Numerical Analysis

uOttawa

Benoit Dionne
University of Ottawa

Adapted version from the notes for the courses MAT3380 and graduate courses in numerical analysis given at the University of Ottawa.

This document is available on the following sites.
uO Research: http://hdl.handle.net/10393/45600
GitHub: https://github.com/BenoitDionne/Numerical_Analysis

Cover Page:
The Rio–Antirio bridge, Greece, photo by Jean Dionne.
The stormy Mediterranean sea, photo by Louise Oegema.

Contents

Preface

This book cover the material normally presented in a two-term course on numerical analysis. It starts with the basic concept normally presented in a first course on numerical analysis. It ends with topics that are more appropriated for a first course in numerical analysis for differential equations.

This book can be used by two different groups of students. If the focus is on the algorithms and the theory is ignored, then the book can be used for an introduction to numerical analysis for engineering and applied science students. The book can also be used as an introduction to numerical analysis for students in mathematics, or students who plan to study more advanced topics in numerical analysis, if the theory is covered. We do not think that there is a need to emphasize the importance of the theory in numerical analysis. No serious progress in numerical analysis is possible without it. Most of the numerical methods presented in this book is accompanied by a code in MATLAB.

The background for this book is a two-term course in linear algebra, a course in real analysis (often called advanced calculus to make the subject less scary), and a course in ordinary differential equation for the last part of the book.

This book is divided into several parts.

After a brief introduction to the arithmetic on computers, the first part on **solving equations** is composed of Chapter 2 on iterative methods to solve nonlinear equations of one unknown variable, Chapter 3 on iterative methods to solve systems of linear equations, Chapter 4 on algebraic methods to solve systems of linear equations, and Chapter 5 on iterative methods to solve system of nonlinear equations.

The second part of the book on **polynomial interpolation** is composed of Chapter 6 on polynomial interpolation of real valued functions and Chapter 7 on spline interpolation; in particular, cubic splines and Bézier curves.

The third part on the **approximation of functions** is composed of three short chapters: Chapter 8 on continuous least square approximation (i.e. in L^2), Chapter 9 on uniform approximation of real valued functions and Chapter 10 on discrete least square approximation (i.e. in ℓ^2).

The fourth part on finding **eigenvalues** of matrices is composed of only one chapter, Chapter 11, on numerical methods to compute eigenvalues of $n \times n$ matrices.

The last part on **differential equations** is composed of Chapter 12 on the numerical differentiation and integration of real valued functions, Chapter 13 on the numerical methods to solve initial value problems for ordinary differential equations, Chapter 14 on the numerical methods to solve boundary value problems for ordinary differential equations, and Chapter 15 on finite difference methods to solve partial differential equations.

There is no chapter on finite element methods to solve partial differential equations. This topic requires some knowledge of functional analysis to be properly covered. To keep the book accessible the undergraduate students (as much as possible), no knowledge of functional analysis is assumed.

The second and third part of the book are related and even intertwine in some cases. There are many ways to approximate a function $f : [a,b] \to \mathbb{R}$ by polynomials. The major approaches are:

1. Given any small ϵ, we could find a polynomial p_ϵ such that
$$\max_{a\leq x\leq b} |f(x) - p_\epsilon(x)| < \epsilon \ .$$
Stone-Weierstrass Theorem, Theorem 9.1.1, states that p_ϵ can always be found if f is a continuous function on $[a,b]$. The function f is **uniformly approximate** by the polynomial p_ϵ. This will be studied in Chapter 9. This very short chapter is more theoretical. Nevertheless, it is important to understand the limitations of polynomial interpolation and splines presented in Chapters 6 and 7.

2. Given any small ϵ, we could find a polynomial p_ϵ such that
$$\int_a^b |f(x) - p_\epsilon(x)|^2 \, \mathrm{d}x < \epsilon \ .$$
The polynomial p_ϵ is a **quadratic approximation** of the function f. This will be studied in Chapter 9. Again, this chapter is more theoretical but essential to understand discrete least square approximation in Chapter 10. This material is also fundamental in the study of numerical analysis; in particular, to develop methods to solve partial differential equations.

3. Given any small ϵ and $a \leq x_0 < x_1 < x_2 < \ldots < x_n \leq b$, we could find a polynomial p_ϵ such that
$$\sum_{i=0}^{n} |f(x_i) - p_\epsilon(x_i)|^2 < \epsilon \ .$$
This is a **discrete least square approximation** of the data set
$$\{(x_i, f(x_i)) : i = 0, 1, 2, \ldots, n\}$$
by a polynomial. This will be the subject of Chapter 10.

4. We could also find a polynomial p of degree at most n such that $p(x_i) = f(x_i)$ for $0 \leq i \leq n$. This is the subject of Chapter 6.

5. Instead of looking for a polynomial of degree n, where n may be large, we could find polynomials p_i of small degrees (usually of degree 3) such that p_i is an approximation of f on the small interval $[x_i, x_{i+1}]$. The polynomials p_i are determined from conditions at the endpoints x_i that provide some degree of smoothness for the piecewise polynomial p defined by $p(x) = p_i(x)$ for $x \in [x_i, x_{i+1}]$. The polynomial p is called a **spline**. We will present the **cubic splines**, **Bézier curves** and **B-Splines** in Sections 7.1, 7.2 and 7.3 of Chapter 7. These piecewise polynomial approximations are superior to the simple polynomial interpolation mentioned in the previous item. Cubic splines, Bézier curves, ... are used in some of the major software for drawing.

There is a strong emphasis in this book on differential equations. This is only a reflect of the principal interest of the author. Contrary to must introductory textbooks in numerical analysis, there is an extensive chapter, Chapter 13, on the numerical methods to solve initial value problems for ordinary differential equations. There is also a full chapter, Chapter 14, on the numerical methods to solve boundary value problems for ordinary differential equations, and a full chapter, Chapter 15, on finite difference methods.

There are many solved exercises at the end of several chapters. Most of the exercises are to reinforce the concepts presented in the text. We have kept the number of theoretical questions to the minimum. This was mainly motivated by the groups of students who took the numerical analysis courses. They were more interested in the applications of numerical analysis than in the theory. Sadly, there are not real life applications of numerical analysis in this book. It would be nice (in the future) to add some realistic projects to illustrate each topics.

The examples should be treated as problems to be solved by the reader. The reader should try to answer each problem before looking at its solution (if it is available).

In this book, we use the following notation for some standard sets of numbers.

Definition

The following well known sets are frequently used in this document.

- $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ is the set of natural numbers.
- $\mathbb{N}^+ = \{1, 2, 3, \ldots\}$ is the set of positive natural numbers.
- $\mathbb{Z} = \{0, 1, -1, 2, -2, 3, -3, \ldots\}$ is the set of integers.
- $\mathbb{Q}$ is the set of rational numbers.
- $\mathbb{R}$ is the set of real numbers.
- $\mathbb{C}$ is the set of complex numbers.

We will also often use the following definition when approximating functions.

Definition

Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ be two functions. We write $f(\mathbf{x}) = O(g(\mathbf{x}))$ near the origin if there exists a positive constant K such that

$$|f(\mathbf{x})| < K\,|g(\mathbf{x})|$$

for $\mathbf{x}$ in a neighbourhood of the origin. We write $f(\mathbf{x}) = o(g(\mathbf{x}))$ near the origin if

$$\lim_{\mathbf{x}\to\mathbf{0}} \frac{f(\mathbf{x})}{g(\mathbf{x})} = 0 \ .$$

Chapter 1

Computer Arithmetic

Before studying algorithms to perform computations with computers, we need to understand how computers perform basic arithmetic operations. It is the goal of this chapter.

1.1 Rounding

Definition 1.1.1

The **normalized scientific notation** for a real number is $\pm 0.d_1 d_2 d_3 \ldots \times 10^m$, where m is an integer, $d_i \in \{0, 1, 2, 3, \ldots, 9\}$ and $d_1 \neq 0$.

Before performing any arithmetic operation with real numbers, we will always assume that they have been expressed in the normalized scientific notation.

When performing arithmetic operations by hand, we often have to consider only the first few decimals (digits after the period) of the numbers used in the operations and ignore the others. This is called **rounding**.

There are different ways to perform rounding. We will mention only two.

Definition 1.1.2

Let $\pm 0.d_1 d_2 \cdots \times 10^N$ be the normalized scientific representation of a real number a, thus $d_1 \neq 0$. For k a positive integer, we define the
k-digit chopping representation of a to be $\pm 0.d_1 d_2 \ldots d_k \times 10^N$, and the
k-digit rounding representation of a to be $\pm 0.d_1 d_2 \ldots d_k \times 10^N + \epsilon\, 10^{-k} \times 10^N$, where $\epsilon = 1$ for $d_{k+1} \geq 5$ and $\epsilon = 0$ for $d_{k+1} < 5$.

If $\tilde{a}$ is the k-digit chopping representation of a, then $|a - \tilde{a}| < 10^{-k} \times 10^N$. If $\tilde{a}$ is k-digit rounding representation of a, then $|a - \tilde{a}| \leq 0.5 \times 10^{-k} \times 10^N$.

Example 1.1.3

Here are some examples of 3-digit rounding representations.

exact value	3-digit rounding approximation
0.19234542×10^6	0.192×10^6
$0.25952100 \times 10^{-5}$	0.260×10^{-5}
0.99950000×10^2	0.100×10^3

♣

Example 1.1.4
If 0.481×10 is a 3-digit rounding approximation of x and 0.12752×10^2 is a 5-digit rounding approximation of y, find the interval that will contain the exact value of $x - y$.

Since $4.805 \leq x < 4.815$ and $12.7515 \leq y < 12.7525$, then $4.805 - 12.7525 < x - y < 4.815 - 12.7515$. Thus $-7.9475 < x - y < -7.9365$. ♣

1.2 Binary Number

Computers only manipulate binary numbers (i.e. numbers in base 2),

Recall that a number in base 2 is a number of the form

$$(b_k b_{k-1} \ldots b_1 b_0 . b_{-1} b_{-2} \ldots)_2 = b_k 2^k + b_{k-1} 2^{k-1} + \ldots + b_1 2 + b_0 + b_{-1} 2^{-1} + b_{-2} 2^{-2} + \ldots$$

where $b_i \in \{0, 1\}$ for all i.

Definition 1.2.1

The **normalized binary numbers** are numbers of the form $\pm(0.b_1 b_2 b_3 \ldots)_2 \times 2^m$, where $b_i \in \{0, 1\}$, $b_1 = 1$ and m is an integer often represented in binary form. Binary numbers in normalized binary form are also said to be in **normalized floating point form**.

To find the binary representation of a positive number x in base 10, one begins by writing x as $x = m + d$, where m is an integer and $d < 1$.

If

$$m = m_j \times 10^j + m_{j-1} \times 10^{j-1} + \ldots + m_1 \times 10 + m_0 ,$$

then

$$(m)_2 = (m_j)_2 \times (10)_2^j + (m_{j-1})_2 \times (10)_2^{j-1} + \ldots + (m_1)_2 \times (10)_2 + (m_0)_2 .$$

The easiest way to evaluate this expression is recursively.

$$\alpha_0 = (m_j)_2$$
$$\alpha_1 = \alpha_0 \times (10)_2 + (m_{j-1})_2$$

$$\begin{aligned}
\alpha_2 &= \alpha_1 \times (10)_2 + (m_{j-2})_2 \\
\vdots \quad & \quad \vdots \\
\alpha_{j-2} &= \alpha_{j-3} \times (10)_2 + (m_2)_2 \\
\alpha_{j-1} &= \alpha_{j-2} \times (10)_2 + (m_1)_2 \\
\alpha_j &= \alpha_{j-1} \times (10)_2 + (m_0)_2
\end{aligned}$$

and $(m)_2 = \alpha_j$.

Let

$$d = d_1 \times 2^{-1} + d_2 \times 2^{-2} + d_3 \times 2^{-3} + \ldots + d_k \times 2^{-k} \ .$$

The first digit d_1 is the integer part of

$$\begin{aligned}
r_1 = 2d &= 2 \times \left(d_1 \times 2^{-1} + d_2 \times 2^{-2} + \ldots + d_{1-k} \times 2^{1-k} + d_k \times 2^{-k}\right) \\
&= d_1 + d_2 \times 2^{-1} + \ldots + d_{k-1} \times 2^{2-k} + d_k \times 2^{1-k} \ .
\end{aligned}$$

The second digit d_2 is the integer part of

$$\begin{aligned}
r_2 &= 2^2(d - d_1 \times 2^{-1}) = 2^2 \times \left(d_2 \times 2^{-2} + d_3 \times 2^{-3} + \ldots + d_{k-1} \times 2^{1-k} + d_k \times 2^{-k}\right) \\
&= d_2 + d_3 \times 2^{-1} + \ldots + d_{k-1} \times 2^{3-k} + d_k \times 2^{2-k} \ . \qquad (1.2.1)
\end{aligned}$$

The third digit d_3 is the integer part of

$$\begin{aligned}
r_3 &= 2^3(d - d_1 \times 2^{-1} - d_2 \times 2^{-2}) \\
&= 2^3 \times \left(d_3 \times 2^{-3} + d_4 \times 2^{-4} + \ldots + d_{k-1} \times 2^{1-k} + d_k \times 2^{-k}\right) \\
&= d_3 + d_4 \times 2^{-1} + \ldots + d_{k-1} \times 2^{4-k} + d_k \times 2^{3-k} \ . \qquad (1.2.2)
\end{aligned}$$

In general, we get that the i^{th} digit d_i is the integer part of

$$r_i = 2^i \left(d - d_1 \times 2^{-1} - d_2 \times 2^{-2} - \ldots - d_{i-1} \times 2^{1-i}\right) \qquad (1.2.3)$$

for $i = 2, 3, 4, \ldots. \ k$.

We however need a more efficient way to find the digits d_i. We have from (1.2.1) that

$$r_2 = 2(2d - d_1) = 2(r_1 - d_1) \ . \qquad (1.2.4)$$

We have from (1.2.2) that

$$r_3 = 2(\ 2(2d - d_1) - d_2) = 2(r_2 - d_2) \ .$$

We prove by induction that

$$r_{i+1} = 2(r_i - d_i) \qquad (1.2.5)$$

for $i = 1, 2, \ldots, k-1$. It follows from (1.2.4) that (1.2.5) is true for $i = 1$. Let's suppose that (1.2.5) is true for $i = j < k-1$. We have from (1.2.3) with $i = j+2$ that

$$r_{j+2} = 2^{j+2}\left(d - d_1 2^{-1} - d_2 2^{-2} - \ldots - d_j 2^{-j} - d_{j+1} 2^{-j-1}\right)$$

$$= 2\left(\underbrace{2^{j-1}\left(d - d_1 2^{-1} - d_2 2^{-2} - \ldots d_j 2^{-j}\right)}_{=r_{j+1} \text{ from (1.2.3) with } i=j+1} + d_{j+1} \right)$$
$$= 2\left(r_{j+1} - d_{j+1}\right) \ .$$

This is (1.2.5) with $i = j + 1$. This complete the proof by induction.

Example 1.2.2
The binary representation of 1/10 is $(0.0\overline{0011})_2$.

Let $(1/10)_2 = (0.d_1 d_2 d_3 \ldots)_2$. We summarize in the table below the computation using $r_1 = 2d$ and $r_{i+1} = 2(r_i - d_i)$ for $i = 1, 2, \ldots$

i	r_i	d_i (the integer part of r_i)
1	$2 \times 1/10 = 1/5$	0
2	$2(r_1 - 0) = 2/5$	0
3	$2(r_2 - 0) = 4/5$	0
4	$2(r_3 - 0) = 8/5 = 1.6$	1
5	$2(r_4 - 1) = 6/5 = 1.2$	1
6	$2(r_5 - 1) = 2/5$	0
$\vdots$	$\vdots$	$\vdots$

Since $r_6 = r_2$ and $d_6 = d_2$, we get that $d_7 = d_3$, $d_8 = d_4$, $d_9 = d_5$ and, in general, $d_i = d_{i-4}$ for $i = 6, 7, \ldots$. ♣

1.3 Computer Numbers

To illustrate the properties of computer arithmetic, we assume that each real number is stored in a 32-bit word. The typical computer representation of a normalized binary number $x = \pm(0.b_1 b_2 b_3 \ldots)_2 \times 2^m$ is given by

$$\boxed{s} \boxed{e_8 e_7 e_6 \ldots e_1} \boxed{b_2 b_3 \ldots b_{24}} \ ,$$

where s indicates the sign of x, $(e_8 e_7 \ldots e_1)_2 = (m)_2 + (1111111)_2$, and $b_1, b_2, b_3, \ldots b_{24}$ are the first 24 binary digits of the normalized representation of x. The part $(b_1 b_3 \ldots b_{24})_2$ is called the **normalized mantissa**.

Remark 1.3.1

1. We did not store the value of b_1 because we always assume that the binary numbers are normalized and so b_1 is always 1.

2. Let e be the decimal representation of the number $(e_8 e_7 \ldots e_1)_2$. Then $0 \leq e < 2^8 = 256$ but, in practice, only $1 \leq e \leq 254$ is used because the values 0 and 255 are often reserved to indicate really small or large numbers, and **NaN** (not a number). We get NaN following an illegal operation like a division by zero.

   To represent negative exponents, we assume that $e = m + 127$. Thus, $-126 \leq m \leq 127$. In binary notation $(m)_2 = (e_8 e_7 \ldots e_1)_2 - (1111111)_2$.

3. 0 has its unique computer representation (associated to $e = 0$ or 255).

♠

The computer representation of a real number x is called the **floating point representation** of x and is denoted by $\mathrm{fl}(x)$. The difference between a real number and its computer representation is called the **rounding error**.

There are major differences between the standard arithmetic and the computer arithmetic. We mention some below.

- Not all real numbers can be represented as computer numbers. There are "holes" in the computer representation of the real line. For instance, the binary representation of 1/10 is $(0.\overline{1100})_2 \times 2^{-(11)_2}$. Hence, the machine representation of this number is

  $$\boxed{0 \mid 1111100 \mid 10011001100110011001100} \; .$$

  This machine number represents in fact the number

  $$\big(2^{-1} + 2^{-2} + 2^{-5} + 2^{-6} + 2^{-9} + 2^{-10} + 2^{-13} + 2^{-14} + 2^{-17} + 2^{-18} + 2^{-21} + 2^{-22}\big)2^{-3} = 0.09999999403954\ldots$$

- Not all real numbers can be represented as computer numbers. There are upper and lower bounds to the real numbers that can be represented on a computer. The largest real number that can be represented as computer number is

  $$R_M = (0.\underbrace{1111\ldots1}_{24 \text{ times}})_2 \times 2^{127} = (1 - 2^{-24}) \times 2^{127} \approx 0.17014117\ldots \times 10^{38}$$

  and the smallest positive number is

  $$R_m = (0.1\underbrace{000\ldots0}_{23 \text{ times}})_2 \times 2^{-126} = 2^{-127} \approx 0.587747\ldots \times 10^{-38} \; .$$

  If the result of a computation is a number bigger than R_M, then we say that we have **overflow**. If the result of a computation is a number smaller than R_m, than we have **underflow**.

- The fundamental algebraic properties of the real number system (commutativity, associativity, …) are not preserved.

  Suppose that the basic computer operations ($+$, $-$, $\times$, $\div$) are defined as follows.

exact operation	computer operation
$x \pm y$	$\mathrm{fl}(\mathrm{fl}(x) \pm \mathrm{fl}(y))$
$x \times y$	$\mathrm{fl}(\mathrm{fl}(x) \times \mathrm{fl}(y))$
$x \div y$	$\mathrm{fl}(\mathrm{fl}(x) \div \mathrm{fl}(y))$

  We also define the computer operation $\mathrm{fl}(\sqrt{\mathrm{fl}(x)})$ to represent the exact operation $\sqrt{x}$. This is not exactly how computers work with computer numbers but it is an acceptable definition to understand why the fundamental algebraic properties of the real number system are not preserved.

If we work in base 10 using 4-digit rounding representations, the computer evaluation of $\pi + (1/3) \times \pi$ is given by

$$\begin{aligned} \mathrm{fl}(\mathrm{fl}(\pi) + \mathrm{fl}(\mathrm{fl}(1/3) \times \mathrm{fl}(\pi))) &= \mathrm{fl}((0.3142 \times 10) + \mathrm{fl}(0.3333 \times (0.3142 \times 10))) \\ &= \mathrm{fl}((0.3142 \times 10) + \mathrm{fl}(1.0472286000000)) \\ &= \mathrm{fl}((0.3142 \times 10) + (0.1047 \times 10)) \\ &= \mathrm{fl}(4.189) = 0.4189 \times 10 \end{aligned}$$

The computer evaluation of $\pi \times (1 + (1/3)) = \pi + (1/3) \times \pi$ is given by

$$\begin{aligned} \mathrm{fl}(\mathrm{fl}(\pi) \times \mathrm{fl}(\mathrm{fl}(1) + \mathrm{fl}(1/3))) &= \mathrm{fl}((0.3142 \times 10) \times \mathrm{fl}((0.1 \times 10) + (0.3333))) \\ &= \mathrm{fl}((0.3142 \times 10) \times \mathrm{fl}(1.3333)) \\ &= \mathrm{fl}((0.3142 \times 10) \times (0.1333 \times 10)) \\ &= \mathrm{fl}(4.188286) = 0.4188 \times 10 \end{aligned}$$

Thus, we do not get the same 4-digit rounding representation for $\pi + (1/3) \times \pi$ and $\pi \times (1 + (1/3))$. The distributive law is not preserved.

Suppose that p is the exact result of a computation and $\tilde{p}$ is the computer result of this computation. The number $\epsilon = |p - \tilde{p}|$ is the **absolute error**. If $p \neq 0$, the number $\epsilon_r = |p - \tilde{p}|/|p| = \epsilon/|p|$ is the **relative error**.

If the absolute error is 0.1, where the numbers p and $\tilde{p}$ are smaller than 1 in absolute value, then the error is enormous. However, when the numbers p and $\tilde{p}$ are larger than 10^6 in absolute value, the same absolute error is very small. The absolute error by itself does not say anything about the accuracy of the computation. The relative error is the useful information about the size of the error.

Example 1.3.2
22/7 and 315/113 are two frequently used approximations of π. We find the absolute and relative errors of these two approximations of π. The absolute and relative error of the approximation 22/7 of $\pi = 3.14159265358979\ldots$ are

$$|3.14159265358979\ldots - 22/7| = 0.126442\ldots \times 10^{-2}$$

and

$$|3.14159265358979\ldots - 22/7|/3.14159265358979\ldots = 0.4024994\ldots \times 10^{-3}$$

respectively. A relative error of about 0.04 %. The absolute and relative error of the approximation 355/113 of π are

$$|3.14159265358979\ldots - 355/113| = 0.2668\ldots \times 10^{-6}$$

and

$$|3.14159265358979\ldots - 355/113|/3.14159265358979\ldots = 0.84914\ldots \times 10^{-7}$$

respectively. A relative error of about 0.0000085 %. ♣

Remark 1.3.3

For our 32-bit computer, if $x = (0.b_1b_2b_3\ldots)_2 \times 2^m$, we have that $\dfrac{|x - \mathrm{fl}(x)|}{|x|} \leq 2^{-24}$ if rounding is used and $\dfrac{|x - \mathrm{fl}(x)|}{|x|} \leq 2^{-23}$ if chopping is used.

i) We prove that $\dfrac{|x - \mathrm{fl}(x)|}{|x|} \leq 2^{-24}$ if rounding is used. The number x is between the computer numbers $x_1 = (0.b_1b_2b_3\ldots b_{24})_2 \times 2^m$ and $x_2 = ((0.b_1b_2b_3\ldots b_{24})_2 + 2^{-24}) \times 2^m$. Hence $\mathrm{fl}(x) = x_1$ if $b_{25} = 0$ and $\mathrm{fl}(x) = x_2$ if $b_{25} = 1$.

If $b_{25} = 0$, then

$$|x - \mathrm{fl}(x)| = |x - x_1| = (0.b_{26}b_{27}\ldots)_2 \times 2^{m-25} \leq 2^{m-25}$$

and

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq \frac{2^{m-25}}{(0.b_1b_2b_3\ldots)_2 \times 2^m} = \frac{1}{(0.b_1b_2b_3\ldots)_2} 2^{-25} \leq 2^{-24}$$

because $(0.b_1b_2b_3\ldots)_2 \geq (0.b_1)_2 = (0.1)_2 = 2^{-1}$.

If $b_{25} = 1$, then

$$|x - \mathrm{fl}(x)| = |x - x_2| = ((1)_2 - (0.b_{25}b_{26}\ldots)_2) \times 2^{m-24} \leq 2^{m-25}$$

because $(1)_2 - (0.b_{25}b_{26}\ldots)_2 \leq 2^{-1}$. Thus

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq \frac{2^{m-25}}{(0.b_1b_2b_3\ldots)_2 \times 2^m} = \frac{1}{(0.b_1b_2b_3\ldots)_2} 2^{-25} < 2^{-24}$$

because $(0.b_1b_2b_3\ldots)_2 \geq (0.b_1)_2 = (0.1)_2 = 2^{-1}$.

ii) To prove that $\dfrac{|x - \mathrm{fl}(x)|}{|x|} \leq 2^{-23}$ if chopping is used. We note that

$$\mathrm{fl}(x) = (0.b_1b_2b_3\ldots b_{24})_2 \times 2^m$$

and

$$|x - \mathrm{fl}(x)| = (0.b_{25}b_{26}b_{27}\ldots)_2 \times 2^{m-24} < 2^{m-24}\ .$$

We do not exclude the possibility that some or all of b_{25}, b_{26}, ... be zero. Thus

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq \frac{2^{m-24}}{(0.b_1b_2b_3\ldots)_2 \times 2^m} = \frac{1}{(0.b_1b_2b_3\ldots)_2} 2^{-24} < 2^{-23}$$

because $(0.b_1b_2b_3\ldots)_2 \geq (0.b_1)_1 = (0.1)_2 = 2^{-1}$. ♠

Definition 1.3.4

Let r be a positive integer. We say that $\tilde{p}$ approximates p to r **significant digits** if

$$|p - \tilde{p}| \leq \frac{1}{2}\beta^{s-r+1} ,$$

where β is the basis used to represent the numbers and s is the largest integer such that $\beta^s \leq |p|$.

For instance, if the basis is $\beta = 10$, then $\tilde{p}$ approximate p to r significant digits if

$$|p - \tilde{p}| \leq \frac{1}{2}\left(10^{s-r+1}\right) = 5 \times 10^{s-r} ,$$

where s is the largest integer such that $10^s \leq |p|$. Thus

$$\frac{|p - \tilde{p}|}{|p|} \leq \frac{|p - \tilde{p}|}{10^s} \leq 5 \times 10^{-r} .$$

The largest positive integer r such that the previous inequality is satisfied is the classical definition of r significant digits.

Example 1.3.5

Both 10.001 and 9.999 approximate 10 to 4 significant digits because the relative error

$$\epsilon_r = \frac{|10.001 - 10|}{10} = \frac{|10 - 9.999|}{10} = 10^{-4} < 5 \times 10^{-4}$$

and 4 is the largest integer r such that $\epsilon_r < 5 \times 10^{-r}$. ♣

1.4 Controlling Errors

From now on and until the end of this chapter, our presentation will be more intuitive. We will not always be mathematically rigorous. Our goal is to help the readers develop their intuition on how to improve the accuracy of numerical computations. This is often referred as the Art of numerical computation.

There are many causes for the loss of accuracy in computations.

1. Loss of accuracy often comes from the cancellation of significant digits due to subtraction of nearly equal numbers.

   Let $x = 5/7 = 0.\overline{714285}$ and $y = 0.714251$. Using 5-digit chopping arithmetic, we get

	Exact values	5-digit chopping arithmetic	absolute error (approx.)	relative error (approx.)	number of significant digits
x	$0.\overline{714285}$	0.71428	0.6×10^{-5}	0.8×10^{-5}	5
y	0.714251	0.71425	0.1×10^{-5}	0.14×10^{-5}	6
$x - y$	$0.34\overline{714285} \times 10^{-4}$	0.3×10^{-4}	0.47×10^{-5}	0.136	1

We have lost a lot of significant digits in the subtraction $x - y$.

2. The rounding error of a computer number is amplified when this number is multiply by a number of large absolute value or divide by a number of small absolute value.

3. A really small number should not be added to a very large number. Let $x = 0.1234 \times 10^5$ and $y = 0.4321$. Using 4-digit rounding arithmetic to add this two numbers, we get $x + y = x$ because $y = 0.000004321$ and so $x + y$ is 0.123404321×10^5. Rounding this number to 4-digits gives $x = 0.1234 \times 10^5$.

When possible, rearranging the order of the arithmetic operations may increase the accuracy of the computation. The following three examples illustrate this technique.

Example 1.4.1
Use 6-digit rounding arithmetic to compute the roots of the polynomial $x^2 - 20\,x + 1 = 0$.

The standard formulae to compute the roots of the polynomial of degree two $ax^2+bx+c = 0$ are

$$x_+ = \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a} \quad \text{and} \quad x_- = \frac{-b - \sqrt{b^2 - 4\,a\,c}}{2\,a} \ . \tag{1.4.1}$$

We get

$$x_+ = \frac{20 + \sqrt{396}}{2} \approx \frac{20 + 19.8997}{2} \approx 19.9499 \ .$$

Since the exact value of this root is $\alpha = 19.9498743710661995\ldots$, the relative error is

$$\frac{19.9499 - \alpha}{\alpha} \approx 0.128 \times 10^{-5} \ .$$

The second root is

$$x_- = \frac{20 - \sqrt{396}}{2} \approx \frac{20 - 19.8997}{2} = \frac{0.1003}{2} = 0.05015 \ .$$

Since the exact value of this root is $\beta = 0.050125628933800\ldots$, the relative error is

$$\frac{0.05015 - \beta}{\beta} =\approx 0.486 \times 10^{-3} \ .$$

This is not really good for 6-digit rounding.

If $c \neq 0$, the roots of the polynomial $ax^2 + bx + c = 0$ are also given by the formulae

$$x_+ = \frac{-2c}{b + \sqrt{b^2 - 4\,a\,c}} \quad \text{and} \quad x_- = \frac{2c}{-b + \sqrt{b^2 - 4\,a\,c}} \ . \tag{1.4.2}$$

Multiply the formula for x_+ in (1.4.1) by $\dfrac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}$ and the formula for x_- in (1.4.1) by $\dfrac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}}$ to get the formulae in (1.4.2).

We get

$$x_- = \frac{2}{20+\sqrt{396}} \approx \frac{2}{20+19.8997} = \frac{2}{39.8997} \approx 0.0501257 \ .$$

The relative error is now

$$\frac{0.0501257-\beta}{\beta} = 0.142 \times 10^{-5} \ .$$

This is good. This is a significant improvement on the previous computation of x_-.

The idea is to avoid the subtraction of almost equal numbers. In the formula for x_- in (1.4.1), we had to compute $20 - 19.8997$ which is the difference of two very close numbers. In the formula for x_- in (1.4.2), we did not have to subtract two very close numbers. This is the reason why, for the polynomial $x^2 - 20\,x + 1 = 0$, the second formula to compute x_- is better than the first one. ♣

Example 1.4.2
Compute

$$f(x) = x^3 - 6x^2 + 3x - 0.149 \tag{1.4.3}$$

at $x = 4.71$ using 3-digit rounding arithmetic.

A direct computation using (1.4.3) and 3-digit rounding arithmetic gives $f(x) = -0.140 \times 10^2$. Using the fact that $f(x) = -14.636489$, we find that the absolute error is 0.636489, the relative error is about 0.04, and the approximation is to 2 significant digits.

A better way to write $f(x)$ is to use the nested form

$$f(x) = -0.149 + x(3 + x(x-6)) \ . \tag{1.4.4}$$

Using (1.4.4) and 3-digit rounding arithmetic, we get $f(x) = -0.146 \times 10^2$. The absolute error is 0.36489×10^{-1}, the relative error is about 0.25×10^{-2}, and the approximation is to 3 significant digits.

The nested form must always be used to evaluate a polynomial because less arithmetic operations are generally involved. For instance, 5 multiplications and 3 additions / subtractions are involved in (1.4.3) while only 2 multiplications and 3 additions / subtractions are involved in (1.4.4). ♣

Example 1.4.3
Using 4-digit chopping arithmetic, add the following numbers in increasing order (from the smallest to the largest) and in decreasing order (from the largest to the smallest).

$$\begin{gathered} x_1 = 0.1580 \ , \ x_2 = 0.2653 \ , \ x_3 = 0.2581 \times 10 \ , \ x_4 = 0.4288 \times 10 \ , \ x_5 = 0.6266 \times 10^2 \ , \\ x_6 = 0.7555 \times 10^2 \ , \ x_7 = 0.7767 \times 10^3 \ , \ x_8 = 0.7889 \times 10^3 \text{ and } x_9 = 0.8999 \times 10^4 \ . \end{gathered}$$

The exact value of the sum is 0.107101023×10^5.

	4-digit chopping arithmetic	absolute error (approx.)	relative error (approx.)	number of significant digits
increasing	0.1071×10^5	0.1023	0.96×10^{-5}	5
decreasing	0.1069×10^5	20.1	0.19×10^{-2}	3

The numbers x_1, x_2, x_3 and x_4 are ignored when the summation is performed in decreasing order. This is another example where adding a really small number to a very large number produces a loss of accuracy. ♣

1.5 Stability

The numerical solution of many problems is approximated by the solution of a difference equation. For instance, the Euler's method, that is taught in calculus and that we will study again later, states that the solution of the difference equation

$$\begin{aligned} w_{j+1} &= w_j + hf(x_j, w_j) \qquad \text{for} \quad j = 0, 1, 2, \ldots \\ w_0 &= y_0 \end{aligned}$$

provides an approximation to the solution of the differential equation $y' = f(x, y)$ with $y(0) = y_0$. Namely, $y(x_j) \approx w_i$ for $j = 0, 1, 2, \ldots$ The x_j's are the **mesh points** defined by $x_j = x_0 + jh$ for $j \geq 0$, where h is the chosen **step size**.

Suppose that the solution of a problem is approximated by the solution of the difference equation

$$x_{n+1} = \frac{10}{21}x_n - \frac{1}{21}x_{n-1} \tag{1.5.1}$$

with the initial conditions $x_0 = 1$ and $x_1 = 1/3$. Using (1.5.1) recursively, we find

n	x_n
x_2	$0.11111111111111\ldots$
x_3	$0.03703703703703\ldots$
$\vdots$	$\vdots$
x_{10}	$0.000016935087808430\ldots$
$\vdots$	$\vdots$
x_{21}	$0.95599066359747\ldots \times 10^{-10}$
$\vdots$	$\vdots$

The exact solution of (1.5.1) is $x_j = (1/3)^j$ for $j = 0, 1, 2, \ldots$ The previous values computed recursively are exact to all written digits.

However, the solution of another problem may be approximated by the solution of the difference equation

$$x_{n+1} = \frac{16}{3}x_n - \frac{5}{3}x_{n-1} \tag{1.5.2}$$

with the initial conditions $x_0 = 1$ and $x_1 = 1/3$. Using (1.5.2) recursively, we find

n	x_n
x_2	$0.11111111111111\ldots$
x_3	$0.03703703703703\ldots$
x_4	$0.01234567901234\ldots$

x_5	$0.00411522633742\ldots$
$\vdots$	$\vdots$
x_{10}	$0.00001693501310\ldots$
$\vdots$	$\vdots$
x_{20}	$-0.00072952204841\ldots$
$\vdots$	$\vdots$
x_{40}	$-0.69572671433304\ldots \times 10^{11}$
$\vdots$	$\vdots$

The exact solution of (1.5.1) is $x_j = (1/3)^j$ for $j = 0, 1, 2, \ldots$ For $j = 2$ and 3, the x_j's are exact to all written digits. However, starting with $j = 14$, there is a growing difference between the exact solution and the computed solution. In fact, the computed solution seems to converge to $-\infty$.

Why can we compute the solution for (1.5.1) but not the solution for (1.5.2)? The general solution of (1.5.1) is of the form

$$x_j = A\left(\frac{1}{3}\right)^j + B\left(\frac{1}{7}\right)^j .$$

The particular solution with $x_0 = 1$ and $x_1 = 1/3$ is given by $A = 1$ and $B = 0$. Numerical rounding has an effect similar to slightly changing (a little perturbation of) the values of A and B. Since $(1/7)^j$ converge to 0 faster than $(1/3)^j$ as $j \to \infty$, the second term of the general solution has little or no significant effect on the compute value of x_j.

However, the general solution of (1.5.1) is of the form

$$x_j = A\left(\frac{1}{3}\right)^j + B4^j .$$

The particular solution for $x_0 = 1$ and $x_1 = 1/3$ is given by $A = 1$ and $B = 0$. Again, numerical rounding has an effect similar to slightly changing (a little perturbation of) the values of A and B. Since 4^j converges to ∞ while $(1/3)^j$ converges to 0 as $j \to \infty$, the term $B4^j$ of the general solution will dominate the computation of x_j as $j \to \infty$ even if B is really small.

We say that a numerical method behaving like (1.5.1) is **stable** and a numerical method behaving like (1.5.2) is **unstable**. We will come back on these concepts several times in the next chapters; in particular in Chapters 13 and 14.

1.6 Conditioning

Will a small perturbation in the data of a numerical process produce a small change or a large change in the result of this numerical process? This type of questions is part of what is called **conditioning**.

We say that a numerical process is **well conditioned** if a small perturbation in the data of this numerical process produces a small change in the result of this numerical process. We say that a numerical process is **ill conditioned** if a small perturbation in the data of this numerical process produces a large change in the result of this numerical process.

A simple example of conditioning is provided by the numerical evaluation of a function. Due to rounding errors (in particular to the rounding error associated to the argument), the numerical evaluation of a function f at x is equal to the exact value of f evaluated at $x+h$, where the perturbation h is small. If, for h small, the exact value $f(x+h)$ is close to the exact value $f(x)$, then we say that the numerical evaluation of f at x is **well conditioned**. Otherwise, we say that the numerical evaluation of f at x is **ill conditioned**.

To give a mathematical meaning to well conditioned and ill conditioned in the context of the evaluation of f at x, we use the Taylor expansion[1] of f at x,

$$f(x+h) = f(x) + f'(x)h + \frac{f''(\zeta)}{2}h^2 ,$$

where $x < \zeta < x+h$. Hence

$$\frac{f(x+h)-f(x)}{f(x)} = \frac{f'(x)}{f(x)}\,h + \frac{f''(\zeta)}{2f(x)}\,h^2 = \left(\frac{xf'(x)}{f(x)}\right)\left(\frac{h}{x}\right) + \frac{f''(\zeta)}{2f(x)}\,h^2 .$$

If h is small enough, we may ignore the term $(f''(\zeta)h^2)/(2f(x))$ because h^2 goes to 0 faster than h. Hence,

$$\frac{f(x+h)-f(x)}{f(x)} \approx \left(\frac{xf'(x)}{f(x)}\right)\left(\frac{h}{x}\right)$$

for h small enough. The relative error of $f(x+h)$ (i.e. the numerical evaluation of a function f at x) is asymptotically proportional to the relative size of the perturbation h with the constant of proportionality

$$\frac{xf'(x)}{f(x)} .$$

This constant is called the **condition number** for the evaluation of the function f at x. This condition number will depend on the function f chosen and the argument x used. If the condition number is large in absolute value, then we say that the evaluation of f at x is ill conditioned. If the condition number is small in absolute value, then we say that the evaluation of f at x is well conditioned.

Example 1.6.1
Is evaluating $f(x) = \tan(x)$ near $x = \pi/2$ well or ill conditioned?

The condition number is

$$\frac{xf'(x)}{f(x)} = \frac{x\sec^2(x)}{\tan(x)} = \frac{x}{\sin(x)\cos(x)} .$$

Since

$$\lim_{x\to\pi/2} \frac{x}{\sin(x)\cos(x)} = +\infty ,$$

[1]See Theorem 2.1.6 in the next chapter.

the conditional number is very large for x near $\pi/2$ and the evaluation of f at x near $\pi/2$ is ill conditioned. ♣

There is also a condition number associated to the numerical process of solving linear systems of equation. This condition number will be defined in the chapter on the algorithms to numerically solve linear systems of equations.

1.7 Exercises

Question 1.1
Compute $\left(\dfrac{1}{3} - \dfrac{3}{11}\right) + \dfrac{3}{20}$ using 3-digit chopping arithmetic and 3-digit rounding arithmetic. Compare the relative error of both computations.

Question 1.2
Using 3-digit chopping arithmetic, compute $\displaystyle\sum_{i=1}^{10} \frac{1}{i^2}$ in ascending and decreasing order. Compute the relative error for each method. Which method is more accurate and why it is so?

Question 1.3
We know that $e = \displaystyle\sum_{n=0}^{\infty} \frac{1}{n!}$. Using 4-digit rounding arithmetic, compute the approximation $\displaystyle\sum_{n=0}^{5} \frac{1}{n!}$ of e using the best method to compute the sum. Compute the absolute error, the relative error and the number of significant digits.

Question 1.4
Assuming that 10-digit rounding arithmetic is used, how many digits of accuracy are lost in the subtraction $1 - \cos(0.25)$?

Question 1.5
If 0.2235 is a 4-digit rounding approximation of x and 0.32145 is a 5-digit rounding approximation of y, find a small interval that will contain x/y.

Question 1.6
If x is an approximation of π with four significant digits, find a small interval that will contain x.

Question 1.7
a) Give the best algebraic formula (the formula with the lowest risk to lose significant digits) to approximate the smallest root x_- of the polynomial $p(x) = x^2 - 235x + 3$. Justify your choice of formula.
b) Using 4-digit rounding arithmetic and the formula that you have given in (a), compute an approximation of x_-. Show all the steps of your computation.
c) The exact value of x_- is $0.012766651010\ldots$. Compute the absolute error, the relative error and the number of significant digits for your approximation in (b).

Question 1.8
What can go wrong with the operation $\sqrt{x^2 + y^2}$ for very large values of x and y. How can you avoid such problem?

Question 1.9
Why is there a loss of significant digits when computing $\ln(1 + x) - \ln(x)$ for x large? How can we rewrite $\ln(1 + x) - \ln(x)$ to avoid this loss of significant digits?

Question 1.10
Transform the expression $1 - \cos(x)$ to an equivalent expression which can be computed "accurately" for small values of x.

Question 1.11
Find a way to compute $f(x) = \sqrt{x^4 + 4} - 2$ for x small that will minimize the loss of significant digits.

Question 1.12
In 1994, a flaw was found on the Intel Pentium computer chip related to the division of large integers. The following results were obtained.

division	$\tilde{x}$: the value obtained with the Intel computer chip	x: the exact value
$\dfrac{5505001}{294911}$	18.66600092909	18.6666519729681...
$\dfrac{4.999999}{14.999999}$	0.333329	0.3333332888888...
$\dfrac{41.95835}{31.45727}$	1.33382	1.33382044913624...

Find the absolute error, relative error and number of significant digits for the values obtained with the Intel computer chip.

Question 1.13
Show that the recurrence relation (i.e. the difference equation)

$$x_n = 2x_{n-1} + x_{n-2} \tag{1.7.1}$$

has a general solution of the form

$$x_n = \alpha_1 \lambda_1^n + \alpha_2 \lambda_2^n$$

for $n = 0, 1, 2, \ldots$ Can we safely use the recurrence relation to compute the values of x_n given initial values x_0 and x_1?

Chapter 2

Iterative Methods to Solve Nonlinear Equations

The classical problem is to find the solutions of the equation

$$f(x) = 0 \ , \tag{2.0.1}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a given function. Namely, the goal is to find the numbers p such that $f(p) = 0$. The numbers p are called the **roots** or **zeros** of f.

2.1 Real Analysis Background

We present some of the well know results in real analysis that will be used to justify the numerical methods presented in this book.

Theorem 2.1.1

If $\{x_n\}_{n=0}^{\infty}$ is a bounded and increasing sequence of $\mathbb{R}$, then it converges to $M = \sup\{x_n : n \geq 0\} \in \mathbb{R}$.

Theorem 2.1.2 (Intermediate Value Theorem)

Let $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$ be a continuous function. If α is between $f(a)$ and $f(b)$ (α may be $f(a)$ or $f(b)$), then there exists c between a and b (c may be a or b) such that $f(c) = \alpha$.

Corollary 2.1.3

Let $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$ be a continuous function. If $f(a)\, f(b) < 0$, then there exists a zero of f in the interval $]a, b[$.

Proof.
Since $f(a)$ and $f(b)$ are of opposite sign, 0 is between $f(a)$ and $f(b)$. By the previous theorem with $\alpha = 0$, there exists c between a and b such that $f(c) = 0$. We have $c \neq a$ and $c \neq b$ because $f(a) \neq 0$ and $f(b) \neq 0$. ∎

Theorem 2.1.4 (**Extremum Theorem**)

Let $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$ be a continuous function. Then there exist x_s and x_i in $[a, b]$ such that

$$f(x_i) \leq f(x) \leq f(x_s)$$

for all $x \in [a, b]$.

Theorem 2.1.5 (**Mean Value Theorem**)

Let $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$ be a continuous function. Suppose that f is differentiable on $]a, b[$. Then there exists c between a and b such that

$$f'(c) = \frac{f(b) - f(a)}{b - a} .$$

Theorem 2.1.6 (**Taylor's Theorem**)

Let $a < b$ be two real numbers. Suppose that $f : [a, b] \to \mathbb{R}$ is a n-time continuously differentiable function on $[a, b]$, that $f^{(n+1)}(x)$ exists for all $x \in]a, b[$, and that $c \in]a, b[$. Then, for every $x \in [a, b]$, there exists $\xi(x, c)$ between x and c such that

$$f(x) = p_n(x) + r_n(x) ,$$

where

$$p_n(x) = f(c) + f'(c)(x - c) + \frac{f'(c)}{2!}(x - c)^2 + \ldots + \frac{f^{(n)}(c)}{n!}(x - c)^n$$

and

$$r_n(x) = \frac{f^{(n+1)}(\xi(x, c))}{(n + 1)!}(x - c)^{n+1} .$$

2.2 Bisection Method

The idea is to construct a sequence of nested intervals $\{[a_n, b_n]\}_{n=0}^{\infty}$ of decreasing length such that the sign of a function f at a_n is different than its sign at b_n. Thus, f must have a root at some point in the interval $[a_n, b_n]$ according to Corollary 2.1.3.

Algorithm 2.2.1 (Bisection)

Suppose that f is continue on $[a,b]$ and $f(a)\,f(b) < 0$.

1. Choose $a_0 = a$ and $b_0 = b$.
2. Stop if $f(a_0)f(b_0) = 0$ because one of a_0 or b_0 is a root of f.
3. Given a_n and b_n such that $f(a_n)\,f(b_n) < 0$, let $x_{n+1} = \dfrac{a_n + b_n}{2}$.
4. Stop if $f(x_{n+1}) = 0$ since $p = x_{n+1}$ is a root of f.
5. If $f(x_{n+1})\,f(a_n) < 0$, set $a_{n+1} = a_n$ and $b_{n+1} = x_{n+1}$. If $f(x_{n+1})\,f(a_n) > 0$, set $a_{n+1} = x_{n+1}$ and $b_{n+1} = b_n$.
6. Repeat (3), (4) and (5) until the interruption criteria are satisfied (more on the interruption criteria later).

Proposition 2.2.2

In the algorithm for the bisection method, $b_n - a_n = (b-a)/2^n$.

Proof.
We prove by induction that the interval $[a_n, b_n]$ is of length $(b-a)/2^n$.

We have $b_0 - a_0 = b - a = (b-a)/2^0$. Hence, the interval $[a_0, b_0]$ is of length $(b-a)/2^0$.

Suppose that the interval $[a_n, b_n]$ is of length $(b-a)/2^n$; namely, $b_n - a_n = (b-a)/2^n$. Since $[a_{n+1}, b_{n+1}]$ is half the length of $[a_n, b_n]$, we have

$$b_{n+1} - a_{n+1} = (b_n - a_n)/2 = (b-a)/2^{n+1} \ ,$$

where we have used the hypothesis of induction for the second equality. Hence, the interval $[a_{n+1}, b_{n+1}]$ is of length $(b-a)/2^{n+1}$.

By induction, we then have that $[a_n, b_n]$ is of length $(b-a)/2^n$ for all $n \geq 0$. ∎

Corollary 2.2.3

In the algorithm for the bisection method,, the approximation x_n is within $(b-a)/2^n$ of a root r of f in the interval $[a,b]$.

Proof.
Since f change sign in the interval $[a_{n-1}, b_{n-1}]$, there is a root r of f in the interval $[a_{n-1}, b_{n-1}]$. Since the approximation x_n of r is the middle point of the interval $[a_{n-1}, b_{n-1}]$, the absolute error $|x_n - r|$ satisfies $|x_n - r| < (b_{n-1} - a_{n-1})/2 = (b-a)/2^n$ according to Proposition 2.2.2. ∎

Proposition 2.2.4

In the algorithm for the bisection method,

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = \lim_{n\to\infty} x_n$$

and this limit is a root of f.

Proof.
Since $a_0 \le a_1 \le a_2 \le \ldots \le b$, the sequence $\{a_n\}_{n=0}^{\infty}$ is an increasing and bounded sequence. It follows from Theorem 2.1.1 that $\{a_n\}_{n=0}^{\infty}$ converges. Let α be this limit.

Similarly, since $b_0 \ge b_1 \ge b_2 \ge \ldots \ge a$, the sequence $\{b_n\}_{n=0}^{\infty}$ is a decreasing and bounded sequence. Thus $\{b_n\}_{n=0}^{\infty}$ converges. Let β be this limit.

Moreover,

$$\alpha - \beta = \lim_{n\to\infty} a_n - \lim_{n\to\infty} b_n = \lim_{n\to\infty} (a_n - b_n) = 0$$

by Proposition 2.2.2.

Since $a_n \le x_{n+1} \le b_n$ for all n, we have by the sandwich theorem that $\{x_n\}_{n=0}^{\infty}$ also converge to $\alpha = \beta$.

Finally, since $f(a_n)f(b_n) \le 0$ for all n, we have

$$(f(\alpha))^2 = f(\alpha)f(\alpha) = \lim_{n\to\infty} f(a_n)f(b_n) \le 0 \ .$$

Hence $f(\alpha) = 0$ and α is a root of f. ∎

Example 2.2.5
Find an approximation of $\sqrt{2}$ using the bisection method. Stop when the length of the interval is less than 10^{-2}. Find a bound on the absolute error.

The question is to find the positive root of $f(x) = x^2 - 2 = 0$. Let $a_0 = 1$ and $b_0 = 2$. Since $f(1) = -1 < 0 < 2 = f(2)$, there is a root of f in the interval $[1, 2]$. If

$$x_{n+1} = \frac{a_n + b_n}{2} \ ,$$

we get

n	x_n	a_n	b_n	$\lvert b_n - a_n\rvert$	$f(x_n)$	$f(a_{n-1})$
0		1	2	1.0		
1	1.500000	1	1.500000	.500000	+	−
2	1.250000	1.250000	1.500000	.250000	−	−
3	1.375000	1.375000	1.500000	.125000	−	−
4	1.437500	1.375000	1.437500	.062500	+	−
5	1.406250	1.406250	1.437500	.031250	−	−
6	1.421875	1.406250	1.421875	.015625	+	−
7	1.4140625	1.4140625	1.421875	.0078125	−	−
8	1.4179688					

The answer is $\sqrt{2} \approx 1.4179688$. There is a root in the interval $[1.4140625, 1.421875]$. So $(1.421875 - 1.4140625)/2 = 1/2^8 = 0.00390625$ is an upper-bound on the absolute error. ♣

Example 2.2.6
Using the formula provided by the bisection method, determine the smallest number of iterations in the previous example to get an absolute error lest than 10^{-4} ?

We choose n such that $|b_n - a_n| = (b-a)/2^n < 10^{-4}$. This is $2^{-n} < 10^{-4}$. Thus,

$$\ln(2^{-n}) < \ln(10^{-4}) \Rightarrow -n \ln(2) < -4 \ln(10) \Rightarrow n > 4 \ln(10)/\ln(2) \approx 13.2877$$

and 14 iterations will be sufficient. ♣

2.3 Interruption criteria

There are three interruption criteria that are usually used in the implementation of iteration methods:

1. Stop after N iterations (N is given).
2. Stop when $|x_{n+1} - x_n| < \epsilon$ (ϵ is given).
3. Stop when $|f(x_n)| < \eta$ (η is given).

We give below an implementation of the bisection method in Matlab, where we make use of the criteria 1 and 3.

Code 2.3.1 **(Bisection)**

To approximate the zeros of a function f.
Input: The function f (funct in the code below).
The endpoints a and b of the interval on which f changes sign.
The error tolerance (tol in the code below).
Output: The approximation x to a root of f.

```
% x = bisection(funct,a,b,tol)

function x = bisection(funct,a,b,tol)
  fa = feval(funct,a);
  fb = feval(funct,b);
  x = NaN;

  if ( a >= b )
    disp(['a must be smaller than b.'])
    return;
  end

  % We compute the theoritical number of iterations needed to reach
```

```
  % the accuracy requested.  This also prevent any infinite loops.
  %
  % From  (b-a)/2^n < tol  we get
  N = ceil(log2((b-a)/tol));

  % We replace fa*fb > 0 by a simple comparison of the signs of these
  % values.  We avoid a multiplication.
  if ( sign(fa) == sign(fb) )
    disp(sprintf('The bisection algorithm cannot be used because f(%f)
= %f and f(%f) = %f have the same sign.',a,b,fa,fb));
    return;
  end

  p = b - a;
  % We stop at i = N-1 because x_N is computed at i = N-1.
  for i=1:N-1
    p = p/2;

    % Instead of using the formula  (a+b)/2  to compute the middle
    % point, we simply add p to a.
    x = a + p;
    fx = feval(funct,x);

    % The test fx == 0 is not reliable because it is extremely rare
    % that the numerical evaluation of a function will give exactly 0.
    % We replace this test by  abs(fx) < 2*realmin , where realmin is the
    % smallest number that the computer may handle.
    if ( abs(fx) <= 2*realmin )
      return;
    end

    % We replace fa*fx < 0 by a simple comparison of the signs of these
    % values.  We avoid a multiplication.
    % We also store the value fx of f at the midpoint x into fa if
    % a takes the value x or into fb if b takes the value x.
    % This eliminates the need to compute f again at x.
    if ( sign(fx) ~= sign(fa) )
      b = x;
      fb = fx;
    else
      a = x;
      fa = fx;
    end
  end
end
```

2.4 Fixed Point Method

To find a root of f, we rewrite (2.0.1) as

$$x = g(x) \ , \tag{2.4.1}$$

where $g : \mathbb{R} \to \mathbb{R}$.

Given x_0, we hope that the sequence x_0, x_1, ... defined by

$$x_{n+1} = g(x_n) \quad \text{for} \quad n = 0, 1, 2, \ldots \tag{2.4.2}$$

will converge to a **fixed point** p of g; namely, a point p such that $g(p) = p$.

We say that (2.0.1) and (2.4.1) are **equivalent** (on a given interval) if a root of f is a fixed point of g and vice-versa. The problem is to choose g and x_0 adequately.

Example 2.4.1

$$f(x) = x^3 + 9\,x - 9 = 0$$

is equivalent to

$$g(x) = (9 - x^3)/9 = x \ .$$

♣

Theorem 2.4.2 (Fixed Point Theorem)

Let g be a real valued function satisfying the following conditions.

1. $g(x) \in [a, b]$ for all $x \in [a, b]$.
2. There exists a number K such that $0 < K < 1$ and $|g(x) - g(y)| \leq K|x - y|$ for all $x, y \in [a, b]$.

Then g has a unique fixed point $p \in [a, b]$ and, given $x_0 \in [a, b]$, the sequence defined by (2.4.2) converges to p as n goes to ∞. Moreover,

$$|x_n - p| \leq K^n \max\{x_0 - a, b - x_0\} \tag{2.4.3}$$

and

$$|x_n - p| \leq \frac{K^n}{1 - K}\,|x_1 - x_0| \ . \tag{2.4.4}$$

Proof.
We begin by proving the existence and uniqueness of the fixed point. Note that the second hypothesis of the theorem implies that g is a continuous function on $[a, b]$.

Since $g(a) \geq a$ and $g(b) \leq b$, the function $h(x) = g(x) - x$ is a continuous function on $[a,b]$ such that $h(b) \leq 0 \leq h(a)$. By the Intermediate Value Theorem, there exists $p \in [a,b]$ such that $h(p) = 0$; namely, $g(p) = p$.

Suppose that p_1 and p_2 are two distinct fixed points of g in $[a,b]$. We have

$$|p_1 - p_2| = |g(p_1) - g(p_2)| \leq K|p_1 - p_2| < |p_1 - p_2| \ .$$

This is a contradiction.

We now prove (2.4.3) and (2.4.4).

Let p be the unique fixed point of g in $[a,b]$ and let x_0 be a point in $[a,b]$. Since $g : [a,b] \to [a,b]$, the sequence $\{x_n\}_{n=0}^{\infty}$ defined by $x_{n+1} = g(x_n)$ for $n \geq 0$ is a well defined sequence in $[a,b]$. Hence,

$$\begin{aligned}|x_n - p| &= |g(x_{n-1}) - g(p)| \leq K|x_{n-1} - p| = K|g(x_{n-2}) - g(p)| \leq K^2|x_{n-2} - p| \\ &= \ldots \leq K^n|x_0 - p| \to 0\end{aligned}$$

as $n \to \infty$ because $0 < K < 1$. Moreover, since $|x_0 - p| \leq \max\{x_0 - a, b - x_0\}$, we get $|x_n - p| \leq K^n \max\{x_0 - a, b - x_0\}$. This prove (2.4.3).

To prove (2.4.4), we write

$$\begin{aligned}|x_{n+1} - x_n| &= |g(x_n) - g(x_{n-1})| \leq K|x_n - x_{n-1}| = K|g(x_{n-1}) - g(x_{n-2})| \leq K^2|x_{n-1} - x_{n-2}| \\ &= \ldots \leq K^n|x_1 - x_0| \ .\end{aligned}$$

Hence, for $m > n$,

$$\begin{aligned}|x_m - x_n| &= |x_m - x_{m-1} + x_{m-1} - x_{m-2} + \ldots - x_{n+1} + x_{n+1} - x_n| \\ &\leq |x_m - x_{m-1}| + |x_{m-1} - x_{m-2}| + \ldots + |x_{n+1} - x_n| \\ &\leq (K^{m-1} + K^{m-2} + \ldots + K^n)|x_1 - x_0| \\ &= K^n(K^{m-n-1} + K^{m-n-2} + \ldots + K + 1)|x_1 - x_0| \ .\end{aligned}$$

If we let m goes to infinity, we get

$$|p - x_n| \leq K^n \left(\sum_{i=0}^{\infty} K^i\right)|x_1 - x_0| = \frac{K^n}{1-K}|x_1 - x_0| \ .$$

The series in the previous expression is the geometric series which converges because $|K| < 1$. ∎

Definition 2.4.3

A continuous function $g : [a,b] \to \mathbb{R}$ for which there exists $0 < K < 1$ satisfying $|g(x) - g(y)| \leq K|x - y|$ for all $x, y, \in [a,b]$ is called a **contraction** on $[a,b]$.

Remark 2.4.4

In Theorem 2.4.2, the second hypothesis is that $g : [a,b] \to [a,b]$ is a contraction.

If g in Theorem 2.4.2 is differentiable and there exists $0 < K < 1$ such that $|g'(x)| \leq K$ for all $x \in [a,b]$, then the second hypothesis is satisfied. This is a consequence of the Mean Value Theorem. For every $x, y \in [a,b]$, there exists η between x and y such that

$$|g(x) - g(y)| = |g'(\eta)|\,|x-y| \leq K|x-y|$$

because $\eta \in [a,b]$. ♠

Example 2.4.5
Find an approximation to a root of $f(x) = x^3 + 9\,x - 9$.

Because $f(0)\,f(1) = -9 < 0$, the function f has a root between 0 and 1. In Example 2.4.1. we saw that $f(x) = x^3 + 9\,x - 9 = 0$ is equivalent to $g(x) = (9 - x^3)/9 = x$. Thus, the problem is to approximate a fixed point of g in $[0,1]$.

We show that g on the interval $[0,1]$ satisfies the hypotheses of the Fixed Point Theorem, Because $g'(x) = -x^2/3 < 0$ for all $x > 0$, the function g is decreasing on $[0,1]$. Hence, $8/9 = g(1) \leq g(x) \leq g(0) = 1$ for all $x \in [0,1]$. We have shown that $g : [0,1] \to [0,1]$ and thus the first hypothesis of the Fixed Point Theorem is satisfied with $[a,b] = [0,1]$. As mentioned in Remark 2.4.4, the second hypothesis of the Fixed Point Theorem is satisfied with $K = 1/3$ because $|g'(x)| = |-x^2/3| \leq 1/3$ for all x in $[0,1]$.

All conditions of the Fixed Point Theorem are satisfied. So, we may use it to approximate a fixed point p of g. The following table gives the first five iterations of $x_{n+1} = g(x_n)$ with $x_0 = 0.5$. The absolute and relative errors have been computed using the exact value of the fixed point p; namely, $p = 0.91490784153366\ldots$.

n	x_n	$\lvert x_n - p\rvert$	$\lvert x_n - p\rvert/\lvert p\rvert$	number of significant digits
0	0.5000000000	0.4149078415	0.4534968690	1
1	0.9861111111	0.0712032696	0.0778256195	1
2	0.8934545158	0.0214533257	0.0234486194	2
3	0.9207544589	0.0058466174	0.0063903894	2
4	0.9132660785	0.0016417630	0.0017944573	3
5	0.9153651027	0.0004572612	0.0004997894	4

♣

Example 2.4.6
Suppose that we want to approximate a root of $f(x) = x^3 + 4\,x^2 - 10$. The function f has a root in $[1,2]$ (show it). The four functions

$$g_1(x) = 10 + x - 4x^2 - x^3\ , \quad g_2(x) = \sqrt{\frac{10}{x} - 4x}\ , \quad g_3(x) = \frac{1}{2}\sqrt{10 - x^3}$$

and

$$g_4(x) = x - \frac{-10 + 4x^2 + x^3}{8x + 3x^2}$$

are equivalent to $f(x) = 0$ on the interval $[1, 2]$. We apply the fixed point method (without checking if the conditions of the Fixed Point Theorem are satisfied) to each function g_i with $x_0 = 1.5$.

n	$x_n = g_1(x_{n-1})$	$x_n = g_2(x_{n-1})$	$x_n = g_3(x_{n-1})$	$x_n = g_4(x_{n-1})$
0	1.5	1.5	1.5	1.5
1	-0.875	0.81649658	1.2869538	1.373333333
2	6.7324219	2.9969088	1.4025408	1.365262015
3	-469.72001		1.3454584	1.365230014
4	1.0275456×10^8		1.3751703	1.365230013
5	$-1.0849339 \times 10^{24}$		1.3600942	1.365230013

g_1 and g_2 generate sequences that do not converge. g_2 even ends up generating complex numbers. This shows that not all functions equivalent to f give converging fixed point iterations. We note the fast convergence of the fixed point iteration for the function g_4. We will show in the next section why it is so. ♣

2.5 Newton's Method

The idea is to construct a sequence $\{x_i\}_{i=0}^{\infty}$ which converges to a root p of a function f.

Algorithm 2.5.1 (**Newton**)

1. Choose x_0 closed to a root p of f (if possible).
2. Given x_n , compute
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.5.1}$$
if $f'(x_n) \neq 0$. If $f'(x_n) = 0$, start over with a better choice of x_0.
3. Repeat (2) until the interruption criteria are satisfied.

This method is also known as **Newton-Raphson's Algorithm**.

There is a nice graphical representation of the Newton's method that can be found in Figure 2.1. Let x_n be an approximation of a root p of f obtained from Newton's method. x_{n+1} is the x coordinate of the intersection of the tangent line to the curve $y = f(x)$ at $(x_n, f(x_n))$ with the x-axis. The equation of the tangent line to the curve $y = f(x)$ at $(x_n, f(x_n))$ is $y = f(x_n) + f'(x_n)(x - x_n)$. Hence x_{n+1} is the solution of $0 = f(x_n) + f'(x_n)(x - x_n)$. If $f'(x_n) \neq 0$, this is
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} .$$

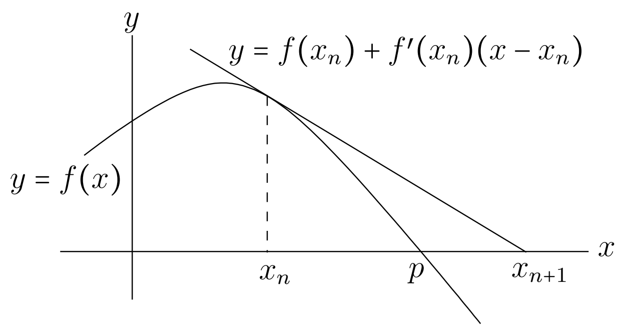


Figure 2.1: Newton's Method

Theorem 2.5.2

Let f be a twice continuously differentiable function on $[a,b]$. Suppose that $p \in [a,b]$ is a root of f such that $f'(p) \neq 0$. Then there exists $\delta > 0$ such that, for any $x_0 \in [p-\delta, p+\delta]$, the sequence defined by (2.5.1) converges to p as n goes to ∞.

This theorem will be proved as part of the more informative Theorem 2.7.3.

Remark 2.5.3

The Newton's method is the fixed point method defined by $x_{n+1} = g(x_n)$ with $g(x) = x - \dfrac{f(x)}{f'(x)}$.

If p is a fixed point of g, then $p = p - \dfrac{f(p)}{f'(p)}$ and we get $f(p) = 0$. ♠

Example 2.5.4

Find an approximation of $\sqrt{2}$ using the Newton's method. Stop when the difference between two consecutive iterations is smaller than 10^{-4}.

As in Example 2.2.5, we find an approximation of the positive root of $f(x) = x^2 - 2$. We have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{x_n^2 + 2}{2x_n}.$$

and start with $x_0 = 2$.

n	x_n (rounded to 6 decimals)	$\|x_{n-1} - x_n\|$	$< 10^{-4}$
0	2		
1	1.5	0.5	no
2	1.416667	0.083333	no
3	1.414216	0.002451	no
4	1.414214	0.000002	yes

The answer we are looking for is $x_4 \approx 1.414214$. ♣

Example 2.5.5
Use Newton's method to find an approximation of a root of f given in Example 2.4.6. Stop when the difference between two consecutive iterations is smaller than 10^{-10}.

We have
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 + 4x_n^2 - 10}{3x_n^2 + 8x_n} = \frac{2\left(x_n^3 + 2x_n^2 + 5\right)}{3x_n^2 + 8x_n} ,$$
and we take $x_0 = 1.5$.

n	x_n (rounded to 13 decimals)	$\lvert x_{n-1} - x_n\rvert$	$< 10^{-10}$
0	1.5		
1	1.3733333333333	0.126667	no
2	1.3652620148746	0.00807132	no
3	1.3652300139162	0.000032001	no
4	1.3652300134141	5.0205×10^{-10}	no
5	1.3652300134141	2.22045×10^{-16}	yes

The required approximation for the root of f is $x_5 \approx 1.3652300134141$. ♣

2.6 Secant Method

As for Newton's method, the idea is to construct a sequence $\{x_i\}_{i=0}^{\infty}$ that converges to a root p of f. The convergence of the secant method is generally slower than the convergence of the Newton's method but this secant method does not use the derivative of f. Moreover, only one evaluation of f is needed at each step of the secant method while one evaluation of f and one evaluation of f' are needed at each step of the Newton's method.

Algorithm 2.6.1 (Secant)

1. Choose two distinct values x_0 and x_1 near a root p of f (if possible)
2. Given two distinct values x_{n-1} and x_n, compute
$$x_{n+1} = x_n - f(x_n)\left(\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}\right)^{-1} = x_n - \frac{f(x_n)\,(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \qquad (2.6.1)$$
if $f(x_n) - f(x_{n-1}) \neq 0$. If $f(x_n) - f(x_{n+1}) = 0$, start over with a better choice of x_0 and x_1.
3. Repeat (2) until the interruption criteria are satisfied.

There is a graphical interpretation of the secant method which is given in Figure 2.2. Let x_{n-1} and x_n be two approximations of a root p of f. the next approximation x_{n+1} of p is the x-coordinate of the intersection of the x-axis with the secant line for the curve $y = f(x)$

through $(x_n, f(x_n))$ and $(x_{n-1}, f(x_{n-1}))$. The equation of the secant line is

$$y = f(x_n) + \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_n) \ .$$

Thus x_{n+1} is the solution of

$$0 = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_n) + f(x_n) \ .$$

If $f(x_n) - f(x_{n-1}) \neq 0$, this is

$$x_{n+1} = x_n - f(x_n)\left(\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}\right)^{-1} \ .$$

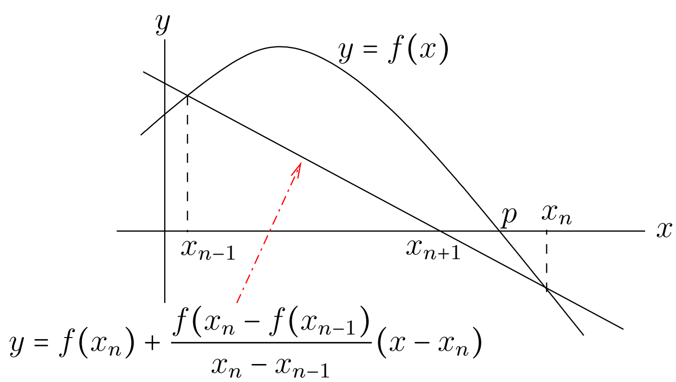


Figure 2.2: Secant Method

Remark 2.6.2

It is preferable to use the formula $x_{n+1} = x_n - \dfrac{f(x_n)\,(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$ instead of $x_{n+1} = x_n - f(x_n)\left(\dfrac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}\right)^{-1}$ to reduce the risk of divisions by numbers (i.e. $x_n - x_{n-1}$) almost equal to 0. ♠

Remark 2.6.3

The secant method is the fixed point method defined by $\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} = g\begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix}$ with

$$\begin{aligned} g : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \begin{pmatrix} x \\ y \end{pmatrix} &\mapsto \begin{pmatrix} y \\ y - \dfrac{f(y)}{F(x,y)} \end{pmatrix} \ , \end{aligned}$$

where F is defined by

$$F(x,y) = \begin{cases} \dfrac{f(x) - f(y)}{x - y} & \text{if} \quad x \neq y \\ f'(x) & \text{if} \quad x = y \end{cases}$$

The point p is a root of f if and only if $\begin{pmatrix} p \\ p \end{pmatrix}$ is a fixed point of g, The sequence $\{x_n\}_{n=0}^{\infty}$ converges to a root p of f if and only if the sequence $\left\{\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix}\right\}_{n=0}^{\infty}$ converges to a fixed point $\begin{pmatrix} p \\ p \end{pmatrix}$ of g.

We will study the fixed point method in $\mathbb{R}^n$ in Chapter 5. ♠

2.7 Order of Convergence

The following definition is used to determine the "quality" of an iterative method.

Definition 2.7.1

Suppose that the sequence $\{x_n\}_{n=0}^{\infty}$ converges to p. Let $e_n = x_n - p$. We say that $\{x_n\}_{n=0}^{\infty}$ **converges to** p **of order** α if there exists a non-zero real number λ such that

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^{\alpha}} = \lambda$$

If $\alpha = 1$, we talk of **linear convergence**. If $\alpha = 2$, we talk of **quadratic convergence**.

Theorem 2.7.2

Let $g : [a, b] \to \mathbb{R}$ be a sufficiently continuously differentiable function. Suppose that p is a fixed point of g in $[a, b]$ such that one of the following conditions is satisfied.

$k = 1$:

$$0 < |g'(p)| < 1 \ .$$

$k = 2, 3, \ldots$:

$$g'(p) = g''(p) = \ldots = g^{(k-1)}(p) = 0 \text{ and } g^{(k)}(p) \neq 0 \ .$$

Then there exists $\delta > 0$ such that, for $x_0 \in [p - \delta, p + \delta]$, the sequence defined by (2.4.2) converges to p of order k as n goes to ∞.

Proof.
Choose K such that $|g'(p)| < K < 1$. By continuity of g', there exists δ such that $|g'(x)| \leq K$ for all x in $[p-\delta, p+\delta]$. Using the Mean Value Theorem, it is easy to see that $g : [p-\delta, p+\delta] \to [p - \delta, p + \delta]$ (show it). Hence, when restricted to $[p - \delta, p + \delta]$, the function g satisfies all the hypothesis of the Fixed Point Theorem.

From the Fixed Point Theorem, p is the unique fixed point of g in $[p-\delta, p+\delta]$. Moreover, if $x_0 \in [p - \delta, p + \delta]$, the sequence $\{x_n\}_{n=0}^{\infty}$ defined by $x_{n+1} = g(x_n)$ for $n \geq 0$ converges to p.

The Taylor series expansion of g at p yields

$$x_{n+1} - p = g(x_n) - g(p) = \frac{1}{k!} g^{(k)}(\xi_n)\,(x_n - p)^k$$

for some ξ_n between x_n and p. If $x_n \to p$ as $n \to \infty$, then $\xi_n \to p$ as $n \to \infty$ because ξ_n is between x_n and p. Hence,

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^k} = \lim_{n\to\infty} \frac{|x_{n+1} - p|}{|x_n - p|^k} = \lim_{n\to\infty} \frac{|g^{(k)}(\xi_n)|}{k!} = \frac{|g^{(k)}(p)|}{k!} \neq 0 \ . \qquad \blacksquare$$

From the proof above, we have that the order of a fixed point method to find a root p of a function g is the order of the first non-null derivative of g at p.

Theorem 2.7.3

Let $f : [a,b] \to \mathbb{R}$ be a twice continuously differentiable function. Suppose that p is a root of f in $[a,b]$ and $f'(p) \neq 0$. Then there exists $\delta > 0$ such that, for $x_0 \in [p-\delta, p+\delta]$, the sequence $\{x_n\}_{n=0}^{\infty}$ produced by the Newton's method defined by (2.5.1) converges to p at least quadratically as n goes to ∞.

Proof.
By continuity of f', there exists δ' such that $f'(x) \neq 0$ for all x in $[p - \delta', p + \delta']$. Consider $g : [p - \delta', p + \delta'] \to \mathbb{R}$ defined by

$$g(x) = x - \frac{f(x)}{f'(x)} \ .$$

This function satisfies the hypotheses of Theorem 2.7.2 with $k > 1$ because

$$g'(x) = \frac{f(x) f''(x)}{(f'(x))^2}$$

on $[p - \delta', p + \delta']$ and so $g'(p) = 0$ because $f(p) = 0$. $\blacksquare$

Remark 2.7.4
If $f'(p) = 0$ in the previous theorem, the convergence (if there is convergence) of the sequence produced by the Newton's method may not be quadratic. If the Newton's method does not produce a sequence converging to a root of the function f, or if it produces a sequence converging very slowly to a root of f, it is possible to slightly modify the Newton's method to obtain a method that will produce a sequence converging quadratically to a root of f.

Suppose that p is a **zero of multiplicity** $k > 1$ **of** f; that is, $f(p) = f'(p) = \ldots = f^{(k-1)}(p) = 0$ and $f^k(p) \neq 0$. Instead of (2.5.1), one uses the fixed point method with the function

$$g(x) = \begin{cases} x - \dfrac{k\, f(x)}{f'(x)} & \text{if} \quad x \neq p \\ p & \text{if} \quad x = p \end{cases}$$

Because p is a zero of multiplicity k of f, it is shown in Question 2.29 that we can write f as $f(x) = (x-p)^k q(x)$ for some function q such that $q(p) \neq 0$. Hence, for $x \neq p$,

$$g(x) = x - \frac{k\,(x-p)^k\,q(x)}{k\,(x-p)^{k-1}\,q(x) + (x-p)^k\,q'(x)} = x - \frac{k\,(x-p)\,q(x)}{k\,q(x) + (x-p)\,q'(x)} \; .$$

This expression of g is well defined at p because the denominator of the fraction is different of 0 at p. In fact, the right-hand side evaluated at p gives p.

Moreover,

$$g'(x) = 1 - \frac{k\,q(x) + k\,(x-p)\,q'(x)}{k\,q(x) + (x-p)\,q'(x)} + \frac{(k\,(x-p)\,q(x))\,(k\,q'(x) + q'(x) + (x-p)\,q''(x))}{(kq(x) + (x-p)q'(x))^2} \; .$$

Hence

$$g'(p) = 1 - \frac{kq(p)}{kq(p)} = 0$$

and the convergence is at least quadratic. ♠

Theorem 2.7.5

Let $f : [a,b] \to \mathbb{R}$ be a twice continuously differentiable function. Suppose that p is a root of f in $[a,b]$, $f'(p) \neq 0$ and $f''(p) \neq 0$. Then there exists $\delta > 0$ such that, for x_0 and x_1 in $[p-\delta, p+\delta]$, the sequence $\{x_n\}_{n=0}^{\infty}$ produced by the secant method defined by (2.6.1) converges to p of order $(1+\sqrt{5})/2 \approx 1.618\ldots$ as n goes to ∞.

The proof of the convergence of the secant method is based on proving that the function g defined in Remark 2.6.3 satisfies the hypothesis of the Fixed Point Theorem. This proof will not be given here.

We will also not prove that there exist $\alpha > 0$ and $\lambda \neq 0$ such that

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda \; . \tag{2.7.1}$$

This proof is tricky. We prove in Remark 6.2.17 of Section 6.2 that if there exist $\alpha > 0$ and $\lambda \neq 0$ such that (2.7.1) is satisfied, then α must be the **golden ratio** $(1+\sqrt{5})/2$. To prove this, we use divide difference formulae that are presented in Chapter 6.

2.8 Aitken's Δ^2 Process and Steffensen's Algorithm

Suppose that p_0 is an initial approximation for a fixed point p of a function g. Moreover, suppose that the sequence $\{p_n\}_{n=0}^{\infty}$ defined by $p_{n+1} = g(p_n)$ for $n \geq 0$ converges linearly to p. We give a procedure to build a new sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ that converges "faster" to p than $\{p_n\}_{n=0}^{\infty}$.

Let

$$\Delta p_n = p_{n+1} - p_n$$

$$
\begin{aligned}
\Delta^2 p_n &= \Delta(\Delta p_n) = \Delta p_{n+1} - \Delta p_n = p_{n+2} - 2p_{n+1} + p_n \\
\dots \quad & \dots \\
\Delta^k p_n &= \Delta(\Delta^{k-1} p_n)
\end{aligned}
$$

for $n \geq 0$.

The sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ defined by

$$
\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} \tag{2.8.1}
$$

converges to p and the order of convergence of $\{\hat{p}_n\}_{n=0}^{\infty}$ to p is greater than 1. The procedure used to construct $\{\hat{p}_n\}_{n=0}^{\infty}$ is called the **Aitken's Δ^2 process**.

$$
\begin{array}{lll}
p_0 \,,\, p_1 \,,\, p_2 & \text{give} & \hat{p}_0 \\
p_1 \,,\, p_2 \,,\, p_3 & \text{give} & \hat{p}_1 \\
p_2 \,,\, p_3 \,,\, p_4 & \text{give} & \hat{p}_2 \\
& \dots &
\end{array}
$$

Since $\hat{p}_n$ is generally a better approximation of p than p_{n+1}, it is better to replace p_n, p_{n+1} and p_{n+2} in (2.8.1) by $\hat{p}_{n-1}$, $g(\hat{p}_{n-1})$ and $g(g(\hat{p}_{n-1}))$. Using this idea, we get the following algorithm.

Algorithm 2.8.1 (Steffensen's)

1. Choose p_0 closed to a fixed point p of g (if possible).
2. Let $\hat{p}_{-1} = p_0$.
3. For $n \geq -1$, compute

$$
\hat{p}_{n+1} = \hat{p}_n - \frac{(g(\hat{p}_n) - \hat{p}_n)^2}{g(g(\hat{p}_n)) - 2g(\hat{p}_n) + \hat{p}_n} \,. \tag{2.8.2}
$$

4. Repeat (3) until the interruption criteria are satisfied.

$$
\begin{array}{lll}
p_0 \,,\, g(p_0) \,,\, g(g(p_0)) & \text{give} & \hat{p}_0 \,. \\
\hat{p}_0 \,,\, g(\hat{p}_0) \,,\, g(g(\hat{p}_0)) & \text{give} & \hat{p}_1 \,. \\
\hat{p}_1 \,,\, g(\hat{p}_1) \,,\, g(g(\hat{p}_1)) & \text{give} & \hat{p}_2 \,. \\
& \dots &
\end{array}
$$

Theorem 2.8.2

Let $g : [a, b] \to \mathbb{R}$ be a 3-time continuously differentiable function. Suppose that p is a fixed point of g in $[a, b]$ and $g'(p) \neq 0$. Then there exists $\delta > 0$ such that, for

$p_0 \in [p-\delta, p+\delta]$, the sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ defined by (2.8.2) converges to p of order two as n goes to ∞.

Proof (Idea).
The idea of the proof is to apply Theorem 2.7.2 with $k=2$ to the function

$$G(x) = \begin{cases} x - \dfrac{(g(x)-x)^2}{g(g(x)) - 2g(x) + x} & \text{if} \quad x \neq p \\ p & \text{if} \quad x = p \end{cases}$$

∎

Remark 2.8.3
Though the order of the Steffensen's Algorithm is greater than the order of the secant method, the Steffensen's Algorithm is not always faster on computer than the secant method because there are two function evaluations, four additions/subtractions and three multiplications/divisions at each step for the Steffensen's Algorithm while there are one function evaluation, three additions/subtractions and two multiplications/divisions at each step for the secant method. A function evaluation may be time consuming. ♠

2.9 Real Roots of Polynomials

In this section, we do not introduce any new iterative algorithms but show how to efficiently use Newton's method to approximate the real roots of a polynomial.

Let p be a polynomial of degree m. If we apply Newton's method to this polynomial, we get the formula

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}$$

for $n \geq 0$. The following theorem gives an algorithm to compute $p(x_n)$ and $p'(x_n)$ with a lot less arithmetic operations than the direct computation of $p(x_n)$ and $p'(x_n)$.

Theorem 2.9.1 (Horner)

Let $p(x) = a_m\, x^m + a_{m-1}\, x^{m-1} + \ldots + a_1\, x + a_0$ and

$$\begin{aligned} b_m &= a_m \\ b_{m-1} &= a_{m-1} + b_m\, \alpha \\ b_{m-2} &= a_{m-2} + b_{m-1}\, \alpha \\ \ldots & \quad \ldots \\ b_k &= a_k + b_{k+1}\, \alpha \\ \ldots & \quad \ldots \\ b_0 &= a_0 + b_1\, \alpha \end{aligned}$$

Then $b_0 = f(\alpha)$ and $p(x) = (x-\alpha)\, q(x) + b_0$ where $q(x) = b_m\, x^{m-1} + b_{m-1}\, x^{m-2} + \ldots + b_2\, x + b_1$.

Moreover $p'(\alpha) = q(\alpha)$.

Proof.
We have

$$\begin{aligned}(x-\alpha)\,q(x) + b_0 &= (x-\alpha)(b_m\,x^{m-1} + b_{m-1}\,x^{m-2} + \ldots + b_2\,x + b_1) + b_0\\ &= b_m\,x^m + b_{m-1}\,x^{m-1} + b_{m-2}\,x^{m-2} + \ldots + b_2\,x^2 + b_1\,x\\ &\quad - b_m\,\alpha\,x^{m-1} - b_{m-1}\,\alpha\,x^{m-2} - \ldots - b_3\,\alpha\,x^2 - b_2\,\alpha\,x - b_1\,\alpha + b_0\\ &= b_m\,x^m + (b_{m-1} - b_m\,\alpha)x^{m-1} + (b_{m-2} - b_{m-1}\,\alpha)x^{m-2} + \ldots\\ &\quad + (b_2 - b_3\,\alpha)x^2 + (b_1 - b_2\,\alpha)x + (b_0 - b_1\,\alpha)\\ &= a_m\,x^m + a_{m-1}\,x^{m-1} + a_{m-2}\,x^{m-2} + \ldots + a_2\,x^2 + a_1\,x + a_0 = f(x)\end{aligned}$$

because $b_m = a_m$ and $b_k - b_{k+1}\,\alpha = a_k$ for $k = 0, 1, 2, \ldots, m-1$. Moreover, $f(\alpha) = (\alpha-\alpha)q(\alpha) + b_0 = b_0$.

Since $p'(x) = (x-\alpha)q'(x) + q(x)$, we get $p'(\alpha) = q(\alpha)$. ∎

At the same time that $p(\alpha)$ is computed with Horner's Algorithm, a second used of Horner's Algorithm with p replaced by q may compute $p'(\alpha) = q(\alpha)$, More precisely, if

$$\begin{aligned} d_m &= b_m\\ d_{m-1} &= b_{m-1} + d_m\,\alpha\\ d_{m-2} &= b_{m-2} + d_{m-1}\,\alpha\\ \cdots &\quad \cdots\\ d_k &= b_k + d_{k+1}\,\alpha\\ \cdots &\quad \cdots\\ d_1 &= b_1 + d_2\,\alpha\end{aligned}$$

then $p'(\alpha) = q(\alpha) = d_1$. This may be expanded to higher order derivatives.

Hence, Horner's theorem gives an efficient way to compute $p(x_n)$ and $p'(x_n)$ in the Newton's method. If $\alpha = x_n$ in Horner's theorem, then $p(x_n) = b_0$ and $p'(x_n) = d_1$.

The computation of $p(x_n)$ and $p'(x_n)$ are combined in the following algorithm.

Code 2.9.2 (Horner's Algorithm)

To evaluate a polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$ and its derivative at a point α.

Input: The coefficients a_i (the vector a in the code below). The coefficient a_n must be given even if it is zero.
The value of α (x in the code below.)
Output: $y = p(\alpha)$ and $z = p'(\alpha)$.

```
% [y,z] = horner(a,x)
```

```
function [y,z] = horner(a,x)
  m = length(a);
  y = a(m);
  z = a(m);
  for i = m-1:-1:2
    y = a(i) + x*y;
    z = y + x*z;
  end
  y = a(1) + x*y;
end
```

If we combine Newton's method and Horner's Algorithm, we get

Code 2.9.3 (Newton's Method with Horner's Algorithm)

To approximate a real root of a polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$

Input: The coefficients a_i (The vector a in the code below) The coefficient a_n must be given even if it is zero.
The initial approximation x_0 (x in the code below) of a root c of p.
The maximal tolerance T.
The maximal number N of iterations.
Output: An approximation (xf in the code below) of the real root c
or
an error message if the real root cannot be approximate with the desired tolerance in less than N iterations.

```
% xf = realroot(a,x,N,tol)

function xf = realroot(a,x,N,tol)
  xf = NaN;
  m = length(a);

  for k=1:N
    y = a(m);
    z = a(m);
    for i=m-1:-1:2
      y = a(i) + x*y;
      z = y + x*z;
    end
    y = a(1) + x*y;

    if ( abs(z) < tol )
      disp 'The derivative is almost null.  Cannot proceed.'
      break;
    end
```

```
    % y = p(x)  and  z = p'(x) .
    ratio = y/z;
    x = x - ratio;
    if (abs(ratio) < tol)
      xf = x;
      return;
    end
  end

  disp 'The program fails to give an approximation to a root of'
  disp 'the polynomial in less than the N iterations.'
  xf = NaN;
end
```

Remark 2.9.4

1. Newton's method may not be so good if we try to approximate a root of multiplicity greater than one of a polynomial. See Remark 2.7.4.

2. A good initial approximation x_0 of a root c of a polynomial p must be given if we want the Newton's method to generate a sequence $\{x_n\}_{n=0}^{\infty}$ that converges to c. A bad choice for x_0 and the sequence may converge toward another root of p or may not converge at all.

3. Small changes in the coefficients of a polynomial of high degree may produce very large changes in the roots of this polynomial.

   For instance, the polynomial
$$p(x) = x^7 - 28x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040$$
   has the roots 1, 2, 3, 4, 5, 6 and 7. However, the polynomial
$$\tilde{p}(x) = x^7 - 28x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13133x^2 + 13068x - 5040 ,$$
   where only the coefficient of x^2 has be changed from 13132 to 13133, has the roots (rounded to seven decimals) 1.0013976, 1.9689208, 3.3183233, 3.5050604, $5.5731849 \pm 0.2641298\, i$ and 7.0599281 which are quite different from those of the initial polynomial.

4. In theory, if we have a real root c of p, we can use Horner's theorem with $\alpha = c$ to express p as $p(x) = (x - c)q(x)$ because $b_0 = p(c) = 0$. To find a second root of p, we only have to find a root of q. the polynomial q is called the **reduced** or **deflated polynomial** associate to p. In reality, we only have an approximation of the root c and Horner's theorem gives only approximations of the coefficients b_j of q. In light of item 3 above, the approximation of a real root of q may have little relation with a real root of p. However, we may use this approximation as x_0 in the Newton's method applied to p to get an approximation of a new root (we hope) of p.

♠

Example 2.9.5
Let $p(x) = x^7 - 28x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040$. Approximate all the roots of p within 10^{-10}.

In the following table

1. $q_0 = p$
2. c_0 is an approximation of a root of q_0 obtained with the Newton's method and the initial value $x_0 = 2.5$ (any other initial value could have been used).
3. $r_0 = c_0$
4. For $i = 1, 2, \ldots, 6$.
   (i) The polynomial q_i is the deflated polynomial obtained from the previous deflated polynomial q_{i-1} with the help of Horner's Algorithm. In theory, we have $q_{i-1} = (x - r_{i-1})q_i$.
   (ii) The number r_i is an approximation of a root of the deflated polynomial q_i obtained with the Newton's method the initial value $x_0 = 2.5$.
   (iii) The number c_i is an approximation of a root of the polynomial p obtained with the Newton's method and the initial value $x_0 = r_i$.

i	q_i	r_i	c_i
0	$-5040 + 13068x - 13132x^2 + 6769x^3 - 1960x^4 + 322x^5 - 28x^6 + x^7$	1	1
1	$5040 - 8028x + 5104x^2 - 1665x^3 + 295x^4 - 27x^5 + x^6$	2	2
2	$-2520 + 2754x - 1175x^2 + 245x^3 - 25x^4 + x^5$	3	3
3	$840 - 638x + 179x^2 - 22x^3 + x^4$	4	4
4	$-210 + 107x - 18x^2 + x^3$	5	5
5	$42 - 13x + x^2$	6	6
6	$-7 + x$	7	7

We get the exact roots after rounding. ♣

Example 2.9.6
Let $p(x) = 5 - 3x - 4x^2 + x^4$. Approximate all the roots of p within 10^{-10}.

In the following table

1. $q_0 = p$
2. c_0 is an approximation of a root of p obtained with the Newton's method and the initial value $x_0 = 2$ (any other initial value could have been used).
3. $r_0 = c_0$

4. For $i = 1$ and 2.

   (i) The polynomial q_i is the deflated polynomial obtained from the previous deflated polynomial q_{i-1} with the help of Horner's Algorithm. In theory, we have $q_{i-1} = (x - r_{i-1})q_i$.

   (ii) If $i = 1$, the number r_i is an approximation of a root of the deflated polynomial q_i obtained with the Newton's method and the initial value $x_0 = 2$.

   (iii) If $i = 1$, the number c_i is an approximation of a root of the polynomial p obtained with the Newton's method and the initial value $x_0 = r_1$.

i	q_i coefficients rounded to 10 decimals	r_i	c_i
0	$5 - 3x - 4x^2 + x^4$	2.0693229488	2.0693229488
1	$-2.4162492389 + 0.2820974665x + 2.0693229488x^2 + x^3$	0.8611735320	0.8611735320
2	$2.8057634715 + 2.9304964809x + x^2$	NaN	NaN

The method using the deflated polynomials combined with Newton's method fails to give all the roots of the polynomial p. The deflated polynomial, where the coefficients have been rounded to 14 decimals,

$$q_2(x) = 2.80576347152215 + 2.93049648085253\,x + x^2$$

does not have real roots. Since q_2 is a polynomial of degree two, we can use the quadratic formula to find the roots of q_2. We find $-1.46524824042627 \pm 0.81167177199277i$, where the real and imaginary parts have been rounded to 14 decimals.

The Newton's method works for the complex roots of polynomials with complex coefficients. So, we may use the Newton's method with p and the initial value x_0 given by one of the complex roots of q_2. Since p has real coefficients, we know that complex roots come in pair. ♣

2.10 Appendix

This section is to illustrate how complex a simple **discrete dynamical system** of the form

$$x_{i+1} = f(x_i) ,$$

where $f : \mathbb{R} \to \mathbb{R}$ is a continuous function, can be. The complexity is even greater if $f : \mathbb{R}^k \to \mathbb{R}^k$ with $k > 1$. Discrete dynamical systems show up in many numerical algorithms. For instance, the Newton's Method to find zeros of functions yields discrete dynamical systems, some numerical methods to solve ordinary differential equations or partial differential equations are discrete dynamical systems, etc. It is therefore important to understand the behaviour of discrete dynamical systems, or at least to be aware of the complex behaviour of these systems.

A good introduction to the subject of this appendix is [12]. It is also a good reference for the proofs of most of results stated in this appendix.

2.10.1 Elementary Concepts of Discrete Dynamical Systems

Definition 2.10.1

Consider a continuous function $f : \mathbb{R} \to \mathbb{R}$ and $x \in \mathbb{R}$. The **forward orbit** of x is the set
$$\mathcal{O}_x^+ = \left\{ x, f(x), f^2(x) = f(f(x)), f^3(x) = f(f(f(x))), \ldots \right\} .$$
If f has an inverse $f^{-1} : \mathbb{R} \to \mathbb{R}$, the **backward orbit** of x is the set
$$\mathcal{O}_x^- = \left\{ x, f^{-1}(x), f^{-2}(x) = f^{-1}(f^{-1}(x)), f^{-3}(x) = f^{-1}(f^{-1}(f^{-1}(x))), \ldots \right\}$$
and the **orbit** of x is $\mathcal{O}_x = \mathcal{O}_x^+ \cup \mathcal{O}_x^-$.

Definition 2.10.2

Consider a continuous function $f : \mathbb{R} \to \mathbb{R}$. The point $p \in \mathbb{R}$ is a **periodic point** of f if there exists a positive integer n such that $f^n(p) = p$. If n is the smallest positive integer such that $f^n(p) = p$, we say that p is of **period** n. When $n = 1$, p is a **fixed point**. We denote by $\mathrm{Per}_n(f)$ the set of all periodic point of f of period n. In particular, $\mathrm{Fix}(f) = \mathrm{Per}_1(f)$ is the set of fixed points of f. If p is a periodic point of f, $\mathcal{O}_p$ is a **periodic orbit**.

Example 2.10.3
Consider the **logistic map**
$$f_\mu(x) = \mu x(1 - x)$$
for $0 \leq x \leq 1$. For $0 \leq \mu \leq 4$, we have $f_\mu : [0,1] \to [0,1]$. Moreover, f_μ has two fixed points: $p_0 = 0$ and $p_\mu = \dfrac{\mu - 1}{\mu}$ for $\mu > 0$.

For $\mu = 3.4$, $p_{3.4} = 0.45195878844045\ldots$ is a periodic point of f_μ of period 2. The orbit of period two is
$$\{0.45195878844045, 0.84215476876273, 0.45195878844045, 0.84215476876273, \ldots\} ,$$
where the values have been chopped to 14 digits after the decimal point. This can be easily seen from the **staircase diagram** or **cobweb** of f_μ shown in Figure 2.3.

Another way to illustrate the behaviour of f_μ is with the **phase portrait** of f_μ shown in Figure 2.4.

Finally, one can plot the histogram of f_μ. Namely, we divide the interval $[0,1]$ into a large number of subintervals of equal lengths and we compute the percentage of iterations that enter each subinterval. For $\mu = 3.4$, the histogram with 200 subintervals of $[0,1]$ and $10,000$ iterations is given in Figure 2.5.

For $\mu = 3.5$, $p_{3.5} = 0.87499726360246\ldots$ is a periodic point of f_μ of period 4. The orbit of period four is
$$\{0.87499726360246, 0.38281968301732, 0.82694070659144, 0.50088421030722, \ldots\} ,$$

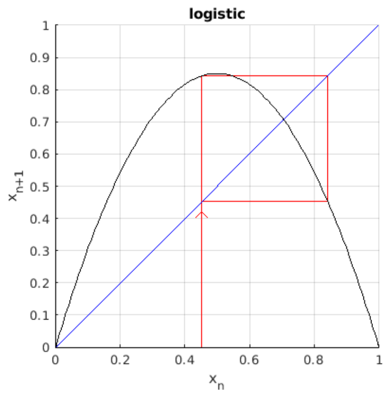


Figure 2.3: Cobweb

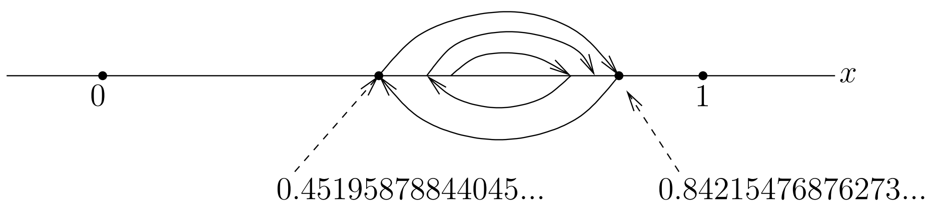


Figure 2.4: Phase Portrait

where the values have been chopped to 14 digits after the decimal point (see Figure 2.6).

♣

Definition 2.10.4

Consider a continuous function $f : \mathbb{R} \to \mathbb{R}$. The points p and q in $\mathbb{R}$ are **forward asymptotic** if

$$\lim_{j\to\infty} \left| f^j(p) - f^j(q) \right| = 0 \ .$$

In particular, if $p \in \mathbb{R}$ is a periodic point of period n, then a point q is **forward asymptotic to** p if

$$p = \lim_{j\to\infty} f^{jn}(q) \ .$$

The set of all points forward asymptotic to p is denoted by $W^s(p)$. There are similar definitions for **backward asymptotic**.

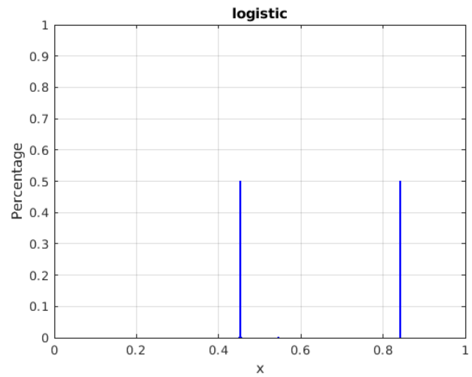


Figure 2.5: Histogram for $f_{3.4}$

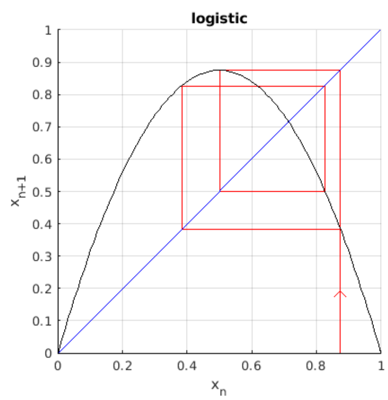


Figure 2.6: Period Four

Definition 2.10.5

Consider a continuous function $f : \mathbb{R} \to \mathbb{R}$. A fixed point p of f is **stable** if, for any open neighbourhood U of p, there exists an open neighbourhood V of p such that $f^i(V) \subset U$ for all $i > 0$. Fixed points that are not stable are called **unstable**. A fixed point p of f is **asymptotically stable** if it is stable and there exists an open neighbourhood W of p such $\lim_{i\to\infty} f^i(x) = p$ for all $x \in W$.

If p is a period point of period n for f, we say that the periodic orbit $\mathcal{O}_p$ is **stable** if p is a stable fixed point of f^n. We say that the periodic orbit is **asymptotically stable** if p is an asymptotically stable fixed point of f^n.

Remark 2.10.6

The previous definition of stability and asymptotic stability for a period orbit is independent of the point p of the orbit used to determine the stability or the asymptotic stability.

Suppose that p is a periodic point of period $n > 1$ for a continuously invertible function $f : \mathbb{R} \to \mathbb{R}$. Suppose that p is stable for f^n and let $q = f^k(p)$ be another point on the orbit $\mathcal{O}_p$. If U_q is an open neighbourhood of q, then $f^{-k}(U_q)$ is an open neighbourhood of p. Since p is a stable fixed point for f^n, there exists an open neighbourhood V_p of p such that $f^{ni}(V_q) \subset f^{-k}(U_q)$ for all $i > 0$. Hence $f^{ni}(f^k(V_p)) = f^{ni+k}(V_q) \subset U_q$ for all $i > 0$, where $f^k(V_p)$ is an open neighbourhood of q. This proves that q is a stable fixed point of f^n.

Suppose furthermore that p is an asymptotically stable fixed point of f^n. Then there exists an open neighbourhood W_p of p such that $\lim_{i\to\infty} f^{ni}(x) = p$ for all $x \in W_p$. Thus $\lim_{i\to\infty} f^{ni}(f^k(x)) = f^k(p) = q$ for all $x \in W_p$; namely, $\lim_{i\to\infty} f^{ni}(y) = q$ for all y in the open neighbourhood $f^k(W_p)$ of q. This proves that q is an asymptotically stable fixed point of f^n. ♠

2.10.2 Qualitative Study

Consider the discrete dynamical system

$$x_{i+1} = f(x_i) \; . \tag{2.10.1}$$

For a qualitative study of this system, we would like to find all the fixed points, periodic orbits, ... We would also like to find the sets of points forward asymptotic to these objects.

We first study the fixed points of (2.10.1).

Definition 2.10.7

A fixed point p of f is **hyperbolic** if $|f'(p)| \neq 1$.

A proof similar to the proof of the Fixed Point Theorem yields the next theorem.

Proposition 2.10.8

Let p be an hyperbolic fixed point of f. If $|f'(p)| < 1$, then p is asymptotically stable However, if $|f'(p)| > 1$, then there exists an open interval I containing p such that for each x in I, $x \neq p$, one can find $j \in \mathbb{N}$ such that $f^j(x) \notin I$. This implies that $f^j(x) \notin I$ for infinitely many values of j.

Definition 2.10.9

If p is an hyperbolic fixed point of f such that $|f'(p)| < 1$, then p is called an **attracting** fixed point or a **sink**. If p is an hyperbolic fixed point of f such that $|f'(p)| > 1$, then p is called a **repelling** fixed point or a **source**.

Example 2.10.10

For the logistic map $f_\mu(x) = \mu x(1-x)$, the origin is a source if $\mu > 1$, and $p_\mu = \dfrac{\mu - 1}{\mu}$ is a sink if $1 < \mu < 3$. This follows from $\left.\dfrac{\partial}{\partial x} f_\mu(x)\right|_{x=0} = \mu$ and $\left.\dfrac{\partial}{\partial x} f_\mu(x)\right|_{x=p_\mu} = 2 - \mu$. ♣

We can expand the notion of hyperbolicity to periodic points.

Definition 2.10.11

A periodic point p of period n for f is **hyperbolic** if $\left|\frac{\mathrm{d}f^n}{\mathrm{d}x}(p)\right| \neq 1$.

A periodic point p of f of period n is a fixed point of f^n. The stability of the periodic point p of f is determined by the stability of the fixed point p of f^n. Proposition 2.10.8 holds for a periodic point p of period n if f is replaced by f^n.

Example 2.10.12
To study the stability of the periodic points of period 2 for the logistic map f_μ with $\mu = 3.4$, we consider the iterative system $x_{i+1} = f_\mu^2(x_i) = f_\mu(f_\mu(x_i))$ for $i = 0, 1, 2, \ldots$

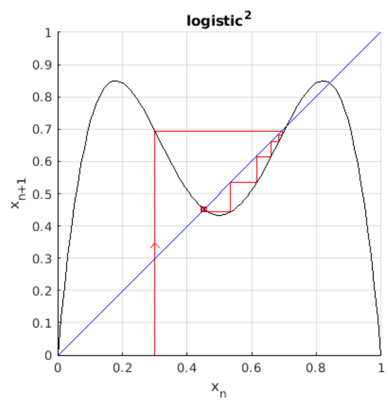


Figure 2.7: Period Two

From the graph of f_μ^2 given in Figure 2.7, we see that the periodic point $0.45195878844045\ldots$ of period 2 is a sink. ♣

2.10.3 Bifurcation

Consider a nice function $f : \mathbb{R}^2 \to \mathbb{R}$. It defines a one-parameter family of functions $f_\mu(x) = f(x, \mu)$.

We say that $\mu = \mu_0$ is a **bifurcation point** of the discrete dynamical system

$$x_{i+1} = f_\mu(x_i)$$

if the qualitative behaviour of the phase portrait changes as μ goes through μ_0. For instance, the number of fixed points change, new periodic solutions appear, etc.

There are three major results related to bifurcation. The first result is a simple consequence of the Implicit Function Theorem applied to the function $g(x, \mu) = f(x, \mu) - x$.

Theorem 2.10.13

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a smooth function and let $f_\mu(x) = f(x,\mu)$. Suppose that $f_{\mu_0}(x_0) = x_0$ and

$$\left.\frac{\partial}{\partial x} f_\mu(x)\right|_{\mu=\mu_0, x=x_0} = \left.\frac{\partial f}{\partial x}(x,\mu)\right|_{\mu=\mu_0, x=x_0} \neq 1 \ ,$$

then there exist an open interval I about x_0, an open interval J about μ_0, and a mapping $p : J \to I$ such that $p(\mu_0) = x_0$ and $f_\mu(p(\mu)) = p(\mu)$ for all $\mu \in U$. Moreover, f_μ has no other fixed point in I (Figure 2.8).

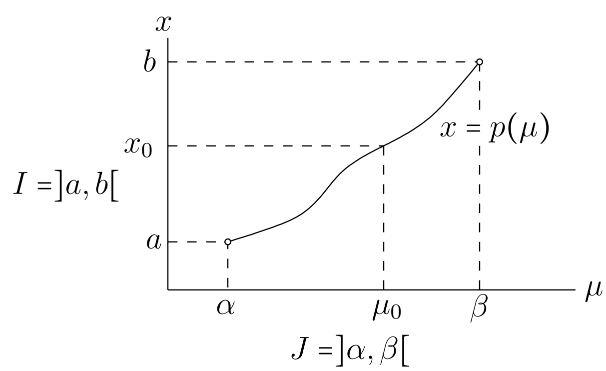


Figure 2.8: The Implicit Function Theorem

The next theorems describe two "generic" types of bifurcation. We have bifurcation only when $\left|\frac{\partial}{\partial x} f_\mu(x)\right| = 1$. We use the word generic because the other conditions to classify these types of bifurcation require only that some derivatives be non-null.

Theorem 2.10.14 (**Saddle-node, tangent or fold bifurcation**)

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a smooth function and let $f_\mu(x) = f(x,\mu)$. Suppose that

1. $f_{\mu_0}(0) = 0$,
2. $\left.\frac{\partial}{\partial x} f_\mu(x)\right|_{\mu=\mu_0, x=0} = 1$,
3. $\left.\frac{\partial^2}{\partial x^2} f_\mu(x)\right|_{\mu=\mu_0, x=0} \neq 0$ and
4. $\left.\frac{\partial}{\partial \mu} f_\mu(x)\right|_{\mu=\mu_0, x=0} \neq 0$.

Then, there exist an interval I about 0 and a mapping $q : I \to \mathbb{R}$ such that $q(0) = \mu_0$ and $f_{q(x)}(x) = x$. Moreover, $q'(0) = 0$ and $q''(0) \neq 0$.

Figure 2.9 illustrates a typical fold bifurcation. The fixed points represented by a dashed curve are sources while those represented by a continuous curve are sinks. The conditions in the statement of the theorem above do not determine which branch of the curve is associated to sources and which branch is associated to sinks. Moreover,

$$q''(0) = \frac{-\left.\dfrac{\partial^2}{\partial x^2} f_\mu(x)\right|_{\mu=\mu_0, x=0}}{\left.\dfrac{\partial}{\partial \mu} f_\mu(x)\right|_{\mu=\mu_0, x=0}}$$

can be used to determine if the curve $\mu = p(x)$ is **supercritical** (namely, $q''(x) > 0$ as in Figure 2.9) or **subcritical** (namely, $q''(x) < 0$).

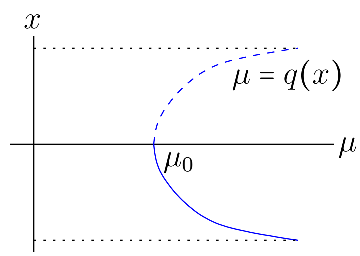


Figure 2.9: A typical fold bifurcation diagram for a discrete map

Theorem 2.10.15 (Period doubling or flip bifurcation)

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a smooth function and let $f_\mu(x) = f(x, \mu)$. Suppose that

1. $f_\mu(0) = 0$ for all μ near μ_0,
2. $\left.\dfrac{\partial}{\partial x} f_\mu(x)\right|_{\mu=\mu_0, x=0} = -1$,
3. $\dfrac{1}{2}\left(\left.\dfrac{\partial^2}{\partial x^2} f_\mu(x)\right|_{\mu=\mu_0, x=0}\right)^2 + \dfrac{1}{3}\left.\dfrac{\partial^3}{\partial x^3} f_\mu(x)\right|_{\mu=\mu_0, x=0} \neq 0$ and
4. $\left.\dfrac{\partial^2}{\partial \mu \partial x} f_\mu^2(x)\right|_{\mu=\mu_0, x=0} \neq 0$,

where $f_\mu^2(x) \equiv f(f(x, \mu), \mu)$. Then, there exist an interval I about 0 and a mapping $q : I \to \mathbb{R}$ such that $q(0) = \mu_0$ and $f_{q(x)}(x) \neq x$ but $f_{q(x)}^2(x) = x$.

Figure 2.10 illustrates a typical period doubling bifurcation. The fixed points are represented by the straight line $x = 0$ and the periodic points of period two are represented by the curve. For μ fixed, a periodic orbit of period two alternates between the lower and the upper curve. The fixed points represented by a dashed curve are sources while those represented by a continuous curve are sinks. The periodic points of period two represented by a dashed curve are unstable while those represented by a continuous curve are asymptotically stable. Moreover,

$$q''(0) = \frac{\left(\left.\frac{\partial^2}{\partial x^2} f_\mu(x)\right|_{\mu=\mu_0,x=0}\right)^2 + \left.\frac{2}{3}\frac{\partial^3}{\partial x^3} f_\mu(x)\right|_{\mu=\mu_0,x=0}}{\left.\frac{\partial^2}{\partial x \partial \mu} f_\mu^2(x)\right|_{\mu=\mu_0,x=0}}$$

can be used to determine if the curve $\mu = q(x)$ is **supercritical** (namely, $q''(x) > 0$ as in Figure 2.10) or **subcritical** (namely, $q''(x) < 0$).

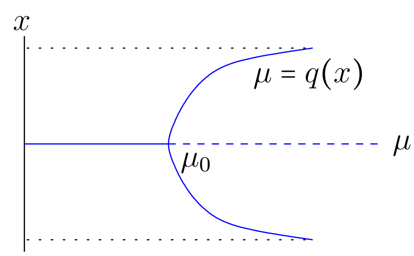


Figure 2.10: A typical period doubling bifurcation diagram for a discrete map

Remark 2.10.16
In the previous theorem, the condition $f_\mu(0) = 0$ for all μ near μ_0 is not necessary. Suppose that

1. $f_{\mu_0}(x_0) = x_0$,
2. $\left.\frac{\partial}{\partial x} f_\mu(x)\right|_{\mu=\mu_0,x=x_0} = -1$,
3. $\frac{1}{2}\left(\left.\frac{\partial^2}{\partial x^2} f_\mu(x)\right|_{\mu=\mu_0,x=x_0}\right)^2 + \left.\frac{1}{3}\frac{\partial^3}{\partial x^3} f_\mu(x)\right|_{\mu=\mu_0,x=x_0} \neq 0$ and
4. $\left.\frac{\partial^2}{\partial \mu \partial x} f_\mu^2(x)\right|_{\mu=\mu_0,x=x_0} \neq 0$.

From Theorem 2.10.13, there exists a function p defined in an open interval J of ν_0 such that $p(\mu_0) = x_0$ and $p(\mu)$ is a fixed point of f_μ for all $\mu \in J$. Let $\hat{f}(x,\mu) \equiv f(x + p(\mu), \mu) - p(\mu)$. We show that $\hat{f}$ satisfies the hypotheses of Theorem 2.10.15.

We have $\hat{f}_\mu(0) = 0$ for all μ near μ_0. Since

$$\frac{\partial^n \hat{f}}{\partial x^n}(x,\mu) = \frac{\partial^n f}{\partial x^n}(x+p(\mu),\mu)$$

for $n = 1, 2, \ldots$, we have

$$\frac{\partial}{\partial x}\hat{f}_\mu(x)\bigg|_{\mu=\mu_0,x=0} = \frac{\partial \hat{f}}{\partial x}(0,\mu_0) = \frac{\partial f}{\partial x}(0+p(\mu_0),\mu_0) = \frac{\partial f}{\partial x}(x_0,\mu_0) = \frac{\partial}{\partial x}f_\mu(x)\bigg|_{\mu=\mu_0,x=x_0} = -1 \ .$$

Moreover,

$$\begin{aligned}
\frac{1}{2}\left(\frac{\partial^2}{\partial x^2}\hat{f}_\mu(x)\bigg|_{\mu=\mu_0,x=0}\right)^2 &+ \frac{1}{3}\frac{\partial^3}{\partial x^3}\hat{f}_\mu(x)\bigg|_{\mu=\mu_0,x=0} = \frac{1}{2}\left(\frac{\partial^2 \hat{f}}{\partial x^2}(0,\mu_0)\right)^2 + \frac{1}{3}\frac{\partial^3 \hat{f}}{\partial x^3}(0,\mu_0) \\
&= \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}(0+p(\mu_0),\mu_0)\right)^2 + \frac{1}{3}\frac{\partial^3 f}{\partial x^3}(0+p(\mu_0),\mu_0) = \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\right)^2 + \frac{1}{3}\frac{\partial^3 f}{\partial x^3}(x_0,\mu_0) \\
&= \frac{1}{2}\left(\frac{\partial^2}{\partial x^2}f_\mu(x)\bigg|_{\mu=\mu_0,x=x_0}\right)^2 + \frac{1}{3}\frac{\partial^3}{\partial x^3}f_\mu(x)\bigg|_{\mu=\mu_0,x=x_0} \neq 0 \ .
\end{aligned}$$

Finally,

$$\frac{\partial^2}{\partial\mu\partial x}\hat{f}^2_\mu(x)\bigg|_{\mu=\mu_0,x=0} = \frac{\partial^2}{\partial\mu\partial x}f^2_\mu(x)\bigg|_{\mu=\mu_0,x=x_0} \neq 1 \ . \tag{2.10.2}$$

To prove the first equality requires a little bit of work. From

$$\hat{f}^2_\mu(x) = f(f(x+p(\mu),\mu),\mu) - p(\mu) \ ,$$

we get

$$\frac{\partial}{\partial x}\hat{f}^2_\mu(x) = \frac{\partial}{\partial x}f(f(x+p(\mu),\mu),\mu) = \frac{\partial f}{\partial x}(f(x+p(\mu),\mu),\mu)\frac{\partial f}{\partial x}(x+p(\mu),\mu)$$

and

$$\begin{aligned}
\frac{\partial^2}{\partial\mu\partial x}\hat{f}^2_\mu(x) &= \frac{\partial^2}{\partial\mu\partial x}f(f(x+p(\mu),\mu),\mu) - p(\mu) \\
&= \frac{\partial}{\partial\mu}\left(\frac{\partial f}{\partial x}(f(x+p(\mu),\mu),\mu)\frac{\partial f}{\partial x}(x+p(\mu),\mu)\right) \\
&= \frac{\partial^2 f}{\partial x^2}(f(x+p(\mu),\mu),\mu)\left(\frac{\partial f}{\partial x}(x+p(\mu),\mu)\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu)\right. \\
&\quad \left.+\frac{\partial f}{\partial\mu}(x+p(\mu),\mu)\right)\frac{\partial f}{\partial x}(x+p(\mu),\mu) + \frac{\partial^2 f}{\partial x\partial\mu}(f(x+p(\mu),\mu),\mu)\frac{\partial f}{\partial x}(x+p(\mu),\mu) \\
&\quad + \frac{\partial f}{\partial x}(f(x+p(\mu),\mu),\mu)\left(\frac{\partial^2 f}{\partial x^2}(x+p(\mu),\mu)\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu) + \frac{\partial^2 f}{\partial x\partial\mu}(x+p(\mu),\mu)\right) \ .
\end{aligned}$$

Hence, using $p(\mu_0) = x_0$, $f(x_0,\mu_0) = x_0$ and $\dfrac{\partial f}{\partial x}(x_0,\mu_0) = \dfrac{\partial}{\partial x} f_\mu(x)\Big|_{\mu=\mu_0,x=x_0} = -1$, we get

$$\begin{aligned}
\frac{\partial^2}{\partial\mu\partial x}\hat{f}_\mu^2(x)\Big|_{\mu=\mu_0,x=0} &= \frac{\partial^2 f}{\partial x^2}(f(x_0,\mu_0),\mu_0)\left(\frac{\partial f}{\partial x}(x_0,\mu_0)\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu_0) + \frac{\partial f}{\partial \mu}(x_0,\mu_0)\right)\frac{\partial f}{\partial x}(x_0,\mu_0) \\
&+ \frac{\partial^2 f}{\partial x\partial\mu}(f(x_0,\mu_0),\mu_0)\frac{\partial f}{\partial x}(x_0,\mu_0) \\
&+ \frac{\partial f}{\partial x}(f(x_0,\mu_0),\mu_0)\left(\frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu_0) + \frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0)\right) \\
&= -\frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\left(-\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu_0) + \frac{\partial f}{\partial\mu}(x_0,\mu_0)\right) - \frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0) \\
&- \left(\frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\frac{\mathrm{d}p}{\mathrm{d}\mu}(\mu_0) + \frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0)\right) = -2\frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0) - \frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\frac{\partial f}{\partial\mu}(x_0,\mu_0)\ .
\end{aligned}$$

Moreover,

$$\begin{aligned}
\frac{\partial^2}{\partial\mu\partial x} f_\mu^2(x) &= \frac{\partial}{\partial\mu}\left(\frac{\partial f}{\partial x}(f(x,\mu),\mu)\frac{\partial f}{\partial x}(x,\mu)\right) \\
&= \frac{\partial^2 f}{\partial x^2}(f(x,\mu),\mu)\frac{\partial f}{\partial\mu}(x,\mu)\frac{\partial f}{\partial x}(x,\mu) + \frac{\partial^2 f}{\partial x\partial\mu}(f(x,\mu),\mu)\frac{\partial f}{\partial x}(x,\mu) \\
&+ \frac{\partial f}{\partial x}(f(x,\mu),\mu)\frac{\partial^2 f}{\partial x\partial\mu}(x,\mu)\ .
\end{aligned}$$

Hence, using $f(x_0,\mu_0) = x_0$ and $\dfrac{\partial f}{\partial x}(x_0,\mu_0) = \dfrac{\partial}{\partial x} f_\mu(x)\Big|_{\mu=\mu_0,x=x_0} = -1$, we get

$$\begin{aligned}
\frac{\partial^2}{\partial\mu\partial x} f_\mu^2(x)\Big|_{\mu=\mu_0,x=x_0} &= \frac{\partial^2 f}{\partial x^2}(f(x_0,\mu_0),\mu_0)\frac{\partial f}{\partial\mu}(x_0,\mu_0)\frac{\partial f}{\partial x}(x_0,\mu_0) \\
&+ \frac{\partial^2 f}{\partial x\partial\mu}(f(x_0,\mu_0),\mu_0)\frac{\partial f}{\partial x}(x_0,\mu_0) + \frac{\partial f}{\partial x}(f(x_0,\mu_0),\mu_0)\frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0) \\
&= -\frac{\partial^2 f}{\partial x^2}(x_0,\mu_0)\frac{\partial f}{\partial\mu}(x_0,\mu_0) - 2\frac{\partial^2 f}{\partial x\partial\mu}(x_0,\mu_0)
\end{aligned}$$

and this proves the first equality of (2.10.2). ♠

2.10.4 Logistic Map

This section is mainly about the logistic equation

$$x_{i+1} = f_\mu(x_i) = \mu x_i(1-x_i)\ .$$

We have already found the fixed points of f_μ with their stability in Examples 2.10.3 and 2.10.10. We have also looked at period points of period 2 for f_μ in Example 2.10.12. To illustrate the complex behaviour that discrete dynamical systems may have, we now present some results about the logistic map f_μ without giving the proofs. A good reference is [12]. Most of the results about the logistic map f_μ that we present in this section are true for mappings having a graph "similar" to the graph of f_μ on $I = [0, 1]$.

Figure 2.11 is the **bifurcation diagram** of f_μ. Namely, for "each" value of μ we plot the fixed points, periodic points, ... of f_μ.

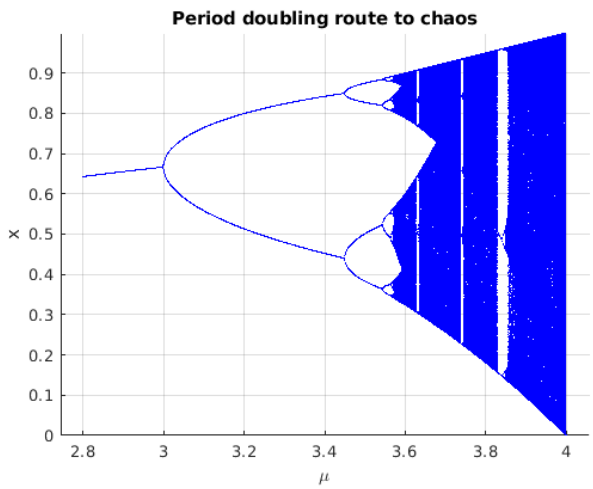


Figure 2.11: Bifurcation diagram for the logistic map

To produce this figure, we have chosen a large number of equally spaced values of μ. For each of these values of μ, we have computed the first 200 iterations of the orbit of 0.5 under f_μ and plotted only the last 80 or so iterations. By increasing the number of iterations to compute and plot, we could have generated a more precise bifurcation diagram.

For $\mu < 3$, all iterations converge to p_μ. When μ crosses above 3, the fixed point p_μ becomes a source and there appears an attracting periodic orbit of period 2; all orbits eventually "bounce back and for" between the two points of the orbit. We have period doubling at $\mu = 3$. This claim can be rigorously prove by showing that all hypothesis of Theorem 2.10.15 are satisfied. Effectively, we have

1. $f_\mu(p_\mu) = p_\mu$ for all μ near $\mu = 3$.

2. $\displaystyle \frac{\partial}{\partial x} f_\mu(x)\bigg|_{\mu=3,x=p_3} = -1.$

3. $\displaystyle \frac{1}{2}\left(\frac{\partial^2}{\partial x^2} f_\mu(x)\bigg|_{\mu=3,x=p_3}\right)^2 + \frac{1}{3}\frac{\partial^3}{\partial x^3} f_\mu(x)\bigg|_{\mu=3,x=p_3} = 2\mu^2\bigg|_{\mu=3} = 18 \neq 0$

4. $\displaystyle \frac{\partial^2}{\partial\mu\partial x} f_\mu^2(x)\big|_{\mu=3,x=p_3} = \left(2\mu - 6\mu^2 x + 18\mu^2 x^2 - 4\mu x - 12\mu^2 x^3\right)\big|_{\mu=3,x=p_3} = 2 \neq 0.$

When μ crosses above $3.236\ldots$, the periodic orbit of period 2 becomes unstable and there appears an attracting periodic orbit of period 4. For μ slightly larger, the periodic orbit of period 4 becomes unstable and there is a bifurcation from the attracting periodic orbit of period 4 to an attracting periodic orbit of period 8, And so on. This is the best known example of a **period doubling cascade**.

For a constant value of μ, if the attracting periodic orbit $\mathcal{O} \subset [0,1]$ of f_μ is of period n with n very large, it is reasonable to expect that $\mathcal{O}$ will "almost cover" some segments of the interval $[0,1]$. This explains the shaded area. Figure 2.12 illustrates the attracting periodic orbit for $\mu = 3.6$. The corresponding histogram with 300 subintervals and 10 millions iterations is given in Figure 2.13.

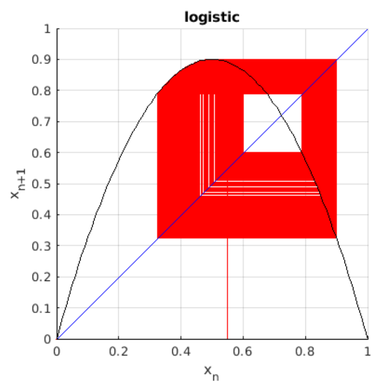


Figure 2.12: Cobweb of a periodic orbit of the logistic map for $\mu = 3.6$. The period of this stable periodic orbit is very large.

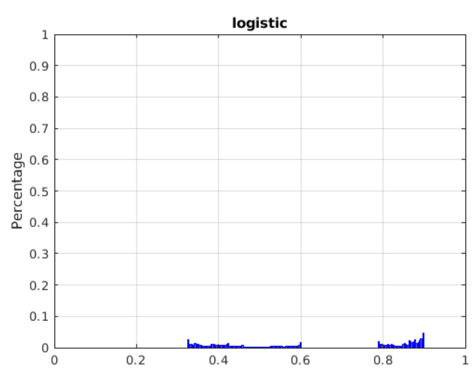


Figure 2.13: Histogram of a periodic orbit of the logistic map for $\mu = 3.6$. The period of this stable periodic orbit is very large.

Period doubling accumulates to $\mu = 3.5699456\ldots$. This number is known as the **Feigenbaum point**. Moreover, let $\mu_0 = 3$, $\mu_1 = 3.236\ldots$, μ_2, μ_3, $\ldots$ be the values of μ for which the logistic mapping undergoes period doubling and let $d_j = \mu_{j+1} - \mu_j$. It has been showed that $\lim_{j\to\infty} \dfrac{d_j}{d_{j+1}} = 4.6692016091029\ldots$. This number is called the **Feigenbaum constant**. Feigenbaum discovered this number in 1975. This constant is universal in the sense that it is the same for a whole class of dynamical systems of the form $x_{n+1} = g_\mu(x_n)$ where the graph of $g_\mu(x)$ looks like the graph of $f_\mu(x)$.

Period doubling is far from being the must complex type of bifurcation. To understand the complex behaviour of the orbits of f_μ, we need the following theorem.

Theorem 2.10.17 (Sarkovskii)

Consider the order on the positive integers defined by

$$\begin{aligned}&3 \gg 5 \gg 7 \gg \ldots \gg 3\times 2 \gg 5\times 2 \gg \ldots \\ &\qquad \gg 3\times 2^2 \gg 5\times 2^2 \gg \ldots \gg 3\times 2^3 \gg 5\times 2^3 \gg \ldots 2^4 \gg 2^3 \gg 2^2 \gg 2 \gg 1\,.\end{aligned}$$

Let $f:\mathbb{R}\to\mathbb{R}$ be a continuous function and k be a prime number. If f has a periodic point of period k, then f has a periodic point of period m for all $m \ll k$.

Example 2.10.18
For $\mu = 3.839\ldots$, $\{0.14988539433432, 0.48917380192271, 0.95930024021836\}$ is an attracting periodic orbit of period 3 of f_μ, where all value have been chopped to 14 digits after the decimal point (Figure 2.14).

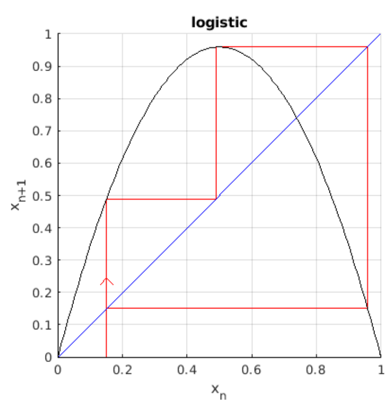


Figure 2.14: Period Three

Hence, $f_{3.839\ldots}$ has periodic orbits of all possible periods. All the periodic orbits, except the one of period 3, are unstable (in fact repelling). So, unless the iteration starts with a point on a repelling periodic orbit, the iterations will converge toward the periodic orbit of period 3. ♣

This is not the end of the story. We now consider f_μ for $\mu > 4$.

Let

$$A_n = \left\{x \in [0,1] : f_\mu^j(x) \in [0,1] \text{ for } 0 \le j \le n \text{ and } f_\mu^j(x) \notin [0,1] \text{ for } j > n\right\} \ .$$

It can be shown that A_n consists of 2^n distinct open subintervals of $[0,1]$. We have that any iteration that starts with $x_0 \in A_0$ (i.e. such that $f_\mu(x_0) > 1$) eventually converges to $-\infty$ (Figure 2.15). We also have that any iteration that starts with $x_0 \in A_1$ (i.e. such that $f_\mu(x_0) \in A_0$) eventually converges to $-\infty$ (Figure 2.16). And so on.

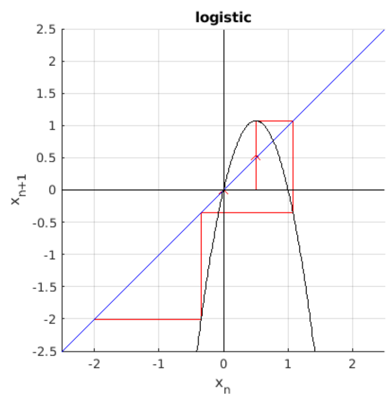


Figure 2.15: $0.5 \in A_0$ for $f_{4.3}$

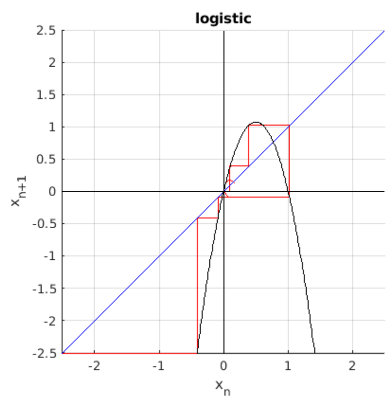


Figure 2.16: $0.1 \in A_1$ for $f_{4.3}$

Consider

$$\Delta_\mu = [0,1] \setminus \bigcup_{n=0}^{\infty} A_n \ .$$

Δ_μ contains all the points x such that $f_\mu^j(x)$ stay in $[0,1]$ for all $j \ge 0$. In particular, Δ_μ contains all the periodic points.

Recall that a **Cantor set** is a set that is closed (contains all its limit points), totally disconnected (does not contain any open interval), and perfect (every point of the set is the limit of other points of the set).

Example 2.10.19
The best known example of a Cantor set is the Cantor Middle-Thirds set. It is also an example of a **Fractal** set because of its self-similarity under zooming. ♣

Theorem 2.10.20

For $\mu > 4$, Δ_μ is a cantor set.

2.10.5 Chaos

The next two definitions are the bases for the definition of chaos.

Definition 2.10.21

Let I be an interval of $\mathbb{R}$ and $f : I \to I$ be a continuous function. f is **topologically transitive** if for any open sets V and W in I there exist $k > 0$ such that $f^k(V) \cap W \neq \emptyset$.

Definition 2.10.22

Let I be an interval of $\mathbb{R}$ and $f : I \to I$ be a continuous function. f has **sensitive dependence on initial conditions** if there exists $\delta > 0$ such that, for any $x \in I$ and neighbourhood $N \subset I$ of x, there exist $y \in N$ and $k > 0$ satisfying $|f^k(x) - f^k(y)| > \delta$.

Definition 2.10.23 (Chaos)

Let I be an interval of $\mathbb{R}$ and $f : I \to I$ be a continuous function. f is said to be **chaotic** on I if

1. f has sensitive dependence on initial conditions.
2. f is topologically transitive.
3. The set of all periodic points of f is dense in I (every non-periodic point of I is the limit of some periodic points).

Remark 2.10.24
It has been proved in [4] that 2 and 3 implies 1. Nevertheless, we keep the tradition of using Definition 2.10.23 as the definition of chaos because it lists three of the must important properties of a chaotic function. Moreover, it has been proved in [3] that 1 and 3 do not imply 2, and 1 and 2 do not imply 3. ♠

Example 2.10.25
The logistic map $f_4 : I \to I$, where $I = [0, 1]$, is chaotic. ♣

We may expand our definition of attracting and repelling periodic orbits to more general sets.

Definition 2.10.26

Let V be a subset of $\mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ be a continuous function. V is an **attracting** (respectively **repelling**) **hyperbolic set** if

1. V is closed and bounded.
2. V is **invariant** under f (i.e. $f(V) \subset V$).
3. There exists $N > 0$ such that $\left|\frac{\mathrm{d}f^n}{\mathrm{d}x}(x)\right| < 1$ (respectively > 1) for all $n \geq N$ and $x \in V$.

Example 2.10.27
It can be proved that for $\mu > 2 + \sqrt{5}$, Δ_μ is a repelling hyperbolic set for the logistic map f_μ. The behaviour of $f_\mu : \Delta_\mu \to \Delta_\mu$ is a lot more complex than we may imagine. f_μ has a dense orbit in Δ_μ. Moreover, $f_\mu : \Delta_\mu \to \Delta_\mu$ is choatic[1]. ♣

2.11 Exercises

Question 2.1
Find small intervals containing the solutions (one solution per interval) of $4x^2 - e^x = 0$. Do not forget to justify your answer.

Question 2.2
Use the bisection method to find an approximation of $\sqrt[3]{25}$ correct to within 10^{-4}.

Question 2.3
In the algorithm for the bisection method, Algorithm 2.2.1, if $a_0 > 0$ and

$$n \geq \frac{\ln(b_0 - a_0) - \ln(\epsilon) - \ln(a_0)}{\ln(2)} \tag{2.11.1}$$

Show that the n^{th} iteration x_n of the bisection method is an approximation of a root r with a relative error smaller than ϵ.

Question 2.4
In the algorithm for the bisection method, Algorithm 2.2.1, show that $|x_n - x_{n+1}| = 2^{-n-1}(b_0 - a_0)$.

Question 2.5
In the algorithm for the bisection method, Algorithm 2.2.1, is it possible to have $a_n < a_{n+1}$ (strict inequalities) for all n? If it is possible, give the conditions under which it is possible. If it is not possible, prove it.

[1] The definition of choatic map can be extended to any topological space I, not just intervals.

Question 2.6
Find a solution accurate to within 10^{-4} for $x = \tan(x)$ on $\pi/2 < x < 3\pi/2$.
a) Use the bisection method.
b) Use the fixed point method.
c) Use Newton's method.

Question 2.7
Find the solutions accurate to within 10^{-5} for $x^2 + 11\cos(x) = 0$.
a) Use the bisection method.
b) Use the fixed point method.
c) Use Newton's method.

Question 2.8
Find the smallest value $x_0 > 0$ such that the Newton's method for $f(x) = \arctan(x)$ does not converge.

Question 2.9
For which functions f is the iterative equation

$$x_{n+1} = 2x_n - Cx_n^2$$

the result of the formula for the Newton's method? C is a constant.

Question 2.10
If $x_0 = 0$ and

$$x_{n+1} = x_n - \frac{\tan(x_n) - 1}{\sec^2(x_n)}$$

for $n = 0,\ 1,\ 2,\ \ldots$ Without doing any computation, find the limit of this sequence.

Question 2.11
Use Newton's method to find an approximation of a root of $f(x) = \tan(x)$ in the interval $[4.8, 7.7]$ with an accuracy of 10^{-8}.

Question 2.12
Use the secant method to find an approximation of the first positive root of $f(x) = e^x - \tan(x)$ with an accuracy of 10^{-8}.
Hint: To choose x_0 and x_1, draw the graph of e^x and $\tan(x)$.

Question 2.13
a) Suppose that the Newton's method is used to generate a sequence $\{x_n\}_{n=0}^{\infty}$ converging to a root r of a function f. Let $e_n = x_n - r$. Show that

$$e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)}\, e_n^2$$

for some ξ_n between r and x_n.
b) Let $f(x) = x - e^{-x}$ and assume that the Newton's method is used to generate a sequence $\{x_n\}_{n=0}^{\infty}$ converging to the root r of f in the interval $[0,1]$. If $e_n = x_n - r$, show that

$$|e_n| \le 2\left(\frac{e_0}{2}\right)^{2^n} \tag{2.11.2}$$

for $n \geq 0$ whenever $x_0 \geq 0$.
c) If $x_0 = 1$ in (b), how many iterations of the Newton's method will be sufficient to get an approximation of the root r of f with an accuracy of 10^{-5}; namely, such that $|e_n| < 10^{-5}$.

Question 2.14
Use Newton's method with Horner's Algorithm to approximate the three roots of $f(x) = x^3 - x$; namely, to approximate $p_1 = -1$, $p_2 = 0$ and $p_3 = 1$. For each value of i, can you find a subinterval I_i of $[-0.451, -0.446]$ such that the Newton's method with Horner's Algorithm converges to the root p_i of $f(x)$? For each i, the subsets of the real line containing the points x_0 such that the Newton's method converges to p_i form a Cantor type of set.

Question 2.15
Suppose that $g : [a, b] \to [a, b]$ satisfies the Fixed Point Theorem and $g'(x) < 0$ for all $x \in [a, b]$. Describe the behaviour of the sequence $\{x_n\}_{n=0}^{\infty}$ given by $x_{n+1} = g(x_n)$ for $x_0 \in [a, b]$ as it converges to the fixed point. You may want to sketch a typical graph.

Question 2.16
Let $g(x) = \dfrac{1}{x^2 + 1}$.
a) Show that g has a unique fixed point in the interval $[0, 1]$.
b) Show that we can use the Fixed Point Theorem to find the fixed point of g in the interval $[0, 1]$.
c) Determine the order of convergence of this fixed point method.

Question 2.17
Consider the function $g(x) = 2^{-x}$.
a) Show that you can use the Fixed Point Theorem to approximate the fixed point of g in the interval $[1/3, 1]$.
b) Find a small value of n ensuring to the approximation x_n of the fixed point of g has an accuracy of 10^{-4}. You may assume that $x_0 = 0.5$.
c) Use the Fixed Point Theorem to find an approximation x_{n+1} to the fixed point of g in the interval $[1/3, 1]$ such that $|x_{n+1} - x_n| < 10^{-4}$. As in (b), you may assume that $x_0 = 0.5$.

Question 2.18
Consider the function
$$g(x) = 12 - \frac{20}{x} \ .$$
a) Explain why this function has two fixed points.
b) Using the Fixed Point Theorem, show that g has a unique fixed point p in the interval $[9.5, 11.5]$, and that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ converge to p whatever the choice $x_0 \in [9.5, 11.5]$.
c) How many iterations are needed to get an approximation of the fixed point of g in the interval $[9.5, 11.5]$ with an accuracy of 10^{-7}? You may assume that $x_0 = 9.5$.
d) What is the order of convergence of the sequence $\{x_n\}_{n=0}^{\infty}$ that is generated by $x_{n+1} = g(x_n)$ with x_0 in $[9.5, 11.5]$.
e) Use Steffensen's Algorithm to find an approximation of the fixed point of g in the interval $[9.5, 11.5]$ with an accuracy of 10^{-7}. Use $x_0 = 9.5$. Why does this method converge faster than the fixed point method?

Question 2.19
Let $f(x) = e^x - 2x - 1$.

a) Show that f has a unique root in the interval $[1,2]$.
b) Show that a root of f is a fixed point of $g(x) = \ln(1+2x)$ and vice-versa.
c) Show that for any $x_0 \in [1,2]$, the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ for $n = 0$, 1, 2, ... converges to the fixed point of g in the interval $[1,2]$.
d) Determine the order of convergence of this fixed point method.

Question 2.20
Our goal is to approximate the value of $\sqrt[3]{25}$ using the Fixed Point Theorem.
a) Show that $\sqrt[3]{25}$ is the unique root of $f(x) = x^3 - 25$.
b) Show that $p > 0$ is a root of f if and only if p is a fixed point of $g(x) = 5/\sqrt{x}$, and conclude that this fixed point p is $\sqrt[3]{25}$.
c) Using the graph of g, give an interval $[a,b]$ with $a > 0$ such that g satisfies the Fixed Point Theorem on $[a,b]$. Verify that g satisfies all the hypotheses of the Fixed Point Theorem.
d) Choose x_0 in the interval $[a,b]$ that you have found in (c). Without doing any iteration, find a small value of n such that x_n, the $(n+1)^{th}$ term in the sequence $\{x_n\}_{n=0}^{\infty}$ produced by the Fixed Point Theorem applied to the function g, is an approximation of p with an accuracy of 10^{-5}.
e) Use the Fixed Point Theorem to find an approximation x_n to the fixed point of g in the interval $[a,b]$ such that $|x_n - x_{n-1}| < 10^{-5}$. Use the x_0 that you have chosen in (d).

Question 2.21
Let $f(x) = e^{2-x} - x^2$.

a) Show that f has a unique root p in the interval $[1,2]$.
b) Find a function g satisfying all the hypotheses of the Fixed Point Theorem such that a root of f in the interval $[1,2]$ is a fixed point of g in $[1,2]$. Verify that your function g satisfies the hypotheses of the Fixed Point Theorem.
c) Let $x_0 = 1$. Without doing any iteration, find a small value of n such that x_n, the $(n+1)^{th}$ term in the sequence $\{x_n\}_{n=0}^{\infty}$ produced by the Fixed Point Theorem applied to the function g, is an approximation of p with an accuracy of 10^{-5}.
d) Determine the order of convergence of this fixed point method.

Question 2.22
The first positive value p such that $p = \tan(p)$ is between π and $3\pi/2$. Let

$$g(x) = \pi + \arctan(x) \ .$$

a) Show graphically that g has a unique fixed point in $[\pi, 3\pi/2]$ and that it is the point p above.
b) Show that g satisfies the hypotheses of the Fixed Point Theorem on the interval $[\pi, 3\pi/2]$.
c) Without doing any iteration, find a small value of n such that x_n, the $(n+1)^{th}$ term in the sequence $\{x_n\}_{n=0}^{\infty}$ produced by the Fixed Point Theorem applied to the function g, is an approximation of p with an accuracy of 10^{-5}.

Question 2.23

Let a be a positive number and

$$g(x) = \frac{x}{2} + \frac{a}{2x} \ . \tag{2.11.3}$$

Given $x_0 > 0$, let $\{x_n\}_{n=0}^{\infty}$ be the sequence generated by $x_{n+1} = g(x_n)$ for $n = 0, 1, 2, \ldots$

a) Show that the positive fixed point of g is $\sqrt{a}$.
b) Use the Fixed Point Theorem to prove that for any $x_0 > 0$ the sequence $\{x_n\}_{n=0}^{\infty}$ converges to the unique positive fixed point of f.
Hint: Show first that if $0 < x_0 < \sqrt{a}$, then $x_1 \geq \sqrt{a}$. Then show that g satisfies the Fixed Point Theorem on any interval of the form $[\sqrt{a}, m]$.

Question 2.24
a) If f' is continuous and positive on $[a,b]$, and $f(a)f(b) < 0$, prove that f has a unique zero in the open interval $]a,b[$.
b) Find λ such that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by the iteration $x_{n+1} = x_n + \lambda f(_n)$ for $n = 0, 1, 2, \ldots$ converges to a zero of f.

Question 2.25
Suppose that g is a continuously differentiable function on an interval $[a,b]$. Let $m = (a+b)/2$ be the middle point of the interval $[a,b]$. If $|g'(x)| < 1$ for all $x \in [a,b]$ and $g(m) = m$, prove or disprove that the sequence $\{x_n\}_{n=0}^{\infty}$ defined by $x_{n+1} = g(x_n)$ converges to the fixed point m of g in $[a,b]$ whatever the choice of $x_0 \in [a,b]$.

Question 2.26
Suppose that $g : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable function and that p is a fixed point of g such that $|g'(p)| > 1$. Prove or disprove that for all $x_0 \in \mathbb{R}$ the sequence $\{x_n\}_{n=0}^{\infty}$ does not converge to p. If there are sequences $\{x_n\}_{n=0}^{\infty}$ that converge to p, describe all of them.

Question 2.27
Suppose that $|g'(x)| \leq \lambda < 1$ for all $x \in [x_0 - \rho, x_0 + \rho]$, where $\rho = \dfrac{|g(x_0) - x_0|}{1 - \lambda}$. Prove that the sequence $\{x_n\}_{n=0}^{\infty}$ defined by $x_{n+1} = g(x_n)$ for $n \geq 0$ converges to a fixed point of g in the interval $[x_0 - \rho, x_0 + \rho]$.

Question 2.28
Prove or disprove that if f is a contraction on $[a,b]$, then f has a unique fixed point and the iterative system $x_{n+1} = f(x_n)$ for $n \geq 0$ yields a sequence $\{x_n\}_{n=0}^{\infty}$ that converges toward this root whatever the choice of $x_0 \in [a,b]$.

Question 2.29
Suppose that f is m times continuously differentiable. Show that $f(x) = (x-p)^m q(x)$ with $q(p) \neq 0$ if and only if $f(p) = f'(p) = \ldots = f^{(m-1)}(p) = 0$ and $f^{(m)}(p) \neq 0$; namely, if and only if f has a zero of multiplicity m at p.

Question 2.30
Let $F(x) = x - f(x)f'(x)$, where f is a three times continuously differentiable function satisfying $f(r) = 0$ and $f'(r) \neq 0$. Find the conditions on f to obtain an iterative method $x_{n+1} = F(x_n)$ for $n \geq 0$ that generates sequences converging toward r and such that the convergence is of order exactly three.

Question 2.31

Let $F(x) = x + f(x)g(x)$, where f and g are sufficiently continuously differentiable functions. Moreover, assume that f satisfies $f(r) = 0$ and $f'(r) \neq 0$. Find the conditions on g to obtain an iterative method $x_{n+1} = F(x_n)$ for $n \geq 0$ that generates sequences converging toward r and such that the order of convergence is exactly three.

Question 2.32
Which of the following sequences converge quadratically?

a) $\left\{\frac{1}{n^2}\right\}_{n=1}^{\infty}$ **b)** $\left\{\frac{1}{2^{2^n}}\right\}_{n=0}^{\infty}$ **c)** $\left\{\frac{1}{\sqrt{n}}\right\}_{n=1}^{\infty}$
d) $\left\{\frac{1}{e^n}\right\}_{n=0}^{\infty}$ **e)** $\left\{\frac{1}{n^n}\right\}_{n=1}^{\infty}$

Question 2.33
a) Show that the convergence of the sequence $p_n = 10^{-k^n}$ to 0 is of order k.
b) Show that the sequence $p_n = 10^{-n^k}$ does not converge to 0 quadratically regardless of the size of the exponent $k > 1$.

Question 2.34
Solve $x - 2^{-x} = 0$ for $x \in [0, 1]$ with an accuracy of 10^{-4} using Steffensen's Algorithm.

Question 2.35
Use the method of deflation to approximate all the roots of

$$p(x) = x^3 - 5.974925987\,x^2 + 9.734512519\,x - 2.617993878$$

with an accuracy of 10^{-10}. Do not used any formula to computer the roots of a polynomial of degree two.

Question 2.36
Use the method of deflation to approximate all the roots of $p(x) = x^4 - 2x^3 - 12x^2 + 16x - 40$ with an accuracy of 10^{-9}. You must use Newton's method with Horner's Algorithm.

Question 2.37
Use the method of deflation to approximate all the roots of $x^3 - 53x^2 + 151x - 3$ with an accuracy of 10^{-3}. You must use Newton's method with Horner's Algorithm.

Question 2.38
Use the method of deflation to approximate all the roots of

$$x^4 - 10.07251864x^3 + 34.83068793x^2 - 44.63745043x + 11.36978427$$

with an acuracy of 10^{-5}. You must use Newton's method with Horner's Algorithm.

Chapter 3

Iterative Methods to Solve Systems of Linear Equations

Our goal is to numerically solve the system of linear equations

$$A\mathbf{x} = \mathbf{b} \ , \tag{3.0.1}$$

where

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \quad , \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \ . \tag{3.0.2}$$

We assume that A is an invertible matrix. Hence (3.0.1) has a unique solution.

In this section, we do not attempt to solve (3.0.1) using Gauss elimination and related direct methods. This is the subject of the next chapter. Instead, we develop iterative methods as we have done to numerically find the roots of real-valued functions. We therefore have to define properly the convergence of vectors and matrices. This is done in the next section.

3.1 Norm and Convergence of Matrices

Definition 3.1.1

A **norm** on a vector space V over the real numbers is a function $N : V \to \mathbb{R}$ satisfying

1. $N(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in V$.
2. $N(\mathbf{x}) = 0$ if and only if $\mathbf{x} = \mathbf{0}$.
3. $N(\alpha\mathbf{x}) = |\alpha| N(\mathbf{x})$ for all $\mathbf{x} \in V$ and $\alpha \in \mathbb{R}$.
4. $N(\mathbf{x} + \mathbf{y}) \leq N(\mathbf{x}) + N(\mathbf{y})$ for all $\mathbf{x}$ and $\mathbf{y}$ in V.

Remark 3.1.2
Three important norms on $V = \mathbb{R}^n$ are the **Euclidean or ℓ^2 norm**

$$N(\mathbf{x}) \equiv \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \ ,$$

the **maximum or ℓ^∞ norm**

$$N(\mathbf{x}) \equiv \|\mathbf{x}\|_\infty = \max_{1\le i\le n} |x_i|$$

and the ℓ^1 **norm**

$$N(\mathbf{x}) \equiv \|\mathbf{x}\|_1 = \sum_{1}^{n} |x_i| \ .$$

♠

Definition 3.1.3

Let $\|\cdot\|$ be any norm on $\mathbb{R}^n$. The **distance** between two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, denoted $d(\mathbf{x},\mathbf{y})$, is defined by $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|$.

Definition 3.1.4

A sequence of vectors $\{\mathbf{x}_k\}_{k=1}^{\infty}$ in $\mathbb{R}^n$ **converges** to a vector $\mathbf{p}$ in $\mathbb{R}^n$ if $\lim_{k\to\infty} \|\mathbf{x}_k - \mathbf{p}\| = 0$.

Remark 3.1.5

1. The definition of convergence in a finite dimensional vector space V does not depend on the chosen norm. It is shown in [17] that for any two norms N_1 and N_2 on V there exist constants c_1 and c_2 such that

   $$c_1 N_1(\mathbf{x}) \le N_2(\mathbf{x}) \le c_2 N_1(\mathbf{x})$$

   for all vector $\mathbf{x} \in \mathbb{R}^n$. For instance, we have

   $$\|\mathbf{x}\|_\infty \le \|\mathbf{x}\|_2 \le \sqrt{n}\|\mathbf{x}\|_\infty \ .$$

2. One can also show that $\{\mathbf{x}_k\}_{k=0}^{\infty}$ converges to $\mathbf{x}$ if and only if $\{x_{k,j}\}_{k=0}^{\infty}$ converges to x_j for $1 \le j \le n$, where x_j is the j^{th} component of the vector $\mathbf{x}$ and $x_{k,j}$ is the j^{th} component of the vector $\mathbf{x}_k$.

♠

Definition 3.1.6

Let $\|\cdot\|$ be any norm on $\mathbb{R}^n$ and A be an $n\times n$ matrix. The **natural** or **induced matrix norm** of A is defined by

$$\|A\| = \sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| \ .$$

Remark 3.1.7

1. The reader is invited to verify that the induced matrix norm satisfies the properties of a norm on the space V of $n \times n$ matrices. We note that the space V of $n \times n$ matrices is linearly isomorphic to $\mathbb{R}^{n^2}$ and so is of finite dimension.
2. It is easy to see that $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{R}^n$. This shows that the mapping $\phi : \mathbb{R}^n \to \mathbb{R}^n$ defined by $\phi(\mathbf{x}) = A\mathbf{x}$ for all $\mathbf{x}$ is a continuous mapping.
3. Since $S = \{\mathbf{x} : \|\mathbf{x}\| = 1\}$ is a compact subset of $\mathbb{R}^n$, the continuous mapping ϕ defined in the previous item reaches its maximum on S at a point in S. For this reason, we may replace sup by max in the definition of the induced norm.
4. If A and B are two $n \times n$ matrices, then $\|AB\| \leq \|A\|\|B\|$.

♠

Theorem 3.1.8

Let A be an $n \times n$ matrix as defined in (3.0.2). The norm of A induced by $\|\cdot\|_\infty$ is given by
$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ ,$$

Proof.
For $\mathbf{x} \in \mathbb{R}^n$ satisfying $\|\mathbf{x}\|_\infty = \max\limits_{1\leq s\leq n} |x_s| = 1$, we have
$$\|A\mathbf{x}\|_\infty = \max_{1\leq i\leq n} \Big|\sum_{j=1}^{n} a_{i,j}x_j\Big| \leq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}||x_j| \leq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ ,$$
where the last inequality is a consequence of $|x_j| \leq \max\limits_{1\leq s\leq n} |x_s| = 1$ for all j. Thus
$$\|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty \leq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ .$$

To prove equality, suppose that k is the index such that
$$\sum_{j=1}^{n} |a_{k,j}| = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ .$$

Define $\mathbf{x} \in \mathbb{R}^n$ by
$$x_j = \begin{cases} 1 & \text{if } a_{k,j} \geq 0 \\ -1 & \text{if } a_{k,j} < 0 \end{cases}$$

Then $\|\mathbf{x}\|_\infty = 1$ and
$$\|A\mathbf{x}\|_\infty = \max_{1\leq i\leq n} \Big|\sum_{j=1}^{n} a_{i,j}x_j\Big| \geq \Big|\sum_{j=1}^{n} a_{k,j}x_j\Big| = \sum_{j=1}^{n} |a_{k,j}| = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ .$$

Thus
$$\|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty \geq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}| \ .$$

∎

Remark 3.1.9
If A is an $n \times n$ matrix, let A^* be the transpose complex conjugate of A. It is usually proved in applied linear algebra that

$$\|A\|_2 = \max\{\sqrt{|\lambda|} : \lambda \text{ is an eigenvalue of } A^*A\} .$$

♠

Definition 3.1.10

The **spectral radius** of a $n \times n$ matrix A, denoted $\rho(A)$, is defined by

$$\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\} .$$

Theorem 3.1.11

Let A be a $n \times n$ matrix, then $\rho(A) = \inf\{\|A\| : \|\cdot\| \text{ is an induced norm}\}$.

Remark 3.1.12
A consequence of Theorem 3.1.11 is that $\rho(A) \leq \|A\|$ for any induced norm $\|\cdot\|$ on the $n \times n$ matrices. ♠

To prove Theorem 3.1.11, we need the following lemma.

Lemma 3.1.13

Every $n \times n$ matrix A is conjugate to an upper-triangular matrix (possibly with complex elements) whose off-diagonal elements can be arbitrary small.

Proof (of the lemma).
From Schur's Theorem, there exists an invertible matrix Q such that

$$QAQ^{-1} = T \equiv \begin{pmatrix} t_{1,1} & t_{1,2} & \ldots & t_{1,n} \\ 0 & t_{2,2} & \ldots & t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & t_{n,n} \end{pmatrix} .$$

Choose $\epsilon > 0$ and let

$$D = \begin{pmatrix} \epsilon & 0 & \ldots & 0 \\ 0 & \epsilon^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \epsilon^n \end{pmatrix} .$$

Then

$$(DQ)A(DQ)^{-1} = DQAQ^{-1}D^{-1} = DTD^{-1} = U ,$$

where

$$u_{i,j} = \begin{cases} \epsilon^{i-j} t_{i,j} & \text{for} \quad j \geq i \\ 0 & \text{for} \quad j < i \end{cases}$$

Since $u_{i,j} = \epsilon^{i-j} t_{i,j} \to 0$ as $\epsilon \to 0$ for all $j > i$, the off-diagonal elements can be arbitrary small. ∎

Proof (of Theorem 3.1.11).
A) We prove first that $\rho(A) \leq \|A\|$ for any induced norm $\|\cdot\|$ on the $n \times n$ matrices.

Let λ be an eigenvalue of A and $\mathbf{x}$ be an eigenvector associated to λ. We may assume that $\mathbf{x}$ is of norm 1. Then

$$|\lambda| = |\lambda| \|\mathbf{x}\| = \|\lambda \mathbf{x}\| = \|A\mathbf{x}\| \leq \|A\| \|\mathbf{x}\| = \|A\| \ .$$

Hence, $\rho(A) \leq \|A\|$.

Thus

$$\rho A \leq \inf \|\|A\| : \|\cdot\| \text{ is an induced norm}\} \ . \tag{3.1.1}$$

B) We construct induced norms $\|\cdot\|_\epsilon$ such that $\|A\|_\epsilon \leq \rho(A) + \epsilon$, where the parameter ϵ can be arbitrary small.

Choose $\epsilon > 0$. From the previous lemma, there exists an invertible matrix Q_ϵ such that $Q_\epsilon A Q_\epsilon^{-1} = D + S_\epsilon$, where D is a diagonal matrix whose elements on the diagonal are the eigenvalues of A and where S_ϵ is a strictly upper-triangular matrix whose elements are assumed to be small enough to get $\|S\|_\infty < \epsilon$. Hence

$$\|Q_\epsilon A Q_\epsilon^{-1}\|_\infty = \|D + S_\epsilon\|_\infty \leq \|D\|_\infty + \|S_\epsilon\|_\infty < \rho(A) + \epsilon$$

because

$$\|D\|_\infty = \max\{|d_{j,j}| : 1 \leq j \leq n\| = \rho(A) \ .$$

Since Q_ϵ is invertible, $\|\mathbf{x}\|_\epsilon = \|Q_\epsilon \mathbf{x}\|_\infty$ for $\mathbf{x} \in \mathbb{R}^n$ defines a norm on $\mathbb{R}^n$. The induced norm of A with respect to the norm $\|\cdot\|_\epsilon$ is

$$\begin{aligned} \|A\|_\epsilon &= \max_{\|\mathbf{x}\|_\epsilon = 1} \|A\mathbf{x}\|_\epsilon = \max_{\|Q_\epsilon \mathbf{x}\|_\infty = 1} \|Q_\epsilon A \mathbf{x}\|_\infty = \max_{\|Q_\epsilon \mathbf{x}\|_\infty = 1} \|Q_\epsilon A Q_\epsilon^{-1}(Q_\epsilon \mathbf{x})\|_\infty \\ &= \max_{\|\mathbf{y}\|_\infty = 1} \|Q_\epsilon A Q_\epsilon^{-1} \mathbf{y}\|_\infty = \|Q_\epsilon A Q_\epsilon^{-1}\|_\infty < \rho(A) + \epsilon \ . \end{aligned}$$

C) From $\|A\|_\epsilon < \rho(A) + \epsilon$, we get that

$$\inf\{\|A\| : \|\cdot\| \text{ is an induced norm}\} < \rho(A) + \epsilon \ .$$

Since ϵ is arbitrary small,

$$\inf\{\|A\| : \|\cdot\| \text{ is an induced norm}\} \leq \rho(A) \ .$$

Combined with (3.1.1), this proves the theorem. ∎

Theorem 3.1.14

Let A be a $n \times n$ matrix and $\|\cdot\|$ be an induced norm on the $n \times n$ matrices. The following statements are equivalent.

(i) $\|A^k\| = \|\underbrace{AA\ldots A}_{k \text{ times}}\|$ converges to zero as k goes to ∞.

(ii) $\rho(A) < 1$.

(iii) Given any $\mathbf{x} \in \mathbb{R}^n$, the sequence $\{A^k\mathbf{x}\}_{k=0}^{\infty}$ converges to $\mathbf{0} \in \mathbb{R}^n$.

Proof.
(i) ⇒ (iii)) Using item 2 of Remark 3.1.7, we have that

$$\|A^k\mathbf{x}\| \leq \|A^k\|\,\|\mathbf{x}\| \to 0$$

as $k \to \infty$ for all $\mathbf{x} \in \mathbb{R}^n$.

(iii) ⇒ (ii)) Suppose that $\rho(A) \geq 1$. There exists an eigenvalue λ such that $|\lambda| \geq 1$. Let $\mathbf{x}$ be an eigenvector associated to λ. We have

$$\|A^k\mathbf{x}\| = \|\lambda^k\mathbf{x}\| = |\lambda|^k\|\mathbf{x}\| \not\to 0$$

as $k \to \infty$. This is a contradiction of (iii).

(ii) ⇒ (i)) From Theorem 3.1.11, there exists an induced norm $\|\cdot\|_\epsilon$ such that $\|A\|_\epsilon < 1$ because $\rho(A) < 1$. From item 4 in Remark 3.1.7, we have $\|A^k\|_\epsilon \leq \|A\|_\epsilon^k$. From Remark 3.1.5, there exists a positive constant c such that $\|B\| \leq c\|B\|_\epsilon$ for all $n \times n$ matrices B since the linear space of $n \times n$ matrices is linearly isomorphic to the finite linear space $\mathbb{R}^{n^2}$. Hence,

$$\|A^k\| \leq c\|A^k\|_\iota \leq c\|A\|_\iota^k \to 0$$

as $k \to \infty$ because $\|A\|_\epsilon < 1$. ∎

3.2 Iterative Methods

3.2.1 Jacobi Iterative Method

Given a vector $\mathbf{x}_0 \in \mathbb{R}^n$, the goal is to generate a sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ that converges to the solution of (3.0.1).

Suppose that $a_{i,i} \neq 0$ for all i, then we can rewrite (3.0.1) as

$$x_i = \frac{1}{a_{i,i}}\left(b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{i,j}x_j\right)$$

for $i = 1, 2, \ldots, n$. This formula motivates the following algorithm.

Algorithm 3.2.1 (**Jacobi Iterative Method**)

1. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).

2. Given the vector $\mathbf{x}_k$, compute the vector $\mathbf{x}_{k+1}$ as follows:

$$x_{k+1,i} = \frac{1}{a_{i,i}}\left(b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{i.j}x_{k,j}\right) \tag{3.2.1}$$

   for $i = 1,\ 2,\ \ldots,\ n$.

3. Repeat (2) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

However, we need conditions on the matrix A to ensure that the sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ converges to a solution of $A\mathbf{x} = \mathbf{b}$. Sufficient conditions will be given shortly.

Code 3.2.2 (**Jacobi Iterative Method**)

To approximate the solution of the linear system $A\mathbf{x} = \mathbf{b}$.
Input: The matrix A.
The column vector $\mathbf{b}$.
The column vector $\mathbf{x}_0$ (denoted x in the code below).
The tolerance tol.
The maximal number of iterations allowed limit
Output: The approximation of the solution.

```
%  xx = jacobi(A,b,x,tol,limit)

function xx = jacobi(A,b,x,tol,limit)
  xx = NaN;
  dim = size(A,1);

  for k = 1:dim
    if ( A(k,k) == 0 )
      disp 'The Jacobi iterative method fails because some of the elements'
      disp 'on the diagonal are zero.'
      return;
    end
  end

  for k = 1:limit
    xx(1,1) = (b(1,1) - A(1,2:dim)*x(2:dim,1))/A(1,1);
    if dim > 2
      for m = 2:dim-1
        xx(m,1) = (b(m,1) - A(m,1:m-1)*x(1:m-1,1) - ...
                    A(m,m+1:dim)*x(m+1:dim,1))/A(m,m);
      end
```

```
    end
    xx(dim,1) = (b(dim,1) - A(dim,1:dim-1)*x(1:dim-1,1))/A(dim,dim);

    if ( norm(xx - x) < tol)
      disp(sprintf('Number of iterations = %d',k))
      return;
    end

    x=xx;
  end

  disp 'The Jacobi iterative method failed to give an approximation to a'
  disp 'solution of A x = b within the required accuracy and maximum'
  disp 'number of iterations allowed.'
  xx = NaN;
end
```

3.2.2 Gauss-Seidel Iterative Method

As for Jacobi iterative method, given a vector $\mathbf{x}_0 \in \mathbb{R}^n$, the goal is to generate a sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ that converges to the solution of (3.0.1).

If we use $x_{k+1,1}$, $x_{k+1,2}$, …, $x_{k+1,i-1}$ instead of $x_{k,1}$, $x_{k,2}$, …, $x_{k,i-1}$ in the formula (3.2.1) to compute $x_{k+1,i}$, hoping that $x_{k+1,1}$, $x_{k+1,2}$, …, $x_{k+1,i-1}$ are better approximations of the first $(i-1)$ coordinates of the solution of (3.0.1) than $x_{k,1}$, $x_{k,2}$, …, $x_{k,i-1}$, then perhaps that will get a new sequence $\{x_k\}_{k=0}^{\infty}$ that converges faster to the solution of (3.0.1). This motivates the following algorithm.

Algorithm 3.2.3 (**Gauss-Seidel Iterative Method**)

1. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).
2. Given the vector $\mathbf{x}_k$, compute the vector $\mathbf{x}_{k+1}$ as follows:
$$x_{k+1,i} = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i.j}x_{k+1,j} - \sum_{j=i+1}^{n} a_{i.j}x_{k,j}\right) \tag{3.2.2}$$
for $i = 1, 2, \ldots, n$.
3. Repeat (2) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

As for the Jacobi iterative method, we need conditions on the matrix A to ensure that the sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ converges to a solution of $A\mathbf{x} = \mathbf{b}$. Sufficient conditions will be given shortly.

Code 3.2.4 (Gauss-Seidel Iterative Method)

To approximate the solution of the linear system $A\mathbf{x} = \mathbf{b}$.
Input: The matrix A.
The column vector $\mathbf{b}$.
The column vector $\mathbf{x}_0$ (denoted x in the code below).
The tolerance tol.
The maximal number of iterations allowed limit
Output: The approximation of the solution.

```
%  xx = gausssiedel(A,b,x,tol,limit)

function xx = gausssiedel(A,b,x,tol,limit)
  xx = NaN;
  dim = size(A,1);

  for k = 1:dim
    if ( A(k,k) == 0 )
      disp 'The Gauss-Seidel iterative method fails because some of the'
      disp 'elements on the diagonal are zero.'
      return;
    end
  end

  for k = 1:limit
    xx(1,1) = (b(1,1) - A(1,2:dim)*x(2:dim,1))/A(1,1);
    if dim > 2
      for m = 2:dim-1
        xx(m,1) = (b(m,1) - A(m,1:m-1)*xx(1:m-1,1) - ...
                     A(m,m+1:dim)*x(m+1:dim,1))/A(m,m);
      end
    end
    xx(dim,1) = (b(dim,1) - A(dim,1:dim-1)*xx(1:dim-1,1))/A(dim,dim);

    if ( norm(xx - x) < tol)
        disp(sprintf('Number of iterations = %d',k))
        return;
    end

    x=xx;
  end

  disp 'The Gauss-Seidel iterative method failed to give an approximation'
  disp 'to a solution of  A x = b  within the required accuracy and'
  disp 'maximum number of iterations allowed.'
  xx = NaN;
end
```

3.2.3 Convergence of Iterative Methods

Let

$$D = \begin{pmatrix} a_{1,1} & 0 & \dots & 0 & 0 \\ 0 & a_{2,2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{n-1,n-1} & 0 \\ 0 & 0 & \dots & 0 & a_{n,n} \end{pmatrix} , \quad U = -\begin{pmatrix} 0 & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & 0 & a_{2,3} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$
$$\text{and} \qquad L = -\begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ a_{2,1} & 0 & 0 & \dots & 0 & 0 \\ a_{3,1} & a_{3,2} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n-1} & 0 \end{pmatrix} . \tag{3.2.3}$$

The equation $A\mathbf{x} = \mathbf{b}$ is equivalent to $(D - U - L)\mathbf{x} = \mathbf{b}$.

Hence, the formula (3.2.1) for the Jacobi iterative method can be rewritten as $\mathbf{x}_{k+1} = D^{-1}(L + U)\mathbf{x}_k + D^{-1}\mathbf{b}$. We have that $\mathbf{x}$ is a solution of $\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}$ if and only if $\mathbf{x}$ is a solution of $A\mathbf{x} = \mathbf{b}$.

As well, the formula (3.2.2) for the Gauss-Seidel iterative method can be rewritten as $\mathbf{x}_{k+1} = (D - L)^{-1}U\mathbf{x}_k + (D - L)^{-1}\mathbf{b}$. Again, $\mathbf{x}$ is a solution of $\mathbf{x} = (D - L)^{-1}U\mathbf{x} + (D - L)^{-1}\mathbf{b}$ if and only if $\mathbf{x}$ is a solution of $A\mathbf{x} = \mathbf{b}$.

Both the Jacobi iterative method and the Gauss-Seidel iterative method are of the form

$$\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c} \tag{3.2.4}$$

for $k = 0, 1, 2, \dots$

For the Jacobi iterative method, $T = D^{-1}(L + U)$ and $\mathbf{c} = D^{-1}\mathbf{b}$. For the Gauss-Seidel iterative method, $T = (D - L)^{-1}U$ and $\mathbf{c} = (D - L)^{-1}\mathbf{b}$.

We now find necessary and sufficient conditions for the convergence of methods of the form (3.2.4).

The following proposition will be used to justify the necessary and sufficient conditions for the convergence of (3.2.4) that will be given in Theorem 3.2.12.

Proposition 3.2.5

Let T be an $n \times n$ matrix. If $\rho(T) < 1$, then $\mathrm{Id}_n - T$ is invertible and

$$(\mathrm{Id}_n - T)^{-1} = \lim_{k\to\infty} (\mathrm{Id}_n + T + T^2 + \dots + T^k) = \lim_{k\to\infty} \sum_{j=0}^{k} T^j .$$

Proof.
Let $S_k = \mathrm{Id}_n + T + T^2 + \dots + T^k$. We have

$$S_k(\mathrm{Id}_n - T) = (\mathrm{Id}_n - T)S_k = \mathrm{Id}_n - T^{k+1} . \tag{3.2.5}$$

Since $\rho(T) < 1$, we get from Theorem 3.1.14 that $\lim_{k\to\infty} T^{k+1} = 0$. Hence,

$$\lim_{k\to\infty} (\mathrm{Id}_n - T^{k+1}) = \mathrm{Id}_n \ .$$

Since all eigenvalues λ of T satisfy $|\lambda| \leq \rho(T) < 1$, all eigenvalues of $\mathrm{Id}_n - T$ (which are of the form $1 - \lambda$ for λ an eigenvalue of T) are non-null. Thus $\mathrm{Id}_n - T$ is invertible.

From (3.2.5), we get

$$\lim_{k\to\infty} S_k = \lim_{k\to\infty} \left((\mathrm{Id}_n - T)^{-1}(I_n - T^{k+1}) \right) = (\mathrm{Id}_n - T)^{-1} \lim_{k\to\infty} (\mathrm{Id}_n - T^{k+1}) = (\mathrm{Id}_n - T)^{-1} \ .$$

We could pull $(\mathrm{Id}_n - T)^{-1}$ out of the limit above because, if $\{A_k\}_{k=1}^{\infty}$ is a sequence of $n \times n$ matrices converging to a matrix A and B is a $n \times n$ matrix, then $\{BA_k\}_{k=1}^{\infty}$ is a sequence of $n \times n$ matrices converging to BA since $\|BA_k - BA\| \leq \|B\| \, \|A_k - A\| \to 0$ as $k \to \infty$. ∎

Corollary 3.2.6

In Proposition 3.2.5, we have

$$\frac{1}{1 + \|T\|} \leq \|(\mathrm{Id}_n - T)^{-1}\| \leq \frac{1}{1 - \|T\|} \ . \tag{3.2.6}$$

Proof.
From $\mathrm{Id}_n = (\mathrm{Id}_n - T)(\mathrm{Id}_n - T)^{-1}$, we get

$$\begin{aligned} 1 = \|\mathrm{Id}_n\| = \|(\mathrm{Id}_n - T)(\mathrm{Id}_n - T)^{-1}\| &\leq \|\mathrm{Id}_n - T\| \, \|(\mathrm{Id}_n - T)^{-1}\| \\ &\leq (\|\mathrm{Id}_n\| + \|T\|) \, \|(\mathrm{Id}_n - T)^{-1}\| = (1 + \|T\|) \, \|(\mathrm{Id}_n - T)^{-1}\| \ . \end{aligned}$$

This proves the first inequality in (3.2.6).

From $(\mathrm{Id}_n - T)^{-1} = \mathrm{Id}_n + T(\mathrm{Id}_n - T)^{-1}$, we get

$$\|(\mathrm{Id}_n - T)^{-1}\| = \|\mathrm{Id}_n + T(\mathrm{Id}_n - T)^{-1}\| \leq \|\mathrm{Id}_n\| + \|T\| \, \|(\mathrm{Id}_n - T)^{-1}\| = 1 + \|T\| \, \|(\mathrm{Id}_n - T)^{-1}\| \ .$$

Thus

$$(1 - \|T\|) \, \|(\mathrm{Id}_n - T)^{-1}\| \leq 1$$

and this proves the second inequality in (3.2.6). ∎

Proposition 3.2.5 and Corollary 3.2.6 are often referenced as the **Banach Lemma**. It will be useful to have the following generalization of the previous corollary.

Corollary 3.2.7

Suppose that P and Q are two $n \times n$ matrices, and P is invertible. If $\|P - Q\| < 1/\|P^{-1}\|$, then Q is invertible and

$$\frac{\|P\|^{-1}}{1+\|P-Q\|\,\|P^{-1}\|} \le \|Q^{-1}\| \le \frac{\|P^{-1}\|}{1-\|P-Q\|\,\|P^{-1}\|} \ .$$

Proof.
Since

$$\|(P-Q)P^{-1}\| \le \|P-Q\|\,\|P^{-1}\| < 1 \ ,$$

we have that $\sigma\left((P-Q)P^{-1}\right) < 1$. It follows from Proposition 3.2.5, that $QP^{-1} = \mathrm{Id}_n - (P-Q)P^{-1}$ is invertible. Since P^{-1} is invertible, we get the Q is invertible.

Moreover, we get from Corollary 3.2.6 with $T = (P-Q)P^{-1}$ that

$$\frac{1}{1+\|(P-Q)P^{-1}\|} \le \|PQ^{-1}\| \le \frac{1}{1-\|(P-Q)P^{-1}\|} \ .$$

Hence

$$\|Q^{-1}\| = \|P^{-1}PQ^{-1}\| \le \|P^{-1}\|\,\|PQ^{-1}\| \le \frac{\|P^{-1}\|}{1-\|(P-Q)P^{-1}\|} \le \frac{\|P^{-1}\|}{1-\|P-Q\|\,\|P^{-1}\|}$$

and

$$\|Q^{-1}\| \ge \frac{\|PQ^{-1}\|}{\|P\|} \ge \frac{\|P\|^{-1}}{1+\|(P-Q)P^{-1}\|} \ge \frac{\|P\|^{-1}}{1+\|P-Q\|\,\|P^{-1}\|} \ .$$

∎

Theorem 3.2.8

Let T be an $n\times n$ matrix. The sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ defined by (3.2.4) converges for all $\mathbf{x}_0 \in \mathbb{R}^n$ to the unique solution $\mathbf{p}$ of $\mathbf{x} = T\mathbf{x}+\mathbf{c}$ if and only if $\rho(T) < 1$.

Proof.
$\Leftarrow$) Since $\rho(T) < 1$, we have from the Proposition 3.2.5 that $\mathrm{Id}_n - T$ is invertible. Thus $\mathbf{x} = T\mathbf{x}+\mathbf{c}$, namely $(\mathrm{Id}_n - T)\mathbf{x} = \mathbf{c}$, has a unique solution given by $\mathbf{p} = (\mathrm{Id}_n - T)^{-1}\mathbf{c}$.

We show by induction that

$$\mathbf{x}_k = T^k\mathbf{x}_0 + \sum_{j=0}^{k-1} T^j\mathbf{c} \ .$$

This is certainly true if $k=1$. If we assume that $\mathbf{x}_{k-1} = T^{k-1}\mathbf{x}_0 + \sum_{j=0}^{k-2} T^j\mathbf{c}$, then

$$\mathbf{x}_k = T\mathbf{x}_{k-1}+\mathbf{c} = T\left(T^{k-1}\mathbf{x}_0 + \sum_{j=0}^{k-2}T^j\mathbf{c}\right) + \mathbf{c} = T^k\mathbf{x}_0 + \sum_{j=0}^{k-2}T^{j+1}\mathbf{c} + \mathbf{c} = T^k\mathbf{x}_0 + \sum_{j=0}^{k-1}T^j\mathbf{c} \ .$$

Using Proposition 3.2.5 and Theorem 3.1.14, we get

$$\lim_{k\to\infty} \mathbf{x}_k = \lim_{k\to\infty} T^k\mathbf{x}_0 + \lim_{k\to\infty} \sum_{j=0}^{k-1} T^j\mathbf{c} = \mathbf{0} + (\mathrm{Id}_n - T)^{-1}\mathbf{c} = \mathbf{p} \ .$$

$\Rightarrow$) Let $\mathbf{x}_0$ be any vector in $\mathbb{R}^n$. We show by induction that

$$\mathbf{p} - \mathbf{x}_k = T^k(\mathbf{p} - \mathbf{x}_0) \ .$$

This is certainly true for $k = 0$. If we assume that $\mathbf{p} - \mathbf{x}_{k-1} = T^{k-1}(\mathbf{p} - \mathbf{x}_0)$, then

$$\mathbf{p} - \mathbf{x}_k = (T\mathbf{p} + \mathbf{c}) - (T\mathbf{x}_{k-1} + \mathbf{c}) = T(\mathbf{p} - \mathbf{x}_{k-1}) = T\left(T^{k-1}(\mathbf{p} - \mathbf{x}_0)\right) = T^k(\mathbf{p} - \mathbf{x}_0) \ .$$

Hence

$$\lim_{k\to\infty} T^k(\mathbf{p} - \mathbf{x}_0) = \lim_{k\to\infty}(\mathbf{p} - \mathbf{x}_k) = \mathbf{0}$$

because $\{\mathbf{x}_k\}_{k=0}^{\infty}$ converges to $\mathbf{p}$ by hypothesis. Since $\mathbf{p} - \mathbf{x}_0$ can be any vector in $\mathbb{R}^n$ by the arbitrary status of $\mathbf{x}_0$, we get that $\rho(T) < 1$ from Theorem 3.1.14. ∎

Corollary 3.2.9

Let T be an $n \times n$ matrix. Suppose that $\|T\| < 1$. Then

$$\|\mathbf{p} - \mathbf{x}_k\| \leq \frac{\|T\|^k}{1 - \|T\|}\|\mathbf{x}_1 - \mathbf{x}_0\| \ .$$

Proof.
We prove by induction that

$$\|\mathbf{x}_{j+1} - \mathbf{x}_j\| \leq \|T\|^j\|\mathbf{x}_1 - \mathbf{x}_0\| \ .$$

This is true for $j = 0$. If we assume that $\|\mathbf{x}_j - \mathbf{x}_{j-1}\| \leq \|T\|^{j-1}\|\mathbf{x}_1 - \mathbf{x}_0\|$, then

$$\begin{aligned}
\|\mathbf{x}_{j+1} - \mathbf{x}_j\| &= \|(T\mathbf{x}_j + \mathbf{c}) - (T\mathbf{x}_{j-1} + \mathbf{c})\| = \|T(\mathbf{x}_j - \mathbf{x}_{j-1})\| \\
&\leq \|T\|\|\mathbf{x}_j - \mathbf{x}_{j-1}\| \leq \|T\|\|T\|^{j-1}\|\mathbf{x}_1 - \mathbf{x}_0\| = \|T\|^j\|\mathbf{x}_1 - \mathbf{x}_0\| \ .
\end{aligned}$$

Hence, for $m > k$,

$$\begin{aligned}
\|\mathbf{x}_m - \mathbf{x}_k\| &= \|\mathbf{x}_m - \mathbf{x}_{m-1} + \mathbf{x}_{m-1} - \mathbf{x}_{m-2} + \ldots - \mathbf{x}_{k+1} + \mathbf{x}_{k+1} - \mathbf{x}_k\| \\
&\leq \|\mathbf{x}_m - \mathbf{x}_{m-1}\| + \|\mathbf{x}_{m-1} - \mathbf{x}_{m-2}\| + \ldots + \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \\
&\leq \left(\|T\|^{m-1} + \|T\|^{m-2} + \ldots \|T\|^k\right)\|\mathbf{x}_1 - \mathbf{x}_0\| \\
&= \|T\|^k\left(\|T\|^{m-k-1} + \|T\|^{m-k-2} + \ldots + \|T\| + 1\right)\|\mathbf{x}_1 - \mathbf{x}_0\| \ .
\end{aligned}$$

If we let m goes to infinity, we get

$$\|\mathbf{p} - \mathbf{x}_k\| \leq \|T\|^k\left(\sum_{j=0}^{\infty}\|T\|^j\right)\|\mathbf{x}_1 - \mathbf{x}_0\| = \frac{\|T\|^k}{1 - \|T\|}\|\mathbf{x}_1 - \mathbf{x}_0\|$$

because $\{\mathbf{x}_m\}_{m=0}^{\infty}$ converges to $\mathbf{p}$ by the previous theorem since $\rho(T) \leq \|T\| < 1$. The series in the previous expression is the geometric series. ∎

Remark 3.2.10
Still in the context of Theorem 3.2.8, since

$$\begin{aligned}\|\mathbf{x}_j - \mathbf{p}\| &= \|\left(T\mathbf{x}_{j-1} + \mathbf{c}\right) - \left(T\mathbf{p} + \mathbf{c}\right)\| = \|T\left(\mathbf{x}_{j-1} - \mathbf{p}\right)\| \leq \|T\|\|\mathbf{x}_{j-1} - \mathbf{p}\| \\ &\leq \|T\|\left(\|\mathbf{x}_{j-1} - \mathbf{x}_j\| + \|\mathbf{x}_j - \mathbf{p}\|\right) ,\end{aligned}$$

we get

$$\|\mathbf{x}_j - \mathbf{p}\| \leq \frac{\|T\|}{1 - \|T\|}\|\mathbf{x}_{j-1} - \mathbf{x}_j\| ,$$

where $\|T\| \neq 1$. This motivate the principle of stopping iterating when $\|\mathbf{x}_j - \mathbf{x}_{j-1}\|$ is small enough. ♠

Definition 3.2.11

An $n \times n$ matrix A is **strictly row diagonally dominant** if

$$\frac{1}{|a_{i,i}|}\sum_{\substack{j=1\\j\neq i}}^{n} |a_{i,j}| < 1$$

for $i = 1, 2, 3, \ldots, n$.

Theorem 3.2.12

If A is strictly row diagonally dominant, then for any choice of $\mathbf{x}_0$, both the Jacobi and the Gauss-Seidel iterative methods generate sequences $\{\mathbf{x}_k\}_{k=0}^{\infty}$ which converge to the unique solution of $A\mathbf{x} = \mathbf{b}$.

Proof.
For Jacobi) Using the notation at the beginning of the section, we have seen that the Jacobi iterative method is of the form (3.2.4), where $T = D^{-1}(L + U)$ and $\mathbf{c} = D^{-1}\mathbf{b}$.

Since A is strictly row diagonally dominant

$$\|T\|_\infty = \max_{1\leq i\leq n}\left(\frac{1}{|a_{i,i}|}\sum_{\substack{j=1\\j\neq i}}^{n} |a_{i,j}|\right) < 1 .$$

Thus $\rho(T) \leq \|T\|_\infty < 1$ by Theorem 3.1.14. The conclusion of the theorem follows from Theorem 3.2.8.

For Gauss-Seidel) The Gauss-Seidel iterative method defined by (3.2.2) is of the form (3.2.4), where $T = (D - L)^{-1}U$ and $\mathbf{c} = (D - L)^{-1}\mathbf{b}$.

Let λ be an eigenvalue of T and $\mathbf{x}$ be an eigenvector associated to λ. We assume that $\|\mathbf{x}\|_\infty = 1$. From $T\mathbf{x} = \lambda\mathbf{x}$, we get $U\mathbf{x} = \lambda(D - L)\mathbf{x}$. Since U is a strictly upper-triangular matrix with $u_{i,j} = -a_{i,j}$ for $j > i$ and $D - L$ is a lower-triangular matrix with $d_{i,j} - l_{i,j} = a_{i,j}$ for $i \geq j$, we get

$$-\sum_{j=i+1}^{n} a_{i,j}x_j = \lambda\sum_{j=0}^{i} a_{i,j}x_j$$

for $i = 1, 2, \dots, n$. This is equivalent to

$$\lambda a_{i,i}x_i = -\sum_{j=i+1}^{n} a_{i,j}x_j - \lambda \sum_{j=0}^{i-1} a_{i,j}x_j$$

for $i = 1, 2, \dots, n$.

If i is the index of $\mathbf{x}$ such that $|x_i| = \|\mathbf{x}\|_\infty = 1$, then

$$|\lambda||a_{i,i}| = |\lambda||a_{i,i}||x_i| \leq \sum_{j=i+1}^{n} |a_{i,j}||x_j| + |\lambda|\sum_{j=0}^{i-1}|a_{i,j}||x_j| \leq \sum_{j=i+1}^{n} |a_{i,j}| + |\lambda|\sum_{j=0}^{i-1}|a_{i,j}|$$

because $|x_j| \leq \|\mathbf{x}\|_\infty = 1$ for all j. Hence,

$$|\lambda| \leq \sum_{j=i+1}^{n} |a_{i,j}| \left(|a_{i,i}| - \sum_{j=0}^{i-1}|a_{i,j}| \right)^{-1} < 1$$

because A is strictly row diagonally dominant; namely, $\displaystyle\sum_{j=0}^{i-1}|a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}| < |a_{i,i}|$.

Since λ was an arbitrary eigenvalue of A, we get $\rho(A) < 1$. The conclusion of the theorem follows from Theorem 3.2.8. ∎

3.3 Relaxation Methods

As for Jacobi and Gauss-Seidel iterative methods, given a vector $\mathbf{x}_0 \in \mathbb{R}^n$, the goal is to generate a sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ that converges to the solution of (3.0.1). The classical **relaxation methods** are given in the following algorithm.

Algorithm 3.3.1 (Relaxation Methods)

1. Choose a real number ω between 0 and 2. The choice of ω will be justified later.
2. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).
3. Given the vector $\mathbf{x}_k$, compute the vector $\mathbf{x}_{k+1}$ as follows:

$$x_{k+1,i} = x_{k,i} + \frac{\omega}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i.j}x_{k+1,j} - \sum_{j=i}^{n} a_{i.j}x_{k,j} \right) \tag{3.3.1}$$

   for $i = 1, 2, \dots, n$.

4. Repeat (3) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

The previous algorithm is called an **under-relaxation method** for $0 < \omega < 1$, and an **over-relaxation method** or a **successive over-relaxation (SOR) method** for $1 < \omega < 2$.

We now give the motivation behind (3.3.1). Using (3.2.3), we can write $A\mathbf{x} = \mathbf{b}$ as

$$-L\mathbf{x} = -D\mathbf{x} + U\mathbf{x} + \mathbf{b} \ .$$

Multiplying both sides by a non-zero factor ω and adding $D\mathbf{x}$ on both sides yield

$$(D - \omega L)\,\mathbf{x} = (1-\omega)D\mathbf{x} + \omega U\mathbf{x} + \omega\mathbf{b} \ .$$

Finally, multiplying by $(D-\omega L)^{-1}$ from the left on both sides of the equality above gives

$$\mathbf{x} = (D-\omega L)^{-1}\left((1-\omega)D + \omega U\right)\mathbf{x} + \omega(D-\omega L)^{-1}\mathbf{b} \ . \tag{3.3.2}$$

This equation equivalent to $A\mathbf{x} = \mathbf{b}$.

With $T = (D-\omega L)^{-1}((1-\omega)D+\omega U)$ and $\mathbf{c} = \omega(D-\omega L)^{-1}\mathbf{b}$, the equation (3.3.2) becomes $\mathbf{x} = T\mathbf{x} + \mathbf{c}$. If the matrix T satisfies Theorem 3.2.8, the sequence $\{\mathbf{x}_k\}_{k=1}^{\infty}$ defined by $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c}$ converges to a solution $\mathbf{p}$ of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$; namely, a solution of $A\mathbf{x} = \mathbf{b}$.

The equation $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c}$ is the one given in (3.3.1).

Code 3.3.2 (Relaxation Methods)

To approximate the solution of the linear system $A\mathbf{x} = \mathbf{b}$.
Input: The matrix A.
The column vector $\mathbf{b}$.
The column vector $\mathbf{x}_0$ (denoted x in the code below).
The value of omega (denoted w in the code below).
The tolerance tol.
The maximal number of iterations allowed limit
Output: The approximation of the solution.

```
%   xx = relaxation(A,b,x,w,tol,limit)

function xx = relaxation(A,b,x,w,tol,limit)
  xx = NaN;
  dim = size(A,1);

  for k = 1:dim
    if ( A(k,k) == 0 )
      disp 'The Relaxation Method fails because some of the'
      disp 'elements on the diagonal are zero.'
      return;
    end
  end

  for k = 1:limit
    xx(1,1) = x(1,1) + w*(b(1,1) - A(1,:)*x(:,1))/A(1,1);
    if dim > 2
      for m = 2:dim
        xx(m,1) = x(m,1) + w*(b(m,1) - A(m,1:m-1)*xx(1:m-1) - ...
```

```
                    A(m,m:dim)*x(m:dim))/A(m,m);
        end
      end

      if ( norm(xx - x) < tol)
          disp(sprintf('Number of iterations = %d',k))
          return;
      end

      x=xx;
    end

    disp 'The Relaxation Method failed to give an approximation to a'
    disp 'solution of  A x = b  within the required accuracy and maximum'
    disp 'number of iterations allowed.'
    xx = NaN;
  end
```

A theorem due to Kahan states that $\rho(T) > |\omega - 1|$. Hence, from $|\rho(T)| < 1$ in Theorem 3.2.8, a necessary condition for the convergence of relaxation methods is that $0 < \omega < 2$. Theorem 3.3.4 below gives a sufficient condition for the convergence of a restricted form of the relaxation methods.

To prove Theorem 3.3.4 below, we consider complex matrices. Recall that the standard scalar product on $\mathbb{C}^n$ is defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^{n} x_j \overline{y_j}$$

for any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{C}^n$. We therefore have that $\langle \mathbf{x}, \lambda\mathbf{y} \rangle = \overline{\lambda} \langle \mathbf{x}, \mathbf{y} \rangle$ for $\lambda \in \mathbb{C}$. Moreover, $\overline{\langle \mathbf{x}, \mathbf{y} \rangle} = \langle \mathbf{y}, \mathbf{x} \rangle$.

The dual A^* of a $n \times n$ complex matrix A is a $n \times n$ matrix such that $\langle A^*\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, A\mathbf{y} \rangle$ for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{C}^n$. Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ be the canonical basis of $\mathbb{C}^n$, then

$$\langle A^*\mathbf{e}_i, \mathbf{e}_j \rangle = \langle \mathbf{e}_i, A\mathbf{e}_j \rangle \Rightarrow a^*_{j,i} = \overline{a}_{i,j} \tag{3.3.3}$$

for $1 \leq i, j \leq n$. Thus A^* is the complex conjugate transpose of A.

A $n \times n$ complex matrix A is **Hermitian** if $A^* = A$; namely, $\langle A\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, A\mathbf{y} \rangle$ for all $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{C}^n$. It follows from (3.3.3) that $a_{j,i} = \overline{a}_{i,j}$ for $1 \leq i, j \leq n$. In particular, for $i = j$, we get $a_{j,j} = \overline{a}_{j,j}$ for $1 \leq j \leq n$. The elements on the diagonal of A are real numbers.

The eigenvalues of an Hermitian matrix A are real numbers. Suppose that λ is an eigenvalue of A and $\mathbf{v}$ is an eigenvector associated to λ. Then

$$\langle A\mathbf{v}, \mathbf{v} \rangle = \langle \mathbf{v}, A\mathbf{v} \rangle \Rightarrow \langle \lambda\mathbf{v}, \mathbf{v} \rangle = \langle \mathbf{v}, \lambda\mathbf{v} \rangle \Rightarrow \lambda \langle \mathbf{v}, \mathbf{v} \rangle = \overline{\lambda} \langle \mathbf{v}, \mathbf{v} \rangle \Rightarrow \lambda = \overline{\lambda} \ . \tag{3.3.4}$$

A $n \times n$ complex matrix A is **strictly positive definite** if A is Hermitian and

$$\langle \mathbf{x}, A\mathbf{x} \rangle = \mathbf{x}^* A\mathbf{x} > 0$$

for all non-zero vector $\mathbf{x} \in \mathbb{C}^n$, where $\mathbf{x}^* = \begin{pmatrix} \overline{x_1} & \overline{x_2} & \ldots & \overline{x_n} \end{pmatrix}$. Since $a_{j,j} = \langle A\mathbf{e}_j, \mathbf{e}_j \rangle > 0$ for $1 \leq j \leq n$, the elements on the diagonal of a strictly positive definite matrix A are positive real numbers. Moreover, the eigenvalues of a strictly positive definite matrix A are positive numbers. Suppose that λ is an eigenvalue of A and $\mathbf{v}$ is an eigenvector associated to λ, then

$$\langle A\mathbf{v}, \mathbf{v} \rangle > 0 \Rightarrow \langle \lambda\mathbf{v}, \mathbf{v} \rangle > 0 \Rightarrow \lambda \underbrace{\langle \mathbf{v}, \mathbf{v} \rangle}_{=\|\mathbf{v}\|^2>0} > 0 \Rightarrow \lambda > 0 \ .$$

Remark 3.3.3

Suppose that A is a $n \times n$ complex matrix which is strictly positive definite. Let $A = D - U - L$, where D, U and L are defined in (3.2.3). Then, D is strictly positive definite because it is obviously Hermitian and

$$\mathbf{x}^* D\mathbf{x} = \left(\sum_{j=1}^{n} \overline{x}_j \mathbf{e}_j^* \right) D \left(\sum_{i=1}^{n} x_i \mathbf{e}_i \right) = \sum_{j=1}^{n} \sum_{i=1}^{n} \overline{x}_j x_i \underbrace{\mathbf{e}_j^* D \mathbf{e}_i}_{\substack{=0 \text{ for } i\neq j \\ =a_{j,j} \text{ for } i=j}} = \sum_{j=1}^{n} \overline{x}_j x_j a_{j,j} = \sum_{j=1}^{n} |x_j|^2 a_{j,j} > 0$$

for all $\mathbf{x} \neq \mathbf{0}$. ♠

Theorem 3.3.4

Suppose that the $n \times n$ matrix A is strictly positive definite. If $0 < \omega < 2$ and $\mathbf{x}_0$ is any vector in $\mathbb{R}^n$, then the relaxation method given by (3.3.1) generates a sequence which converges to the only solution of $A\mathbf{x} = \mathbf{b}$.

Proof.

The conclusion of the theorem is a consequence of Theorem 3.2.8 if we prove that $\rho(T) < 1$, where $T = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$.

Let $\lambda \in \mathbb{C}$ be an eigenvalue of T and $\mathbf{x} \in \mathbb{C}^n$ be an eigenvector associated to λ. We first note that $\lambda \neq 1$. If $\lambda = 1$, we get from $T\mathbf{x} = \mathbf{x}$ that

$$\begin{aligned} (D - \omega L)\,\mathbf{x} = (1 - \omega) D\mathbf{x} + \omega U\mathbf{x} &\Rightarrow D\mathbf{x} - \omega L\mathbf{x} = D\mathbf{x} - \omega D\mathbf{x} + \omega U\mathbf{x} \\ &\Rightarrow \omega(D - U - L)\mathbf{x} = \omega A\mathbf{x} = \mathbf{0} \ . \end{aligned}$$

Since A is invertible, we get $\mathbf{x} = \mathbf{0}$ which cannot be because $\mathbf{x}$ is an eigenvector associated to λ.

We now construct a relation between ω and λ that will be used to show that $|\lambda| < 1$. Since

$$(1 - \omega)D + \omega U = D - \omega(D - U) = D - \omega L - \omega(D - U - L) = (D - \omega L) - \omega A \ ,$$

we get that $T = \operatorname{Id} - Q^{-1}A$, where $Q = \dfrac{1}{\omega}(D - \omega L)$. Hence, $\operatorname{Id} - T = Q^{-1}A$ and $(D - \omega L)Q^{-1}\mathbf{y} = \omega\mathbf{y}$ for all $\mathbf{y}$. We get

$$(1 - \lambda)(D - \omega L)\mathbf{x} = (D - \omega L)\left((1 - \lambda)\mathbf{x}\right) = (D - \omega L)(\operatorname{Id} - T)\mathbf{x} = (D - \omega L)Q^{-1}A\mathbf{x} = \omega A\mathbf{x} \ .$$

Thus

$$(D - \omega L)\mathbf{x} = \frac{\omega}{1-\lambda} A\mathbf{x} \ . \tag{3.3.5}$$

Moreover, from $T = \mathrm{Id} - Q^{-1}A$, we also have that $Q(\mathrm{Id} - T) = A$. Hence

$$(1-\lambda)QT\mathbf{x} = QT\left((1-\lambda)\mathbf{x}\right) = QT(\mathrm{Id} - T)\mathbf{x} = Q(\mathrm{Id} - T)T\mathbf{x} = AT\mathbf{x} = \lambda A\mathbf{x} \ .$$

We get

$$QT\mathbf{x} = \frac{\lambda}{1-\lambda} A\mathbf{x} \ . \tag{3.3.6}$$

It follows from the definitions of T and Q, and (3.3.6) that

$$(1-\omega)D\mathbf{x} + \omega U\mathbf{x} = (D - \omega L)T\mathbf{x} = \omega QT\mathbf{x} = \frac{\lambda\omega}{1-\lambda} A\mathbf{x} \ . \tag{3.3.7}$$

From (3.3.5) and (3.3.7), we respectively get

$$\langle D\mathbf{x}, \mathbf{x}\rangle - \omega \langle L\mathbf{x}, \mathbf{x}\rangle = \frac{\omega}{1-\lambda} \langle A\mathbf{x}, \mathbf{x}\rangle \tag{3.3.8}$$

and

$$\langle \mathbf{x}, D\mathbf{x}\rangle - \omega \langle \mathbf{x}, D\mathbf{x}\rangle + \omega \langle \mathbf{x}, U\mathbf{x}\rangle = \frac{\omega\overline{\lambda}}{1-\overline{\lambda}} \langle \mathbf{x}, A\mathbf{x}\rangle \ . \tag{3.3.9}$$

Since $A = A^*$, we have that $\langle \mathbf{x}, U\mathbf{x}\rangle = \langle L\mathbf{x}, \mathbf{x}\rangle$ and $\langle \mathbf{x}, D\mathbf{x}\rangle = \langle D\mathbf{x}, \mathbf{x}\rangle$ because the transposed conjugate of U is L and the elements on the diagonal of A are real. Adding (3.3.8) and (3.3.9), we get

$$(2-\omega)\langle D\mathbf{x}, \mathbf{x}\rangle = \omega\left(\frac{1}{1-\lambda} + \frac{\overline{\lambda}}{1-\overline{\lambda}}\right)\langle A\mathbf{x}, \mathbf{x}\rangle = \frac{\omega(1-|\lambda|^2)}{|1-\lambda|^2}\langle A\mathbf{x}, \mathbf{x}\rangle \ .$$

Since $2 - \omega > 0$, $|1-\lambda|^2 > 0$, $\langle D\mathbf{x}, \mathbf{x}\rangle > 0$ (Remark 3.3.3) and $\langle A\mathbf{x}, \mathbf{x}\rangle > 0$ because A is strictly positive definite , we must have $|\lambda| < 1$. Since this is true for any eigenvalue λ of T, we get $\rho(T) < 1$ as desired. ∎

We state without proving.

Theorem 3.3.5

Let A be a tridiagonal, strictly positive definite matrix. Let $T_j = D^{-1}(L+U)$ (Jacobi iterative method) and $T_{gs} = (D-L)^{-1}U$ (Gauss-Seidel iterative method). Then $\rho(T_{gs}) = (\rho(T_j))^2 < 1$ and $\omega = 2/(1+\sqrt{1-\rho(T_{gs})}) = 2/(1+\sqrt{1-\rho^2(T_j)})$ is the optimal choice of ω for the relaxation method.

Remark 3.3.6

The proof of Theorem 3.3.4 is true if we replace $A = D - U - L$ by $A = D - C - C^*$, where D is strictly positive definite. By modifying the decomposition of A as stated in the previous statement, we can generate other relaxation methods than the classical one. ♠

3.4 Extrapolation

There are two steps to the method of **extrapolation** of the solutions.

1. The first step of extrapolation consists in embedding the equation $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ into a family of equations of the form $\mathbf{x} = T_s\mathbf{x} + \mathbf{c}_s$ for $s \in \mathbb{R}$ such that:
   (a) $\mathbf{x} = T_s\mathbf{x} + \mathbf{c}_s$ and $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ have the same solutions.
   (b) $\rho(T_s) < 1$ for some $s \in \mathbb{R}$.
2. The second step of extrapolation is to choose s_0 such that $\rho(T_{s_0}) < 1$ and solve $\mathbf{x} = T_{s_0}\mathbf{x} + \mathbf{c}_{s_0}$ using the iterative procedure $\mathbf{x}_{k+1} = T_{s_0}\mathbf{x}_k + \mathbf{c}_{s_0}$ for $k = 1, 2, 3, \ldots$ Since $\rho(T_{s_0}) < 1$, this iterative procedure converges toward a solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

If $\rho(T_s)$ has an absolute minimum at $s = s_0$, we may expect that, among all converging iterative procedures of the form $\mathbf{x}_{k+1} = T_s\mathbf{x}_k + \mathbf{c}_s$ for $s \in \mathbb{R}$, the iterative procedure with $s = s_0$ will converge the fastest.

In this section, we consider the special case $T_s = sT + (1-s)\,\mathrm{Id}$ and $\mathbf{c}_s = s\mathbf{c}$. Simple algebraic manipulations show that $\mathbf{x} = T_s\mathbf{x} + \mathbf{c}_s$ can be reduced to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ is $s \neq 0$.

The eigenvalues of $T_s = sT + (1-s)\,\mathrm{Id}$ are of the form $s\lambda + (1-s)$, where λ is an eigenvalue of T. Hence,

$$\rho(T_s) = \max\{|s\lambda + (1-s)| : \lambda \text{ is an eigenvalue of T}\} .$$

The following theorem gives a formula to compute the value s_0 where $\rho(T_s)$ has an absolute minimum.

Theorem 3.4.1

Consider the family of iterative procedures $\mathbf{x}_{k+1} = T_s\mathbf{x}_k + \mathbf{c}_s$, where $T_s = sT + (1-s)\,\mathrm{Id}$ and $\mathbf{c}_s = s\,\mathbf{c}$. Suppose that $a \leq \lambda \leq b$ for all eigenvalues λ of T and $1 \notin [a,b]$. Then, $\rho(T_{s_0}) < 1 - |s_0|d < 1$ for $s_0 = 2/(2-a-b)$ and the distance d between 1 and $[a,b]$. Moreover, if $[a,b]$ is the smallest interval containing the eigenvalues of T, then the absolute minimum of $\rho(T_s)$ is reached at s_0.

Proof.
We consider the case where $1 < a < b$. The case $a < b < 1$ is similar.

The eigenvalues of $T_s = sT + (1-s)\,\mathrm{Id}$ are of the form $s\lambda + (1-s) = s(\lambda - 1) + 1$, where λ is an eigenvalue of T. Since $\lambda \geq a > 1$ for all eigenvalues of T, we have $s(\lambda-1)+1 \geq 1$ for $s \geq 0$. We must therefore assume that $s < 0$.

We have $d = a - 1$ and $2 - a - b = (1-a) + (1-b) < 0$. Hence, $s_0 < 0$ and

$$0 < |s_0|d = \left|\frac{2}{2-a-b}\right|(a-1) = \frac{2(a-1)}{a+b-2} = \frac{2(a-1)}{(a-1)+(b-1)} < 1$$

because $b-1 > a-1 > 0$.

From $s<0$ and $a \leq \lambda \leq b$ for all eigenvalues λ of T, we get that

$$sa + (1-s) \geq \mu \geq sb + (1-s) \tag{3.4.1}$$

for all eigenvalues μ of T_s. Thus,

$$\mu \geq s_0 b + (1-s_0) = s_0(b+a-2) - s_0(a-1) + 1 = -s_0(a-1) - 1 = -s_0 d - 1$$

and

$$\mu \leq s_0 a + (1-s_0) = s_0(a-1) + 1 = s_0 d + 1$$

for all eigenvalues μ of T_{s_0}. Hence $\rho(T_{s_0}) < |1+s_0 d| = 1 - |s_0| d < 1$ because $s_0 d < 0 < |s_0| d < 1$.

If $[a,b]$ is the smallest interval such that $a \leq \lambda \leq b$ for all eigenvalue λ of T, then $s(a-1)+1$ and $s(b-1)+1$ are eigenvalues of T_s with $1 > s(a-1)+1 > s(b-1)+1$. If $|s(a-1)+1| > |s(b-1)+1|$, then $\rho(T_s) = s(a-1)+1$ and it increases as $s<0$ increases. If $|s(b-1)+1| > |s(a-1)+1|$, then $\rho(T_s) = |s(b-1)+1| = -s(b-1)-1$ and it increases as $s<0$ decreases. The minimum is therefore when $s(a-1)+1 = -s(b-1)-1$. Solving for s gives $s = s_0$ as desired. ∎

3.5 Steepest Descent and Conjugate Gradient

We consider systems of the form $A\mathbf{x} = \mathbf{b}$, where A is a strictly positive definite matrix (this also means that A is symmetric).

3.5.1 Steepest Descent

The basis for the method of steepest descent is the result of the following proposition.

Proposition 3.5.1

Let A be a strictly positive definite matrix and $\mathbf{b} \in \mathbb{R}^n$. The solution $\mathbf{p}$ of the linear system $A\mathbf{x} = \mathbf{b}$ is the point where the quadratic function

$$\begin{aligned} g : \mathbb{R}^n &\to \mathbb{R} \\ \mathbf{x} &\mapsto \langle \mathbf{x}, A\mathbf{x} \rangle - 2\langle \mathbf{x}, \mathbf{b} \rangle \end{aligned}$$

reaches its strict absolute minimum.

Proof.
Let $\mathbf{p}$ and $\mathbf{u}$ be any two vectors and let $q(t) = g(\mathbf{p} + t\mathbf{u})$.

For the standard scalar product on $\mathbb{R}^n$,

$$\langle \mathbf{u}, A\mathbf{p} \rangle = \langle A\mathbf{u}, \mathbf{p} \rangle = \langle \mathbf{p}, A\mathbf{u} \rangle$$

because A is symmetric, Hence,

$$\begin{aligned} q(t) &= \langle \mathbf{p} + t\mathbf{u}, A(\mathbf{p} + t\mathbf{u})\rangle - 2\langle \mathbf{p} + t\mathbf{u}, \mathbf{b}\rangle \\ &= g(\mathbf{p}) + t\langle \mathbf{u}, A\mathbf{p}\rangle + t\langle \mathbf{p}, A\mathbf{u}\rangle + t^2\langle \mathbf{u}, A\mathbf{u}\rangle - 2t\langle \mathbf{u}, \mathbf{b}\rangle \\ &= g(\mathbf{p}) + 2t\langle \mathbf{u}, A\mathbf{p} - \mathbf{b}\rangle + t^2\langle \mathbf{u}, A\mathbf{u}\rangle \ . \end{aligned}$$

We note that $\langle \mathbf{u}, A\mathbf{u}\rangle > 0$ for $\mathbf{u} \neq \mathbf{0}$ because A is strictly positive definite. Since the coefficient of t^2 in $q(t)$ is positive, we have a quadratic polynomial which is concave upward. Its minimum is reached at

$$t = t_m = \frac{\langle \mathbf{u}, \mathbf{b} - A\mathbf{p}\rangle}{\langle \mathbf{u}, A\mathbf{u}\rangle}$$

and the minimum value is

$$q(t_m) = g(\mathbf{p}) + 2t_m\langle \mathbf{u}, A\mathbf{p} - \mathbf{b}\rangle + t_m^2\langle \mathbf{u}, A\mathbf{u}\rangle = g(\mathbf{p}) - \frac{\left(\langle \mathbf{u}, \mathbf{b} - A\mathbf{p}\rangle\right)^2}{\langle \mathbf{u}, A\mathbf{u}\rangle} \ . \tag{3.5.1}$$

If $\mathbf{p}$ is a solution of $A\mathbf{x} = \mathbf{b}$, then $A\mathbf{p} - \mathbf{b} = \mathbf{0}$. Therefore, for all directions $\mathbf{u}$, we have $\langle \mathbf{u}, A\mathbf{p} - \mathbf{b}\rangle = \mathbf{0}$. Thus, $q(t)$ reaches its strict absolute minimum of $g(\mathbf{p})$ at $t = 0$ whatever the direction $\mathbf{u}$. Hence, $\mathbf{p}$ is the point where $g(\mathbf{x})$ reaches its strict absolute minimum.

Conversely, if $\mathbf{p}$ is the strict absolute minimum for $g(\mathbf{x})$, we get from (3.5.1) that $\langle \mathbf{u}, A\mathbf{p} - \mathbf{b}\rangle = 0$ for all $\mathbf{u} \in \mathbb{R}^n$. The only vector orthogonal to all vectors in $\mathbb{R}^n$, in particular to itself, is $\mathbf{0}$. Thus $A\mathbf{p} - \mathbf{b} = \mathbf{0}$ and $\mathbf{p}$ is the solution of $A\mathbf{x} = \mathbf{b}$. ∎

The previous proposition suggests the following algorithm.

Algorithm 3.5.2 (**Steepest Descent**)

1. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).
2. Given the vector $\mathbf{x}_k$, choose a direction $\mathbf{u}_k$ such that $\langle \mathbf{u}_k, \mathbf{b} - A\mathbf{x}_k\rangle \neq \mathbf{0}$. If no such vector exists, then $A\mathbf{x}_k = \mathbf{b}$ and the solution has been found.
3. Compute
$$t_k = \frac{\langle \mathbf{u}_k, \mathbf{b} - A\mathbf{x}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle}$$
and let $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k\mathbf{u}_k$.
4. Repeat (2) to (3) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

The steepest descent algorithm is illustrated in Figure 3.1.

The third step of the steepest descent algorithm is deduced as in the proof of Proposition 3.5.1 by considering $q(t) = g(\mathbf{x}_k + t\mathbf{u}_k)$ instead of $q(t) = g(\mathbf{p} + t\mathbf{u})$. In this algorithm, we have not been specific about the choice of the vectors $\mathbf{u}_k$. We present a slight variation of this algorithm in Question 3.15. We now try to choose the vectors $\mathbf{u}_k$ to speed up the

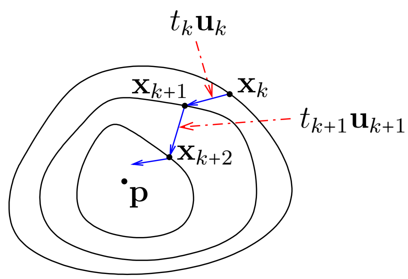


Figure 3.1: A graphical representation of steepest descent algorithm. We have drawn some level curves of the function g defined in Proposition 3.5.1

convergence of the algorithm. In particular, we need to control the size of t_k if we do not want to "overshoot" the solution.

The next proposition shows that, in theory, the steepest descent algorithm ends after a finite number of iterations. However, this does not generally happen for the computer implementation of this algorithm because of round off errors, ill-conditioning, ...

Proposition 3.5.3

Let A be a strictly positive definite matrix and $\mathbf{b} \in \mathbb{R}^n$. Suppose that the vectors $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$ are A-orthogonal vectors in $\mathbb{R}^n$; namely, $\langle \mathbf{u}_i, A\mathbf{u}_j \rangle = 0$ for $i \neq j$. Then the steepest descent algorithm produces the solution of $A\mathbf{x} = \mathbf{b}$ after n steps.

Proof.
Let

$$t_j = \frac{\langle \mathbf{u}_j, \mathbf{b} - A\mathbf{x}_j \rangle}{\langle \mathbf{u}_j, A\mathbf{u}_j \rangle} \tag{3.5.2}$$

and

$$\mathbf{x}_{j+1} = \mathbf{x}_j + t_j \mathbf{u}_j \tag{3.5.3}$$

for $j = 1, 2, \ldots, n$.

We first show by induction that

$$A\mathbf{x}_{k+1} = A\mathbf{x}_1 + t_1 A\mathbf{u}_1 + t_2 A\mathbf{u}_2 + \ldots + t_k A\mathbf{u}_k \tag{3.5.4}$$

for $k = 1, 2, \ldots, n$. Multiplying both sides of (3.5.3) with $j = 1$ by A from the left proves that the previous statement is true for $k = 1$. If we assume that (3.5.4) is true for $k < n$, multiplying both sides of (3.5.3) with $j = k + 1$ by A from the left and using the induction hypothesis yield

$$\begin{aligned} A\mathbf{x}_{k+2} &= A\mathbf{x}_{k+1} + t_{k+1} A\mathbf{u}_{k+1} = \left(A\mathbf{x}_1 + t_1 A\mathbf{u}_1 + t_2 A\mathbf{u}_2 + \ldots + t_k A\mathbf{u}_k \right) + t_{k+1} A\mathbf{u}_{k+1} \\ &= A\mathbf{x}_1 + t_1 A\mathbf{u}_1 + t_2 A\mathbf{u}_2 + \ldots + t_{k+1} A\mathbf{u}_{k+1} \ . \end{aligned}$$

This is (3.5.4) with k replaced by $k+1$.

Using (3.5.2), (3.5.4) and the A-orthogonality of the vectors $\mathbf{u}_j$, we find that

$$\begin{aligned}\langle \mathbf{b} - A\mathbf{x}_{n+1}, \mathbf{u}_k\rangle &= \langle \mathbf{b} - A\mathbf{x}_1 - t_1 A\mathbf{u}_1 - t_2 A\mathbf{u}_2 - \ldots - t_k A\mathbf{u}_k, \mathbf{u}_k\rangle \\ &= \langle \mathbf{b} - A\mathbf{x}_1, \mathbf{u}_k\rangle - \langle \mathbf{b} - A\mathbf{x}_k, \mathbf{u}_k\rangle \\ &= \langle \mathbf{b} - A\mathbf{x}_1, \mathbf{u}_k\rangle - \langle \mathbf{b} - A\mathbf{x}_1 - t_1 A\mathbf{u}_1 - t_2 A\mathbf{u}_2 - \ldots - t_{k-1} A\mathbf{u}_{k-1}, \mathbf{u}_k\rangle \\ &= \langle \mathbf{b} - A\mathbf{x}_1, \mathbf{u}_k\rangle - \langle \mathbf{b} - A\mathbf{x}_1, \mathbf{u}_k\rangle = 0\end{aligned}$$

for $k = 1, 2, \ldots n$. Since $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ is a basis of $\mathbb{R}^n$, the previous equations shows that $A\mathbf{x}_{n+1} = \mathbf{b}$. ∎

3.5.2 Conjugate Gradient

The **conjugate gradient** algorithm is a special case of the steepest descent algorithm, where the vectors $\mathbf{u}_j$ are chosen such that the vectors $\mathbf{r}_j = \mathbf{b} - A\mathbf{x}_j$ are mutually orthogonal.

Algorithm 3.5.4 (Conjugate Gradient)

1. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).
2. Let $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{u}_0 = \mathbf{r}_0$.
3. Given the vectors $\mathbf{u}_k \neq \mathbf{0}$ and $\mathbf{r}_k$, compute
$$t_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle} .$$
4. Given the vectors $\mathbf{x}_k$ and $\mathbf{r}_k$, let $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k\mathbf{u}_k$ and $\mathbf{r}_{k+1} = \mathbf{r}_k - t_k A\mathbf{u}_k$.
5. Stop if $\|\mathbf{r}_{k+1}\|_2^2 < \epsilon$, where ϵ is given.
6. Compute
$$s_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1}\rangle}{\langle \mathbf{r}_k, \mathbf{r}_k\rangle} .$$
7. Let $\mathbf{u}_{k+1} = \mathbf{r}_{k+1} + s_k\mathbf{u}_k$.
8. Repeat (3) to (7) until the condition in (5) is satisfied.

The next theorem[1] shows that the t_k's used in the conjugate gradient algorithm are of the form
$$t_k = \frac{\langle \mathbf{b} - A\mathbf{x}_k, \mathbf{u}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle}$$

[1] In fact, the items (I) to (IV) in the proof of this theorem are as important as the results in the statement of the theorem.

as required for the steepest descent method.

Theorem 3.5.5

The vectors $\mathbf{x}_k$ and $\mathbf{r}_k$ of the conjugate gradient algorithm satisfy $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ and $\langle \mathbf{r}_i, \mathbf{r}_j \rangle = 0$ for $i \neq j$ as long as $\mathbf{u}_k \neq \mathbf{0}$.

Proof.
The proof is by induction. The hypothesis of induction is

$$\begin{array}{llllll} I) & \langle \mathbf{r}_i, \mathbf{u}_j \rangle = 0 & II) & \langle \mathbf{u}_i, A\mathbf{u}_j \rangle = 0 & III) & \langle \mathbf{r}_i, \mathbf{r}_j \rangle = 0 \\ IV) & \langle \mathbf{r}_i, \mathbf{r}_i \rangle = \langle \mathbf{r}_i, \mathbf{u}_i \rangle & V) & \mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i & VI) & \mathbf{r}_i \neq \mathbf{0} \end{array}$$

for $0 \leq j < i$.

$\mathbf{i = 1}$)

We prove that the hypothesis of induction is true for $i = 1$. Recall that $\mathbf{u}_0 = \mathbf{r}_0$ in the conjugate gradient algorithm. Hence.

$$\begin{aligned} \langle \mathbf{r}_1, \mathbf{u}_0 \rangle &= \langle \mathbf{r}_0 - t_0 A\mathbf{u}_0, \mathbf{u}_0 \rangle = \langle \mathbf{r}_0, \mathbf{u}_0 \rangle - t_0 \langle A\mathbf{u}_0, \mathbf{u}_0 \rangle \\ &= \langle \mathbf{r}_0, \mathbf{r}_0 \rangle - t_0 \langle \mathbf{u}_0, A\mathbf{u}_0 \rangle = \langle \mathbf{r}_0, \mathbf{r}_0 \rangle - \langle \mathbf{r}_0, \mathbf{r}_0 \rangle = 0 \ . \end{aligned}$$

Thus $\underline{I \text{ and } III \text{ are true for } i = 1}$.

From I with $i = 1$, we get

$$\langle \mathbf{r}_1, \mathbf{u}_1 \rangle = \langle \mathbf{r}_1, \mathbf{r}_1 + s_0 \mathbf{u}_0 \rangle = \langle \mathbf{r}_1, \mathbf{r}_1 \rangle + s_0 \langle \mathbf{r}_1, \mathbf{u}_0 \rangle = \langle \mathbf{r}_1, \mathbf{r}_1 \rangle$$

and this proves $\underline{IV \text{ for } i = 1}$.

Since $A\mathbf{u}_0 = t_0^{-1}(\mathbf{r}_0 - \mathbf{r}_1)$, we get from I with $i = 1$ that $\langle \mathbf{u}_0, A\mathbf{u}_0 \rangle = t_0^{-1} \langle \mathbf{r}_0, \mathbf{r}_0 \rangle$. Combined with $s_0 \langle \mathbf{r}_0, \mathbf{r}_0 \rangle = \langle \mathbf{r}_1, \mathbf{r}_1 \rangle$, we get from III with $i = 1$ that

$$\begin{aligned} \langle \mathbf{u}_1, A\mathbf{u}_0 \rangle &= \langle \mathbf{r}_1 + s_0 \mathbf{u}_0, A\mathbf{u}_0 \rangle = \langle \mathbf{r}_1, A\mathbf{u}_0 \rangle + s_0 \langle \mathbf{u}_0, A\mathbf{u}_0 \rangle \\ &= t_0^{-1} \langle \mathbf{r}_1, \mathbf{r}_0 - \mathbf{r}_1 \rangle + t_0^{-1} \langle \mathbf{r}_1, \mathbf{r}_1 \rangle = t_0^{-1} \langle \mathbf{r}_1, \mathbf{r}_0 \rangle = 0 \end{aligned}$$

and this proves $\underline{II \text{ for } i = 1}$.

$\underline{V \text{ for } i = 1}$ is a consequence of

$$\mathbf{b} - A\mathbf{x}_1 = \mathbf{b} - A(\mathbf{x}_0 + t_0 \mathbf{u}_0) = \mathbf{b} - A\mathbf{x}_0 - t_0 A\mathbf{u}_0 = \mathbf{r}_0 - t_0 A\mathbf{u}_0 = \mathbf{r}_1 \ .$$

Since we assume that A is strictly positive definite, it follows that from II with $i = 1$ that

$$\begin{aligned} 0 < \langle \mathbf{u}_1, A\mathbf{u}_1 \rangle &= \langle \mathbf{r}_1 + s_0 \mathbf{u}_0, A\mathbf{u}_1 \rangle = \langle \mathbf{r}_1, A\mathbf{u}_1 \rangle + s_0 \langle \mathbf{u}_0, A\mathbf{u}_1 \rangle \\ &= \langle \mathbf{r}_1, A\mathbf{u}_1 \rangle + s_0 \langle A\mathbf{u}_0, \mathbf{u}_1 \rangle = \langle \mathbf{r}_1, A\mathbf{u}_1 \rangle \end{aligned}$$

as long as $\mathbf{u}_1 \neq \mathbf{0}$. We have used the fact that A is a symmetric matrix for the second to last equality. Hence, $\underline{VI \text{ is true for } i = 1}$.

i = k implies i = k + 1)

We now assume that the hypothesis of induction is true for $i = k$ and shows that this implies that the hypothesis is also true for $i = k+1$. Let $\mathbf{u}_{-1} = \mathbf{0}$ and $s_{-1} = 0$.

From IV with $i = k$, we get

$$\langle \mathbf{r}_{k+1}, \mathbf{u}_k \rangle = \langle \mathbf{r}_k - t_k A\mathbf{u}_k, \mathbf{u}_k \rangle = \langle \mathbf{r}_k, \mathbf{u}_k \rangle - t_k \langle A\mathbf{u}_k, \mathbf{u}_k \rangle = \langle \mathbf{r}_k, \mathbf{u}_k \rangle - \langle \mathbf{r}_k, \mathbf{r}_k \rangle = 0 \ .$$

From I and II with $i = k$, we also get

$$\langle \mathbf{r}_{k+1}, \mathbf{u}_j \rangle = \langle \mathbf{r}_k - t_k A\mathbf{u}_k, \mathbf{u}_j \rangle = \langle \mathbf{r}_k, \mathbf{u}_j \rangle - t_k \langle A\mathbf{u}_k, \mathbf{u}_j \rangle = \langle \mathbf{r}_k, \mathbf{u}_j \rangle - t_k \langle \mathbf{u}_k, A\mathbf{u}_j \rangle = 0$$

for $j < k$. The previous two equations show that $\underline{I \text{ is true for } i = k+1}$.

From I with $i = k+1$, we get

$$\langle \mathbf{r}_{k+1}, \mathbf{u}_{k+1} \rangle = \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} + s_k \mathbf{u}_k \rangle = \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle + s_k \langle \mathbf{r}_{k+1}, \mathbf{u}_k \rangle = \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle$$

and this proves $\underline{IV \text{ for } i = k+1}$.

Using the definition of $\mathbf{u}_i$ in step 7 with $i = k+1$, $i = j+1$ and $i = j$, and the definition of $\mathbf{r}_{j+1}$ in step 4, we get for $j < k$ that

$$\begin{aligned}
\langle \mathbf{u}_{k+1}, A\mathbf{u}_j \rangle &= \langle \mathbf{r}_{k+1} + s_k \mathbf{u}_k, A\mathbf{u}_j \rangle = \langle \mathbf{r}_{k+1}, A\mathbf{u}_j \rangle + s_k \langle \mathbf{u}_k, A\mathbf{u}_j \rangle \\
&= t_j^{-1} \langle \mathbf{r}_{k+1}, \mathbf{r}_j - \mathbf{r}_{j+1} \rangle + s_k \langle \mathbf{u}_k, A\mathbf{u}_j \rangle \\
&= t_j^{-1} \langle \mathbf{r}_{k+1}, \mathbf{u}_j - s_{j-1}\mathbf{u}_{j-1} - \mathbf{u}_{j+1} + s_j \mathbf{u}_j \rangle + s_k \langle \mathbf{u}_k, A\mathbf{u}_j \rangle \\
&= t_j^{-1} \left(\langle \mathbf{r}_{k+1}, \mathbf{u}_j \rangle - s_{j-1} \langle \mathbf{r}_{k+1}, \mathbf{u}_{j-1} \rangle - \langle \mathbf{r}_{k+1}, \mathbf{u}_{j+1} \rangle + s_j \langle \mathbf{r}_{k+1}, \mathbf{u}_j \rangle \right) + s_k \langle \mathbf{u}_k, A\mathbf{u}_j \rangle = 0
\end{aligned}$$

because the first four scalar products are null according to I with $i = k+1$ and the last scalar product is null according to II with $i = k$. Moreover, as above, we have

$$\begin{aligned}
\langle \mathbf{u}_{k+1}, A\mathbf{u}_k \rangle = t_k^{-1} &\left(\langle \mathbf{r}_{k+1}, \mathbf{u}_k \rangle - s_{k-1} \langle \mathbf{r}_{k+1}, \mathbf{u}_{k-1} \rangle - \langle \mathbf{r}_{k+1}, \mathbf{u}_{k+1} \rangle + s_k \langle \mathbf{r}_{k+1}, \mathbf{u}_k \rangle \right) \\
&+ s_k \langle \mathbf{u}_k, A\mathbf{u}_k \rangle \ .
\end{aligned}$$

The first, second and fourth scalar products are null according to I with $i = k+1$. We therefore have that

$$\begin{aligned}
\langle \mathbf{u}_{k+1}, A\mathbf{u}_k \rangle &= -t_k^{-1} \langle \mathbf{r}_{k+1}, \mathbf{u}_{k+1} \rangle + s_k \langle \mathbf{u}_k, A\mathbf{u}_k \rangle \\
&= -\frac{\langle \mathbf{u}_k, A\mathbf{u}_k \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle} \langle \mathbf{r}_{k+1}, \mathbf{u}_{k+1} \rangle + \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle} \langle \mathbf{u}_k, A\mathbf{u}_k \rangle = 0
\end{aligned}$$

due to IV with $i = k+1$. We have thus proved that $\underline{II \text{ is true for } i = k+1}$.

$\underline{V \text{ for } i = k+1}$ is a consequence of

$$\mathbf{b} - A\mathbf{x}_{k+1} = \mathbf{b} - A(\mathbf{x}_k + t_k \mathbf{u}_k) = \mathbf{b} - A\mathbf{x}_k - t_k A\mathbf{u}_k = \mathbf{r}_k - (\mathbf{r}_k - \mathbf{r}_{k+1}) = \mathbf{r}_{k+1} \ ,$$

where we have used V with $i = k$ and the definition of $\mathbf{r}_{k+1}$.

III for $i = k+1$ is a consequence of I with $i = k+1$ since it implies that

$$\langle \mathbf{r}_{k+1}, \mathbf{r}_j \rangle = \langle \mathbf{r}_{k+1}, \mathbf{u}_j - s_{j-1}\mathbf{u}_{j-1} \rangle = \langle \mathbf{r}_{k+1}, \mathbf{u}_j \rangle - s_{j-1} \langle \mathbf{r}_{k+1}, \mathbf{u}_{j-1} \rangle = 0$$

for $j < k+1$.

Finally, since we assume that A is strictly positive definite, it follows from II with $i = k+1$ that

$$\begin{aligned} 0 &< \langle \mathbf{u}_{k+1}, A\mathbf{u}_{k+1} \rangle = \langle \mathbf{r}_{k+1} + s_k \mathbf{u}_k, A\mathbf{u}_{k+1} \rangle = \langle \mathbf{u}_{k+1}, \mathbf{r}_{k+1} \rangle + s_k \langle \mathbf{u}_k, A\mathbf{u}_{k+1} \rangle \\ &= \langle \mathbf{u}_{k+1}, \mathbf{r}_{k+1} \rangle + s_k \langle A\mathbf{u}_k, \mathbf{u}_{k+1} \rangle = \langle \mathbf{r}_{k+1}, \mathbf{u}_{k+1} \rangle \end{aligned}$$

as long as $\mathbf{u}_{k+1} \neq \mathbf{0}$. We have used the fact that A is a symmetric matrix for the second to last equality. Hence, VI is true for $i = k+1$. ∎

3.5.3 Preconditioned Conjugate Gradient

The conjugate gradient method is often used to approximate the solutions of linear systems $A\mathbf{x} = \mathbf{b}$, where A is not well conditioned – Namely, the condition number $\kappa(A)$ of the matrix A is large (Section 4.4). Instead of working with the original system $A\mathbf{x} = \mathbf{b}$, one often transforms this system into an equivalent system $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, where $\tilde{A} = T^\top A T$, $\tilde{\mathbf{x}} = T^{-1}\mathbf{x}$ and $\tilde{\mathbf{b}} = T^\top \mathbf{b}$ for an invertible matrix T.

Instead of computing $\tilde{A}$ and $\tilde{\mathbf{b}}$, and using the conjugate gradient algorithm directly to approximate the solution of $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, we derive an algorithm from the conjugate gradient algorithm that gives us an approximation of the solution of $A\mathbf{x} = \mathbf{b}$ without having to compute $\tilde{A} = T^\top A T$ and $\tilde{\mathbf{b}} = T^\top \mathbf{b}$.

To compare the conjugate gradient algorithm applied to both systems $A\mathbf{x} = \mathbf{b}$ and $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, we let $\tilde{\mathbf{x}}_k = T^{-1}\mathbf{x}_k$ and $\tilde{\mathbf{u}}_k = T^{-1}\mathbf{u}_k$.

We have that

$$\tilde{\mathbf{r}}_k = \tilde{\mathbf{b}} - \tilde{A}\tilde{\mathbf{x}}_k = T^\top \mathbf{b} - (T^\top A T)(T^{-1}\mathbf{x}_k) = T^\top (\mathbf{b} - A\mathbf{x}_k) = T^\top \mathbf{r}_k \ .$$

For the preconditioned conjugate gradient method, we assume that $T\,T^\top$ is an invertible matrix. If $Q^{-1} = T\,T^\top$, then Q^{-1} is a strictly positive definite matrix.

In the conjugate gradient algorithm applied to $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, we have

$$\tilde{t}_k = \frac{\langle \tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k \rangle}{\langle \tilde{\mathbf{u}}_k, \tilde{A}\tilde{\mathbf{u}}_k \rangle} = \frac{\langle T^\top \mathbf{r}_k, T^\top \mathbf{r}_k \rangle}{\langle T^{-1}\mathbf{u}_k, (T^\top A T) T^{-1}\mathbf{u}_k \rangle} = \frac{\langle Q^{-1}\mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k \rangle} \ . \tag{3.5.5}$$

From $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \tilde{t}_k \tilde{\mathbf{u}}_k$, we get $T^{-1}\mathbf{x}_{k+1} = T^{-1}\mathbf{x}_k + \tilde{t}_k T^{-1}\mathbf{u}_k$. Multiplying both sides of this equality by T from the left, we get

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tilde{t}_k \mathbf{u}_k \ . \tag{3.5.6}$$

From $\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - \tilde{t}_k \tilde{A}\,\tilde{\mathbf{u}}_k$, we get $T^\top \mathbf{r}_{k+1} = T^\top \mathbf{r}_k + \tilde{t}_k (T^\top A T)(T^{-1}\mathbf{u}_k)$. Multiplying both sides of this equality by $(T^{-1})^\top$ from the left, we get

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \tilde{t}_k A \mathbf{u}_k \ . \tag{3.5.7}$$

We also have

$$\tilde{s}_k = \frac{\langle \tilde{\mathbf{r}}_{k+1}, \tilde{\mathbf{r}}_{k+1}\rangle}{\langle \tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k\rangle} = \frac{\langle T^\top \mathbf{r}_{k+1}, T^\top \mathbf{r}_{k+1}\rangle}{\langle T^\top \mathbf{r}_k, T^\top \mathbf{r}_k\rangle} = \frac{\langle Q^{-1}\mathbf{r}_{k+1}, \mathbf{r}_{k+1}\rangle}{\langle Q^{-1}\mathbf{r}_k, \mathbf{r}_k\rangle} \ , \tag{3.5.8}$$

Finally, from $\tilde{\mathbf{u}}_{k+1} = \tilde{\mathbf{r}}_{k+1} + \tilde{s}_k\,\tilde{\mathbf{u}}_k$, we get $T^{-1}\mathbf{u}_{k+1} = T^\top \mathbf{r}_{k+1} + \tilde{s}_k T^{-1}\mathbf{u}_k$. Multiplying both sides of this equality by T from the left, we get

$$\mathbf{u}_{k+1} = Q^{-1}\mathbf{r}_{k+1} + \tilde{s}_k \mathbf{u}_k \ . \tag{3.5.9}$$

From (3.5.5) to (3.5.9), we deduce the following algorithm.

Algorithm 3.5.6 (Preconditioned Conjugate Gradient)

1. Choose a vector $\mathbf{x}_0$ closed to the solution of $A\mathbf{x} = \mathbf{b}$ (if possible).
2. Let $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{u}_0 = \mathbf{r}_0$.
3. Solve $Q\tilde{\mathbf{v}}_0 = \mathbf{r}_0$.
4. Given the vectors $\mathbf{u}_k \neq \mathbf{0}$, $\mathbf{r}_k$ and $\tilde{\mathbf{v}}_k$, compute
$$\tilde{t}_k = \frac{\langle \tilde{\mathbf{v}}_k, \mathbf{r}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle} \ .$$
5. Given the vectors $\mathbf{x}_k$ and $\mathbf{r}_k$, let $\mathbf{x}_{k+1} = \mathbf{x}_k + \tilde{t}_k \mathbf{u}_k$ and $\mathbf{r}_{k+1} = \mathbf{r}_k - \tilde{t}_k A\mathbf{u}_k$.
6. Solve $Q\tilde{\mathbf{v}}_{k+1} = \mathbf{r}_{k+1}$.
7. If $\langle \tilde{\mathbf{v}}_{k+1}, \mathbf{r}_{k+1}\rangle < \epsilon$, where $\epsilon > 0$ is given, compute $\|\mathbf{r}_{k+1}\|_2^2$. Stop if this last expression is smaller than ϵ.
8. Compute
$$\tilde{s}_k = \frac{\langle \tilde{\mathbf{v}}_{k+1}, \mathbf{r}_{k+1}\rangle}{\langle \tilde{\mathbf{v}}_k, \mathbf{r}_k\rangle} \ .$$
9. Let $\mathbf{u}_{k+1} = \tilde{\mathbf{v}}_{k+1} + \tilde{s}_k \mathbf{u}_k$.
10. Repeat (4) to (9) until the condition in (7) is satisfied.

A few comments are necessary. From the point of view of the number and complexity of operations, the only difference between the regular conjugate gradient algorithm and the preconditioned conjugate gradient method is the need to solve the systems $Q\tilde{\mathbf{v}}_k = \mathbf{r}_k$. A good choice of Q (and so of T) may reduce the condition number of Q significantly and so possibly accelerate the convergence toward the solution of $A\mathbf{x} = \mathbf{b}$. However, the systems $Q\tilde{\mathbf{v}}_k = \mathbf{r}_k$ may be as difficult to solve as our original system $A\mathbf{x} = \mathbf{b}$. To develop a good preconditioned conjugate gradient algorithm, we need to find the right balance between speeding up the convergence and keeping the systems $Q\tilde{\mathbf{v}}_k = \mathbf{r}_k$ easy to solve.

In (7) of the preconditioning conjugate gradient algorithm, we compute $\|\mathbf{r}_{k+1}\|_2^2$ only if $\langle \tilde{\mathbf{v}}_{k+1}, \mathbf{r}_{k+1} \rangle < \epsilon$ because $\|\mathbf{r}_{k+1}\|_2^2$ is not used to compute $\tilde{s}_k$ and $\tilde{t}_{k+1}$, and eventually $\mathbf{x}_{k+2}$ as it is the case in the original conjugate gradient algorithm. So, to avoid extra computations, we compute $\|\mathbf{r}_{k+1}\|_2^2$ only when we feel that there is a good chance that it is smaller than ϵ. Note that, since $Q^{-1} = TT^\top$,

$$\langle \tilde{\mathbf{v}}_{k+1}, \mathbf{r}_{k+1} \rangle = \left\langle Q^{-1}\mathbf{r}_{k+1}, \mathbf{r}_{k+1} \right\rangle = \langle T^\top \mathbf{r}_{k+1}, T^\top \mathbf{r}_{k+1} \rangle = \|T^\top \mathbf{r}_{k+1}\| > 0$$

for all $\mathbf{r}_{k+1} \neq \mathbf{0}$.

3.6 Exercises

Question 3.1
Prove that $\|\mathbf{x}\| = \sum_{i=1}^{n} 2^{-i}|x_i|$ defines a norm on $\mathbb{R}^n$.

Question 3.2
If $\|\cdot\|$ is a norm on $\mathbb{R}^n$, show that

$$\|\mathbf{x} - \mathbf{y}\| \geq \big|\|\mathbf{x}\| - \|\mathbf{y}\|\big| \tag{3.6.1}$$

for any vector $\mathbf{x}$ and $\mathbf{y}$.

Question 3.3
Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Show that the induced norm on the $n \times n$ matrices satisfies

$$\|A\| = \sup_{\mathbf{x} \neq \mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

for any $n \times n$ matrix A.

Question 3.4
If A is an $n \times n$ matrix, show that the induce norm $\|A\|_1$ is given by

$$\|A\|_1 = \max_{0 \leq j \leq n} \left\{ \sum_{i=0}^{n} |a_{i,j}| \right\} .$$

Question 3.5
Let

$$A = \begin{pmatrix} 4 & -3 & 2 \\ -1 & 0 & 5 \\ 2 & 6 & -2 \end{pmatrix} .$$

Among all vectors $\mathbf{x}$ such that $\|\mathbf{x}\|_\infty = 1$, find a vector where $\|A\mathbf{x}\|_\infty$ reaches its maximum value. What is this maximum value?

Question 3.6
If $\|\cdot\|$ is an induced norm on the space of $n \times n$ matrices, is it true that $\|AB\| = \|BA\|$ for all matrix A and B? Justify your answer.

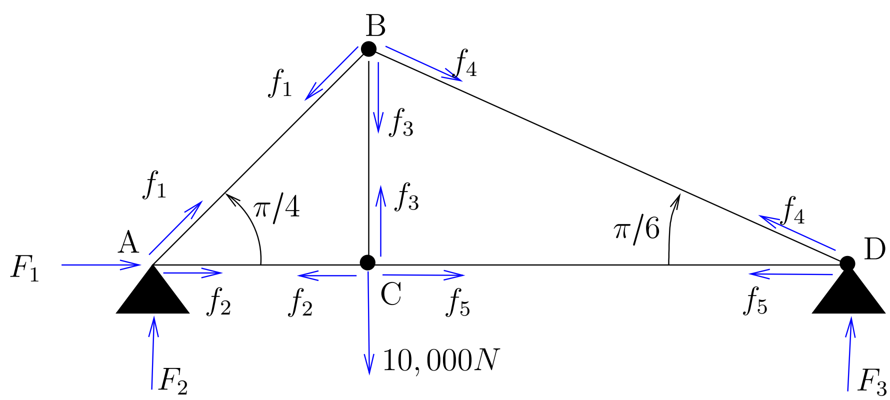
Question 3.7
Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ and A be an $n \times n$ matrix. Prove that $\|A\mathbf{x}\| \le \|A\| \, \|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{R}^n$. Moreover, prove that $\|A\|$ is the smallest number C such that $\|A\mathbf{x}\| \le C\,\|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{R}^n$.

Question 3.8
Consider the system of linear equations

$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1 \\ 2x_1 + x_2 - 4x_3 &= 0 \\ x_1 + 3x_2 - x_3 &= 1 \end{aligned}$$

a) Rewrite this system in the form $A\mathbf{x} = \mathbf{b}$ for which the Gauss-Seidel iterative method converges. You must proof the convergence.
b) Use the Gauss-Seidel iterative method to approximate the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5} if the infinite norm is used. Start with $\mathbf{x}_0 = \mathbf{0} \in \mathbb{R}^3$.

Question 3.9
The following figure illustrates a simple bridge truss.



A load of 10,000 Newtons is at the joint C. At each joint, the horizontal and vertical components of the resultant internal forces must be zero. Verify that the horizontal components of the resultant internal forces are

$$F_1 + \frac{\sqrt{2}}{2} f_1 + f_2 = 0 \ , \quad -\frac{\sqrt{2}}{2} f_1 + \frac{\sqrt{3}}{2} f_4 = 0 \ , \quad -f_2 + f_5 = 0 \quad \text{and} \quad -\frac{\sqrt{3}}{2} f_4 - f_5 = 0$$

at A, B, C and D respectively. Verify that the vertical components of the resultant internal forces are

$$F_2 + \frac{\sqrt{2}}{2} f_1 = 0 \ , \quad -\frac{\sqrt{2}}{2} f_1 - f_3 + \frac{1}{2} f_4 = 0 \ , \quad f_3 - 10,000 = 0 \quad \text{and} \quad F_3 - \frac{1}{2} f_4 = 0$$

at A, B, C and D respectively. To be complete, the problem should also consider the horizontal and vertical components of the resultant external forces, and the sum of the moments must be zero. We will not consider these equations.

a) Use Jacobi iterative method to approximate the solution of this system of forces to within 10^{-3}.
b) Use Gauss-Seidel iterative method to approximate the solution of this system of forces to within 10^{-3}.
c) Use a relaxation method to approximate the solution of this system of forces to within 10^{-3}.

Start the iteration with $f_i = F_j = 1$ for all i and j. Note that you may have to reorder the equations to ensure that the methods are applicable.

Question 3.10
Consider the linear system $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 4 & 1 & -1 & 1 \\ 1 & 5 & 1 & -1 \\ 2 & -1 & 6 & 2 \\ -1 & 1 & -2 & 5 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 5 \\ 2 \\ -14 \\ 25 \end{pmatrix} .$$

a) Show that both Jacobi and Gauss-Seidel iteration methods converge.
b) Use Jacobi iteration method to approximate the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}.
c) Use Gauss-Seidel iteration method to approximate the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}.
d) Use a relaxation method to approximate the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}. You must first show that the method converges with your choice of ω. Experiment with different values of ω. For your choice of $\mathbf{x}_0$, determine roughly the value(s) of ω for which the relaxation method converges the fastest (i.e. with the smallest number of iterations to satisfy the accuracy.)

Question 3.11
Consider the iterative system $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c}$, where T is an $n \times n$ matrix whose spectral radius $\rho(T)$ is bigger or equal to 1. Give a vector $\mathbf{x}_0$ such that the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ does not converge to a solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ is there is a solution.

Question 3.12
a) Let A be an $n \times n$ upper-triangular matrix. Show that Jacobi iterative method converges to the solution of $A\mathbf{x} = \mathbf{b}$ for any initial vector $\mathbf{x}_0$.
b) Suppose that

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} .$$

Choose any initial vector $\mathbf{x}_0$ and show that only a finite number of iterations of Jacobi iterative method is necessary to get the solution of $A\mathbf{x} = \mathbf{b}$.
c) If A is a general $n \times n$ upper-triangular matrix and $\mathbf{b} \in \mathbb{R}^n$, show that a finite number of iterations of the Jacobi iterative method is sufficient to get the solution of $A\mathbf{x} = \mathbf{b}$.

Question 3.13
Let A be an $n \times n$ upper-triangular matrix and $\mathbf{b} \in \mathbb{R}^n$, show that the Gauss-Seidel iterative method converges to the solution of $A\mathbf{x} = \mathbf{b}$ for any initial vector $\mathbf{x}_0$, and that it does so in a finite number of iterations

Question 3.14
Suppose that $A = \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix}$ and $\mathbf{x}_0$ is any vector in $\mathbb{R}^2$.

a) Show that the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ generated by the relaxation method converges to the solution of $A\mathbf{x} = \mathbf{b}$ whatever the choice of $\mathbf{x}_0$ if and only if $\omega \in]0, 2[$
b) What is the optimal value of ω; namely, what is the value of ω for which we expect the fastest convergence?
c) If $\omega \notin]0, 2[$, show that there exists $\mathbf{x}_0$ for which the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ generated by the relaxation method does not converge. So, it certainly does not converge to a solution of $A\mathbf{x} = \mathbf{b}$. Give such a vector $\mathbf{x}_0$.

Question 3.15
A variant of the steepest descent method presented in in Algorithm 3.5.2 is to replace the second step by

2'. If $\mathbf{b} \neq A\mathbf{x}_k$, let $\mathbf{u}_k = \mathbf{b} - A\mathbf{x}_k$

Obviously, if $\mathbf{b} = A\mathbf{x}_k$, then we have the solution $\mathbf{x}_k$ and we stop the iteration.

a) Prove that $\mathbf{u}_k$ is parallel to the gradient of $g(x) = \langle \mathbf{x}, A\mathbf{x} \rangle - 2\langle \mathbf{x}, \mathbf{b} \rangle$ at $\mathbf{x} = \mathbf{x}_k$. Therefore, perpendicular to the level curve of g at $\mathbf{x} = \mathbf{x}_k$.
b) Prove that $\mathbf{u}_{k+1}$ is perpendicular to $\mathbf{u}_k$.
c) Draw a figure similar to Figure 3.1 to illustrate this version of the steepest descent method.

Question 3.16
Prove that if $\mathbf{u}_k = \mathbf{0}$ in Algorithm 3.5.4, then $A\mathbf{x}_k = \mathbf{b}$.

Question 3.17
If A is a strictly positive definite matrix and $\mathbf{b}$ is a given vector. Show that the scalar product of the residual error $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ and the error vector $\mathbf{e} = A^{-1}\mathbf{b} - \mathbf{x}$ is positive unless $A\mathbf{x} = \mathbf{b}$.

Chapter 4

Algebraic Methods to Solve Systems of Linear Equations

As in the previous chapter, our goal is to numerically solve the system of linear equations $A\mathbf{x} = \mathbf{b}$, where A is an invertible $n \times n$ matrix and $\mathbf{b} \in \mathbb{R}^n$ is given. However, we will only consider classical direct methods in this chapter.

4.1 Gaussian Elimination with Backward Substitution

Gaussian elimination is a well known method to solve systems of linear equations of the form

$$A\mathbf{x} = \mathbf{b} \ , \tag{4.1.1}$$

where

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \quad , \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \ .$$

We assume that A is an invertible matrix. Hence, the solution exists and is unique.

We first review the Gaussian elimination method before implementing it. The **augmented matrix** associated to the system (4.1.1) is the matrix

$$\begin{bmatrix} A & \mathbf{b} \end{bmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} & b_n \end{pmatrix} \ .$$

Let $M(1) = \begin{bmatrix} A & \mathbf{b} \end{bmatrix}$. Suppose that, after several row operations, we have the matrix

$$M(k) = \left(\begin{array}{cccccccc|c} a_{1,1}^{[1]} & a_{1,2}^{[1]} & \ldots & a_{1,k-1}^{[1]} & a_{1,k}^{[1]} & \ldots & a_{1,n}^{[1]} & & b_1^{[1]} \\ 0 & a_{2,2}^{[2]} & \ldots & a_{1,k-1}^{[2]} & a_{1,k}^{[2]} & \ldots & a_{2,n}^{[2]} & & b_2^{[2]} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ldots & \vdots & & \vdots \\ \vdots & \vdots & \ddots & a_{k-1,k-1}^{[k-1]} & a_{k-1,k}^{[k-1]} & \ldots & a_{k-1,n}^{[k-1]} & & b_{k-1}^{[k-1]} \\ \vdots & \vdots & \vdots & 0 & a_{k,k}^{[k]} & \ldots & a_{k,n}^{[k]} & & b_k^{[k]} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & \ldots & \ldots & 0 & a_{n,k}^{[k]} & \ldots & a_{n,n}^{[k]} & & b_n^{[k]} \end{array}\right).$$

We may assume that $a_{k,k}^{[k]} \neq 0$. If $a_{k,k}^{[k]} = 0$, there exists $i > k$ such that $a_{i,k}^{[k]} \neq 0$ because A is invertible. We interchange the k^{th} and i^{th} rows.

To get $M(k+1)$ from $M(k)$, we subtract $a_{i,k}^{[k]}/a_{k,k}^{[k]}$ times the k^{th} row from the i^{th} row and write the result back in the i^{th} row for each $i > k$. Namely,

$$a_{i,j}^{[k+1]} = a_{i,j}^{[k]} - \frac{a_{i,k}^{[k]}}{a_{k,k}^{[k]}} a_{k,j}^{[k]} \quad \text{and} \quad b_i^{[k+1]} = b_i^{[k]} - \frac{a_{i,k}^{[k]}}{a_{k,k}^{[k]}} b_k^{[k]} \tag{4.1.2}$$

for $i = k+1, k+2, \ldots, n$ and $j = k+1, k+2, \ldots, n$. We have $a_{i,k}^{[k+1]} = 0$ for $i > k$. Repeating these operations from $k = 1$ to $k = n-1$, we get

$$M(n) = \left(\begin{array}{ccccc|c} a_{1,1}^{[1]} & a_{1,2}^{[1]} & \ldots & a_{1,n-1}^{[1]} & a_{1,n}^{[1]} & b_1^{[1]} \\ 0 & a_{2,2}^{[2]} & \ldots & a_{2,n-1}^{[2]} & a_{2,n}^{[2]} & b_2^{[2]} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & a_{n-1,n-1}^{[n-1]} & a_{n-1,n}^{[n-1]} & b_n^{[n-1]} \\ 0 & \ldots & \ldots & 0 & a_{n,n}^{[n]} & b_n^{[n]} \end{array}\right).$$

To compute x_i for $1 \leq i \leq n$, we use **backward substitution**; namely,

$$x_n = \frac{b_n^{[n]}}{a_{n,n}^{[n]}}$$

and

$$x_k = \frac{b_k^{[k]} - a_{k,k+1}^{[k]} x_{k+1} - a_{k,k+2}^{[k]} x_{k+2} - \ldots - a_{k,n}^{[k]} x_n}{a_{k,k}^{[k]}} \tag{4.1.3}$$

for $k = n-1, n-2, \ldots, 1$.

The following code implements Gaussian elimination with backward substitution.

Code 4.1.1 (Gaussian Elimination with Backward Substitution)

To compute the solution of the linear system of equations $A\mathbf{x} = \mathbf{b}$, where A is invertible.
Input: The matrix A and the column vector $\mathbf{b}$.
Output: The solution $\mathbf{x}$ of the system (in theory).

```
% function x = gauss(A,b)
```

```
function x = gauss(A,b)
  dim = size(A,1);
  x = NaN;

  % To avoid expensive row interchanges, we only interchange the
  % indices of the rows.  We create the vector N = ( 1 2 3 ... dim )
  % to keep track of the permutations of the rows.  N(i) will contain the
  % index of the row in the original matrix  A  which is now located
  % in row  i .
  N=linspace(1,dim,dim);

  for k = 1:dim-1
    % We find the smallest index j such that A^k_{j,k} is non null.
    j = k;
    while (A(N(j),k) == 0)
      j = j+1;
      if (j > dim)
        % A is not invertible.
        return;
      end
    end

    % We interchange the k'th and j'th rows.
    temp = N(j);
    N(j) = N(k);
    N(k) = temp;

    % We eliminate the entries in the k'th column which are in the
    % rows below the k'th row.
    for i = k+1:dim
      m = A(N(i),k)/A(N(k),k);
      A(N(i),k+1:dim) = A(N(i),k+1:dim) - m*A(N(k),k+1:dim);
      b(N(i),1) = b(N(i),1)-m*b(N(k),1);
    end
  end

  % We use backward substitution,
  x(dim,1) = b(N(dim),1)/A(N(dim),dim);
  for k = dim-1:-1:1
    x(k)= (b(N(k),1) - A(N(k),k+1:dim)*x(k+1:dim,1))/A(N(k),k);
  end
end
```

Remark 4.1.2
If $|a^{[k]}_{i,k}| \gg |a^{[k]}_{k,k}|$, then $a^{[k]}_{i,k}/a^{[k]}_{k,k}$ is very large. Hence, when performing (4.1.2), we may magnify the rounding error. Moreover, when performing the backward substitution (4.1.3), we may

also magnify the rounding error if we divide by a small number $a_{k,k}^{[k]}$.

A strategy to minimize the problem with rounding error is to use **maximal column pivoting** also called **partial pivoting**. Before performing (4.1.2), we choose the index i such that

$$|a_{i,k}^{[k]}| = \max_{k\leq j\leq n} |a_{j,k}^{[k]}|$$

and interchange the i^{th} and k^{th} rows.

Another strategy to minimize the problem with rounding error is to use **scaled column pivoting**. This time, before performing (4.1.2), we choose the index i such that

$$\frac{|a_{i,k}^{[k]}|}{\max_{k\leq j\leq n} |a_{i,j}^{[k]}|} \geq \frac{|a_{s,k}^{[k]}|}{\max_{k\leq j\leq n} |a_{s,j}^{[k]}|}$$

for $k \leq s \leq n$, and interchange the i^{th} and k^{th} rows.

There is another strategy which is better than the previous two but is not often used because of the number of operations needed to perform it. In **total pivoting**, before performing (4.1.2), we choose the indices i and j such that

$$|a_{i,j}^{[k]}| = \max_{\substack{k\leq s\leq n\\ k\leq r\leq n}} |a_{s,r}^{[k]}|$$

and interchange the i^{th} and k^{th} rows and the j^{th} and k^{th} columns. With this strategy, the indices of $\mathbf{x}$ also have to be permuted. ♠

Example 4.1.3
Consider the system

$$\begin{aligned} 3x_1 + 15660x_2 &= 15690 \\ 0.3454x_1 - 2.436x_2 &= 1.018 \end{aligned}$$

The exact solution is $x_1 = 10$ and $x_2 = 1$.

We first solve this system using Gaussian elimination with backward substitution, without row interchange and with 5-digit rounding arithmetic.

$$M(1) = \begin{pmatrix} 0.3\times 10 & 0.1566\times 10^5 & 0.1569\times 10^5 \\ 0.3454 & -0.2436\times 10 & 0.1018\times 10 \end{pmatrix} .$$

Let

$$m = \frac{0.3454}{0.3\times 10} \approx 0.11513$$

To get $M(2)$, we subtract m times the first row from the second row and write the result in the second row.

$$M(2) = \begin{pmatrix} 0.3\times 10 & 0.1566\times 10^5 & 0.1569\times 10^5 \\ 10^{-5} & -0.18053\times 10^4 & -0.18054\times 10^4 \end{pmatrix} .$$

Using backward substitution, we get

$$x_2 \approx \frac{-0.18054 \times 10^4}{-0.18053 \times 10^4} \approx 0.10001 \times 10$$

and

$$x_1 \approx \frac{0.1569 \times 10^5 - 0.1566 \times 10^5 \times 0.10001 \times 10}{0.3 \times 10} \approx 0.93333 \times 10 \ .$$

We have a good approximation of x_2 but a bad approximation of x_1.

If we use maximal column pivoting, we get the same answer because the method does not require to interchange the rows.

If we use scaled column pivoting, we have to interchange the rows because

$$\frac{|a_{2,1}^{[1]}|}{\max_{1\leq j\leq 2} |a_{2,j}^{[1]}|} = \frac{1727}{12180} > \frac{1}{5220} = \frac{|a_{1,1}^{[1]}|}{\max_{1\leq j\leq 2} |a_{1,j}^{[1]}|} \ .$$

Hence

$$M(1) = \begin{pmatrix} 0.3454 & -0.2436 \times 10 & 0.1018 \times 10 \\ 0.3 \times 10 & 0.1566 \times 10^5 & 0.1569 \times 10^5 \end{pmatrix} \ .$$

Let

$$m = \frac{0.3 \times 10}{0.3454} \approx 0.86856 \times 10 \ .$$

To get $M(2)$, we subtract m times the first row from the second row and write the result in the second row.

$$M(2) = \begin{pmatrix} 0.3454 & -0.2436 \times 10 & 0.1018 \times 10 \\ 0 & 0.15681 \times 10^5 & 0.15681 \times 10^5 \end{pmatrix} \ .$$

Using backward substitution, we get

$$x_2 \approx \frac{0.15681 \times 10^5}{0.15681 \times 10^5} \approx 1$$

and

$$x_1 \approx \frac{0.1018 \times 10 + 0.2436 \times 10 \times 11}{0.3454} \approx 10 \ .$$

We get the exact values of x_1 and x_2. ♣

We now give the code that implements Gaussian elimination with backward substitution, and maximum column pivoting or scaled column pivoting.

Code 4.1.4 (Gaussian Elimination with Backward Substitution and Pivoting Strategy)

To compute the solution of the linear system of equations $A\mathbf{x} = \mathbf{b}$, where A is invertible. Maximal column or scaled column pivoting can be used.
Input: The matrix A and the column vector $\mathbf{b}$.
The option selected: maximal column or scaled column pivoting.
Output: The solution $\mathbf{x}$ of the system (in theory).

```
%  x = gauss(A,b,option)
%
%  We use gaussian elimination with maximal column pivoting
%  (obtion = 1) or scaled column pivoting (obtion = 2) to solve
%  a system of linear equations of the form
%
%    A(1,1)*x(1)     + ... + A(1,dim)*x(dim)   = b(1,:)
%           . . . .
%    A(dim,1)*x(x(1) + ... + A(dim,dim)*x(dim) = b(dim,:)
%
%  The following must be given:
%    The matrix A
%    The matrix ( b(:,i) ) for i = 1, 2, ..., M ; the M linear
%      systems A x = b(:,i) are solved simultaneously.
%    The option  option  chosen: option = 1 for partial column
%      pivoting and option = 2 for scaled column pivoting.
%
%  The program gives an approximation  x(:,i)  of the solution of
%  the linear system associated to b(:,i) for i=1, 2, ..., M.

function x = gauss(A,b,option)
  dim = size(A,1);
  x = NaN;

  if ( (option ~= 1) & (option ~= 2) )
    disp 'There is no such algorithm.';
    return;
  end

  % To avoid expensive row interchanges, we only interchange the
  % indices of the rows.  We create the vector N = ( 1 2 3 ... dim )
  % to keep track of the permutations of the rows.  N(i) will contain the
  % index of the row in the original matrix  A  which is now located
  % in row  i .
  N = linspace(1,dim,dim);

  % We use gaussian elimination to write the system in echelon form.
```

```
for k=1:(dim-1)
  % If option = 1, then we use the maximal colum pivoting algorithm.
  % If option = 2, then we use the scaled column pivoting algorithm.

  if (option == 1)
    j = k;
    max = abs( A(N(k),k) );
    for i=(k+1):dim
      if (abs( A(N(i),k) ) > max)
        max = abs( A(N(i),k) );
        j = i;
      end
    end
    if (max == 0)
      disp 'The matrix  A  is not invertible.';
      return;
    end
  else
    % We find the index j such that
    % |a^k_{j,k}|/max_{k\leq i \leq n}|a^k_{j,i}| >=
    %   |a^k_{s,k}|/max_{k\leq i \leq n}|a^k_{s,i}|
    % for k <= s <= dim.
    j = k;
    rowmax = norm(A(N(k),k:dim), inf);
    if (rowmax == 0)
      disp 'The matrix  A  is not invertible.';
      return;
    end
    max = abs( A(N(k),k) )/rowmax;
    for i=(k+1):dim
      rowmax = norm(A(N(i),k:dim), inf);
      if (rowmax == 0)
        disp 'The matrix  A  is not invertible.';
        return;
      end
      test = abs( A(N(i),k) )/rowmax;
      if (test > max)
        max = test;
        j = i;
      end
    end
  end

  % We interchange the k^{th} and j^{th} rows.
  if (k ~= j)
    ncopy = N(k);
```

```
      N(k) = N(j);
      N(j) = ncopy;
    end

    for i=(k+1):dim
      m = A(N(i),k)/A(N(k),k);
      A(N(i),(k+1):dim) = A(N(i),(k+1):dim) - m*A(N(k),(k+1):dim);
      b(N(i),:)=b(N(i),:) - m*b(N(k),:);
    end
  end

  % We now use backward substitution to get an approximation of the
  % solution of the system.
  x(dim,:) = b(N(dim),:)/A(N(dim),dim);
  for i=(dim-1):-1:1
    x(i,:) = b(N(i),:);
    for j=(i+1):dim
      x(i,:) = x(i,:) - A(N(i),j)*x(j,:);
    end
    x(i,:) = x(i,:)/A(N(i),i);
  end
end
```

4.2 LU Factorization

We consider a system of linear equation of the form (4.1.1) where A is an $n \times n$ invertible matrix and $\mathbf{b}$ is an $n \times 1$ column vector.

Suppose that we can write A as the product PLU where P is a permutation matrix, L is an invertible lower-triangular matrix and U is an invertible upper-triangular matrix. It is then easy to solve (4.1.1). We first solve $L\mathbf{y} = \mathbf{c} = P^{-1}\mathbf{b}$. Note that $\mathbf{c}$ is obtained from $\mathbf{b}$ by permuting the indices of $\mathbf{b}$. The solution of $A\mathbf{x} = \mathbf{b}$ is then the solution of $U\mathbf{x} = \mathbf{y}$.

To solve $L\mathbf{y} = \mathbf{c}$, we use **forward substitution** as implemented in the next code.

Code 4.2.1 (Forward Substitution)

To solve $L\mathbf{y} = \mathbf{c}$ where L is an invertible lower-triangular matrix.
Input: The matrix L and the column vector $\mathbf{c}$.
Output: The solution $\mathbf{y}$ of the system.

```
% y = forward(L,c)

function y = forward(L,c)
  dim = size(L,1);
  y(1,1) = c(1,1)/L(1,1);
```

```
  for i = 2:dim
    y(i,1) = (c(i,1) - L(i,1:i-1)*c(1:i-1))/L(i,i);
  end
end
```

To solve $U\mathbf{x} = \mathbf{y}$, we use backward substitution as implemented in the next code.

Code 4.2.2 (Backward Substitution)

To solve $U\mathbf{x} = \mathbf{y}$ where U is an invertible upper-triangular matrix.
Input: The matrix U and the column vector $\mathbf{y}$.
Output: The solution $\mathbf{x}$ of the system.

```
% x = backward(U,y)

function x = backward(U,y)
  dim = size(U,1);
  x(dim,1) = y(dim,1)/U(dim,dim);
  for i = dim-1:-1:1
    x(i,1) = (y(i,1) - U(i,i+1:dim)*y(1:i+1:dim,1))/U(i,i);
  end
end
```

The matrices P, L and U are obtained from the Gaussian elimination procedure described in the previous section. Using the same notation than in the previous section, the matrices U and L are respectively given by $u_{i,j} = a_{i,j}^{[i]}$ and $\ell_{i,j} = a_{i,j}^{[j]}/a_{j,j}^{[j]}$. Recall that $a_{i,j}^{[i]} = 0$ and $\ell_{i,j} = 0$ if $j < i$.

We prove that $A = LU$. To simplify the discussion, we assume for now that no row-interchange has been used. From (4.1.2), we get

$$a_{i,j}^{[k+1]} = a_{i,j}^{[k]} - \frac{a_{i,k}^{[k]}}{a_{k,k}^{[k]}} a_{k,j}^{[k]} = a_{i,j}^{[k]} - \ell_{i,k} a_{k,j}^{[k]}$$

for $i = k+1,\ k+2,\ \ldots,\ n$ and $j = 1,\ 2,\ \ldots,\ n$. Note that we only have null values for $1 \leq j < k+1$. If we substitute k for $k-1$ in this formula, we get

$$a_{i,j}^{[k]} = a_{i,j}^{[k-1]} - \ell_{i,k-1} a_{k-1,j}^{[k-1]}$$

for $i = k,\ k+1,\ \ldots,\ n$ and $j = 1,\ 2,\ \ldots,\ n$. Hence,

$$a_{i,j}^{[k+1]} = a_{i,j}^{[k-1]} - \ell_{i,k-1} a_{k-1,j}^{[k-1]} - \ell_{i,k} a_{k,j}^{[k]}$$

for $i = k+1,\ k+2,\ \ldots,\ n$ and $j = 1,\ 2,\ \ldots,\ n$. By induction,

$$a_{i,j}^{[k+1]} = a_{i,j}^{[1]} - \ell_{i,1} a_{1,j}^{[1]} - \ell_{i,2} a_{2,j}^{[2]} - \ldots - \ell_{i,k} a_{k,j}^{[k]}$$

for $i = k+1, k+2, \ldots, n$ and $j = 1, 2, \ldots, n$. In particular, if $i = k+1$, we get

$$a_{k+1,j}^{[k+1]} = a_{k+1,j}^{[1]} - \ell_{k+1,1}a_{1,j}^{[1]} - \ell_{k+1,2}a_{2,j}^{[2]} - \ldots - \ell_{k+1,k}a_{k,j}^{[k]}$$

for $j = 1, 2, \ldots, n$. Thus

$$a_{k+1,j}^{[1]} = \ell_{k+1,1}a_{1,j}^{[1]} + \ell_{k+1,2}a_{2,j}^{[2]} + \ldots + \ell_{k+1,k}a_{k,j}^{[k]} + a_{k+1,j}^{[k+1]}$$

for $j = 1, 2, \ldots, n$. Because $\ell_{k+1,k+1} = 1$ and $\ell_{k+1,s} = 0$ for $s > k+1$, we get

$$a_{k+1,j}^{[1]} = \sum_{s=1}^{n} \ell_{k+1,s}a_{s,j}^{[s]} = \sum_{s=1}^{n} \ell_{k+1,s}u_{s,j} \tag{4.2.1}$$

for $j = 1, 2, \ldots, n$. Since (4.2.1) is true for $k = 0, 1, \ldots, n-1$, we get $A = LU$.

If row interchanges are needed in Gaussian elimination, these row interchanges can be performed on A and $\mathbf{b}$ before starting Gaussian elimination. Let P^{-1} be the permutation matrix that performs all the needed row interchanges. Note that $P = P^{-1}$. From our previous discussion, we can write $P^{-1}A$ as $P^{-1}A = LU$ for a lower-triangular matrix L and an upper-triangular matrix U. No row interchange is needed to reduce $P^{-1}A$ to an upper-triangular matrix using Gaussian elimination. Hence $A = PLU$.

To solve $A\mathbf{x} = \mathbf{b}$, we only have to solve $LU\mathbf{x} = P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$. From a computational point of view, the row interchanges do not cause any problem. The formulae for U and L given above are still valid if we performed the same row interchanges on the vector $\mathbf{b}$ than the ones performed during Gaussian elimination.

Code 4.2.3 (LU Decomposition)

To approximate the solution of the linear system of equations $A\mathbf{x} = \mathbf{b}$, where A is invertible. Maximal column or scaled column pivoting can be used.
Input: The matrix A and the column vector $\mathbf{b}$.
Output: The solution $\mathbf{x}$ of the system (in theory).

```
%  x = LUfactor(A,b,option)
%
%  We use PLU factorization with maximal column pivoting
%  (option 1) or scaled column pivoting (option 2) to solve
%  a system of linear equations of the form
%
%    A(1,1)*x(1)      + ... + A(1,dim)*x(dim)   = b(1,:)
%    . . .
%    A(dim,1)*x(x(1) + ... + A(dim,dim)*x(dim) = b(dim,:)
%
%  The following must be given:
%    The square matrix A
%    The matrix ( b(:,i) ) for i=1, 2, ..., M ; the M linear
%      systems A x = b(:,i) are solved simultaneously.
%    The option  option  chosen: option = 1 for maximal column
%      pivoting and option = 2 for scaled column pivoting.
```

```
%
%  The program gives an approximation  x(:,i)  of the solution of
%  the linear system A x = b(:,i) for i=1, 2, ..., M.

function x = LUfactor(A,b,option)
  dim = size(A,1);
  x = NaN;

  if ( (option ~= 1) & (option ~= 2) )
    disp 'There is no such algorithm.';
    return;
  end

  % To avoid expensive row interchanges, we only interchange the
  % indices of the rows.  We create the vector N = ( 1 2 3 ... dim )
  % to keep track of the permutations of the rows.  N(i) will contain the
  % index of the row in the original matrix  A  which is now located
  % in row  i .
  N=linspace(1,dim,dim);

  % We compute the entries of  U  and  L .
  for k=1:(dim-1)
    % If option = 1, then we use the maximal colum pivoting algorithm.
    % If option = 2, then we use the scaled column pivoting algorithm.

    if (option==1)
      j = k;
      max = abs( A(N(k),k) );
      for i=(k+1):dim
        if (abs( A(N(i),k) ) > max)
          max = abs( A(N(i),k) );
          j = i;
        end
      end
      if (max == 0)
        disp 'The matrix  A  is not invertible.';
        return;
      end
    else
      % We find the index k such that
      % |a^k_{k,k}|/\max_{k\leq i \leq n}|a^k_{k,i}| >=
      %   |a^k_{s,k}|/\max_{k\leq i \leq n}|a^k_{s,i}|
      % for k <= s <= dim.
      j = k;
      rowmax = norm(A(N(k),k:dim),inf);
      if (rowmax == 0)
```

```
      disp 'The matrix  A  is not invertible.';
      return;
    end
    max = abs( A(N(k),k) )/rowmax;
    for i=(k+1):dim
      rowmax = norm(A(N(i),k:dim),inf);
      if (rowmax == 0)
        disp 'The matrix  A  is not invertible.';
        return;
      end
      test = abs( A(N(i),k) )/rowmax;
      if (test > max)
        max = test;
        j = i;
      end
    end
  end

  % We interchange the k'th and j'th rows.
  if (k ~= j)
    ncopy = N(k);
    N(k) = N(j);
    N(j) = ncopy;
  end

  % We perform the Gaussian elimination.
  % We store the factors l_{i,k} = A(N(i),k)/A(N(k),k)  used in
  % gaussian elimination for row  N(i)  in  A(N(i),k)  which
  % is zero after elimination.
  for i=(k+1):dim
    A(N(i),k) = A(N(i),k)/A(N(k),k);
    A(N(i),(k+1):dim) = A(N(i),(k+1):dim) ...
                          - A(N(i),k)*A(N(k),(k+1):dim);
  end
end

% Only at this point do we need the value of  b .
% We now use forward substitution to sole  Ly = c.
y(1,:) = b(N(1),:);
for i=2:dim
  y(i,:) = b(N(i),:);
  for j=1:(i-1)
    y(i,:) = y(i,:) - A(N(i),j)*y(j,:);
  end
end
```

```
  % We now use backward substitution to get an approximation of the
  % solution of the system.
  x(dim,:) = y(dim,:)/A(N(dim),dim);
  for i=(dim-1):-1:1
    x(i,:) = y(i,:);
    for j=(i+1):dim
      x(i,:) = x(i,:) - A(N(i),j)*x(j,:);
    end
    x(i,:) = x(i,:)/A(N(i),i);
  end
end
```

We did not call the functions defined in Codes 4.2.1 and 4.2.2 in the previous code to save storage space for the matrices and to take advantage of the special form of the lower-triangular matrix L that has 1 everywhere on the diagonal.

Moreover, b in the previous code can be a matrix. Thus, the previous code can be used to find the inverse of a matrix A by posing $b = \mathrm{Id}$.

4.3 Cholesky Factorization

This is a special case of the LU factorization. We assume that the matrix A in (4.1.1) is real, symmetric and strictly positive definite. It can then be proved that A has a LU factorization that does not need pivoting. The proof is based on the fact that all the sub-matrices $\begin{pmatrix} a_{1,1} & \dots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k} \end{pmatrix}$ for $k = 1, 2, \dots, n$ have positive determinants.

Suppose that $A = LU$ as in the previous section. Let

$$D = \begin{pmatrix} \sqrt{a_{1,1}^{[1]}} & 0 & \dots & 0 \\ 0 & \sqrt{a_{2,2}^{[2]}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sqrt{a_{n,n}^{[n]}} \end{pmatrix} ,$$

$M = LD$ and $N = D^{-1}U$. Recall that the elements on the diagonal of a strictly positive definite matrix are all positive numbers. Then $A = MN$, where M is lower-triangular and $N = M^{\top}$.

To prove that $M = N^{\top}$, we use the relation $A = MN$ to get

$$m_{k,k} = n_{k,k} = \sqrt{a_{k,k} - \sum_{i=1}^{k-1} m_{k,i} n_{i,k}} , \tag{4.3.1}$$

$$n_{k,j} = \frac{1}{m_{k,k}} \{a_{k,j} - \sum_{i=1}^{k-1} m_{k,i} n_{i,j}\} , \tag{4.3.2}$$

$$m_{j,k} = \frac{1}{n_{k,k}}\{a_{j,k} - \sum_{i=1}^{k-1} m_{j,i}n_{i,k}\} \tag{4.3.3}$$

and

$$m_{k,j} = n_{j,k} = 0$$

for $j > k \geq 1$. The summations in the formulae above are ignored when $k = 1$. It remains to show that $m_{j,k} = n_{k,j}$ for $j > k \geq 1$. We use induction on k. For $k = 1$, we have

$$n_{1,j} = \frac{a_{1,j}}{m_{1,1}} = \frac{a_{j,1}}{n_{1,1}} = m_{j,1}$$

for $j > 1$ because A is symmetric and $m_{1,1} = n_{1,1}$ from (4.3.1). We assume that $m_{j,i} = n_{i,j}$ for $j > i \geq 1$ and $i \leq k$ and show that $m_{j,k+1} = n_{k+1,j}$ for $j > k+1$. We rewrite (4.3.2) and (4.3.3) with k replaced by $k+1$ to get

$$n_{k+1,j} = \frac{1}{m_{k+1,k+1}}\{a_{k+1,j} - \sum_{i=1}^{k} m_{k+1,i}n_{i,j}\} \ , \tag{4.3.4}$$

and

$$m_{j,k+1} = \frac{1}{n_{k+1,k+1}}\{a_{j,k+1} - \sum_{i=1}^{k} m_{j,i}n_{i,k+1}\} \ . \tag{4.3.5}$$

Since $a_{k+1,j} = a_{j,k+1}$ because A is symmetric, $m_{k+1,i} = n_{i,k+1}$ for $1 \leq i \leq k$ and $m_{j,i} = n_{i,j}$ for $1 \leq i \leq k < j$ by induction, we get that the summations in (4.3.4) and (4.3.5) are equal for $j > k+1$.

From (4.3.1), (4.3.2) and (4.3.3), we can get the following implementation of the Cholesky factorization. This algorithm is faster than the previous algorithms to solve $A\mathbf{x} = \mathbf{b}$ with pivoting because it requires less computation. However, A has to be real symmetric and positive definite.

Code 4.3.1 (**Cholesky Factorization**)

To compute the solution of the linear system of equations $A\mathbf{x} = \mathbf{b}$, where A is real, symmetric and strictly positive definite.
Input: The matrix A and the column vector $\mathbf{b}$.
Output: The solution $\mathbf{x}$ of the system and the matrix M in $A = MN$.

```
% function x = cholesky(A,b)

function [x,M] = cholesky(A,b)
  dim = size(A,1);

  % In theory, we do not have to use pivoting.  Moreover, we only need
  % to compute  M  because  N  is the trampose of  M.
  M = zeros(dim,dim);
```

```
  M(1,1) = sqrt(A(1,1));
  M(2:dim,1) = A(2:dim,1)/M(1,1);
  for k = 2:dim-1
    M(k,k) = sqrt(A(k,k) - sum(M(k,1:k-1).^2));
    for j = k+1:dim
      M(j,k) = (A(j,k) - sum(M(j,1:k-1).*M(k,1:k-1)))/M(k,k);
    end
  end
  M(dim,dim) = sqrt(A(dim,dim) - sum(M(dim,1:dim-1).^2));

  % Only at this point do we need the value of  b .
  % We use forward substitution to sole  My = b.

  y(1,1) = b(1,1)/M(1,1);
  for i = 2:dim
    y(i,1) = (b(i,1) - M(i,1:i-1)*y(1:i-1,1) )/M(i,i);
  end

  % We now use backward substitution to get an approximation of the
  % solution of the system.

  x(dim,1) = y(dim,1)/M(dim,dim);
  for i = dim-1:-1:1
    x(i,1) = (y(i,1)-M(i+1:dim,i)'*x(i+1:dim,1))/M(i,i);
  end
end
```

4.4 Error estimates

Let $\mathbf{x}_a$ be an approximation of the solution $\mathbf{p}$ of (3.0.1) such that $\|\mathbf{b} - A\mathbf{x}_a\|$ is small. Is $\|\mathbf{p} - \mathbf{x}_a\|$ small?

Example 4.4.1

Consider the system $A\mathbf{x} = \mathbf{b}$ where $A = \begin{pmatrix} 3 & 6 \\ 2.9999 & 6 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 9 \\ 8.9999 \end{pmatrix}$. The unique solution is $\mathbf{p} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

If $\mathbf{x}_a = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$, let $\mathbf{r} = \mathbf{b} - A\mathbf{x}_a = \begin{pmatrix} 0 \\ 0.0002 \end{pmatrix}$. We have that $\|\mathbf{r}\|_\infty = 0.0002$ is a small number but $\|\mathbf{p} - \mathbf{x}_a\|_\infty = 2$ is a large number. ♣

Definition 4.4.2

Let A be an invertible $n \times n$ matrix.

1. If $\mathbf{x}_a$ is an approximation of the unique solution of $A\mathbf{x} = \mathbf{b}$, then the vector $\mathbf{r} = \mathbf{b} - A\mathbf{x}_a$ is called the **residual vector** for $\mathbf{x}_a$.
2. The **condition number** of A is the number $K(A) = \|A\|\,\|A^{-1}\|$.

Theorem 4.4.3

Let A be an invertible matrix and $\mathbf{x}_a$ be an approximation of the unique solution $\mathbf{p}$ of $A\mathbf{x} = \mathbf{b}$. The residual vector $\mathbf{r} = \mathbf{b} - A\mathbf{x}_a$ for A satisfies

$$\|\mathbf{x}_a - \mathbf{p}\| \leq K(A)\frac{\|\mathbf{r}\|}{\|A\|}$$

and

$$\frac{\|\mathbf{x}_a - \mathbf{p}\|}{\|\mathbf{p}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$$

if $\mathbf{p} \neq \mathbf{0}$.

Proof.
From $\mathbf{r} = \mathbf{b} - A\mathbf{x}_a = A(\mathbf{p} - \mathbf{x}_a)$, we get $\mathbf{p} - \mathbf{x}_a = A^{-1}\mathbf{r}$. Thus

$$\|\mathbf{p} - \mathbf{x}_a\| \leq \|A^{-1}\|\,\|\mathbf{r}\| = K(A)\frac{\|\mathbf{r}\|}{\|A\|} \,. \tag{4.4.1}$$

Since $\|\mathbf{b}\| = \|A\mathbf{p}\| \leq \|A\|\,\|\mathbf{p}\|$, we get

$$\frac{1}{\|A\|} \leq \frac{\|\mathbf{p}\|}{\|\mathbf{b}\|} \,.$$

If we combine this last inequality with (4.4.1), we get

$$\|\mathbf{p} - \mathbf{x}_a\| \leq K(A)\frac{\|\mathbf{r}\|\,\|\mathbf{p}\|}{\|\mathbf{b}\|}$$

and so

$$\frac{\|\mathbf{p} - \mathbf{x}_a\|}{\|\mathbf{p}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \,.$$

∎

Definition 4.4.4

An invertible matrix A is **well-conditioned** when $K(A)$ is small (near 1) and **ill-conditioned** otherwise.

Remark 4.4.5
Suppose that the matrix A in the statement of the previous theorem is a well-conditioned matrix, then the absolute error $\|\mathbf{x}_a - \mathbf{p}\|$ is small when the residual vector $\mathbf{r}$ is small. Moreover,

the relative error $\|\mathbf{x}_a - \mathbf{p}\|/\|\mathbf{p}\|$ is small when the relative size of the residual vector $\mathbf{r}$ with respect to the vector $\mathbf{b}$ is small. ♠

Example 4.4.6

In the previous example $A = \begin{pmatrix} 3 & 6 \\ 2.9999 & 6 \end{pmatrix}$. Hence $\|A\|_\infty = 9$.

Since $A^{-1} = \begin{pmatrix} 10^4 & -10^4 \\ -4999.8\overline{3} & 5000 \end{pmatrix}$, we have $\|A^{-1}\|_\infty = 2\times 10^4$. Thus $K(A) = 1.8\times 10^5$ is really large. A is ill-conditioned.

We have inequalities in Theorem 4.4.3 but we "may" in practice treat them as equalities because they suggest the potential for large errors as we have seen in the previous example. ♣

Due to rounding errors in representing on computers the entries of A and $\mathbf{b}$, solving numerically the system

$$A\mathbf{x} = \mathbf{b} \tag{4.4.2}$$

is equivalent to solving exactly the perturbed system

$$(A + \Delta A)\mathbf{x} = (\mathbf{b} + \Delta\mathbf{b}) \ , \tag{4.4.3}$$

where ΔA is an $n\times n$ matrix near the $n\times n$ null matrix and $\Delta\mathbf{b}$ is a vector of $\mathbb{R}^n$ near $\mathbf{0}\in\mathbb{R}^n$. The next theorem gives an estimate of the difference between the exact solution of (4.4.2) and the exact solution of (4.4.3).

Theorem 4.4.7

If $\|\Delta A\| < \|A^{-1}\|^{-1}$, we have that the exact solution $\mathbf{p}$ of (4.4.2) and the exact solution $\mathbf{q}$ of (4.4.3) satisfy

$$\frac{\|\mathbf{q}-\mathbf{p}\|}{\|\mathbf{p}\|} \leq \frac{K(A)}{1 - K(A)\|\Delta A\|/\|A\|}\left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\Delta A\|}{\|A\|}\right) \ .$$

Proof.

From Proposition 3.2.5, $\mathrm{Id}_n + A^{-1}\Delta A$ is invertible because

$$\|A^{-1}\Delta A\| \leq \|\Delta A\|\,\|A^{-1}\| < 1$$

by hypothesis. Moreover, Corollary 3.2.6 gives

$$\left\|\left(\mathrm{Id}_n + A^{-1}\Delta A\right)^{-1}\right\| \leq \frac{1}{1 - \|A^{-1}\Delta A\|} \leq \frac{1}{1 - \|A^{-1}\|\,\|\Delta A\|} \ . \tag{4.4.4}$$

Multiplying both sides of

$$(A + \Delta A)\left(\mathbf{p} + (\mathbf{q}-\mathbf{p})\right) = (\mathbf{b} + \Delta\mathbf{b})$$

from the left by A^{-1}, we get

$$\left(\mathrm{Id}_n + A^{-1}\Delta A\right)(\mathbf{q}-\mathbf{p}) + \mathbf{p} + A^{-1}\Delta A\,\mathbf{p} = \mathbf{p} + A^{-1}\,\Delta\mathbf{b}$$

because $A\mathbf{p} = \mathbf{b}$. Thus

$$\mathbf{q}-\mathbf{p} = \left(\mathrm{Id}_n + A^{-1}\Delta A\right)^{-1}\left(A^{-1}\Delta\mathbf{b} - A^{-1}\,\Delta A\,\mathbf{p}\right) .$$

Taking the norm on both sides, we get from (4.4.4) that

$$\begin{aligned}\|\mathbf{q}-\mathbf{p}\| &\leq \|\left(\mathrm{Id}_n + A^{-1}\Delta A\right)^{-1}\|\left(\|A^{-1}\|\,\|\Delta\mathbf{b}\| + \|A^{-1}\|\,\|\Delta A\|\,\|\mathbf{p}\|\right)\\ &\leq \frac{1}{1-\|A^{-1}\|\,\|\Delta A\|}\|A^{-1}\|\left(\|\Delta\mathbf{b}\| + \|\Delta A\|\,\|\mathbf{p}\|\right) .\end{aligned}$$

Thus

$$\begin{aligned}\frac{\|\mathbf{q}-\mathbf{p}\|}{\|\mathbf{p}\|} &\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\,\|\Delta A\|}\left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{p}\|} + \|\Delta A\|\right) \leq \frac{\|A^{-1}\|\,\|A\|}{1-\|A^{-1}\|\,\|\Delta A\|}\left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{p}\|\,\|A\|} + \frac{\|\Delta A\|}{\|A\|}\right)\\ &\leq \frac{K(A)}{1-K(A)\|\Delta A\|/\|A\|}\left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\Delta A\|}{\|A\|}\right) ,\end{aligned}$$

where we have used $K(A) = \|A\|\,\|A^{-1}\|$ and $\|\mathbf{b}\| \leq \|A\|\,\|\mathbf{p}\|$ from $\mathbf{b} = A\mathbf{p}$. ∎

Remark 4.4.8

1. Let A be an invertible $n\times n$ matrix and $\mathbf{p}$ be the unique solution of a system $A\mathbf{x} = \mathbf{b}$. It has been proved[1] that the residual vector $\mathbf{r}$ of $\mathbf{x}_a$ obtained from Gaussian elimination with backward substitution and t-digit rounding arithmetic satisfies

   $$\|\mathbf{r}\| \approx 10^{-t}\,\|A\|\|\mathbf{x}_a\| ,$$

   where $\mathbf{r}$ has been computed using $2t$-digit rounding arithmetic.

   Moreover, if $\mathbf{y}$ is (an approximation of) the solution of the equation $A\mathbf{x} = \mathbf{r}$, then

   $$\|\mathbf{y}\| \approx \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\|\,\|\mathbf{r}\| \approx \|A^{-1}\|\left(10^{-t}\,\|A\|\,\|\mathbf{x}_a\|\right) = 10^{-t}K(A)\|\mathbf{x}_a\| .$$

   Thus $10^t\dfrac{\|\mathbf{y}\|}{\|\mathbf{x}_a\|}$ may be used as an rough approximation of $K(A)$.

2. Let A be an invertible matrix. The method of **iterative refinement**, to numerically solve a system of the form $A\mathbf{x} = \mathbf{b}$ with accuracy ϵ, can be summarized as follows.

   (a) Using Gauss elimination with maximal column pivoting and single precision, find $\mathbf{x}_a$ such that $A\mathbf{x}_a \approx \mathbf{b}$.

   (b) Compute the residual vector $\mathbf{r} = \mathbf{b} - A\mathbf{x}_a$ in double precision. More precision most be used because the computations involve many almost identical numbers.

[1] Forsythe, G.E. and Moler, E.B., **Computer Solution of Linear Algebraic Systems**, Prentice-Hall, 1967

(c) Using Gauss elimination with maximal column pivoting and single precision, find $\mathbf{x_c}$ such that $A\mathbf{x_c} \approx \mathbf{r}$. The steps for this Gauss elimination are already known from (a).

(d) Let $\mathbf{x}_b = \mathbf{x}_a + \mathbf{x_c}$.

(e) If $10^t\|\mathbf{x_c}\|/\|\mathbf{x}_b\| < \epsilon$, the requested accuracy, then the vector $\mathbf{x}_b$ should be the desired approximation of the solution $\mathbf{p}$ of $A\mathbf{x} = \mathbf{b}$ and hopefully a better approximation of $\mathbf{p}$ than $\mathbf{x}_a$. If $10^t\|\mathbf{x_c}\|/\|\mathbf{x}_b\| \not< \epsilon$. replace $\mathbf{x}_a$ by $\mathbf{x}_b$, and repeat from step (b).

♠

4.5 Exercises

Question 4.1
The following questions could have come from a basic Linear algebra course.

a) Prove that if A and B are two $n \times n$ matrices such that AB is invertible, then A and B are invertible.
b) Prove that the product of two lower-triangular (resp. upper-triangular) matrices is a lower-triangular (resp. upper-triangular) matrix.
c) Suppose that A is a $n \times n$ invertible matrix. Prove that A^{-1} is lower-triangular (resp. upper-triangular) if A is lower-triangular (resp. upper-triangular).
d) Prove that the **triangular factorization** of a $n \times n$ matrix is unique; namely, prove that if an invertible matrix A can be expressed as $A = L_1U_1 = L_2U_2$, where L_1 and L_2 are two lower-triangular matrices with 1 as elements on the diagonal, and U_1 and U_2 are two upper-triangular matrices, then $L_1 = L_2$ and $U_1 = U_2$.
e) If A is $n \times n$ symmetric matrix and $A = LU$, where L is lower-triangular with 1 as elements on its diagonal and U is upper-triangular, prove that $U = DL^\top$, where D is a diagonal matrix whose diagonal is the diagonal of the matrix U.
f) A matrix A is **tridiagonal** if $a_{i,j} = 0$ for $|i - j| \geq 2$. If A is $n \times n$ tridiagonal matrix and $A = LU$, where L is lower-triangular and U is upper-triangular, prove that L and U are also tridiagonal.

Question 4.2
Suppose that A is a $n \times n$ symmetric matrix.

a) If Gauss elimination without pivoting is used on the first column of A to reduce it to the $n \times n$ matrix B. Prove that the $(n-1) \times (n-1)$ matrix obtained from B by removing the first column and the first row is also symmetric.
b) Use the result in (a) to write an algorithm to solve the system $A\mathbf{x} = \mathbf{b}$ that reduce the number of operations by about half.

Question 4.3
Consider the matrix

$$A = \begin{pmatrix} 2 & 4 & 3 \\ 2.001 & 4 & 3 \\ 0 & 2 & 1 \end{pmatrix} .$$

Use Gaussian elimination with backward substitution and scaled column pivoting to compute the inverse of A. Use 5-digit rounding arithmetic. Compute the condition number of A. Is A ill-conditioned?
Hint: the i^{th} column of A^{-1} is the solution of $A\mathbf{x} = \mathbf{e}_i$.

Question 4.4
If A is an invertible matrix, prove that the condition number $K(A)$ satisfies $K(A) \geq 1$.

Question 4.5
Consider the system of linear equations $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 1 & 2 \\ 1.00001 & 2 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 3 \\ 3.00001 \end{pmatrix}.$$

Use row operations and 7-digit rounding arithmetic to compute the solution $\mathbf{x}_p$ of the perturbed system $A\mathbf{x} = \mathbf{b}_p$, where $\mathbf{b}_p = \begin{pmatrix} 3.00001 & 3.00003 \end{pmatrix}^{\top}$. Compare $\mathbf{x}_p$ with the solution $\mathbf{x}_s = \begin{pmatrix} 1 & 1 \end{pmatrix}^{\top}$ of the unperturbed system. Compute the condition number using the ℓ^{∞}-norm. Is the system ill-conditioned or well-conditioned?

Question 4.6
Let A be the $n \times n$ lower-triangular matrix defined by

$$a_{i,j} = \begin{cases} 0 & \text{if } j > i \\ 1 & \text{if } j = i \\ -1 & \text{if } j < i \end{cases}$$

Compute the condition number $K(A) = \|A\|_{\infty} \|A^{-1}\|_{\infty}$. Is A well conditioned?

Question 4.7
Consider the system of linear equations $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 0.04 & 0.01 & -0.01 \\ 0.2 & 0.5 & -0.2 \\ 1 & 2 & 4 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 0.0601 \\ 0.302 \\ 11.03 \end{pmatrix}.$$

Suppose that $\mathbf{q} = \begin{pmatrix} 1.8 \\ 0.64 \\ 1.9 \end{pmatrix}$ is an approximation of the solution $\mathbf{p} = \begin{pmatrix} 1.83 \\ 0.66 \\ 1.97 \end{pmatrix}$. Without computing A^{-1}, can you determine if the system is ill-conditioned or well-conditioned?

Chapter 5

Iterative Methods to Solve Systems of Nonlinear Equations

The problem is to find the solutions of the equation

$$f(\mathbf{x}) = \mathbf{0} \ , \tag{5.0.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a given function. Namely, we have to find the vectors $\mathbf{p} \in \mathbb{R}^n$ such that $f(\mathbf{p}) = 0$. As for real-valued functions, the vectors $\mathbf{p}$ are called the **roots** or **zeros** of f.

5.1 Fixed Point Method

To find a root of f, we rewrite (5.0.1) as

$$\mathbf{x} = g(\mathbf{x}) \ , \tag{5.1.1}$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$, and a fixed point of g is a root of f and vice-versa. Recall that a vector $\mathbf{p} \in \mathbb{R}^n$ is a **fixed point** of g if $g(\mathbf{p}) = \mathbf{p}$. We say that (5.0.1) and (5.1.1) are **equivalent** (on a given set) if a root of f is a fixed point of g and vice-versa.

Given $\mathbf{x}_0$, we hope that the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}_{k+1} = g(\mathbf{x}_k) \quad , \quad k = 0, 1, 2, \ldots \tag{5.1.2}$$

will converge to a fixed point $\mathbf{p}$ of g and therefore a root of f. The problem is to choose g and $\mathbf{x}_0$ adequately.

The following theorem gives conditions that guarantee the convergence of the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ defined by (5.1.2) to a fixed point of g.

Theorem 5.1.1 (Fixed Point Theorem for Mappings)

Let S be a closed and bounded subset of $\mathbb{R}^n$ and suppose that $g : S \to \mathbb{R}^n$ satisfies:

1. $g(\mathbf{x}) \in S$ for all $x \in S$.
2. There exists $0 < K < 1$ such that $\|g(\mathbf{x}) - g(\mathbf{y})\| \leq K\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}$ and $\mathbf{y}$ in S.

Then g has a unique fixed point $\mathbf{p} \in S$ and, given $\mathbf{x}_0 \in S$, the sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ defined by (5.1.2) converges to $\mathbf{p}$. Moreover,

$$\|\mathbf{x}_k - \mathbf{p}\| \leq \frac{K^k}{1-K}\|\mathbf{x}_1 - \mathbf{x}_0\| \ .$$

Remark 5.1.2

1. The proof of Theorem 5.1.1 is identical to the proof of the Fixed Point Theorem, Theorem 2.4.2, for $g : \mathbb{R} \to \mathbb{R}$ if the absolute value is replaced by the norm.
2. Suppose that $g : S \to \mathbb{R}^n$ is continuously differentiable; namely, that all the partial derivatives $\dfrac{\partial g_i}{\partial x_j}(\mathbf{x})$ of g exist and are continuous in S. Therefore the derivative $\mathrm{D}g(\mathbf{x})$ of g at $\mathbf{x}$ exists and is given by

$$\mathrm{D}g(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial g_1}{\partial x_1}(\mathbf{x}) & \dfrac{\partial g_1}{\partial x_2}(\mathbf{x}) & \dots & \dfrac{\partial g_1}{\partial x_n}(\mathbf{x}) \\ \dfrac{\partial g_2}{\partial x_1}(\mathbf{x}) & \dfrac{\partial g_2}{\partial x_2}(\mathbf{x}) & \dots & \dfrac{\partial g_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial g_n}{\partial x_1}(\mathbf{x}) & \dfrac{\partial g_n}{\partial x_2}(\mathbf{x}) & \dots & \dfrac{\partial g_n}{\partial x_n}(\mathbf{x}) \end{pmatrix} .$$

If S is convex and

$$\max_{\mathbf{x} \in S} \|\mathrm{D}g(x)\|_\infty < 1 \ , \tag{5.1.3}$$

then $K \equiv \max_{\mathbf{x} \in S} \|\mathrm{D}g(x)\|_\infty$ can be used to satisfy the second hypothesis of the Fixed Point Theorem because we have that $\|g(\mathbf{x}) - g(\mathbf{y})\|_\infty \leq K\|\mathbf{x} - \mathbf{y}\|_\infty$ for all $\mathbf{x}$ and $\mathbf{y}$ in S. This is a consequence of the mean value theorem for real valued functions on $\mathbb{R}^n$. The previous inequality is also true for the norm $\|\cdot\|_2$ but the norm $\|\mathrm{D}g(x)\|_2$ is a lot harder to compute in general.

Thus, to find S and g that satisfy (5.1.3), one needs to find S and g such that

$$\sum_j^n \left| \frac{\partial g_i}{\partial x_j}(\mathbf{x}) \right| \leq K < 1$$

for all i and all $\mathbf{x} \in S$.

♠

Example 5.1.3
Find a solution of

$$\begin{aligned} x_1^2 - 10x_1 + x_2^2 + 8 &= 0 \\ x_1 x_2^2 + x_1 - 10x_2 + 8 &= 0 \end{aligned}$$

with an accuracy of 3×10^{-5} using the norm $\|\cdot\|_\infty$.

We consider the function

$$g(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \dfrac{x_1^2 + x_2^2 + 8}{10} \\ \dfrac{x_1 x_2^2 + x_1 + 8}{10} \end{pmatrix}$$

and

$$S = \left\{ \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} : 0 \le x_i \le \frac{3}{2} \text{ for } i = 1, 2 \right\} .$$

We show that the conditions of the Fixed Point Theorem for Mappings are satisfied. Since

$$0 \le \frac{x_1^2 + x_2^2 + 8}{10} \le \frac{5}{4} < \frac{3}{2} \quad \text{and} \quad 0 \le \frac{x_1 x_2^2 + x_1 + 8}{10} \le \frac{103}{80} < \frac{3}{2}$$

for $\mathbf{x} \in S$, we have that $g(\mathbf{x}) \in S$ if $\mathbf{x} \in S$.

Since g is continuously differentiable on S, we may use the second item of Remark 5.1.2 to find a K satisfying the Fixed Point Theorem for Mappings. We have

$$\begin{gathered} \left| \frac{\partial g_1}{\partial x_1}(\mathbf{x}) \right| = \left| \frac{x_1}{5} \right| \le \frac{3}{10} , \quad \left| \frac{\partial g_1}{\partial x_2}(\mathbf{x}) \right| = \left| \frac{x_2}{5} \right| \le \frac{3}{10} , \\ \left| \frac{\partial g_2}{\partial x_1}(\mathbf{x}) \right| = \left| \frac{x_2^2 + 1}{10} \right| \le \frac{13}{40} \quad \text{and} \quad \left| \frac{\partial g_2}{\partial x_2}(\mathbf{x}) \right| = \left| \frac{x_1 x_2}{5} \right| \le \frac{9}{20} . \end{gathered}$$

Hence $K = \max_{\mathbf{x} \in S} \|\mathrm{D}g(\mathbf{x})\|_\infty \le \max\{3/10 + 3/10, 13/40 + 9/20\} = 31/40 < 1$.

With $\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, we get $\mathbf{x}_1 = \begin{pmatrix} 0.8 \\ 0.8 \end{pmatrix}$, $\mathbf{x}_2 = \begin{pmatrix} 0.928 \\ 0.9312 \end{pmatrix}$, $\ldots$, $\mathbf{x}_{10} = \begin{pmatrix} 0.9999570565 \\ 0.9999570577 \end{pmatrix}$, $\mathbf{x}_{11} = \begin{pmatrix} 0.9999828232 \\ 0.9999828234 \end{pmatrix}$, $\mathbf{x}_{12} = \begin{pmatrix} 0.9999931294 \\ 0.9999931294 \end{pmatrix}$. We get $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty < 3 \times 10^{-5}$ only for $k \ge 12$. So $\mathbf{x}_{12}$ is the desired approximation. All the previous computations were done with as much precision as possible but the written values were rounded to 10 decimals.

With $K = 31/40$, we get

$$\|\mathbf{x}_{11} - \mathbf{p}\|_\infty \le \frac{K^{11}}{1 - K} \|\mathbf{x}_1 - \mathbf{x}_0\|_\infty = \frac{(31/40)^{11}}{1 - 31/40} \left\| \begin{pmatrix} 0.8 \\ 0.8 \end{pmatrix} \right\|_\infty = 0.2154\ldots$$

This is a very large upper bound for the error. This motivates the use of the condition $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty < 3 \times 10^{-5}$ to stop the iteration. ♣

5.2 Newton's Method

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable function; namely, all partial derivatives $\dfrac{\partial f_i}{\partial x_j}(\mathbf{x})$ exist and are continuous. Then the derivative $\mathrm{D}f(\mathbf{x})$ of f at $\mathbf{x}$ is

$$\mathrm{D}f(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \dfrac{\partial f_2}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

The Newton's Method for mappings is as follows:

Algorithm 5.2.1 (Newton's Method for Mappings)

1. Choose $\mathbf{x}_0$ closed to a solution $\mathbf{p}$ of $f(\mathbf{x}) = \mathbf{0}$ if possible.
2. Given $\mathbf{x}_k$, compute
$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathrm{D}f(\mathbf{x}_k))^{-1} f(\mathbf{x}_k) \tag{5.2.1}$$
if $\mathrm{D}f(\mathbf{x}_k)$ is invertible. If $\mathrm{D}f(\mathbf{x}_k)$ is not invertible, start over with a better choice for $\mathbf{x}_0$.
3. Repeat (2) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

Theorem 5.2.2

Suppose that $\mathbf{p}$ is a solution of $f(\mathbf{x}) = \mathbf{0}$. Let $S = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{p}\| \le \eta\}$. Suppose that $\mathrm{D}f(\mathbf{x})$ is invertible for all $\mathbf{x} \in S$ and let $g(\mathbf{x}) = \mathbf{x} - \mathrm{D}f(\mathbf{x})^{-1} f(\mathbf{x})$. If the partial derivatives of order two $\dfrac{\partial^2 g_i}{\partial x_j \partial x_k}$ exist and are continuous on S for $1 \le i, j, k \le n$, then there exists a positive number $\delta \le \eta$ such that the sequence defined by (5.2.1) converges at least quadratically to $\mathbf{p}$ if $\|\mathbf{x}_0 - \mathbf{p}\| < \delta$.

The proof of this theorem is similar to the proof of the order of convergence for the Newton's Method for functions $f : \mathbb{R} \to \mathbb{R}$.

Remark 5.2.3
Any norm on $\mathbb{R}^n$ can be used. However, the norm $\|\cdot\|_\infty$ is often used because it is usually easy to compute. ♠

Example 5.2.4
Use Newton's Method for Mappings to approximate a solution of

$$3x_1^2 - x_2^2 = 0$$

$$3x_1x_2^2 + x_1^3 - 1 = 0$$

near $\begin{pmatrix}1\\1\end{pmatrix}$ with an accuracy of 10^{-6} using $\|\cdot\|_\infty$.

We have

$$f(\mathbf{x}) = \begin{pmatrix}f_1(\mathbf{x})\\f_2(\mathbf{x})\end{pmatrix} = \begin{pmatrix}3x_1^2 - x_2^2\\3x_1x_2^2 - x_1^3 - 1\end{pmatrix}$$

and

$$\mathrm{D}f(\mathbf{x}) = \begin{pmatrix}6x_1 & -2x_2\\3(x_2^2 - x_1^2) & 6x_1x_2\end{pmatrix}.$$

Let $\mathbf{x}_0 = \begin{pmatrix}1\\1\end{pmatrix}$.

1. $f(\mathbf{x}_0) = \begin{pmatrix}2\\1\end{pmatrix}$ and $\mathrm{D}f(\mathbf{x}_0) = \begin{pmatrix}6 & -2\\0 & 6\end{pmatrix}$. The solution of $\mathrm{D}f(\mathbf{x}_0)\mathbf{y} = f(\mathbf{x}_0)$ is $\mathbf{y} = \begin{pmatrix}0.3\overline{8}\\0.1\overline{6}\end{pmatrix}$. Hence $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{y} = \begin{pmatrix}0.6\overline{1}\\0.8\overline{3}\end{pmatrix}$.
2. $f(\mathbf{x}_1) = \begin{pmatrix}0.4\overline{259}\\0.044924554\end{pmatrix}$ and $\mathrm{D}f(\mathbf{x}_1) = \begin{pmatrix}3.\overline{6} & -1.\overline{6}\\0.\overline{962} & 3.0\overline{5}\end{pmatrix}$. The solution of $\mathrm{D}f(\mathbf{x}_1)\mathbf{y} = f(\mathbf{x}_1)$ is $\mathbf{y} = \begin{pmatrix}0.10745204\\-0.019161089\end{pmatrix}$. Hence, $\mathbf{x}_2 = \mathbf{x}_1 - \mathbf{y} = \begin{pmatrix}0.50365909\\0.85249442\end{pmatrix}$.
3. And so on.

We get $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty < 10^{-6}$ for $k \geq 5$. So $\mathbf{x}_5 = \begin{pmatrix}0.50000000\\0.86602540\end{pmatrix}$ is the desired approximation. All the previous computations were done with as much precision as possible but the written values were rounded to 8 decimals. ♣

5.3 Quasi-Newton Methods

We consider the problem of finding a solution of the equation

$$f(\mathbf{x}) = \mathbf{0}\ , \tag{5.3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable.

To use Newton's Method, we need to compute $\mathrm{D}f(\mathbf{x})$. It is not always possible to compute $\mathrm{D}f(\mathbf{x})$ at each step or it may be costly to compute it at each step. It would be nice to have a method like the secant method to solve systems of non-linear equations.

The method proposed by Broyden produces a sequence $\{\mathbf{x}\}_{k=0}^\infty$ from an iterative formula of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - A_k^{-1} f(\mathbf{x}_k) \quad , \quad k = 0, 1, 2, \ldots \tag{5.3.2}$$

where $\mathbf{x}_0$ is the given initial value and the matrix A_k is an approximation of $\mathrm{D}f(\mathbf{x}_k)$.

The method developed by Broyden gives an approximation A_k of $\mathrm{D}f(\mathbf{x}_k)$ such that the approximation A_{k+1} of $\mathrm{D}f(\mathbf{x}_{k+1})$ can be easily obtained from A_k. Only $\mathrm{D}f(\mathbf{x}_0)$ is needed to start the iteration. The method reduces the number of functions evaluation at each steps. However, it also produces an iterative method with a rate of convergence inferior to the quadratic rate of convergence of the Newton's Method. The iterative method proposed by Broyden has a superlinear rate of convergence; namely,

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{p}\|}{\|\mathbf{x}_k - \mathbf{p}\|} = 0 \ ,$$

where $\mathbf{p}$ is the limit of the sequence $\{\mathbf{x}\}_{k=0}^{\infty}$ and thus a solution of $f(\mathbf{x}) = \mathbf{0}$. Unlike Newton's Method, the iterative method developed by Broyden is not "self correcting" Newton's Method will correct round-off errors as one keeps iterating. This is not so for the method presented in this section.

The sequence $\{\mathbf{x}\}_{k=0}^{\infty}$ produced by the iterative formula (5.3.2) will generally converges to a solution $\mathbf{p}$ of $f(\mathbf{x}) = \mathbf{0}$ if $\mathbf{x}_0$ is closed enough to $\mathbf{p}$.

The approximation A_k of $\mathrm{D}f(\mathbf{x}_k)$ is given recursively as follows.

1. $A_0 = J_f(\mathbf{x}_0)$. This is the only time that $J_f(\mathbf{x})$ needs to be computed.

2. Given A_k, $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, the approximation A_{k+1} is a matrix which satisfies

$$A_{k+1}\left(\mathbf{x}_{k+1} - \mathbf{x}_k\right) = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)$$

and

$$\mathcal{N}(A_{k+1} - A_k) \supset E \equiv \{\lambda(\mathbf{x}_{k+1} - \mathbf{x}_k) : \lambda \in \mathbb{R}\}^{\perp} \ ,$$

where $\mathcal{N}(A_{k+1} - A_k)$ denotes the kernel of the linear mapping associated to the matrix $A_{k+1} - A_k$.

The second condition in item (2) can be expressed as follows. $A_{k+1}\mathbf{x} = A_k\mathbf{x}$ for all $\mathbf{x}$ such that $\langle \mathbf{x}, (\mathbf{x}_{k+1} - \mathbf{x}_k)\rangle = 0$. In other words, $A_{k+1} = A_k$ on the orthogonal complement of E. It is easy to check that the matrix A_{k+1} satisfying the item (2) above is

$$A_{k+1} = A_k + \frac{1}{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2}\Big(f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) - A_k(\mathbf{x}_{k+1} - \mathbf{x}_k)\Big)(\mathbf{x}_{k+1} - \mathbf{x}_k)^{\top} \ . \tag{5.3.3}$$

There is an additional benefit in using the iterative method above. There is no need to solve a linear system of the form $A_k\mathbf{y} = \mathbf{x}_k$ at each iterative step. It is easy to compute recursively A_k^{-1}. To explain how to do this, we need the following proposition.

Proposition 5.3.1 (Sherman and Morrison)

If A is an $n\times n$ nonsingular matrix and $\mathbf{x}$, $\mathbf{y}$ are two vectors such that $\mathbf{y}^\top A^{-1}\mathbf{x}+1\neq 0$, then $A+\mathbf{x}\mathbf{y}^\top$ is nonsingular and

$$\left(A+\mathbf{x}\mathbf{y}^\top\right)^{-1} = A^{-1} - \frac{1}{1+\mathbf{y}^\top A^{-1}\mathbf{x}} A^{-1}\mathbf{x}\mathbf{y}^\top A^{-1} \ . \tag{5.3.4}$$

Proof.
Since

$$A+\mathbf{x}\mathbf{y}^\top = A\left(\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top\right)$$

and A is nonsingular, it is enough to prove that $\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top$ is nonsingular to prove that $A+\mathbf{x}\mathbf{y}^\top$ is nonsingular.

If $\mathbf{z}\in\mathcal{N}\left(\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top\right)$, we get that

$$\underbrace{\left(1+\mathbf{y}^\top A^{-1}\mathbf{x}\right)}_{\neq 0}\mathbf{y}^\top\mathbf{z} = \mathbf{y}^\top\left(\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top\right)\mathbf{z} = \mathbf{y}^\top\mathbf{0} = 0 \ .$$

Thus $\mathbf{y}^\top\mathbf{z}=0$ and

$$\mathbf{0} = \left(\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top\right)\mathbf{z} = \mathbf{z} + A^{-1}\mathbf{x}\left(\mathbf{y}^\top\mathbf{z}\right) = \mathbf{z} \ .$$

We conclude that $\mathcal{N}\left(\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top\right)=\{\mathbf{0}\}$ and $\mathrm{Id}+A^{-1}\mathbf{x}\mathbf{y}^\top$ is nonsingular.

To prove that the right hand side of (5.3.4) is the inverse of $A+\mathbf{x}\mathbf{y}^\top$, it suffices to multiply the right hand side of (5.3.4) by $A+\mathbf{x}\mathbf{y}^\top$. This is a simple computation left to the reader. ∎

Let

$$\mathbf{u}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$$

and

$$\mathbf{v}_{k+1} = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \quad , \quad k = 0,1,2,\ldots$$

If we substitute

$$\mathbf{y} = \mathbf{u}_{k+1} \ , \quad A = A_k \quad \text{and} \quad \mathbf{x} = \frac{1}{\|\mathbf{u}_{k+1}\|^2}\left(\mathbf{v}_{k+1} - A_k\mathbf{u}_{k+1}\right)$$

in (5.3.4), we get from (5.3.3) that

$$\begin{aligned}A_{k+1}^{-1} &= \left(A_k + \frac{1}{\|\mathbf{u}_{k+1}\|^2}\left(\mathbf{v}_{k+1} - A_k\mathbf{u}_{k+1}\right)\mathbf{u}_{k+1}^\top\right)^{-1}\\ &= A_k^{-1} - \left(1+\mathbf{u}_{k+1}^\top A_k^{-1}\left(\frac{1}{\|\mathbf{u}_{k+1}\|^2}\left(\mathbf{v}_{k+1} - A_k\mathbf{u}_{k+1}\right)\right)\right)^{-1}\\ &\qquad \left(A_k^{-1}\left(\frac{1}{\|\mathbf{u}_{k+1}\|^2}\left(\mathbf{v}_{k+1} - A_k\mathbf{u}_{k+1}\right)\right)\mathbf{u}_{k+1}^\top A_k^{-1}\right)\end{aligned}$$

$$= A_k^{-1} + \left(\mathbf{u}_{k+1}^\top A_k^{-1}\mathbf{v}_{k+1}\right)^{-1}\left(\mathbf{u}_{k+1} - A_k^{-1}\mathbf{v}_{k+1}\right)\mathbf{u}_{k+1}^\top A_k^{-1} \quad , \quad k = 0, 1, 2, \ldots$$

Once A_0^{-1} has been computed, it becomes "relatively easy" to compute the A_k^{-1} with the iterative formula above.

5.4 Steepest Descent for Nonlinear Systems

The problem of finding a solution of the equation

$$f(\mathbf{x}) = 0 \ , \tag{5.4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, can be solved using a method similar to the steepest descent method that has been introduced earlier to solve linear systems of the form $A\mathbf{x} = \mathbf{b}$.

The **steepest descent** algorithm that we present below is based on the following observations. At a point $\mathbf{x}_k$, the direction in which the function $f(\mathbf{x})$ decreases the fastest is the direction of the vector $-\nabla f(\mathbf{x}_k)$. The descent method is based on minimizing $q(t) = f\left(\mathbf{x}_k - t\nabla f(\mathbf{x}_k)\right)$ for t near the origin. If t_k is the value of t nearest 0 where a minimum of q is reached, the next approximation of the solution of $f(\mathbf{x}) = \mathbf{0}$ is given by $\mathbf{x}_{k+1} = \mathbf{x}_k - t_k\nabla f(\mathbf{x}_k)$. Repeating this procedure, we hope to get a sequence of vectors $\{\mathbf{x}_k\}_{k=0}^{\infty}$ converging toward the solution of $f(\mathbf{x})$.

Algorithm 5.4.1 (Steepest descent)

1. Choose $\mathbf{x}_0$ closed to a solution $\mathbf{p}$ of $f(\mathbf{x}) = \mathbf{0}$ if possible.
2. Given $\mathbf{x}_k$, compute $\nabla f(\mathbf{x}_k)$.
3. Find the value of t_k nearest 0 for which $q(t) = f\left(\mathbf{x}_k - t\nabla f(\mathbf{x}_k)\right)$ reaches a minimum.
4. Let $\mathbf{x}_{k+1} = \mathbf{x}_k - t_k\nabla f(\mathbf{x}_k)$.
5. Repeat (2) to (4) until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$, where ϵ is given.

To minimize q in (3), one may look for the roots of q'; namely, the critical points of q. We will not elaborate on the techniques to minimize q. This is part of the important subject of optimization that we unfortunately do not cover in this book.

5.5 Exercises

Question 5.1
Consider the function

$$g(\mathbf{x}) = \begin{pmatrix} \cos^2(x_1 + x_2)/6 \\ \sin(x_1)\cos(x_2)/5 \end{pmatrix} .$$

a) Show that g satisfies the hypothesis of the Fixed Point Theorem for mapping on $S = \{\mathbf{x} \in \mathbb{R}^2 : -1 \le x_1, x_2 \le 1\}$.
b) Use the fixed point method to approximate the fixed point of g in S with an accuracy of 10^{-5}.
c) Let $\mathbf{p}$ be the fixed point of g in S, find a small value of n for which $\|\mathbf{x}_n - \mathbf{p}\|_\infty < 10^{-5}$, where $\{\mathbf{x}_n\}_{n=1}^{\infty}$ is the sequence generated by $\mathbf{x}_{n+1} = g(\mathbf{x}_n)$ with $\mathbf{x}_0 = \mathbf{0}$.

Question 5.2
a) Show that a solution of

$$f(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} x_1^3 + 12x_1 - x_2 - 3 \\ 2x_1 + x_2^3 - 12x_2 + 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

is a fixed point of

$$g(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} (x_2 - x_1^3 + 3)/12 \\ (2x_1 + x_2^3 + 2)/12 \end{pmatrix}$$

and vice-versa.
b) Use a sketch of the two level curves defined by $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 0$ to show that there is at least one solution to $f(\mathbf{x}) = \mathbf{0}$.
c) Check that the function g satisfies all the hypotheses of the Fixed Point Theorem for mappings on $S = \{\mathbf{x} : 0 \le x_1, x_2 \le 1\}$.
d) Use the fixed point method to approximate a solution of $f(\mathbf{x}) = \mathbf{0}$ with an accuracy of 10^{-5}. Start with $\mathbf{x}_0 = \mathbf{0}$.
e) Determine a small value of n for which $\|\mathbf{x}_n - \mathbf{p}\|_\infty < 10^{-5}$, where $\mathbf{p}$ is the unique fixed point of g in S and the vectors $\mathbf{x}_n$ are generated by the fixed point method from $\mathbf{x}_0 = \mathbf{0}$.

Question 5.3
Use the fixed point method to approximate a solution of $f(\mathbf{x}) = \mathbf{0}$ to within 10^{-5}, where

$$f(\mathbf{x}) = \begin{pmatrix} 4x_1 - x_2 - 5 \\ 1 + \sqrt{x_1} - (x_2 + 1)^3 \end{pmatrix} .$$

Don't forget to verify the hypothesis of the Fixed-Point Theorem first.

Question 5.4
a) Show that a root of

$$f(\mathbf{x}) = \begin{pmatrix} 2(x_1 - 1)^2 - 2x_2 - 1 \\ x_1^2 + 4x_2^2 - 4 \end{pmatrix} \tag{5.5.1}$$

is a fixed point of

$$g(\mathbf{x}) = \begin{pmatrix} (2x_1^2 - 2x_2 + 1)/4 \\ (-x_1^2 - 4x_2^2 + 8x_2 + 4)/8 \end{pmatrix}$$

and vice-versa.
b) Use a sketch of the two level curves defined by $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 0$ to show that there is at least one solution to $f(\mathbf{x}) = \mathbf{0}$.
c) Verify that the function g satisfies all the hypotheses of the Fixed Point Theorem for mappings on $S = \{\mathbf{x} : -1/4 \le x_1 \le 1/4 \ , \ 3/4 \le x_2 \le 1\}$.

d) Use the fixed point method to approximate a solution of (5.5.1) with an accuracy of 10^{-5}. Start with $\mathbf{x}_0 = \begin{pmatrix} 0 & 1 \end{pmatrix}^\top$.
e) Find a small value of n for which $\|\mathbf{x}_n - \mathbf{p}\|_\infty < 10^{-5}$, where $\mathbf{p}$ is the unique fixed point of g in S and the vectors $\mathbf{x}_n$ are generated by the fixed point method from $\mathbf{x}_0$ given in (d).

Question 5.5
a) Show that a root of

$$f(\mathbf{x}) = \begin{pmatrix} 3 - 15x_1 + x_2^2 + 4x_3 \\ 5 + x_1^2 - 10x_2 + x_3 \\ 22 + x_2^3 - 25x_3 \end{pmatrix} \tag{5.5.2}$$

is a fixed point of

$$g(\mathbf{x}) = \begin{pmatrix} (x_2^2 + 4x_3 + 3)/15 \\ (5 + x_1^2 + x_3)/10 \\ (x_2^3 + 22)/25 \end{pmatrix}$$

and vice-versa.
b) Verify that the function g satisfies all the hypotheses of the Fixed Point Theorem for mappings on $S = \{\mathbf{x} : 0 \le x_i \le 3/2\}$.
c) Use the fixed point method to approximate a solution of (5.5.2) with an accuracy of 10^{-5}. Start with $\mathbf{x}_0 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^\top$.
d) Find a small value of n for which $\|\mathbf{x}_n - \mathbf{p}\|_\infty < 10^{-5}$, where $\mathbf{p}$ is the unique fixed point of g in S and the vectors $\mathbf{x}_n$ are generated by the fixed point method from $\mathbf{x}_0$ given in (c).

Question 5.6
Use Newton's Method to approximate a solution of $f(\mathbf{x}) = \mathbf{0}$ with an accuracy 10^{-6}, where

$$f(\mathbf{x}) = \begin{pmatrix} x_1^3 + x_1^2 x_2 - x_1 x_3 + 6 \\ e^{x_1} + e^{x_2} - x_3 \\ x_2^2 - 2x_1 x_3 - 4 \end{pmatrix}$$

for $-2 \le x_1, x_2 \le -1$ and $0 \le x_3 \le 1$.

Chapter 6

Polynomial Interpolation

Suppose that an unknown function f governs some physical phenomenon and that the results of an experiment gives the data $(x_i, f(x_i))$ for $i = 0, 1, 2, \ldots, n$. Could we use these data to approximate $f(x)$ at $x \neq x_i$ for $i = 0, 1, \ldots, n$? In this chapter, we present some methods that answer this question using (piecewise) polynomial approximations of f.

Definition 6.0.1

Suppose that p is a (piecewise) polynomial approximations of f, if x is inside the smallest interval containing the x_i's, we say that $p(x)$ is a **polynomial interpolation** of $f(x)$. Otherwise, we say that $p(x)$ is a **polynomial extrapolation** of $f(x)$.

6.1 Lagrange Interpolation

Definition 6.1.1

If $f : [a, b] \to \mathbb{R}$ is a function and $a \leq x_0 < x_1 < x_2 < \ldots < x_n \leq b$, then the polynomial p of degree n defined by

$$p(x) = \sum_{i=0}^{n} f(x_i) \prod_{\substack{j=0 \\ j\neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \tag{6.1.1}$$

is such that $p(x_i) = f(x_i)$ for $0 \leq i \leq n$. The polynomial p is called the **Lagrange Interpolating Polynomial** of f at $x_0, x_1, \ldots, x_n$.

The polynomial p is often used as an approximation of f on the interval $[x_0, x_n]$.

Since polynomials of degree n have exactly n complex roots counted with multiplicity, the Lagrange Interpolating Polynomial in (6.1.1) is the unique polynomial of degree at most n satisfying $p(x_i) = f(x_i)$ for $0 \leq i \leq n$. To prove this statement, suppose that q is another polynomial of degree less than or equal to n such that $q(x_i) = f(x_i)$ for $0 \leq i \leq n$, then $p - q$

is a polynomial of degree at most n such that $(p-q)(x_i) = 0$ at $n+1$ distinct values. Namely, $p-q$ is a polynomial of degree at most n with $n+1$ roots. The only possibility is $p-q=0$.

(6.1.1) is not the best form of the interpolating polynomial of a function but it is an important tool to develop formulas for derivation and integration later on. We will present another form of the interpolating polynomial below.

6.2 Newton Interpolation

We now extend the definition of interpolating polynomial of a function at $(n+1)$ distinct points $x_0, x_1, \ldots, x_n$ to the case where the x_i's are not all distinct.

Definition 6.2.1

Let $f :]a,b[\to \mathbb{R}$ and $g :]a,b[\to \mathbb{R}$ be two functions sufficiently differentiable. Suppose that $x_0, x_1, \ldots, x_n$ are $(n+1)$ points in $]a,b[$ (not necessarily distinct). We say that **f and g agree at the points** $x_0, x_1, \ldots, x_n$ if

$$\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(z) = \frac{\mathrm{d}^j g}{\mathrm{d}x^j}(z)$$

for $j = 0, 1, \ldots, m-1$ whenever z appears m times in the list $x_0, x_1, \ldots, x_n$. Obviously, we set

$$\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(z) = f(z)$$

for $j = 0$.

Theorem 6.2.2

Let $f :]a,b[\to \mathbb{R}$ be a function sufficiently differentiable. Suppose that $x_0, x_1, \ldots, x_n$ are $(n+1)$ points in $]a,b[$ not necessarily distinct. Then there is a unique polynomial p of degree at most n such that f and p agree at $x_0, x_1, \ldots, x_n$.

Proof.
See Section 6.3 below. ∎

Definition 6.2.3

The polynomial p in Theorem 6.2.2 is called the **interpolating polynomial** of f at the **interpolatory points** $x_0, x_1, \ldots, x_n$.

Definition 6.2.4

Let $f :]a,b[\to \mathbb{R}$ be a function sufficiently differentiable. The k^{th} **divided difference** of f at $k+1$ not necessarily distinct points $x_0, x_1, \ldots, x_k$ in $]a,b[$, denoted $f[x_0, x_1, \ldots, x_k]$, is the coefficient of x^k in the unique polynomial of degree at most k

that agrees with f at x_0, x_1, …, x_k.

Theorem 6.2.5

Let $f :]a,b[\to \mathbb{R}$ be a function sufficiently differentiable. Suppose that x_0, x_1, …, x_n are $n+1$ not necessarily distinct points in $]a,b[$. Then the unique polynomial p of degree at most n that agrees with f at x_0, x_1, …, x_n is given by

$$\begin{aligned} p(x) = f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) + \ldots \\ + f[x_0,x_1,\ldots,x_n]\,(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \ . \end{aligned} \tag{6.2.1}$$

Moreover, for $x_j \neq x_{j+k}$,

$$f[x_j,x_{j+1},\ldots,x_{j+k}] = \frac{f[x_{j+1},x_{j+2},\ldots,x_{j+k}] - f[x_j,x_{j+1},\ldots,x_{j+k-1}]}{x_{j+k}-x_j} \tag{6.2.2}$$

and, for $x_j = x_{j+1} = x_{j+2} = \ldots = x_{j+k}$,

$$f[x_j,x_{j+1},\ldots,x_{j+k}] = \frac{1}{k!}\,\frac{\mathrm{d}^k f}{\mathrm{d}x^k}(x_j) \ . \tag{6.2.3}$$

Finally,

$$f(x) = p(x) + f[x_0,x_1,\ldots,x_n,x]\,(x-x_0)(x-x_1)\ldots(x-x_{n-1})(x-x_n) \ . \tag{6.2.4}$$

Proof.
See Section 6.3 below. ∎

It is easy to deduce the first divided difference of f

a) The interpolating polynomial p of f of degree 0 at x_0 is given by the constant function $p(x) = f(x_0)$ for all x. Hence, the coefficient $f[x_0]$ of x^0 is

$$f[x_0] = f(x_0) \ .$$

b) The interpolating polynomial p of f of degree at most 1 at x_0, x_1 is given by

$$p(x) = \begin{cases} f(x_0) + \dfrac{f(x_1)-f(x_0)}{x_1-x_0}\,(x-x_0) & \text{if} \quad x_0 \neq x_1 \\ f(x_0) + f'(x_0)\,(x-x_0) & \text{if} \quad x_0 = x_1 \end{cases}$$

In the first case, it is the equation of the secant line through $(x_0, f(x_0))$ and $(x_1, f(x_1))$. In the second case, it is the equation of the tangent line at x_0 because we must have $p(x_0) = f(x_0)$ and $p'(x_0) = f'(x_0)$. Hence, the coefficient $f[x_0,x_1]$ of x^1 is

$$f[x_0,x_1] = \begin{cases} \dfrac{f(x_1)-f(x_0)}{x_1-x_0} & \text{if} \quad x_0 \neq x_1 \\ f'(x_0) & \text{if} \quad x_0 = x_1 \end{cases}$$

c) The interpolating polynomial p of f of degree at most 2 at x_0, x_1, x_2 is of the form

$$p(x) = A + B\,(x - x_0) + C\,(x - x_0)(x - x_1)\ .$$

If the x_i's are distinct, the polynomial p must satisfy

$$\begin{aligned} f(x_0) &= p(x_0) = A\ , \\ f(x_1) &= p(x_1) = A + B\,(x_1 - x_0) \end{aligned}$$

and

$$f(x_2) = p(x_2) = A + B\,(x_2 - x_0) + C\,(x_2 - x_0)(x_2 - x_1)\ .$$

Hence,

$$A = f(x_0) = f[x_0]\ , \quad B = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f[x_0, x_1]$$

and

$$\begin{aligned} C &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left(f(x_2) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\,(x_2 - x_0) \right) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left(f(x_2) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\,((x_2 - x_1) + (x_1 - x_0)) \right) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left(f(x_2) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\,(x_2 - x_1) - f(x_1) + f(x_0) \right) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left(f(x_2) - f(x_1) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\,(x_2 - x_1) \right) \\ &= \frac{1}{(x_2 - x_0)} \left(\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) \\ &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}\ . \end{aligned}$$

Hence, the coefficient $f[x_0, x_1, x_2]$ of x^2 is

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

if the x_i's are distinct. We get a similar formula if $x_0 = x_1 \neq x_2$ or $x_1 \neq x_2 = x_3$. Recall that we assume that if a value z appears more than once in $\{x_0, x_1, x_3\}$, then all occurrences of z are contiguous.

If $x_0 = x_1 = x_2$, the interpolating polynomial p of f of degree at most 2 is given by the Taylor polynomial of degree 2 at x_0; namely,

$$p(x) = f(x_0) + f'(x_0)\,(x - x_0) + \frac{1}{2!}\,f''(x_0)\,(x - x_0)^2\ .$$

It is easy to check that $p(x_0) = f(x_0)$, $f'(x_0) = p'(x_0)$ and $f''(x_0) = p''(x_0)$. Hence, the coefficient $f[x_0, x_1, x_2]$ of x^2 is

$$f[x_0, x_1, x_2] = \frac{1}{2!}\,f''(x_0)$$

if $x_0 = x_1 = x_2$.

Remark 6.2.6

1. Because of Theorem 6.2.2, (6.1.1) and (6.2.1) are two ways to represent the polynomial of degree at most n that agrees with f at the $n+1$ distinct points x_0, x_1, $\ldots$, x_n,

2. Because the interpolating polynomial of degree at most n of f at x_0, x_1, x_2, $\ldots$, x_n is independent of the order in which the x_i's are listed, in particular the coefficient of x^n is not going to change if the order of the x_i's is changed, we have that
$$f[x_0, x_1, \ldots, x_k] = f[x_{\sigma(0)}, x_{\sigma(1)}, \ldots, x_{\sigma(k)}]$$
for any permutation σ of $\{0, 1, 2, 3, \ldots, k\}$.

3. To be able to use (6.2.3) and thus get simple divided difference formulae, we assume that if a value z appears more than once in $\{x_0, x_1, x_2, \ldots, x_n\}$, then all occurrences of z are contiguous.

4. From a computational point of view, (6.2.1) is better than (6.1.1) because there are lest operations needed to evaluate $p(x)$ if we use the nested form to evaluate polynomials.

5. The form (6.2.1) of the interpolating polynomial of f at x_0, x_1, $\ldots$, x_n can be easily extended to the form (6.2.1) of the interpolating polynomial of f at the $n+1$ points x_0, x_1, $\ldots$, x_n and x_{n+1}, where x_{n+1} is a new point. Only the divided difference $f[x_0, x_1, \ldots, x_n, x_{n+1}]$ needs to be computed because we already have the divided differences $f[x_0]$, $f[x_0, x_1]$, $\ldots$, $f[x_0, x_1, \ldots, x_n]$ from the interpolating polynomial of f at x_0, x_1, $\ldots$, x_n.

♠

The divided differences have the following properties.

Theorem 6.2.7

Let x_0, x_1, $\ldots$, x_k be $k+1$ points in $]a, b[$ and $f :]a, b[\to \mathbb{R}$ be a sufficiently continuously differentiable function. Then
$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!} \frac{\mathrm{d}^k f}{\mathrm{d}x^k}(\xi)$$
for some ξ in the smallest interval containing x_0, x_1, $\ldots$, x_k.
Moreover,
$$\frac{\mathrm{d}^j}{\mathrm{d}x^j} f[x_0, x_1, \ldots, x_k, x] = j!\, f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{j+1 \text{ times}}] \tag{6.2.5}$$
for $j \geq 0$.

Proof.
See Section 6.3 below. ∎

To motivate (6.2.5), we prove that

$$\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x] = f[x_0, x, x] \quad \text{and} \quad \frac{\mathrm{d}^2}{\mathrm{d}x^2} f[x_0, x] = 2 f[x_0, x, x, x] \ .$$

We have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x] &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{f(x) - f(x_0)}{x - x_0}\right) = \frac{f'(x)(x - x_0) - (f(x) - f(x_0))}{(x - x_0)^2} \\
&= \frac{f'(x) - \dfrac{f(x) - f(x_0)}{x - x_0}}{x - x_0} = \frac{f[x, x] - f[x_0, x]}{x - x_0} = f[x_0, x, x]
\end{aligned}$$

at $x \neq x_0$ and

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}x^2} f[x_0, x] &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{f'(x)(x - x_0) - (f(x) - f(x_0))}{(x - x_0)^2}\right) \\
&= \frac{f''(x)(x - x_0)^3 - 2f'(x)(x - x_0)^2 + 2(f(x) - f(x_0))(x - x_0)}{(x - x_0)^4} \\
&= 2\,\frac{\dfrac{f''(x)}{2} - \dfrac{1}{x - x_0}\left(f'(x) - \dfrac{f(x) - f(x_0)}{x - x_0}\right)}{x - x_0} \\
&= 2\,\frac{f[x, x, x] - \dfrac{1}{x - x_0}\left(f[x, x] - f[x, x_0]\right)}{x - x_0} \\
&= 2\,\frac{f[x, x, x] - f[x_0, x, x]}{x - x_0} = 2 f[x_0, x, x, x]
\end{aligned}$$

at $x \neq x_0$.

Before proving Theorems 6.2.2, 6.2.5 and 6.2.7, we illustrate how these theorems are used.

6.2.1 Linear Interpolation

Suppose that we only have two points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ with $x_0 \neq x_1$. Then

$$p(x) = f[x_0] + f[x_0, x_1]\,(x - x_0) \ ,$$

where

$$f[x_0] = f(x_0) \quad \text{and} \quad f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \ .$$

Example 6.2.8
If $(x_0, f(x_0)) = (2.2, 6.2)$ and $(x_1, f(x_1)) = (2.5, 6.7)$, find an approximation of $f(x)$ at $x = 2.35$.

We have $f[2.2] = f(2.2) = 6.2$ and

$$f[2.2, 2.5] = \frac{f(2.5) - f(2.2)}{2.5 - 2.2} = \frac{6.7 - 6.2}{2.5 - 2.2} = 1.\overline{6} \ .$$

Thus $p(x) = 6.2 + 1.\overline{6}(x - 2.2)$ and $f(2.35) \approx p(2.35) = 6.45$. ♣

Example 6.2.9
Suppose that only the values of $f(x) = \cos(x)$ at $x = 0$ and $x = \pi/6$ are known, find an approximation of $\cos(0.2)$.

We compute the interpolating polynomial at the points

$$(0, \cos(0)) = (0,1) \quad \text{and} \quad (\pi/6, \cos(\pi/6)) = \left(\pi/6, \sqrt{3}/2\right) .$$

We have $f[0] = f(0) = 1$ and

$$f[0,\pi/6] = \frac{f(pi/6) - f(0)}{\pi/6 - 0} = \frac{\sqrt{3}/2 - 1}{\pi/6 - 0} = \frac{3\sqrt{3} - 6}{\pi} .$$

Thus

$$p(x) = 1 + \left(\frac{3\sqrt{3} - 6}{\pi}\right) x$$

and $\cos(0.2) \approx p(0.2) \approx 0.948825$. Rounded after six digits, $\cos(0.2) = 0.980067$. Thus, the absolute error is about 0.031242. ♣

Remark 6.2.10
MATLAB can be used to plot the graph of a function f associated to the data set

$$\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\} ,$$

where $x_0 < x_1 < \ldots < x_n$. MATLAB plots the graph of the **piecewise linear function** p that passes through each point of the data set. Namely, f is approximated by the piecewise polynomial p (each piece is a polynomial of degree one) defined by

$$p(x) = f[x_i] + f[x_i, x_{i+1}]\,(x - x_i) \quad , \quad x_i \leq x \leq x_{i+1} ,$$

for $i = 0, 1, 2, \ldots, n-1$. ♠

6.2.2 Quadratic Interpolation

Suppose that we have the points $(x_0, f(x_0))$, $(x_1, f(x_1))$ and $(x_2, f(x_2))$ with $x_i \neq x_j$ for $i \neq j$. Then

$$p(x) = f[x_0] + f[x_0, x_1]\,(x - x_0) + f[x_0, x_1, x_2]\,(x - x_0)(x - x_1) ,$$

where

$$f[x_0] = f(x_0) , \quad f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad \text{and} \quad f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} .$$

Moreover, we need to compute $f[x_1, x_2] = \dfrac{f(x_2) - f(x_1)}{x_2 - x_1}$ in the formula of $f[x_0, x_1, x_2]$.

Example 6.2.11
Given the data $(x_0, f(x_0)) = (2.2, 6.2)$, $(x_1, f(x_1)) = (2.5, 6.7)$ and $(x_2, f(x_2)) = (2.7, 6.5)$. Find the approximation of $f(x)$ at $x = 2.35$.

We have $f[2.2] = f(2.2) = 6.2$,

$$\begin{aligned} f[2.2, 2.5] &= \frac{f(2.5) - f(2.2)}{2.5 - 2.2} = \frac{6.7 - 6.2}{2.5 - 2.2} = 1.\overline{6} \ , \\ f[2.5, 2.7] &= \frac{f(2.7) - f(2.5)}{2.7 - 2.5} = \frac{6.5 - 6.7}{2.7 - 2.5} = -1 \end{aligned}$$

and

$$f[2.2, 2.5, 2.7] = \frac{f[2.5, 2.7] - f[2.2, 2.5]}{2.7 - 2.2} = \frac{-1 - 1.\overline{6}}{2.7 - 2.2} = -5.\overline{3} \ .$$

Thus

$$\begin{aligned} p(x) &= 6.2 + 1.\overline{6}(x - 2.2) - 5.\overline{3}(x - 2.2)(x - 2.5) \\ &= 6.2 + (x - 2.2)(1.\overline{6} - 5.\overline{3}(x - 2.5)) \end{aligned}$$

and $f(2.35) \approx p(2.35) = 6.57$. ♣

6.2.3 General Interpolation

We now consider the general interpolating polynomial at the points $x_0, x_1, \ldots, x_n$.

If the x_i's are distinct (i.e. $x_i \neq x_j$ for all i and j), then Table 6.1(a) gives the formulas to compute the first Newton divided differences of f and thus the first coefficients of the interpolating polynomial (6.2.1) of f of degree at most n at $x_0, x_1, \ldots, x_n$. The coefficients are on the top line of the table.

When $x_0, x_1, \ldots, x_n$ are not all distinct, we use (6.2.3) to evaluate the entries in the table of Newton divided differences as show in Table 6.1(b). We assume that if a value z appears more than once in $\{x_0, x_1, x_2, \ldots, x_n\}$, then all occurrences of z are contiguous. For instance,

$$x_1 = x_0 \Rightarrow f[x_1, x_0] = f'(x_0) \quad , \quad x_0 = x_1 = x_2 \Rightarrow f[x_2, x_1, x_0] = \frac{1}{2} f''(x_0)$$

and

$$x_3 = x_4 = x_5 = x_6 \Rightarrow f[x_6, x_5, x_4, x_3] = \frac{1}{3!} f'''(x_3) \ .$$

Definition 6.2.12

1. (6.2.1) is the **Newton form** or **Newton-Cotes form** of the interpolating polynomial of f at x_0, x_1 , ..., x_n.

a) $x_0 < x_1 < x_2 < x_3$

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$
x_0	$f(x_0)$	$f[x_1,x_0] = \dfrac{f(x_1)-f(x_0)}{x_1-x_0}$	$f[x_2,x_1,x_0] = \dfrac{f[x_2,x_1]-f[x_1,x_0]}{x_2-x_0}$	$f[x_3,x_2,x_1,x_0] = \dfrac{f[x_3,x_2,x_1]-f[x_2,x_1,x_0]}{x_3-x_0}$
x_1	$f(x_1)$	$f[x_2,x_1] = \dfrac{f(x_2)-f(x_1)}{x_2-x_1}$	$f[x_3,x_2,x_1] = \dfrac{f[x_3,x_2]-f[x_2,x_1]}{x_3-x_1}$	
x_2	$f(x_2)$	$f[x_3,x_2] = \dfrac{f(x_3)-f(x_2)}{x_3-x_2}$		
x_3	$f(x_3)$			

b) $x_0 < x_1 = x_2 = x_3$

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$
x_0	$f(x_0)$	$f[x_1,x_0] = \dfrac{f(x_1)-f(x_0)}{x_1-x_0}$	$f[x_2,x_1,x_0] = \dfrac{f[x_2,x_1]-f[x_1,x_0]}{x_2-x_0}$	$f[x_3,x_2,x_1,x_0] = \dfrac{f[x_3,x_2,x_1]-f[x_2,x_1,x_0]}{x_3-x_0}$
x_1	$f(x_1)$	$f[x_2,x_1] = f'(x_1)$	$f[x_3,x_2,x_1] = \dfrac{1}{2}f''(x_1)$	
x_2	$f(x_2)$	$f[x_3,x_2] = f'(x_2)$		
x_3	$f(x_3)$			

Table 6.1: Some tables of Newton divided differences

2. If n is odd and $x_0 = x_1 < x_2 = x_3 < \ldots < x_{2k} = x_{2k+1} < \ldots < x_{n-1} = x_n$, then (6.2.1) is also called the **Hermite's interpolating polynomial** for f at x_0, x_1, $\ldots$, x_n.

Example 6.2.13
If $(x_0, f(x_0)) = (1.0, 2.4)$, $(x_1, f(x_1)) = (1.3, 2.2)$, $(x_2, f(x_2)) = (1.5, 2.3)$ and $(x_3, f(x_3)) = (1.7, 2.4)$, find an approximation for $f(1.4)$.

The following table gives the divided differences needed for the Newton form of the interpolating polynomial of f at x_0, x_1, x_2 and x_3.

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$
1.0	2.4	$-0.\overline{6}$	$2.\overline{3}$	$-3.\overline{3}$
1.3	2.2	0.5	0.0	
1.5	2.3	0.5		
1.7	2.4			

Hence,

$$\begin{aligned} p(x) &= 2.4 - 0.\overline{6}\,(x-1.0) + 2.\overline{3}\,(x-1.0)(x-1.3) - 3.\overline{3}(x-1.0)(x-1.3)(x-1.5) \\ &= 2.4 + (x-1.0)(-0.\overline{6} + (x-1.3)(2.\overline{3} - 3.\overline{3}\,(x-1.5))) \end{aligned}$$

and $f(1.4) \approx p(1.4) = 2.24$. ♣

Code 6.2.14 (**Newton Divided Differences**)

To produce the table of divided differences to generate the coefficients a_i and the interpolating polynomial

$$p(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \ldots + a_{n-1}(x-x_0)(x-x_1)\ldots(x-x_{n-2})$$

at the points x_i for $0 \le i < n$.
Input: The matrix

$$\begin{pmatrix} x_0 & f_0 \\ x_1 & f_1 \\ \vdots & \vdots \\ x_{n-1} & f_{n-1} \end{pmatrix}$$

(d in the code below), where $x_0 \le x_1 \le \ldots \le x_{n-1}$ an $f_k = f^{(j)}(x_k)$ if $x_{k-j-1} < x_{k-j} = x_{x-j+1} = \ldots = x_k$; namely, the point x_k appears j times before.
Output: The table of divided difference and the coefficients a_i for $0 \le i < n$ (a in the code below) of the interpolating polynomial.

```
%  [a, table] = divideddiff(d)

function [a, table] = divideddiff(d)
  n = size(d,1);
  md = 0;
```

```
  % generate the table of divided differences
  table = repmat(NaN,n,n+1);

  table(:,1) = d(:,1);
  table(1,2) = d(1,2);
  for i=2:n
    if ( table(i,1) == table(i-1,1) )
      table(i,2) = table(i-1,2);
    else
      table(i,2) = d(i,2);
    end
  end
  for k=3:n+1
    for i=1:n-k+2
      m = i+k-2;
      if ( table(m,1) == table(i,1) )
        if ( md == 0 )
          md = m;
        end
        table(i,k) = d(md,2)/factorial(k-2);
      else
        table(i,k)=(table(i+1,k-1)-table(i,k-1))/(table(m,1)-table(i,1));
        md = 0;
      end
    end
    md = 0;
  end

  a = table(1,2:n+1);
end
```

Code 6.2.15 (Nested Form)

To evaluate a polynomial

$$p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_{n-1}(x - x_0)(x - x_1)\ldots(x - x_{n-2})$$

using its nested form.
Input: The coefficients a_i for $0 \leq i < n$ (a in the code below).
The points x_i for $0 \leq i < n - 1$ (X in the code below).
The value x where to evaluate the polynomial (x in the code below).
Output: The value of $p(x)$.

```
%  v = polynomial(X,a,x)

function v = polynomial(X,a,x)
  n=length(a);
```

```
  v = a(n);
  for i=(n-1):-1:1
    v=a(i)+(x-X(i)).*v;
  end
end
```

Remark 6.2.16

1. Let $f : [a,b] \to \mathbb{R}$ be a continuous function and p be the interpolating polynomial of f at the interpolatory points $x_0 < x_1 < \ldots x_n$. Suppose that $c \in [a,b]$ and $c \neq x_i$ for all i. Will $p(c)$ approach $f(c)$ as the number of interpolatory points n increases? Namely, will $p(c)$ give a better approximation of $f(c)$ as n increases? In general, the answer is no.

   When equally spaced points are used in the interpolating polynomial, the approximation of $f(x)$ given by $p(x)$ is generally getting worse as n increases if x is near the endpoints of the interval $[a,b]$. The approximation of $f(x)$ given by $p(x)$ is slowly getting better as n increases if x is near the centre of the interval $[a,b]$.

   Consider the function $f(x) = |x|$ on $[-1,1]$. We list in the next table the result of $p_n(x)$ for some values of x and n, where p_n is the interpolating polynomial of f at the $n+1$ equally spaced points $x_i = -1 + (2i)/n$ for $i = 0, 1, \ldots, n$.

$p_n(x)$		x		
		0	0.8	0.9
n	3	0.25	0.73	0.8575
	7	0.097656250	0.775586650	0.857468784
	15	0.043878794	0.852540445	0.748920770

   The approximation of $f(0)$ improves really slowly when n increases. However, the approximations of $f(0.8)$ and $f(0.9)$ are getting worse when n increases. The results in the table above were rounded to 9 decimals. The graphs of f, p_3, p_7 and p_{15} are given in Figure 6.1.

2. Consider the function $f(x) = 1/(1+x^2)$ on $[-5,5]$. Suppose that p_n is the interpolating polynomial of f at the equally spaced points $x_i = -5 + (10i)/n$ for $i = 0, 1, \ldots, n$. The uniform distance
$$\max_{x\in[-5,5]} |p_n(x) - f(x)|$$
   increases as n increases. See Figure 6.2.

   The approximation of $f(x)$ given by $p_n(x)$ is generally getting worse as n increases if x is near -5 or 5. the approximation of $f(x)$ given by $p_n(x)$ is generally getting slowly better as n increases if x is near the origin.

3. Let $f : [a,b] \to \mathbb{R}$ be a continuous function and p_n be the interpolating polynomial of f at the interpolatory points $x_0 < x_1 < \ldots x_n$. One way to improve the uniform approximation of f by the polynomial p is to use more interpolatory points near the ends of the interval $[a,b]$ than near the middle of the interval $[a,b]$.

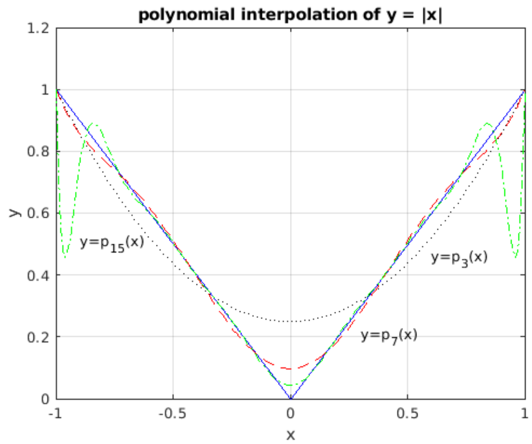


Figure 6.1: The solid blue line is the graph of $f(x) = |x|$ for $-1 \leq x \leq 1$. The dotted grey line is the graph of the interpolating polynomial p_3 of degree at most 3 at $x_i = -1 + 2i/3$ for $i = 0, 1, 2$ and 3. The dashed red line is the graph of the interpolating polynomial p_7 of degree at most 7 at $x_i = -1 + 2i/7$ for $i = 0, 1, \ldots, 7$. The dashed-dotted green line is the graph of the interpolating polynomial p_{15} of degree at most 15 at $x_i = -1 + 2i/15$ for $i = 0, 1, \ldots, 15$.

Good interpolatory points are the **Chebyshev points** x_i adjusted to the interval $[a, b]$ which are defined by

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\,\pi\right)$$

for $i = 0, 1, \ldots, n$.

Among all polynomials of order at most n interpolating f at $(n+1)$ points of $[a, b]$, we will show in Section 9.2 that the interpolating polynomial p_n of f at the Chebyshev points above is the "best uniform approximation" of f on $[a, b]$. Moreover, the approximation $p_n(x)$ of $f(x)$ is as good for x near the endpoints of the interval $[a, b]$ as it is for x near the middle of the interval $[a, b]$.

For instance, let $f(x) = 1/(1+x^2)$ on $[-5, 5]$ as before. From Theorems 6.2.5 and 6.2.7, we have

$$f(x) - p_n(x) = \frac{1}{(n+1)!} \frac{\mathrm{d}^{n+1} f}{\mathrm{d}x^{n+1}}(\xi) \prod_{i=0}^{n} (x - x_i)$$

for some ξ in the smallest interval containing the x_i and x.

We may assume that

$$\frac{1}{(n+1)!} \frac{\mathrm{d}^{n+1} f}{\mathrm{d}x^{n+1}}(\xi)$$

is almost constant on $[-5, 5]$. Hence, the behaviour of $|f(x) - p_n(x)|$ is roughly described by $\left|\prod_{i=0}^{n} (x - x_i)\right|$. We have in Figure 6.3 the graph of $\left|\prod_{i=0}^{n} (x - x_i)\right|$ for $-5 \leq x \leq 5$, where

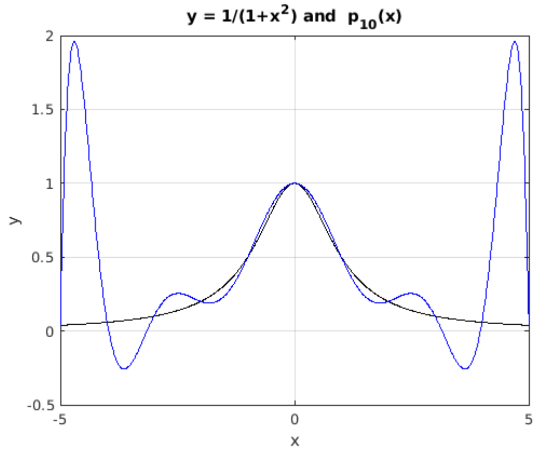


Figure 6.2: The black line is the graph of $f(x) = 1/(1+x^2)$ and the blue line is the graph of the interpolating polynomial p_{10} of f at the 11 equally spaced points $x_i = -5+i$ for $i = 0, 1, \ldots, 10$.

(i) $x_i = -5\cos\left(\dfrac{2i+1}{16}\,\pi\right)$ for $i = 0, 1, \ldots, 7$ are the Chebyshev points adjusted to the interval $[-5,5]$ and

(ii) $x_i = -5+(10i)/7$ with $i = 0, 1, \ldots, 7$ are equally spaced points.

♠

Remark 6.2.17

As we promised in Section 2.7, we now show that if there exist $\alpha > 0$ and $\lambda \neq 0$ such that

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda \ , \tag{6.2.6}$$

then α must be the **golden ratio** $(1+\sqrt{5})/2$. Namely, the order of convergence of the sequence method must be $(1+\sqrt{5})/2$. Obviously, we assume that the secant method yields a sequence $\{x_n\}_{n=0}^\infty$ that converges to a root p of a function f so that (6.2.6) can be stated.

For any distinct real numbers a, b and x, we have

$$f(x) = f(a) + f[a,b](x-a) + f[a,b,x](x-a)(x-b) \ ,$$

where

$$f[a,b] = \frac{f(b)-f(a)}{b-a} \ , \quad f[b,x] = \frac{f(x)-f(b)}{x-b} \quad \text{and} \quad f[a,b,x] = \frac{f[b,x]-f[a,b]}{x-a} \ .$$

From,

$$0 = f(p) = f(a) + f[a,b](p-a) + f[a,b,p](p-a)(p-b) \ ,$$

we get

$$p = a - \frac{f(a)}{f[a,b]} - \frac{f[a,b,p]}{f[a,b]}(p-a)(p-b) \ .$$

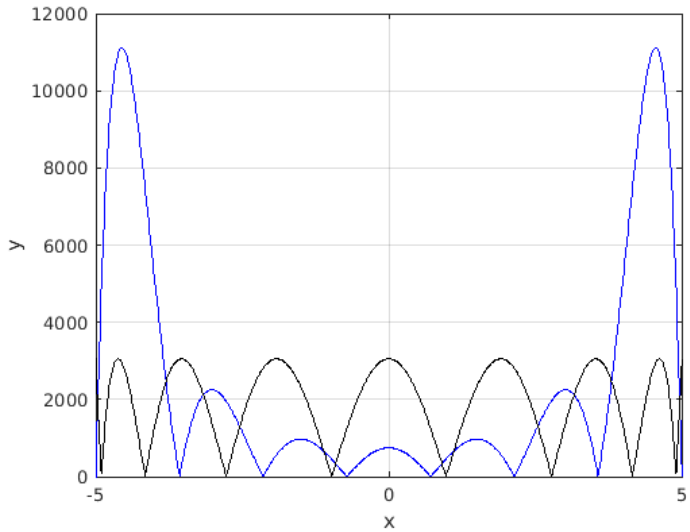


Figure 6.3: The blue line represents the qualitative behavior of the error for interpolating polynomials using equally spaced points and the black line represents the qualitative behavior of the error for interpolating polynomial using Chebyshev points adjusted to the interval $[-5,5]$.

If $a = x_n$ and $b = x_{n-1}$, we get

$$\begin{aligned} p &= \underbrace{x_n - \frac{f(x_n)}{f[x_n, x_{n-1}]}}_{\text{secant method}} - \frac{f[x_n, x_{n-1}, p]}{f[x_n, x_{n-1}]}(p - x_n)(p - x_{n-1}) \\ &= x_{n+1} - \frac{f[x_n, x_{n-1}, p]}{f[x_n, x_{n-1}]}(p - x_n)(p - x_{n-1}) \end{aligned}$$

and thus

$$e_{n+1} = \frac{f[x_n, x_{n-1}, p]}{f[x_n, x_{n-1}]} e_n e_{n-1} \ . \tag{6.2.7}$$

We can rewrite (6.2.7) as $e_{n+1} = c_n e_n e_{n-1}$, where $c_n = \left| \dfrac{f[x_n, x_{n-1}, p]}{f[x_n, x_{n-1}]} \right|$. Hence,

$$\frac{|e_{n+1}|}{|e_n|^\alpha} = \frac{|c_n e_n e_{n-1}|}{|e_n|^\alpha} = c_n |e_n|^{1-\alpha} |e_{n-1}| = c_n \left(\frac{|e_n|}{|e_{n-1}|^\alpha} \right)^\beta ,$$

where β satisfies $\beta = 1 - \alpha$ and $\alpha\beta = -1$. Thus, $-\alpha^2 + \alpha + 1 = 0$. The roots of this polynomial are $(1 \pm \sqrt{5})/2$. We are only interested in the positive root $\alpha = (1 + \sqrt{5})/2$. We then have that $\beta = -1/\alpha = (1 - \sqrt{5})/2$ and $-1 < \beta < 0$.

If $y_n = \dfrac{|e_n|}{|e_{n-1}|^\alpha}$, we get $y_{n+1} = c_n y_n^\beta$. Taking the limit $n \to \infty$ on both sides, we get $\lambda = c_\infty \lambda^\beta$, where

$$c_\infty = \lim_{n\to\infty} c_n = \frac{f''(p)}{2f'(p)} \neq 0 \ .$$

Thus,

$$\lambda = c_\infty^{1/(1-\beta)} = c_\infty^{1/\alpha} \neq 0$$

as expected.

We have shown that, with $\alpha = \dfrac{1+\sqrt{5}}{2}$, we have $\displaystyle \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda = \left(\frac{f''(p)}{2f'(p)}\right)^{1/\alpha} \neq 0$. ♠

6.3 Proofs of Theorems 6.2.2, 6.2.5 and 6.2.7

Definition 6.3.1

Suppose that $f : [a,b] \to \mathbb{R}$ is sufficiently differentiable and $z \in]a,b[$. Then z is a **zero of f of order k** if

$$\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(z) = 0 \quad \text{for} \quad 0 \leq j < k \quad \text{and} \quad \frac{\mathrm{d}^k f}{\mathrm{d}x^k}(z) \neq 0 \ .$$

If $k = 0$, the previous condition is reduced to $f(z) \neq 0$. As usual,

$$\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(z) = f(z)$$

for $j = 0$.

Lemma 6.3.2

If z is a zero of order k of a polynomial p of degree $n \geq k$, then one can write $p(x) = (x-z)^k q(x)$, where q is a polynomial of degree $n-k$ such that $q(z) \neq 0$.

This factorization can be obtained with the help of the Horner Algorithm.

If z is a root of p, then Horner's Theorem, Theorem 2.9.1, yields $p(x) = (x-z)\, q_1(x)$, where q_1 is a polynomial of degree $n-1$. Since $p'(x) = q_1(x) + (x-z)\, q_1'(x)$, we get $p'(z) = q_1(z)$.

If $k > 1$, then $q_1(z) = p'(z) = 0$. Since z is a root of q_1, Horner's Theorem yields $q_1(x) = (x-z)\, q_2(x)$, where q_2 is a polynomial of degree $n-2$. Hence $p(x) = (x-z)^2 q_2(x)$. Since $p''(x) = 2\, q_2(x) + 4(x-z)\, q_2'(x) + (x-z)^2 q_2''(x)$, we get $p''(z) = 2\, q_2(z)$.

If $k > 2$, then $q_2(z) = p''(z)/2 = 0$ and we can again use Horner's Theorem. Since z is a root of q_2, Horner's Theorem yields $q_2(x) = (x-z) q_3(x)$, where q_3 is a polynomial of degree $n-3$. Hence $p(x) = (x-z)^3 q_3(x)$.

It becomes clear that the claim of the first paragraph of the remark can be proved by induction. A shorter proof is given by the Taylor polynomial of p at z.

Proof of Lemma 6.3.2.
The Taylor polynomial of degree n of p at z is p itself because $\dfrac{\mathrm{d}^j p}{\mathrm{d}x^j} = 0$ for $j > n$. Hence,

$$\begin{aligned}
p(x) &= p(z) + \frac{\mathrm{d}p}{\mathrm{d}x}(z)\,(x-z) + \frac{1}{2!}\,\frac{\mathrm{d}^2 p}{\mathrm{d}x^2}(z)\,(x-z)^2 + \ldots + \frac{1}{k!}\,\frac{\mathrm{d}^k p}{\mathrm{d}x^k}(z)\,(x-z)^k \\
&\qquad + \frac{1}{(k+1)!}\,\frac{\mathrm{d}^{k+1} p}{\mathrm{d}x^{k+1}}(z)\,(x-z)^{k+1} + \ldots + \frac{1}{n!}\,\frac{\mathrm{d}^n p}{\mathrm{d}x^n}(z)\,(x-z)^n \\
&= \frac{1}{k!}\,\frac{\mathrm{d}^k p}{\mathrm{d}x^k}(z)\,(x-z)^k + \frac{1}{(k+1)!}\,\frac{\mathrm{d}^{k+1} p}{\mathrm{d}x^{k+1}}(z)\,(x-z)^{k+1} + \ldots + \frac{1}{n!}\,\frac{\mathrm{d}^n p}{\mathrm{d}x^n}(z)\,(x-z)^n \\
&= (x-z)^k \underbrace{\left(\frac{1}{k!}\,\frac{\mathrm{d}^k p}{\mathrm{d}x^k}(z) + \frac{1}{(k+1)!}\,\frac{\mathrm{d}^{k+1} p}{\mathrm{d}x^{k+1}}(z)\,(x-z) + \ldots + \frac{1}{n!}\,\frac{\mathrm{d}^n p}{\mathrm{d}x^n}(z)\,(x-z)^{n-m}\right)}_{=q(x)} \\
&= (x-z)^k q(x) \ ,
\end{aligned}$$

where we have used $\dfrac{\mathrm{d}^j p}{\mathrm{d}x^j}(z) = 0$ for $j = 0, 1, \ldots, k-1$. Moreover, $q(z) = \dfrac{1}{k!}\dfrac{\mathrm{d}^k p}{\mathrm{d}x^k}(z) \neq 0$. ∎

Proof (of Theorem 6.2.2).
Since polynomials of degree n have exactly n (complex) roots (counted with multiplicity), there could be only one polynomial of degree at most n which agrees with f at $x_0, x_1, \ldots, x_n$. Suppose that p_1 and p_2 are two polynomials of degree at most n such that $p_1(x_i) = p_2(x_i)$ for $i = 0, 1, 2, \ldots, n$. Then $p(x) = p_1(x) - p_2(x)$ is a polynomial of degree at most n with $n+1$ roots (counted with multiplicity). Thus, p is the zero polynomial.

The proof of the existence of the polynomial p satisfying the conclusion of the theorem is by induction on n. Without lost of generality, we may assume that $x_0 \leq x_1 \leq x_2 \leq \ldots \leq x_n$

If $n = 1$, then

$$p(x) = \frac{x - x_0}{x_1 - x_0}\,f(x_1) + \frac{x - x_1}{x_0 - x_1}\,f(x_0)$$

satisfies the conclusion of the theorem if $x_0 < x_1$ and

$$p(x) = f(x_0) + f'(x_0)(x - x_0)$$

satisfies the conclusion of the theorem if $x_0 = x_1$.

We assume that the conclusion of the theorem is true for $n = k$ and show that it is then true for $n = k+1$.

$\mathbf{x_0 = x_{k+1}}$) Since $x_0 \leq x_1 \leq \ldots \leq x_{k+1}$, we get $x_0 = x_1 = \ldots = x_{k+1}$. Let p be the Taylor polynomial of degree $k+1$ of f at x_0; namely,

$$\begin{aligned}
p(x) = f(x_0) &+ \frac{\mathrm{d}f}{\mathrm{d}x}(x_0)\,(x - x_0) + \frac{1}{2!}\,\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(x_0)\,(x - x_0)^2 + \ldots \\
&+ \frac{1}{(k+1)!}\,\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(x_0)\,(x - x_0)^{k+1} \ .
\end{aligned} \tag{6.3.1}$$

We obviously have that f and p agree at x_0, x_1, …, x_{k+1}.

$\mathbf{x_0 < x_{k+1}}$) By induction, there exist polynomials q_1 and q_2 of degree k such that q_1 agrees with f at x_0, x_1, … x_k and q_2 agrees with f at x_1, x_2, …, x_{k+1}. Let

$$p(x) = \frac{x - x_0}{x_{k+1} - x_0}\, q_2(x) + \frac{x_{k+1} - x}{x_{k+1} - x_0}\, q_1(x) \ . \tag{6.3.2}$$

We show that f agree with p at x_0, x_1, …, x_{k+1}.

We have that

$$p(x_i) = \frac{x_i - x_0}{x_{k+1} - x_0} q_2(x_i) + \frac{x_{k+1} - x_i}{x_{k+1} - x_0} q_1(x_i) = \frac{x_i - x_0}{x_{k+1} - x_0} f(x_i) + \frac{x_{k+1} - x_i}{x_{k+1} - x_0} f(x_i) = f(x_i)$$

for all $0 < i < k+1$,

$$p(x_0) = \frac{x_0 - x_0}{x_{k+1} - x_0} q_2(x_0) + \frac{x_{k+1} - x_0}{x_{k+1} - x_0} q_1(x_0) = q_1(x_0) = f(x_0)$$

and

$$p(x_{k+1}) = \frac{x_{k+1} - x_0}{x_{k+1} - x_0} q_2(x_{k+1}) + \frac{x_{k+1} - x_{k+1}}{x_{k+1} - x_0} q_1(x_{k+1}) = q_2(x_{k+1}) = f(x_{k+1}) \ .$$

Suppose now that $x_i = x_{i+1} = \ldots = x_{i+r} \neq x_m$ for $m \notin \{i, i+1, \ldots, i+r\}$ with $r > 0$.

The polynomial p has degree at most $k+1$ and

$$\begin{aligned}\frac{\mathrm{d}^j p}{\mathrm{d}x^j}(x) &= \left(\frac{x - x_0}{x_{k+1} - x_0}\right) \frac{\mathrm{d}^j q_2}{\mathrm{d}x^j}(x) + \left(\frac{x_{k+1} - x}{x_{k+1} - x_0}\right) \frac{\mathrm{d}^j q_1}{\mathrm{d}x^j}(x) \\ &\quad + \frac{j}{x_{k+1} - x_0} \left(\frac{\mathrm{d}^{j-1} q_2}{\mathrm{d}x^{j-1}}(x) - \frac{\mathrm{d}^{j-1} q_1}{\mathrm{d}x^{j-1}}(x)\right)\end{aligned} \tag{6.3.3}$$

for all $j \geq 1$ by induction.

If $i = 0$, then

$$\frac{\mathrm{d}^j q_1}{\mathrm{d}x^j}(x_i) = \frac{\mathrm{d}^j q_2}{\mathrm{d}x^j}(x_i) = \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i)$$

for $j = 0$, 1, …, $r - 1$ and

$$\frac{\mathrm{d}^r q_1}{\mathrm{d}x^r}(x_i) = \frac{\mathrm{d}^r f}{\mathrm{d}x^r}(x_i)$$

by definition of q_1 and q_2. Hence, we get from (6.3.3) that

$$\begin{aligned}\frac{\mathrm{d}^j p}{\mathrm{d}x^j}(x_i) &= \underbrace{\left(\frac{x_i - x_0}{x_{k+1} - x_0}\right)}_{=0} \frac{\mathrm{d}^j q_2}{\mathrm{d}x^j}(x_i) + \underbrace{\left(\frac{x_{k+1} - x_i}{x_{k+1} - x_0}\right)}_{=1} \frac{\mathrm{d}^j q_1}{\mathrm{d}x^j}(x_i) \\ &\quad + \frac{j}{x_{k+1} - x_0} \left(\frac{\mathrm{d}^{j-1} q_2}{\mathrm{d}x^{j-1}}(x_i) - \frac{\mathrm{d}^{j-1} q_1}{\mathrm{d}x^{j-1}}(x_i)\right) \\ &= \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i) + \frac{j}{x_{k+1} - x_0} \left(\frac{\mathrm{d}^{j-1} f}{\mathrm{d}x^{j-1}}(x_i) - \frac{\mathrm{d}^{j-1} f}{\mathrm{d}x^{j-1}}(x_i)\right) = \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i)\end{aligned}$$

for $j = 1, 2, \ldots, r$. If $i + r = k + 1$, a similar argument yields

$$\frac{\mathrm{d}^j p}{\mathrm{d}x^j}(x_{i+r}) = \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_{i+r})$$

for $j = 1, 2, \ldots, r$.

If $i \neq 0$ and $i + r \neq k + 1$, then

$$\frac{\mathrm{d}^j q_1}{\mathrm{d}x^j}(x_i) = \frac{\mathrm{d}^j q_2}{\mathrm{d}x^j}(x_i) = \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i)$$

for $j = 0, 1, \ldots, r$ by definition of q_1 and q_2. From (6.3.3), we get

$$\begin{aligned}
\frac{\mathrm{d}^j p}{\mathrm{d}x^j}(x_i) &= \left(\frac{x_i - x_0}{x_{k+1} - x_0}\right)\frac{\mathrm{d}^j q_2}{\mathrm{d}x^j}(x_i) + \left(\frac{x_{k+1} - x_i}{x_{k+1} - x_0}\right)\frac{\mathrm{d}^j q_1}{\mathrm{d}x^j}(x_i) + \frac{j}{x_{k+1} - x_0}\left(\frac{\mathrm{d}^{j-1} q_2}{\mathrm{d}x^{j-1}}(x_i) - \frac{\mathrm{d}^{j-1} q_1}{\mathrm{d}x^{j-1}}(x_i)\right) \\
&= \left(\frac{x_i - x_0}{x_{k+1} - x_0}\right)\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i) + \left(\frac{x_{k+1} - x_i}{x_{k+1} - x_0}\right)\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i) + \frac{j}{x_{k+1} - x_0}\left(\frac{\mathrm{d}^{j-1} f}{\mathrm{d}x^{j-1}}(x_i) - \frac{\mathrm{d}^{j-1} f}{\mathrm{d}x^{j-1}}(x_i)\right) \\
&= \left(\frac{x_i - x_0}{x_{k+1} - x_0} + \frac{x_{k+1} - x_i}{x_{k+1} - x_0}\right)\frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i) = \frac{\mathrm{d}^j f}{\mathrm{d}x^j}(x_i)
\end{aligned}$$

for $j = 1, 2, \ldots, r$. This proves that, for $x_0 < x_{k+1}$, f and p agree at $x_0, x_1, \ldots, x_{k+1}$.

This complete the proof by induction. ∎

Remark 6.3.3

We have from (6.3.1) (after replacing k by $n-1$) that the coefficient of x^n in the interpolating polynomial of f at $x_0, x_1, \ldots, x_n$ is

$$f[x_0, x_1, \ldots, x_n] = \frac{1}{n!}\frac{\mathrm{d}^n f}{\mathrm{d}x^n}(x_0) \tag{6.3.4}$$

when $x_0 = x_1 = \ldots = x_n$.

We have from (6.3.2) (after replacing k by $n-1$) that the coefficient of x^n in the interpolating polynomial of f at $x_0, x_1, \ldots, x_n$ is

$$\begin{aligned}
f[x_0, x_1, \ldots, x_n] &= \frac{1}{x_n - x_0} f[x_1, x_2, \ldots, x_n] - \frac{1}{x_n - x_0} f[x_0, x_1, \ldots, x_{n-1}] \\
&= \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} ,
\end{aligned}$$

when $x_0 \neq x_n$, because $f[x_0, x_1, \ldots, x_{n-1}]$ (resp. $f[x_1, x_2, \ldots, x_n]$) is the coefficient of x^{n-1} in the interpolating polynomial of f at $x_0, x_1, \ldots, x_{n-1}$ (resp. $x_1, x_2, \ldots, x_n$.) ♠

Before proving Theorems 6.2.5 and 6.2.7, we need the following results.

Proposition 6.3.4

Suppose that $f :]a, b[\to \mathbb{R}$ is a n times continuously differentiable fonctions and that $x_0, x_1, x_2, \ldots, x_n$ are $n+1$ distinct points in $]a, b[$, then there exists ξ in the smallest closed interval containing the x_i's such that

$$f[x_0, x_1, x_2, \ldots, x_n] = \frac{1}{n!}\frac{\mathrm{d}^n f}{\mathrm{d}x^n}(\xi) \ . \tag{6.3.5}$$

Proof.
Without lost of generality, we may assume that $x_0 < x_1 < x_2 < \ldots < x_n$.

For $n = 1$, (6.3.5) is the statement of the Mean Value Theorem. There exists ξ between x_0 and x_1 such that

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(\xi) \ .$$

Let p be the interpolating polynomial of degree at most n of f at x_0, x_1, $\ldots$, x_n. Let $g = f - p$. The fonction g has $n+1$ distinct roots at x_0, x_1, $\ldots$, x_n.

We prove by induction that $\dfrac{\mathrm{d}^j g}{\mathrm{d}x^j}$ has $n - j + 1$ distinct roots between x_0 and x_n for $j = 1$, 2, $\ldots$, n.

By the Mean Value Theorem, for each pair of points $\{x_i, x_{i+1}\}$ with $i \in \{0, 1, 2, \ldots, n-1\}$, there exists μ_i between x_i and x_{i+1} such that

$$0 = g(x_{x+1}) - g(x_i) = g'(\mu_i)(x_{i+1} - x_i) \ .$$

Hence $g'(\mu_i) = 0$ for $i = 0$, 1, $\ldots$, $n-1$. The function g' has therefore $n = n - 1 + 1$ distinct roots between x_0 and x_n. The hypothesis of induction is true if $j = 1$.

Suppose that the hypothesis of induction is true for $j = k$; namely, $\dfrac{\mathrm{d}^k g}{\mathrm{d}x^k}$ has $n - k + 1$ distinct roots at ζ_0, ζ_1, $\ldots$, ζ_{n-k} between x_0 and x_n. By the Mean Value Theorem, for each pair of points $\{\zeta_i, \zeta_{i+1}\}$, where $i \in \{0, 1, 2, \ldots, n-k-1\}$, there exists ν_i between ζ_i and ζ_{i+1} such that

$$0 = \frac{\mathrm{d}^k g}{\mathrm{d}x^k}(\zeta_{x+1}) - \frac{\mathrm{d}^k g}{\mathrm{d}x^k}(\zeta_i) = \frac{\mathrm{d}^{k+1} g}{\mathrm{d}x^{k+1}}(\nu_i)(\zeta_{i+1} - \zeta_i) \ .$$

Hence $\dfrac{\mathrm{d}^{k+1} g}{\mathrm{d}x^{k+1}}(\nu_i) = 0$ for $i = 0$, 1, $\ldots$, $n-k-1$. The function $\dfrac{\mathrm{d}^{k+1} g}{\mathrm{d}x^{k+1}}$ has therefore $n - k = n - (k+1) + 1$ distinct roots at ν_0, ν_1, $\ldots$, ν_{n-k-1} between x_0 and x_n. This proves the hypothesis of induction.

We have that $\dfrac{\mathrm{d}^n g}{\mathrm{d}x^n}$ has $n - n + 1 = 1$ root between x_0 and x_n. Let ξ be this root, Since p is a polynomial of degree at most n whose coefficient for x^n is $f[x_0, x_1, \ldots, x_n]$, we have

$$\frac{\mathrm{d}^n g}{\mathrm{d}x^n}(x) = \frac{\mathrm{d}^n f}{\mathrm{d}x^n}(x) - n!\, f[x_0, x_1, \ldots, x_n] \ .$$

Hence,

$$0 = \frac{\mathrm{d}^n f}{\mathrm{d}x^n}(\xi) - n!\, f[x_0, x_1, \ldots, x_n] \ .$$

This proves the proposition. ∎

Proposition 6.3.5

Suppose that $f :]a,b[\to \mathbb{R}$ is a n times continuously differentiable fonctions, then

$$f[x_0, x_1, \ldots, x_n] = \frac{1}{n!} \frac{\mathrm{d}^n f}{\mathrm{d}x^n}(\xi) \tag{6.3.6}$$

for some ξ in the smallest closed interval containing x_0, x_1, ..., x_n.
Moreover, if $\{x_{i,j}\}_{j=0}^{\infty}$ are sequences such that

$$\lim_{j\to\infty} x_{i,j} = x_i$$

for $i = 0, 1, \ldots, n$, then

$$\lim_{j\to\infty} f[x_{0,j}, x_{1.j}, x_{2,j}, \ldots, x_{n,j}] = f[x_0, x_1, x_2, \ldots, x_n] \ . \tag{6.3.7}$$

Proof.
Without lost of generality, we may assume that $x_0 \leq x_1 \leq x_2 \leq \ldots \leq x_n$.

For $n = 0$, we have $f[x_0] = f(x_0)$ and the conclusion of the proposition are obviously true. Note that the case $k = 1$ is also obvious. (6.3.6) is the Mean Value Theorem when $x_0 \neq x_1$ and $f[x_0, x_1] = f'(x_0)$ when $x_0 = x_1$. As for (6.3.7), it follows from the continuity of f when $x_0 \neq x_1$, the definition of f' when $x_0 = x_1$, of the continuity of f' when $x_{1,j} = x_{2,j}$ for all j (large enough).

We suppose that the conclusion of the theorem is true for $n = k$ and show that it is true for $n = k + 1$.

I) First, we prove (6.3.7) with $n = k + 1$ and $x_0 < x_{k+1}$. Since $x_{i,j} \to x_i$ as $j \to \infty$ and $x_0 \neq x_{k+1}$, there exists $J > 0$ such that $x_{0,j} \neq x_{k+1,j}$ for $j \geq J$. Hence, for $j \geq J$, we have

$$\begin{aligned} f[x_{0,j}, x_{1,j}, \ldots, x_{k+1,j}] &= \frac{f[x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] - f[x_{0,j}, x_{1,j}, \ldots, x_{k,j}]}{x_{k+1,j} - x_{0,j}} \\ &\to \frac{f[x_1, x_2, \ldots, x_{k+1}] - f[x_0, x_1, \ldots, x_k]}{x_{k+1} - x_0} \\ &= f[x_0, x_1, x_2, \ldots, x_{k+1}] \end{aligned}$$

because $f[x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] \to f[x_1, x_2, \ldots, x_{k+1}]$ and
$f[x_{0,j}, x_{1,j}, \ldots, x_{k,j}] \to f[x_0, x_1, \ldots, x_k]$ as $j \to \infty$ by hypothesis of induction for $n = k$.

II) Second, we prove (6.3.6) with $n = k+1$. If $x_0 = x_1 = \ldots = x_{k+1}$, then (6.3.6) with $n = k+1$ is just (6.3.4) with $n = k + 1$. If $x_0 < x_{k+1}$, we choose sequences $\{x_{i,j}\}_{j=0}^{\infty}$ in $]a,b[$ such that

$$\lim_{j\to\infty} x_{i,j} = x_i$$

for $i = 0, 1, 2, \ldots, k+1$ and

$$x_0 \leq x_{0,j} < x_{1,j} < x_{2,j} < \ldots < x_{k+1,j} \leq x_{k+1}$$

for all j. From Proposition 6.3.4, there exist $\xi_j \in [x_{0,j}, x_{k+1,j}]$ such that

$$f[x_{0,j}, x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] = \frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi_j)$$

for all j. From (I), we have that

$$\lim_{j\to\infty}\left(\frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi_j)\right) = \lim_{j\to\infty} f[x_{0,j}, x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] = f[x_0, x_1, \ldots, x_{k+1}] \ .$$

Consider

$$\begin{aligned} g :]a,b[&\to \mathbb{R} \\ x &\mapsto \frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(x) \end{aligned}$$

We have $\lim_{j\to\infty} g(\xi_j) = f[x_0, x_1, \ldots, x_{k+1}]$. Since $\xi_j \in [x_{0,j}, x_{k+1,j}] \subset [x_0, x_{k+1}]$ for all j, we have that $g(\xi_j) \in g([x_0, x_{k+1}])$ for all j. Since g is a continuous function by assumption and since the image of a closed and bounded interval (a compact and connected set) by a continuous function like g is a closed and bounded interval (another compact and connected set), it follows that $g([x_0, x_{k+1}])$ is closed and $\lim_{j\to\infty} g(\xi_j) \in g([x_0, x_{k+1}])$. Thus, there exists $\xi \in [x_0, x_{k+1}]$ such that

$$f[x_0, x_1, \ldots, x_{k+1}] = \lim_{j\to\infty}\frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi_j) = \lim_{j\to\infty} g(\xi_j) = g(\xi) = \frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi) \ .$$

This proves (6.3.6) for $n = k+1$.

III) Finally, we prove (6.3.7) with $n = k+1$ and $x_0 = x_1 = \ldots = x_{k+1}$. Let $\{x_{i,j}\}_{j=0}^{\infty}$ be any sequences such that $\lim_{j\to\infty} x_{i,j} = x_i$ for $i = 0, 1, \ldots, k+1$. From (II), there exist $\xi_j \in [x_{0,j}, x_{k+1,j}]$ such that

$$f[x_{0,j}, x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] = \frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi_j)$$

for all j. Moreover, since

$$\lim_{j\to\infty} x_{0,j} = \lim_{j\to\infty} x_{k+1,j} = x_0$$

and $x_{0,j} \leq \xi_j \leq x_{k+1,j}$ for all j, we have that

$$\lim_{j\to\infty} \xi_j = x_0 \ .$$

Hence, by continuity of $\dfrac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}$,

$$\begin{aligned} \lim_{j\to\infty} f[x_{0,j}, x_{1,j}, x_{2,j}, \ldots, x_{k+1,j}] &= \lim_{j\to\infty}\left(\frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(\xi_j)\right) \\ &= \frac{1}{(k+1)!}\frac{\mathrm{d}^{k+1} f}{\mathrm{d}x^{k+1}}(x_0) = f[x_0, x_1, \ldots, x_{k+1}] \ , \end{aligned}$$

where we have used (6.3.4) with $n = k+1$.

This completes the proof by induction. ∎

Proof (of Theorem 6.2.5).
We first prove (6.2.1) by induction on n.

If $n = 0$, then $p(x) = f(x_0)$ for all x is the interpolating polynomial of degree 0 of f at x_0. In addition, we can even note that for $n = 1$, the interpolating polynomial of degree at most 1 of f at x_0 and x_1 is $p(x) = f(x_0)+f[x_0,x_1](x-x_0)$, where $f[x_0,x_1] = (f(x_1)-f(x_0))/(x_1-x_0)$ for $x_0 \neq x_1$ and $f[x_0,x_1] = f'(x_0)$ for $x_0 = x_1$.

We assume that (6.2.1) is true for $n = k$ and show that it is also true for $n = k+1$.

Let p be the interpolating polynomial of degree at most $k+1$ of f at x_0, x_2, ..., x_{k+1}. By definition, the coefficient of x^{k+1} is $f[x_0,x_1,x_2,\ldots,x_{k+1}]$. Let

$$q(x) = p(x) - f[x_0,x_1,x_2,\ldots,x_{k+1}](x-x_0)(x-x_1)\ldots(x-x_k) \ .$$

q is a polynomial of degree at most k that agree with f at x_0, x_2, ..., x_k. By the hypothesis of induction, we have

$$\begin{aligned} q(x) = f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) + \ldots \\ + f[x_0,x_1,\ldots,x_k]\,(x-x_0)(x-x_1)\ldots(x-x_{k-1}) \ . \end{aligned}$$

Hence,

$$\begin{aligned} p(x) &= q(x) + f[x_0,x_1,x_2,\ldots,x_{k+1}](x-x_0)(x-x_1)\ldots(x-x_k) \\ &= f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) + \ldots \\ &\quad + f[x_0,x_1,\ldots,x_{k+1}]\,(x-x_0)(x-x_1)\ldots(x-x_k) \ . \end{aligned}$$

Thus (6.2.1) is true for $k+1$ and this completes the proof of (6.2.1).

(6.2.2) and (6.2.3) follow from Remark 6.3.3 if the interpolating polynomial of f is at x_j, x_{j+1}, ..., x_{j+k} only.

Finally, to prove (6.2.4), let p be the interpolating polynomial of degree at most $n+1$ of f at x_0, x_1, x_2, ..., x_n given by (6.2.1). Let $\tilde{p}$ be the interpolating polynomial of degree at most $n+2$ of f at x_0, x_1, x_2, ..., x_n and $\tilde{x}$. We have

$$\begin{aligned} \tilde{p}(x) &= \sum_{j=0}^{n} f[x_0,x_1,\ldots,x_j](x-x_0)(x-x_1)\ldots(x-x_{j-1}) \\ &\qquad + f[x_0,x_1,\ldots,x_n,\tilde{x}]\,(x-x_0)(x-x_1)\ldots(x-x_n) \\ &= p(x) + f[x_0,x_1,\ldots,x_n,\tilde{x}]\,(x-x_0)(x-x_1)\ldots(x-x_n) \ . \end{aligned}$$

Hence, since $\tilde{p}(\tilde{x}) = f(\tilde{x})$ by construction,

$$f(\tilde{x}) = p(\tilde{x}) + f[x_0,x_1,\ldots,x_n,\tilde{x}]\,(\tilde{x}-x_0)(\tilde{x}-x_1)\ldots(\tilde{x}-x_n) \ .$$

This is (6.2.4) if we substitute $\tilde{x}$ by x. ∎

Proof (of Theorem 6.2.7).
The first part of Theorem 6.2.7 is the first par of Proposition 6.3.5.

To prove (6.2.5) in Theorem 6.2.7, we proceed by induction on j. For $j = 1$, let

$$g(x) = f[x_0, x_1, x_2, \ldots, x_k, x] \ .$$

Since

$$\begin{aligned}\lim_{h\to 0} \frac{g(x+h) - g(x)}{h} &= \lim_{h\to 0} \frac{f[x_0, x_1, x_2, \ldots, x_k, x+h] - f[x_0, x_1, x_2, \ldots, x_k, x]}{h} \\ &= \lim_{h\to 0} f[x_0, x_1, x_2, \ldots, x_k, x, x+h] = f[x_0, x_1, x_2, \ldots, x_k, x, x]\end{aligned}$$

because of (6.3.7), we get

$$\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, x_2, \ldots, x_k, x] = g'(x) = f[x_0, x_1, x_2, \ldots, x_k, x, x] \ .$$

Suppose that (6.2.5) is true for $j = m$. We have by induction that

$$\begin{aligned}&\frac{\mathrm{d}^{m+1}}{\mathrm{d}x^{m+1}} f[x_0, x_1, \ldots, x_k, x] = \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}^m}{\mathrm{d}x^m} f[x_0, x_1, \ldots, x_k, x]\right) \\ &\quad = \frac{\mathrm{d}}{\mathrm{d}x}\left(m!\, f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m+1 \text{ times}}]\right) = m!\, \frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, \ldots, x_k, x, x, \ldots, x] \ .\end{aligned} \tag{6.3.8}$$

Moreover,

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, \ldots, x_k, x, x, \ldots, x] \\ &\qquad = \lim_{h\to 0} \frac{f[x_0, x_1, \ldots, x_k, x+h, x+h, \ldots, x+h] - f[x_0, x_1, \ldots, x_k, x, x, \ldots, x]}{h} \ .\end{aligned}$$

Hence,

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, \ldots, x_k, x, x, \ldots, x] \\ &= \lim_{h\to 0}\Bigg(\big(f[x_0, x_1, \ldots, x_k, x+h, x+h, \ldots, x+h] - f[x_0, x_1, \ldots, x_k, x, x+h, \ldots, x+h]\big) \\ &\qquad + \big(f[x_0, x_1, \ldots, x_k, x, x+h, \ldots, x+h] - f[x_0, x_1, \ldots, x_k, x, x, x+h, \ldots, x+h]\big) \\ &\qquad + \ldots + \big(f[x_0, x_1, \ldots, x_k, x, x, \ldots, x, x+h] - f[x_0, x_1, \ldots, x_k, x, x, \ldots, x]\big)\Bigg)\frac{1}{h} \\ &= \lim_{h\to 0}\Bigg(\frac{1}{h}\big(f[x_0, x_1, \ldots, x_k, x+h, x+h, \ldots, x+h] - f[x_0, x_1, \ldots, x_k, x, x+h, \ldots, x+h]\big) \\ &\qquad + \frac{1}{h}\big(f[x_0, x_1, \ldots, x_k, x, x+h, \ldots, x+h] - f[x_0, x_1, \ldots, x_k, x, x, x+h, \ldots, x+h]\big) \\ &\qquad + \ldots + \frac{1}{h}\big(f[x_0, x_1, \ldots, x_k, x, x, \ldots, x, x+h] - f[x_0, x_1, \ldots, x_k, x, x, \ldots, x]\big)\Bigg) \\ &= \lim_{h\to 0}\Bigg(f[x_0, x_1, \ldots, x_k, x, \underbrace{x+h, \ldots, x+h}_{m+1 \text{ times}}] + f[x_0, x_1, \ldots, x_k, x, x, \underbrace{x+h, \ldots, x+h}_{m \text{ times}}]\end{aligned}$$

$$+ \ldots + f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m+1 \text{ times}}, x+h]\Bigg) .$$

There are $m+1$ divided differences in the previous equality, all converging to

$$f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m+2 \text{ times}}]$$

according to (6.3.7) of Proposition 6.3.5. Hence,

$$\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m=1 \text{ times}}] = (m+1) f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m+2 \text{ times}}] .$$

If we combine with (6.3.8), we get

$$\frac{\mathrm{d}^{m+1}}{\mathrm{d}x^{m+1}} f[x_0, x_1, \ldots, x_k, x] = (m+1)! f[x_0, x_1, \ldots, x_k, \underbrace{x, x, \ldots, x}_{m+2 \text{ times}}] .$$

This completes the proof by induction. ∎

6.4 Exercises

Question 6.1
Let x_0, x_1, ..., x_n be $n+1$ distinct points and let

$$\ell_j(x) = \prod_{\substack{i=0\\ i\neq j}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right)$$

for $j = 0, 1, 2, \ldots, n$. Suppose that p is the Lagrange interpolating polynomial of a function f at x_0, x_1, ..., x_n. Show that

$$f(x) - p(x) = \sum_{j=0}^{n} \left(f(x) - p(x_j)\right) \ell_j(x) .$$

Question 6.2
We would like to find a polynomial p of degree at most two such that $p(0) = \alpha$, $p(1) = \beta$ and $p'(\xi) = \gamma$, where the constants α, β and γ are given. Describe analytically and graphically the variation in the answers as ξ varies. Does your observations contradict the existence and uniqueness of the interpolating polynomial of f?

Question 6.3
Suppose that x_0, x_1, x_2, ..., x_n are $n+1$ distinct points. If p is the interpolating polynomial of f of degree at most $n-1$ at x_0, x_1, ..., x_{n-1} and q is the interpolating polynomial of f of degree at most $n-1$ at x_1, x_2, ..., x_n, show that

$$r(x) = \frac{x - x_n}{x_0 - x_n} p(x) + \frac{x - x_0}{x_n - x_0} q(x)$$

is the interpolating polynomial of degree at most n at x_0, x_1, ..., x_n.

Question 6.4

If p is the interpolating polynomial of f of degree at most n at the $n+1$ distinct points x_0, x_1, ..., x_n, show that the coefficient of x^n in p is $\sum_{i=0}^{n} f(x_i)\ell_i$, where $\ell_i = \prod_{\substack{j=0\\i\neq j}}^{n}\left(\frac{1}{x_i - x_j}\right)$. Conclude that $\sum_{i=0}^{n} f(x_i)\ell_i = 0$ if f is a polynomial of degree less than n.

Question 6.5

We have seen in Question 6.4 that the coefficient of x^n in the interpolating polynomial of f of degree at most n at the $n+1$ distinct points x_0, x_1, x_2, ..., x_n is

$$f[x_0, x_1, x_2, \ldots, x_n] = \sum_{i=0}^{n} f(x_i)\ell_i \ , \tag{6.4.1}$$

where

$$\ell_i = \prod_{\substack{i=0\\i\neq j}}^{n}\left(\frac{1}{x_i - x_j}\right) \ .$$

a) If $x_0 < x_1 < x_2 < \ldots < x_n$, show that the ℓ_j alternate sign.
b) Show that

$$\sum_{j=0}^{n} x_i^n \ell_j = 1 \tag{6.4.2}$$

and

$$\sum_{j=0}^{n} \ell_j = \begin{cases} 1 & \text{if} \quad n = 0 \\ 0 & \text{if} \quad n > 0 \end{cases} \tag{6.4.3}$$

Question 6.6

Prove that

$$f[0, 1, 2, \ldots, m] = \frac{1}{m!}\sum_{j=0}^{m}(-1)^{m-j}\binom{m}{j} f(j) \ . \tag{6.4.4}$$

Hint: $\binom{m}{j-1} + \binom{m}{j} = \binom{m+1}{j}$, where $\binom{p}{q} = 0$ for $q > p$ by convention.

Question 6.7

Some values of a function f are given in the following table.

x	$f(x)$
1	1
1.1	0.904837418
1.3	0.740818221
1.4	0.670320046
1.6	0.548811636
1.8	0.449328964

Use Newton divided difference formula to construct an interpolating polynomial p of degree at most 5 for f at the points 1, $1,1$, $1,3$, 1.4, 1.6 and 1.8.

Approximate $f(1.35)$ using the nested form of the polynomial.

Question 6.8
a) Find the interpolating polynomial p of degree at most 3 of $f(x) = e^{x/2}$ such that $p(0) = f(0)$, $p(2) = f(2)$, $p'(2) = f'(2)$ and $p''(2) = f''(2)$.
b) Use the nested form of the interpolating polynomial that you have found in (a) to approximate $f(1)$.
c) Find an upper bound on the truncation error of the interpolating polynomial p of f on $[0, 2]$.

Question 6.9
a) Find the interpolating polynomial p of degree at most 4 of a function f using all the following information on f.

x	$f(x)$	$f'(x)$	$f''(x)$
0	e		
1	1	-1	1
2	e^{-1}		

b) Use the nested form of the interpolating polynomial that you have found in (a) to approximate $f(1.1)$.
c) Knowing that $|f^{(5)}(x)| \leq e$ for $0 \leq x \leq 2$, find an upper bound on the truncation error of the interpolating polynomial of f on $[0, 2]$.

Question 6.10
Some values of a function f and its derivatives are given in the following table.

x	$f(x)$	$f'(x)$	$f''(x)$
1	1.7165256995	-1.4444065708	0.28798342609
1.8	0.79675974510		
2.4	0.4783590320	-0.32311391318	

Use Newton divided difference formula to construct an interpolating polynomial of degree at most 5 for f at 1, 1, 1, 1.8, 2.4 and 2.4 .

Approximate $f(1.75)$ using the nested form of the polynomial. The exact value is $f(1.75) = 0.83673651441075\ldots$ Compute the absolute error, the relative error and the number of significant digits.

Question 6.11
Let $f(x) = 3xe^x - e^{2x}$. Approximate $f(1.03)$ using the Hermite interpolating polynomial of degree at most five at the points $x_0 = x_1 = 1$, $x_2 = x_3 = 1.05$ and $x_4 = x_5 = 1.07$.

Question 6.12
The following data are given by a polynomial p of unknown degree.

x	0	1	2
$p(x)$	2	1	4

If all third order forward divided differences are 1, find the polynomial p.

Question 6.13

a) Find the interpolating polynomial p of degree at most 4 of

$$f(x) = \cos\left(\frac{\pi}{2} - x\right)$$

that satisfies the following requirements.

x	$f(x)$	$f'(x)$	$f''(x)$
0	0		
$\pi/4$	$\sqrt{2}/2$	$\sqrt{2}/2$	$-\sqrt{2}/2$
$\pi/2$	1		

Use at lest 10-digit rounding arithmetic for all your computations.

b) Use the nested form of the interpolating polynomial that you have found in (a) to approximate $f(\pi/8)$.

c) Find an upper bound on the truncation error of the interpolating polynomial p of f on $[0, \pi/4]$.

d) Sketch the graphs of f and p on the same coordinate system to assess the quality of the interpolating polynomial.

Chapter 7

Splines

In the previous chapter, we showed how to generate a polynomial whose graph traverses a set of points $(x_i, f(x_i))$ for $i = 0, 1, 2, \ldots, n$. This polynomial could be of high degree and not be a very good fit for the function f that produced the points. In the present chapter, we describe several methods to generate a piecewise polynomial function p that may provide a good fit for the function f. For some methods, the piecewise polynomial function p may traverse all the points $(x_i, f(x_i))$ but this is not a necessity. Because p is a piecewise functions, it is possible to impose some conditions on p at the points $(x_i, f(x_i))$ (using what is called "control points") to provide a good fit for the function f.

Some of the methods presented in this chapter could be use to generate a piecewise parametric curve that traverses some points (x_i, y_i) for $0 \leq i \leq n$, and satisfies some conditions at these points by adding some "control points".

7.1 Cubic Spline Interpolation

Let $f : [a, b] \to \mathbb{R}$ be a continuously differentiable function and $a = x_0 < x_1 < \ldots < x_n = b$. We have seen in Remark 6.2.10 that MATLAB uses a piecewise linear function through the points $(x_i, f(x_i))$ for $0 \leq i \leq n$ to sketch the graph of f; to be precise, we should say that MATLAB plot a piecewise linear curve that looks like the graph of f. Instead of using linear interpolation to join the points $(x_{i-1}, f(x_{i-1}))$ and $(x_i, f(x_i))$, we now propose to use cubic polynomial interpolation on the intervals $[x_{i-1}, x_i]$. The function p that we get is called a **piecewise cubic polynomial**. Using cubic polynomials, we can impose a better fit between f and p than with linear polynomials. Cubic spline interpolation is ideal to approximate function with discontinuous derivatives.

Definition 7.1.1

Let $f : [a, b] \to \mathbb{R}$ be a continuously differentiable function and $a = x_0 < x_1 < \ldots < x_n = b$. A **free or natural spline interpolant** for the function f on the **nodes** x_0, x_1, ...,

x_n is a piecewise cubic polynomial p defined as follows.

$$p(x) = p_i(x) \quad \text{if} \quad x_i \le x \le x_{i+1} \ ,$$

where the p_i are polynomials of degree three that satisfy

1. $p_i(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n-1$,
2. $p_i(x_{i+1}) = f(x_{i+1})$ for $i = 0, 1, \ldots, n-1$,
3. $p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$ for $i = 0, 1, \ldots, n-2$,
4. $p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$ for $i = 0, 1, \ldots, n-2$,
5. $p_0''(x_0) = p_{n-1}''(x_n) = 0$.

Definition 7.1.2

If the fifth condition in Definition 7.1.1 is replaced by

5. $p_0'(x_0) = f'(x_0)$ and $p_{n-1}'(x_n) = f'(x_n)$,

then p is called a **clamped spline interpolant** for the function f on the nodes x_0, $x_1, \ldots, x_n$.
If the third, fourth and fifth conditions in Definition 7.1.1 are replaced by

1. $p_i'(x_i) = f'(x_i)$ for $i = 0, 1, \ldots, n-1$,
2. $p_i'(x_{i+1}) = f'(x_{i+1})$ for $i = 0, 1, \ldots, n-1$,

Then p is called a **piecewise cubic Hermite interpolant** for the function f on the nodes x_0, x_1,$\ldots$, x_n.

We now describe how to find the cubic polynomials p_i needed for the spline interpolant. Let $z_i = p''(x_i)$ for $i = 0, 1, \ldots, n$. We need to find the values of the z_i's to satisfy the natural or clamped cubic splines.

Since we assume that the p_i's are cubic polynomials, p_i'' is a linear function through the points (x_i, z_i) and (x_{i+1}, z_{i+1}). Recall that $\Delta x_i = x_{i+1} - x_i$. Hence,

$$p_i''(x) = \frac{z_{i+1} - z_i}{\Delta x_i} x + \frac{x_{i+1} z_i - x_i z_{i+1}}{\Delta x_i} = \left(\frac{z_i}{\Delta x_i}\right)(x_{i+1} - x) + \left(\frac{z_{i+1}}{\Delta x_i}\right)(x - x_i) \ .$$

Integrating twice gives

$$\begin{aligned} p_i(x) &= \left(\frac{z_i}{6\Delta x_i}\right)(x_{i+1} - x)^3 + \left(\frac{z_{i+1}}{6\Delta x_i}\right)(x - x_i)^3 + A_i x + B_i \\ &= \left(\frac{z_i}{6\Delta x_i}\right)(x_{i+1} - x)^3 + \left(\frac{z_{i+1}}{6\Delta x_i}\right)(x - x_i)^3 + C_i(x - x_i) + D_i(x_{i+1} - x) \ , \end{aligned} \tag{7.1.1}$$

where $C_i - D_i = A_i$ and $D_i x_{i+1} - C_i x_i = B_i$.

From $p_i(x_i) = f(x_i)$ and $p_i(x_{i+1}) = f(x_{i+1})$, we get

$$f(x_i) = \left(\frac{z_i}{6}\right)(\Delta x_i)^2 + D_i \Delta x_i \quad \text{and} \quad f(x_{i+1}) = \left(\frac{z_{i+1}}{6}\right)(\Delta x_i)^2 + C_i \Delta x_i \ .$$

Solving for C_i and D_i, we get

$$C_i = \frac{f(x_{i+1})}{\Delta x_i} - \frac{z_{i+1}\Delta x_i}{6} \quad \text{and} \quad D_i = \frac{f(x_i)}{\Delta x_i} - \frac{z_i \Delta x_i}{6} \ .$$

If we substitute these values of C_i and D_i in (7.1.1), we get

$$\begin{aligned} p_i(x) = &\left(\frac{z_i}{6\Delta x_i}\right)(x_{i+1} - x)^3 + \left(\frac{z_{i+1}}{6\Delta x_i}\right)(x - x_i)^3 \\ &+ \left(\frac{f(x_{i+1})}{\Delta x_i} - \frac{z_{i+1}\Delta x_i}{6}\right)(x - x_i) + \left(\frac{f(x_i)}{\Delta x_i} - \frac{z_i\Delta x_i}{6}\right)(x_{i+1} - x) \ . \end{aligned} \tag{7.1.2}$$

To determine the values of the z_i's, we will used the property that $p_i'(x_i) = p_{i-1}'(x_i)$ for $1 \le i \le n-1$. This gives $n-1$ equations to determine the $n+1$ variables z_i for $i = 0, 1, \ldots, n$.

7.1.1 Natural Spline

For the natural spline interpolant, we set $z_0 = z_n = 0$ and determine the values of the other z_i's using $p_i'(x_i) = p_{i-1}'(x_i)$ for $1 \le i \le n-1$.

From (7.1.2), we get

$$\begin{aligned} p_i'(x) = &-\left(\frac{z_i}{2\Delta x_i}\right)(x_{i+1} - x)^2 + \left(\frac{z_{i+1}}{2\Delta x_i}\right)(x - x_i)^2 \\ &+ \left(\frac{f(x_{i+1})}{\Delta x_i} - \frac{z_{i+1}\Delta x_i}{6}\right) - \left(\frac{f(x_i)}{\Delta x_i} - \frac{z_i\Delta x_i}{6}\right) \end{aligned} \tag{7.1.3}$$

for $i = 0, 1, \ldots, n-1$. Hence,

$$\begin{aligned} p_i'(x_i) &= -\left(\frac{z_i}{2}\right)\Delta x_i + \left(\frac{f(x_{i+1})}{\Delta x_i} - \frac{z_{i+1}\Delta x_i}{6}\right) - \left(\frac{f(x_i)}{\Delta x_i} - \frac{z_i\Delta x_i}{6}\right) \\ &= -\frac{z_{i+1}\Delta x_i}{6} - \frac{z_i\Delta x_i}{3} + \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} \ . \end{aligned}$$

Similarly,

$$\begin{aligned} p_{i-1}'(x) = &-\left(\frac{z_{i-1}}{2\Delta x_{i-1}}\right)(x_i - x)^2 + \left(\frac{z_i}{2\Delta x_{i-1}}\right)(x - x_{i-1})^2 \\ &+ \left(\frac{f(x_i)}{\Delta x_{i-1}} - \frac{z_i\Delta x_{i-1}}{6}\right) - \left(\frac{f(x_{i-1})}{\Delta x_{i-1}} - \frac{z_{i-1}\Delta x_{i-1}}{6}\right) \end{aligned}$$

for $i = 1, 2, \ldots, n$. Hence,

$$
\begin{aligned}
p'_{i-1}(x_i) &= \left(\frac{z_i \Delta x_{i-1}}{2}\right) + \left(\frac{f(x_i)}{\Delta x_{i-1}} - \frac{z_i \Delta x_{i-1}}{6}\right) - \left(\frac{f(x_{i-1})}{\Delta x_{i-1}} - \frac{z_{i-1}\Delta x_{i-1}}{6}\right) \\
&= \frac{z_i \Delta x_{i-1}}{3} + \frac{z_{i-1}\Delta x_{i-1}}{6} + \frac{f(x_i) - f(x_{i-1})}{\Delta x_{i-1}} \ .
\end{aligned}
$$

The relation $p'_i(x_i) = p'_{i-1}(x_i)$ yields

$$
\begin{aligned}
&z_{i+1}\Delta x_i + 2z_i\left(\Delta x_i + \Delta x_{i-1}\right) + z_{i-1}\Delta x_{i-1} \\
&\quad = \frac{6}{\Delta x_i}\left(f(x_{i+1}) - f(x_i)\right) - \frac{6}{\Delta x_{i-1}}\left(f(x_i) - f(x_{i-1})\right)
\end{aligned}
\tag{7.1.4}
$$

for $i = 1, 2, \ldots, n-1$. We conclude that the z_i's for $1 \le i \le n-1$ are given by the solution of the $n-1$ dimensional linear system $A\mathbf{z} = \mathbf{b}$, where

$$
A = \begin{pmatrix}
d_1 & u_1 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
l_2 & d_2 & u_2 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & l_3 & d_3 & u_3 & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & 0 & l_4 & d_4 & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \vdots & 0 & l_5 & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & u_{n-5} & 0 & \ldots & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & d_{n-4} & u_{n-4} & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & l_{n-3} & d_{n-3} & u_{d-3} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_{n-2} & d_{n-2} & u_{n-2} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_{n-1} & d_{n-1}
\end{pmatrix}
\tag{7.1.5}
$$

and

$$
\mathbf{b} = \begin{pmatrix}
-z_0\Delta x_0 + \dfrac{6}{\Delta x_1}\left(f(x_2) - f(x_1)\right) - \dfrac{6}{\Delta x_0}\left(f(x_1) - f(x_0)\right) \\
\dfrac{6}{\Delta x_2}\left(f(x_3) - f(x_2)\right) - \dfrac{6}{\Delta x_1}\left(f(x_2) - f(x_1)\right) \\
\vdots \\
\dfrac{6}{\Delta x_{n-2}}\left(f(x_{n-1}) - f(x_{n-2})\right) - \dfrac{6}{\Delta x_{n-3}}\left(f(x_{n-2}) - f(x_{n-3})\right) \\
-z_n\Delta x_{n-1} + \dfrac{6}{\Delta x_{n-1}}\left(f(x_n) - f(x_{n-1})\right) - \dfrac{6}{\Delta x_{n-2}}\left(f(x_{n-1}) - f(x_{n-2})\right)
\end{pmatrix}
$$

with $d_i = 2(\Delta x_{i-1} + \Delta x_i)$, $u_i = \Delta x_i$ and $l_i = \Delta x_{i-1}$.

To evaluate the polynomial p_i defined in (7.1.2), we rewrite it in nested form. If we expand p_i around $x = x_i$, we get

$$
\begin{aligned}
p_i(x) &= \frac{z_{i+1} - z_i}{6\Delta x_i}(x - x_i)^3 + \frac{z_i}{2}(x - x_i)^2 \\
&\quad + \left(-\frac{z_i \Delta x_i}{3} - \frac{z_{i+1}\Delta x_i}{6} + \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i}\right)(x - x_i) + f(x_i) \ .
\end{aligned}
$$

Thus,

$$p_i(x) = \left(\left(\alpha_i(x - x_i) + \beta_i\right)(x - x_i) + \gamma_i\right)(x - x_i) + \delta_i \ , \tag{7.1.6}$$

where

$$\begin{aligned}
\delta_i &= f(x_i) \ , \\
\gamma_i &= -\frac{z_i \Delta x_i}{3} - \frac{z_{i+1}\Delta x_i}{6} + \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} \ , \\
\beta_i &= \frac{z_i}{2} \ , \\
\alpha_i &= \frac{z_{i+1} - z_i}{6\Delta x_i} \ .
\end{aligned} \tag{7.1.7}$$

Example 7.1.3

Using the information in the table below, construct the natural spline interpolant for f on the nodes 0, 1, 2, 3 and 5.

x	$f(x)$
0	1
1	0.540302305868140
2	−0.416146836547142
3	−0.989992496600445
5	0.283662185463226

All the numerical results displayed below will be rounded to 10 digits. The computations are done with more precision.

We have

$$p_i(x) = \left(\left(\alpha_i(x - x_i) + \beta_i\right)(x - x_i) + \gamma_i\right)(x - x_i) + \delta_i$$

on $[x_i, x_{i+1}]$ for $0 \leq i \leq 3$, where $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$ and $x_4 = 5$.

Let $\mathbf{w} \in \mathbb{R}^3$ be the solution of $A\mathbf{w} = \mathbf{b}$, where

$$A = \begin{pmatrix} 2(x_2 - x_0) & x_2 - x_1 & 0 \\ x_2 - x_1 & 2(x_3 - x_1) & x_3 - x_2 \\ 0 & x_3 - x_2 & 2(x_4 - x_2) \end{pmatrix} = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 6 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 6\,\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - 6\,\dfrac{f(x_1) - f(x_0)}{x_1 - x_0} \\ 6\,\dfrac{f(x_3) - f(x_2)}{x_3 - x_2} - 6\,\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} \\ 6\,\dfrac{f(x_4) - f(x_3)}{x_4 - x_3} - 6\,\dfrac{f(x_3) - f(x_2)}{x_3 - x_2} \end{pmatrix} = \begin{pmatrix} -2.9805086897 \\ 2.29562089 \\ 7.26403801 \end{pmatrix} .$$

We find

$$\mathbf{w} = \begin{pmatrix} -0.8728068282 \\ 0.5107186229 \\ 1.125553231 \end{pmatrix} .$$

Let

$$\mathbf{z} = \begin{pmatrix} 0 \\ -0.8728068282 \\ 0.5107186229 \\ 1.125553231 \\ 0 \end{pmatrix} .$$

The coefficients of p_i are given by

$$\begin{aligned} \delta_i &= f(x_i) \ , \\ \gamma_i &= -\frac{z_i(x_{i+1}-x_i)}{3} - \frac{z_{i+1}(x_{i+1}-x_i)}{6} + \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} \ , \\ \beta_i &= \frac{z_i}{2} \end{aligned}$$

and

$$\alpha_i = \frac{z_{i+1}-z_i}{6(x_{i+1}-x_i)}$$

for $i = 0$, 1, 2 and 3.

The following table gives the values of the coefficients of p_i.

i	α_i	β_i	γ_i	δ_i
0	−0.1454678047	0	−0.3142298894	1
1	0.2305875752	−0.4364034141	−0.7506333035	0.5403023059
2	0.1024724346	0.2553593115	−0.9316774061	−0.4161468365
3	−0.09379610255	0.5627766153	−0.1135414794	−0.9899924966

♣

7.1.2 Clamped Spline

For the clamped spline interpolant, z_0 and z_n are free but we have the additional constraints $p'(x_0) = p_0'(x_0) = f'(x_0)$ and $p'(x_n) = p_{n-1}'(x_n) = f'(x_n)$. Using (7.1.3), we get

$$f'(x_0) = p_0'(x_0) = -\frac{z_0 \Delta x_0}{3} - \frac{z_1 \Delta x_0}{6} + \frac{f(x_1)-f(x_0)}{\Delta x_0}$$

and

$$f'(x_n) = p_{n-1}'(x_n) = \frac{z_n \Delta x_{n-1}}{3} + \frac{z_{n-1}\Delta x_{n-1}}{6} + \frac{f(x_n)-f(x_{n-1})}{\Delta x_{n-1}} \ .$$

Hence

$$2z_0\Delta x_0 + z_1 \Delta x_0 = -6f'(x_0) + \frac{6}{\Delta x_0}\left(f(x_1)-f(x_0)\right)$$

and

$$z_{n-1}\Delta x_{n-1} + 2z_n\Delta x_{n-1} = 6f'(x_n) - \frac{6}{\Delta x_{n-1}}\left(f(x_n) - f(x_{n-1})\right) \ .$$

These two equations give two linear equations that we may add to the $(n-1)$ linear equations given in (7.1.4).

If we define $\Delta x_{-1} = 0$ and $\Delta x_n = 0$, the z_i's for $0 \leq i \leq n$ are given by the solution of the $n+1$ dimensional linear system $A\mathbf{z} = \mathbf{b}$, where

$$A = \begin{pmatrix} d_0 & u_0 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ l_1 & d_1 & u_1 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & l_2 & d_2 & u_2 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & 0 & l_3 & d_3 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & 0 & l_4 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & u_{n-4} & 0 & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & d_{n-3} & u_{n-3} & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & l_{n-2} & d_{n-2} & u_{d-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_{n-1} & d_{n-1} & u_{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_n & d_n \end{pmatrix} \tag{7.1.8}$$

and

$$\mathbf{b} = \begin{pmatrix} -6f'(x_0) + \dfrac{6}{\Delta x_0}\left(f(x_1) - f(x_0)\right) \\ \dfrac{6}{\Delta x_1}\left(f(x_2) - f(x_1)\right) - \dfrac{6}{\Delta x_0}\left(f(x_1) - f(x_0)\right) \\ \dfrac{6}{\Delta x_2}\left(f(x_3) - f(x_2)\right) - \dfrac{6}{\Delta x_1}\left(f(x_2) - f(x_1)\right) \\ \vdots \\ \dfrac{6}{\Delta x_{n-2}}\left(f(x_{n-1}) - f(x_{n-2})\right) - \dfrac{6}{\Delta x_{n-3}}\left(f(x_{n-2}) - f(x_{n-3})\right) \\ \dfrac{6}{\Delta x_{n-1}}\left(f(x_n) - f(x_{n-1})\right) - \dfrac{6}{\Delta x_{n-2}}\left(f(x_{n-1}) - f(x_{n-2})\right) \\ 6f'(x_n) - \dfrac{6}{\Delta x_{n-1}}\left(f(x_n) - f(x_{n-1})\right) \end{pmatrix}$$

with d_i, u_i and l_i for $0 \leq i \leq n$ defined as for the natural spline before.

The expression for p_i given in (7.1.6) and (7.1.7) is still valid for the clamped cubic spline interpolant.

Example 7.1.4

Using the information in the table below, construct the clamped spline interpolant for f on the nodes 0, 0.3 and 1.

x	0	0.3	1
$f(x)$	1	0.548811636094027	0.135335283236613
$f'(x)$	-2		-0.270670566473225

All the numerical results displayed will be rounded to 10 digits. The computations are done with more precision.

We have
$$p_i(x) = \left(\left(\alpha_i(x-x_i)+\beta_i\right)(x-x_i)+\gamma_i\right)(x-x_i)+\delta_i$$
on $[x_i, x_{i+1}]$ for $i = 0$ and 1, where $x_0 = 0$, $x_1 = 0.3$ and $x_2 = 1$.

Let $\mathbf{z} \in \mathbb{R}^3$ be the solution of $A\mathbf{z} = \mathbf{b}$, where
$$A = \begin{pmatrix} 2(x_1-x_0) & x_1-x_0 & 0 \\ x_1-x_0 & 2(x_2-x_0) & x_2-x_1 \\ 0 & x_2-x_1 & 2(x_2-x_1) \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 & 0 \\ 0.3 & 2 & 0.7 \\ 0 & 0.7 & 1.4 \end{pmatrix}$$
and
$$\mathbf{b} = \begin{pmatrix} -6f'(x_0) + 6\,\dfrac{f(x_1)-f(x_0)}{x_1-x_0} \\ 6\,\dfrac{f(x_2)-f(x_1)}{x_2-x_1} - 6\,\dfrac{f(x_1)-f(x_0)}{x_1-x_0} \\ 6f'(x_2) - 6\,\dfrac{f(x_2)-f(x_1)}{x_2-x_1} \end{pmatrix} = \begin{pmatrix} 2.976232722 \\ 5.479684254 \\ 1.920059626 \end{pmatrix} .$$
We find
$$\mathbf{z} = \begin{pmatrix} 3.949875177 \\ 2.021025387 \\ 0.3609584679 \end{pmatrix} .$$

The coefficients of p_i are given by
$$\begin{aligned} \delta_i &= f(x_i) \ , \\ \gamma_i &= -\frac{z_i \Delta x_i}{3} - \frac{z_{i+1}\Delta x_i}{6} + \frac{f(x_{i+1})-f(x_i)}{\Delta x_i} \ , \\ \beta_i &= \frac{z_i}{2} \end{aligned}$$
and
$$\alpha_i = \frac{z_{i+1}-z_i}{6\Delta x_i}$$
for $i = 0$ and 1.

The following table gives the values of the coefficients of p_i.

i	α_i	β_i	γ_i	δ_i
0	−1.071583217	1.974937588	−2	1
1	−0.3952540283	1.010512693	−1.104364916	0.5488116361

♣

We give below a code to find the clamped cubic spline interpolant. We leave the task of writing a code to find the natural cubic spline interpolant to the reader.

Code 7.1.5 (Clamped Cubic Spline Interpolant - System)

This program computes the tridiagonal matrix A and the right hand side $\mathbf{b}$ associated to the clamped cubic spline interpolant.
Input: The nodes x_i for $0 \leq i \leq n$ (x(i+1) in the code below).
The values $f(x_i)$ for $0 \leq i \leq n$ (f(i+1) in the code below).
The values $f'(x_0)$ and $f'(x_n)$ (fx(1) and fx(2) respectively in the code below).
Output: The lower diagonal L, the diagonal D and the upper diagonal U of the tridiagonal matrix A.
The right hand side $\mathbf{b}$ of $A\mathbf{x} = \mathbf{b}$.

```
%  [L,D,U,b] = clampedsplinematrix(f,fx,x)

function [L,D,U,b] = campledsplinematrix(f,fx,x)
  N = length(x);
  L = repmat(NaN,1,N-1);
  U = repmat(NaN,1,N-1);
  D = repmat(NaN,1,N);
  b = repmat(NaN,1,N);

  dx = x(2)-x(1);
  if (dx == 0)
    return;
  end
  ratio = (f(2)-f(1))/dx;
  D(1) = 2*dx;
  U(1) = dx;
  b(1) = 6*(ratio - fx(1));

  for n=2:N-1
    prevdx = dx;
    dx = x(n+1)-x(n);
    if (dx == 0)
      return;
    end
    prevratio = ratio;
    ratio = (f(n+1)-f(n))/dx;
    L(n-1) = prevdx;
    D(n) = 2*(dx+prevdx);
    U(n) = dx;
    b(n) = 6*(ratio - prevratio);
  end

  L(N-1) = dx;
  D(N) = 2*dx;
  b(N) = 6*(fx(2) - ratio);
end
```

Code 7.1.6 (Tridiagonal Matrix)

To solve a system of the form $A\mathbf{x} = \mathbf{b}$, where A is a tridiagonal matrix.
Input: The lower diagonal L, the diagonal D and the upper diagonal U of the tridiagonal matrix A. None of the components of the diagonal D can be null.
The right hand side $\mathbf{b}$.
Output: The solution if the system can be solved.

```
%  z = tridmatrix(L,D,U,b)

function z = tridmatrix(L,D,U,b)
  m = length(D);
  z = repmat(NaN,1,m);

  for n=2:m
    if (D(n-1) == 0)
      return;
    end
    q = L(n-1)/D(n-1);
    D(n) = D(n)-q*U(n-1);
    b(n) = b(n)-q*b(n-1);
  end

  if (D(m) == 0)
    return;
  end

  % Backward substitution
  z(m) = b(m)/D(m);
  for n=(m-1):-1:1
    z(n)=(b(n)-U(n)*z(n+1))/D(n);
  end
end
```

Code 7.1.7 (Cubic Spline Interpolant - Polynomial)

To evaluate a cubic spline interpolant defined by

$$p(x) = (\alpha_i(x - x_i) + \beta_i) * (x - x_i) + \gamma_i) * (x - x_i) + \delta_i$$

for $x_i < x \leq x_{i+1}$.
Input: The points x_i for $0 \leq i \leq n$ (x(i+1) in the code below).
The values $f(x_i)$ for $0 \leq i \leq n$ (f(i+1) in the code below).
The solution $\mathbf{z}$ of the system $A\mathbf{z} = \mathbf{b}$ associated to the cubic spline used.
The values of x where the cubic spline interpolant must be evaluated (X in the code below).
Output: The value of the cubic spline interpolant at all the given values of x.

The coefficients for each polynomials

$$p_i(x) = ((c_{i,1}(x - x_i) + c_{i,2})(x - x_i) + c_{i,3})(x - x_i) + c_{i,4}$$

for $i = 1$,2, ..., $n - 1$ (the matrix coeffs in the code below).

```
function [y, coeffs] = splinepoly(z,f,x,X)
  npoints = length(x);
  N = length(X);
  y = repmat(NaN,1,N);

  for m=1:1:npoints-1
    coeffs(m,4) = f(m);
    dx = x(m+1)-x(m);
    df = f(m+1)-f(m);
    coeffs(m,3) = -(2*z(m)+z(m+1))*dx/6 + df/dx;
    coeffs(m,2) = z(m)/2;
    coeffs(m,1) = (z(m+1)-z(m))/(6*dx);
  end

  for n=1:1:N
    J = 0;
    if ( X(n) >= x(1) && X(n) <= x(npoints) )
      for m = 2:1:npoints
        if ( X(n) <= x(m) )
          J = m-1;
          break;
        end
      end
      dx = X(n) - x(J);
      y(n) = ((coeffs(J,1)*dx + coeffs(J,2))*dx + coeffs(J,3))*dx ...
             + coeffs(J,4);
    end
  end
end
```

7.1.3 Existence of Interpolants

The are a few questions that come naturally after the presentation of the natural and clamped spline interpolants. First, do the linear systems of the form $A\mathbf{z} = \mathbf{b}$ used to find the natural and clamped spline interpolants have always a solution? If so, it is unique? How good are the natural and clamped spline interpolant? We answer these questions below.

Proposition 7.1.8

If B is a strictly diagonally dominant $n \times n$ matrix, then B is invertible.

Proof.
Suppose that $\mathbf{x} \neq \mathbf{0}$ satisfies $B\mathbf{x} = \mathbf{0}$. Let k be an index such that

$$|x_k| = \|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i| \ .$$

We have $|x_k| > 0$ because $\mathbf{x} \neq \mathbf{0}$.

From $\displaystyle\sum_{j=1}^{n} b_{k,j}x_j = 0$, we get

$$b_{k,k} = \sum_{\substack{j=0\\ j\neq k}}^{n} b_{k,j}\left(\frac{x_j}{x_k}\right) \ .$$

Hence

$$|b_{k,k}| \leq \sum_{\substack{j=0\\ j\neq k}}^{n} |b_{k,j}|\left|\frac{x_j}{x_k}\right| \leq \sum_{\substack{j=0\\ j\neq k}}^{n} |b_{k,j}| \ .$$

This contradict that B is strictly diagonally dominant. ∎

Theorem 7.1.9

Let $f : [a, b] \to \mathbb{R}$ be a continuously differentiable function and $a = x_0 < x_1 < \ldots < x_n = b$. There exists a unique natural cubic spline interpolant for f on the nodes x_0, x_1, ..., x_n. Similarly, There exists a unique clamped cubic spline interpolant for f on the nodes x_0, x_1, ..., x_n.

Proof.
Any natural cubic spline on the nodes x_0, x_2, ..., x_n has to satisfy the system $A\mathbf{z} = \mathbf{b}$ for A given in (7.1.5). Since A is strictly diagonally dominant, it follows from the previous proposition that A is invertible. Thus, the solution of $A\mathbf{z} = \mathbf{b}$ is unique. The same reasoning is true for clamped cubic splines with A given in (7.1.8). ∎

Theorem 7.1.10

If p is the natural cubic spline interpolant for a function f of class C^2 on the nodes $a = x_0 < x_1 < \ldots < x_n = b$, then

$$\int_a^b (p''(x))^2\,\mathrm{d}x \leq \int_a^b (f''(x))^2\,\mathrm{d}x \ .$$

Proof.
Let $g = f - p$. We have

$$\begin{aligned}\int_a^b (f''(x))^2\,\mathrm{d}x &= \int_a^b (g''(x) + p''(x))^2\,\mathrm{d}x\\ &= \int_a^b (g''(x))^2\,\mathrm{d}x + \int_a^b (p''(x))^2\,\mathrm{d}x + 2\int_a^b g''(x)p''(x)\,\mathrm{d}x \ .\end{aligned}$$

To prove the theorem, we show that $\displaystyle\int_a^b g''(x)p''(x)\,dx = 0$.

Using integration by parts and $p''_{n-1}(x_n) = p''_0(x_0) = 0$ for the natural cubic spline, we get

$$\begin{aligned}\int_a^b g''(x)p''(x)\,dx &= \sum_{j=0}^{n-1}\int_{x_j}^{x_{j+1}} g''(x)p''_j(x)\,dx \\ &= \sum_{j=0}^{n-1}\left(g'(x_{j+1})p''_j(x_{j+1}) - g'(x_j)p''_j(x_j)\right) - \sum_{j=0}^{n-1}\int_{x_j}^{x_{j+1}} g'(x)p'''_j(x)\,dx \\ &= g'(x_n)p''_{n-1}(x_n) - g'(x_0)p''_0(x_0) - \sum_{j=0}^{n-1}\int_{x_j}^{x_{j+1}} g'(x)\left(\frac{z_{i+1}-z_i}{\Delta x_i}\right)dx \\ &= -\sum_{j=0}^{n-1}\left(\frac{z_{i+1}-z_i}{\Delta x_i}\right)\left(g(x_{j+1})-g(x_j)\right) = 0 \ .\end{aligned}$$

The last equality comes from $g(x_j) = 0$ for all j because $p(x_j) = f(x_j)$ for all j. ∎

Using the approach presented in the next section, it is possible to prove the following theorem.

Theorem 7.1.11

Let $f : [a,b] \to \mathbb{R}$ be a four times continuously differentiable function and suppose that

$$\max_{x\in[a,b]} |f^{(4)}(x)| < M \ .$$

If p is the clamped cubic spline interpolant for f on x_0, $x_1, \ldots,$ x_n in $]a,b[$, then

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{5M}{384} \max_{0\leq i<n-1} |\Delta x_i|^4$$

and

$$\max_{x\in[a,b]} |f'(x) - p'(x)| \leq \frac{M}{24} \max_{0\leq i<n-1} |\Delta x_i|^3 \ .$$

It follows from the previous theorem that the clamped cubic spline polynomial p can be a good fit for a function f if $\displaystyle\max_{0\leq i<n}|\Delta x_i|$ is small enough.

7.1.4 Another Approach

The presentation of the cubic spline that follows is based on [10].

Suppose that p is a cubic spline defined in Definition 7.1.1. If we express p_i as

$$p_i(x) = p(x_i) + p[x_i,x_i](x-x_i) + p[x_i,x_i,x_{i+1}](x-x_i)^2 + p[x_i,x_i,x_{i+1},x_{i+1}](x-x_i)^2(x-x_{i+1})$$

and substitute $(x - x_{i+1}) = (x - x_i) + (x_i - x_{i+1})$, we get

$$\begin{aligned} p_i(x) &= p(x_i) + p[x_i, x_i](x - x_i) + \left(p[x_i, x_i, x_{i+1}] - (x_{i+1} - x_i)p[x_i, x_i, x_{i+1}, x_{i+1}]\right)(x - x_i)^2 \\ &\quad + p[x_i, x_i, x_{i+1}, x_{i+1}](x - x_i)^3 \ . \end{aligned}$$

Hence, we have

$$p_i(x) = a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3 \ ,$$

where

$$\begin{aligned} a_i &= p(x_i) \ , \\ b_i &= p[x_i, x_i] = p'(x_i) \ , \\ d_i &= p[x_i, x_i, x_{i+1}, x_{i+1}] = \frac{p[x_i, x_{i+1}, x_{i+1}] - p[x_i, x_i, x_{i+1}]}{\Delta x_i} \\ &= \frac{p[x_{i+1}, x_{i+1}] - 2p[x_i, x_{i+1}] + p[x_i, x_i]}{(\Delta x_i)^2} \\ &= \frac{b_{i+1} - 2f[x_i, x_{i+1}] + b_i}{(\Delta x_i)^2} \ , \\ c_i &= p[x_i, x_i, x_{i+1}] - (x_{i+1} - x_i)p[x_i, x_i, x_{i+1}, x_{i+1}] \\ &= \frac{p[x_i, x_{i+1}] - p[x_i, x_i]}{\Delta x_i} - p[x_i, x_i, x_{i+1}, x_{i+1}]\Delta x_i = \frac{f[x_i, x_{i+1}] - b_i}{\Delta x_i} - d_i \Delta x_i \\ &= \frac{-2b_i - b_{i+1} + 3f[x_i, x_{i+1}]}{\Delta x_i} \end{aligned} \tag{7.1.9}$$

for $i = 0, 1, \ldots, n-1$. We have that $p[x_i, x_{i+1}] = f[x_i, x_{i+1}]$ because $p(x_i) = f(x_i)$ for all i. We also have that $p[x_i, x_i] = p'(x_i)$ for $0 \leq i \leq n$.

There are only $n+1$ unknowns in (7.1.9); namely, b_i for $i = 0, 1, \ldots, n$.

The conditions $p''_{i-1}(x_i) = p''_i(x_i)$ for $1 \leq i \leq n-1$ imply that

$$2c_{i-1} + 6d_{i-1}\Delta x_{i-1} = 2c_i$$

for $1 \leq i \leq n-1$. Using the definitions of c_i and d_i in (7.1.9), we get

$$(\Delta x_i)b_{i-1} + 2(\Delta x_i + \Delta x_{i-1})b_i + (\Delta x_{i-1})b_{i+1} = 3\left(f[x_{i-1}, x_i]\Delta x_i + f[x_i, x_{i+1}]\Delta x_{i-1}\right)$$

for $1 \leq i \leq n-1$. Since we have $n+1$ unknowns and $n-1$ equations, we have two free variables. It is natural to take b_0 and b_n as free variables.

For the clamped cubic spline interpolant, we require $p'_0(x_0) = f'(x_0)$ and $p'_{n-1}(x_n) = f'(x_n)$. Since $p'_0(x_0) = b_0$ and

$$\begin{aligned} p'_{n-1}(x_n) &= b_{n-1} + 2c_{n-1}(x_n - x_{n-1}) + 3d_{n-1}(x_n - x_{n-1})^2 \\ &= b_{n-1} + 2\left(\frac{-2b_{n-1} - b_n + 3f[x_{n-1}, x_n]}{\Delta x_{n-1}}\right)\Delta x_{n-1} \\ &\quad + 3\left(\frac{b_n - 2f[x_{n-1}, x_n] + b_{n-1}}{(\Delta x_{n-1})^2}\right)(\Delta x_{n-1})^2 = b_n \ , \end{aligned}$$

we have $b_0 = f'(x_0)$ and $b_n = f'(x_n)$. The other b_i's are given by the solution of $n-1$ dimensional linear system $A\mathbf{b} = \mathbf{q}$ where

$$A = \begin{pmatrix} d_1 & u_0 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ l_2 & d_2 & u_1 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & l_3 & d_3 & u_2 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & 0 & l_4 & d_4 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & 0 & l_5 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & u_{n-6} & 0 & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & d_{n-4} & u_{n-5} & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & l_{n-3} & d_{n-3} & u_{d-4} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_{n-2} & d_{n-2} & u_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & l_{n-1} & d_{n-1} \end{pmatrix}$$

and

$$\mathbf{q} = \begin{pmatrix} -b_0\Delta x_1 + 3\left(f[x_0,x_1]\Delta x_1 + f[x_1,x_2]\Delta x_0\right) \\ 3\left(f[x_1,x_2]\Delta x_2 + f[x_2,x_3]\Delta x_1\right) \\ \vdots \\ 3\left(f[x_{n-3},x_{n-2}]\Delta x_{n-2} + f[x_{n-2},x_{n-1}]\Delta x_{n-3}\right) \\ -b_n\Delta x_{n-2} + 3\left(f[x_{n-2},x_{n-1}]\Delta x_{n-1} + f[x_{n-1},x_n]\Delta x_{n-2}\right) \end{pmatrix}$$

with $d_i = 2(\Delta x_{i-1} + \Delta x_i)$, $u_i = \Delta x_i$ and $l_i = \Delta x_{i-1}$.

This gives us another formulation for the clamped cubic spline.

Remark 7.1.12
To find the piecewise cubic Hermite interpolant p for a function f on the nodes x_0, x_1,..., x_n, one uses the formulas above with $b_i = f[x_i,x_i] = f'(x_i)$ for $i = 0, 1, \ldots, n$. The piecewise cubic Hermite interpolant gives a good approximation of f but requires almost twice as much information about f than the clamped or free spline interpolant. We need to know $f'(x_i)$ for $i = 0, 1, \ldots, n$. ♠

Example 7.1.13
Using the information in the table below and the approach developed in this subsection, construct the clamped spline interpolant for f on the nodes 0, 0.3 and 1.

x	0	0.3	1
$f(x)$	1	0.548811636094027	0.135335283236613
$f'(x)$	−2		−0.270670566473225

All the numerical results displayed will be rounded to 10 digits. The computations are done with more precision.

We have

$$p_i(x) = a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3$$

on $[x_i, x_{i+1}]$ for $i = 0$ and 1, where $x_0 = 0$, $x_1 = 0.3$ and $x_2 = 1$.

The coefficients of p_i are given by

$$a_i = p(x_i) \ , \quad b_i = p'(x_i) \ , \quad d_i = \frac{b_{i+1} - 2f[x_i, x_{i+1}] + b_i}{(\Delta x_i)^2}$$

and

$$c_i = \frac{-2b_i - b_{i+1} + 3f[x_i, x_{i+1}]}{\Delta x_i}$$

for $i = 0$ and 1. We have $b_0 = f'(0)$, $b_2 = f'(1)$ and b_1 is the solution of

$$(\Delta x_1)b_0 + 2(\Delta x_0 + \Delta x_1)b_1 + (\Delta x_0)b_2 = 3\left(f[x_0, x_1]\Delta x_1 + f[x_1, x_2]\Delta x_0\right) \ ;$$

namely,

$$0.7f'(0) + 2b_1 + 0.3f'(1) = 3\left(\frac{0.7}{0.3}\left(f(0.3) - f(0)\right) + \frac{0.3}{0.7}\left(f(1) - f(0.3)\right)\right) \ .$$

Thus $b_1 = -1.104364916$.

The following table gives the values of the coefficients of p_i.

i	d_i	c_i	b_i	a_i
0	−1.071583217	1.974937588	−2	1
1	−0.3952540283	1.010512693	−1.104364916	0.5488116361

As expected, we find the same clamped cubic spline as in Example 7.1.4. ♣

7.2 Parametric Curves: Bézier Curves

A general curves C in the plane is the image of a vector valued function $\phi : [a, b] \to \mathbb{R}^2$. The function ϕ is called a **parametric representation** of the curve C. The parametric representation of a curve is not unique.

It is not always possible to describe a curve C by the graph of a function $y = f(x)$ for $a \leq x \leq b$. When it is possible, $\phi : [a, b] \to \mathbb{R}^2$ defined by $\phi(x) = (x, f(x))$ is a parametric representation of the curve C.

Example 7.2.1
The circle C of radius 1 centred at the origin has the following well known parametric representation.

$$\phi(\theta) = (\cos(\theta), \sin(\theta))$$

for $0 \leq \theta \leq 2\pi$. It is impossible to describe the full circle as the graph of a function $y = f(x)$. ♣

Given $n+1$ points $\mathbf{p}_0 = (x_0, y_0)$, $\mathbf{p}_1 = (x_1, y_1)$, $\ldots$, $\mathbf{p}_n = (x_n, y_n)$, the goal is to find polynomial maps of degree three $\phi_i : [0, 1] \to \mathbb{R}^2$ such that $\phi_i(0) = \mathbf{p}_i$ and $\phi_i(1) = \mathbf{p}_{i+1}$

for $0 \leq i < n$. By pasting all the mappings ϕ_i together, we hope to get a nice parametric representation of a curve.

The curves that we are going to describe are called **cubic Bézier curves**. The mapping $\phi_i : [0, 1] \to \mathbb{R}^2$ between the points $\mathbf{p}_i = (x_i, y_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1})$, is defined by

1. $\phi_i(0) = \mathbf{p}_i$,
2. $\phi_i(1) = \mathbf{p}_{i+1}$,
3. $\phi_i'(0) = 3(\alpha_i, \beta_i)$ and
4. $\phi_i'(1) = 3(\alpha_{i+1}, \beta_{i+1})$,

where the α_i's and β_i's are parameters to be described later.

Let $\check{\mathbf{q}}_i = (x_i + \alpha_i, y_i + \beta_i)$ and $\hat{\mathbf{q}}_{i+1} = (x_{i+1} - \alpha_{i+1}, y_{i+1} - \beta_{i+1})$. It is easy to see that

$$\phi_i(t) = (1-t)^3\mathbf{p}_i + 3t(1-t)^2\check{\mathbf{q}}_i + 3t^2(1-t)\hat{\mathbf{q}}_{i+1} + t^3\mathbf{p}_{i+1} \tag{7.2.1}$$

satisfies the four conditions above.

The points $\check{\mathbf{q}}_i$ and $\hat{\mathbf{q}}_{i+1}$ are called the **control points** of the Bézier curve with endpoints $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$.

For the parametric representation between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$,

$$\left.\frac{\partial y}{\partial x}\right|_{x=\mathbf{p}_i} = \frac{\beta_i}{\alpha_i} \ .$$

As long as the ratio β_i/α_i is constant, the parametric representation has the same slope at $\mathbf{p}_i$. By taking α_i and β_i very large, we flatten the image of the representation near $\mathbf{p}_i$.

We illustrate graphically the meaning of the parameters α_i, β_i. α_{i+1} and β_{i+1} in Figure 7.1.

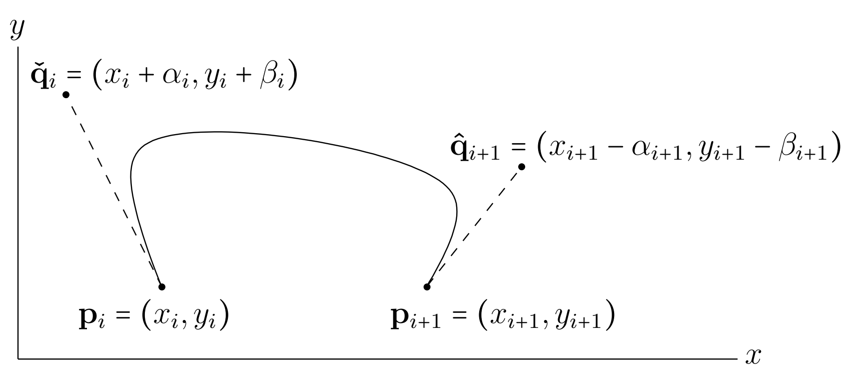


Figure 7.1: Piece of a Bézier curve

(7.2.1) can be rewritten as

$$\begin{aligned}\phi_i &= (-t^3 + 3t^2 - 3t + 1)\mathbf{p}_i + (3t^3 - 6t^2 + 3t)\check{\mathbf{q}}_i + (-3t^3 + 3t^2)\hat{\mathbf{q}}_{i+1} + t^3\mathbf{p}_{i+1} \\ &= \left(-\mathbf{p}_i + 3\check{\mathbf{q}}_i - 3\hat{\mathbf{q}}_{i+1} + \mathbf{p}_{i+1}\right) t^3 + \left(3\mathbf{p}_i - 6\check{\mathbf{q}}_i + 3\hat{\mathbf{q}}_{i+1}\right) t^2 \\ &\qquad + \left(-3\mathbf{p}_i + 3\check{\mathbf{q}}_i\right) t + \mathbf{p}_i \ . \end{aligned} \tag{7.2.2}$$

It is (7.2.2) instead of (7.2.1) that is used in computer codes to draw Bézier curves. The coefficients are computed once and the nested form of (7.2.2) is used. The number of arithmetic operations is minimal.

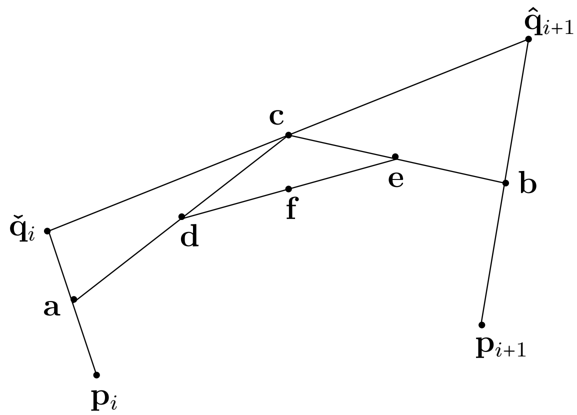


Figure 7.2: Construction of a Bézier curve

Remark 7.2.2
There is a nice geometric interpretation of the Bézier curve.

Let $\mathbf{a}$ be the middle point of the line from $\mathbf{p}_i$ to $\check{\mathbf{q}}_i$, $\mathbf{b}$ be the middle point of the line from $\mathbf{p}_{i+1}$ to $\hat{\mathbf{q}}_{i+1}$, $\mathbf{c}$ be the middle point of the line from $\check{\mathbf{q}}_i$ to $\hat{\mathbf{q}}_{i+1}$, $\mathbf{d}$ be the middle point of the line from $\mathbf{a}$ to $\mathbf{c}$, $\mathbf{e}$ be the middle point of the line from $\mathbf{b}$ to $\mathbf{c}$, and $\mathbf{f}$ be the middle point of the line from $\mathbf{d}$ to $\mathbf{e}$.

The point $\mathbf{f}$ is on the Bézier curve Γ with endpoints $\mathbf{p}_i$, $\mathbf{p}_{i+1}$ and control points $\check{\mathbf{q}}_i$, $\hat{\mathbf{q}}_{i+1}$. Moreover, the Bézier curve Γ is the pasting of the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{f}$ and control points $\mathbf{a}$, $\mathbf{d}$, and the Bézier curve with endpoints $\mathbf{f}$, $\mathbf{p}_{i+1}$ and control points $\mathbf{e}$, $\mathbf{b}$.

We can apply the previous construction to the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{f}$ and control points $\mathbf{a}$, $\mathbf{d}$ to find another point $\mathbf{f}'$ on the the Bézier curve Γ. Similarly, the Bézier curve with endpoints $\mathbf{f}$, $\mathbf{p}_{i+1}$ and control points $\mathbf{e}$, $\mathbf{b}$ gives another point $\mathbf{f}''$ on the Bézier curve Γ. Repeating this construction on smaller and smaller portion of the Bézier curve Γ gives a sequence of points on the Bézier curve Γ. To draw the Bézier curve Γ, one may draw straight lines between the points of Γ that have been found when the distance between them is smaller than a given small value. It is not suggested however to use this method to draw Bézier curves because of the large number of operations needed to draw the curve.

We first show that $\mathbf{f}$ is on the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{p}_{i+1}$ and control points $\check{\mathbf{q}}_i$, $\hat{\mathbf{q}}_{i+1}$. We have that

$$\begin{aligned}
\mathbf{f} &= \frac{1}{2}(\mathbf{d}+\mathbf{e}) = \frac{1}{2}\left(\frac{1}{2}(\mathbf{a}+\mathbf{c}) + \frac{1}{2}(\mathbf{c}+\mathbf{b})\right) = \frac{1}{4}\mathbf{a} + \frac{1}{2}\mathbf{c} + \frac{1}{4}\mathbf{b} \\
&= \frac{1}{4}\left(\frac{1}{2}(\mathbf{p}_i + \check{\mathbf{q}}_i)\right) + \frac{1}{2}\left(\frac{1}{2}(\check{\mathbf{q}}_i + \hat{\mathbf{q}}_{i+1})\right) + \frac{1}{4}\left(\frac{1}{2}(\hat{\mathbf{q}}_{i+1} + \mathbf{p}_{i+1})\right) \\
&= \frac{1}{8}(\mathbf{p}_i + 3\check{\mathbf{q}}_i + 3\hat{\mathbf{q}}_{i+1} + \mathbf{p}_{i+1}) = \phi_i\left(\frac{1}{2}\right) .
\end{aligned}$$

We now show that the first half of the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{p}_{i+1}$ and control points $\check{\mathbf{q}}_i$, $\hat{\mathbf{q}}_{i+1}$, namely $\phi_i(t)$ for $0 \le t \le 1/2$, is the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{f}$ and control points $\mathbf{a}$, $\mathbf{d}$. We leave to the reader the proof that the second half of the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{p}_{i+1}$ and control points $\check{\mathbf{q}}_i$, $\hat{\mathbf{q}}_{i+1}$, namely $\phi_i(t)$ for $1/2 \le t \le 1$, is the Bézier curve with endpoints $\mathbf{f}$, $\mathbf{p}_{i+1}$ and control points $\mathbf{e}$, $\mathbf{b}$.

The parametric representation of the Bézier curve with endpoints $\mathbf{p}_i$, $\mathbf{f}$ and control points $\mathbf{a}$, $\mathbf{d}$ is given by

$$\psi(s) = (1-s)^3\mathbf{p}_i + 3s(1-s)^2\mathbf{a} + 3s^2(1-s)\mathbf{d} + s^3\mathbf{f}$$

for $0 \le s \le 1$. Hence,

$$\begin{aligned}
\psi(s) &= (1-s)^3\mathbf{p}_i + 3s(1-s)^2\mathbf{a} + 3s^2(1-s)\mathbf{d} + s^3\frac{1}{2}(\mathbf{d}+\mathbf{e}) \\
&= (1-s)^3\mathbf{p}_i + 3s(1-s)^2\mathbf{a} + \left(3s^2(1-s) + \frac{1}{2}s^3\right)\mathbf{d} + \frac{1}{2}s^3\mathbf{e} \\
&= (1-s)^3\mathbf{p}_i + 3s(1-s)^2\mathbf{a} + \left(3s^2(1-s) + \frac{1}{2}s^3\right)\left(\frac{1}{2}(\mathbf{a}+\mathbf{c})\right) + \frac{1}{2}s^3\left(\frac{1}{2}(\mathbf{c}+\mathbf{b})\right) \\
&= (1-s)^3\mathbf{p}_i + \left(3s(1-s)^2 + \frac{3}{2}s^2(1-s) + \frac{1}{4}s^3\right)\mathbf{a} + \left(\frac{3}{2}s^2(1-s) + \frac{1}{2}s^3\right)\mathbf{c} + \frac{1}{4}s^3\mathbf{b} \\
&= (1-s)^3\mathbf{p}_i + \left(3s(1-s)^2 + \frac{3}{2}s^2(1-s) + \frac{1}{4}s^3\right)\left(\frac{1}{2}(\mathbf{p}_i + \check{\mathbf{q}}_i)\right) \\
&\quad + \left(\frac{3}{2}s^2(1-s) + \frac{1}{2}s^3\right)\left(\frac{1}{2}(\check{\mathbf{q}}_i + \hat{\mathbf{q}}_{i+1})\right) + \frac{1}{4}s^3\left(\frac{1}{2}(\mathbf{p}_{i+1} + \hat{\mathbf{q}}_{i+1})\right) \\
&= \left((1-s)^3 + \frac{3}{2}s(1-s)^2 + \frac{3}{4}s^2(1-s) + \frac{1}{8}s^3\right)\mathbf{p}_i \\
&\quad + \left(\frac{3}{2}s(1-s)^2 + \frac{3}{2}s^2(1-s) + \frac{3}{8}s^3\right)\check{\mathbf{q}}_i + \left(\frac{3}{4}s^2(1-s) + \frac{3}{8}s^3\right)\hat{\mathbf{q}}_{i+1} + \frac{1}{8}s^3\mathbf{p}_{i+1} \\
&= \left(\frac{s}{2} + (1-s)\right)^3\mathbf{p}_i + \frac{3s}{2}\left(\frac{s}{2} + (1-s)\right)^2\check{\mathbf{q}}_i + 3\left(\frac{s}{2}\right)^2\left(\frac{s}{2} + (1-s)\right)\hat{\mathbf{q}}_{i+1} + \left(\frac{s}{2}\right)^3\mathbf{p}_{i+1} \\
&= \left(1 - \frac{s}{2}\right)^3\mathbf{p}_i + \frac{3s}{2}\left(1 - \frac{s}{2}\right)^2\check{\mathbf{q}}_i + 3\left(\frac{s}{2}\right)^2\left(1 - \frac{s}{2}\right)\hat{\mathbf{q}}_{i+1} + \left(\frac{s}{2}\right)^3\mathbf{p}_{i+1} = \phi_i\left(\frac{s}{2}\right)
\end{aligned}$$

for $0 \le s \le 1$. ♠

Example 7.2.3

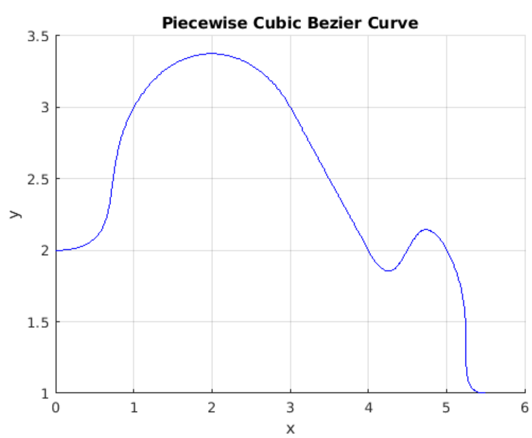
We want to construct a piecewise cubic Bézier curve that satisfy the following conditions.

i	$\mathbf{p}_i$	$\mathbf{p}_{i+1}$	$\check{\mathbf{q}}_i$	$\hat{\mathbf{q}}_{i+1}$
0	$(0,2)$	$(1,3)$	$(1,2)$	$(0.5,2.5)$
1	$(1,3)$	$(3,3)$	$(1.5,3.5)$	$(2.5,3.5)$
2	$(3,3)$	$(4,2)$	$(3.5,2.5)$	$(3.5,2.5)$
3	$(4,2)$	$(5,2)$	$(4.5,1.5)$	$(4.5,2.5)$
4	$(5,2)$	$(5.5,1)$	$(5.5,1.5)$	$(5,1)$

The pieces of the curve are given by

$$
\begin{aligned}
\phi_i(t) &= \left(-\mathbf{p}_i + 3\check{\mathbf{q}}_i - 3\hat{\mathbf{q}}_{i+1} + \mathbf{p}_{i+1}\right) t^3 + \left(3\mathbf{p}_i - 6\check{\mathbf{q}}_i + 3\hat{\mathbf{q}}_{i+1}\right) t^2 \\
&\qquad + \left(-3\mathbf{p}_i + 3\check{\mathbf{q}}_i\right) t + \mathbf{p}_i \\
&= \begin{cases}
\begin{pmatrix} 2.5 \\ -0.5 \end{pmatrix} t^3 + \begin{pmatrix} -4.5 \\ 1.5 \end{pmatrix} t^2 + \begin{pmatrix} 3.0 \\ 0.0 \end{pmatrix} t + \begin{pmatrix} 0.0 \\ 2.0 \end{pmatrix} & \text{if} \quad i = 0 \\
\begin{pmatrix} -1.0 \\ 0.0 \end{pmatrix} t^3 + \begin{pmatrix} 1.5 \\ -1.5 \end{pmatrix} t^2 + \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix} t + \begin{pmatrix} 1.0 \\ 3.0 \end{pmatrix} & \text{if} \quad i = 1 \\
\begin{pmatrix} 1.0 \\ -1.0 \end{pmatrix} t^3 + \begin{pmatrix} -1.5 \\ 1.5 \end{pmatrix} t^2 + \begin{pmatrix} 1.5 \\ -1.5 \end{pmatrix} t + \begin{pmatrix} 3.0 \\ 3.0 \end{pmatrix} & \text{if} \quad i = 2 \\
\begin{pmatrix} 1.0 \\ -3.0 \end{pmatrix} t^3 + \begin{pmatrix} -1.5 \\ 4.5 \end{pmatrix} t^2 + \begin{pmatrix} 1.5 \\ -1.5 \end{pmatrix} t + \begin{pmatrix} 4.0 \\ 2.0 \end{pmatrix} & \text{if} \quad i = 3 \\
\begin{pmatrix} 2.0 \\ 0.5 \end{pmatrix} t^3 + \begin{pmatrix} -3.0 \\ 0.0 \end{pmatrix} t^2 + \begin{pmatrix} 1.5 \\ -1.5 \end{pmatrix} t + \begin{pmatrix} 5.0 \\ 2.0 \end{pmatrix} & \text{if} \quad i = 4
\end{cases}
\end{aligned}
$$

for $0 \leq t \leq 1$. The graph of the piecewise cubic Bézier curve is given below.



♣

Remark 7.2.4

The **Bernstein polynomial** of degree $m \in \mathbb{N}^+$ for a function $f : [0,1] \to \mathbb{R}$ is the polynomial

$$
B_m(t; f) = \sum_{k=0}^{m} f\left(\frac{k}{m}\right) \binom{m}{k} t^k (1-t)^{m-k} . \tag{7.2.3}
$$

One can prove that $B_m(\cdot;f) \to f$ uniformely on $[0,1]$ as $m \to \infty$ if f is continuous on $[0,1]$. One of the proofs of the Stone-Weierstrass Theorem, Theorem 9.1.1, is effectively based on the Bernstein polynomials.

The Bézier curve (7.2.1) can be written as the Bernstein polynomial

$$\phi_i(t) = \sum_{k=0}^{3} \mathbf{c}_{i,k} \binom{3}{k} t^k (1-t)^{3-k}$$

for $0 \leq i < n$, where $\mathbf{c}_{i,k} \in \mathbb{R}^2$ satisfies

$$\binom{3}{k}\mathbf{c}_{i,k} = \begin{cases} \mathbf{p}_i & \text{if} \quad k=0 \\ \check{\mathbf{q}}_i & \text{if} \quad k=1 \\ \hat{\mathbf{q}}_{i+1} & \text{if} \quad k=2 \\ \mathbf{p}_{i+1} & \text{if} \quad k=3 \end{cases}$$

for $0 \leq i < n$.

Similarly, we may generalize Bézier curves to more than two control points. For each $m \geq 3$, we define the Bézier curve with $m-1$ control points as the curve defined by

$$\phi_i(t) = \sum_{k=0}^{m} \mathbf{c}_{i,k} \binom{m}{k} t^k (1-t)^{m-k} \ . \tag{7.2.4}$$

The control points are $\binom{m}{k}\mathbf{c}_{i,k}$ for $k = 1, 2, \ldots, m-1$. In particular, if we assume that

$$\binom{m}{k}\mathbf{c}_{i,k} = \begin{cases} \mathbf{p}_i & \text{if} \quad k=0 \\ \check{\mathbf{q}}_i & \text{if} \quad k=1 \\ \hat{\mathbf{q}}_{i+1} & \text{if} \quad k=m-1 \\ \mathbf{p}_{i+1} & \text{if} \quad k=m \end{cases}$$

then $\phi_i(0) = \mathbf{p}_i$ and $\phi_i(1) = \mathbf{p}_{i+1}$ for $0 \leq i < n$. Moreover, since

$$\phi_i'(t) = m \sum_{k=0}^{m-1} \left(\mathbf{c}_{i,k+1} - \mathbf{c}_{i,k}\right) \binom{m-1}{k} t^k (1-t)^{m-1-k} \ ,$$

we get

$$\phi_i'(0) = m\left(\mathbf{c}_{i,1} - \mathbf{c}_{i,0}\right) = m\left(\check{\mathbf{q}}_i - \mathbf{p}_i\right) = m(\alpha_i, \beta_i)$$

and

$$\phi_i'(1) = m\left(\mathbf{c}_{i,m} - \mathbf{c}_{i,m-1}\right) = m\left(\mathbf{p}_{i+1} - \hat{\mathbf{q}}_{i+1}\right) = m(\alpha_{i+1}, \beta_{i+1}) \ .$$

We still get a curve which is tangent to (α_i, β_i) at $\mathbf{p}_i$ and tangent to $(\alpha_{i+1}, \beta_{i+1})$ at $\mathbf{p}_{i+1}$. ♠

7.3 B-Spline Interpolation

In this section, we consider an infinite sequence of knots

$$\ldots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \ldots$$

such that $\lim\limits_{i\to-\infty} t_i = -\infty$ and $\lim\limits_{i\to\infty} t_i = +\infty$.

Definition 7.3.1

The **B-splines of degree** 0 are defined by

$$B_i^0(t) = \begin{cases} 1 & \text{if} \quad t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \mathbb{Z}$.
The **B-splines of degree** $k > 0$ are defined by the recurrence relation

$$B_i^k(t) = v_i^k(t) B_i^{k-1}(t) + \left(1 - v_{i+1}^k(t)\right) B_{i+1}^{k-1}(t) \tag{7.3.1}$$

for $i \in \mathbb{Z}$, where $v_i^k(t) = \dfrac{t - t_i}{t_{i+k} - t_i}$.

We sketch in Figure 7.3 a B-spline of degree 0 and a B-spline of degree 1.

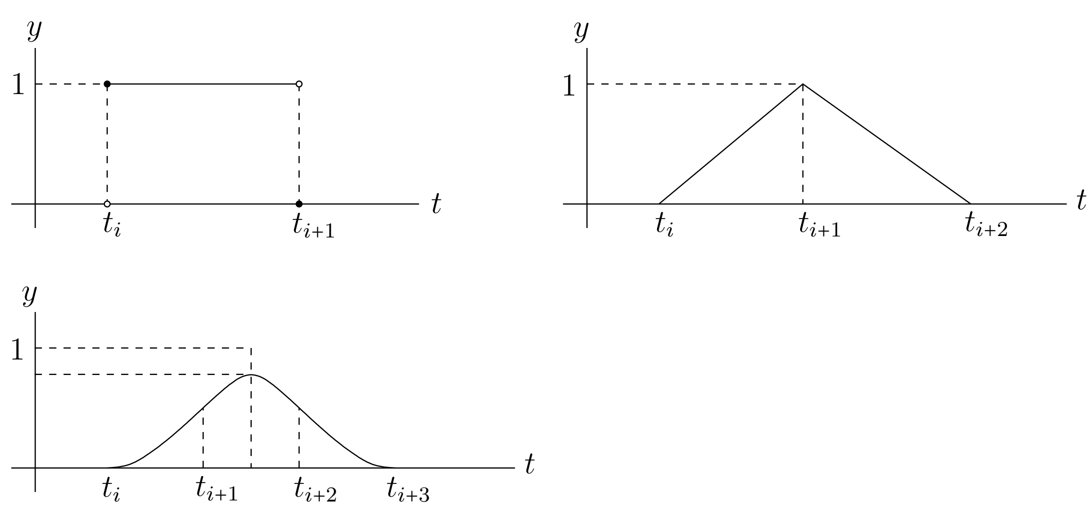


Figure 7.3: Clockwise from the top left corner: B_i^0, B_i^1 and B_i^2.

The next propositions state some of the properties of the B-splines. We give some of the proofs and refer the reader to [21] for the missing proofs and more information.

Proposition 7.3.2

The B-splines of degree 0 are piecewise constant functions which are continuous from the right. For $k > 0$, the B-splines of degree k are piecewise polynomials of degree k and class C^{k-1}.

That the B-splines of degree $k > 0$ are piecewise polynomials of degree k is proved by induction using (7.3.1). To proof that they are of class C^{k-1} requires induction and tedious computations to show that

$$\frac{\mathrm{d}}{\mathrm{d}t} B_i^k(t) = \left(\frac{k}{t_{i+k} - t_i}\right) B_i^{k-1}(t) - \left(\frac{k}{t_{i+k+1} - t_{i+1}}\right) B_{i+1}^{k-1}(t) \tag{7.3.2}$$

for $k > 1$. This formula is also true for $k = 1$ as long as $t \neq t_j$ for $j = i$, $i+1$ and $i+2$.

A simple proof by induction based on (7.3.1) gives the following result.

Proposition 7.3.3

$B_i^0(t) > 0$ for $t_i \leq t < t_{i+1}$ and $B_i^0(t) = 0$ otherwise. For $k > 0$, $B_i^k(t) > 0$ for $t_i < t < t_{i+k+1}$ and $B_i^k(t) = 0$ otherwise.

The next result is quite useful to evaluate B-splines.

Proposition 7.3.4

Suppose that

$$p(t) = \sum_{i=-\infty}^{\infty} C_i^k(t) B_i^k(t) \quad .$$

Given $t \in [t_j, t_{j+1}[$, if we use the relation

$$\begin{aligned} C_i^{r-1}(t) &= C_i^r(t) v_i^r(t) + C_{i-1}^r \left(1 - v_i^r(t)\right) \\ &= \frac{1}{t_{r+i} - t_i} \left((t - t_i) C_i^r(t) + (t_{r+i} - t) C_{i-1}^r(t)\right) \end{aligned} \tag{7.3.3}$$

for $r = k, k-1, \ldots, 0$ to generate the table

$$\begin{array}{ccccc} C_j^k(t) & C_j^{k-1}(t) & \ldots & C_j^1(t) & C_j^0(t) \\ C_{j-1}^k(t) & C_{j-1}^{k-1}(t) & \ldots & C_{j-1}^1(t) & \\ \vdots & \vdots & ⋰ & & \\ C_{j-k+1}^k(t) & C_{j-k+1}^{k-1}(t) & & & \\ C_{j-k}^k(t) & & & & \end{array}$$

then $p(t) = C_j^0(t)$.

Proof.

The proof of the previous proposition is based on the relation

$$\begin{aligned}\sum_{i=-\infty}^{\infty} C_i^r(t)B_i^r(t) &= \sum_{i=-\infty}^{\infty} C_i^r(t)\left(v_i^r(t)B_i^{r-1}(t) + \left(1-v_{i+1}^r(t)\right)B_{i+1}^{r-1}\right)\\
&= \sum_{i=-\infty}^{\infty} C_i^r(t)v_i^r(t)B_i^{r-1}(t) + \sum_{i=-\infty}^{\infty} C_i^r(t)\left(1-v_{i+1}^r(t)\right)B_{i+1}^{r-1}\\
&= \sum_{i=-\infty}^{\infty} C_i^r(t)v_i^r(t)B_i^{r-1}(t) + \sum_{i=-\infty}^{\infty} C_{i-1}^r(t)\left(1-v_i^r(t)\right)B_i^{r-1}\\
&= \sum_{i=-\infty}^{\infty} \left(C_i^r(t)v_i^r(t) + C_{i-1}^r(t)\left(1-v_i^r(t)\right)\right)B_i^{r-1}\\
&= \sum_{i=-\infty}^{\infty} C_i^{r-1}(t)B_i^{r-1}(t)\end{aligned}$$

and a simple proof by induction to get

$$p(t) = \sum_{i=-\infty}^{\infty} C_i^k(t)B_i^k(t) = \sum_{i=-\infty}^{\infty} C_i^0(t)B_i^0(t) \ .$$

Don't forget that, for t given, all sums are finite. ∎

Remark 7.3.5
The spline interpolant that we will present later will be of the form

$$p(t) = \sum_{i=-\infty}^{\infty} c_i^k B_i^k(t)$$

for some constants c_i^k. If we set $C_i^r(t) = c_i^r$, we can use the method presented in the proposition above to compute $p(t)$. ♠

Proposition 7.3.6

$\displaystyle\sum_{i=-\infty}^{+\infty} B_i^k(t) = 1$ for all $t \in \mathbb{R}$ and $k \geq 0$.

Proof.
We use the previous proposition with $C_i^k(t) = 1$ for all i. We have

$$C_i^{k-1}(t) = C_i^k(t)v_i^k(t) + C_{i-1}^k\left(1-v_i^k(t)\right) = v_i^k(t) + \left(1-v_i^k(t)\right) = 1$$

for all i. A simple proof by induction shows that $C_i^r(t) = 1$ for all i and all r with $0 \leq r \leq k$. Thus,

$$\sum_{i=-\infty}^{\infty} B_i^k(t) = \sum_{i=-\infty}^{\infty} B_i^0(t) \ .$$

For $t \in [t_j, t_{j+1}[$, we get

$$\sum_{i=-\infty}^{\infty} B_i^k(t) = \sum_{i=-\infty}^{\infty} B_i^0(t) = B_j^0(t) = 1 \ .$$ ∎

Proposition 7.3.7

The set $\left\{B_j^k, B_{j+1}^k, \ldots, B_{j+k}^k\right\}$ is linearly independent on $]t_{j+k}, t_{j+k+1}[$.

We note that the only B-splines B_i^k that are not trivially null on $]t_{j+k}, t_{j+k+1}[$ are those for $j \leq i \leq j+k$. The proof of this proposition is by induction and requires the formula for the derivative of the B-splines B_i^k that was required for the proof of Proposition 7.3.2.

Proposition 7.3.8

The set of B-splines $\left\{B_{-k}^k, B_{-k+1}^k, \ldots, B_{n-1}^k\right\}$ is a basis for the space S_n^k of functions p of class C^{k-1} on $[t_0, t_n]$ such that $p\big|_{[t_i,t_{i+1}]}$ is a polynomials of degree at most k for $0 \leq i < n$.

Proof.
We note that the only B-splines B_i^k that are not trivially null on $]t_0, t_n[$ are those for $-k \leq i \leq n-1$.

Linear Independence: Suppose that $\displaystyle\sum_{i=-k}^{n-1} c_i B_i^k = 0$ on $[t_0, t_n]$. We therefore also have that

$$\sum_{i=-k}^{0} c_i B_i^k = \sum_{i=-k}^{n-1} c_i B_i^k = 0$$

on $]t_0, t_1[$. It follows from the previous proposition that $c_i = 0$ for $-k \leq i \leq 0$.

Suppose that $j < n$ is the smallest index such that $c_j \neq 0$. From the previous discussion, we have $j > 0$. Hence, for $t \in [t_j, t_j + 1[$, we have

$$0 = \sum_{i=-k}^{n-1} c_i B_i^k(t) = \sum_{i=j}^{n-1} c_i B_i^k(t) = c_j B_j^k(t) \ .$$

Since $B_j^k(t) > 0$, we get $c_j = 0$. This is a contradiction that $c_j \neq 0$. So, there is no such j between 0 and n such $c_j \neq 0$.

In particular, this proves that S_n^k is at least of dimension $n+k$.

Generating set: We prove that all functions $p \in S_n^k$ can be expressed as a linear combination over $\mathbb{R}$ of the following $n+k$ functions in S_n^k: t^j for $0 \leq j \leq k$ and $H(t-t_j)(t-t_j)^k$ for $1 \leq j \leq n-1$, where H is the Heavyside function defined by

$$H(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases}$$

This will prove that S_n^k is at most of dimension $n+k$. Combined with what we proved in the first part of the proof, this shows that S_n^k is of dimension $n+k$ and that $\left\{B_{-k}^k, B_{-k+1}^k, \ldots, B_{n-1}^k\right\}$ is a basis of S_n^k.

Given $p \in S_n^k$, we have that $p_0 = p\big|_{[t_0,t_1]}$ is a polynomial of degree at most k. So $p_0(t) = \sum_{i=0}^{k} a_i t^i$ for some $a_i \in \mathbb{R}$.

We prove by induction that there exist constants a_{k+i} such that

$$p(t) = \sum_{i=0}^{k} a_i t^i + \sum_{i=1}^{j} a_{k+i} H(t-t_i)(t-t_i)^k \tag{7.3.4}$$

for $t_0 \le t \le t_{j+1}$ with $1 \le j \ < n$.

We have that $P_1 = p\big|_{[t_1,t_2]}$ is a polynomial of degree at most k. Since p is of class C^{k-1}, we have that

$$\frac{\mathrm{d}^m}{\mathrm{d}t^m}(p_1 - p_0)(t_1) = 0$$

for $0 \le m < k$. Since $p_1 - p_0$ is a polynomial of degree at most k, it follows from Lemma 6.3.2 that $(p_1 - p_0)(t) = a_{k+1}(t-t_1)^k$ for some $a_{k+1} \in \mathbb{R}$. Thus

$$p(t) = \sum_{i=0}^{k} a_i t^i + a_{k+1} H(t-t_1)(t-t_1)^k$$

for $t_0 \le t \le t_2$. This proves (7.3.4) for $j = 1$.

Suppose that (7.3.4) is true for j. We have that $p_{j+1} = p\big|_{[t_{j+1},t_{j+2}]}$ is a polynomial of degree at most k. Moreover,

$$p_j = p\big|_{[t_j,t_{j+1}]} = \left(\sum_{i=0}^{k} a_i t^i + \sum_{i=1}^{j} a_{k+i} H(t-t_i)(t-t_i)^k \right)\Bigg|_{[t_j,t_{j+1}]}$$

is a polynomial of degree at most k. Since p is of class C^{k-1}, we have that

$$\frac{\mathrm{d}^m}{\mathrm{d}t^m}(p_{j+1} - p_j)(t_{j+1}) = 0$$

for $0 \le m < k$. Since $p_{j+1} - p_j$ is a polynomial of degree at most k, it again follows from Lemma 6.3.2 that $(p_{j+1} - p_j)(t) = a_{k+j+1}(t - t_{j+1})^k$ for some $a_{k+j+1} \in \mathbb{R}$. thus

$$p(t) = \sum_{i=0}^{k} a_i t^i + \sum_{i=1}^{j+1} a_{k+i} H(t-t_i)(t-t_i)^k$$

for $t_0 \le t \le t_{j+2}$. This proves (7.3.4) for j replaced by $j+1$.

(7.3.4) with $j = n-1$ shows that f is a linear combination of the $n+k$ functions t^j for $0 \le j \le k$ and $H(t-t_j)(t-t_j)^k$ for $1 \le j \le n-1$ as claimed. ∎

Our interpolation problem is as follows. Given points (x_1, y_1), (x_2, y_2), $\ldots$, (x_{n+k}, y_{n+k}) such that $x_j < x_{j+1}$ for $1 \le j < n+k$, and $x_j \in [t_0, t_n]$ for $1 \le j \le n+k$, can we find constants c_i with $-k \le i \le n-1$ such that

$$\sum_{i=-k}^{n-1} c_i B_i^k(x_j) = y_j \tag{7.3.5}$$

for $1 \leq j \leq n+k$? If such c_i exist, then

$$p(x) = \sum_{i=-k}^{n-1} c_i B_i^k(x) \tag{7.3.6}$$

is a **spline interpolant** on the nodes x_1, x_2, ..., x_{n+k}.

The answer to this question is a consequence of the following result.

Theorem 7.3.9 (Schoemberg-Whitney)

Given $q \in \mathbb{Z}$ and $x_1 < x_2 < \ldots < x_m$, consider the $m \times m$ matrix Q with the entries $Q_{j,i} = B_{i+q}^k(x_j)$ for $1 \leq i,j \leq m$. Then, Q is invertible if and only if $Q_{j,j} \neq 0$ for $1 \leq j \leq m$.

To find the c_i^k required to satisfy (7.3.5), we have to solve the linear system $Q\mathbf{z} = \mathbf{y}$, where $Q_{j,i} = B_{i-k-1}^k(x_j)$ and $z_i = c_{i-k-1}$ for $1 \leq i,j \leq n+k$. We get the following result from Schoemberg-Whitney Theorem and Proposition 7.3.3.

Proposition 7.3.10

The system $Q\mathbf{z} = \mathbf{y}$ defined above has a solution if and only if $Q_{j,j} = B_{j-k-1}^k(x_j) \neq 0$ for $1 \leq j \leq n+k$; namely, if $t_{j-k-1} < x_j < t_j$ for $1 \leq j \leq n+k$.

If we consider the set of B-splines $\left\{B_{-k}^k, B_{-k+1}^k, \ldots, B_{n-1}^k\right\}$, then Schoemberg-Whitney Theorem not only gives a condition for the existence of a solution to $Q\mathbf{z} = \mathbf{y}$ but it also shows that this solution is unique. However, there is no obligation to specify all the $n+k$ points (x_1, y_1), (x_2, y_2), ..., (x_{n+k}, y_{n+k}). Namely, we do not have to use all the equations (7.3.5) for $1 \leq j \leq n+k$ to determine the c_i. We may use other conditions to determine some of the c_i. We will do just that in an example below.

Remark 7.3.11

We will need the following information for the next example. Let

$$p(t) = \sum_{-\infty}^{\infty} c_i B_i^k(t) \ .$$

If we derive f using (7.3.2), we get

$$p'(t) = k \sum_{-\infty}^{\infty} \left(\frac{c_i - c_{i-1}}{t_{i+k} - t_i}\right) B_i^{k-1}(t)$$

for $k > 1$. This formula is also true for $k = 1$ as long as $t \neq t_i$ for all i. Again, if we derive f' using (7.3.2), we get

$$p''(t) = k(k-1) \sum_{-\infty}^{\infty} \left(\frac{1}{t_{i+k-1} - t_i}\right)\left(\frac{c_i - c_{i-1}}{t_{i+k} - t_i} - \frac{c_{i-1} - c_{i-2}}{t_{i+k-1} - t_{i-1}}\right) B_i^{k-2}(t)$$

for $k > 2$. This formula is also true for $k = 2$ as long as $t \neq t_i$ for all i. ♠

Example 7.3.12
Suppose that $f : \mathbb{R} \to \mathbb{R}$. We will give a natural cubic interpolant of f on the nodes $x_1 < x_2 < \ldots < x_n$.

We have $k = 3$ and we select the knots t_i such that $x_j = t_{j-1}$ for $1 \leq j \leq n$. The spline interpolant p that we are looking for will be determined by the points $(x_j, y_j) = (t_{j-1}, f(t_{j-1}))$ for $1 \leq j \leq n$. We need $j - 4 \leq i < j$ to possibly have that $B_i^3(x_j)$ is non null.

The spline interpolant p is of the form

$$p(x) = \sum_{i=-3}^{n-2} c_i B_i^3(x) \ ,$$

where

$$p(x_j) = \sum_{i=-3}^{n-2} c_i B_i^3(x_j) = \sum_{i=j-3}^{j-2} c_i B_i^3(x_j) = y_j \tag{7.3.7}$$

for $1 \leq j \leq n$. We have dropped the term for $i \notin \{-3, -2, \ldots, n-2\}$ from the summation because $B_i^3(x_j) = B_i^3(t_{j-1}) = 0$ for these values of i. There are n equations with $n+2$ variables c_i for $-3 \leq i \leq n-2$. We use the two extra variables to satisfy the conditions for a natural cubic spline interpolant; namely, $p''(x_1) = p''(x_n) = 0$.

From the result stated in the previous remark, we have

$$\begin{aligned}
p''(x_1) = p''(t_0) &= 6 \sum_{-\infty}^{\infty} \left(\frac{1}{t_{i+2} - t_i} \right) \left(\frac{c_i - c_{i-1}}{t_{i+3} - t_i} - \frac{c_{i-1} - c_{i-2}}{t_{i+2} - t_{i-1}} \right) B_i^1(t_0) \\
&= 6 \left(\frac{1}{t_1 - t_{-1}} \right) \left(\frac{c_{-1} - c_{-2}}{t_2 - t_{-1}} - \frac{c_{-2} - c_{-3}}{t_1 - t_{-2}} \right) = 0
\end{aligned}$$

and

$$p''(x_n) = p''(t_{n-1}) = 6 \left(\frac{1}{t_n - t_{n-2}} \right) \left(\frac{c_{n-2} - c_{n-3}}{t_{n+1} - t_{n-2}} - \frac{c_{n-3} - c_{n-4}}{t_n - t_{n-3}} \right) = 0 \ .$$

These give two extra equations,

$$(t_1 - t_{-2})c_{-1} - (t_2 + t_1 - t_{-1} - t_{-2})c_{-2} + (t_2 - t_{-1})c_{-3} = 0$$

and

$$(t_n - t_{n-3})c_{n-2} - (t_{n+1} + t_n - t_{n-2} - t_{n-3})c_{n-3} + (t_{n+1} - t_{n-2})c_{n-4} = 0 \ ,$$

to combine with the n equations in (7.3.7) to determine the $n+2$ variables c_i^3 for $-3 \leq i \leq n-2$. ♣

Consider a function $f : I \to \mathbb{R}$, where I is a sub-interval of $\mathbb{R}$, and $\delta > 0$. The **modulus of continuity** of f on I is defined by

$$\omega(f; \delta, I) = \sup\{|f(x) - f(y)| : x, y \in I \text{ and } |x - y| \leq \delta\} \quad .$$

For a uniformly continuous function f on I, we can have $\omega(f; \delta, I)$ as small as we want by taking δ small enough.

Theorem 7.3.13

Let $q(t) = \sum_{i=-\infty}^{\infty} f(t_{i+2})B_i^k(t)$ for $t \in \mathbb{R}$ and $k \geq 2$. If $f : [t_{-k}, t_{n+1}] \to \mathbb{R}$, then

$$\sup_{t_0 \leq t \leq t_n} |f(t) - q(t)| \leq k\omega(f; \delta, [t_{-k}, t_{n+1}])$$

for $\delta = \max_{-k \leq i \leq n+1} |t_i - t_{i-1}|$.

Proof.
Using Propositions 7.3.3 and 7.3.6, we may write

$$\begin{aligned}|f(t) - q(t)| &= \left| f(t) \sum_{i=-\infty}^{\infty} B_i^k(t) - \sum_{i=-\infty}^{\infty} f(t_{i+2})B_i^k(t) \right| = \left| \sum_{i=-\infty}^{\infty} \left(f(t) - f(t_{i+2})\right) B_i^k(t) \right| \\ &\leq \sum_{i=-\infty}^{\infty} |f(t) - f(t_{i+2})| \, B_i^k(t) = \sum_{i=j-k}^{j} |f(t) - f(t_{i+2})| \, B_i^k(t)\end{aligned}$$

for $t \in [t_j, t_{j+1}]$ and $0 \leq j \leq n-1$. Hence,

$$|f(t) - q(t)| \leq \max_{j-k \leq i \leq j} |f(t) - f(t_{i+2})| \underbrace{\sum_{i=j-k}^{j} B_i^k(t)}_{\leq 1} \leq \max_{j-k \leq i \leq j} |f(t) - f(t_{i+2})|$$

for $t \in [t_j, t_{j+1}]$. For $i = j$, we have

$$|f(t) - f(t_{i+2})| = |f(t) - f(t_{j+2})| \leq |f(t) - f(t_{j+1})| + |f(t_{j+1}) - f(t_{j+2})| \leq 2\omega(f, \delta, [t_{-k}, t_{n+1}])$$

for $t \in [t_j, t_{j+1}]$. For $i = j-1$, we have

$$|f(t) - f(t_{i+2})| = |f(t) - f(t_{j+1})| \leq \omega(f, \delta, [t_{-k}, t_{n+1}])$$

for $t \in [t_j, t_{j+1}]$. For $i = j-2$, we have

$$|f(t) - f(t_{i+2})| = |f(t) - f(t_j)| \leq \omega(f, \delta, [t_{-k}, t_{n+1}])$$

for $t \in [t_j, t_{j+1}]$. For $i = j-s$ with $3 \leq s \leq k$, we have

$$\begin{aligned}&|f(t) - f(t_{i+2})| = |f(t) - f(t_{j-s+2})| \\ &\quad \leq |f(t) - f(t_j)| + |f(t_j) - f(t_{j-1})| + \ldots + |f(t_{j-s+3}) - f(t_{j-s+2})| \leq (s-1)\omega(f, \delta, [t_{-k}, t_{n-1}])\end{aligned}$$

for $t \in [t_j, t_{j+1}]$. In all cases, we have $|f(t) - f(t_{i+2})| \leq k\omega(f, \delta, [t_{-k}, t_{n-1}])$ for $t \in [t_j, t_{j+1}]$. The conclusion of the theorem follows since this is true for all j such that $0 \leq j < n$.

Note: As the proof shows, we could have used only the interval $[t_{-k+2}, t_{n+1}]$ instead of $[t_{-k}, t_{n+1}]$ in the statement of the theorem. We have used the second one because the statement was nicer. ∎

Since every element of S_n^k is a linear combination of B_j^k for $-k \leq j < n$, we get the following result from the previous theorem.

Corollary 7.3.14

We have that
$$\operatorname{dist}\left(f, S_n^k\right) \leq k\omega(f;\delta,[t_{-k},t_{n+1}])$$
for all $f : [t_{-k}, t_{n+1}] \to \mathbb{R}$.

If f is continuous on $[t_{-k}, t_{n-1}]$, and so uniformly continuous on $[t_{-k}, t_{n-1}]$, we have that $\omega(f;\delta,[t_{-k},t_{n+1}] \to 0$ as $\delta \to 0$. Therefore, to theoretically improve the accuracy of the interpolation of a function f on a given interval, we may increase the number of knots t_i in the interval (increase n) while decreasing the distance between them (decreasing δ).

7.4 Other Spline Methods

Consider $n+1$ points $\mathbf{p}_i = (x_i, y_i)$ for $i = 0, 1, \ldots, n$. We now define a piecewise polynomial, parametric representation of a curve that shadows the points $\mathbf{p}_i$ but does not include the points $\mathbf{p}_i$.

First, we add the points $\mathbf{p}_{-2} = \mathbf{p}_{-1} = \mathbf{p}_0$ and $\mathbf{p}_{n+2} = \mathbf{p}_{n+1} = \mathbf{p}_n$. There are other approaches to handle the end points $\mathbf{p}_0$ and $\mathbf{p}_n$. With this approach, $\mathbf{p}_0$ and $\mathbf{p}_n$ are on the spline.

For $-1 \leq i \leq n$, we define the curve with the parametric representation
$$\phi_i(t) = \sum_{j=-1}^{2} b_j(t)\mathbf{p}_{i+j} \ , \tag{7.4.1}$$
where $b_{-1}(t) = -\dfrac{t^3}{6} + \dfrac{t^2}{2} - \dfrac{t}{2} + \dfrac{1}{6}$, $b_0(t) = \dfrac{t^3}{2} - t^2 + \dfrac{2}{3}$, $b_1(t) = -\dfrac{t^3}{2} + \dfrac{t^2}{2} + \dfrac{t}{2} + \dfrac{1}{6}$ and $b_2(t) = \dfrac{t^3}{6}$ for $0 \leq t \leq 1$. Each component of the parametric representation is a polynomial of degree three in t.

The small curve defined by the parametric representation $\phi_i(t)$ with $0 \leq t \leq 1$ is in the convex hull of the points $\mathbf{p}_j$ for $j = i-1$, i, $i+1$ and $i+2$. The coordinates of $\phi_i(t)$ are the weighted sums of the coordinates of $\mathbf{p}_j$ for $j = i-1$, i, $i+1$ and $i+2$ because $\displaystyle\sum_{j=-1}^{2} b_j(t) = 1$ for all t.

The parametric representations $\phi_i(t)$ satisfy the following properties:
$$\phi_i(1) = \phi_{i+1}(0) \ , \quad \phi_i'(1) = \phi_{i+1}'(0) \quad \text{and} \quad \phi_i''(1) = \phi_{i+1}''(0)$$
for $i = -1, 0, 1, \ldots, n$.

The parametric representation $\phi_i(t)$ given in (7.4.1) can be rewritten as
$$\begin{aligned}\phi_i(t) &= \frac{1}{6}\left(-\mathbf{p}_{i-1} + 3\mathbf{p}_i - 3\mathbf{p}_{i+1} + \mathbf{p}_{i+2}\right)t^3 + \frac{1}{6}\left(3\mathbf{p}_{i-1} - 6\mathbf{p}_i + 3\mathbf{p}_{i+1}\right)t^2 \\ &\quad + \frac{1}{6}\left(-3\mathbf{p}_{i-1} + 3\mathbf{p}_{i+1}\right)t + \frac{1}{6}\left(\mathbf{p}_{i-1} + 4\mathbf{p}_i + \mathbf{p}_{i+1}\right) \ .\end{aligned} \tag{7.4.2}$$

Note the resemblance between (7.4.2) and the definition of Bézier curves in Section 7.2.

7.5 Exercises

Question 7.1
Construct the clamped cubic spline interpolant to f associated to the data of the following table.

x	0	1	3	4	5	5.5
$f(x)$	2	3	3	2	2	1
$f'(x)$	0					−2

Plot the graph of this cubic spline for $0 \le x \le 5.5$.

Question 7.2
Write a code similar to Code 7.1.5 for the natural cubic spline interpolation and use it to draw the natural cubic spline interpolant to f associated to the data of the following table.

x	0	1	3	4	5	5.5
$f(x)$	2	3	3	2	2	1

Chapter 8

Least Square Approximation (in L^2)

To understand the foundation of least square approximation, we first need to briefly review L^2 spaces. The reader will notice many similarities with linear algebra in $\mathbb{R}^n$.

8.1 L^2 spaces

Readers who have not studied measure theory and functional analysis before may skip the review of the theory and only read the examples in this sections.

Suppose that μ is a measure on a measurable space Ω. Let $L^2(\Omega)$ be the space of measurable functions $f : \Omega \to \mathbb{C}$ such that $\displaystyle \int_\Omega f^2 \,\mathrm{d}\mu$ is finite. We can define a scalar product on $L^2(\Omega)$ by

$$\langle f, g \rangle = \int_\Omega f\,\overline{g}\,\mathrm{d}\mu \quad , \quad f, g \in L^2(\Omega) \ .$$

The associated L^2-norm is

$$\|f\|_2 = \sqrt{\langle f, f \rangle} = \left(\int_\Omega |f|^2 \,\mathrm{d}\mu \right)^{1/2} \quad , \quad f \in L^2(\Omega) \ .$$

Equipped with this norm, $L^2(\Omega)$ is a **Hilbert space**.

Definition 8.1.1

A set of functions $\{\phi_\alpha\}_{\alpha \in A} \subset L^2(\Omega)$, where A is some index set, is **linearly independent** if, for any finite subset $\{\alpha_i\}_{i=1}^n \subset A$, $\displaystyle \sum_{i=1}^n c_i \phi_{\alpha_i} = 0$ with $c_i \in \mathbb{C}$ implies that $c_1 = c_2 = \ldots = c_n = 0$.

Definition 8.1.2

A set of functions $S = \{\phi_\alpha\}_{\alpha\in A} \subset L^2(\Omega)$, where A is some index set, is **orthonormal** if

$$\langle \phi_{\alpha_1}, \phi_{\alpha_2} \rangle = \begin{cases} 0 & \text{if} \quad \alpha_1 \neq \alpha_2 \\ 1 & \text{if} \quad \alpha_1 = \alpha_2 \end{cases}$$

If instead of $\langle \phi_{\alpha_1}, \phi_{\alpha_2} \rangle = 1$ for $\alpha_1 \neq \alpha_2$, we have $\langle \phi_{\alpha_1}, \phi_{\alpha_2} \rangle \neq 0$ for $\alpha_1 = \alpha_2$, we say that S is an **orthogonal set**.

Orthogonal sets (and so orthonormal sets) are linear independent.

Definition 8.1.3

A **complete orthonormal set** or **orthonormal basis** is an orthonormal set $S = \{\phi_\alpha : \alpha \in A\} \subset L^2(\Omega)$, where A is some index set, such that the set of all finite linear combinations of elements of S is dense in $L^2(\Omega)$. If we replace "orthonormal" by "orthogonal" in the previous sentence, we get the definition of a **complete orthogonal set** and **orthogonal basis**.

It is proved in Functional Analysis that an orthogonal (or orthonormal) set of functions $\{\phi_\alpha : \alpha \in A\}$, where A is some index set, is complete if $\langle f, \phi_\alpha \rangle = 0$ for all $\alpha \in A$ implies that $f = 0$ almost everywhere on Ω.

Definition 8.1.4

Let $S = \{\phi_\alpha : \alpha \in A\}$, where A is some index set, be a orthonormal basis of $L^2(\Omega)$. The **Fourier series** of a function $f \in L^2(\Omega)$ with respect to the orthonormal basis S is

$$f \sim \sum_{\alpha\in A} a_\alpha \phi_\alpha \ ,$$

where

$$a_\alpha = \langle f, \phi_\alpha \rangle = \int_\Omega f\, \overline{\phi_\alpha}\, \mathrm{d}\mu \quad , \quad \alpha \in A \ .$$

If S is only an orthogonal basis, then the **Fourier series** of a function $f \in L^2(\Omega)$ with respect to the orthogonal basis S is

$$f \sim \sum_{\alpha\in A} a_\alpha \phi_\alpha \ ,$$

where

$$a_\alpha = \frac{\langle f, \phi_\alpha \rangle}{\langle \phi_\alpha, \phi_\alpha \rangle} \quad , \quad \alpha \in A \ .$$

It is proved in Functional Analysis that $a_\alpha \neq 0$ for at most a countable number of indices. Moreover,

$$\left(\int_\Omega \left| f - \sum_{j=1}^{J} a_{\alpha_j} \phi_{\alpha_j} \right|^2 \mathrm{d}\mu \right)^{1/2} \to 0 \quad \text{as} \quad J \to \infty$$

whatever the ordering $\{\alpha_j\}_{j=0}^{\infty}$ of the indices $\alpha \in A$ such that $a_\alpha \neq 0$.

The following result gives the theoretical justification to the method of least square approximation of functions that we will present later.

Theorem 8.1.5

Let $S = \{\phi_j : 1 \leq j \leq J\}$ be a finite orthonormal subset of $L^2[a,b]$. Given $f \in L^2[a,b]$, we have

$$\left\| f - \sum_{j=1}^{J} \langle f, \phi_j \rangle \, \phi_j \right\|_2 \leq \left\| f - \sum_{j=1}^{J} \lambda_j \phi_j \right\|_2$$

for all $\lambda_j \in \mathbb{C}$, and

$$\left\| f - \sum_{j=1}^{J} \langle f, \phi_j \rangle \, \phi_j \right\|_2 = \left\| f - \sum_{j=1}^{J} \lambda_j \phi_j \right\|_2$$

if and only if $\lambda_j = \langle f, \phi_j \rangle$ for all j.

Proof.
We have

$$\begin{aligned}
\left\| f - \sum_{j=1}^{J} \lambda_j \phi_j \right\|^2 &= \left\langle f - \sum_{j=1}^{J} \lambda_j \phi_j , f - \sum_{j=1}^{J} \lambda_j \phi_j \right\rangle = \langle f, f\rangle - \sum_{j=1}^{J} \overline{\lambda_i} \, \langle f, \phi_j \rangle - \sum_{j=1}^{J} \lambda_j \, \langle \phi, f \rangle + \sum_{j=1}^{J} |\lambda_j|^2 \\
&= \langle f, f\rangle - \sum_{j=1}^{J} \overline{\lambda_j} \, \langle f, \phi_j \rangle - \sum_{j=1}^{J} \lambda_j \overline{\langle f, \phi \rangle} + \sum_{j=1}^{J} |\lambda_j|^2 \\
&= \langle f, f\rangle + \underbrace{\sum_{j=1}^{J} |\lambda_j - \langle f, \phi_j \rangle|^2}_{\geq 0} - \sum_{j=1}^{J} |\langle \phi_j, f \rangle|^2 \ .
\end{aligned}$$

Hence,

$$\left\| f - \sum_{j=1}^{J} \lambda_i \phi_j \right\|^2 \geq \langle f, f\rangle - \sum_{j=1}^{J} |\langle \phi_j, f \rangle|^2$$

for all λ_j with $j = 1, 2, \ldots, J$. We have equality if and only if $\lambda_j = \langle f, \phi_j \rangle$ for $j = 1, 2, \ldots, J$. ∎

We give an example of how this theorem can be used, another example will be given in the next section.

Example 8.1.6
Let $\phi_n(x) = e^{nxi}$ for $n \in \mathbb{Z}$, where i is the complex number satisfying $i^2 = -1$. Since

$$\begin{aligned}
\int_{-\pi}^{\pi} \phi_k(x) \overline{\phi_j(x)} \, \mathrm{d}x &= \int_{-\pi}^{\pi} e^{kxi} e^{-jxi} \, \mathrm{d}x = \int_{-\pi}^{\pi} e^{(k-j)xi} \, \mathrm{d}x \\
&= \begin{cases} \dfrac{1}{(k-j)i} e^{(k-j)xi} \Big|_{x=-\pi}^{\pi} = \dfrac{1}{(k-j)i} \left(e^{(k-j)\pi i} - e^{-(k-j)\pi i} \right) = 0 & \text{if} \quad k \neq j \\ x \Big|_{x=-\pi}^{\pi} = 2\pi & \text{if} \quad k = j \end{cases}
\end{aligned}$$

The set $S = \{e^{nxi} : n \in \mathbb{Z}\}$ is an orthogonal set in the space $L^2[-\pi,\pi]$ with the Lebesgue measure. If we replace ϕ_n by $(2\pi)^{-1/2}\phi_n$, we get an orthonormal set. It can be shown that the set of all finite linear combinations of elements of S is dense in $L^2[-\pi,\pi]$. Hence, S is a complete orthogonal set in $L^2[-\pi,\pi]$.

For $f \in L^2[-\pi,\pi]$, we have that

$$\left(\int_{-\pi}^{\pi}\left(f(x) - \sum_{n=-N}^{N} a_n e^{nxi}\right)^2 \mathrm{d}x\right)^{1/2} \to 0 \quad \text{as} \quad N \to \infty \ , \tag{8.1.1}$$

where

$$a_n = \frac{\langle f, \phi_n\rangle}{\langle \phi_n, \phi_n\rangle} = \frac{1}{2\pi}\int_{\pi}^{\pi} f(x)\overline{\phi_n(x)}\,\mathrm{d}x = \frac{1}{2\pi}\int_{\pi}^{\pi} f(x)e^{-nxi}\,\mathrm{d}x \quad , \quad n \in \mathbb{Z} \ . \tag{8.1.2}$$

$\displaystyle\sum_{n\in\mathbb{Z}} a_n e^{nxi}$ is the **(complex) Fourier series** of f. We may write

$$f = \sum_{n\in\mathbb{Z}} a_n e^{nxi} \ ,$$

if the equality is interpreted in the sense of (8.1.1). We do not necessarily have pointwise convergence.

From Theorem 8.1.5, the minimum of

$$I(r_{-N}, r_{-N+1}, \ldots, r_{N-1}, r_N) = \int_{-\pi}^{\pi}\left(f(x) - \sum_{n=-N}^{N} r_n e^{rxi}\right)^2 \mathrm{d}x$$

for $r_n \in \mathbb{C}$ is given by $r_n = a_n$ in (8.1.2). ♣

For the rest of this section, we assume that Ω is an interval $[a,b]$ and the measure is $\mathrm{d}\mu(x) = w(x)\,\mathrm{d}x$, where $w : [a,b] \to \mathbb{R}$ is a piecewise continuous function on the interval $[a,b]$ such that $w(x) > 0$ for almost all $x \in [a,b]$. The function w is called a **weight function** on $[a,b]$. We consider only real valued functions. The following discussion is also valid if we replace the interval $[a,b]$ by an open interval $]a,b[$, a semi-open interval $[a,b[$, or an unbounded interval.

$L^2[a,b]$ is the space of measurable functions $f : [a,b] \to \mathbb{R}$ such that $\displaystyle\int_a^b f^2(x)w(x)\,\mathrm{d}x$ is finite. The scalar product on $L^2[a,b]$ is

$$\langle f, g\rangle = \int_a^b f(x)g(x)w(x)\,\mathrm{d}x \quad , \quad f, g \in L^2[a,b] \ . \tag{8.1.3}$$

The L^2-norm is

$$\|f\|_2 = \sqrt{\langle f, f\rangle} = \left(\int_a^b f^2(x)w(x)\,\mathrm{d}x\right)^{1/2} \quad , \quad f \in L^2[a,b] \ . \tag{8.1.4}$$

The space $L^2[a,b]$ has a countable orthonormal basis. Suppose that

$$S = \{\phi_n : n \in \mathbb{N}\} \subset L^2[a,b]$$

is an orthonormal basis for $L^2[a,b]$, the Fourier series of a function $f \in L^2[a,b]$ with respect to this basis is

$$f \sim \sum_{n=0}^{\infty} a_n \phi_n \ ,$$

where

$$a_n = \langle f, v_n \rangle = \int_a^b f(x)\phi_n(x)w(x)\,\mathrm{d}x \quad , \quad n = 0,1,2,\ldots$$

Sometime, we only have a complete orthogonal set of functions

$$S = \{\phi_n : n \in \mathbb{N}\} \subset L^2[a,b]$$

In this case, the Fourier series of $f \in L^2[a,b]$ with respect to this set of functions is

$$f \sim \sum_{n=0}^{\infty} a_n \phi_n \ ,$$

where

$$a_n = \frac{\langle f, \phi_n\rangle}{\|\phi_n\|^2} = \left(\int_a^b (\phi_n(x))^2 w(x)\,\mathrm{d}x\right)^{-1} \int_a^b f(x)\phi_n(x)w(x)\,\mathrm{d}x \quad , \quad n = 0,1,2,\ldots$$

We have

$$\left\| f - \sum_{n=1}^{N} a_n\phi_n \right\|_2 = \left(\int_a^b \left(f(x) - \sum_{n=0}^{N} a_n\phi_n(x)\right)^2 w(x)\,\mathrm{d}x\right)^{1/2} \to 0 \quad \text{as} \quad N \to \infty \ .$$

We say that $\displaystyle\sum_{n=0}^{N} a_n\phi_n$ converges in L^2 to f as $N \to \infty$.

Example 8.1.7

There is a real form for the trigonometric polynomials of Example 8.1.6. As stated at the beginning of the section, we consider only real valued functions. Let

$$\phi_0(x) = \frac{1}{\sqrt{2\pi}} \ , \quad \phi_{2j}(x) = \frac{1}{\sqrt{\pi}}\cos(jx) \quad \text{and} \quad \phi_{2j-1}(x) = \frac{1}{\sqrt{\pi}}\sin(jx)$$

for $j = 1,\ 2,\ \ldots$ It is easy to verify that

$$\int_{-\pi}^{\pi} \phi_i(x)\phi_j(x)\,\mathrm{d}x = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Thus $S = \{\phi_n : n \in \mathbb{N}\}$ is a set of orthonormal functions in $L^2[-\pi,\pi]$, where the weight function is $w(x) = 1$ for all x. It is possible to show that the set of all linear combination of elements of S is dense in $L^2[-\pi,\pi]$.

Hence, for $f \in L^2[-\pi,\pi]$, we have that

$$\left(\int_{-\pi}^{\pi} \left(f(x) - a_0 - \sum_{n=0}^{N} a_n\cos(nx) - \sum_{n=0}^{N} b_n\sin(nx)\right)^2 \mathrm{d}x\right)^{1/2} \to 0 \quad \text{as} \quad N \to \infty \ , \qquad (8.1.5)$$

where

$$a_0 = \left\langle f, \frac{1}{\sqrt{2\pi}} \right\rangle = \frac{1}{\sqrt{2\pi}} \int_{\pi}^{\pi} f(x)\,\mathrm{d}x \ , \tag{8.1.6}$$

$$a_n = \left\langle f, \frac{1}{\sqrt{\pi}} \cos(nx) \right\rangle = \frac{1}{\sqrt{\pi}} \int_{-\pi}^{\pi} f(x)\cos(nx)\,\mathrm{d}x \tag{8.1.7}$$

and

$$b_n = \left\langle f, \frac{1}{\sqrt{\pi}} \sin(nx) \right\rangle = \frac{1}{\sqrt{\pi}} \int_{-\pi}^{\pi} f(x)\sin(nx)\,\mathrm{d}x \tag{8.1.8}$$

for $n = 1, 2, 3, \ldots$ We write

$$f = a_0 + \sum_{n=0}^{\infty} a_n \cos(nx) + \sum_{n=0}^{\infty} b_n \sin(nx) \ .$$

This is the **classical Fourier series** of f. As for the complex Fourier series, the equality in the expression above is in the sense of convergence in $L^2[-\pi,\pi]$; namely, (8.1.5) is satisfied. We may not have pointwise convergence for all $x \in [a,b]$.

From Theorem 8.1.5, the minimum of

$$\begin{aligned} &I(r_0, r_1, r_2, \ldots, r_N, s_1, s_2, \ldots, s_N) \\ &\quad = \int_{-\pi}^{\pi} \left(f(x) - r_0 - \sum_{n=0}^{N} r_n \cos(nx) - \sum_{n=0}^{N} s_n \sin(nx) \right)^2 \mathrm{d}x \end{aligned}$$

for r_n and s_n in $\mathbb{R}$ is reached at $r_n = a_n$ and $s_n = b_n$ defined in (8.1.6), (8.1.7) and (8.1.8). ♣

In the following section, we will only consider bases formed on polynomials.

8.2 Bases of Polynomial

For each $n \in \mathbb{N}$, let $P_n(x) = \sum_{j=0}^{n} \alpha_{n,j} x^j$ be a polynomial of degree exactly n; namely, $\alpha_{n,n} \neq 0$. These polynomials can be considered as elements of $L^2[a,b]$.

We now prove that for any finite subset A of $\mathbb{N}$, if $\sum_{n\in A} c_n P_n = 0$ with $c_n \in \mathbb{R}$, then $c_n = 0$ for all i. The proof is by induction on the cardinality of the set of indices A.

If A is of cardinality one, say $A = \{n_1\} \subset \mathbb{N}$, then $c_1 P_{n_1} = \sum_{j=0}^{n_1} c_1 \alpha_{n_1,j} x^j = 0$ implies that $c_a \alpha_{n_1,n_1} = 0$ with $\alpha_{n_1,n_1} \neq 0$. Thus $c_1 = 0$.

Our hypothesis of induction is that for any set $A = \{n_1, n_2, \ldots, n_k\} \subset \mathbb{N}$ of cardinality k (in particular, $n_j \neq n_i$ for $i \neq j$), we have that $\sum_{i=1}^{k} c_i P_{n_i} = 0$ with $c_i \in \mathbb{R}$ implies that $c_i = 0$ for all

i. Suppose that $\displaystyle\sum_{i=1}^{k+1} c_i P_{n_i} = 0$ with $c_i \in \mathbb{R}$ for a set $A = \{n_1, n_2, \ldots, n_k, n_{k+1}\} \subset \mathbb{N}$ of cardinality $k+1$. Let $n_j = \displaystyle\max_{1\leq i\leq k+1}\{n_i\}$. The only term of degree n_j in $\displaystyle\sum_{i=1}^{k+1} c_i P_{n_i} = 0$ is $c_j \alpha_{n_j,n_j} x^{n_j}$. Hence $c_j \alpha_{n_j,n_j} x^{n_j} = 0$ with $\alpha_{n_j,n_j} \neq 0$. Thus $c_j = 0$. We therefore get $\displaystyle\sum_{\substack{i=1\\ i\neq j}}^{k+1} c_i P_{n_i} = 0$ with $c_i \in \mathbb{R}$. Since the set of indices in this sum is of cardinality k, we get by induction that $c_i = 0$ for all i.

We can also prove by induction on the degree that all polynomials of degree less then or equal to k can be expressed as a linear combination of P_0, P_1, P_2, $\ldots$, P_k.

The result is obviously true for $k = 0$ because $P_0(x) = \alpha_{0,0} \neq 0$ for all x, and every real number can be expressed as the product of $\alpha_{0,0}$ with another real number.

We assume by induction that all polynomials of degree less then or equal to k can be expressed as a linear combination of P_0, P_1, P_2, $\ldots$, P_k. Consider p, a polynomial of degree $k+1$. Suppose that c is the coefficient of x^{k+1} in $p(x)$. Since p and P_{k+1} are of degree $k+1$, we have that $c \neq 0$ and $\alpha_{k+1,k+1} \neq 0$. Thus, $p - (c/\alpha_{k+1,k+1})P_{k+1}$ is a polynomial of degree k. By induction, we may write

$$p - \frac{c}{\alpha_{k+1,k+1}} P_{k+1} = \sum_{j=0}^{k} c_j P_j$$

for some $c_j \in \mathbb{R}$. Hence

$$p = \sum_{j=0}^{k+1} c_j P_j$$

with $c_{k+1} = c/\alpha_{k+1,k+1}$ and the other c_j as before.

Using Stone-Weierstrass Theorem, Theorem 9.1.1, and the density of continuous functions in $L^2[a,b]$, we may show that the set of all finite linear combinations of elements of the set $P = \{P_n : n \in \mathbb{N}\}$ is dense in $L^2[a,b]$. Combined with the linear independence of P, this shows that P is a basis of $L^2[a,b]$.

Remark 8.2.1

A more direct proof of the linear independence of P can also be given. Suppose that $\displaystyle\sum_{i=1}^{k} c_i P_{n_i} = 0$ with $c_i \in \mathbb{R}$.

Without loss of generality, we may assume that $n_1 < n_2 < \ldots < n_k$. The previous equation can be written

$$\sum_{i=0}^{k} \left(\sum_{j=0}^{n_i} c_i \alpha_{n_i,j} x^j \right) = 0 \ .$$

If we consider only the terms in x^{n_j}, we get the following system of linear equations.

$$0 = c_k\, \alpha_{n_k,n_k} \ , \tag{8.2.1}$$
$$0 = c_k\, \alpha_{n_k,n_{k-1}} + c_{k-1}\, \alpha_{n_{k-1},n_{k-1}} \ , \tag{8.2.2}$$
$$0 = c_k\, \alpha_{n_k,n_{k-2}} + c_{k-1}\, \alpha_{n_{k-1},n_{k-2}} + c_{k-2}\, \alpha_{n_{k-2},n_{k-2}} \ , \tag{8.2.3}$$
$$0 = c_k\, \alpha_{n_k,n_{k-3}} + c_{k-1}\, \alpha_{n_{k-1},n_{k-3}} + c_{k-2}\, \alpha_{n_{k-2},n_{k-3}} + c_{k-3}\, \alpha_{n_{k-3},n_{k-3}} \ , \tag{8.2.4}$$

$$
\begin{aligned}
\vdots &= \quad \vdots \\
0 &= c_k\, \alpha_{n_k,n_0} + c_{k-1}\, \alpha_{n_{k-1},n_0} + c_{k-2}\, \alpha_{n_{k-2},n_0} + \ldots + c_1\, \alpha_{n_1,n_0} \ .
\end{aligned}
\tag{8.2.5}
$$

Using forward substitution to solve for the c_i, we find that $c_i = 0$ for all i. In other words, since $\alpha_{n_k,n_k} \neq 0$, (8.2.1) implies that $c_k = 0$. Since $c_k = 0$ and $\alpha_{n_{k-1},n_{k-1}} \neq 0$, (8.2.2) implies that $c_{k-1} = 0$. Since $c_k = c_{k-1} = 0$ and $\alpha_{n_{k-2},n_{k-2}} \neq 0$, (8.2.3) implies that $c_{k-2} = 0$. Inductively, we get $c_i = 0$ for all i. ♠

Remark 8.2.2

A direct proof that all polynomials of degree less then or equal to k can be expressed as a linear combination of P_0, P_1, P_2, $\ldots$, P_k is as it follows. Since P_0, P_1, $\ldots$, P_k are $k+1$ linearly independent elements of the space of polynomials of degree less than or equal to k, and since this space is of dimension $k+1$, we have that $\{P_0, P_1, \ldots, P_k\}$ is a basis of the space of polynomials of degree less than or equal to k. ♠

The following theorem gives a simple procedure to generate families of orthogonal polynomials. As shown in Question 8.2, the procedure is even simpler for families of orthonormal polynomials.

Theorem 8.2.3

Let $\{P_0, P_1, P_2, \ldots\}$ be an orthogonal set of polynomials on $[a,b]$ with respect to a weight function w. Moreover, suppose that P_k is of degree exactly k for all k. Then,

1. Any polynomial $p(x)$ of degree at most n can be expressed as a linear combination $p = \displaystyle\sum_{k=0}^{n} c_k\, P_k$ for some constants c_0, c_1, $\ldots$, c_n.
2. If p is a polynomial of degree less than k, then p is orthogonal to P_k.
3. For each positive integer k, P_k has exactly k distinct real roots in $]a,b[$.
4. If the coefficient of x^k in P_k is $\alpha_{k,k}$, then

$$
P_{k+1}(x) = A_k(x - B_k)P_k(x) - C_k P_{k-1}(x)
$$

for $k \geq 0$, where

$$
P_{-1} = 0 \ , \quad A_k = \frac{\alpha_{k+1,k+1}}{\alpha_{k,k}} \quad \text{for} \quad k \geq 0 \ , \quad B_k = \frac{\displaystyle\int_a^b x P_k^2(x) w(x)\,\mathrm{d}x}{\displaystyle\int_a^b P_k^2(x) w(x)\,\mathrm{d}x} \quad \text{for} \quad k \geq 0 \ ,
$$

$$
C_0 = 0 \quad \text{and} \quad C_k = \frac{A_k \displaystyle\int_a^b P_k^2(x) w(x)\,\mathrm{d}x}{A_{k-1} \displaystyle\int_a^b P_{k-1}^2(x) w(x)\,\mathrm{d}x} \quad \text{for} \quad k > 0 \ .
$$

Proof.

1) We use induction on the degree of the polynomial p.

If p is of degree 0, then $p(x) = b$ for all x, where b is a constant. Since P_0 is a non-trivial polynomial of degree 0 by assumption, $P_0(x) = \alpha_{0,0} \neq 0$ for all x. Hence, $p = a_0 P_0$ with $a_0 = b/\alpha_{0,0}$.

Assume that every polynomial of degree less than n can be expressed as a linear combination of P_0, P_1, $\ldots$, P_{n-1}. Let p be a polynomial of degree exactly n. Let b be the coefficient of x^n in p. The constant b is non-null because p is of degree exactly n. Similarly, the coefficient $\alpha_{n,n}$ of x^n in P_n is non null because P_n is of degree exactly n. Hence, $p - a_n P_n$ with $a_n = b/\alpha_{n,n}$ is a polynomial of degree $n-1$ that can therefore be expressed as a linear combination of the polynomials P_i for $0 \leq i < n$ by the hypothesis of induction. Namely,

$$p - a_n P_n = \sum_{j=0}^{n-1} a_j P_j$$

for some constants a_0, a_1, $\ldots$, a_{n-1}. Hence,

$$p = \sum_{j=0}^{n} a_j P_j \ .$$

2) Let p be a polynomial of degree less than k. According to (1), we can write p as a linear combination

$$p = \sum_{j=0}^{k-1} b_j P_j$$

for some constants b_0, b_1, $\ldots$, b_{k-1}. Then

$$\int_a^b p(x) P_k(x) w(x)\,\mathrm{d}x = \int_a^b \left(\sum_{j=0}^{k-1} b_j P_j(x) \right) P_k(x) w(x)\,\mathrm{d}x = \sum_{j=0}^{k-1} b_j \int_a^b P_j(x) P_k(x) w(x)\,\mathrm{d}x = 0$$

because $\displaystyle \int_a^b P_j(x) P_k(x) w(x)\,\mathrm{d}x = 0$ for all $j < k$ by hypothesis.

3) We note that a consequence of the Fundamental Theorem of Algebra is that P_k cannot have more than k roots and so more than k distinct real roots in $]a,b[$. Suppose that P_k change sign at $r < k$ distinct points in $]a,b[$ only. Let x_1, x_2, $\ldots$, x_r be these r distinct points and choose $\hat{x} \in]a,b[$ such that $x_j < \hat{x}$ for $1 \leq j \leq r$. Then

$$p(x) = P_k(\hat{x})(x - x_1)(x - x_2)\ldots(x - x_r)$$

is a polynomial of degree $r < k$ such that $p(x)P_k(x) > 0$ for all $x \in]a,b[\setminus \{x_1, x_2, \ldots, x_r\}$. Hence,

$$\int_a^b p(x) P_k(x) w(x)\,\mathrm{d}x > 0$$

contradicts the orthogonality result of (2).

4) We write $P_{k+1}(x) = A_k x P_k(x) + q(x)$, where $A_k = \alpha_{k+1.k+1}/\alpha_{k,k}$ and $q(x)$ is a polynomial of degree at most k. From 1, we may write q as a linear combination

$$q(x) = \sum_{j=0}^{k} b_j P_j$$

for some constants b_0, b_1, ..., b_k. Hence,

$$P_{k+1}(x) = A_k x P_k(x) + \sum_{j=0}^{k} b_j P_j(x) \ . \tag{8.2.6}$$

We have that

$$\begin{aligned}\int_a^b P_{k+1}(x)P_i(x)w(x)\,\mathrm{d}x = A_k \int_a^b xP_k(x)P_i(x)w(x)\,\mathrm{d}x \\ + \sum_{j=0}^{k} b_j \int_a^b P_j(x)P_i(x)w(x)\,\mathrm{d}x\end{aligned} \tag{8.2.7}$$

for all i. From (2), (8.2.7) yields

$$0 = b_i \int_a^b P_i^2(x)w(x)\,\mathrm{d}x$$

for $0 \le i \le k-2$; namely,

$$b_i = 0 \quad , \quad 0 \le i \le k-2 \ . \tag{8.2.8}$$

For $i = k$, (8.2.7) yields

$$0 = A_k \int_a^b xP_k^2(x)w(x)\,\mathrm{d}x + b_k \int_a^b P_k^2(x)w(x)\,\mathrm{d}x \ .$$

Thus,

$$b_k = -A_k \frac{\int_a^b xP_k^2(x)w(x)\,\mathrm{d}x}{\int_a^b P_k^2(x)w(x)\,\mathrm{d}x} = -A_k B_k \ . \tag{8.2.9}$$

For $i = k-1$, (8.2.7) yields

$$0 = A_k \int_a^b xP_k(x)P_{k-1}(x)w(x)\,\mathrm{d}x + b_{k-1} \int_a^b P_{k-1}^2(x)w(x)\,\mathrm{d}x \ . \tag{8.2.10}$$

However, from (1), we may write

$$xP_{k-1}(x) - \frac{\alpha_{k-1,k-1}}{\alpha_{k,k}} P_k(x) = \sum_{j=0}^{k-1} c_j P_j(x)$$

for some constants c_0, c_1, ..., c_{k-1}. Hence,

$$\begin{aligned}\int_a^b xP_k(x)P_{k-1}(x)w(x)\,\mathrm{d}x &= \frac{\alpha_{k-1,k-1}}{\alpha_{k,k}} \int_a^b P_k^2(x)w(x)\,\mathrm{d}x + \sum_{j=0}^{k-1} c_j \int_a^b P_j(x)P_k(x)w(x)\,\mathrm{d}x \\ &= \frac{\alpha_{k-1,k-1}}{\alpha_{k,k}} \int_a^b P_k^2(x)w(x)\,\mathrm{d}x = \frac{1}{A_{k-1}} \int_a^b P_k^2(x)w(x)\,\mathrm{d}x\end{aligned}$$

by (2). Thus, from (8.2.10),

$$b_{k-1} = -\frac{A_k \int_a^b P_k^2(x)w(x)\,\mathrm{d}x}{A_{k-1}\int_a^b P_{k-1}^2(x)w(x)\,\mathrm{d}x} = -C_k \ . \tag{8.2.11}$$

Substituting (8.2.8), (8.2.9) and (8.2.11) into (8.2.6) gives (4). ∎

Example 8.2.4
Find the first three polynomials of the orthogonal set $\{P_0, P_1, P_2, \ldots\}$ if the interval is $[a,b] = [-1,1]$, the weight function is $w(x) = \sqrt{1-x^2}$, and the coefficient $\alpha_{k,k}$ of x^k in the polynomial P_k is 1 for all k.

Due to our assumption on the coefficient of x^i in P_i, we have that $P_0(x) = 1$ for all x. For the sake of the computations. we let $P_{-1}(x) = 0$ for all x and $C_0 = 0$. We have

$$P_1 = A_0(x - B_0)P_0 - C_0 P_{-1} \ ,$$

where $A_0 = \alpha_{1,1}/\alpha_{0,0} = 1$ and

$$B_0 = \frac{\int_{-1}^1 x P_0^2(x)\, w(x)\, \mathrm{d}x}{\int_{-1}^1 P_0^2(x)\, w(x)\, \mathrm{d}x} = \frac{\int_{-1}^1 x\sqrt{1-x^2}\, \mathrm{d}x}{\int_{-1}^1 \sqrt{1-x^2}\, \mathrm{d}x} = 0 \ .$$

The integral on the numerator is zero because it is the integral of an odd function on the symmetric interval $[-1,1]$. To compute the integral in the denominator, one may use the trigonometric substitution $x = \sin(\theta)$ for $-\pi/2 \leq \theta \leq \pi/2$ or note that the integral is equal to $\pi/2$, half the area of the disk of radius 1. Thus,

$$P_1(x) = x$$

for all x.

We have

$$P_2 = A_1(x - B_1)P_1 - C_1 P_0 \ ,$$

where $A_1 = \alpha_{2,2}/\alpha_{1,1} = 1$,

$$B_1 = \frac{\int_{-1}^1 x P_1^2(x)\, w(x)\, \mathrm{d}x}{\int_{-1}^1 P_1^2(x)\, w(x)\, \mathrm{d}x} = \frac{\int_{-1}^1 x^3 \sqrt{1-x^2}\, \mathrm{d}x}{\int_{-1}^1 x^2 \sqrt{1-x^2}\, \mathrm{d}x} = 0$$

and

$$C_1 = \frac{A_1 \int_{-1}^1 P_1^2(x)\, w(x)\, \mathrm{d}x}{A_0 \int_{-1}^1 P_0^2(x)\, w(x)\, \mathrm{d}x} = \frac{\int_{-1}^1 x^2 \sqrt{1-x^2}\, \mathrm{d}x}{\int_{-1}^1 \sqrt{1-x^2}\, \mathrm{d}x} = \frac{1}{4} \ .$$

Again, the integral on the numerator of B_1 is zero because it is the integral of an odd function on the symmetric interval $[-1,1]$. To compute the other integrals, the trigonometric substitution $x = \sin(\theta)$ for $-\pi/2 \leq \theta \leq \pi/2$ may be used. Hence,

$$P_2(x) = x^2 - \frac{1}{4}$$

for all x. ♣

Example 8.2.5 (Normalized Legendre Polynomials)
If, in the fourth item of Theorem 8.2.3, we take $a = -1$, $b = 1$, $w(x) = 1$, $\alpha_{0,0} = 1$ and

$$\alpha_{k+1,k+1} = \frac{2k+1}{k+1}\, \alpha_{k,k}$$

for $k \geq 0$, we get $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = \dfrac{3}{2}\left(x^2 - \dfrac{1}{3}\right)$, $P_3(x) = \dfrac{5}{2}\left(x^3 - \dfrac{3}{5}x\right)$, and in general
$$P_{k+1}(x) = \frac{1}{k+1}\big((2k+1)xP_k(x) - kP_{k-1}(x)\big)$$
for $k = 1, 2, 3, \ldots$ These polynomials are said to be normalized because $P_k(1) = 1$ for all i.

The usual approach to derive the recursion relation above is to show that the Legendre polynomial P_n is the only bounded solution of the second order differential equation
$$(1-x^2)y'' - 2xy' + n(n-1)y = 0 \quad , \quad -1 < x < 1 \ .$$
One can then show that
$$P_n(x) = \frac{1}{n!\,2^n}\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n$$
for $n \geq 0$. A good reference on the subject of Legendre polynomials is [29]. ♣

Example 8.2.6 (**Normalized Chebyshev Polynomials**)
If, in the fourth item of Theorem 8.2.3, we take $a = -1$, $b = 1$, $w(x) = (1-x^2)^{-1/2}$, $\alpha_{0,0} = 1$ and $\alpha_{k,k} = 2^{k-1}$ for $k \geq 1$, we get $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = 2x^2 - 1$, $P_3(x) = 4x^3 - 3x$, and in general
$$P_{k+1}(x) = 2x\,P_k(x) - P_{k-1}(x)$$
for $k = 1, 2, 3, \ldots$

As for the Legendre polynomials, the usual approach to derive the recursion relation above is to show that the Chebyshev polynomial P_n is the only bounded solution of the second order differential equation
$$(1-x^2)y'' - xy' + n^2y = 0 \quad , \quad -1 < x < 1 \ .$$
One can then show that
$$P_n(x) = \cos(n\arccos(x))$$
for $n \geq 0$. We prove this result in Section 9.2. A good reference on the subject of Chebyshev polynomials is [29]. ♣

Example 8.2.7
There are many more sets of orthogonal polynomials.

1. If, in the fourth item of Theorem 8.2.3, we take $a = -1$, $b = 1$ and $w(x) = (1-x)^\alpha(1+x)^\beta$ with $\alpha, \beta > -1$, we get the Jacobi polynomials $P_k^{[\alpha,\beta]}$ for $k = 0, 1, 2, \ldots$ For each value of α and β, we get a different sets of orthogonal polynomials. The sets of orthogonal polynomials that we have seen in the two previous examples are associated to particular values of α and β. For $\alpha = \beta = 0$, we have the Legendre polynomials. For $\alpha = \beta = -1/2$, we have the Chebyshev polynomials.

2. If, in the fourth item of Theorem 8.2.3, we take $a = 0$, $b = \infty$ and $w(t) = x^\alpha e^{-x}$ with $\alpha > -1$, we get the Laguerre polynomials $l_k^{[\alpha]}$.

3. If, in the fourth item of Theorem 8.2.3, we take $a = -\infty$, $b = +\infty$ and $w(x) = e^{-x^2}$, we get the Hermite polynomials H_k.

Note that these orthogonal polynomials are not normalized.

As we mentioned before for the Legendre and Chebyshev polynomials, These classical sets of orthogonal polynomials are generally introduced when studying their associated differential equations. They represent polynomial solutions to these differential equations. The recurrence formulae and many other properties of these orthogonal polynomials are more naturally deduced using their presentation in the context of differential equations. ♣

8.3 Orthogonal Polynomials and Least Square Approximation

For $n \in \mathbb{N}$, let $P_n : \mathbb{R} \to \mathbb{R}$ be polynomials of degree exactly n. These polynomials can be considered as elements of $L^2[a,b]$.

We want to find the "best approximation" of $f \in L^2[a,b]$ by a finite linear combination of the polynomials P_n. More precisely, let $S = \{P_n : 0 \leq n \leq k\}$. We are looking for real values a_0, a_1, a_2, ..., a_k that minimize

$$I(a_0, a_1, \ldots, a_k) = \left\| f - \sum_{i=0}^{k} a_i P_i \right\|_2^2 = \int_a^b \left(f(x) - \sum_{i=0}^{k} a_i P_i(x) \right)^2 w(x) \, \mathrm{d}x \ .$$

The good old calculus will help us to solve this problem, If I has a minimum at $\mathbf{a} = \begin{pmatrix} a_0 & a_1 & \ldots & a_k \end{pmatrix}^\top$, then $\nabla I(\mathbf{a}) = \mathbf{0}$. Assuming that we may differentiate under the integral sign (e.g. if f is continuous on the close interval $[a,b]$), we get

$$0 = -2a_j \int_a^b \left(f(x) - \sum_{i=0}^{k} a_i P_i(x) \right) P_j(x) w(x) \, \mathrm{d}x$$

for $0 \leq j \leq k$. These equations yield the system of linear equations

$$D\mathbf{a} = \mathbf{c} \ , \tag{8.3.1}$$

where

$$d_{j,i} = \int_a^b P_i(x) P_j(x) w(x) \, \mathrm{d}x$$

for $0 \leq i, j \leq k$ and

$$c_j = \int_a^b f(x) P_j(x) w(x) \, \mathrm{d}x$$

for $0 \leq j \leq k$. There are $k+1$ equations and $k+1$ unknowns.

For general polynomials P_n, this system may be hard to solve. For instance, suppose that the weight function is $w(x) \equiv 1$ and the polynomials are $P_n(x) = x^n$ for all $n \in \mathbb{N}$. Then

$$d_{j,i} = \int_a^b x^i x^j \, \mathrm{d}x = \frac{b^{i+j+1} - a^{i+j+1}}{i+j+1}$$

for $i, j = 0, 1, 2, \ldots, k$. The matrix D in (8.3.1) is then a **Hilbert matrix**. This type of matrices is ill-conditioned. In particular, no pivoting technique can be used to get a good approximation of the solution if the matrix is large. For this reason, we must choose a good set $S = \{P_n : n \in \mathbb{N}\}$ of polynomials, where we still have that P_n is a polynomial of degree exactly n for $n \in \mathbb{N}$. The obvious choice is an orthogonal (or even an orthonormal) set S of polynomials. With this choice, D in (8.3.1) is a diagonal matrix and it is easy to find the solution $\mathbf{a}$ of (8.3.1). More precisely,

$$a_i = \frac{\int_a^b f(x)P_i(x)w(x)\,\mathrm{d}x}{\int_a^b P_i^2(x)w(x)\,\mathrm{d}x}$$

for $0 \leq i \leq k$ as predicted by Theorem 8.1.5.

8.4 Exercises

Question 8.1
Suppose that $w : [a, b] \to [0, \infty[$ is a weight function. Without referring to Theorem 8.2.3, prove that we cannot have two distinct orthogonal families of monic polynomials $\{P_k\}_{k=0}^{\infty}$ such that P_k is of degree k and

$$\langle p, P_k\rangle = \int_a^b p(x)\,P_k(x)\,w(x)\,\mathrm{d}x = 0$$

for all polynomial p of degree less than k. Recall that a monic polynomial $p(x)$ of degree n is a polynomial where the coefficient of the term in x^n is 1.

Question 8.2
Let $\{P_0, P_1, P_2, \ldots\}$ be an orthonornal set of polynomials on $[a, b]$ with respect to a weight function w. Suppose that $P_k(x) = \sum_{j=0}^{k} a_{k,j}x^j$ with $a_{k,k} \neq 0$ for all k. Prove that

$$P_{k+1}(x) = A_k(x - B_k)P_k(x) - C_kP_{k-1}(x) \tag{8.4.1}$$

for $k = 0, 1, 2, \ldots$, where $P_{-1} = 0$, $A_k = \dfrac{a_{k+1,k+1}}{a_{k,k}}$, $B_k = \dfrac{a_{k,k-1}}{a_{k,k}} - \dfrac{a_{k+1,k}}{a_{k+1,k+1}}$ and $C_k = \dfrac{a_{k+1,k+1}a_{k-1,k-1}}{a_{k,k}^2}$ for $k \geq 0$ if we set $a_{-1,-1} = a_{0,-1} = 0$.

Question 8.3
Prove that the normalized Legendre polynomial $P_k(x)$ satisfies

$$\int_{-1}^{1} |P_k(x)|^2\,\mathrm{d}x = \frac{2}{2k+1}\ .$$

Chapter 9

Uniform Approximation

Suppose that $f : \mathbb{R} \to \mathbb{R}$ is a sufficiently differential function. The Taylor expansion of the function f at a point $c \in \mathbb{R}$ is given by

$$f(x) = p_n(x) + r_n(x) \ ,$$

where

$$p_n(x) = f(c) + f'(c)(x-c) + \frac{f'(c)}{2!}(x-c)^2 + \ldots + \frac{f^{(n)}(c)}{n!}(x-c)^n$$

and

$$r_n(x) = \frac{f^{(n+1)}(\xi(x,c))}{(n+1)!}(x-c)^{n+1}$$

for some $\xi(x,c)$ between x and c. If $c \in [a,b]$ and $\left|f^{(n+1)}(x)\right| < M$ for $[a \leq x \leq b$, then we get that

$$\sup_{a\leq x\leq b} |f(x) - p_n(x)| \leq \frac{M}{(n-1)!}(b-a)^{n+1} \ .$$

This is an uniform approximation of f by a polynomial p_n. If M is small and $b - a \leq 1$, then p can be a good uniform approximation of f. So, instead of evaluating $f(x)$, it may be simpler and sufficiently accurate to evaluate $p_n(x)$. Unfortunately, in practice, it may be very hard to compute the derivatives of f and to find an upper bound on $|f^{n+1}|$. Moreover, the interval $[a,b]$ may be of length greater than 1 and so $(b-a)^{n+1} \to \infty$ as $n \to \infty$.

Therefore, it is still preferable to use interpolation polynomials as presented in Chapter 6 to approximate f. However, among all the possible interpolating polynomials, it is possible to choose the points of interpolation to minimize the degree of the interpolating polynomial and the error. This is the major result of this chapter that we present in Section 9.2.1 below.

9.1 Stone-Weierstrass Theorem

The fundamental theorem in this chapter is the following.

Theorem 9.1.1 (Stone-Weierstrass)

Given a continuous function $f : [a,b] \to \mathbb{C}$, there exists a sequence of polynomials $\{p_n\}_{n=0}^{\infty}$ over $\mathbb{C}$ such that

$$\max_{a\leq x\leq b} |f(x) - p_n(x)| \to 0 \quad \text{as} \quad n \to \infty \ .$$

If $f : [a,b] \to \mathbb{R}$, the polynomials can be assumed to be over $\mathbb{R}$.

Basically, the theorem states that for any continuous function $f : [a,b] \to \mathbb{C}$, we can find a sequence of polynomials converging to f uniformly on $[a,b]$. We will not prove this theorem. There exist many proofs of it. The reader can find one of them in any good analysis textbook.

9.2 Chebyshev Polynomials

We have seen in Example 8.2.6 that the Chebyshev polynomials were defined by

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

for $n = 1, 2, 3, \ldots$ with $T_0(x) = 1$ and $T_1(x) = x$. The tradition is to denote the Chebyshev polynomials with the letter T instead of P because the translation from Russian to French of Chebyshev is Tchébyshev.

There is an equivalent way to define the Chebyshev polynomials from which it is easier to deduce some of the properties of the Chebyshev polynomials.

The Chebyshev polynomial T_n is also defined by

$$T_n(x) = \cos(n \arccos(x)) \quad , \quad -1 \leq x \leq 1 \ . \tag{9.2.1}$$

To verify that this is true, we note that $T_0(x) = \cos(0) = 1$ and $T_1(x) = \cos(\arccos(x)) = x$ for $x \in [-1,1]$. Moreoevr, it follows from the addition formulae for the cosine function that

$$\cos((n+1)\theta) = \cos(n\theta + \theta) = \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta)$$

and

$$\cos((n-1)\theta) = \cos(n\theta - \theta) = \cos(n\theta)\cos(\theta) + \sin(n\theta)\sin(\theta) \ .$$

Hence

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2\cos(n\theta)\cos(\theta) \ .$$

If we substitute $\theta = \arccos(x)$ in this last equation, we get the recurrence relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{9.2.2}$$

for $n = 1, 2, 3, \ldots$ used to defined the Chebyshev polynomials. Thus (9.2.1) is another way to define the Chebyshev polynomials.

We know from Theorem 8.2.3 that the Chebyshev polynomials are orthogonal on $L^2[-1,1]$, where the weight function is $w(x) = \dfrac{1}{\sqrt{1-x^2}}$. This can be directly proved from (9.2.1). Using the substitution $\theta = \arccos(x)$ for $-1 < x < 1$ and $\mathrm{d}\theta = \dfrac{-1}{\sqrt{1-x^2}}\,\mathrm{d}x$, we get

$$\begin{aligned}\int_{-1}^{1} \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}}\,\mathrm{d}x &= \int_{-1}^{1} \frac{\cos(i\arccos(x))\cos(j\arccos(x))}{\sqrt{1-x^2}}\,\mathrm{d}x = -\int_{\pi}^{0}\cos(i\theta)\cos(j\theta)\,\mathrm{d}\theta \\ &= \int_0^{\pi}\left(\frac{1}{2}\cos((i+j)\theta) + \frac{1}{2}\cos((i-j)\theta)\right)\mathrm{d}\theta \\ &= \left(\frac{\sin((i+j)\theta)}{2(i+j)} + \frac{\sin((i-j)\theta)}{2(i-j)}\right)\Bigg|_{\theta=0}^{\pi} = 0\end{aligned}$$

for $i \neq j$. The Chebyshev polynomials are not of norm one because, using the substitution for the previous integral, we have

$$\begin{aligned}\int_{-1}^{1} \frac{T_i^2(x)}{\sqrt{1-x^2}}\,\mathrm{d}x &= \int_{-1}^{1} \frac{\cos^2(i\arccos(x))}{\sqrt{1-x^2}}\,\mathrm{d}x = -\int_{\pi}^{0}\cos^2(i\theta)\,\mathrm{d}\theta \\ &= \int_0^{\pi}\frac{1}{2}\left(1+\cos(2i\theta)\right)\,\mathrm{d}\theta = \frac{1}{2}\left(\theta - \frac{1}{2i}\sin(2i\theta)\right)\Bigg|_{\theta=0}^{\pi} = \frac{\pi}{2}\end{aligned}$$

for $i \neq 0$, and

$$\int_{-1}^{1}\frac{T_0^2(x)}{\sqrt{1-x^2}}\,\mathrm{d}x = \int_{-1}^{1}\frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x = -\int_{\pi}^{0}\mathrm{d}x = \int_0^{\pi}\mathrm{d}x = \pi\ .$$

Proposition 9.2.1

For $n \geq 1$, T_n has n distinct roots and they are in the interval $[-1,1]$. These roots are

$$r_i = \cos\left(\frac{(2i-1)\pi}{2n}\right) \quad , \quad i = 1,2,3,\ldots,n\ .$$

Proof.
It is easy to verify by substituting in (9.2.1) that the r_i's are n distinct roots of T_n. The r_i's are in the interval $[-1,1]$ since $-1 < \cos(\theta) < 1$ for all $\theta \neq n\pi$. Since T_n is a polynomial of degree n, it has no more roots according to the fundamental theorem of algebra. ∎

Proposition 9.2.2

For $n > 0$, T_n reaches its absolute extrema in the interval $[-1,1]$ at the points

$$s_i = \cos\left(\frac{i\pi}{n}\right) \quad , \quad i = 0,1,2,\ldots,n\ .$$

Moreover, $T_n(s_i) = (-1)^i$.

Proof.
We have $T_n'(x) = \dfrac{n\sin(n\arccos(x))}{\sqrt{1-x^2}}$. Since $T_n'(s_i) = 0$ for $0 < i < n$, the s_i's for $0 < i < n$ are critical points of T_n. Since T_n' is a polynomial of degree $n-1$, it has at most $n-1$ roots. Thus T_n has exactly $n-1$ critical points in $]-1,1[$ given by s_i for $0 < i < n$. These critical points are the only points in the interval $]-1,1[$ where T_n reaches its extrema. A direct computation shows that $T_n(s_i) = \cos(i\pi) = (-1)^i$.

The other two possible points where T_n may reach its extrema are at the endpoints -1 and 1. Since $T_n(-1) = \cos(n\pi) = (-1)^n$ and $T_n(1) = \cos(0) = 1$, the endpoints are also two points where T_n reaches its extrema. ∎

Definition 9.2.3

The **monic Chebyshev polynomials** $\tilde{T}_n$ for $n \geq 0$ are defined by $\tilde{T}_0(x) = 1$ and $\tilde{T}_n(x) = \dfrac{1}{2^{n-1}}T_n(x)$ for $n > 0$.

Remark 9.2.4
The recurrence relation (9.2.2) becomes $\tilde{T}_2(x) = x\tilde{T}_1(x) - \dfrac{1}{2}\tilde{T}_0(x)$ and $\tilde{T}_{n+1}(x) = x\tilde{T}_n(x) - \dfrac{1}{4}\tilde{T}_{n-1}(x)$ for $n > 1$. Using these relations, it follows by induction on the degree of the polynomials that the coefficient of x^n in $\tilde{T}_n(x)$ is 1, hence the name monic given to the polynomials $\tilde{T}_n$.

The roots of $\tilde{T}_n$ are the roots r_i of T_n. $\tilde{T}_n$ and T_n reach their extrema at the same points; namely, at s_i for $0 \leq i \leq n$. However, $\tilde{T}_n(s_i) = \dfrac{(-1)^i}{2^{n-1}}$ for $0 \leq i \leq n$ and $n > 0$. ♠

Proposition 9.2.5

Let $\tilde{\Pi}_n$ be the set of all monic polynomials of degree exactly n. We have

$$\frac{1}{2^{n-1}} = \max_{-1\leq x\leq 1}\left|\tilde{T}_n(x)\right| \leq \max_{-1\leq x\leq 1}|p(x)| \quad , \quad p \in \tilde{\Pi}_n \ .$$

We have equality for $p = \tilde{T}_n$.

Proof.
Suppose that $p \in \tilde{\Pi}_n$ satisfies

$$\max_{-1\leq x\leq 1}|p(x)| \leq \frac{1}{2^{n-1}} \ . \tag{9.2.3}$$

Since p and $\tilde{T}_n$ are both monic of degree n, we have that $q = \tilde{T}_n - p$ is a polynomial of degree at most $n-1$. Moreover, for $0 \leq i \leq n$, we have

$$q(s_i) = \frac{(-1)^i}{2^{n-1}} - p(s_i) \leq 0$$

if i is odd and

$$q(s_i) = \frac{(-1)^i}{2^{n-1}} - p(s_i) \geq 0$$

if i is even because of (9.2.3). By the Intermediate Value Theorem, q has at least one root between s_i and s_{i+1} for $0 \leq i < n$. Hence, q is a polynomial of degree $n-1$ with at least n roots. The only possibility is if $q(x) = 0$ for all $x \in [-1,1]$. ∎

In Item 3 of Remark 6.2.16, we said that Chebyshev points adjusted to the interval $[a,b]$ were the "best" choice of interpolatory points for Lagrange interpolation on the interval $[a,b]$. We now justify this statement.

Without loss of generality, we may assume that $[a,b] = [-1,1]$. If q is the Lagrange interpolating polynomial of a sufficiently differentiable function f on the interval $[-1,1]$ and $0 \leq x_0 < x_1 < \ldots < x_n \leq 1$ are the interpolatory points, then the error is given by

$$|f(x) - q(x)| = \frac{1}{(n+1)!}\, f^{(n+1)}(\xi(x)) \prod_{i=0}^{n}(x - x_i)$$

for $-1 \leq x \leq 1$, where $\xi : [-1,1] \to [-1,1]$. If we assume that $f^{(n+1)}$ is (almost) constant on the interval $[-1,1]$, then we have to minimize $p(x) = \prod_{i=0}^{n}(x - x_i)$ for $-1 \leq x \leq 1$ to minimize the error. Note that p is a monic polynomial of degree $n+1$, hence

$$\frac{1}{2^n} = \max_{-1\leq x\leq 1} \left|\tilde{T}_{n+1}(x)\right| \leq \max_{-1\leq x\leq 1} |p(x)| \quad , \quad p \in \tilde{\Pi}_{n+1} \ ,$$

according to the previous proposition. We have equality when $p = \tilde{T}_{n+1}$; namely, when $x_i = \cos\left(\frac{(2i-1)\pi}{2(n+1)}\right)$ for $1 \leq i \leq n+1$. These are the roots of T_{n+1} which were called Chebychev points in Remark 6.2.16.

Proposition 9.2.6

Let f be a sufficiently differentiable function f defined on the interval $[-1,1]$. If q is the Lagrange interpolating polynomial of f at the Chebyshev points $x_i = \cos\left(\frac{(2i-1)\pi}{2(n+1)}\right)$ for $1 \leq i \leq n+1$, then

$$\max_{-1\leq x\leq 1} |f(x) - q(x)| \leq \frac{1}{2^n(n+1)!} \max_{-1\leq x\leq 1} \left|f^{(n+1)}(x)\right| \ .$$

Proof.
We have

$$f(x) - q(x) = \frac{1}{(n+1)!}\, f^{(n+1)}(\xi(x)) \prod_{i=0}^{n}(x - x_i) = \frac{1}{(n+1)!}\, f^{(n+1)}(\xi(x)) \tilde{T}_{n+1}(x) \ .$$

Hence,

$$\begin{aligned}
\max_{-1\le x\le 1} |f(x) - q(x)| &= \frac{1}{(n+1)!} \max_{-1\le x\le 1} \left| f^{(n+1)}(\xi(x)) \tilde{T}_{n+1}(x) \right| \\
&\le \frac{1}{(n+1)!} \max_{-1\le x\le 1} \left| f^{(n+1)}(\xi(x)) \right| \max_{-1\le x\le 1} \left| \tilde{T}_{n+1}(x) \right| \\
&= \frac{1}{2^n (n+1)!} \max_{-1\le x\le 1} \left| f^{(n+1)}(\xi(x)) \right| ,
\end{aligned}$$

where the last equality comes from Proposition 9.2.5. ∎

9.2.1 How to reduce the Degree of an Interpolating Polynomial with a Minimal Loss of Accuracy

Suppose that $q(x) = \sum_{j=0}^{n} a_j x^j$, where $a_n \neq 0$, is an interpolating polynomial of a fonction f on the interval $[-1,1]$. The goal is to find a polynomial p of degree less than n such that $\max_{-1\le x\le 1} |p(x) - q(x)|$ is as small as possible. If p is of degree less than n, then $(q-p)/a_n$ is a monic polynomial of degree n. Hence,

$$\max_{-1\le x\le 1} |q(x) - p(x)| = |a_n| \max_{-1\le x\le 1} \left| \frac{q(x)-p(x)}{a_n} \right| \ge |a_n| \max_{-1\le x\le 1} \left| \tilde{T}_n(x) \right| = \frac{|a_n|}{2^{n-1}} .$$

We have equality when $\dfrac{q(x)-p(x)}{a_n} = \tilde{T}_n(x)$. We should therefore take $p = q - a_n \tilde{T}_n$. For this choice,

$$\max_{-1\le x\le 1} |q(x) - p(x)| = \frac{|a_n|}{2^{n-1}} .$$

9.3 Exercises

Question 9.1
Consider $f(x) = x - \sin(x)$. Find a small value of n such that the truncation error of the Taylor polynomial $p_n(x)$ of degree n of f about the origin for does not exceeding 10^{-9} for $|x| < 1$.

Question 9.2
Consider $f(x) = \dfrac{1}{1-x}$. Give the Taylor polynomial p_n of degree n of f about the origin as well as the truncation formula for this polynomial. Find a small value of n such that p_n uniformly approximates f to within 10^{-6} on the interval $[0, 1/4]$.

Question 9.3
Find the Taylor polynomial $p_2(x)$ of degree two about the origin for the function $f(x) = e^x \cos(x)$. Approximate $f(0.5)$ using $p_2(x)$. Find an upper bound on the error $|f(0,5) - p_2(0.5)|$ using the truncation formula for the Taylor polynomial of degree two. Compare this bound with the real error.

Chapter 10

Least Square Approximation (in ℓ^2)

Consider the data provided in Figure 10.1

It is not reasonable to use polynomial interpolation at all these points to describe their distribution. There seem to be a pattern in the distribution of these points that polynomial interpolation will completely miss. Instead, it makes more sense to find a curve that "best fit" the data. This curve may not intersect any of the given points but may better describe the distribution of these points; in particular if we want to extrapolate from this set of data.

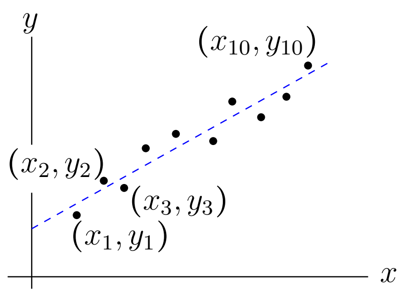


Figure 10.1: Least square approximation of a set of data by a straight line

If we assume that the data $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$ represent a line as in Figure 10.1, the best known methods to fit a line $y = p(x) = ax + b$ through this set of points are the following methods.

Minimax : Find a and b that minimize $I(a, b) = \max_{1 \leq i \leq n} |y_i - (ax_i + b)|$.

Absolute Deviation : Find a and b that minimize $I(a, b) = \sum_{i=1}^{n} |y_i - (ax_i + b)|$.

Least Square : Find a and b that minimize $I(a, b) = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$.

Instead of a straight line, we may have assumed that the data in Figure 10.1 represent a parabola $y = p(x) = ax^2 + bx + c$ or some other functions. We will say more on this later. We have chosen a straight line to illustrate the discrete least square method. The least square method is the most convenient method for the following reasons.

1. We can use elementary calculus to determine the values of a and b that minimize $I(a,b)$ because I is a differentiable function. I is not differentiable everywhere for the other two methods.
2. The method does not assign too much weight to the few points which are far away (vertically) from the straight line that we try to fit.
3. The method is statistically significant.
4. The method is theoretically significant. It is closely related to the notion of L^2 approximation that we will study in the next sections.

10.1 Linear Modeling

Let
$$I(a,b) = \sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)^2 \ .$$
To minimize I, we first find the critical points of I.
$$\frac{\partial I}{\partial a} = -2\sum_{i=1}^{n} x_i \left(y_i - (ax_i + b)\right) = 0$$
and
$$\frac{\partial I}{\partial b} = -2\sum_{i=1}^{n} \left(y_i - (ax_i + b)\right) = 0 \ .$$
This yields the system of linear equation
$$\begin{pmatrix} \displaystyle\sum_{i=1}^{n} x_i^2 & \displaystyle\sum_{i=1}^{n} x_i \\ \displaystyle\sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{i=1}^{n} x_i y_i \\ \displaystyle\sum_{i=1}^{n} y_i \end{pmatrix} \ .$$
The solution of this system is
$$a = \frac{n\displaystyle\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\left(\displaystyle\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2}$$

and

$$b = \frac{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i\right) - \left(\sum_{i=1}^{n} x_i y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2} .$$

Since $I : \mathbb{R}^2 \to [0, \infty[$ is a quadratic polynomial function, its only critical point must be a local and absolute minimum.

10.2 Nonlinear Modelling

We will just present a couple of examples of nonlinear modelling to give a feeling of the subject. We do not plan to investigate this topic very deeply.

If instead of a line, we assume that the data $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$ represent a polynomial $p(x) = \sum_{j=0}^{m} a_j x^j$, then we must minimize

$$I(\mathbf{a}) = \sum_{i=1}^{n} (y_i - p(x_i))^2 = \sum_{i=1}^{n} y_i^2 - 2\sum_{j=0}^{m} a_j \left(\sum_{i=1}^{n} y_i x_i^j\right) + \sum_{j_1=0}^{m}\sum_{j_2=0}^{m} a_{j_1} a_{j_2} \left(\sum_{i=1}^{n} x_i^{j_1+j_2}\right) ,$$

where $\mathbf{a} = \begin{pmatrix} a_0 & a_1 & \ldots & a_n \end{pmatrix}^\top$. The critical points are given by

$$\frac{\partial I}{\partial a_k} = -2\sum_{i=1}^{n} y_i x_i^k + 2\sum_{j=0}^{m} a_j \left(\sum_{i=1}^{n} x_i^{j+k}\right) = 0 \quad \text{for} \quad k = 0, 1, 2, \ldots, m .$$

This yields the system of linear equations $H\mathbf{a} = \mathbf{b}$, where

$$h_{k,j} = \sum_{i=1}^{n} x_i^{j+k} \quad \text{and} \quad b_k = \sum_{i=1}^{n} y_i x_i^k \quad \text{for} \quad k, j = 0, 1, 2, \ldots, m .$$

This is a system of linear equations with $m+1$ equations and $m+1$ unknowns. This system has always a unique solution because the matrix H is a **Hilbert matrix**.

As a final example, suppose that the data $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$ represent an exponential curve $p(x) = be^{ax}$ (or $p(x) = bx^a = be^{a\ln(x)}$). In theory, we have to minimize

$$I(a, b) = \sum_{i=1}^{n} (y_i - be^{ax_i})^2 .$$

However, this is not an easy function to minimize exactly or numerically (the reader should try to do it). So, instead, we minimize

$$J(a, b) = \sum_{i=1}^{n} (\ln(y_i) - \ln(be^{ax_i}))^2 = \sum_{i=1}^{n} (\ln(y_i) - \ln(b) - ax_i)^2 .$$

The unique critical point of J is given by

$$a = \frac{n\sum_{i=1}^{n} x_i \ln(y_i) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} \ln(y_i)\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2}$$

and

$$\ln(b) = \frac{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} \ln(y_i)\right) - \left(\sum_{i=1}^{n} x_i \ln(y_i)\right)\left(\sum_{i=1}^{n} x_i\right)}{n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2} .$$

These values of a and b are not the values of a and b that minimize I. Thus, $y = be^{ax}$ may not be a "best fit" for the set of data as we will expect with the least square method.

10.3 Trigonometric Polynomial Approximation (Real Case)

Suppose that $\{(x_n, y_n) : n = 0, 1, 2, \ldots, 2N-1\}$ are $2N$ data points, where $x_n = \dfrac{n\pi}{N}$ for $n = 0$, 1, 2,..., $2N-1$. We suppose that these data come from (the approximation of) a 2π-periodic function $f : \mathbb{R} \to \mathbb{R}$; namely, $y_n = f(x_n)$ for all n.

Our goal is to find the coefficients a_0, a_1, ..., a_K, b_1, b_2, ..., b_K of the trigonometric polynomial

$$p(x) = a_0 + \sum_{k=1}^{K} a_k \cos(kx) + \sum_{k=1}^{K} b_k \sin(kx) \tag{10.3.1}$$

that minimizes

$$I(a_0, a_1, \ldots, a_K, b_1, b_2, \ldots, b_K) = \sum_{n=0}^{2N-1} (y_n - p(x_n))^2 .$$

We assume that $K \leq N$. The main goal of this section is to efficiently generalize the method of least square approximation presented in Section 10.1. For that, we need a couple of results about the trigonometric polynomial given in (10.3.1).

Lemma 10.3.1

Assume that $r \in \mathbb{Z}$. If r is not a multiple of $2N$, then

$$\sum_{n=0}^{2N-1} \cos(rx_n) = \sum_{n=0}^{2N-1} \sin(rx_n) = 0 . \tag{10.3.2}$$

Moreover, if r is not a multiple of N, then

$$\sum_{n=0}^{2N-1} \cos^2(rx_n) = \sum_{n=0}^{2N-1} \sin^2(rx_n) = N \ . \tag{10.3.3}$$

Proof.
Using complex notation, we have

$$\sum_{n=0}^{2N-1} \cos(rx_n) + i \sum_{n=0}^{2N-1} \sin(rx_n) = \sum_{n=0}^{2N-1} e^{irx_n} = \sum_{n=0}^{2N-1} e^{(rn\pi/N)i} = \left(\frac{1-\left(e^{(r\pi/N)i}\right)^{2N}}{1-e^{(r\pi/N)i}} \right) = 0$$

because $\left(e^{(r\pi/N)i}\right)^{2N} = e^{2r\pi i} = 1$ whatever r, and $e^{(r\pi/N)i} \neq 1$ since r is not a multiple of $2N$. Thus (10.3.2) follows from setting the real and imaginary parts of the right hand side of the previous relation to 0.

If r is not a multiple of N, then $2r$ is not a multiple of $2N$ and it follows from (10.3.2) that $\sum_{n=0}^{2N-1} \cos(2rx_n) = 0$. Hence

$$\sum_{n=0}^{2N-1} \cos^2(rx_n) = \sum_{n=0}^{2N-1} \frac{1}{2}\left(1+\cos(2rx_n)\right) = \frac{1}{2}\sum_{n=0}^{2N-1} 1 + \frac{1}{2}\sum_{n=0}^{2N-1} \cos(2rx_n) = N \ .$$

A similar proof gives $\sum_{n=0}^{2N-1} \sin^2(rx_n) = 1$. ∎

Proposition 10.3.2

Suppose that $K \leq N$. Let $S_K = \{\phi_k\}_{k=0}^{2K}$, where $\phi_0(x) = 1/2$, $\phi_{2k}(x) = \cos(kx)$ and $\phi_{2k-1}(x) = \sin(kx)$ for $k = 1, 2, \ldots, K$. The set S_K is an orthogonal set of functions with respect to the **pseudo scalar product**

$$\langle\langle f, g \rangle\rangle = \sum_{n=0}^{2N-1} f(x_n)g(x_n) \quad , \quad f, g : [0, 2\pi[\to \mathbb{R} \ . \tag{10.3.4}$$

Namely, $\langle\langle \phi_k, \phi_j \rangle\rangle = 0$ for $k \neq j$.

Remark 10.3.3
We call (10.3.4) a **pseudo scalar product** because $\langle\langle f, f \rangle\rangle = 0$ implies only that $f(x_n) = 0$ for $0 \leq n < 2N$. We do not necessarily get $f(x) = 0$ for all $x \in [0, 2\pi[$. All the other properties for a scalar product are satisfied by (10.3.4). We could consider that (10.3.4) defines a scalar product if we consider only functions defined on the set $\{x_0, x_1, \ldots, x_{2N-1}\}$. ♠

Proof.
Using basic trigonometric identities and (10.3.2), we get

$$\begin{aligned}\langle\langle \phi_{2j}, \phi_{2k} \rangle\rangle &= \sum_{n=0}^{2N-1} \cos(jx_n)\cos(kx_n) = \sum_{n=0}^{2N-1} \frac{1}{2}\left(\cos((j+k)x_n) + \cos((j-k)x_n)\right) \\ &= \frac{1}{2}\sum_{n=0}^{2N-1} \cos((j+k)x_n) + \frac{1}{2}\sum_{n=0}^{2N-1} \cos((j-k)x_n) = 0\end{aligned}$$

for $k \neq j$ and $0 < k, j \leq K$ because $0 < |j+k| < 2K$ and $0 < |j-k| < 2K$; so neither $j+k$ nor $j-k$ is a multiple of $2N$.

Similarly,

$$\langle\langle \phi_{2j-1}, \phi_{2k-1} \rangle\rangle = \sum_{n=0}^{2N-1} \sin(jx_n)\sin(kx_n) = 0$$

for $k \neq j$ and $0 < k, j \leq K$, and

$$\langle\langle \phi_{2j}, \phi_{2k-1} \rangle\rangle = \sum_{n=0}^{2N-1} \cos(jx_n)\sin(kx_n) = 0$$

for $0 < k, j \leq K$. Finally,

$$\langle\langle \phi_0, \phi_{2j} \rangle\rangle = \sum_{n=0}^{2N-1} \frac{1}{2}\cos(jx_n) = 0 \quad \text{and} \quad \langle\langle \phi_0, \phi_{2j-1} \rangle\rangle = \sum_{n=0}^{2N-1} \frac{1}{2}\sin(jx_n) = 0$$

according to (10.3.2). ∎

We may now give the formulae to compute the values of a_k for $0 \leq k \leq K$ and b_k for $1 \leq k \leq K$ that minimize $I(a_0, a_1, \ldots, a_K, b_1, b_2, \ldots, b_K)$.

Theorem 10.3.4

Suppose that $K \leq N$. The values of the coefficients a_0, a_1, …, a_K, b_1, b_2, …, b_K of the trigonometric polynomial p defined in (10.3.1) that minimizes

$$I(a_0, a_1, \ldots, a_K, b_1, b_2, \ldots, b_K) = \sum_{n=0}^{2N-1} (y_n - p(x_n))^2$$

are given by

$$a_0 = \frac{1}{2N}\sum_{n=0}^{2N-1} y_n \ , \ a_k = \frac{1}{N}\sum_{n=0}^{2N-1} y_n \cos(kx_n) \quad \text{and} \quad b_k = \frac{1}{N}\sum_{n=0}^{2N-1} y_n \sin(kx_n) \qquad (10.3.5)$$

for $k = 1, 2, \ldots, K$.

Proof.
The idea of the proof is not profound. We just have to find the critical points of I. We get from

$$\frac{\partial I}{\partial b_k} = -2\sum_{n=0}^{2N-1} (y_n - p(x_n))\sin(kx_n) = 0$$

that

$$\sum_{n=0}^{2N-1} y_n \sin(kx_n) = a_0 \sum_{n=0}^{2N-1} \sin(kx_n) + \sum_{j=0}^{K} a_j \left(\sum_{n=0}^{2N-1} \cos(jx_n) \sin(kx_n) \right) + \sum_{j=0}^{K} b_j \left(\sum_{n=0}^{2N-1} \sin(jx_n) \sin(kx_n) \right) .$$

Using (10.3.2) and (10.3.3) of Proposition 10.3.2, we can simplify this expression to get

$$\sum_{n=0}^{2N-1} y_n \sin(kx_n) = b_k \sum_{n=0}^{2N-1} \sin^2(kx_n) = N b_k .$$

Solving for b_k gives the formula in (10.3.5). A very similar reasoning yields the formula for a_k in (10.3.5). For a_0, we have

$$\frac{\partial I}{\partial a_0} = -2 \sum_{n=0}^{2N-1} (y_n - p(x_n)) = 0 .$$

Thus

$$\sum_{n=0}^{2N-1} y_n = a_0 \sum_{n=0}^{2N-1} 1 + \sum_{k=0}^{K} a_k \left(\sum_{n=0}^{2N-1} \cos(kx_n) \right) + \sum_{k=0}^{K} b_k \left(\sum_{n=0}^{2N-1} \sin(kx_n) \right) = 2N a_0 ,$$

where we have used (10.3.2) to get the last equality.

Finally, since $I : \mathbb{R}^{2N+1} \to [0, \infty[$ is a quadratic polynomial function with a single critical point, this critical point is a local and absolute minimum. ∎

Remark 10.3.5
If we use $y_n = f(x_n)$ for all n, we get from Theorem 10.3.4 that

$$a_k = \frac{\langle\langle f(x), \cos(kx) \rangle\rangle}{\langle\langle \cos(kx), \cos(kx) \rangle\rangle} = \left(\sum_{n=0}^{2N-1} f(x_n) \cos(kx_n) \right) \Bigg/ \left(\sum_{n=0}^{2N-1} \cos^2(kx_n) \right)$$

for $k = 0, 1, 2, \ldots, K$ and

$$b_k = \frac{\langle\langle f(x), \sin(kx) \rangle\rangle}{\langle\langle \sin(kx), \sin(kx) \rangle\rangle} == \left(\sum_{n=0}^{2N-1} f(x_n) \sin(kx_n) \right) \Bigg/ \left(\sum_{n=0}^{2N-1} \sin^2(kx_n) \right)$$

for $k = 1, 2, \ldots, K$. ♠

Remark 10.3.6
It was shown in Example 8.1.7 of Chapter 8 that the trigonometric polynomials defined in Proposition 10.3.2 form an orthogonal set of functions in the space of square integrable functions on the interval $[0, 2\pi]$, where the scalar product is defined by the integral $\langle f, g \rangle = \int_0^{2\pi} f(x) g(x) \, dx$ for f and g two square integrable functions. In the space of square integrable functions, the least square problem is to find $a_0, a_1, \ldots, a_K, b_1, b_2, \ldots, b_K$ that

minimize $I(a_0, \ldots, b_K) = \displaystyle\int_{-\pi}^{\pi} (f(x) - p(x))^2 \, \mathrm{d}x$, where p is defined in (10.3.1). The values of a_k and b_k that minimize I are

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) \, \mathrm{d}x \ , \ a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) \, \mathrm{d}x \quad \text{and} \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) \, \mathrm{d}x$$

for $k = 1, 2, \ldots, K$. These coefficients can also be expressed as

$$a_k = \frac{\langle f(x), \cos(kx) \rangle}{\langle \cos(kx), \cos(kx) \rangle} = \left(\int_0^{2\pi} f(x) \cos(kx) \, \mathrm{d}x \right) \Big/ \left(\int_0^{2\pi} \cos^2(kx) \, \mathrm{d}x \right)$$

for $k = 0, 1, 2, \ldots, K$ and

$$b_k = \frac{\langle f(x), \sin(kx) \rangle}{\langle \sin(kx), \sin(kx) \rangle} == \left(\int_0^{2\pi} f(x) \sin(kx) \, \mathrm{d}x \right) \Big/ \left(\int_0^{2\pi} \sin^2(kx) \, \mathrm{d}x \right)$$

for $k = 1, 2, \ldots, K$.

This information is not needed in this section but shows the similarities between the discrete least square method and the least square method in the space of square integrable functions studied in Chapter 8. ♠

10.4 Trigonometric Polynomial Approximation (Complex Case)

Instead of limiting the theory to 2π-periodic real value functions as we have done in the previous section, we now consider 2π-periodic complex valued functions. In the context of complex valued functions, the complex trigonometric polynomials are finite linear combinations of e^{kxi} for $k \in \mathbb{Z}$, where i is the complex number satisfying $i^2 = -1$. In particular, we will consider trigonometric polynomials of the form

$$p(x) = \sum_{k=-K}^{K} r_k e^{kxi} \tag{10.4.1}$$

for $r_k \in \mathbb{C}$.

Suppose that $\{(x_n, y_n) : n = 0, 1, 2, \ldots, N-1\}$ are N data points, where $x_n = \dfrac{2n\pi}{N}$ for $n = 0$, $1, 2, \ldots, N-1$.

We suppose that these data comes from (the approximation of) a 2π-periodic function $f : \mathbb{R} \to \mathbb{C}$; namely, $y_n = f(x_n)$ for all n. Unlike the least square for real trigonometric polynomials of the previous section, we now accept an odd number of data points.

The least square method in the present context is to find the coefficients r_k in (10.4.1) that minimize

$$I(r_{-K}, r_{-K+1}, \ldots, r_K) = \sum_{n=0}^{N-1} |y_n - p(x_n)|^2 \ . \tag{10.4.2}$$

The points x_n are called **sampling points**. The values $f(x_n)$ are called the **sampling values**. $2\pi/N$ is the **sampling interval** and $N/(2\pi)$ is the **sampling frequency**.

In the context of complex valued functions, the pseudo scalar product (10.3.4) becomes

$$\langle\langle f, g\rangle\rangle = \sum_{n=0}^{N-1} f(x_n)\overline{g(x_n)} \quad , \quad f, g : [0, 2\pi] \to \mathbb{C} \; . \tag{10.4.3}$$

The set $S = \{e^{kxi}\}_{k\in\mathbb{Z}}$ is not orthogonal with respect to this pseudo scalar product. However, we have a result similar to orthogonality.

Proposition 10.4.1

The set $S = \{e^{kxi}\}_{k\in\mathbb{Z}}$ satisfies

$$\left\langle\!\left\langle e^{kxi}, e^{jxi}\right\rangle\!\right\rangle = \begin{cases} N & \text{if} \quad k \equiv j \pmod N \\ 0 & \text{if} \quad k \not\equiv j \pmod N \end{cases}$$

Proof.
The proof is similar to the proof of Lemma 10.3.1. For $k \not\equiv j \pmod N$, we have

$$\left\langle\!\left\langle e^{kxi}, e^{jxi}\right\rangle\!\right\rangle = \sum_{n=0}^{N-1} e^{(k-j)x_n i} = \sum_{n=0}^{N-1} e^{(2n\pi(k-j)/N)i} = \frac{1 - \left(e^{(2\pi(k-j)/N)i}\right)^N}{1 - e^{(2\pi(k-j)/N)i}} = 0$$

because $\left(e^{(2\pi(k-j)/N)i}\right)^N = e^{2(k-j)\pi i} = 1$ and $e^{(2\pi(k-j)/N)i} \neq 1$ since $k-j$ is not a multiple of N.

For $k \equiv j \pmod N$, we have

$$\left\langle\!\left\langle e^{kxi}, e^{jxi}\right\rangle\!\right\rangle = \sum_{n=0}^{N-1} e^{(k-j)x_n i} = \sum_{n=0}^{N-1} e^{(2n\pi(k-j)/N)i} = \sum_{n=0}^{N-1} 1 = N$$

because $e^{(2n\pi(k-j)/N)i} = 1$ since $k-j$ is a multiple of N. ∎

Based on our experience in the previous section with 2π-periodic real valued functions, it would tempting to say that the coefficients r_k in (10.4.1) that minimize 10.4.2 are

$$r_k = \frac{\langle\langle f(x), e^{kxi}\rangle\rangle}{\langle\langle e^{kxi}, e^{kxi}\rangle\rangle} = \frac{1}{N}\sum_{n=0}^{N-1} f(x_n)e^{-kx_n i} \quad , \quad -K \le k \le K \; . \tag{10.4.4}$$

Unfortunately, this is only true for $K < N/2$. For $K = N/2$, the coefficients r_{-K} and r_K have to be modified. More precisely,

$$r_{-K} = \frac{1}{2N}\sum_{n=0}^{N-1} f(x_n)e^{-Kx_n i} \quad \text{and} \quad r_K = \frac{1}{2N}\sum_{n=0}^{N-1} f(x_n)e^{Kx_n i} \; . \tag{10.4.5}$$

The reason for this exception comes from the fact that $\left\langle\!\left\langle f(x), e^{Kxi}\right\rangle\!\right\rangle = \left\langle\!\left\langle f(x), e^{-Kxi}\right\rangle\!\right\rangle$ for $K = N/2$ because

$$e^{-Kx_n i} = e^{-n\pi i} = e^{n\pi i} = e^{Kx_n i} = \begin{cases} 1 & n \text{ is even} \\ -1 & n \text{ is odd} \end{cases}$$

Thus

$$\sum_{n=0}^{N-1} f(x_n) e^{Kx_n i} = \sum_{n=0}^{N-1} f(x_n) e^{-Kx_n i} = \sum_{n=0}^{N-1} (-1)^n f(x_n) \ .$$

A proof similar to the proof of Theorem 10.3.4 could be given to show that the coefficients r_k defined above for $K \leq N/2$ minimize (10.4.2) (This is left to the reader). We will proceed differently. The proof that we will give requires some knowledge of Fourier series of complex valued functions in L^2. This was the subject of Chapter 8. We only state the result that is needed from that chapter. This approach has the advantage of linking the discrete least square method above to L^2 approximation in the space of 2π-periodic square integrable functions. However, it has the disadvantage or requesting that f be continuously differentiable. This is an extra condition which is not required to determine the coefficients r_k minimizing (10.4.2).

We saw in Chapter 8 that if f is an L^2-integrable function on $[0, 2\pi]$, then we may write

$$f = \sum_{k\in\mathbb{Z}} A_k e^{kxi} \ ,$$

where the convergence is the L^2 convergence and

$$A_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-kxi} \quad , \quad k \in \mathbb{Z} \ .$$

If f is a 2π-periodic differentiable function, then it can be proved [27] that the series $\sum_{k\in\mathbb{Z}} A_k e^{kxi}$ converges absolutely to $f(x)$ for all x and uniformly on $\mathbb{R}$ to f. In particular, this implies that $\sum_{k=0}^{\infty} A_{\alpha_k} e^{\alpha_k xi}$ converges pointwise to $f(x)$ for any reordering $\{\alpha_k\}_{k=0}^{\infty}$ of the natural numbers[1]. Hence, we can change the order of the summation without changing the limit.

Let

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) e^{-kx_n i} \quad , \quad k \in \mathbb{Z} \ . \tag{10.4.6}$$

Before stating the relation between the a_k's and the A_k's, we show that $\left\langle\!\left\langle f, e^{j_1 xi}\right\rangle\!\right\rangle = \left\langle\!\left\langle f, e^{j_2 xi}\right\rangle\!\right\rangle$ for $j_1 \equiv j_2 \pmod N$. This is at the root of the relation that we will find between the a_k's and A_k's. Suppose that $j_2 = sN + j_1$ with $s \in \mathbb{Z}$, then

$$e^{j_2 x_n i} = e^{(2\pi(sN+j_1)n/N)i} = e^{2\pi sni} e^{(2\pi j_1 n/N)i} = e^{j_1 x_n i}$$

for $n = 0, 1, 2, \ldots, N-1$ because $e^{2\pi sni} = 1$ for all n. Thus

$$\left\langle\!\left\langle f, e^{j_1 xi}\right\rangle\!\right\rangle = \sum_{n=0}^{N-1} f(x_n) e^{j_1 x_n i} = \sum_{n=0}^{N-1} f(x_n) e^{j_2 x_n i} = \left\langle\!\left\langle f, e^{j_2 xi}\right\rangle\!\right\rangle \ .$$

[1]Namely, $k \mapsto \alpha_k$ is an injective and surjective mapping of $\mathbb{Z}$ to itself.

Proposition 10.4.2

If f is a 2π-periodic differentiable function, then

$$a_k = \sum_{j \equiv k \pmod N} A_j \ .$$

Proof.
We have

$$\begin{aligned} a_k &= \frac{1}{N}\sum_{n=0}^{N-1} f(x_n)e^{-kx_n i} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\lim_{J\to\infty}\sum_{j=-J}^{J} A_j e^{jx_n i}\right)e^{-kx_n i} \\ &= \lim_{J\to\infty}\sum_{j=-J}^{J} A_j\left(\frac{1}{N}\sum_{n=0}^{N-1} e^{jx_n i}e^{-kx_n i}\right) = \sum_{j\equiv k \pmod N} A_j \ . \end{aligned}$$

The third equality comes from the absolute convergence of the series $\sum_{k\in\mathbb{Z}} A_k e^{kxi}$ to $f(x)$ for all x. The last equality comes from Proposition 10.4.1 and the absolute convergence of the series $\sum_{k\in\mathbb{Z}} A_k$. ∎

The result of the previous proposition is called **aliasing**.

Theorem 10.4.3

Assume that f is a 2π-periodic differentiable function and that a_k is defined by (10.4.6) for all k.

1. If $K < N/2$, then $I(r_{-K}, r_{-K+1}, \ldots, r_K) \leq I(b_{-K}, b_{-K+1}, \ldots, b_K)$ for all $b_k \in \mathbb{C}$ with $|k| \leq K$ if $r_k = a_k$ for $k) \leq K$.

2. If $K = N/2$, then $I(r_{-K}, r_{-K+1}, \ldots, r_K) \leq I(b_{-K}, b_{-K+1}, \ldots, b_K)$ for all $b_k \in \mathbb{C}$ with $|k| \leq K$ if $r_k = a_k$ for $|k| < K$ and $r_{-K} = r_K = \dfrac{1}{2}a_K$.

Proof.
i) If $K < N/2$ and $b_k = 0$ for $K < |k| \leq N/2$, we have

$$\begin{aligned} \sum_{n=0}^{N-1}\left|f(x_n) - \sum_{k=-K}^{K} b_k e^{kx_n i}\right|^2 &= \sum_{n=0}^{N-1}\left|\left(\sum_{j\in\mathbb{Z}} A_j e^{jx_n i}\right) - \sum_{k=-K}^{K} b_k e^{kx_n i}\right|^2 \\ &= \sum_{n=0}^{N-1}\left|\sum_{-N/2<k\leq N/2}\left(\sum_{j\equiv k \pmod N} A_j\right)e^{kx_n i} - \sum_{k=-K}^{K} b_k e^{kx_n i}\right|^2 \\ &= \sum_{n=0}^{N-1}\left|\sum_{-N/2<k\leq N/2}\left(\left(\sum_{j\equiv k \pmod N} A_j\right) - b_k\right)e^{kx_n i}\right|^2 \end{aligned}$$

$$= \sum_{n=0}^{N-1} \left| \sum_{-N/2<k\leq N/2} (a_k - b_k)\, e^{kx_n i} \right|^2 .$$

To get the first equality, we have used $e^{jx_n i} = e^{kx_n i}$ for $j \equiv k \pmod N$ and the absolute convergence of the series to rearrange the summation. Hence

$$\sum_{n=0}^{N-1} \left| f(x_n) - \sum_{k=-K}^{K} b_k e^{kx_n i} \right|^2 = \sum_{\substack{-N/2<k_1\leq N/2\\ -N/2<k_2\leq N/2}} (a_{k_1} - b_{k_1})\, \overline{(a_{k_2} - b_{k_2})} \underbrace{\left(\sum_{n=0}^{N-1} e^{k_1 x_n i} e^{-k_2 x_n i} \right)}_{= \begin{cases} 0 & \text{if } k_1 \neq k_2 \\ N & \text{if } k_1 = k_2 \end{cases}}$$

$$= N \sum_{-N/2<k\leq N/2} |a_k - b_k|^2$$

because of Lemma 10.4.1. Therefore, the minimum $N \left(\sum_{\substack{-N/2<k<-K\\ K<k\leq N/2}} |a_k|^2 \right)$. is reached at $b_k = a_k$ for $|k| \leq K$.

i) If $K = N/2$, we have as for the case $K < N/2$ above that

$$\sum_{n=0}^{N-1} \left| f(x_n) - \sum_{k=-K}^{K} b_k e^{kx_n i} \right|^2 = \sum_{n=0}^{N-1} \left| \left(\sum_{j\in\mathbb{Z}} A_j e^{jx_n i} \right) - \sum_{k=-K}^{K} b_k e^{kx_n i} \right|^2$$

$$= \sum_{n=0}^{N-1} \left| \sum_{-N/2<k\leq N/2} \left(\sum_{j\equiv k \pmod N} A_j \right) e^{kx_n i} - \sum_{k=-K}^{K} b_k e^{kx_n i} \right|^2$$

$$= \sum_{n=0}^{N-1} \left| \sum_{|k|<N/2} \left(\left(\sum_{j\equiv k \pmod N} A_j \right) - b_k \right) e^{kx_n i} + \left(\left(\sum_{j\equiv K \pmod N} A_j \right) - b_K - b_{-K} \right) e^{Kx_n i} \right|^2 .$$

Hence,

$$\sum_{n=0}^{N-1} \left| f(x_n) - \sum_{k=-K}^{K} b_k e^{kx_n i} \right|^2 = \sum_{n=0}^{N-1} \left| \sum_{|k|<N/2} (a_k - b_k)\, e^{kx_n i} + (a_K - b_K - b_{-K})\, e^{Kx_n i} \right|^2$$

$$= N \left(\sum_{-N/2<k<N/2} |a_k - b_k|^2 + |a_K - b_{-K} - b_K|^2 \right)$$

because of Lemma 10.4.1. Therefore, the minimum is 0 when $b_k = a_k$ for $|k| < K$ and $b_{-K} = b_K = \dfrac{a_K}{2}$.

Note that other choices of b_{-K} and b_K are possible as long as $a_K = b_{-K} + b_K$. ∎

Remark 10.4.4

1. There is another proof for the case $K < N/2$ in Theorem 10.4.3. According to Proposition 10.4.1, the set $S_N = \left\{ e^{kxi} \right\}_{|k|\leq K}$ is a orthogonal set with respect to the pseudo

scalar product (10.4.3). Hence, a theorem similar to Theorem 8.1.5 in Chapter 8 says that $I(r_{-K}, r_{-K+1}, \ldots, r_K) = \langle\!\langle f - p, f - p \rangle\!\rangle$, where p is defined in (10.4.1), reaches its minimum if and only if r_k is given by (10.4.4) for $|k| \leq K$.

2. Since $e^{-Kx_n i} = e^{Kx_n i}$ for all n when $K = N/2$, it follows from the proof of the previous theorem that
$$p(x) = \sum_{-N/2<k\leq N/2} a_k e^{kxi}$$
minimizes $\displaystyle\sum_{n=0}^{N-1} |y_n - p(x_n)|^2$ among all trigonometric polynomials of the form $\displaystyle\sum_{|k|\leq\lfloor N/2\rfloor} a_k e^{kxi}$, where $\lfloor N/2 \rfloor$ is the largest integer less than or equal to $N/2$.

3. Let $p(x) = \displaystyle\sum_{|k|\leq K} r_k e^{kxi}$ be the polynomial given by Theorem 10.4.3 when $K = \lfloor N/2 \rfloor$. Since
$$f(x_n) = \sum_{j\in\mathbb{Z}} A_j e^{jx_n i} = \sum_{-N/2<k\leq N/2} \left(\sum_{k\equiv j \pmod N} A_j \right) e^{kx_n i} = \sum_{-N/2<k\leq N/2} a_k e^{kx_n i} = p(x_n)$$
for $0 \leq n < N$, the polynomial p is an interpolating polynomial of f at $x_0, x_1, \ldots, x_{N-1}$.

4. If f is a 2π-periodic real valued function, then $a_{-k} = \overline{a_k}$ for all k. In particular, $a_0 \in \mathbb{R}$. Hence, for $K < N/2$, we have
$$\begin{aligned} p(x) &= \sum_{k=-K}^{K} a_k e^{kxi} = a_0 + \sum_{k=1}^{K} \left(a_k e^{kxi} + \overline{a_k e^{kxi}} \right) = a_0 + 2\sum_{k=1}^{K} \operatorname{Re}\left(a_k e^{kxi} \right) \\ &= a_0 + 2\sum_{k=1}^{K} \big(\operatorname{Re}(a_k)\cos(kx) - \operatorname{Im}(a_k)\sin(kx) \big) \\ &= \tilde{a}_0 + \sum_{k=1}^{K} \tilde{a}_k \cos(kx) + \sum_{k=1}^{K} \tilde{b}_k \sin(kx) , \end{aligned}$$
where
$$\begin{aligned} \tilde{a}_0 &= a_0 = \frac{1}{N}\sum_{n=0}^{N-1} f(x_n) , \\ \tilde{a}_k &= 2\operatorname{Re}(a_k) = 2\operatorname{Re}\left(\frac{1}{N}\sum_{n=0}^{N-1} f(x_n) e^{-kx_n i} \right) = \frac{2}{N}\sum_{n=0}^{N-1} f(x_n)\cos(kx_n) \end{aligned}$$
and
$$\tilde{b}_k = -2\operatorname{Im}(a_k) = -2\operatorname{Im}\left(\frac{1}{N}\sum_{n=0}^{N-1} f(x_n) e^{-kx_n i} \right) = \frac{2}{N}\sum_{n=0}^{N-1} f(x_n)\sin(kx_n) .$$
Therefore, the real case is a special case of the complex case when $K < N/2$.

♠

Remark 10.4.5
We considered in Example 8.1.6 of Chapter 8 the following least square problem. Let f be a 2π-periodic complex valued functions, find $r_k \in \mathbb{C}$ for $-K \leq k \leq K$ that minimize $I(r_{-K}, r_{-K+1}, \ldots, r_K) = \int_0^{2\pi} (f(x) - p(x))^2 \, dx$, where p is defined in (10.4.1). We showed in Example 8.1.6 that the choice of r_k that minimize I is given by

$$r_k = A_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-kxi} \, dx \quad , \quad k \in \mathbb{Z} \ .$$

It is interesting to approximate the coefficients A_k using a numerical method like the composite trapezoidal rule given in Theorem 12.4.1. Suppose that $g : \mathbb{R} \to \mathbb{C}$ is a 2π-periodic function. If we use the partition of the interval $[0, 2\pi]$ given by $x_n = \dfrac{2n\pi}{N}$ for $0 \leq n \leq N$, we get

$$\int_0^{2\pi} g(x) \, dx \approx \frac{\pi}{N} g(0) + \frac{2\pi}{N} \sum_{n=1}^{N-1} g(x_n) + \frac{\pi}{N} g(2\pi) = \frac{2\pi}{N} g(0) + \frac{2\pi}{N} \sum_{n=1}^{N-1} g(x_n) = \frac{2\pi}{N} \sum_{n=0}^{N-1} g(x_n) \ .$$

In particular, if $g(x) = f(x) e^{-kxi}$ with f a 2π-periodic complex valued function, we get

$$A_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-kxi} \, dx \approx= \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) e^{-kx_n i} = a_k \ .$$

So a_k is approximately A_k. The terms other than A_k in $a_k = \sum_{j \equiv k \pmod{N}} A_j$ are almost negligible. ♠

Example 10.4.6
Find the trigonometric polynomial $p(x) = \sum_{|k| \leq 1} b_k e^{kxi}$ that interpolates $f(x) = \sin^2(x)$ at $x_n = 2n\pi/3$ for $n = 0$, 1 and 2.

According to Item 3 of Remark 10.4.4, the answer is given by Theorem 10.4.3 with $N = 3$ and $K = 1$. Since f is a real valued function, we may used Item 4 of Remark 10.4.4, to write

$$p(x) = a_0 + a_1 \cos(x) + b_1 \sin(x) \ ,$$

where

$$\begin{aligned}
a_0 &= \frac{1}{3} \sum_{n=0}^{2} \sin^2(x_n) = \frac{1}{3} \left(\sin^2(0) + \sin^2\left(\frac{2\pi}{3}\right) + \sin^2\left(\frac{4\pi}{3}\right) \right) = \frac{1}{3} \left(0 + \frac{3}{4} + \frac{3}{4} \right) = \frac{1}{2} \ , \\
a_1 &= \frac{2}{3} \sum_{n=0}^{2} \sin^2(x_n) \cos(x_n) = \frac{2}{3} \left(\sin^2(0)\cos(0) + \sin^2\left(\frac{2\pi}{3}\right) \cos\left(\frac{2\pi}{3}\right) + \sin^2\left(\frac{4\pi}{3}\right) \cos\left(\frac{4\pi}{3}\right) \right) \\
&= \frac{2}{3} \left(0 + \frac{3}{4}\left(\frac{-1}{2}\right) + \frac{3}{4}\left(\frac{-1}{2}\right) \right) = -\frac{1}{2}
\end{aligned}$$

and

$$b_1 = \frac{2}{3}\sum_{n=0}^{2}\sin^2(x_n)\sin(x_n) = \frac{2}{3}\left(\sin^3(0) + \sin^3\left(\frac{2\pi}{3}\right) + \sin^3\left(\frac{4\pi}{3}\right)\right)$$
$$= \frac{2}{3}\left(0 + \frac{3}{4}\left(\frac{\sqrt{3}}{2}\right) + \frac{3}{4}\left(\frac{-\sqrt{3}}{2}\right)\right) = 0 \ .$$

Hence,

$$p(x) = \frac{1}{2} - \frac{1}{2}\cos(x) \ .$$

♣

Remark 10.4.7
Suppose that N is odd and let $K = \lfloor N/2 \rfloor$. Let Π_K be the space of all trigonometric polynomials of the form $p(x) = \sum_{|k|\leq K} b_k e^{kxi}$. Suppose that f is a 2π-periodic continuous function and let $\operatorname{dist}(f, \Pi_K) = \inf_{q\in\Pi_K} \|f - q\|_\infty$, where $\|h\|_\infty = \max_{0\leq x\leq 2\pi} |h(x)|$ for all continuous function $h : [0, 2\pi] \to \mathbb{C}$. If $p(x) = \sum_{|k|\leq K} r_k e^{kxi}$ is the trigonometric polynomial given by Theorem 10.4.3, then one can prove that

$$\|f - p\|_\infty \leq C \operatorname{dist}(f, \Pi_K)$$

for some constant C [10].

Moreover, if f is j-times differentiable with $f^{(j)}$ piecewise continuous, one can prove that $|A_k| = O(|k|^{-j-1})$ and that this implies that $\operatorname{dist}(f, \Pi_K) = O(K^{-j})$ [10]. ♠

10.5 Fast Fourier Transform

As we have seen in the previous section, the main task for the complex case of trigonometric polynomial approximation was to compute the coefficients a_k defined in (10.4.6); namely,

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} f(x_n) e^{-2\pi kni/N}$$

for $k \in \mathbb{Z}$, where $f : \mathbb{R} \to \mathbb{C}$ is a 2π-periodic function. We present in this section fast algorithms to compute these coefficients. The Fast Fourier Transform algorithms that we will introduce have many other applications.

Definition 10.5.1

Let Π_N be the space of all periodic functions $\mathbf{z} : \mathbb{Z} \to \mathbb{C}$ of period N and let ω_N be the N^{th} root of unity defined by $\omega_N = e^{2\pi i/N}$. The **Discrete Fourier Transform** is the mapping $\mathcal{F}_N : \Pi_N \to \Pi_N$ such that $\mathbf{y} = \mathcal{F}_N \mathbf{x}$ is defined by

$$\mathbf{y}(n) \equiv \frac{1}{N} \sum_{k=0}^{N-1} \omega_N^{-nk} \mathbf{x}(k)$$

for $n \in \mathbb{Z}$.

The Discrete Fourier Transform has the following property.

Proposition 10.5.2

$\mathcal{F}_N : \Pi_N \to \Pi_N$ is one-to-one and onto. The inverse of $\mathcal{F}_N$ is the mapping $\mathcal{F}_N^{-1} : \Pi_N \to \Pi_N$ such that $\mathbf{x} = \mathcal{F}_N^{-1}\mathbf{y}$ is defined by

$$\mathbf{x}(n) \equiv \sum_{k=0}^{N-1} \omega_N^{nk} \mathbf{y}(k)$$

for $n \in \mathbb{Z}$.

The functions $\mathbf{x} : \Pi_N \to \Pi_N$ and $\mathbf{y} : \Pi_N \to \Pi_N$ could also be written as the infinite sequences $\{x_n\}_{n\in\mathbb{Z}}$ and $\{y_n\}_{n\in\mathbb{Z}}$ respectively. We do not use this notation to avoid complicated indices in the formulae that we will introduce later.

In this subsection, we will develop some **Fast Fourier Transform** algorithms to compute the Discrete Fourier Transform on Π_N. There are many Fast Fourier Transform algorithms; one for each integer decomposition of N. The Fast Fourier Transform algorithms that we give below is based on the work of Cooley and Tukey [11]. The Fast Fourier Transform is used in signal processing, image compression, ... Fast Poisson Solver is a technique to solve some types of partial differential equations using the Fast Fourier Transform. We will not cover any of these applications in this book. Henrici [16] has a nice overview of the applications of the Fast Fourier Transforms. Another good starting reference for the applications of the Fast Fourier Transforms is Strang [30].

To simplify the notation, we consider

$$\mathcal{F}_N^* \mathbf{x} \equiv N \mathcal{F}_N \mathbf{x}$$

for $\mathbf{x} \in \Pi_N$. Hence, $\mathbf{y}^* = \mathcal{F}_N^* \mathbf{x}$ is given by

$$\mathbf{y}^*(n) \equiv \sum_{k=0}^{N-1} \omega_N^{-nk} \mathbf{x}(k)$$

for $n \in \mathbb{Z}$. In many books, $\mathcal{F}_N^*$ is used as the definition of the Discrete Fourier Transform.

The idea behind the Fast Fourier Transforms is to construct a sequence $\tilde{\mathbf{y}}_m$, $\tilde{\mathbf{y}}_{m-1}$, ..., $\tilde{\mathbf{y}}_0$ of functions from $\mathbb{Z} \times \mathbb{Z}$ into $\mathbb{R}$ such that $\tilde{\mathbf{y}}_{j-1}$ is obtained from $\tilde{\mathbf{y}}_j$ for $j = m, m-1, \ldots, 2, 1$. Moreover, $\tilde{\mathbf{y}}_m(\cdot, 0)$ is $\mathbf{x}$ and $\tilde{\mathbf{y}}_0(0, \cdot)$ is $\mathbf{y}^* = \mathcal{F}_N^* \mathbf{x}$.

Definition 10.5.3

1. Given $\mathbf{x} \in \Pi_N$, if $N = AB$ with A and B two integers, we define $\mathbf{x}_{A,a} \in \Pi_B$ with $a \in \mathbb{N}$ by
$$\mathbf{x}_{A,a}(n) = \mathbf{x}(a + An)$$
for $n \in \mathbb{Z}$.

2. Suppose that $N = P_1 P_2 P_3 \ldots P_m$ and let $N = A_k P_k B_k$, where $A_k = P_1 P_2 \ldots P_{k-1}$ and $B_k = P_{k+1} P_{k+2} \ldots P_m$. If $k = 0$, we set $P_k = 1$ and $A_k = 1$. If $k = m$, we set $B_k = 1$.

   For each $q \in \mathbb{Z}$, we define the function $\mathbf{y}^*_{A_k P_k, q} \in \Pi_{B_k}$ by
$$\mathbf{y}^*_{A_k P_k, q} = \mathcal{F}^*_{B_k} \mathbf{x}_{A_k P_k, q} \ .$$
   Namely,
$$\mathbf{y}^*_{A_k P_k, q}(b) = \sum_{s=0}^{B_k - 1} x(q + A_k P_k s) \omega_{B_k}^{-ns} \bigg|_{n=b} = \sum_{s=0}^{B_k - 1} \mathbf{x}(q + A_k P_k s) \omega_{B_k}^{-sb} \tag{10.5.1}$$
   for $b \in \mathbb{Z}$.

3. $\mathbf{y}^*_{A_k P_k, q}$ can be used to define the function
$$\begin{aligned} \tilde{\mathbf{y}}_k : \mathbb{Z} \times \mathbb{Z} &\to \mathbb{R} \\ (q, b) &\mapsto \mathbf{y}^*_{A_k P_k, q}(b) \end{aligned}$$
   The function $\tilde{\mathbf{y}}_k$ is of period B_k in its second variable and of period N in its first variable.

Remark 10.5.4

1. For $k = m$, we have the special case $A_k = P_1 P_2 \ldots P_{m-1}$, $P_k = P_m$ and $B_k = 1$ in (10.5.1). Thus,
$$\mathbf{y}^*_{A_m P_m, q}(0) = \mathcal{F}^*_1 \mathbf{x}_{N,q}(n)\bigg|_{n=0} = \sum_{s=0}^{0} \mathbf{x}(q + Ns) \omega_1^{-ns}\bigg|_{n=0} = \mathbf{x}(q)$$
   for $q \in \mathbb{N}$. Namely, $\tilde{\mathbf{y}}_m(q, 0) = \mathbf{x}(q)$ for $q \in \mathbb{N}$.

2. For $k = 0$, we have $A_k = P_k = 1$ and $B_k = N$ in (10.5.1). Thus,
$$\mathbf{y}^*_{A_0 P_0, 0}(b) = \mathcal{F}^*_N \mathbf{x}_{1,0}(n)\bigg|_{n=b} = \sum_{s=0}^{N-1} \mathbf{x}(s) \omega_N^{-ns}\bigg|_{n=b} = \mathbf{y}^*(b) = \mathcal{F}^*_N \mathbf{x}(n)\bigg|_{n=b}$$
   for $b \in \mathbb{N}$. Namely, $\tilde{\mathbf{y}}_0(0, b) = \mathcal{F}^*_N \mathbf{x}(b)$ for $b \in \mathbb{N}$.

♠

The next proposition justifies the method to compute $\tilde{\mathbf{y}}_j$ from $\tilde{\mathbf{y}}_{j-1}$ that will be introduced later.

Proposition 10.5.5

$$\sum_{s=0}^{P_k-1} \tilde{\mathbf{y}}_k(a + A_k s, b)\omega_{P_kB_k}^{-s(b+B_kp)} = \tilde{\mathbf{y}}_{k-1}(a, b + B_kp)$$

for a, b and p in $\mathbb{N}$.

Proof.
We have

$$\begin{aligned}
\sum_{s=0}^{P_k-1} \mathcal{F}_{B_k}^{*}\mathbf{x}_{A_kP_k,a+A_ks}(b)\omega_{P_kB_k}^{-s(b+B_kp)} &= \sum_{s=0}^{P_k-1}\left(\sum_{r=0}^{B_k-1}\mathbf{x}(a + A_ks + A_kP_kr)\omega_{B_k}^{-br}\right)\omega_{P_kB_k}^{-s(b+B_kp)} \\
&= \sum_{s=0}^{P_k-1}\sum_{r=0}^{B_k-1}\mathbf{x}(a + A_k(s + P_kr))\omega_{B_kP_k}^{-brP_k}\omega_{P_kB_k}^{-s(b+B_kp)} \\
&= \sum_{s=0}^{P_k-1}\sum_{r=0}^{B_k-1}\mathbf{x}(a + A_k(s + P_kr))\omega_{B_kP_k}^{-(s+rP_k)(b+B_kp)} \\
&= \mathcal{F}_{P_kB_k}^{*}\mathbf{x}_{A_k,a}(b + B_kp)
\end{aligned}$$

because $\omega_{P_kB_k}^{-mP_kB_k} = 1^{-m} = 1$ for all $m \in \mathbb{Z}$. Thus

$$\sum_{s=0}^{P_k-1} \tilde{\mathbf{y}}_k(a + A_k s, b)\omega_{P_kB_k}^{-s(b+B_kp)} = \tilde{\mathbf{y}}_{k-1}(a, b + B_kp) \ . \qquad \blacksquare$$

A consequence of Proposition 10.5.5 for $0 \leq a < A_k$, $0 \leq b < B_k$ and $0 \leq p < P_k$ is the following Fast Fourier Transform algorithms.

Code 10.5.6 (Fast Fourier Transform)

To compute the Fast Fourier Transform of $\mathbf{x}$ in Π_N. We assume that $N = P_1P_2\ldots P_m$.
Input: The vector $(P_1, P_2, \ldots P_m)$ and the column vector $\mathbf{x}$ which are respectively denoted `MP` and `X` in the code below. Since $\mathbf{x}$ is N-periodic, only the components $\mathbf{x}(j)$ for $0 \leq j < N$ are needed.
Output: The Fast Fourier Transform $\mathbf{y}^* = \mathcal{F}_N^*\mathbf{x}$ which is denoted `Z` in the code below. As for the input, only the components $\mathbf{y}^*(j)$ for $0 \leq j < N$ are returned because of the periodicity of $\mathbf{y}^*$.

```
function Z = FFT(X,MP)
  m = length(MP);

  % For k = m, Y = X
  Y(:,1) = X;
  for k = m:-1:1
    if k < m
      B = prod(MP(k+1:m));
    else
      B = 1;
```

```
    end
    P = MP(k);
    if ( k > 1 )
      A = prod(MP(1:k-1));
    else
      A = 1;
    end

    x = 1;
    omega = exp(-(2*pi*i)/(P*B))
    p = [0:P-1]';
    for b = 0:B-1
      omega_p =  omega.^(b+B*p);
      for a = 0:A-1

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Do not forget that the column vectors in Matlab are
        % indexed (1,1),(2,1), ...
        Ytempo = ones(P,1)*Y(1+b+B*a+A*B*(P-1),1);
        for s = P-2:-1:0
          Ytempo = Ytempo.*omega_p + Y(1+b+B*a+A*B*s,1);
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        % We transfer the information to Z for the next value of m.
        for s = 0:P-1
          Z(1+b+B*s+B*P*a,1) = Ytempo(s+1);
        end
      end
    end

    % The value of Y for the next value of m.
    Y = Z;
  end

  % The final result
  % For m = 1, Z is the Fast Fourier Transform of X.
end
```

Remark 10.5.7

1. The portion of code between the two lines of %'s is

$$\sum_{s=0}^{P_k-1} \tilde{\mathbf{y}}_k(a + A_k s, b)\omega_{P_k B_k}^{-s(b+B_k p)} \tag{10.5.2}$$

computed for all the values of p at the same time using Matlab matrix operations. We repeat this operation for $0 \leq a < A_k$ and $0 \leq b < B_k$ to get the full vector $\tilde{\mathbf{y}}_{k-1}(a, b+B_k p)$.

We have also used the nested form of the polynomial in $\omega_{P_kB_k}^{-(b+B_kp)}$ to evaluate the expression (10.5.2) above. To formulate (10.5.2) in MATLAB, we have to note that

- `Z(1+b+B*p+B*P*a,1)` represents $\tilde{\mathbf{y}}_{k-1}(a, b + B_k p)$ and
- `Y(1+b+B*a+A*B*s,1) = Y(1+b+B*(a+A*s),1)` represents $\tilde{\mathbf{y}}_k(a + A_k s, b)$

for $0 \le a < A_k$, $0 \le b < B_k$ and $0 \le p, s < P_k$.

2. About $N(P_1 + P_2 + \ldots + P_m) = N\log(N)$ operations are needed in the code above to compute $\mathcal{F}_N^*\mathbf{x}$. The evaluation of omega and copying data has been ignored when computing the number of operations. The number of operations to compute $\mathcal{F}_N^*\mathbf{x}$ directly from the definition is about N^2. This is much larger than $N(P_1+P_2+\ldots+P_m)$ in general.

3. When $N = 2^m$, an efficient Fast Fourier Transform algorithm can be developed. It is probably the must often used Fast Fourier Transform algorithm.

$$
\begin{aligned}
\mathcal{F}_N^*\mathbf{x}(n) &= \sum_{k=0}^{N-1} \omega_N^{-nk}\mathbf{x}(k) = \sum_{k=0}^{2^m-1} \omega_{2^m}^{-nk}\mathbf{x}(k) \\
&= \sum_{k=0}^{2^{m-1}-1} \omega_{2^m}^{-n(2k)}\mathbf{x}(2k) + \sum_{k=0}^{2^{m-1}-1} \omega_{2^m}^{-n(2k+1)}\mathbf{x}(2k+1) \\
&= \sum_{k=0}^{2^{m-1}-1} \omega_{2^{m-1}}^{-nk}\mathbf{x}(2k) + \omega_{2^m}^{-n} \sum_{k=0}^{2^{m-1}-1} \omega_{2^{m-1}}^{-nk}\mathbf{x}(2k+1) \\
&= \mathcal{F}_{2^{m-1}}^*\mathbf{x}_e(n) + \omega_{2^m}^{-n}\,\mathcal{F}_{2^{m-1}}^*\mathbf{x}_o(n)\ ,
\end{aligned}
$$

where $\mathbf{x}_e(k) = \mathbf{x}(2k)$ and $\mathbf{x}_o(k) = \mathbf{x}(1+2k)$ for $k \in \mathbb{N}$. We have used the relation $\omega_{2^m}^{2n} = \omega_{2^{m-1}}^{n}$ to get the fourth equality. Moreover, since $\omega_{2^m}^{-j-2^{m-1}} = -\omega_{2^m}^{-j}$ for $0 \le j < 2^{m-1}$, we get

$$
\mathcal{F}_N^*\mathbf{x}(n) = \mathcal{F}_{2^{m-1}}^*\mathbf{x}_e(n) + \omega_{2^m}^{-n}\,\mathcal{F}_{2^{m-1}}^*\mathbf{x}_o(n)
$$

and

$$
\begin{aligned}
\mathcal{F}_N^*\mathbf{x}\left(n + 2^{m-1}\right) &= \mathcal{F}_{2^{m-1}}^*\mathbf{x}_e\left(n + 2^{m-1}\right) + \omega_{2^m}^{-(n+2^{m-1})}\,\mathcal{F}_{2^{m-1}}^*\mathbf{x}_o\left(n + 2^{m-1}\right) \\
&= \mathcal{F}_{2^{m-1}}^*\mathbf{x}_e(n) - \omega_{2^m}^{-n}\,\mathcal{F}_{2^{m-1}}^*\mathbf{x}_o(n)
\end{aligned}
$$

for $0 \le n < 2^{m-1}$. The last equality, comes from the fact that $\mathcal{F}_{2^{m-1}}^*\mathbf{x}_e$ and $\mathcal{F}_{2^{m-1}}^*\mathbf{x}_o$ are of period 2^{m-1} because $\mathbf{x}_e$ and $\mathbf{x}_o$ are of period 2^{m-1}. This gives the following simple algorithm.

Code 10.5.8 (Fast Fourier Transform)

To compute the Fast Fourier Transform of $\mathbf{x}$ in Π_N, where $N = 2^m$.
Input: The column vector $\mathbf{x}$ (denoted X in the code below). Since $\mathbf{x}$ is N-periodic, only the components $\mathbf{x}(j)$ for $0 \le j < N$ are needed.

Output: The Fast Fourier Transform $\mathbf{y}^* = \mathcal{F}_N^* \mathbf{x}$ (denoted Z in the code below). As for the input, only the components $\mathbf{z}^*(j)$ for $0 \le j < N$ are returned because of the periodicity of $\mathbf{z}^*$.

```
function Z = recursiveFFT(X)
  N = length(X);
  if N == 1
    Z = X;
  else
    % We compute the Fourier Transform for x_{2k}
    Y1 = recursiveFFT( X(1:2:N) );

    % We compute the Fast Fourier Transform for x_{1+2k}
    Y2 = recursiveFFT( X(2:2:N) );

    a = [0:N/2-1]';
    Y3 = Y2.*exp(-(2*pi*i)*a/N);
    Z = [Y1+Y3 ; Y1-Y3];
  end
end
```

♠

Chapter 11

Iterative Methods to Approximate Eigenvalues

11.1 Background in Linear Algebra

Before developing methods to approximate eigenvalues of linear operators, we need to review some basic concepts of Linear Algebra. We also present some theoretical results about the location of the eigenvalues. In Section 3.1 of Chapter 3, we have already given some properties of the eigenvalues of a linear operator. We refer in particular to Definition 3.1.10, Theorem 3.1.11 and Remarks 3.1.9 and 3.1.12,

11.1.1 Orthogonality

To really understand the Gram-Schmidt orthogonalization process that we give in Definition 11.6.1, we need to review some useful concepts including projections on subspace of $\mathbb{R}^n$.

Definition 11.1.1

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$. A set of non-null vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is **orthogonal** if $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ for $i \neq j$.

Proposition 11.1.2

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$ and $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ be a set of orthogonal vectors. Then S is a set of linearly independent vectors.

Proof.
Suppose that $\mathbf{0} = \sum_{j=1}^{k} a_j \mathbf{v}_j$. We have

$$0 = \langle \mathbf{v}_i, \mathbf{0} \rangle = \left\langle \mathbf{v}_i, \sum_{j=1}^{k} a_j \mathbf{v}_j \right\rangle = \sum_{j=1}^{k} a_j \underbrace{\langle \mathbf{v}_i, \mathbf{v}_j \rangle}_{=0 \text{ for } j \neq i} = a_i \langle \mathbf{v}_i, \mathbf{v}_i \rangle$$

for $i = 1, 2, \ldots, k$. Since $\langle \mathbf{v}_i, \mathbf{v}_i \rangle \neq 0$ because $\mathbf{v}_i \neq \mathbf{0}$, we get $a_i = 0$. ∎

Definition 11.1.3

Let $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ be a set of vectors in $\mathbb{R}^n$. The **span** of S is the subspace, denoted $\text{span}(S)$, defined by

$$\text{span}(S) = \left\{ \sum_{j=1}^{k} a_j \mathbf{v}_j : a_j \in \mathbb{R} \right\} \quad .$$

Definition 11.1.4

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$. Let $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ be an orthogonal set of $\mathbb{R}^n$ and $V = \text{span}(S)$. The **orthogonal projection** P on V is the mapping defined by $P(\mathbf{x}) = \sum_{j=1}^{k} a_j \mathbf{v}_j$, where $a_j = \dfrac{\langle \mathbf{v}_j, \mathbf{x} \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle}$.

Proposition 11.1.5

The orthogonal project P defined in Definition 11.1.4 is a linear mapping such that $\mathbf{x} - P(\mathbf{x}) \perp V$ for all $\mathbf{x} \in \mathbb{R}^n$; namely, $\langle \mathbf{v}, \mathbf{x} - P(\mathbf{x}) \rangle = 0$ for all $\mathbf{v} \in V$.

Proof.
That P is a linear mapping is a consequence of the linearity in the second component of the scalar product. We leave it to the reader to verify that $P(a\mathbf{x} + b\mathbf{y}) = aP(\mathbf{x}) + bP(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $a, b \in \mathbb{R}$.

Choose $\mathbf{x} \in \mathbb{R}^n$. To prove that $\mathbf{x} - P(\mathbf{x}) \perp V$, it suffices to prove that $\langle \mathbf{v}_i, \mathbf{x} - P(\mathbf{x}) \rangle = 0$ for $1 \leq i \leq k$.

Let $P(\mathbf{x}) = \sum_{j=1}^{k} a_j \mathbf{v}_j$ with $a_j = \dfrac{\langle \mathbf{v}_j, \mathbf{x} \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle}$. We have

$$\begin{aligned} \langle \mathbf{v}_i, \mathbf{x} - P(\mathbf{x}) \rangle &= \left\langle \mathbf{v}_i, \mathbf{x} - \sum_{j=1}^{k} a_j \mathbf{v}_j \right\rangle = \langle \mathbf{v}_i, \mathbf{x} \rangle - \sum_{j=1}^{k} a_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \\ &= \langle \mathbf{v}_i, \mathbf{x} \rangle - a_i \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 0 \end{aligned}$$

for $1 \leq i \leq k$ by definition of a_i. ∎

We illustrate in Figure 11.1 an orthogonal projection P on a subspace V of $\mathbb{R}^3$ generated by two orthogonal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

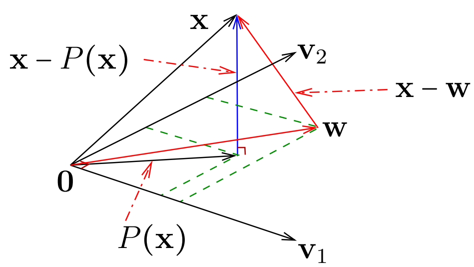


Figure 11.1: Sketch of the image of a vector $\mathbf{x}$ by the orthogonal projection P on a subspace V of $\mathbb{R}^3$ generated by two orthogonal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

Proposition 11.1.6

The orthogonal project P defined in Definition 11.1.4 has the following property.

$$\|\mathbf{x} - P(\mathbf{x})\| < \|\mathbf{x} - \mathbf{w}\|$$

for all $\mathbf{w} \in V$ such that $\mathbf{w} \neq P(\mathbf{x})$. The norm of a vector $\mathbf{y} \in \mathbb{R}^n$ is obviously defined by $\|\mathbf{y}\| = \sqrt{\langle \mathbf{y}, \mathbf{y} \rangle}$.

Proof.
The conclusion of the proposition is illustrated in Figure 11.1.

Suppose that $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$. Since $P(\mathbf{x}) - \mathbf{w} \in V$, we have from the previous proposition that $\langle P(\mathbf{x}) - \mathbf{w}, \mathbf{x} - P(\mathbf{x}) \rangle = 0$. Hence

$$\begin{aligned}
\|\mathbf{x} - \mathbf{w}\|^2 &= \langle \mathbf{x} - \mathbf{w}, \mathbf{x} - \mathbf{w} \rangle \\
&= \langle (\mathbf{x} - P(\mathbf{x})) + (P(\mathbf{x}) - \mathbf{w}), (\mathbf{x} - P(\mathbf{x})) + (P(\mathbf{x}) - \mathbf{w}) \rangle \\
&= \langle \mathbf{x} - P(\mathbf{x}), \mathbf{x} - P(\mathbf{x}) \rangle + \langle P(\mathbf{x}) - \mathbf{w}, \mathbf{x} - P(\mathbf{x}) \rangle + \langle \mathbf{x} - P(\mathbf{x}), P(\mathbf{x}) - \mathbf{w} \rangle \\
&\qquad + \langle P(\mathbf{x}) - \mathbf{w}, P(\mathbf{x}) - \mathbf{w} \rangle \\
&= \langle \mathbf{x} - P(\mathbf{x}), \mathbf{x} - P(\mathbf{x}) \rangle + \langle P(\mathbf{x}) - \mathbf{w}, P(\mathbf{x}) - \mathbf{w} \rangle \\
&= \|\mathbf{x} - P(\mathbf{x})\|^2 + \|P(\mathbf{x}) - \mathbf{w}\|^2 > \|\mathbf{x} - P(\mathbf{x})\|^2
\end{aligned}$$

unless $\|P(\mathbf{x}) - \mathbf{w}\| = 0$; namely, unless $\mathbf{w} = P(\mathbf{x})$. ∎

Remark 11.1.7
The previous proposition shows that the orthogonal projection P defined in Definition 11.1.4 is independent of the orthogonal set S generating the subspace V. ♠

Definition 11.1.8

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$ and V be a subspace of $\mathbb{R}^n$. A set of non-null vector $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is **orthogonal basis** of V if it is a basis of V and it is orthogonal. It is an **orthonormal basis** of V if it an orthogonal basis of V such that $\|\mathbf{v}_j\| = 1$ for $1 \leq j \leq k$.

Proposition 11.1.9

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$. If $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is an orthogonal basis of a subspace V of $\mathbb{R}^n$, then $\mathbf{v} = \displaystyle\sum_{j=1}^{k} a_j \mathbf{v}_j$ with $a_j = \dfrac{\langle \mathbf{v}_j, \mathbf{x} \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle}$ for all $\mathbf{v} \in V$.

Proof.

Given $\mathbf{v} \in V$, since S is a basis of V, we have $\mathbf{v} = \displaystyle\sum_{j=1}^{k} a_j \mathbf{v}_j$ for some $a_j \in \mathbb{R}$. Hence

$$\langle \mathbf{v}_i, \mathbf{v} \rangle = \left\langle \mathbf{v}_i, \sum_{j=1}^{k} a_j \mathbf{v}_j \right\rangle = \sum_{j=1}^{k} a_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle = a_i \langle \mathbf{v}_i, \mathbf{v}_i \rangle$$

for $1 \leq i \leq k$. Thus, $a_i = \dfrac{\langle \mathbf{v}_i, \mathbf{x} \rangle}{\langle \mathbf{v}_i, \mathbf{v}_i \rangle}$ for $1 \leq i \leq k$. ∎

11.1.2 Self-adjoint and Unitary Operators

Let V be a vector space over the complex numbers. A linear functional L on V is a linear mapping from V to $\mathbb{C}$. The vector space of all linear functional on V is denoted V^*. It is called the **dual** of V.

Theorem 11.1.10 (Reisz)

Let V be a vector space over the complex numbers and $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C}$ be an hermitian product. Given a linear functional L on V, there exist a unique $\mathbf{w} \in V$ such that $L(\mathbf{v}) = \langle \mathbf{v}, \mathbf{w} \rangle$ for all $\mathbf{v} \in V$. The mapping $\mathbf{w} \mapsto \langle \cdot, \mathbf{w} \rangle$ is a **conjugate-linear isomorphism** from V to V^* (because $\lambda \mathbf{w} \mapsto \overline{\lambda} \langle \cdot, \mathbf{w} \rangle$).

Corollary 11.1.11

Let V be a vector space over the complex numbers and $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C}$ be an hermitian product. Suppose that A is a linear mapping from V into itself. There exists a unique linear mapping B from V into itself such that $\langle A\mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{v}, B\mathbf{w} \rangle$ for all $\mathbf{v}$ and $\mathbf{w}$ in V.

Definition 11.1.12

The linear mapping B in Corollary 11.1.11 is called the **adjoint** of A and is denoted A^*. If $V = \mathbb{C}^n$ and $\langle\cdot,\cdot\rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ is the standard hermitian product, then this definition of adjoint corresponds to the definition of adjoint for a matrix. Namely, $A^* = \overline{A}^\top$.
We say that A is **hermitian** or **self-adjoint** if $A = A^*$.

Definition 11.1.13

Let V be a vector space over the complex numbers and $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{C}$ be an hermitian product. A linear mapping $A : V \to V$ is **(complex) unitary** if $\langle A\mathbf{v}, A\mathbf{w}\rangle = \langle \mathbf{v}, \mathbf{w}\rangle$ for all $\mathbf{v}$ and $\mathbf{w}$ in V.

Theorem 11.1.14

Let V be a vector space over the complex numbers and $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{C}$ be an hermitian product. Let $\|\cdot\| : V \to [0,\infty[$ be the norm induced by the scalar product on V; namely, $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle}$ for all $\mathbf{v} \in V$. Let $A : V \to V$ be a linear mapping. The following statement are equivalent:

1. A is complex unitary.
2. $\|A\mathbf{v}\| = \|\mathbf{v}\|$ for all $\mathbf{v} \in V$.
3. $A^* A = A A^* = \mathrm{Id}$.

11.1.3 Symmetric and Orthogonal Operators

The notion of Hermitian and unitary Operators can be restricted to vector spaces over the real numbers. To do this, we need the following theorem.

Theorem 11.1.15

Let V be a vector space over the real numbers and $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{R}$ be a scalar product. Suppose that $A : V \to V$ is a linear mapping. Then there exists a unique linear mapping $B : V \to V$ such that $\langle A\mathbf{v}, \mathbf{w}\rangle = \langle \mathbf{v}, B\mathbf{w}\rangle$ for all $\mathbf{v}$ and $\mathbf{w}$ in V.

Definition 11.1.16

The linear mapping B in Corollary 11.1.15 is called the **transpose** of A and is denoted $A^\top$. If $V = \mathbb{R}^n$ and $\langle\cdot,\cdot\rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is the standard scalar product, then this definition of transpose corresponds to the definition of transpose for a matrix. We say that A is **symmetric** if $A = A^\top$.

Definition 11.1.17

Let V be a vector space over the real numbers and $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ be a scalar product. A linear mapping $A : V \to V$ is **real unitary** or **orthogonal** if $\langle A\mathbf{v}, A\mathbf{w} \rangle = \langle \mathbf{v}, \mathbf{w} \rangle$ for all $\mathbf{v}$ and $\mathbf{w}$ in V.

Theorem 11.1.18

Let V be a vector space over the real numbers and $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ be a scalar product. Let $\| \cdot \| : V \to [0, \infty[$ be the norm induced by the scalar product on V; namely, $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$. Let $A : V \to V$ be a linear mapping. The following statement are equivalent:

1. A is orthogonal.
2. $\|A\mathbf{v}\| = \|\mathbf{v}\|$ for all $\mathbf{v} \in V$.
3. $A^\top A = A A^\top = \mathrm{Id}$.

11.1.4 Triangular and Diagonal Matrices

Definition 11.1.19

Two $n \times n$ matrices A and B are **similar** if there exists an invertible $n \times n$ matrix N such that $B = N^{-1} A N$.

Theorem 11.1.20

Let A and B be two similar $n \times n$ matrices as defined in the previous definition. Then A and B have the same characteristic polynomial. In particular, $\mathbf{x}$ is an eigenvector of A associated to the eigenvalue λ if and only if $N^{-1}\mathbf{x}$ is an eigenvector of B associated to the eigenvalue λ.

Theorem 11.1.21 (Schur Form and decomposition)

Let A be an $n \times n$ matrix with entries in $\mathbb{C}$. Then there exists an unitary matrix U such that $U^* A U$ is upper-triangular. We say that A is **unitary similar** to an upper-triangular matrix.

Theorem 11.1.22

Let V be a vector space over the real numbers and $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ be a scalar product. Suppose that $A : V \to V$ is a symmetric linear mapping. Then there exists an orthogonal basis of V consisting of eigenvectors of A. In particular, all the eigenvalues of A are real.

Corollary 11.1.23

Let A be a $n \times n$ symmetric matrix with entries in $\mathbb{R}$ and $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be the standard scalar product. Then there exists a real unitary matrix U such that $U^\top A U$ is diagonal. If $\mathcal{E}$ is the canonical basis in $\mathbb{R}^n$ and $\mathcal{B}$ is the orthogonal basis of eigenvectors of A given in Theorem 11.1.22, then $U = Q_{\mathcal{E}}^{\mathcal{B}}(\mathrm{Id}_n)$, where $Q_{\mathcal{E}}^{\mathcal{B}}(\mathrm{Id}_n)$ is the matrix of change of basis from $\mathcal{B}$ to $\mathcal{E}$.

11.1.5 Definite Positive Matrices

Definition 11.1.24

A **quadratic form** on $\mathbb{R}^n$ (resp. $\mathbb{C}^n$) is a real-valued function Q of the form $Q(\mathbf{x}) = \mathbf{x}^* A \mathbf{x}$ for $\mathbf{x}$ in $\mathbb{R}^n$ (resp. $\mathbb{C}^n$), where A is a $n \times n$ symmetric (resp. hermitian) matrix.

Definition 11.1.25

1. A quadratic form $Q(\mathbf{x})$ is **positive definite** if $Q(\mathbf{x}) \geq 0$ for all $\mathbf{x}$.
2. A quadratic form $Q(\mathbf{x})$ is **strictly positive definite** if $Q(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$.
3. A quadratic form $Q(\mathbf{x})$ is **indefinite** if there exists $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $Q(\mathbf{x}_1) < 0 < Q(\mathbf{x}_2)$.
4. A $n \times n$ matrix A is **positive definite** (resp. **strictly positive definite**) if $Q(\mathbf{x}) = \mathbf{x}^* A\mathbf{x}$ is positive definite (resp. strictly positive definite).

Theorem 11.1.26

Let A be a $n \times n$ symmetric matrix. A is positive definite if and only if all the eigenvalues of A are greater than 0.

Proof.
i) Suppose that A is positive definite. Let λ be an eigenvalue of A and $\mathbf{v}$ be an eigenvector associated to λ. We have

$$0 < \mathbf{v}^\top A\mathbf{v} = \mathbf{v}^\top (\lambda \mathbf{v}) = \lambda \mathbf{v}^\top \mathbf{v} = \lambda \|\mathbf{v}\|^2 .$$

Thus $\lambda > 0$.

ii) Suppose that all eigenvalues of A are positive. Let $\{\mathbf{v}_i\}_{i=1}^n$ be an orthogonal basis of eigenvectors of A given by Theorem 11.1.22. Let λ_j be the eigenvalue associated to the eigenvector $\mathbf{v}_j$ for $j = 1, 2, \ldots, n$. Given $\mathbf{x} \in \mathbb{R}^n$, we may write $\mathbf{x} = \sum_{j=1}^n \beta_j \mathbf{v}_j$ for some unique

$\beta_j \in \mathbb{R}$. Hence,

$$\begin{aligned}\mathbf{x}^\top A\mathbf{x} &= \mathbf{x}^\top \left(\sum_{j=1}^n \beta_j A\mathbf{v}_j\right) = \mathbf{x}^\top \left(\sum_{j=1}^n \lambda_j\beta_j\mathbf{v}_j\right) = \sum_{i=1}^n \left(\sum_{j=1}^n \lambda_j\beta_j\beta_i\mathbf{v}_i^\top\mathbf{v}_j\right)\\ &= \sum_{i=1}^n \left(\sum_{j=1}^n \lambda_j\beta_j\beta_i\delta_{i,j}\right) = \sum_{j=1}^n \lambda_j\beta_j^2 > 0\end{aligned}$$

if $\mathbf{x} \neq \mathbf{0}$. ∎

Theorem 11.1.27

Let A be a $n \times n$ symmetric (or hermitian) matrix.

1. A is positive definite if and only if $\det(A_k) > 0$ for all **principal submatrices**

$$A_k = \begin{pmatrix} a_{1,1} & \ldots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \ldots & a_{k,k}\end{pmatrix}$$

   of A.

2. A is positive definite if and only if all the pivots used in the reduction process of A to a row-echelon form, without interchanging rows, are positive.

11.1.6 Gerschgorin's Theorem

Theorem 11.1.28 (**Gerschgorin's Circles**)

Let A be an $n \times n$ matrix and λ be an eigenvalue of A. Then there exists an index i with $1 \leq i \leq n$, such that

$$|\lambda - a_{i,i}| \leq \sum_{\substack{j=1\\ j\neq i}}^n |a_{i,j}| \ .$$

More precisely, each component (i.e. a connected set which is not properly contained in a larger connected set) of the union $\bigcup_{i=1}^n U_i$ of the **Gerschgorin's circles**

$$U_i = \left\{\lambda : |\lambda - a_{i,i}| \leq \sum_{\substack{j=1\\ j\neq i}}^n |a_{i,j}|\right\}$$

contains exactly as many eigenvalues of A (counted with algebraic multiplicity) as circles U_i forming the component.

Proof.

i) Suppose that λ is an eigenvalue of A and that $\mathbf{v}$ is an eigenvector associated to λ. Let k be an integer such that $|v_k| = \|\mathbf{v}\|_\infty = \max_{1\leq j\leq n} |v_j|$. Note that $v_k \neq 0$ because $\mathbf{v} \neq \mathbf{0}$. From $A\mathbf{v} = \lambda\mathbf{v}$, we get

$$\sum_{j=1}^{n} a_{k,j} v_j = \lambda v_k \ .$$

Thus

$$|(a_{k.k} - \lambda) v_k| = \left| - \sum_{\substack{j=1 \\ j\neq k}} a_{k,j} v_j \right| \ .$$

After dividing both sides of this equality by $|v_k|$, we get

$$|a_{k.k} - \lambda| \leq \sum_{\substack{j=1 \\ j\neq k}} |a_{k,j}| \frac{|v_j|}{|v_k|} \leq \sum_{\substack{j=1 \\ j\neq k}} |a_{k,j}| \ .$$

This implies that $\lambda \in U_k$.

ii) To prove the second statement of the theorem, suppose that U is a component of the form

$$U = \bigcup_{i\in I} U_i \ ,$$

where I is a subset of $\{1, 2, \ldots, n\}$. Let $A(t) = tA + (1-t)D$. where D is the diagonal matrix defined by

$$D = \begin{pmatrix} a_{1,1} & 0 & 0 & \ldots & 0 \\ 0 & a_{2,2} & 0 & \ldots & 0 \\ 0 & 0 & a_{3,3} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & a_{n,n} \end{pmatrix} ,$$

Let $R(t) = \bigcup_{i\in I} R_i(t)$ with

$$R_i(t) = \left\{ z : |z - a_{i,i}| \leq t \sum_{\substack{j=1 \\ j\neq i}} |a_{i,j}| \right\} \ .$$

At $t = 0$, there are obviously $|I|$ eigenvalues of $A(0) = D$ in $R(0) = \{a_{j,j} : j \in I\}$ (counted with algebraic multiplicity); these eigenvalues are $a_{j,j}$ for $j \in I$.

The eigenvalues of $A(t)$ are in $\bigcup_{i=1}^{n} R_i(t)$ because of (i). Moreover, $R(t)$ is a closed set such that $R(t) \cap R_i(t) = \emptyset$ for all $i \notin I$ and all $0 \leq t \leq 1$ because $R_i(t) \subset R_i(1) = U_i$ for all $0 \leq t \leq 1$ and all i, and $U \cap U_i = \emptyset$ for all $i \notin I$ (Figure 11.2). Since the eigenvalues of $A(t)$ are continuous functions of t, because the roots of the characteristic polynomial $p_t(\lambda) = \det(A(t) - \lambda \, \mathrm{Id})$ are continuous functions of its coefficients which are continuous functions of t, the number of eigenvalues of $A(t)$ in $R(t)$ (counted with algebraic multiplicity) is constant. No eigenvalue of $A(t)$ can jump from $R(t)$ to one of the $R_i(t)$ with $i \notin I$ by continuity.

Therefore, $R(1) = U$ contains $|I|$ eigenvalues (counted with algebraic multiplicity) as $R(0)$. ∎

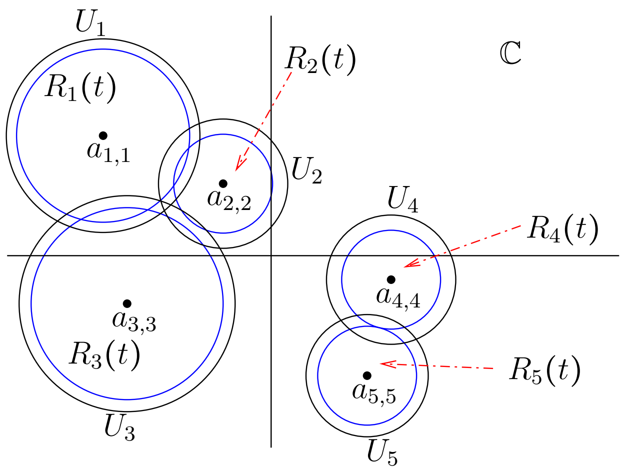


Figure 11.2: Example of the Gerschgorin's circles in the case of a 5×5 matrix A. $U = U_1 \cup U_2 \cup U_3$ is a component containing three eigenvalues (counted with multiplicity) of A.

11.2 Power Method

The first method that we present can be used to approximate the largest eigenvalue in absolute value of an $n \times n$ matrix A.

Suppose that the $n \times n$ matrix A has m distinct eigenvalues such that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_m| \ . \tag{11.2.1}$$

We assume that there is a basis of eigenvectors of A for $\mathbb{R}^n$. So, every vector in $\mathbf{x} \in \mathbb{R}^n$ can be expressed uniquely as a sum $\mathbf{x} = \sum_{j=1}^{m} \mathbf{v}_j$, where $\mathbf{v}_j$ is an eigenvector associated to λ_j or the null vector.

Given $\mathbf{y} \neq \mathbf{0}$, we may write $\mathbf{y}$ as $\mathbf{y} = \sum_{j=1}^{m} \mathbf{v}_j$ where $\mathbf{v}_j$ is an eigenvector associated to λ_j for each j. It is easy to prove by induction that

$$A^j \mathbf{y} = \sum_{i=1}^{m} \lambda_i^j \mathbf{v}_i = \lambda_i^j \left(\mathbf{v}_1 + \sum_{i=2}^{m} \left(\frac{\lambda_i}{\lambda_1} \right)^j \mathbf{v}_i \right)$$

for $j \geq 0$. If
$$\psi_j = \sum_{i=2}^{m} \left(\frac{\lambda_i}{\lambda_1}\right)^j \mathbf{v}_i \ ,$$
we have that $\psi_j \to 0$ as $j \to 0$ because of (11.2.1), and so $\lambda_1^{-j} A^j \mathbf{y} = \mathbf{v}_1 + \psi_j \to \mathbf{v}_1$ as $j \to \infty$.

Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a linear functional such that $\phi(\mathbf{v}_1) \neq 0$. We have that
$$\mu_j = \frac{\phi(A^j\mathbf{y})}{\phi(A^{j-1}\mathbf{y})} = \lambda_1 \frac{\phi\left(\mathbf{v}_1 + \psi_j\right)}{\phi\left(\mathbf{v}_1 + \psi_{j-1}\right)} \to \lambda_1 \frac{\phi\left(\mathbf{v}_1\right)}{\phi\left(\mathbf{v}_1\right)} = \lambda_1 \quad \text{as} \quad j \to \infty \ .$$
There are infinitely many possible choices for the linear functional ϕ. A linear functional that is often used is defined by $\phi(\mathbf{x}) = x_k$, the k^{th} component of the vector $\mathbf{x}$, for k constant.

Generally, the sequence $\{\mu_j\}_{j=1}^{\infty}$ converge linearly to λ_1. We have
$$\begin{aligned}
\left|\frac{\mu_{j+1} - \lambda_1}{\mu_j - \lambda_1}\right| &= \left|\left(\lambda_1 \frac{\phi\left(\mathbf{v}_1 + \psi_{j+1}\right)}{\phi\left(\mathbf{v}_1 + \psi_j\right)} - \lambda_1\right)\left(\lambda_1 \frac{\phi\left(\mathbf{v}_1 + \psi_j\right)}{\phi\left(\mathbf{v}_1 + \psi_{j-1}\right)} - \lambda_1\right)^{-1}\right| \\
&= \left|\left(\frac{\phi\left(\mathbf{v}_1 + \psi_{j+1}\right) - \phi\left(\mathbf{v}_1 + \psi_j\right)}{\phi\left(\mathbf{v}_1 + \psi_j\right) - \phi\left(\mathbf{v}_1 + \psi_{j-1}\right)}\right)\left(\frac{\phi\left(\mathbf{v}_1 + \psi_{j-1}\right)}{\phi\left(\mathbf{v}_1 + \psi_j\right)}\right)\right| \\
&= \left|\left(\frac{\phi\left(\psi_{j+1}\right) - \phi\left(\psi_j\right)}{\phi\left(\psi_j\right) - \phi\left(\psi_{j-1}\right)}\right)\left(\frac{\phi\left(\mathbf{v}_1\right) + \phi\left(\psi_{j-1}\right)}{\phi\left(\mathbf{v}_1\right) + \phi\left(\psi_j\right)}\right)\right| \ ,
\end{aligned}$$
where the linearity of ϕ has been used to get the last equality. If we assume that $|\lambda_3| < |\lambda_2|$, then
$$\begin{aligned}
\frac{\phi\left(\psi_{j+1}\right) - \phi\left(\psi_j\right)}{\phi\left(\psi_j\right) - \phi\left(\psi_{j-1}\right)} &= \frac{\displaystyle\sum_{i=2}^{m}\left(\frac{\lambda_i}{\lambda_1}\right)^{j+1}\phi(\mathbf{v}_i) - \sum_{i=2}^{m}\left(\frac{\lambda_i}{\lambda_1}\right)^{j}\phi(\mathbf{v}_i)}{\displaystyle\sum_{i=2}^{m}\left(\frac{\lambda_i}{\lambda_1}\right)^{j}\phi(\mathbf{v}_i) - \sum_{i=2}^{m}\left(\frac{\lambda_i}{\lambda_1}\right)^{j-1}\phi(\mathbf{v}_i)} \\
&= \left(\frac{\lambda_2}{\lambda_1}\right)\frac{\displaystyle\left(\frac{\lambda_2}{\lambda_1} - 1\right)\phi(\mathbf{v}_2) + \sum_{i=3}^{m}\left(\frac{\lambda_i}{\lambda_1} - 1\right)\left(\frac{\lambda_i}{\lambda_2}\right)^{j}\phi(\mathbf{v}_i)}{\displaystyle\left(\frac{\lambda_2}{\lambda_1} - 1\right)\phi(\mathbf{v}_2) + \sum_{i=3}^{m}\left(\frac{\lambda_i}{\lambda_1} - 1\right)\left(\frac{\lambda_i}{\lambda_2}\right)^{j-1}\phi(\mathbf{v}_i)} \to \left(\frac{\lambda_2}{\lambda_1}\right) \neq 0 \quad \text{as} \quad j \to \infty
\end{aligned}$$
because $|\lambda_i|/|\lambda_2| < 1$ for $3 \leq i \leq n$. Moreover
$$\frac{\phi\left(\mathbf{v}_1\right) + \phi\left(\psi_{j-1}\right)}{\phi\left(\mathbf{v}_1\right) + \phi\left(\psi_j\right)} \to \frac{\phi\left(\mathbf{v}_1\right)}{\phi\left(\mathbf{v}_1\right)} = 1 \quad \text{as} \quad j \to \infty \ .$$
Thus
$$\lim_{j\to\infty}\left|\frac{\mu_{j+1} - \lambda_1}{\mu_j - \lambda_1}\right| = \left|\frac{\lambda_2}{\lambda_1}\right| \neq 0$$
if $|\lambda_3| < |\lambda_2|$. The method may converge less than linearly if $|\lambda_3| = |\lambda_2|$. Even when the convergence is linear, it could still be very slow if $|\lambda_1| \approx |\lambda_2|$. There is also the danger

of divisions by very small numbers if $A^j\mathbf{y}$ approaches the origin when $j \to \infty$ [1]. So, the power method is not that powerful. It will need to be improved. One may use Aitken's Δ^2 procedure to accelerate the convergence toward the eigenvalue λ_1 but even that is not a huge improvement.

If λ_1 is real and positive, the sequence $\{\mathbf{w}\}_{j=0}^\infty$ defined by $\mathbf{w}_j = \dfrac{1}{\|A^j\mathbf{y}\|} A^j\mathbf{y}$ converges to $\dfrac{1}{\|\mathbf{v}_1\|}\mathbf{v}_1$, an eigenvector of norm one associated to the eigenvalue λ_1, because

$$\mathbf{w}_j = \frac{1}{\|A^j\mathbf{y}\|} A^j\mathbf{y} = \left\| \sum_{i=1}^m \lambda_i^j \mathbf{v}_i \right\|^{-1} \left(\sum_{i=1}^m \lambda_i^j \mathbf{v}_i \right) = \frac{1}{\|\mathbf{v}_1 + \psi_j\|} (\mathbf{v}_1 + \psi_j) \to \frac{1}{\|\mathbf{v}_1\|}\mathbf{v}_1 \quad \text{as} \quad j \to \infty \ .$$

In general, the vector $\mathbf{w}_j$ is getting "more parallel" to the direction of the eigenvector $\mathbf{v}_1$ as $j \to \infty$.

11.3 Rayleigh Quotient for Symmetric Matrices

We consider a $n \times n$ symmetric matrix A. As for the iterative power method of the previous section, we assume that A has m distinct eigenvalues which are real according to Theorem 11.1.22.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_m| \ .$$

Moreover, there exists an orthonormal basis of eigenvectors of A. As we mentioned in the previous section in such case, every vector in $\mathbf{x} \in \mathbb{R}^n$ can be expressed uniquely as a sum $\mathbf{x} = \sum_{j=1}^m \mathbf{v}_j$, where $\mathbf{v}_j$ is an eigenvector associated to λ_j or the null vector.

Definition 11.3.1

The **Rayleigh Quotient** of the symmetric matrix A is the function

$$\rho_A(\mathbf{x}) = \frac{\langle \mathbf{x}, A\mathbf{x}\rangle}{\langle \mathbf{x}, \mathbf{x}\rangle}$$

for $\mathbf{x} \neq \mathbf{0}$.

Given $\mathbf{y} \neq \mathbf{0}$, we write $\mathbf{y} = \sum_{j=1}^m \mathbf{v}_j$, where $\mathbf{v}_j$ is an eigenvector associated to λ_j. The sequence $\{\mu_j\}_{j=1}^\infty$ defined by $\mu_j = \rho_A(A^j\mathbf{y})$ converges to λ_1. To prove it, let

$$\psi_k = \sum_{i=2}^m \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i^\top \mathbf{v}_i \ .$$

[1] To avoid divisions by very small numbers, one may generate the $\mathbf{y}_j$ as it follows: $\mathbf{y}_0 = \|\mathbf{y}\|^{-1}\mathbf{y}$ and $\mathbf{y}_j = \|A\mathbf{y}_{j-1}\|^{-1}A\mathbf{y}_{j-1}$ for $j > 0$. Then $\mu_j = \dfrac{\phi(\mathbf{y}_j)}{\phi(\mathbf{y}_{j-1})}$ for $j > 0$.

We then have

$$\mu_j = \frac{(A^j\mathbf{y})^\top A(A^j\mathbf{y})}{(A^j\mathbf{y})^\top(A^j\mathbf{y})} = \frac{\mathbf{y}^\top A^{2j+1}\mathbf{y}}{\mathbf{y}^\top A^{2j}\mathbf{y}} = \frac{\sum_{i=1}^m \lambda_i^{2j+1}\mathbf{v}_i^\top\mathbf{v}_i}{\sum_{i=1}^m \lambda_i^{2j}\mathbf{v}_i^\top\mathbf{v}_i} = \lambda_1\left(\frac{\mathbf{v}_1^\top\mathbf{v}_1 + \psi_{2j+1}}{\mathbf{v}_1^\top\mathbf{v}_1 + \psi_{2j}}\right) \to \lambda_1 \quad \text{as} \quad j \to \infty$$

because $|\lambda_i/\lambda_1| < 1$ for all $i > 1$.

The sequence $\{\mu_j\}_{j=1}^\infty$ converges much faster to the eigenvalue λ_1 than the simple power method of the previous section.

As for the power method of the previous section, if $\lambda_1 > 0$, the sequence $\{\mathbf{w}\}_{j=0}^\infty$ defined by $\mathbf{w}_j = \dfrac{1}{\|A^j\mathbf{y}\|}A^j\mathbf{y}$ converges to $\dfrac{1}{\|\mathbf{v}_1\|}\,\mathbf{v}_1$, an eigenvector of norm one associated to the eigenvalue λ_1. If $\lambda_1 < 0$, the sequence $\{\mathbf{w}\}_{j=0}^\infty$ defined by $\mathbf{w}_j = \dfrac{1}{\|A^{2j}\mathbf{y}\|}A^{2j}\mathbf{y}$ converges to $\dfrac{1}{\|\mathbf{v}_1\|}\,\mathbf{v}_1$.

11.4 Inverse Power Method

Until now, we have presented methods to approximate the largest eigenvalue in absolute value of a $n \times n$ matrix A. How can we find the other eigenvalues? We present in this section one possible method to answer this question. Suppose that $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of A counted with their algebraic multiplicity.

Choose $q \neq \lambda_i$ for all i. The matrix $A - q\,\mathrm{Id}$ is invertible and the eignevalues of $(A - q\,\mathrm{Id})^{-1}$ are of the form $1/(\lambda_i + q)$, where λ_i is an eigenvalue of A. In fact, $\mathbf{v}_i$ is an eigenvector of A associated to the eigenvalue λ_i if and only if

$$(\lambda_i - q)\mathbf{v}_i = (A - q\,\mathrm{Id})\,\mathbf{v}_i \ .$$

This is if and only if

$$(A - q\,\mathrm{Id})^{-1}\,\mathbf{v}_i = \frac{1}{\lambda_i - q}\mathbf{v}_i \ .$$

This last equation says that $\mathbf{v}_i$ is an eigenvector of $A - q\,\mathrm{Id}$ associated to the eigenvalue $1/(\lambda_i - q)$.

Suppose that k is an index such that $1/|\lambda_k - q| > 1/|\lambda_i - q|$ for $i \neq k$. We may then use the iterative power method with $(A - q\,\mathrm{Id})^{-1}$ instead of A, to approximate the eigenvalue $1/(\lambda_k - q)$ and an eigenvector associated to this eigenvalue. This gives us an approximation of the eigenvalue λ_k of A.

If A is symmetric, then $(A - q\,\mathrm{Id})^{-1}$ is also a symmetric matrix. Hence, we may use the Rayleigh quotient to approximate the eigenvalue $1/(\lambda_k - q)$ of $(A - q\,\mathrm{Id})^{-1}$.

11.5 Householder's Matrices and Hessemberg Forms

The **(principal) subdiagonal** of an $n \times n$ matrix B is the set formed by the components $b_{i+1,i}$ for $i = 1, 2, \ldots n-1$. Given an $n \times n$ matrix A, the goal of this section is to find a matrix B conjugate to A such that the elements below the principal subdiagonal are zero (i.e. $b_{i,j} = 0$ for $i > j+1$). If A is symmetric, then B is also symmetric. Thus B satisfies $b_{i,j} = 0$ for $|i-j| \geq 2$. Such matrices are called **tridiagonal matrices**.

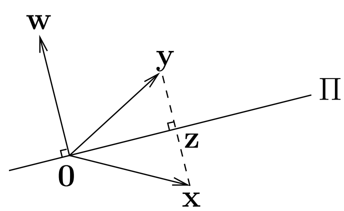
In the next section, we will present a method to find eignevalues of symmetric tridiagonal matrices like B above. But first, we have to review some concepts in linear algebra.

Definition 11.5.1

Let $\mathbf{w} \in \mathbb{R}^n$ be a non-null vector. the $n \times n$ **Householder matrix** $H_{\mathbf{w}}$ is defined by

$$H_{\mathbf{w}} = \mathrm{Id}_n - \left(\frac{2}{\mathbf{w}^\top \mathbf{w}}\right) \mathbf{w}\mathbf{w}^\top \ .$$

We present a geometric interpretation of the Householder matrix. Let Π be the $n-1$ dimensional subspace of $\mathbb{R}^n$ defined by $\Pi = \{\mathbf{v} \in \mathbb{R}^n : \mathbf{v} \perp \mathbf{w}\}$ and $L : \mathbb{R}^n \to \mathbb{R}^n$ be the reflection through Π.



L is a linear mapping. If $\mathbf{y} = L(\mathbf{x})$, we have that $\mathbf{y} = \mathbf{x} + \alpha\mathbf{w}$ for some $\alpha \in \mathbb{R}$. Thus $\mathbf{z} = \mathbf{x} + \dfrac{\alpha}{2}\mathbf{w} \in \Pi$ and $\mathbf{z} \perp \mathbf{w}$.

From

$$0 = \mathbf{w}^\top \mathbf{z} = \mathbf{w}^\top \left(\mathbf{x} + \frac{\alpha}{2}\mathbf{w}\right) = \mathbf{w}^\top \mathbf{x} + \frac{\alpha}{2}\mathbf{w}^\top \mathbf{w} \ ,$$

we get

$$\alpha = -\frac{2\,\mathbf{w}^\top \mathbf{x}}{\mathbf{w}^\top \mathbf{w}} \ .$$

Hence,

$$L(\mathbf{x}) = \mathbf{x} + \alpha\mathbf{w} = \mathbf{x} - \left(\frac{2\,\mathbf{w}^\top \mathbf{x}}{\mathbf{w}^\top \mathbf{w}}\right)\mathbf{w} = \mathbf{x} - \left(\frac{2}{\mathbf{w}^\top \mathbf{w}}\right)\mathbf{w}\mathbf{w}^\top \mathbf{x} = H_{\mathbf{w}}(\mathbf{x}) \ .$$

Thus $H_{\mathbf{w}}$ is the reflection through the subspace orthogonal to $\mathbf{w}$.

Theorem 11.5.2

Let $H_{\mathbf{w}}$ be an $n \times n$ Householder matrix. Then

1. $H_{\mathbf{w}}$ is symmetric and orthogonal.

2. $H_{\mathbf{w}}(\mathbf{x})$ is the reflection of the vector $\mathbf{x}$ through the subspace orthogonal to $\mathbf{w}$.
3. $\det(H_{\mathbf{w}}) = -1$.
4. For any $\mathbf{x}$ and $\mathbf{y}$ with $\mathbf{x} \neq \mathbf{y}$, there exists $\mathbf{w} \in \mathbb{R}^n$ such that $H_{\mathbf{w}}(\mathbf{x})$ is a scalar multiple of $\mathbf{y}$. In fact,

   (i) if $\mathbf{x} \neq \lambda\mathbf{y}$ with $\lambda > 0$, then we can take $\mathbf{w} = \mathbf{x} - \dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\mathbf{y}$. We get $H_{\mathbf{w}}(\mathbf{x}) = \dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\mathbf{y}$.

   (ii) if $\mathbf{x} \neq \lambda\mathbf{y}$ with $\lambda < 0$, then we can take $\mathbf{w} = \mathbf{x} + \dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\mathbf{y}$. We get $H_{\mathbf{w}}(\mathbf{x}) = -\dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\mathbf{y}$.

Proof.
1) We have

$$\begin{aligned} H_{\mathbf{w}}^{\top} &= \left(\mathrm{Id}_n - \left(\frac{2}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top}\right)^{\top} = \mathrm{Id}_n^{\top} - \left(\frac{2}{\mathbf{w}^{\top}\mathbf{w}}\right)(\mathbf{w}\mathbf{w}^{\top})^{\top} \\ &= \mathrm{Id}_n - \left(\frac{2}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top} = H_{\mathbf{w}} \ . \end{aligned}$$

Thus, $H_{\mathbf{w}}$ is symmetric. Moreover,

$$\begin{aligned} H_{\mathbf{w}}^2 &= \left(\mathrm{Id}_n - \left(\frac{2}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top}\right)\left(\mathrm{Id}_n - \left(\frac{2}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top}\right) \\ &= \mathrm{Id}_n - \left(\frac{4}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top} + \left(\frac{4}{(\mathbf{w}^{\top}\mathbf{w})^2}\right)\mathbf{w}\mathbf{w}^{\top}\mathbf{w}\mathbf{w}^{\top} \\ &= \mathrm{Id}_n - \left(\frac{4}{\mathbf{w}^{\top}\mathbf{w}}\right)\mathbf{w}\mathbf{w}^{\top} + \left(\frac{4}{(\mathbf{w}^{\top}\mathbf{w})^2}\right)\mathbf{w}\left(\mathbf{w}^{\top}\mathbf{w}\right)\mathbf{w}^{\top} = \mathrm{Id}_n \ . \end{aligned}$$

Thus $H_{\mathbf{w}}^{-1} = H_{\mathbf{w}} = H_{\mathbf{w}}^{\top}$ implies that $H_{\mathbf{w}}$ is orthogonal.

2) This has been proved before the statement of the theorem.

3) Let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-1}\}$ be an orthogonal basis of $\Pi = \{\mathbf{v} \in \mathbb{R}^n : \mathbf{v} \perp \mathbf{w}\}$. Then $\{\mathbf{w}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-1}\}$ is an orthogonal basis of $\mathbb{R}^n$. Since $H_{\mathbf{w}}(\mathbf{w}) = -\mathbf{w}$ and $H_{\mathbf{w}}(\mathbf{v}_i) = \mathbf{v}_i$ for all i, we have that -1 is an eigenvalue of algebraic and geometric multiplicity one while 1 is an eigenvalue of algebraic and geometric multiplicity $n-1$. Hence, since $\det(H_{\mathbf{w}})$ is equal to the product of the eigenvalues, we have that $\det(H_{\mathbf{w}}) = -1$.

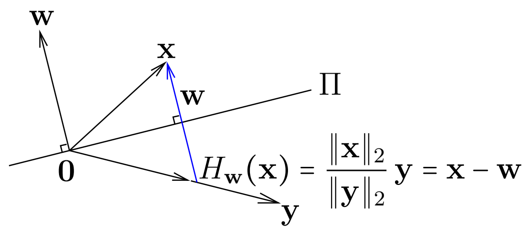
Another way to show that $\det(H_{\mathbf{w}}) = -1$ is to consider the $n \times n$ matrix $A = \begin{pmatrix} \mathbf{w} & \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_{n-1} \end{pmatrix}$. We have that $H_{\mathbf{w}}A = \begin{pmatrix} -\mathbf{w} & \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_{n-1} \end{pmatrix}$. Since $H_{\mathbf{w}}A$ is obtained from A by multiplying the first column of A by -1, we have that

$$\det(H_{\mathbf{w}})\det(A) = \det(H_{\mathbf{w}}A) = -\det(A)$$

Since $\det(A) \neq 0$, we get $\det(H_{\mathbf{w}}) = -1$.

4) For (i). it is enough to prove that $\dfrac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}} = 1$ for $\mathbf{w} = \mathbf{x} - \dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}$ because this will implies that

$$H_{\mathbf{w}}(\mathbf{x}) = \mathbf{x} - \left(\frac{2}{\mathbf{w}^\top\mathbf{w}}\right)\mathbf{w}\mathbf{w}^\top\mathbf{x} = \mathbf{x} - \left(\frac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}}\right)\mathbf{w} = \mathbf{x} - \mathbf{w} = \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\ .$$



Since $\mathbf{x} \neq \lambda\mathbf{y}$ with $\lambda > 0$, we have that $\mathbf{w} \neq \mathbf{0}$. Hence,

$$\mathbf{w}^\top\mathbf{x} = \left(\mathbf{x} - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\right)^\top \mathbf{x} = \mathbf{x}^\top\mathbf{x} - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x} = \|\mathbf{x}\|_2^2 - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x}$$

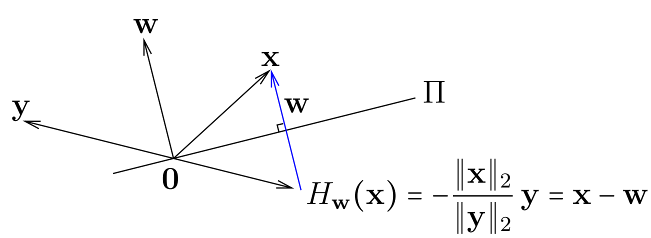
and

$$\begin{aligned}\mathbf{w}^\top\mathbf{w} &= \left(\mathbf{x}^\top - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\right)\left(\mathbf{x} - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\right) = \mathbf{x}^\top\mathbf{x} - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\underbrace{(\mathbf{y}^\top\mathbf{x} + \mathbf{x}^\top\mathbf{y})}_{=2\,\mathbf{y}^\top\mathbf{x}} + \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{y}\|_2^2}\,\mathbf{y}^\top\mathbf{y} \\ &= 2\left(\|\mathbf{x}\|_2^2 - \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x}\right)\ .\end{aligned}$$

Thus $\dfrac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}} = 1$.

For (ii). it is also enough to prove that $\dfrac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}} = 1$ for $\mathbf{w} = \mathbf{x} + \dfrac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}$ because this will implies that

$$H_{\mathbf{w}}(\mathbf{x}) = \mathbf{x} - \left(\frac{2}{\mathbf{w}^\top\mathbf{w}}\right)\mathbf{w}\mathbf{w}^\top\mathbf{x} = \mathbf{x} - \left(\frac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}}\right)\mathbf{w} = \mathbf{x} - \mathbf{w} = -\frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\ .$$



Since $\mathbf{x} \neq \lambda\mathbf{y}$ with $\lambda < 0$, we have that $\mathbf{w} \neq \mathbf{0}$. Hence,

$$\mathbf{w}^\top\mathbf{x} = \left(\mathbf{x} + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\right)^\top \mathbf{x} = \mathbf{x}^\top\mathbf{x} + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x} = \|\mathbf{x}\|_2^2 + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x}$$

and

$$\mathbf{w}^\top\mathbf{w} = \left(\mathbf{x}^\top + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\right)\left(\mathbf{x} + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}\right) = \mathbf{x}^\top\mathbf{x} + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\underbrace{\left(\mathbf{y}^\top\mathbf{x} + \mathbf{x}^\top\mathbf{y}\right)}_{=2\,\mathbf{y}^\top\mathbf{x}} + \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{y}\|_2^2}\,\mathbf{y}^\top\mathbf{y}$$

$$= 2\left(\|\mathbf{x}\|_2^2 + \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}\,\mathbf{y}^\top\mathbf{x}\right) .$$

Thus $\dfrac{2\,\mathbf{w}^\top\mathbf{x}}{\mathbf{w}^\top\mathbf{w}} = 1$. ∎

Algorithm 11.5.3 (QR Decomposition with Householder Matrices)

Let A be a $n\times m$ matrix with entries in $\mathbb{R}$.

1. Use item 4 of Theorem 11.5.2 to find $\mathbf{w} \in \mathbb{R}^n$ such that $H_{\mathbf{w}}$ maps the first column of A to a non-negative multiple of $\mathbf{e}_1$ in $\mathbb{R}^n$. If the first column of A is already a multiple of $\mathbf{e}_1 \in \mathbb{R}^n$, take $\mathbf{w} = \mathbf{0}$.

2. Let $Q_1 = H_{\mathbf{w}}$ (when the first column of A is already a multiple of $\mathbf{e}_1 \in \mathbb{R}^n$, then $Q_1 = \mathrm{Id}_n$) and let $A_1 = Q_1 A$. The matrix Q_1 is an orthogonal matrix. A_1 is of the form
$$A_1 = \begin{pmatrix} R_1 & B_1 \\ 0 & C_1 \end{pmatrix} ,$$
where $R_1 \in \mathbb{R}$.

3. Suppose that A_i is of the form
$$A_i = \begin{pmatrix} R_i & B_i \\ 0 & C_i \end{pmatrix} ,$$
where R_i is an $i\times i$ upper-triangular matrix. Use item 4 of Theorem 11.5.2 to find $\mathbf{w} \in \mathbb{R}^{n-i}$ such that $H_{\mathbf{w}}$ maps the first column of C_i to a non-negative multiple of $\mathbf{e}_1$ in $\mathbb{R}^{n-i}$. If the first column of C_i is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-i}$, take $\mathbf{w} = \mathbf{0}$.

4. Let
$$Q_{i+1} = \begin{pmatrix} \mathrm{Id}_i & 0 \\ 0 & H_{\mathbf{w}} \end{pmatrix}$$
(when the first column of C_i is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-i}$, then $Q_{i+1} = \mathrm{Id}_n$) and let $A_{i+1} = Q_{i+1}A_i$. The matrix Q_{i+1} is an orthogonal matrix. A_{i+1} is of the form
$$A_{i+1} = \begin{pmatrix} R_{i+1} & B_{i+1} \\ 0 & C_{i+1} \end{pmatrix} ,$$
where R_i is an $(i+1)\times(i+1)$ upper-triangular matrix.

5. Repeat (3) and (4) with i replace by $i+1$ until $i = n-1$.

Then $Q_nQ_{n-1}\cdots Q_1A = R$ is an upper-triangular matrix with non-negative entries on the main diagonal. If $Q = Q_1Q_2\cdots Q_n$, then Q is an orthogonal matrix such that $A = QR$.

The QR decomposition with Householder matrices gives a method to solve linear systems of equations of the form $A\mathbf{x} = \mathbf{b}$, where A is an $n \times m$ matrix and $\mathbf{b} \in \mathbb{R}^n$. Suppose that $A = QR$ is the QR decomposition of A. Since $Q^{-1} = Q^\top$, $\mathbf{x}$ is the solution of $A\mathbf{x} = \mathbf{b}$ if and only if $\mathbf{x}$ is the solution of $R\mathbf{x} = Q^\top\mathbf{b}$. The solution $\mathbf{x}$ of $R\mathbf{x} = Q^\top\mathbf{b}$ is found using backward substitution.

Definition 11.5.4

We say that an $n \times n$ matrix M is in **Hessemberg form** if $m_{i,j} = 0$ for $j + 1 < i \leq n$ and $1 \leq j \leq n - 2$.

Algorithm 11.5.5 (Hessemberg form)

Let A be a $n \times n$ -matrix with entries in $\mathbb{R}$.

1. Suppose that
$$A = \begin{pmatrix} T & \mathbf{r}^\top \\ \mathbf{s} & C \end{pmatrix},$$
where $T \in \mathbb{R}$. Use item 4 of Theorem 11.5.2 to find $\mathbf{w}_1 \in \mathbb{R}^{n-1}$ such that $H_{\mathbf{w}_1}$ maps $\mathbf{s}$ to a multiple of $\mathbf{e}_1$ in $\mathbb{R}^{n-1}$. If $\mathbf{s}$ is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-1}$, take $\mathbf{w}_1 = \mathbf{0}$.

2. Let
$$G_1 = \begin{pmatrix} 1 & 0 \\ 0 & H_{\mathbf{w}_1} \end{pmatrix}$$
(when $\mathbf{s}$ is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-1}$, then $G_1 = \mathrm{Id}_n$) and let $A_1 = G_1 A G_1$. The matrix G_1 is an orthogonal matrix. A_1 is of the form
$$A_1 = \begin{pmatrix} T_1 & B_1 \\ 0 & C_1 \end{pmatrix},$$
where T_1 is an 2×1 matrix.

3. Suppose that A_i is of the form
$$A_i = \begin{pmatrix} T_i & B_i \\ 0 & C_i \end{pmatrix},$$
where $M = T_i$ is an $(i+1) \times i$ matrix satisfying $M_{j,k} = 0$ for $j > k + 1$. Use item 4 of Theorem 11.5.2 to find $\mathbf{w}_{i+1} \in \mathbb{R}^{n-i-1}$ such that $H_{\mathbf{w}_{i+1}}$ maps the first column of C_i to a multiple of $\mathbf{e}_1$ in $\mathbb{R}^{n-i-1}$. If the first column of C_i is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-i-1}$, take $\mathbf{w}_{i+1} = \mathbf{0}$.

4. Let
$$G_{i+1} = \begin{pmatrix} \mathrm{Id}_{i+1} & 0 \\ 0 & H_{\mathbf{w}_{i+1}} \end{pmatrix}$$

(when the first column of C_i is already a non-negative multiple of $\mathbf{e}_1 \in \mathbb{R}^{n-i-1}$, then $G_{i+1} = \mathrm{Id}_n$) and let $A_{i+1} = G_{i+1}A_iG_{i+1}$. The matrix G_{i+1} is an orthogonal matrix. A_{i+1} is of the form

$$A_{i+1} = \begin{pmatrix} T_{i+1} & B_{i+1} \\ 0 & C_{i+1} \end{pmatrix} ,$$

where $M = T_{i+1}$ is an $(i+2)\times(i+1)$ matrix satisfying $M_{j,k} = 0$ for $j > k+1$.

5. Repeat (3) and (4) with i replace by $i+1$ until $i = n-3$.

Then $T = G_{n-2}G_{n-3}\ldots G_1AG_1G_2\cdots G_{n-2}$ is a matrix in the Hessemberg form. If $G = G_1G_2\cdots G_{n-2}$, then the matrix G is an orthogonal matrix such that $A = G^{\top}TG$.

Our goal is now to implement efficiently the previous theorem to get, for any given matrix A, an Hessemberg form conjugate to A. The implementation presented is based on [6].

11.5.1 Finding the vector $\mathbf{w}_i$

The first task is to find an efficient way to find the vector $\mathbf{w}_i$. Without lost of generality, we will assume that $\|\mathbf{w}_i\|_2 = 1$. This rule out the possibility of using item 4 of Theorem 11.5.2 to find $\mathbf{w}_i$. Though this complicates the procedure to find $\mathbf{w}_i$, it is a small price to pay to get a more efficient procedure to compute $G_iA_{i-1}G_i$ later.

We have

$$G_i = \begin{pmatrix} \mathrm{Id}_i & 0 \\ 0 & H_{\mathbf{w}_i} \end{pmatrix} ,$$

where $M = H_{\mathbf{w}_i}$ is an $(n-i)\times(n-i)$ matrix with the components

$$m_{j,k} = \begin{cases} 1 - 2w_{i,j}^2 & \text{if} \quad j = k \\ -2w_{i,j}w_{i,k} & \text{if} \quad j \neq k \end{cases}$$

for $1 \leq j,k \leq n-i$, and $w_{i,j}$ is the j^{th} coordinates of the vector $\mathbf{w}_i$. We can write A_{i-1} as

$$A_{i-1} = \begin{pmatrix} B & C \\ D & E \end{pmatrix} ,$$

where B is an $i\times i$ matrix in Hessemberg form, C is an $i\times(n-i)$ matrix, D is an $(n-i)\times i$ matrix of the form

$$D = \begin{pmatrix} 0 & 0 & \ldots & 0 & d_{1,i} \\ 0 & 0 & \ldots & 0 & d_{2,i} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & d_{n-i,i} \end{pmatrix}$$

and E is an $(n-i)\times(n-i)$ matrix. We then have

$$G_iA_{i-1}G_i = \begin{pmatrix} \mathrm{Id}_i & 0 \\ 0 & H_{\mathbf{w}_i} \end{pmatrix}\begin{pmatrix} B & C \\ D & E \end{pmatrix}\begin{pmatrix} \mathrm{Id}_i & 0 \\ 0 & H_{\mathbf{w}_i} \end{pmatrix} = \begin{pmatrix} B & CH_{\mathbf{w}_i} \\ H_{\mathbf{w}_i}D & H_{\mathbf{w}_i}EH_{\mathbf{w}_i} \end{pmatrix} ,$$

where

$$H_{\mathbf{w}_i} D = H_{\mathbf{w}_i} \begin{pmatrix} 0 & 0 & \dots & 0 & d_{1,i} \\ 0 & 0 & \dots & 0 & d_{2,i} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & d_{n-i,i} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \dots & 0 & d_{1,i} - 2\sum_{k=1}^{n-i} w_{i,1} w_{i,k} d_{k,i} \\ 0 & 0 & \dots & 0 & d_{2,i} - 2\sum_{k=1}^{n-i} w_{i,2} w_{i,k} d_{k,i} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & d_{n-i,i} - 2\sum_{k=1}^{n-i} w_{i,n-i} w_{i,k} d_{k,i} \end{pmatrix} .$$

We need to have

$$d_{m,i} - 2w_{i,m} \sum_{k=1}^{n-i} w_{i,k} d_{k,i} = 0 \tag{11.5.1}$$

for $m = 2, 3, \dots, n-i$. Let $\sigma_i = \sum_{k=1}^{n-i} w_{i,k} d_{k,i}$. We have

$$\sum_{m=1}^{n-i} \left(d_{m,i} - 2w_{i,m}\sigma_i\right)^2 = \sum_{m=1}^{n-i} d_{m,i}^2 - 4\sigma_i \sum_{m=1}^{n-i} w_{i,m} d_{m,i} + 4\sigma_i^2 \sum_{m=1}^{n-i} w_{i,m}^2 = \sum_{m=1}^{n-i} d_{m,i}^2 \ , \tag{11.5.2}$$

where we have used $\|\mathbf{w}_i\|_2 = 1$ and the definition of σ_i to get the last equality. From (11.5.1), we have $d_{m,i} - 2w_{i,m}\sigma_i = 0$ for $m = 2, 3, \dots, n-i$. Hence, we get from (11.5.2) that

$$d_{1,i} - 2w_{i,1}\sigma_i = \epsilon \sqrt{\sum_{m=1}^{n-i} d_{m,i}^2} \ , \tag{11.5.3}$$

where $\epsilon = 1$ or -1. Let

$$s_i = \sqrt{\sum_{m=1}^{n-i} d_{m,i}^2} \ . \tag{11.5.4}$$

We may assume that $s_i \neq 0$. If $s_i = 0$, then we may take $\mathbf{w}_i = \mathbf{0}$ because $d_{m,i} = 0$ for $m = 1, 2, \dots, n-i$. It follows from the definition of σ_i, (11.5.1), (11.5.3) and $\|\mathbf{w}_i\|_2 = 1$ that

$$\begin{aligned} \sigma_i &= w_{i,1} d_{1,i} + \sum_{k=2}^{n-i} w_{i,k} d_{k,i} = \left(2w_{i,1}\sigma_i + \epsilon s_i\right) w_{i,1} + \sum_{k=2}^{n-i} w_{i,k}\left(2w_{i,k}\sigma_i\right) \\ &= \epsilon s_i w_{i,1} + 2\sigma_i \sum_{k=1}^{n-i} w_{i,k}^2 = \epsilon s_i w_{i,1} + 2\sigma_i \ . \end{aligned}$$

Thus, $\sigma_i = -\epsilon s_i w_{i,1}$. If we substitute this expression in (11.5.3), we get $d_{1,i} + 2\epsilon w_{i,1}^2 s_i = \epsilon s_i$. Hence,

$$w_{i,1}^2 = \frac{s_i - \epsilon d_{1,i}}{2s_i} \ . \tag{11.5.5}$$

To avoid the possibility of a subtraction of two numbers almost equal, we take $\epsilon = -\operatorname{sgn}(d_{1,i})$. The formulae to compute $w_{i,k}$ for $k > 1$ will involve a division by $w_{i,1}$. It is therefore important to compute $w_{i,1}$ as accurately as we can. The formula to compute $w_{i,1}$ is

$$w_{i,1}^2 = \frac{s_i + |d_{1,i}|}{2s_i} \ . \tag{11.5.6}$$

For the other components of $\mathbf{w}_i$, we use (11.5.1) to get

$$0 = d_{m,i} - 2w_{i,m}\sigma_i = d_{m,i} + 2w_{i,m}\left(\epsilon s_i w_{i,1}\right)$$

for $m = 2, 3, \ldots, n-i$. Thus

$$w_{i,m} = \frac{-d_{m,i}}{2\epsilon w_{i,1} s_i} = \operatorname{sgn}(d_{1,i})\frac{d_{m,i}}{2w_{i,1}s_i} \tag{11.5.7}$$

for $m = 2, 3, \ldots, n-i$.

Note that

$$\begin{aligned}\mathbf{w}_i^\top \mathbf{w}_i &= \sum_{k=1}^{n-i} w_{i,k}^2 = w_{i,1}^2 + \sum_{k=2}^{n-i} w_{i,k}^2 = \frac{s_i + \epsilon d_{1,i}}{2s_i} + \sum_{k=2}^{n-i}\frac{d_{k,i}^2}{4w_{i,1}^2 s_i^2} \\ &= \frac{s_i + \epsilon d_{1,i}}{2s_i} + \frac{1}{4w_{i,1}^2 s_i^2}\left(\sum_{k=1}^{n-i} d_{k,i}^2 - d_{1,i}^2\right) = \frac{s_i + \epsilon d_{1,i}}{2s_i} + \frac{1}{4w_{i,1}^2 s_i^2}\left(s_i^2 - d_{1,i}^2\right) \ . \end{aligned} \tag{11.5.8}$$

Since

$$4w_{i,1}^2 s_i^2 = 4\left(\frac{s_i + \epsilon d_{1,i}}{2s_i}\right)s_i^2 = 2\left(s_i + \epsilon d_{1,i}\right)s_i \ ,$$

we get from (11.5.8) that

$$\begin{aligned}\mathbf{w}_i^\top \mathbf{w}_i &= \frac{s_i + \epsilon d_{1,i}}{2s_i} + \frac{1}{2\left(s_i + \epsilon d_{1,i}\right)s_i}\left(s_i^2 - d_{1,i}^2\right) \\ &= \frac{\left(s_i + \epsilon d_{1,i}\right)\left(s_i + \epsilon d_{1,i}\right) + \left(s_i^2 - d_{1,i}^2\right)}{2\left(s_i + \epsilon d_{1,i}\right)s_i} = \frac{2s_i^2 + 2\epsilon d_{1,i}s_i}{2\left(s_i + \epsilon d_{1,i}\right)s_i} = 1\end{aligned}$$

as expected.

(11.5.6) and (11.5.7) are the formulae used to find the vectors $\mathbf{w}_i \in \mathbb{R}^{n-i}$ for $i = 1, 2, \ldots, n-2$.

11.5.2 Computing $\mathbf{G_i A_{i-1} G_i}$

We now give an efficient way to compute $G_i A_{i-1} G_i$ for each i. Let

$$\mathbf{v}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \operatorname{sgn}(d_{1,i})s_i + d_{1,i} \\ d_{2,i} \\ \vdots \\ d_{n-i,i} \end{pmatrix} . \tag{11.5.9}$$

Recall that the vector $\mathbf{w}_i$ is represented algebraically by a $(n-i)\times 1$ column matrix. We get from (11.5.5) and (11.5.7) that

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{w}_i \end{pmatrix} = \frac{\operatorname{sgn}(d_{1,i})}{2w_{i,1}s_i}\mathbf{v}_i \ .$$

Thus

$$\begin{aligned} G_i = \begin{pmatrix} \mathrm{Id}_i & 0 \\ 0 & H_{\mathbf{w}_i} \end{pmatrix} &= \mathrm{Id}_n - 2\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{w}_i \end{pmatrix}\begin{pmatrix} 0 & 0 & \cdots & 0 & \mathbf{w}_i^\top \end{pmatrix} \\ &= \mathrm{Id}_n - 2\left(\frac{\operatorname{sgn}(d_{1,i})}{2w_{i,1}s_i}\right)^2 \mathbf{v}_i\mathbf{v}_i^\top = \mathrm{Id}_n - \frac{1}{2w_{i,1}^2 s_i^2}\mathbf{v}_i\mathbf{v}_i^\top \ . \end{aligned}$$

From (11.5.6), we get

$$2w_{i,1}^2 s_i^2 = 2\left(\frac{s_i + |d_{1,i}|}{2s_i}\right)s_i^2 = (s_i + |d_{1,i}|)\, s_i \ .$$

Hence, if we define

$$\alpha_i = \frac{1}{(s_i + |d_{1,i}|)\, s_i} \ , \tag{11.5.10}$$

then $G_i = \mathrm{Id}_n - \alpha_i \mathbf{v}_i\mathbf{v}_i^\top$. Let

$$\begin{aligned} &\mathbf{x}_i = \alpha_i A_{i-1}\mathbf{v}_i \quad , \quad \mathbf{y}_i = \alpha_i A_{i-1}^\top \mathbf{v}_i \quad , \quad \mu_i = \frac{1}{2}\alpha_i \mathbf{v}_i^\top \mathbf{x}_i \ , \\ &\mathbf{p}_i = \mathbf{y}_i - \mu_i\mathbf{v}_i \quad \text{and} \quad \mathbf{q}_i = \mathbf{x}_i - \mu_i \mathbf{v}_i \ . \end{aligned} \tag{11.5.11}$$

We have

$$\begin{aligned} G_iA_{i-1}G_i &= \left(\mathrm{Id}_n - \alpha_i\mathbf{v}_i\mathbf{v}_i^\top\right) A_{i-1} \left(\mathrm{Id}_n - \alpha_i\mathbf{v}_i\mathbf{v}_i^\top\right) \\ &= A_{i-1} - \alpha_i\mathbf{v}_i\mathbf{v}_i^\top A_{i-1} - \alpha_i A_{i-1}\mathbf{v}_i\mathbf{v}_i^\top + \alpha_i^2\mathbf{v}_i\mathbf{v}_i^\top A_{i-1}\mathbf{v}_i\mathbf{v}_i^\top \\ &= A_{i-1} - \mathbf{v}_i\left(\alpha_i\mathbf{v}_i^\top A_{i-1} - \frac{1}{2}\alpha_i^2\mathbf{v}_i^\top A_{i-1}\mathbf{v}_i\mathbf{v}_i^\top\right) \\ &\qquad - \left(\alpha_i A_{i-1}\mathbf{v}_i - \frac{1}{2}\alpha_i^2\mathbf{v}_i\mathbf{v}_i^\top A_{i-1}\mathbf{v}_i\right)\mathbf{v}_i^\top \\ &= A_{i-1} - \mathbf{v}_i\left(\mathbf{y}_i^\top - \mu_i\mathbf{v}_i^\top\right) - \left(\mathbf{x}_i - \mu_i\mathbf{v}_i\right)\mathbf{v}_i^\top \\ &= A_{i-1} - \mathbf{v}_i\mathbf{p}_i^\top - \mathbf{q}_i\mathbf{v}_i^\top \ . \end{aligned} \tag{11.5.12}$$

We can combine (11.5.4), (11.5.9), (11.5.10), (11.5.11) and (11.5.12) to get the following code.

Code 11.5.6 (Householder Reduction Algorithm)

To produce a matrix B in the Hessemberg form which is conjugate to the given matrix A.
Input: The matrix A.
Output: The matrix B.

```
% function B = householder(A)

function B = householder(A)
  dim = size(A,1);

  for i = 1:dim-2
    v = zeros(dim,1);
    z = v;

    % s_i
    s = norm(A(i+1:dim,i));

    if ( s != 0 )
      % alpha_i
      alpha = 1/( (s + abs(A(i+1,i)))*s );

      % v_i
      v(i+1,1) = sign(A(i+1,i))*s + A(i+1,i);
      z(i+1,1) = alpha*v(i+1,1);
      v(i+2:dim,1) = A(i+2:dim,i);
      z(i+2:dim,1) = alpha*v(i+2:dim,1);

      % x_i  and  y_i
      x = A*z;
      y = A'*z;

      % mu_i
      mu = (alpha * (v' *x))/2;

      % p_i  and  q_i
      z = mu*v;
      p = y - z;
      q = x - z;

      %  A_i
      A = A - v * p' - q*v';
    end
  end
  B = A;
end
```

Remark 11.5.7

1. The previous algorithm requires $O(4n^2)$ multiplications to compute A_i from A_{i-1}. The direct product $G_i A_{i-1} G_i$ requires $O(2n^3)$ multiplications.

2. If A is symmetric, it is easy to prove by induction that the matrices A_i are symmetric for all i because the G_i are symmetric. The resulting matrix B is a symmetric tridiagonal matrix. The previous algorithm may also be improved because $\mathbf{x}_i = \mathbf{y}_i$ and $\mathbf{p}_i = \mathbf{q}_i$.

♠

Since the resulting matrix T given by the Householder Reduction Algorithm is conjugate to the given matrix A, the matrices A and T have the same eigenvalues. In particular, if A is symmetric, then T is a symmetric tridiagonal matrix. The next section will present a method to compute the eigenvalues of a symmetric tridiagonal matrix T, hence the eigenvalues of A.

We now present a theoretical method to find the eigenvalues of a symmetric tridiagonal $n \times n$ matrix $T = (t_{i,j})$. We say theoretical because it is not the best method to compute eigenvalues of a matrix. Let

$$M_i = \begin{pmatrix} t_{1,1}-\lambda & t_{1,2} & 0 & 0 & \dots & 0 & 0 & 0 \\ t_{2,1} & t_{2,2}-\lambda & t_{2,3} & 0 & \dots & 0 & 0 & 0 \\ 0 & t_{3,2} & t_{3,3}-\lambda & t_{3,4} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & t_{i-1,i-2} & t_{i-1,i-1}-\lambda & t_{i-1,i} \\ 0 & 0 & 0 & 0 & \dots & 0 & t_{i,i-1} & t_{i,i}-\lambda \end{pmatrix}$$

and $p_i(\lambda) = \det M_i$ for $i = 1, 2, \dots, n$. We have $T = M_n$. Let $p_0(\lambda) = 1$. Developing the determinant of M_i along the last row and using the symmetry of T (in particular, $t_{i,i-1} = t_{i-1,i}$), we get

$$p_i(\lambda) = (t_{i,i}-\lambda)p_{i-1}(\lambda) - t_{i,i-1}^2 p_{i-2}(\lambda) \quad , \quad i = 2, 4, \dots, n \; . \tag{11.5.13}$$

Only $3n-6$ multiplications are needed to compute the determinant of T; a lot less than the $n!$ multiplication needed for a full ordinary $n \times n$ matrix.

Theorem 11.5.8

Consider a symmetric tridiagonal matrix T. If $t_{i,i-1} \neq 0$ for $2 \leq i \leq n$, then the roots of p_i are distinct and between any two consecutive roots of p_i there is a root of p_{i-1}.

Proof.

1) We first show that p_i and p_{i-1} cannot have a common root. Suppose that c is a common root of p_i and p_{i-1} for some i. If $i \geq 2$, we get from (11.5.13) that c is also a root of p_{i-2} because $t_{i,i-1} \neq 0$. Thus, inductively, c is a root of p_0 which is impossible because $p_0(x) = 1$ for all x.

2) We prove by induction that the roots of p_i are distinct and between any two roots of p_i there is a root of p_{i-1}

We have $p_1(\lambda) = t_{1,1} - \lambda$ and $p_2(\lambda) = (t_{2,2} - \lambda)(t_{1,1} - \lambda) - t_{2,1}^2$. The only root of p_1 is $t_{1,1}$. Since $p_2(t_{1,1}) = -t_{2,1}^2 < 0$ and p_2 is a polynomial of degree 2 of the form $p_2(\lambda) = \lambda^2 + l.o.t.$ ($l.o.t.$ stands for lower order terms in λ), it has two distinct roots; one smaller than $t_{1,1}$ and one bigger that $t_{1,1}$. Thus, the hypothesis of induction is true for $i = 2$.

Let's assume that the hypothesis of induction is true for i. We have that

$$p_{i+1}(\lambda) = (t_{i+1,i+1} - \lambda)p_i(\lambda) - t_{i+1,i}^2 p_{i-1}(\lambda) \quad .$$

Let α be the largest root of p_i. We assume first that i is even. We have

$$p_{i-1}(\lambda) = (-1)^{i-1}\lambda^{i-1} + l.o.t. = -\lambda^{i-1} + l.o.t.$$

Since p_{i-1} does not have roots bigger than α because the roots of p_{i-1} are between the roots of p_i by the induction hypothesis, we have that $p_{i-1}(\alpha) < 0$. Since

$$p_{i+1}(\alpha) = (t_{i+1,i+1} - \alpha)p_i(\alpha) - t_{i+1,i}^2 p_{i-1}(\alpha) = -t_{i+1,i}^2 p_{i-1}(\alpha) \quad ,$$

we have that $p_{i+1}(\alpha) > 0$. But we also have that

$$p_{i+1}(\lambda) = (-1)^{i+1}\lambda^{i+1} + l.o.t. = -\lambda^{i+1} + l.o.t.$$

Thus, there must be a root of p_{i+1} greater than α.

Similarly, if i is odd, we have

$$p_{i-1}(\lambda) = (-1)^{i-1}\lambda^{i-1} + l.o.t. = \lambda^{i-1} + l.o.t.$$

Since p_{i-1} does not have roots bigger than α because the roots of p_{i-1} are between the roots of p_i by the induction hypothesis, we have that $p_{i-1}(\alpha) > 0$. Since

$$p_{i+1}(\alpha) = (t_{i+1,i+1} - \alpha)p_i(\alpha) - t_{i+1,i}^2 p_{i-1}(\alpha) = -t_{i+1,i}^2 p_{i-1}(\alpha) \quad ,$$

we have that $p_{i+1}(\alpha) < 0$. But we also have that

$$p_{i+1}(\lambda) = (-1)^{i+1}\lambda^{i+1} + l.o.t. = \lambda^{i+1} + l.o.t.$$

Thus, again, there must be a root of p_{i+1} greater than α.

Proceeding as we did for the largest root of p_i, we can show that p_{i+1} has a root smaller than the smallest root of p_i.

Let's $\alpha < \beta$ be two consecutive roots of p_i. Since there is one (and only one) root of p_{i-1} between two consecutive roots of p_i by the hypothesis of induction, $p_{i-1}(\alpha)$ and $p_{i-1}(\beta)$ must be of opposite sign. However, since

$$p_{i+1}(\alpha) = (t_{i+1,i+1} - \alpha)p_i(\alpha) - t_{i+1,i}^2 p_{i-1}(\alpha) = -t_{i+1,i}^2 p_{i-1}(\alpha)$$

and

$$p_{i+1}(\beta) = (t_{i+1,i+1} - \beta)p_i(\beta) - t_{i+1,i}^2 p_{i-1}(\beta) = -t_{i+1,i}^2 p_{i-1}(\beta) \quad ,$$

we have that $p_{i+1}(\alpha)$ and $p_{i+1}(\beta)$ must also be of opposite sign. So there is a root of p_{i+1} between the two consecutive roots, α and β, of p_i.

The distinct roots β_j for $1 \leq j \leq i$ of p_i divide the real line into $n+1$ subintervals $]-\infty, \beta_1[$, $]\beta_j, \beta_{j+1}[$ for $1 \leq j < i$, and $]\beta_i, \infty[$. We have shown that there is a root of p_{i+1} in each of these subintervals. Since p_{i+1} is of degree $i+1$, those are all the roots of p_{i+1} and they separate the roots of p_i. This complete the proof by induction. ∎

The number $N_i(\beta)$ of **sign agreements** at $\lambda = \beta \in \mathbb{R}$ is the number of times that $\operatorname{sgn}(p_j(\beta)) = \operatorname{sgn}(p_{j-1}(\beta))$ for $j = 1, 2, \ldots, i$. By convention, we assume that there is a sign agreement when $p_j(\beta) = 0$. For instance, there are three sign agreements in the sequence $+,+,+,-,-$. There are also three sign agreements in the sequence $+,+,0,-,-$.

The following result follows from the previous theorem.

Proposition 11.5.9

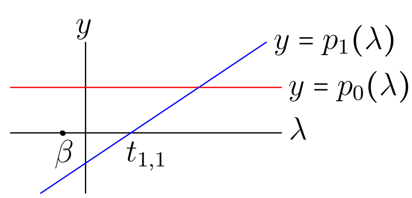
Consider a symmetric tridiagonal matrix T. $N_i(\beta)$ is equal to the number of roots of p_i which are greater or equal to $\beta \in \mathbb{R}$.

Proof.
As in the previous theorem, we will first assume that $t_{i,i-1} \neq 0$ for $2 \leq i \leq n$. So, we have that the roots p_i are distinct and between any two consecutive roots of p_i there is a root of p_{i-1}.

The proof is by induction on i, the degree of the polynomial p_i.

Since $p_0(\lambda) = 1$ and $p_1(\lambda) = t_{1,1} - \lambda$, we have that

$$N_1(\beta) = \begin{cases} 1 & \text{if} \quad \beta \leq t_{1,1} \\ 0 & \text{if} \quad \beta > t_{1,1} \end{cases}$$



Effectively, p_1 has one root greater or equal to β if $\beta \leq t_{1,1}$ and no root greater or equal to β if $\beta > t_{1,1}$. Thus, the induction hypothesis is true for $i = 1$.

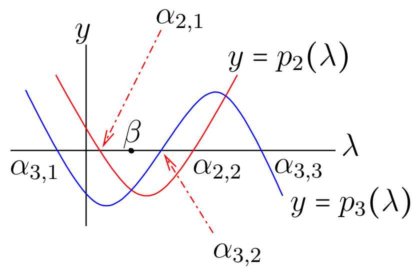
Let's assume that the result is true for i; namely, p_i has $N_i(\beta)$ roots greater or equal to β.

Let $\alpha_{i,1} < \alpha_{i,2} < \ldots < \alpha_{i,i}$ and $\alpha_{i+1,1} < \alpha_{i+1,2} \ldots < \alpha_{i+1,i+1}$ be the roots of p_i and p_{i+1} respectively. From the previous theorem, we know that

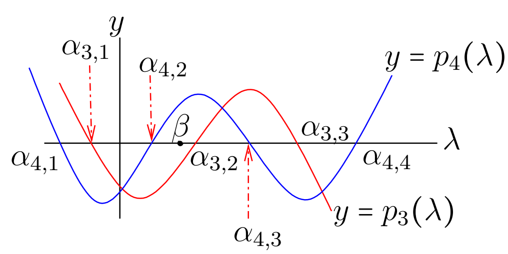
$$\alpha_{i+1,1} < \alpha_{i,1} < \alpha_{i+1,2} < \alpha_{i,2} < \ldots < \alpha_{i,i} < \alpha_{i+1,i+1} \quad .$$

Suppose that $N_i(\beta) = i - j$. By induction, this means that $\beta \leq \alpha_{i,1}$ if $j = 0$, $\alpha_{i,j} < \beta \leq \alpha_{i,j+1}$ if $0 < j < i$, or $\beta > \alpha_{i,i}$ if $j = i$.

When i is even, we have $p_i(\lambda) = \lambda^i + l.o.t.$ and $p_{i+1}(\lambda) = -\lambda^{i+1} + l.o.t.$. We have sketched the case $i = 2$ in the following figure.



When i is odd, we have $p_i(\lambda) = -\lambda^i + l.o.t.$ and $p_{i+1}(\lambda) = \lambda^{i+1} + l.o.t.$. We have sketched the case $i = 3$ in the following figure.



We consider first the case for $0 < j < i$. We have that $\alpha_{i,j} < \alpha_{i+1,j+1} \leq \alpha_{i,j+1}$. Either $\alpha_{i,j} < \beta \leq \alpha_{i+1,j+1}$ or $\alpha_{i+1,j+1} < \beta \leq \alpha_{i,j+1}$

1. When $\alpha_{i,j} < \beta \leq \alpha_{i+1,j+1}$, we have that $\text{sgn}(p_i(\beta)) = \text{sgn}(p_{i+1}(\beta))$ or $p_{i+1}(\beta) = 0$. Thus, we have an additional sign agreement and $N_{i+1}(\beta) = 1+N_i(\beta) = 1+(i-j)$. We effectively have $i+1-j$ roots of p_{i+1} greater or equal to β; namely, $\alpha_{i+1,k}$ for $k > j$.

2. When $\alpha_{i+1,j+1} < \beta \leq \alpha_{i,j+1}$, we have that $\text{sgn}(p_i(\beta)) \neq \text{sgn}(p_{i+1}(\beta))$. Thus, we have no additional sign agreement and $N_{i+1}(\beta) = N_i(\beta) = i-j$. We effectively have $(i+1)-(j+1)$ roots of p_{i+1} greater or equal to β; namely, $\alpha_{i+1,k}$ for $k > j+1$.

A similar argument can be used for $j = 0$ and $j = i$ to prove that p_{i+1} has $N_{i+1}(\beta)$ roots greater or equal to β.

To show that the assumption that $t_{i,i-1} \neq 0$ for $2 \leq i \leq n$ is not needed, we note that any symmetric tridiagonal matrix with some null elements on its subdiagonal is the limit of symmetric tridiagonal matrices with no null element on its subdiagonal. We leave the details to the reader. ∎

We can use this theorem and the bisection method to find all the roots of p_n.

Algorithm 11.5.10

Algorithm Suppose that $[a, b]$ is an interval containing all the roots of p_n. Such an interval can be found with the help of Gerschgorin theorem. We obviously have that

$N(a) = n$ and $N(b) = 0$. To find all the roots of p_n, one may proceed as follows for $i = 1, 2, \ldots, n$.

1. Let $\alpha = a$ and $\beta = b$.
2. Let $m = (\alpha + \beta)/2$. m is the midpoint of the interval $[\alpha, \beta]$.
3. If $N_n(m) = i$, then a single root of p_n exists in $[m, \beta]$. The bisection method may be used to approximate the root of p_n in the interval $[m, \beta]$.
   If $N_n(m) > i$, set $\alpha = m$ and go back to (2).
   If $N_n(m) < i$, set $\beta = m$ and go back to (2).
4. Let $\beta = m$ and go back to (1) until all n roots of p_n have been found.

This method to find all the roots of p_n may not work if the distance between two roots of p_n is smaller than the accuracy of the computer used. If $t_{i,i-1} \neq 0$ for $2 \leq i \leq n$ is not satisfied, there may be roots of algebraic multiplicity greater than one. The method will not provide the algebraic multiplicity of the roots. At the third step of the algorithm above, if $N_n(m) > i$ for all computed midpoints. One may temporary say that b is a root of algebraic multiplicity $N_n(m) - i + 1$ and proceed to step (4) with $i = N_n(m) + 1$. The roots of p_n very closed to b have to be determined using another approach.

11.6 QR Algorithm

The mean goal of this section is to present a method to compute the eigenvalues of a symmetric tridiagonal matrix T.

Let A be an $n \times n$ matrix. Starting with $A_0 = A$, we produce recursively a sequence of $n \times n$ matrices $\{A_i\}_{i=0}^{\infty}$ which are all conjugated to A as follows. Given the $n \times n$ matrix A_i, we write A_i as $A_i = Q_i R_i$, where Q_i is an orthogonal matrix and R_i is an upper-triangular matrix. The next matrix is defined by $A_{i+1} = R_i Q_i$.

We have that

$$A_{i+1} = R_i Q_i = Q_i^{\top} A_i Q_i \ .$$

By induction,

$$A_{i+1} = Q_i^{\top} Q_{i-1}^{\top} \ldots Q_0^{\top} A_0 Q_0 \ldots Q_{i-1} Q_i = (Q_0 \ldots Q_{i-1} Q_i)^{\top} A_0 (Q_0 \ldots Q_{i-1} Q_i) \ . \tag{11.6.1}$$

Thus A_i is orthogonally conjugate to A. In particular, A_i and A have the same eigenvalues with the same algebraic multiplicity.

In Section 11.6.2, we explain how to express a $n \times n$ matrix A as the product $A = QR$, where Q is an orthogonal matrix and R is an upper-triangular matrix. The tool that we will use to do this is the Gram-Schmidt orthogonalization process that we cover in the next section.

11.6.1 Gram-Schmidt Orthogonalization Process

Definition 11.6.1 (Gram-Schmidt Orthogonalization Process)

Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{R}^n$ and let $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ be a subset of $\mathbb{R}^n$. We define the set $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ as follows:

1. $\mathbf{v}_1 = \mathbf{u}_1$
2. For $2 \leq i \leq k$, $\mathbf{v}_i = \mathbf{u}_i - \displaystyle\sum_{j=1}^{i-1} r_{j,i}\mathbf{v}_j$, where

$$r_{j,i} = \begin{cases} \dfrac{\langle \mathbf{v}_j, \mathbf{u}_i \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle} & \text{if} \quad \mathbf{v}_j \neq \mathbf{0} \\ 0 & \text{if} \quad \mathbf{v}_j = \mathbf{0} \end{cases}$$

for $1 \leq j < i$.

The set $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ given by the Gram-Schmidt Orthogonalization Process has the following properties.

Proposition 11.6.2

Let $\mathbf{u}_j$ and $\mathbf{v}_j$ for $1 \leq j \leq k$ be the vectors defined in Definition 11.6.1. Let $S_i = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_i\}$ and $V_i = \text{span}(S_i)$ for $1 \leq i \leq k$.

1. $V_i = \text{span}(T_i)$ where $T_i = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i\}$.
2. $\langle \mathbf{v}_p, \mathbf{v}_q \rangle = 0$ for $1 \leq p < q \leq i$.
3. $\mathbf{v}_i = 0$ if and only if $\mathbf{u}_i \in V_{i-1}$.
4. If S_k is a basis of V_k, then T_k is an orthogonal basis of V.

Remark 11.6.3

If P_j is the orthogonal projection of $\mathbb{R}^n$ onto V_j for $1 \leq j \leq k$, we have that $\mathbf{v}_i = \mathbf{u}_i - P_{i-1}(\mathbf{u}_i)$. Items 2 and 4 of the previous proposition imply that all finite dimensional vector spaces have an orthogonal basis. ♠

Proof.

1) The proof is by induction on i.

Since $\mathbf{v}_1 = \mathbf{u}_1$, we have that $V_1 = \text{span}(T_1)$. So, the result is true for $i = 1$.

Suppose that the result is true for i; namely, $V_i = \text{span}(T_i)$. To show that $V_{i+1} = \text{span}(T_{i+1})$, it suffices to show that $\mathbf{u}_{i+1} \in \text{span}(T_{i+1})$ because $\mathbf{u}_j \in \text{span}(T_i) \subset \text{span}(T_{i+1})$ for $1 \leq j \leq i$ by the hypothesis of induction. From

$$\mathbf{v}_{i+1} = \mathbf{u}_{i+1} - \sum_{j=1}^{i} r_{j,i+1}\mathbf{v}_j \ ,$$

we get

$$\mathbf{u}_{i+1} = \mathbf{v}_{i+1} + \sum_{j=1}^{i} r_{j,i+1}\mathbf{v}_j \ .$$

Thus $\mathbf{u}_{i+1} \in \operatorname{span}(T_{i+1})$.

2) The proof is again by induction on i.

Since $\mathbf{v}_2 = \mathbf{u}_2 - r_{1,2}\mathbf{v}_1$ with

$$r_{1,2} = \begin{cases} \dfrac{\langle \mathbf{v}_1, \mathbf{u}_2\rangle}{\langle \mathbf{v}_1, \mathbf{v}_1\rangle} & \text{if} \quad \mathbf{v}_1 \neq \mathbf{0} \\ 0 & \text{if} \quad \mathbf{v}_1 = \mathbf{0} \end{cases}$$

we get

$$\langle \mathbf{v}_1, \mathbf{v}_2\rangle = \langle \mathbf{v}_1, \mathbf{u}_2 - r_{1,2}\mathbf{v}_1\rangle = \langle \mathbf{v}_1, \mathbf{u}_2\rangle - r_{1,2}\langle \mathbf{v}_1, \mathbf{v}_1\rangle = 0 \ .$$

So the result is true for $i = 2$.

Suppose that the result is true for i; namely, $\langle \mathbf{v}_p, \mathbf{v}_q\rangle = 0$ for $1 \le p < q \le i$. To prove that $\langle \mathbf{v}_p, \mathbf{v}_q\rangle = 0$ for $1 \le p < q \le i+1$, it suffices to prove that $\langle \mathbf{v}_m, \mathbf{v}_{i+1}\rangle = 0$ for $1 \le m \le i$. Since $\mathbf{v}_{i+1} = \mathbf{u}_{i+1} - \sum_{j=1}^{i} r_{j,i+1}\mathbf{v}_j$, where

$$r_{j,i+1} = \begin{cases} \dfrac{\langle \mathbf{v}_j, \mathbf{u}_{i+1}\rangle}{\langle \mathbf{v}_j, \mathbf{v}_j\rangle} & \text{if} \quad \mathbf{v}_j \neq \mathbf{0} \\ 0 & \text{if} \quad \mathbf{v}_j = \mathbf{0} \end{cases}$$

we get

$$\begin{aligned} \langle \mathbf{v}_m, \mathbf{v}_{i+1}\rangle &= \left\langle \mathbf{v}_m, \mathbf{u}_{i+1} - \sum_{j=1}^{i} r_{j,i+1}\mathbf{v}_j \right\rangle = \langle \mathbf{v}_m, \mathbf{u}_{i+1}\rangle - \sum_{j=1}^{i} r_{j,i+1}\langle \mathbf{v}_m, \mathbf{v}_j\rangle \\ &= \langle \mathbf{v}_m, \mathbf{u}_{i+1}\rangle - r_{m,i+1}\langle \mathbf{v}_m, \mathbf{v}_m\rangle = 0 \end{aligned}$$

for $1 \le m \le i$. So the result is true for $i+1$.

3) If $\mathbf{v}_i = \mathbf{0}$, then $\mathbf{u}_i - \sum_{j=1}^{i-1} r_{j,i}\mathbf{v}_j = \mathbf{0}$. This gives $\mathbf{u}_i = \sum_{j=1}^{i-1} r_{j,i}\mathbf{v}_j$. Thus $\mathbf{u}_i \in V_{i-1}$ because $V_{i-1} = \operatorname{span}(T_{i-1})$ by (1). Conversely, if $\mathbf{u}_i \in V_{i-1} = \operatorname{span}(T_{i-1})$, we get from Proposition 11.1.9 that $\mathbf{u}_i = \sum_{j=1}^{i} a_j\mathbf{v}_j$ with $a_j = r_{j,i}$. Thus, $\mathbf{v}_{i+1} = \mathbf{u}_i - \sum_{j=1}^{i-1} r_{j,i}\mathbf{v}_j = \mathbf{0}$.

4) If we use (2), this follows from (1) with $i = k$. ∎

Proposition 11.6.4

Let $\langle\cdot,\cdot\rangle$ be the standard scalar product on $\mathbb{R}^n$ and let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ be an orthonormal basis of a subspace V of $\mathbb{R}^n$. Let $Q = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_k \end{pmatrix}$. Then

1. $Q^\top Q = \mathrm{Id}_k$.
2. The orthogonal projection P on V is given by $P = QQ^\top$.
3. P is symmetric (i.e. $P^\top = P$).
4. $P^2 = P$.
5. $P(\mathrm{Id}_n - P) = (\mathrm{Id}_n - P)P = 0$.
6. $(\mathrm{Id}_n - P)Q = 0$.

Proof.

1) The component on the i^{th} row and j^{th} column of $Q^\top Q$ is

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{i,j} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

2) Given $\mathbf{v} \in \mathbb{R}^n$,

$$QQ^\top \mathbf{v} = Q \begin{pmatrix} \mathbf{v}_1^\top \mathbf{x} \\ \mathbf{v}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{v}_k^\top \mathbf{x} \end{pmatrix} = \sum_{j=1}^{k} \langle \mathbf{v}_j, \mathbf{x} \rangle \, \mathbf{v}_j = \sum_{j=1}^{k} \frac{\langle \mathbf{v}_j, \mathbf{x} \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle} \mathbf{v}_j = P\mathbf{x}$$

because $\langle \mathbf{v}_j, \mathbf{x} \rangle = \mathbf{v}_1^\top \mathbf{x}$ and $\langle \mathbf{v}_j, \mathbf{v}_j \rangle = 1$ for $1 \leq j \leq k$.

3) We have

$$P^\top = (QQ^\top)^\top = (Q^\top)^\top Q^\top = QQ^\top = P \quad .$$

4) We have

$$P^2 = (QQ^\top)(QQ^\top) = Q(Q^\top Q)Q^\top = Q \,\mathrm{Id}_n\, Q^\top = QQ^\top = P \quad .$$

5) We have

$$P(\mathrm{Id}_n - P) = P - P^2 = P - P = 0 \qquad \text{and} \qquad (\mathrm{Id}_n - P)P = P - P^2 = P - P = 0 \quad .$$

6) We have

$$(\mathrm{Id}_n - P)Q = Q - (QQ^\top)Q = Q - Q(Q^\top Q) = Q - Q\,\mathrm{Id}_k = Q - Q = 0 \quad .$$

∎

11.6.2 Normalized QR Decomposition

The Gram-Schmidt orthogonalization process given in Definition 11.6.1 can be summarize as follows. Consider the $n \times k$ matrices $A = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_k \end{pmatrix}$ and $Q_0 = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_k \end{pmatrix}$. Let R_0 be the $k \times k$ upper-triangular matrix

$$\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \ldots & r_{1,k} \\ r_{2,1} & r_{2,2} & r_{2,3} & \ldots & r_{2,k} \\ r_{3,1} & r_{3,2} & r_{3,3} & \ldots & r_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{k,1} & r_{k,2} & r_{k,3} & \ldots & r_{k,k} \end{pmatrix} ,$$

where

$$r_{j,i} = \begin{cases} 0 & \text{if} \quad j > i \\ 1 & \text{if} \quad j = i \\ \dfrac{\langle \mathbf{v}_j, \mathbf{u}_i \rangle}{\langle \mathbf{v}_j, \mathbf{v}_j \rangle} & \text{if} \quad j < i \quad \text{and} \quad \mathbf{v}_j \neq \mathbf{0} \\ 0 & \text{if} \quad j < i \quad \text{and} \quad \mathbf{v}_j = \mathbf{0} \end{cases} .$$

Then $A = Q_0 R_0$. This is called the **unnormalized QR decomposition** of the matrix A.

If we eliminate the null columns of Q_0, we get a $n \times p$ matrix $Q_1 = \begin{pmatrix} \mathbf{v}_{j_1} & \mathbf{v}_{j_2} & \ldots & \mathbf{v}_{j_p} \end{pmatrix}$ for some $j_1 < j_2, < \ldots < j_p$ in $\{1, 2, \ldots, k\}$ with $p \leq k$. If we eliminate the rows other than the rows j_1, j_2, $\ldots$, j_p from R_0, we get the $p \times k$ upper-triangular matrix

$$R_1 = \begin{pmatrix} r_{j_1,1} & r_{j_1,2} & r_{j_1,3} & \ldots & r_{j_1,k} \\ r_{j_2,1} & r_{j_2,2} & r_{j_2,3} & \ldots & r_{j_2,k} \\ r_{j_3,1} & r_{j_3,2} & r_{j_3,3} & \ldots & r_{j_3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{j_p,1} & r_{j_p,2} & r_{j_p,3} & \ldots & r_{j_p,k} \end{pmatrix} ,$$

We have $A = Q_1 R_1$. Finally, if we define the $n \times p$ matrix

$$Q = Q_1 \begin{pmatrix} \dfrac{1}{\|\mathbf{v}_{j_1}\|} & 0 & \ldots & 0 \\ 0 & \dfrac{1}{\|\mathbf{v}_{j_2}\|} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \dfrac{1}{\|\mathbf{v}_{j_p}\|} \end{pmatrix}$$

and the $p \times k$ matrix

$$R = \begin{pmatrix} \|\mathbf{v}_{j_1}\| & 0 & \ldots & 0 \\ 0 & \|\mathbf{v}_{j_2}\| & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \|\mathbf{v}_{j_p}\| \end{pmatrix} R_1 ,$$

then $A = QR$. This is the **(normalized) QR decomposition** of A. We have that $Q^\top Q = \mathrm{Id}_k$ and R is upper-triangular. The columns of Q are the normalized columns of Q_1.

The following result is an interesting consequence of the QR decomposition of a matrix.

Proposition 11.6.5

Let $S = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ be a subset of $\mathbb{R}^n$ and $V = \operatorname{span}(S)$. If $A = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_k \end{pmatrix}$ and $A = QR$ is the normalized QR decomposition of A, then $P = QQ^\top$ is the projection on V.

Proof.
The set $\left\{ \frac{1}{\|\mathbf{v}_{j_1}\|}\mathbf{v}_{j_1}, \frac{1}{\|\mathbf{v}_{j_2}\|}\mathbf{v}_{j_2}, \ldots, \frac{1}{\|\mathbf{v}_{j_p}\|}\mathbf{v}_{j_p}, \right\}$ formed of the columns of Q is an orthonormal basis of V. It follows from Proposition 11.6.4 that $P = QQ^\top$ is the projection on V. ∎

Since our goal is to find the eigenvalues of a matrix, the interesting case of QR decomposition is when A is a $n \times n$ matrix. Thus $k = n$ in the previous presentation of the QR decomposition. Moreover, we will assume that A is invertible. Thus, the set formed of the n columns of A is linearly independent. This implies that $Q_0 = Q_1$ and $R_0 = R_1$ in the previous discussion of the QR decomposition. Moreover, item 1 of Proposition 11.6.4 implies that Q is an orthogonal matrix.

We now summarize the algorithm to compute the QR decomposition of an $n \times n$ invertible matrix.

Algorithm 11.6.6 (Normalized QR decomposition)

Let $A = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n \end{pmatrix}$ and $\mathbf{q}_i$ be the i^{th} column of the matrix Q in the QR decomposition $A = QR$.

$$\begin{aligned} r_{1,1} &= \|\mathbf{u}_1\| \\ \mathbf{q}_1 &= \frac{1}{r_{1,1}} \mathbf{u}_1 \end{aligned}$$

For $i = 1,\ 2,\ \ldots n-1$

$$\begin{aligned} r_{j,i+1} &= \mathbf{u}_{i+1}^\top \mathbf{q}_j \quad \text{for} \quad j = 1, 2, \ldots, i \\ r_{i+1,i+1} &= \left\| \mathbf{u}_{i+1} - \sum_{j=1}^{i} r_{j,i+1} \mathbf{q}_j \right\| \\ \mathbf{q}_{i+1} &= \frac{1}{r_{i+1.i+1}} \left(\mathbf{u}_{i+1} - \sum_{j=1}^{i} r_{j,i+1} \mathbf{q}_j \right) \end{aligned}$$

Since $r_{i,i} > 0$ for all i (i.e. the elements on the diagonal of R are all positive), Q is uniquely determined.

Remark 11.6.7
For full non-singuliar matrix A, the algorithm above will take $O(n^3)$ multiplications to produce the QR decomposition of A. However, if A in the Hessemberg form, the algorithm will take only $O(n^2)$ multiplications to produce the QR decomposition of A. Even better,

if A is a symmetric tridiagonal matrix, the algorithm will take only $O(n)$ multiplications to produce the QR factorization of A. ♠

11.6.3 General QR Algorithm

We have presented all the techniques needed to execute the following algorithm.

Algorithm 11.6.8 (QR Algorithm)

Let A be an $n \times n$ matrix.

1. Let $A_0 = A$.
2. Given the $n \times n$ matrix A_i, find a QR decomposition $A_i = Q_i R_i$,
3. Let $A_{i+1} = R_i Q_i$. Then $A_{i+1} = Q^\top A_i Q$ and A_{i+1} is orthogonally similar to A_i.
4. Repeat (2) and (3) with i replace by $i+1$.

The matrices A_i are orthogonally similar to A.

We shall now justify why this algorithm is useful to find the eigenvalues of a matrix.

Theorem 11.6.9 (Francis)

If A is a $n \times n$ matrix with n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $|\lambda_1| < |\lambda_2| < \ldots < |\lambda_n|$, then the sequence $\{A_i\}_{i=0}^{\infty}$ converges toward an upper-triangular matrix B.

A proof of this result can be found in the article *The QR Transformation: A Unitary Analogue to the LR Transformation, Part 1*, J. G. F. Francis, The Computer Journal, Vol. **4**, Issue 3, 1961, pp. 265–271.

It follows from the previous theorem that the diagonal elements of A_i converge toward the eigenvalues of A since the A_i's are conjuagte to A.

If some of the eigenvalues of A have equal magnitude in absolute value, then the diagonal of B may contain subblocks whose eigenvalues are the eigenvalues of equal magnitude. If the subblocks are large (i.e. larger than 2×2 matrix), it may be difficult to compute these eigenvalues.

Since $|\lambda_1| < |\lambda_2| < \ldots < |\lambda_n|$ is a very restrictive condition for the eigenvalues of A, we need a less restrictive condition on the eigenvalues of A. To do that, we first consider the convergence of the sequence $\{A_i\}_{i=0}^{\infty}$.

Remark 11.6.10

If A is an Hessemberg form, then the matrices A_i produced by the QR algorithm are also in Hessemberg form. We present a "graphical" proof of this claim. It is as good as an algebraic proof without the mess of the indices. Suppose that A_i is in Hessemberg form and $A_i = Q_i R_i$, where Q_i is orthogonal and R_i is upper-triangular. Since R_i^{-1} is also upper-triangular, we have that

$$
\begin{aligned}
Q_i &= A_i R_i^{-1} \\
&= \begin{pmatrix}
* & * & * & \dots & * & * & * \\
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & * & *
\end{pmatrix}
\begin{pmatrix}
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
0 & 0 & 0 & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & 0 & *
\end{pmatrix}
= \begin{pmatrix}
* & * & * & \dots & * & * & * \\
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & * & *
\end{pmatrix}
\end{aligned}
$$

is in Hessemberg form. Hence,

$$
\begin{aligned}
A_{i+1} &= R_i Q_i \\
&= \begin{pmatrix}
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
0 & 0 & 0 & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & 0 & *
\end{pmatrix}
\begin{pmatrix}
* & * & * & \dots & * & * & * \\
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & * & *
\end{pmatrix}
= \begin{pmatrix}
* & * & * & \dots & * & * & * \\
* & * & * & \dots & * & * & * \\
0 & * & * & \dots & * & * & * \\
0 & 0 & * & \dots & * & * & * \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & 0 & * & *
\end{pmatrix}
\end{aligned}
$$

is in Hessember form.

Hence, if A is in Hessemberg form, the matrices A_i provided by the QR algorithm are also in Hessemberg form. ♠

For the sake of determining the eigenvalues of a matrix A in Hessemberg form, the convergence of the elements on the subdiagonal plays a fundamental role. If they converge rapidly toward zero, then we will rapidly get good approximations for the eigenvalues of A even if the components above the diagonal have not reached their limit yet. Suppose that all the elements on the subdiagonal of A_i are null (the situation is rarely that simple), then the diagonal of A_i has the eigenvalues of A because A_i is conjugate to A. This is true even if all the other components of A_i have not reached their limit yet. This justify the following definition.

Definition 11.6.11

Let A be a matrix in Hessemberg form. We say that the sequence $\{A_i\}_{i=0}^{\infty}$ produce by the QR algorithm **converges** if

$$
\max_{\substack{2\leq j\leq n-1 \\ M=A_i}} |m_{j+1,j}\, m_{j,j-1}| \to 0 \quad \text{as} \quad i \to \infty \ .
$$

The following theorem demonstrates the importance of the elements on the subdiagonal of the matrices A_i.

Theorem 11.6.12 (Parlett)

Let A be a $n \times n$ matrix in Hessemberg form. The sequence of matrices $\{A_i\}_{i=0}^{\infty}$ produced with the QR algorithm converges as defined in the previous definition if and only if each set of eigenvalues of A of the same magnitude in absolute value contains at most two eigenvalues of even algebraic multiplicity or two eigenvalues of odd algebraic multiplicity.

A proof of this theorem is given in the article *Global Convergence of the Basic QR Algorithm On Hessemberg Matrices*, B. Parlett, Mathematics of Computation, Vol. **22**, No. 104 (Oct. 1968), pp. 803-817.

The limit of the sequence $\{A_i\}_{i=0}^{\infty}$ predicted by the previous theorem will be a matrix having sub-blocks of dimension at most 2×2 on the diagonal. The eigenvalues of these sub-blocks are the eigenvalues of A.

The convergence of the sequence $\{A_i\}_{i=0}^{\infty}$ if A is in Hessemberg matrices A is not fast. Moreover, the convergence of the sequence $\{A_i\}_{i=0}^{\infty}$ is not faster for a symmetric tridiagonal matrices A but the QR factorization of the A_i's is fast (of the order of $O(n)$ multiplications as we have seen in Remark 11.6.7). In the next section, we present an efficient algorithm to find an orthogonal matrix Q_i and an upper-triangular matrix R_i for the factorization $A_i = Q_i R_i$ in the case where A_i is a symmetric tridiagonal matrix.

Obviously, not all matrices are in Hessemberg form. However, we have seen that for any given matrix, we can use Householder matrices to find a Hessemberg matrix conjugate to it. Then the QR algorithm can be applied to this Hessemberg matrix. We summarize this algorithm in the next theorem. In this statement, we also use Householder matrices to find the QR decomposition instead of Gram-Schmidt as we have done before. We will not elaborate on this approach in these notes. It is an interesting theoretical approach but not computationally efficient.

Algorithm 11.6.13 (The QR Algorithm)

Let A be a $n \times n$ matrix with entries in $\mathbb{R}$.

1. Let $G_1, G_2, \ldots, G_{n-2}$ be $n-2$ Householder matrices (given by Algorithm 11.5.5) such that $A_1 = G^\top A G$ is a matrix in Hessemberg form for $G = G_1 G_2 \cdots G_{n-2}$.
2. Given the $n \times n$ matrix A_i in Hessemberg form, let $Q_1, Q_2, \ldots, Q_n$ be n Householder matrices (given by Algorithm 11.5.3) such that $A_n = QR$ for $Q = Q_1 Q_2 \cdots Q_n$ and R an upper-triangular $n \times n$ matrix.
3. Let $A_{i+1} = RQ$. Then $A_{i+1} = Q^\top A_i Q$ and A_{i+1} is orthogonally similar to A_i.
4. Repeat (2) and (3) with i replace by $i + 1$.

The matrices A_i are orthogonally similar to A.

11.6.4 QR Factorization for Symmetric Tridiagonal Matrices

Let

$$P_1(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & \mathrm{Id}_{n-2} \end{pmatrix} , \quad P_{n-1}(\theta) = \begin{pmatrix} \mathrm{Id}_{n-2} & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} ,$$

and

$$P_j(\theta) = \begin{pmatrix} \mathrm{Id}_{j-1} & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & \mathrm{Id}_{n-j-1} \end{pmatrix}$$

for $j = 2, 3 ,\ldots, n-2$. Suppose that A is a symmetric tridiagonal matrix. Let $A_0 = A$. We explain how to use the matrices $P_j(\theta)$ to find a QR decomposition $A_i = Q_i R_i$ for $i \geq 0$. Recall that $A_{i+1} = Q_i R_i$.

Suppose that

$$A_i = \begin{pmatrix} a_1 & b_1 & 0 & \ldots & 0 & 0 & 0 \\ b_1 & a_2 & b_2 & \ldots & 0 & 0 & 0 \\ 0 & b_2 & a_3 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & \ldots & 0 & b_{n-1} & a_n \end{pmatrix} .$$

To compute Q_i and R_i in $A_i = Q_i R_i$, choose θ_1 such that

$$B_1 = P_1(\theta_1) A_i = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 & \ldots & 0 & 0 \\ 0 & x_2 & y_2 & 0 & \ldots & 0 & 0 \\ 0 & b_2 & a_3 & b_3 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & b_{n-1} & a_n \end{pmatrix} ;$$

namely,

$$\begin{aligned} \alpha_1 &= a_1 \cos(\theta_1) + b_1 \sin(\theta_1) \ , \\ \beta_1 &= b_1 \cos(\theta_1) + a_2 \sin(\theta_1) \ , \\ \gamma_1 &= b_2 \sin(\theta_1) \ , \\ x_2 &= -b_1 \sin(\theta_1) + a_2 \cos(\theta_1) \ , \\ y_2 &= b_2 \cos(\theta_1) \end{aligned} \tag{11.6.2}$$

and

$$0 = -a_1 \sin(\theta_1) + b_1 \cos(\theta_1) \ .$$

We have that $\cos^2(\theta_1) + \sin^2(\theta_1) = 1$ and $0 = -a_1 \sin(\theta_1) + b_1 \cos(\theta_1)$ are satisfied by

$$\cos(\theta_1) = \frac{a_1}{\sqrt{a_1^2 + b_1^2}} \quad \text{and} \quad \sin(\theta_1) = \frac{b_1}{\sqrt{a_1^2 + b_1^2}} \ . \tag{11.6.3}$$

This is a possible choice.

Suppose that we have found θ_j and B_j for $j = 1, 2, \ldots, k$ with $k < n-2$ such that

$$B_k = P_k(\theta_k)P_{k-1}(\theta_{k-1})\ldots P_1(\theta_1)A_i$$
$$= \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \alpha_k & \beta_k & \gamma_k & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & x_{k+1} & y_{k+1} & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & b_{k+1} & a_{k+2} & b_{k+2} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & b_{n-1} & a_n \end{pmatrix} .$$

Choose θ_{k+1} such that

$$B_{k+1} = P_{k+1}(\theta_{k+1})B_k$$
$$= \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \alpha_{k+1} & \beta_{k+1} & \gamma_{k+1} & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & x_{k+2} & y_{k+2} & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & b_{k+2} & a_{k+3} & b_{k+3} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & b_{n-1} & a_n \end{pmatrix} ;$$

namely,

$$\begin{aligned} \alpha_{k+1} &= x_{k+1}\cos(\theta_{k+1}) + b_{k+1}\sin(\theta_{k+1}) \ , \\ \beta_{k+1} &= y_{k+1}\cos(\theta_{k+1}) + a_{k+2}\sin(\theta_{k+1}) \ , \\ \gamma_{k+1} &= b_{k+2}\sin(\theta_{k+1}) \ , \\ x_{k+2} &= -y_{k+1}\sin(\theta_{k+1}) + a_{k+2}\cos(\theta_{k+1}) \ , \\ y_{k+2} &= b_{k+2}\cos(\theta_{k+1}) \end{aligned} \tag{11.6.4}$$

and

$$0 = -x_{k+1}\sin(\theta_{k+1}) + b_{k+1}\cos(\theta_{k+1}) \ .$$

We have that $\cos^2(\theta_{k+1}) + \sin^2(\theta_{k+1}) = 1$ and $0 = -x_{k+1}\sin(\theta_{k+1}) + b_{k+1}\cos(\theta_{k+1})$ are satisfied by

$$\cos(\theta_{k+1}) = \frac{x_{k+1}}{\sqrt{x_{k+1}^2 + b_{k+1}^2}} \quad \text{and} \quad \sin(\theta_{k+1}) = \frac{b_{k+1}}{\sqrt{x_{k+1}^2 + b_{k+1}^2}} \ . \tag{11.6.5}$$

Proceeding inductively, we can find θ_j and B_j for $j = 1, 2, \ldots, n-2$ such that

$$B_{n-2} = P_{n-2}(\theta_{n-2})P_{n-1}(\theta_{n-1})\ldots P_1(\theta_1)A_i$$

$$= \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \alpha_{n-2} & \beta_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & 0 & \ldots & 0 & x_{n-1} & y_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & b_{n-1} & a_n \end{pmatrix} .$$

Choose θ_{n-1} such that

$$\begin{aligned} B_{n-1} &= P_{n-1}(\theta_{n-1}) B_{n-2} \\ &= \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \alpha_{n-2} & \beta_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & 0 & \ldots & 0 & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \alpha_n \end{pmatrix} ; \end{aligned}$$

namely,

$$\begin{aligned} \alpha_{n-1} &= x_{n-1}\cos(\theta_{n-1}) + b_{n-1}\sin(\theta_{n-1}) \ , \\ \beta_{n-1} &= y_{n-1}\cos(\theta_{n-1}) + a_n\sin(\theta_{n-1}) \ , \\ \alpha_n &= -y_{n-1}\sin(\theta_{n-1}) + a_n\cos(\theta_{n-1}) \end{aligned} \tag{11.6.6}$$

and

$$0 = -x_{n-1}\sin(\theta_{n-1}) + b_{n-1}\cos(\theta_{n-1}) \ .$$

We have that $\cos^2(\theta_{n-1}) + \sin^2(\theta_{n-1}) = 1$ and $0 = -x_{n-1}\sin(\theta_{n-1}) + b_{n-1}\cos(\theta_{n-1})$ are satisfied by

$$\cos(\theta_{n-1}) = \frac{x_{n-1}}{\sqrt{x_{n-1}^2 + b_{n-1}^2}} \quad \text{and} \quad \sin(\theta_{n-1}) = \frac{b_{n-1}}{\sqrt{x_{n-1}^2 + b_{n-1}^2}} \ . \tag{11.6.7}$$

We end up with the upper-triangular matrix

$$B_{n-1} = P_{n-1}(\theta_{n-1}) P_{n-2}(\theta_{n-2}) \ldots P_1(\theta_1) A_i \ .$$

Let

$$R_i = B_{n-1}$$

and

$$\begin{aligned} Q_i &= \left(P_{n-1}(\theta_{n-1}) P_{n-2}(\theta_{n-2}) \ldots P_1(\theta_1)\right)^\top = P_1(\theta_1)^\top P_2(\theta_2)^\top \ldots P_{n-1}(\theta_{n-1})^\top \\ &= P_1(-\theta_1) P_2(-\theta_2) \ldots P_{n-1}(-\theta_{n-1}) \ . \end{aligned}$$

We have $A_i = Q_i R_i$, where R_i is an upper-triangular matrix and Q_i is an orthogonal matrix.

To complete the justification of the QR algorithm above for the symmetric tridiagonal matrices, we now show that A_i is symmetric tridiagonal for all i. Since A is symmetric, it follows by induction from (11.6.1) that A_i is symmetric (i.e. $A_i^\top = A_i$) for all i. Moreover, since $A_0 = A$ is in Hessemberg form, it follows from Remark 11.6.10 that A_i is in Hessemberg form for all i. Thus, the matrix A_i is symmetric tridiagonal for all i.

11.6.5 Shifting Technique

We have mentioned before that the convergence of the sequence $\{A_i\}_{i=0}^{\infty}$ provided by the QR algorithm is not fast, even if $A_0 = A$ is symmetric tridiagonal. To accelerate the convergence of the sequences, we present a technique similar to the shifting technique used for the inverse power method.

Suppose that we have computed

$$A_i = \begin{pmatrix} a_{i;1} & b_{i;1} & 0 & \dots & 0 & 0 & 0 \\ b_{i;1} & a_{i;2} & b_{i;2} & \dots & 0 & 0 & 0 \\ 0 & b_{i;2} & a_{i;3} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b_{i;n-2} & a_{i;n-1} & b_{i;n-1} \\ 0 & 0 & 0 & \dots & 0 & b_{i;n-1} & a_{i;n} \end{pmatrix} .$$

To compute A_{i+1}, we consider the matrix $A_i - s_i \,\mathrm{Id}$, where s_i is the eigenvalue of

$$\begin{pmatrix} a_{i;n-1} & b_{i;n-1} \\ b_{i;n-1} & a_{i;n} \end{pmatrix}$$

which is closest to $a_{i;n}$. Since $A_i - s_i \,\mathrm{Id}$ is symmetric tridiagonal, we may use the QR factorization method of the previous section to write $A_i - s_i \,\mathrm{Id} = Q_i R_i$, where Q_i is an orthogonal matrix and R_i is an upper-triangular matrix. The matrix A_{i+1} is defined by $A_{i+1} = R_i Q_i$ as usual.

We first prove by induction that

$$A_{i+1} = Q_i^\top Q_{i-1}^\top \dots Q_0^\top A_0 Q_0 Q_1 \dots Q_i - \sum_{j=0}^{i} s_j \,\mathrm{Id} \ . \tag{11.6.8}$$

Since $A_0 - s_0 \,\mathrm{Id} = Q_0 R_0$ and $A_1 = R_0 Q_0$, we get

$$A_1 = Q_0^\top \left(A_0 - s_0 \,\mathrm{Id}\right) Q_0 = Q_0^\top A_0 Q_0 - s_0 \,\mathrm{Id} \ .$$

This proves (11.6.8) for $i = 0$. Suppose that (11.6.8) is true for $i = k$; namely,

$$A_{k+1} = Q_k^\top Q_{k-1}^\top \dots Q_0^\top A_0 Q_0 Q_1 \dots Q_k - \sum_{j=0}^{k} s_j \,\mathrm{Id} \ .$$

Then $A_{k+1} - s_{k+1} \,\mathrm{Id} = Q_{k+1} R_{k+1}$ and $A_{k+2} = R_{k+1} Q_{k+1}$ yield

$$\begin{aligned} A_{k+2} &= Q_{k+1}^\top \left(A_{k+1} - s_{k+1} \,\mathrm{Id}\right) Q_{k+1} \\ &= Q_{k+1}^\top \left(Q_k^\top Q_{k-1}^\top \dots Q_0^\top A_0 Q_0 Q_1 \dots Q_k - \sum_{j=0}^{k} s_j \,\mathrm{Id} - s_{k+1} \,\mathrm{Id} \right) Q_{k+1} \\ &= Q_{k+1}^\top Q_k^\top \dots Q_0^\top A_0 Q_0 Q_1 \dots Q_{k+1} - \left(\sum_{j=0}^{k+1} s_j \right) Q_{k+1}^\top \,\mathrm{Id}\, Q_{k+1} \end{aligned}$$

$$= Q_{k+1}^\top Q_k^\top \ldots Q_0^\top A_0 Q_0 Q_1 \ldots Q_{k+1} - \sum_{j=0}^{k+1} s_j \,\mathrm{Id} \ ,$$

where we have used the hypothesis of induction for the second equality. This proves that (11.6.8) is true for $i = k+1$. By induction, (11.6.8) is true for all i.

Hence, the eigenvalues of A are of the form $\lambda + \sum_{j=0}^{i} s_j$, where λ is an eigenvalue of A_{i+1}. If $b_{i+1;n-1}$ is negligible (to be defined by the user), we may assume that $b_{i+1,n-1} = 0$ and thus $a_{i+1;n} + \sum_{j=0}^{i} s_j$ is an eigenvalue of A.

To find the other eigenvalues of A, we consider the $(n-1) \times (n-1)$ matrix

$$C = \begin{pmatrix} a_{i+1;1} & b_{i+1;1} & 0 & \ldots & 0 & 0 & 0 \\ b_{i+1;1} & a_{i+1;2} & b_{i+1;2} & \ldots & 0 & 0 & 0 \\ 0 & b_{i+1;2} & a_{i+1;3} & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & b_{i+1;n-3} & a_{i+1;n-2} & b_{i+1;n-2} \\ 0 & 0 & 0 & \ldots & 0 & b_{i+1;n-2} & a_{i+1;n-1} \end{pmatrix} .$$

The eigenvalues of A_{i+1} other than (one copy of) $a_{i+1;n} + \sum_{j=0}^{i} s_j$ "are" the eigenvalues of C. We used quotation marks in the previous sentence because it is rigorously true only if $b_{i+1,n-1} = 0$, not just when $b_{i+1,n-1}$ is negligible. We repeat the previous QR algorithm with shifting with A_0 replaced by C to find an eigenvalue λ of C. We have that $\lambda + \sum_{j=0}^{i} s_j$ is an eigenvalue of A.

In general, we can repeat recursively this procedure to approximate all eigenvalues of A. The QR algorithm with shifting suffers from some of the weaknesses that the standard QR algorithm has.

The following code implement the QR algorithm with shifting. The equations (11.6.2) to (11.6.7) inclusively have been used to create this algorithm.

Code 11.6.14 (QR Algorithm with Shifting)

To find the eigenvalues of a symmetric tridiagonal matrix A.
Input: The components $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_{n-1}$ of the symmetric tridiagonal matrix A.
The maximum number N of iterations for the QR decomposition.
The value d such that numbers b satisfying $|b| < d$ may be considered as null.
Output: An approximation for each eigenvalue of A if it is possible.

```
function E = QRshifting(a, b, N, d)
  n = length(a);
  E = repmat(NaN,n,1);
  nE = 0;
  s = 0;           % sum of the shifts
```

```
for k = 1:N
  fprintf('%d ... ',k);

  % If n=1, we are done
  if ( n == 1 )
    nE = nE + 1;
    E(nE) = a(1) + s;
    return;
  end

  % If the matrix can be splitted into two symmetric tridiagonal
  % matrices, we do so.
  for j = 1:n-1
    if ( abs(b(j)) < d )
      disp 'Splitting the matrix';
      E(nE+1:nE+j,1) = QRshifting(a(1:j),b(1:j-1), N, d) + s;
      E(nE+j+1:n,1) = QRshifting(a(j+1:n),b(j+1:n-1), N, d) + s;
      return;
    end
  end

  % We compute the eigenvalues of the matrix
  %    [ a_{n-1} b_{n-1} ]
  %    [ b_{n-1} a_n     ]
  % We use the appropriate form of the formula to find the roots
  % of a quadratic equation to avoid subtraction of almost equal
  % numbers.
  B = -(a(n-1) + a(n));
  C = a(n)*a(n-1) - b(n-1)*b(n-1);
  D = sqrt(B^2-4*C);
  if ( B > 0 )
    r1 = -2*C/(B+D);
    r2 = -(B+D)/2;
  else
    r1 = (D-B)/2;
    r2 = 2*C/(D-B);
  end

  % If we have only a 2 x 2 matrix, we have found approximations
  % for the last two eigenvalues of A.
  if ( n == 2)
    nE = nE + 1;
    E(nE,1) = r1 + s;
    nE = nE + 1;
    E(nE,1) = r2 + s;
    return;
```

```
    end

    % Chose the appropriate shift
    if ( abs(r1-a(n)) < abs(r2 -a(n)) )
      stmp = r1;
    else
      stmp = r2;
    end
    s = s + stmp;
    a = a - stmp;

    % Get the QR decomposition
    x = a(1);
    y = b(1);
    for j = 1:n-1
      alpha(j) = sqrt( x^2 +b(j)^2 );
      ccc(j) = x/alpha(j);
      sss(j) = b(j)/alpha(j);
      beta(j) = y*ccc(j) + a(j+1)*sss(j);
      x = - y*sss(j) + a(j+1)*ccc(j);
      if ( j ~= n-1 )
        gamma(j) = b(j+1)*sss(j);
        y = b(j+1)*ccc(j);
      end
    end
    alpha(n) = x;

    % Compute RQ knowing that the result is a symmetric tridiagonal matrix
    a(1) = alpha(1)*ccc(1) + beta(1)*sss(1);
    b(1) = alpha(2)*sss(1);
    for j = 2:n-1;
      a(j) = alpha(j)*ccc(j-1)*ccc(j) + beta(j)*sss(j);
      b(j) = alpha(j+1)*sss(j);
    end
    a(n) = alpha(n)*ccc(n-1);
  end
end
```

Chapter 12

Numerical Differentiation and Integration

Readers have probably learned many techniques to compute derivatives and integrals in their calculus courses. They probably remember how tricky and convoluted the computations can be when the function is a little bit complex. They probably have also seen some examples of integrals that cannot be evaluated using any of the integration methods. Powerful programs doing symbolic computations can, to some extend, compute all the derivatives and integrals but their answers are sometime very complicated formulae that still have to be evaluated numerically. It is often less costly (in computer time) and more accurate to simply use numerical methods to compute the derivative of a function at a point or the integral of a function on an interval. This chapter introduces some of the most often used numerical methods to compute derivative and evaluate integrals.

12.1 Numerical Differentiation

Let $f :]a,b[\to \mathbb{R}$ be a sufficiently continuously differentiable function and let p be the interpolating polynomial of f at some well chosen nodes x_0. x_1, ..., x_n in $]a,b[$. To develop formulae to approximate $f'(x)$ at $x \in]a,b[$, we use the interpolating polynomial p.

Theorem 12.1.1

Let f be a three times continuously differentiable function near $a \in \mathbb{R}$. Then

$$f'(a) = \frac{f(a+h) - f(a)}{h} - \frac{1}{2} f''(\eta)\, h \tag{12.1.1}$$

for some η between a and $a+h$.

Remark 12.1.2

1. (12.1.1) is called a **forward difference formula** if $h > 0$ and a **backward difference formula** if $h < 0$.

2. If h is small, we have $f'(a) \approx (f(a+h) - f(a))/h$. The term $-f''(\eta)\,h/2$, which has been dropped, is the **truncation error**.

♠

Proof.
If we use two nodes x_0 and x_1, we have

$$f(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x](x - x_0)(x - x_1) \ .$$

Hence,

$$\begin{aligned} f'(x) &= f[x_0, x_1] + f[x_0, x_1, x]\left((x - x_0) + (x - x_1)\right) + \left(\frac{\mathrm{d}}{\mathrm{d}x} f[x_0, x_1, x]\right)(x - x_0)(x - x_1) \\ &= f[x_0, x_1] + f[x_0, x_1, x]\left((x - x_0) + (x - x_1)\right) + f[x_0, x_1, x, x]\,(x - x_0)(x - x_1) \\ &= f[x_0, x_1] + \frac{1}{2} f''(\eta)\big((x - x_0) + (x - x_1)\big) + \frac{1}{3!} f'''(\xi)(x - x_0)(x - x_1) \qquad (12.1.2) \end{aligned}$$

for some η and ξ in the smallest interval containing x_0, x_1 and x. If we choose $x = x_0 = a$ and $x_1 = a + h$ with $h \in \mathbb{R}$, (12.1.2) becomes

$$f'(a) = f[a, a + h] - \frac{1}{2} f''(\eta)\, h$$

for some η between a and $a + h$. ∎

Theorem 12.1.3 (Central Difference Formula)

Let f be a four times continuously differentiable function near $a \in \mathbb{R}$. Then

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} - \frac{1}{6} f^{(3)}(\eta)\, h^2 \qquad (12.1.3)$$

for some η between $a - h$ and $a + h$.

Remark 12.1.4
If h is small, $f'(a) \approx (f(a+h) - f(a-h))/(2h)$ and the truncation error is the term $-\frac{1}{6} f^{(3)}(\eta)\, h^2$.

♠

Proof.
If we use three nodes x_0, x_1 and x_2, we have

$$\begin{aligned} f(x) = f(x_0) &+ f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ f[x_0, x_1, x_2, x](x - x_0)(x - x_1)(x - x_2) \ . \end{aligned}$$

Hence,

$$f'(x) = f[x_0, x_1] + f[x_0, x_1, x_2]\big((x - x_0) + (x - x_1)\big)$$

$$
\begin{aligned}
&\quad + f[x_0,x_1,x_2,x]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\
&\quad + \left(\frac{\mathrm{d}}{\mathrm{d}x} f[x_0,x_1,x_2,x]\right)(x-x_0)(x-x_1)(x-x_2) \\
&= f[x_0,x_1] + f[x_0,x_1,x_2]\big((x-x_0)+(x-x_1)\big) \\
&\quad + f[x_0,x_1,x_2,x]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\
&\quad + f[x_0,x_1,x_2,x,x](x-x_0)(x-x_1)(x-x_2) \\
&= f[x_0,x_1] + f[x_0,x_1,x_2]\big((x-x_0)+(x-x_1)\big) \\
&\quad + \frac{1}{3!} f^{(3)}(\eta)\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\
&\quad + \frac{1}{4!} f^{(4)}(\xi)(x-x_0)(x-x_1)(x-x_2) \qquad (12.1.4)
\end{aligned}
$$

for some η and ξ in the smallest interval containing x_0, x_1, x_2 and x. If we choose $x = x_1 = a$, $x_0 = a-h$ and $x_2 = a+h$ with $h \in \mathbb{R}$, (12.1.4) becomes

$$
\begin{aligned}
f'(a) &= f[a-h,a] + f[a-h,a,a+h]\,h - \frac{1}{3!} f^{(3)}(\eta)\,h^2 \\
&= \frac{f(a+h)-f(a-h)}{2h} - \frac{1}{3!} f^{(3)}(\eta)\,h^2
\end{aligned}
$$

for some η between $a-h$ and $a+h$. ∎

Remark 12.1.5
Due to rounding error, numerical differentiation is unstable. The truncation error decreases as h decreases but rounding error increases as h decreases.

To illustrate this phenomenon, we consider the central difference formula (12.1.3). Let f_{a-h} be the computed value of $f(a-h)$ and f_{a+h} be the computed value of $f(a+h)$. The rounding errors in computing $f(a-h)$ and $f(a+h)$ are respectively $E_- = f_{a-h} - f(a-h)$ and $E_+ = f_{a+h} - f(a+h)$.

From (12.1.3), we have

$$
f'(a) = \frac{(f_{a+h}-E_+) - (f_{a-h}-E_-)}{2h} - \frac{1}{6} f^{(3)}(\eta)\,h^2 \ .
$$

for η between $a-h$ and $a+h$. The computed value used to approximate $f'(a)$ is

$$
\frac{f_{a+h}-f_{a-h}}{2h} \qquad (12.1.5)
$$

and the error is

$$
R(h) = \frac{E_- - E_+}{2h} - \frac{1}{6} f^{(3)}(\eta)\,h^2 \ . \qquad (12.1.6)
$$

We have assumed that the subtraction and division in (12.1.5) can be performed without rounding error to simplify the discussion.

We may assume that E_- and E_+ have the same (small) magnitude. Moreover, if we assume that $f^{(3)}$ is almost constant near a, then we see from (12.1.6) that $|R(h)|$ increases as h decreases. For instance, if $f(x) = \ln(x)$ and $a = 2$, we have

$$f'(2) \approx \frac{\ln(2+h) - \ln(2-h)}{2h} \tag{12.1.7}$$

with error

$$R(h) = \frac{E_- - E_+}{2h} + \frac{h^2}{3\eta^3}$$

for some η between $2-h$ and $2+h$. If we assume that $|E_- - E_+| \approx 10^{-8}$ and $\eta \approx 2$, then

$$|R(h)| \approx \frac{10^{-8}}{2h} + \frac{h^2}{3 \times 2^3} = \frac{10^{-8}}{2h} + \frac{h^2}{24} .$$

The graph of $|R(h)|$ is given in Figure 12.1. $|R(h)|$ effectively increases as h decreases. Moreover, $|R(h)|$ is minimal at $h \approx 0.003915$. For this value, (12.1.7) gives $f'(2) \approx 0.50000064$ which is a good approximation of $f'(2) = 0.5$. ♠

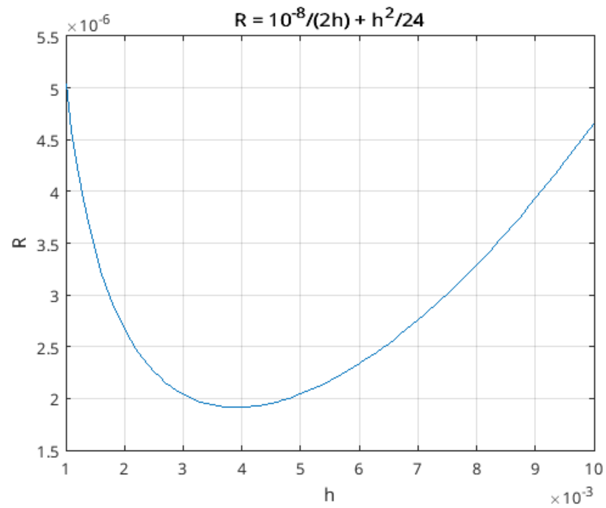


Figure 12.1: Graph of $|R(h)|$ where $R(y) = 10^{-8}/(2h) + h^2/24$.

12.2 Richardson Extrapolation

Richardson extrapolation is also called **extrapolation to the limit**.

Let $f : \mathbb{R} \to \mathbb{R}$ be a sufficiently continuously differentiable function and $L(f) = f'(c)$ for some $c \in \mathbb{R}$ (or $L(f) = \displaystyle\int_a^b f(x)\,\mathrm{d}x$ as we will see later). Suppose that $L_h(f)$ is an approximation of $L(f)$ satisfying

$$L(f) = L_h(f) + \sum_{j=1}^{\infty} K_j h^{2j} \tag{12.2.1}$$

for h near the origin. We now describe a procedure that generates better approximations of $L(f)$ than $L_h(f)$ from a truncation point of view.

If we replace h in (12.2.1) by $h/2$ and $h/2^2$, we respectively get

$$L(f) = L_{h/2}(f) + \sum_{j=1}^{\infty} K_j \left(\frac{h}{2}\right)^{2j} . \tag{12.2.2}$$

and

$$L(f) = L_{h/2^2}(f) + \sum_{j=1}^{\infty} K_j \left(\frac{h}{2^2}\right)^{2j} . \tag{12.2.3}$$

If we subtract (12.2.1) from 4 times (12.2.2) and divide the result by $4-1$, we get

$$\begin{aligned} L(f) &= L^1_{h/2}(f) + \frac{1}{4-1}\left(4\sum_{j=1}^{\infty} K_j \left(\frac{h}{2}\right)^{2j} - \sum_{j=1}^{\infty} K_j\, h^{2j}\right) \\ &= L^1_{h/2}(f) + \frac{1}{4-1}\left(\sum_{j=1}^{\infty}\left(4K_j\left(\frac{h}{2}\right)^{2j} - 4^j K_j \left(\frac{h}{2}\right)^{2j}\right)\right) \\ &= L^1_{h/2}(f) + \sum_{j=2}^{\infty} \frac{4-4^j}{4-1} K_j \left(\frac{h}{2}\right)^{2j} , \end{aligned} \tag{12.2.4}$$

where

$$L^1_{h/2}(f) = \frac{4L_{h/2}(f) - L_h(f)}{4-1}$$

is an approximation of $L(f)$ with truncation error $-K_2\, h^4/4 + O(h^6)$. Recall that the expression $O(h^k)$ replaces a function $g(h)$ for which there exists a constant M such that $|g(h)| \le M h^k$ for h closed to the origin. In theory, $L^1_{h/2}(f)$ is a better approximation of $L(f)$ than $L_{h/2}(f)$ because, for $h < 1$ given, the truncation error for $L^1_{h/2}(f)$ in (12.2.4) is generally smaller than the truncation error for $L_{h/2}(f)$ in (12.2.2).

If we subtract (12.2.2) from 4 times (12.2.3) and divide the result by $4-1$, we get

$$\begin{aligned} L(f) &= L^1_{h/2^2}(f) + \frac{1}{4-1}\left(4\sum_{j=1}^{\infty} K_j \left(\frac{h}{2^2}\right)^{2j} - \sum_{j=1}^{\infty} K_j \left(\frac{h}{2}\right)^{2j}\right) \\ &= L^1_{h/2^2}(f) + \frac{1}{4-1}\left(\sum_{j=1}^{\infty}\left(4K_j\left(\frac{h}{2^2}\right)^{2j} - 4^j K_j \left(\frac{h}{2^2}\right)^{2j}\right)\right) \\ &= L^1_{h/2^2}(f) + \sum_{j=2}^{\infty} \frac{4-4^j}{4-1} K_j \left(\frac{h}{2^2}\right)^{2j} , \end{aligned} \tag{12.2.5}$$

where

$$L^1_{h/2^2}(f) = \frac{4L_{h/2^2}(f) - L_{h/2}(f)}{4-1}$$

is an approximation of $L(f)$ with truncation error $-K_2\, h^4/4^3 + O(h^6)$.

In the spirit of the discussion above, we can easily prove by induction that

$$L(f) = L^1_{h/2^k}(f) + \sum_{j=2}^{\infty} \frac{4-4^j}{4-1} K_j \left(\frac{h}{2^k}\right)^{2j} \quad , \tag{12.2.6}$$

where

$$L^1_{h/2^k}(f) = \frac{4L_{h/2^k}(f) - L_{h/2^{k-1}}(f)}{4-1}$$

is an approximation of $L(f)$ with truncation error $O(h^4)$.

If we subtract (12.2.4) from 4^2 times (12.2.5) and divide the result by 4^2-1, we get

$$\begin{aligned}
L(f) &= L^2_{h/2^2}(f) + \frac{1}{4^2-1}\left(4^2 \sum_{j=2}^{\infty} \frac{4-4^j}{4-1} K_j \left(\frac{h}{2^2}\right)^{2j} - \sum_{j=2}^{\infty} \frac{4-4^j}{4-1} K_j \left(\frac{h}{2}\right)^{2j}\right) \\
&= L^2_{h/2^2}(f) + \frac{1}{4^2-1} \sum_{j=2}^{\infty}\left(4^2 \frac{4-4^j}{4-1} K_j \left(\frac{h}{2^2}\right)^{2j} - 4^j \frac{4-4^j}{4-1} K_j \left(\frac{h}{2^2}\right)^{2j}\right) \\
&= L^2_{h/2^2}(f) + \sum_{j=3}^{\infty} \frac{(4^2-4^j)(4-4^j)}{(4^2-1)(4-1)} K_j \left(\frac{h}{2^2}\right)^{2j} \quad ,
\end{aligned} \tag{12.2.7}$$

where

$$L^2_{h/2^2}(f) = \frac{4^2 L^1_{h/2^2}(f) - L^1_{h/2}(f)}{4^2-1}$$

is an approximation of $L(f)$ with truncation error $K_3 h^6/64 + O(h^8)$. We may assume that $L^2_{h/2^2}(f)$ is the best approximation of $L_h(f)$ that we have found so far in this section. For small $h < 1$, the truncation error is generally smaller for $L^2_{h/2^2}(f)$ than for the other approximations.

In general, we generate the following table:

Order of the truncation error

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
$L^0_h(f)$			
$L^0_{h/2}(f)$	$L^1_{h/2}(f) = \frac{4L^0_{h/2}(f)-L^0_h(f)}{4-1}$		
$L^0_{h/4}(f)$	$L^1_{h/4}(f) = \frac{4L^0_{h/4}(f)-L^0_{h/2}(f)}{4-1}$	$L^2_{h/4}(f) = \frac{4^2L^1_{h/4}(f)-L^1_{h/2}(f)}{4^2-1}$	
$L^0_{h/8}(f)$	$L^1_{h/8}(f) = \frac{4L^0_{h/8}(f)-L^0_{h/4}(f)}{4-1}$	$L^2_{h/8}(f) = \frac{4^2L^1_{h/8}(f)-L^1_{h/4}(f)}{4^2-1}$	$L^3_{h/8}(f) = \frac{4^3L^2_{h/8}(f)-L^2_{h/4}(f)}{4^3-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

where $L^0_{h/2^k}(f) = L_{h/2^k}(f)$. The general formula is

$$L^n_{h/2^k}(f) = \frac{4^n L^{n-1}_{h/2^k}(f) - L^{n-1}_{h/2^{k-1}}(f)}{4^n-1} \tag{12.2.8}$$

for $k \geq n > 0$.

Proposition 12.2.1

Given any non-negative integer n, we have that

$$L(f) = L^n_{h/2^k}(f) + \sum_{j=n+1}^{\infty} \hat{K}_{j,n}\left(\frac{h}{2^k}\right)^{2j} , \tag{12.2.9}$$

where

$$\hat{K}_{j,n} = \begin{cases} K_j & \text{if} \quad n = 0 \\ \dfrac{(4^n - 4^j)(4^{n-1} - 4^j)\ldots(4 - 4^j)}{(4^n - 1)(4^{n-1} - 1)\ldots(4 - 1)} K_j & \text{if} \quad n > 0 \end{cases}$$

for $j \geq n$. The K_n are defined in (12.2.1). In particular, $L(f) = L^n_{h/2^k} + O(h^{2n+2})$.

Proof.
The proof is by induction on n. From (12.2.1), we have that

$$L(f) = L^0_h(f) + \sum_{j=1}^{\infty} K_j\, h^{2j}$$

for all h. Replacing h by $h/2^k$ with $k \geq 0$ gives (12.2.9) for $n = 0$. The case $n = 1$ is (12.2.6).

We assume that (12.2.9) is true for n replaced by $n - 1$; namely,

$$L(f) = L^{n-1}_{h/2^k}(f) + \sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\frac{h}{2^k}\right)^{2j}$$

with

$$\hat{K}_{j,n-1} = \frac{(4^{n-1} - 4^j)(4^{n-2} - 4^j)\ldots(4 - 4^j)}{(4^{n-1} - 1)(4^{n-2} - 1)\ldots(4 - 1)} K_j \,.$$

Then (12.2.8) yields

$$\begin{aligned} L^n_{h/2^k}(f) &= \frac{4^n L^{n-1}_{h/2^k}(f) - L^{n-1}_{h/2^{k-1}}(f)}{4^n - 1} \\ &= \frac{1}{4^n - 1}\left(4^n\left(L(f) - \sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\frac{h}{2^k}\right)^{2j}\right) - \left(L(f) - \sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\frac{h}{2^{k-1}}\right)^{2j}\right)\right) \\ &= \frac{1}{4^n - 1}\left((4^n - 1)L(f) - \sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(4^n - 2^{2j}\right)\left(\frac{h}{2^k}\right)^{2j}\right) \\ &= L(f) - \sum_{j=n}^{\infty} \frac{4^n - 4^j}{4^n - 1}\hat{K}_{j,n-1}\left(\frac{h}{2^k}\right)^{2j} = L(f) - \sum_{j=n+1}^{\infty} \hat{K}_{j,n}\left(\frac{h}{2^k}\right)^{2j} \end{aligned}$$

which is (12.2.9). This complete the proof by induction. ∎

Remark 12.2.2

1. Before using $L^n_{h/2^k}$ as a good approximation of $L(f)$, we should verify that
$$\frac{L^{n-1}_{h/2^k}(f) - L^{n-1}_{h/2^{k-1}}(f)}{L^{n-1}_{h/2^{k+1}}(f) - L^{n-1}_{h/2^k}(f)} \approx 4^n \ . \tag{12.2.10}$$
This rule is motivated by the following observation. From the previous proposition,
$$\begin{aligned}\frac{L^{n-1}_{h/2^k}(f) - L^{n-1}_{h/2^{k-1}}(f)}{L^{n-1}_{h/2^{k+1}}(f) - L^{n-1}_{h/2^k}(f)} &= \frac{\displaystyle\sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\left(\frac{h}{2^k}\right)^{2j} - \left(\frac{h}{2^{k-1}}\right)^{2j}\right)}{\displaystyle\sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\left(\frac{h}{2^{k+1}}\right)^{2j} - \left(\frac{h}{2^k}\right)^{2j}\right)} \\ &= \frac{\displaystyle\sum_{j=n}^{\infty} \hat{K}_{j,n-1}\left(\left(\frac{h}{2^k}\right)^{2j} - \left(\frac{h}{2^{k-1}}\right)^{2j}\right)}{\displaystyle\sum_{j=n}^{\infty} \hat{K}_{j,n-1}2^{-2j}\left(\left(\frac{h}{2^k}\right)^{2j} - \left(\frac{h}{2^{k-1}}\right)^{2j}\right)} \ .\end{aligned}$$
If we assume that terms for $j > n$ are negligible and drop them, we get (12.2.10). This is a nice theoretical observation but, in concrete computations, this criterion has a big weakness that limits its usefulness as shown in Question 12.13. Since we may expect that $L^{n-1}_{h/2^{k+1}}(f) \approx L^{n-1}_{h/2^k}(f)$, the formula in (12.2.10) involves a division by a number closed to 0. Thus, there is a large round off error in the computation of (12.2.10).

2. Let $f : [a, b] \to \mathbb{R}$ be an analytic function at $c \in [a, b]$. The central difference formula
$$L_h(f) = \frac{f(c+h) - f(c-h)}{2h}$$
is an approximation of $L(f) = f'(c)$ that satisfies (12.2.1). The Taylor series of f around c gives
$$f(c+h) = \sum_{j=0}^{\infty} \frac{1}{j!} f^{(j)}(c)h^j \quad \text{and} \quad f(c-h) = \sum_{j=0}^{\infty} (-1)^j \frac{1}{j!} f^{(j)}(c)h^j \ .$$
Hence,
$$\begin{aligned}L_h(f) = \frac{f(c+h) - f(c-h)}{2h} &= \frac{1}{2h}\left(\sum_{j=0}^{\infty}\left(1 - (-1)^j\right)\frac{1}{j!} f^{(j)}(c)h^j\right) \\ &= \sum_{j=0}^{\infty} \frac{1}{(2j+1)!} f^{(2j+1)}(c)h^{2j} \ .\end{aligned}$$
Solving for $f'(c)$ gives
$$f'(c) = L(f) = L_h(f) - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!} f^{(2j+1)}(c)h^{2j} \ .$$
So, it is justified to use Richardson extrapolation with the central difference formula.

3. The justification of Richardson extrapolation could have been based only on the hypothesis that
$$L(f) = L_h(f) + \sum_{j=1}^{m} K_j h^{2j} + O(h^{2n+1})$$
for m large enough instead of (12.2.1).

♠

Example 12.2.3
Use Richardson extrapolation with the central difference formula to approximate $f'(1)$ where $f(x) = \sin(x)$.

We have
$$L(f) = f'(1)$$
and
$$L_h(f) = \frac{\sin(1+h) - \sin(1-h)}{2h}\ .$$
With $h = 1.6/2^n$ for $0 \leq n \leq 7$, we give the values of (12.2.8) and (12.2.10) in Table 12.1.

A good approximation of $f'(1)$ is given by $L^4_{0.05}(f) \approx 0.54030230587$. The exact value is $f'(1) = \cos(1) = 0.54030230586814\ldots$. ♣

Code 12.2.4 (Richardson Table)

To generate the full Richardson table.
Input: The first column $T(1,1) = L_h(f)$, $T(2,1) = L_{h/2}(f)$, $\ldots$, $T(N,1) = L_{h/2^{N-1}}(f)$ of the Richardson table.
Output: The full Richardson table as represented in Table 12.1.

```
% T = richardson(col1)

function T = richardson(col1)
  % default arguments
  arguments
    col1 (:,1) double;
  end

  N = size(col1,1);
  T = repmat(NaN,N,2*N-1);
  T(:,1) = col1;

  for i=2:1:N
    T(i:1:N,2*i-1) = ((4^(i-1))*T(i:1:N,2*i-3) - T(i-1:1:N-1,2*i-3))...
                     /(4^(i-1)-1);
    if ( i < N )
      T(i:1:N-1,2*(i-1)) = (T(i:1:N-1,2*i-3) - T(i-1:1:N-2,2*i-3))...
```

h	$L_h(f)$		$L_h^1(f)$		$L_h^2(f)$		$L_h^3(f)$	
1.6	0.33754495163							
0.8	0.48448643755	3.538829	0.53346693286					
0.4	0.52600907074	3.881194	0.53984994847	15.065717	0.54027548285			
0.2	0.53670748767	$\boxed{3.970075}$	0.54027362665	15.761627	0.54030187186	61.776295	0.54030229073	
0.1	0.53940225217	$\boxed{3.992505}$	0.54030050700	$\boxed{15.940102}$	0.54030229903	$\boxed{63.436040}$	0.54030230581	250.129748
0.05	0.54007720805	$\boxed{3.998125}$	0.54030219334	$\boxed{15.985007}$	0.54030230576	$\boxed{63.859838}$	0.54030230587	$\boxed{256.958353}$
0.025	0.54024602614	$\boxed{3.999531}$	0.54030229883	$\boxed{15.996248}$	0.54030230587	$\boxed{63.924418}$	0.54030230587	117.388889
0.0125	0.54028823561		0.54030230543		0.54030230587		0.54030230587	

n	$L_h^4(f)$		$L_h^5(f)$		$L_h^6(f)$	$L_h^7(f)$
1.6						
0.8						
0.4						
0.2						
0.1	0.54030230587					
0.05	0.54030230587	−1388.888888889	0.54030230587			
0.025	0.54030230587	−0.8181818182	0.54030230587	−2.0000000000	0.54030230587	
0.0125	0.54030230587		0.54030230587		0.54030230587	0.54030230587

Table 12.1: Richardson table to approximate $\sin(x)$ near $x = 1$

```
                                    ./(T(i+1:1:N,2*i-3)-T(i:1:N-1,2*i-3));
    end
  end
end
```

12.3 Closed and Open Newton-Cotes Formulae

Let $f : [a, b] \to \mathbb{R}$ be a sufficiently continuously differentiable function on $[a, b]$. An approximation of $\int_a^b f(x)\,\mathrm{d}x$ is given by $\int_a^b p(x)\,\mathrm{d}x$, where p is an interpolating polynomial of f at some notes $x_0 < x_1 < \ldots x_n$. The **closed Newton-Cotes formulae** are based on interpolating polynomials p of f at the points $a = x_0 < x_1 < x_2 < \ldots < x_n = b$. The **open Newton-Cotes formulae** are based on interpolating polynomials p of f at the points $a < x_0 < x_1 < x_2 < \ldots < x_n < b$.

The following result from Analysis will be quite useful to derive integration formulae.

Theorem 12.3.1 (**Mean Value Theorem for Integrals**)

Let $a < b$ be two real numbers and $f : [a, b] \to \mathbb{R}$ be a continuous function. Let $g : [a, b] \to \mathbb{R}$ be an integrable function on $[a, b]$ such that g does not change sign on $[a, b]$. Then there exists c between a and b such that

$$\int_a^b f(x)\,g(x)\,\mathrm{d}x = f(c)\int_a^b g(x)\,\mathrm{d}x\ .$$

Theorem 12.3.2 (**Trapezoidal Rule**)

Suppose that $f : [a, b] \to \mathbb{R}$ is a twice continuously differentiable function on $[a, b]$. Then

$$\int_a^b f(x)\,\mathrm{d}x = \frac{f(a) + f(b)}{2}\,(b - a) - \frac{f''(\xi)\,(b - a)^3}{12}$$

for some ξ between a and b.

Proof.
We consider the interpolating polynomial p of f at the points $x_0 = a$ and $x_1 = b$; namely, $p(x) = f(a) + f[a, b](x - a)$. We have that $f(x) = p(x) + f[a, b, x](x - a)(x - b)$. Thus

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b f(a)\,\mathrm{d}x + \int_a^b f[a, b](x - a)\,\mathrm{d}x + \int_a^b f[a, b, x](x - a)(x - b)\,\mathrm{d}x\ .$$

We have that

$$\int_a^b f(a)\,\mathrm{d}x + \int_a^b f[a, b](x - a)\,\mathrm{d}x = f(a)\,(b - a) + f[a, b]\frac{(b - a)^2}{2} = \frac{f(a) + f(b)}{2}\,(b - a)$$

and the truncation error is

$$\begin{aligned}\int_a^b f[a,b,x](x-a)(x-b)\,\mathrm{d}x &= f[a,b,\eta]\int_a^b (x-a)(x-b)\,\mathrm{d}x = \frac{f''(\xi)}{2}\int_a^b (x-a)(x-b)\,\mathrm{d}x \\ &= \frac{f''(\xi)}{2}\left(\frac{x^3}{3}-(a+b)\frac{x^2}{2}+abx\right)\Bigg|_a^b = -\frac{f''(\xi)\,(b-a)^3}{12}\ .\end{aligned}$$

The first equality is a consequence of the Mean Value Theorem for Integrals because $(x-a)(x-b)$ does not change sign on $[a,b]$. The value η is between a and b. The second equality comes from Theorem 6.2.5 for some ξ between a and b. ∎

Remark 12.3.3

1. The trapezoidal rule is a closed Newton-Cotes formula.
2. If $|a-b|$ is small, $\displaystyle\int_a^b f(x)\,\mathrm{d}x \approx \frac{f(a)+f(b)}{2}(b-a)$ with the truncation error $\displaystyle -f''(\xi)\frac{(b-a)^3}{2}$.

♠

Theorem 12.3.4 (Simpson's Rule)

Suppose that $f:[a,b]\to\mathbb{R}$ is a four times continuously differentiable function on $[a,b]$. Then

$$\int_a^b f(x)\,\mathrm{d}x = \frac{f(a)+4f\big((a+b)/2\big)+f(b)}{6}\,(b-a) - \frac{f^{(4)}(\xi)\,(b-a)^5}{2880}$$

for some ξ between a and b.

Proof.
We consider the interpolating polynomial p of f at the points $x_0 = a$, $x_1 = (a+b)/2$ and $x_2 = b$; namely,

$$p(x) = f(a) + f[a,b](x-a) + f\left[a,\frac{a+b}{2},b\right](x-a)\left(x-\frac{a+b}{2}\right)\ .$$

We have that

$$f(x) = p(x) + f\left[a,\frac{a+b}{2},b,x\right](x-a)\left(x-\frac{a+b}{2}\right)(x-b)\ .$$

Thus

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \int_a^b f(a)\,\mathrm{d}x + \int_a^b f\left[a,\frac{a+b}{2}\right](x-a)\,\mathrm{d}x \\ &\quad + \int_a^b f\left[a,\frac{a+b}{2},b\right](x-a)\left(x-\frac{a+b}{2}\right)\mathrm{d}x \\ &\quad + \int_a^b f\left[a,\frac{a+b}{2},b,x\right](x-a)\left(x-\frac{a+b}{2}\right)(x-b)\,\mathrm{d}x\ .\end{aligned}$$

Expanding the divide differences, we get

$$\begin{aligned}
&\int_a^b f(a)\,\mathrm{d}x + \int_a^b f\left[a, \frac{a+b}{2}\right](x-a)\,\mathrm{d}x + \int_a^b f\left[a, \frac{a+b}{2}, b\right](x-a)\left(x - \frac{a+b}{2}\right)\mathrm{d}x \\
&\quad = f(a)\,(b-a) + f\left[a, \frac{a+b}{2}\right]\frac{(b-a)^2}{2} + f\left[a, \frac{a+b}{2}, b\right]\frac{(b-a)^3}{12} \\
&\quad = \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)\left(\frac{b-a}{6}\right)\ .
\end{aligned}$$

The truncation error is

$$R = \int_a^b f\left[a, \frac{a+b}{2}, b, x\right](x-a)\left(x - \frac{a+b}{2}\right)(x-b)\,\mathrm{d}x\ .$$

We cannot use the Mean Value Theorem for Integrals to evaluate this integral because $(x-a)\left(x - \frac{a+b}{2}\right)(x-b)$ changes same sign on $[a,b]$. But, from

$$f\left[\frac{a+b}{2}, a, \frac{a+b}{2}, b, x\right] = \frac{f\left[a, \frac{a+b}{2}, b, x\right] - f\left[\frac{a+b}{2}, a, \frac{a+b}{2}, b\right]}{x - \frac{a+b}{2}}\ ,$$

we get

$$\begin{aligned}
R &= \int_a^b f\left[\frac{a+b}{2}, a, \frac{a+b}{2}, b\right](x-a)\left(x - \frac{a+b}{2}\right)(x-b)\,\mathrm{d}x \\
&\quad + \int_a^b f\left[\frac{a+b}{2}, a, \frac{a+b}{2}, b, x\right](x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)\,\mathrm{d}x\ .
\end{aligned}$$

The first integral is 0 because $(x-a)\left(x - \frac{a+b}{2}\right)(x-b)$ is like an odd function with respect to the line $x = \frac{a+b}{2}$. We can use the Mean Value Theorem for Integrals for the second integral because $(x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)$ does not change sign on $[a,b]$. Hence, there exists η between a and b such that

$$\begin{aligned}
R &= f\left[\frac{a+b}{2}, a, \frac{a+b}{2}, b, \eta\right]\int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)\,\mathrm{d}x \\
&= \frac{f^{(4)}(\xi)}{4!}\int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)\,\mathrm{d}x = -\frac{f^{(4)}(\xi)}{4!}\,\frac{(b-a)^5}{120} = -\frac{f^{(4)}(\xi)(b-a)^5}{2880}\ .
\end{aligned}$$

The second equality follows from Theorem 6.2.5 for some ξ between a and b. ∎

Remark 12.3.5

1. The Simpson's rule is also a closed Newton-Cotes formula.
2. If $|a-b|$ is small, $\int_a^b f(x)\,\mathrm{d}x \approx \frac{1}{2}\left(f(a) + f\left(\frac{a+b}{2}\right) + f(b)\right)(b-a)$ and the truncation error is $-\frac{f^{(4)}(\xi)}{2880}\,(b-a)^5$.

♠

Theorem 12.3.6 (Midpoint Rule)

Suppose that $f:[a,b]\to\mathbb{R}$ is twice continuously differentiable function on $[a,b]$. Then

$$\int_a^b f(x)\,\mathrm{d}x = f\left(\frac{a+b}{2}\right)(b-a) + \frac{f''(\xi)}{24}\,(b-a)^3$$

for some ξ between a and b.

Proof.

We consider the interpolating polynomial p of f at the point $x_0 = \dfrac{a+b}{2}$; namely, $p(x) = f\left(\dfrac{a+b}{2}\right)$. We have that $f(x) = f\left(\dfrac{a+b}{2}\right) + f\left[\dfrac{a+b}{2}, x\right]\left(x - \dfrac{a+b}{2}\right)$. Thus

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b f\left(\frac{a+b}{2}\right)\mathrm{d}x + \int_a^b f\left[\frac{a+b}{2}, x\right](x-a)\,\mathrm{d}x\ .$$

We have that

$$\int_a^b f\left(\frac{a+b}{2}\right)\mathrm{d}x = f\left(\frac{a+b}{2}\right)(b-a)\ .$$

The truncation error is

$$R = \int_a^b f\left[\frac{a+b}{2}, x\right]\left(x-\frac{a+b}{2}\right)\mathrm{d}x\ .$$

We cannot use the Mean Value Theorem for Integrals to evaluate this integral because $\left(x - \dfrac{a+b}{2}\right)$ changes same sign on $[a,b]$. But, from

$$f\left[\frac{a+b}{2}, \frac{a+b}{2}, x\right] = \frac{f\left[\frac{a+b}{2}, x\right] - f\left[\frac{a+b}{2}, \frac{a+b}{2}\right]}{x - \frac{a+b}{2}}\ ,$$

we get

$$R = \int_a^b f\left[\frac{a+b}{2}, \frac{a+b}{2}\right]\left(x-\frac{a+b}{2}\right)\mathrm{d}x + \int_a^b f\left[\frac{a+b}{2}, \frac{a+b}{2}, x\right]\left(x-\frac{a+b}{2}\right)^2\mathrm{d}x\ .$$

The first integral is 0 because $\left(x - \dfrac{a+b}{2}\right)$ is like an odd function with respect to the line $x = \dfrac{a+b}{2}$. We can use the Mean Value Theorem for Integrals in the second integral because $\left(x - \dfrac{a+b}{2}\right)^2$ does not change sign on $[a,b]$. Hence, there exists η between a and b such that

$$\begin{aligned}R &= f\left[\frac{a+b}{2}, \frac{a+b}{2}, \eta\right]\int_a^b \left(x-\frac{a+b}{2}\right)^2\mathrm{d}x = \frac{f''(\xi)}{2!}\int_a^b\left(x-\frac{a+b}{2}\right)^2\mathrm{d}x \\ &= \frac{f''(\xi)}{2!}\,\frac{\left(x-\frac{a+b}{2}\right)^3}{3}\Bigg|_a^b = \frac{f''(\xi)}{24}\,(b-a)^3\ .\end{aligned}$$

The second equality follows from Theorem 6.2.5 for some ξ between a and b. ∎

Remark 12.3.7

1. The midpoint rule is an open Newton-Cotes formula.

2. If $|a-b|$ is small, $\displaystyle\int_a^b f(x)\,\mathrm{d}x \approx f\left(\frac{a+b}{2}\right)(b-a)$ and the truncation error is $\dfrac{f''(\xi)}{24}(b-a)^3$.

♠

12.4 Composite Numerical Integration

Let $f : [a,b] \to \mathbb{R}$ be a sufficiently continuously differentiable function. The trapezoidal, Simpson's and midpoint rules do not give good approximations of $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ if the interval $[a,b]$ is large. To get better approximations of $\displaystyle\int_a^b f(x)\,\mathrm{d}x$, we divide the interval $[a,b]$ into small subintervals of equal lengths and apply the trapezoidal, Simpson's and midpoint rules on each subintervals.

1. Let $x_0 = a < x_1 < \ldots < x_n = b$. Since $\displaystyle\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x$, the sum of the approximation of $\displaystyle\int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x$ for $1 \le i \le n$ gives an approximation of $\displaystyle\int_a^b f(x)\,\mathrm{d}x$. If a linear interpolating polynomial of f at the two endpoints x_{i-1} and x_i is used on each subinterval $[x_{i-1}, x_i]$ to approximate f, we get the composite trapezoidal rule.

2. Let $x_0 = a < x_1 < \ldots < x_{n=2m} = b$. Since $\displaystyle\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{m} \int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x$, the sum of the approximation of $\displaystyle\int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x$ for $1 \le i \le m$ gives an approximation of $\displaystyle\int_a^b f(x)\,\mathrm{d}x$. If a quadratic interpolating polynomial of f at the three points x_{2i-2}, x_{2i-1} and x_{2i} is used on each subinterval $[x_{2i-2}, x_{2i}]$ to approximate f, we get the composite Simpson's rule.

3. Let $x_0 = a < x_1 < \ldots < x_{n=2m} = b$. Since $\displaystyle\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{m} \int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x$, the sum of the approximation of $\displaystyle\int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x$ for $1 \le i \le m$ gives an approximation of $\displaystyle\int_a^b f(x)\,\mathrm{d}x$. If a constant interpolating polynomial of f at the middle point x_{2i-1} is used on each subinterval $[x_{2i-2}, x_{2i}]$ to approximate f, namely $f(x)$ is approximated by $f(x_{2i-1})$ for all $x \in [x_{2i-2}, x_{2i}]$, we get the composite midpoint rule.

12.4.1 Composite Trapezoidal Rule

Theorem 12.4.1 (Composite Trapezoidal Rule)

Let $f : [a,b] \to \mathbb{R}$ be a twice continuously differentiable function. Let $h = (b-a)/n$ and $x_j = a + j\,h$ for $j = 0, 1, \ldots, n$. Then

$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) - \frac{f''(\xi)\,(b-a)}{12}\,h^2$$

for some $\xi \in [a,b]$.

Proof.
Using the trapezoidal rule on $[x_{j-1}, x_j]$ for $1 \leq j \leq n$, we get that

$$\begin{aligned}\int_{x_{j-1}}^{x_j} f(x)\,\mathrm{d}x &= \frac{f(x_{j-1}) + f(x_j)}{2}\,(x_j - x_{j-1}) - \frac{f''(\xi_j)}{12}\,(x_j - x_{j-1})^3 \\ &= \frac{f(x_{j-1}) + f(x_j)}{2}\,h - \frac{f''(\xi_j)}{12}\,h^3\end{aligned}$$

for some $\xi_j \in [x_{j-1}, x_j]$, where we have used $x_j - x_{j-1} = h$.

Since

$$\min_{x\in[a,b]} f''(x) \leq \frac{1}{n}\sum_{j=1}^{n} f''(\xi_j) \leq \max_{x\in[a,b]} f''(x)\ ,$$

there exists $\xi \in [a,b]$ such that

$$f''(\xi) = \frac{1}{n}\sum_{j=1}^{n} f''(\xi_j)$$

by the Intermediate Value Theorem. Hence,

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \sum_{j=1}^{n}\int_{x_{j-1}}^{x_j} f(x)\,\mathrm{d}x &= \frac{h}{2}\sum_{j=1}^{n}\left(f(x_{j-1}) + f(x_j)\right) - \frac{h^3}{12}\sum_{j=1}^{n} f''(\xi_j) \\ &= \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) - \frac{h^3}{12}\,n f''(\xi) \\ &= \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) - \frac{f''(\xi)\,(b-a)}{12}\,h^2\end{aligned}$$

because $h = (b-a)/n$. ∎

Remark 12.4.2
We have $\displaystyle\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right)$ and the truncation error is $\displaystyle -\frac{f''(\xi)\,(b-a)}{12}\,h^2$. ♠

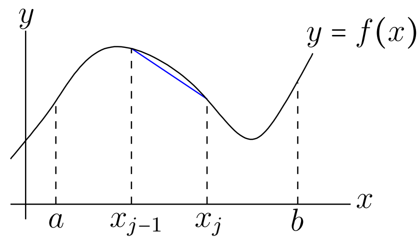


Figure 12.2: Trapezoidal Rule

Example 12.4.3

Use the composite trapezoidal rule to approximate $\int_0^1 e^{x^2}\,\mathrm{d}x$. Choose the number of subintervals such that the magnitude of the truncation error is smaller than 10^{-4}.

We first choose n such that the magnitude of the truncation error in the composite trapezoidal rule (Theorem 12.4.1) is smaller than 10^{-4}; namely,

$$\left| \frac{f''(\xi)\,(b-a)}{12}\, h^2 \right| < 10^{-4} \ .$$

We have $f(x) = e^{x^2}$, $a = 0$, $b = 1$, $x_j = jh$ and $h = 1/n$. Since $|f''(x)| = (2+4x^2)e^{x^2}$, we have that $|f''(x)| \le 6e$ for $x \in [0,1]$. The magnitude of the truncation error is at most $\dfrac{e}{2n^2}$. We choose n such that $\dfrac{e}{2n^2} < 10^{-4}$; namely, $n > 116.5821991\ldots$. With $n = 117$, we get $h = 1/117$ and

$$\begin{aligned}
\int_a^b f(x)\,\mathrm{d}x &\approx \frac{1}{234}\left(f(0) + 2\sum_{j=1}^{116} f(j/117) + f(1) \right) \\
&= \frac{1}{234}\left(1 + 2e^{(1/117)^2} + 2e^{(2/117)^2} + \ldots + 2e^{(115/117)^2} + 2e^{(116/117)^2} + e \right) \approx 1.46268 \ .
\end{aligned}$$

♣

12.4.2 Composite Simpson's Rule

Theorem 12.4.4 (Composite Simpson's Rule)

Let $f : [a,b] \to \mathbb{R}$ be a four times continuously differentiable function. Let $n = 2m$, $h = (b-a)/n$ and $x_j = a + j\,h$ for $j = 0,\ 1,\ \ldots,\ n$. then

$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + f(x_n) \right) - \frac{f^{(4)}(\xi)\,(b-a)}{180}\, h^4$$

for some $\xi \in [a,b]$.

Proof.
Using the Simpson's rule on $[x_{2j-2}, x_{2j}]$ for $1 \le j \le m$, we get that

$$\int_{x_{2j-2}}^{x_{2j}} f(x)\,\mathrm{d}x = \frac{f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})}{6}\,(x_{2j} - x_{2j-2}) - \frac{f^{(4)}(\xi_j)}{2880}\,(x_{2j} - x_{2j-2})^5$$
$$= \frac{f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})}{3} h - \frac{f^{(4)}(\xi_j)}{90}\,h^5$$

for some $\xi_j \in [x_{2j-2}, x_{2j}]$, where we have used $x_{2j} - x_{2j-2} = 2h$.

Since

$$\min_{x\in[a,b]} f^{(4)}(x) \le \frac{1}{m}\sum_{j=1}^{m} f^{(4)}(\xi_j) \le \max_{x\in[a,b]} f^{(4)}(x) \ ,$$

there exists $\xi \in [a, b]$ such that

$$f^{(4)}(\xi) = \frac{1}{m}\sum_{j=1}^{m} f^{(4)}(\xi_j)$$

by the Intermediate Value Theorem. Hence,

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{j=1}^{m}\int_{x_{2j-2}}^{x_{2j}} f(x)\,\mathrm{d}x = \frac{h}{3}\sum_{j=1}^{m}\left(f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\right) - \frac{h^5}{90}\sum_{j=1}^{m} f^{(4)}(\xi_j)$$
$$= \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + f(x_{2m})\right) - \frac{h^5}{90}\,m f^{(4)}(\xi)$$
$$= \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + f(x_{2m})\right) - \frac{h^4(b-a)}{180} f^{(4)}(\xi)$$

because $h = (b-a)/(2m)$. ∎

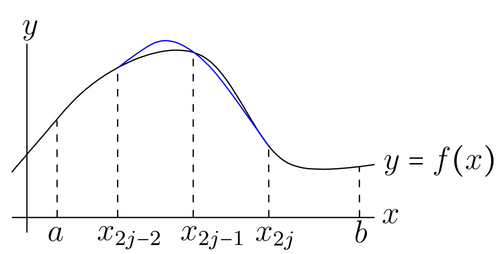


Figure 12.3: Simpson's Rule

Remark 12.4.5
We have $\displaystyle\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + f(x_n)\right)$ with the truncation error $\displaystyle -\frac{f^{(4)}(\xi)\,(b-a)}{180}\,h^4$. ♠

Example 12.4.6

Use the composite Simpson's rule to approximate $\displaystyle\int_0^1 e^{x^2}\,\mathrm{d}x$. Choose the number of subintervals such that the magnitude of the truncation error is smaller than 10^{-4}.

We first choose m such that the magnitude of the truncation error in the composite Simpson's rule (Theorem 12.4.4) is smaller than 10^{-4}; namely,

$$\left|\frac{f^{(4)}(\xi)\,(b-a)}{180}\,h^4\right| < 10^{-4} \ .$$

We have $f(x) = e^{x^2}$, $a = 0$, $b = 1$, $x_j = jh$ and $h = 1/n$ where $n = 2m$. Since $|f^{(4)}(x)| = 4e^{x^2}\left(3 + 12\,x^2 + 4\,x^4\right)$, we have that $|f^{(4)}(x)| \leq 76e$ on $[0,1]$. Thus, the magnitude of the truncation error is at most $\dfrac{76e}{180(2m)^4} = \dfrac{19e}{720m^4}$. We choose m such that $\dfrac{19e}{720m^4} < 10^{-4}$; namely, $m > 5.175220955\ldots$. With $m = 6$, we get $h = 1/12$ and

$$\begin{aligned}\int_0^1 e^{x^2}\,\mathrm{d}x &\approx \frac{1}{36}\left(f(0) + 2\sum_{j=1}^{5} f(j/6) + 4\sum_{j=0}^{5} f((2j+1)/12) + f(1)\right)\\ &= \frac{1}{36}\left(1 + 2\left(e^{(2/12)^2} + \ldots + e^{(10/12)^2}\right) + 4\left(e^{(1/12)^2} + \ldots + e^{(11/12)^2}\right) + e\right) \approx 1.46267 \ .\end{aligned}$$

♣

Code 12.4.7 (Composite Simpson's Rule)

To approximate the value of the integral

$$\int_a^b f(x)\,\mathrm{d}x \ .$$

Input: The function f (Denoted funct in the code below).
The endpoints a and b.
The number m which is half the number of subintervals that will be used.
Output: The approximation to the value of the integral.

```
%  s = simpson(funct,a,b,m)

function s = simpson(funct,a,b,m)
  N = 2*m;
  h = (b-a)/N;
  if ( m > 1)
    x = linspace(a,b,N+1);
    x4 = x(2:2:N);
    x2 = x(3:2:N-1);
    s = h*(funct(a) + funct(b) + 2*sum(funct(x2)) + 4*sum(funct(x4)))/3;
  else
    s = h*(funct(a) + funct(b) + 4*funct(a+h))/3;
  end
end
```

12.4.3 Composite Midpoint Rule

Theorem 12.4.8 (Composite Midpoint Rule)

Let $f : [a,b] \to \mathbb{R}$ be a twice continuously differentiable function. Let $n = 2m$, $h = (b-a)/n$ and $x_j = a + j\,h$ for $j = 0, 1 \ldots, 2m$. Then

$$\int_a^b f(x)\,\mathrm{d}x = 2h \sum_{j=1}^{m} f(x_{2j-1}) + \frac{f''(\xi)\,(b-a)}{6}\,h^2$$

for some $\xi \in [a,b]$.

Proof.
Using the midpoint rule on $[x_{2j-2}, x_{2j}]$ for $1 \leq j \leq m$, we get that

$$\int_{x_{2j-2}}^{x_{2j}} f(x)\,\mathrm{d}x = f(x_{2j-1})\,(x_{2j} - x_{2j-2}) + \frac{f''(\xi_j)}{24}\,(x_{2j} - x_{2j-2})^3 = 2h\,f(x_{2j-1}) + \frac{f''(\xi_j)\,h^3}{3}$$

for some $\xi_j \in [x_{2j-2}, x_{2j}]$, where we have used $x_{2j} - x_{2j-2} = 2h$.

Again, as in the proofs of Theorem 12.4.1 for the composite trapezoidal rule and Theorem (12.4.4) for the Simpson's rule, since

$$\min_{x\in[a,b]} f''(x) \leq \frac{1}{m}\sum_{j=1}^{m} f''(\xi_j) \leq \max_{x\in[a,b]} f''(x)\ ,$$

there exists $\xi \in [a,b]$ such that

$$f''(\xi) = \frac{1}{m}\sum_{j=1}^{m} f''(\xi_j)$$

by the Intermediate Value Theorem. Hence,

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \sum_{j=1}^{m}\int_{x_{2j-2}}^{x_{2j}} f(x)\,\mathrm{d}x = 2h\sum_{j=1}^{m} f(x_{2j-1}) + \frac{h^3}{3}\sum_{j=1}^{m} f''(\xi_j) \\ &= 2h\sum_{j=1}^{m} f(x_{2j-1}) + \frac{h^3}{3}\,m\,f''(\xi) = 2h\sum_{j=1}^{m} f(x_{2j-1}) + \frac{f''(\xi)\,(b-a)}{6}\,h^2\end{aligned}$$

because $h = (b-a)/(2m)$. ∎

Remark 12.4.9
We have that $\displaystyle\int_a^b f(x)\,\mathrm{d}x \approx 2h\sum_{j=1}^{m} f(x_{2j-1})$ and the truncation error is $\dfrac{f''(\xi)\,(b-a)}{6}\,h^2$. ♠

Example 12.4.10
Use the composite midpoint rule to approximate $\displaystyle\int_0^1 e^{x^2}\,\mathrm{d}x$. Choose the number of subintervals such that the magnitude of the truncation error is smaller than 10^{-4}. Compare with Examples 12.4.6 and 12.4.3.

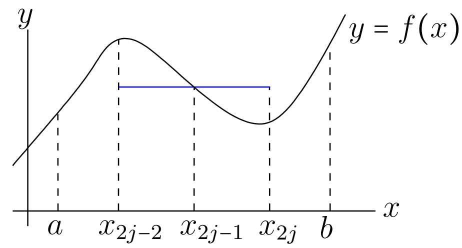


Figure 12.4: Midpoint Rule

We first choose m such that the magnitude of the truncation error in the composite midpoint rule (Theorem 12.4.8) is smaller than 10^{-4}; namely,

$$\left|\frac{f''(\xi)\,(b-a)}{6}\,h^2\right| < 10^{-4} \ .$$

We have $f(x) = e^{x^2}$, $a = 0$, $b = 1$, $x_j = j\,h$ and $h = 1/n$ where $n = 2m$. Since $|f''(x)| = (2+4x^2)e^{x^2}$, we have that $|f''(x)| \leq 6e$ for $x \in [0,1]$. The magnitude of the truncation error is at most $\dfrac{e}{(2m)^2}$. We choose m such that $\dfrac{e}{(2m)^2} < 10^{-4}$; namely, $m > 82.436\ldots$. With $m = 83$, we get $h = 1/166$ and

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \frac{1}{83}\sum_{j=1}^{83} f((2j-1)/166) = \frac{1}{83}\left(e^{(1/166)^2} + e^{(3/166)^2} + \cdots + e^{(163/166)^2} + e^{(165/166)^2}\right) \\ &\approx 1.4626189 \ .\end{aligned}$$

♣

Remark 12.4.11
Contrary to numerical differentiation, numerical integration is stable with respect to rounding error. We demonstrate this with the composite Simpson's rule.

Let $f : [a,b] \to \mathbb{R}$ be a four times continuously differentiable function. Moreover, let $n = 2m$, $h = (b-a)/n$ and $x_j = a + j\,h$ for $j = 0, 1, \ldots, n$. Finally, let f_i be the computed value of $f(x_i)$ and $e_i = f_i - f(x_i)$ be the rounding error in computing $f(x_i)$.

From Theorem 12.4.4, there exists $\xi \in [a,b]$ such that

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + f(x_n)\right) - \frac{f^{(4)}(\xi)\,(b-a)}{180}\,h^4 \\ &= \frac{h}{3}\left(f_0 + 2\sum_{j=1}^{m-1} f_{2j} + 4\sum_{j=0}^{m-1} f_{2j+1} + f_n\right) - R(h) \ ,\end{aligned}$$

where

$$R(h) = \frac{h}{3}\left(e_0 + 2\sum_{j=1}^{m-1} e_{2j} + 4\sum_{j=0}^{m-1} e_{2j+1} + e_n\right) + \frac{f^{(4)}(\xi)\,(b-a)}{180}\,h^4$$

is the error. We have assumed that the arithmetic operations in

$$\frac{h}{3}\left(f_0 + 2\sum_{j=1}^{m-1} f_{2j} + 4\sum_{j=0}^{m-1} f_{2j+1} + f_n\right)$$

can be performed without rounding error to simplify the discussion.

Suppose that the rounding errors e_i are uniformly bounded by r, namely $|e_i| < r$ for all i, and $M = \sup_{x\in[a,b]} |f^{(4)}(x)|$. Then

$$\begin{aligned}|R(h)| &\leq \frac{h}{3}\left(r + 2\sum_{j=1}^{m-1} r + 4\sum_{j=0}^{m-1} r + r\right) + \frac{M\,(b-a)}{180}\,h^4 \\ &= 2hmr + \frac{M\,(b-a)}{180}\,h^4 = (b-a)r + \frac{M\,(b-a)}{180}\,h^4\end{aligned}$$

because $h = (b-a)/(2m)$. Thus $R(h)$ is bounded for small h. $R(h)$ does not blow up as h gets smaller. ♠

12.5 Romberg Integration

Romberg integration is nothing else than Richardson extrapolation with $L(f) = \int_a^b f(x)\,\mathrm{d}x$ and

$$L_h(f) = \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) , \tag{12.5.1}$$

where $x_j = a + j\,h$ with $h = (b-a)/n$. $L_h(f)$ is the approximation formula for the composite trapezoidal rule.

Remark 12.5.1

1. We will prove in Theorem 12.8.5 of Section 12.8 that (12.2.1) is satisfied for h small if f is smooth. Thus, Richardson extrapolation can be used with the trapezoidal rule.

2. The value of $L_h(f)$ can be used to reduce the number of operations in the computation of $L_{h/2}(f)$.

   Suppose that $L_h(f)$ is given by (12.5.1). Let $\tilde{n} = 2n$, $\tilde{h} = \dfrac{b-a}{\tilde{n}} = \dfrac{h}{2}$ and $\tilde{x}_j = a + j\,\tilde{h}$. Then, because $\tilde{x}_{2j} = x_j$ for all j, we get

$$\begin{aligned}L_{h/2}(f) &= \frac{\tilde{h}}{2}\left(f(\tilde{x}_0) + 2\sum_{j=1}^{\tilde{n}-1} f(\tilde{x}_j) + f(\tilde{x}_{\tilde{n}})\right) \\ &= \frac{1}{2}\left(\frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) + h\sum_{j=1}^{n} f(\tilde{x}_{2j-1})\right) \\ &= \frac{1}{2}\left(L_h(f) + M_{h/2}(f)\right) ,\end{aligned}$$

where
$$M_{h/2}(f) = h \sum_{j=1}^{n} f(\tilde{x}_{2j-1})$$
is the approximation formula given by the composite midpoint rules with $2n$ subintervals.

♠

Romberg Integration may be used with functions which are only continuous and, therefore, do not satisfy (12.2.1) in theory.

Theorem 12.5.2

If $f : [a, b] \leftarrow \mathbb{R}$ is a continuous function, then
$$L^n_{h/2^k}(f) \to L(f) \quad \text{as} \quad k \to \infty \, . \tag{12.5.2}$$

Proof.
The proof of (12.5.2) is by induction on n.

We rewrite $L^0_{h/2^k}(f)$ as
$$L^0_{h/2^k}(f) = \frac{1}{2}\left(\sum_{j=0}^{2^k-1} f(x_j)h + \sum_{j=1}^{2^k} f(x_j)h\right) ,$$
where $x_j = a + j\,h$ and $h = \dfrac{b-a}{2^k n}$. The two sums are Riemann sums (the left and right sums) that converge to the value of the integral $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ as $k \to \infty$. Hence,
$$\lim_{k\to\infty} L^0_{h/2^k}(f) = \frac{1}{2}\left(\int_a^b f(x)\,\mathrm{d}x + \int_a^b f(x)\,\mathrm{d}x\right) = \int_a^b f(x)\,\mathrm{d}x \, .$$
This proves (12.5.2) for $n = 0$.

We assume that (12.5.2) is true for n: that is $L^n_{h/2^k}(f) \to L(f)$ as $k \to \infty$. Then,
$$\begin{aligned}\lim_{k\to\infty} L^{n+1}_{h/2^k}(f) &= \frac{4^{n+1}\lim_{k\to\infty} L^n_{h/2^k}(f) - \lim_{k\to\infty} L^n_{h/2^{k-1}}(f)}{4^{n+1}-1} \\ &= \frac{4^{n+1}\displaystyle\int_a^b f(x)\,\mathrm{d}x - \int_a^b f(x)\,\mathrm{d}x}{4^{n+1}-1} = \int_a^b f(x)\,\mathrm{d}x \, .\end{aligned}$$
This proves (12.5.2) for $n+1$ instead of n and complete the proof by induction. ∎

12.6 Adaptive Quadrature Methods

Let $f : [a, b] \to \mathbb{R}$ be a sufficiently continuously differentiable function. Our goal is to approximate the integral

$$\int_a^b f(x)\,\mathrm{d}x$$

with an accuracy of $\epsilon > 0$.

In the composite methods of Section 12.4, the **step size** (the distance between the points x_i in the partition of the interval $[a, b]$) was constant. To get a good approximation of the integral, it would be advantageous to choose a smaller step size where the function f varies more rapidly. This is the idea motivating the adaptive quadrature methods.

There are many adaptive quadrature methods. The adaptive quadrature method that we consider in this subsection is based on the composite Simpson's rule.

Let $a = x_0 < x_1 < \ldots < x_n = b$ be a partition of $[a, b]$. The x_i may not be equally spaced. We compute two approximations of

$$I_i = \int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x$$

using the composite Simpson's rule. With $m = 1$, $a = x_{i-1}$, $b = x_i$ and $h = h_i = (x_i - x_{i-1})/2$ in Theorem 12.4.4, we get $I_i = S_i + R_i$, where

$$S_i = \frac{h_i}{3}\left(f(x_{i-1}) + 4f(x_{i-1} + h_i) + f(x_i)\right)$$

and

$$R_i = -\frac{f^{(4)}(\eta_i)\,(x_i - x_{i-1})}{180}\,h_i^4 = -\frac{f^{(4)}(\eta_i)}{90}\,h_i^5$$

for η_i between x_{i-1} and x_i. With $m = 2$, $a = x_{i-1}$, $b = x_i$ and $h = h_i/2 = (x_i - x_{i-1})/4$ in Theorem 12.4.4, we get $I_i = \tilde{S}_i + \tilde{R}_i$, where

$$\tilde{S}_i = \frac{h_i}{6}\left(f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{2}\right) + 2f(x_{i-1} + h_i) + 4f\left(x_{i-1} + \frac{3h_i}{2}\right) + f(x_i)\right)$$

and

$$\tilde{R}_i = -\frac{f^{(4)}(\mu_i)\,(x_i - x_{i-1})}{180}\left(\frac{h_i}{2}\right)^4 = -\frac{f^{(4)}(\mu_i)}{90 \times 16}\,h_i^5\ ,$$

for $h_i = (x_i - x_{i-1})/2$ and μ_i between x_{i-1} and x_i.

If we assume that $f^4(x)$ is almost constant on $[x_{i-1}, x_i]$, we may suppose that $f^4(\eta_i) \approx f^4(\mu_i)$. Hence $R_i \approx 16\tilde{R}_i$.

If we subtract $I_i = \tilde{S}_i + \tilde{R}_i$ from $I_i = S_i + R_i =\approx S_i + 16\tilde{R}_i$, we get $0 \approx (S_i - \tilde{S}_i) + 15\tilde{R}_i$. Thus $\tilde{R}_i \approx \dfrac{1}{15}\left(\tilde{S}_i - S_i\right)$.

In summary,

$$I_i \approx \tilde{S}_i + \frac{1}{15}\left(\tilde{S}_i - S_i\right) .$$

Thus $\tilde{S}_i$ is an approximation of I_i with truncation error almost equals to $\frac{1}{15}\left(\tilde{S}_i - S_i\right)$.

Suppose that $x_0, x_1, \ldots, x_n$ is a partition of $[a,b]$ such that

$$\frac{1}{15}\left(\tilde{S}_i - S_i\right) < \frac{x_i - x_{i-1}}{b-a}\,\epsilon \tag{12.6.1}$$

for $1 \leq i \leq n$. Then

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x \approx \sum_{i=1}^n \tilde{S}_i$$

and the approximation $\sum_{i=1}^n \frac{1}{15}\left(\tilde{S}_i - S_i\right)$ of the truncation error satisfies

$$\left|\sum_{i=1}^n \frac{1}{15}\left(\tilde{S}_i - S_i\right)\right| \leq \sum_{i=1}^n \left|\frac{1}{15}\left(\tilde{S}_i - S_i\right)\right| < \sum_{i=1}^n \frac{x_i - x_{i-1}}{b-a}\,\epsilon = \epsilon .$$

If

$$\frac{1}{15}\left(\tilde{S}_i - S_i\right) \geq \frac{x_{i+1} - x_i}{b-a}\,\epsilon$$

for some i, we add a point in $]x_{i-1}, x_i[$ to our partition of $[a,b]$. Usually, we choose the midpoint of $]x_i, x_{i+1}[$. This has the effect of splitting the interval $[x_i, x_{i+1}]$ into two smaller intervals. We hope that, with this new finer partition that we also call $x_0, x_1, \ldots, x_n$, the relation (12.6.1) will be satisfied for all i. If (12.6.1) is not satisfy for all i, we keep on adding points to the partition as we just did. We hope that after having added a finite number of points to the initial partition of $[a,b]$, (12.6.1) will be satisfied for all i.

The following code implement the adaptive method above.

Code 12.6.1 (Adaptive Method Based on the Composite Simpson's Rule)

To approximate the integral

$$\int_a^b f(x)\,\mathrm{d}x .$$

Input: The endpoints of the interval $[a,b]$.
The function f (denoted funct in the code below).
The maximal tolerated error T.
The maximal number of times Max that the program may subdivide the interval $[a,b]$.
Output: The program gives the approximation of the integral if it does not have to subdivide the interval $[a,b]$ more than Max times to reach the accuracy T.

```
function sum = simpson_adapt(funct,a,b,T,Max)
  sum = nested_adaptive(funct,a,b,T,Max,simpsonNC(funct,a,b));
end

function sum = nested_adaptive(funct,a,b,T,Max,S)
```

```
    if (Max < 0 )
      sum = NaN;
      return;
    end

    mid = (a+b)/2;
    sum1 = S;
    sum2L = simpsonNC(funct,a,mid);
    sum2R = simpsonNC(funct,mid,b);
    sum2 = sum2L + sum2R;

    if ( abs(sum1-sum2)/15  < T)
      sum = sum2;
    else
      sum = nested_adaptive(funct,a,mid,T/2,Max-1,sum2L) + ...
            nested_adaptive(funct,mid,b,T/2,Max-1,sum2R);
    end
  end

  function sum = simpsonNC(funct,a,b)
    sum = (b-a).*(funct(a) + 4*funct((a+b)/2) + funct(b))/6;
  end
```

Example 12.6.2
Use the adaptive quadrature method defined above to approximate

$$\int_0^1 \sqrt{x}\,\mathrm{d}x$$

with an accuracy of 0.0005.

For this purpose and to simplify the discussion, let $S(a,b,h)$ is the approximation of $\displaystyle\int_a^b \sqrt{x}\,\mathrm{d}x$ given by the composite Simpson's rule, Theorem 12.4.4, with $m = (b-a)/(2h)$. The values displayed in the following computations have been rounded to at least 6 significant digits though the computations have been done with as many digits as possible.

Level 0:

i	1	2	3	4	5
x_i	0	1/4	1/2	3/4	1

$h = 0.5$, $T = 0.0005$, $S_{[0,1]} = S(0,1,0.5) = 0.63807119$,
$S_1 = S(0,0.5,0.25) = 0.22559223$, $S_2 = S(0.5,1,0.25) = 0.43093403$,
$\tilde{S}_{[0,1]} = S(0,1,0.25) = S_1 + S_2$ and $\tilde{R}_{[0,1]} \approx \dfrac{1}{15}\left|\tilde{S}_{[0,1]} - S_{[0,1]}\right| \approx 0.123034 \times 10^{-2} \not< 0.0005$.

Level 1:

i	1	2	3	4	5
x_i	0.0	1/8	1/4	3/8	1/2

$h = 0.25$, $T = 0.00025$ (stored for $[0.5,1]$), $S_{[0,0.5]} = S(0,0.5,0.25) = 0.22559223$,
$S_1 = S(0,0.25,0.125) = 0.07975890$, $S_2 = S(0.25,0.5,0.125) = 0.15235819$,

$\tilde{S}_{[0,0.5]} = S(0, 0.5, 0.125) = S_1 + S_2$ and
$\tilde{R}_{[0,0.5]} \approx \frac{1}{15}\left|\tilde{S}_{[0,0.5]} - S_{[0,0.5]}\right| \approx 0.43499 \times 10^{-3} \nless 0.00025.$

Level 2:

i	1	2	3	4	5
x_i	0.0	1/16	1/8	3/16	1/4

$h = 0.125$, $T = 0.000125$ (stored for $[0.25, 0.5]$), $S_{[0,0.25]} = S(0, 0.25, 0.125) = 0.07975890$,
$S_1 = S(0, 0.125, 0.0625) = 0.02819903$, $S_2 = S(0.125, 0.25, 0.0625) = 0.05386675$,
$\tilde{S}_{[0,0.25]} = S(0, 0.25, 0.0625) = S_1 + S_2$ and
$\tilde{R}_{[0,0.25]} \approx \frac{1}{15}\left|\tilde{S}_{[0,0.25]} - S_{[0,0.25]}\right| \approx 0.153792 \times 10^{-3} \nless 0.000125.$

Level 3:

i	1	2	3	4	5
x_i	0.0	1/32	1/16	3/32	1/8

$h = 0.0625$, $T = 0.0000625$ (stored for $[0.125, 0.25]$),
$S_{[0,0.125]} = S(0, 0.125, 0.0625) = 0.02819903$,
$S_1 = S(0, 0.0625, 0.03125) = 0.00996986$, $S_2 = S(0.0625, 0.125, 0.03125) = 0.01904477$,
$\tilde{S}_{[0,0.125]} = S(0, 0.125, 0.03125) = S_1 + S_2$ and
$\tilde{R}_{[0,0.125]} \approx \frac{1}{15}\left|\tilde{S}_{[0,0.125]} - S_{[0,0.125]}\right| \approx 0.5437 \times 10^{-4} < 0.0000625.$

So, we accept $\tilde{S}_{[0,0.125]}$ as an approximation of $\int_0^{1/8} \sqrt{x}\,\mathrm{d}x$; namely,
$\int_0^{1/8} \sqrt{x}\,\mathrm{d}x \approx \tilde{S}_{[0,0.125]} = 0.02901464.$

Level 3:

i	1	2	3	4	5
x_i	1/8	5/32	3/16	7/32	1/4

$h = 0.0625$, $T = 0.0000625$ (retrieved from $[0, 0.125]$),
$S_{[0.125,0.25]} = S(0.125, 0.25, 0.0625) = 0.05386675$,
$S_1 = S(0.125, 0.1875, 0.03125) = 0.02466359$, $S_2 = S(0.1875, 0.25, 0.03125) = 0.02920668$,
$\tilde{S}_{[0.125,0.25]} = S(0.125, 0.25, 0.03125) = S_1 + S_2$ and
$\tilde{R}_{[0.125,0.25]} \approx \frac{1}{15}\left|\tilde{S}_{[0.125,0.25]} - S_{[0.125,0.25]}\right| \approx 0.2347 \times 10^{-6} < 0.0000625.$

So, we accept $\tilde{S}_{[0.125,0.25]}$ as an approximation of $\int_{1/8}^{1/4} \sqrt{x}\,\mathrm{d}x$; namely,
$\int_{1/8}^{1/4} \sqrt{x}\,\mathrm{d}x \approx \tilde{S}_{[0.125,0.25]} = 0.05387027.$

Hence, $\int_0^{1/4} \sqrt{x}\,\mathrm{d}x = \int_0^{1/8} \sqrt{x}\,\mathrm{d}x + \int_{1/8}^{1/4} \sqrt{x}\,\mathrm{d}x \approx 0.08288491.$

Level 2:

i	1	2	3	4	5
x_i	1/4	5/16	3/8	7/16	1/2

$h = 0.125$, $T = 0.000125$ (retrieved from $[0, 0.25]$),
$S_{[0.25,0.5]} = S(0.25, 0.5, 0.125) = 0.15235819$,

$S_1 = S(0.25, 0.375, 0.06125) = 0.06975918,\ S_2 = S(0.375, 0.5, 0.0615) = 0.08260897,$
$\tilde{S}_{[0.25,0.5]} = S(0.25, 0.5, 0.06125) = S_1 + S_2$ and
$\tilde{R}_{[0.25,0.5]} \approx \dfrac{1}{15}\left|\tilde{S}_{[0.25,0.5]} - S_{[0.25,0.5]}\right| \approx 0.664 \times 10^{-6} < 0.000125.$

So, we accept $\tilde{S}_{[0.25,0.5]}$ as an approximation of $\displaystyle\int_{1/4}^{1/2} \sqrt{x}\,\mathrm{d}x$; namely,

$\displaystyle\int_{1/4}^{1/2} \sqrt{x}\,\mathrm{d}x \approx \tilde{S}_{[0.25,0.5]} = 0.15236815.$

Hence, $\displaystyle\int_0^{1/2} \sqrt{x}\,\mathrm{d}x = \int_0^{1/4} \sqrt{x}\,\mathrm{d}x + \int_0^{1/4} \sqrt{x}\,\mathrm{d}x \approx= 0.23525305.$

Level 1:

i	1	2	3	4	5
x_i	1/2	5/8	3/4	7/8	1

$h = 0.25$, $T = 0.00025$ (retrieved from $[0, 0.5]$), $S_{[0.5,1]} = S(0.5, 1, 0.25) = 0.43093403,$
$S_1 = S(0.5, 0.75, 0.125) = 0.19730874,\ S_2 = S(0.75, 1, 0.125) = 0.23365345,$
$\tilde{S}_{[0.5,1]} = S(0.5, 1, 0.125) = S_1 + S_2$ and
$\tilde{R}_{[0.5,1]} \approx \dfrac{1}{15}\left|\tilde{S}_{[0.5,1]} - S_{[0.5,1]}\right| \approx 0.18773 \times 10^{-5} < 0.00025.$

So, we accept $\tilde{S}_{[0.5,1]}$ as an approximation of $\displaystyle\int_{1/2}^{1} \sqrt{x}\,\mathrm{d}x$; namely,

$\displaystyle\int_{1/2}^{1} \sqrt{x}\,\mathrm{d}x \approx \tilde{S}_{[0.5,1]} = 0.43096219.$

Hence, $\displaystyle\int_0^{1} \sqrt{x}\,\mathrm{d}x = \int_0^{1/2} \sqrt{x}\,\mathrm{d}x + \int_{1/2}^{1} \sqrt{x}\,\mathrm{d}x = 0.66621525.$

Level 0: We have found that

$$\int_0^1 \sqrt{x}\,\mathrm{d}x \approx 0.66621525\ .$$

The exact answer is $2/3 = 0.\overline{6}$. ♣

12.7 Gaussian Quadrature

Let $f :]a, b[\to \mathbb{R}$ be an integrable function on $]a, b[$. f may not be a nice function to integrate. For instance, f may not be bounded at the endpoints, thus $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ is in improper integral.

In this section, we assume that we can write f as the product of two functions g and w, where g is a nice function on $]a, b[$ and w is a function on $]a, b[$ taking only non-negative values (and almost everywhere non-null).

Definition 12.7.1

A **Gaussian quadrature** is a formula of the form

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b g(x)w(x)\,\mathrm{d}x \approx \sum_{i=1}^n c_i g(x_i) \ , \tag{12.7.1}$$

where we choose the **nodes** $x_1, x_2, \ldots, x_n$ in $[a,b]$ and the **weights** $c_1, c_2, \ldots, c_n$ such that the formula is exact for all polynomials g of degree lest than a given constant k (usually $k = 2n$).

It is easy to find a Gaussian quadrature that is exact for polynomials of degree less than n. Given any nodes $a \leq x_1 < x_2 < \ldots < x_n \leq b$, let

$$\ell_i(x) = \prod_{\substack{j=1\\ j\neq i}}^n \frac{x - x_j}{x_i - x_j}$$

and

$$c_i = \int_a^b \ell_i(x)w(x)\,\mathrm{d}x \tag{12.7.2}$$

for $1 \leq i \leq n$. Since any polynomial p of degree less than n can be written as

$$p(x) = \sum_{i=1}^n p(x_i)\ell_i(x) \ ,$$

we have

$$\int_a^b p(x)w(x)\,\mathrm{d}x = \sum_{i=1}^n p(x_i)\left(\int_a^b \ell_i(x)w(x)\,\mathrm{d}x\right) = \sum_{i=1}^n c_i p(x_i) \ .$$

We would like to do better than that.

Example 12.7.2

Find x_1 and c_1 such that

$$\int_a^b f(x)\,\mathrm{d}x \approx c_1 f(x_1)$$

is exact for polynomial of degree less than 2. The node x_1 and the weight c_1 must satisfy $\displaystyle\int_a^b 1\,\mathrm{d}x = c_1$ and $\displaystyle\int_a^b x\,\mathrm{d}x = c_1 x_1$; namely, $a + b = c_1$ and $\dfrac{b^2}{2} - \dfrac{a^2}{2} = c_1 x_1$. The values of c_1 and x_1 satisfying these two equations are $c_1 = b - a$ and $x_1 = (b+a)/2$. The quadrature formula is therefore

$$\int_a^b f(x)\,\mathrm{d}x \approx (b-a) f(\frac{b+a}{2}) \ .$$

This is the Midpoint rule. The truncation error for $f(x) = x^2$ is

$$\int_a^b x^2\,\mathrm{d}x - (b-a)\left(\frac{b+a}{2}\right)^2 = \frac{b^3 - a^3}{3} - \frac{(b-a)(b+a)^2}{4} = -\frac{(b-a)^3}{12} \ .$$

This is $-f''(\xi)(b-a)^3/24$. ♣

Example 12.7.3
Find x_1, x_2, c_1 and c_2 such that

$$\int_a^b f(x)\,\mathrm{d}x \approx c_1 f(x_1) + c_2 f(x_2)$$

is exact for polynomial of degree less than 4. The nodes x_1, x_2 and the weights c_1, c_2 must satisfy $\int_a^b 1\,\mathrm{d}x = c_1 + c_2$, $\int_a^b x\,\mathrm{d}x = c_1 x_1 + c_2 x_2$, $\int_a^b x^2\,\mathrm{d}x = c_1 x_1^2 + c_2 x_2^2$ and $\int_a^b x^3\,\mathrm{d}x = c_1 x_1^3 + c_2 x_2^3$; Namely, $b - a = c_1 + c_2$, $\dfrac{b^2 - a^2}{2} = c_1 x_1 + c_2 x_2$, $\dfrac{b^3 - a^3}{3} = c_1 x_1^2 + c_2 x_2^2$ and $\dfrac{b^4 - a^4}{4} = c_1 x_1^3 + c_2 x_2^3$. The values of c_1, c_2, x_1 and x_2 satisfying these four nonlinear equations are $c_1 = c_2 = \dfrac{b-a}{2}$, $x_1 = z$ and $x_2 = a + b - z$, where z is the positive root of $6z^2 - 6(a+b)z + a^2 + b^2 + 4ab$.

If $a = -1$ and $b = 1$, we get $c_1 = c_2 = 1$ and $x_1 = -x_2 = 1/\sqrt{3}$. Hence,

$$\int_{-1}^1 f(x)\,\mathrm{d}x \approx f\left(\frac{1}{\sqrt{3}}\right) + f\left(\frac{-1}{\sqrt{3}}\right) .$$

We will see later that this is the Gauss-Legendre quadrature formula for $n = 2$. ♣

We now show in general how to choose the nodes x_1, x_2, ..., x_n and the weights c_1, c_2, ..., c_n such that (12.7.1) is exact for polynomials of degree less than $2n$.

Remark 12.7.4
Using the polynomial $p(x) = \prod_{i=1}^n (x - x_i)^2$, we ask the reader in Question 12.40 to show that $2n$ is the largest value k such that (12.7.1) is exact for polynomials of degree less than k. ♠

Theorem 12.7.5

Let $\{P_0, P_1, P_2, \ldots\}$ be an orthogonal set of polynomials on $[a, b]$ with respect to a weight function w. Suppose that P_n is of degree exactly n. If p is a polynomial of degree less than $2n$, then

$$\int_a^b p(x) w(x)\,\mathrm{d}x = \sum_{j=1}^n c_j\, p(x_j) ,$$

where

$$c_j = \int_a^b \left(\prod_{\substack{i=1\\ i\neq j}}^n \frac{x - x_i}{x_j - x_i} \right) w(x)\,\mathrm{d}x$$

and x_1, x_2, ..., x_n are the roots of the polynomial P_n of degree n.

Proof.
A) If the degree of p is less than n.

Using the Lagrange's form of the interpolating polynomial of p at the roots x_1, x_2, …, x_n of P_n, formula (6.1.1), we have

$$p(x) = \sum_{j=1}^{n} \left(\prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{x - x_i}{x_j - x_i} \right) p(x_j) \ .$$

Recall that there is a unique polynomial of degree less than n that interpolates p at the points x_1, x_2, …, x_n. It must therefore be p itself. Hence

$$\int_a^b p(x)w(x)\,\mathrm{d}x = \sum_{i=j}^{n} p(x_j) \int_a^b \left(\prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{x - x_i}{x_j - x_i} \right) w(x)\,\mathrm{d}x = \sum_{j=1}^{n} c_j\, p(x_j) \ .$$

B) If the degree of p is greater or equal to n but less than $2n$.

If we divide p by P_n, we get $p = qP_n + r$, where q and r are polynomials of degree less than n.

From the first conclusion of Theorem 8.2.3, we have that $q = \sum_{i=0}^{n-1} \alpha_i P_i$ for some constants α_0, α_1, α_2, …, α_{n-1}. Hence,

$$\begin{aligned}
\int_a^b p(x)w(x)\,\mathrm{d}x &= \int_a^b q(x)P_n(x)w(x)\,\mathrm{d}x + \int_a^b r(x)w(x)\,\mathrm{d}x \\
&= \sum_{i=0}^{n-1} \alpha_i \int_a^b P_i(x)P_n(x)w(x)\,\mathrm{d}x + \int_a^b r(x)w(x)\,\mathrm{d}x \\
&= \int_a^b r(x)w(x)\,\mathrm{d}x = \sum_{i=1}^{n} c_i\, r(x_i) \ , \qquad (12.7.3)
\end{aligned}$$

where x_1, x_2, …, x_n are the roots of P_n. The third equality comes from the orthogonality property of the polynomials P_i. The last equality comes from the previous case for the polynomials of degree less than n.

Finally, since x_1, x_2, …, x_n are the roots of P_n, then $p(x_i) = q(x_i)P_n(x_i) + r(x_i) = r(x_i)$ for $1 \leq i \leq n$, and the conclusion of the theorem follows from (12.7.3). ∎

Remark 12.7.6
If we substitute $g(x) = 1$ in a Gaussian quadrature formula

$$\int_a^b g(x)w(x)\,\mathrm{d}x \approx \sum_{j=1}^{n} c_j g(x_j)$$

which is exact for polynomials of degree less than $2n$, we find that $\int_a^b w(x)\,\mathrm{d}x = \sum_{j=1}^{n} c_j$. ♠

12.7.1 Gauss-Legendre quadrature

If we use the Legendre polynomials and $w(x) = 1$ for $-1 \leq x \leq 1$ in Theorem 12.7.5, we get the **Gauss-Legendre quadrature**. The integral $\int_{-1}^{1} f(x)\,\mathrm{d}x$ is approximately equal to $\sum_{j=1}^{n} c_j f(x_j)$, where the x_j's are the roots of the Legendre polynomial P_n and the c_j's are given by Theorem 12.7.5 with $a = -1$, $b = 1$ and $w(x) = 1$ for $-1 \leq x \leq 1$. The values of c_j and x_j for $n = 2$, 3, 4 and 5 are given in the following table.

n	roots x_j of $P_n(x)$	coefficients c_j
2	0.5773502692	1.0
	−0.5773502692	1.0
3	0.7745966692	0.5555555556
	0.0	0.8888888889
	−0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	−0.3399810436	0.6521451549
	−0.8611363116	0.3478548451
5	−0.9061798459	0.2369268851
	−0.5384693101	0.4786286705
	0.0	0.5688888889
	0.5384693101	0.4786286705
	0.9061798459	0.2369268851

Example 12.7.7
Use Gauss-Legendre quadrature with $n = 3$ to approximate

$$\int_1^3 \frac{\sin^2(x)}{x}\,\mathrm{d}x \ .$$

Using the change of variable $t = x - 2$, we get

$$\begin{aligned}\int_1^3 \frac{\sin^2(x)}{x}\,\mathrm{d}x = \int_{-1}^1 \frac{\sin^2(t+2)}{t+2}\,\mathrm{d}t &\approx \sum_{i=1}^{3} c_i \frac{\sin^2(x_i+2)}{x_i+2} \\ &\approx 0.5555555556\,\frac{\sin^2(2+0.7745966692)}{2+0.7745966692} + 0.8888888889\,\frac{\sin^2(2)}{2} \\ &\quad + 0.5555555556\,\frac{\sin^2(2-0.7745966692)}{2-0.7745966692} \approx 0.79465267\ .\end{aligned}$$

♣

Remark 12.7.8
In general, to transform an integral of the form $\int_a^b f(x)\,\mathrm{d}x$ into an integral of the form $\int_{-1}^1 g(t)\,\mathrm{d}t$, one uses the substitution $t = \dfrac{x-(a+b)/2}{(b-a)/2}$, where $(a+b)/2$ is the middle of the interval $[a,b]$ and $(b-a)/2$ is half the length of $[a,b]$. ♠

12.7.2 Gauss-Chebyshev quadrature

If we use the Chebyshev polynomials and $w(x) = 1/\sqrt{1-x^2}$ for $-1 < x < 1$ in Theorem 12.7.5, we get the **Gauss-Chebyshev quadrature**. The integral $\displaystyle\int_{-1}^{1} \frac{g(x)}{\sqrt{1-x^2}}\,dx$ is approximately equal to $\displaystyle\sum_{j=1}^{n} c_j g(x_j)$, where the x_j's are the roots of the Chebyshev polynomial T_n and the c_j's are given by Theorem 12.7.5 with $a = -1$, $b = 1$ and $w(x) = 1/\sqrt{1-x^2}$ for $-1 < x < 1$. So, $x_j = \cos((2j-1)\pi/(2n))$ for $1 \leq j \leq n$, and one can prove that $c_j = \pi/n$ for all j.

Example 12.7.9
Use Gauss-Chebyshev quadrature to approximate

$$\int_{-1}^{1} \frac{x^2}{\sqrt{1-x^2}}\,dx \ .$$

We use Gauss-Chebyshev quadrature with $n = 2$. We have that

$$\int_{-1}^{1} \frac{x^2}{\sqrt{1-x^2}}\,dx = c_1 x_1^2 + c_2 x_2^2 = \frac{\pi}{2}(\cos(\pi/4))^2 + \frac{\pi}{2}(\cos(3\pi/4))^2 = 1.57079632679\ldots$$

because Gauss-Chebyshev quadrature with $n = 2$ is exact for polynomial of degree less than $2n = 4$. In general, $\displaystyle\int_{-1}^{1} \frac{p(x)}{\sqrt{1-x^2}}\,dx = c_1 p(x_1) + c_2 p(x_2)$ for any polynomial $p(x)$ of degree less than 4. ♣

12.7.3 Convergence and accuracy

Theorem 12.7.10

Let $g : [a,b] \to \mathbb{R}$ be a continuous function. Suppose that, for each positive integer n,

$$\int_a^b g(x)w(x)\,dx \approx \sum_{j=1}^{n} c_j g(x_j)$$

is a Gaussian quadrature formula which is exact for polynomials of degree less than $2n$ (as given in Theorem 12.7.5 for instance). Then,

$$\sum_{j=1}^{n} c_j g(x_j) \to \int_a^b g(x)w(x)\,dx \quad \text{as} \quad n \to \infty \ .$$

Proof.
Given $\epsilon > 0$, Stone-Weierstrass Theorem, Theorem 9.1.1, gives a polynomial p such that

$$\max_{a\leq x\leq b} |g(x) - p(x)| < \frac{\epsilon}{2\displaystyle\int_a^b w(x)\,dx} \ .$$

Hence, since the Gaussian quadrature formula is exact for polynomials of degree less than $2n$, we have for $2n$ greater than the degree of p that

$$\begin{aligned}
&\left| \int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j g(x_j) \right| \\
&\quad = \left| \int_a^b g(x)w(x)\,\mathrm{d}x - \int_a^b p(x)w(x)\,\mathrm{d}x + \sum_{j=1}^n c_j p(x_j) - \sum_{j=1}^n c_j g(x_j) \right| \\
&\quad \leq \int_a^b |g(x) - p(x)|\, w(x)\,\mathrm{d}x + \sum_{j=1}^n c_j \,|p(x_j) - g(x_j)| \\
&\quad \leq \frac{\epsilon}{2\displaystyle\int_a^b w(x)\,\mathrm{d}x} \left(\int_a^b w(x)\,\mathrm{d}x + \sum_{j=1}^n c_j \right) = \epsilon \ .
\end{aligned}$$

The last equality comes from Remark 12.7.6. ∎

Theorem 12.7.11

Let $g : [a,b] \to \mathbb{R}$ be a twice continuously differentiable function and suppose that the hypotheses of Theorem 12.7.5 are satisfied. Then,

$$\begin{aligned}
&\int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j\, g(x_j) \\
&\quad = \int_a^b g[x_1, x_2, \ldots, x_n, x_1, x_2, \ldots, x_n, x] \prod_{i=1}^n (x - x_i)^2 w(x)\,\mathrm{d}x \ .
\end{aligned}$$

Moreover, if g is continuously differentiable of order $2n$,

$$\int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j\, g(x_j) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b \prod_{i=1}^n (x - x_i)^2 w(x)\,\mathrm{d}x$$

for some $\xi \in [a,b]$.

Proof.

The interpolating polynomial p of $g(x)$ at the points x_1, x_2, $\ldots$, x_n, x_1, x_2, $\ldots$, x_n satisfies

$$g(x) = p(x) + g[x_1, x_2, \ldots, x_n, x_1, x_2, \ldots, x_n, x] \prod_{i=1}^n (x - x_i)^2 \ .$$

The polynomial p is of degree less than $2n$. The divided difference $g[x_1, x_2, \ldots, x_n, x_1, x_2, \ldots, x_n, x]$ is well defined because $g(x)$ is twice continuously differentiable. The nodes x_j come in pairs and the only cases where we can have three equal nodes are when $x = x_j$ for some j.

Since p is of degree less than $2n$,

$$\int_a^b p(x)w(x)\,\mathrm{d}x = \sum_{j=1}^n c_j\, p(x_j) \ .$$

Hence,

$$\begin{aligned}\int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j\, g(x_j) &= \int_a^b p(x)w(x)\,\mathrm{d}x \\ &\quad + \int_a^b g[x_1,x_2,\ldots,x_n,x_1,x_2,\ldots,x_n,x]\prod_{i=1}^n (x-x_i)^2 w(x)\,\mathrm{d}x \\ &\quad - \sum_{j=1}^n c_j\, p(x_j) - \sum_{j=1}^n c_j\, g[x_1,x_2,\ldots,x_n,x_1,x_2,\ldots,x_n,x_j]\prod_{i=1}^n (x_j-x_i)^2 \\ &= \int_a^b g[x_1,x_2,\ldots,x_n,x_1,x_2,\ldots,x_n,x]\prod_{i=1}^n (x-x_i)^2 w(x)\,\mathrm{d}x\ .\end{aligned}$$

The last sum of the first equality is zero because the divided differences $g[x_1,x_2,\ldots,x_n,x_1,x_2,\ldots,x_n,x_j]$ are well defined and $\prod_{i=1}^n (x_j-x_i)^2 = 0$ for all j.

Since $\prod_{i=1}^n (x-x_i)^2 w(x) \geq 0$ for $x \in [a,b]$, we get from the Mean Value Theorem for Integrals that

$$\int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j\, g(x_j) = g[x_1,x_2,\ldots,x_n,x_1,x_2,\ldots,x_n,\nu]\int_a^b \prod_{i=1}^n (x-x_i)^2 w(x)\,\mathrm{d}x$$

for some $\nu \in [a,b]$. If $g(x)$ is $2n$ continuously differentiable, Theorem 6.2.5 gives

$$\int_a^b g(x)w(x)\,\mathrm{d}x - \sum_{j=1}^n c_j\, g(x_j) = \frac{g^{(2n)}(\xi)}{(2n)!}\int_a^b \prod_{i=1}^n (x-x_i)^2 w(x)\,\mathrm{d}x$$

for some $\xi \in [a,b]$ ∎

12.8 Bernoulli Polynomials

One of the major results of this section is Theorem 12.8.5. It proves that Richardson extrapolation can be applied to the composite trapezoidal rule to get Romberg integration.

Definition 12.8.1

The polynomials $B_n(x)$ defined recursively by the series

$$\sum_{j=0}^n \binom{n+1}{j} B_j(x) = (n+1)x^n \tag{12.8.1}$$

for $n = 0,\ 1,\ 2,\ \ldots$ are the **Bernoulli polynomials**.

Remark 12.8.2

The first four Bernoulli polynomials are $B_0(x) = 1$, $B_1(x) = x - \frac{1}{2}$, $B_2(x) = x^2 - x + \frac{1}{6}$ and $B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x$. ♠

Proposition 12.8.3

The Bernoulli polynomials satisfy the following properties.

1. $B_n'(x) = nB_{n-1}(x)$ for $n \geq 1$.
2. $B_n(x+1) - B_n(x) = nx^{n-1}$ for $n \geq 1$.
3. $B_n(x) = \displaystyle\sum_{j=0}^{n} \binom{n}{j} B_j(0)x^{n-j}$ for $n \geq 0$.
4. $B_n(1-x) = (-1)^n B_n(x)$ for $n \geq 0$.

Proof.
1) We prove (1) by induction. The case $n = 1$ is a consequence of $B_0(x) = 1$ and $B_1(x) = x - \dfrac{1}{2}$. We assume that $B_j'(x) = jB_{j-1}(x)$ for $1 \leq j < n$. The derivative on both side of (12.8.1) yields

$$\sum_{j=1}^{n} \binom{n+1}{j} B_j'(x) = (n+1)nx^{n-1} = (n+1)\sum_{j=0}^{n-1}\binom{n}{j}B_j(x) \ , \tag{12.8.2}$$

where the second equality comes from (12.8.1) with n replaced by $n-1$.

From the hypothesis of induction, we get

$$\begin{aligned}\sum_{j=1}^{n} \binom{n+1}{j} B_j'(x) &= \binom{n+1}{n} B_n'(x) + \sum_{j=1}^{n-1}\binom{n+1}{j} jB_{j-1}(x) \\ &= (n+1)B_n'(x) + (n+1)\sum_{j=1}^{n-1}\binom{n}{j-1}B_{j-1}(x) \\ &= (n+1)B_n'(x) + (n+1)\sum_{j=0}^{n-2}\binom{n}{j}B_j(x) \ .\end{aligned}$$

because

$$\binom{n+1}{j} j = \left(\frac{(n+1)!}{j!(n+1-j)!}\right) j = (n+1)\,\frac{n!}{(j-1)!(n-(j-1))!} = (n+1)\binom{n}{j-1} \ .$$

Hence, after dividing both sides of (12.8.2) by $(n+1)$, we get

$$B_n'(x) + \sum_{j=0}^{n-2}\binom{n}{j}B_j(x) = \sum_{j=0}^{n-1}\binom{n}{j}B_j(x) \ .$$

After cancelling the terms that are equal from both sides, we get $B_n'(x) = \dbinom{n}{n-1}B_{n-1}(x) = nB_{n-1}(x)$ which proved (1).
2) From (1), we have that

$$B_n^{(j)}(x) = n(n-1)(n-2)\ldots(n-j+1)B_{n-j}(x)$$

for $j > 0$ and $n > 0$. Hence,

$$\begin{aligned}
B_n(x+h) &= \sum_{j=0}^{n} \frac{1}{j!} B_n^{(j)}(x) h^j = B_n(x) + \sum_{j=1}^{n} \frac{n(n-1)(n-2)\ldots(n-(j-1))}{j!} B_{n-j}(x) h^j \\
&= B_n(x) + \sum_{j=1}^{n} \binom{n}{j} B_{n-j}(x) h^j = B_n(x) + \sum_{j=1}^{n} \binom{n}{n-j} B_{n-j}(x) h^j \\
&= B_n(x) + \sum_{j=0}^{n-1} \binom{n}{j} B_j(x) h^{n-j} \qquad (12.8.3)
\end{aligned}$$

With $h = 1$, we get

$$B_n(x+1) = B_n(x) + \sum_{j=0}^{n-1} \binom{n}{j} B_j(x) = B_n(x) + n x^{n-1} .$$

where the last equality comes from (12.8.1) with n replaced by $n-1$. This prove (2).

3) The case $n = 0$ can be verified directly. For $n > 0$, if we set $x = 0$ in (12.8.3), we get

$$B_n(h) == \sum_{j=0}^{n} \binom{n}{j} B_j(0) h^{n-j}$$

which is (3).

4) If we substitute x by $-x$ in (2), we get

$$B_n(1-x) - B_n(-x) = n(-x)^{n-1} = (-1)^{n-1}\left(n x^{n-1}\right) = (-1)^{n-1}\left(B_n(x+1) - B_n(x)\right)$$

for $n > 0$. Hence,

$$(-1)^n B_n(x+1) - B_n(-x) = (-1)^n B_n(x) - B_n(1-x) . \qquad (12.8.4)$$

If $F(x) = (-1)^n B_n(x) - B_n(1-x)$ for all x, then (12.8.4) shows that $F(x+1) = F(x)$ for all x. Thus, F is a periodic function of period 1. Since F is also a polynomial, we must have that $F(x) = C_n$, a constant, for all x.

Hence

$$0 = F'(x) = (-1)^n B_n'(x) + B_n'(1-x) = (-1)^n n B_{n-1}(x) + n B_{n-1}(1-x)$$

and (4) with n replaced by $n-1$ follows after a division by n. ∎

Lemma 12.8.4

Given any positive integer n, we have that $B_{2n}(x) - B_{2n}(0) \neq 0$ for all $x \in]0,1[$.

Proof.
Let $G_n(x) = B_{2n}(x) - B_{2n}(0)$ for all $x \in [0,1]$.

From (2) of Proposition 12.8.3 with $x = 0$, we get that $B_j(0) = B_j(1)$ for $j \geq 2$. Thus $G_n(0) = G_n(1) = 0$ for $n > 0$.

Moreover, from (2) and (4) of Proposition 12.8.3 with $x = 0$, we get $B_j(0) = B_j(1) = (-1)^j B_j(0)$ for all $j \geq 2$. Thus, $B_{2j-1}(0) = B_{2j-1}(1) = 0$ for $j = 2, 3, \ldots$

We may assume that $n > 1$. Since $G_1(x) = B_2(x) - B_2(0)$ is a polynomial of degree 2 that already vanishes at 0 and 1, it cannot have another zero.

Suppose that G_n with $n > 1$ vanishes at a point $\eta \in]0,1[$. We first prove by induction that this implies that B_{2k-1} has always two distinct zeros in $]0,1[$ for $k = n, n-1, \ldots, 2$.

Since $G_n(0) = G_n(1) = 0$, the Mean Value Theorem on $[0,\eta]$ and $[\eta,1]$ yields two distinct zeros, $\eta_1 \in]0,\eta[$ and $\eta_2 \in]\eta,1[$, of $G_n'(x) = B_{2n}'(x) = 2nB_{2n-1}(x)$ in $]0,1[$. Thus B_{2n-1} has two distinct zeros $\eta_1 < \eta_2$ in $]0,1[$. The induction hypothesis is thus true for $k = n$.

Suppose that B_{2k-1} for $2 < k \leq n$ has two distinct zeros $\eta_1 < \eta_2$ in $]0,1[$. Since B_{2k-1} vanishes also at 0 and 1, the Mean value Theorem on the intervals $[0,\eta_1]$, $[\eta_1,\eta_2]$ and $[\eta_2,1]$ yields three distinct zeros of $B_{2k-1}'(x) = (2k-1)B_{2k-2}(x)$ in $]0,1[$. Let $\eta_3 \in]0,\eta_1[$, $\eta_4 \in]\eta_1,\eta_2[$ and $\eta_5 \in]\eta_2,1[$ be these three distinct zeros. Thus B_{2n-2} has three distinct zeros $\eta_3 < \eta_4 < \eta_5$ in $]0,1[$. The Mean Value Theorem on the intervals $[\eta_3,\eta_4]$ and $[\eta_4,\eta_5]$ yields two distinct zeros, $\eta_6 \in]\eta_3,\eta_4[$ and $\eta_7 \in]\eta_4,\eta_5[$, of $B_{2k-2}'(x) = (2k-2)B_{2k-3}(x)$ in $]0,1[$. Thus B_{2k-3} has two distinct zeros $\eta_6 < \eta_7$ in $]0,1[$. The induction hypothesis is true for $k-1$ instead of k. This completes the proof by induction.

This shows that B_3 has four zeros: 0, 1 and the two distinct zeros in $]0,1[$. But this is impossible because B_3 is a non-trivial polynomial of degree 3. The assumption that G_n vanishes at a point $\eta \in]0,1[$ yields a contradiction. ∎

Theorem 12.8.5

If $f : [a,b] \to \mathbb{R}$ is a $2n$-continuously differentiable function,

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left(f(a)+f(b)\right) - \sum_{j=0}^{n-1} \frac{B_{2j}(0)}{(2j)!}\left(f^{(2j-1)}(b) - f^{(2j-1)}(a)\right)h^{2j} \\ - \frac{B_{2n}(0)}{(2n)!} f^{(2n)}(\eta)h^{2n+1}\end{aligned} \tag{12.8.5}$$

for some $\eta \in [a,b]$, where $h = b-a$.

Proof.
If we substitute $x = a + th$ in the integral, we get

$$\int_a^b f(x)\,\mathrm{d}x = h\int_0^1 f(a+th)\,\mathrm{d}t\ .$$

Let $g(t) = f(a+th)$. We will show that

$$\int_0^1 g(t)\,\mathrm{d}t = \frac{1}{2}\left(g(0)+g(1)\right) - \sum_{j=0}^{n-1}\frac{B_{2j}(0)}{(2j)!}\left(g^{(2j-1)}(1) - g^{(2j-1)}(0)\right) - \frac{B_{2n}(0)}{(2n)!}g^{(2n)}(\nu) \tag{12.8.6}$$

for some $\nu \in [0,1]$. This yields (12.8.5) if $g(t) = f(a+th)$ since

$$g^{(k)}(t) = \frac{\mathrm{d}^k}{\mathrm{d}t^k} g(t) = \frac{\mathrm{d}^k}{\mathrm{d}t^k} f(a+th) = f^{(k)}(a+th)\, h^k \ .$$

We first prove by induction that

$$\begin{split}\int_0^1 g(t)\,\mathrm{d}t = \frac{1}{2}\left(g(1)+g(0)\right) - \sum_{j=1}^{k} \frac{B_{2j}(0)}{(2j)!}\left(g^{(2j-1)}(1) - g^{(2j-1)}(0)\right)\\ + \frac{1}{(2k)!}\int_0^1 g^{(2k)}(t) B_{2k}(t)\,\mathrm{d}t\end{split} \tag{12.8.7}$$

for $k = 1, 2, \ldots, n$.

Using integration by parts, we have

$$\begin{split}\int_0^1 g(t)\,\mathrm{d}t &= \int_0^1 g(t)B_1'(t)\,\mathrm{d}t = \left(g(t)B_1(t)\right)\Big|_{t=0}^1 - \int_0^1 g'(t)B_1(t)\,\mathrm{d}t\\ &= \frac{1}{2}\left(g(1)+g(0)\right) - \int_0^1 g'(t)B_1(t)\,\mathrm{d}t\end{split}$$

because $B_1(1) = -B_1(0) = 1/2$. Another integration by parts and (1) of Proposition 12.8.3 yield

$$\begin{split}\int_0^1 g'(t)B_1(t)\,\mathrm{d}t &= \frac{1}{2}\int_0^1 g'(t)B_2'(t)\,\mathrm{d}t = \frac{1}{2}\left(g'(t)B_2(t)\right)\Big|_{t=0}^1 - \frac{1}{2}\int_0^1 g''(t)B_2(t)\,\mathrm{d}t\\ &= \frac{B_2(0)}{2}\left(g'(1)-g'(0)\right) - \frac{1}{2}\int_0^1 g''(t)B_2(t)\,\mathrm{d}t\end{split}$$

because $B_2(0) = B_2(1)$ as can be seen from (2) of Proposition 12.8.3 with $x = 0$. Hence

$$\int_0^1 g(t)\,\mathrm{d}t = \frac{1}{2}\left(g(1)+g(0)\right) - \frac{B_2(0)}{2}\left(g'(1)-g'(0)\right) + \frac{1}{2}\int_0^1 g''(t)B_2(t)\,\mathrm{d}t \ .$$

This prove (12.8.7) for $k = 1$.

Suppose that (12.8.7) is true for k. From (2) and (4) of Proposition 12.8.3 with $x = 0$, we find that $B_{2j+1}(0) = B_{2j+1}(1) = 0$ for all $j > 0$. Hence,

$$\begin{split}\int_0^1 g^{(2k)}(t)B_{2k}(t)\,\mathrm{d}t &= \frac{1}{2k+1}\int_0^1 g^{(2k)}(t)B_{2k+1}'(t)\,\mathrm{d}t\\ &= \frac{1}{2k+1}\left(g^{(2k)}(t)B_{2k+1}(t)\right)\Big|_{t=0}^1 - \frac{1}{2k+1}\int_0^1 g^{(2k+1)}(t)B_{2k+1}(t)\,\mathrm{d}t\\ &= -\frac{1}{2k+1}\int_0^1 g^{(2k+1)}(t)B_{2k+1}(t)\,\mathrm{d}t \ .\end{split} \tag{12.8.8}$$

Moreover,

$$\int_0^1 g^{(2k+1)}(t)B_{2k+1}(t)\,\mathrm{d}t = \frac{1}{2k+2}\int_0^1 g^{(2k+1)}(t)B_{2k+2}'(t)\,\mathrm{d}t$$

$$
\begin{aligned}
&= \frac{1}{2k+2}\left(g^{(2k+1)}(t)B_{2k+2}(t)\right)\Big|_{t=0}^{1} - \frac{1}{2k+2}\int_0^1 g^{(2k+2)}(t)B_{2k+2}(t)\,\mathrm{d}t \\
&= \frac{B_{2k+2}(0)}{2k+2}\left(g^{(2k+1)}(1) - g^{(2k+1)}(0)\right) - \frac{1}{2k+2}\int_0^1 g^{(2k+2)}(t)B_{2k+2}(t)\,\mathrm{d}t
\end{aligned} \tag{12.8.9}
$$

because $B_{2j}(0) = B_{2j}(1)$ for all $j > 0$ as can been seen from (2) of Proposition 12.8.3 with $x = 0$. Hence (12.8.8) and (12.8.9) imply that

$$
\begin{aligned}
\int_0^1 g^{(2k)}(t)B_{2k}(t)\,\mathrm{d}t = &-\frac{B_{2k+2}(0)}{(2k+1)(2k+2)}\left(g^{(2k+1)}(1) - g^{(2k+1)}(0)\right) \\
&+ \frac{1}{(2k+1)(2k+2)}\int_0^1 g^{(2k+2)}(t)B_{2k+2}(t)\,\mathrm{d}t\ .
\end{aligned}
$$

If we substitute this expression in (12.8.7), we get (12.8.7) for k replaced by $k+1$. This completes the proof by induction.

(12.8.7) for $k = n$ gives

$$
\begin{aligned}
\int_0^1 g(t)\,\mathrm{d}t = \frac{1}{2}\left(g(1) + g(0)\right) &- \sum_{j=1}^{n} \frac{B_{2j}(0)}{(2j)!}\left(g^{(2j-1)}(1) - g^{(2j-1)}(0)\right) \\
&+ \frac{1}{(2n)!}\int_0^1 g^{(2n)}(t)B_{2n}(t)\,\mathrm{d}t\ .
\end{aligned} \tag{12.8.10}
$$

Since the last term of the series in (12.8.10) is

$$
\frac{B_{2n}(0)}{(2n)!}\left(g^{(2n-1)}(1) - g^{(2n-1)}(0)\right) = \frac{B_{2n}(0)}{(2n)!}\int_0^1 g^{(2n)}(t)\,\mathrm{d}t\ ,
$$

we can rewrite (12.8.10) as

$$
\begin{aligned}
\int_0^1 g(t)\,\mathrm{d}t = \frac{1}{2}\left(g(1) + g(0)\right) &- \sum_{j=1}^{n-1} \frac{B_{2j}(0)}{(2j)!}\left(g^{(2j-1)}(1) - g^{(2j-1)}(0)\right) \\
&+ \frac{1}{(2n)!}\int_0^1 g^{(2n)}(t)\left(B_{2n}(t) - B_{2n}(0)\right)\,\mathrm{d}t\ .
\end{aligned} \tag{12.8.11}
$$

From the previous lemma, $B_{2n}(t) - B_{2n}(0)$ does not change sign on the interval $[0,1]$. Hence, from the Mean Value Theorem for Integrals, we may write

$$
\frac{1}{(2n)!}\int_0^1 g^{(2n)}(t)\left(B_{2n}(t) - B_n(0)\right)\,\mathrm{d}t = \frac{g^{(2n)}(\nu)}{(2n)!}\int_0^1 \left(B_{2n}(t) - B_n(0)\right)\,\mathrm{d}t
$$

for some $\nu \in [0,1]$. Moreover,

$$
\int_0^1 B_{2n}(t)\,\mathrm{d}t = \frac{1}{2n+1}\int_0^1 B'_{2n+1}(t)\,\mathrm{d}t = \frac{1}{2n+1}\left(B_{2n+1}(1) - B_{2n+1}(0)\right) = 0
$$

because $B_{2n+1}(0) = B_{2n+1}(1) = 0$. Thus,

$$
\frac{1}{(2n)!}\int_0^1 g^{(2n)}(t)\left(B_{2n}(t) - B_n(0)\right)\,\mathrm{d}t = -\frac{g^{(2n)}(\nu)}{(2n)!}B_n(0)\ ,
$$

and substituting this expression in (12.8.11) gives (12.8.6). ∎

12.9 Exercises

Question 12.1
Using polynomial interpolation, derive the following formula with its truncation error.

$$f'(x) \approx \frac{1}{2h}\left(-3f(x) + 4f(x+h) - f(x+2h)\right) \ . \tag{12.9.1}$$

Question 12.2
Using polynomial interpolation, derive the following formula with its truncation error.

$$f''(x) \approx \frac{1}{h^2}\left(f(x) - 2f(x+h) + f(x+2h)\right) \ . \tag{12.9.2}$$

Question 12.3
Develop a method similar to the Richardson extrapolation method given in Section 12.2 if $L_h(f)$ is an approximation of $L(f)$ with

$$L(f) = L_h(f) + \sum_{j=1}^{\infty} a_j h^{2j-1} \ . \tag{12.9.3}$$

Question 12.4
Develop a method similar to the Richardson extrapolation method given in Section 12.2 if $L_h(f)$ is an approximation of $L(f)$ with

$$L(f) = L_h(f) + \sum_{j=1}^{\infty} a_j h^{3j} \ . \tag{12.9.4}$$

Question 12.5
Use Richardson extrapolation with the centrale difference formula to approximate the derivative of $f(x) = \sin(\ln(x))$ at $x = 3$ with an accuracy of 10^{-7}. Start with $h = 0.8$.

Question 12.6
Use the composite midpoint rule to approximate

$$\int_0^{\pi/2} \sin(x)\,\mathrm{d}x$$

with an accuracy of 10^{-5}. You have to find a number of subintervals of $[1,3]$ (and so a step size h) such that the local truncation error is smaller than 10^{-5}.

Question 12.7
Use the composite midpoint rule to approximate

$$\int_1^3 \left(x\ln(x) + \frac{x^3}{24} - 5x^2\right)\mathrm{d}x$$

with an accuracy of 10^{-5}. You have to find a number of subintervals of $[1, 3]$ (and so a step size h) such that the local truncation error is smaller than 10^{-5}.

Question 12.8
Use the composite Simpson rule to find an approximation of the integral

$$\int_2^4 (x+1)^{1/3}\,\mathrm{d}x$$

with an accuracy of 10^{-5}. You have to find a number of subintervals of $[2, 4]$ (and so a step size h) such that the local truncation error is smaller than 10^{-5}.

Question 12.9
For each of the integration methods below, determine the theoretical value of n (and h) that will give an approximation of

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x$$

to within 10^{-5} and compute the approximation with this value of n.
a) The composite midpoint rule.
b) The composite trapezoidal rule.
c) The composite Simpson's rule.

Compare with the exact answer.

Question 12.10
Show that Simpson's rule is exact for polynomials of degree up to 3 but not (generally) exact for degrees higher than 3.

Question 12.11
Use Romberg integration to approximate the integral

$$\int_2^4 (x+1)^{1/3}\,\mathrm{d}x$$

with an accuracy of 10^{-5}. Start with two subdivisions of the interval $[2, 4]$.

Question 12.12
Use Romberg integration to approximate the integral

$$\int_3^5 (x-2)^{1/4}\,\mathrm{d}x$$

with an accuracy of 10^{-5}. Start with one subdivision of the interval $[3, 5]$.

Question 12.13
Use Romberg integration to approximate the integral

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x\ .$$

Stop when the difference between two successive iterations on the diagonal line is smaller than 10^{-5}.

Question 12.14
Show that the formula used to generate the second column of the table associated to Romberg integration is the Simpson's rule.

Question 12.15
Use an adaptive method based on the composite Simpson rule to approximate the integral

$$\int_3^5 (x-2)^{1/4}\,\mathrm{d}x$$

with an accuracy of 10^{-5}. Start with one subdivision of the interval $[3,5]$.

Question 12.16
Show that the following formula is exact for all polynomials of degree less than or equal to 4.

$$\int_0^1 f(x)\,\mathrm{d}x = \frac{1}{90}\left(7\,f(0) + 32\,f(1/4) + 12\,f(1/2) + 32\,f(3/4) + 7\,f(1)\right) \ . \tag{12.9.5}$$

Use this formula to deduce a formula for the integral $\displaystyle\int_a^b f(x)\,\mathrm{d}x$, where a and b are any real numbers.

Question 12.17
Find, if possible, an integration formula of the form

$$\int_0^1 f(x)\,\mathrm{d}x \approx A\,\left(f(x_0) + f(x_1)\right)$$

that is exact for polynomials of degree less or equal to 2.

Question 12.18
Find the values of A, B and C such that

$$\int_0^2 x\,f(x)\,\mathrm{d}x \approx A\,f(0) + B\,f(1) + C\,f(2)$$

is exact for polynomials of degree as high as possible.

Question 12.19
Find a formula of the form

$$\int_0^{2\pi} f(x)\,\mathrm{d}x = A\,f(0) + B\,f(\pi)$$

that is exact for $f(x) = \cos(kx)$ with $k = 0$ and $k = 1$. Show that it is exact for any function of the form

$$f(x) = \sum_{k=0}^{n}\left(a_k\,\cos((2k+1)x) + b_k\sin(k\,x)\right) \ . \tag{12.9.6}$$

Question 12.20
Use an interpolating polynomial to derive a formula of the form

$$\int_a^b f(x)\,\mathrm{d}x \approx Af\left(a + \frac{b-a}{3}\right) + Bf\left(a + \frac{2(b-a)}{3}\right) \ .$$

If there exists a constant M such that $|f''(x)| < M$ for all $x \in [a, b]$, find a bound for the truncation error of this formula.

Question 12.21
Use polynomial interpolation to derive a formula of the form

$$\int_a^b f(x)\,\mathrm{d}x \approx A\,f\left(a + \frac{b-a}{4}\right) + B\,f\left(a + \frac{b-a}{2}\right) + C\,f\left(a + \frac{3(b-a)}{4}\right) .$$

Find a bound on the truncation error if there exists a constant M such that $|f^{(3)}(x)| < M$ for all $x \in [a, b]$.

Question 12.22
a) Use polynomial interpolation to find an integration formula of the form

$$\int_0^h f(x)\,\mathrm{d}x \approx h\left(Af(0) + Bf(-h) + Cf(-2h)\right)$$

with its truncation error.
b) Use the formula in (a) to deduce a formula for the integral $\int_a^b f(x)\,\mathrm{d}x$ and its truncation error.
c) Use the formula that you have found in (b) to approximate the value of the solution of the differential equation $y' = f(x, y)$ at $a + h$ if you know the values of $y(a)$, $y(a-h)$ and $y(a-2h)$.

Note: The formula that you use in (c) is a "Fourth-Order Adams-Bashforth Formula." We will study such methods of integration in Section 13.5. They are called "multistep methods" because they use nodes, $a - h$ and $a - 2h$, before a to approximate $y(a + h)$.

Question 12.23
a) Use polynomial interpolation to find an integration formula of the form

$$\int_{-h}^h f(x)\,\mathrm{d}x \approx h\left(A\,f(0) + B\,f(-h) + C\,f(-2h)\right)$$

with its truncation error.
b) Use the formula in (a) to deduce a formula for the integral between a and b, and its truncation error.

Question 12.24
Suppose that $a \leq x_1 < x_2 < \ldots < x_k \leq b$ and $w : [a, b] \to [0, \infty[$ is a weight function. Give a different proof than the one given at the beginning of Section 12.7 that there exist constants $c_1, c_2, \ldots, c_k$ such that

$$\int_a^b p(x)\,w(x)\,\mathrm{d}x = \sum_{j=1}^k c_j\,p(x_j) \tag{12.9.7}$$

for all polynomials p of degree less than k.

Question 12.25
Let $w : [a, b] \to [0, \infty[$ be a weight function. Suppose that P is a polynomial of degree $k > 0$ with k distinct roots $x_1, x_2, \ldots, x_k$ in the interval $[a, b]$ such that

$$\langle p, P\rangle = \int_a^b p(x)\,P(x)\,w(x)\,\mathrm{d}x = 0$$

for all polynomials p of degree less than $m \leq k$. Let $c_1, c_2, \ldots, c_k$ be the coefficients given in (12.7.2). Show that (12.9.7) is exact for all polynomials of degree less than $k + m$.

Moreover, if $\langle x^m, P\rangle \neq 0$, show that (12.9.7) is not true for all polynomials of degree equal to $k + m$.

Question 12.26
Let $w : [a, b] \to [0, \infty[$ be a weight function and suppose that there exist nodes x_j in $[a, b]$ and weight c_j for $1 \leq j \leq k$ such that

$$\int_a^b p(x)\, w(x)\, \mathrm{d}x = \sum_{j=1}^{k} b_j\, p(c_j)$$

for all polynomials p of degree less than q. Show that exists a constant $K = K(a, b, w, k, q, b_j)$ such that

$$\left| \int_a^b f(x)\, w(x)\, \mathrm{d}x - \sum_{j=1}^{k} b_j\, f(c_j) \right| \leq K(b-a)^q \max_{a \leq x \leq b} |f^{(q)}(x)|$$

for all q-time continuously differentiable functions f on an open interval containing $[a, b]$.

Question 12.27
Approximate

$$\int_0^{\pi/4} x^2 \sin(x)\, \mathrm{d}x$$

using Gauss-Legendre quadrature with $n = 5$.

Question 12.28
Approximate

$$\int_1^3 x^2 \ln(x)\, \mathrm{d}x \tag{12.9.8}$$

using Gauss Legendre quadrature with $n = 5$.

Question 12.29
Use the appropriate Gaussien quadrature (Legendre or Chebyshev) with $n = 3$ to approximate the integral

$$\int_2^3 \frac{\sin(x)}{\sqrt{(x-2)(3-x)}}\, \mathrm{d}x \ .$$

Question 12.30
Use the appropriate Gaussian quadrature (Legendre or Chebyshev) with $n = 3$ to find an approximation of the integral

$$\int_2^3 \frac{\sin(x)}{\sqrt{-6 + 5x - x^2}}\, \mathrm{d}x \ .$$

Question 12.31
Use the appropriate Gaussian quadrature formula with $n = 3$ to approximate the following integral. Determine if the approximation is exact?

$$\int_0^2 \frac{1}{\sqrt{(4 - x^2)\cos(x/4)}}\, \mathrm{d}x \ .$$

Question 12.32
Use the appropriate Gaussian quadrature (Legendre or Chebyshev) with $n = 3$ to find an approximation of the integral
$$\int_0^1 \frac{e^x}{\sqrt{x(1-x)}}\,\mathrm{d}x\ .$$

Question 12.33
Use the appropriate Gaussian quadrature (Legendre or Chebyshev) to compute **exactly** the integral
$$\int_0^1 \frac{x^2}{\sqrt{1-x^2+x^4-x^6}}\,\mathrm{d}x\ .$$

Question 12.34
Use the appropriate Gaussian quadrature (Legendre or Chebyshev) to compute **exactly** the value of the integral
$$\int_{-3}^1 \frac{(1+x)^4}{\sqrt{(1-x)(3+x)}}\,\mathrm{d}x\ .$$

Question 12.35
Use the appropriate Gaussian quadrature (Legendre or Chebyshev) to compute **exactly** the integral
$$\int_0^2 \frac{x^4+5}{\sqrt{4-x^2}}\,\mathrm{d}x\ .$$

Question 12.36
Find a Gaussian quadrature formula of the form
$$\int_0^1 x\,f(x)\,\mathrm{d}x \approx A\,f(x_1) + B\,f(x_2) \tag{12.9.9}$$
that is exact for polynomial f of degree up to 3.

Question 12.37
Find a Gaussian quadrature formula of the form
$$\int_{-1}^1 x^2 f(x)\,\mathrm{d}x \approx A\,f(x_1) + B\,f(x_2) \tag{12.9.10}$$
that is exact for polynomial f of degree up to 3.

Question 12.38
Let $f : [a,b] \to \mathbb{R}$ be a sufficiently differentiable function. If $\max\limits_{a\le x\le b} |f^{(n+1)}(x)| < M$ for some constant M, find a bound for the truncation error of the Gauss-Chebyshev quadrature with $n > 0$.

Question 12.39
If the formula

$$\int_a^b f(x)\,w(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} a_i\,f(x_i)$$

is exact for all polynomials of degree less than $2n$, show that $\prod_{j=1}^{n}(x-x_j)$ is orthogonal to all polynomials of degree less than n with respect to the weight function w.

Question 12.40
Could a Gaussian quadrature formula of the form

$$\int_a^b f(x)\,w(x)\,\mathrm{d}x = \sum_{j=1}^{n} c_j\,f(x_j) \tag{12.9.11}$$

be exact for polynomial of degree $2n$?

Question 12.41
Use polynomial interpolation to derive a Gaussian quadrature formula of the form

$$\int_a^b f(x)\,\mathrm{d}x \approx c_0 f(a) + c_1 f(b) + c_2 f'(a) + c_3 f'(b)\ . \tag{12.9.12}$$

What is the highest value n such that (12.9.12) is exact for polynomials of degree smaller than n. This type of Gaussian quadrature is called **Gauus-Hermite quadrature**.

Chapter 13

Initial Value Problems for Ordinary Differential Equations

13.1 Introduction to Ordinary Differential Equations

We consider the initial value problem

$$\begin{aligned} \frac{\mathrm{d}y}{\mathrm{d}t}(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(t_0) &= y_0 \end{aligned} \tag{13.1.1}$$

where $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$.

In this section, we develop numerical algorithms to approximate the solution y of (13.1.1) on $[t_0, t_f]$. Before attempting to numerically solve (13.1.1), we have to get a positive answer to the following questions.

Question(s)

Is the initial value problem (13.1.1) "well-posed?" Namely, is there a solution to (13.1.1) and, if there is one, is it unique? Moreover, does a small "perturbation" of (13.1.1) implies only a "small variation" in the solution of (13.1.1)?

If the initial value problem is not well-posed, then there is not point in attempting to numerically solve (13.1.1). Suppose that (13.1.1) is deduced from experimental data associated to a given physical phenomenon, then (13.1.1) is a "perturbation" of the real ordinary differential equation governing this physical phenomenon. Hence, if the problem is not "well-posed", then the analytical solution of (13.1.1) may not be related to the analytical solution of the real ordinary differential equation governing the physical phenomenon. The same can be said about the numerical solution. Moreover, even if (13.1.1) is the real ordinary differential equation governing the physical phenomenon, solving (13.1.1) numerically is equivalent to solving analytically a "perturbation" of (13.1.1). Hence, if the problem is not "well-posed", then the numerical solution of (13.1.1) may not be related to the analytic solution of (13.1.1) but to the analytic solution of the "perturbation" of (13.1.1)

Before giving conditions on f that guarantee that an initial value problem (13.1.1) is "well-posed", we have to clarify the meaning of "perturbation" and "well-posed" problem.

Definition 13.1.1

A **perturbation** of (13.1.1) is an initial value problem of the form

$$\begin{aligned} \frac{\mathrm{d}z}{\mathrm{d}t}(t) &= f(t, z(t)) + \delta(t) \quad , \quad t_0 \le t \le t_f \\ z(t_0) &= y_0 + \delta_0 \end{aligned} \tag{13.1.2}$$

where $\delta : [t_0, t_f] \to \mathbb{R}$ is a continuous function and δ_0 is a constant.

Definition 13.1.2

The initial value problem (13.1.1) is **well posed** if:

1. There is a unique solution y to (13.1.1).
2. There exist positive constants K and E such that for any positive $\epsilon \le E$, the solution $z(t)$ of the perturbed problem (13.1.2) satisfies

$$|y(t) - z(t)| < K\epsilon$$

for all $t \in [t_0, t_f]$ if $|\delta(t)| < \epsilon$ for all $t \in [t_0, t_f]$ and $|\delta_0| < \epsilon$ (Figure 13.1).

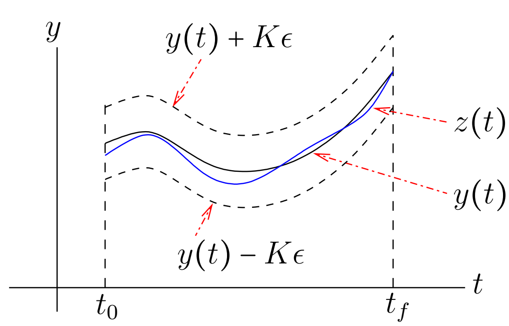


Figure 13.1: Uniform approximation of y by z on $[t_0, t_f]$.

The following theorem gives conditions for the initial value problem (13.1.1) to be well-posed.

Theorem 13.1.3

Let $D = \{(t,y) : t_0 \le t \le t_f \text{ and } -\infty < y < \infty\}$. Suppose that $f : D \to \mathbb{R}$ is continuous and that there exists a constant L such that

$$|f(t,y) - f(t,\tilde{y})| \le L|y - \tilde{y}| \tag{13.1.3}$$

for all (t,y) and $(t,\tilde{y})$ in D. Then the initial value problem (13.1.1) is well-posed.

Remark 13.1.4

If (13.1.3) is satisfied, we say that f satisfies a **Lipschitz condition** with respect to its second variable on D or that f is **Lipschitz continuous** with respect to its second variable on D. L is called a **Lipschitz constant**. ♠

Proof (partial).

The existence and uniqueness of the solution of the initial value problem (13.1.1) is usually proved in good introductory textbooks on ordinary differential equations. The main idea for the proof of existence given by Peano is to construct a contraction mapping whose fixed point is the local solution of the ordinary differential equations.

We prove the second condition of the definition of well-posed problem.

Let $r(t) = z(t) - y(t)$ where $y(t)$ is the solution of (13.1.1) and $z(t)$ is the solution of (13.1.2). If we subtract (13.1.1) from (13.1.2), we get

$$\begin{aligned} r'(t) &= f(t,z(t)) - f(t,y(t)) + \delta(t) \\ r(t_0) &= \delta_0 \end{aligned}$$

As required in Definition 13.1.2, let's assume that $|\delta(t)| < \epsilon$ for all $t \in [t_0,t_f]$ and $|\delta_0| < \epsilon$. We get from (13.1.3) that

$$r(t) - r(t_0) = \int_{t_0}^{t} r'(s)\,\mathrm{d}s = \int_{t_0}^{t} f(s,z(s)) - f(s,y(s))\,\mathrm{d}s + \int_{t_0}^{t} \delta(s)\,\mathrm{d}s\ .$$

Hence,

$$|r(t)| \le |r(t_0)| + \int_{t_0}^{t} |f(s,z(s)) - f(s,y(s))|\,\mathrm{d}s + \int_{t_0}^{t} |\delta(s)|\,\mathrm{d}s \le \epsilon + L\int_{t_0}^{t} |r(s)|\,\mathrm{d}s + \epsilon(t_f - t_0)$$

for all $t \in [t_0,t_f]$. It follows from Gronwall's Lemma[1] that

$$|r(t)| \le \epsilon(1 + t_f - t_0)e^{L(t-t_0)} \le \epsilon(1 + t_f - t_0)e^{L(t_f-t_0)}$$

for all $t \in [t_0,t_f]$. We conclude that

$$|r(t)| \le K\epsilon$$

with

$$K = (1 + t_f - t_0)e^{L(t_f-t_0)}$$

for $t \in [t_0,t_f]$. ∎

[1] Another fundamental result that one can find in a good introductory textbook on ordinary differential equations.

13.2 Euler's Method

We introduce in this section the simplest numerical method to approximate the solution of an initial value problem. Even if it is not the best numerical method, it is still a good method to introduce most of the concepts and issues involved in the numerical approximation of solutions of initial value problem.

Suppose that (13.1.1) is well-posed. The general procedure to **approximate the solution** of (13.1.1) is as follows:

1. Choose a positive integer N.
2. Select $N+1$ **mesh points** $t_0 < t_1 < t_2 < \ldots < t_N = t_f$ (usually equally spaced.)
3. Find an approximation w_i of $y_i = y(t_i)$ for $i = 1, 2, \ldots, N$.
4. Use linear interpolation at the points (t_i, w_i) (or higher order polynomial interpolation) to approximate $y(t)$ at $t \neq t_i$ for $i = 0, 1, \ldots, N$ (Figure 13.2).

Our first procedure to compute an approximation w_i to y_i is the Euler's method.

Definition 13.2.1 (Euler's Method)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= w_i + h\, f(t_i, w_i) \quad , \quad 0 \leq i < N \\ w_0 &= y_0 \end{aligned} \tag{13.2.1}$$

Remark 13.2.2

1. The mesh points in the presentation of the Euler's method are equally spaced. However, the mesh points t_i do not have to be equally spaced. We may simply require that $h_i = t_{i+1} - t_i$ for $0 \leq i < N$ satisfy $\displaystyle \max_{0 \leq i < N} |h_i| < K \min_{0 \leq i < N} |h_i|$ for a constant K.
2. The Euler's method can be justified as follows. Suppose that f is continuously differential with respect to both variables. From Theorem 2.1.6, we have
$$y(t_{i+1}) = y(t_i) + y'(t_i)\,(t_{i+1} - t_i) + \frac{y''(\xi_i)}{2}\,(t_{i+1} - t_i)^2$$
for some ξ_i between t_i and t_{i+1}. If we substitute $y'(t_i) = f(t_i, y(t_i))$, $y_i = y(t_i)$ and $h = t_{i+1} - t_i$ in the previous equation, we get
$$y_{i+1} = y_i + f(t_i, y_i)\, h + \frac{y''(\xi_i)}{2}\, h^2 \tag{13.2.2}$$
for some ξ_i between t_i and t_{i+1}. If we assume that $y''(\xi_i)h^2/2$ is much smaller than $y_i + f(t_i, y_i)\, h$ for all i and for h small enough, we get the Euler's method by removing this term from (13.2.2).

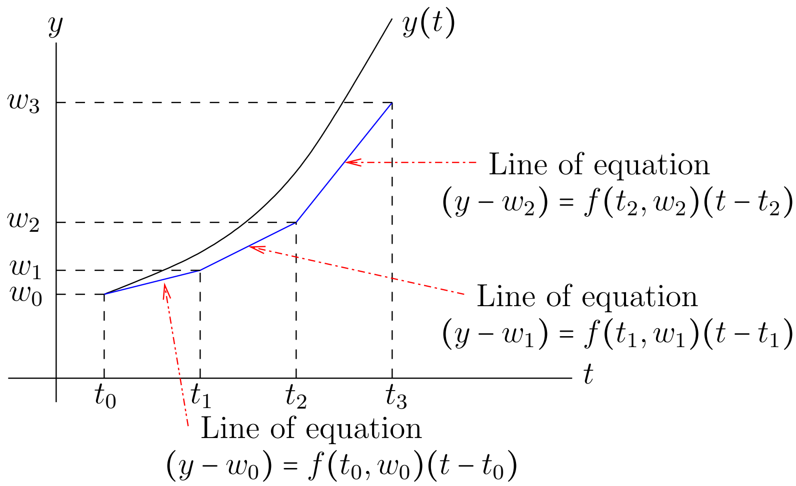


Figure 13.2: Graph of the solution y and its approximation given by the Euler's method.

3. The **local discretization error** for the Euler's method is $y''(\xi_i)h^2/2$.

♠

Example 13.2.3
Use Euler's method with $N = 5$ to approximate the solution y of

$$\begin{aligned} y'(t) &= 0.2\,ty \quad , \quad 1 \le t \le 1.5 \\ y(1) &= 1 \end{aligned} \tag{13.2.3}$$

We have $t_0 = 1$, $t_f = t_5 = 1.5$, $y_0 = 1$ and $f(t,y) = 0.2\,ty$. Hence $h = (t_5 - t_0)/5 = 0.1$, $t_i = t_0 + ih = 1 + 0.1i$ and the approximation w_i of $y_i = y(t_i)$ is given by

$$\begin{aligned} w_0 &= 1 \\ w_{i+1} &= w_i + 0.02\,t_i w_i \quad , \quad 0 \le i < 5. \end{aligned}$$

The results of these computations are given in the following table.

i	t_i	w_i	y_i	absolute error	relative error
0	1.00	1.0000	1.0000	0.0	0.0
1	1.10	1.02	1.0212220516	−0.0012220516	0.0011966561
2	1.20	1.04244	1.0449823549	−0.0025423549	0.0024329166
3	1.30	1.06745856	1.0714362091	−0.0039776491	0.0037124461
4	1.40	1.0952124826	1.1007590640	−0.0055465814	0.0050388696
5	1.50	1.1258784321	1.1331484531	−0.0072700210	0.0064157710

Since the differential equation in (13.2.3) is separable, it is easy to find the exact solution $y(t) = e^{0.1t^2-0.1}$ of (13.2.3). We have used this formula to compute y_i. Our approximation w_5 of y_5 has a relative error of about 0.64 %. This is good. ♣

Example 13.2.4
Use the Euler's method with $N = 5$ to approximate the solution y of

$$\begin{aligned} y'(t) &= 2\,ty \quad , \quad 1 \le t \le 1.5 \\ y(1) &= 1 \end{aligned} \tag{13.2.4}$$

As in the previous example, we have $t_0 = 1$, $t_f = t_5 = 1.5$ and $y_0 = 1$. However, $f(t,y) = 2\,ty$. Hence $h = (t_5 - t_0)/5 = 0.1$, $t_i = t_0 + hi = 1 + 0.1i$ and the approximation w_i of $y_i = y(t_i)$ is given by

$$\begin{aligned} w_0 &= 1 \\ w_{i+1} &= w_i + 0.2\,t_i w_i \quad , \quad 0 \le i < 5 \ . \end{aligned}$$

The results of these computations are given in the following table.

i	t_i	w_i	y_i	absolute error	relative error
0	1.0	1.0000	1.0000	0.0	0.0
1	1.1	1.2000	1.2336780600	−0.0336780600	0.0272989048
2	1.2	1.4640	1.5527072185	−0.0887072185	0.0571306795
3	1.3	1.81536	1.9937155332	−0.1783555332	0.0894588673
4	1.4	2.2873536	2.6116964734	−0.3243428734	0.1241885789
5	1.5	2.927812608	3.4903429575	−0.5625303495	0.1611676435

Since the differential equation in (13.2.4) is separable, it is easy to find the exact solution $y(t) = e^{t^2-1}$ of (13.2.4). We have used this formula to compute y_i. Our approximation w_5 of y_5 has a relative error of about 16.12 %. This is not good. The Euler's method does not give good approximations of y_i for $1 \le i \le 5$.

To find the reason behind these poor numerical results compared to those of the previous example, we have to compare the graphs of the solutions. The graph of the solution of (13.2.3) is concave up and so is the graph of the solution of (13.2.4) because $f(t,y) > 0$ for $t > 0$ and $y > 0$ in both cases. However, the solution of (13.2.4) increases a lot faster than the solution of (13.2.3). It is therefore easy to imagine (Figure 13.3) that the distance between the graph of the solution of (13.2.4) and the graph of its numerical approximation increases faster than the distance between the graph of the solution of (13.2.3) and the graph of its numerical approximation. ♣

We now investigate the effect of local discretization and rounding error on the Euler's method. Due to rounding error, solving (13.2.1) numerically is equivalent to solving

$$\begin{aligned} u_{i+1} &= u_i + h\,f(t_i, u_i) + \delta_{i+1} \\ u_0 &= y_0 + \delta_0 \end{aligned} \tag{13.2.5}$$

exactly, where δ_0 is the error in approximating y_0 and δ_{1+i} is the rounding error in computing $w_i + h\,f(t_i, w_i)$. For $0 \le i \le N$, the value u_i represents the computed value of w_i.

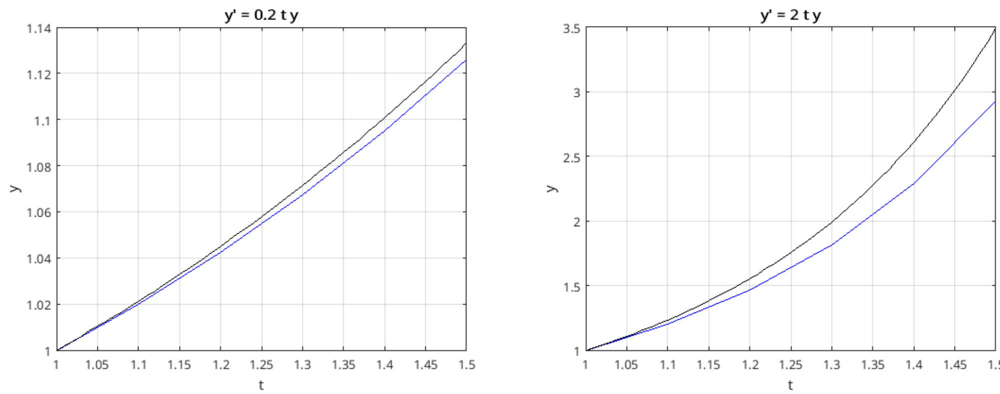


Figure 13.3: Graphs of the solution (in black) of (13.2.3) on the left and of (13.2.4) on the right with the graphs of their approximation (in blue) given by the Euler's method.

Theorem 13.2.5

Let $e_i = y_i - u_i$ for $i = 0, 1, \ldots, N$. Suppose that:

1. There exists δ such that $|\delta_i| < \delta$ for $i = 1, 2, \ldots, N$. (Note that for a given computer, this assumption makes sense.)
2. The function f satisfies the Lipschitz condition (13.1.3) on $[t_0, t_f] \times \mathbb{R}$.
3. There exists $M > 0$ such that $|y''(t)| < M$ for all t in $[t_0, t_f]$.

Then,
$$|e_i| \leq \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\left(e^{L(t_i - t_0)} - 1\right) + |\delta_0|\, e^{L(t_i - t_0)} . \tag{13.2.6}$$

Proof.
If we subtract the first equation of (13.2.5) from (13.2.2), we get
$$\begin{aligned} e_{i+1} &= e_i + h\left(f(t_i, y_i) - f(t_i, u_i)\right) + \frac{y''(\xi_i)}{2}\, h^2 - \delta_{1+i} \\ e_0 &= -\delta_0 \end{aligned} \tag{13.2.7}$$
Since f satisfies the Lipschitz condition (13.1.3) on $[t_0, t_f] \times \mathbb{R}$, we have
$$|f(t, y_i) - f(t, u_i)| \leq L|y_i - u_i| .$$
Hence, if we take the absolute value on both sides of the equations in (13.2.7), we get
$$\begin{aligned} |e_{i+1}| &\leq |e_i| + hL|e_i| + \frac{M}{2}\, h^2 + \delta \\ |e_0| &= |\delta_0| \end{aligned} \tag{13.2.8}$$

Consider the difference equation

$$\begin{aligned} \eta_{i+1} &= \eta_i + hL\eta_i + \frac{M}{2}\,h^2 + \delta \\ \eta_0 &= |\delta_0| \end{aligned} \tag{13.2.9}$$

The solution of (13.2.9) is

$$\eta_i = A(1+hL)^i + B \ ,$$

where

$$A = |\delta_0| + \frac{1}{hL}\left(\frac{M}{2}\,h^2 + \delta\right) = |\delta_0| + \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)$$

and

$$B = -\frac{1}{hL}\left(\frac{M}{2}\,h^2 + \delta\right) = -\frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right) \ .$$

We can easily show by induction that $|e_i| \leq \eta_i$ for $i = 0, 1, \ldots, N$. This is true for $i = 0$ because $\eta_0 = |\delta_0| = |e_0|$. Suppose that $|e_i| \leq \eta_i$ for $i = 0, 1, \ldots, k$. Then it follows from (13.2.8) that

$$|e_{k+1}| \leq |e_k| + hL|e_k| + \frac{M}{2}\,h^2 + \delta \leq \eta_k + hL\eta_k + \frac{M}{2}\,h^2 + \delta = \eta_{k+1} \ .$$

Thus, $|e_i| \leq \eta_i$ for all i by induction.

Hence,

$$\begin{aligned} |e_i| \leq \eta_i = A(1+hl)^i + B &= \left(|\delta_0| + \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\right)(1+hL)^i - \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right) \\ &\leq \left(|\delta_0| + \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\right) e^{ihL} - \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right) \\ &= \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\left(e^{L(t_i - t_0)} - 1\right) + |\delta_0|\, e^{L(t_i - t_0)} \ , \end{aligned}$$

where we have used $(1+x)^n \leq e^{nx}$ for $x > 0$ in the second inequality, and $ih = t_i - t_0$ in the last equality. ∎

Remark 13.2.6

1. If we assume the idealistic case where there are no rounding and approximation errors, namely $\delta = \delta_0 = 0$, then (13.2.6) in Theorem 13.2.5 yields

   $$|w_i - y(t_i)| = |e_i| \leq \frac{M}{2L}\left(e^{L(t_f - t_0)} - 1\right) h$$

   for all i. It follows that

   $$\lim_{h \to 0} \max_{0 \leq i \leq N} |w_i - y(t_i)| = 0 \ .$$

2. As for numerical differentiation, Euler's method is sensitive to rounding error. Theorem 13.2.5 suggests that

$$|e_i| \approx \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\left(e^{L(t_i - t_0)} - 1\right) + |\delta_0|\, e^{L(t_i - t_0)} ,$$

where the factor $\dfrac{Mh}{2} + \dfrac{\delta}{h}$ goes to infinity as h goes to 0.

♠

13.3 Higher-Order Taylor Methods

The Euler's method is a nice method to introduce the concept of numerical solution of initial value problems of the form (13.1.1). However, it is not a really good method. We now turn our attention to "better methods." Before that, we need some concepts to define what we mean by "better methods."

Definition 13.3.1

The **local truncation error** of a method of the form

$$\begin{aligned} w_{i+1} &= w_i + h\phi(t_i, w_i) \quad , \quad 0 \le i < N \\ w_0 &= y_0 \end{aligned} \tag{13.3.1}$$

to numerically solve (13.1.1) is defined as

$$\tau_{i+1}(h) = (y_{i+1} - y_i)/h - \phi(t_i, y_i) \tag{13.3.2}$$

with $y_i = y(t_i)$ for $i = 0,\ 1,\ 2, \ldots,\ N$.
If there exist a function $\tau : \mathbb{R} \to \mathbb{R}$ such that $|\tau_{i+1}(h)| \le \tau(h)$ for all i, and a positive integer k (as large as possible) such that $\tau(h) = O(h^k)$ near the origin, then we say that the method (13.3.1) is of **order k**.

Example 13.3.2
For the Euler's method $\phi(t, y) = f(t, y)$ in (13.3.1) It follows from (13.2.2) that the local truncation error for the Euler's method is

$$\tau_{i+1}(h) = (y_{i+1} - y_i)/h - f(t_i, y_i) = \frac{y''(\xi_i)}{2}\, h$$

for some ξ_i in $[t_{i-1}.t_i]$. If there exists a constant M such that $|y''(t)| \le M$ for all $t \in [t_0, t_f]$, then $|\tau_{i+1}(h)| \le \tau(h) \equiv M|h|/2$ for all i and $\tau(h) = O(h)$ near the origin. So, the Euler's method is of order 1. ♣

Remark 13.3.3
Since $|h^p|$ is a decreasing function of p for $|h| < 1$ fixed, the local truncation error is generally smaller for high order methods than for low order methods. ♠

Definition 13.3.4 (Taylor Method of order 2)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= w_i + h\phi(t_i, w_i) \quad , \quad 0 \leq i < N \\ w_0 &= y_0 \end{aligned}$$

where

$$\phi(t,y) = f(t,y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t,y) + \left(\frac{\partial f}{\partial y}(t,y)\right) f(t,y)\right) .$$

Remark 13.3.5

1. Assuming that f is sufficiently differentiable, we may derive Taylor methods of order $n > 2$ by differentiating $n-1$ times with respect to t the expression $f(t, y(t))$. The Taylor method of order n can be justified as follows. From Theorem 2.1.6, we have

$$\begin{aligned} y(t_{i+1}) &= y(t_i) + y'(t_i)\,(t_{i+1} - t_i) + \frac{y''(t_i)}{2}\,(t_{i+1} - t_i)^2 + \ldots + \frac{y^{(n)}(t_i)}{n!}\,(t_{i+1} - t_i)^n \\ &\quad + \frac{y^{(n+1)}(\xi_i)}{(n+1)!}\,(t_{i+1} - t_i)^{n+1} \end{aligned}$$

for some ξ_i between t_i and t_{i+1}. We also have that

$$\begin{aligned} y'(t) &= f(t, y(t)) \ , \\ y''(t) &= \frac{\mathrm{d}}{\mathrm{d}t} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \left(\frac{\partial f}{\partial y}(t, y(t))\right) y'(t) \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \left(\frac{\partial f}{\partial y}(t, y(t))\right) f(t, y(t)) \ , \\ y^{(3)}(t) &= \ldots \end{aligned}$$

Since $h = t_{i+1} - t_i$, $y_i = y(t_i)$, $y'(t_i) = f(t_i, y_i)$, $y''(t_i) = \dfrac{\partial f}{\partial t}(t_i, y_i) + \dfrac{\partial f}{\partial y}(t_i, y_i)\, f(t_i, y_i)$, $\ldots$, we get

$$\begin{aligned} y_{i+1} &= y_i + h\underbrace{\left(f(t_i, y_i) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + \frac{\partial f}{\partial y}(t_i, y_i)\, f(t_i, y_i)\right) + \ldots\right)}_{=\phi(t_i, y_i)} \\ &\quad + \frac{y^{(n+1)}(\xi_i)}{(n+1)!}\, h^{n+1} \end{aligned} \tag{13.3.3}$$

for some ξ_i between t_i and t_{i+1}. If we assume that $\dfrac{y^{(n+1)}(\xi_i)}{(n+1)!}\, h^{n+1}$ is small for all i, we get the Taylor method of order n by removing this term from (13.3.3).

2. The **local discretization error** for the Taylor method of order n is
$$\frac{1}{(n+1)!}\, y^{(n+1)}(\xi_i) h^{n+1} \ .$$

3. The local truncation error is
$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i) = \frac{y^{(n+1)}(\xi_i)}{(n+1)!} h^n$$
for $\xi_i \in [t_i, t_{i+1}]$. If there exists a constant M such that $|y^{(n+1)}(t)| \leq M$ for all $t \in [t_0, t_f]$, then $|\tau_{i+1}(h)| \leq \tau(h) \equiv M|h|^n/(n+1)!$ for all i and $\tau(h) = O(h^n)$ near the origin. This justifies the name Taylor method of order n.

4. From a numerical point of view, the Taylor methods of order $n > 1$ are not very useful. We may use these methods only when $\phi(t, y)$ can be easily computed symbolically. Moreover, though the local truncation error is smaller for the Taylor methods of order $n > 1$ than it is for the Euler's method, rounding error is generally larger for the Taylor methods of order $n > 1$ because of the number of numerical operations necessary to evaluate $\phi(t_i, w_i)$ for $i = 0, 1, \ldots, N-1$.

♠

13.4 Runge-Kutta Methods

In this section, we develop numerical methods to approximate the solution of (13.1.1) that are of order greater than one and do not require the evaluation of complicate functions like $\phi(t, y)$ in the high order Taylor methods.

Definition 13.4.1 (**General Form of the Runge-Kutta Method**)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation
$$\begin{aligned} w_{i+1} &= w_i + h \sum_{j=1}^{s} \gamma_j K_j \quad , \quad 0 \leq i < N \\ w_0 &= y_0 \end{aligned}$$
where s is a positive integer,
$$K_j = f(t_i + \alpha_j h, w_i + h \sum_{m=1}^{s} \beta_{j,m} K_m) \quad , \quad 1 \leq j \leq s$$
and α_j, $\beta_{j,m}$ and γ_j are constants such that
$$\alpha_j = \sum_{m=1}^{s} \beta_{j,m} \quad \text{and} \quad \sum_{j=1}^{s} \gamma_j = 1 \ . \tag{13.4.1}$$

The values K_m are called the **stages** and the method is described as a s**-stage Runge-Kutta method**.
If $\beta_{j,m} = 0$ for $m \geq j$, the Runge-Kutta method is called an **explicit method**. Otherwise, it is called an **implicit method**. If $\beta_{j,m} = 0$ for $m > j$, the Runge-Kutta method is called a **semi-implicit method**.

The classical way to describe a Runge-Kutta method is with its **Butcher array**.

$$
\begin{array}{c|cccc}
\alpha_1 & \beta_{1,1} & \beta_{1,2} & \ldots & \beta_{1,1} \\
\alpha_2 & \beta_{2,1} & \beta_{2,1} & \ldots & \beta_{2,1} \\
\alpha_3 & \beta_{3,1} & \beta_{3,2} & \ldots & \beta_{3,1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\alpha_s & \beta_{s,1} & \beta_{s,2} & \ldots & \beta_{s,s} \\
\hline
 & \gamma_1 & \gamma_2 & \ldots & \gamma_s
\end{array}
$$

The Butcher array for the explicit Runge-Kutta methods is as follows.

$$
\begin{array}{c|ccccc}
\alpha_1 & & & & & \\
\alpha_2 & \beta_{2,1} & & & & \\
\alpha_3 & \beta_{3,1} & \beta_{3,2} & & & \\
\vdots & \vdots & \vdots & & & \\
\alpha_s & \beta_{s,1} & \beta_{s,2} & \ldots & \beta_{s,s-1} & \\
\hline
 & \gamma_1 & \gamma_2 & \ldots & \gamma_{s-1} & \gamma_s
\end{array}
$$

The Runge-Kutta methods do not only get some information from the solution through $(t_0, y(t_0))$ but also from solutions that are near the solution through $(t_0, y(t_0))$. This information comes from $f(t, y)$. Let's consider the explicit Runge-Kutta methods. We first note that $\alpha_1 = 0$ because of (13.4.1).

- We have $K_1 = f(t_i, w_i)$.
- $K_2 = f(t_i + \alpha_2 h, w_i + \beta_{2,1} h K_1)$, where $w_i + \beta_{2,1} h K_1 = w_i + \alpha_2 h f(t_i, w_i)$ is the approximation of $y(t)$ at $t = t_i + \alpha_2 h$ given by the Euler's method.
- $K_3 = f(t_i + \alpha_3 h, w_i + \beta_{3,1} h K_1 + \beta_{3,2} h K_2)$, where $\beta_{3,1} K_1 + \beta_{3,2} K_2 = (\alpha_3 - \beta_{3,2}) K_1 + \beta_{3,2} K_2$ is a weighted average of the approximations of $y'(t)$ at t_i and $t_i + \alpha_2 h$ respectively.
- Similarly, $K_4 = f(t_i + \alpha_4 h, w_i + \beta_{4,1} h K_1 + \beta_{4,2} h K_2 + \beta_{4,3} h K_3)$, where $\beta_{4,1} K_1 + \beta_{4,2} K_2 + \beta_{4,3} K_3 = (\alpha_4 - \beta_{4,2} - \beta_{4,3}) K_1 + \beta_{4,2} K_2 + \beta_{4,3} K_3$ is a weighted average of the approximations of $y'(t)$ at t_i, $t_i + \alpha_2 h$ and $t_i + \alpha_3 h$.
- And so on for all the K_5, K_6, $\ldots$

Hence, K_j is an approximation of $y'(t)$ at $t_i + \alpha_j h$ for $1 \leq j \leq s$ and $w_{i+1} = w_i + h \sum_{k=1}^{s} \gamma_k K_k$, where $\sum_{k=1}^{s} \gamma_k K_k$ is a weighted average of these approximations.

Remark 13.4.2
We will see later that the conditions (13.4.1) are necessary conditions for an s-stage Runge-Kutta method to be of order s. ♠

We now present some of the most famous explicit Runge-Kutta methods.

Definition 13.4.3 (Runge-Kutta Methods of order two)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= w_i + h\left(\gamma_1 f(t_i, w_i) + \gamma_2 f(t_i + \alpha_2 h, w_i + \beta_{2,1} h f(t_i, w_i))\right) \quad , \quad 0 \leq i < N \\ w_0 &= y_0 \end{aligned}$$

where α_2, $\beta_{2,1}$, γ_1 and γ_2 are constants satisfying $\gamma_1 + \gamma_2 = 1$, $\alpha_2 = \beta_{2,1}$ and $\alpha_2\gamma_2 = 1/2$.

Remark 13.4.4

1. Some well known Runge-Kutta methods of order two are:

   Midpoint method: $\alpha_2 = \beta_{2,1} = 1/2$, $\gamma_1 = 0$ and $\gamma_2 = 1$, Its Butcher array is

   $$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

   Modified Euler's method: $\alpha_2 = \beta_{2,1} = 1$ and $\gamma_1 = \gamma_2 = 1/2$, Its Butcher array is

   $$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

   Heun's method: $\alpha_2 = \beta_{2,1} = 2/3$, $\gamma_1 = 1/4$ and $\gamma_2 = 3/4$. Its Butcher array is

   $$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline & 1/4 & 3/4 \end{array}$$

2. The motivation for the Runge-Kutta methods of order two is as follows. We assume that all the mixed partial derivatives of $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ exist and are continuous up to order two. Up to $O(h^2)$, we replace the function $\phi(t, y)$ in the Taylor method of order 2 (Definition 13.3.4) by an expression of the form $\gamma_1 K_1 + \gamma_2 K_2$, where $K_1 = f(t, y)$ and $K_2 = f(t + \alpha_2 h, y + \beta_{2,1} h K_1)$ for some γ_1, γ_2, α_2 and $\beta_{2,1}$.

   Using Taylor expansion theorem in two variables, we have

   $$f(t + \alpha_2 h, y + \beta_{2,1} h K_1) = f(t, y) + \alpha_2 h \frac{\partial f}{\partial t}(t, y) + \beta_{2,1} h K_1 \frac{\partial f}{\partial y}(t, y) + O(h^2) \ .$$

Hence

$$\begin{aligned}\gamma_1 K_1 + \gamma_2 K_2 = (\gamma_1 + \gamma_2) f(t,y) + \alpha_2 \gamma_2 h \frac{\partial f}{\partial t}(t,y) \\ + \beta_{2,1}\gamma_2 h f(t,y) \frac{\partial f}{\partial y}(t,y) + O(h^2)\ .\end{aligned} \tag{13.4.2}$$

If we match the coefficients of $f(t,y)$, $h\dfrac{\partial f}{\partial t}(t,y)$ and $hf(t,y)\dfrac{\partial f}{\partial y}(t,y)$ in (13.4.2) with those in

$$\phi(t,y) = f(t,y) + \frac{1}{2} h \frac{\partial f}{\partial t}(t,y) + \frac{1}{2} h f(t,y) \frac{\partial f}{\partial y}(t,y)\ ,$$

we get $\gamma_1 + \gamma_2 = 1$, $\alpha_2 \gamma_2 = 1/2$ and $\beta_{2,1}\gamma_2 = 1/2$.

3. If we assume that all the mixed partial derivatives of $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ exist and are continuous up to order two, then the local truncation errors of the Runge-Kutta methods in Definition 13.4.3 are bounded by a function τ (found for the Taylor method of order 2) such that $\tau(h) = O(h^2)$ near the origin as their name suggests.

♠

Example 13.4.5
Use the modified Euler's method with $N = 5$ to approximate the solution y of the initial value problem (13.2.4) of Example 13.2.4.

We have $t_0 = 1$, $t_f = t_5 = 1.5$, $y_0 = 1$ and $f(t,y) = 2ty$. Hence $h = (t_5 - t_0)/5 = 0.1$ and $t_j = t_0 + hi = 1 + 0.1i$. The approximation w_i of $y_i = y(t_i)$ is given by

$$\begin{aligned} w_0 &= 1 \\ \left.\begin{aligned} w_i^* &= w_i + 0.2\, t_i w_i \\ w_{i+1} &= w_i + 0.1\,(t_i w_i + t_{i+1} w_i^*) \end{aligned}\right\} &\quad , \quad 0 \le i < 5 \end{aligned}$$

The results of these computations are given in the following table:

i	t_i	w_i	y_i	absolute error	relative error
0	1.00	1.00000000	1.0000	0.0	0.0
1	1.10	1.23200000	1.23367806	0.00167806	0.00136021
2	1.20	1.54788480	1.55270722	0.00482242	0.00310581
3	1.30	1.98315006	1.99371553	0.01056553	0.00529942
4	1.40	2.59078717	2.61169647	0.02090931	0.00800602
5	1.50	3.45092851	3.49034296	0.03941445	0.01129243

We get approximation of y_i that are much better than those given by the Euler's method in Example 13.2.4. ♣

Definition 13.4.6 (Runge-Kutta Method of Order Four)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{6}(K_1 + 2\,K_2 + 2\,K_3 + K_4) \quad , \quad 0 \le i < N \\ w_0 &= y_0 \end{aligned}$$

where $K_1 = f(t_i, w_i)$, $K_2 = f(t_i + h/2, w_i + hK_1/2)$, $K_3 = f(t_i + h/2, w_i + hK_2/2)$ and $K_4 = f(t_{i+1}, w_i + hK_3)$.

This method is often called the **classical Runge-Kutta method**. A graphical interpretation of the Runge-Kutta method classic is given in Figure 13.4. Its Butcher array is

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

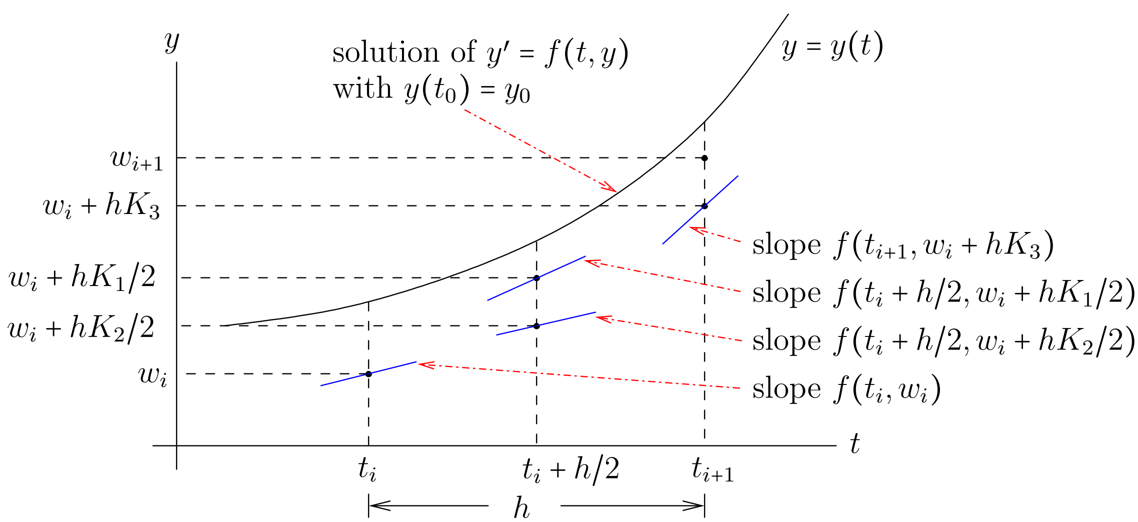


Figure 13.4: The expression $K_1/6 + 2K_2/3 + 2K_3/3 + K_4/6$ in the formula for the Runge-Kutta classic is a weighted average of the four slopes shown in the figure above.

Remark 13.4.7

1. The motivation for the Runge-Kutta methods of order four is as follows. We assume that all the mixed partial derivatives of $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ exist and are continuous up to order four. Up to $O(h^4)$, we replace the function $\phi(t, y)$ in the Taylor method

of order 4 (Definition 13.3.4) by an expression of the form $\gamma_1 K_1 + \gamma_2 K_2 + \gamma_3 K_3 + \gamma_4 K_4$ where $K_1 = f(t,y)$, $K_2 = f(t+\alpha_2 h, y + \beta_{2,1} h K_1)$, $K_3 = f(t+\alpha_3 h, y+\beta_{3,1} h K_1 + \beta_{3,2} h K_2)$ and $K_4 = f(t+\alpha_4 h, y+\beta_{4,1} h K_1 + \beta_{4,2} h K_3 + \beta_{4,3} h K_3)$ for some γ_j, α_j and $\beta_{j,m}$. After a long computation that can be found in [23], we find the following conditions on γ_j, α_j and $\beta_{j,m}$:

$$\begin{gathered}
\beta_{2,1} = \alpha_2 \ , \quad \beta_{3,1} + \beta_{3,2} = \alpha_3 \ , \quad \beta_{4,1} + \beta_{4,2} + \beta_{4,3} = \alpha_4 \ , \\
\sum_{j=1}^{4} \gamma_j = 1 \ , \quad \sum_{j=2}^{4} \gamma_j \alpha_j^k = \frac{1}{k+1} \quad \text{for} \quad k = 1,2,3 \ , \\
\gamma_3 \alpha_2 \beta_{3,2} + \gamma_4 (\alpha_2 \beta_{4,2} + \alpha_3 \beta_{4,3}) = \frac{1}{6} \ , \quad \gamma_3 \alpha_2 \alpha_3 \beta_{3,2} + \gamma_4 \alpha_4 (\alpha_2 \beta_{4,2} + \alpha_3 \beta_{4,3}) = \frac{1}{8} \ , \\
\gamma_3 \alpha_2^2 \beta_{3,2} + \gamma_4 (\alpha_2^2 \beta_{4,2} + \alpha_3^2 \beta_{4,3}) = \frac{1}{12} \quad \text{and} \quad \gamma_4 \alpha_2 \beta_{3,2} \beta_{4,3} = \frac{1}{24} \ .
\end{gathered}$$

The Runge-Kutta of order four given in the definition above corresponds to a particular choice for these constants. We will present in the following sections other techniques to develop Runge-Kutta methods.

2. If we assume that all the mixed partial derivatives of $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ exist and are continuous up to order four, then the local truncation error of the Runge-Kutta method in the previous definition is $O(h^4)$ as its name suggests.

3. Note that $\displaystyle\sum_{j=1}^{s} \gamma_j = 1$ is a necessary condition for the s-stage Runge-Kutta method to be of order s.

4. There is no explicit Runge-Kutta method of order s for $s \geq 5$ that have at most s stages. We have the following relation between the order of an explicit Runge-Kutta method and the number s of stage.

Order	1	2	3	4	5	6	7	8	9	10
Minimum stage number	1	2	3	4	6	7	9	11	$12 \leq s \leq 17$	$13 \leq s \leq 17$

♠

Example 13.4.8
Use Runge-Kutta classic of order four with $N = 5$ to approximate the solution y of (13.2.4) in Example 13.2.4.

We have $t_0 = 1$, $t_f = t_5 = 1.5$, $y_0 = 1$ and $f(t,y) = 2ty$. Hence $h = (t_5 - t_0)/5 = 0.1$ and $t_i = 1 + 0.1i$ for $i = 0, 1, \ldots, 5$. The approximation w_i of $y_i = y(t_i)$ is given by

$$\begin{aligned}
w_0 &= 1.0 \\
w_{i+1} &= w_i + \frac{0.1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad , \quad 0 \leq i < 5
\end{aligned}$$

where

$$K_1 = 2\,t_i w_i$$

$$K_2 = 2(t_i + 1/20)(w_i + K_1/20)$$
$$K_3 = 2(t_i + 1/20)(w_i + K_2/20)$$
$$K_4 = 2t_{i+1}(w_i + K_3/10)$$

The results of this computation are given in the following table:

i	t_i	w_i	y_i	absolute error	relative error
0	1.00	1.0000000000	1.0000000000	0.0	0.0
1	1.10	1.2336743500	1.2336780600	0.000003710	0.0000030072
2	1.20	1.5526953980	1.5527072185	0.000011820	0.0000076128
3	1.30	1.9936867693	1.9937155332	0.000028764	0.0000144273
4	1.40	2.6116332332	2.6116964734	0.000063240	0.0000242142
5	1.50	3.4902106364	3.4903429575	0.000132321	0.0000379106

We get approximations of y_i that are much better than those given by the modified Euler's method in Example 13.4.5 and the Euler's method in Example 13.2.4. ♣

Code 13.4.9 (Runge-Kutta of Order Four)

To approximate the solution of the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t \geq t_0 \\ y(0) &= y_0 \end{aligned}$$

Input: The function $f(t, y)$ (funct in the code below).
The step-size h.
The number of steps N.
The initial time t_0 (t0 in the code below) and the initial conditions y_0 (y0 in the code below) at t_0.
Output: The approximations w_i (ww(i+1) in the code below) of $y(t_i)$ at t_i (tt(i+1) in the code below).

```
function [tt,ww] = rgkt4(funct,h,N,t0,y0)
  tt(1) = t0;
  ww(1) = y0;
  h2 = h/2;
  for j=1:N
    tt(j+1) = tt(1)+j*h;
    k1 = h*funct(tt(j),ww(j));
    k2 = h*funct(tt(j)+h2,ww(j)+k1/2);
    k3 = h*funct(tt(j)+h2,ww(j)+k2/2);
    k4 = h*funct(tt(j+1),ww(j)+k3);
    ww(j+1) = ww(j) + (k1+2*(k2+k3)+k4)/6;
  end
end
```

13.4.1 Derivation of Runge-Kutta Methods – Collocation Method

We present a method to derive some Runge-Kutta methods in this section. A more general method will be presented in the next section.

As usual, we consider the initial value problem (13.1.1) and assume that we have a partition $t_0 < t_1 < \ldots < t_N = t_f$ of $[t_0, t_f]$ such that $t_{i+1} - t_i = h$ for $i = 0, 1, \ldots, N-1$.

Definition 13.4.10 (Collocation Method)

We consider k distinct nodes $\alpha_1 < \alpha_2 < \ldots < \alpha_k$ in $[0,1]$. Assuming that we have $w_i \approx y(t_i)$, we seek a polynomial p of degree k such that

$$\begin{aligned} p(t_i) &= w_i \\ p'(t_i + \alpha_j h) &= f(t_i + \alpha_j h, p(t_i + \alpha_j h)) \quad , \quad 1 \leq j \leq k \ . \end{aligned}$$

The approximation w_{i+1} of $y(t_{i+1})$ is given by $p(t_{i+1})$. We repeat this construction for $i = 0, 1, \ldots, N-1$.

The idea behind the collocation method is to use a polynomial of degree k on each interval $[t_i, t_{i+1}]$ to approximate the solution between t_i and t_{i+1}.

Theorem 13.4.11

Let

$$\ell_m(t) = \prod_{\substack{j=1 \\ j \neq m}}^{k} \frac{t - \alpha_j}{\alpha_m - \alpha_j}$$

for $1 \leq m \leq k$ and

$$\beta_{j,m} = \int_0^{\alpha_j} \ell_m(t) \, \mathrm{d}t \quad \text{and} \quad \gamma_j = \int_0^1 \ell_j(t) \, \mathrm{d}t$$

for $1 \leq j, m \leq k$. Then, the collocation method presented in Definition 13.4.10 is an implicit Runge-Kutta method with Butcher array

$$\begin{array}{c|cccc} \alpha_1 & \beta_{1,1} & \beta_{1,2} & \ldots & \beta_{1,k} \\ \alpha_2 & \beta_{2,1} & \beta_{2,2} & \ldots & \beta_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_k & \beta_{k,1} & \beta_{k,2} & \ldots & \beta_{k,k} \\ \hline & \gamma_1 & \gamma_2 & \ldots & \gamma_k \end{array}$$

Proof.

Suppose that p is the polynomial given in Definition 13.4.10. Let

$$q(t) = \sum_{m=1}^{k} p'(t_i + \alpha_m h) \, \ell_m\left(\frac{t - t_i}{h}\right) \ .$$

q and p' are two polynomials of degree $k-1$ that coincide at the k points $t_i+\alpha_j h$ for $1 \le j \le k$; namely, $q(t_i+\alpha_j h) = p'(t_i+\alpha_j h)$ for $1 \le j \le k$. Therefore $q(t) = p'(t)$ for all $t \in [t_i, t_{i+1}]$. Hence

$$p'(t) = \sum_{m=1}^{k} p'(t_i+\alpha_m h)\,\ell_m\left(\frac{t-t_i}{h}\right) = \sum_{m=1}^{k} f(t_i+\alpha_m h, p(t_i+\alpha_m h))\,\ell_m\left(\frac{t-t_i}{h}\right)$$

and

$$\begin{aligned} p(t) = p(t_i) + \int_{t_i}^{t} p'(s)\,\mathrm{d}s &= w_i + \sum_{m=1}^{k}\left(f(t_i+\alpha_m h, p(t_i+\alpha_m h)) \int_{t_i}^{t} \ell_m\left(\frac{s-t_i}{h}\right)\,\mathrm{d}s\right) \\ &= w_i + h\sum_{m=1}^{k}\left(f(t_i+\alpha_m h, p(t_i+\alpha_m h)) \int_0^{(t-t_i)/h} \ell_m(s)\,\mathrm{d}s\right)\ . \end{aligned} \tag{13.4.3}$$

Let $K_m = f(t_i+\alpha_m h, p(t_i+\alpha_m h))$ for $1 \le m \le k$. If we substitute $t = t_i + \alpha_j h$ in (13.4.3), we get

$$p(t_i+\alpha_j h) = w_i + h\sum_{m=1}^{k} \beta_{j,m}\,K_m \tag{13.4.4}$$

for $1 \le j \le k$. Thus,

$$K_j = f\left(t_i+\alpha_j h, w_i + h\sum_{m=1}^{k} \beta_{j,m}\,K_m\right) \tag{13.4.5}$$

for $1 \le j \le k$.

If we now substitute $t = t_{i+1}$ in (13.4.3), we get

$$w_{i+1} = w_i + h\sum_{j=1}^{k} \gamma_j\,K_j\ . \tag{13.4.6}$$

(13.4.5) and (13.4.6) define the expected implicit Runge-Kutta method. ∎

Remark 13.4.12

1. Not all Runge-Kutta methods comes from collocation methods. For instance

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 2/3 & 1/3 & 1/3 \\ \hline & 1/4 & 3/4 \end{array} \quad \text{and} \quad \begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array}$$

   are two Runge-Kutta methods but there is a unique collocation method associated to the nodes $\alpha_1 = 0$ and $\alpha_2 = 2/3$.

2. The choice of $\beta_{j,m}$ for $1 \le m, j \le k$ is such that

$$\int_0^{\alpha_j} q(t)\,\mathrm{d}t = \sum_{m=1}^{k} \beta_{j,m}\,q(\alpha_m) \tag{13.4.7}$$

is true for all polynomials q of degree less than k because all polynomials of degree less than k can be written as
$$q(t) = \sum_{m=1}^{k} q(\alpha_m)\,\ell_m(t)$$
since we assume that the α_m are distinct. Similarly, we have
$$\int_0^1 q(t)\,\mathrm{d}t = \sum_{j=1}^{k} \gamma_j\, q(\alpha_j) \tag{13.4.8}$$
is true for polynomial q of degree less than k. Question 13.12 expands on this subject. In particular, the $\beta_{i,m}$ and γ_m are uniquely determined by (13.4.7) and (13.4.8).

♠

We state the next proposition in the context of an initial value problem in $\mathbb{R}^n$; namely, $f : [t_0, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$ with $n > 1$. The statement of this proposition is more interesting in this context despite the fact that we will only use it for $n = 1$.

Proposition 13.4.13 (Alekseev-Gröbner Lemma)

Let $\mathbf{y} : [a, b] \to \mathbb{R}^n$ be the solution of
$$\begin{aligned} \mathbf{y}'(t) &= f(t, \mathbf{y}(t)) \quad , \quad t_0 \le t \le t_f \\ \mathbf{y}(t_0) &= \mathbf{y}_0 \end{aligned}$$
where the function $f : [t_0, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. Suppose that $\mathbf{v} : [t_0, t_f] \to \mathbb{R}^n$ is a continuously differentiable function and $\mathbf{v}(t_0) = \mathbf{y}_0$. Then $\mathbf{v}$ satisfies
$$\mathbf{v}(t) = \mathbf{y}(t) + \int_{t_0}^{t} A(t, s, \mathbf{v}(s))\,\big(\mathbf{v}'(s) - f(s, \mathbf{v}(s))\big)\,\mathrm{d}s$$
for $t_0 \le t \le t_f$, where A is the Jacobian matrix with respect to $\mathbf{w}$ of the solution $\mathbf{u} = \mathbf{u}(t, s, \mathbf{w})$ of
$$\begin{aligned} \mathbf{u}'(t) &= f(t, \mathbf{u}(t)) \quad , \quad t \ge s \\ \mathbf{u}(s) &= \mathbf{w} \end{aligned}$$
for every $s \ge t_0$.

We will not prove this proposition. However, we will illustrate it for $n = 1$.

Example 13.4.14
Consider the initial value problem
$$\begin{aligned} y'(t) &= a y(t) \quad , \quad t_0 \le t \le t_f \\ y(t_0) &= y_0 \end{aligned}$$
Its solution is $y(t) = e^{a(t - t_0)} y_0$ for $t_0 \le t \le t_f$.

Suppose that $v : [t_0, t_f] \to \mathbb{R}$ is a continuously differentiable function such that $v(t_0) = y_0$. Consider the initial value problem

$$\begin{aligned} u'(t) &= au(t) \quad , \quad t \geq s \\ u(s) &= w \end{aligned}$$

Its solution is $u(t) = e^{a(t-s)}w$ for $t \geq s$.

According to Alekseev-Gröbner Lemma, we have

$$v(t) = e^{a(t-t_0)}y_0 + \int_{t_0}^{t} e^{a(t-s)}\,(v'(s) - av(s))\,\mathrm{d}s \tag{13.4.9}$$

for $t_0 \leq t \leq t_f$. This is a well known result because $v(t)$, the solution of

$$\begin{aligned} v'(t) &= av(t) + g(t) \quad , \quad t_0 \leq t \leq t_f \\ v(t_0) &= y_0 \end{aligned}$$

where $g(t) = v'(t) - av(t)$ for $t_0 \leq t \leq t_f$, is given by (13.4.9). ♣

Theorem 13.4.15

Let $q(t) = \prod_{j=1}^{k}(t - \alpha_j)$, where the α_j are the nodes given in Definition 13.4.10. Suppose that m is the largest integer such that $0 < m \leq k$ and

$$\int_0^1 q(t)\,t^j\,\mathrm{d}t = 0 \tag{13.4.10}$$

for $0 \leq j < m$. Then, the collocation method in Definition 13.4.10 is of order $k + m$ (Definition 13.3.1) if we assume that f is sufficiently continuously differentiable.

Proof.

From Alekseev-Gröbner lemma with t_0 replaced by t_i, t by t_{i+1} and $v(t)$ by the collocation polynomial $p(t)$ in Definition 13.4.10, we get

$$w_{i+1} - y(t_{i+1}) = \int_{t_i}^{t_{i+1}} g(s)\,\mathrm{d}s = h\int_0^1 g(t_i + sh)\,\mathrm{d}s \ , \tag{13.4.11}$$

where

$$g(s) = A(t_{i+1}, s, p(s))\,(p'(s) - f(s, p(s))) \ .$$

As we have seen at the beginning of Section 12.7, if we use the nodes α_j and the weight γ_j for $1 \leq j \leq k$ given in Theorem 13.4.11, we get from (13.4.11) that

$$w_{i+1} - y(t_{i+1}) = h\sum_{j=1}^{k}\gamma_j\,g(t_i + \alpha_j h) + h\,E_i \ ,$$

where E_i is the discretization error of the quadrature formula. However, in the Definition 13.4.10 of the collocation method, we have that

$$p'(t_i + \alpha_j h) - f(t_i + \alpha_j h, p(t_i + \alpha_j h)) = 0$$

for $1 \leq j \leq k$. Thus $w_{i+1} - y(t_{i+1}) = h\,E_i$. From (13.4.10) and Question 12.25, we have that the quadrature formula is exact for polynomials of degree less than $k+m$. It follows from Question 12.26 that there exist a constant $K = K(k,m,\gamma_j)$ such that

$$\begin{aligned}|E_i| &= \left| \int_0^1 g(t_i + sh)\,\mathrm{d}s - \sum_{j=1}^k \gamma_j\, g(t_i + \alpha_j h) \right| \leq K \max_{0\leq s\leq 1} \left| \frac{\mathrm{d}^{k+m}}{\mathrm{d}s^{k+m}} g(t_i + sh) \right| \\ &= K h^{k+m} \max_{0\leq s\leq 1} \left| g^{(k+m)}(t_i + sh) \right| \leq K h^{k+m} \max_{a\leq t\leq b} \left| g^{(k+m)}(t) \right|\end{aligned}$$

for $0 \leq i < N$. Let $\phi(t_i, w_i) = \sum_{j=1}^k \gamma_j K_j$ be the formula for the Runge-Kutta method provided by the collocation method. Then

$$\begin{aligned}\tau_{i+1}(h) &= \frac{y(t_{i+1}) - y(t_i)}{h} - \phi(t_i, y(t_i)) = \frac{(w_{i+1} - hE_i) - (w_i - hE_{i-1})}{h} - \phi(t_i, w_i - hE_{i-1}) \\ &= \underbrace{\frac{w_{i+1} - w_i}{h} - \phi(t_i, w_i)}_{=0} - E_i + E_{i-1} - h\frac{\partial \phi}{\partial w}(t_i, \xi_i) E_{i-1}\end{aligned}$$

for some ξ_i between w_i and $w_i - E_{i-1}$ is we assume that $|h| \leq 1$.

Since $E_i = O(h^{k+m})$ near the origin for all i, we may assume that there exists an interval $[c,d]$ such that w_i and $w_i - E_i$ are in $[c,d]$ for all i and all small h (i.e. for all partitions $a \leq t_0 < t_1 < \cdots < t_N = b$ with $t_{i+1} - t_i = h$ small enough). Since ϕ is composed of f and some of its partial derivatives, there exists a constant M such that $\left| \frac{\partial \phi}{\partial w}(t,w) \right| \leq M$ for all $(t,w) \in [a,b] \times [c,d]$. Hence

$$|\tau_{i+1}(h)| \leq \tau(h) = (2 + |h|M)\, K |h|^{k+m} \max_{a\leq t\leq b} \left| g^{(k+m)}(t) \right| = O(h^{k+m})$$

near the origin. Proving that the collocation method is of order at least $k+m$.

To prove that the collocation method is not of order greater than $k+m$, it suffices to apply the collocation method to

$$\begin{aligned} y'(t) &= (k+m+1)t^{k+m} \\ y(0) &= 0 \end{aligned}$$

∎

Corollary 13.4.16

Let $q(t) = \prod_{j=1}^k (t - \alpha_j)$, where the α_j are the nodes used in the collocation method given in Definition 13.4.10. Suppose that

$$\int_0^1 q(t)\, t^i \, \mathrm{d}t = 0$$

for $i = 0, 1, \ldots, k-1$. Then, the collocation method is of order $2k$.

Example 13.4.17 (**Gauss-Legendre Methods**)
Suppose that $q(t) = t - 1/2$, the Gauss-Legendre polynomial of degree 1, and $\alpha_1 = 1/2$. The previous Corollary is satisfied with $k = 1$. From Theorem 13.4.11, we get the Runge-Kutta method of order two associated to the Butcher array[2]

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$$

This is the Implicit Midpoint Rule.

Suppose that $q(t) = t^2 - t + 1/6$, the Gauss-Legendre polynomial of degree 2. $q(t) = (t-\alpha_1)(t-\alpha_2)$, where $\alpha_1 = (3-\sqrt{3})/6$ and $\alpha_2 = (3+\sqrt{3})/6$. The previous Corollary is satisfied with $k = 2$. From Theorem 13.4.11, we get the Runge-Kutta method of order four associated to the Butcher array

$$\begin{array}{c|cc} (3-\sqrt{3})/6 & 1/4 & (3-2\sqrt{3})/12 \\ (3+\sqrt{3})/6 & (3+2\sqrt{3})/12 & 1/4 \\ \hline & 1/2 & 1/2 \end{array}$$

♣

13.4.2 Derivation of Runge-Kutta Methods – Rooted Trees

In this section, we will use trees to derive Runge-Kutta methods. None of the results from graph theory will be proved. This could be the subject for another book. Moreover, we will consider initial value problems like (13.1.1) where $f : [t_o, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$ with $n > 1$. Namely, we will consider the initial value problem

$$\begin{aligned} \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t) &= f(t, \mathbf{y}(t)) \quad , \quad t_0 \le t \le t_f \\ \mathbf{y}(t_0) &= \mathbf{y}_0 \in \mathbb{R}^n \end{aligned} \tag{13.4.12}$$

where $f : [t_0, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$ and $n > 1$.

It is true that most of what we have said for initial value problems with $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ is also true for initial value problems with $f : [t_0, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$ and $n > 1$. But there are

[2] This is the case $k = 1$ in Theorem 13.4.11. We then have that $\ell_1(t) = 1$ for all t by definition. The empty product is defined to be 1, the neutral element for the multiplication, as the empty sum is defined to be 0, the neutral element for the addition.

some differences. One property that is influenced by the dimension of the space is the order of the method. As we will show later in this section, some Runge-Kutta methods do not have the same order in $\mathbb{R}$ than in $\mathbb{R}^n$ with $n > 1$.

Some good references on the subject of this section are [8, 19, 23, 24]. The proof of many of the results stated in this section can be found in those references. They also include good references to the publications on the rooted tree approach.

13.4.2.1 Elementary differentials

For simplicity and without too much lost of generality, we assume that f in (13.4.12) is independent of t.

Let $\mathcal{L}^m(\mathbb{R}^n, \mathbb{R}^n)$ be the space of multilinear mappings from $\mathbb{R}^n \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n$ (m times) to $\mathbb{R}^n$.

Definition 13.4.18

The **Frechet derivative** of degree n of $f : \mathbb{R}^n \to \mathbb{R}^n$ is the mapping $\mathrm{D}^m f : \mathbb{R}^n \to \mathcal{L}^m(\mathbb{R}^n, \mathbb{R}^n)$ defined by

$$\begin{aligned}
&\mathrm{D}^m f(\mathbf{y})(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_m) \\
&\quad = \sum_{i=1}^{n} \left(\sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \ldots \sum_{j_m=1}^{n} \frac{\partial^m f}{\partial y_{j_1} \partial y_{j_2} \ldots \partial y_{j_m}}(\mathbf{y})\, k_{1,j_1}\, k_{2,j_2} \ldots k_{m,j_m} \right) \mathbf{e}_i \ ,
\end{aligned}$$

where $\mathbf{k}_i = \begin{pmatrix} k_{i,1} & k_{i,2} & \ldots & k_{1,n} \end{pmatrix}^\top$.

Example 13.4.19
If $f : \mathbb{R}^2 \to \mathbb{R}^2$,

$$\mathrm{D}^2 f(\mathbf{y})(\mathbf{k}_1, \mathbf{k_2}) = \begin{pmatrix} \displaystyle\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \frac{\partial^2 f_1}{\partial y_{j_1} \partial y_{j_2}}(\mathbf{y})\, k_{1,j_1}\, k_{2,j_2} \\ \displaystyle\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \frac{\partial^2 f_2}{\partial y_{j_1} \partial y_{j_2}}(\mathbf{y})\, k_{1,j_1}\, k_{2,j_2} \end{pmatrix} .$$

♣

Definition 13.4.20

The **elementary differentials** of $f : \mathbb{R}^n \to \mathbb{R}^n$ and their **order** are defined recursively.

1. $f : \mathbb{R}^n \to \mathbb{R}^n$ is the only elementary differential of order 1.
2. if $g_1 : \mathbb{R}^n \to \mathbb{R}^n$, $g_2 : \mathbb{R}^n \to \mathbb{R}^n$, ..., $g_r : \mathbb{R}^n \to \mathbb{R}^n$ are r elementary differentials of f of order m_1, m_2, ..., m_r respectively, then $\mathrm{D}^r f(\cdot)(g_1(\cdot), g_2(\cdot), \ldots, g_r(\cdot)) : \mathbb{R}^n \to \mathbb{R}^n$, defined by

$$\mathrm{D}^r f(\mathbf{y})\big(g_1(\mathbf{y}), g_2(\mathbf{y}), \ldots, g_r(\mathbf{y})\big)$$

$$= \sum_{i=1}^{n}\left(\sum_{j_1=1}^{n}\sum_{j_2=1}^{n}\cdots\sum_{j_r=1}^{n}\frac{\partial^r f}{\partial y_{j_1}\partial y_{j_2}\ldots\partial y_{j_r}}(\mathbf{y})\, g_{1,j_1}(\mathbf{y})\, g_{2,j_2}(\mathbf{y}),\ldots g_{r,j_r}(\mathbf{y})\right)\mathbf{e}_i$$

for $\mathbf{y} \in \mathbb{R}^n$, is an elementary differential of order $1 + \sum_{i=1}^{r} m_i$ of $\mathbf{f}$.

For simplicity, $\mathrm{D}^r f(\cdot)(g_1(\cdot), g_2(\cdot), \ldots, g_r(\cdot))$ is denoted $\{g_1\ g_2\ \ldots\ g_r\}$.

The order of an elementary differential is not related to the degree of the Frechet derivatives of f that are used in the definition of the elementary differential.

Example 13.4.21
The elementary differential of $f : \mathbb{R}^n \to \mathbb{R}^n$ of order 2 is $\{f\} = \mathrm{D}f(\cdot)(f(\cdot))$ defined by

$$\mathrm{D}f(\mathbf{y})(f(\mathbf{y})) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{\partial f_i}{\partial y_j}(\mathbf{y}) f_j(\mathbf{y})\right)\mathbf{e}_i\ .$$

This is $\mathbf{y}''(t)$ if $\mathbf{y}'(t) = f(\mathbf{y}(t))$. This is the motivation for the definition of the order of an elementary differential. Only partial derivative of order 1 of f are used but it is associated to the second order derivative of $\mathbf{y}$ when $\mathbf{y}'(t) = f(\mathbf{y}(t))$. Note that $\mathrm{D}f \equiv \mathrm{D}^1 f$.

Here are two elementary differentials of $f : \mathbb{R}^n \to \mathbb{R}^n$ of order 2:
$\{\{f\}\} = \mathrm{D}f(\cdot)\big(\mathrm{D}f(\cdot)(f(\cdot))\big)$ defined by

$$\mathrm{D}f(\mathbf{y})\big(\mathrm{D}f(\mathbf{y})(f(\mathbf{y}))\big) = \sum_{i=1}^{n}\left(\sum_{k=1}^{n}\frac{\partial f_i}{\partial y_k}(\mathbf{y})\left(\sum_{j=1}^{n}\frac{\partial f_k}{\partial y_j}(\mathbf{y}) f_j(\mathbf{y})\right)\right)\mathbf{e}_i$$

for $\mathbf{y} \in \mathbb{R}^n$, and $\{f\ f\} = \mathrm{D}^2 f(\cdot)\big(f(\cdot), f(\cdot)\big)$ defined by

$$\mathrm{D}^2 f(\mathbf{y})\big(f(\mathbf{y}), f(\mathbf{y})\big) = \sum_{i=1}^{n}\left(\sum_{j_1=1}^{n}\sum_{j_2=1}^{n}\frac{\partial^2 f_i}{\partial y_{j_1}\partial y_{j_1}}(\mathbf{y}) f_{j_1}(\mathbf{y})\, f_{j_2}(\mathbf{y})\right)\mathbf{e}_i$$

for $\mathbf{y} \in \mathbb{R}^n$.

From now on, we will ignore the dependent variable $\mathbf{y}$ and write $\{\{f\}\} = \mathrm{D}f\,(\mathrm{D}f\,(f))$ and $\{f\ f\} = \mathrm{D}^2 f\,(f, f)$ to simplify the notation. ♣

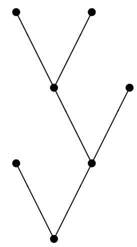
13.4.2.2 Rooted Trees

In this section, we briefly introduce some concepts about rooted trees without giving any proof. We only introduce the concepts that will provide the tools to compute elementary differentials. The proofs of the results mentioned in this section can be found in [23].

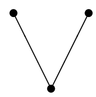
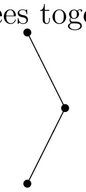
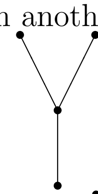
The easiest way to define the **rooted trees** is to give some examples of them.

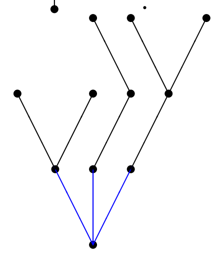
A rooted tree of order 1:

A rooted tree of order 2:

A rooted tree of order 7: 

We can combine rooted trees together to form another rooted tree.

Let $\tau_1 =$  , $\tau_2 =$  and $\tau_3 =$  .

The new rooted tree $[\tau_1\ \tau_2\ \tau_3]$ is defined by  . The three rooted trees were combined using the rooted tree in blue.

Remark 13.4.22
It is interesting to know that if a_i is the number of rooted trees of order i, then

$$a_1 + a_2 u + a_2 u^2 + a_3 u^3 + \ldots = (1-u)^{-a_1}\,(1-u^2)^{-a_2}\,(1-u^3)^{-a_3}\ldots$$

♠

Definition 13.4.23

Let τ be a rooted tree. We define the following values associated to the rooted tree τ.

1. $r(\tau)$ is the **order** of τ.
2. $\sigma(\tau)$ is the **symmetry** of τ.
3. $\gamma(\tau)$ is the **density** of τ.
4. $\alpha(\tau)$ is the number of "distinct ways of numbering the nodes" of τ such that the numbers increase along the branches if we start from the root.

The order, symmetry and density are defined recursively.
If τ is a rooted tree of order one, then $r(\tau) = \sigma(\tau) = \gamma(\tau) = 1$.
If

$$\tau = [\underbrace{\tau_1\ \tau_1\ \ldots\ \tau_1}_{n_1 \text{ times}}\ \underbrace{\tau_2\ \tau_2\ \ldots\ \tau_2}_{n_2 \text{ times}} \ldots \underbrace{\tau_q\ \tau_q\ \ldots\ \tau_q}_{n_q \text{ times}}]\ ,$$

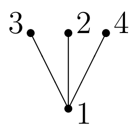
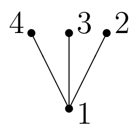
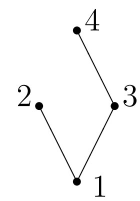
then

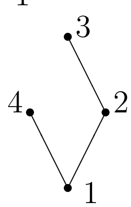
$$r(\tau) = 1 + n_1\, r(\tau_1) + n_2\, r(\tau_2) + \ldots + n_q\, r(\tau_q)$$

$$\sigma(\tau) = n_1!\,n_2!\dots n_q!\,(\sigma(\tau_1))^{n_1}(\sigma(\tau_2))^{n_2}\dots(\sigma(\tau_q))^{n_q}$$
$$\gamma(\tau) = r(\tau)\,(\gamma(\tau_1))^{n_1}\,(\gamma(\tau_2))^{n_2}\dots(\gamma(\tau_q))^{n_q}$$

Remark 13.4.24

We explain the expression "distinct ways of numbering the nodes" of a rooted tree with the help of some examples. The following numberings are not considered to be distinct:

 and  . However, the following numberings are distinct: 

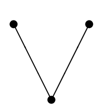
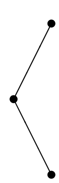
and  ♠

The number of distinct ways of numbering the nodes of a rooted tree τ is given by the following theorem.

Theorem 13.4.25

If τ is a rooted tree, then
$$\alpha(\tau) = \frac{r(\tau)!}{\sigma(\tau)\gamma(\tau)}\ .$$

We give in Table 13.1 the order, symmetry, density and number of distinct ways of numbering the nodes for some of the basic rooted trees.

rooted tree	name	order	symmetry	density	numbering
	τ	1	1	1	1
	$[\tau]$	2	1	2	1
	$[\tau\ \tau]$	3	2	3	1
	$[[\tau]]$	3	1	6	1

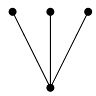
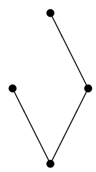
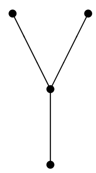
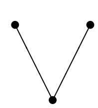
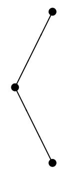
rooted tree	name	order	symmetry	density	numbering
	$[\tau\ \tau\ \tau]$	4	6	4	1
	$[\tau\ [\tau]]$	4	1	8	3
	$[[\tau\ \tau]]$	4	2	12	1
	$[[[\tau]]]$	4	1	24	1

Table 13.1: The order, symmetry, density and number of distinct ways of numbering the rooted trees of order 1 to 4 inclusively.

13.4.2.3 Relation Between Elementary Differentials and Rooted Trees

We define a mapping F which associates to each rooted tree τ an elementary differential $F(\tau)$ of a function f. The easiest way to explain how F associates elementary differentials to rooted trees is to give some examples.

rooted tree τ	elementary differential $F(\tau)$
	f
	$\{f\} = \mathrm{D}f\,(f)$
	$\{f\ f\} = \mathrm{D}^2 f\,(f, f)$

τ	$F(\tau)$
	$\{\{f\}\} = \mathrm{D}f\,(\mathrm{D}f\,(f))$
$\vdots$	$\vdots$

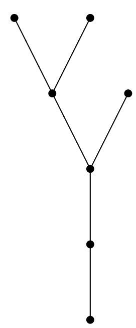
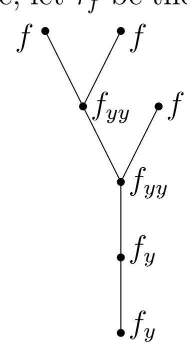
The following proposition follows easily from the definitions.

Proposition 13.4.26

1. If τ is the rooted tree associated to the elementary differential g of f (i.e. $g = F(\tau)$), then g and τ have the same order.

2. If $g_1, g_2, \ldots, g_s$ are elementary differentials of f associated to the rooted trees $\tau_1, \tau_2, \ldots, \tau_s$ respectively, then the elementary differential $\{g_1\ g_2\ \ldots\ g_s\} = \mathrm{D}^s f\,(g_1, g_2, \ldots, g_s)$ is associated to the rooted tree $[\tau_1\ \tau_2\ \ldots\ \tau_s]$.

Example 13.4.27
If τ_1, τ_2 and τ_3 are the rooted trees defined at the beginning of Section 13.4.2.2, we have that $g_1 = \{f\ f\}$ is associated to τ_1, $g_2 = \{\{f\}\}$ is associated to τ_2 and $g_3 = \{\{f\ f\}\}$ is associated to τ_3. Thus $\{g_1\ g_2\ g_3\}$ is associated to the rooted tree $[\tau_1\ \tau_2\ \tau_3]$. ♣

Remark 13.4.28
If we have $f : \mathbb{R} \to \mathbb{R}$, the relation between rooted trees and elementary differentials is simple. For instance, let τ_f be the rooted tree . We associate to this rooted tree the rooted tree 

. Let τ be the rooted tree of order one. Then $\tau_f = [[[\tau\ [\tau\ \tau]]]]$ and the elementary differential of f associate to $\tau_f =$ is $\{\{\{f\ \{f\ f\}\}\}\} = f_{yy}^2\, f_y^2\, f^3$. We only have to multiply the derivatives that appear in the second rooted tree. ♠

13.4.2.4 Runge-Kutta Methods

We now use rooted trees and elementary differentials to develop Runge-Kutta methods.

Theorem 13.4.29

We consider the initial value problem (13.4.12) where f does not depend on the time. Thus, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{y}'(t) = f(\mathbf{y}(t))$. We have that

$$\mathbf{y}^{(q)}(t) = \sum_{r(\tau)=q} \alpha(\tau) F(\tau) \ ,$$

where $F(\tau)$ is evaluated at $\mathbf{y}(t)$.

Example 13.4.30

$$\begin{aligned}\mathbf{y}^{(4)} &= \{f\ f\ f\} + 3\{f\ \{f\}\} + \{\{f\ f\}\} + \{\{\{f\}\}\} \\ &= \mathrm{D}^3 f(f,f,f) + 3\,\mathrm{D}^2 f(f, \mathrm{D}f\,(f)) + \mathrm{D}f(\mathrm{D}^2 f(f,f)) + \mathrm{D}f\,(\mathrm{D}f\,(\mathrm{D}f\,(f))) \ .\end{aligned}$$

If we have $f : \mathbb{R} \to \mathbb{R}$, we then get

$$y^{(4)} = f_{yyy} f^3 + 3 f_{yy} f_y f^2 + f_{yy} f_y f^2 + f_y^3 f = f_{yyy} f^3 + 4 f_{yy} f_y f^2 + f_y^3 f \ .$$

♣

From now on, we consider the general definition of the Runge-Kutta methods given in Definition 13.4.1.

We define a mapping Ψ which associates to each rooted tree τ a sum $\Psi(\tau)$ constructed from some elements of the Butcher array

$$\begin{array}{c|cccc} \alpha_1 & \beta_{1,1} & \beta_{1,2} & \ldots & \beta_{1,s} \\ \alpha_2 & \beta_{2,1} & \beta_{2,2} & \ldots & \beta_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_s & \beta_{s,1} & \beta_{s,2} & \ldots & \beta_{s,s} \\ \hline & \gamma_1 & \gamma_2 & \ldots & \gamma_s \end{array}$$

If τ is a rooted tree, $\Psi(\tau) = \psi_{s+1}(\tau)$, where the function ψ_{s+1} is defined recursively as follows. Let $\beta_{s+1,j} = \gamma_j$ for $1 \leq j \leq s$.

1. If τ is the rooted tree of order 1, then $\psi_j(\tau) \equiv \sum_{k=1}^{s} \beta_{j,k}$ for $1 \leq j \leq s+1$.
2. If $\tau = [\tau_1\ \tau_2\ \ldots\ \tau_q]$, where τ_1, τ_2, ..., τ_q are rooted trees, then

$$\psi_i(\tau) \equiv \sum_{j=1}^{s} \beta_{i,j}\, \psi_j(\tau_1)\, \psi_j(\tau_2) \ldots \psi_j(\tau_q)$$

for $1 \leq i \leq s+1$.

There is an easy way to compute $\Psi(\tau)$. We illustrate it with some examples in Table 13.3 below.

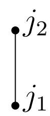
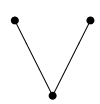
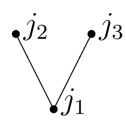
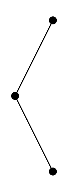
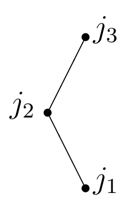
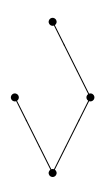
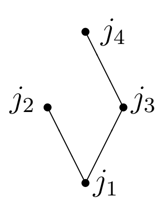
τ		$\Psi(\tau)$
		$\sum_{j=1}^{s} \beta_{s+1,j} = \sum_{j=1}^{s} \gamma_j = 1$
		$\sum_{j_1=1}^{s} \left(\beta_{s+1,j_1} \sum_{j_2=1}^{s} \beta_{j_1,j_2} \right)$ $= \sum_{j_1=1}^{s} \gamma_{j_1} \alpha_{j_1}$
		$\sum_{j_1=1}^{s} \left(\beta_{s+1,j_1} \left(\sum_{j_2=1}^{s} \beta_{j_1,j_2} \right) \left(\sum_{j_3=1}^{s} \beta_{j_1,j_3} \right) \right)$ $= \sum_{j_1=1}^{s} \gamma_{j_1} \alpha_{j_1}^2$
		$\sum_{j_1=1}^{s} \left(\beta_{s+1,j_1} \left(\sum_{j_2=1}^{s} \beta_{j_1,j_2} \left(\sum_{j_3=1}^{s} \beta_{j_2,j_3} \right) \right) \right)$ $= \sum_{j_1=1}^{s} \left(\gamma_{j_1} \left(\sum_{j_2=1}^{s} \beta_{j_1,j_2} \alpha_{j_2} \right) \right)$
		$\sum_{j_1=1}^{s} \left(\beta_{s+1,j_1} \left(\sum_{j_2=1}^{s} \beta_{j_1,j_2} \right) \left(\sum_{j_3=1}^{s} \beta_{j_1,j_3} \left(\sum_{j_4=1}^{s} \beta_{j_3,j_4} \right) \right) \right)$ $= \sum_{j_1=1}^{s} \left(\gamma_{j_1} \alpha_{j_1} \left(\sum_{j_3=1}^{s} \beta_{j_1,j_3} \alpha_{j_3} \right) \right)$

Table 13.3: Computation of $\Psi(\tau)$ for some basic tress

We define the function

$$\mathbf{Y}_i(z) = \mathbf{y}(t_i) + \mathbf{y}'(t_i)\, z + \frac{1}{2!}\mathbf{y}''(t_i)\, z^2 + \frac{1}{3!}\mathbf{y}^{(3)}(t_i)\, z^3 + \ldots$$

We have that $\mathbf{y}_{i+1} = \mathbf{y}(t_{i+1}) = \mathbf{Y}(h)$. We are assuming that the Taylor series of $\mathbf{y}$ at t_i has a radius of convergence greater than h. From the Runge-Kutta method, we also define the function

$$\mathbf{W}_i(z) = \mathbf{w}_i + z \sum_{j=1}^{s} \gamma_j K_j \ ,$$

where

$$K_j = f\left(t_i + \alpha_j z, \mathbf{w}_i + z\sum_{k=1}^{s} \beta_{j,k} K_k\right)$$

for $1 \le j \le s$. We have that $\mathbf{w}_{i+1} = \mathbf{W}_i(h)$.

Our goal is to match the series expansions of $\mathbf{W}_i$ and $\mathbf{Y}_i$ near the origin to generate Runge-Kutta methods of high order. To be rigorous, the only think that we need is Taylor polynomial expansions of $\mathbf{Y}_i$ and $\mathbf{W}_i$ of degree sufficiently large. We use the series expansions with a large enough radius of convergence to simplify the presentation.

Proposition 13.4.31

We have that

$$\frac{\mathrm{d}^q \mathbf{W}_i}{\mathrm{d}z^q}(0) = \sum_{r(\tau)=q} \alpha(\tau)\,\gamma(\tau)\,\Psi(\tau)\,F(\tau)\ ,$$

where $F(\tau)$ is evaluated at $\mathbf{W}_i(0) = \mathbf{w}_i$

Theorem 13.4.32

A Runge-Kutta method is of order p if $\Psi(\tau) = 1/\gamma(\tau)$ for all rooted trees τ of order less than or equal to p, and $\Psi(\tau) \neq 1/\gamma(\tau)$ for at least one rooted tree of order $p+1$.

Proof.
We have

$$\mathbf{Y}_i(z) = \mathbf{y}_i + \mathbf{y}'(t_i)\,z + \frac{1}{2!}\mathbf{y}''(t_i)\,z^2 + \frac{1}{3!}\mathbf{y}^{(3)}(t_i)\,z^3 + \ldots = \mathbf{y}_i + \sum_{q=1}^{\infty}\frac{1}{q!}\left(\sum_{r(\tau)=q}\alpha(\tau)\,F(\tau)\right)z^q\ ,$$

where $F(\tau)$ is evaluated at $\mathbf{y}_i$, and

$$\mathbf{W}_i(z) = \mathbf{w}_i + \frac{\mathrm{d}\mathbf{W}_i}{\mathrm{d}z}(0)\,z + \frac{1}{2}\,\frac{\mathrm{d}^2\mathbf{W}_i}{\mathrm{d}z^2}(0)\,z^2 + \ldots = \mathbf{w}_i + \sum_{q=1}^{\infty}\frac{1}{q!}\left(\sum_{r(\tau)=q}\alpha(\tau)\,\gamma(\tau)\,\Psi(\tau)\,F(\tau)\right)z^q\ ,$$

where $F(\tau)$ is evaluated at $\mathbf{W}_i(0) = \mathbf{w}_i$.

To compute the local truncation error, we make use of the localisation assumption $\mathbf{y}_i = \mathbf{w}_i$.

Hence, if $\gamma(\tau)\,\Psi(\tau) = 1$ for all rooted trees of order less than or equal to p, then the series expansions of $\mathbf{Y}_i$ and $\mathbf{W}_i$ have identical terms in z^q for $q \le p$. We thus have that

$$\mathbf{Y}(z) = \mathbf{W}_i(z) + \sum_{q=p+1}^{\infty}\left(\frac{1}{q!}\sum_{r(\tau)=q}\alpha(\tau)\,(1-\gamma(\tau)\,\Psi(\tau))\,F(\tau)\right)z^q\ ,$$

where $F(\tau)$ is evaluated at $\mathbf{y}_i = \mathbf{w}_i$.

Hence, the local truncation error is

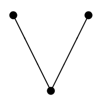
$$\begin{aligned}\tau_{i+1}(h) &= \frac{\mathbf{y}(t_{i+1}) - \mathbf{y}(t_i)}{h} - \phi(t_i, \mathbf{y}(t_i)) = \frac{\mathbf{Y}_i(h) - \mathbf{W}_i(h)}{h} \\ &= \frac{h^p}{(p+1)!}\sum_{r(\tau)=p+1}\alpha(\tau)\,(1-\gamma(\tau)\,\Psi(\tau))\,F(\tau) + O(h^{p+1})\ ,\end{aligned} \tag{13.4.13}$$

where $F(\tau)$ is evaluated at $\mathbf{y}_i = \mathbf{w}_i$. Therefore, the local truncation error is of order at least p. It will be exactly of order p if there exists a rooted tree of order $p+1$ such that $\gamma(\tau)\,\Psi(\tau) \neq 1$ (Remark 13.4.35 below). ∎

Example 13.4.33
We consider the 3-stage explicit Runge-Kutta methods. Since the methods are explicit, we have $\alpha_1 = 0$ and $\beta_{i,j} = 0$ for $j \geq i$. We look for methods of order at least 3.

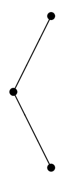
For the rooted tree τ of order one, we get from $\Psi(\tau) = 1/\gamma(\tau)$ that $1 = \displaystyle\sum_{j=1}^{3} \gamma_j$.

For the rooted tree τ of order two, we get from $\Psi(\tau) = 1/\gamma(\tau)$ that $\dfrac{1}{2} = \displaystyle\sum_{j=2}^{3} \alpha_j\,\gamma_j$ since $\alpha_1 = 0$.

For the tree $\tau =$  of order three, we get from $\Psi(\tau) = 1/\gamma(\tau)$ that

$$\frac{1}{3} = \sum_{i=1}^{3} \gamma_i \left(\sum_{j=1}^{3} \beta_{i,j}\right)\left(\sum_{k=1}^{3} \beta_{i,k}\right) = \sum_{i=1}^{3} \gamma_i \alpha_i^2 = \sum_{i=2}^{3} \gamma_i \alpha_i^2$$

since $\alpha_1 = 0$.

For the tree $\tau =$  of order three, we get from $\Psi(\tau) = 1/\gamma(\tau)$ that

$$\frac{1}{6} = \sum_{i=1}^{3} \gamma_i \left(\sum_{j=1}^{3} \beta_{i,j}\left(\sum_{k=1}^{3} \beta_{j,k}\right)\right) = \sum_{i=1}^{3} \gamma_i \left(\sum_{j=1}^{3} \beta_{i,j}\,\alpha_j\right) = \gamma_3\,\beta_{3,2}\,\alpha_2$$

since $\alpha_1 = 0$ and $\beta_{i,j} = 0$ for $j \geq i$.

Two possible Butcher arrays that satisfy $\gamma_1+\gamma_2+\gamma_3 = 1$, $\alpha_2\gamma_2+\alpha_3\gamma_3 = 1/2$ and $\gamma_3\beta_{3,2}\alpha_2 = 1/6$ are:

Heun's method:

$$\begin{array}{c|ccc}
0 & & & \\
1/3 & 1/3 & & \\
2/3 & 0 & 2/3 & \\
\hline
 & 1/4 & 0 & 3/4
\end{array}$$

Kutta's method of order three:

$$\begin{array}{c|ccc}
0 & & & \\
1/2 & 1/2 & & \\
1 & -1 & 2 & \\
\hline
 & 1/6 & 2/3 & 1/6
\end{array}$$

These method are therefore of order at least three. The reader can verify that, for each of these methods, there is a rooted three of order four τ such that $\Psi(\tau) \neq 1/\gamma(\tau)$. So the method are of order three. ♣

Example 13.4.34
We consider the 2-stage implicit Runge-Kutta methods of order four. We have the relations

$$\begin{aligned}\alpha_1 &= \beta_{1,1} + \beta_{1,2} \\ \alpha_2 &= \beta_{2,1} + \beta_{2,2}\end{aligned}$$

from the definition of the Runge-Kutta methods. From Theorem 13.4.32, we also have the following relations.

From the rooted tree of order one, we get $1 = \gamma_1 + \gamma_2$.

From the rooted tree of order two, we get $\displaystyle \frac{1}{2} = \sum_{j=1}^{2} \gamma_j \alpha_j$.

From the two rooted trees of order three, we get $\displaystyle \frac{1}{3} = \sum_{j=1}^{2} \gamma_j \alpha_j^2$ and $\displaystyle \frac{1}{6} = \sum_{j=1}^{2}\sum_{k=1}^{2} \gamma_j \beta_{j,k}\alpha_k$.

From the four rooted trees of order four, we get $\displaystyle \frac{1}{4} = \sum_{j=1}^{2} \gamma_j \alpha_j^3$, $\displaystyle \frac{1}{8} = \sum_{j=1}^{2}\sum_{k=1}^{2} \gamma_j \alpha_j \beta_{j,k}\alpha_k$, $\displaystyle \frac{1}{12} = \sum_{j=1}^{2}\sum_{k=1}^{2} \gamma_j \beta_{j,k}\alpha_k^2$ and $\displaystyle \frac{1}{24} = \sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{l=1}^{2} \gamma_j \beta_{j,k}\beta_{k,l}\alpha_l$.

Miraculously, there is a unique solution (modulo conjugacy) of all these equations. We get the Butcher array

$$\begin{array}{c|cc} (3-\sqrt{3})/6 & 1/4 & (3-2\sqrt{3})/12 \\ (3+\sqrt{3})/6 & (3+2\sqrt{3})/12 & 1/4 \\ \hline & 1/2 & 1/2 \end{array}$$

that we have already found in Example 13.4.17, using collocation methods. ♣

Remark 13.4.35
We show that the 4-stage Runge-Kutta method given by the Butcher array

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ -1 & 1/2 & -3/2 & & \\ 1 & 0 & 4/3 & -1/3 & \\ \hline & 1/6 & 2/3 & 0 & 1/6 \end{array}$$

is of order 4 if $f : \mathbb{R} \to \mathbb{R}$ and of order 3 if $f : \mathbb{R}^n \to \mathbb{R}^n$ with $n > 1$.

In the proof of Theorem 13.4.32, we use the conditions $\Psi(\tau)\gamma(\tau) = 1$ for all rooted trees τ of order $q \leq p$ to ensure that

$$\sum_{r(\tau)=q} \alpha(\tau)\, F(\tau) = \sum_{r(\tau)=q} \alpha(\tau)\,\gamma(\tau)\,\Psi(\tau)\, F(\tau) \tag{13.4.14}$$

for $q \le p$. This was sufficient to ensure that the coefficient of coefficient of z^q in $\mathbf{Y}_i(z)$ was equal to the coefficient of z^q in $\mathbf{W}_i(z)$. However, for $n = 1$, the condition $\Psi(\tau)\gamma(\tau) = 1$ for all rooted trees τ of order q is not always necessary to satisfy (13.4.14).

We leave it to the reader to verify that, for the 4-stage Runge-Kutta method above, the condition $\Psi(\tau)\gamma(\tau) = 1$ for all rooted trees τ of order $q \le 3$ is satisfied. However, this is not true for $q = 4$. Despite that, the 4-stage Runge-Kutta method is of order four for $n = 1$ but not for $n > 1$.

Let $\tau_1 = [[\tau\ \tau]]$ and $\tau_2 = [\tau\ [\tau]]$ where τ is the tree of order one. τ_1 and τ_2 are two trees of order four (Table 13.1).

If $f : \mathbb{R} \to \mathbb{R}$, then

$$F(\tau_1) = \{\{f\ f\}\} = \mathrm{D} f(\mathrm{D}^2 f(f, f))) = f_{yy}\, f_y\, f^2$$

and

$$F(\tau_2) = \{f\ \{f\}\} = \mathrm{D}^2 f(f, \mathrm{D} f(f)) = f_{yy}\, f_y\, f^2\ .$$

Since $F(\tau_1) = F(\tau_2)$, we can replace $\Psi(\tau_j)\gamma(\tau_j) = 1$ for $j = 1$ and 2 by

$$\alpha(\tau_1)\gamma(\tau_1)\Psi(\tau_1) + \alpha(\tau_2)\gamma(\tau_2)\Psi(\tau_2) = \alpha(\tau_1) + \alpha(\tau_1)$$

with $\Psi(\tau)\gamma(\tau) = 1$ for the other rooted trees τ of order four. This ensures that the coefficient of z^4 in $\mathbf{Y}_i$ is still equal to the coefficient of z^4 in $\mathbf{W}_i$.

The condition $\alpha(\tau_1)\gamma(\tau_1)\Psi(\tau_1) + \alpha(\tau_2)\gamma(\tau_2)\Psi(\tau_2) = \alpha(\tau_1) + \alpha(\tau_1)$, instead of the two conditions $\Psi(\tau_j)\gamma(\tau_j) = 1$ for $j = 1$ and 2, was used to obtain the 4-stage Runge-Kutta method above. We leave it to the reader to verify that the 4-stage Runge-Kutta method above verify this condition.

When $f : \mathbb{R}^n \to \mathbb{R}^n$ with $n > 1$, the relation $F(\tau_1) = F(\tau_2)$ is not necessary true and so the condition $\alpha(\tau_1)\gamma(\tau_1)\Psi(\tau_1)+\alpha(\tau_2)\gamma(\tau_2)\Psi(\tau_2) = \alpha(\tau_1)+\alpha(\tau_1)$ may not guarantee that the coefficient of z^4 in $\mathbf{Y}_i$ is equal to the coefficient of z^4 in $\mathbf{W}_i$.

As a simple example for $F(\tau_1) \ne F(\tau_2)$, consider the initial value problem

$$\begin{aligned} &y'(t) = f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ &y(t_0) = y_0 \end{aligned} \tag{13.4.15}$$

where $f : \mathbb{R}^2 \to \mathbb{R}$. We can rewrite this initial value problem as a system

$$\begin{aligned} &\mathbf{z}'(t) = \tilde{f}(\mathbf{z}(t)) \quad , \quad t_0 \le s \le t_f \\ &\mathbf{z}(t_0) = \mathbf{z}_0 \end{aligned}$$

where $\tilde{f}(\mathbf{z}) = \begin{pmatrix} f(z_2, z_1) \\ 1 \end{pmatrix}$ and $\mathbf{z}_0 = \begin{pmatrix} y_0 \\ t_0 \end{pmatrix}$. Hence,

$$F(\tau_1) = \{\{\tilde{f}\ \tilde{f}\}\} = \begin{pmatrix} f_{z_1}\,(f_{z_2 z_2} + 2f\, f_{z_1 z_2} + f^2\, f_{z_1 z_1}) \\ 0 \end{pmatrix}$$

and

$$F(\tau_2) = \{\tilde{f}\ \{\tilde{f}\}\} = \begin{pmatrix} (f_{z_2 z_1} + f\, f_{z_1 z_1})\,(f_{z_2} + f\, f_{z_1}) \\ 0 \end{pmatrix}$$

are generally different.

Note: This also shows that the 4-stage Runge-Kutta method above is only of order three for the initial value problem (13.4.15). ♠

Remark 13.4.36
Consider the following three statements about a fixed Runge-Kutta method applied to the initial value problem (13.4.12).

A The method is of order p with $f : \mathbb{R}^n \to \mathbb{R}^n$, $n > 1$. Note that f does not depend on time.

B The method is of order p with $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Note that f depends on time.

C The method is of order p with $f : \mathbb{R} \to \mathbb{R}$. Note that f does not depend on time.

It has been proved that

1. $A \Leftrightarrow B \Leftrightarrow C$ if $1 \leq p \leq 3$.
2. $A \Leftrightarrow B \Rightarrow C$ and $C \not\Rightarrow B$ if $p = 4$.
3. $A \Rightarrow B \Rightarrow C$, $C \not\Rightarrow B$ and $B \not\Rightarrow A$ if $p > 4$.

♠

13.4.2.5 Maximal Order of Explicit Runge-Kutta Methods

Theorem 13.4.37

An s-stage explicit Runge-Kutta method cannot be of order greater than s.

Proof.
Let τ be the rooted tree of order one. Consider the rooted tree τ_p of order p defined by

$$\tau_p = \underbrace{[[\ldots[}_{p-1 \text{ times}}\ \tau\ \ ]\ldots]]\ .$$

We have that $\gamma(\tau_p) = p!$ and

$$\Psi(\tau_p) = \sum_{j_1=1}^{s} \sum_{j_2=1}^{s} \ldots \sum_{j_p=1}^{s} \gamma_{j_1}\, \beta_{j_1,j_2} \ldots \beta_{j_{p-1},j_p}\ .$$

Since $\beta_{i,j} = 0$ for $j \geq i$, $\Psi(\tau_p) = 0$ unless $s \geq j_1 > j_2 > \ldots > j_p$. This is possible only if $s \geq p$. So, for $p > s$, $\Psi(\tau_p) = 0$ and we cannot get $1 = \gamma(\tau_p)\Psi(\tau_p)$. ∎

We have an even stronger result for the 5-stage explicit Runge-Kutta methods.

Theorem 13.4.38

There is no 5-stage explicit Runge-Kutta method of order five.

13.4.3 Variable Step-Size Methods

Up until now, we have only considered methods with equally spaced mesh points t_i for $i = 0$, 1, 2, ..., N. It will be advantageous to have some control on the **step-size** (i.e. the distance) between two consecutive mesh points. A large step-size could be used on the portions of the interval $[t_0, t_f]$ where the solution y of (13.1.1) varies slowly and a small step-size could be used on the portions of the interval $[t_0, t_f]$ where the solution y of (13.1.1) varies rapidly.

A method often used to control the step-size between each pair of mesh points is the **Runge-Kutta-Fehlberg method**. Let

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + h\phi(t_i, w_i) \end{aligned} \tag{13.4.16}$$

be a Runge-Kutta method of order four and

$$\begin{aligned} \tilde{w}_i &= w_i \\ \tilde{w}_{i+1} &= \tilde{w}_i + h\tilde{\phi}(t_i, \tilde{w}_i) \end{aligned} \tag{13.4.17}$$

be a Runge-Kutta method of order five. The functions ϕ and $\tilde{\phi}$ associated to the Runge-Kutta-Fehlberg method will be given below.

Let $\tau_{i+1}(h)$ be the local truncation error for the Runge-Kutta method of order four (13.4.16). Combining the Runge-Kutta methods of orders four and five, (13.4.16) and (13.4.17) respectively, we can determine the step-size h between t_i and t_{i+1} such that $\tau_{i+1}(h) < \epsilon$ for a ϵ given.

The Runge-Kutta-Fehlberg method can be summarized as follows:

Algorithm 13.4.39 (Runge-Kutta-Fehlberg Method)

1. $w_0 = y_0$.
2. Stop if $t_i = t_f$.
3. Suppose that w_i is an approximation of $y_i = y(t_i)$ and $h > 0$ is given. Compute a first approximation w_{i+1} of y_{i+1} using (13.4.16) and a second approximation $\tilde{w}_{i+1}$ of y_{i+1} using (13.4.17) with $\tilde{w}_i = w_i$.
4. If $|(\tilde{w}_{i+1} - w_{i+1})/h| < \epsilon$, accept w_{i+1} as an approximation of $y_{i+1} = y(t_i + h)$. Substitute h by qh where $q = |\epsilon h/(\tilde{w}_{i+1} - w_{i+1})|^{1/4}$.
5. If $|(\tilde{w}_{i+1} - w_{i+1})/h| \geq \epsilon$, reject w_{i+1} as an approximation of $y(t_i + h)$. Substitute h by qh where $q = |\epsilon\, h/(\tilde{w}_{i+1} - w_{i+1})|^{1/4}$.

6. If $h > t_f - t_i$, replace h by $t_f - t_i$.
7. If w_{i+1} has been accepted, go back to 2 with i replaced by $i+1$ and the new value of h. If w_{i+1} has been rejected, go back to 2 with i again and the new smaller value of h.

The function $\phi(t_i, w_i)$ in (13.4.16) is defined by

$$\phi(t_i, w_i) = \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5$$

and the function $\tilde{\phi}(t_i, \tilde{w}_i)$ in (13.4.17) (recall that $w_i = \tilde{w}_i$) is defined by

$$\tilde{\phi}(t_i, w_i) = \frac{16}{135}K_1 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6 ,$$

where

$$\begin{aligned}
K_1 &= f\left(t_i, w_i\right) , \\
K_2 &= f\left(t_i + \frac{h}{4}, w_i + \frac{h\,K_1}{4}\right) , \\
K_3 &= f\left(t_i + \frac{3h}{8}, w_i + \frac{3h\,K_1}{32} + \frac{9h\,K_2}{32}\right) , \\
K_4 &= f\left(t_i + \frac{12h}{13}, w_i + \frac{1932h\,K_1}{2197} - \frac{7200h\,K_2}{2197} + \frac{7296h\,K_3}{2197}\right) , \\
K_5 &= f\left(t_i + h, w_i + \frac{439h\,K_1}{216} - 8h\,K_2 + \frac{3680h\,K_3}{513} - \frac{845h\,K_4}{4104}\right)
\end{aligned}$$

and

$$K_6 = f\left(t_i + \frac{h}{2}, w_i - \frac{8h\,K_1}{27} + 2h\,K_2 - \frac{3544h\,K_3}{2565} + \frac{1859h\,K_4}{4104} - \frac{11h\,K_5}{40}\right) .$$

Both Runge-Kutta methods can be summarized in the following Butcher array

$$\begin{array}{r|rrrrrr}
0 & & & & & & \\
1/4 & 1/4 & & & & & \\
3/8 & 3/32 & 9/32 & & & & \\
12/13 & 1932/2197 & -7200/2197 & 7296/2197 & & & \\
1 & 439/216 & -8 & 3680/513 & -845/4104 & & \\
1/2 & -8/27 & 2 & -3544/2565 & 1859/4104 & -11/40 & \\
\hline
 & 25/216 & 0 & 1408/2565 & 2197/4104 & -1/5 & \\
\hline
 & 16/135 & 0 & 6656/12825 & 28561/56430 & -9/50 & 2/55
\end{array}$$

Remark 13.4.40

A non-rigorously justification of the Runge-Kutta-Fehlberg method is as follows. Let $\tilde{\tau}_{i+1}(h)$

be the local truncation error for the Runge-Kutta method of order five (13.4.17). Suppose that $y_i \approx w_i = \tilde{w}_i$, then

$$y_{i+1} - w_{i+1} = y_{i+1} - w_i - h\phi(t_i, w_i) \approx y_{i+1} - y_i - h\phi(t_i, y_i) = h\tau_{i+1}(h) \ .$$

Similarly,

$$y_{i+1} - \tilde{w}_{i+1} \approx h\tilde{\tau}_{i+1}(h) \ .$$

Hence,

$$\begin{aligned}\tau_{i+1}(h) &\approx \frac{1}{h}\left(y_{i+1} - w_{i+1}\right) = \frac{1}{h}\left(y_{i+1} - \tilde{w}_{i+1} + \tilde{w}_{i+1} - w_{i+1}\right) \\ &= \frac{1}{h}\left(y_{i+1} - \tilde{w}_{i+1}\right) + \frac{1}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right) \approx \tilde{\tau}_{i+1}(h) + \frac{1}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right) \ .\end{aligned}$$

Since $\tau_{i+1}(h) = O(h^4)$ and $\tilde{\tau}_{i+1}(h) = O(h^5)$, $(\tilde{w}_{i+1} - w_{i+1})/h$ is the dominant term on the right hand side for h small. Thus, we may assume that

$$\tau_{i+1}(h) \approx \frac{1}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right) \ . \tag{13.4.18}$$

If $|(\tilde{w}_{i+1} - w_{i+1})/h| < \epsilon$, we may assume that $|\tau_{i+1}(h)| < \epsilon$. Therefore, w_{i+1} is an acceptable approximation of $y(t_i + h)$.

If $|(\tilde{w}_{i+1} - w_{i+1})/h| \geq \epsilon$, then w_{i+1} is probably not an acceptable approximation of $y(t_i + h)$. We choose a new (smaller) step-size h. We repeat (13.4.16) and (13.4.17) starting at (t_i, w_i) again and using the new step-size.

How do we select a new step-size h to go from t_i to t_{i+1}? Formula (13.4.18) is used to find a new value of h such that $|(\tilde{w}_{i+1} - w_{i+1})/h| < \epsilon$. Since $\tau_{i+1}(h) = O(h^4)$, we may assume that $\tau_{i+1}(h) \approx Ch^4$ for some constant C and h small. Let q be a positive constant and suppose that (13.4.16) is used to approximate $y(t_i + qh)$. The local truncation error in this case is

$$\tau_{i+1}(hq) \approx Cq^4h^4 \approx q^4\tau_{i+1}(h) \approx \frac{q^4}{h}\left(\tilde{w}_{i+1} - w_{i+1}\right) \ .$$

If we require $|\tau_{i+1}(qh)| < \epsilon$, then $q^4\,|(\tilde{w}_{i+1} - w_{i+1})/h| < \epsilon$ or

$$q < \left|\frac{\epsilon h}{\tilde{w}_{i+1} - w_{i+1}}\right|^{1/4} \ . \tag{13.4.19}$$

The new step-size that is used is qh where q satisfies (13.4.19). ♠

Remark 13.4.41
In step 4 of the Runge-Kutta-Fehlberg method, we replace h by qh even if w_{i+1} is accepted. The reason is simple. If $|(\tilde{w}_{i+1} - w_{i+1})/h|$ is small, then q should be greater than one and the step-size is increased. This corresponds to taking a large step-size when y varies slowly. ♠

We now implement the Runge-Kutta-Fehlberg method.

Code 13.4.42 (Runge-Kutta-Fehlberg Method)

To approximate the solution of the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(0) &= y_0 \end{aligned}$$

Input: The maximal step-size hmax.
The minimal step-size hmin.
The maximal tolerated error T.
The initial time t_0 (t0 in the code below) and final time t_f (tf in the code below).
The initial conditions y_0 (y0 in the code below) at t_0.
The function $f(t, y)$ (funct in the code below).
Output: The approximations w_i (gw(i+1) in the code below) of $y(t_i)$ at t_i (gt(i+1) in the code below) with the requested tolerance and the step-size between hmin and hmax if it is possible.

```
function [gt,gw] = rgktfb(funct,t0,y0,tf,hmin,hmax,T)
  h = hmax;
  gt(1) = t0;
  gw(1) = y0;
  t = t0;
  w = y0;

  while (0 == 0)
    k1 = h*funct(t,w);
    k2 = h*funct(t+h/4,w+k1/4);
    k3 = h*funct(t+3*h/8,w+3*k1/32+9*k2/32);
    k4 = h*funct(t+12*h/13,w+1932*k1/2197-7200*k2/2197+7296*k3/2197);
    k5 = h*funct(t+h,w+439*k1/216-8*k2+3680*k3/513-845*k4/4104);
    k6 = h*funct(t+0.5*h,w-8*k1/27+2*k2-3544*k3/2565+1859*k4/4104-11*k5/40);

    sigma = abs(k1/360-128*k3/4275-2197*k4/75240+k5/50+2*k6/55);
    if (sigma < T)
      % We accept w as an approximation of y(t) .  w is an approximation
      % of  y(t)  given by a Runge-Kutta method of order four.
      t = t+h;
      w = w+25*k1/216+1408*k3/2565+2197*k4/4104-k5/5;
      gt = [gt;t];
      gw = [gw;w];
    end

    % We have reached  tf  and the program should stop.
    if (t >= tf)
      return;
    end

    if (sigma == 0)
```

```
        % We choose a large value for  q  if the error seems to be negligable.
        q = 5;
      else
        q = (T/sigma)^(0.25);
      end

      % We choose the step-size less than hmax and larger than hmin
      % such that the local error should still be less than  T.
      if (q < 0.1)
        % We do not reduce the step-size h to less than 1/10
        % its original size.
        h = 0.1*h;
      elseif (q > 4)
        % We do not increase the step-size h to more than 4 times
        % its original size or hmax.
        h = min(4*h,hmax);
      else
        h = min(h*q,hmax);
      end

      % We make sure than the step-size is not smaller than hmin.
      if (h < hmin)
        break;
        gt = NaN;
        gw = NaN;
      end

      % We adjust the step-size if we are going to exceed  tf  at the next
      % step.
      if (t + h > tf)
        h = tf - t;
      end
    end
  end
```

13.5 Multistep Methods

Up until now, we have only considered **one-step methods**; namely, methods where the approximation w_{i+1} of y_{i+1} is obtained from the approximation w_i of y_i. We fix $m > 0$ and consider methods where the approximation w_{i+1} of y_{i+1} is obtained from a combination of the approximations w_j of y_j for $j = i+1,\ i,\ i-1,\ \ldots,\ i-m$.

In this section, we assume that $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ in (13.1.1) is nice; namely, all the mixed derivatives of f that we need exist and are continuous. This implies that the solution y of (13.1.1) is sufficiently differentiable.

Definition 13.5.1 (General Form of a Multistep Method)

Let $0 < m < N$, $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= \sum_{j=0}^{m} a_j w_{i-j} + h \sum_{j=-1}^{m} b_j f(t_{i-j}, w_{i-j}) \quad , \quad i = m, m+1, \ldots, N-1 \\ w_i &= y_i \quad , \quad i = 0, 1, \ldots, m \end{aligned} \tag{13.5.1}$$

for some given constants a_i and b_i. If $b_{-1} = 0$, the method is called an **explicit** or **open method**. If $b_{-1} \neq 0$. the method is called an **implicit** or **closed method**.

Definition 13.5.2

For a multistep method, the **local truncation error** is defined by

$$\tau_{i+1}(h) = \frac{1}{h}\left(y_{i+1} - \sum_{j=0}^{m} a_j y_{i-j}\right) - \sum_{j=-1}^{m} b_j f(t_{i-j}, y_{i-j}) \quad , \quad m \leq i < N \ .$$

If, for all well-posed initial value problems (13.1.1), there exists a function $\tau : \mathbb{R} \to \mathbb{R}$ such that $|\tau_{i+1}(h)| \leq \tau(h) = O(h^p)$ near the origin for all i, we say that the **method is of order** p.

13.5.1 Classical Methods

We consider (13.1.1) with the usual partition $t_0 < t_1 < \ldots < t_N = t_f$ and $h = (t_f - t_0)/N$.

If we approximate $f(t, y(t))$ on $t_i \leq t \leq t_{i+1}$ by the average value

$$\frac{f(t_{i+1}, y(t_{i+1})) + f(t_i, y(t_i))}{2} ,$$

then

$$\begin{aligned} y(t_{i+1}) &= y(t_i) + \int_{t_i}^{t_{i+1}} y'(t)\,\mathrm{d}t = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y(t))\,\mathrm{d}t \\ &\approx y(t_i) + \int_{t_i}^{t_{i+1}} \frac{f(t_{i+1}, y(t_{i+1})) + f(t_i, y(t_i))}{2}\,\mathrm{d}t \\ &= y(t_i) + \frac{f(t_{i+1}, y(t_{i+1})) + f(t_i, y(t_i))}{2}\, h \ . \end{aligned}$$

If we suppose that $w_i \approx y(t_i)$, we get the following method.

Definition 13.5.3 (Trapezoidal Method)

Consider the initial value problem (13.1.1). Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$w_{i+1} = w_i + \frac{h}{2}\big(f(t_{i+1}.w_{i+1}) + f(t_i,w_i)\big) \quad , \quad 0 \le i < N$$
$$w_0 = y_0$$

The trapezoidal method is an implicit rule because w_{i+1} appears on both sides of the equation. Note that a one-step method like the trapezoidal method is still a multistep method.

To compute the order of the trapezoidal method, we use Taylor series expansions of $y(t)$ and $y'(t)$ for t near t_i. Namely,

$$\begin{aligned}\tau_{i+1}(h) &= \frac{y(t_{i+1}) - y(t_i)}{h} - \frac{1}{2}\big(f(t_i,y(t_i)) + f(t_{i+1},y(t_{i+1}))\big) \\ &= \frac{y(t_{i+1}) - y(t_i)}{h} - \frac{1}{2}\big(y'(t_i) + y'(t_{i+1})\big) \\ &= \frac{\big(y(t_i) + y'(t_i)\,h + y''(t_i)\,h^2/2 + y^{(3)}(\xi_i)\,h^3/6\big) - y(t_i)}{h} \\ &\qquad - \frac{1}{2}\left(y'(t_i) + \left(y'(t_i) + y''(t_i)\,h + \frac{1}{2}\,y^{(3)}(\eta_i)\,h^2\right)\right) = M(\xi_i,\eta_i)\,h^2\end{aligned}$$

for some ξ_i and η_i between t_i and t_{i+1}, where

$$M(\xi,\eta) = \left(\frac{1}{6}\,y^{(3)}(\xi) - \frac{1}{4}\,y^{(3)}(\eta)\right) .$$

If f is twice continuously differentiable on $[t_0,t_f]\times\mathbb{R}$, then $|y^{(3)}(t)|$ is continuous on $[t_0,t_f]$ and reaches is maximum at a point on the interval $[t_0,t_f]$. Let K be the maximum of $|y^{(3)}(t)|$ on $[t_0,t_f]$, then

$$|\tau_{i+1}(h)| \le \tau(h) \equiv \left(\frac{1}{6} + \frac{1}{4}\right)K h^2 = \frac{5K}{12}\,h^2 = O(h^2)$$

for all i. Hence, the trapezoidal method is of order 2.

Remark 13.5.4

The Trapezoidal Method is part of a family of methods called the **Theta Method**.

We consider (13.1.1) with the usual partition $a = t_0 < t_1 < \ldots < t_N = t_f$ and $h = (t_f - t_0)/N$. The Theta Method is defined by

$$w_{i+1} = w_i + h\,\big((1-\theta)\,f(t_{i+1},w_{i+1}) + \theta\,f(t_i,w_i)\big) \quad , \quad 1 \le i < N$$
$$w_0 = y(t_0)$$

If $\theta = 0$, we get the **Backward Euler's method**. If $\theta = 1/2$, we get the Trapezoidal Method. If $\theta = 1$, we get the Euler's method.

We compute the local truncation error of the Theta Method with the help of the Taylor series expansions of $y(t)$ and $y'(t)$ for t near t_i. We have

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - \big(\theta\,f(t_i,y(t_i)) + (1-\theta)\,f(t_{i+1},y(t_{i+1}))\big)$$

$$
\begin{aligned}
&= \frac{y(t_{i+1}) - y(t_i)}{h} - \big(\theta\, y'(t_i) + (1-\theta)\, y'(t_{i+1})\big) \\
&= \frac{\big(y(t_i) + y'(t_i)\, h + y''(t_i)\, h^2/2 + y^{(3)}(\xi_i)\, h^3/6\big) - y(t_i)}{h} \\
&\qquad - \left(\theta\, y'(t_i) + (1-\theta)\left(y'(t_i) + y''(t_i)\, h + \frac{1}{2}\, y^{(3)}(\eta_i)\, h^2\right)\right) \\
&= \left(\theta - \frac{1}{2}\right) y''(t_i)\, h + \left(\frac{1}{6}\, y^{(3)}(\xi_i) + \frac{\theta-1}{2}\, y^{(3)}(\eta_i)\right) h^2
\end{aligned}
$$

for some ξ_i and η_i between t_i and t_{i+1}. If we assume that f is twice continuously differentiable on $[t_0, t_f] \times \mathbb{R}$, then $|y''(t)|$ and $|y^{(3)}(t)|$ are continuous on $[t_0, t_f]$ and reaches their maximum at a point on the interval $[t_0, t_f]$. Let M_2 be the maximum of $|y''(t)|$ on $[t_0, t_f]$ and M_3 be the maximum of $|y^{(3)}(t)|$ on $[t_0, t_f]$. For $\theta = 1/2$, we have

$$
|\tau_{i+1}(h)| = \left|\frac{1}{6}\, y^{(3)}(\xi) + \frac{|\theta - 1|}{2}\, y^{(3)}(\eta)\right| h^2 \leq \tau(h) \equiv \left(\frac{1}{6} + \frac{1}{2}\right) M_3\, h^2 = O(h^2) \ .
$$

Hence, the Theta Method with $\theta = 1/2$ is of order 2. However, for $\theta \neq 1/2$, we have

$$
\begin{aligned}
|\tau_{i+1}(h)| &= \left|\left(\theta - \frac{1}{2}\right) y''(t_i)\, h + \left(\frac{1}{6}\, y^{(3)}(\xi) + \frac{|\theta-1|}{2}\, y^{(3)}(\eta)\right) h^2\right| \\
&\leq \tau(h) \equiv \frac{1}{2}\, M_2 h + \left(\frac{1}{6} + \frac{1}{2}\right) M_3\, h^2 = \left(\frac{M_2}{2} + \frac{2M_3}{3}\, h\right) h = O(h) \ .
\end{aligned}
$$

Hence, the Theta Methods with $\theta \neq 1/2$ are of order 1. The Trapezoidal Method is the best method of this family. ♠

13.5.2 General Approach

The general procedure to derive explicit multistep methods is as follows. We assume that m, N, h and t_i are as in Definition 13.5.1.

We consider the **Newton backward divided difference formula** of the interpolating polynomial p of $g(t) = f(t, y(t))$ at t_i, t_{i-1}, …, t_{i-m}. Namely,

$$
\begin{aligned}
p(t) &= g[t_i] + g[t_i, t_{i-1}](t - t_i) + g[t_i, t_{i-1}, t_{i-2}](t - t_i)(t - t_{i-1}) + \ldots \\
&\quad + g[t_i, t_{i-1}, \ldots, t_{i-m}](t - t_i)(t - t_{i-1}) \ldots (t - t_{i-m+1}) \ .
\end{aligned}
\tag{13.5.2}
$$

We have

$$
g(t) = p(t) + g[t_i, t_{i-1}, \ldots, t_{i-m}, t] \prod_{j=i-m}^{i} (t - t_j) \ . \tag{13.5.3}
$$

If we substitute $t = t_i + sh$ in (13.5.2), we get

$$
p(t) = \sum_{j=0}^{m} (-1)^j \binom{-s}{j} \nabla^j g_i \ , \tag{13.5.4}
$$

where $g_k = g(t_k)$ for $0 \leq k \leq N$,

$$\binom{r}{j} = \begin{cases} 1 & \text{if } j = 0 \\ \dfrac{r(r-1)(r-2)\ldots(r-j+1)}{j!} & \text{if } j > 0 \end{cases}$$

for $r \in \mathbb{R}$, and $\nabla^k g_i$ for $k \in \mathbb{N}$ is the k^{th} **backward difference** of g_i defined by

$$\begin{aligned} \nabla g_i &= g_i - g_{i-1} \ , \\ \nabla^2 g_i &= \nabla(g_i - g_{i-1}) = \nabla g_i - \nabla g_{i-1} = g_i - 2g_{i-1} + g_{i-2} \end{aligned}$$

and in general

$$\nabla^k g_i = \nabla^{k-1}(\nabla g_i)$$

for $k > 1$. (13.5.4) is called the **Newton backward difference formula** for the interpolating polynomial of g at t_i, t_{i-1}, $\ldots$, t_{i-m}.

If we substitute $t = t_i + sh$ in the error term of the polynomial interpolation p of g given in (13.5.3), we get

$$g[t_i, t_{i-1}, \ldots, t_{i-m}, t] \prod_{j=i-m}^{i} (t - t_j) = (-1)^{m+1} \binom{-s}{m+1} g^{(m+1)}(t_i + \eta_i(s)h) h^{m+1}$$

for some $\eta_i(s)$ in the smallest interval containing s, 0, -1, $\ldots$, $-m$; namely, $t_i + \eta_i(s)h$ is in the smallest interval containing t, t_i, t_{i-1}, $\ldots$, t_{i-m}.

Given $0 \leq q \leq m$, since

$$y_{i+1} - y_{i-q} = \int_{t_{i-q}}^{t_{i+1}} y'(t)\,\mathrm{d}t = \int_{t_{i-q}}^{t_{i+1}} g(t)\,\mathrm{d}t \ ,$$

we get

$$\begin{aligned} y_{i+1} - y_{i-q} = h \sum_{j=0}^{m} (-1)^j \nabla^j g_i \int_{-q}^{1} \binom{-s}{j} \mathrm{d}s \\ + (-1)^{m+1} h^{m+2} \int_{-q}^{1} \binom{-s}{m+1} g^{(m+1)}(t_i + \eta_i(s)h)\,\mathrm{d}s \ . \end{aligned} \tag{13.5.5}$$

The explicit multistep methods comes from this formula if we ignore the local discretization error

$$h^{m+2} \int_{-q}^{1} (-1)^{m+1} \binom{-s}{m+1} g^{(m+1)}(t_i + \eta_i(s)h)\,\mathrm{d}s \ .$$

The case $m = 3$ and $q = 0$ in (13.5.5) gives

$$\begin{aligned} y_{i+1} = y_i + \frac{h}{24}\left(55\, f(t_i . y_i) - 59\, f(t_{i-1}, y_{i-1}) + 37\, f(t_{i-2}, y_{i-2}) - 9\, f(t_{i-3}, y_{i-3})\right) \\ + (251/720)\, y^{(5)}(\xi_i)\, h^5 \end{aligned}$$

for some $\xi_i \in [t_{i-3}, t_{i+1}]$ and $3 \leq i < N$. We had to use the Mean Value Theorem for Integrals, Theorem 12.3.1, to get the discretization error $(251/720)\, y^{(5)}(\xi_i)\, h^5$; namely,

$$\begin{aligned}
\int_0^1 (-1)^4 \binom{-s}{4} g^{(4)}(t_i + \eta_i(s)h)\,\mathrm{d}s &= \int_0^1 (-1)^4 \frac{(-s)(-s-1)(-s-2)(-s-3)}{4!} y^{(5)}(t_i + \eta_i(s)h)\,\mathrm{d}s \\
&= \frac{1}{24} \int_0^1 \underbrace{s(s+1)(s+2)(s+3)}_{\geq 0}\, y^{(5)}(t_i + \eta_i(s)h)\,\mathrm{d}s \\
&= \frac{1}{24} y^{(5)}(t_i + \tilde{\eta}_i h) \int_0^1 s(s+1)(s+2)(s+3)\,\mathrm{d}s
\end{aligned}$$

for some $\tilde{\eta}_i \in [-3, 1]$. If we let $\xi_i = t_i + \tilde{\eta}_i h$ and compute the integral, we get $(251/720)\, y^{(5)}(\xi_i)\, h^5$.

We get the following famous explicit method.

Definition 13.5.5 (Adams-Bashforth Method of Order Four)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned}
w_{i+1} &= w_i + \frac{h}{24}\big(55\, f(t_i . w_i) - 59\, f(t_{i-1}, w_{i-1}) + 37\, f(t_{i-2}, w_{i-2}) \\
&\qquad -9\, f(t_{i-3}, w_{i-3})\big) \quad , \quad 3 \leq i < N \\
w_i &= y_i \quad , \quad 0 \leq i < 4
\end{aligned}$$

The local truncation error $\tau_{i+1}(h)$ is $(251/720)\, y^{(5)}(\xi_i)\, h^4$ for some $\xi_i \in [t_{i-3}, t_{i+1}]$ and $3 \leq i < N$.

The procedure to derive implicit multistep methods is as follows. We assume that m, N, h and t_i are as in Definition 13.5.1.

We consider the Newton backward divided difference formula of the interpolating polynomial p of $g(t) = f(t, y(t))$ at t_{i+1}, t_i, $\ldots$, t_{i-m}. Namely,

$$\begin{aligned}
p(t) &= g[t_{i+1}] + g[t_{i+1}, t_i](t - t_{i+1}) + g[t_{i+1}, t_i, t_{i-1}](t - t_{i+1})(t - t_i) + \ldots \\
&\quad + g[t_{i+1}, t_i, \ldots, t_{i-m}](t - t_{i+1})(t - t_i)\ldots(t - t_{i-m+1}) \; .
\end{aligned} \tag{13.5.6}$$

We have

$$g(t) = p(t) + g[t_{i+1}, t_i, \ldots, t_{i-m}, t] \prod_{j=i-m}^{i+1} (t - t_j) \; . \tag{13.5.7}$$

If we substitute $t = t_i + sh$ in (13.5.6), we get

$$p(t) = \sum_{j=0}^{m+1} (-1)^j \binom{1-s}{j} \nabla^j g_{i+1} \; .$$

If we substitute $t = t_i + sh$ in the error term of the polynomial interpolation p of g given in (13.5.7), we get

$$g[t_{i+1}, t_i, \ldots, t_{i-m}, t] \prod_{j=i-m}^{i+1} (t - t_j) = (-1)^{m+2} \binom{1-s}{m+2} g^{(m+2)}(t_i + \eta_i(s)h) h^{m+2}$$

for some $\eta_i(s)$ in the smallest interval containing s, 1, 0, −1, ..., $-m$.

Given $0 \leq q \leq m$, since

$$y_{i+1} - y_{i-q} = \int_{t_{i-q}}^{t_{i+1}} y'(t)\,\mathrm{d}t = \int_{t_{i-q}}^{t_{i+1}} g(t)\,\mathrm{d}t \ ,$$

we get

$$\begin{aligned} y_{i+1} - y_{i-q} &= h \sum_{j=0}^{m+1} (-1)^j \nabla^j g_{i+1} \int_{-q}^{1} \binom{1-s}{j} \mathrm{d}s \\ &\quad + (-1)^{m+2} h^{m+3} \int_{-q}^{1} \binom{1-s}{m+2} g^{(m+2)}(t_i + \eta_i(s)h)\,\mathrm{d}s \ . \end{aligned} \tag{13.5.8}$$

The implicit multistep methods comes from this formula if we ignore the local discretization error

$$h^{m+3} \int_{-q}^{1} (-1)^{m+2} \binom{1-s}{m+1} g^{(m+2)}(t_i + \eta_i(s)h)\,\mathrm{d}s \ .$$

The case $m = 2$ and $q = 0$ in (13.5.8) gives

$$\begin{aligned} y_{i+1} = y_i &+ \frac{h}{24} \left(9 f(t_{i+1}, y_{i+1}) + 19 f(t_i . y_i) - 5 f(t_{i-1}, y_{i-1}) + f(t_{i-2}, y_{i-2})\right) \\ &- (19/720)\, y^{(5)}(\xi_i)\, h^5 \end{aligned}$$

for some $\xi_i \in [t_{i-2}, t_{i+1}]$ and $2 \leq i < N$. As for the previous explicit method, we had to use the Mean Value Theorem for Integrals to get the discretization error $-(19/720)\, y^{(5)}(\xi_i)\, h^5$.

We get the following famous implicit method.

Definition 13.5.6 (**Adams-Moulton Method of Order Four**)

Let $h = (t_f - t_0)/N$, $t_i = t_0 + ih$ and $y_i = y(t_i)$ for $i = 0, 1, 2, \ldots, N$. The approximation w_i of y_i is the solution of the difference equation

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{24} \left(9 f(t_{i+1}, w_{i+1}) + 19 f(t_i . w_i) - 5 f(t_{i-1}, w_{i-1})\right. \\ &\qquad \left. + f(t_{i-2}, w_{i-2})\right) \quad , \quad 2 \leq i < N \\ w_i &= y_i \quad , \quad 0 \leq i < 3 \end{aligned}$$

The local truncation error $\tau_{i+1}(h)$ is $-(19/720)\, y^{(5)}(\xi_i)\, h^4$ for some $\xi_i \in [t_{i-2}, t_{i+1}]$ and $2 \leq i < N$.

By varying m and q in (13.5.5) and (13.5.8), we can find many more multistep methods.

Example 13.5.7

It is generally impossible to solve explicitly for w_{i+1} the finite difference equations of the implicit multistep methods. For instance, the Adams-Moulton method of order four applied to the initial value problem

$$\begin{aligned} y'(t) &= e^{y(t)} \quad , \quad 0 \le t \le 0.25 \\ y(0) &= 1 \end{aligned}$$

gives the equation

$$w_{i+1} = w_i + \frac{h}{24}\left(9e^{w_{i+1}} + 19e^{w_i} - 5e^{w_{i-1}} + e^{w_{i-2}}\right)$$

which cannot be solved explicitly for w_{i+1}. ♣

Remark 13.5.8

1. Iterations are used to find the approximation w_{i+1} of y_{i+1} in the implicit multistep methods (13.5.1). Suppose that a first approximation $w_{i+1}^{[0]}$ of w_{i+1} is given — We will provide in the next section a method to obtain a first approximation. The solution w_{i+1} of (13.5.1) is approximated using the iterative system

$$w_{i+1}^{[k+1]} = \sum_{j=0}^{m} a_j w_{i-j} + hb_{-1} f\left(t_{i+1}, w_{i+1}^{[k]}\right) + h\sum_{j=0}^{m} b_j f(t_{i-j}, w_{i-j}) \tag{13.5.9}$$

for $k = 0, 1, \ldots$

We now show that if h is small enough such that $|b_{-1}h|L < 1$, where L is the Lipschitz constant associated to f as in (13.1.3), then $\left\{w_{i+1}^{[k]}\right\}_{k=0}^{\infty}$ converges to the unique solution w_{i+1} of (13.5.1). The reader will recognize that the following proof is "basically identical" to the proof of the Fixed Point Theorem, Theorem 2.4.2.

The iterative system (13.5.9) can be rewritten as

$$w_{i+1}^{[k+1]} = A_i + hb_{-1} G_i\left(w_{i+1}^{[k]}\right) + hF_i \ , \tag{13.5.10}$$

where

$$A_i = \sum_{j=0}^{m} a_j w_{i-j} \quad \text{and} \quad F_i = \sum_{j=0}^{m} b_j f(t_{i-j}.w_{i-j})$$

are constant, and $G_i\left(w_{i+1}^{[k]}\right) = f\left(t_{i+1}, w_{i+1}^{[k]}\right)$.

We first prove that if there is a solution to

$$w = A_i + hb_{-1} G_i(w) + hF_i \ , \tag{13.5.11}$$

then it is unique. Suppose that w and w^* are two distinct solutions of (13.5.11); namely, if

$$w = A_i + b_{-1} h G_i(w) + hF_i$$

and

$$w^* = A_i + b_{-1}hG_i(w^*) + hF_i \ .$$

Then

$$|w - w^*| = |b_{-1}h(G_i(w) - G_i(w^*))| \le |b_{-1}h|L|w - w^*| < |w - w^*| \ .$$

This is a contradiction.

We prove that the sequence $\left\{w_{i+1}^{[k]}\right\}_{k=0}^{\infty}$ defined by (13.5.10) converges. In fact, we prove that it is a Cauchy sequence. Let ϵ be a small number. We find a positive integer N such that $\left|w_{i+1}^{[r]} - w_{i+1}^{[s]}\right| < \epsilon$ whenever $r, s \ge N$.

First, we prove by induction that

$$\left|w_{i+1}^{[k+1]} - w_{i+1}^{[k]}\right| \le |hb_{-1}L|^k \left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| \ . \tag{13.5.12}$$

for all k. We have that (13.5.12) is obviously true for $k = 0$. Suppose that (13.5.12) is true for k, then

$$\begin{aligned}\left|w_{i+1}^{[k+2]} - w_{i+1}^{[k+1]}\right| &= \left|hb_{-1}\left(G\left(w_{i+1}^{[k+1]}\right) - G\left(w_{i+1}^{[k]}\right)\right)\right| \le |hb_{-1}L| \ \left|w_{i+1}^{[k+1]} - w_{i+1}^{[k]}\right| \\ &\le |hb_{-1}L| \ |hb_{-1}L|^k \left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| = |hb_{-1}L|^{k+1} \left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| \ ,\end{aligned}$$

where the first inequality comes from the Lipschitz continuity of G and the second inequality comes from the hypothesis of induction. Hence, (13.5.12) is true for $k+1$. This complete the proof by induction.

Hence, if $|hb_{-1}L| < 1$ and

$$r > s \ge N > \frac{\ln(\epsilon) + \ln(1 - |hb_{-1}L|) - \ln\left(\left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right|\right)}{\ln(|hb_{-1}L|)} \ ,$$

we have

$$\begin{aligned}\left|w_{i+1}^{[r]} - w_{i+1}^{[s]}\right| &\le \left|w_{i+1}^{[r]} - w_{i+1}^{[r-1]}\right| + \left|w_{i+1}^{[r-1]} - w_{i+1}^{[r-2]}\right| + \ldots + \left|w_{i+1}^{[s+1]} - w_{i+1}^{[s]}\right| \\ &\le \left(|hb_{-1}L|^{r-s-1} + |hb_{-1}L|^{r-s-2} + \ldots + |hb_{-1}L| + 1\right)|hb_{-1}L|^s \left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| \\ &= \frac{1 - |hb_{-1}L|^{r-s}}{1 - |hb_{-1}L|}|hb_{-1}L|^s \left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| \le \frac{|hb_{-1}L|^s}{1 - |hb_{-1}L|}\left|w_{i+1}^{[1]} - w_{i+1}^{[0]}\right| < \epsilon\end{aligned}$$

Finally, let w_{i+1} be the limit of $\left\{w_{i+1}^{[k]}\right\}_{k=0}^{\infty}$. We show that w_{i+1} is a solution of (13.5.11). Since G is a continuous function, if we take the limit with respect to k on both sides of (13.5.10), we get

$$w_{i+1} = \lim_{k\to\infty} w_{i+1}^{[k+1]} = \lim_{k\to\infty}\left(A_i + hb_{-1}G_i\left(w_{i+1}^{[k]}\right) + hF_i\right) = A_i + hb_{-1}G_i(w_{i+1}) + hF_i \ .$$

2. Implicit multistep methods may seem to be inefficient methods to find an approximation w_i of y_i because iterations have to be done to find this approximation. However, for some multistep methods, the number of iterations necessary to find a good approximation of y_i is small and the step-size h can be taken relatively large. Moreover, implicit multistep methods are usually "stable" as we will see later. They are also very useful to solve "stiff" differential equations as we will also see soon.

♠

Remark 13.5.9
Instead of using the Newton backward divided difference formula of the interpolating polynomial p of $g(t) = f(t, y(t))$ to derive explicit and implicit multistep methods to approximate the solution of (13.1.1), we can use the Lagrange Interpolating Polynomial

$$p(t) = \sum_{j=i-m}^{i} \left(f(t_j, y(t_j)) \prod_{\substack{k=i-m \\ k\neq j}}^{i} \left(\frac{t - t_k}{t_j - t_k} \right) \right)$$

to get the formula

$$w_{i+1} = w_{i-q} + \sum_{j=i-m}^{i} \left(f(t_j, w_j) \int_{t_{i-q}}^{t_{i+1}} \prod_{\substack{k=i-m \\ k\neq j}}^{i} \left(\frac{t - t_k}{t_j - t_k} \right) \mathrm{d}t \right)$$

for the explicit multistep methods, and the Lagrange Interpolating Polynomial

$$p(t) = \sum_{j=i-m}^{i+1} \left(f(t_j, y(t_j)) \prod_{\substack{k=i-m \\ k\neq j}}^{i+1} \left(\frac{t - t_k}{t_j - t_k} \right) \right)$$

to get the formula

$$w_{i+1} = w_{i-q} + \sum_{j=i-m}^{i+1} \left(f(t_j, w_j) \int_{t_{i-q}}^{t_{i+1}} \prod_{\substack{k=i-m \\ k\neq j}}^{i+1} \left(\frac{t - t_k}{t_j - t_k} \right) \mathrm{d}t \right)$$

for the implicit multistep methods. The two integrals above can be computed using the substitution $t = t_i + sh$. We get respectively

$$\begin{aligned}\int_{t_{i-q}}^{t_{i+1}} \prod_{\substack{k=i-m \\ k\neq j}}^{i} \left(\frac{t - t_k}{t_j - t_k} \right) \mathrm{d}t &= h \int_{-q}^{1} \prod_{\substack{k=0 \\ k\neq i-j}}^{m} \left(\frac{t_i + sh - t_{i-k}}{t_{i-(i-j)} - t_{i-k}} \right) \mathrm{d}s = h \int_{-q}^{1} \prod_{\substack{k=0 \\ k\neq i-j}}^{m} \left(\frac{s + k}{-(i-j) + k} \right) \mathrm{d}s \\ &= \frac{h(-1)^{i-j}}{(i-j)!\,(m-i+j)!} \int_{-q}^{1} \prod_{\substack{k=0 \\ k\neq i-j}}^{m} (s+k)\,\mathrm{d}s\end{aligned}$$

after substituting k by $i - k$ and noting that $0 \le i - j \le m$, and

$$\int_{t_{i-q}}^{t_{i+1}} \prod_{\substack{k=i-m \\ k\neq j}}^{i+1} \left(\frac{t - t_k}{t_j - t_k} \right) \mathrm{d}t = h \int_{-q}^{1} \prod_{\substack{k=0 \\ k\neq i-j+1}}^{m+1} \left(\frac{t_i + sh - t_{i-k+1}}{t_{i-(i-j)} - t_{i-k+1}} \right) \mathrm{d}s$$

$$= h \int_{-q}^{1} \prod_{\substack{k=0 \\ k \neq i-j+1}}^{m+1} \left(\frac{s+k-1}{-(i-j+1)+k} \right) \mathrm{d}s = \frac{h(-1)^{i-j+1}}{(i-j+1)!\,(m-i+j)!} \int_{-q}^{1} \prod_{\substack{k=0 \\ k \neq i-j+1}}^{m+1} (s+k-1)\,\mathrm{d}s \ .$$

after substituting k by $i-k+1$ and noting that $0 \le i-j+1 \le m+1$.

This approach obviously yields the same formulae than those found with the approach that we have chosen. However, it does not provide the local truncation error that we have been able to find with our approach. ♠

13.5.3 Another Approach to Multistep Methods

There is still another approach to develop multistep methods based on the following theorem.

Theorem 13.5.10

The multistep method (13.5.1) is of order $p \geq 1$ if and only if there exists $c \neq 0$ such that

$$p(w) - q(w)\ln(w) = c(w-1)^{p+1} + O((w-1)^{p+2})$$

for w near 1, where

$$p(w) = w^{m+1} - \sum_{j=0}^{m} a_j\, w^{m-j} \quad \text{and} \quad q(w) = \sum_{j=-1}^{m} b_j\, w^{m-j} \ .$$

The polynomial p is called the **characteristic polynomial** of the multistep method.

The beauty of this approach is, as we will see in Section 13.6, that the polynomials p and q in the previous theorem play in important role in the study of "consistency", "stability" and "convergence" of numerical methods to approximate solutions of ordinary differential equations.

Proof.
As usual, we assume that f is smooth enough such that we can express the solution y of (13.1.1) as a Taylor series of radius at least mh about any point $t_i \in [a,b]$. Hence, since $f(t_i, y(t_i)) = y'(t_i)$, we get

$$\begin{aligned}
h\,\tau_{i+1}(h) &= y(t_i+h) - \sum_{j=0}^{m} a_j\, y(t_i - jh) - h \sum_{j=-1}^{m} b_j\, f(t_i - jh, y(t_i - jh)) \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} y^{(k)}(t_i)\, h^k - \sum_{j=0}^{m} a_j \left(\sum_{k=0}^{\infty} \frac{1}{k!} y^{(k)}(t_i)\,(-j)^k h^k \right) - h \sum_{j=-1}^{m} b_j \left(\sum_{k=0}^{\infty} \frac{1}{k!} y^{(k+1)}(t_i)\,(-j)^k h^k \right) \\
&= \left(1 - \sum_{j=0}^{m} a_j \right) y(t_i) + \left(1 + \sum_{j=0}^{m} a_j\, j - \sum_{j=-1}^{m} b_j \right) y'(t_i) h \\
&\quad + \sum_{k=2}^{\infty} \frac{1}{k!} \left(1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k \right) y^{(k)}(t_i)\, h^k - h \sum_{k=1}^{\infty} \frac{1}{k!} \left((-1)^k \sum_{j=-1}^{m} b_j\, j^k \right) y^{(k+1)}(t_i)\, h^k
\end{aligned}$$

$$
\begin{aligned}
&= \left(1 - \sum_{j=0}^{m} a_j\right) y(t_i) + \left(1 + \sum_{j=0}^{m} a_j\, j - \sum_{j=-1}^{m} b_j\right) y'(t_i) h \\
&\quad + \sum_{k=2}^{\infty} \frac{1}{k!}\left(1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k - (-1)^{k-1} k \sum_{j=-1}^{m} b_j\, j^{k-1}\right) y^{(k)}(t_i)\, h^k \ .
\end{aligned}
$$

Thus, (13.5.1) is of order $p \geq 1$ if and only if

$$
1 - \sum_{j=0}^{m} a_j = 0 \quad \text{and} \quad 1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k - (-1)^{k-1} k \sum_{j=-1}^{m} b_j\, j^{k-1} = 0 \tag{13.5.13}
$$

for $1 \leq k \leq p$, and

$$
c \equiv 1 - (-1)^{p+1} \sum_{j=0}^{m} a_j\, j^{p+1} - (-1)^{p} (p+1) \sum_{j=-1}^{m} b_j\, j^{p} \neq 0 \ . \tag{13.5.14}
$$

In this case, we have that

$$
h\, \tau_{i+1}(h) = c\, \frac{y^{(p+1)}(t_i)}{(p+1)!}\, h^{p+1} + O(h^{p+2}) \ .
$$

Moreover, if we set $w = e^x$, we get

$$
\begin{aligned}
\frac{p(w) - q(w)\, \ln(w)}{e^m} &= e^x - \sum_{j=0}^{m} a_j\, e^{-jx} - x \sum_{j=-1}^{m} b_j\, e^{-jx} \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} x^k - \sum_{j=0}^{m} a_j \left(\sum_{k=0}^{\infty} \frac{1}{k!} (-j)^k x^k\right) - x \sum_{j=-1}^{m} b_j \left(\sum_{k=0}^{\infty} \frac{1}{k!} (-j)^k x^k\right) \\
&= \left(1 - \sum_{j=0}^{m} a_j\right) + \left(1 + \sum_{j=0}^{m} a_j\, j - \sum_{j=-1}^{m} b_j\right) x \\
&\quad + \sum_{k=2}^{\infty} \frac{1}{k!}\left(1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k\right) x^k - \sum_{k=1}^{\infty} \frac{1}{k!}\left((-1)^k \sum_{j=-1}^{m} b_j\, j^k\right) x^{k+1} \\
&= \left(1 - \sum_{j=0}^{m} a_j\right) + \left(1 + \sum_{j=0}^{m} a_j\, j - \sum_{j=-1}^{m} b_j\right) x \\
&\quad + \sum_{k=2}^{\infty} \frac{1}{k!}\left(1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k - (-1)^{k-1} k \sum_{j=-1}^{m} b_j\, j^{k-1}\right) x^k \ .
\end{aligned}
$$

So,

$$
p(w) - q(w)\, \ln(w) = c\, x^{p+1} + O(x^{p+2}) = c\,(w-1)^{p+1} + O((w-1)^{p+2})
$$

if and only if (13.5.13) and (13.5.14) are satisfied. Recall that $x = \ln(w) = (w-1) + O((w-1)^2)$ for w near 1. ∎

Remark 13.5.11
We will see in Section 13.6 that the condition

$$
p(1) = 1 - \sum_{j=0}^{m} a_j = 0
$$

is necessary for the method to be "consistent." We will see that consistency and the "root condition" imply that the method is "convergent." The "root condition" also implies that the method is "zero-stable." But we are getting ahead of ourselves and should go back to multistep methods. ♠

Example 13.5.12
The multistep method

$$w_{i+1} = w_i + h\left(\frac{23}{12}f(t_i,w_i) - \frac{4}{3}f(t_{i-1},w_{i-1}) + \frac{5}{12}f(t_{i-2},w_{i-2})\right) \quad , \quad 0 \leq i < N$$

is of order 3. We use the previous theorem with $m = 2$ to prove this statement. We have $p(w) = w^3 - w^2$ and

$$q(w) = \frac{23}{12}w^2 - \frac{4}{3}w + \frac{5}{12} \ .$$

To develop $p(w) - q(w)\ln w$ near $w = 1$, we set $v = w - 1$. Hence

$$\begin{aligned} p(w) &= (v+1)^3 - (v+1)^2 = v^3 + 2v^2 + v \ , \\ q(w) &= \frac{23}{12}(v+1)^2 - \frac{4}{3}(v+1) + \frac{5}{12} = \frac{23}{12}v^2 + \frac{5}{2}v + 1 \end{aligned}$$

and

$$\begin{aligned} p(w) - q(w)\ln w &= (v^3 + 2v^2 + v) - \left(\frac{23}{12}v^2 + \frac{5}{2}v + 1\right)\left(\sum_{k=1}^{\infty}(-1)^{k+1}\frac{v^k}{k}\right) \\ &= \frac{3}{8}v^4 + O(v^5) = \frac{3}{8}(w-1)^4 + O((w-1)^5) \ . \end{aligned}$$

♣

We can use Theorem 13.5.10 to construct multistep methods of any order.

To construct an explicit multistep method of order p, choose a polynomial $p(w) = w^{m+1} - \sum_{j=0}^{m} a_j\, w^{m-j}$ such that $p(1) = 0$. The polynomial $q(w)$ is the polynomial of degree m (if such polynomial exists) given by the relation

$$p(w) - q(w)\ln(w) = O((w-1)^{p+1}) \ .$$

To construct an implicit multistep method of order p, choose $p(w)$ as before but this time the polynomial $q(w)$ is the polynomial of degree $m+1$ (if such a polynomial exists) given by the relation

$$p(w) - q(w)\ln(w) = O((w-1)^{p+1}) \ .$$

As we will realize in the next examples, it is useful to note that $\dfrac{v}{\ln(v+1)}$ can be defined at $v = 0$ by the extension

$$g(v) = \begin{cases} \dfrac{v}{\ln(v+1)} & \text{if} \quad v \neq 0 \\ 1 & \text{if} \quad v = 0 \end{cases}$$

Moreover,

$$g(v) = 1 + \frac{v}{2} - \frac{v^2}{12} + \frac{v^3}{24} + O(v^4) \ .$$

Example 13.5.13
To construct an implicit method of order 4 from $p(w) = w^3 - w^2$, let $v = w - 1$. Then

$$\begin{aligned}\frac{p(w)}{\ln(w)} &= \frac{w^2(w-1)}{\ln(w)} = (v+1)^2 \frac{v}{\ln(v+1)} \\ &= (v^2 + 2v + 1)\left(1 + \frac{v}{2} - \frac{v^2}{12} + \frac{v^3}{24} + O(v^4)\right) \\ &= 1 + \frac{5}{2} v + \frac{23}{12} v^2 + \frac{3}{8} v^3 + O(v^4) \\ &= \frac{1}{24} - \frac{5}{24} w + \frac{19}{24} w^2 + \frac{3}{8} w^3 + O((w-1)^4) \ .\end{aligned}$$

Thus

$$q(w) = \frac{1}{24} - \frac{5}{24} w + \frac{19}{24} w^2 + \frac{3}{8} w^3$$

and the multistep method is

$$w_{i+1} = w_i + h\left(\frac{3}{8} f(t_{i+1}, w_{i+1}) + \frac{19}{24} f(t_i, w_i) - \frac{5}{24} f(t_{i-1}, w_{i-1}) + \frac{1}{24} f(t_{i-2}, w_{i-2})\right) \ , \ \leq i < N \ .$$

This is our famous Adams-Moulton method of order four. ♣

Example 13.5.14
To construct an explicit method of order 1 from $p(w) = w^2 - w$, let $v = w - 1$. Then

$$\begin{aligned}\frac{p(w)}{\ln(w)} &= \frac{w(w-1)}{\ln(w)} = (v+1) \frac{v}{\ln(v+1)} = (v+1)\left(1 + \frac{v}{2} - O(v^2)\right) \\ &= 1 + \frac{3}{2} v + O(v^2) = -\frac{1}{2} + \frac{3}{2} w + O(|w-1|^2) \ .\end{aligned}$$

Thus $q(w) = -\frac{1}{2} + \frac{3}{2} w$ and the multistep method is

$$w_{i+1} = w_i + h\left(\frac{3}{2} f(t_i, w_i) - \frac{1}{2} f(t_{i-1}, w_{i-1})\right) \quad , \quad 1 \leq i < N \ .$$

♣

Remark 13.5.15
If we consider $p(w) = w^{m-1}(w-1)$, we get the **Adams methods**. The implicit methods are called **Adams-Moulton methods** and the explicit methods are called **Adams-Bashforth methods**.

If we consider $p(w) = w^{m-2}(w^2 - 1)$, the explicit methods (of order m) are called **Nystron methods** and the implicit methods (of order $m+1$) are called **Milne methods**. ♠

13.5.4 Backward Difference Formulae

Definition 13.5.16

A multistep method of the form (13.5.1) and of order $m+1$ is called a **Backward Difference Formula** if $b_{-1} \neq 0$ and $b_j = 0$ for $0 \leq j \leq m$.

Proposition 13.5.17

For a Backward Difference Formula, we have $b_{-1} = \left(\sum_{j=1}^{m+1} \frac{1}{j}\right)^{-1}$ and the characteristic polynomial is $p(w) = b_{-1} \sum_{j=1}^{m+1} \frac{1}{j}\, w^{m+1-j}(w-1)^j$.

Proof.
We will use Theorem 13.5.10. We have by hypothesis that $q(w) = b_{-1}w^{m+1}$ for some non-zero constant $b_{-1} \in \mathbb{R}$. Since the Backward Difference Formula is of order $m+1$, we have

$$p(w) - b_{-1}w^{m+1}\ln(w) = O\left((w-1)^{m+2}\right) \tag{13.5.15}$$

for w near 1. If we substitute $w = 1/v$ in this equation and multiply it by v^{m+1}, we get

$$v^{m+1}p\left(\frac{1}{v}\right) = b_{-1}\ln(v) + O\left((v-1)^{m+2}\right)$$

for v near 1. Since

$$\ln(v) = \ln(1+(v-1)) = \sum_{j=1}^{m+1} \frac{(-1)^{j-1}}{j}(v-1)^j + O\left((v-1)^{m+2}\right)$$

for v near 1, we get

$$v^{m+1}p\left(\frac{1}{v}\right) = b_{-1}\sum_{j=1}^{m+1} \frac{(-1)^{j}}{j}(v-1)^j + O\left((v-1)^{m+2}\right) .$$

If we rewrite this equation in function of w, we get

$$\begin{aligned} p(w) &= b_{-1}w^{m+1}\sum_{j=1}^{m+1} \frac{(-1)^{j}}{j}(1-w)^j\, w^{-j} + O\left((w-1)^{m+2}\right) \\ &= b_{-1}\sum_{j=1}^{m+1} \frac{1}{j}(w-1)^j\, w^{m+1-j} + O\left((w-1)^{m+2}\right) . \end{aligned}$$

Since $p(w)$ is a polynomial of degree $m+1$ and any extra terms of the form $(w-1)^k$ with $k \geq m+2$ will not affect (13.5.15), we may assume that

$$p(w) = b_{-1}\sum_{j=1}^{m+1} \frac{1}{j}(w-1)^j\, w^{m+1-j} .$$

To get p of the form $p(w) = w^{m+1} - \sum_{j=0}^{m} a_j w^{m-j}$, we need $1 = b_{-1}\sum_{j=1}^{m+1} \frac{1}{j}$. ∎

Example 13.5.18
The case $m = 0$ gives $b_{-1} = 1$ and $p(w) = w - 1$. We get the Backward Euler's method $w_{i+1} = w_i + h\,f(t_{i+1}, w_{i+1})$ for $0 \leq i < N$.

The case $m = 1$ gives $b_{-1} = 2/3$ and

$$p(w) = \frac{2}{3}\left((w-1)w + \frac{1}{2}(w-1)^2\right) = w^2 - \frac{4}{3}\,w + \frac{1}{3}\ .$$

We get the backward method

$$w_{i+1} = \frac{4}{3}\,w_i - \frac{1}{3}\,w_{i-1} + \frac{2}{3}\,hf(t_{i+1}, w_{i+1}) \quad , \quad 1 \leq i < N\ .$$

♣

13.5.5 Predictor-Corrector Methods

Since it is generally impossible to solve explicitly for w_{i+1} the finite difference equations of the implicit multistep methods, we do not use implicit multistep methods to approximate y_{i+1} but we use them to improve the approximation of y_{i+1} given by the explicit methods.

The combination of an explicit and an implicit multistep method of the same order gives a predictor-corrector method. We illustrate this idea with the Adams-Bashforth method of order four and the Adams-Moulton method of order four. Both are multistep methods of order four.

Algorithm 13.5.19 (Predictor-Corrector Method)

1. Use Runge-Kutta Method of order four to get approximations w_i, of y_i for $i = 1$, 2, and 3. Recall that $w_0 = y_0$.

2. (P) Suppose that we have found the approximation w_i of y_i for $i \geq 3$. Use Adams-Bashforth formula to get a first approximation

$$w_{i+1}^{[0]} = w_i + \frac{h}{24}\left(55\,f(t_i.w_i) - 59\,f(t_{i_1}, w_{i-1}) + 37\,f(t_{i-2}, w_{i_2}) - 9\,f(t_{i-3}, w_{i-3})\right)$$

of y_{i+1}.

3. (C) Use Adams-Moulton formula to get a better (we hope) approximation

$$w_{i+1}^{[1]} = w_i + \frac{h}{24}\left(9\,f\left(t_{i+1}, w_{i+1}^{[0]}\right) + 19\,f(t_i.w_i) - 5\,f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})\right)$$

of y_{i+1}. Accept $w_{i+1}^{[1]}$ as the approximation w_{i+1} of y_{i+1}.

4. Go back to (2) if $i < N$.

Remark 13.5.20
Generally, no more than two iterations are done. If the iterative process does not give a "good" approximation after two iterations. The step-size is usually reduced. ♠

We now look a little deeper into the theory to determine the order of the predictor-corrector method resulting from combining two multistep methods.

We consider two multistep methods to approximate the solution of (13.1.1). The first method is an explicit method of order p given by

$$w_{i+1} = \sum_{j=0}^{m} a_j\, w_{i-j} + h \sum_{j=0}^{m} b_j\, f(t_{i-j}, w_{i-j}) \tag{13.5.16}$$

for $m \leq i < N$ and the second method is an implicit method of order $\tilde{p}$ given by

$$\tilde{w}_{i+1} = \sum_{j=0}^{\tilde{m}} \tilde{a}_j\, \tilde{w}_{i-j} + h \sum_{j=-1}^{\tilde{m}} \tilde{b}_j\, f(t_{i-j}, \tilde{w}_{i-j}) \tag{13.5.17}$$

for $\tilde{m} \leq i < N$. We have that w_{i-j} is the approximation of $y(t_{i-j})$ given by (13.5.16) and $\tilde{w}_{i-j}$ is the approximation of $y(t_{i-j})$ given by (13.5.17).

We combine these two multistep methods to create a predictor-corrector method as follows.

Let $M = \max\{m, \tilde{m}\}$. Suppose that w_0, w_1, …, w_M have been obtained from a method of order at least equal to $\max\{p, \tilde{p}\}$.

Assuming that we have the values of $w_{i-j} = \tilde{w}_{i-j}$ and $f_{i-j} = f(t_{i-j}, w_{i-j})$ for $0 \leq J \leq M$, we compute the value of w_{i+1} for $i \geq M$ as follows.

P: Prediction

$$w_{i+1}^{[0]} = \sum_{j=0}^{m} a_j\, w_{i-j} + h \sum_{j=0}^{m} b_j\, f_{i-j} \tag{13.5.18}$$

(EC)$^\nu$: Evaluation and Correction for $k = 0, 1, \ldots \nu - 1$.

$$\begin{aligned} f_{i+1}^{[k+1]} &= f(t_{i+1}, w_{i+1}^{[k]}) \\ w_{i+1}^{[k+1]} &= \sum_{j=0}^{\tilde{m}} \tilde{a}_j\, w_{i-j} + h \left(\sum_{j=0}^{\tilde{m}} \tilde{b}_j\, f_{i-j} + \tilde{b}_{i+1}\, f_{i+1}^{[k+1]} \right) \end{aligned} \tag{13.5.19}$$

We set $w_{i+1} = \tilde{w}_{i+1} = w_{i+1}^{[\nu]}$.

E: Evaluation

$$f_{i+1} = f(t_{i+s}, w_{i+s})$$

This predictor-corrector method is named $P(EC)^\nu E$. In general, the number of iterations ν should be small.

In the proof of Theorem 13.5.10, we have shown that a multistep method of the form (13.5.1) satisfies

$$\begin{aligned} &y(t_i+h) - \sum_{j=0}^{m} a_j\, y(t_i - jh) - h \sum_{j=-1}^{m} b_j\, f(t_i - jh, y(t_i - jh)) \\ &= \left(1 - \sum_{j=0}^{m} a_j\right) y(t_i) + \left(1 + \sum_{j=0}^{m} a_j\, j - \sum_{j=-1}^{m} b_j\right) y'(t_i)\, h \\ &\quad + \sum_{k=2}^{\infty} \frac{1}{k!} \left(1 - (-1)^k \sum_{j=0}^{m} a_j\, j^k - (-1)^{k-1} k \sum_{j=-1}^{m} b_j\, j^{k-1}\right) y^{(k)}(t_i)\, h^k\ . \end{aligned} \tag{13.5.20}$$

To compute the order of the predictor-corrector method, we make the localisation assumption that $w_{i-j} = y_{i-j} = y(t_{i-j})$ for $0 \le j \le m$. Using (13.5.20), we find that our explicit multistep method ($b_{-1} = 0$) of order p satisfies

$$y(t_{i+1}) - w_{i+1}^{[0]} = C_p h^{p+1} y^{(p+1)}(t_i) + O(h^{p+2})\ , \tag{13.5.21}$$

where

$$C_p = \frac{1}{(p+1)!} \left(1 + (-1)^p \sum_{j=0}^{m} a_j\, j^{p+1} - (-1)^p (p+1) \sum_{j=0}^{m} b_j\, j^p\right) .$$

Again, using (13.5.20), we find that our implicit multistep method of order $\tilde{p}$ satisfies

$$\begin{aligned} y(t_{i+1}) - w_{i+1}^{[k+1]} &= h\tilde{b}_{-1} \left(f(t_{i+1}, y(t_{i+1})) - f(t_{i+1}, w_{i+1}^{[k]})\right) + \left(y(t_{i+1}) - w_{i+1}\right) \\ &= h\tilde{b}_{-1} \left(f(t_{i+1}, y(t_{i+1})) - f(t_{i+1}, w_{i+1}^{[k]})\right) + \tilde{D}_{\tilde{p}} h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \\ &= h\tilde{b}_{-1} \frac{\partial f}{\partial y}(t_{i+s}, \eta_i) \left(y(t_{i+s}) - w_{i+s}^{[k]}\right) + \tilde{D}_{\tilde{p}} h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \end{aligned} \tag{13.5.22}$$

for some $\eta_{i,k}$ between $y(t_{i+1})$ and $w_{i+1}^{[k]}$, where

$$\tilde{D}_{\tilde{p}} = \frac{1}{(\tilde{p}+1)!} \left(1 + (-1)^{\tilde{p}} \sum_{j=0}^{\tilde{m}} \tilde{a}_j\, j^{\tilde{p}+1} - (-1)^{\tilde{p}} (\tilde{p}+1) \sum_{j=-1}^{\tilde{m}} \tilde{b}_j\, j^{\tilde{p}}\right) .$$

If $p \ge \tilde{p}$, we get from (13.5.22) with $k = 0$ that

$$\begin{aligned} y(t_{i+1}) - w_{i+1}^{[1]} &= = h\tilde{b}_{-1} \frac{\partial f}{\partial y}(t_{i+1}, \eta_{i.0}) \underbrace{\left(y(t_{i+1}) - w_{i+1}^{[0]}\right)}_{=O(h^{p+1})} + \tilde{D}_p h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \\ &= \tilde{D}_p h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \end{aligned}$$

because of (13.5.21). If we substitute this result into (13.5.22) with $k = 1$, we get

$$y(t_{i+s}) - w_{i+s}^{[2]} = = h\tilde{b}_{-1} \frac{\partial f}{\partial y}(t_{i+1}, \eta_{i,1}) \underbrace{\left(y(t_{i+1}) - w_{i+1}^{[1]}\right)}_{=O(y^{\tilde{p}+1})} + \tilde{D}_p h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2})$$

$$= \tilde{D}_p h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \ .$$

Proceeding this way with $k = 2, 3, \ldots, \nu - 1$, we get

$$y(t_{i+s}) - w_{i+s}^{[\nu]} = \tilde{D}_p h^{\tilde{p}+1} y^{(\tilde{p}+1)}(t_i) + O(h^{\tilde{p}+2}) \ .$$

This shows that the principal part of the local truncation error (the term in h with the smallest exponent) for the predictor-corrector method is given by the principal part of the corrector only. The predictor-corrector method is of order $\tilde{p}$.

Proceeding as we have just done, we find that

1. If $p < \tilde{p}$ and $\nu \geq \tilde{p} - p$, the predictor-corrector method has the same order as the corrector method. However, the principal part of the local truncation error for the predictor-corrector method is not the principal part of the local truncation error for the corrector method.

2. If $p < \tilde{p}$ and $\nu < \tilde{p} - p$, the predictor-corrector method is of order $p + \nu$. Each iteration of the implicit multistep method increases the order of the method by 1.

13.5.6 Variable Step-Size Multistep methods

We show how the predictor-corrector method of the previous section can be adapted to control the step-size.

Let $\tilde{\tau}_{i+1}(h)$ be the local truncation error for the Adams-Moulton method of order four. Combining the Adams-Bashforth Method of order four and the Adams-Moulton method of order four, we can determine the step-size h between t_i and t_{i+1} such that $\tilde{\tau}_{i+1}(h) < \epsilon$ where ϵ is given.

The following procedure outlines this variable step-size multistep method based on the Adams-Moulton method of order four and the Adams-Bashforth method of order four.

Algorithm 13.5.21 (Variable Step-Size Multistep method)

1. Let $i = 0$, $\tilde{t}_0 = t_0$ and $\tilde{w}_0 = y(t_0)$.

2. Use Runge-Kutta Method of order four starting with $\tilde{w}_0$ as approximation of $y(t_0)$ to get approximations $\tilde{w}_j$ of $y(t)$ at $\tilde{t}_j = \tilde{t}_0 + jh$ for $1 \leq j \leq 3$. Let $w_{i-j} = \tilde{w}_{3-j}$ and $t_{i-j} = \tilde{t}_{3-j}$ for $0 \leq j \leq 3$.

3. Use Adams-Bashforth formula to get a first approximation

$$w_{i+1}^{[0]} = w_i + \frac{h}{24}\left(55\, f(t_j.w_j) - 59\, f(t_{j-1}, w_{j-1}) + 37\, f(t_{j-2}, w_{j-2}) - 9\, f(t_{j-3}, w_{j-3})\right)$$

of $y(t)$ at $t_{j+1} = t_i + h$. .

4. Use Adams-Moulton formula to get a better (we hope) approximation
$$w_{i+1}^{[1]} = w_i + \frac{h}{24}\left(9\,f\left(t_{i+1}, w_{i+1}^{[0]}\right) + 19\,f(t_i.w_i) - 5\,f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})\right)$$
of $y(t)$ at $t_{i+1} = t_i + h$.

5. If
$$\frac{19}{270}\left|\frac{w_{i+1}^{[1]} - w_{i+1}^{[0]}}{h}\right| < \epsilon\ ,$$
we accept $w_{i+1} = w_{i+1}^{[1]}$, w_i, w_{i-1} and w_{i-2} as approximations of $y(t)$ at $t_{i+1} = t_i + h$, t_i, $t_{i-1} = t_0 - h$, and $t_{i-2} = t_0 - 2h$ respectively.

   (a) We choose a bigger step-size if
$$\frac{19}{270}\left|\frac{w_{i+1}^{[1]} - w_{i+1}^{[0]}}{h}\right| < \frac{\epsilon}{2}\ .$$
   We replace h by qh, where
$$q = \left|\left(\frac{270}{19}\right)\frac{h\epsilon}{w_{i+1}^{[1]} - w_{i+1}^{[0]}}\right|^{1/4}\ . \tag{13.5.23}$$
   q should be greater than 1 as we will show below. Set $\tilde{t}_0 = t_{i+1}$ and $\tilde{w}_0 = w_{i+1}$, increase i by 4, and go back to step 2.

   (b) We do not change the step-size if
$$\frac{\epsilon}{2} \leq \frac{19}{270}\left|\frac{w_{i+1}^{[1]} - w_{i+1}^{[0]}}{h}\right| < \epsilon\ .$$
   Increase i by 1 and go back to step 3.. Changing the step-size is expensive. So, we do not change it if there is little or no gain to make.

6. If
$$\frac{19}{270}\left|\frac{w_{i+1}^{[1]} - w_{i+1}^{[0]}}{h}\right| \geq \epsilon\ ,$$
we reduce the step-size. We replace h by qh, where q is defined in (13.5.23). q should be less than 1. If w_j for $i - 3 \leq j \leq i$ have already been accepted as good approximations (i.e. w_i comes from the Adams-Moulton method of order four), set $\tilde{t}_0 = t_i$ and $\tilde{w}_0 = w_i$, go back to step 2. Otherwise (i.e. w_i comes from the Runge-Kutta method of order four), just go back to step 2 with the same $\tilde{t}_0$ and $\tilde{w}_0$ but the new h.

We now justify non-rigorously this variable step-size multistep method.

We suppose that w_{i-j} has been accepted as an approximation of $y(t_{i-j})$ with the local truncation error for the Adams-Moulton method of order four less than ϵ for $0 \leq j \leq 3$. Moreover, we make the localization assumption that $w_{i-j} \approx y(t_{i-j})$ for $0 \leq j \leq 3$. Then, from Definition 13.5.5, we get

$$\frac{y(t_{i+1}) - w^{[0]}_{i+1}}{h} \approx \tau_{i+1}(h) = \frac{251}{720}\, y^{(5)}(\eta)h^4 \tag{13.5.24}$$

for some η between t_{i-3} and t_{i+1}, where $\tau_{i+1}(h)$ denotes the local truncation error for the Adams-Bashforth method of order four.

If we also assume that $y(t_{i+1}) \approx w^{[0]}_{i+1}$, we get from Definition 13.5.6 that

$$\frac{y(t_{i+1}) - w^{[1]}_{i+1}}{h} \approx \tilde{\tau}_{i+1}(h) = -\frac{19}{720}\, y^{(5)}(\xi)h^4 \tag{13.5.25}$$

for some ξ between t_{i-2} and t_{i+1}, where $\tilde{\tau}_{i+1}(h)$ denotes the local truncation error for the Adams-Moulton method of order four.

Finally, if we assume that $y^{(5)}(t)$ is almost constant on $[t_{i-3}, t_{i+1}]$ and subtract (13.5.25) from (13.5.24), we get

$$\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h} \approx \frac{270}{720}\, Y\, h^4 = \frac{3}{8}\, Y\, h^4 \ ,$$

where $Y \approx y^{(5)}(\xi) \approx y^{(5)}(\eta)$.

Thus $Y \approx \dfrac{8}{3}\left(\dfrac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h^5}\right)$. If we substitute $y^{(5)}(\xi)$ by Y in (13.5.25), we get

$$|\tilde{\tau}_{i+1}(h)| \approx \frac{19}{720}\left(\frac{8}{3}\left|\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h^5}\right|\right) h^4 = \frac{19}{270}\left|\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h}\right| \ .$$

If $\dfrac{19}{270}\left|\dfrac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h}\right| < \epsilon$, we may expect that $|\tilde{\tau}_{i+1}(h)| < \epsilon$.

If we use qh instead of h in the previous discussion (we keep the same estimate for Y), we get

$$|\tilde{\tau}_{i+1}(qh)| \approx \frac{19}{720}\, Y\,(qh)^4 \approx \frac{19}{720}\left(\frac{8}{3}\left|\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h^5}\right|\right)(qh)^4 = \frac{19}{270}\left|\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h}\right| q^4 \ .$$

To get $|\tilde{\tau}_{i+1}(qh)| < \epsilon$, we choose q such that

$$\frac{19}{270}\left|\frac{w^{[1]}_{i+1} - w^{[0]}_{i+1}}{h}\right| q^4 < \epsilon \ ;$$

namely,

$$q < \left(\frac{270\epsilon}{19}\left|\frac{h}{w^{[1]}_{i+1} - w^{[0]}_{i+1}}\right|\right)^{1/4} \ .$$

The following code implement the variable step-size multistep method outlined above.

Code 13.5.22 (Variable Step-Size Multistep Method)

To approximate the solution of the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(t_0) &= y_0 \end{aligned} .$$

Input: The maximal step-size hmax.
The minimal step-size hmin.
The maximal tolerated error T.
The initial time t_0 (t0 in the code below).
The final time t_f (tf in the code below).
The initial condition y_0 (y0 in the code below).
The function $f(t, y)$ (funct in the code below).
Output: The approximations w_i (gw(i+1) in the code below) of $y(t_i)$ at t_i (gt(i+1) in the code below) if all the requested requirements can be met.

```
function [gt,gw] = multistepABM(funct,t0,y0,tf,hmin,hmax,T)
  % last = 1  if we have reached  tf  or  h < hmin  at some point,
  % and  last = 0  otherwise.
  last = 0;
  h = hmin;
  gt(1) = t0;
  gw(1) = y0;
  t(1) = t0;
  w(1) = y0;

  % Given t0 and y0, we use Runge-Kutta of order four, to compute
  % an approximation w(i+1) of y(t0+i*h) for i = 1, 2 and 3 .
  % The code for the function rgkt4() was given previously.
  [t,w] = rgkt4(funct,h,3,t(1),w(1));

  % rkflag = 1  if the last stage used Runge-Kutta of order four and
  % rkflag = 0  otherwise,
  rkflag = 1;
  i=1:4;
  f(i) = funct(t(i),w(i));

  while (1==1)
    t(5) = t(4) + h;
    % We use the predictor-corrector method
    predict = w(4) + h*(55*f(4) - 59*f(3) + 37*f(2) - 9*f(1))/24;
    f(5) = funct(t(5),predict);
    correct = w(4) + h*(9*f(5) + 19*f(4) - 5*f(3) + f(2))/24;
    sigma = 19*abs(predict-correct)/(270*h);
    if (sigma < T)
      w(5) = correct;
      f(5) = funct(t(5),correct);
```

```
j=1:4;
t(j) = t(j+1);
w(j) = w(j+1);
f(j) = f(j+1);

if (rkflag==1)
  %  We accept the three values obtained from Runge-Kutta of order
  %  four and the one obtained with the predictor-corrector method.
  for j=1:4
    gt = [gt,t(j)];
    gw = [gw,w(j)];
  end
else
  % We accept the new value obtained with the predictor-corrector
  % method.  The other values have already been accepted.
  % It is at least the second time in a row that we apply
  % the predictor-corrector method.
  gt = [gt,t(4)];
  gw = [gw,w(4)];
end

if (last == 1)
  break;
end

% We have now executed at least one iteration of the
% predictor-corrector method
rkflag = 0;

if ( (t(4)+h > tf) || (sigma < T/2) )
  % We now choose a bigger step-size.
  if (sigma == 0)
    q = 4.0;
  else
    q = (T/sigma)^0.25;
  end
  h = min([hmax,4*h,q*h]);

  % We check that after the next stage  t  will not exceed  tf.
  if (t(4) + h > tf)
    %  We divide by four because we are now going to use
    %  rgkt4 and one step of Adams-Moulton to
    %  complete the integration;  we must therefore
    %  have that t(4) + 4*h = tf.
    h = (tf-t(4))/4;
    last = 1;
```

```
        end

        t(1) = t(4);
        w(1) = w(4);
        f(1) = f(4);
        [t,w] = rgkt4(funct,h,3,t(1),w(1));
        rkflag = 1;
        j=2:4;
        f(j) = funct(t(j),w(j));
      end
    else
      % We choose a smaller step-size.
      q = max([0.1, (T/sigma)^0.25]);
      h = q*h;

      if (h < hmin)
        gt = NaN;
        gw = NaN;
        break
      end

      % We start Runge-Kutta with  t(4)  and  w(4)  if
      % we have used the predictor-corrector method at the previous
      % stage.
      if (rkflag == 0)
        t(1) = t(4);
        w(1) = w(4);
        f(1) = f(4);
      end
      [t,w] = rgkt4(funct,h,3,t(1),w(1));
      rkflag = 1;
      last = 0;
      j=2:4;
      f(j) = feval(funct,t(j),w(j));
    end
  end
end
```

We now describe non-rigorously how to control the step-size for general variable step-size multistep methods.

We consider two multistep methods of order p to approximate the solution of (13.1.1). The first method is an explicit method given by

$$w_{i+1} = \sum_{j=0}^{m} a_j\, w_{i-j} + h \sum_{j=0}^{m} b_j\, f(t_{i-j}, w_{i-j}) \tag{13.5.26}$$

for $m \leq i < N$ and the second method is an implicit method given by

$$\tilde{w}_{i+1} = \sum_{j=0}^{\tilde{m}} \tilde{a}_j \, \tilde{w}_{i-j} + h \sum_{j=-1}^{\tilde{m}+1} \tilde{b}_j \, f(t_{i-j}, \tilde{w}_{i-j}) \tag{13.5.27}$$

for $\tilde{m} \leq i < N$. We have that w_{i-j} is the approximation of $y(t_{i-j})$ given by (13.5.26) and $\tilde{w}_{i-j}$ is the approximation of $y(t_{i-j})$ given by (13.5.27).

In the proof of Theorem 13.5.10, we have shown that a multistep method of the form (13.5.1) satisfies

$$\begin{aligned}
&y(t_i + h) - \sum_{j=0}^{m} a_j \, y(t_i - jh) - h \sum_{j=-1}^{m} b_j \, f(t_i - jh, y(t_i - jh)) \\
&= \left(1 - \sum_{j=0}^{m} a_j\right) y(t_i) + \left(1 + \sum_{j=0}^{m} a_j \, j - \sum_{j=-1}^{m} b_j\right) y'(t_i) \, h \\
&\quad + \sum_{k=2}^{\infty} \frac{1}{k!} \left(1 - (-1)^k \sum_{j=0}^{m} a_j \, j^k - (-1)^{k-1} k \sum_{j=-1}^{m} b_j \, j^{k-1}\right) y^{(k)}(t_i) \, h^k \ .
\end{aligned} \tag{13.5.28}$$

Using (13.5.28) with $b_{-1} = 0$ and the localization assumption $w_{i-j} = y_{i-j} = y(t_{i-j})$ for $0 \leq j \leq m$, we find that the first multistep method of order p satisfies

$$y(t_{i+1}) - w_{i+1} = C_p h^{p+1} y^{(p+1)}(t_i) + O(h^{p+2}) \ , \tag{13.5.29}$$

where

$$C_p = \frac{1}{(p+1)!} \left(1 + (-1)^p \sum_{j=0}^{m} a_j \, j^{p+1} - (-1)^p (p+1) \sum_{j=0}^{m} b_j \, j^p\right) .$$

Similarly, $\tilde{w}_{i-j} = y_{i-j} = y(t_{i-j})$ for $0 \leq j \leq m$ and $w_{i+1} \approx y_{i+1} = y(t_{i+1})$, the second multistep method of order p satisfies

$$y(t_{i+1}) - \tilde{w}_{i+1} = \tilde{D}_p h^{p+1} y^{(p+1)}(t_i) + O(h^{p+2}) \ , \tag{13.5.30}$$

$$\tilde{C}_p = \frac{1}{(p+1)!} \left(1 + (-1)^p \sum_{j=0}^{\tilde{m}} a_j \, j^{p+1} - (-1)^p (p+1) \sum_{j=-1}^{\tilde{m}} b_j \, j^p\right) .$$

If we assume that $y^{(p+1)}(t)$ is almost constant on an interval $[t_{i-m}, t_{i+1}]$, we may choose K such that $y^{(p+1)}(t_i) \approx K$. Hence,

$$y(t_{i+1}) - w_{i+1} \approx C_p K h^{p+1} + O(h^{p+2}) \quad \text{and} \quad y(t_{i+1}) - \tilde{w}_{i+1} = \tilde{C}_p K h^{p+1} + O(h^{p+2}) \ .$$

Thus, after subtracting the second expression from the first expression, we get

$$\tilde{w}_{i+1} - w_{i+1} \approx (C_p - \tilde{C}_p) K h^{p+1} + O(h^{p+2}) \ .$$

If we ignore the small error of order h^{p+2}, we get

$$K h^{p+1} \approx \frac{\tilde{w}_{i+1} - w_{i+1}}{C_p - \tilde{C}_p}$$

if $C_p \neq \tilde{C}_p$. If we substitute this expression into

$$y(t_{i+s}) - w_{i+s} \approx C_p K h^{p+1} + O(h^{p+2}) \ ,$$

we get

$$y(t_{i+s}) - w_{i+s} \approx \frac{C_p}{C_p - \tilde{C}_p} \left(\tilde{w}_{i+1} - w_{i+1}\right) \ . \tag{13.5.31}$$

Let δ be a small number. We may require

$$\left| \frac{C_p}{C_p c - \tilde{D}_p} \left(\tilde{w}_{i+s} - w_{i+s}\right) \right| < \delta$$

at each step. This is called **error control per step**. If the requirement is not satisfied, we reduce the step-size h. We may instead require

$$\left| \frac{C_p}{C_p c - \tilde{C}_p} \left(\tilde{w}_{i+s} - w_{i+s}\right) \right| < \delta h$$

at each step. This is called **error control per unit step**. This takes care of the accumulation of error at each step (assuming that the error is evenly distributed among the integration steps). We have that $N\delta h = \delta(b-a)$ is the **cumulative error**.

13.6 Convergence, Consistency and Stability

The content of this section is based in great part on [19, 20].

We consider the initial value problem (13.1.1), where we assume that $f : [t_0, t_f] \times \mathbb{R} \to \mathbb{R}$ has continuous mixed derivatives of sufficiently high order. This implies that the solution y of (13.1.1) is sufficiently differentiable.

In this section, we consider multistep methods of the form

$$\begin{aligned} w_{i+1} &= \sum_{j=0}^{m} a_j w_{i-j} + hF(h, \mathbf{t}, \mathbf{w}, f) \quad , \quad m \leq i < N \\ w_i &= y(t_i) \quad , \quad 0 \leq i \leq m \end{aligned} \tag{13.6.1}$$

where $a_m \neq 0$, $\mathbf{t} = \begin{pmatrix} t_{i-m} & \ldots & t_{i-1} & t_i & t_{i+1} \end{pmatrix}^\top$ and $\mathbf{w} = \begin{pmatrix} w_{i-m} & \ldots & w_{i-1} & w_i & w_{i+1} \end{pmatrix}^\top$. We assume that

$$F(h, \mathbf{t}, \mathbf{w}, 0) = 0 \tag{13.6.2}$$

and

$$\left| F\left(h, \mathbf{t}, \mathbf{w}^{[1]}, f\right) - F\left(h, \mathbf{t}, \mathbf{w}^{[2]}, f\right) \right| \leq R \left\| \mathbf{w}^{[1]} - \mathbf{w}^{[2]} \right\|_1 = R \sum_{j=-1}^{m} \left| w_{i-j}^{[1]} - w_{i-j}^{[2]} \right| \tag{13.6.3}$$

for a constant R.

1. The Multistep methods defined in Definition 13.5.1 are of the form (13.6.1), and satisfy (13.6.2) and (13.6.3).

   We have
$$F(h,\mathbf{t},\mathbf{w},f) = \sum_{j=-1}^{m} b_j f(t_{i-j},w_{i-j}) \ .$$

   Obviously, $F(h,\mathbf{t},\mathbf{w},0)=0$. Suppose that L is the Lipschitz constant in the definition of a well posed differential equation; namely, L is such that $|f(t,x)-f(t,y)|\leq L|x-y|$ for all (t,x) and (t,y) in the domain of f. We have that
$$\begin{aligned}\left|F\left(h,\mathbf{t},\mathbf{w}^{[1]},f\right)-F\left(h,\mathbf{t},\mathbf{w}^{[2]},f\right)\right| &\leq \sum_{j=-1}^{m}|b_j|\left|f\left(t_{i-j},w_{i-j}^{[1]}\right)-f\left(t_{i-j},w_{i-j}^{[2]}\right)\right| \\ &\leq L\sum_{j=-1}^{m}|b_j|\left|w_{i-j}^{[1]}-w_{i-j}^{[2]}\right| \leq L\max_{-1\leq j\leq m}|b_j|\sum_{j=-1}^{m}\left|w_{i-j}^{[1]}-w_{i-j}^{[2]}\right| = R\left\|\mathbf{w}^{[1]}-\mathbf{w}^{[2]}\right\|_1\end{aligned}$$

   for $R = L\max_{-1\leq j\leq m}|b_j|$.

2. The Runge-Kutta methods defined in Definition 13.4.1 are also of the form (13.6.1), and satisfy (13.6.2) and (13.6.3).

   We have
$$F(h,\mathbf{t},\mathbf{w},f) = \sum_{j=1}^{s}\gamma_j K_j \ ,$$

   where $\mathbf{t}=(t_i)$ and $\mathbf{w}=(w_i)$ since Runga-Kutta methods are one-step methods.

   By definition of the K_j, we have that $F(h,\mathbf{t},\mathbf{w},0)=0$. As before, let L be the Lipschitz constant in the definition of a well posed differential equation; namely, L is such that $|f(t,x)-f(t,y)|\leq L|x-y|$ for all (t,x) and (t,y) in the domain of f. Let
$$K_j^{[k]} = f(t_i+\alpha_j h, w_i^{[k]} + h\sum_{m=1}^{s}\beta_{j,m}K_m^{[k]})$$

   for $k=1$ and 2. To verify this claim, we first note that
$$\begin{aligned}\left|K_j^{[1]}-K_j^{[2]}\right| &= \left|f(t_i+\alpha_j h, w_i^{[1]}+h\sum_{m=1}^{s}\beta_{j,m}K_m^{[1]}) - f(t_i+\alpha_j h, w_i^{[2]}+h\sum_{m=1}^{s}\beta_{j,m}K_m^{[2]})\right| \\ &\leq L\left|w_i^{[1]}-w_i^{[2]}\right| + Lh\sum_{m=1}^{s}\beta_{j,m}\left|K_m^{[1]}-K_m^{[2]}\right| \quad , \quad 1\leq j\leq s \ .\end{aligned} \tag{13.6.4}$$

   Let
$$B = \begin{pmatrix}\beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,s}\\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,s}\\ \vdots & \vdots & \ddots & \vdots\\ \beta_{s,1} & \beta_{s,2} & \dots & \beta_{s,s}\end{pmatrix} \ , \ \mathbf{K} = \begin{pmatrix}\left|K_1^{[1]}-K_1^{[2]}\right|\\ \left|K_2^{[1]}-K_2^{[2]}\right|\\ \vdots\\ \left|K_s^{[1]}-K_s^{[2]}\right|\end{pmatrix} \text{ and } \mathbf{W} = \begin{pmatrix}\left|w_i^{[1]}-w_i^{[2]}\right|\\ \left|w_i^{[1]}-w_i^{[2]}\right|\\ \vdots\\ \left|w_i^{[1]}-w_i^{[2]}\right|\end{pmatrix} \ .$$

We can rewrite (13.6.4) as

$$\mathbf{K} \leq L\mathbf{W} + LhB\mathbf{K} \ ,$$

where the inequality is component by component. We have that

$$\|\mathbf{K}\|_1 \leq L\|\mathbf{W}\|_1 + Lh\|B\|_1\|\mathbf{K}\|_1 \ .$$

Thus

$$\left|K_j^{[1]} - K_j^{[2]}\right| \leq \|\mathbf{K}\|_1 \leq \frac{L}{1 - Lh\|B\|_1}\|\mathbf{W}\|_1 \leq 2L\|\mathbf{W}\|_1 = 2Ls\left|w_i^{[1]} - w_i^{[2]}\right| \quad , \quad 1 \leq j \leq s \ ,$$

if we assume that h is small enough to have $Lh\|B\|_1 < 1/2$ [3]. Finally, we have that

$$\begin{aligned}\left|F\left(h, \mathbf{t}, \mathbf{w}^{[1]}, f\right) - F\left(h, \mathbf{t}, \mathbf{w}^{[2]}, f\right)\right| &\leq \sum_{j=1}^{s} \gamma_j \left|K_j^{[1]} - K_j^{[2]}\right| \leq \underbrace{\sum_{j=1}^{s} \gamma_j}_{=1} \ 2Ls\left|w_i^{[1]} - w_i^{[2]}\right| \\ &= 2Ls\left|w_i^{[1]} - w_i^{[2]}\right| = R\left|w_i^{[1]} - w_i^{[2]}\right|\end{aligned}$$

for $R = 2Ls$.

To define the stability of a numerical method, we consider a perturbation of (13.6.1) given by the difference equation

$$\begin{aligned} u_{i+1} &= \sum_{j=0}^{m} a_j u_{i-j} + hF(h, \mathbf{t}, \mathbf{w}, f) + \delta_{i+1}(h) \quad , \quad m \leq i < N \\ u_i &= y(t_i) + \delta_i(h) \quad , \quad 0 \leq i \leq m \end{aligned} \tag{13.6.5}$$

Convergence and consistency are two primordial concepts. Convergence obviously does not need any motivation.

Definition 13.6.1

A multistep method of the form (13.6.1) [4] is **convergent** if, for all well-posed initial value problems (13.1.1),

$$\lim_{h \to 0} \max_{0 \leq i \leq N} |y_i - u_i| = 0 \ ,$$

where u_i is the numerical approximation of w_i.

Remark 13.6.2

To prove convergence of a multistep method, we will require that $\max_{0\leq i\leq N} |\delta_i(h)| \to 0$ as $h \to 0$. Obviously, in practice, this is not realistic. We do not expect round off errors to go to 0 as h goes to 0. However, our theoretical result shows that by having $\max_{0\leq i\leq N} |\delta_i(h)|$ very small, we

[3] Any value smaller than 1 could have been used.

[4] From now on, this will refer to Runge-Kutta methods (Definition 13.4.1) and multistep methods (Definition 13.5.1).

may hope that our numerical approximation of the solution be very accurate. To decrease round off errors (by choosing the right algorithm, by efficiently programming it, ...) is one of the big challenges in numerical analysis. ♠

Remark 13.6.3
A definition of convergence that is often given in textbooks is the following.

A multistep method is convergent if, for all well-posed initial value problems (13.1.1),

$$\lim_{h\to 0} \max_{0\leq i\leq N} |y_i - w_i| = 0 \ .$$

All rounding errors are assumed to be null.

For the Euler's method, if we ignore rounding errors in Theorem 13.2.5 (i.e. $\delta = \delta_0 = 0$), we get

$$\max_{0\leq i\leq N} |y_i - w_i| \leq \frac{Mh}{2L}\left(e^{L(t_f-t_0)} - 1\right) \to 0$$

as $h \to 0$. So, the Euler's method is converging in this weak sense. Unfortunately, this does not prove that the Euler's method is convergent according to Definition 13.6.1. In fact, we will need to assume (the unrealistic assumption) that $\max\limits_{0\leq i\leq N} |\delta_i(h)| = O(h^2)$ to be able to show that Euler's method is converging in the sense of Definition 13.6.1.

Another example is given by the Trapezoidal Method. It is convergent in the weak sense above. For this example, we refer to Definition 13.5.3 and the paragraphs following this definition. We prove that

$$\lim_{h\to 0} \max_{0\leq i\leq N} |w_i - y(t_i)| = 0 \ . \tag{13.6.6}$$

Let $e_i = w_i - y(t_i)$. If we subtract

$$y(t_{i+1}) = y(t_i) + \frac{h}{2}\left(f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right) + M(\xi_i,\eta_i)\, h^3$$

from

$$w_{i+1} = w_i + \frac{h}{2}\left(f(t_{i+1}, w_{i+1}) + f(t_i, w_i)\right) \ ,$$

we get

$$e_{i+1} = e_i + \frac{h}{2}\Bigg(f(t_i, w_i) - f(t_i, y(t_i)) + f(t_{i+1}, w_{i+1}) - f(t_{i+1}, y(t_{i+1}))\Bigg) - M(\xi_i,\eta_i)\, h^3 \ . \tag{13.6.7}$$

As we have seen when computing the order of the Trapezoidal Method if we assume that f is twice continuously differentiable on $[t_0, t_f] \times \mathbb{R}$, then $|M(\xi_i,\eta_i)| \leq Q$ for all i, where $Q = 5K/12$ and K is the maximum of $y^{(3)}(t)$ on $[t_0, t_f]$. If we use this property and the assumption that f satisfies the Lipschitz condition (13.1.3), we get from (13.6.7) that

$$|e_{i+1}| \leq |e_i| + \frac{hL}{2}\left(|e_i| + |e_{i+1}|\right) + Q\, h^3 \ . \tag{13.6.8}$$

Since $h \to 0$, we may assume that $hL/2 < 1$. Hence, we get from (13.6.8) that

$$|e_{i+1}| \leq \frac{1 + hL/2}{1 - hL/2}|e_i| + \frac{Q\, h^3}{1 - hL/2} \ . \tag{13.6.9}$$

We now show by induction that

$$|e_i| \leq \frac{Q}{L}\left(\left(\frac{1+hL/2}{1-hL/2}\right)^i - 1\right)h^2 \quad , \quad 0 \leq i \leq N \ . \tag{13.6.10}$$

The result is true for $i = 0$ because we assume that $w_0 = y_0$. Suppose (13.6.10) it is true for i. Using (13.6.9) and (13.6.10), we get

$$\begin{aligned}
|e_{i+1}| &\leq \frac{1+hL/2}{1-hL/2}|e_i| + \frac{Q\,h^3}{1-hL/2} \leq \frac{1+hL/2}{1-hL/2}\left(\frac{Q}{L}\left(\left(\frac{1+hL/2}{1-hL/2}\right)^i - 1\right)h^2\right) + \frac{Q\,h^3}{1-hL/2} \\
&= \frac{Q}{L}\left(\frac{1+hL/2}{1-hL/2}\right)^{i+1} h^2 + \frac{Qh^2}{L(1-hL/2)}\left(-\left(1+\frac{hL}{2}\right) + Lh\right) \\
&= \frac{Q}{L}\left(\frac{1+hL/2}{1-hL/2}\right)^{i+1} h^2 - \frac{Qh^2}{L} = \frac{Q}{L}\left(\left(\frac{1+hL/2}{1-hL/2}\right)^{i+1} - 1\right)h^2 \ .
\end{aligned}$$

So (13.6.10) is true for i replaced by $i+1$, completing the proof by induction. Since $0 < hL/2 < 1$, we have that

$$0 < \frac{1+hL/2}{1-hL/2} = 1 + \frac{hL}{1-hL/2} \leq \sum_{j=0}^{\infty} \frac{1}{j!}\left(\frac{hL}{1-hL/2}\right)^j = e^{hL/(1-hL/2)} \ .$$

Thus

$$|e_i| \leq \frac{Q}{L}\left(\frac{1+hL/2}{1-hL/2}\right)^i h^2 \leq \frac{Q}{L}e^{ihL/(1-hL/2)}\,h^2 \leq \frac{Q}{L}e^{(t_f-t_0)L/(1-hL/2)}\,h^2 \quad , \quad 0 \leq i \leq N \ .$$

Hence

$$\max_{0\leq i\leq N}|w_i - y(t_i)| = \max_{0\leq i\leq N}|e_i| \leq \frac{Q}{L}e^{(t_f-t_0)L/(1-hL/2)}\,h^2 \to 0$$

as $h \to 0$. This proves (13.6.6). ♠

Consistency ensure that the numerical method approximates adequately the differential equation.

Definition 13.6.4

The **local truncation error** of a multistep method of the form (13.6.1) is defined by

$$\tau_{i+1}(h) = \frac{1}{h}\left(y_{i+1} - \sum_{j=0}^{m} a_j y_{i-j}\right) - F(h, \mathbf{t}, \mathbf{y}, h) \quad , \quad 0 \leq i < N \ ,$$

where $\mathbf{y} = \begin{pmatrix} y_{i-m} & \ldots & y_{i-1} & y_i & y_{i+1}\end{pmatrix}^\top$ and $y_i = y(t_i)$ for all i as usual.
we say that a multistep method is **consistent** if, for each well posed initial value problems (13.1.1), there exists a function $\tau : \mathbb{R} \to \mathbb{R}$ such that

$$\max_{0\leq i<N}|\tau_{i+1}(h)|| \leq \tau(h) \to 0$$

as $h \to 0$.

A finite difference problem of order greater than 0 is consistent.

For the Runga-Kutta methods given in Definition 13.4.1, the local truncation error is defined by

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - h \sum_{j=1}^{s} \gamma_j K_j \quad , \quad 0 \leq i < N \ ,$$

where

$$K_j = f(t_i + \alpha_j h, y_i + h \sum_{k=1}^{s} \beta_{j,k} K_k) \ ,$$

For the multistep methods given in Definition 13.5.1, the local truncation error is defined by

$$\tau_{i+1}(h) = \frac{1}{h}\left(a_{i+1} - \sum_{j=0}^{m} a_j y(t_{i+j}) \right) - \sum_{j=-1}^{m} b_j f(t_{i-j}, y_{i-j}) \quad , \quad m \leq i < N \ .$$

Remark 13.6.5
In the definitions of convergence and consistency, we consider the limit when $h \to 0$. We have to keep in mind that $h = (t_f - t_0)/N$ and that $N \to \infty$. It would have been more appropriate to write h_N instead of h in these definitions but we will stick to the tradition of only writing h. ♠

Example 13.6.6
One of the simplest multistep methods is obviously the Euler's method.

Assume that f in the initial value problem (13.1.1) satisfies a Lipschitz condition on $[t_0, t_f] \times \mathbb{R}$ with respect to the second variable and that L is the Lipschitz constant. Moreover, assume that $|y''|$ is bounded by M on $[t_0, t_f]$ where y is the solution of (13.1.1).

The Euler's method is consistent with respect to (13.1.1) because

$$|\tau_{i+1}(h)| = \left| \frac{h}{2} y''(\xi_i) \right| \leq \tau(h) \equiv \frac{Mh}{2} \to 0$$

as $h \to 0$. ♣

Example 13.6.7
Consider the following Adams-Bashforth Method of order two.

$$\begin{aligned} w_{i+1} &= w_{i-1} + 2hf(t_i, w_i) \quad , \quad 1 \leq i < N \\ w_0 &= y_0 \\ w_1 &= y_1 \end{aligned}$$

This method is obtained by taking $m = 1$ and $q = 1$ in (13.5.5). We now show that this is a consistent method of order two. Since

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(t_i) + \frac{h^3}{3!} y'''(\xi_i)$$

and

$$y(t_{i-1}) = y(t_i) - hy'(t_i) + \frac{h^2}{2}y''(t_i) - \frac{h^3}{3!}y'''(\nu_i)$$

for some number ξ_i and ν_i between t_{i-1} and t_{i+1}, and $f(t_i, y_i) = y'(t_i)$, we get

$$\begin{aligned}\tau_{i+1}(h) &= \frac{y_{i+1} - y_{i-1}}{h} - 2f(t_i, y_i) \\ &= \frac{\left(y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{3!}y'''(\xi_i)\right) - \left(y(t_i) - hy'(t_i) + \frac{h^2}{2}y''(t_i) - \frac{h^3}{3!}y'''(\nu_i)\right)}{h} \\ &\qquad - 2y'(t_i) \\ &= \left(\frac{y'''(\xi_i)}{3!} + \frac{y'''(\nu_i)}{3!}\right)h^2 \ .\end{aligned}$$

Hence,

$$|\tau_{i+1}(h)| \leq \tau(h) \equiv \frac{2}{3}\max_{t_0\leq t\leq t_f}|y'''(t)|h^2 = \frac{2M}{3}\,h^2 \quad , \quad 1 \leq i < N \ ,$$

where $M = \max_{t_0\leq t\leq t_f}|y'''(t)|$. Therefore, the method is of order two and $|\tau_{i+1}(h)| \leq \tau(h) \to 0$ as $h \to 0$. The method is therefore consistent. ♣

The definition of stability that we adopt is given below.

Definition 13.6.8

A multistep method of the form (13.6.1) is **zero-stable** if, for any well-posed initial value problems (13.1.1), there exist S and h_0 such that for any partition of $[t_0, t_f]$ with $h < h_0$, any solution $\{u_i^{[j]}\}_{i=0}^N$ of (13.6.5) with $\delta_i = \delta_i^{[j]}$ for $j = 1$ and 2, then

$$|u_i^{[1]} - u_i^{[2]}| < S\epsilon$$

for $i = 0, 1, \ldots, N$ whenever $|\delta_i^{[1]} - \delta_i^{[2]}| < \epsilon$ for $i = 0, 1, \ldots, N$.

Remark 13.6.9
This definition is reminiscent of the definition of stability for systems of linear equations. A system of the form $A\mathbf{x} = \mathbf{b}$, where A is a $n \times n$ matrix, is stable is there exists a constant K such that $\|\mathbf{x}\| \leq K\|A\mathbf{x}\|$ for all $\mathbf{x}$. This ensures that if $\tilde{\mathbf{b}}$ is a slight perturbation of $\mathbf{b}$ than the solution $\mathbf{x}_{\tilde{b}}$ of $A\mathbf{x} = \tilde{\mathbf{b}}$ is a slight perturbation of the solution $\mathbf{x}_b$ of $A\mathbf{x} = \mathbf{b}$ because

$$\|\mathbf{x}_{\tilde{b}} - \mathbf{x}_b\| \leq K\|A\mathbf{x}_{\tilde{b}} - A\mathbf{x}_b\| = K\|\tilde{\mathbf{b}} - \mathbf{b}\| \ .$$

♠

13.6.1 Consistency

Proposition 13.6.10

Runge-Kutta methods are consistent if and only if

$$\sum_{j=1}^{s} \gamma_j = 1 \ .$$

Proof.
Since $y(t_{i+1}) = y(t_i) + hy'(\xi_i)$ for some ξ_i between t_i and t_{i+1}, we have

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - \sum_{j=1}^{s} \gamma_j K_j = y'(\xi_i) - \sum_{j=1}^{s} \gamma_j K_j$$

for $0 \leq i < N$. Let $c \in [a,b]$ be a fixed value such that $t_i \leq c \leq t_{i+1}$ for all h. So i will increase and converge to ∞ as h goes to 0 to ensure that $t_i \leq c \leq t_{i+1}$. We have that $t_{i+1} \to c$, $t_i \to c$, $y_i \to y(c)$ and $\xi_i \to c$ as $h \to 0$. Thus

$$\lim_{h\to 0} \tau_{i+1}(h) = y'(c) - \sum_{j=1}^{s} \gamma_i f(c, y(c)) = y'(c)\left(1 - \sum_{j=1}^{s} \gamma_j\right) = 0$$

if and only if

$$\sum_{j=1}^{s} \gamma_i = 1 \ . \qquad \blacksquare$$

So, all our Runge-Kutta methods are consistent since we require $\displaystyle\sum_{j=1}^{s} \gamma_j = 1$.

Proposition 13.6.11

If the multistep method given in Definition 13.5.1 is consistent, then

$$1 = \sum_{i=0}^{m} a_i \quad \text{and} \quad \sum_{k=-1}^{m} b_k = \sum_{k=0}^{m} a_k\,(k+1) \ .$$

Proof.
From the Mean Value Theorem, we have that $y_{i-k} = y(t_{i-k}) = y(t_{i+1}) - (k+1)hy'(\xi_{i-k})$ for some $\xi_{i-k} \in [t_{i-k}, t_{i+1}]$, where $0 \leq k \leq m$. If we substitute these expressions in the definition of local truncation error given in Definition 13.6.4, we get

$$\tau_{i+1}(h) = \left(1 - \sum_{k=0}^{m} a_k\right)\frac{y_{i+1}}{h} + \sum_{k=0}^{m} a_k\,(k+1)y'(\xi_{i-k}) - \sum_{k=-1}^{m} b_k f(t_{i-k}, y_{i-k}) \ . \tag{13.6.11}$$

Choose an increasing sequence $\{N_j\}_{j=1}^{\infty}$ of positive integers converging to ∞ such that there always is a value i_j of i such that $t_{i_j+1} = c$, a constant value, when $N = N_j$. We have that $i_j \to \infty$, $h = h_j \equiv \dfrac{t_f - t_0}{N_j} \to 0$, $t_{i_j-k} \to c$ and $\xi_{i-k} \to c$ for all $0 \leq k \leq m$ as $N_j \to \infty$ because

$t_{i+1} - t_{i-m} = (m+1)h_j \to 0$ as $h_j \to 0$. If we assume that the multistep method is consistent, then $h = h_j \to 0$ in (13.6.11) yields

$$0 = \left(1 - \sum_{k=0}^{m} a_k\right) y(c) \quad \text{and} \quad 0 = \sum_{k=0}^{m} a_k\,(k+1)y'(c) - \sum_{k=-1}^{m} b_m f(c, y(c)) \ .$$

We get $1 - \sum_{k=0}^{m} a_k = 0$ from the first equation and, because $y'(t) = f(t, y(t))$, we get

$$0 = \sum_{k=0}^{m} a_k\,(k+1) - \sum_{k=-1}^{m} b_k$$

from the second equation. ∎

Remark 13.6.12
We can easily show that if a consistent multistep method is converging according to the definition given Remark 13.6.3; namely $\max_{0 \le i \le N} |w_i - y_i| \to 0$ as $h \to 0$, then $1 = \sum_{i=0}^{m} a_i$. ♠

13.6.2 Finite Difference Equations

Before diving deeper into the analysis of multistep methods, we need to introduce some notions about finite difference equations.

Definition 13.6.13

Consider the **finite difference equation**

$$\begin{aligned} \sum_{j=0}^{s} a_j\, u_{i-j} &= C_i \quad , \quad i \ge s \\ u_i &= v_i \quad , \quad 0 \le i < s \end{aligned} \tag{13.6.12}$$

where the constants a_j for $0 \le j \le s$, v_j for $0 \le j < s$, and C_i for $i \ge s$ are given. A sequence $\{u_i\}_{i=0}^{\infty}$ that satisfies (13.6.12) is called a **solution** of (13.6.12).
If $a_s\, a_0 \ne 0$, the finite difference equation is said to be of **order** s.

Theorem 13.6.14

If (13.6.12) is of order s, then there is a unique solution of (13.6.12).

Proof.
Since $a_0 \ne 0$, the existence of the solution follows recursively from

$$u_i = -\frac{1}{a_0} \sum_{j=1}^{s} a_j\, u_{i-j} + C_i \quad , \quad i \ge s \ , \tag{13.6.13}$$

with $u_i = v_i$ for $0 \leq i < s$.

Suppose that there are two solutions $\left\{u_i^{[1]}\right\}_{i=0}^{\infty}$ and $\left\{u_i^{[2]}\right\}_{i=0}^{\infty}$ of (13.6.12). Then, $\{u_i\}_{i=0}^{\infty}$ with $u_i = u_i^{[1]} - u_i^{[2]}$ for all $i \geq 0$ is a solution of (13.6.12) with $C_i = 0$ for $i \geq s$ and $u_i = 0$ for $0 \leq i < s$. It follows from (13.6.13) that $u_i = u_i^{[1]} - u_i^{[2]} = 0$ for $i \geq 0$. ∎

Consider the **homogeneous finite difference equation**

$$\sum_{j=0}^{s} a_j\, u_{i-j} = 0 \quad , \quad i \geq s \; , \tag{13.6.14}$$

of order s.

It is clear that a linear combination of solutions of (13.6.14) is a solution of (13.6.14). Moreover, it follows from (13.6.13) with $C_i = 0$ for all $i \geq s$ that the linear independence of solutions $\{u_i^{[j]}\}_{i=0}^{\infty}$ of (13.6.14) for $1 \leq j \leq k$ is completely determined by the linear independence of the vectors $\begin{pmatrix} u_0^{[j]} & u_1^{[j]} & \ldots & u_{s-1}^{[j]} \end{pmatrix}^\top$ in $\mathbb{R}^s$ for $1 \leq j \leq k$. It follows that (13.6.14) may have s linearly independent solutions. For instance, given $0 \leq k < s$, let $\left\{u_i^{[k]}\right\}_{i=0}^{\infty}$ be the solution of

$$\begin{aligned} \sum_{j=0}^{s} a_j\, u_{i-j} &= 0 \quad , \quad i \geq s \\ u_i &= \delta_{k,i} \quad , \quad 0 \leq i < s \end{aligned}$$

where

$$\delta_{k,i} = \begin{cases} 0 & \text{if} \quad i \neq k \\ 1 & \text{if} \quad i = k \end{cases}$$

is the well known Dirac Delta function. The solutions $\left\{u_i^{[0]}\right\}_{i=0}^{\infty}$, $\left\{u_i^{[1]}\right\}_{i=0}^{\infty}$, $\ldots$, $\left\{u_i^{[s-1]}\right\}_{i=0}^{\infty}$ form a set of s linearly independent solutions of (13.6.14).

Definition 13.6.15

A set of s linearly independent solutions of an homogeneous finite difference equation of order s is called a **fundamental set of solutions**.

Theorem 13.6.16

Let $\{u_i^{[k]}\}_{i=0}^{\infty}$ for $0 \leq k < s$ be a fundamental set of solutions of the homogeneous finite difference equation (13.6.14). Then, the solution of

$$\begin{aligned} \sum_{j=0}^{s} a_j\, u_{i-j} &= 0 \quad , \quad i \geq s \\ u_i &= v_i \quad , \quad 0 \leq i < s \end{aligned}$$

can be expressed uniquely as a linear combination of $\left\{u_i^{[k]}\right\}_{i=0}^{\infty}$ for $0 \leq k < s$.

Proof.
We have to find α_0, α_1, ..., α_{s-1} such that

$$\sum_{k=0}^{s-1} \alpha_k \, u_i^{(k)} = v_i$$

for $0 \leq i < s$. Namely, we have to solve $A\alpha = \mathbf{v}$ where

$$A = \begin{pmatrix} u_0^{(0)} & u_0^{(1)} & \dots & u_0^{(s-1)} \\ u_1^{(0)} & u_1^{(1)} & \dots & u_1^{(s-1)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{s-1}^{(0)} & u_{s-1}^{(1)} & \dots & u_{s-1}^{(s-1)} \end{pmatrix} \quad , \quad \mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ u_{s-1} \end{pmatrix} \quad \text{and} \quad \alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{s-1} \end{pmatrix} .$$

Since $\{u_i^{[k]}\}_{i=0}^{\infty}$ for $0 \leq k < s$ are linearly independent, the columns of A must also be linearly independent. Thus the matrix A is invertible. Hence there is a unique solution to $A\alpha = \mathbf{v}$. ∎

Suppose that $u_i = z^i$ for $i \geq 0$ is a solution of the homogeneous finite difference equation (13.6.14). If we substitute this formula for u_i in (13.6.14) and factor out z^{i-s}, we get the **characteristic polynomial**

$$\sum_{j=0}^{s} a_j \, z^{s-j} = 0 \ . \tag{13.6.15}$$

If r is a real root of the characteristic polynomial, then $\{u_i\}_{i=0}^{\infty}$ with $u_i = r^i$ is a solution for (13.6.14). We note that $r \neq 0$ because we assume that $a_s \neq 0$. If we could find s distinct roots of the characteristic polynomial, then we will have s solutions that provide a fundamental set of solutions for (13.6.14). Unfortunately, not all polynomials of degree s have s distinct real roots. If we work in $\mathbb{C}$, we can describe all the solutions of (13.6.14).

Proposition 13.6.17

If $r \in \mathbb{C}$ is a root of algebraic multiplicity m of the characteristic polynomial (13.6.15) with $a_s \neq 0$, then $\left\{u_i^{[k]}\right\}_{i=0}^{\infty}$ with $u_i = i^k r^i$ and $0 \leq k < m$ are m linearly independent solutions for (13.6.14).

Proof.
Let

$$p(z) = \sum_{j=0}^{s} a_j \, z^{s-j}$$

If r is a root of p of algebraic multiplicity m (so a zero of order m of p), we have that $p(r) = p'(r) = \ldots = p^{(m-1)}(r) = 0$ and $p^{(m)}(r) \neq 0$. Let $q(z) = z^{i-s}p(z)$ for some $i \geq s$. We have

$$q^{(k)}(z) = \sum_{j=0}^{k} \binom{k}{j} \left(\frac{\partial^j}{\partial z^j} z^{i-s} \right) p^{(k-j)}(z) \ .$$

Hence, $q(r) = q'(r) = \ldots = q^{(m-1)}(r) = 0$ and $q^{(m)}(r) = r^{i-s}p^{(m)}(r) \neq 0$. Therefore,

$$q^{(k)}(r) = \sum_{j=0}^{s} a_j \, (i-j)(i-j-1)\ldots(i-j-k+1)r^{i-j-k} = 0$$

for $0 < k < m$. Hence

$$z^k q^{(k)}(r) = \sum_{j=0}^{s} a_j\,(i-j)(i-j-1)\ldots(i-j-k+1)r^{i-j} = 0$$

for $0 < k < m$. We have shown that $\{v_i^{[k]}\}_{i=0}^{\infty}$ with $v_i = i(i-1)\ldots(i-k+1)r^i$ and $0 < k < m$ are solutions for (13.6.14). Since

$$i\prod_{n=1}^{N}(i-a_n) = i^{N+1} + \underbrace{\left(-\sum_{n_1=1}^{N} a_{n_1}\right)}_{=A_1} i^N + \underbrace{\left((-1)^2 \sum_{\substack{n_1,n_2=1\\ n_1\neq n_2}}^{N} a_{n_1}a_{n_2}\right)}_{=A_2} i^{N-1} + \ldots + \underbrace{\left((-1)^N \prod_{n_1=1}^{N} a_{n_1}\right)}_{=A_N} i\ ,$$

we have that

$$i(i-1)\ldots(i-k+1) = i^k + \sum_{j=1}^{k-1} A_j i^{k-j}$$

for $0 < k < m$ and the appropriate definitions of the A_j; for instance, $A_1 = -\sum_{n=1}^{k-1} n$. Using the fact that a linear combination of solutions of (13.6.14) is a solution of (13.6.14), we have that $\{u_i^{[0]}\}_{i=0}^{\infty}$ with $u_i^{(0)} = r^i$, $\{u_i^{[1]}\}_{i=0}^{\infty}$ with $u_i^{[1]} = v_i^{[1]} = ir^i$, and in general $\{u_i^{[k]}\}_{i=0}^{\infty}$ with $1 < k < m$ and

$$u_i^{[k]} = v_i^{[k]} - \sum_{j=1}^{k-1} A_j u_i^{[j]} = i^k r^i$$

are linearly independent solutions of (13.6.14). ∎

Suppose that $z_1, z_2, \ldots, z_q$ are the roots of the characteristic polynomial of multiplicities $k_1, k_2, \ldots, k_q$ respectively. It follows from the previous proposition that the general solution $\{u_i\}_{i=0}^{\infty}$ of (13.6.14) is given by

$$u_i = \sum_{j=1}^{q}\left(\sum_{m=0}^{k_j-1} c_{j,m}\, i^m\right) z_j^i$$

for $i \geq 0$, where the constants $c_{j,m}$ are determined by the initial conditions $u_0, u_1, \ldots, u_{s-1}$. We note that $s = \sum_{j=1}^{q} k_j$.

Example 13.6.18
Consider the finite difference equation

$$u_{i+3} - 9\,u_{i+2} + 24\,u_{i+1} - 20\,u_i = 0$$

for $i \geq 0$ with initial conditions $u_0 = u_1 = 0$ and $u_2 = 1$. The characteristic polynomial is $z^3 - 9z^2 + 24z - 20 = (z-2)^2(z-5)$. There are two distinct roots: 2 of multiplicity 2 and 5 of multiplicity 1. The general solution is

$$u_i = (c_{1.0} + c_{1,1}\,i)2^i + c_{2,0}\,5^i$$

for $i \geq 1$. From the initial conditions, we get

$$\begin{aligned} u_0 = 0 &\Rightarrow c_{1,0} + c_{2,0} = 0 \\ u_1 = 0 &\Rightarrow 2(c_{1,0} + c_{1,1}) + 5c_{2,0} = 0 \\ u_2 = 1 &\Rightarrow (c_{1,0} + 2c_{1,1})\, 2^2 + c_{2,0}\, 5^2 = 1 \end{aligned}$$

Solving, we find $c_{1,0} = -1/9$, $c_{2,0} = 1/9$ and $c_{1,1} = -1/6$.

The solution $\{u_i\}_{i=0}^{\infty}$ is given by $u_i = -\left(\frac{1}{9} + \frac{1}{6}\,i\right)2^i + \frac{1}{9}\,5^i$ for $i = 0, 1, 2, \ldots$ ♣

Theorem 13.6.19

Suppose that (13.6.12) is of order s and let $\{u_i^{[k]}\}_{i=0}^{\infty}$ be the solution of the homogeneous finite difference equation

$$\begin{aligned} \sum_{j=0}^{s} a_j\, u_{i-j} &= 0 \quad , \quad i \geq s \\ u_i &= \delta_{k,i} \quad , \quad 0 \leq i < s \end{aligned}$$

for $0 \leq k < s$. Then, the solution $\{u_i\}_{i=0}^{\infty}$ of (13.6.12) is given by

$$u_i = \sum_{k=0}^{s-1} v_i\, u_i^{[k]} + w_i \quad , \quad i \geq 0 \ , \tag{13.6.16}$$

where

$$w_i = \begin{cases} \dfrac{1}{a_0} \displaystyle\sum_{k=0}^{i-s} C_{s+k}\, u_{i-k-1}^{[s-1]} & \text{if} \quad i \geq s \\ 0 & \text{if} \quad 0 \leq i < s \end{cases} \tag{13.6.17}$$

Proof.
It is easy to see that the first sum in (13.6.16) satisfies the homogeneous finite difference problem

$$\begin{aligned} \sum_{j=0}^{s} a_j\, u_{i-j} &= 0 \quad , \quad i \geq s \\ u_i &= v_i \quad , \quad 0 \leq i < s \end{aligned}$$

We now proof that (13.6.17) satisfies

$$\begin{aligned} \sum_{j=0}^{s} a_j\, w_{i-j} &= C_i \quad , \quad i \geq s \\ w_i &= 0 \quad , \quad 0 \leq i < s \end{aligned}$$

We obviously have $w_i = 0$ for $0 \leq i < s$ by definition of the w_i. We have that

$$\sum_{j=0}^{s} a_j\, w_{i-j} = \frac{1}{a_0} \sum_{j=0}^{s} a_j \left(\sum_{k=0}^{i-j-s} C_{s+k}\, u_{i-j-k-1}^{[s-1]} \right) = \frac{1}{a_0} \sum_{j=0}^{s} a_j \left(\sum_{k=0}^{i-s} C_{s+k}\, u_{i-j-k-1}^{[s-1]} \right)$$

because $i-j-k-1 < s-1$ for $k > i-j-s$ implies that $u^{[s-1]}_{i-j-k-1} = 0$. To simplify the notation, we assume that $u^{[s-1]}_i = 0$ for $i < 0$. Hence,

$$\sum_{j=0}^{s} a_j\, w_{i-j} = \frac{1}{a_0}\sum_{k=0}^{i-s} C_{s+k}\left(\sum_{j=0}^{s} a_j\, u^{[s-1]}_{i-j-k-1}\right) = \frac{1}{a_0} C_i\left(\sum_{j=0}^{s} a_j\, u^{[s-1]}_{s-j-1}\right) = C_i$$

for $i \geq s$. We note that $s-1 \leq i-k-1 \leq i-1$ for $0 \leq k \leq i-s$. Hence, the second equality comes from $\displaystyle\sum_{j=0}^{s} a_j\, u^{(s-1)}_{(i-k-1)-j} = 0$ for $i-k-1 \geq s$ because $\{u^{(s-1)}_i\}_{i=0}^{\infty}$ is a solution of the homogeneous difference equation. The last equality, for $i-k-1 = s-1$, comes from

$$u^{(s-1)}_{(i-k-1)-j} = u^{(s-1)}_{s-j-1} = \begin{cases} 1 & \text{if} \quad j = 0 \\ 0 & \text{if} \quad j > 0 \end{cases} \qquad \blacksquare$$

13.6.3 Convergence

Our study of the convergence of multistep methods will use the notion of "root condition" that we first define.

If $F \equiv 0$ in (13.6.1), we get $w_{i+1} = \displaystyle\sum_{k=0}^{m} a_k w_{i-k}$. If we substitute λ^i for w_i in this expression, we get $\lambda^{i+1} = \displaystyle\sum_{k=0}^{m} a_k \lambda^{i-k}$. If we multiply both sides of this equation by λ^{m-i}, we get $p(\lambda) = 0$ for $p(\lambda) = -\lambda^{m+1} + \displaystyle\sum_{k=0}^{m} a_k \lambda^{m-k}$.

Definition 13.6.20

The **characteristic polynomial** of the multistep method (13.6.1) is the polynomial $p(\lambda) = -\lambda^{m+1} + \displaystyle\sum_{k=0}^{m} a_k \lambda^{m-k}$.

Definition 13.6.21

1. A multistep method satisfies the **root condition** if all the roots of its characteristic polynomial have absolute values less than or equal to one and those equal to one are simple roots.

2. A multistep method is **strongly stable** if all the roots of its characteristic polynomial have absolute values less than one except for one root which is equal to one.

3. A multistep method is **weakly stable** if it satisfies the root condition and has more than one root of absolute value one.

4. A multistep method is **unstable** if it does not satisfy the root condition.

Example 13.6.22

1. The characteristic polynomial of Adams-Bashforth method of order four is $p(\lambda) = -\lambda^4 + \lambda^3$. 1 is a root of multiplicity one and 0 is a root of multiplicity three. The method is strongly stable.

2. The characteristic polynomial of the Adams-Bashforth method of order two from Example 13.6.7 is $p(\lambda) = -\lambda^2 + 1$. 1 and -1 are the two roots of this polynomial. The method is weakly stable.

♣

Proposition 13.6.23

If the finite difference method in (13.6.1) satisfies (13.6.2) and is convergent, then (13.6.1) satisfies the root condition.

Proof.
We give a special initial value problem with specific values for the δ_i in (13.6.5) such that the multistep method is not convergent if the root condition is not satisfied.

Consider the initial value problem

$$\begin{aligned} y'(t) &= 0 \quad , \quad t_0 \leq t \leq t_f \\ y(a) &= 0 \end{aligned} \tag{13.6.18}$$

Thus $F \equiv 0$ in (13.6.1). If we assume that $\delta_{i+1} = 0$ for $m \leq i < N$, the finite difference problem (13.6.5) becomes

$$\begin{aligned} u_{i+1} - \sum_{j=0}^{m} a_j\, u_{i-j} = 0 \quad &, \quad m \leq i < N \\ u_i = \delta_i \quad &, \quad 0 \leq i \leq m \end{aligned} \tag{13.6.19}$$

The solution of (13.6.18) is $y(t) = 0$ for all t. We show that the finite difference problem is not convergent for our initial value problem; namely, we do not have that

$$\lim_{h\to 0} \max_{0\leq i\leq N} |u_i| = 0 \ . \tag{13.6.20}$$

Suppose that c is a root of the characteristic polynomial $p(z)$ such that $|c| > 1$. Let

$$u_i = \begin{cases} hc^i & \text{if} \quad c \in \mathbb{R} \\ h\left(c^i + \bar{c}^i\right) & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

for $0 \leq i \leq N$, where $h = (t_f - t_0)/N$ as usual. We have that $\{u_i\}_{i=0}^N$ is a solution of (13.6.19) if we set $\delta_i = u_i$ for $0 \leq i \leq m$. For $c \in \mathbb{C} \setminus \mathbb{R}$, $\{u_i\}_{i=0}^N$ is linear combination of the two solutions, $\{c^i\}_{i=0}^N$ and $\{\bar{c}^i\}_{i=0}^N$. However,

$$|u_N| = \begin{cases} \dfrac{t_f - t_0}{N}|c^N| & \text{if} \quad c \in \mathbb{R} \\ \dfrac{t_f - t_0}{N}|c^N + \bar{c}^N| & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

does not converge to 0 as $N \to \infty$ (Remark 13.6.24 below). Thus (13.6.20) is not satisfied.

Suppose that c is a root of the characteristic polynomial such that $|c| = 1$ and c is not simple. Let

$$u_i = \begin{cases} hic^i & \text{if} \quad c \in \mathbb{R} \\ hi\left(c^i + \bar{c}^i\right) & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

for $0 \leq i \leq N$, where $h = (t_f - t_0)/N$. Again, $\{u_i\}_{i=0}^N$ is a solution of (13.6.19) if we set $\delta_i = u_i$ for $0 \leq i \leq m$. For $c \in \mathbb{C} \setminus \mathbb{R}$, $\{u_i\}_{i=0}^N$ is linear combination of the two solutions: $\{ic^i\}_{i=0}^N$ and $\{i\bar{c}^i\}_{i=0}^N$ (Proposition 13.6.17). However,

$$|u_N| = \begin{cases} (t_f - t_0)|c^N| & \text{if} \quad c \in \mathbb{R} \\ (t_f - t_0)|c^N + \bar{c}^N| & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

does not converge to 0 as $N \to \infty$ (Remark 13.6.24 below). Thus (13.6.20) is not satisfied. ∎

Remark 13.6.24

1. In the proof of the previous proposition, we could have used a fixed value of $t \in [t_0, t_f]$ associated to another index than N. Suppose that $t = t_{i(N)}$; namely, we have

   $$\frac{t - t_0}{i(N)} = \frac{t_f - t_0}{N} = h$$

   or, stated differently,

   $$i(N) = \frac{t - t_0}{t_f - t_0} N \ .$$

   We have that $i(N) = CN$ with $C = (t - t_0)/(t_f - t_0)$. If we select an increasing sequence $\{N_j\}_{j=0}^\infty$ of positive integers (e.g. $N_{j+1} = 2N_j$) such that CN_j reminds an integer, then $t = t_{i(N_j)}$ is always one of the nodes of the partition of $[t_0, t_f]$ and $i(N_j)$ increase proportionally to N_j according to $i(N_j) = CN_j$. It is then easy to modify the reasoning in the proof of the previous proposition to show that $u_{i(N_j)}$, the approximation of $y(t) = y(t_{i(N_j)})$, does not converge to 0 as $j \to \infty$.

2. In the proof of the previous proposition, we have used the following claims:

   (a) $\{|c^j|/j\}_{j=1}^\infty$ does not converge to 0 if $c \in \mathbb{R}$ satisfies $|c| > 1$.

   (b) $\{|c^j + \bar{c}^j|/j\}_{j=1}^\infty$ does not converge to 0 if $c \in \mathbb{C} \setminus \mathbb{R}$ satisfies $|c| > 1$.

   (c) $\{|c^j|\}_{j=1}^\infty$ does not converge to 0 if $c \in \mathbb{R}$ satisfies $|c| = 1$.

   (d) $\{|c^j + \bar{c}^j|\}_{j=1}^\infty$ does not converge to 0 if $c \in \mathbb{C} \setminus \mathbb{R}$ satisfies $|c| = 1$.

   The two cases where $c \in \mathbb{R}$ are easy to prove because $|c|^j/j \to \infty$ as $j \to \infty$ when $|c| > 1$, and $|c|^j = 1$ for all j when $|c| = 1$.

   If $c \in \mathbb{C} \setminus \mathbb{R}$, we have $c = |c|e^{i\theta}$ for some $\theta \neq n\pi$ for $n \in \mathbb{Z}$. Thus,

   $$c^j + \bar{c}^j = |c|^j e^{j\theta\, i} + |c|^j e^{-j\theta\, i} = 2|c|^j \cos(j\theta) \ .$$

We now show that there exists a strictly increasing sequence $\{j_k\}_{k=1}^{\infty}$ of positive integers and a constant $C > 0$ depending on θ such that $|\cos(j_k\theta)| \geq C$ for all k.

If $\theta = \dfrac{m\pi}{n}$ for two positive integer m and n such that m/n is in its reduced form and $n \neq 2$, then we can take $j_k = 2kn + 1$ and $C = |\cos(\theta)|$. We have

$$|\cos(j_k\theta)| = |\cos((2kn+1)\theta)| = |\cos(2km\pi + \theta)| = |\cos(\theta)| = C$$

for all k.

If $\theta = \dfrac{m\pi}{2}$ for $m = 1$ or 3, then we can take $j_k = 2k$ and $C = 1$. We have

$$|\cos(j_k\theta)| = |\cos(2k\theta)| = |\cos(km\pi)| = 1$$

for all k.

If $\theta/\pi \in \mathbb{R} \setminus \mathbb{Q}$ with $0 < \theta < 2\pi$, we need to use the fact that $\{e^{j\theta}\}_{j=0}^{\infty}$ is dense on the unit circle. Thus, there exist an infinite strictly increasing sequence $\{j_k\}_{k=0}^{\infty}$ such that $j_k\theta$ is between $\pi/6$ and $\pi/3$ modulo 2π. We then have

$$|\cos(j_k\theta)| \geq \left|\cos\left(\frac{\pi}{3}\right)\right| = \frac{1}{2}$$

for all k. We can take these j_k and $C = 1/2$.

Hence,

$$\frac{|c^{j_k} + \bar{c}^{j_k}|}{j_k} = \frac{2|c|^{j_k}|\cos(j_k\theta)|}{j_k} \geq 2C\,\frac{|c|^{j_k}}{j_k} \ .$$

Since $\displaystyle\lim_{k\to\infty} \frac{|c|^{j_k}}{j_k} = \infty$ because $|c| > 1$, we get that $\displaystyle\lim_{k\to\infty} \frac{|c_k^j + \bar{c}_k^j|}{j_k} = \infty$.

Similarly,

$$|c^{j_k} + \bar{c}^{j_k}| = 2|c|^{j_k}|\cos(j_k\theta)| \geq 2C$$

for $|c| = 1$ and all k. So $\{|c^{j_k} + \bar{c}^{j_k}|\}_{k=0}^{\infty}$ does not converge to 0.

♠

Proposition 13.6.25

If the finite difference method in (13.6.1) satisfies (13.6.2) and is zero-stable, then (13.6.1) satisfies the root condition.

Proof.

We proceed as in the proof of Proposition 13.6.23. We give a special initial value problem with specific values for the δ_i in (13.6.5) such that the multistep method is not zero-stable if the root condition is not satisfied.

Consider the initial value problem

$$y'(t) = 0 \quad , \quad t_0 \leq t \leq t_f$$

$$y(a) = 0$$

Thus $F \equiv 0$ in (13.6.1). If we assume that $\delta_{i+1} = 0$ for $m \le i < N$, the finite difference problem (13.6.5) becomes

$$\begin{aligned} u_{i+1} - \sum_{j=0}^{m} a_j\, u_{i-j} = 0 \quad &, \quad m \le i < N \\ u_i = \delta_i \quad &, \quad 0 \le i \le m \end{aligned} \tag{13.6.21}$$

Moreover, we consider the perturbed finite difference problem given by (13.6.5) with $\delta_i = 0$ for $0 \le i \le N$; namely,

$$\begin{aligned} \tilde{u}_{i+1} - \sum_{j=0}^{m} a_j\, \tilde{u}_{i-j} = 0 \quad &, \quad m \le i < N \\ \tilde{u}_i = \tilde{\delta}_i = 0 \quad &, \quad 0 \le i \le m \end{aligned}$$

The solution of this perturbed finite difference problem is obviously $\{\tilde{u}_i\}_{i=0}^{N}$, where $\tilde{u}_i = 0$ for $0 \le i \le N$.

We show that the multistep method is not zero-stable for our initial value problem; namely, given $\epsilon > 0$, there does not exist S and h_0 such that

$$\max_{0 \le i \le N} |u_i - \tilde{u}_i| = \max_{0 \le i \le N} |u_i| < S\epsilon \tag{13.6.22}$$

if $|\delta_i - \tilde{\delta}_i| = |\delta_i| < \epsilon$ for $0 \le i \le N$ and $h < h_0$.

Suppose that c is a root of the characteristic polynomial $p(z)$ such that $|c| > 1$. Let

$$u_i = \begin{cases} \delta\ c^i & \text{if} \quad c \in \mathbb{R} \\ \delta\left(c^i + \bar{c}^i\right) & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

for $0 \le i \le N$. We have that $\{u_i\}_{i=0}^{N}$ is a solution of (13.6.21) if we set $\delta_i = u_i$ for $0 \le i \le m$. We select δ small enough to get $|\delta_i - \tilde{\delta}_i| = |u_i| < \epsilon$ for $0 \le i \le m$. However,

$$|u_N| = \begin{cases} \delta\, |c|^N & \text{if} \quad c \in \mathbb{R} \\ \delta\, |c^N + \bar{c}^N| & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

does not converge to 0 as $N \to \infty$. There are strictly increasing sequences $\{N_j\}_{j=0}^{\infty}$ of positive integers such that $\{|u_{N_j}|\}_{j=0}^{\infty}$ converges to ∞ (Remark 13.6.24). Thus, we can take N_j large enough such that $h = (t_f - t_0)/N_j < h_0$ and $|u_{N_j}| > S\epsilon$ for whatever S and h_0 that we choose. Therefore, contradicting (13.6.22).

Suppose that c is a root of the characteristic polynomial $p(z)$ such that $|c| = 1$ and c is not simple. Let

$$u_i = \begin{cases} \delta\, i c^i & \text{if} \quad c \in \mathbb{R} \\ \delta\, i\left(c^i + \bar{c}^i\right) & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

for $0 \leq i \leq N$. Again, $\{u_i\}_{i=0}^N$ is a solution of (13.6.19) if we set $\delta_i = u_i$ for $0 \leq i \leq m$. We also can select δ small enough to get $|\delta_i - \tilde{\delta}_i| = |u_i| < \epsilon$ for $0 \leq i \leq m$. However,

$$|u_N| = \begin{cases} \delta\, N|c|^N & \text{if} \quad c \in \mathbb{R} \\ \delta\, N|c^N + \bar{c}^N| & \text{if} \quad c \in \mathbb{C} \setminus \mathbb{R} \end{cases}$$

does not converge to 0 as $N \to \infty$. There are strictly increasing sequences $\{N_j\}_{j=0}^\infty$ of positive integers such that $\{|u_{N_j}|\}_{j=0}^\infty$ converges to ∞ (Remark 13.6.24). Again, we can take N_j large enough such that $h = (t_f - t_0)/N_j < h_0$ and $|u_{N_j}| > S\epsilon$ for whatever S and h_0 that we choose. Therefore, contradicting (13.6.22). ∎

Theorem 13.6.26 (Dahlquist)

Suppose that the finite difference method in (13.6.1) satisfies (13.6.2) and (13.6.3), and that $\max_{0\leq i\leq N} |\delta_i(h)| \leq \delta(h) = O(h^2)$ in (13.6.5). If the finite difference problem (13.6.1) is consistent, then (13.6.1) is convergent if and only if it satisfies the root condition.

Remark 13.6.27

It is not too outrageous to require $\delta(h) = O(h^2)$ in the statement of Theorem 13.6.26.

Suppose that $\{u_i\}_{i=0}^N$ is a solution of

$$\frac{1}{h}\left(u_{i+1} - \sum_{j=0}^m a_j u_{i-j}\right) + F(h, \mathbf{t}, \mathbf{u}, f) + \sigma_{i+1}(h) \quad , \quad m \leq i < N \ ,$$

where $\sigma_{i+1}(h)$ represents the perturbation. Then,

$$u_{i+1} - \sum_{j=0}^m a_j u_{i-j} + hF(h, \mathbf{t}, \mathbf{w}, f) + h\sigma_{i+1}(h) \quad , \quad m \leq i < N \ .$$

We may set $\delta_i(h) = h\sigma_i(h)$. So, the real assumption that we make is that $\max_{0\leq i\leq N} |\sigma_i(h)| \leq \sigma(h) = O(h)$ near the origin. ♠

Proof (of Dahlquist's theorem).

That convergence implies that the root condition is satisfied is a consequence of Proposition 13.6.23. We need only prove the converse. Suppose that the root condition is satisfied.

As mentioned in the previous remark, we may assume that $\delta_i(h) = h\sigma_i(h)$ and $\delta(h) = h\sigma(h)$, where both $\sigma_i(h)$ and $\sigma(h)$ are $O(h)$ near the origin.

If we subtract

$$y(t_{i+1}) - \sum_{j=0}^m a_j y(t_{i-j}) - hF(h, \mathbf{t}, \mathbf{y}, f) = h\tau_{i+1}(h)$$

from

$$u_{i+1} - \sum_{j=0}^m a_j u_{i-j} - hF(h, \mathbf{t}, \mathbf{u}, f) = h\sigma_{i+1}(h)$$

for $m \leq i < N$, we get

$$\sum_{j=-1}^{m} a_j\, e_{i-j} = C_i \quad , \quad m \leq i < N \ , \tag{13.6.23}$$

where $a_{-1} = -1$, $e_j = u_j - y(t_j)$ for $0 \leq j \leq N$ and

$$C_i = -h\left(F(h,\mathbf{t},\mathbf{u},f) - F(h,\mathbf{t},\mathbf{y},f)\right) - h(\sigma_{i+1}(h) - \tau_{i+1}(h)) \quad , \quad m \leq i < N \ .$$

If we replace j by $j-1$ and i by $i-1$ in (13.6.23), we get the finite difference equation

$$\begin{aligned} \sum_{j=0}^{m+1} a_{j-1}\, e_{i-j} &= C_{i-1} \quad \text{for} \quad m < i \leq N \\ e_i &= h\sigma_i(h) \quad \text{for} \quad 0 \leq i \leq m \end{aligned} \tag{13.6.24}$$

Let $\left\{u_i^{[k]}\right\}_{i=0}^{\infty}$ be the solution of

$$\begin{aligned} \sum_{j=0}^{m+1} a_{j-1}\, u_{i-j} &= 0 \quad , \quad i > m \\ u_i &= \delta_{k,i} \quad , \quad 0 \leq i \leq m \end{aligned}$$

for $0 \leq k \leq m$. From Theorem 13.6.19 (with $v_i = e_i$, a_j replaced by a_{j-1}, C_i replaced by C_{i-1} and $s = m+1$ in (13.6.12)), the solution of (13.6.24) is

$$\begin{aligned} e_i &= \sum_{k=0}^{m} e_i\, u_i^{[k]} + \begin{cases} \dfrac{1}{a_{-1}} \displaystyle\sum_{k=0}^{i-m-1} C_{k+m}\, u_{i-k-1}^{[m]} & \text{if} \quad m < i \leq N \\ 0 & \text{if} \quad 0 \leq i \leq m \end{cases} \\ &= \sum_{k=0}^{m} e_i\, u_i^{[k]} - \begin{cases} \displaystyle\sum_{k=0}^{i-m-1} C_{k+m}\, u_{i-k-1}^{[m]} & \text{if} \quad m < i \leq N \\ 0 & \text{if} \quad 0 \leq i \leq m \end{cases} \end{aligned} \tag{13.6.25}$$

The root condition implies that there exist a constant $Q \geq 1$ such that $|u_i^{[k]}| \leq Q$ for all i and k. Recall that all solutions of the homogeneous difference equation

$$\sum_{j=0}^{m+1} a_{j-1}\, u_{i-j} = 0 \quad , \quad i > m \ , \tag{13.6.26}$$

are linear combinations of solutions with terms of the form $e_i = i^n c^i$, where c is a root of the characteristic polynomial

$$\sum_{j=0}^{m+1} a_{j-1}\, z^{m+1-j} = \sum_{j=-1}^{m} a_j\, z^{m-j} = 0$$

and n is a non-negative integer smaller than the multiplicity of c.

It follows from the definition of C_i and (13.6.3) that

$$|C_{k+m}| \leq h\left(R \sum_{j=k}^{k+m+1} |e_j| + \sigma(h) + \tau(h)\right) \leq h\left(R\,(m+2) \max_{k \leq j \leq k+m+1} |e_j| + \sigma(h) + \tau(h)\right)$$

$$\leq h\left(R\,(m+2)\max_{0\leq j\leq i}|e_j| + \sigma(h) + \tau(h)\right)$$

for $0 \leq k \leq i-m-1$. Hence, from (13.6.25), we get

$$|e_i| \leq \begin{cases} Q(m+1)\displaystyle\max_{0\leq j\leq m}|e_j| \\ \qquad +Q(i-m+1)\,h\left(R\,(m+2)\displaystyle\max_{0\leq j\leq i}|e_j| + \sigma(h)+\tau(h)\right) & \text{if} \quad m < i \leq N \\ Q(m+1)\displaystyle\max_{0\leq j\leq m}|e_j| & \text{if} \quad 0 \leq i \leq m \end{cases}$$

$$\leq \begin{cases} Q(m+1)\displaystyle\max_{0\leq j\leq m}|e_j| \\ \qquad +Qih\left(R\,(m+2)\displaystyle\max_{0\leq j\leq i}|e_j| + \sigma(h)+\tau(h)\right) & \text{if} \quad m < i \leq N \\ Q(m+1)\displaystyle\max_{0\leq j\leq m}|e_j| & \text{if} \quad 0 \leq i \leq m \end{cases} \tag{13.6.27}$$

We get from (13.6.27) that

$$\max_{0\leq j\leq i}|e_j| \leq Q(m+1)\max_{0\leq j\leq m}|e_j| + Mih\max_{0\leq j\leq i}|e_j| + Qih\,(\sigma(h)+\tau(h)) \quad , \quad m < i \leq N\,, \tag{13.6.28}$$

where $M = RQ(m+2)$.

We choose h small enough (i.e. N large enough) such that $1/(2Mh) > m+1$ and consider $m < i \leq 1/(2Mh)$. We get from (13.6.28) that

$$\max_{0\leq j\leq i}|e_j| \leq Q(m+1)\max_{0\leq j\leq m}|e_j| + \frac{1}{2}\max_{0\leq j\leq i}|e_j| + Qih\,(\sigma(h)+\tau(h))$$

for $m < i \leq 1/(2Mh)$. If we isolate $\displaystyle\max_{0\leq j\leq i}|e_j|$, we get

$$\max_{0\leq j\leq i}|e_j| \leq 2\,Q\left((m+1)\max_{0\leq j\leq m}|e_j| + \frac{1}{2M}\,(\sigma(h)+\tau(h))\right) \tag{13.6.29}$$

for $m < i \leq 1/(2Mh)$.

Let $i_k = \lfloor k/(2Mh)\rfloor$ for $1 \leq k \leq K$, where $K = \lfloor 2M(t_f - t_0)\rfloor$. Recall that $\lfloor a\rfloor$ is the largest integer smaller than or equal to a. Let $i_{-1} = 0$, $i_0 = m$ and $i_{K+1} = N$.

If we repeat the same argument on the interval $I_k = [t_0 + k/(2M), t_0 + (k+1)/(2M)]$ for $1 \leq k \leq K$ with the initial conditions at t_j given by e_j for $i_k - m \leq j \leq i_k$, we get

$$\max_{i_k-m\leq j\leq i}|e_j| \leq 2Q\left((m+1)\max_{i_1-m\leq j\leq i_1}|e_j| + \frac{1}{2M}\,(\sigma(h)+\tau(h))\right) \tag{13.6.30}$$

for $i_k < i \leq i_{k+1}$ and $1 \leq k \leq K$.

Let $E_k = \displaystyle\max_{i_k\leq i\leq i_{k+1}}|e_i|$ for $-1 \leq k \leq K$. We deduce from (13.6.29) that

$$E_0 \leq 2\,Q\left((m+1)\,E_{-1} + \frac{1}{2M}\,(\sigma(h)+\tau(h))\right)$$

and from (13.6.30) that

$$E_k \le 2Q\left((m+1)E_{k-1} + \frac{1}{2M}\left(\sigma(h)+\tau(h)\right)\right)$$

for $1 \le k \le K$. By induction, we find

$$\begin{aligned} E_k &\le \left(2Q(m+1)\right)^{k+1} E_{-1} + \left(\frac{1-(2Q(m+1))^{k+1}}{1-2Q(m+1)}\right)\frac{Q}{M}\left(\sigma(h)+\tau(h)\right) \\ &\le \left(2Q(m+1)\right)^{K+1} E_{-1} + \left(\frac{1-(2Q(m+1))^{K+1}}{1-2Q(m+1)}\right)\frac{Q}{M}\left(\sigma(h)+\tau(h)\right) \end{aligned}$$

for $0 \le k \le K$ because $Q(m+1) \ge 1$ by assumption. Therefore,

$$\max_{0\le i\le N} |u_i - y(y_i)| = \max_{-1\le k\le K} E_k \to 0$$

as $h \to 0$ independently of f. Recall that $E_{-1} \to 0$ as $h \to 0$ because $E_{-1} = \max_{0\le j\le m} |e_j| = \max_{0\le j\le m} |\delta_j(h)| = O(h^2)$. ∎

Remark 13.6.28
If we use the definition of convergence given in remark 13.6.3 which is equivalent to assuming that $\delta_i(h) \equiv 0$ for all i in (13.6.5), the previous theorem can be stated as follows.

Suppose that the finite difference method in (13.6.1) satisfies (13.6.2) and (13.6.3). If the finite difference problem (13.6.1) is consistent, then (13.6.1) is convergent if and only if it satisfies the root condition.

There is no reference to perturbations δ_i in this statement. ♠

Theorem 13.6.29

Suppose that the finite difference method in (13.6.1) satisfies (13.6.2) and (13.6.3), and that $\max_{0\le i\le N} |\delta_i(h)| \le \delta(h) = O(h^2)$ in (13.6.5). Then (13.6.1) is zero-stable if and only if it satisfies the root condition.

Proof.
We have from Proposition 13.6.25 that zero-stable implies root condition. We only need to prove the converse.

The proof is similar to the proof of the previous theorem. If we subtract

$$u_{i+1}^{[1]} - \sum_{j=0}^{m} a_j u_{i-j}^{[1]} - hF(h, \mathbf{t}, \mathbf{u}^{[1]}, f) = \delta_{i+1}^{[1]}(h)$$

from

$$u_{i+1}^{[2]} - \sum_{j=0}^{m} a_j u_{i-j}^{[2]} - hF(h, \mathbf{t}, \mathbf{u}^{[2]}, f) = \delta_{i+1}^{[2]}(h)$$

for $m \leq i < N$, we get
$$\sum_{j=-1}^{m} a_j\, e_{i-j} = C_i \quad , \quad m \leq i < N \ ,$$
where $a_{-1} = -1$, $e_j = u_j^{[2]} - u_j^{[1]}$ for $0 \leq j \leq N$ and
$$C_i = -h\left(F(h,\mathbf{t},\mathbf{u}^{[2]},f) - F(h,\mathbf{t},\mathbf{u}^{[1]},f)\right) - \left(\delta_{i+1}^{[2]} - \delta_{i+1}^{[1]}\right) \quad , \quad m \leq i < N \ .$$

Let $\delta_i^{[j]}(h) = h\sigma_i^{[j]}(h)$ for $j = 1$ and 2. Moreover, let
$$\sigma(h) = \max_{0\leq i\leq N} \left|\sigma_i^{[2]}(h) - \sigma_i^{[1]}(h)\right| \ .$$
We then have that[5]
$$\left|\delta_i^{[2]}(h) - \delta_i^{[1]}(h)\right| \leq h\sigma(h) \quad , \quad 0 \leq i \leq N \ .$$
Proceeding as we did in the proof of Dahlquist's theorem, we show that
$$E_k \leq (2Q(m+1))^{K+1}\, E_{-1} + \left(\frac{1-(2Q(m+1))^{K+1}}{1-2Q(m+1)}\right)\frac{Q}{M}\,\sigma(h)$$
for $0 \leq k \leq K$, where
$$E_{-1} = \max_{0\leq i\leq m} |e_i| = \max_{0\leq i\leq m} \left|u_i^{[2]} - u_i^{[1]}\right| = \max_{0\leq i\leq m} \left|h\sigma_i^{[2]} - h\sigma_i^{[1[}\right| \leq h\sigma(h)$$
because $u_i^{[1]} = y(t_i) + \delta_i^{[1]}(h)$ and $u_i^{[2]} = y(t_i) + \delta_i^{[2]}(h)$ for $0 \leq i \leq m$. Thus
$$E_k \leq \left((2Q(m+1))^{K+1}h + \left(\frac{1-(2Q(m+1))^{K+1}}{1-2Q(m+1)}\right)\frac{Q}{M}\right)\sigma(h)$$
for $-1 \leq k \leq M$.

Let
$$K = (2Q(m+1))^{K+1} + \left(\frac{1-(2Q(m+1))^{K+1}}{1-2Q(m+1)}\right)\frac{Q}{M} \ .$$
Given $\epsilon > 0$, choose $h_0 < 1$ such that $\sigma(h) < \epsilon$ for $|h| < h_0$. This is possible because $\sigma(h) \to 0$ as $h \to 0$. Then
$$\max_{0\leq i\leq N} |u_i^{[1]} - u_i^{[2]}| = \max_{-1\leq k\leq K} E_k < K\epsilon$$
namely, the definition of zero-stability is satisfied with these values of K and h_0. ∎

We end this section by stating (without proofs) a couple of results providing some constraints on the maximal order of some numerical methods if stability and convergence have to be preserved.

[5]Instead of requiring $\max_{0\leq i\leq N} |\delta_i(h)| \leq \delta(h) = O(h^2)$ in the statement of the theorem, we could have only required $\max_{0\leq i\leq N} \left|\delta_i^{[2]}(h) - \delta_i^{[1]}(h)\right| \leq \delta(h) = O(h^2)$.

Theorem 13.6.30 (Dahlquist First Barrier)

The maximum order of a zero-stable implicit multistep method of the form (13.5.1) is $m+3$ when m is even and $m+1$ when m is odd. For a zero-stable explicit multistep method of the form (13.5.1), the maximum order is $m+1$.

Proposition 13.6.31

The Backward Difference Formulae satisfy the root condition if and only if $0 \leq m \leq 5$ (i.e. they are method of order 1 to 6 inclusively). Since these methods are consistent by construction, they are convergent if and only if $0 \leq m \leq 5$.

13.6.4 Absolute Stability and A-Stability

Stability is a delicate concepts. There are several ways to define it. The goal is always to ensure (as much as possible) that errors do not increase as we iterate; namely, that $|u_i - y(t_i)|$ does not increase as i increases, where u_i is the numerical approximation of w_i.

To define the new notion of stability, we consider the simple linear initial value problem

$$\begin{aligned} y'(t) &= \mu\, y(t) \quad , \quad t_0 \leq t \leq t_f \\ y(t_0) &= y_0 \end{aligned} \tag{13.6.31}$$

where $\operatorname{Re}\mu < 0$.

We will start with Runge-Kutta methods before turning our attention to multistep methods (with $m > 0$).

13.6.4.1 Runge-Kutta Methods

If we apply the general Runge-Kutta method given in Definition 13.4.1 to (13.6.31), we get

$$\begin{aligned} w_{i+1} &= w_i + h \sum_{j=1}^{s} \gamma_j K_j \\ w_0 &= y_0 \end{aligned}$$

for $0 \leq i < N$, where

$$K_j = \mu \left(w_i + h \sum_{m=1}^{s} \beta_{j,m} K_m \right)$$

for $1 \leq j \leq s$.

We can rewrite these two formulae in a more compact way using vectors and matrices. Let

$$B = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,s} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{s,1} & \beta_{s,2} & \dots & \beta_{s,s} \end{pmatrix} \ , \quad \mathbf{c} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_s \end{pmatrix} \ , \quad \mathbf{K} = \begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_s \end{pmatrix} \ , \quad \mathbf{u} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{w}_i = w_i \mathbf{u} \ .$$

We can rewrite the formulae above as

$$\begin{aligned} w_{i+1} &= w_i + h\mathbf{c}^\top \mathbf{K} \\ w_0 &= y_0 \end{aligned}$$

for $0 \leq i < N$, where

$$\mathbf{K} = \mu\left(\mathbf{w}_i + hB\mathbf{K}\right) .$$

If we solve this last equation for $\mathbf{K}$, we get

$$\mathbf{K} = \mu\left(\mathrm{Id} - h\mu B\right)^{-1}\mathbf{w}_i$$

Thus

$$w_{i+1} = w_i + h\mu\mathbf{c}^\top(\mathrm{Id} - h\mu B)^{-1}\mathbf{w}_i = w_i\left(1 + h\mu\mathbf{c}^\top(\mathrm{Id} - h\mu B)^{-1}\mathbf{u}\right) \tag{13.6.32}$$

for $0 \leq i < N$.

Definition 13.6.32

The **region of absolute stability** of a Runge-Kutta method is the set of all values $h\mu \in \mathbb{C}$ such that $\lim_{i\to+\infty} w_i = 0$ for all solutions $\{w_i\}_{i=0}^{\infty}$ of the difference equation associated to the Runge-Kutta method given in Definition 13.4.1 applied to the initial value problem (13.6.31).
A Runge-Kutta method is **A-stable** if the region of absolute stability contains the half-plane to the left of the imaginary axis (complex numbers with a negative real part.)

Remark 13.6.33
In the previous definition, we have to remember that we assume that $\mathrm{Re}\,\mu < 0$. Hence, all solutions $y(t) = e^{\mu(t-t_0)}y_0$ of the differential equation (13.6.31) satisfy $\lim_{t\to\infty} y(t) = 0$. ♠

Example 13.6.34
We find the region of absolute stability for the Runge-Kutta method of order two given in Definition 13.4.3. The computations for the other explicit Runge-Kutta methods are similar but more convoluted.

The recursive formula for the Runge-Kutta method of order two is

$$w_{i+1} = w_i + h\left(\gamma_1 f(t_i, w_i) + \gamma_2 f(t_i + \alpha_2 h, w_i + \beta_{2,1} h f(t_i, w_i))\right) .$$

If we use the Runge-Kutta method of order two to solve the non-trivial initial value problem (13.6.31), the iterative formula becomes

$$\begin{aligned} w_{i+1} &= w_i + h\left(\gamma_1 \mu\, w_i + \gamma_2 \mu\left(w_i + \beta_{2,1} h\mu\, w_i\right)\right) \\ &= \left(1 + (\gamma_1 + \gamma_2)\mu\, h + \gamma_2\beta_{2,1}(\mu\, h)^2\right) w_i . \end{aligned} \tag{13.6.33}$$

If we substitute λ^i for w_i, we get

$$\lambda^{i+1} = \left(1 + (\gamma_1 + \gamma_2)\mu\, h + \gamma_2\beta_{2,1}(\mu\, h)^2\right)\lambda^i$$

and, after dividing by λ^i, we get the only non-null root

$$\lambda = \left(1 + (\gamma_1 + \gamma_2)\mu h + \gamma_2\beta_{2,1}(\mu h)^2\right) .$$

Hence, the general solution of (13.6.33) is $w_i = \lambda^i w_0$ for $i = 0, 1, 2, \ldots$ From the condition $\gamma_1 + \gamma_2 = 1$ and $\beta_{2,1}\gamma_2 = 1/2$ (Definition 13.4.3), we get

$$\lambda = \left(1 + \mu h + \frac{1}{2}(\mu h)^2\right) . \tag{13.6.34}$$

We need

$$|\lambda| = \left|1 + \mu h + \frac{1}{2}(\mu h)^2\right| < 1 \tag{13.6.35}$$

to get $\lim_{i\to\infty} w_i = 0$.

The region of absolute stability of the Runge-Kutta methods of order two is the set of all $h\mu \in \mathbb{C}$ such that (13.6.35) is satisfied. The set of values $z \in \mathbb{C}$ such that $1 + z + z^2/2$ is on the unit circle is the black curve shown in Figure 13.5. To draw this black curve, we may use the fact that $1 + z + z^2/2 = e^{i\theta}$ is a quadratic equation whose solutions are given by $z = -1 \pm \sqrt{-1 + 2e^{i\theta}}$. The number i in the previous sentence is the complex number such that $i^2 = -1$ and not the index i in the Runge-Kutta method. As θ goes from 0 to 2π, we move along the (upper and lower branches of the) black curve.

The region of absolute stability is inside the black curve. To determine if the region of absolute stability is inside or outside the continuous curve, we have drawn the curve generated by the set of points z such that $|1 + z + z^2/2| = 1.2$, the red curve in Figure 13.5, and the curve generated by the set of points z such that $|1 + z + z^2/2| = 0.8$, the blue curve in Figure 13.5. In the first case, the points z correspond to values of $h\mu$ for which $|\lambda| > 1$, so they are outside the region of absolute stability, while in the second case they correspond to values of $h\mu$ for which $|\lambda| < 1$, so they are inside the region of absolute stability.

One can show that all the Runge-Kutta methods of order p fixed have the same region of absolute stability. It is certainly true for $p = 2$ as we have just shown. ♣

Proposition 13.6.35

Consider the general Runge-Kutta method from Definition 13.4.1. There exists a rational function $r : \mathbb{C} \to \mathbb{C}$ such that $w_i = (r(h\mu))^i w_0$ for $0 \leq i \leq N$. If the Runge-Kutta method is explicit, r is a polynomial.

Proof.

We get by induction from (13.6.32) that

$$w_i = w_0 \left(1 + h\mu \mathbf{c}^\top (\mathrm{Id} - h\mu B)^{-1}\mathbf{u}\right)^i$$

for $0 \leq i \leq N$. Thus, we have to show that

$$r(z) = 1 + z\mathbf{c}^\top (\mathrm{Id} - zB)^{-1}\mathbf{u}$$

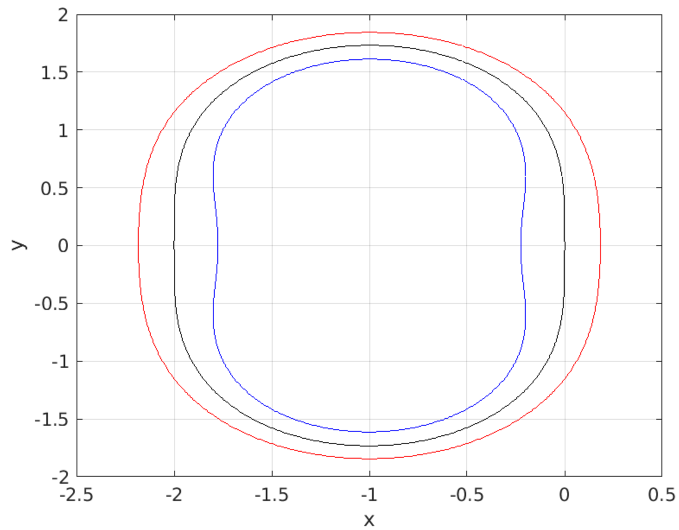


Figure 13.5: Boundaries of the region of absolute stability for the Runge-Kutta method of order two (black curve) and the curve generated by the set of points $z \in \mathbb{C}$ such that $|1 + z + z^2/2| = 1.2$ and 0.8 (red and blue curves respectively). The region of absolute stability is inside the black curve.

is a rational function. It is enough to show that $\mathbf{c}^\top(\mathrm{Id} - zB)^{-1}\mathbf{u}$ is a rational function in z. This comes from

$$(\mathrm{Id} - zB)^{-1} = \frac{1}{\det(\mathrm{Id} - zB)}\left(\mathrm{adj}(\mathrm{Id} - zB)\right)^\top ,$$

where $\det(\mathrm{Id} - zB)$ is a polynomial in z of degree at most s, and $\mathrm{adj}(\mathrm{Id} - zB)$ is an $s \times s$ matrix whose (i,j) entry is the **cofactor** $(-1)^{i+j}\det A_{i,j}$, where $A_{i,j}$ is obtained from $\mathrm{Id} - zB$ by removing the i^{th} row and j^{th} column. The cofactors are polynomials in z of degree at most $s-1$. Hence, $\mathbf{c}^\top(\mathrm{Id} - zA)^{-1}\mathbf{u}$ is the quotient of a polynomial of degree at most $s-1$ by a polynomial of degree at most s.

If the Runge-Kutta method is explicit, $\mathrm{Id} - zB$ is a lower-triangular matrix with only 1 on the diagonal. Thus $\det(\mathrm{Id} - zB) = 1$. ∎

Corollary 13.6.36

The stability domain of a general Runge-Kutta method is $\{z \in \mathbb{C} : |r(z)| < 1\}$.

Proof.
The result follows from

$$\lim_{i\to+\infty} w_i = 0 \Leftrightarrow \lim_{i\to+\infty}\left(r(h\mu)\right)^i = 0 \Leftrightarrow |r(h\mu)| < 1 \ .$$ ∎

Remark 13.6.37
To motivate the definition of stability, suppose that u_i is the numerical approximation of w_i. Let $r(h\mu) = 1 + h\mu\mathbf{c}^\top(\mathrm{Id} - h\mu B)^{-1}\mathbf{u}$. We have from (13.6.32) that

$$w_{i+1} = r(h\mu)w_i \quad , \quad 0 \leq i < N \ .$$

We have by definition of the local truncation error that

$$y_{i+1} - w_{i+1} = r(h\mu)(y_i - w_i) + h\tau_{i+1}(h) \quad , \quad 0 \leq i < N \ , \tag{13.6.36}$$

where $y_i = y(t_i)$ for $0 \leq i \leq N$.

Moreover, we may assume that u_i is the exact solution of

$$u_{i+1} = r(h\mu)u_i + \delta_i \quad , \quad 0 \leq i < N \ ,$$

where δ_i represents the error for each computation. Hence,

$$u_{i+1} - w_{i+1} = r(h\mu)(u_i - w_i) + \delta_i \quad , \quad 0 \leq i < N \ . \tag{13.6.37}$$

If we subtract (13.6.37) from (13.6.36), we get

$$(y_{i+1} - u_{i+1}) = r(h\mu)(y_i - u_i) + h\tau_{i+1}(h) - \delta_i \quad , \quad 0 \leq i < N \ . \tag{13.6.38}$$

We now prove by induction that

$$|y_i - u_i| \leq |r(h\mu)|^i \, |y_0 - u_0| + \sum_{j=0}^{i-1} |r(h\mu)|^j \big(|h\tau_{i-j}(h)| + |\delta_{i-1-j}| \big) \quad , \quad 0 < i \leq N \ . \tag{13.6.39}$$

It follows from (13.6.38) with $i = 0$ that (13.6.39) is true for $i = 1$. Suppose that (13.6.39) is true. then (13.6.38) and the induction hypothesis imply that

$$\begin{aligned}
|y_{i+1} - u_{i+1}| &\leq |r(h\mu)| \, |y_i - u_i| + |h\tau_{i+1}(h)| + |\delta_i| \\
&\leq |r(h\mu)| \left(|r(h\,u)|^i |y_0 - u_0| + \sum_{j=0}^{i-1} |r(h\mu)|^j \big(|h\tau_{i-j}(h)| + |\delta_{i-1-j}| \big) \right) + |h\tau_{i+1}(h)| + |\delta_i| \\
&= |r(h\,u)|^{i+1} |y_0 - u_0| + \sum_{j=0}^{i-1} |r(h\mu)|^{j+1} \big(|h\tau_{i-j}(h)| + |\delta_{i-1-j}| \big) + |h\tau_{i+1}(h)| + |\delta_i| \\
&= |r(h\,u)|^{i+1} |y_0 - u_0| + \sum_{j=1}^{i} |r(h\mu)|^j \big(|h\tau_{i+1-j}(h)| + |\delta_{i-j}| \big) + |h\tau_{i+1}(h)| + |\delta_i| \\
&= |r(h\,u)|^{i+1} |y_0 - u_0| + \sum_{j=0}^{i} |r(h\mu)|^j \big(|h\tau_{i+1-j}(h)| + |\delta_{i-j}| \big)
\end{aligned}$$

This is (13.6.39) with i replaced by $i + 1$. Thus completing the proof by induction.

Suppose that the Runge-Kutta method is consistent; namely, $\max\limits_{0 \leq i < N} |\tau_{i+1}(h)| \leq \tau(h) \to 0$ as $h \to 0$. Moreover, suppose that $|\delta_i| < \delta$ for all i.

If $h\mu$ is in the region of absolute stability, then $|r(h\mu)| < 1$. Hence, (13.6.39) yields

$$
\begin{aligned}
|y_i - u_i| &\leq |r(h\,u)|^i\,|y_0 - u_0| + \sum_{j=0}^{i-1} |r(h\mu)|^j \big(h\tau(h) + \delta\big) \\
&= |r(h\,u)|^i\,|u_0 - w_0| + \frac{1 - |r(h\mu)|^i}{1 - |r(h\mu)|}\big(h\tau)h) + \delta\big) \\
&\leq |u_0 - w_0| + \frac{h\tau(h)}{1 - |r(h\mu)|} + \frac{\delta}{1 - |r(h\mu)|} \quad , \quad 1 < i \leq N \ .
\end{aligned}
$$

Thus, the error of the approximation u_i does not increase as i increases.

If we consider the previous example for the Runge-Kutta method of order two, we have $r(h\mu) = 1 + h\mu + (h\mu)^2/2$. Then

$$
\begin{aligned}
1 - |r(h\mu)| &= 1 - \left(1 - r(h\mu)\,\overline{r(h\mu)}\right)^{1/2} = 1 - \left(1 - (\mu + \overline{\mu})h + O(h^2)\right)^{1/2} \\
&= 1 - \left(1 - \frac{(\mu + \overline{\mu})h}{2} + O(h^2)\right) = \frac{(\mu + \overline{\mu})h}{2} + O(h^2) \ .
\end{aligned}
$$

Thus,

$$
\frac{h\tau(h)}{1 - |r(h\mu)|} = \frac{\tau(h)}{(\mu + \overline{\mu})/2 + O(h)} \to 0
$$

as $h \to 0$. However, since $r(h\mu) \to 1$ as $h \to 0$, $\delta/(1 - |r(h\mu)|)$ increases as $h \to 0$. So, the numerical approximations u_i may not get better as $h \to 0$. As for the Euler's method (see the remark after Theorem 13.2.5), we may suspect that there is an optimal value of h to reduce round off errors. This will require a thorough analysis of $\dfrac{h\tau(h)}{1 - |r(h\mu)|} + \dfrac{\delta}{1 - |r(h\mu)|}$. ♠

Corollary 13.6.38

No explicit Runge-Kutta method is A-stable

Proof.
For an explicit Runge-Kutta method, the function r in Proposition 13.6.35 is a polynomial. There is no polynomial r of degree greater than zero that is bounded on $\{z : \mathrm{Re}\, z < 0\}$. If r is constant, then $r(z) = 1$ for all z because $r(0) = 1$. So, there is no polynomial r such that $|r(z)| < 1$ for all z in $\{z : \mathrm{Re}\, z < 0\}$. ∎

Example 13.6.39
Consider the Runge-Kutta method given by the Butcher array

$$
\begin{array}{c|cc}
0 & 1/4 & -1/4 \\
2/3 & 1/4 & 5/12 \\
\hline
 & 1/4 & 3/4
\end{array}
$$

So $B = \begin{pmatrix} 1/4 & -1/4 \\ 1/4 & 5/12 \end{pmatrix}$ and $\mathbf{c} = \begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix}$. Thus, $\mathrm{Id} - zB = \begin{pmatrix} 1 - z/4 & z/4 \\ -z/4 & 1 - 5z/12 \end{pmatrix}$ and

$$\begin{aligned} r(z) &= 1 + z\mathbf{c}^\top (\mathrm{Id} - zB)^{-1}\mathbf{u} = 1 + \frac{z}{(1-z/4)(1-5z/12) + z^2/16}\,\mathbf{c}^\top \begin{pmatrix} 1 - 5z/12 & -z/4 \\ z/4 & 1 - z/4 \end{pmatrix}\mathbf{u} \\ &= \frac{1 + z/3}{1 - 2z/3 + z^2/6} \ . \end{aligned}$$

To show that this method is A-stable, we show that $|r(z)| < 1$ for $z = \rho e^{i\theta}$ with $\rho > 0$ and $\pi/2 < \theta < 3\pi/2$. We have

$$\begin{aligned} |r(z)| < 1 &\Leftrightarrow \left|1 + \frac{\rho e^{i\theta}}{3}\right|^2 < \left|1 - \frac{2}{3}\rho e^{i\theta} + \frac{1}{6}\rho^2 e^{2i\theta}\right|^2 \\ &\Leftrightarrow 2\rho(1 + \frac{1}{9}\rho^2)\cos(\theta) < \frac{2}{3}\rho^2\cos^2(\theta) + \frac{1}{36}\rho^4 \ . \end{aligned}$$

Since the last inequality is always true for $\rho > 0$ and $\pi/2 < \theta < 3\pi/2$ (because $2\rho(1+\rho^2/9) > 0$, $\cos(\theta) < 0$ and $2\rho^2\cos^2(\theta)/3 + \rho^4/36 > 0$), the method is A-stable. We will see shortly that this is typical for a large class of implicit Runge-Kutta methods. ♣

Lemma 13.6.40

Let r be a rational non-constant function. $|r(z)| < 1$ in $\{z \in \mathbb{C} : \mathrm{Re}\, z < 0\}$ if and only if r has no pole in $\{z \in \mathbb{C} : \mathrm{Re}\, z \leq 0\}$ and $|r(z)| \leq 1$ for all z on the imaginary axis.

Proof.
Let $D = \{z \in \mathbb{C} : \mathrm{Re}\, z < 0\}$. So, $\overline{D} = \{z \in \mathbb{C} : \mathrm{Re}\, z \leq 0\}$.

Suppose that $|r(z)| < 1$ in D. Then $|r(z)| \leq 1$ in $\overline{D}$ by continuity. Thus, r has no pole in $\overline{D}$ and $|r(z)| \leq 1$ for all z on the imaginary axis.

Conversely, suppose that r has no pole in $\overline{D}$ and $|r(z)| \leq 1$ for all z on the imaginary axis. The function r cannot reach its absolute maximum in D because it is an analytic and non constant function on the open set D[6]. However, r must reach its absolute maximum at one point of $\overline{D}$ because it is continuous on $\overline{D}$ [7]. Since r is not constant, r reaches its absolute maximum on the imaginary axis only; the boundary of $\overline{D}$. Since $r(z) \leq 1$ for all z on the imaginary axis, then $r(z) < 1$ in D. ∎

Example 13.6.41 (**Example 13.6.39 continued**)
The poles of

$$r(z) = \frac{1 + z/3}{1 - 2z/3 + z^2/6}$$

are the roots of $1 - 2z/3 + z^2/6$; namely, $z_\pm = 2 \pm i\sqrt{2}$. The poles of $r(z)$ are not in the half-plane $\{z \in \mathbb{C} : \mathrm{Re}\, z \leq 0\}$.

[6] We use the Maximum Modulus Theorem from complex analysis.

[7] Since r has no pole at infinity, We may consider that r is a continuous function on the compactification of $\overline{D}$.

Moreover, on the imaginary axis, $z = ti$ with $t \in \mathbb{R}$. Thus,

$$r(ti) = \frac{1 + ti/3}{1 - 2it/3 - t^2/6}$$

and

$$|r(ti)| \leq 1 \Leftrightarrow \left|1 + \frac{ti}{3}\right|^2 \leq \left|1 - \frac{2it}{3} - \frac{t^2}{6}\right|^2 \Leftrightarrow 0 \leq \frac{t^4}{36} \ .$$

Since the last inequality is true for all $t \in \mathbb{R}$, $|r(z)| \leq 1$ on the imaginary axis. Thus, from the previous lemma, $|r(z)| < 1$ for all z in $\{z \in \mathbb{C} : \operatorname{Re} z < 0\}$. Namely, the method is A-stable as we have already shown in example 13.6.39. ♣

Lemma 13.6.42

Suppose that r is the rational function associated to a Runge-Kutta method as in Proposition 13.6.35 and that the Runge-Kutta method is of order p, then $r(z) = e^z + O(z^{p+1})$ as $z \to 0$.

Proof.
By definition of order, $y(t_{i+1}) = y(t_i) + h\,\phi(t_i, y(t_i)) + O(h^{p+1})$ where $\phi(t_i, w_i)$ represents the right hand side summation in the Runge-Kutta method. Thus, for $i = 0$, we get

$$y(t_1) = y(t_0) + h\,\phi(t_0, y(t_0)) + O(h^{p+1}) = w_0 + h\,\phi(t_0, w_0) + O(h^{p+1}) = w_1 + O(h^{p+1})$$

because $w_0 = y_0 = y(t_0)$. But $w_{i+1} = r(h\mu)w_i$ for $i \geq 0$ and the solution of (13.6.31) is $y(t) = e^{t\mu}w_0$. Thus,

$$e^{h\mu}w_0 = r(h\mu)w_0 + O(h^{p+1}) \ .$$

We get the conclusion of the lemma after a division by w_0 on both sides of the previous equality. ∎

Theorem 13.6.43

A Runge-Kutta method given by a collocation method satisfying Corollary 13.4.16 with $k > 0$ is A-stable.

Proof.
From Corollary 13.4.16, the Runge-Kutta method is of order $2k$. For the initial value problem (13.6.31), we have from Proposition 13.6.35 that the approximation w_j of $y(t_j)$ given by the Runge-Kutta method is $w_i = (r(h\lambda))^i w_0$ for $i \geq 0$, where $r(z)$ is the quotient of two polynomials of degree at most k. From Lemma 13.6.42, $r(z)$ is a "Padé approximation" of e^z of order $2k$. From Wanner-Hairer-Norsett theorem[8], r is associated to a A-stable method. ∎

[8] Roughly, this theorem states that a $r(z) = p(z)/q(z)$, a "Padé approximante to the exponential function", is A-acceptable if and only if the p and q have a specific form and $\deg p \leq \deg q \leq 2 + \deg p$.

13.6.4.2 Multistep Methods

The finite difference formula (13.5.1) applied to (13.6.31) becomes

$$w_{i+1} = \sum_{k=0}^{m} a_k w_{i-k} + h\mu \sum_{k=-1}^{m} b_k w_{i-k} \ .$$

If we substitute λ^i for w_i, we get

$$\lambda^{i+1} = \sum_{k=0}^{m} a_k \lambda^{i-k} + h\mu \sum_{k=-1}^{m} b_k \lambda^{i-k} \ .$$

If we subtract λ^{i+1} from both sides of this equality and multiply them by λ^{m-i}, we get

$$p(\lambda) + h\mu\, q(\lambda) = 0 \ ,$$

where

$$p(\lambda) = -\lambda^{m+1} + \sum_{k=0}^{m} a_k \lambda^{m-k} \quad \text{and} \quad q(\lambda) = \sum_{k=-1}^{m} b_k \lambda^{m-k} \ .$$

We have already defined $p(\lambda)$ as the characteristic polynomial of the multistep method given in Definition 13.5.1. We add another definition.

Definition 13.6.44

The **stability polynomial** of the multistep method given in Definition 13.5.1 is the polynomial $p(\lambda) + h\mu\, q(\lambda)$.

Remark 13.6.45

1. We have from Proposition 13.6.11 that $1 = \sum_{i=0}^{m} a_i$ (i.e. $p(1) = 0$) if the multistep method is consistent. Thus, a necessary condition for the consistency of a multistep method is the existence of 1 has a root of its characteristic polynomial.

2. For the multistep method given in Definition 13.5.1, we have seen that the finite difference formula (13.5.1) was derived from a formula of the form

$$y_{i+1} = \sum_{k=0}^{m} a_k y_{i-k} + h \sum_{k=-1}^{m} b_k f(t_{i-k}, y_{i-k}) + h\tau_{i+1}(h)$$

for $m \leq i < N$. Since $y(t) = Ae^{\mu t}$ is the general solution of $y' = \mu y$,

$$y_i = Ae^{\mu(t_0+ih)} = Ae^{\mu t_0}\left(e^{\mu h}\right)^i$$

is a solution of

$$y_{i+1} = \sum_{k=0}^{m} a_k y_{i-k} + h\mu \sum_{k=-1}^{m} b_k y_{i-k} + h\tau_{i+1}(h) \ .$$

Thus

$$\left(e^{\mu h}\right)^{i+1} = \sum_{k=0}^{m} a_k \left(e^{\mu h}\right)^{i-k} + h\mu \sum_{k=-1}^{m} b_k \left(e^{\mu h}\right)^{i-k} + h\left(Ae^{\mu t_0}\right)^{-1} \tau_{i+1}(h) \ .$$

If the multistep method is consistent for (13.6.31), one of the roots of the stability polynomial must approximate $e^{\mu h}$ for h small. This root is called the **principal root** of the stability polynomial.

3. Because the roots of the stability polynomial are continuous functions of h, the roots of the characteristic polynomial can be use to approximate the roots of the stability polynomial for h small.

♠

We can now restate the definition of absolute stability for the multistep methods defined in Definition 13.5.1.

Definition 13.6.46

The **region of absolute stability** of a multistep method as defined in definition 13.5.1 is the set of all values $h\mu \in \mathbb{C}$ such that $\lim_{i\to+\infty} w_i = 0$ for all solutions $\{w_i\}_{i=0}^{\infty}$ of the difference equation (13.5.1) applied to the initial value problem (13.6.31).
We say that a multistep method is **absolutely stable** for the value $h\mu$ if $h\mu$ is in the region of absolute stability.
A multistep method is **A-stable** if the region of absolute stability contains the half-plane to the left of the imaginary axis (complex numbers with a negative real part.)

Remark 13.6.47
As for the Runge-Kutta methods, we have to remember that $\operatorname{Re}\mu < 0$ in the previous definition. Hence, all solutions $y(t) = e^{\mu(t-t_0)}y_0$ of the differential equation (13.6.31) satisfy $\lim_{t\to\infty} y(t) = 0$. ♠

The following example illustrates the crucial role played by absolute stability.

Example 13.6.48
Consider the initial value problem

$$\begin{aligned} y'(t) &= 1 - 2y(t) \quad , \quad 0 \leq t \leq 4 \\ y(0) &= 1 \end{aligned} \tag{13.6.40}$$

The exact general solution of $y' = 1 - 2y$ is $y(t) = ce^{-2t} + 1/2$. The initial condition $y(0) = 1$ gives $c = 1/2$.

If we use the Euler's Method with $N = 128$, then $h = (t_f - t_0)/N = 1/32$, $t_i = t_0 + ih = i/32$ for $i = 0, 1, \ldots, 128$ and the approximations w_i of y_i are given by the difference equation

$$\begin{aligned} w_0 &= 1 \\ w_{i+1} &= w_i + h(1 - 2w_i) \end{aligned}$$

for $i = 0, 1, \ldots, 127$. The values of some of the w_i are given in Table 13.4 and the graph of the approximation of y given by the w_i can be found in Figure 13.6.

i	t_i	w_i	y_i	$w_i - y_i$	$\lvert y_i - w_i\rvert/\lvert y_i\rvert$
0	0	1	1	0	0
8	0.25	0.79835974	0.80326533	−0.00490559	0.00610706
16	0.5	0.67803707	0.68393972	−0.00590266	0.00863038
24	0.75	0.60623818	0.61156508	−0.00532690	0.00871027
32	1	0.56339439	0.56766764	−0.00427325	0.00752773
40	1.25	0.53782867	0.54104250	−0.00321383	0.00594007
48	1.5	0.52257310	0.52489353	−0.00232043	0.00442076
56	1.75	0.51346981	0.51509869	−0.00162888	0.00316227
64	2	0.50803770	0.50915782	−0.00112012	0.00219995
72	2.25	0.50479625	0.50555450	−0.00075825	0.00149983
80	2.5	0.50286202	0.50336897	−0.00050696	0.00100713
88	2.75	0.50170782	0.50204339	−0.00033556	0.00066840
96	3	0.50101909	0.50123938	−0.00022029	0.00043948
104	3.25	0.50060811	0.50075172	−0.00014361	0.00028679
112	3.5	0.50036287	0.50045594	-0.9307×10^{-4}	0.00018596
120	3.75	0.50021653	0.50027654	-0.6001×10^{-4}	0.00011995
128	4	0.50012921	0.50016773	-0.3852×10^{-4}	0.7702×10^{-4}

Table 13.4: Some results from the Euler's method used in Example 13.6.48 to approximate the solution of (13.6.40).

If we use the Adams-Bashforth method of order two from Example 13.6.7 with $N = 128$, then $h = (t_f - t_0)/N = 1/32$, $t_i = t_0 + ih = i/32$ for $i = 0, 1, \ldots, 128$ and the approximations w_i of y_i are given by the difference equation

$$\begin{aligned} w_0 &= 1 \\ w_1 &= y(1/32) = (e^{-1/16} + 1)/2 = 9.69706531\ldots \times 10^{-1} \\ w_{i+1} &= w_{i-1} + 2hf(t_i, w_i) = w_{i-1} + 2h(1 - 2w_i) \end{aligned}$$

for $i = 1, 2, \ldots, 127$. The values of some of the w_i are given in Table 13.5 and the graph of the approximation of y given by the w_i can be found in Figure 13.6.

For the first steps, the magnitude of the absolute error for the Euler's method is bigger than the magnitude of the absolute error for the Adams-Bashforth method of order two. However, the magnitude of the absolute error for the Euler's method decreases as the index i increases while the magnitude of the absolute error for the Adams-Bashforth method of order two increases as the index i increases. As i approaches 128, the values of w_i start to oscillate between values above and values below the exact value of y at t_i. For i near 128 the approximation w_i given by the Adams-Bashforth method of order two has only one significant digit.

To find out why the Adams-Bashforth method of order two gives a poor approximation of the solution of (13.6.40), we rewrite the equation $w_{i+1} = w_{i-1} + 2h(1 - 2w_i)$ as

$$w_{i+1} + 4hw_i - w_{i-1} = 2h \ . \tag{13.6.41}$$

i	t_i	w_i	y_i	$w_i - y_i$	$\lvert y_i - w_i\rvert/\lvert y_i\rvert$
0	0	1	1	0	
1	0.03125	0.96970653	0.96970653	0	0
8	0.25	0.80337381	0.80326533	0.00010848	0.00013505
16	0.5	0.68408166	0.68393972	0.00014194	0.00020754
24	0.75	0.61171439	0.61156508	0.00014931	0.00024415
32	1	0.56782462	0.56766764	0.00015698	0.00027654
40	1.25	0.54122426	0.54104250	0.00018176	0.00033594
48	1.5	0.52513250	0.52489353	0.00023897	0.00045527
56	1.75	0.51544736	0.51509869	0.00034867	0.00067691
64	2	0.50969996	0.50915781	0.00054215	0.00106479
72	2.25	0.50642522	0.50555450	0.00087072	0.00172230
80	2.5	0.50478834	0.50336897	0.00141937	0.00281974
81	2.53125	0.50167598	0.50316486	−0.00148887	0.00295902
82	2.5625	0.50457884	0.50297311	0.00160573	0.00319249
83	2.59375	0.50110363	0.50279298	−0.00168935	0.00335993
88	2.75	0.50437208	0.50204339	0.00232869	0.00463843
89	2.78125	0.49945547	0.50191958	−0.00246412	0.00490939
90	2.8125	0.50444014	0.50180328	0.00263686	0.00525477
91	2.84375	0.49890045	0.50169403	−0.00279358	0.00556829
96	3	0.50507032	0.50123938	0.00383094	0.00764293
104	3.25	0.50706105	0.500751720	0.00630933	0.01259971
112	3.5	0.51085173	0.50045594	0.01039579	0.02077264
113	3.53125	0.48936667	0.50042832	−0.01106164	0.02210435
114	3.5625	0.51218090	0.50040237	0.01177853	0.02353812
115	3.59375	0.48784406	0.50037799	−0.01253393	0.02504892
120	3.75	0.51740867	0.50027654	0.01713212	0.03424531
121	3.78125	0.48202618	0.50025979	−0.01823360	0.03644827
128	4	0.52840330	0.50016773	0.02823557	0.05645221

Table 13.5: Some results from the Adams-Bashforth method of order two used in Example 13.6.48 to approximate the solution of (13.6.40).

The general solution of (13.6.41) is

$$w_i = c_1\lambda_1^i + c_2\lambda_2^i + \frac{1}{2} \ , \tag{13.6.42}$$

where

$$\lambda_1 = -2h + \sqrt{1+4h^2} = 1 - 2h + O(h^2) \quad \text{and} \quad \lambda_2 = -2h - \sqrt{1+4h^2} = -1 - 2h + O(h^2)$$

are the roots of $\lambda^2 + 4h\lambda - 1 = 0$, and c_1 and c_2 are arbitrary constants. Note that $w_i = 1/2$ for all i is a solution of the non-homogeneous equation (13.6.41) and $w_i = c_1\lambda^i + c_2\lambda_2^i$ is the general solution of the homogeneous equation

$$w_{i+1} + 4hw_i - w_{i-1} = 0 \ .$$

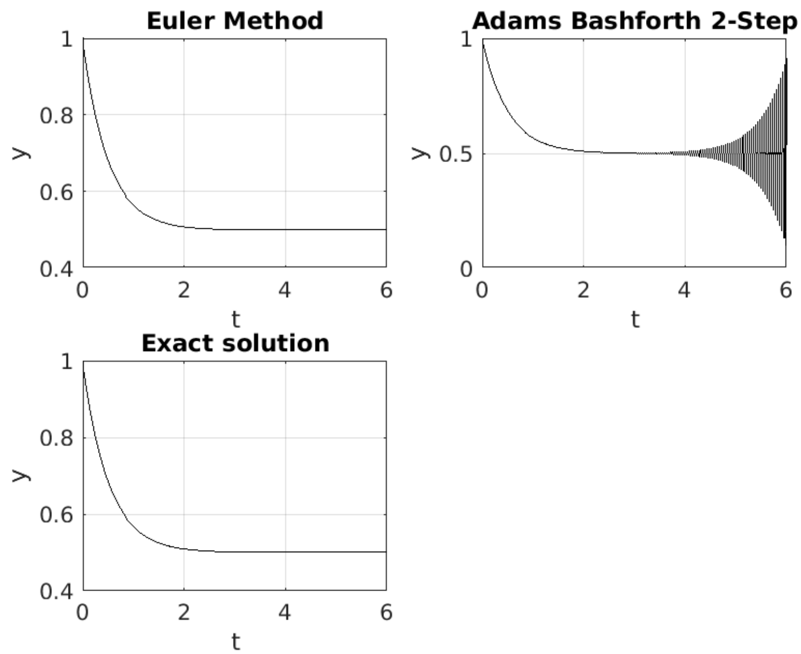


Figure 13.6: Approximation of the solution of $y'(t) = 1 - 2y(t)$ where $0 \leq t \leq 4$ and $y(0) = 1$ given by three different numerical methods.

The initial conditions $w_0 = y(0) = 1$ and $w_1 = y(1/32) = 0.96970\ldots$ gives $c_1 = 0.50793\ldots \cong 1/2$ and $c_2 = 0.0079\ldots \cong 0$ because of round off errors. The exact values are $c_1 = 1/2$ and $c_2 = 0$. Even if the initial conditions were such that $c_2 = 0$, the effect of rounding error is equivalent to having $c_2 \neq 0$. Because $|\lambda_2| > 1$, the magnitude of the second term of (13.6.42) increases as i increases. This explains the increase of the errors as i increases.

If we substitute λ_1 and λ_2 in (13.6.42), we get

$$\begin{aligned} w_i &= c_1\left(-2h + \sqrt{1+4h^2}\right)^i + c_2\left(-2h - \sqrt{1+4h^2}\right)^i + \frac{1}{2} \\ &= c_1\left(1 - 2h + O(h^2)\right)^i + c_2(-1)^i\left(1 + 2h + O(h^2)\right)^i + \frac{1}{2} \\ &\approx c_1(1-2h)^i + c_2(-1)^i(1+2h)^i + \frac{1}{2} \approx c_1 e^{-2t_i} + c_2(-1)^i e^{2t_i} + \frac{1}{2} \end{aligned} \tag{13.6.43}$$

for h very small. For the last approximation above, we note that

$$(1-2h)^i = \left((1-2h)^{-1/(2h)}\right)^{-2ih} ,$$

where $\lim_{h\to 0}(1-2h)^{-1/(2h)} = e$. Thus, for h very small, we may assume that $(1-2h)^i \approx e^{-2ih} = e^{-2t_i}$. Similarly, $(1+2h)^i \approx e^{2t_i}$ for h very small.

From (13.6.43) we may conclude that the first term in (13.6.42) is associated to the solution of $y' = 1 - 2y$ but the second term in (13.6.42) exists only because we have transformed the first order differential equation $y' = 1 - 2y$ into a second order difference equation $w_{i+1} = w_{i-1} + 2h(1 - 2w_i)$.

This example shows that it is important to study the magnitude of the roots of the stability polynomial associated to a multistep method. ♣

Remark 13.6.49
The Adams-Bashforth method of order two from Example 13.6.7 is consistent and, as we saw in Example 13.6.22, satisfies the root condition. Thus, this Adams-Bashforth method of order two is convergent according to Theorem 13.6.26. However, we saw in the numerical experiment of Example 13.6.48 that this Adams-Bashforth method of order two does not converge. Is there anything wrong with Theorem 13.6.26? No, there is nothing wrong mathematically but the theory assumes that $\delta_i(h) = O(h^2)$ which is rarely satisfied by round off errors (round off errors do not generally go to 0 as h decreases). Moreover, the theory does not take into account the information provided by the stability polynomial that may have many roots. This illustrate the limits of the theory presented so far where we ignore the information provided by the stability polynomial and the full effect of round off errors. We therefore need a stability criteria which is stronger than the root condition. Absolute stability is this criteria. ♠

Proposition 13.6.50

The region of absolute stability is the set of all value $h\mu \in \mathbb{C}$ with $\operatorname{Re}\mu < 0$ such that all the roots of the stability polynomial have absolute values less than one.

Proof.
The multistep method (13.5.1) applied to this initial value problem (13.6.31) is the finite difference equation

$$\sum_{k=-1}^{m} (a_k + h\mu\, b_k) w_{i-k} = 0 \quad , \quad i \geq m \ ,$$

where $a_{-1} = -1$. A solution $\{w_i\}_{i=0}^{\infty}$ of this finite difference equation is a linear combination of solutions of the form $\{i^n \lambda^i\}_{i=0}^{\infty}$, where $\lambda \in \mathbb{C}$ is a root of multiplicity s of the stability polynomial

$$p(\lambda) + h\mu q(\lambda) = \sum_{k=-1}^{m} (a_k + h\mu\, b_k)\lambda^{m-k} \tag{13.6.44}$$

and $0 \leq n < s$. Hence, $|\lambda| < 1$ for all roots of (13.6.44) if and only if any non-trivial solution $\{w_i\}_{i=0}^{\infty}$ of the finite difference equation above satisfies $\lim_{i\to+\infty} w_i = 0$. Namely, if and only if $h\mu$ is in the region of absolute stability. ∎

Example 13.6.51
We illustrate with the Euler's method why we should choose $h\mu$ in the region of absolute stability.

Consider the initial value problem (13.6.31). The exact solution is $y(t) = y_0\, e^{\mu t}$. The approximation w_i of y_i given by the Euler's method (Definition 13.2.1) is the solution of

$$w_0 = y_0$$

$$w_{i+1} = w_i + h\mu\, w_i = (1 + h\mu)w_i$$

Thus, $w_i = (1 + h\mu)^i y_0$ for $i \geq 0$.

Since $\operatorname{Re}\mu < 0$, we have that $y(t) \to 0$ as $t \to 0$. To get the same behaviour for w_i (i.e. $w_i \to 0$ as $i \to \infty$), we need $|1 + h\mu| < 1$.

Suppose that an error δ_0 is introduced in the initial condition; namely, $w_0 = y_0 + \delta_0$. The new value of w_i is $(1 + h\mu)^i y_0 + (1 + h\mu)^i \delta_0$ which differ from the unperturbed value of w_i by $(1 + h\mu)^i \delta_0$. To have this difference decreases as $i \to \infty$, we need $|1 + h\mu| < 1$.

Let us show that $|1+h\mu| < 1$ is the condition for $h\mu$ to be in the region of absolute stability. The stability polynomial of the Euler's method is $-\lambda + (1 + h\mu)$. Obviously, the only root is $\lambda = 1 + h\mu$. The region of absolute stability is $\{h\mu : |1 + h\mu| < 1\}$. This is the open disk of radius 1 centred at $(-1, 0)$ in the complex plane. ♣

Remark 13.6.52
Suppose that $p(\lambda)+h\mu\, q(\lambda)$ is the stability polynomial of a multistep method. If we draw the graph of $z = -p(\lambda)/q(\lambda)$ for λ on the unit circle in the complex plane, we get the boundary of the region of absolute stability of the multistep method; namely, the values of $h\mu$ for which $|\lambda| = 1$. ♠

Example 13.6.53
The stability polynomial for the Adams-Bashforth method of order two from Example 13.6.7 is

$$p(\lambda) + h\mu q(\lambda) = -\lambda^2 + 1 + 2h\mu\lambda \ .$$

If λ is a root of this polynomial such that $|\lambda| = 1$, we may assume that $\lambda = e^{\theta i}$ for some $\theta \in [0, 2\pi[$. We get

$$-e^{2\theta i} + 1 + 2h\mu e^{\theta i} = 0 \Rightarrow h\mu = \frac{e^{2\theta i} - 1}{2e^{\theta i}} = \frac{e^{\theta i} - e^{\theta i}}{2} = i\sin(\theta) \ .$$

Thus, the boundary of the region of absolute stability is the segment $\{ri : -1 \leq r \leq 1\}$ on the imaginary axis. All points outside this segment are on curves given by $\lambda = re^{\theta i}$ for both $r > 1$ and $r < 1$. The region of absolute stability has no interior (Figure 13.7). Thus, the region of absolute stability is empty. This explains why this method fails to converge in Example 13.6.48. ♣

Example 13.6.54
The boundaries for the region of absolute stability for the Adams-Bashforth method of order four is drawn in Figure 13.8. To produce this boundary, we drew the graph of $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ with $0 \leq \theta < 2\pi$. For the Adams-Bashforth method of order four $p(\lambda) = -\lambda^4 + \lambda^3$ and $q(\lambda) = (55\lambda^3 - 59\lambda^2 + 37\lambda - 9)/24$.

The boundaries for the region of absolute stability for the Adams-Moulton method of order four is drawn in Figure 13.9. To produce this boundary, we drew the graph of $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ with $0 \leq \theta < 2\pi$. For the Adams-Moulton method of order four $p(\lambda) = -\lambda^3 + \lambda^2$ and $q(\lambda) = (9\lambda^3 + 19\lambda^2 - 5\lambda + 1)/24$.

The regions inside the boundary curves (the black curve) and to the left of the imaginary axis are the regions of absolute stability. The two lobes for the Adams-Bashforth method of

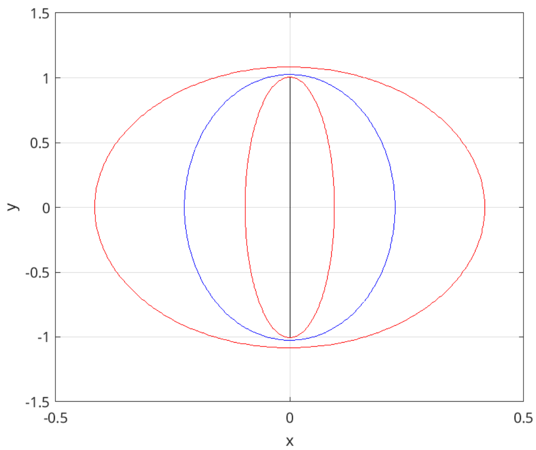


Figure 13.7: Regions of absolute stability for the Adams-Bashforth method of order two of Example 13.6.7. We have drawn the curve $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ (black curve), for $\lambda = 1.1e^{i\theta}$ and $\lambda = 1.5e^{i\theta}$ (red curves) and $\lambda = 0.8e^{i\theta}$ (blue curve). The region of absolute stability is empty.

order four do not represent regions of absolute stability. To justify this, we have also drawn the curves $z = -p(\lambda)/q(\lambda)$ for $\lambda = 1.2e^{i\theta}$ and $\lambda = 0.8e^{i\theta}$ with $0 \leq \theta < 2\pi$ for the two methods. The points z on the curves in red associated to $\lambda = 1.2e^{i\theta}$ correspond to values of $h\mu$ for which the stability polynomial has a root λ of absolute value 1.2, these points are therefore outside the region of absolute stability, whereas the points z on the curves in blue associated to $\lambda = 0.8e^{i\theta}$ correspond to values of $h\mu$ for which the stability polynomial has a root λ of absolute value 0.8, these points are inside the region of absolute stability. ♣

Example 13.6.55

The region of absolute stability of the Trapezoidal method is the half-plane to the left of the imaginary axis. Hence, the trapezoidal method is A-stable.

The stability polynomial of the trapezoidal method is $p(\lambda) + h\mu\, q(\lambda)$ where $p(\lambda) = -\lambda^2 + \lambda$ and $q(\lambda) = (\lambda^2 + \lambda)/2$. We draw the graph of $z = -p(\lambda)/q(\lambda)$ for λ on the unit circle in Figure 13.10.

We now prove rigorously that the region of absolute stability of the Trapezoidal Method is the half-plane to the left of the imaginary axis.

The Trapezoidal Method applied to the initial value problem (13.6.31) gives

$$w_{i+1} = w_i + \frac{h}{2}\left(\mu\, w_{i+1} + \mu\, w_i\right) \quad , \quad 0 \leq i < N \ .$$

If we solve for w_{i+1}, we get

$$w_{i+1} = \left(\frac{1 + h\mu/2}{1 - h\mu/2}\right) w_i \ .$$

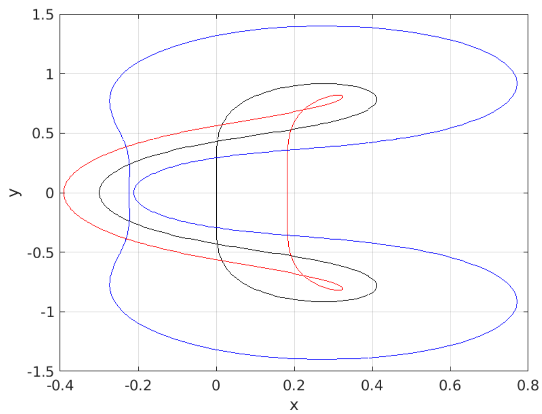


Figure 13.8: Regions of absolute stability for the Adams-Bashforth method of order four. We have drawn the curve $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ (black curve), for $\lambda = 1.2e^{i\theta}$ (red curve) and $\lambda = 0.8e^{i\theta}$ (blue curve). The region to the left of the imaginary axis and inside the black curve is the region of absolute stability.

Hence, by induction,

$$w_{i+1} = \left(\frac{1 + h\mu/2}{1 - h\mu/2}\right)^{i+1} w_0 \quad , \quad i \geq 0 \ .$$

The region of absolute stability is

$$\left\{h\mu : \left|\frac{1 + h\mu/2}{1 - h\mu/2}\right| < 1\right\} = \{z : \operatorname{Re} z < 0\}$$

because

$$\left|\frac{1 + h\mu/2}{1 - h\mu/2}\right| < 1 \Leftrightarrow \left|1 + \frac{h\mu}{2}\right|^2 < \left|1 - \frac{h\mu}{2}\right|^2 \Leftrightarrow \operatorname{Re} h\mu < 0 \ .$$

♣

Remark 13.6.56
The fact that the trapezoidal method is A-stable does not mean that all values of h can be used. Consider the differential equation

$$\begin{aligned} y'(t) &= \mu(t)y(t) \quad , \quad t \geq 0 \\ y(0) &= y_0 \end{aligned}$$

where μ is a differentiable function such that $\mu(t) < 0$ and $\mu'(t) > 0$ for all $t > 0$. All solutions y of this differential equation satisfy $\lim_{t\to\infty} y(t) = 0$.

The trapezoidal method gives

$$w_{i+1} = \frac{1 + h\mu(t_i)/2}{1 - h\mu(t_{i+1})/2}\, w_i$$

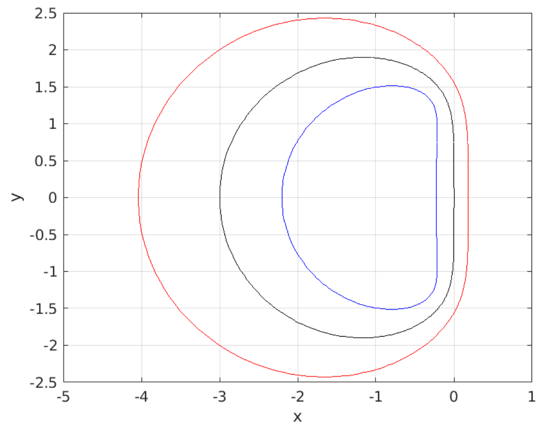


Figure 13.9: Regions of absolute stability for the Adams-Moulton method of order four. We have drawn the curve $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ (black curve), for $\lambda = 1.2e^{i\theta}$ (red curve) and $\lambda = 0.8e^{i\theta}$ (blue curve). The region to the left of the imaginary axis and inside the black curve is the region of absolute stability.

for $i \geq 0$. Since $\mu(t) < 0$ for all $t > 0$, we still need

$$\left| \frac{1 + h\mu(t_i)/2}{1 - h\mu(t_{i+1})/2} \right| < 1$$

to ensure that $w_i \to 0$ as $i \to \infty$. However, if $\mu(t) \to 0$ as $t \to \infty$, we will have that $\left| \dfrac{1 + h\mu(t_i)/2}{1 - h\mu(t_{i+1})/2} \right| \to 1$ as $i \to \infty$. Thus, the convergence of w_i to 0 will be really slow and taking h smaller will further slow the convergence. ♠

Example 13.6.57
Consider the initial value problem

$$\begin{aligned} &y'(t) = 100y(t) + 100t^2 - 2t - 100 \quad , \quad 0 \leq t \leq 1 \\ &y(0) = 1 \end{aligned} \tag{13.6.45}$$

If we use the modified Euler's method, the Runge-Kutta method of order four and the Adams-Bashfort method of order four to approximate $y(1)$, we get the following results:

number N of steps	Approximation of $y(1)$: Modified Euler's method	Runge-Kutta Method of order 4	Adams-Bashforth Method of order 4
10	$5.9445\ldots \times 10^{14}$	$3.9941\ldots \times 10^{24}$	$2.9627\ldots \times 10^{14}$
20	$7.8754\ldots \times 10^{21}$	$2.0564\ldots \times 10^{32}$	$2.9976\ldots \times 10^{19}$
30	$1.4899\ldots \times 10^{26}$	$2.6381\ldots \times 10^{35}$	$3.2001\ldots \times 10^{23}$
40	$9.7746\ldots \times 10^{28}$	$5.5303\ldots \times 10^{36}$	$4.2923\ldots \times 10^{26}$

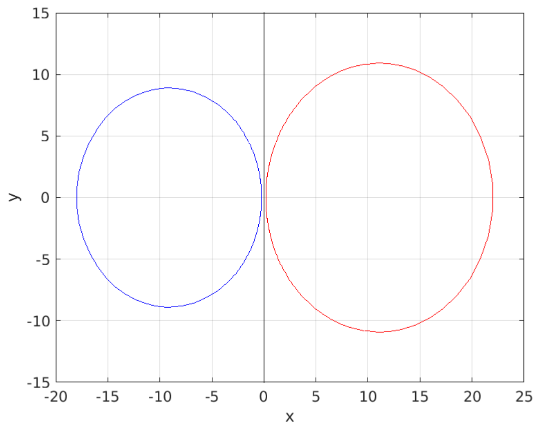


Figure 13.10: Regions of absolute stability for the trapezoidal method. We have drawn the curve $z = -p(\lambda)/q(\lambda)$ for $\lambda = e^{i\theta}$ (black curve), for $\lambda = 1.2e^{i\theta}$ (red curve) and $\lambda = 0.8e^{i\theta}$ (blue curve). The region to the left of the imaginary axis is the region of absolute stability.

The exact solution of (13.6.45) is $y(t) = 1 - t^2 + Ce^{100t}$. The initial condition $y(0) = 1$ implies that $C = 0$. However, because of round off error, the solution that we compute is one with $C \neq 0$ small.

We now use the trapezoidal method to approximate $y(1)$. First, we have to explain how to implement the trapezoidal method.

Given w_i, we have to solve $w_{i+1} = w_i + \dfrac{h}{2}\left(f(t_{i+1}.w_{i+1}) + f(t_i, w_i)\right)$ for w_{i+1}. This is an implicit equation for w_{i+1}. In general, this equation cannot be solved explicitly for w_{i+1}. To compute w_{i+1}, we use Euler's method to get a first approximation of w_{i+1} and then use Newton's Method to approximate a root of $0 = z - w_i - \dfrac{h}{2}\left(f(t_{i+1}.z) + f(t_i, w_i)\right)$. The root of this equation is the value of w_{i+1}.

To get a first approximation of w_{i+1}, we apply the Euler's method with $N = 1$ to

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_i \leq t \leq t_{i+1} \\ y(t_i) &= w_i \end{aligned}$$

to get the first approximation of w_{i+1}. The Euler's method gives an approximation of $y(t_{i+1})$ if $w_i = y_i$.

The following code is an implementation of the trapezoidal method. Using this code with $t_0 = 0$, $t_f = 1$, $y_0 = 1$, the number of subinterval $N = 10$, the tolerance $T = 10^{-5}$ and the maximum number of iterations for the Newton's Method $M = 10$, we get

i	0	1	2	3	4	5	6	7	8	9	10
t_i	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
w_i	1.00	0.99	0.96	0.91	0.84	0.75	0.64	0.51	0.36	0.19	0.0

rounded to two decimal places. The approximations w_i of y_i are exact. ♣

Code 13.6.58 (Trapezoidal Method)

To approximate the solution of the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(0) &= y_0 \end{aligned}$$

Input: The initial time t_0 (t0 in the code below).
The final time t_f (tf in the code below).
The number N of subintervals of $[t_0, t_f]$.
The initial condition y_0 (y0 in the code below).
The tolerance T for the Newton's Method.
The maximum number of iterations M for the Newton's Method.
The function $f(t, y)$ (funct in the code below).
The derivative of $f(t, y)$ with respect to y (functprime in the code below).
Output: The approximations w_i (w(i+1) in the code below) of $y(t_i)$ for $i = 0, 1, \ldots,$ N, where $t_i = t_0 + ih$ (t(i+1) in the code below) with $h = (t_f - t_0)/N$.

```
function [t,w] = trapez(funct,functprime,t0,y0,tf,N,M,T)
  h = (tf-t0)/N;
  half = h/2;
  t(1) = t0;
  w(1) = y0;

  for i = 1:1:n
    % We start the iteration with the approximation of y(t_0 + h)
    % given by the Euler method.
    k = feval(funct,t(i),w(i));
    w0 = w(i) + h*k;
    t(i+1) = t(1) + i*h;

    % Newton-Raphson iterations
    for j = 1:M
      numer = w0 - w(i)- half*(funct(t(i+1),w0) + k);
      denum = 1-half*functprime(t(i+1),w0);

      if (denum == 0)
        % Newton-Raphson iterative method does not converge (fast enough).
        t = NaN;
        w = NaN;
        return;
      end

      w1 = w0 - numer/denum;
      if (abs(w1-w0) < T)
        w(i+1) = w1;
```

```
        break;
      else
        w0 = w1;
        if (j == max)
          % The maximum number of iterations has been reached
          % before getting an approximation of w(i+1) within the
          % required tolerance.
          t = NaN;
          w = NaN;
          return;
        end
      end
    end
  end
end
```

We conclude this section with a couple more results. This can be used as a starting point for further reading on the subject of this chapter.

Proposition 13.6.59

The multistep method (13.5.1) is A-stable if and only if $b_{-1} > 0$ and, for each $h\mu$ on the imaginary axis, the roots of the stability polynomial are less than or equal to 1 in absolute value.

Lemma 13.6.60 (Cohn-Schur Criterion)

Consider the quadratic equation $p(z) = az^2 + bz + c$, where $a, b, c \in \mathbb{C}$ and $a \neq 0$. Then, the roots of p are inside the closed disk of radius 1 centred at the origin if and only if $|a| \geq |c|$ and $\left||a|^2 - |c|^2\right| \geq |a\bar{b} - b\bar{c}|$

The proofs of these two results is sketched in [19]. We show in the next example how these results can be used.

Example 13.6.61
We prove that the backward divided difference

$$w_{i+1} - \frac{4}{3}w_i + \frac{1}{3}w_{i-1} = \frac{2}{3}hf(t_{i+1}, w_{i+1}) \quad , \quad 1 \leq i < N \tag{13.6.46}$$

is A-stable.

If we apply this multistep method to the initial value problem (13.6.31), we get

$$w_{i+1} - \frac{4}{3}w_i + \frac{1}{3}w_{i-1} = \frac{2}{3}h\mu\, w_{i+1}$$

for $1 \leq i \leq N-1$. Thus, the stability polynomial is

$$p(\lambda) + h\mu q(\lambda) = \left(-1 + \frac{2}{3}h\mu\right)\lambda^2 + \frac{4}{3}\lambda - \frac{1}{3} = 0 \ .$$

1. We have that $b_{-1} = 2/3 > 0$.

2. We show that the roots of the characteristic equation with $h\mu$ on the imaginary axis are less than or equal to 1.

   If we substitute $h\mu = ti$ with $t \in \mathbb{R}$ in the stability polynomial, we get
$$\left(-1 + \frac{2}{3}ti\right)\lambda^2 + \frac{4}{3}\lambda - \frac{1}{3} = 0 \ . \tag{13.6.47}$$

   We now apply Lemma 13.6.60 to this polynomial equation. We have $a = -1 + 2ti/3$, $b = 4/3$ and $c = -1/3$. Thus, for all $t \in \mathbb{R}$, we have that

   (a) $a \neq 0$.
   (b) $|a|^2 - |c|^2 = |1 - 2ti/3|^2 - |1/3|^2 = 4(2 + t^2)/9 > 0$. Thus $|a| > |c|$.
   (c) $\left(|a|^2 - |c|^2\right)^2 - |a\bar{b} - b\bar{c}|^2 = 16(2 + t^2)^2/81 - 64(1 + t^2)/81 = 16t^4/81 \geq 0$.

   Hence, the conditions of Lemma 13.6.60 are satisfied and we can conclude that, for all $t \in \mathbb{R}$, the roots of (13.6.47) are smaller than or equal to 1 in absolute value.

1 and 2 imply that the conditions of Proposition 13.6.59 are satisfied and so the method (13.6.46) is A-stable. ♣

The next theorem tells us that the Trapezoidal Method is basically the best multistep method that we can hope for if A-stability is required.

Theorem 13.6.62 (Dahlquist Second Barrier)

The highest order of an A-stable multistep method is 2.

Remark 13.6.63
We should not completely reject the higher order multistep methods. There are multistep methods of order higher than 2, though not A-stable, that are **A(α)-stable**; namely, the stability region contain the cone $\{z \in \mathbb{C} : z = \rho e^{\theta}$ with $\rho > 0$ and $|\theta - \pi| < \alpha\}$. For some multistep methods, α may be closed to $\pi/2$. ♠

13.6.5 Conclusion

Theorem 13.6.26 is a beautiful and simple theoretical result to ensure convergence of a numerical method to solve initial value problems. However, it can only be use as a necessary criteria to ensure convergence because of the strong hypothesis on the size of round off error. That was the motivation to introduce a stronger stability criteria; namely, absolute stability. However, even absolute stability is not ideal. It may impose strict conditions on the step-size h depending on the region of absolute stability. We will see in the next section that for some initial value problems (particularly in higher dimension), there may be conditions on the step-size that are almost impossible (if not impossible) to satisfy.

This demonstrates that solving initial value problems is still a subject of research since we are now intensively using initial value problems to model physical phenomena.

Though solving numerically initial value problems is not simple, it is still a lot simpler than solving partial differential equation as we will see in Chapter 15.

13.7 Stiff Systems and Stability

We consider the initial value problem

$$\begin{aligned} &\frac{\mathrm{d}y}{\mathrm{d}t}(t) = f(t, \mathbf{y}(t)) \quad , \quad t_0 \le t \le t_f \\ &\mathbf{y}(t_0) = \mathbf{y}_0 \end{aligned} \tag{13.7.1}$$

where $f : [t_0, t_f] \times \mathbb{R}^n \to \mathbb{R}^n$.

As we will see, there are additional constraints on the step-size h than those needed to get a converging and stable Runge-Kutta or multistep method to numerically solve (13.7.1). We already know that the requirements on the step size h to get a stable method may be stronger than what is necessary for the required accuracy. If an implicit multistep method is used with an iterative algorithm, h may have to be small enough for the iterations to converge as we have seen in Remark 13.5.8). In that remark, we showed that h had to be small enough to get $|b_{-1}hL| < 1$, where L is the Lipschitz constant associated to the second variable of the function f; namely, $|f|t, x) - f(t, y)| \le L|x - y|$ for all (t, x) and (t, y) in the domain of f. We will add to this list the case where some "components" of the solution vary much faster than others. This will be one of the major characteristics of Stiff differential equations.

The differential equation (13.7.1) is **stiff** when basically no reasonable choice of h can address all the constraints. A mistake often made is to say that a system is stiff when in fact the system is unstable.

Example 13.7.1
Consider the initial value problem

$$\begin{aligned} \mathbf{y}' &= A\mathbf{y} \quad , \quad 0 \le t \le t_f \\ \mathbf{y}(0) &= \mathbf{y}_0 \end{aligned} \tag{13.7.2}$$

where $\mathbf{y} : \mathbb{R} \to \mathbb{R}^2$, $\mathbf{y}_0 = \begin{pmatrix} y_{0,1} \\ y_{0,2} \end{pmatrix} \in \mathbb{R}^2$, and $A = \begin{pmatrix} -\lambda & 1 \\ 0 & -0.1 \end{pmatrix}$ for a large positive number λ.

Since $A = QBQ^{-1}$, where $Q = \begin{pmatrix} 1 & 1 \\ 0 & \lambda - 0.1 \end{pmatrix}$ and $B = \begin{pmatrix} -\lambda & 0 \\ 0 & -0.1 \end{pmatrix}$, the solution of (13.7.2) is

$$\mathbf{y} = e^{tA}\mathbf{y}_0 = Qe^{tB}Q^{-1}\mathbf{y}_0 = \begin{pmatrix} e^{-\lambda t} & (e^{-0.1t} - e^{-\lambda t})/(\lambda - 0.1) \\ 0 & e^{-0.1t} \end{pmatrix} \begin{pmatrix} y_{0,1} \\ y_{0,2} \end{pmatrix} \ . \tag{13.7.3}$$

Choose $N \in \mathbb{N}$. Let $h = t_f/N$ and $t_j = jh$ for $0 \leq j \leq N$. With the Euler's method for systems of ordinary differential equations, approximations $\mathbf{w}_j$ of the exact values $\mathbf{y}(t_j)$ of the solution of (13.7.2) are given by

$$\begin{aligned} \mathbf{w}_{j+1} &= (I_2 + hA)\mathbf{w}_j \quad , \quad 0 \leq j < N \\ \mathbf{w}_0 &= \mathbf{y}_0 \end{aligned} \tag{13.7.4}$$

The solution $\{\mathbf{w}_j\}_{j=0}^{\infty}$ of (13.7.4) is given by

$$\begin{aligned} \mathbf{w}_j &= (I_2 + hA)^j \mathbf{w}_0 = Q(I_2 + hB)^j Q^{-1}\mathbf{w}_0 \\ &= Q\begin{pmatrix} (1-\lambda h)^j & 0 \\ 0 & (1-0.1h)^j \end{pmatrix} Q^{-1}\mathbf{w}_0 \ . \end{aligned} \tag{13.7.5}$$

Suppose that $\mathbf{y}_0 = \begin{pmatrix} 1 \\ \lambda - 0.1 \end{pmatrix}$. This is an eigenvector of A associated to the eigenvalue -0.1. The solution of (13.7.2) for $0 \leq t \leq t_f$ is then given by $\mathbf{y}(t) = e^{-0.1t}\mathbf{y}_0$.

If h is small enough such that

$$(1-0.1h)^j \approx e^{-0.1jh} \tag{13.7.6}$$

and

$$|1 - \lambda h| < 1 \ , \tag{13.7.7}$$

then (13.7.5) will give a good approximation of $\mathbf{y}(t_j)$.

Unfortunately, because λ is a large positive number, the restriction (13.7.7) is much more severe than the restriction (13.7.6). So (13.7.7) may force h to be smaller than the computer accuracy. Therefore, the Euler's method will not give a good approximation of the solution of (13.7.2). This type of problems occurs in system that have two quite different "time-scales" as we have for the present system with the two requirements on h. ♣

Remark 13.7.2

Lambert [23] defines stiffness as follows. The differential equation (13.7.1) is **stiff** for $t_0 < t < t_f$ if, for all t between t_0 and t_f, $\operatorname{Re}\lambda_i(t) < 0$ for all eigenvalues $\lambda_i(t)$ of $\mathrm{D}_{\mathbf{y}} f(t, \mathbf{y}(t))$ and $\max_i\{\operatorname{Re}\lambda_i(t)\} \gg \min_i\{\operatorname{Re}\lambda_i(t)\}$. The ratio

$$\max_i\{\mathrm{Re}\ \lambda_i(t)\}/\min_i\{\mathrm{Re}\ \lambda_i(t)\}$$

is called the **stiffness ratio**.

We do not use this definition because it does not cover all the cases of stiffness as we have defined it. Note that if the stiffness ratio is large then the Lipschitz constant L will be large. ♠

Example 13.7.3

We use the Trapezoidal Method to approximate the solution of the initial value problem (13.7.2) considered in the previous example. Namely, we consider

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \frac{h}{2}\left(A\mathbf{w}_{i+1} + A\mathbf{w}_i\right) \quad , \quad 0 \leq i < N \ .$$

If we solve for $\mathbf{w}_{i+1}$, we get

$$\mathbf{w}_{i+1} = \left(\mathrm{Id} - \frac{h}{2}\,A\right)^{-1}\left(\mathrm{Id} + \frac{h}{2}\,A\right)\mathbf{w}_i \ .$$

Note that $\mathrm{Id} - (h/2)A$ is invertible for all h. By induction, we get that

$$\mathbf{w}_{i+1} = \left(\mathrm{Id} - \frac{h}{2}\,A\right)^{-i-i}\left(\mathrm{Id} + \frac{h}{2}\,A\right)^{i+1}\mathbf{w}_0 \ .$$

Since $A = QBQ^{-1}$ for $Q = \begin{pmatrix} 1 & 1 \\ 0 & \lambda - 0.1 \end{pmatrix}$ and $B = \begin{pmatrix} -\lambda & 0 \\ 0 & -0.1 \end{pmatrix}$, we get

$$\mathbf{w}_{i+1} = Q\left(\mathrm{Id} - \frac{h}{2}\,B\right)^{-i-i}\left(\mathrm{Id} + \frac{h}{2}\,B\right)^{i+1} Q^{-1}\mathbf{w}_0 \ .$$

Since

$$\mathrm{Id} - \frac{h}{2}B = \begin{pmatrix} 1 + \lambda h/2 & 0 \\ 0 & 1 + h/20 \end{pmatrix} \quad \text{and} \quad \mathrm{Id} + \frac{h}{2}B = \begin{pmatrix} 1 - \lambda h/2 & 0 \\ 0 & 1 - h/20 \end{pmatrix}$$

commute, we get

$$\begin{aligned} \mathbf{w}_{i+1} &= Q\left(\left(\mathrm{Id} - \frac{h}{2}\,B\right)^{-1}\left(\mathrm{Id} + \frac{h}{2}\,B\right)\right)^{i+1} Q^{-1}\mathbf{w}_0 \\ &= Q\begin{pmatrix} \left(\dfrac{1 - \lambda h/2}{1 + \lambda h/2}\right)^{i+1} & 0 \\ 0 & \left(\dfrac{1 - h/20}{1 + h/20}\right)^{i+1} \end{pmatrix} Q^{-1}\mathbf{w}_0 \ . \end{aligned}$$

We finally get

$$\mathbf{w}_{i+1} = \begin{pmatrix} \left(\dfrac{1 - \lambda h/2}{1 + \lambda h/2}\right)^{i+1} & \dfrac{1}{\lambda - 0.1}\left(\left(\dfrac{1 - h/20}{1 + h/20}\right)^{i+1} - \left(\dfrac{1 - \lambda h/2}{1 + \lambda h/2}\right)^{i+1}\right) \\ 0 & \left(\dfrac{1 - h/20}{1 + h/20}\right)^{i+1} \end{pmatrix}\mathbf{w}_0 \quad , \quad 0 \leq i < N \ .$$

Independently of the choice of h, we have that $\mathbf{w}_i \to \mathbf{0}$ as $i \to \infty$ because $\left|\dfrac{1 - \lambda h/2}{1 + \lambda h/2}\right| < 1$ and $\left|\dfrac{1 - h/20}{1 + h/20}\right| < 1$ for all h.

So there is no constraints on h other than the one that we may impose for the accuracy.

In general, we should use A-stable methods, like the Trapezoidal Method, to solve stiff differential equations. ♣

13.8 Exercises

Question 13.1
Show that the following initial value problems are well posed.

a) $$\begin{aligned} y'(t) &= 2y(t) + 2 \quad , \quad 0 \le t \le 1 \\ y(0) &= 1 \end{aligned}$$

b) $$\begin{aligned} y'(t) &= t^2 y(t) + 1 \quad , \quad 0 \le t \le 1 \\ y(0) &= 1 \end{aligned}$$

c) $$\begin{aligned} y'(t) &= t^2 \sin(y(t)) + y(t) \quad , \quad 0 \le t \le 1 \\ y(0) &= 1 \end{aligned}$$

Question 13.2
Consider the initial value problem

$$\begin{aligned} y' &= 1 - y \quad , \quad 0 \le t \le 1 \\ y(0) &= 0 \end{aligned}$$

a) Estimate the value of h (the step size) that minimize the error bound for the Euler's method. Assume that all rounding errors have magnitude less than 10^{-8}.
b) With the value of h found in (a), compute the error bound on the interval $[0, 1]$.

Question 13.3
Consider the initial value problem

$$\begin{aligned} y' &= \frac{y+t}{t} \quad , \quad 1 \le t \le 2 \\ y(1) &= 0 \end{aligned}$$

a) Show that this initial value problem is well posed.
b) Estimate the value of the step size h that minimizes the error bound for the Euler's method. Assume that all rounding errors have magnitude less than 10^{-8}.
c) Use the Euler's method with the value of h found in (b) (after a slight adjustment if needed) to find an approximation of the solution to the initial value problem above.
d) With the value of h found in (b), compute the predicted error bound at $t = 2$ with the actual error. What can you conclude?

Question 13.4
Use Runge-Kutta method of order four to approximate the solution of the initial value problem

$$\begin{aligned} y' &= 1 + (t - y)^2 \quad , \quad 2 \le t \le 3 \\ y(2) &= 1 \end{aligned} \tag{13.8.1}$$

Use a step size of 0.1 and compute the absolute and relative error at each step. You obviously need to find the analytic solution of the initial value problem (13.8.1) to compute the errors.

Question 13.5
Use Runge-Kutta method of order four to approximate the solution of

$$\begin{aligned} y' &= \sin(t-y) \quad , \quad 2 \le t \le 3 \\ y(2) &= 1 \end{aligned}$$

Use different step sizes.

Question 13.6
Consider the Runge-Kutta Method

$$\mathbf{w}_{i+1} = \mathbf{w}_i + h\left(\left(\frac{1}{2}+\beta\right)K_1 + \left(\frac{1}{2}-\beta\right)K_2\right) \ ,$$

where

$$K_1 = f(\mathbf{w}_i + \beta h K_1) \quad \text{and} \quad K_2 = f(\mathbf{w}_i + hK_1 + \beta h K_2) \ .$$

a) Give the Butcher array associated to this Runge-Kutta Method.
b) Use Theorem 13.4.32 to determine the order of the method? Show that it does not depend on β.
c) Find the first term of the local truncation error.
d) If we use the Runge-Kutta Method above to find an approximation of the solution of the initial value problem

$$\begin{aligned} \mathbf{y}'(t) &= A\mathbf{y}(t) \quad , \quad a \le t \le b \\ \mathbf{y}(a) &= \mathbf{y}_0 \end{aligned} \tag{13.8.2}$$

where A is a $n \times n$ matrix, can we chose β to get a method of order greater than the ordre found in (b)?
e) Show that the Runge-Kutta Method above applied to (13.8.2) yields a finite difference equation of the form

$$\mathbf{w}_{i+1} = R(hA, \beta)\mathbf{w}_i \ ,$$

where R is a rational function.

Question 13.7
Compute the local truncation error of the trapezoidal method

$$w_{i+1} = w_i + \frac{h}{2}\left(f(t_i, w_i) + f(t_{i+1}, w_{i+1})\right) \quad , \quad 0 < i < N \ .$$

What is the order of this method? Is the method consistent?

Question 13.8
Show that two successive steps of the trapezoidal method

$$w_{i+1} = w_i + \frac{h}{2}\left(f(t_i, w_i) + f(t_{i+1}, w_{i+1})\right)$$

(t_i to t_{i+1} followed by t_{i+1} to t_{i+2}) yields one step of a 3-stage Runge-Kutta Method (t_i to t_{i+2}). Give the Butcher array of the Runge-Kutta Method. What is the order of this method?

Question 13.9
Find the 2-stage Runge-Kutta Method given by the collocation method associated to the nodes $\alpha_1 = 1/3$ and $\alpha_2 = 2/3$. Find the order of this method? What is the maximal order of a 2-stage Runge-Kutta Method that we can get with the collocation method?

Question 13.10
Consider an implicit Runge-Kutta Method given by the Butcher array

$$\begin{array}{c|cccc} \alpha_1 & \beta_{1,1} & \beta_{1,2} & \ldots & \beta_{1,k} \\ \alpha_2 & \beta_{2,1} & \beta_{2,2} & \ldots & \beta_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_k & \beta_{k,1} & \beta_{k,2} & \ldots & \beta_{k,k} \\ \hline & \gamma_1 & \gamma_2 & \ldots & \gamma_k \end{array}$$

Assume that the order of the method is greater or equal to p and $\alpha_i \neq \alpha_j$ for $i \neq j$. Show that this method is given by the collocation method if and only if

$$\sum_{j=1}^{k} \beta_{i,j}\alpha_j^{n-1} = \frac{\alpha_i^n}{n} \quad \text{and} \quad \sum_{j=1}^{k} \gamma_j \alpha_j^{n-1} = \frac{1}{n} \tag{13.8.3}$$

for $1 \leq i, n \leq k$.

Question 13.11
Consider the initial value problem

$$\begin{aligned} y'(t) &= \mu y(t) \quad , \quad t \geq 0 \\ y(0) &= y_0 \end{aligned}$$

Show that all semi-implicit Runge-Kutta Method applied to this initial value problem is of the form $w_i = (r(h\mu))^i w_0$, where $r(z)$ is a rational function whose denominator is a product of factor of degree one.

Question 13.12
Prove without using Theorem 13.6.43 that the Runge-Kutta Method given by the Butcher array

$$\begin{array}{c|cc} (3-\sqrt{3})/6 & 1/4 & (3-2\sqrt{3})/12 \\ (3+\sqrt{3})/6 & (3+2\sqrt{3})/12 & 1/4 \\ \hline & 1/2 & 1/2 \end{array}$$

is A-stable?

Question 13.13
Consider the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \leq t \leq t_f \\ y(0) &= y_0 \end{aligned} \tag{13.8.4}$$

An explicit method to approximate the solution of (13.8.4) is defined as follows. The approximation w_i of $y(t_i)$ is given by the solution of

$$w_{i+1} = w_i + \frac{h}{2}\left(3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\right) \ , \quad w_1 = y_1 \quad \text{and} \quad w_0 = y_0 \ .$$

a) Show that this method is of order 2 and that it is consistent.
b) Is this method strongly stable?
c) Is this method convergent?

Question 13.14
Consider the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(0) &= y_0 \end{aligned} \tag{13.8.5}$$

An implicit method to approximate the solution of (13.8.5) is defined as follows. The approximation w_i of $y(t_i)$ is given by the solution of

$$w_{i+1} = w_{i-1} + \frac{2}{3} h \left(f(t_{i+1}, w_{i+1}) + f(t_i, w_i) + f(t_{i-1}, w_{i-1}) \right) \ , \quad w_1 = y_1 \quad \text{and} \quad w_0 = y_0 \ .$$

a) Show that this method is of order 2 and that it is consistent.
b) Does this method satisfy the root condition?
c) Is this method convergent?

Question 13.15
Consider the initial value problem

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad , \quad t_0 \le t \le t_f \\ y(0) &= y_0 \end{aligned} \tag{13.8.6}$$

The Simpson's method is an implicit method to approximate the solution of (13.8.6) defined as follows. The approximation w_i of $y(t_i)$ is given by the solution of

$$w_{i+1} = w_{i-1} + \frac{h}{3} \left(f(t_{i+1}, w_{i+1}) + 4 f(t_i, w_i) + f(t_{i-1}, w_{i-1}) \right) \ , \quad w_1 = y_1 \quad \text{and} \quad w_0 = y_0 \ .$$

a) Apply the Simpson rule of integration to

$$\int_{t_{i-1}}^{t_{i+1}} f(t, y(t)) \, \mathrm{d}t$$

to derive the Simpson method above and its local truncation error.
b) Show that Simpson's method is consistent.
c) Does the Simpson's method satisfy the root condition?
d) Is the Simpson's method convergent?

Question 13.16
To approximate the solution of the initial value problem

$$\begin{aligned} y'(t) &= t \quad , \quad 0 \le t \le 5 \\ y(0) &= 0 \end{aligned} \tag{13.8.7}$$

we use the multistep method

$$w_{i+1} = w_i + \frac{h}{12} \left(4 f(t_{i+1}, w_{i+1}) + 9 f(t_i, w_i) - f(t_{i-1}, w_{i-1}) \right) \ , \tag{13.8.8}$$

$$w_1 = y_1 \quad \text{and} \quad w_0 = y_0 \ .$$

for $1 \leq i < N$ with $t_i = ih$ and $h = 5/N$

a) Write the difference equation associated to (13.8.7).
b) Find the general solution of the difference equation in (a). A particular solution of this equation is of the form $w_i = Ai^2 + Bi$ for some constants A and B.
c) Find the solution of the difference equation in (a) with $w_0 = 0$.
d) Does the solution that you have found in (c) converge to the solution of (13.8.7) as $h \to 0$?
e) Does this multistep method satisfy the root condition?
f) Is this multistep method consistent?

Question 13.17
Use the technique presented in Section 13.5.3 to answer the following questions.

a) Construct a multistep method of order at least 2 of the form

$$\begin{aligned} w_{i+1} &= w_{i-2} + \sum_{j=-1}^{m} b_j f(t_{i-j}, w_{i-j}) \quad , \quad m \leq i < N \\ w_i &= y_i \quad , \quad 0 \leq i \leq m \end{aligned}$$

b) Construct a multistep method of order at least 3 of the form

$$\begin{aligned} w_{i+1} &= w_{i-2} + \sum_{j=-1}^{m} b_j f(t_{i-j}, w_{i-j}) \quad , \quad m \leq i < N \\ w_i &= y_i \quad , \quad 0 \leq i \leq m \end{aligned}$$

Question 13.18
Consider the multistep method

$$\begin{aligned} w_{i+1} &= \sum_{j=0}^{m} a_j w_{i-j} + h \sum_{j=-1}^{m} b_j f(t_{i-j}, w_{i-j}) \quad , \quad m \leq i < N \\ w_i &= y_i \quad , \quad 0 \leq i \leq m \end{aligned}$$

Let

$$p(w) = 1 - \sum_{j=0}^{m} a_j w^j \quad \text{and} \quad q(w) = \sum_{j=-1}^{m} b_j w^j \ .$$

Moreover, let $p_1(w) = p(w)$, $p_{k+1}(w) = 1 - w p_k'(w)$ for $k > 0$, $q_1(w) = q(w)$ and $q_{k+1}(w) = -w q_k'(w)$ for $k > 0$. Show that the method is of order r if and only if $p_1(1) = 0$, $p_{k+1}(1) = k q_k(1)$ for $1 \leq k \leq r$, and $p_{r+2}(1) \neq (p+1) q_{r+1}(1)$.

Question 13.19
a) Find the multistep method of the form

$$w_{i+1} = a_0 w_i + a_1 w_{i-1} + h \left(b_0 f(t_i, w_i) + b_1 f(t_{i-1}, w_{i-1}) \right)$$

of highest order.
b) What is the order of the method?
c) Is this method A-stable?

Question 13.20
If the multistep method

$$w_{i+1} = \sum_{j=0}^{m} a_j w_{i-j} + \sum_{j=-1}^{m} b_j f(t_{i-j}, w_{i-j})$$

is convergent, shows that 0 is on the boundary of the region of absolute stability for this method.

Chapter 14

Boundary Value Problems for Ordinary Differential Equations

The content of this chapter is based in great part on [22].

14.1 Introduction

Example 14.1.1
A simple example of a **boundary value problem** is given by the second order differential equation

$$\begin{aligned} &y''(t) + y(t) = 0 \quad , \quad 0 \le t \le \frac{\pi}{2} \\ &y(0) = 0 \quad \text{and} \quad y(\pi/2) = 1 \end{aligned} \tag{14.1.1}$$

The conditions that y must satisfy at 0 and $\pi/2$ are the **boundary conditions**. The general solution of (14.1.1) is $y(t) = a\cos(t) + b\sin(t)$ where a and b are constants. $y(0) = 0$ implies that $a = 0$ and $y(\pi/2) = 1$ implies that $b = 1$. The solution of the boundary value problem is therefore $y(t) = \sin(t)$. ♣

We have to be prudent when solving boundary value problems because there may not exist a solution.

Example 14.1.2
The boundary value problem

$$\begin{aligned} &y''(t) + y(t) = 0 \quad , \quad 0 \le t \le \pi \\ &y(0) = 0 \quad \text{and} \quad y(\pi) = 1 \end{aligned}$$

does not have a solution as can be seen by trying to satisfy the boundary conditions with $y(t) = a\cos(t) + b\sin(t)$. ♣

14.2 Shooting Methods

This section is based on Keller's lectures [22].

Consider the boundary value problem

$$\begin{aligned} &y'' = f(t, y, y') \quad , \quad a \le t \le b \\ &y(a) = \alpha \quad \text{and} \quad y(b) = \beta \end{aligned} \tag{14.2.1}$$

Assuming that this problem has a solution (which is not always true even for nice boundary value problems), a possible approach to solve this problem is to use our knowledge of initial value problems. We have seen several analytical and numerical methods to solve initial value problems. We solve the initial value problem

$$\begin{aligned} &y'' = f(t, y, y') \quad , \quad a \le t \le b \\ &y(a) = \alpha \quad \text{and} \quad y'(a) = x \end{aligned} \tag{14.2.2}$$

to find a solution $y(t) = y(t, x)$ for $a \le t \le b$. Then we find x_b such that $y(b, x_b) - \beta = 0$ is satisfied to get the solution $y(t) = y(t, x_b)$ of (14.2.1).

When no analytical solution of (14.2.2) is available, numerical solutions of the initial value problem have to be found to approximate $y(b)$. This means that a value of x has to be chosen and a numerical solution of (14.2.2) has to be found to be able to compute $y(b) = y(b, x)$. If $y(b) \neq \beta$, then another value of x has to be chosen and another numerical solution of (14.2.2) has to be found to get a new value of $y(b) = y(b, x)$. This has to be repeated until we find x_b such that $y(b) = y(b, x_b)$ is closed enough to β to meet the required accuracy. This approach bears some resemblance to shooting where one tries to adjust the initial velocity to reach the target.

We present a more general approach of the shooting method than the one usually found in textbooks. Solving (14.2.2) using the numerical methods that we have presented requires rewriting (14.2.2) has a system of first order differential equations. Moreover, the boundary conditions may be more complex than the simple ones that we used given above. Our approach will take all that into consideration.

14.2.1 Shooting Method for Linear Boundary Value Problems

Let $\mathrm{GL}(n)$ be the group of $n \times n$ matrices with real entries. We consider the **boundary value problem**

$$\begin{aligned} P(\mathbf{y}(t)) \equiv \mathbf{y}'(t) - A(t)\mathbf{y}(t) = f(t) \quad &, \quad a \le t \le b \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{y}_c \end{aligned} \tag{14.2.3}$$

where $\mathbf{y} : [a, b] \to \mathbb{R}^n$, $A : [a, b] \to \mathrm{GL}(n)$ and $f : [a, b] \to \mathbb{R}^n$ are sufficiently differentiable functions, and $B_a, B_b \in \mathrm{GL}(n)$.

To solve this problem, we proceed as follows:

Algorithm 14.2.1 (Shooting Method)

1. We solve the initial value problems
$$\begin{aligned} P(\mathbf{y}_0(t)) &= f(t) \quad , \quad a \leq t \leq b \\ \mathbf{y}_0(a) &= \mathbf{y}_c \end{aligned} \tag{14.2.4}$$
and
$$\begin{aligned} P(\mathbf{y}_j(t)) &= \mathbf{0} \quad , \quad a \leq t \leq b \\ \mathbf{y}_j(a) &= \mathbf{e}_j \end{aligned} \tag{14.2.5}$$
for $j = 1, 2, \ldots, n$. Any other vector than $\mathbf{y}_c$ would have been acceptable.

2. The general solution $\mathbf{y}_g : [a,b] \to \mathbb{R}^n$ of the differential equation in (14.2.3) is of the form
$$\mathbf{y}_g(t) = \mathbf{y}_0(t) + \sum_{j=1}^{n} d_j \mathbf{y}_j(t) = \mathbf{y}_0(t) + Y(t)\mathbf{d} \ ,$$
where $Y(t) = \begin{pmatrix} \mathbf{y}_1(t) & \mathbf{y}_2(t) & \ldots & \mathbf{y}_n(t) \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} d_1 & d_2 & \ldots & d_n \end{pmatrix}^\top \in \mathbb{R}^n$.

3. $\mathbf{y}_g$ will be the solution of the boundary value problem (14.2.3) if there exists $\mathbf{d} \in \mathbb{R}^n$ such that
$$\mathbf{y}_c - B_a \mathbf{y}_0(a) - B_b \mathbf{y}_0(b) = Q\mathbf{d} \ , \tag{14.2.6}$$
where $Q = B_a + B_b Y(b)$. With this value of $\mathbf{d}$, we have that $B_a \mathbf{y}_g(a) + B_b \mathbf{y}_g(b) = \mathbf{y}_c$.

Example 14.2.2 (Example 14.1.1 continued)
The boundary value problem of Example 14.1.1 can be restated in the format (14.2.3) with $a = 0$, $b = \pi/2$, $\mathbf{y}(t) = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$, $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, $f(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $B_a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $B_b = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$ and $\mathbf{y}_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Since the general solution of $P(\mathbf{y}(t)) = \mathbf{y}'(t) - A(t)\mathbf{y}(t) = \mathbf{0}$ is
$$\mathbf{y}(t) = e^{tA}\mathbf{y}(0) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix} \mathbf{y}(0) \ ,$$
we get $\mathbf{y}_0(t) = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$, $\mathbf{y}_1(t) = \begin{pmatrix} \cos(t) \\ -\sin(t) \end{pmatrix}$, $\mathbf{y}_2(t) = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$ and $Y(t) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$. Thus
$$\mathbf{y}_g(t) = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix} + \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$
and $Q = \mathrm{Id}$. Since $\mathbf{y}_c - B_a \mathbf{y}_0(a) - B_b \mathbf{y}_0(b) = \mathbf{0}$ and Q is non-singular, the only solution of (14.2.6) is $\mathbf{d} = \mathbf{0}$. We find the solution $\mathbf{y}_g(t) = \mathbf{y}_0(t)$ as expected. ♣

Theorem 14.2.3

If $A : [a,b] \to \mathrm{GL}(n)$ and $f : [a,b] \to \mathbb{R}^n$ are functions of class C^r, then the boundary value problem (14.2.3) has a unique solution of class C^{r+1} if and only if Q defined in step 3 above is invertible.

Proof.
The existence (and uniqueness) of the solutions to (14.2.4) and (14.2.5) is proved in a basic course on ordinary differential equations. It is proved in basic linear algebra that (14.2.6) has a unique solution if and only if Q is invertible. ∎

The following code implement the shooting method for our linear boundary value problem.(14.2.3).

Code 14.2.4 (Shooting Method)

To approximate the solution of the boundary value problem $y' - A(t) = f(f)$ with $B_a y(a) + B_b y(b) = y_c$ for $a \leq t \leq b$. The classical fourth order Runga-Kutta is use to solve initial value problems in the algorithm. For N given, the step size is $h = (b-a)/N$ and $t_i = a + ih$ for $0 \leq i \leq N$.
Input: The vector valued function $f : [a,b] \to \mathbb{R}^n$ (f in the code below).
The $n \times n$ matrix valued function A defined on $[a,b]$ (A in the code below).
The $n \times n$ matrix B_a (Ba in the code below).
The $n \times n$ matrix B_b (Bb in the code below).
The (column) vector y_c (yc in the code below).
The number N of equal partitions of $[a,b]$.
The endpoints a and b of the interval of integration $[a,b]$
Output: The $n \times (N+1)$ matrix ww that contains the approximation $w_{k,i}$ of $y_{k,i} = y_k(t_i)$ for $1 \leq k \leq n$ and $0 \leq i \leq N$, and the vector tt that contains t_i for $0 \leq i \leq N$.

```
function [tt,ww] = shooting(f,A,Ba,Bb,yc,N,a,b)
  funct1 = @(t,y) A(t)*y + f(t);
  funct2 = @(t,y) A(t)*y;
  h = (b-a)/N;
  n = length(yc);

  %   Solve the initial value problem
  %   y'(t) - A(t) y(t) = f(t)  with  y(a) = y_c
  [tt,ww1] = rgkt4(funct1,h,N,a,yc);

  %   Solve the initial value problems
  %   y'(t) - A(t) y(t) = 0  with  y(a) = e_i
  %   for 1 <= j <= n
  WW = repmat(NaN,n,N+1,n);
  for j=1:1:n
    yj = zeros(n,1);
    yj(j) = 1;
```

```
    [tt,ww2] = rgkt4(funct2,h,N,a,yj);
    WW(:,:,j) = ww2;
  end

  %  Solve yc -B_a y_0(a) - B_b y_0(b) = Q d
  %  with Q = B_a + B_b Y(b)
  Y = yc - Ba*ww1(:,1) -Bb*ww1(:,N+1);
  Q = Ba + Bb*squeeze(WW(:,N+1,:));
  d = linsolve(Q,Y);

  ww2 = repmat(0,n,N+1);
  for j=1:1:n
    ww2 =  ww2 + d(j)*squeeze(WW(:,:,j));
  end
  ww = ww1 + ww2;
end
```

We could have used one of the "ode" solvers in Matlab. However, we have chosen to use our own implementation of Runage-Kutta in $\mathbb{R}^n$. It is basically the same code that we have presented in Code 13.4.9. We give it below.

Code 14.2.5 (Runge-Kutta of Order Four)

To approximate the solution of the initial value problem

$$\begin{aligned} \mathbf{y}'(t) &= f(t, \mathbf{y}(t)) \quad , \quad t \geq t_0 \\ \mathbf{y}(0) &= \mathbf{y}_0 \end{aligned}$$

Input: The function $f(t, \mathbf{y})$ (funct in the code below).
The step-size h.
The number of steps N.
The initial time t_0 (t0 in the code below) and the initial conditions $\mathbf{y}_0$ (y0 in the code below) at t_0.
Output: The approximations $\mathbf{w}_i$ (ww(:,i+1) in the code below) of $\mathbf{y}(t_i)$ at t_i (tt(i+1) in the code below).

```
function [tt,ww] = rgkt4(funct,h,N,t0,y0)
  tt(1) = t0;
  ww(:,1) = y0;
  h2 = h/2;
  for j=1:N
    tt(j+1) = tt(1)+j*h;
    k1 = h*funct(tt(j),ww(:,j));
    k2 = h*funct(tt(j)+h2,ww(:,j)+k1/2);
    k3 = h*funct(tt(j)+h2,ww(:,j)+k2/2);
    k4 = h*funct(tt(j+1),ww(:,j)+k3);
    ww(:,j+1) = ww(:,j) + (k1+2*(k2+k3)+k4)/6;
```

```
  end
end
```

Example 14.2.6
Consider the following boundary value problem

$$y_1'(t) = y_2(t) \quad , \quad y_2'(t) = 4y_1(t) - 3e^t$$

with

$$y_1(0) = 1 \quad , \quad y_2(1) = e$$

This problem can be restated as $\mathbf{y}'(t) = A(t)\mathbf{y}(t) + f(t)$ with $B_a\mathbf{y}(0) + B_b\mathbf{y}(1) = \mathbf{y}_c$, where

$$\mathbf{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \ , \ A(t) = \begin{pmatrix} 0 & 1 \\ 4 & 0 \end{pmatrix} \ , \ f(t) = \begin{pmatrix} 0 \\ -3e^t \end{pmatrix} \ , \ B_a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \ , \ B_b = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{y}_c = \begin{pmatrix} 1 \\ e \end{pmatrix} \ .$$

If we use the code above with $N = 25$, we find the following approximations of the solution.

i	t_i	$w_{1,i}$	$w_{2,i}$
0	0	1.0	1.0
1	0.04	1.0408108	1.0408111
2	0.08	1.0832871	1.0832873
3	0.12	1.1274969	1.1274971
4	0.16	1.1735109	1.1735111
5	0.20	1.2214028	1.2214030
6	0.24	1.2712492	1.2712494
7	0.28	1.3231299	1.3231300
8	0.32	1.3771278	1.3771280
⋮	⋮	⋮	⋮

i	t_i	$w_{1,i}$	$w_{2,i}$
17	0.68	1.9738778	1.9738778
18	0.72	2.0544333	2.0544332
19	0.76	2.1382763	2.1382762
20	0.80	2.2255410	2.2255409
21	0.84	2.3163670	2.3163669
22	0.88	2.4108997	2.4108996
23	0.92	2.5092904	2.5092903
24	0.96	2.6116965	2.6116963
25	1.00	2.7182818	2.7182816

where $w_{1,i} \approx y_{1,i} = y_1(t_i)$ and $w_{2,i} \approx y_{2,i} = y_2(t_i)$ for all i. All the approximations have at least 6-digit accuracy. The exact solution is $\mathbf{y}(t) = \begin{pmatrix} e^t \\ e^t \end{pmatrix}$.

For the sake of completeness, here is the code used to call the shooting method.

Code 14.2.7

```
format long
f = @(t) [ 0 ; -3*exp(t) ];
A = @(t) [ 0 1 ; 4 0 ];
Ba = [ 1 0 ; 0 0 ];
Bb = [ 0 0 ; 1 0 ];
yc = [ 1 ; exp(1) ];
N = 25;
[t,w] = shooting(f,A,Ba,Bb,yc,N,0,1)
```

♣

Example 14.2.8
The following example was used in [9] to test the shooting method and the parallel shooting method that we will see shortly.

Consider the boundary value problem

$$y^{(4)}(t) - 401y''(t) + 400y(t) + 1 - 200t^2 = 0$$

with

$$y(0) = 1 \ , y'(0) = 1 \ , \ y(1) = \frac{3}{2} + \sinh(1) \text{ and } y'(1) = 1 + \cosh(1) \ .$$

This problem can be rewritten as $\mathbf{y}'(t) = A(t)\mathbf{y}(t) + f(t)$ with $B_a\mathbf{y}(0) + B_b\mathbf{y}(1) = \mathbf{y}_c$, where

$$\mathbf{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \end{pmatrix} , \ A(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -400 & 0 & 401 & 0 \end{pmatrix} , \ f(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 + 200t^2 \end{pmatrix} , \ B_a = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ,$$

$$B_b = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \text{ and } \mathbf{y}_c = \begin{pmatrix} 1 \\ 1 \\ 3/2 + \sinh(1) \\ 1 + \cosh(1) \end{pmatrix} .$$

If we use the code above with $N = 25$, we find the following approximations of the solution $y(t) = y_1(t)$.

i	t_i	$w_{1,i}$	i	t_i	$w_{1,i}$	i	t_i	$w_{1,i}$
0	0	1.0000000	9	0.36	1.4326265	18	0.72	2.0430405
1	0.04	1.0408107	10	0.40	1.4907523	19	0.76	2.1241049
2	0.08	1.0832854	11	0.44	1.5511354	20	0.80	2.2081060
3	0.12	1.1274882	12	0.48	1.6138455	21	0.84	2.2951282
4	0.16	1.1734835	13	0.52	1.6789536	22	0.88	2.3852584
5	0.20	1.2213360	14	0.56	1.7465317	23	0.92	2.4785857
6	0.24	1.2711106	15	0.60	1.8166536	24	0.96	2.5752018
7	0.28	1.3228730	16	0.64	1.8893942	25	1.00	2.6752012
8	0.32	1.3766894	17	0.68	1.9648304			

where $w_{1,i} \approx y_i = y(t_i)$ for all i because $y_1(t) = y(t)$ for all t. All the approximations have at least 7-digit accuracy. The exact solution is $\mathbf{y}(t) = 1 + t^2/2 + \sinh(t)$.

It is interesting to note how much more accurate our results are then those in [9]. The difference is not in the algorithm used because we both use a simple shooting method. The difference is in the fact that they use single precision arithmetic (common for the main frame computers at that time) while we use double precision arithmetic.

Moreover, the matrix $A(t)$ has eigenvalues ± 1 and ± 20. So, we have a stiff ordinary differential equation. However, the solution that we are approximating is the one associated to the eigenvalues ± 1. Fortunately, the fact that we use double precision arithmetic and that we have imposed a condition at $t = 1$ eliminate the part associated to the eigenvalue 20. This explain why the shooting method could give us a reasonably good solution. The reader is invited to numerically solve the initial value problem $\mathbf{y}'(t) = A(t)\mathbf{y}(t) + f(t)$ with $\mathbf{y}(0) = \mathbf{y}_c$ using the classical fourth order Runge-Kutta methods. The solution obtained is not even remotely closed to $\mathbf{y}(t) = 1 + t^2/2 + \sinh(t)$. The part of the general solution associated to the eigenvalue 20 dominates. ♣

14.2.2 Numerical Aspect of the Shooting Method

Let $\{t_i\}_{i=0}^N$ be a partition of $[a,b]$. More precisely, $t_0 = a$, $t_N = b$, $t_{i+1} = t_i + h_i$ with $h_i > 0$ for $0 \leq i < N$ and $h = \max_{0\leq i<N} h_i \leq \theta \min_{0\leq i<N} h_i$ for some constant θ.

Remark 14.2.9 (Important)
The constant $\theta \geq 1$ is an absolute constant for the entire chapter. In particular, θ does not vary with the choice of partitions. If $\theta = 1$, we have that $h_i = h$ for all i. The step size is constant. ♠

We assume that a stable and convergent numerical method is used to numerically solve (14.2.4) and (14.2.5). Let $\mathbf{w}_{j,i}$ be the numerical approximation of $\mathbf{y}_j(t_i)$ given by the numerical method for $0 \leq i \leq N$ and $0 \leq j \leq n$. Suppose that

$$\|\mathbf{w}_{j,i} - \mathbf{y}_j(t_i)\| = O(h^p) \quad , \quad 0 \leq i \leq N \ , \tag{14.2.7}$$

for $0 \leq j \leq n$.

If we set $Q_T = B_a + B_b W_N$, where $W_i = \begin{pmatrix} \mathbf{w}_{1,i} & \mathbf{w}_{2,i} & \dots & \mathbf{w}_{n,i} \end{pmatrix} \in \mathrm{GL}(n)$ for $0 \leq i \leq N$, then (14.2.6) becomes

$$\mathbf{y}_c - B_a\mathbf{w}_{0,0} - B_b\mathbf{w}_{0,N} = Q_T\mathbf{d}_T \tag{14.2.8}$$

for some $\mathbf{d}_T \in \mathbb{R}^n$. Since $\|Q - Q_T\| = O(h^p)$ from (14.2.7) and Q is invertible, we get from Banach Lemma that Q_T is invertible for h small enough. To be more precise, if h is small enough to have $\|Q - Q_T\| < 1/\|Q^{-1}\|$, then Q_T is invertible. Thus (14.2.8) has a unique solution. Note that (14.2.8) is a system of linear equations which is not necessarily easy to solve.

An approximation of the solution $\mathbf{y}_g$ of the boundary value problem (14.2.3) is given by

$$\mathbf{w}_i = \mathbf{w}_{0,i} + W_i\mathbf{d}_T \quad , \quad 0 \leq i \leq N \ .$$

We now show that the approximation of the solution given by the shooting method also satisfies

$$\|\mathbf{w}_i - \mathbf{y}(t_i)\| = O(h^p) \ .$$

Since $\|Q - Q_T\| = O(h^p)$, we have that $\|Q_T\|$ is uniformly bounded for h small enough. To prove this, choose h_0 such that $\|Q - Q_T\| < 1/(2\|Q^{-1}\|)$ for $h < h_0$ and note that

$$\|Q_T^{-1}\| - \|Q^{-1}\| \leq \|Q_T^{-1} - Q^{-1}\| = \|Q_T^{-1}(Q - Q_T)Q^{-1}\| \leq \|Q_T^{-1}\|\,\|Q - Q_T\|\,\|Q^{-1}\|$$

implies

$$\|Q_T^{-1}\| \leq \frac{\|Q^{-1}\|}{1 - \|Q - Q_T\|\,\|Q^{-1}\|} \leq 2\|Q^{-1}\|$$

for $h < h_0$. It follows that

$$\|Q^{-1} - Q_T^{-1}\| = \|Q^{-1}(Q_T - Q)Q_T^{-1}\| \leq \underbrace{\|Q^{-1}\|}_{\text{bounded}} \underbrace{\|Q_T - Q\|}_{=O(h^p)} \underbrace{\|Q_T^{-1}\|}_{\text{bounded}} = O(h^p) \ . \tag{14.2.9}$$

Since

$$\begin{aligned}\|\mathbf{y}(t_i) - \mathbf{w}_i\| &= \|\mathbf{y}_0(t_i) + Y(t_i)\mathbf{d} - \mathbf{w}_{0,i} - W_i\mathbf{d}_T\| \\ &\leq \underbrace{\|\mathbf{y}_0(t_i) - \mathbf{w}_{0,i}\|}_{=O(h^p)\text{ by (14.2.7)}} + \underbrace{\|Y(t_i) - W_i\|}_{=O(h^p)\text{ by (14.2.7)}} \|\mathbf{d}\| + \underbrace{\|W_i\|}_{\text{bounded}} \|\mathbf{d} - \mathbf{d}_T\|\end{aligned}$$

and

$$\begin{aligned}\|\mathbf{d} - \mathbf{d}_T\| &= \|Q^{-1}\left(\mathbf{y}_c - B_a\mathbf{y}_0(a) - B_b\mathbf{y}_0(b)\right) - Q_T^{-1}\left(\mathbf{y}_c - B_a\mathbf{w}_{0,0} - B_b\mathbf{w}_{0,N}\right)\| \\ &\leq \|Q^{-1}\|\left(\|B_a\| \underbrace{\|\mathbf{y}_0(a) - \mathbf{w}_{0,0}\|}_{=O(h^p)\text{ by (14.2.7)}} + \|B_b\| \underbrace{\|\mathbf{y}_0(b) - \mathbf{w}_{0,N}\|}_{=O(h^p)\text{ by (14.2.7)}}\right) \\ &\quad + \underbrace{\|Q^{-1} - Q_T^{-1}\|}_{=O(h^p)\text{ by (14.2.9)}} \underbrace{\|\mathbf{y}_c - B_a\mathbf{w}_{0,0} - B_b\mathbf{w}_{0,N}\|}_{\text{bounded}} \ ,\end{aligned}$$

we get

$$\|\mathbf{y}(t_i) - \mathbf{w}_i\| = O(h^p) \ .$$

This shows that the order of the shooting method is determined by the order of the numerical methods used to solve the initial value problems (14.2.4) and (14.2.5).

Obviously, in the previous discussion, we have ignored rounding errors.

14.2.3 Separated and Partially Separated Boundary Conditions

For the boundary value problem (14.2.3), we generally assumed that

$$\operatorname{rank}\begin{pmatrix} B_a & B_b \end{pmatrix} = n \tag{14.2.10}$$

to get n linearly independent boundary conditions. This is a necessary condition to get Q invertible.

Suppose that $\operatorname{rank} B_b = q < n$. There exists an $n \times n$ invertible matrix R_b (built from operations on the rows of B_b) such that

$$R_b B_b = \begin{pmatrix} 0 \\ B_b^{[b]} \end{pmatrix} \ ,$$

where $B_b^{[b]}$ is a $q \times n$ matrix of rank q.

We define

$$\begin{pmatrix} \mathbf{y}_c^{[a]} \\ \mathbf{y}_c^{[b]} \end{pmatrix} = R_b \mathbf{y}_c \ ,$$

where $\mathbf{y}_c^{[b]} \in \mathbb{R}^q$ and $\mathbf{y}_c^{[a]} \in \mathbb{R}^{n-q}$, and

$$\begin{pmatrix} B_a^{[a]} \\ B_a^{[b]} \end{pmatrix} = R_b B_a \ ,$$

where $B_a^{[a]}$ is an $(n-q)\times n$ matrix and $B_a^{[b]}$ is a $q\times n$ matrix. We have applied the operations on the rows of B_b above to the rows of B_a. Note that $B_a^{[a]}$ is of rank $n-q$ because of (14.2.10).

The boundary conditions in (14.2.3) can then be rewritten as

$$\begin{aligned} B_a^{[a]}\mathbf{y}(a) &= \mathbf{y}_c^{[a]} \\ B_a^{[b]}\mathbf{y}(a) + B_b^{[b]}\mathbf{y}(b) &= \mathbf{y}_c^{[b]} \end{aligned} \tag{14.2.11}$$

The boundary conditions are **separable** if $B_a^{[b]} = 0$ and **partially separable** if $B_a^{[b]} \neq 0$.

We now explain how to solve the boundary value problem (14.2.3). Let D_a be a $q \times n$ matrix such that

$$M_a = \begin{pmatrix} B_a^{[a]} \\ D_a \end{pmatrix}$$

is invertible. This is possible because $B_a^{[a]}$ is of rank $n-q$ and thus the rows of $B_a^{[a]}$ are linearly independent. Let F_a be the $n \times q$ matrix defined by

$$M_a^{-1} = \begin{pmatrix} E_a & F_a \end{pmatrix} \ .$$

Note that F_a is of rank q because M_a is invertible.

To solve this problem, we may proceed as follows:

Algorithm 14.2.10

1. We solve the initial value problems

$$\begin{aligned} P(\mathbf{y}_0(t)) &= f(t) \quad , \quad a \le t \le b \\ B_a^{[a]}\mathbf{y}_0(a) &= \mathbf{y}_c^{[a]} \end{aligned} \tag{14.2.12}$$

   and

$$\begin{aligned} P(\mathbf{y}_j(t)) &= \mathbf{0} \quad , \quad a \le t \le b \\ \mathbf{y}_j(a) &= F_a\mathbf{e}_j \end{aligned}$$

   for $\mathbf{e}_j \in \mathbb{R}^q$ and $j = 1, 2, \ldots, q$. Since $q < n$, there are less initial value problems to solve.

2. The general solution $\mathbf{y}_g : [a,b] \to \mathbb{R}^n$ of the differential equation in (14.2.3) is of the form

$$\mathbf{y}_g(t) = \mathbf{y}_0(t) + \sum_{j=1}^{q} d_j\mathbf{y}_j(t) = \mathbf{y}_0(t) + V(t)\mathbf{d} \ ,$$

where $V(t) = \begin{pmatrix} \mathbf{y}_1(t) & \mathbf{y}_2(t) & \ldots & \mathbf{y}_q(t) \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} d_1 & d_2 & \ldots & d_q \end{pmatrix}^\top \in \mathbb{R}^q$.

3. $\mathbf{y}_g$ above will be a solution of the boundary value problem (14.2.3) if there exists $\mathbf{d} \in \mathbb{R}^q$ such that
$$B_a^{[b]}\mathbf{y}_g(a) + B_b^{[b]}\mathbf{y}_g(b) = \mathbf{y}_c^{[b]} \ . \tag{14.2.13}$$

Note that
$$B_a^{[a]}\mathbf{y}_g(a) = B_a^{[a]}\mathbf{y}_0(a) + B_a^{[a]}F_a\mathbf{d} = B_a^{[a]}\mathbf{y}_0(a) = \mathbf{y}_c^{[a]} \ .$$
The second equality in the previous equation comes from $B_a^{[a]}F_a = 0$ because $M_a\, M_a^{-1} = \mathrm{Id}$ and the last equality comes from (14.2.12).

Using the general form of $\mathbf{y}_g$ and $V(a) = F_a\,\mathrm{Id}_q$, we get from (14.2.13) that
$$\left(B_a^{[b]}F_a + B_b^{[b]}V(b)\right)\mathbf{d} = \mathbf{y}_c^{[b]} - B_a^{[b]}\mathbf{y}_0(a) - B_b^{[b]}\mathbf{y}_0(b) \ . \tag{14.2.14}$$
Since $q < n$, we have a smaller system of linear equations to solve than in the general shooting method.

Since $V(t) = Y(t)F_a$, where Y is the fundamental solution given in (14.2.5), the equation in (14.2.14) can be rewritten
$$Q_b\mathbf{d} = \mathbf{y}_c^{[b]} - B_a^{[b]}\mathbf{y}_0(a) - B_b^{[b]}\mathbf{y}_0(b) \ ,$$
where $Q_b = \left(B_a^{[b]} + B_b^{[b]}Y(b)\right)F_a$ because $V(b) = Y(b)F_a$. The matrix Q_b is an invertible $q\times q$ matrix if and only if $Q = \tilde{B}_a + \tilde{B}_bY(b)$ with $\tilde{B}_a = \begin{pmatrix} B_a^{[a]} \\ B_a^{[b]} \end{pmatrix}$ and $\tilde{B}_b = \begin{pmatrix} 0 \\ B_b^{[b]} \end{pmatrix}$ is invertible. To prove the last sentence, it suffices to note that
$$QM_a^{-1} = \begin{pmatrix} B_a^{[a]} \\ B_a^{[b]} + B_b^{[b]}Y(b) \end{pmatrix}\begin{pmatrix} E_a & F_a \end{pmatrix} = \begin{pmatrix} Id & 0 \\ * & Q_b \end{pmatrix}$$
because $M_aM_a^{-1} = \mathrm{Id}$.

Remark 14.2.11
Similarly, if $\operatorname{rank} B_a = p < n$, we can rewrite the boundary conditions in (14.2.3) as
$$\begin{aligned} B_a^{[a]}\mathbf{y}(a) + B_b^{[a]}\mathbf{y}(b) &= \mathbf{y}_c^{[a]} \\ B_b^{[b]}\mathbf{y}(b) &= \mathbf{y}_c^{[b]} \end{aligned}$$
where $B_a^{[a]}$ is a $p\times n$ matrix of rank p, $B_b^{[b]}$ is a $(n-p)\times n$ matrix of rank $n-p$, and $B_b^{[a]}$ is a $p\times n$ matrix.

Obviously, there is also the alternative to reorder the coordinates of $\mathbf{y}$ to reduce this case to the previous case. ♠

14.2.4 Parallel Shooting for Linear Boundary Value Problems

The parallel shooting method that we present in this section and the procedure presented in the next section to determine the F_i and $\mathbf{y}_{c,i}$ used in the parallel shooting method are based on [9, 22].

A potential serious issue with the simple shooting method is that the solutions $\mathbf{y}_j(t)$ given by (14.2.5) may become more and more dependent as t increases; namely, the matrix $Y(t) = \begin{pmatrix} \mathbf{y}_1(t) & \mathbf{y}_2(t) & \ldots & \mathbf{y}_n(t) \end{pmatrix}$ may become more and more singular, and so ill-conditioned, as t increases. In particular, $Y(b)$ could be ill-conditioned. Therefore, solving (14.2.6) may lead to serious numerical errors.

The **first step to address the issue above** is to integrate on shorter interval of time instead of the full interval $[a,b]$. It is crucial to do so if the solution is rapidly increasing in some regions of the interval $[a,b]$. The basic idea is to apply the shooting method on each interval $[t_{i-1}, t_i]$.

We consider the boundary value problem (14.2.3). As usual, we let $\{t_i\}_{i=0}^N$ be a partition of $[a,b]$ with $t_0 = a$, $t_N = b$, $t_{i+1} = t_i + h_i$ with $h_i > 0$ for $0 \leq i < N$ and $h = \max_{0\leq i<N} h_i \leq \theta \min_{0\leq i<N} h_i$ for some constant θ.

On each subinterval $[t_i, t_{i+1}]$, we solve the initial value problems

$$\begin{aligned} P(\mathbf{y}_{i,0}(t)) &= f(t) \quad , \quad t_i \leq t \leq t_{i+1} \\ \mathbf{y}_{i,0}(t_i) &= \mathbf{y}_{c,i} \end{aligned} \tag{14.2.15}$$

and

$$\begin{aligned} P(\mathbf{y}_{i,j}(t)) &= \mathbf{0} \quad , \quad t_i \leq t \leq t_{i+1} \\ \mathbf{y}_{i,j}(t_i) &= F_i \mathbf{e}_j \end{aligned} \tag{14.2.16}$$

for the canonical vectors $\mathbf{e}_j \in \mathbb{R}^n$ and $1 \leq j \leq n$. The matrices F_i of rank n and the vector $\mathbf{y}_{c,i}$ will be defined in Section 14.2.5 below.

We look for a solution $\mathbf{y}_g : [a,b] \to \mathbb{R}^n$ of the boundary value problem (14.2.3) defined on each interval $[t_i, t_{i+1}]$ by

$$\mathbf{y}_g(t) = \mathbf{y}_i(t) = \mathbf{y}_{i,0}(t) + V_i(t)\mathbf{d}_i \quad , \quad t_i \leq t \leq t_{i+1} \ ,$$

where $V_i(t) = \begin{pmatrix} \mathbf{y}_{i,1}(t) & \mathbf{y}_{i,2}(t) & \ldots & \mathbf{y}_{i,n}(t) \end{pmatrix}$ for $t_i \leq t \leq t_{i+1}$ and $\mathbf{d}_i \in \mathbb{R}^n$.

To get a continuous solution $\mathbf{y}$ at the points t_i for $0 < i < N$, we impose the condition

$$\mathbf{y}_i(t_i) = \mathbf{y}_{i-1}(t_i) \quad , \quad 1 < i < N \ .$$

Namely,

$$\mathbf{y}_{c,i} + F_i \mathbf{d}_i = \mathbf{y}_{i-1,0}(t_i) + V_{i-1}(t_i)\mathbf{d}_{i-1} \quad , \quad 1 < i < N \ . \tag{14.2.17}$$

Moreover, the boundary condition in (14.2.3) gives

$$B_a\left(\mathbf{y}_{c,0} + F_0 \mathbf{d}_0\right) + B_b\left(\mathbf{y}_{N-1,0}(b) + V_{N-1}(b)\mathbf{d}_{N-1}\right) = \mathbf{y}_c \ . \tag{14.2.18}$$

We can combine (14.2.17) and (14.2.18) to get the system $A_S\mathbf{d} = B_S$, where

$$A_S = \begin{pmatrix} B_aF_0 & 0 & 0 & \dots & 0 & B_bV_{N-1}(b) \\ -V_0(t_1) & F_1 & 0 & \dots & 0 & 0 \\ 0 & -V_1(t_2) & F_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -V_{N-2}(t_{N-1}) & F_{N-1} \end{pmatrix}, \tag{14.2.19}$$

$$\mathbf{d} = \begin{pmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N-1} \end{pmatrix} \quad \text{and} \quad B_S = \begin{pmatrix} \mathbf{y}_c - B_a\mathbf{y}_{c,0} - B_b\mathbf{y}_{N-1.0}(b) \\ \mathbf{y}_{0,0}(t_1) - \mathbf{y}_{c,1} \\ \vdots \\ \mathbf{y}_{N-2,0}(t_{N-1}) - \mathbf{y}_{c,N-1} \end{pmatrix}. \tag{14.2.20}$$

Remark 14.2.12

1. If the F_i for $0 \le i < N$ are invertible, the parallel shooting method is equivalent to the simple shooting method. In fact, we have

$$A_S = \begin{pmatrix} Q_0 & Q_1 & Q_2 & \dots & Q_{N-1} \\ 0 & Id & 0 & \dots & 0 \\ 0 & 0 & Id & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & Id \end{pmatrix} \begin{pmatrix} F_0 & 0 & 0 & \dots & 0 \\ -V_0(t_1) & F_1 & 0 & \dots & 0 \\ 0 & -V_1(t_2) & F_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -V_{N-2}(t_{N-1}) & F_{N-1} \end{pmatrix},$$

   where

$$Q_i = \begin{cases} B_bV_i(b)F_i^{-1} & \text{for} \quad i = N-1 \\ Q_{i+1}V_i(t_{i+1})F_i^{-1} & \text{for} \quad i = N-2, N-3, \dots, 1 \\ B_a + Q_{i+1}V_i(t_{i+1})F_i^{-1} & \text{for} \quad i = 0 \end{cases}$$

   Since $V_i(t) = Y_i(t)F_i$ for $t_i \le t \le t_{i+1}$, where $Y_i(t)$ is the fundamental solution of $P(\mathbf{y}(t)) = \mathbf{0}$ on $[t_i, t_{i+1}]$, in particular $Y_i(t_i) = \mathrm{Id}$, we have that

$$\begin{aligned} Q_0 &= B_a + B_bV_{N-1}(t_N)F_{N-1}^{-1}V_{N-2}(t_{N-1})F_{N-2}^{-1}\dots V_0(t_1)F_0^{-1} \\ &= B_a + B_bY_{N-1}(b)Y_{N-2}(t_{N-1})\dots Y_0(t_1) \\ &= B_a + B_bY(b) = Q\ , \end{aligned}$$

   where Y is the fundamental solution of $P(\mathbf{y}(t)) = \mathbf{0}$ on $[a,b]$. To get the second to last equality in the previous equation, we have use the uniqueness of solutions for initial value problems.

   Thus A_S is invertible if and only if Q is invertible.

2. Note that the decomposition of A_S above is an LU decomposition of A_S that may be used (with care) to solve $A_S\mathbf{d} = B_S$.

3. If we have separated or partially separated end-conditions, then we may assume that the row operations (i.e. R_b in Section 14.2.3) have been performed to get $\mathbf{y}_c = \begin{pmatrix} \mathbf{y}_c^{[a]} \\ \mathbf{y}_c^{[b]} \end{pmatrix}$,

$B_a = \begin{pmatrix} B_a^{[a]} \\ B_a^{[b]} \end{pmatrix}$ and $B_b = \begin{pmatrix} 0 \\ B_b^{[b]} \end{pmatrix}$. The matrices F_i in (14.2.16) are now $n \times q$ matrices of rank q. We can then repeat the reasoning in this section to get a system of linear equations $A_S\mathbf{d} = B_S$ with A_S, B_S and $\mathbf{d}$ defined by

$$A_S = \begin{pmatrix} B_a^{[a]}F_0 & 0 & 0 & \dots & 0 & 0 \\ -V_0(t_1) & F_1 & 0 & \dots & 0 & 0 \\ 0 & -V_1(t_2) & F_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -V_{N-2}(t_{N-1}) & F_{N-1} \\ B_a^{[b]}F_0 & 0 & 0 & \dots & 0 & B_b^{[b]}V_{N-1}(b) \end{pmatrix} , \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N-1} \end{pmatrix}$$

and

$$B_S = \begin{pmatrix} \mathbf{y}_c^{[a]} - B_a^{[a]}\mathbf{y}_{c,0} \\ \mathbf{y}_0(t_1) - \mathbf{y}_{c,1} \\ \vdots \\ \mathbf{y}_{N-2}(t_{N-1}) - \mathbf{y}_{c,N-2} \\ \mathbf{y}_c^{[b]} - B_a^{[b]}\mathbf{y}_{c,0} - B_b^{[b]}\mathbf{y}_{N-1,0}(b) \end{pmatrix} .$$

The first $n-q$ rows of A_S above are the first $n-q$ rows of A_S in (14.2.19), and the last q rows of A_S above are the q rows from the $(n-q+1)^{th}$ to the n^{th} row inclusively of A_S in (14.2.19). We have a similar statement for B_S above and B_S in (14.2.20).

Matrices like A_S (i.e. lower or almost lower block triangular) often appear in the numerically solution of systems of partial differential equations. This type of matrices has been extensively studied in numerical analysis.

♠

14.2.5 The Choice of F_i and $\mathbf{y}_{c,i}$

The simple shooting method is given by $F_i = \text{Id}$ for $0 \leq i < N$, $\mathbf{y}_{c,i} = \mathbf{0}$ for $1 \leq i < N$, and $\mathbf{y}_{c,0} = \mathbf{y}_c$, where $\mathbf{y}_c$ is defined in (14.2.3). But this is not the one interesting us.

The **second step to address the issue** mentioned at the beginning of Section 14.2.4 is to replace (14.2.17) and (14.2.18) by equations that no longer involve the $V_{i-1}(t_i)$ but only matrices F_{i-1} that have orthonormal columns.

The **last step to address the issue** mentioned at the beginning of Section 14.2.4 is to ensure that $\mathbf{y}_{c,i}$ is not in the range of $V_{i-1}(t_i)$ in order to provided a transition from the integration on the interval $[t_{i-1}, t_i]$ to the interval $[t_i, t_{i+1}]$. It is traditional to take $\mathbf{y}_{c,i}$ in the orthogonal complement of the column span of $V_{i-1}(t_i)$.

We implement these two steps below.

From now on, we assume that the boundary conditions are partially separated.

1. Let $F_0 = F_a$ and $\mathbf{y}_{c,0} = \mathbf{y}_0(a)$, where F_a and $\mathbf{y}_0$ are defined in Section 14.2.3.

2. Suppose that we have determined $V_{i-1}(t_i)$. The q columns of F_i are the q columns of $V_{i-1}(t_i)$ after they have been orthonormalized. Therefore, $F_i = V_{i-1}(t_i)P_{i-1}$ for some $q \times q$ upper-triangular matrix P_{i-1}. Usually, the Gram-Schmidt method seen in linear algebra is used for this purpose.

3. $\mathbf{y}_{c,i}$ is the projection of $\mathbf{y}_{i-1,0}(t_i)$ on the orthogonal complement of the column span of F_i. Therefore, $\mathbf{y}_{c,i} = (\mathrm{Id} - F_i F_i^\top)\mathbf{y}_{i-1,0}(t_i)$.

We now explain how to compute the $\mathbf{d}_i$ for $0 \leq i < N$ that are used to define the $\mathbf{y}_i$ of the parallel shooting method.

We rewrite (14.2.17) as

$$\mathbf{y}_{c,i} + F_i\mathbf{d}_i = \mathbf{y}_{i-1,0}(t_i) + V_{i-1}(t_i)\mathbf{d}_{i-1}$$

for $i = N-1,\ N-2,\ \ldots,\ 1$. If we interpret this equation for the shooting method with partially separated boundary conditions, we get

$$\left(\mathrm{Id} - V_{i-1}(t_i)P_{i-1}F_i^\top\right)\mathbf{y}_{i-1,0}(t_i) + V_{i-1}(t_i)P_{i-1}\mathbf{d}_i = \mathbf{y}_{i-1,0}(t_i) + V_{i-1}(t_i)\mathbf{d}_{i-1}$$

for $i = N-1,\ N-2,\ \ldots,\ 1$. This equation can be simplified to yield

$$V_{i-1}(t_i)P_{i-1}\left(\mathbf{d}_i - F_i^\top \mathbf{y}_{i-1,0}(t_i)\right) = V_{i-1}(t_i)\mathbf{d}_{i-1}$$

for $i = N-1,\ N-2,\ \ldots,\ 1$.

Since $V_{i-1}(t_i)$ has rank q because the columns of V_{i-1} are q linearly independent solutions of $P(\mathbf{y}(t)) = \mathbf{0}$ [1], we can simplify the previous equation to get

$$\mathbf{d}_{i-1} = P_{i-1}\left(\mathbf{d}_i - F_i^\top \mathbf{y}_{i-1,0}(t_i)\right) \in \mathbb{R}^q \tag{14.2.21}$$

for $j = N,\ N-1,\ N-2,\ \ldots,\ 1$. Note that we have extended (14.2.21) to $i = N$. the extra F_N and $\mathbf{y}_{c,N}$ are also given by the previous 3-step procedure.

We first show that the condition

$$B_a^{[a]}F_0\mathbf{d}_0 = \mathbf{y}_c^{[a]} - B_a^{[a]}\mathbf{y}_{c,0} \tag{14.2.22}$$

from the first $n-q$ rows of (14.2.18) (Item 3 of Remark 14.2.12) is automatically satisfied by construction. We have that (14.2.22) is equivalent to $B_a^{[a]}\mathbf{y}_{c,0} = \mathbf{y}_c^{[a]}$ because $F_0 = F_a$ and $B_a^{[a]}F_a = 0$ since $MM^{-1} = \mathrm{Id}$. Moreover, it follows from (14.2.12) that $B_a^{[a]}\mathbf{y}_{c,0} = \mathbf{y}_c^{[a]}$ is satisfied because we assume that $\mathbf{y}_{c,0} = \mathbf{y}_0(a)$.

We now consider the condition

$$B_a^{[b]}F_0\mathbf{d}_0 + B_b^{[b]}V_{N-1}(b)\mathbf{d}_{N-1} = \mathbf{y}_c^{[b]} - B_a^{[b]}\mathbf{y}_{c,0} - B_b^{[b]}\mathbf{y}_{N-1,0}(b)$$

[1] We use the uniqueness of solutions for ordinary differential equations to conclude that if $\{y_{i,j}(t)\}_{j=1}^q$ is a linearly independent set of solutions, then $\{y_{i,j}(s)\}_{j=1}^q$ is a linear independent set of vectors in $\mathbb{R}^n$ for every $s \in [t_i, t_{i+1}]$.

from the last q rows of (14.2.18) (Item 3 of Remark 14.2.12). Using (14.2.21) for $i = N$, we get

$$
\begin{aligned}
&B_a^{[b]} F_0 \mathbf{d}_0 + B_b^{[b]} \underbrace{V_{N-1}(b) P_{N-1}}_{=F_N} (\mathbf{d}_N - F_N^\top \mathbf{y}_{N-1,0}(t_N)) = \mathbf{y}_c^{[b]} - B_a^{[b]} \mathbf{y}_{c,0} - B_b^{[b]} \mathbf{y}_{N-1,0}(b) \\
&\Rightarrow B_a^{[b]} F_0 \mathbf{d}_0 + B_b^{[b]} F_N \mathbf{d}_N - B_b^{[b]} F_N F_N^\top \mathbf{y}_{N-1,0}(t_N) = \mathbf{y}_c^{[b]} - B_a^{[b]} \mathbf{y}_{c,0} - B_b^{[b]} \mathbf{y}_{N-1,0}(b) \\
&\Rightarrow B_a^{[b]} F_0 \mathbf{d}_0 + B_b^{[b]} F_N \mathbf{d}_N = \mathbf{y}_c^{[b]} - B_a^{[b]} \mathbf{y}_{c,0} - B_b^{[b]} \underbrace{(\mathrm{Id} - F_N F_N^\top)\, \mathbf{y}_{N-1,0}(t_N)}_{=\mathbf{y}_{c,N}} \\
&\Rightarrow B_a^{[b]} F_0 \mathbf{d}_0 + B_b^{[b]} F_N \mathbf{d}_N = \mathbf{y}_c^{[b]} - B_a^{[b]} \mathbf{y}_{c,0} - B_b^{[b]} \mathbf{y}_{c,N} \ .
\end{aligned}
$$

Therefore, the vectors $\mathbf{d}_i$ are given by the system of linear equations $A_S \mathbf{d} = B_S$, where

$$
A_S = \begin{pmatrix}
-\mathrm{Id} & P_0 & 0 & \dots & 0 & 0 \\
0 & -\mathrm{Id} & P_1 & \dots & 0 & 0 \\
0 & 0 & -\mathrm{Id} & \dots & 0 & 0 \\
 & & & \ddots & & \\
0 & 0 & 0 & \dots & -\mathrm{Id} & P_{N-1} \\
B_a^{[b]} F_0 & 0 & 0 & \dots & 0 & B_b^{[b]} F_N
\end{pmatrix} \quad , \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_N \end{pmatrix}
$$

and

$$
B_S = \begin{pmatrix}
P_0 F_1^\top \mathbf{y}_{0,0}(t_1) \\
\vdots \\
P_{N-1} F_N^\top \mathbf{y}_{N-1,0}(t_N) \\
\mathbf{y}_c^{[b]} - B_a^{[b]} \mathbf{y}_{c,0} - B_b^{[b]} \mathbf{y}_{c,N}
\end{pmatrix} .
$$

Remark 14.2.13

The method proposed above could be improved. Since orthonormalization is costly in computation time and prone to numerical round off errors, it may be preferable to perform it only when the matrix $V_{i-1}(t_i)$ is "nearly singular" only. A method to determine if orthonormalization is required is to test for the size of the angles between the column vectors of $V_{i-1}(t_i)$. If the angles get "too close" to 0, orthonormalization should be used. Recall that the cosine of the angle between two vectors $\mathbf{a}$ and $\mathbf{b}$ is determined by $\langle \mathbf{a}, \mathbf{a} \rangle / (\|\mathbf{a}\| \, \|\mathbf{b}\|)$. Unfortunately, even this method is kind of computer time intensive.

The method could also be improved by appropriately choosing the mesh points $\{t_i\}_{i=0}^N$ such that $t_{i+1} - t_i$ is "small" when the solution is "rapidly" increasing. We select the mesh points as i increases to ensure that $V_{i-1}(t_i)$ does not get too "large." ♠

Code 14.2.14 (Parallel Shooting Method for Linear Problems with Partially Separated End-conditions)

To approximate the solution of the boundary value problem $y' - A(t) = f(f)$ with $B_a y(a) + B_b y(b) = y_c$ for $a \leq t \leq b$. We consider the intervals $[t_i, t_{i+1}]$ for $0 \leq i < N$ with $t_i = a + iH$ and $H = (b-a)/N$. We use the classical fourth order Runge-Kutta on each interval $[t_i, t_{i+1}]$ with the step size $h = (t_i - t_{i+1})/M$ to solve initial value problems.

Let $t_{i,j} = t_i + jh$ for $0 \leq i < N$ and $0 \leq j \leq M$.
Input: The vector valued function $f : [a,b] \to \mathbb{R}^n$ (f in the code below).
The $n \times n$ matrix valued function A defined on $[a,b]$ (A in the code below).
The $(n-q) \times n$ matrix $B_a^{[a]}$ (Baa in the code below).
The $q \times n$ matrix $B_a^{[b]}$ (Bab in the code below).
The $q \times n$ matrix $B_b^{[b]}$ (Bbb in the code below).
The (column) vector $y_c \in R^n$ (yc in the code below).
The number N of partitions of $[a,b]$.
The number M of partitions of each $[t_i, t_{i+1}]$.
The endpoints a and b of the interval of integration $[a,b]$.
Output: The $n \times (MN+1)$ matrix ww that contains the approximations $w_{k,iM+j}$ of $y_k(t_{i,j})$ and the vector tt that contains $t_{iM+j} = t_{i,j}$ for $1 \leq k \leq n$, $0 \leq i < N$, and $0 \leq j < M$ if $i < N-1$ or $0 \leq j \leq M$ if $i = N-1$.

```
function [tt,ww] = par_shooting(f,A,Baa,Bab,Bbb,yc,M,N,a,b)
  funct1 = @(t,y) A(t)*y + f(t);
  funct2 = @(t,y) A(t)*y;
  n = length(yc);
  q = size(Bbb,1);
  nmq = n - q;
  ttt = repmat(NaN,M+1,N);
  WW1 = repmat(NaN,n,M+1,N);
  WWW = repmat(NaN,n,M+1,q,N);
  PPi = repmat(NaN,q,q,N+1);
  FFi = repmat(NaN,n,q,N+1);
  yci = repmat(NaN,n,N+1);

  % We choose the matrix D_a such that M_a is invertible
  Da = zeros(q,n);
  s = 1;
  for j=1:1:n
    v = zeros(1,n);
    v(j) = 1;
    MM = [Baa ; v];
    if ( rank(MM) > nmq )
      Da(s,:) = v;
      s = s +1;
    end
    if ( rank(Da) == q )
      break;
    end
  end

  % We find the matrix F_a
  Ma = [Baa ; Da];
  Mainv = inv(Ma);
```

```
Fa = Mainv(:,(q+1):n);

% We now compute the approximation of y_i(t_j) and
% V_i(t_j) for 0 <= i < N and 0 \leq j \leq M, and the
% F_i and R_i for 0 <= i <= N.
% Warning: the indices i and j in matlab are shifted by 1
%          because vectors start with the index 1.
H = (b-a)/N;
h = H/M;
ti = a;
FFi(:,:,1) = Fa;

% We solve M_a y_{c,0} = y_c instead of B_a^{[a]} y_{c,0} = y_c^{[a]}
% to ensure that there is only one solution for Matlab to find.
yci(:,1) = linsolve(Ma,yc);

for i=1:1:N+1
  %   Solve the initial value problem
  %   y'(t) - A(t) y(t) = f(t)  with  y_{i,0}(t_i) = y_{c,i}
  if ( i <= N )
    [t,ww1] = rgkt4(funct1,h,M,ti,yci(:,i));
    WW1(:,:,i) = ww1;
  end

  %   Solve the initial value problems
  %   y'(t) - A(t) y(t) = 0  with  y_{i,j}(t_i) = F_i e_j
  %   for 1 <= j <= q
  WW = repmat(NaN,n,M+1,q);
  for j=1:1:q
    yj = zeros(q,1);
    yj(j) = 1;
    y = FFi(:,:,i)*yj;
    [tt,ww2] = rgkt4(funct2,h,M,ti,y);
    WW(:,:,j) = ww2;
  end
  if ( i <= N )
    ttt(:,i) = tt;
    WWW(:,:,:,i) = WW;
  end

  % We choose F_{i+1} and Y_{c,i+1} for the next interval
  % The function par_QR is defined in the following code.
  % It is used to find F_i and P_{i-1}.
  Vi = squeeze(WW(:,M+1,:));
  [Fi,R] = par_QR(Vi);
  FFi(:,:,i+1) = Fi;
```

```
    PPi(:,:,i) = inv(R);
    if ( i <= N )
      yci(:,i+1) = (eye(n) - Fi*Fi')*ww1(:,M+1);
    end
    ti = ti + H;
  end

  % We now find the vector d_i for 0 <= i < N .
  qN = q*N;
  qNp1 = q*(N+1);
  As = zeros(qNp1,qNp1);
  Bs = zeros(qNp1,1);
  for i=1:1:N
    qi = q*i;
    qim1 = qi - q + 1;
    As(qim1:qi, qim1:qi) = - eye(q);
    As(qim1:qi, qi+1:qi+q) = squeeze(PPi(:,:,i));
    Bs(qim1:qi,1) = PPi(:,:,i)*(FFi(:,:,i+1)')*WW1(:,M+1,i);
  end
  As(qN+1:qNp1, 1:q) = Bab*FFi(:,:,1);
  As(qN+1:qNp1, qN+1:qNp1) = Bbb*FFi(:,:,N+1);
  Bs(qN+1:qNp1,1) = yc(nmq+1:n,1) - Bab*yci(:,1) - Bbb*yci(:,N+1);

  D = linsolve(As,Bs);

  % The results
  ww = [];
  tt = [];
  for i=1:1:N
    w = zeros(n,M+1);
    for j=1:1:q
      w =  w + D((i-1)*q+j)*squeeze(WWW(:,:,j,i));
    end
    if ( i < N )
      ww = [ww, WW1(:,1:M,i) + w(:,1:M)];
      tt = [tt, ttt(1:M,i)'];
    else
      ww = [ww, WW1(:,:,i) + w];
      tt = [tt, ttt(:,i)'];
    end
  end
end
```

Finding the QR decomposition of a matrix is normally seen in a first course on linear algebra. We also presented it in Section 11.6.1i of Chapter 11.

Code 14.2.15 (QR Decomposition)

Find the QR decomposition of a matrix A: namely, $A = QR$ where the columns of Q are orthonormal and R is upper triangular. The columns of A must be linearly independent.
Input: The $n \times q$ matrix A.
Output: The matrices Q and R of the QR decomposition of A.

```
% [Q,R] = par_QR(A)
%
function [Q,R] = par_QR(A)
  n = size(A,1);
  q = size(A,2);
  Q = repmat(NaN,n,q);
  R = zeros(q);

  if ( rank(A) < q )
    return;
  end

  R(1,1) = norm(A(:,1));
  Q(:,1) = (1/R(1,1))*A(:,1);
  for i = 2:1:q
      v = A(:,i);
      for j = 1:1:i-1
          R(j,i) = Q(:,j)'*A(:,i);
          v = v - R(j,i)*Q(:,j);
      end
      R(i,i) = norm(v);
      Q(:,i) = (1/R(i,i))*v;
  end
end
```

We now revisit the two examples that we have considered with our code for the parallel shooting method.

Example 14.2.16 (Example 14.2.6 Continued)
If we use the Code 14.2.14 with $N = M = 10$, we get the following approximations of the

solution.

i	t_i	$w_{1,i}$	$w_{2,i}$	i	t_i	$w_{1,i}$	$w_{2,i}$
0	0	1	1	93	0.92	2.5092904	2.5092904
1	).01	1.0100502	1.0100502	93	0.93	2.5345092	2.5345092
2	0.02	1.0202013	1.0202013	94	0.94	2.5599814	2.5599814
3	0.03	1.0304545	1.0304545	95	0.95	2.5857097	2.5857097
4	0.04	1.0408108	1.0408108	96	0.96	2.6116965	2.6116965
5	0.05	1.0512711	1.0512711	97	0.97	2.6379445	2.6379445
6	0.06	1.0618365	1.0618365	98	0.98	2.6644562	2.6644562
7	0.07	1.0725082	1.0725082	99	0.99	2.6912345	2.6912345
8	0.08	1.0832871	1.0832871	100	1.00	2.7182818	2.7182818
$\vdots$	$\vdots$	$\vdots$	$\vdots$				

where $w_{1,i} \approx y_{1,i} = y_1(t_i)$ and $w_{2,i} \approx y_{2,i} = y_2(t_i)$ for all i. All the approximations have at least 8-digit accuracy.

For the sake of completeness, here is the code used to call the parallel shooting method.

Code 14.2.17

```
format long
f = @(t) [ 0 ; -3*exp(t) ];
A = @(t) [ 0 1 ; 4 0 ];
Baa = [ 1 0 ];
Bab = [ 0 0 ];
Bbb = [ 1 0 ];
yc = [ 1 ; exp(1) ];
N = 10;
M = 10;
[t,w] = par_shooting(f,A,Baa,Bab,Bbb,yc,M,N,0,1)
```

♣

Example 14.2.18 (Example 14.2.6 Continued)
If we use the Code 14.2.14 with $N = M = 10$, we get the following approximations of the solution.

i	t_i	$w_{1,i}$	i	t_i	$w_{1,i}$	i	t_i	$w_{1,i}$
0	0.00	1.00000000000	9	0.09	1.09417154921	92	0.92	2.47858567444
1	0.01	1.01005016666	10	0.10	1.10516675001	93	0.93	2.50242773364
2	0.02	1.02020133336	11	0.11	1.11627196757	94	0.94	2.52647679150
3	0.03	1.03045450020	12	0.12	1.12748820742	95	0.95	2.55073431794
4	0.04	1.04081066751	13	0.13	1.13881647619	96	0.96	2.57520179373
5	0.05	1.05127083593	14	0.14	1.15025778172	97	0.97	2.59988071063
6	0.06	1.06183600647	15	0.15	1.16181313314	98	0.98	2.62477257154
7	0.07	1.07250718067	16	0.16	1.17348354101	99	0.99	2.64987889067
8	0.08	1.08328536063	$\vdots$	$\vdots$	$\vdots$	100	1.00	2.67520119364

where $w_{1,i} \approx y_i = y(t_i)$ for all i. All the approximations have at least 10-digit accuracy. ♣

14.2.6 Shooting Method for Non-Linear Boundary Value Problems

We consider the boundary value problem

$$\begin{aligned} \mathbf{y}'(t) &= f(t, \mathbf{y}(t)) \quad , \quad a \le t \le b \\ g(\mathbf{y}(a), \mathbf{y}(b)) &= \mathbf{0} \end{aligned} \tag{14.2.23}$$

This problem can be reformulated as follows. Find $\mathbf{s} \in \mathbb{R}^n$ such that the solution $\mathbf{u}(t, \mathbf{s})$ of

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t}(t, \mathbf{s}) &= f(t, \mathbf{u}(t, \mathbf{s})) \quad , \quad a \le t \le b \\ \mathbf{u}(a, \mathbf{s}) &= \mathbf{s} \end{aligned} \tag{14.2.24}$$

satisfies

$$\phi(\mathbf{s}) = g(\mathbf{s}, \mathbf{u}(b, \mathbf{s})) = \mathbf{0} \ . \tag{14.2.25}$$

We have reduced the problem to finding the roots of $\phi(\mathbf{s})$. Hence, $\mathbf{u}(t, \mathbf{s})$ will be a solution of (14.2.23) if $\mathbf{s}$ is a root of $\phi(\mathbf{s})$.

The following theorem (assuming that g in (14.2.23) is also sufficiently differentiable) will ensure that the solution of (14.2.23), if there is one, is sufficiently differentiable. Moreover, the following theorem will also be used to justify the use of the Newton Method to find a root of $\phi(\mathbf{s})$.

Theorem 14.2.19

Suppose that $\mathbf{y}_g$ is a solution of (14.2.23) and that there exist two positive constants K and δ such that

$$\|f(t, \mathbf{v}) - f(t, \mathbf{w})\| \le K \|\mathbf{v} - \mathbf{w}\|$$

for all $(t, \mathbf{v}), (t, \mathbf{w}) \in T_\delta(\mathbf{y})$, where

$$T_\delta(\mathbf{y}) = \{(t, \mathbf{v}) \in \mathbb{R} \times \mathbb{R}^n : a \le t \le b \text{ and } \|\mathbf{y}(t) - \mathbf{v}\| \le \delta\} \ .$$

(Figure 14.1) If $\mathbf{s} \in \{\mathbf{s} \in \mathbb{R}^n : \|\mathbf{y}(a) - \mathbf{s}\| \le \delta e^{-K(b-a)}\}$, then there exists a unique solution $\mathbf{u}$ of (14.2.24). Moreover, if f is continuously differentiable on $T_\delta(\mathbf{y})$, then $U(t, \mathbf{s}) = \mathrm{D}_{\mathbf{s}} \mathbf{u}(t, \mathbf{s})$ exists for $a \le t \le b$ and is the fundamental solution of

$$U'(t) = \mathrm{D}_{\mathbf{y}} f(t, \mathbf{u}(t, \mathbf{s})) U(t) \quad , \quad a \le t \le b \ . \tag{14.2.26}$$

In particular, $U(a, \mathbf{s}) = \mathrm{Id}$.

Proof.
The domain $T_\delta(\mathbf{y})$ is sketched in Figure 14.1. These are classical results of ordinary differential equations. ∎

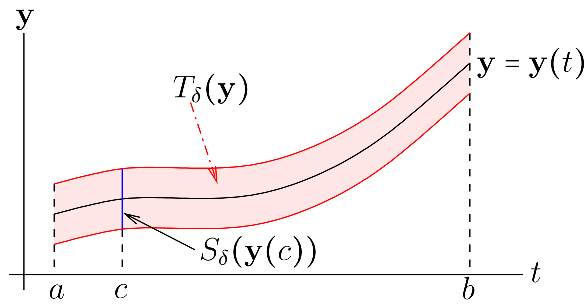


Figure 14.1: The domain $T_f(\mathbf{y})$ of Theorem 14.2.19 used to define conditions that will ensure solutions of an initial value problem. Note that $S_\delta(\mathbf{y}(c)) = \{\mathbf{v} : \|\mathbf{v} - \mathbf{y}(c)\| < \delta\}$.

The Newton Method applied to (14.2.25) is

$$Q(\mathbf{s}^{[j]})\left(\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\right) = -\phi(\mathbf{s}^{[j]}) \quad , \quad j \geq 0 \ , \tag{14.2.27}$$

where

$$Q(\mathbf{s}) = \mathrm{D}_{\mathbf{s}}\phi(\mathbf{s}) = \mathrm{D}_{\mathbf{y}_1} g(\mathbf{s}, \mathbf{u}(b,\mathbf{s})) + \mathrm{D}_{\mathbf{y}_2} g(\mathbf{s}, \mathbf{u}(b,\mathbf{s}))U(b,\mathbf{s})$$

for

$$\begin{aligned} g : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^n \\ (\mathbf{y}_1, \mathbf{y}_2) &\mapsto g(\mathbf{y}_1, \mathbf{y}_2) \end{aligned}$$

The following theorem justifies the use of the Newton Method to find a root of ϕ. It also gives a (not that useful) hint on how to choose $\mathbf{s}_0$.

Theorem 14.2.20

Suppose that ϕ has an isolate root $\mathbf{s}_*$ and that there exist $\rho_* > 0$, β and γ such that

$$\|Q^{-1}(\mathbf{s}_*)\| < \beta \ ,$$
$$\|Q(\mathbf{s}) - Q(\tilde{\mathbf{s}})\| \leq \gamma \|\mathbf{s} - \tilde{\mathbf{s}}\|$$

for all $\mathbf{s}, \tilde{\mathbf{s}} \in S_{\rho_*}(\mathbf{s}_*) = \{\mathbf{s} : \|\mathbf{s} - \mathbf{s}_*\| \leq \rho_*\}$, and $\rho_*\beta\gamma < \dfrac{2}{3}$.
Then, for all $\mathbf{s}^{[0]} \in S_{\rho_*}(\mathbf{s}_*)$, the sequence $\{\mathbf{s}^{[j]}\}_{j=0}^{\infty}$ given by the iterative method defined in (14.2.27) stays in $S_{\rho_*}(\mathbf{s}_*)$ and converge to $\mathbf{s}_*$. The convergence is quadratic; namely,

$$\|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| \leq \alpha \|\mathbf{s}^{[j]} - \mathbf{s}_*\|^2 \ , \tag{14.2.28}$$

where $\alpha = \dfrac{\beta\gamma}{2(1 - \rho_*\beta\gamma)}$.

Proof.
For all $\mathbf{s}$,

$$Q(\mathbf{s}) = Q(\mathbf{s}_*)\left(\mathrm{Id} - Q^{-1}(\mathbf{s}_*)\left(Q(\mathbf{s}_*) - Q(\mathbf{s})\right)\right) .$$

Since

$$\|Q^{-1}(\mathbf{s}_*)\left(Q(\mathbf{s}) - Q(\mathbf{s}_*)\right)\| \le \|Q^{-1}(\mathbf{s}_*)\|\,\|Q(\mathbf{s}_*) - Q(\mathbf{s})\| \le \beta\gamma\|\mathbf{s} - \mathbf{s}_*\| \le \beta\gamma\rho_* < \frac{2}{3} < 1$$

for all $\mathbf{s} \in S_{\rho_*}(\mathbf{s}_*)$, it follows from the Banach Lemma (Proposition 3.2.5 and Corollary 3.2.6) that $\mathrm{Id} - Q^{-1}(\mathbf{s}_*)\left(Q(\mathbf{s}_*) - Q(\mathbf{s})\right)$ is invertible for all $\mathbf{s} \in S_{\rho_*}(\mathbf{s}_*)$. Thus $Q(\mathbf{s})$ is invertible and

$$\|Q^{-1}(\mathbf{s})\| < \|\left(\mathrm{Id} - Q^{-1}(\mathbf{s}_*)\left(Q(\mathbf{s}_*) - Q(\mathbf{s})\right)\right)^{-1}\|\,\|Q^{-1}(\mathbf{s}_*)\| \le \frac{\beta}{1 - \rho_*\beta\gamma}$$

for all $\mathbf{s} \in S_{\rho_*}(\mathbf{s}_*)$.

We prove that $\mathbf{s}^{[j]} \in S_{\rho_*}(\mathbf{s}_*)$ for all j by induction. We have that $\mathbf{s}^{[0]} \in S_{\rho_*}(\mathbf{s}_*)$. We assume that $\mathbf{s}^{[j]} \in S_{\rho_*}(\mathbf{s}_*)$ and show that this implies that $\mathbf{s}^{[j+1]} \in S_{\rho_*}(\mathbf{s}_*)$. Since $\phi(\mathbf{s}_*) = \mathbf{0}$ and $Q^{-1}(\mathbf{s})$ exists for all $\mathbf{s} \in S_{\rho_*}(\mathbf{s}_*)$, we get from (14.2.27) that

$$\begin{aligned}\mathbf{s}^{[j+1]} - \mathbf{s}_* &= \left(\mathbf{s}^{[j]} - \mathbf{s}_*\right) + Q^{-1}(\mathbf{s}^{[j]})\left(\phi(\mathbf{s}_*) - \phi(\mathbf{s}^{[j]})\right) \\ &= Q^{-1}(\mathbf{s}^{[j]})\left(Q(\mathbf{s}^{[j]}) - Q(\mathbf{s}_*, \mathbf{s}^{[j]})\right)\left(\mathbf{s}^{[j]} - \mathbf{s}_*\right) ,\end{aligned}$$

where

$$Q(\mathbf{s}, \tilde{\mathbf{s}}) = \int_0^1 Q(\theta\mathbf{s} + (1-\theta)\tilde{\mathbf{s}})\,\mathrm{d}\theta .$$

Note that

$$Q(\theta\mathbf{s} + (1-\theta)\tilde{\mathbf{s}})\,(\mathbf{s} - \tilde{\mathbf{s}}) = \frac{\mathrm{d}}{\mathrm{d}\theta}\phi(\theta\mathbf{s} + (1-\theta)\tilde{\mathbf{s}})$$

since $Q(\mathbf{s}) = \mathrm{D}_{\mathbf{s}}\phi(\mathbf{s})$. Hence,

$$\begin{aligned}\|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| &\le \|Q^{-1}(\mathbf{s}^{[j]})\|\,\|Q(\mathbf{s}^{[j]}) - Q(\mathbf{s}_*, \mathbf{s}^{[j]})\|\,\|\mathbf{s}^{[j]} - \mathbf{s}_*\| \\ &\le \underbrace{\left(\frac{\beta}{1 - \rho_*\beta\gamma}\right)\frac{\gamma}{2}}_{=\alpha}\|\mathbf{s}^{[j]} - \mathbf{s}_*\|^2 ,\end{aligned} \tag{14.2.29}$$

where the last inequality comes from

$$\begin{aligned}\|Q(\mathbf{s}^{[j]}) - Q(\mathbf{s}_*, \mathbf{s}^{[j]})\| &= \left\|\int_0^1 \left(Q(\mathbf{s}^{[j]}) - Q(\theta\mathbf{s}_* + (1-\theta)\mathbf{s}^{[j]})\right)\mathrm{d}\theta\right\| \\ &\le \int_0^1 \left\|Q(\mathbf{s}^{[j]}) - Q(\theta\mathbf{s}_* + (1-\theta)\mathbf{s}^{[j]})\right\|\,\mathrm{d}\theta \\ &\le \gamma\int_0^1 \left\|\mathbf{s}^{[j]} - \theta\mathbf{s}_* - (1-\theta)\mathbf{s}^{[j]})\right\|\,\mathrm{d}\theta \\ &= \gamma\left\|-\mathbf{s}_* + \mathbf{s}^{[j]}\right\|\int_0^1 \theta\,\mathrm{d}\theta = \frac{\gamma}{2}\left\|-\mathbf{s}_* + \mathbf{s}^{[j]}\right\| .\end{aligned} \tag{14.2.30}$$

It follows from (14.2.29) that

$$\|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| \le \alpha \rho_*^2 < \rho_*$$

because $\alpha < 3\beta\gamma/2 < 1/\rho_*$ which is a consequence of $\rho_*\beta\gamma < 2/3$. Thus $\mathbf{s}^{[j]} \in S_{\rho_*}(\mathbf{s}_*)$ for all j by induction.

Since $S_{\rho_*}(\mathbf{s}_*)$ is complete, there is a subsequence of $\{\mathbf{s}^{[j]}\}_{j=0}^{\infty}$ that converges. However, we also have from (14.2.29) that

$$\|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| \le \underbrace{\left(\frac{\rho_*\beta\gamma}{2(1-\rho_*\beta\gamma)}\right)}_{<1} \|\mathbf{s}^{[j]} - \mathbf{s}_*\| \tag{14.2.31}$$

again because $\rho_*\beta\gamma < 2/3$. As we have done in the proof of the Fixed Point Theorem, Theorem 2.4.2, we can show that $\{\mathbf{s}^{[j]}\}_{j=0}^{\infty}$ converges to $\mathbf{s}_*$.

Finally, we also have from (14.2.29) that (14.2.28) is satisfied. ∎

Remark 14.2.21
To compute $\mathbf{s}^{[j+1]}$, we must solve (14.2.24) and compute the fundamental solution of (14.2.26), where $\mathbf{s}$ is replaced by $\mathbf{s}^{[j]}$. ♠

14.2.7 Error Analysis

Numerical computations are never exact. We now consider the effect of truncation (e.g. in numerical integration) on the Newton Method (14.2.27). To simplify the discussion, we assume that there is no round off error which, in practice, is not negligible. Because of truncation, instead of (14.2.27), we actually compute

$$Q(\tilde{\mathbf{s}}^{[j]})\left(\tilde{\mathbf{s}}^{[j+1]} - \tilde{\mathbf{s}}^{[j]}\right) = -\phi(\tilde{\mathbf{s}}^{[j]}) + \delta_{j+1}(h) \tag{14.2.32}$$

for $j \ge 0$, where h is the maximum step size of the partition of $[a,b]$ for the converging and stable numerical method used to solve the differential equation. We assume that, for all j, $\|\delta_j(h)\| \le Mh^p$ for some constant M and positive integer p.

Theorem 14.2.22

Suppose that the hypothesis of Theorem 14.2.20 are satisfied with ρ_* replaced by $\tilde{\rho} = \rho_* + \delta_*$ and $\rho_*\beta\gamma < 2/3$ replaced by $\tilde{\rho}\beta\gamma < 1/2$. Suppose that $\theta \in]0,1[$ satisfies

$$2\gamma M h^p \left(\frac{\beta\gamma}{1-2\tilde{\rho}\beta\gamma}\right)^2 \le \theta \tag{14.2.33}$$

and

$$\sigma \equiv \frac{1}{1+\sqrt{1-\theta}}\left(\frac{2\beta M h^p}{1-2\tilde{\rho}\beta\gamma}\right) \le \delta_* \ . \tag{14.2.34}$$

Then, if $\mathbf{s}^{[0]} \in S_{\rho_*}(\mathbf{s}_*)$ satisfies $\|\mathbf{s}^{[1]} - \mathbf{s}^{[0]}\| \leq \rho_*$, the sequences $\{\mathbf{s}^{[j]}\}_{j=0}^{\infty}$ of (14.2.27) and $\{\tilde{\mathbf{s}}^{[j]}\}_{j=0}^{\infty}$ of (14.2.32) with $\tilde{\mathbf{s}}^{[0]} = \mathbf{s}^{[0]}$ satisfy

$$\|\mathbf{s}^{[j]} - \tilde{\mathbf{s}}^{[j]}\| \leq \sigma \quad \text{and} \quad \|\tilde{\mathbf{s}}^{[j]} - \mathbf{s}_*\| \leq \frac{1}{\alpha}\left(\alpha\|\mathbf{s}^{[0]} - \mathbf{s}_*\|\right)^{2^j} + \sigma$$

for all $j \geq 0$, where α is defined in the statement of Theorem 14.2.20.

Proof.
As in Theorem 14.2.20, we can show that $Q(\mathbf{s})$ is invertible and

$$\|Q^{-1}(\mathbf{s})\| \leq \frac{\beta}{1 - \tilde{\rho}\beta\gamma}$$

for all $\mathbf{s} \in S_{\tilde{\rho}}(\mathbf{s}_*)$. Moreover, as in Theorem 14.2.20, we can show that $\mathbf{s}^{[j]} \in S_{\rho_*}(\mathbf{s}_*)$ and

$$\|\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\| \leq \rho_* \tag{14.2.35}$$

for $j \geq 0$ if $\mathbf{s}_0 \in S_{\rho_*}(\mathbf{s}_*)$ because $\rho_*\beta\gamma \leq \tilde{\rho}\beta\gamma < 1/2 < 2/3$. In particular, (14.2.35) follows from the hypothesis that $\|\mathbf{s}^{[1]} - \mathbf{s}^{[0]}\| \leq \rho_*$ and

$$\|\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\| \leq \lambda\|\mathbf{s}^{[j]} - \mathbf{s}^{[j-1]}\|$$

with $\lambda = \dfrac{\rho_*\beta\gamma}{2(1 - \rho_*\beta\gamma)} < 1$ that can be proved as (14.2.31) was proved.

Let $\mathbf{r}^{[j]} = \mathbf{s}^{[j]} - \tilde{\mathbf{s}}^{[j]}$. If we subtract (14.2.32) from (14.2.27), we get

$$\begin{aligned} Q(\tilde{\mathbf{s}}^{[j]})\,\mathbf{r}^{[j+1]} &= \left(Q(\tilde{\mathbf{s}}^{[j]}) - Q(\mathbf{s}^{[j]}, \tilde{\mathbf{s}}^{[j]})\right)\mathbf{r}^{[j]} \\ &\quad + \left(Q(\tilde{\mathbf{s}}^{[j]}) - Q(\mathbf{s}^{[j]})\right)\left(\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\right) - \Delta_j(h) \ . \end{aligned} \tag{14.2.36}$$

A) If $\tilde{\mathbf{s}}^{[j]} \in S_{\tilde{\rho}}(\mathbf{s}_*)$, we show that $\|\mathbf{r}^{[j]}\| < \sigma$ and $\tilde{\mathbf{s}}^{[j+1]} \in S_{\tilde{\rho}}(\mathbf{s}_*)$. From (14.2.36), we get

$$\begin{aligned} \|\mathbf{r}^{[j+1]}\| &\leq \|Q^{-1}(\tilde{\mathbf{s}}^{[j]})\|\Bigg(\underbrace{\|Q(\tilde{\mathbf{s}}^{[j]}) - Q(\mathbf{s}^{[j]}, \tilde{\mathbf{s}}^{[j]})\|}_{\leq(\gamma/2)\|\mathbf{r}^{[j]}\| \text{ as in } (14.2.30)} \|\mathbf{r}^{[j]}\| \\ &\qquad + \underbrace{\|Q(\tilde{\mathbf{s}}^{[j]}) - Q(\mathbf{s}^{[j]})\|}_{\substack{\leq\gamma\|\mathbf{r}^{[j]}\| \text{ by Hypothesis of} \\ \text{Theorem 14.2.20 with} \\ \rho_* \text{ replaced by } \tilde{\rho}}} \ \underbrace{\|\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\|}_{\leq\tilde{\rho} \text{ from } (14.2.35)} + \underbrace{\|\delta_{j+1}(h)\|}_{\leq Mh^p} \Bigg) \\ &= \frac{\beta}{1 - \tilde{\rho}\beta\gamma}\left(\frac{\gamma}{2}\|\mathbf{r}^{[j]}\|^2 + \gamma\tilde{\rho}\|\mathbf{r}^{[j]}\| - \frac{1 - \tilde{\rho}\beta\gamma}{\beta}\|\mathbf{r}^{[j]}\| + Mh^p\right) + \|\mathbf{r}^{[j]}\| \\ &= q(\|\mathbf{r}^{[j]}\|) + \|\mathbf{r}^{[j]}\| \ , \end{aligned} \tag{14.2.37}$$

where

$$q(z) = \frac{\beta}{1 - \tilde{\rho}\beta\gamma}\left(\frac{\gamma z^2}{2} - \left(\frac{1 - 2\tilde{\rho}\beta\gamma}{\beta}\right)z + Mh^p\right)$$

because

$$\gamma\tilde{\rho} - \frac{1-\tilde{\rho}\beta\gamma}{\beta} = \frac{2\tilde{\rho}\beta\gamma - 1}{\beta} \ .$$

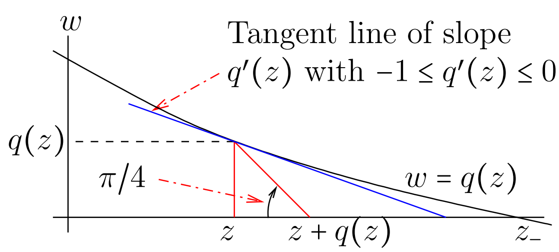
Since $((1-2\tilde{\rho}\beta\gamma)/\beta)^2 - 2\gamma M h^p > 0$ from (14.2.33), and $(1-2\tilde{\rho}\beta\gamma)/\beta > 0$ because $\tilde{\rho}\beta\gamma < 1/2$, the quadratic polynomial $q(z)$ has two positive roots $z_+ > z_-$. We show by induction that $\|\mathbf{r}^{[j]}\| \leq z_-$ for all j. The result is true for $j = 0$ because $\|\mathbf{r}^{[0]}\| = 0$. Suppose that $\|\mathbf{r}^{[j]}\| \leq z_-$. Since

$$q'(z) = \frac{\beta}{1-\tilde{\rho}\beta\gamma}\left(\gamma z - \left(\frac{1-2\tilde{\rho}\beta\gamma}{\beta}\right)\right) \ ,$$

we have that

$$-1 < -\frac{1-2\tilde{\rho}\beta\gamma}{1-\tilde{\rho}\beta\gamma} = q'(0) \leq q'(z) \leq q'(z_-) < 0$$

for $0 \leq z \leq z_-$ and q is concave up. We get the following figure



It follows that $z + q(z) < z_-$ for $0 \leq z \leq z_-$. Therefore $\|\mathbf{r}^{[j+1]}\| \leq z_-$ from (14.2.37). This completes the proof by induction.

We now show that $z_- \leq \sigma$. Since $z_+ z_- = 2Mh^p/\gamma$ and

$$z_+ = \frac{1-2\tilde{\rho}\beta\gamma}{\beta\gamma}\left(1 + \sqrt{1 - 2\gamma M h^p\left(\frac{\beta}{1-2\tilde{\rho}\beta\gamma}\right)^2}\right) \geq \frac{1-2\tilde{\rho}\beta\gamma}{\beta\gamma}\left(1+\sqrt{1-\theta}\right) > 0$$

because

$$2\gamma M h^p\left(\frac{\beta}{1-2\tilde{\rho}\beta\gamma}\right)^2 \leq \theta$$

according to (14.2.33), it follows that

$$z_- = \frac{2Mh^p}{\gamma z_+} \leq \frac{2Mh^p}{1+\sqrt{1-\theta}}\left(\frac{\beta}{1-2\tilde{\rho}\beta\gamma}\right) = \sigma \ .$$

Finally,

$$\begin{aligned}\|\tilde{\mathbf{s}}^{[j+1]} - \mathbf{s}_*\| &\leq \|\tilde{\mathbf{s}}^{[j+1]} - \mathbf{s}^{[j+1]}\| + \|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| = \|\mathbf{r}^{[j+1]}\| + \|\mathbf{s}^{[j+1]} - \mathbf{s}_*\| \\ &\leq \sigma + \rho_* \leq \delta_* + \rho_* = \tilde{\rho} \ ,\end{aligned}$$

where the last inequality comes from the hypothesis $\sigma < \delta_*$. Thus, $\tilde{\mathbf{s}}^{[j+1]} \in S_\rho(\mathbf{s}_*)$.

B) By induction, we get from (14.2.28) that

$$\|\mathbf{s}^{[j]} - \mathbf{s}_*\| \leq \alpha^{2^j-1}\|\mathbf{s}^{[0]} - \mathbf{s}_*\|^{2^j} \ ,$$

where $\alpha = \dfrac{\beta\gamma}{2(1-\rho_*\beta\gamma)}$. Hence,

$$\|\tilde{\mathbf{s}}^{[j]} - \mathbf{s}_*\| \leq \|\tilde{\mathbf{s}}^{[j]} - \mathbf{s}^{[j]}\| + \|\mathbf{s}^{[j]} - \mathbf{s}_*\| \leq \delta + \alpha^{2^j-1}\|\mathbf{s}^{[0]} - \mathbf{s}_*\|^{2^j} \leq \delta + \frac{1}{\alpha}\left(\alpha\|\mathbf{s}^{[0]} - \mathbf{s}_*\|\right)^{2^j} . \qquad \blacksquare$$

It follows from the previous theorem that the accuracy of the approximation $\tilde{\mathbf{s}}^{[j]}$ of $\mathbf{s}_*$ is limited by σ. Moreover, recall that $\alpha\|\mathbf{s}^{[0]} - \mathbf{s}_*\| \leq \alpha\rho_* < 2/3 < 1$. Thus, the error $\|\tilde{\mathbf{s}}^{[j]} - \mathbf{s}_*\|$ does not grow as j increases.

14.2.8 Parallel Shooting for Non-Linear Boundary Value Problems

As usual, let $\{t_i\}_{i=0}^N$ be a partition of $[a,b]$ such that $t_0 = a$, $t_N = b$, $t_{i+1} = t_i + h_i$ with $h_i > 0$ for $0 \leq i < N$ and $h = \max_{0\leq i<N} h_i \leq \theta \min_{0\leq i<N} h_i$ for some constant θ.

Parallel Shooting Method applied to the boundary value problem (14.2.23) can be summarized as follows. Solve the initial value problems

$$\begin{aligned} \mathbf{y}_i'(t,\mathbf{s}_i) &= f(t,\mathbf{y}_i(t,\mathbf{s}_i)) \quad , \quad t_i \leq t \leq t_{i+1} \\ \mathbf{y}_i(t_i,\mathbf{s}_i) &= \mathbf{s}_i \end{aligned}$$

for $0 \leq i < N$, where the initial conditions $\mathbf{s}_i$ are such that the function $\mathbf{y} : [a,b] \to \mathbb{R}^n$ defined by

$$\mathbf{y}_g(t) = \mathbf{y}_i(t,\mathbf{s}_i) \quad , \quad t_i \leq t \leq t_{i+1}$$

is a solution of the differential equation in (14.2.23) satisfying

$$\phi(\mathbf{s}) \equiv \begin{pmatrix} g(\mathbf{s}_0, \mathbf{y}_{N-1}(b,\mathbf{s}_{N-1})) \\ \mathbf{s}_1 - \mathbf{y}_0(t_1,\mathbf{s}_0) \\ \vdots \\ \mathbf{s}_{N-1} - \mathbf{y}_{N-2}(t_{N-1},\mathbf{s}_{N-2}) \end{pmatrix} = \mathbf{0} \ , \quad \text{where} \quad \mathbf{s} = \begin{pmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_{N-1} \end{pmatrix} .$$

The first n equations in $\phi(\mathbf{s}) = \mathbf{0}$ are the boundary conditions and the other equations are to ensure that we get a continuous (and differentiable) solution at t_i for $1 \leq i \leq N-1$.

The Parallel Shooting Method can be rewritten as a simple Shooting Method. Let

$$\mathbf{z}_i(\tau) = \mathbf{y}((t_{i-1} + \tau(t_i - t_{i-1})) \quad \text{for} \quad 1 \leq i \leq N \ ,$$

$$\mathbf{z}(\tau) = \begin{pmatrix} \mathbf{z}_i(\tau) \\ \mathbf{z}_2(\tau) \\ \vdots \\ \mathbf{z}_N(\tau) \end{pmatrix} \quad , \quad F(\tau,\mathbf{z}(\tau)) = \begin{pmatrix} (t_1-t_0)f(t_0+\tau(t_1-t_0),\mathbf{z}_1(\tau)) \\ (t_2-t_1)f(t_1+\tau(t_2-t_1),\mathbf{z}_2(\tau)) \\ \vdots \\ (t_N-t_{N-1})f(t_{N-1}+\tau(t_N-t_{N-1}),\mathbf{z}_N(\tau)) \end{pmatrix}$$

$$\text{and} \quad G(\mathbf{v},\mathbf{w}) = \begin{pmatrix} g(\mathbf{v}_1,\mathbf{w}_N) \\ \mathbf{v}_2 - \mathbf{w}_1 \\ \vdots \\ \mathbf{v}_N - \mathbf{w}_{N-1} \end{pmatrix}$$

for $0 \le \tau \le 1$ and $\mathbf{v}, \mathbf{w} \in (\mathbb{R}^n)^N \cong \mathbb{R}^{nN}$.

The boundary value problem (14.2.23) can be rewritten as

$$\begin{aligned} \mathbf{z}'(\tau) &= F(\tau, \mathbf{z}(\tau)) \quad , \quad 0 \le \tau \le 1 \\ G(\mathbf{z}(0), \mathbf{z}(1)) &= \mathbf{0} \end{aligned} \tag{14.2.38}$$

We get the Shooting Method

$$\begin{aligned} \mathbf{u}'(\tau, \mathbf{s}) &= F(\tau, \mathbf{u}(\tau, \mathbf{s}) \quad , \quad 0 \le \tau \le 1 \\ \mathbf{u}(0, \mathbf{s}) &= \mathbf{s} \end{aligned} \tag{14.2.39}$$

where $\mathbf{s} \in \mathbb{R}^{nN}$ is a solution of

$$\phi(\mathbf{s}) \equiv G(\mathbf{s}, \mathbf{u}(1, \mathbf{s})) = \mathbf{0} \tag{14.2.40}$$

and

$$u(\tau, \mathbf{s}) = \begin{pmatrix} u_1(\tau, \mathbf{s}_1) \\ u_2(\tau, \mathbf{s}_2) \\ \vdots \\ u_N(\tau, \mathbf{s}_N) \end{pmatrix} .$$

Theorem 14.2.23

Suppose that $t_i - t_{i-1} = h > 0$ for $1 \le i \le N$, and that the hypothesis of Theorem 14.2.19 are satisfied (with $\mathbf{y}$ replaced by $\mathbf{z}$ and (14.2.23) replaced by 14.2.38). Then, there exists a unique solution of (14.2.39) for any

$$\mathbf{s} \in \left\{ \mathbf{s} \in (\mathbb{R}^n)^N \cong \mathbb{R}^{nN} : \|\mathbf{s} - \mathbf{z}(0)\| = \left(\sum_{i=1}^N \|\mathbf{s}_i - \mathbf{z}_i(0)\|^2 \right)^{1/2} \le \delta e^{-Kh} \right\} .$$

Proof.

The conclusion follows from Theorem 14.2.19 (with K replaced by hK and $[a,b]$ by $[0,1]$) after we note that

$$\begin{aligned} \|F(\tau, \mathbf{u}) - F(\tau, \tilde{\mathbf{u}})\| &= \left(\sum_{i=1}^N \|h f_i(\tau, \mathbf{u}_i) - h f_i(\tau, \tilde{\mathbf{u}}_i)\|^2 \right)^{1/2} \\ &\le \left(\sum_{i=1}^N h^2 K^2 \|\mathbf{u}_i - \tilde{\mathbf{u}}_i\|^2 \right)^{1/2} = hK \|\mathbf{u} - \tilde{\mathbf{u}}\| \end{aligned}$$

for all $(\tau, \mathbf{u})$ and $(\tau, \tilde{\mathbf{u}})$ in

$$\left\{ (\tau, \mathbf{u}) \in [0,1] \times (\mathbb{R}^n)^N : 0 \le \tau \le 1 \text{ and } \|\mathbf{u} - \mathbf{z}(\tau)\| = \left(\sum_{i=1}^N \|\mathbf{u}_i - \mathbf{z}_i(\tau)\|^2 \right)^{1/2} < \delta \right\} . \qquad \blacksquare$$

It follows from the previous theorem that we only need to solve (14.2.40) to get the solution of (14.2.38).

If Newton method is used to approximate the solution of $\phi(\mathbf{s}) = \mathbf{0}$ in (14.2.40), then the initial condition $\mathbf{s}_0$ should be taken from the disk of radius δe^{-Kh} centred at $\mathbf{z}(0)$.

If we compare with the simple Shooting Method of Section 14.2.6, The dimension of the system for the Parallel Shooting Method (i.e. nN) is larger than the dimension of the system for the simple Shooting Method (i.e. n). However, the initial condition $\mathbf{s}_0$ for the Newton method can be chosen from a disk of larger radius for the Parallel Shooting Method (i.e. δe^{-Kh}) than for the simple Shooting Method (i.e. $\delta e^{-K(b-a)}$). The integration time is also shorter for the Parallel Shooting Method (i.e. h) than for the simple Shooting Method (i.e. $b-a$) though repeated n times.

Remark 14.2.24
Theorem 14.2.20 can also be applied to (14.2.38) and (14.2.40). The Newton Method is

$$Q(\mathbf{s}^{[j]})\left(\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\right) = -\phi(\mathbf{s}^{[j]}) \quad , \quad j \geq 0 \ ,$$

where

$$Q(\mathbf{s}) = \mathrm{D}_{\mathbf{s}}\phi(\mathbf{s}) =$$
$$\begin{pmatrix} \mathrm{D}_{\mathbf{y}_1} g(\mathbf{s}_1, \mathbf{u}_N(1,\mathbf{s}_N)) & 0 & \ldots & 0 & \mathrm{D}_{\mathbf{y}_2} g(\mathbf{s}_1, \mathbf{u}_N(1,\mathbf{s}_N)) U_N(1,\mathbf{s}_N) \\ -U_1(1,\mathbf{s}_1) & \mathrm{Id} & \ldots & 0 & 0 \\ 0 & -U_2(1,\mathbf{s}_2) & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & -U_{N-1}(1,\mathbf{s}_{N-1}) & \mathrm{Id} \end{pmatrix},$$

where

$$U_i(\tau, \mathbf{s}_i) = \mathrm{D}_{\mathbf{s}_i} \mathbf{u}_i(\tau, \mathbf{s}_i)$$

for $1 \leq i \leq N$. ♠

14.2.9 Family of Solutions

One of the difficulty with the Shooting Method is to choose $\mathbf{s}^{[0]}$ in the Newton Method. If the boundary value problem that we want to solve is "closed" to another boundary value problem for which we know how to choose $\mathbf{s}^{[0]}$, we may perhaps use this information to guess $\mathbf{s}^{[0]}$ for our boundary value problem. For us, two "closed" boundary value problems will mean that they are two "closed" members of a family of boundary value problems.

Suppose that f and g in (14.2.23) depend on a parameter $\sigma \in [\sigma_a, \sigma_b]$. We get the following **family of boundary value problems**

$$\begin{aligned} \mathbf{y}'(t,\sigma) &= f(t, \mathbf{y}(t,\sigma), \sigma) \quad , \quad a \leq t \leq b \\ g(\mathbf{y}(a,\sigma), \mathbf{y}(b,\sigma), \sigma) &= \mathbf{0} \end{aligned} \tag{14.2.41}$$

for $\sigma_a \leq \sigma \leq \sigma_b$. Recall that $\mathbf{y}'(t,\sigma) = \dfrac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t,\sigma)$. We generally assume that (14.2.41) has a unique isolated solution $\mathbf{y}(t,\sigma)$ for each σ and that the dependence of $\mathbf{y}(t,\sigma)$ on σ is sufficiently differentiable. We get the **family of solutions** $\{\mathbf{y}(t,\sigma) : \sigma_a \leq \sigma \leq \sigma_b\}$.

A simple boundary value problem like (14.2.23) can be included in a family of boundary value problems as in (14.2.41) by defining

$$F(t,\mathbf{y},\sigma) = \sigma f(t,\mathbf{y}) + (1-\sigma)\left(A(t)\mathbf{y} + \mathbf{g}(t)\right)$$

and

$$G(\mathbf{v},\mathbf{w},\sigma) = \sigma g(\mathbf{v},\mathbf{w}) + (1-\sigma)\left(B_a\mathbf{v} + B_b\mathbf{w} - \mathbf{y}_c\right)$$

in (14.2.41). We have a linear boundary value problem (that we may have chosen) for $\sigma = 0$, and our original non-linear boundary value problem for $\sigma = 1$.

To compute a branch of solutions of (14.2.41), we solve

$$\begin{aligned} \mathbf{u}'(t,\mathbf{s},\sigma) &= F(t,\mathbf{u}(t,\mathbf{s},\sigma),\sigma) \quad , \quad a \leq t \leq b \\ \mathbf{u}(a,\mathbf{s},\sigma) &= \mathbf{s} \end{aligned} \tag{14.2.42}$$

where $\mathbf{s}$ is a solution of

$$\phi(\mathbf{s},\sigma) \equiv G(\mathbf{s},\mathbf{u}(b,\mathbf{s},\sigma),\sigma) = \mathbf{0} \ . \tag{14.2.43}$$

Theorem 14.2.25

Suppose that:

1. There is a solution $\mathbf{u}$ of (14.2.42) and (14.2.43) for $\sigma = \sigma_* \in [\sigma_a, \sigma_b]$ and $\mathbf{s} = \mathbf{s}_*$.
2. There exist η_1 and η_2 such that $F(t,\mathbf{w},\sigma)$ is of class C^1 in the tubular neighbourhood of $\{(t,\mathbf{u}(t,\mathbf{s}_*,\sigma_*),\sigma_*) : a \leq t \leq b\}$ defined by
$$\{(t,\mathbf{w},\sigma) : a \leq t \leq b, |\sigma - \sigma_*| < \eta_1 \text{ and } \|\mathbf{w} - \mathbf{u}(t,\mathbf{s}_*,\sigma_*)\| < \eta_2\} \ ,$$
and $G(\mathbf{r},\mathbf{w},\sigma)$ is of class C^1 in the neighbourhood of $(\mathbf{s}_*,\mathbf{u}(b,\mathbf{s}_*,\sigma_*),\sigma_*)$ defined by
$$\{(\mathbf{r},\mathbf{w},\sigma) : \|\mathbf{r} - \mathbf{s}_*\| < \eta_2 \ , \ \|\mathbf{w} - \mathbf{u}(b,\mathbf{s}_*,\sigma_*)\| < \eta_2 \text{ and } |\sigma - \sigma_*| < \eta_1\} \ .$$
3. $\mathrm{D}_{\mathbf{s}}\phi(\mathbf{s}_*,\sigma_*)$ is non-singular.

Then, there exist $\delta > 0$ and a continuously differentiable function $\mathbf{s} :]\sigma_* - \delta, \sigma_* + \delta[\to \mathbb{R}^n$ such that $s(\sigma_*) = \mathbf{s}_*$ and $\mathbf{u}(t,\mathbf{s}(\sigma),\sigma)$ for $a \leq t \leq b$ is a solution of (14.2.42) and (14.2.43).

Proof.
We have that $\phi(\mathbf{s}_*,\sigma_*) = \mathbf{0}$ and $\mathrm{D}_{\mathbf{s}}\phi(\mathbf{s}_*,\sigma_*)$ is non-singular. It follows from the implicit function theorem that all solutions of $\phi(\mathbf{s},\sigma) = \mathbf{0}$ in a sufficiently small open neighbourhood of $(\mathbf{s}_*,\sigma_*)$ is of the form $(\mathbf{s}(\sigma),\sigma)$ for a differentiable function $\mathbf{s} :]\sigma_* - \delta, \sigma_* + \delta[\to \mathbb{R}^n$ with δ sufficiently small. Moreover, $\mathbf{s}(\sigma_*) = \mathbf{s}_*$.

The continuous differentiability of $\mathbf{s}$ comes from our usual assumption that f is as smooth as needed. In the present case, we need f to be continuously differentiable. ∎

Remark 14.2.26

1. Since $\mathbf{s} :]\sigma_* - \delta, \sigma_* + \delta[\to \mathbb{R}^n$ given by the previous theorem is of class C^1, we may derive $\phi(\mathbf{s}(\sigma), \sigma) = \mathbf{0}$ with respect to σ to get

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\phi(\mathbf{s}(\sigma), \sigma) = \mathrm{D}_{\mathbf{s}}\phi(\mathbf{s}, \sigma)\bigg|_{\mathbf{s}=\mathbf{s}(\sigma)} \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}\sigma}(\sigma) + \frac{\partial \phi}{\partial \sigma}(\mathbf{s}(\sigma), \sigma) = \mathbf{0} \ .$$

This is a differential equation for $\mathbf{s}(\sigma)$ with initial condition $\mathbf{s}(\sigma_*) = \mathbf{s}_*$. Note that

$$\frac{\partial \phi}{\partial \sigma}(\mathbf{s}, \sigma) = \frac{\partial G}{\partial \sigma}(\mathbf{s}, \mathbf{u}(b, \mathbf{s}, \sigma), \sigma) + \mathrm{D}_{\mathbf{y}_2} G(\mathbf{s}, \mathbf{u}(b, \mathbf{s}, \sigma), \sigma)\, V(b, \mathbf{s}, \sigma) \ ,$$

where

$$V(t, \mathbf{s}, \sigma) = \frac{\partial \mathbf{u}}{\partial \sigma}(t, \mathbf{s}, \sigma)$$

is the solution of

$$V'(t, \mathbf{s}, \sigma) = \mathrm{D}_{\mathbf{u}} F(t, \mathbf{u}(t, \mathbf{s}, \sigma), \sigma)\, V(t, \mathbf{s}, \sigma) + \frac{\partial F}{\partial \sigma}(t, \mathbf{u}(t, \mathbf{s}, \sigma), \sigma) \ .$$

2. From $\mathbf{s}(\sigma_* + \delta) = \mathbf{s}(\sigma_*) + \mathbf{s}'(\sigma_*)\,\delta + O(\delta^2)$, we may choose $\mathbf{s}(\sigma_*) + \mathbf{s}'(\sigma_*)\delta$ as initial value in the Newton iterative method for the boundary value problem (14.2.42) and (14.2.43) given by $\sigma = \sigma_* + \delta$. Recursively, we may be able to "find" a branch of solutions $\mathbf{s} : [\sigma_a, \sigma_b] \to \mathbb{R}^n$ for the family of boundary value problems given by (14.2.42) and (14.2.43). This subject of "path following" is another exciting subject of numerical analysis that unfortunately we will not address in this book.

♠

14.3 Finite Difference Methods

This section is based on Keller's lectures [22] and Ascher et al.'s book [2].

The next chapter will cover finite difference methods to solve partial differential equations. The present section can be seen as an introduction to this broader subject since a boundary value problem for ordinary differential equation is a one-dimensional boundary value problem for partial differential equation.

The general boundary value problem is of the form

$$\begin{aligned} P(\mathbf{y}(t)) \equiv \mathbf{y}'(t) - f(t, \mathbf{y}(t)) &= \mathbf{0} \quad , \quad a \leq t \leq b \\ g(\mathbf{y}(a), \mathbf{y}(b)) &= \mathbf{0} \end{aligned} \tag{14.3.1}$$

As usual, let $\{t_i\}_{i=0}^N$ be a partition of $[a, b]$ such that $t_0 = a$, $t_N = b$, $t_{i+1} = t_i + h_i$ with $h_i > 0$ for $0 \leq i < N$ and $h = \max_{0 \leq i < N} h_i \leq \theta \min_{0 \leq i < N} h_i$ for some constant θ.

The associated general form of a finite difference method to approximate the solution of (14.3.1) is

$$\begin{aligned} P_{i,h}(\mathbf{W}) &= \mathbf{0} \quad , \quad 0 \le i < N \\ g(\mathbf{w}_0, \mathbf{w}_N) &= \mathbf{0} \end{aligned} \tag{14.3.2}$$

where $\mathbf{W} = \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1}$. We hope that the solution $\{\mathbf{w}_i\}_{i=0}^N$ of this finite difference equation (14.3.2) will provide an approximation of the solution of (14.3.1). Namely, we hope that $\mathbf{w}_i$ will be a good approximation of $\mathbf{y}_i \equiv \mathbf{y}(t_i)$ for $0 \le i \le N$.

Example 14.3.1
The **trapezoidal method** or **scheme** to solve a general boundary value problem is a one-step method defined by

$$\begin{aligned} P_{i,h}(\mathbf{W}) = \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{h_i} - \frac{1}{2}\left(f(t_{i+1}, \mathbf{w}_{i+1}) + f(t_i, \mathbf{w}_i)\right) &= \mathbf{0} \quad , \quad 0 \le i < N \\ g(\mathbf{w}_0, \mathbf{w}_N) &= \mathbf{0} \end{aligned}$$

where $h_i = t_{i+1} - t_i$. ♣

Remark 14.3.2 (Important)
In this section, when we write $\lim_{h\to 0} E = 0$ for some expression E that depends on h, we mean that for each $\epsilon > 0$, there exist $h_\epsilon > 0$ such that $|E| < \epsilon$ for all partition $\{t_i\}_{i=0}^N$ as defined above such that $h < h_\epsilon$. If N is included in the expression E, it is the N associated to the partition with maximum step size h under consideration in the expression E.

The same consideration applies if we say that an expression E that depends on h is true for $h < h_0$. Namely, it means that E is true for all partition $\{t_i\}_{i=0}^N$ as defined above such that $h < h_0$ and, if N is included in the expression E, then N is associated to the partition with maximum step size h under consideration in the expression E. ♠

To determine the quality of a finite difference method to approximate the solution of a boundary value problem, we will use concepts similar to those used before for the initial value problems; namely, convergence, consistency and stability.

Definition 14.3.3

The method (14.3.2) is **convergent** if, for all well-posed boundary value problem (14.3.1),

$$\lim_{h\to 0} \max_{0\le i\le N} \|\mathbf{y}(t_i) - \mathbf{w}_i\| = 0 \ .$$

Remark 14.3.4
As for the shooting methods, we will not consider any perturbation of (14.3.1) as we did for the initial value problems. We will come back on convergence of finite difference methods in Chapter 15. ♠

Definition 14.3.5

The **local truncation error** of a finite difference method as in (14.3.2) is

$$\tau_i(\mathbf{y}) = P_{i,h}(\mathbf{Y}) \quad , \quad 0 \le i < N \ ,$$

where $\mathbf{Y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1}$. Recall that $\mathbf{y}_i = \mathbf{y}(t_i)$ for $0 \le i \le N$, where $\{t_i\}_{i=0}^N$ is any partition of $[a,b]$ with the maximum step-size h.
The method (14.3.2) is of **order** $p > 0$ if there exist a function $\tau : \mathbb{R}^n \to [0,\infty[$ such that $\|\tau_i(\mathbf{y})\| \le \tau(\mathbf{y}) = O(h^p)$ for $0 \le i < N$.

Definition 14.3.6

The finite difference method (14.3.2) is **consistent** if, for all well-posed boundary value problem (14.3.1),

$$\lim_{h\to 0} \max \left\{ \max_{0\le i<N} \|\tau_i(\mathbf{y})\| . \|g(\mathbf{y}_0, \mathbf{y}_N\| \right\} = 0 \ .$$

Definition 14.3.7

The finite difference method (14.3.2) is **stable** if, for any well-posed boundary value problem (14.3.1), there exist $K > 0$, $h_0 > 0$ and $\delta > 0$ such that

$$\|\mathbf{u}_i - \mathbf{v}_i\| \le K \max \left\{ \|g(\mathbf{u}_0, \mathbf{u}_N) - \mathbf{g}(\mathbf{v}_0, \mathbf{v}_N)\|, \max_{1\le i<N} \|P_{i,h}(\mathbf{U}) - P_{i,h}(\mathbf{V})\| \right\} \tag{14.3.3}$$

for all $\mathbf{U} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{pmatrix}$ and $\mathbf{V} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{pmatrix}$ in

$$S_\delta(\mathbf{y}) = \left\{ \mathbf{w} \in (\mathbb{R}^n)^{N+1} : \|\mathbf{w}_i - \mathbf{y}_i\| < \delta \quad \text{for} \quad 0 \le i \le N \right\} \ ,$$

and all $h < h_0$.

The following theorem will be proved in Chapter 15 (Theorem 15.3.9) about finite difference methods for partial differential equations.

Theorem 14.3.8

If a method like (14.3.2) is stable and consistent for the linear boundary value problem (14.3.1), then the method is convergent.

We will prove a version of this theorem for the linear boundary value problems in the next section.

14.3.1 Finite Difference Methods for Linear Boundary Value Problems

As we did for the shooting methods, we start with the linear boundary value problem

$$\begin{aligned} P(\mathbf{y}(t)) = \mathbf{y}'(t) - A(t)\mathbf{y}(t) - f(t) = \mathbf{0} \quad &, \quad a \le t \le b \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) - \mathbf{y}_c = \mathbf{0} & \end{aligned} \tag{14.3.4}$$

The general form of a finite difference method to approximate the solution of (14.3.4) is

$$\begin{aligned} P_{i,h}(\mathbf{W}) = L_{i,h}(\mathbf{W}) - F_{i,h}(f) = \mathbf{0} \quad &, \quad 0 \le i < N \\ B_a\mathbf{w}_0 + B_b\mathbf{w}_N - \mathbf{y}_c = \mathbf{0} & \end{aligned} \tag{14.3.5}$$

where $\mathbf{W} = \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1}$ and $L_{i,h}(\mathbf{W})$ is associated to the linear part $L(\mathbf{y}(t)) = \mathbf{y}'(t) - A(t)\mathbf{y}(t)$.

Example 14.3.9
The **midpoint scheme** or **centred Euler scheme** to solve linear boundary value problems is a one-step method defined by

$$\begin{aligned} P_{i,h}(\mathbf{W}) = L_{i,h}(\mathbf{W}) - F_{i,h}(f) = \mathbf{0} \quad &, \quad 0 \le i < N \\ B_a\mathbf{w}_0 + B_b\mathbf{w}_N = \mathbf{y}_c & \end{aligned}$$

where $L_{i,h}(\mathbf{W}) = \dfrac{\mathbf{w}_{i+1} - \mathbf{w}_i}{h_i} - \dfrac{1}{2}A(t_i + h_i/2)\left(\mathbf{w}_{i+1} + \mathbf{w}_i\right)$ and $F_{i,h}(f) = f(t_i + h_i/2)$. ♣

Example 14.3.10
The trapezoidal scheme to solve linear boundary value problems is a one-step method defined by

$$\begin{aligned} P_{i,h}(\mathbf{W}) = L_{i,h}(\mathbf{W}) - F_{i,h}(f) = \mathbf{0} \quad &, \quad 0 \le i < N \\ B_a\mathbf{w}_0 + B_b\mathbf{w}_N = \mathbf{y}_c & \end{aligned}$$

where $L_{i,h}(\mathbf{W}) = \dfrac{\mathbf{w}_{i+1} - \mathbf{w}_i}{h} - \dfrac{1}{2}\left(A(t_i + h)\mathbf{w}_{i+1} + A(t_i)\mathbf{w}_i\right)$ and $F_{i,h}(f) = \dfrac{1}{2}\left(f(t_i + h) + f(t_i)\right)$.
♣

Because of the very special form of (14.3.5), in particular the linearity of $L_{i,h}$, the stability condition (14.3.3) can be reduced to

$$\|\mathbf{u}_i\| \le K \max\left\{\|B_a\mathbf{u}_0 + B_b\mathbf{u}_N\|, \max_{0 \le j < N} \|L_{j,h}(\mathbf{U})\|\right\} \quad , \quad 0 \le i \le N \ , \tag{14.3.6}$$

for all $\mathbf{U} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1}$ and all $h < h_0$.

Proposition 14.3.11

If a method like (14.3.5) is stable and consistent for the linear boundary value problem (14.3.4), then it is convergent.

Proof.

To prove this result, let $\mathbf{r}_i = \mathbf{y}(t_i) - \mathbf{w}_i$ for $0 \leq i \leq N$, and $\mathbf{R} = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_N \end{pmatrix}$. The local truncation error is

$$\tau_i(\mathbf{y}) = L_{i,h}(\mathbf{Y}) - F_{i,h}(f) = L_{i,h}(\mathbf{R}) + \underbrace{L_{i,h}(\mathbf{W}) - F_{i,h}(f)}_{=P_{i,h}(\mathbf{W})=\mathbf{0}} = L_{i,h}(\mathbf{R}) \quad , \quad 0 \leq i < N \ ,$$

and $B_a\mathbf{r}_0 + B_b\mathbf{r}_N = \mathbf{0}$ because $B_a\mathbf{w}_0 + B_b\mathbf{w}_N - \mathbf{y}_c = \mathbf{0}$ and $B_a\mathbf{y}_0 + B_b\mathbf{y}_N - \mathbf{y}_c = \mathbf{0}$. Since the method is consistent, $\lim_{h\to 0} \max_{0\leq i<N} \|\tau_i(\mathbf{y})\| = 0$. Finally, since the method is stable, we have from the remark before the statement of the proposition that

$$\|\mathbf{r}_i\| \leq K \max\left\{ \|B_a\mathbf{r}_0 + B_b\mathbf{r}_N\|, \max_{1\leq j<N} \|L_{j,h}(\mathbf{R})\| \right\} \leq K \max_{0\leq j<N} \|\tau_j(\mathbf{y})\|$$

for some constant K and $0 \leq i \leq N$. Thus

$$0 \leq \lim_{h\to 0} \max_{1\leq i\leq N} \|\mathbf{r}_i\| \leq K \lim_{h\to 0} \max_{0\leq j<N} \|\tau_j(\mathbf{y})\| = 0 \ ,$$

where, as usual, N is associated to the chosen partition of $[a,b]$ of maximum size h. ∎

The result is also true for finite difference methods (14.3.2) applied to the general boundary value problem (14.3.1) (Theorem 14.3.8 above) but the proof is not as direct as for the linear boundary value problems above.

The next two propositions will be used to prove that the method (14.3.5) applied to the linear boundary value problem (14.3.4) is stable and consistent if it is stable and consistent when applied to the initial value problem

$$\begin{aligned} L(\mathbf{y}(t)) = \mathbf{y}'(t) - A(t)\mathbf{y}(t) &= \mathbf{0} \\ \mathbf{y}(a) &= \mathbf{s} \in \mathbb{R}^n \end{aligned}$$

(Corollary 14.3.16 below).

Proposition 14.3.12

Consider two linear boundary value problems

$$\begin{aligned} L(\mathbf{y}(t)) = \mathbf{y}'(t) - A(t)\mathbf{y}(t) = f(t) \quad &, \quad a \leq t \leq b \\ B_a^{[\nu]}\mathbf{y}(a) + B_b^{[\nu]}\mathbf{y}(b) = \mathbf{y}_c & \end{aligned} \tag{14.3.7}$$

for $\nu = 0$ and 1.

1. For ν fixed,

$$\begin{aligned} L(Y^{[\nu]}(t)) = 0 \quad &, \quad a \leq t \leq b \\ B_a^{[\nu]}Y^{[\nu]}(a) + B_b^{[\nu]}Y^{[\nu]}(b) = \mathrm{Id} & \end{aligned} \tag{14.3.8}$$

   has a unique solution $Y^{[\nu]}(t)$ if and only if (14.3.7) has a unique solution.

2. Moreover, if (14.3.7) for $\nu = 0$ has a unique solution, then (14.3.7) for $\nu = 1$ has a unique solution if and only if $B_a^{[1]}Y^{[0]}(a) + B_b^{[1]}Y^{[0]}(b)$ is invertible.

Proof.
1) For this part, we assume that ν is fixed. From Theorem 14.2.3, the boundary value problem (14.3.7) has a unique solution if and only if $Q^{[\nu]} = B_a^{[\nu]} + B_b^{[\nu]}Y(b)$ is invertible, where $Y(t)$ is the (fundamental) solution of $L(Y) = 0$ with $Y(a) = \mathrm{Id}$.

Moreover, it follows from the second step in Algorithm 14.2.1 with $\mathbf{y}_c = 0$ and $\mathbf{y}_0(t) = 0$ for all t that the solution of (14.3.8) is of the form $Y^{[\nu]}(t) = Y(t)R^{[\nu]}$, where $R^{[\nu]}$ is a constant matrix satisfying

$$\mathrm{Id} = B_a^{[\nu]}Y^{[\nu]}(a) + B_b^{[\nu]}Y^{[\nu]}(b) = \left(B_a^{[\nu]} + B_b^{[\nu]}Y(b)\right)R^{[\nu]} = Q^{[\nu]}R^{[\nu]} .$$

Such a system has a unique solution $R^{[\nu]}$ if and only if $Q^{[\nu]}$ is non-singular. In that case, $R^{[\nu]} = \left(Q^{[\nu]}\right)^{-1}$.

2) To prove this part of the theorem, suppose that (14.3.7) with $\nu = 0$ has a unique solution. Then (14.3.8) with $\nu = 0$ has a unique solution given by $Y^{[0]}(t) = Y(t)\left(Q^{[0]}\right)^{-1}$. In particular, $Q^{[0]}$ is invertible. Since

$$B_a^{[1]}Y^{[0]}(a) + B_b^{[1]}Y^{[0]}(b) = \left(B_a^{[1]} + B_b^{[1]}Y(b)\right)\left(Q^{[0]}\right)^{-1} = Q^{[1]}\left(Q^{[0]}\right)^{-1} ,$$

we have that $Q^{[1]}$ is invertible if and only if $B_a^{[1]}Y^{[0]}(a) + B_b^{[1]}Y^{[0]}(b)$ is invertible. Thus, (14.3.8) with $\nu = 1$ has a unique solution if and only if $B_a^{[1]}Y^{[0]}(a) + B_b^{[1]}Y^{[0]}(b)$ is invertible. The conclusion from the first part for $\nu = 1$. ∎

The general form (14.3.5) of a finite difference method for a linear boundary value problem can be written explicitly as

$$\begin{aligned} L_{i,h}(\mathbf{W}) = \sum_{k=0}^{N} C_{i,k}\mathbf{w}_k = F_{i,h}(f) \quad , \quad 0 \le i < N \\ B_a\mathbf{w}_0 + B_b\mathbf{w}_N = \mathbf{y}_c \end{aligned} \tag{14.3.9}$$

The $C_{i,k}$ may depend on h and t_i. If we set

$$A = \begin{pmatrix} B_a & 0 & \dots & B_b \\ C_{0,0} & C_{0,1} & \dots & C_{0,N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-1,0} & C_{N-1,1} & \dots & C_{N-1,N} \end{pmatrix} , \ \mathbf{W} = \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{w}_q \\ \vdots \\ \mathbf{w}_N \end{pmatrix} \quad \text{and} \quad \mathbf{F} = \begin{pmatrix} \mathbf{y}_c \\ F_{0,h}(f) \\ \vdots \\ F_{N-1,h}(f) \end{pmatrix} ,$$

we can rewrite (14.3.9) as

$$A\mathbf{W} = \mathbf{F} . \tag{14.3.10}$$

Proposition 14.3.13

The finite difference method (14.3.9) is stable for the linear boundary value problem (14.3.4) if and only if there exist two constants K and h_0 such that A^{-1} exists and $\|A^{-1}\|_\infty < K$ for $0 < h < h_0$.

Proof.
A) Suppose that (14.3.9) is stable for the linear boundary value problem (14.3.4). Thus, there exist $K > 0$ and $h_0 > 0$ such that (14.3.6) is satisfied for all $\mathbf{U} \in (\mathbb{R}^n)^{N+1}$ and $h < h_0$. But (14.3.6) is another way of saying that $\|\mathbf{U}\|_\infty \le K\|A\mathbf{U}\|_\infty$ for all $\mathbf{U} \in (\mathbb{R}^n)^{N+1}$ and $h < h_0$. From this relation, we have that A is one-to-one and therefore invertible. We can then write that $\|A^{-1}\mathbf{U}\|_\infty \le K\|\mathbf{U}\|_\infty$ for all $\mathbf{U} \in (\mathbb{R}^n)^{N+1}$ and $h < h_0$; namely, $\|A^{-1}\|_\infty \le K$ for $h < h_0$.

B) Suppose that there exist two constants K and h_0 such that A^{-1} exists and $\|A^{-1}\|_\infty < K$ for $0 < h < h_0$. From the definition of the norm of matrices, we get $\|A^{-1}\mathbf{U}\|_\infty \le K\|\mathbf{U}\|_\infty$ for all $\mathbf{U} \in (\mathbb{R}^n)^{N+1}$ and $h < h_0$. Thus, $\|\mathbf{U}\|_\infty \le K\|A\mathbf{U}\|_\infty$ for all $\mathbf{U} \in (\mathbb{R}^n)^{N+1}$ and $h < h_0$. This is exactly the statement of (14.3.6); namely, (14.3.9) is stable for the linear boundary value problem (14.3.4). ∎

Theorem 14.3.14

Consider the linear boundary value problems (14.3.7) and the finite difference methods

$$\begin{aligned} L_{i,h}(\mathbf{W}) = \sum_{k=0}^{N} C_{i,k}\,\mathbf{w}_k = F_{i,h}(f) \quad , \quad 0 \le i < N \\ B_a^{[\nu]}\mathbf{w}_0 + B_b^{[\nu]}\mathbf{w}_N = \mathbf{y}_c \end{aligned} \tag{14.3.11}$$

for $\nu = 0$ and 1. Suppose that both linear boundary value problems in (14.3.7) have a unique solution. The method (14.3.11) with $\nu = 0$ is stable and consistent for (14.3.7) with $\nu = 0$ if and only if the method (14.3.11) with $\nu = 1$ is stable and consistent for (14.3.7) with $\nu = 1$.

Remark 14.3.15
Before proving this theorem, it will help to review some of the properties of the infinity norm.

We have $\|\mathbf{u}\|_\infty = \max_{1\le i\le n}|u_i|$ for $\mathbf{u}\in\mathbb{R}^n$ and $\|\mathbf{U}\|_\infty = \max_{0\le i\le N}\|\mathbf{u}_i\|_\infty$ for $\mathbf{U} = \begin{pmatrix}\mathbf{u}_0\\ \mathbf{u}_1\\ \vdots\\ \mathbf{u}_N\end{pmatrix} \in (\mathbb{R}^n)^{N+1}$. The associated norm of the linear mapping from $(\mathbb{R}^n)^{N+1}$ to itself defined by the matrix

$$M = \begin{pmatrix} M_{0,0} & M_{0,1} & \dots & M_{0,N}\\ M_{1,0} & M_{1,1} & \dots & M_{1,N}\\ \vdots & \vdots & \ddots & \vdots\\ M_{N,0} & M_{N,1} & \dots & M_{N,N}\end{pmatrix},$$

where the $M_{i,j}$ are $n\times n$ matrices, is given by $\|M\|_\infty = \max_{\substack{\mathbf{U}\in(\mathbb{R}^n)^{N+1}\\ \mathbf{U}\neq\mathbf{0}}}\dfrac{\|M\mathbf{U}\|_\infty}{\|\mathbf{U}\|_\infty}$.

We now show that there exists $H>1$ such that

$$\frac{1}{H}\max_{0\le i\le N}\sum_{j=0}^N \|M_{i,j}\|_\infty \le \|M\|_\infty \le \max_{0\le i\le N}\sum_{j=0}^N\|M_{i,j}\|_\infty\ . \tag{14.3.12}$$

Since all norms on a finite dimensional space are equivalent, there exist constants C_1 and C_2 such that

$$C_1\max_{0\le i\le N}\sum_{j=0}^N \|M_{i,j}\|_\infty \le \|M\|_\infty \le C_2\max_{0\le i\le N}\sum_{j=0}^N\|M_{i,j}\|_\infty \tag{14.3.13}$$

because $\max_{0\le i\le N}\sum_{j=0}^N\|M_{i,j}\|_\infty$ is a norm on the space of $n(N+1)\times n(N+1)$ matrices.

If (14.3.13) is true with $C_1\ge 1$, then it is obviously true if we replace C_1 by a constant $1/H$ with $H>1$. We can take $C_2=1$ because

$$\begin{aligned}\|M\mathbf{U}\|_\infty &= \max_{0\le i\le N}\Big\|\sum_{j=0}^N M_{i,j}\mathbf{u}_j\Big\|_\infty \le \max_{0\le i\le N}\left(\sum_{j=0}^N\|M_{i,j}\|_\infty\|\mathbf{u}_j\|_\infty\right)\\ &\le \max_{0\le i\le N}\left(\sum_{j=0}^N\|M_{i,j}\|_\infty\right)\max_{0\le j\le N}\|\mathbf{u}_j\|_\infty = \max_{0\le i\le N}\left(\sum_{j=0}^N\|M_{i,j}\|_\infty\right)\|\mathbf{U}\|_\infty\end{aligned}$$

for all $\mathbf{U}\in(\mathbb{R}^n)^{N+1}$. Thus $\|M\|_\infty \le \max_{0\le i\le N}\sum_{j=0}^N\|M_{i,j}\|_\infty$. ♠

Proof of Theorem 14.3.14.
Since $P_{i,h}(\mathbf{W}) = L_{i,h}(\mathbf{W}) - F_{i,h}(f)$ is independent of ν, the local truncation error $\tau_i(\mathbf{y}^{[\nu]})$ for $0\le i<N$ are identical for both problems. Hence the methods are either both consistent or both non-consistent. Recall that the boundary conditions are exactly satisfied in both cases.

Suppose that the method (14.3.11) with $\nu = 0$ is stable and consistent for (14.3.7) with $\nu = 0$. Let

$$A^{[\nu]} = \begin{pmatrix} B_a^{[\nu]} & \mathbf{0} & \dots & B_b^{[\nu]} \\ C_{0,0} & C_{0,1} & \dots & C_{0,N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-1,0} & C_{N-1,1} & \dots & C_{N-1,N} \end{pmatrix} .$$

For h small enough, we can write

$$A^{[1]} = \left(\mathrm{Id} + D\left(A^{[0]}\right)^{-1}\right) A^{[0]} , \tag{14.3.14}$$

where

$$D = A^{[1]} - A^{[0]} = \begin{pmatrix} B_a^{[1]} - B_a^{[0]} & 0 & \dots & B_b^{[1]} - B_b^{[0]} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} ,$$

because $(A^{[0]})^{-1}$ exists. To be precise, according to Proposition 14.3.13, there exist $K > 0$ and $h_0 > 0$ such that $(A^{[0]})^{-1}$ exists and $\| (A^{[0]})^{-1} \|_\infty \leq K$ if $0 < h < h_0$.

Suppose that

$$\left(A^{[0]}\right)^{-1} = \begin{pmatrix} Z_{0,0} & Z_{0,1} & \dots & Z_{0,N} \\ Z_{1,0} & Z_{1,1} & \dots & Z_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{N,0} & Z_{N,1} & \dots & Z_{N,N} \end{pmatrix} ,$$

where the matrices $Z_{i,j}$ are $n \times n$ matrices. Since $A^{[0]} (A^{[0]})^{-1} = \mathrm{Id}$, we get $B_a^{[0]} Z_{0,j} + B_b^{[0]} Z_{N,j} = 0$ for $1 \leq j \leq N$ and $B_a^{[0]} Z_{0,0} + B_b^{[0]} Z_{N,0} = \mathrm{Id}$. Therefore,

$$\mathrm{Id} - D\left(A^{[0]}\right)^{-1} = \begin{pmatrix} Q_{0,0} & Q_{0,1} & \dots & Q_{0,N} \\ 0 & \mathrm{Id} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{Id} \end{pmatrix} ,$$

where

$$Q_{0,j} = B_a^{[1]} Z_{0,j} + B_b^{[1]} Z_{N,j}$$

for $j = 0, 1, \ldots, N$.

Moreover, $A^{[0]} (A^{[0]})^{-1} = \mathrm{Id}$ implies that $B_a^{[0]} Z_{0,0} + B_b^{[0]} Z_{N,0} = \mathrm{Id}$ and $L_{i,h}(Z_0) = 0$ for $1 \leq i < N$ where $Z_0 = \begin{pmatrix} Z_{0,0} \\ Z_{1,0} \\ \vdots \\ Z_{N,0} \end{pmatrix}$. Thus $\{Z_{i,0}\}_{i=0}^{N}$ is an approximation of the solution of

$$\begin{aligned} L(Y(t)) = Y'(t) - A(t)Y(t) = 0 \quad &, \quad a \leq t \leq b \\ B_a^{[0]} Y(a) + B_b^{[0]} Y(b) = \mathrm{Id} & \end{aligned}$$

Since the method (14.3.11) with $\nu = 0$ is consistent and stable, it is convergent according to Proposition 14.3.11. Therefore, if we use the method (14.3.11) for the linear boundary value problem (14.3.7) with $\nu = 0$ and $f = 0$, we get that $\lim_{h\to 0}\max_{0\leq i\leq N} \|Z_{i,0} - Y^{[0]}(t_i)\|_\infty = 0$. Hence

$$\begin{aligned}
&\left\| Q_{0,0} - B_a^{[1]} Y^{[0]}(a) - B_b^{[1]} Y^{[0]}(b) \right\|_\infty \\
&\qquad = \left\| B_a^{[1]}(Z_{0,0} - Y^{[0]}(a)) - B_b^{[1]}(Z_{N,0} - Y^{[0]}(b)) \right\|_\infty \to 0
\end{aligned} \tag{14.3.15}$$

as $h \to 0$. From Proposition 14.3.12 and our hypothesis about the uniqueness of the solutions, we have that $B_a^{[1]} Y^{[0]}(a) + B_b^{[1]} Y^{[0]}(b)$ is invertible. Thus, if we select $\tilde{h}_0 < h_0$ small enough such that

$$\left\| Q_{0,0} - B_a^{[1]} Y^{[0]}(a) - B_b^{[1]} Y^{[0]}(b) \right\|_\infty < \frac{1}{2} \left\| \left(B_a^{[1]} Y^{[0]}(a) + B_b^{[1]} Y^{[0]}(b) \right)^{-1} \right\|_\infty^{-1}$$

for $h < \tilde{h}_0$, then it follows from the Banach Lemma (Corollary 3.2.7) that $Q_{0,0}$ is invertible for $h < \tilde{h}_0$ and

$$\left\| Q_{0,0}^{-1} \right\| \leq T \equiv 2 \left\| \left(B_a^{[1]} Y^{[0]}(a) + B_b^{[1]} Y^{[0]}(b) \right)^{-1} \right\|_\infty .$$

Therefore, $\mathrm{Id} - D\left(A^{[0]}\right)^{-1}$ is invertible for $h < \tilde{h}_0$ and it follows from (14.3.14) that $A^{[1]}$ is invertible for $h < \tilde{h}_0$. In fact, we have

$$\left(A^{[1]}\right)^{-1} = \left(A^{[0]}\right)^{-1} \begin{pmatrix} Q_{0,0}^{-1} & -Q_{0,0}^{-1} Q_{0,1} & \cdots & -Q_{0,0}^{-1} Q_{0,N} \\ 0 & \mathrm{Id} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{Id} \end{pmatrix} . \tag{14.3.16}$$

Finally, we now use (14.3.12) in Remark 14.3.15 to obtain

$$\begin{aligned}
\sum_{j=1}^N \|Q_{0,j}\|_\infty &= \sum_{j=1}^N \|B_a^{[1]} Z_{0,j} + B_b^{[1]} Z_{N,j}\|_\infty \leq \|B_a^{[1]}\|_\infty \underbrace{\sum_{j=1}^N \|Z_{0,j}\|_\infty}_{\leq H \|(A^{[0]})^{-1}\|_\infty} + \|B_b^{[1]}\|_\infty \underbrace{\sum_{j=1}^N \|Z_{N,j}\|_\infty}_{\leq H \|(A^{[0]})^{-1}\|_\infty} \\
&\leq H \left\| \left(A^{[0]}\right)^{-1} \right\|_\infty \left(\|B_a^{[1]}\|_\infty + \|B_b^{[1]}\|_\infty \right) \leq KH \left(\|B_a^{[1]}\|_\infty + \|B_b^{[1]}\|_\infty \right) .
\end{aligned}$$

Hence, we get from (14.3.16) that

$$\left\| \left(A^{[1]}\right)^{-1} \right\|_\infty \leq KT \left(1 + KH \left(\|B_a^{[1]}\|_\infty + \|B_b^{[1]}\|_\infty \right) \right)$$

for $h < \tilde{h}_0$. Thus, from Proposition 14.3.13, the finite difference method (14.3.11) with $\nu = 1$ is stable for the linear boundary value problem (14.3.7) with $\nu = 1$.

The opposite implication follows by interchanging $\nu = 0$ and $\nu = 1$. ∎

Corollary 14.3.16

Suppose that (14.3.4) has a unique solution. The finite difference method (14.3.9) is stable and consistent for (14.3.4) if and only if the finite difference method

$$\begin{aligned} L_{i,h}(\mathbf{W}) &= \sum_{k=0}^{N} C_{i,k}\,\mathbf{w}_k = F_{i,h}(f) \quad , \quad 0 \leq i < N \\ \mathbf{w}_0 &= \mathbf{y}_c \end{aligned} \tag{14.3.17}$$

is stable and consistent for the initial value problem

$$\begin{aligned} L(\mathbf{y}(t)) &= \mathbf{y}'(t) - A(t)\mathbf{y}(t) = f(t) \quad , \quad a \leq t \leq b \\ \mathbf{y}(a) &= \mathbf{y}_c \end{aligned} \; . \tag{14.3.18}$$

Proof.
The conclusion follows from Theorem 14.3.14 with (14.3.4) and (14.3.9) as the linear boundary value problem with its associated finite difference method for $\nu = 1$, and (14.3.18) and (14.3.17) as the linear boundary value problem with its associated finite difference method for $\nu = 0$. Note that (14.3.18) has a unique solution. ∎

14.3.2 Numerical Aspect of the One-Step Finite Difference Method for Linear Boundary Value Problems

If only $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$ are used in (14.3.9), we say that the method is a one-step finite difference method.

Example 14.3.17

1. For the midpoint scheme, we have $C_{i,i} = -\dfrac{1}{h_i}\,\mathrm{Id} - \dfrac{1}{2}\,A(t_i + h_i/2)$, $C_{i,i+1} = \dfrac{1}{h_i}\,\mathrm{Id} - \dfrac{1}{2}\,A(t_i + h_i/2)$ and $F_{i,h}(f) = f(t_i + h_i/2)$ for $0 \leq i < N$. We also have that $C_{i,j} = 0$ otherwise.

2. For the trapezoidal scheme, we have $C_{i,i} = -\dfrac{1}{h_i}\,\mathrm{Id} - \dfrac{1}{2}\,A(t_i)$, $C_{i,i+1} = \dfrac{1}{h_i}\,\mathrm{Id} - \dfrac{1}{2}\,A(t_i + h)$ and $F_{i,h}(f) = \dfrac{1}{2}\left(f(t_i) + f(t_i + h)\right)$ for $0 \leq i < N$. We also have that $C_{i,j} = 0$ otherwise.

♣

If the boundary conditions are separable, namely $B_a^{[b]} = 0$ in (14.2.11) of Section 14.2.3, we can rewrite A as

$$A = \begin{pmatrix} B_a^{[a]} & 0 & \dots & 0 & 0 \\ C_{0,0} & C_{0,1} & \dots & 0 & 0 \\ 0 & C_{1,1} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & C_{N-1,N-1} & C_{N-1,N} \\ 0 & 0 & \dots & 0 & B_b^{[b]} \end{pmatrix} , \tag{14.3.19}$$

where $B_a^{[a]}$ is a $(n-q)\times n$ matrix and $B_b^{[b]}$ is a $q\times n$ matrix. We also rewrite $\mathbf{F}$ to get

$$\mathbf{F} = \begin{pmatrix} \mathbf{y}_c^{[a]} \\ F_0(f) \\ \vdots \\ F_{N-1}(f) \\ \mathbf{y}_c^{[b]} \end{pmatrix} .$$

As we can see, the problem now is to solve a large system of linear equations. The matrix A in (14.3.19) is a block tridiagonal matrix of the form

$$A = \begin{pmatrix} A_0 & C_0 & 0 & \ldots & 0 & 0 \\ B_1 & A_1 & C_1 & \ldots & 0 & 0 \\ 0 & B_2 & A_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & A_{N-1} & C_{N-1} \\ 0 & 0 & 0 & \ldots & B_N & A_N \end{pmatrix} ,$$

where each block is a $n\times n$ matrix, the q last rows of the $n\times n$ matrices B_j and the $n-q$ first rows of the $n\times n$ matrices C_j are null. Moreover, if A is nonsingular then we can express it as $A = LU$, where

$$L_h = \begin{pmatrix} L_{0,0} & 0 & \ldots & 0 & 0 \\ L_{1,0} & L_{1,1} & \ldots & 0 & 0 \\ 0 & L_{2,1} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & L_{N,N-1} & L_{N,N} \end{pmatrix} \quad \text{and} \quad U_h = \begin{pmatrix} U_{0,0} & U_{0,1} & 0 & \ldots & 0 \\ 0 & U_{1,1} & U_{1,2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & U_{N-1,N} \\ 0 & 0 & 0 & \ldots & U_{N,N} \end{pmatrix} .$$

The $n\times n$ matrices $L_{i,j}$ and $U_{i,j}$ satisfy

$$\begin{gathered} L_{0,0}U_{0,0} = A_0 \\ \left.\begin{aligned} L_{i-1,i-1}U_{i-1,i} &= C_{i-1} \\ L_{i,i-1}U_{i-1,i-1} &= B_i \\ L_{i,i}U_{i,i} &= A_i - L_{i,i-1}U_{i-1,i} \end{aligned}\right\} \quad , \quad i = 1, 2, \ldots, N \end{gathered}$$

The LU decomposition of A above is not unique. To determine a unique LU decomposition, it is standard to set $L_{i,i} = \mathrm{Id}$ for $0 \le i \le N$. We do that below for the case of the partially separable boundary conditions. It is proved in [22] that this decomposition can be obtained using row interchanges on the first $n-q$ rows, the last q rows, and the n rows between the $(jn-q+1)^{th}$ and $((j+1)n-q)^{th}$ rows for $1 \le j \le N$.

Let $\mathbf{F} = \begin{pmatrix} \tilde{\mathbf{f}}_0 \\ \tilde{\mathbf{f}}_1 \\ \vdots \\ \tilde{\mathbf{f}}_N \end{pmatrix}$, where $\tilde{\mathbf{f}}_j \in \mathbb{R}^n$ for all j. To solve $A\mathbf{W} = \mathbf{F}$ for $\mathbf{W} \in (\mathbb{R}^n)^{N+1}$, we first solve $L\mathbf{V} = \tilde{\mathbf{F}}$ for $\mathbf{V} \in (\mathbb{R}^n)^{N+1}$. Namely, we use the forward substitution $L_{0,0}\mathbf{v}_0 = \tilde{\mathbf{f}}_0$ and

$L_{i,i}\mathbf{v}_i = \tilde{\mathbf{f}}_i - L_{i,i-1}\mathbf{v}_{i-1}$ for $i = 1, 2, \ldots, N$. Then we solve $U\mathbf{W} = \mathbf{V}$ for $\mathbf{W} \in (\mathbb{R}^n)^{N+1}$. Namely, we use the backward substitution $U_{N,N}\mathbf{w}_N = \mathbf{v}_N$ and $U_{i,i}\mathbf{w}_i = \mathbf{v}_i - U_{i,i+1}\mathbf{w}_{i+1}$ for $i = N-1$, $N-2, \ldots, 0$.

If the boundary condition are only partially separable, namely $B_a^{[b]} \neq 0$ in (14.2.11) of Section 14.2.3, we can rewrite A as

$$A = \begin{pmatrix} B_a^{[a]} & 0 & \ldots & 0 & 0 \\ C_{0,0} & C_{0,1} & \ldots & 0 & 0 \\ 0 & C_{1,1} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & C_{N-1,N-1} & C_{N-1,N} \\ B_a^{[b]} & 0 & \ldots & 0 & B_b^{[b]} \end{pmatrix}, \tag{14.3.20}$$

where $B_a^{[a]}$ is a $(n-q)\times n$ matrix, and $B_b^{[b]}$ and $B_a^{[b]}$ are $q \times n$ matrices.

A in (14.3.20) is of the form

$$A = \begin{pmatrix} A_0 & C_0 & 0 & \ldots & 0 & 0 \\ B_1 & A_1 & C_1 & \ldots & 0 & 0 \\ 0 & B_2 & A_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & A_{N-1} & C_{N-1} \\ C_N & 0 & 0 & \ldots & B_N & A_N \end{pmatrix},$$

where each block is a $n \times n$ matrix, the q last rows of the $n \times n$ matrices B_j and the $n-q$ first rows of the $n \times n$ matrices C_j are null. If A is non-singular, we can express it as $A = LU$, where

$$L = \begin{pmatrix} \mathrm{Id} & 0 & \ldots & 0 & 0 \\ L_{1,0} & \mathrm{Id} & \ldots & 0 & 0 \\ 0 & L_{2,1} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{N,0} & L_{N,1} & \ldots & L_{N,N-1} & \mathrm{Id} \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} U_{0,0} & U_{0,1} & \ldots & 0 & 0 \\ 0 & U_{1,1} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & U_{N-1,N-1} & U_{N-1,N} \\ 0 & 0 & \ldots & 0 & U_{N,N} \end{pmatrix}.$$

The $n \times n$ matrices $L_{i,j}$ and $U_{i,j}$ satisfy

$$\begin{aligned} U_{0,0} &= A_0 \\ U_{0,1} &= C_0 \\ L_{N,0}U_{0,0} &= C_N \\ &\left.\begin{aligned} L_{i,i-1}U_{i-1,i-1} &= B_i \\ U_{i,i} &= A_i - L_{i,i-1}U_{i-1,i} \\ U_{i,i+1} &= C_i \\ L_{N,i}U_{i,i} &= -L_{N,i-1}U_{i-1,i} \end{aligned}\right\} \quad , \quad i = 1, 2, \ldots, N-2 \\ L_{N-1,N-2}U_{N-2,N-2} &= B_{N-1} \\ U_{N-1,N-1} &= A_{N-1} - L_{N-1,N-2}U_{N-2,N-1} \end{aligned}$$

$$\begin{aligned}
U_{N-1,N} &= C_{N-1} \\
L_{N,N-1}U_{N-1,N-1} &= B_N - L_{N,N-2}U_{N-2,N-1} \\
U_{N,N} &= A_N - L_{N,N-1}U_{N-1,N}
\end{aligned}$$

To solve $A\mathbf{W} = \mathbf{F}$ for $\mathbf{W} \in (\mathbb{R}^n)^{N+1}$, we first solve $L\mathbf{V} = \tilde{\mathbf{F}}$ for $\mathbf{V} \in (\mathbb{R}^n)^{N+1}$ using forward substitution as we did for the separable boundary conditions case above; the last substitution is now $\mathbf{v}_N = \tilde{\mathbf{f}}_N - \sum_{j=0}^{N-1} L_{N,j}\mathbf{v}_j$. The second step is to solve $U\mathbf{W} = \mathbf{V}$ for $\mathbf{W} \in (\mathbb{R}^n)^{N+1}$ using backward substitution as for the separable boundary conditions case above.

As for the separable case, the LU decomposition of A above is not unique. To determine a unique LU decomposition, it is standard to require that $L_{i,i} = \mathrm{Id}$ for $0 \le i \le N$ as we did above. It is proved in [22] that this decomposition can be obtained with the same restrictions on the row interchanges as above if h is small enough, the linear boundary value problem (14.3.4) with the boundary conditions expressed as in (14.2.11) has a unique solution, and

$$\begin{aligned}
L_{i,h}(\mathbf{W}) &= \sum_{k=0}^{N} C_{i,k}\,\mathbf{w}_k = F_{i,h}(f) \quad , \quad 0 \le i < N \\
\mathbf{w}_0 &= \mathbf{y}_v
\end{aligned}$$

is consistent and stable for the initial value problem

$$\begin{aligned}
L(\mathbf{y}(t)) &= \mathbf{y}'(t) - A(t)\mathbf{y}(t) = f(t) \quad , \quad a \le t \le b \\
\mathbf{y}(a) &= \mathbf{y}_c
\end{aligned}$$

Code 14.3.18 (One-Step Finite Difference Method for Linear Boundary Value Problems)

To approximate the solution of the boundary value problem $y' - A(t) = f(f)$ with $B_a y(a) + B_b y(b) = y_c$ for $a \le t \le b$. We consider the intervals $[t_i, t_{i+1}]$ for $0 \le i < N$ with $t_i = a + ih$ and $h = (b-a)/N$.
Input: The vector valued function $F : [a,b] \times \mathbb{R} \to \mathbb{R}^n$ (F in the code below).
The $n \times n$ matrix valued function $C_{i,i}$ defined on $[a,b] \times \mathbb{R}$ (Ci in the code below).
The $n \times n$ matrix valued function $C_{i,i+1}$ defined on $[a,b] \times \mathbb{R}$ (Cii in the code below).
The $(n-q) \times n$ matrix $B_a^{[a]}$ (Baa in the code below).
The $q \times n$ matrix $B_a^{[b]}$ (Bab in the code below).
The $q \times n$ matrix $B_b^{[b]}$ (Bbb in the code below).
The (column) vector $y_c \in R^n$ (yc in the code below).
The number $N > 2$ of partitions of $[a,b]$.
The endpoints a and b of the interval of integration $[a,b]$.
Output: The $n \times (N+1)$ matrix ww that contains the approximations $\mathbf{w}_i$ of $\mathbf{y}(t_i)$ and the vector tt that contains t_i for $0 \le i \le N$.

```
function [tt,ww] = linearFDM(F,Ci,Cii,Baa,Bab,Bbb,yc,N,a,b)
  n = length(yc);
```

```
q = size(Bbb,1);
nmq = n - q;
h = (b-a)/N;

% We construct the matrix A and the vector F
A = zeros(n,n,N+1);        % A(:,:,i) = A_{i-1}  for 1 <= i <= N+1
B = zeros(n,n,N);          % B(:,:,i) = B_i      for 1 <= i <= N
C = zeros(n,n,N+1);        % C(:,:,i) = C_{i-1}  for 1 <= i <= N+1
FF = zeros(n,N+1);         % F(:,:,i) = \tilde{f}_{i-1}  for 1 <= i <= N+1

A(1:nmq,:,1) = Baa;
C(nmq+1:n,:,N+1) = Bab;
t = (N-1)*h;
Civ = Ci(t,h);
Ciiv = Cii(t+h,h);
B(1:nmq,:,N) = Civ(q+1:n,:);
A(1:nmq,:,N+1) = Ciiv(q+1:n,:);
A(nmq+1:n,:,N+1) = Bbb;
FF(1:nmq,1) = yc(1:nmq,1);
tt = [];
for i=1:1:N
  t = a + (i-1)*h;
  tt = [tt t];
  Civ = Ci(t,h);
  Ciiv = Cii(t+h,h);
  A(nmq+1:n,:,i) = Civ(1:q,:);
  C(nmq+1:n,:,i) = Ciiv(1:q,:);
  B(1:nmq,:,i) = Civ(q+1:n,:);
  A(1:nmq,:,i+1) = Ciiv(q+1:n,:);
  v = F(t,h);
  FF(nmq+1:n,i) = v(1:q,1);
  FF(1:nmq,i+1) = v(q+1:n,1);
end
FF(nmq+1:n,N+1) = yc(nmq+1:n,1);
tt = [tt b];

% We construct the matrices L nad U
Ud = zeros(n,n,N+1);       % Ud(:,:,i) = U_{i-1,i-1} , 1 <= i <= N+1
Uu = zeros(n,n,N);         % Uu(:,:,i) = U_{i-1,i} , 1 <= i <= N
Ll = zeros(n,n,N);         % Ll(:,:,i) = L_{i,i-1} , 1 <= i <= N
Lr = zeros(n,n,N);         % Lr(:,:i) = L_{N.i-1} ,  1 <= i <= N-1

Ud(:,:,1) = A(:,:,1);
Uu(:,:,1) = C(:,:,1);
% Lr(:,:,1) = C(:,:,N+1)*inv(Ud(:,:,1));
Lr(:,:,1) = linsolve(Ud(:,:,1)',C(:,:,N+1)')';
```

```
  for i=1:1:N-2
    % Ll(:,:,i) = B(:,:,i)*inv(Ud(:,:,i));
    Ll(:,:,i) = linsolve(Ud(:,:,i)',B(:,:,i)')';
    Ud(:,:,i+1) = A(:,:,i+1) - Ll(:,:,i)*Uu(:,:,i);
    Uu(:,:,i+1) = C(:,:,i+1);
    % Lr(:,:,i+1) = -Lr(:,:,i)*Uu(:,:,i)*inv(Ud(:,:,i+1));
    Lr(:,:,i+1) = -linsolve(Ud(:,:,i+1)', Uu(:,:,i)'*Lr(:,:,i)')';
  end
  % Ll(:,:,N-1) = B(:,:,N-1)*inv(Ud(:,:,N-1));
  Ll(:,:,N-1) = linsolve(Ud(:,:,N-1)',B(:,:,N-1)')';
  Ud(:,:,N) = A(:,:,N) - Ll(:,:,N-1)*Uu(:,:,N-1);
  Uu(:,:,N) = C(:,:,N);
  % Ll(:,:,N) = (B(:,:,N) - Lr(:,:,N-1)*Uu(:,:,N-1))*inv(Ud(:,:,N));
  Ll(:,:,N) = linsolve(Ud(:,:,N)',(B(:,:,N) - Lr(:,:,N-1)*Uu(:,:,N-1))')';
  Ud(:,:,N+1) = A(:,:,N+1) - Ll(:,:,N)*Uu(:,:,N);

  % We now solve the system A W = F
  % First, we solve L V = F
  V = zeros(n,N+1);
  V(:,1) = FF(:,1);
  for i=2:1:N+1
    V(:,i) = FF(:,i) - Ll(:,:,i-1)*V(:,i-1);
  end
  for i=1:1:N-1
    V(:,N+1) = V(:,N+1) - Lr(:,:,i)*V(:,i);
  end

  % Second, we solve U W = V
  W = zeros(n,N+1);
  % W(:,N+1) = inv(Ud(:,:,N+1))*V(:,N+1);
  W(:,N+1) = linsolve(Ud(:,:,N+1),V(:,N+1));
  for i=N:-1:1
    % W(:,i) = inv(Ud(:,:,i))*(V(:,i) - Uu(:,:,i)*W(:,i+1));
    W(:,i) = linsolve(Ud(:,:,i),V(:,i) - Uu(:,:,i)*W(:,i+1));
  end
  ww = W;
end
```

Example 14.3.19 (**Example 14.2.6 Continued**)
Recall that the boundary value problem was $\mathbf{y}'(t) = A(t)\mathbf{y}(t) + f(t)$ with $B_a\mathbf{y}(0) + B_b\mathbf{y}(1) = \mathbf{y}_c$, where

$$\mathbf{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}\ ,\ A(t) = \begin{pmatrix} 0 & 1 \\ 4 & 0 \end{pmatrix}\ ,\ f(t) = \begin{pmatrix} 0 \\ -3e^t \end{pmatrix}\ ,\ B_a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\ ,\ B_b = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{y}_c = \begin{pmatrix} 1 \\ e \end{pmatrix}\ .$$

We use the previous code with the trapezoidal method to numerically solve the boundary value problem. We need to set

$$C_{i,i} = -\frac{1}{h}\,\mathrm{Id} - \frac{1}{2}\,A(t_i) = \begin{pmatrix} -1/h & -1/2 \\ -2 & -1/h \end{pmatrix}\ ,\ C_{i,i+1} = \frac{1}{h}\,\mathrm{Id} - \frac{1}{2}\,A(t_i+h) = \begin{pmatrix} 1/h & -1/2 \\ -2 & 1/h \end{pmatrix}\ ,$$

$$F_{i,h}(f) = \frac{1}{2}\left(f(t_i) + f(t_i+h)\right) = \begin{pmatrix} 0 \\ -3\left(e^{t_1} + e^{t_i+h}\right)/2 \end{pmatrix}\ ,\ B_a^{[a]} = \begin{pmatrix} 1 & 0 \end{pmatrix}\ ,\ B_a^{[b]} = \begin{pmatrix} 0 & 0 \end{pmatrix}\ ,$$

$$B_b^{[b]} = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{y}_c = \begin{pmatrix} 1 \\ e^1 \end{pmatrix}\ .$$

If we use the code above with $N = 100$, we find the following approximations of the solution.

i	t_i	$w_{1,i}$	$w_{2,i}$
0	0	1	0.9999829
1	0.01	1.0100501	1.0100332
2	0.02	1.0202012	1.0201844
3	0.03	1.0304543	1.0304377
4	0.04	1.0408104	1.0407940
5	0.05	1.0512707	1.0512544
6	0.06	1.0618361	1.0618199
7	0.07	1.0725076	1.0724916
8	0.08	1.0832864	1.0832706
9	0.09	1.0941736	1.0941578
10	0.10	1.1051701	1.1051545
$\vdots$	$\vdots$	$\vdots$	$\vdots$

i	t_i	$w_{1,i}$	$w_{2,i}$
90	0.90	2.4596020	2.4595917
91	0.91	2.4843215	2.4843112
92	0.92	2.5092895	2.5092793
93	0.93	2.5345084	2.5344983
94	0.94	2.5599807	2.5599707
95	0.95	2.5857091	2.5856991
96	0.96	2.6116960	2.6116861
97	0.97	2.6379449	2.6379343
98	0.98	2.6644560	2.6644463
99	0.99	2.6912343	2.6912248
100	1.00	2.7182818	2.7182723

where $w_{1,i} \approx y_{1,i} = y_1(t_i)$ and $w_{2,i} \approx y_{2,i} = y_2(t_i)$ for all i. All the approximations have at least 5-digit accuracy. These results are not as good as those that we found with the parallel shooting method but we have to keep in mind that the trapezoidal method is of order 2 while the classical forth order Runge-Kutta method that we have used for the parallel shooting if of order four. There are finite difference schemes that can give better results. However, those schemes will generally not be one-step scheme and, therefore, the matrix A will not be as nice as the one that we have for one-step schemes.

Here is the code used to call the finite difference method.

Code 14.3.20

```
format long
F = @(t,h) [ 0 ; -3*(exp(t)+exp(t+h))/2 ];
Ci = @(t,h) [ -1/h -1/2 ; -2 -1/h];
Cii = @(t,h) [ 1/h -1/2 ; -2  1/h];
Baa = [ 1 0 ];
Bab = [ 0 0 ];
Bbb = [ 1 0 ];
yc = [ 1 ; exp(1) ];
N = 100;
[t,w] = linearFDM(F,Ci,Cii,Baa,Bab,Bbb,yc,N,0,1)
```

♣

Unfortunately, the trapezoidal method cannot be used to numerically solve the boundary value problem of Example 14.2.8. The matrix A generated by this method is singular. Other finite difference schemes must be used. We will not develop finite difference methods with more than one-step. This is left to the adventurous readers.

14.3.3 Finite Difference Methods for Non-Linear Boundary Value Problems

We consider finite difference methods of the form (14.3.2) that may be used to approximate the solution of a boundary value problem of the form (14.3.1).

In this subsection, we first study the stability of finite difference methods like (14.3.2) for boundary value problems like (14.3.1). At the end, we will give a constructive proof of the existence of a numerical approximation to the solution of (14.3.1).

Consider the following linear boundary value problem obtained from the linearisation of (14.3.1).

$$\begin{aligned} L^{[\mathbf{y}]}(\mathbf{u}(t)) = \mathbf{u}'(t) - \mathrm{D}_{\mathbf{y}} f(t,\mathbf{y}(t))\mathbf{u}(t) = \mathbf{0} \quad &, \quad a \leq t \leq b \\ \mathrm{D}_{\mathbf{y}_1} g(\mathbf{y}(a),\mathbf{y}(b))\mathbf{u}(a) + \mathrm{D}_{\mathbf{y}_2} g(\mathbf{y}(a),\mathbf{y}(b))\mathbf{u}(b) = \mathbf{0} & \end{aligned} \tag{14.3.21}$$

where $\mathbf{y}$ is the solution of the boundary value problem (14.3.1).

We also consider the finite difference method obtained from the linearisation of (14.3.2); namely,

$$\begin{aligned} L_{i,h}^{[\mathbf{W}]}(\mathbf{U}) = \sum_{k=0}^{N} C_{i,k}(\mathbf{W})\,\mathbf{u}_k = \mathbf{0} \quad &, \quad 0 \leq i < N \\ B_a(\mathbf{W})\,\mathbf{u}_0 + B_b(\mathbf{W})\,\mathbf{u}_n = \mathbf{0} & \end{aligned} \tag{14.3.22}$$

where

$$C_{i,k}(\mathbf{W}) = \mathrm{D}_{\mathbf{y}_k} P_{i,h}(\mathbf{Y})\Big|_{\mathbf{Y}=\mathbf{W}} \quad , \quad B_a(\mathbf{W}) = \mathrm{D}_{\mathbf{y}_1} g(\mathbf{y}_0,\mathbf{y}_N)\Big|_{\mathbf{Y}=\mathbf{W}} \; ,$$

$$B_b[\mathbf{W}) = \mathrm{D}_{\mathbf{y}_2} g(\mathbf{y}_0,\mathbf{y}_N)\Big|_{\mathbf{Y}=\mathbf{W}} \quad , \quad \mathbf{W} = \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1} \quad \text{and} \quad \mathbf{U} = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1} \; .$$

let

$$A(\mathbf{Z}) = \begin{pmatrix} B_a(\mathbf{Z}) & 0 & \ldots & B_b(\mathbf{Z}) \\ C_{0,0}(\mathbf{Z}) & C_{1,1}(\mathbf{Z}) & \ldots & C_{0,N}(\mathbf{Z}) \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-1,0}(\mathbf{Z}) & C_{N,1}(\mathbf{Z}) & \ldots & C_{N-1,N}(\mathbf{Z}) \end{pmatrix} \quad \text{with} \quad \mathbf{Z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{pmatrix} \in (\mathbb{R}^n)^{N+1} \; .$$

Theorem 14.3.21

Suppose that $\mathbf{y}$ is an isolate solution of (14.3.1). Let $\{t_i\}_{i=0}^N$ be a partition of $[a,b]$ satisfying our standard conditions, and let $\mathbf{Y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{pmatrix}$, where $\mathbf{y}_i = \mathbf{y}(t_i)$ for $0 \leq i \leq N$.

Suppose that the finite difference method

$$\begin{aligned} L_{i,h}^{[\mathbf{Y}]}(\mathbf{U}) &= \mathbf{0} \quad , \quad 0 \leq i < N \\ \mathbf{u}_0 &= \mathbf{y}_c \in \mathbb{R}^n \end{aligned} \tag{14.3.23}$$

is stable and consistent for the initial value problem

$$\begin{aligned} L^{[\mathbf{y}]}(\mathbf{u}(t)) &= \mathbf{0} \quad , \quad a \leq t \leq b \\ \mathbf{u}(a) &= \mathbf{y}_c \in \mathbb{R}^n \end{aligned} \tag{14.3.24}$$

Suppose that $L_{i,h}^{[\mathbf{W}]}$ is Lipschitz continuous with respect to $\mathbf{W}$ in a neighbourhood of $\mathbf{Y}$; namely, there exist constants $\delta > 0$, $K_L > 0$ and $h_0 > 0$ such that

$$\|L_{i,h}^{[\mathbf{W}]} - L_{i,h}^{[\tilde{\mathbf{W}}]}\|_\infty \leq K_L \|\mathbf{W} - \tilde{\mathbf{W}}\|_\infty \tag{14.3.25}$$

for all $\mathbf{W}$ and $\tilde{\mathbf{W}}$ in

$$S_\delta(\mathbf{Y}) = \left\{ \mathbf{Z} \in (\mathbb{R}^n)^{N+1} : \|\mathbf{z}_i - \mathbf{y}_i\|_\infty < \delta \quad \text{for} \quad 0 \leq i \leq N \right\}$$

and all $h < h_0$. Moreover, in this context, suppose that

$$\begin{aligned} &\max\left\{ \|B_a(\mathbf{W}) - B_a(\tilde{\mathbf{W}})\|_\infty, \|B_b(\mathbf{W}) - B_b(\tilde{\mathbf{W}})\|_\infty \right\} \\ &\quad \leq \frac{K_L}{2} \max\left\{ \|\mathbf{w}_0 - \tilde{\mathbf{w}}_0\|_\infty, \|\mathbf{w}_N - \tilde{\mathbf{w}}_N\|_\infty \right\} \end{aligned} \tag{14.3.26}$$

for all $\mathbf{W}, \tilde{\mathbf{W}} \in S_\delta(\mathbf{Y})$ and all $h < h_0$. Then, if δ is small enough, $A_h(\mathbf{Z})$ has a uniformly bounded inverse for all $\mathbf{Z} \in S_\delta(\mathbf{Y})$ and h small enough.
Moreover, 14.3.22 is stable for the linear boundary value problem 14.3.21 [2].

Proof.
Since (14.3.23) is stable and consistent for (14.3.24), we get from Corollary 14.3.16 that (14.3.22) is stable and consistent for (14.3.21). It follows from Proposition 14.3.13 that there exist $h_1 > 0$ and $K > 0$ such that $(A(\mathbf{Y}))^{-1}$ exists and $\|(A(\mathbf{Y}))^{-1}\| \leq K$ for $h < h_1$. We may assume that $h_0 < h_1$ by shrinking h_0 if necessary.

Hence, if $\mathbf{Z}$ is closed enough to $\mathbf{Y}$, (14.3.25) and (14.3.26) imply that $A(\mathbf{Z})$ is as closed as we want of the invertible matrix $A(\mathbf{Y})$ independently of $h < h_0$. If we choose δ_0 small

[2]It can be shown that this implies that (14.3.2) is stable for the nonlinear boundary value problem (14.3.1).

enough to have $\|A(\mathbf{Y}) - A(\mathbf{Z})\|_\infty \left\|(A(\mathbf{Y}))^{-1}\right\|_\infty < 1/2$ for $\delta < \delta_0$, then it follows from the Banach Lemma that $(A(\mathbf{Z}))^{-1}$ exists. Moreover, from

$$\begin{aligned}
\|\,(A(\mathbf{Z}))^{-1}\,\|_\infty - \|\,(A(\mathbf{Y}))^{-1}\,\|_\infty &\leq \|\,(A(\mathbf{Z}))^{-1} - (A(\mathbf{Y}))^{-1}\,\|_\infty \\
&= \|\,(A(\mathbf{Y}))^{-1}\,(A(\mathbf{Y}) - A(\mathbf{Z}))\,(A(\mathbf{Z}))^{-1}\,\|_\infty \\
&\leq \underbrace{\|\,(A(\mathbf{Y}))^{-1}\,\|_\infty\,\|\,(A(\mathbf{Y}) - A(\mathbf{Z}))\,\|_\infty}_{<1/2}\,\|\,(A(\mathbf{Z}))^{-1}\,\|_\infty \ ,
\end{aligned}$$

we get

$$\|\,(A(\mathbf{Z}))^{-1}\,\|_\infty \leq \frac{\|\,(A(\mathbf{Y}))^{-1}\,\|_\infty}{1 - \|\,(A(\mathbf{Y}))^{-1}\,\|_\infty\,\|A(\mathbf{Y}) - A(\mathbf{Z})\|_\infty} \leq 2\|\,(A(\mathbf{Y}))^{-1}\,\|_\infty$$

for all $\mathbf{Z} \in S_\delta(\mathbf{Y})$ with $h < h_0$ and $\delta < \delta_0$. We could have used Corollary 3.2.7 to directly draw the previous conclusion. So $(A(\mathbf{Z}))^{-1}$ is uniformly bounded for h and δ small enough.

Let

$$\Psi(\mathbf{W}) = \begin{pmatrix} g(\mathbf{w}_0, \mathbf{w}_N) \\ P_{0,h}(\mathbf{W}) \\ \vdots \\ P_{N-1,h}(\mathbf{W}) \end{pmatrix} . \tag{14.3.27}$$

Then

$$\Psi(\mathbf{Z}) - \Psi(\tilde{\mathbf{Z}}) = A(\mathbf{Z}, \tilde{\mathbf{Z}})\left(\mathbf{Z} - \tilde{\mathbf{Z}}\right) \ , \tag{14.3.28}$$

where

$$A(\mathbf{Z}, \tilde{\mathbf{Z}}) \equiv \int_0^1 A(s\mathbf{Z} + (1-s)\tilde{\mathbf{Z}})\, \mathrm{d}s$$

since $P_{i,h}$ and g are assumed to be continuously differentiable, and $A(\mathbf{W}) = \mathrm{D}_{\mathbf{Z}}\Psi(\mathbf{Z})\big|_{\mathbf{Z}=\mathbf{W}}$.

Moreover, from (14.3.25) and (14.3.26), we get

$$\begin{aligned}
&\|A(\mathbf{Z}, \tilde{\mathbf{Z}}) - A(\mathbf{Y})\|_\infty \\
&\qquad \leq \max\left\{B_a(\mathbf{W}) - B_a(\tilde{\mathbf{W}})\|_\infty + \|B_b(\mathbf{W}) - B_b(\tilde{\mathbf{W}})\|_\infty, \max_{0\leq i<N} \|L_{i,h}^{[\mathbf{W}]} - L_{i,h}^{[\tilde{\mathbf{W}}]}\|_\infty\right\} \leq K_L\delta
\end{aligned}$$

for $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ in $S_\delta(\mathbf{Y})$ and $h < h_0$. Thus, if δ is small enough to have

$$\|A(\mathbf{Z}, \tilde{\mathbf{Z}}) - A(\mathbf{Y})\|_\infty \left\|(A(\mathbf{Y}))^{-1}\right\|_\infty \leq \delta K_L K < 1 \ ,$$

then it follows from the Banach Lemma that $\left(A(\mathbf{Z}, \tilde{\mathbf{Z}})\right)^{-1}$ exists and, as we have shown above for $(A(\mathbf{Z}))^{-1}$,

$$\|\left(A(\mathbf{Z}, \tilde{\mathbf{Z}})\right)^{-1}\|_\infty \leq \frac{\|\,(A(\mathbf{Y}))^{-1}\,\|_\infty}{1 - \|\,(A(\mathbf{Y}))^{-1}\,\|_\infty\,\|A(\mathbf{Y}) - A(\mathbf{Z}, \tilde{\mathbf{Z}})\|_\infty} \leq \frac{K}{1 - \delta K_L K}$$

for $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ in $S_\delta(\mathbf{Y})$ with δ and h small enough.

The stability of (14.3.2) follows from

$$\left(\mathbf{Z} - \tilde{\mathbf{Z}}\right) = \left(A(\mathbf{Z}, \tilde{\mathbf{Z}})\right)^{-1}\left(\Psi(\mathbf{Z}) - \Psi(\tilde{\mathbf{Z}})\right)$$

by taking the norm on both sides and using the uniform upper bound on $\left(A(\mathbf{Z},\tilde{\mathbf{Z}})\right)^{-1}$ for $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ in $S_\delta(\mathbf{Y})$ with δ and h small enough. ∎

We now show how we can use the Newton Method to find an approximation of the solution of (14.3.1) if $\mathbf{Z}^{[0]} \in S_\delta(\mathbf{Y})$ is chosen appropriately, where δ is given in the previous theorem. More precisely, we show that if h and $\delta_0 < \delta$ are small enough and $\mathbf{Z}^{[0]} \in S_{\delta_0}(\mathbf{Y})$, then the sequence $\{\mathbf{Z}^{[k]}\}_{k=0}^\infty$ defined by

$$A(\mathbf{Z}^{[k]})\left(\mathbf{Z}^{[k+1]} - \mathbf{Z}^{[k]}\right) = -\Psi(\mathbf{Z}^{[k]}) \quad , \quad k = 0,1,2,\ldots \tag{14.3.29}$$

stays in $S_{\delta_0}(\mathbf{Y})$ and converges toward a solution $\mathbf{W}$ of (14.3.2).

We can rewrite (14.3.29) as

$$\begin{aligned} L_{i,h}^{[\mathbf{Z}^{[k]}]}\left(\mathbf{Z}^{[k+1]} - \mathbf{Z}^{[k]}\right) &= -P_{i,h}(\mathbf{Z}^{[k]}) \quad , \quad 0 \le i < N \\ B_a(\mathbf{Z}^{[k]})\left(\mathbf{z}_0^{[k+1]} - \mathbf{z}_0^{[k]}\right) + B_b(\mathbf{Z}^{[k]})\left(\mathbf{z}_N^{[k+1]} - \mathbf{z}_N^{[k]}\right) &= -g(\mathbf{z}_0^{[k]}, \mathbf{z}_N^{[k]}) \end{aligned}$$

for $k = 0, 1, 2, \ldots$

The following theorem will be useful shortly. A proof of this theorem can be found in [25].

Theorem 14.3.22 (Newton-Kantorovich)

Suppose that $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a sufficiently differentiable function and let $Q(\mathbf{x}) = \mathrm{D}_{\mathbf{x}}\phi(\mathbf{x})$. Suppose that there exists γ such that

$$\|Q(\mathbf{x}) - Q(\tilde{\mathbf{x}})\|_\infty \le \gamma \|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty \tag{14.3.30}$$

for all $\mathbf{x}, \tilde{\mathbf{x}}$ in an open convex set $D \subset \mathbb{R}^n$. Suppose also that, for some $\mathbf{x}_0 \in D$, there exist constants α and β such that

$$\|Q^{-1}(\mathbf{x}_0)\|_\infty \le \beta \ , \tag{14.3.31}$$

$$\|Q^{-1}(\mathbf{x}_0)\phi(\mathbf{x}_0)\|_\infty \le \alpha \ , \tag{14.3.32}$$

and

$$\alpha\beta\gamma < \frac{1}{2} \ . \tag{14.3.33}$$

let

$$\delta_\pm = \frac{1 \pm \sqrt{1 - 2\alpha\beta\gamma}}{\beta\gamma} \ .$$

If $S_{\delta_-}(\mathbf{x}_0) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x_0}\|_\infty < \delta_-\} \subset D$, then, the sequence $\{\mathbf{x}_k\}_{k=0}^\infty$ generated by

$$Q(\mathbf{x}_k)\,(\mathbf{x}_{k+1} - \mathbf{x}_k) = -\phi(\mathbf{x}_k) \quad , \quad k = 0,1,2,\ldots$$

remains in $S_{\delta_-}(\mathbf{x}_0)$ for all k and converges quadratically to the unique root of ϕ in $S_{\delta_+}(\mathbf{x}_0) \cap D$.

If D is very large so that $S_{\delta_-}(\mathbf{x}_0) \subset D$ is "almost" always satisfied, then the previous theorem does not require the explicit knowledge of the exact root of ϕ to determine conditions to get a converging sequence $\{\mathbf{x}^{[k]}\}_{k=0}^{\infty}$ to a root of ϕ.

Theorem 14.3.23

Suppose that all the hypothesis of Theorem 14.3.21 are satisfied. Suppose that the local truncation error of (14.3.2) with respect to (14.3.1) is of order $p > 0$. Then, there exist $0 < \delta_0 < \delta$ (δ given Theorem 14.3.21) and $h_0 > 0$ such that (14.3.2) has a solution $\mathbf{W}$ in $S_\delta(\mathbf{Y})$ if $h \leq h_0$. The Newton Method (14.3.29) with $\mathbf{Z}^{[0]}$ such that $\mathbf{Z}^{[0]} \in S_{\delta_0}(\mathbf{Y})$ can be used to approximate this solution. The convergence is quadratic.

Proof.
We prove that the hypotheses of Newton-Kantorovich Theorem are satisfied. We replace $\mathbf{x}_k$ by $\mathbf{Z}^{[k]}$, ϕ by Ψ, $Q(\mathbf{Z})$ by $A(\mathbf{Z}) = \mathrm{D}_{\mathbf{W}}\Psi(\mathbf{W})\big|_{\mathbf{W}=\mathbf{Z}}$ and D by $S_\delta(\mathbf{Y}) = \{\mathbf{Z} : \|\mathbf{Z} - \mathbf{Y}\|_\infty < \delta\}$ in Newton-Kantorovich Theorem, where $\mathbf{Y}$ and δ are given Theorem 14.3.21.

From (14.3.25) and (14.3.26), we have that

$$\|A(\mathbf{W}) - A(\tilde{\mathbf{W}})\| \leq K_L \|\mathbf{W} - \tilde{\mathbf{W}}\|_\infty$$

for all $\mathbf{W}$ and $\tilde{\mathbf{W}}$ in $S_\delta(\mathbf{Y})$ for δ given in the statement of Theorem 14.3.21. So (14.3.30) is satisfied with with $\gamma = K_L$.

Suppose that $\delta_0 < \delta$. We will precise the value of δ_0 later. Let $\mathbf{Z}^{[0]}$ be any element in $S_{\delta_0}(\mathbf{Y})$.

Proceeding as in the proof of Theorem 14.3.21, we have that $\|A_h^{-1}(\mathbf{Z}, \tilde{\mathbf{Z}})\|_\infty \leq K/(1 - \delta_0 K_L K)$ for $\mathbf{Z}, \tilde{\mathbf{Z}} \in S_{\delta_0}(\mathbf{Y})$ if h is small enough and $\delta_0 < \delta$. Recall that $\delta K_L K < 1$. If we take $\mathbf{Z} = \tilde{\mathbf{Z}} = \mathbf{Z}^{[0]}$, we get that (14.3.31) is satisfies with $\beta = K/(1 - \delta_0 K_L K)$; namely, $\|A^{-1}(\mathbf{Z}^{[0]})\|_\infty \leq \beta$.

Since

$$A^{-1}(\mathbf{Z}^{[0]})A(\mathbf{Z}^{[0]}, \mathbf{Y}) = Id + A^{-1}(\mathbf{Z}^{[0]})\left(A(\mathbf{Z}^{[0]}, \mathbf{Y}) - A(\mathbf{Z}^{[0]})\right) ,$$

we get

$$\|A^{-1}(\mathbf{Z}^{[0]})A(\mathbf{Z}^{[0]}, \mathbf{Z})\|_\infty \leq 1 + \left(\frac{K}{1 - \delta_0 K_L K}\right) K_L \delta_0$$

for $\mathbf{Z}^{[0]} \in S_{\delta_0}(\mathbf{Y})$. Moreover, from (14.3.28), we get

$$A^{-1}(\mathbf{Z}^{[0]})\Psi(\mathbf{Z}^{[0]}) = A^{-1}(\mathbf{Z}^{[0]})\left(\Psi(\mathbf{Y}) + A(\mathbf{Z}^{[0]}, \mathbf{Y})\left(\mathbf{Z}^{[0]} - \mathbf{Y}\right)\right) .$$

Thus,

$$\|A^{-1}(\mathbf{Z}^{[0]})\Psi(\mathbf{Z}^{[0]})\|_\infty \leq \left(\frac{K}{1 - \delta_0 K_L K}\right) K_0 h^p + \left(1 + \left(\frac{K}{1 - \delta_0 K_L K}\right) K_L \delta_0\right) \delta_0$$

for some constant K_0 and h small enough. The factor $K_0 h^p$ comes from the assumption that the method is of order p. Thus (14.3.32) is satisfied with $\alpha = \left(\frac{K}{1 - \delta_0 K_L K}\right) K_0 h^p + \left(1 + \left(\frac{K}{1 - \delta_0 K_L K}\right) K_L \delta_0\right) \delta_0$.

We need to choose δ_0 and h small enough to satisfy (14.3.33); namely,

$$\alpha\beta\gamma = \left(\left(\frac{K}{1-\delta_0 K_L K}\right) K_0 h^p + \left(1 + \left(\frac{K}{1-\delta_0 K_L K}\right) K_L \delta_0\right)\delta_0\right)\left(\frac{K}{1-\delta_0 K_L K}\right) K_L < \frac{1}{2}$$

We also need to choose δ_0 small enough to have $S_{\delta_-}(\mathbf{Z}^{[0]}) \subset B_\delta(\mathbf{Y})$ to be able to apply Newton-Kantorovich Theorem.

First, we may assume that KK_L is large enough to have $1/(KK_L) < \delta/4$. Hence,

$$\delta_- < \delta_+ = \frac{1+\sqrt{1-2\alpha\beta\gamma}}{\beta\gamma} < \frac{2}{\beta\gamma} < \frac{2}{KK_L} < \frac{\delta}{2}\ . \tag{14.3.34}$$

Moreover, we may assume that δ_0 is small enough to have $\delta_0 K_L K < 1/2$. Then, we select h such that

$$\left(\frac{K}{1-\delta_\delta K_L K}\right)^2 K_L K_0 h^p < 4K^2 K_L K_0 h^p < \frac{1}{4}\ . \tag{14.3.35}$$

We choose δ_o small enough to have

$$\delta_0\left(1 + \left(\frac{K}{1-\delta_0 K_L K}\right) K_L \delta_0\right)\left(\frac{K}{1-\delta_0 K_L K}\right) K_L < 2\delta_0\left(1 + 2KK_l\delta_0\right) KK_L < \frac{1}{4}\ .$$

Combine with (14.3.35), this implies that (14.3.33) is satisfied.

Finally, we choose δ_0 small enough to have

$$2\delta_0 KK_L < 1 - \sqrt{1-2K^2K_LK_0h^p}\ .$$

We then have that

$$\delta_0 < \frac{1-\sqrt{1-2K^2K_LK_0h^p}}{2KK_L} \le \frac{1-\sqrt{1-2\alpha\beta\gamma}}{\beta\gamma} = \delta_-\ .$$

If follow from (14.3.34) that $\delta_0 < \delta_- < \delta/2$. Thus for any $\mathbf{Z}^{[0]} \in S_{\delta_0}(\mathbf{Y})$, we have $S_{\delta_-}\left(\mathbf{Z}^{[0]}\right) \subset S_\delta(\mathbf{Y})$ as required. ∎

14.3.4 Collocation and Implicit Runge-Kutta

We consider a simple case to illustrate how collocation and Runge-Kutta methods can be use to develop method to solve boundary value problems.

In this subsection, we consider the partition $a = t_0 < t_1 < \ldots < t_N = b$ of the interval $[a, b]$ with $t_{i+1} - t_i = h_i$ for $i = 0, 1, \ldots, N-1$. Let $0 \le \theta_0 < \theta_1 < \ldots < \theta_{J-1} < \theta_J \le 1$. We subdivide each interval $[t_i, t_{i+1}]$ with a partition $t_i \le t_{i,0} < t_{i,1} < \ldots < t_{i,J-1} < t_{i,J} \le t_{i+1}$ where $t_{i,j} = t_i + \theta_j h_i$ for $j = 0, 1, \ldots, J$.

From now on, we assume that $\theta_0 = 0$ and $\theta_J = 1$ to simplify the presentation.

We approximate the solution $\mathbf{y}$ of (14.3.1) on $[t_i, t_{i+1}]$ by a polynomial mapping $\mathbf{p}_i(t)$ of degree $J+1$ such that

$$\mathbf{p}_i'(t_{i,j}) = f(t_{i,j}, \mathbf{p}_i(t_{i,j})) \quad , \quad 0 \leq i < N \text{ and } 0 \leq j \leq J \tag{14.3.36}$$

$$\mathbf{0} = g(\mathbf{p}_0(t_{0,0}), \mathbf{p}_{N-1}(t_{N-1,J})) \tag{14.3.37}$$

$$\mathbf{p}_i(t_{i,0}) = \mathbf{p}_{i-1}(t_{i-1,J}) \quad , \quad 0 < i < N \tag{14.3.38}$$

Condition (14.3.38) implies that $\mathbf{p} : [a,b] \to \mathbb{R}^n$ defined by $\mathbf{p}(t) = \mathbf{p}_i(t)$ for $t_i \leq t \leq t_{i+1}$ is a piecewise continuous polynomial mapping.

(14.3.36) and (14.3.38) are exactly the conditions that we have used with the collocation method to derive implicit Runge-Kutta Method in Section 13.4.1.

If we use Proposition 13.4.11, in particular (13.4.4), we get

$$\mathbf{p}(t_{i,j}) = \mathbf{p}(t_{i,0}) + h_i \sum_{m=0}^{J} \beta_{j,m} K_{i,m} \quad , \quad 0 \leq i \leq N \text{ and } 0 \leq j \leq J \ ,$$

where

$$K_{i,m} = f(t_{i,m}, \mathbf{p}(t_{i,m})) \quad , \quad 0 \leq i \leq N \text{ and } 0 \leq m \leq J \ ,$$

and

$$\beta_{j,m} = \int_{\theta_0}^{\theta_j} \ell_m(\theta)\,\mathrm{d}\theta = \int_{\theta_0}^{\theta_j} \left(\prod_{\substack{k=0 \\ k\neq m}}^{J} \frac{\theta - \theta_k}{\theta_m - \theta_k} \right) \mathrm{d}\theta \quad , \quad 0 \leq j, m \leq J \ .$$

The solution $\mathbf{y}$ of (14.3.1) may therefore be approximated by the scheme

$$\mathbf{w}_{i,j} = \mathbf{w}_{i,0} + h_i \sum_{m=0}^{J} \beta_{j,m} f(t_{i,m}, \mathbf{w}_{i,m}) \quad , \quad 0 \leq i < N \text{ and } 0 \leq j \leq J \tag{14.3.39}$$

$$\mathbf{0} = g(\mathbf{w}_{0,0}, \mathbf{w}_{N-1,J}) \tag{14.3.40}$$

We hope that $\mathbf{w}_{i,j} \approx y(t_{i,j})$ for all i and j.

Remark 14.3.24

1. Note that (14.3.39) is one step method, from t_i to t_{i+1}, of an implicit Runge-Kutta method. Since we assume that $\theta_0 = 0$, we have that $\beta_{0,m} = 0$ for all m. Since we assume that $\theta_J = 1$, we have that $\beta_{J,m} = \gamma_m$ for all m. Thus (14.3.39) with $j = J$ yields (13.4.6).

2. The Runge-Kutta method (14.3.39) is stable for the initial value problem

$$\mathbf{y}'(t) = f(t, \mathbf{y}(t))$$
$$\mathbf{y}(a) = \mathbf{y}_c \in \mathbb{R}^n$$

3. The local truncation error of the Runge-Kutta method (14.3.39) is at least of order J.

♠

Theorem 14.3.25

1. Suppose that the polynomial mappings $\mathbf{p}_i$ satisfy (14.3.36), (14.3.37) and (14.3.38). Then, $\mathbf{w}_{i,j} = \mathbf{p}(t_{i,j})$ satisfy (14.3.39) and (14.3.40).

2. Suppose that the $\mathbf{w}_{i,j}$ satisfy (14.3.39) and (14.3.40). For $0 \leq i < N$, let $\mathbf{p}_i$ be the unique interpolating polynomial mapping of degree $J+1$ at the points $(t_{i,j}, \mathbf{w}_{i,j})$ for $0 \leq j \leq J$ that satisfies $\mathbf{p}_i'(t_{i,0}) = f(t_{i,0}, \mathbf{p}_i(t_{i,0}))$. Then, the $\mathbf{p}_i$ satisfy (14.3.36), (14.3.37) and (14.3.38).

Proof.
1) Since $\mathbf{p}_i$ is a polynomial mapping of degree $J+1$, $\mathbf{p}_i'$ is a polynomial mapping of degree J. Since the quadrature formula

$$\int_{\theta_0}^{\theta_j} q(\theta)\,\mathrm{d}\theta = \sum_{m=0}^{J} \beta_{j,m} q(\theta_m) \tag{14.3.41}$$

with $0 \leq m \leq J$ is true for polynomial q of degree up to at least J by construction, we have

$$\begin{aligned}
\mathbf{p}_i(t_{i,j}) &= \mathbf{p}_i(t_{i,0}) + \int_{t_{i,0}}^{t_{i,j}} \mathbf{p}_i'(t)\,\mathrm{d}t = \mathbf{p}_i(t_{i,0}) + h_i \int_{\theta_0}^{\theta_j} \mathbf{p}_i'(t_i + \theta h_i)\,\mathrm{d}\theta \\
&= \mathbf{p}_i(t_{i,0}) + h_i \sum_{m=0}^{J} \beta_{j,m}\,\mathbf{p}_i'(t_{i,m}) \\
&= \mathbf{p}_i(t_{i,0}) + h_i \sum_{m=0}^{J} \beta_{j,m}\, f(t_{i,m}, \mathbf{p}_i(t_{i,m})) \quad , \quad 0 \leq j \leq J \ ,
\end{aligned}$$

where the last equality comes from (14.3.36). So, we get (14.3.39) with $\mathbf{w}_{i,j} = \mathbf{p}_i(t_{i,j})$ for all $0 \leq i < N$ and $0 \leq j \leq J$. Obviously, (14.3.37) implies (14.3.40).

2) Again, since $\mathbf{p}_i$ is a polynomial of degree $J+1$, $\mathbf{p}_i'$ is a polynomial of degree J. Since (14.3.41) is true for polynomial q of degree up to at least J by construction, we get

$$\begin{aligned}
\mathbf{w}_{i,j} - \mathbf{w}_{i,0} &= \mathbf{p}(t_{i,j}) - \mathbf{p}(t_{i,0}) = \int_{t_{i,0}}^{t_{i,j}} \mathbf{p}_i'(t)\,\mathrm{d}t \\
&= h_i \int_{\theta_0}^{\theta_j} \mathbf{p}_i'(t_i + \theta h_i)\,\mathrm{d}\theta = h_i \sum_{m=0}^{J} \beta_{j,m} \mathbf{p}_i'(t_{i,m}) \quad , \quad 0 \leq j \leq J \ .
\end{aligned}$$

Moreover, from (14.3.39), we have that

$$\mathbf{w}_{i,j} - \mathbf{w}_{i,0} = h_i \sum_{k=0}^{J} \beta_{j,k}\, f(t_{i,k}, \mathbf{w}_{i,k}) = h_i \sum_{m=0}^{J} \beta_{j,m}\, f(t_{i,m}, \mathbf{p}_i(t_{i,m})) \quad , \quad 0 \leq j \leq J \ .$$

Thus,

$$\sum_{m=0}^{J} \beta_{j,m} \left(\mathbf{p}_i'(t_{i,m}) - f(t_{i,m}, \mathbf{p}_i(t_{i,m}))\right) = \mathbf{0} \quad , \quad 0 \leq j \leq J \ .$$

Since we assume that

$$\mathbf{p}_i'(t_{i,0}) = f(t_{i,0}, \mathbf{p}_i(t_{i,0})) \quad , \quad 0 \leq i < N \ ,$$

namely that (14.3.36) with $j = 0$ is satisfied, we have

$$\sum_{m=1}^{J} \beta_{j,m} \left(\mathbf{p}_i'(t_{i,m}) - f(t_{i,m}, \mathbf{p}(t_{i,m}))\right) = \mathbf{0} \quad , \quad 1 \leq j \leq J \ .$$

This can be rewritten as a linear system of the form $B\mathbf{X} = \mathbf{0}$, where

$$B = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \ldots & \beta_{1,J} \\ \beta_{2,1} & \beta_{2,2} & \ldots & \beta_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{J,1} & \beta_{J,2} & \ldots & \beta_{J,J} \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{p}_i'(t_{i,1}) - f(t_{i,1}, \mathbf{p}(t_{i,1})) \\ \mathbf{p}_i'(t_{i,2}) - f(t_{i,2}, \mathbf{p}(t_{i,2})) \\ \vdots \\ \mathbf{p}_i'(t_{i,J}) - f(t_{i,J}, \mathbf{p}(t_{i,J})) \end{pmatrix} .$$

Since B is an invertible matrix[3], the only solution is $\mathbf{X} = \mathbf{0}$. Thus (14.3.36) with $1 \leq j \leq J$ must also be satisfied.

(14.3.38) is satisfied because $\mathbf{p}_i(t_{i,0}) = \mathbf{w}_{i,0} = \mathbf{w}_{i-1,J} = \mathbf{p}_{i-1}(t_{i-1,J})$ for $1 < i < N$. Finally, (14.3.37) is satisfied because $g(\mathbf{p}_0(t_{0,0}), \mathbf{p}(t_{N-1,J})) = g(\mathbf{w}_{0,0}, \mathbf{w}_{N-1,J}) = \mathbf{0}$. ∎

Remark 14.3.26
There exist collocation methods with smoother polynomial mappings than the piecewise continuous polynomial mappings that we have considered here. These methods are more efficient. ♠

14.4 Analytic Eigenvalue Problems

This section is based on Keller's lectures [22] and Ascher et al.'s book [2].

Eigenvalue problems are the major source of boundary value problems. It is therefore important to say a few words about eigenvalues problems.

We consider the generalised eigenvalue problem

$$\begin{aligned} \mathbf{y}'(t) - A(t,\lambda)\mathbf{y}(t) &= \mathbf{0} \quad , \quad a \leq t \leq b \\ B_a(\lambda)\mathbf{y}(a) + B_b(\lambda)\mathbf{y}(b) &= \mathbf{0} \end{aligned} \tag{14.4.1}$$

where $B_a(\lambda)$ and $B_b(\lambda)$ are analytic in λ, and $A(t,\lambda)$ is analytic in λ uniformly in $t \in [a,b]$. Moreover, we assume that $\operatorname{rank}\begin{pmatrix} B_a(\lambda) & B_b(\lambda)\end{pmatrix} = n$ for all λ. This is a necessary condition for the existence of a solution for (14.4.1).

Remark 14.4.1
The eigenvalue problem (14.4.1) has partially separated boundary conditions if

$$B_a(\lambda) = \begin{pmatrix} B_a^{[a]}(\lambda) \\ B_a^{[b]}(\lambda) \end{pmatrix} \quad \text{and} \quad B_b(\lambda) = \begin{pmatrix} 0 \\ B_b^{[b]}(\lambda) \end{pmatrix} ,$$

where $B_a^{[a]}(\lambda)$ is a $(n-q)\times n$ matrix, and $B_a^{[b]}(\lambda)$ and $B_b^{[b]}(\lambda)$ are $q \times n$ matrices. ♠

[3]Use (14.3.41) with $q(\theta) = \theta^m$ for $1 \leq m \leq J$ to show that B is an invertible Vandermonde matrix.

The **fundamental solution** associated to (14.4.1) is the solution of

$$\begin{aligned} Y'(t,\lambda) - A(t,\lambda)Y(t,\lambda) &= 0 \quad , \quad a \le t \le b \\ Y(a,\lambda) &= \mathrm{Id} \end{aligned}$$

for all λ. It can be shown that $Y(t,\lambda)$ is analytic in λ uniformly in $t \in [a,b]$.

Every solution of (14.4.1) is of the form

$$y(t,\lambda) = Y(t,\lambda)\mathbf{y}_c \ ,$$

where $\mathbf{y}_c$ is a solution of

$$Q(\lambda)\mathbf{y}_c \equiv \left(B_a(\lambda) + B_b(\lambda)Y(b,\lambda)\right)\mathbf{y}_c = \mathbf{0} \ . \tag{14.4.2}$$

Theorem 14.4.2

For the generalised eigenvalue problem (14.4.1), only one of the following two cases is possible.

1. Every λ is an eigenvalue of (14.4.1).

2. There are at most a countable number of distinct eigenvalues λ_k with no accumulation point.

In the second case, λ_k has geometric multiplicity

$$r_k = \dim \ker(Q(\lambda_k)) \le n \ .$$

If we have partially separated boundary conditions as in Remark 14.4.1, then $r_k \le q$.

Proof.
λ is an eigenvalue of (14.4.1) if (14.4.2) has a non-trivial solution, and (14.4.2) has a non-trivial solution if and only if $\det(Q(\lambda)) = 0$. Thus, the geometric multiplicity of λ is the dimension of the solution space of (14.4.2); namely, the dimension of the kernal of $Q(\lambda)$.

Since $\det(Q(\lambda))$ is an analytic function of λ, $\det(Q(\lambda)) = 0$ for all λ if and only if there is an accumulation point for the zeros of $\det(Q(\lambda))$; namely, the eigenvalues of (14.4.1). Thus, we have either (1) or (2).

For the partially separated boundary conditions case

$$Q(\lambda_k)\mathbf{y}_c = \begin{pmatrix} B_a^{[a]}(\lambda_k) \\ B_a^{[b]}(\lambda_k) + B_b^{[b]}(\lambda_k)Y(b,\lambda_k) \end{pmatrix}\mathbf{y}_c = \mathbf{0} \ ,$$

where $B_a^{[a]}(\lambda_k)$ has full rank $n-q$. Thus $r_k = \dim\ \ker(Q(\lambda_k)) \le q$. ∎

14.5 Exercises

Question 14.1
Show that the midpoint scheme of Example 14.3.9 is consistent and stable for the linear boundary value problem (14.3.4), and therefore convergent.

Chapter 15

Finite Difference Methods

Compare to solving partial differential equations numerically, solving ordinary differential equations is very simple. All the numerical methods behave similarly with all types of ordinary differential equations. The only exception is with stiff ordinary differential equations.

The situation for partial differential equations is a lot more complex. There are no numerical methods that can be used for all types of partial differential equations to generate an accurate numerical solution. This is even true for the three types of linear partial differential equations of order two with constant coefficients; namely the parabolic, elliptic and hyperbolic equations. In fact, as we will show, hyperbolic partial differential equations cannot be solved accurately with finite difference schemes without imposing strict constrains on the step sizes. Some other methods, like finite element methods, need to be used with such partial differential equations.

Suppose that u is the solution of a partial differential equation

$$P\left(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \ldots\right) = F(x,y)$$

on a domain

$$R = [a,b] \times [c,d] = \{(x,y) : a \leq x \leq b \text{ and } c \leq y \leq d\} \ ,$$

where P and F are "nice" functions. Choose N and M, two positive integers, and let $\Delta x = (b-a)/N$ and $\Delta y = (d-c)/M$. The set

$$R_\Delta = \{(x_i, y_j) : x_i = a + i\Delta x \text{ for } 0 \leq i \leq N \text{ and } y_j = c + j\Delta y \text{ for } 0 \leq j \leq M\}$$

forms a **grid** of the domain D. Each point (x_i, y_j) is called a **mesh point**. The **step sizes** are the values of Δx and Δy [1]. A **numerical solution** of the partial differential equation is a set

$$\{w_{i,j} : 0 \leq i \leq N \text{ and } 0 \leq j \leq M\}$$

such that $w_{i,j} \approx u(x_i, y_j)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$.

[1] In our presentation, we assume that the distance between the x_i and the distance between the y_j are constant. Finite difference schemes could be developed for non-constant step sizes but this is for a more advanced text.

The goal of this chapter is to develop some **finite difference schemes or methods**; namely, some finite difference equations to compute the values $w_{i,j}$. The finite difference equations are obtained from the partial differential equations by substituting the partial derivatives in the partial differential equations by finite difference formulae approximating these partial derivatives.

The reader should not expect a complete listing of methods to solve partial differential equations. Only some basic partial differential equations and finite difference schemes are considered. There is however enough material to get a good understanding of the complexity and procedure to numerically solve partial differential equations.

15.1 Finite Difference Formulae

To develop finite difference schemes, we need to use finite difference formulae to approximate the partial derivatives of sufficiently differentiable functions. Let $u : \mathbb{R}^2 \to \mathbb{R}$ be a sufficiently differentiable function. We use Taylor expansions of u at (x_i, y_j) to derive finite difference formulae of the partial derivatives of u at (x_i, y_j). We provide below a few examples of the derivation of finite difference formulae. More finite difference formulae will be introduced later on.

15.1.1 First Order Derivatives

We begin by deriving a finite difference formula for $\dfrac{\partial u}{\partial x}$ at the mesh point (x_i, y_j). If we assume that u is of class C^2, we have

$$u(x_{i+1}, y_j) = u(x_i, y_j) + \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)(\Delta x)^2$$

for some $\zeta_{i,j} \in]x_i, x_{i+1}[$. So

$$\frac{\partial u}{\partial x}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - u(x_i, y_j)}{\Delta x} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\Delta x \ . \tag{15.1.1}$$

Since $\dfrac{\partial^2 u}{\partial x^2}$ is continuous on the close set R, there exists a constant $K > 0$ such that

$$\left| \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x, y) \right| < K$$

for $(x, y) \in R$. Hence,

$$\left| \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\Delta x \right| < K\Delta x$$

because $\zeta_{i,j} \in]x_i, x_{i+1}[$ and therefore $(\zeta_{i,j}, y_j) \in R$. We have shown that

$$\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - u(x_i, y_j)}{\Delta x} \tag{15.1.2}$$

and the truncation error $\dfrac{1}{2}\dfrac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\Delta x$ satisfies

$$\frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\,\Delta x = O(\Delta x)$$

for Δx near 0. The truncation error converges to zero as Δx converges to zero.

Instead of using the points (x_i, y_j) and (x_{i+1}, y_j) to derive a finite difference formula for $\dfrac{\partial u}{\partial x}$ at the mesh point (x_i, y_j), we could use (x_i, y_j) and (x_{i-1}, y_j).

If we assume that u is of class C^2, we have

$$u(x_{i-1}, y_j) = u(x_i, y_j) - \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)(\Delta x)^2$$

for some $\zeta_{i,j} \in]x_{i-1}, x_i[$. So

$$\frac{\partial u}{\partial x}(x_i, y_j) = \frac{u(x_i, y_j) - u(x_{i-1}, y_j)}{\Delta x} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\Delta x\ . \tag{15.1.3}$$

As we did above, if $\dfrac{\partial^2 u}{\partial x^2}$ is continuous on the close set R, we may assume that there exists a constant $K > 0$ such that

$$\left|\frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\,\Delta x\right| < K\Delta x\ .$$

Hence,

$$\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{u(x_i, y_j) - u(x_{i-1}, y_j)}{\Delta x} \tag{15.1.4}$$

and the truncation error $\dfrac{1}{2}\dfrac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\Delta x$ satisfies

$$\frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\zeta_{i,j}, y_j)\,\Delta x = O(\Delta x)$$

for Δx near 0.

To derive finite difference formulae which are "more accurate" than (15.1.2) (i.e. with a smaller truncation error when Δx approach 0), we need to consider Taylor expansion of order higher than two. For instance, if we assume that u is of class C^3, we have that

$$u(x_{i+1}, y_j) = u(x_i, y_j) + \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j)(\Delta x)^2 + \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(\zeta_{i,j}, y_j)(\Delta x)^3$$

and

$$u(x_{i-1}, y_j) = u(x_i, y_j) - \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j)(\Delta x)^2 - \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(\eta_{i,j}, y_j)(\Delta x)^3$$

for $\zeta_{i,j} \in]x_i, x_{i+1}[$ and $\eta_{i,j} \in]x_{i-1}, x_i[$. If we subtract the second equation from the first equation, we get

$$u(x_{i+1}, y_j) - u(x_{i-1}, y_j) = 2\frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{3!}\left(\frac{\partial^3 u}{\partial x^3}(\zeta_{i,j}, y_j) + \frac{\partial^3 u}{\partial x^3}(\eta_{i,j}, y_j)\right)(\Delta x)^3\ .$$

Hence

$$\frac{\partial u}{\partial x}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2\Delta x} - \frac{1}{12}\left(\frac{\partial^3 u}{\partial x^3}(\zeta_{i,j}, y_j) + \frac{\partial^3 u}{\partial x^3}(\eta_{i,j}, y_j)\right)(\Delta x)^2 . \quad (15.1.5)$$

We have found that

$$\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2\Delta x}$$

and, because $\dfrac{\partial^3 u}{\partial x^3}$ is continuous, we can show as we did for the previous finite difference formulae that the truncation error $\dfrac{1}{12}\left(\dfrac{\partial^3 u}{\partial x^3}(\zeta_{i,j}, y_j) + \dfrac{\partial^3 u}{\partial x^3}(\eta_{i,j}, y_j)\right)(\Delta x)^2$ satisfies

$$\frac{1}{12}\left(\frac{\partial^3 u}{\partial x^3}(\zeta_{i,j}, y_j) + \frac{\partial^3 u}{\partial x^3}(\eta_{i,j}, y_j)\right)(\Delta x)^2 = O\left((\Delta x)^2\right)$$

for Δx near 0.

15.1.2 Second Order Derivatives

Using the Taylor Expansion Theorem, we may also derive finite difference formulae for second order derivatives. If u is of class C^4, we have

$$\begin{aligned} u(x_{i+1}, y_j) = u(x_i, y_j) &+ \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j)(\Delta x)^2 + \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(x_1, y_j)(\Delta x)^3 \\ &+ \frac{1}{4!}\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j)(\Delta x)^4 \end{aligned}$$

and

$$\begin{aligned} u(x_{i-1}, y_j) = u(x_i, y_j) &- \frac{\partial u}{\partial x}(x_i, y_j)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j)(\Delta x)^2 - \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(x_i, y_j)(\Delta x)^3 \\ &+ \frac{1}{4!}\frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)(\Delta x)^4 \end{aligned}$$

for $\zeta_{i,j} \in]x_i, x_{i+1}[$, and $\eta_{i,j} \in]x_{i-1}, x_i[$. If we add these two equations, we get

$$\begin{aligned} &u(x_{i+1}, y_j) + u(x_{i-1}, y_j) \\ &\quad = 2u(x_i, y_j) + \frac{\partial^2 u}{\partial x^2}(x_i, y_j)(\Delta x)^2 + \frac{1}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)\right)(\Delta x)^4 . \end{aligned}$$

Solving for $\dfrac{\partial^2 u}{\partial x^2}(x_i, y_j)$, we get

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x_i, y_j) = &\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{(\Delta x)^2} \\ &- \frac{1}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)\right)(\Delta x)^2 . \end{aligned} \quad (15.1.6)$$

We have found that

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{(\Delta x)^2} \tag{15.1.7}$$

and, because $\dfrac{\partial^4 u}{\partial x^4}$ is continuous, we can show as we did before that the truncation error $\dfrac{1}{4!}\left(\dfrac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j) + \dfrac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)\right)(\Delta x)^2$ satisfies

$$\frac{1}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)\right)(\Delta x)^2 = O\big((\Delta x)^2\big)$$

for Δx near 0.

We can proceed likewise to find other finite difference formulae.

15.2 Explicit and Implicit Schemes

We develop finite difference schemes for the three types of linear partial differential equations of order two with constant coefficients. More precisely, we develop finite difference schemes for one representative of each of these types of partial differential equations. This will be enough to understand the peculiarities of each type.

1. For the parabolic equations, we consider the heat equation $\dfrac{\partial u}{\partial t} = c^2 \dfrac{\partial^2 u}{\partial x^2}$.
2. For the elliptic equations, we consider the Dirichlet equation $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f$.
3. For the hyperbolic equation, we consider the wave equation $\dfrac{\partial^2 u}{\partial t^2} = c^2 \dfrac{\partial^2 u}{\partial x^2}$.

15.2.1 Parabolic Equations

15.2.1.1 An Explicit Scheme

We consider the heat equation with forcing

$$\frac{\partial u}{\partial t} - c^2 \frac{\partial^2 u}{\partial x^2} = f(x, t) \quad , \quad 0 < x < L \text{ and } 0 < t < T \ , \tag{15.2.1}$$

with the boundary conditions

$$u(0, t) = h_0(t) \text{ and } u(L, t) = h_L(t) \quad , \quad 0 \le t \le T \ , \tag{15.2.2}$$

and the initial condition

$$u(x, 0) = g(x) \quad , \quad 0 \le x \le L \ , \tag{15.2.3}$$

where $g(0) = h_0(0)$ and $g(L) = h_L(0)$. The forcing is provided by the function f.

We develop a finite difference scheme for the heat equation with forcing given in (15.2.1), (15.2.2) and (15.2.3).

Given two integers $N \geq 2$ and $M \geq 1$, we set $\Delta x = L/N$ $\Delta t = T/M$, $x_i = i\Delta x$, $t_j = j\Delta t$ and $u_{i,j} = u(x_i, t_j)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$. From (15.1.1) and (15.1.6), we get

$$\begin{aligned}\frac{u_{i,j+1} - u_{i,j}}{\Delta t} - \frac{1}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \rho_{i,j})\,\Delta t - c^2\,\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \\ + \frac{c^2}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, t_j) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, t_j)\right)(\Delta x)^2 = f(x_i, t_j)\end{aligned} \tag{15.2.4}$$

for $\rho_{i,j} \in]t_j, t_{j+1}[$, $\zeta_{i,j} \in]x_{i-1}, x_{i+1}[$ and $\eta_{i,j} \in]x_{i-1}, x_{i+1}[$. For Δt and Δx small, we have

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} - c^2\,\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \approx f(x_i, t_j)\ .$$

This suggests the following finite difference equation.

$$\frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2\,\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} = f(x_i, t_j) \tag{15.2.5}$$

for $0 < i < N$ and $0 \leq j < M$. The boundary conditions impose $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \leq j \leq M$. The initial condition imposes $w_{i,0} = g(x_i)$ for $0 \leq i \leq N$.

Following some simple algebra, we get the following finite difference scheme to approximate the solution of the heat equation with forcing in (15.2.1).

Algorithm 15.2.1

$$w_{i,j+1} - w_{i,j} - \alpha\,(w_{i+1,j} - 2w_{i,j} + w_{i-1,j}) = f(x_i, t_j)\Delta t$$

for $1 < i < N$ and $0 \leq j < M$, where $\alpha = \dfrac{c^2\Delta t}{(\Delta x)^2}$, $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \leq j \leq M$, and $w_{i,0} = g(x_i)$ for $0 \leq i \leq N$.

This scheme is illustrated in Figure 15.1. It can be expressed as a linear system $A\mathbf{w} = \mathbf{B}$. The (column) vector $\mathbf{w}$ is defined by

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} \quad \text{with} \quad \mathbf{w}_j = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{N-1,j} \end{pmatrix}$$

for $0 < j \leq M$. The matrix A is a $(N-1)M \times (N-1)M$ matrix of the form

$$A = \begin{pmatrix} \mathrm{Id} & 0 & 0 & \dots & 0 & 0 \\ K & \mathrm{Id} & 0 & \dots & 0 & 0 \\ 0 & K & \mathrm{Id} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & K & \mathrm{Id} \end{pmatrix},$$

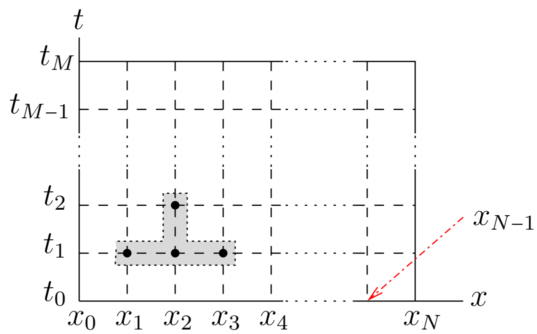


Figure 15.1: Schematic representation of the finite difference scheme given in Algorithm 15.2.1.

where Id is the $(N-1)\times(N-1)$ identity matrix and

$$K = \begin{pmatrix} -1+2\alpha & -\alpha & 0 & 0 & 0 & \dots & 0 & 0 \\ -\alpha & -1+2\alpha & -\alpha & 0 & 0 & \dots & 0 & 0 \\ 0 & -\alpha & -1+2\alpha & -\alpha & 0 & \dots & 0 & 0 \\ 0 & 0 & -\alpha & -1+2\alpha & -\alpha & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\alpha & -1+2\alpha \end{pmatrix} \tag{15.2.6}$$

is a $(N-1)\times(N-1)$ matrix. The (column) vector $\mathbf{B}$ is defined by

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_M \end{pmatrix}, \text{ where } \mathbf{B}_1 = \begin{pmatrix} w_{1,0} + \alpha\left(w_{0,0} - 2w_{1,0} + w_{2,0}\right) + f(x_1,t_0)\Delta t \\ w_{2,0} + \alpha\left(w_{1,0} - 2w_{2,0} + w_{3,0}\right) + f(x_2,t_0)\Delta t \\ w_{3,0} + \alpha\left(w_{2,0} - 2w_{3,0} + w_{4,0}\right) + f(x_3,t_0)\Delta t \\ \vdots \\ w_{N-1,0} + \alpha\left(w_{N-2,0} - 2w_{N-1,0} + w_{N,0}\right) + f(x_{N-1},t_0)\Delta t \end{pmatrix}$$

and

$$\mathbf{B}_j = \begin{pmatrix} \alpha w_{0,j-1} + f(x_1,t_{j-1})\Delta t \\ f(x_2,t_{j-1})\Delta t \\ \vdots \\ f(x_{N-2},t_{j-1})\Delta t \\ \alpha w_{N,j-1} + f(x_{N-1},t_{j-1})\Delta t \end{pmatrix}$$

for $2 \le j \le M$.

15.2.1.2 An Implicit Scheme, Crank-Nicolson Scheme

We will see in Section 15.3 that the finite difference scheme in Algorithm 15.2.1 is not really good. Another scheme often used to numerically solve the heat equation with forcing is due to Crank and Nicolson. Before introducing this scheme, we need to introduce the following finite difference scheme.

Using (15.1.4) and (15.1.7) at (x_i, t_{j+1}), we may write

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \rho_{i,j})\,\Delta t - c^2\,\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2} + \frac{c^2}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, t_{j+1}) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, t_{j+1})\right)(\Delta x)^2 = f(x_i, t_{j+1})$$

for $\rho_{i,j} \in]t_j, t_{j+1}[$, $\zeta_{i,j} \in]x_{i-1}, x_{i+1}[$ and $\eta_{i,j} \in]x_{i-1}, x_{i+1}[$. For Δt and Δx small, we have

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} - c^2\,\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{(\Delta x)^2} \approx f(x_i, t_{j+1})\ .$$

This suggests the following finite difference equation.

$$\frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2\,\frac{w_{i+1,j+1} - 2w_{i,j+1} + w_{i-1,j+1}}{(\Delta x)^2} = f(x_i, t_{j+1}) \tag{15.2.7}$$

for $0 < i < N$ and $0 \leq j < M$.

The Crank-Nicolson scheme comes from adding $1/2$ times (15.2.5) and $1/2$ times (15.2.7) to get the finite difference equation[2]

$$\begin{aligned}\frac{w_{i,j+1} - w_{i,j}}{\Delta t} &- \frac{c^2}{2}\left(\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} + \frac{w_{i+1,j+1} - 2w_{i,j+1} + w_{i-1,j+1}}{(\Delta x)^2}\right)\\ &= \frac{1}{2}\big(f(x_i, t_j) + f(x_i, t_{j+1})\big)\end{aligned} \tag{15.2.8}$$

for $0 < i < N$ and $0 \leq j < M$. The boundary conditions and initial condition still give $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \leq j \leq M$, and $w_{i,0} = g(x_i)$ for $0 \leq i \leq N$ respectively.

Following some simple algebra, we find the following finite difference scheme for the heat equation with forcing in (15.2.1).

Algorithm 15.2.2 (Crank-Nicholson)

$$\begin{aligned}w_{i,j+1} - w_{i,j} - \alpha\,(w_{i+1,j} - 2w_{i,j} + w_{i-1,j} + w_{i+1,j+1} - 2w_{i,j+1} + w_{i-1,j+1})\\ = \frac{1}{2}\big(f(x_i, t_j) + f(x_i, t_{j+1})\big)\Delta t\end{aligned}$$

for $0 < i < N$ and $0 \leq j < M$, where $\alpha = \dfrac{c^2 \Delta t}{2(\Delta x)^2}$, $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \leq j \leq M$, and $w_{i,0} = g(x_i)$ for $0 \leq i \leq N$.

This scheme is illustrated in Figure 15.2. This is an implicit scheme because the value of u at (x_i, t_{j+1}) is approximated using values of u at (x_{i-1}, t_{j+1}) and (x_{i+1}, t_{j+1}), two values for $t = t_{j+1}$ that are not explicitly known.

[2]more generally, we could have added λ times (15.2.5) and $1 - \lambda$ times (15.2.7) to get a family of finite difference scheme for $0 \leq \lambda \leq 1$.

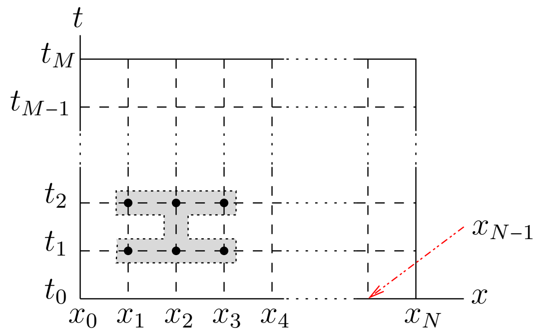


Figure 15.2: Schematic representation of the Crank-Nicolson scheme given in Algorithm 15.2.2.

As with the finite difference scheme in Algorithm 15.2.1, the Crank-Nicolson scheme can be expressed as a linear system $A\mathbf{w} = \mathbf{B}$. The (column) vector $\mathbf{w}$ is again defined by

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} \quad \text{with} \quad \mathbf{w}_j = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{N-1,j} \end{pmatrix}$$

for $0 < j \le M$. The matrix A is a $(N-1)M \times (N-1)M$ matrix of the form

$$A = \begin{pmatrix} J & 0 & 0 & \dots & 0 & 0 \\ K & J & 0 & \dots & 0 & 0 \\ 0 & K & J & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & K & J \end{pmatrix},$$

where

$$J = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & 0 & 0 & \dots & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 & \dots & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & -\alpha & 0 & \dots & 0 & 0 \\ 0 & 0 & -\alpha & 1+2\alpha & -\alpha & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\alpha & 1+2\alpha \end{pmatrix} \tag{15.2.9}$$

is a $(N-1) \times (N-1)$ matrix and K is defined in (15.2.6). The (column) vector $\mathbf{B}$ is the column matrix defined by

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_M \end{pmatrix},$$

where

$$\mathbf{B}_1 = \begin{pmatrix} w_{1,0} + \alpha\left(w_{0,0} - 2w_{1,0} + w_{2,0} + w_{0,1}\right) + \big(f(x_1,t_0) + f(x_1,t_1)\big)\Delta t/2 \\ w_{2,0} + \alpha\left(w_{1,0} - 2w_{2,0} + w_{3,0}\right) + \big(f(x_2,t_0) + f(x_2,t_1)\big)\Delta t/2 \\ w_{3,0} + \alpha\left(w_{2,0} - 2w_{3,0} + w_{4,0}\right) + \big(f(x_3,t_0) + f(x_3,t_1)\big)\Delta t/2 \\ \vdots \\ w_{N-1,0} + \alpha\left(w_{N-2,0} - 2w_{N-1,0} + w_{N,0} + w_{N,1}\right) + \big(f(x_{N-1},t_0) + f(x_{N-1},t_1)\big)\Delta t/2 \end{pmatrix}$$

and

$$\mathbf{B}_j = \begin{pmatrix} \alpha\left(w_{0,j-1} + w_{0,j}\right) + \big(f(x_1,t_{j-1}) + f(x_1,t_j)\Delta t/2 \\ \big(f(x_2,t_{j-1}) + f(x_2,t_j)\big)\Delta t/2 \\ \vdots \\ \big(f(x_{N-2},t_{j-1}) + f(x_{N-2},t_j)\big)\Delta t/2 \\ \alpha\left(w_{N,j-1} + w_{N,j}\right) + \big(f(x_{N-1},t_{j-1}) + f(x_{N-1},t_j)\big)\Delta t/2 \end{pmatrix}$$

for $2 \le j \le M$.

Code 15.2.3 (**Crank-Nicholson**)

To approximate the solution of the heat equation with forcing

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial y^2} + f(x,y)$$

on the region $R = [a,b] \times [t_1,t_2]$ with the initial condition $u(x,t_1) = g_b(x)$ for $a \le x \le b$, and the boundary conditions $u(a,t) = g_l(t)$ and $u(b,t) = g_r(t)$ for $t_1 \le t \le t_2$.
Input: The right hand side f.
The initial condition $g_b(x)$ when $t = t_1$.
The boundary condition $g_l(t)$ when $x = a$.
The boundary condition $g_r(t)$ when $x = b$.
The number of partitions N of $[a,b]$ with $\Delta x = (b-a)/N$.
The number of partitions M of $[t_1,t_2]$ with $\Delta t = (t_2 - t_1)/M$.
The endpoints $a < b$ of the x-interval for the domain R.
The endpoints $t_1 < t_2$ of the t-interval for the domain R.
Output: X contains the x-coordinates x_0, x_1, ..., x_N of the mesh points in the domain R.
T contains the t-coordinates t_0, t_1, ..., t_M of the mesh points in the domain R.
U contains the approximations of u at the mesh points. $U_{i,j} \approx u(x_{i-1},t_{j-1})$ for $1 \le i \le N+1$ and $1 \le j \le M+1$.

```
function [X,T,U] = crank_nicolson(f,gb,gl,gr,c,N,M,a,b,t1,t2)
  np1 = N + 1;
  mp1 = M + 1;
  U = repmat(NaN,np1,mp1);
  X = linspace(a,b,np1);
  T = linspace(t1,t2,mp1);

  % Initial condition
  for i=1:1:np1
```

```
    U(i,1) = gb(X(i));
end
% Boundary conditions
for j=1:1:mp1
    U(1,j) = gl(T(j));
    U(np1,j) = gr(T(j));
end

deltax = (b-a)/N;
deltat = (t2-t1)/M;
alpha = deltat*(c/deltax)^2/2;

nm1 = N - 1;
J = repmat(0,nm1,nm1);
K = repmat(0,nm1,nm1);
for i=1:1:nm1
    for j=i-1:1:i+1
        if (i == j)
            K(i,j) = -1 + 2 * alpha;
            J(i,j) = 1 + 2 * alpha;
        elseif ( j > 0 && j < N )
            K(i,j) = -alpha;
            J(i,j) = -alpha;
        end
    end
end

% We use the fact that the matrix Q is block lower triangular,
% and only the diagonal and lower diagonal contain non-trivial blocks.
nm2 = N - 2;
B = repmat(NaN,nm1,1);
B(1,1) = U(2,1) + alpha*(U(1,1) -2*U(2,1) + U(3,1) + U(1,2)) ...
         + (f(X(2),T(1)) + f(X(2),T(2)))*deltat/2;
for k=2:1:nm2
    B(k,1) = U(k+1,1) + alpha*(U(k,1) -2*U(k+1,1) + U(k+2,1)) ...
             + (f(X(k),T(1)) + f(X(k),T(2)))*deltat/2;
end
B(nm1,1) = U(N,1) + alpha*(U(nm1,1) -2*U(N,1) + U(np1,1) +U(np1,2)) ...
    + (f(X(N),T(1)) + f(X(N),T(2)))*deltat/2;
% Solve the system J U_2 = B_1 with matlab library
U(2:N,2) = linsolve(J,B);
% Solve the system J U_2 = B_1 with gauss()
% U(2:N,2) = gauss(J,B,1);

for j=2:1:M
    jp1 = j + 1;
```

```
        B(1,1) = alpha*(U(1,j) + U(1,jp1)) ...
                + (f(X(2),T(j)) + f(X(2),T(jp1)))*deltat/2;
        for k=2:1:nm2
            B(k,1) = (f(X(k),T(j)) + f(X(k),T(jp1)))*deltat/2;
        end
        B(nm1,1) = alpha*(U(np1,j) + U(np1,jp1)) ...
            + (f(X(N),T(j)) + f(X(N),T(jp1)))*deltat/2;
        % Solve the system J U_{j+1} = B_j - K U_j with matlab library
        U(2:N,jp1) = linsolve(J,B-K*U(2:N,j));
        % Solve the system J U_{j+1} = B_j - K U_j with gauss()
        % U(2:N,jp1) = gauss(J,B-K*U(2:N,j),1);
    end
end
```

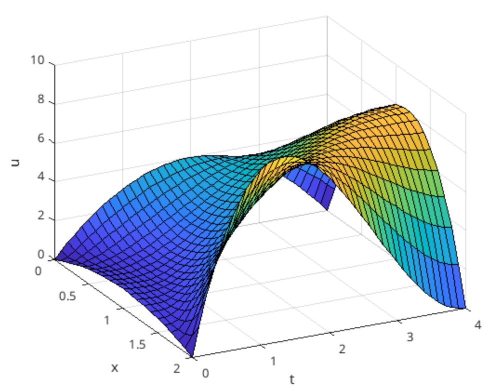
The comment in Remark 15.2.8 below is very relevant for the previous code because the matrix J can still be large.

Example 15.2.4
Use the Crank-Nicolson scheme to approximate the solution to the following heat equation with forcing.

$$\frac{\partial u}{\partial t} = 0.5^2 \frac{\partial u}{\partial x} 2 + xy;$$

on the domain $R = [0,2] \times [0,4]$ with the initial condition $u(x,0) = x(2-x)$ for $0 \le x \le 2$, and the boundary conditions $u(0,t) = t(4-t)$ and $u(2,t) = t(4-t)^2$ for $0 \le t \le 4$.

With the matlab code below, we got the following graph for the approximation of the solution u.



Code 15.2.5

```
f = @(x,y) x.*y;
gb = @(x) x.*(2-x);
gl = @(y) y.*(4-y);
gr = @(y) y.*(4-y).^2;
c = 0.5;
```

```
a = 0;
b = 2;
t1 = 0;
t2 = 4;
N = 20;
M = 40;

[X,T,U] = crank_nicolson(f,gb,gl,gr,c,N,M,a,b,t1,t2);
[XX,TT] = meshgrid(X,T);

% We need to transpose the matrix U because meshgrid()
% transposes the coordinates.
surf(XX,TT,U');
xlabel('x')
ylabel('t')
zlabel('u')
```

♣

15.2.2 Elliptic Equations

We consider the Dirichlet equation

$$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \tag{15.2.10}$$

on the set $R = \{(x,y) : 0 \leq x \leq a \ , \ 0 \leq y \leq b\}$ with the boundary conditions

$$u\big|_{\partial R} = g \ .$$

We assume that $f : R \to \mathbb{R}$ and $g : \partial R \to \mathbb{R}$ are continuous functions.

Given two integers $N \geq 2$ and $M \geq 2$, we set $\Delta x = a/N$, $\Delta y = b/M$, $x_i = i\Delta x$, $y_j = j\Delta y$ and $u_{i,j} = u(x_i, y_j)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$. From (15.1.6) in terms of x and y, we get

$$\begin{split}\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} - \frac{1}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\zeta_{i,j}, y_j) + \frac{\partial^4 u}{\partial x^4}(\eta_{i,j}, y_j)\right)(\Delta x)^2 \\ + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} - \frac{1}{4!}\left(\frac{\partial^4 u}{\partial y^4}(x_i, \mu_{i,j}) + \frac{\partial^4 u}{\partial y^4}(x_i, \nu_{i,j})\right)(\Delta y)^2 = f(x_i, y_j)\end{split} \tag{15.2.11}$$

for $\zeta_{i,j}, \eta_{i,j} \in]x_{i-1}, x_{i+1}[$ and $\mu_{i,j}, \nu_{i,j} \in]y_{j-1}, y_{j+1}[$. For Δx and Δy small, we have

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} \approx f(x_i, y_j) \ .$$

This suggests the following finite difference equation.

$$\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta y)^2} = f(x_i, y_j) \tag{15.2.12}$$

for $0 < i < N$ and $0 < j < M$. The boundary conditions impose

$$w_{i,0} = g(x_i, c) \text{ and } w_{i,M} = g(x_i, d) \text{ for } 0 \leq i \leq N \ ,$$

and

$$w_{0,j} = g(a, y_j) \text{ and } w_{a,j} = g(b, y_j) \text{ for } 0 \leq j \leq M \ .$$

We get the following finite difference scheme for the Dirichlet equation (15.2.10).

Algorithm 15.2.6

$$w_{i,j+1} - 2w_{i,j} + w_{i,j-1} + \alpha\,(w_{i+1,j} - 2w_{i,j} + w_{i-1,j}) = f(x_i, y_j)(\Delta y)^2$$

for $0 < i < N$ and $0 < j < M$, where $\alpha = \dfrac{(\Delta y)^2}{(\Delta x)^2}$, $w_{i,0} = g(x_i, c)$ and $w_{i,M} = g(x_i, d)$ for $0 \leq i \leq N$, and $w_{0,j} = g(a, x_j)$ and $w_{N,j} = g(b, x_j)$ for $0 \leq j \leq M$.

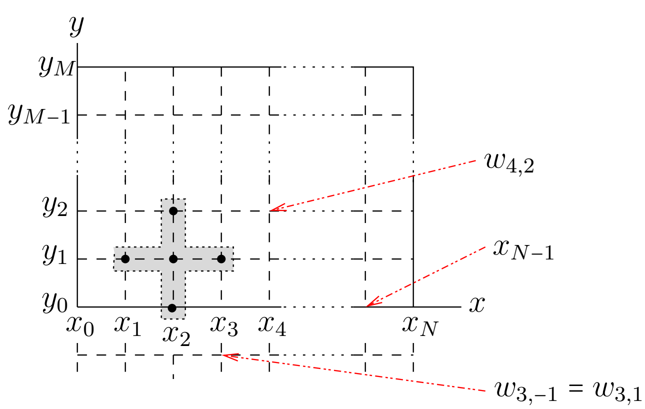


Figure 15.3: Schematic representation of the finite difference scheme given in Algorithm 15.2.6.

This finite difference scheme is illustrated in Figure 15.3.

It can be expressed as a linear system $A\mathbf{w} = \mathbf{B}$. The (column) vector $\mathbf{w}$ is defined by

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{M-1} \end{pmatrix} \quad \text{with} \quad \mathbf{w}_j = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{N-1,j} \end{pmatrix}$$

for $0 < j < M$. The matrix A is a $(N-1)(M-1) \times (N-1)(M-1)$ matrix of the form

$$A = \begin{pmatrix} J & \mathrm{Id} & 0 & 0 & \dots & 0 & 0 & 0 \\ \mathrm{Id} & J & \mathrm{Id} & 0 & \dots & 0 & 0 & 0 \\ 0 & \mathrm{Id} & J & \mathrm{Id} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathrm{Id} & J & \mathrm{Id} \\ 0 & 0 & 0 & 0 & \dots & 0 & \mathrm{Id} & J \end{pmatrix},$$

where Id is the $(N-1) \times (N-1)$ identity matrix and

$$J = \begin{pmatrix} -2-2\alpha & \alpha & 0 & 0 & 0 & \dots & 0 & 0 \\ \alpha & -2-2\alpha & \alpha & 0 & 0 & \dots & 0 & 0 \\ 0 & \alpha & -2-2\alpha & \alpha & 0 & \dots & 0 & 0 \\ 0 & 0 & \alpha & -2-2\alpha & \alpha & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha & -2-2\alpha \end{pmatrix} \tag{15.2.13}$$

is a $(N-1) \times (N-1)$ matrix. The vector $\mathbf{B}$ is defined by

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_{M-1} \end{pmatrix}, \text{where} \quad \mathbf{B}_1 = \begin{pmatrix} -w_{1,0} - \alpha w_{0,1} + f(x_1, y_1)(\Delta y)^2 \\ -w_{2,0} + f(x_2, y_1)(\Delta y)^2 \\ -w_{3,0} + f(x_3, y_1)(\Delta y)^2 \\ \vdots \\ -w_{N-1,0} - \alpha w_{N,1} + f(x_{N-1}, y_1)(\Delta y)^2 \end{pmatrix},$$

$$\mathbf{B}_j = \begin{pmatrix} -\alpha w_{0,j} + f(x_1, y_j)(\Delta y)^2 \\ f(x_2, y_j)(\Delta y)^2 \\ f(x_3, y_j)(\Delta y)^2 \\ \vdots \\ -\alpha w_{N,j} + f(x_{N-1}, y_j)(\Delta y)^2 \end{pmatrix}$$

for $1 < j < M-1$, and

$$\mathbf{B}_{M-1} = \begin{pmatrix} -w_{1,M} - \alpha w_{0,M-1} + f(x_1, y_{M-1})(\Delta y)^2 \\ -w_{2,M} + f(x_2, y_{M-1})(\Delta y)^2 \\ -w_{3,M} + f(x_3, y_{M-1})(\Delta y)^2 \\ \vdots \\ -w_{N-1,M} - \alpha w_{N,M-1} + f(x_{N-1}, y_{M-1})(\Delta y)^2 \end{pmatrix}.$$

Code 15.2.7

To approximate the solution of the Dirichlet equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

on the region $R = [a, b] \times [c, d]$ with the boundary conditions $u(x, c) = g_b(x)$ and

$u(x,d) = g_t(x)$ for $a \le x \le b$, and $u(a,y) = g_l(y)$ and $u(b,y) = g_r(y)$ for $c \le y \le d$.
Input: The right hand side f.
The boundary condition $g_b(x)$ when $y = c$.
The boundary condition $g_t(x)$ when $y = d$.
The boundary condition $g_l(y)$ when $x = a$.
The boundary condition $g_r(y)$ when $x = b$.
The number of partitions N of $[a,b]$ with $\Delta x = (b-a)/N$.
The number of partitions M of $[c,d]$ with $\Delta y = (d-c)/M$.
The endpoints $a < b$ of the x-interval for the domain R.
The endpoints $c < d$ of the y-interval for the domain R.
Output: X contains the x-coordinates x_0, x_1, ..., x_N of the mesh points in the domain R.
Y contains the y-coordinates y_0, y_1, ..., y_M of the mesh points in the domain R.
U contains the approximations of u at the mesh points. $U_{i,j} \approx u(x_{i-1}, y_{j-1})$ for $1 \le i \le N+1$ and $1 \le j \le M+1$.

```
function [X,Y,U] = dirichletS1(f,gb,gt,gl,gr,N,M,a,b,c,d)
  np1 = N + 1;
  mp1 = M + 1;
  U = repmat(NaN,np1,mp1);
  X = linspace(a,b,np1);
  Y = linspace(c,d,mp1);

  % Boundary conditions
  for i=1:1:np1
     U(i,1) = gb(X(i));
     U(i,mp1) = gt(X(i));
  end
  for j=1:1:mp1
     U(1,j) = gl(Y(j));
     U(np1,j) = gr(Y(j));
  end

  deltax = (b-a)/N;
  deltay = (d-c)/M;
  alpha = (deltay/deltax)^2;

  nm1 = N - 1;
  J = repmat(0,nm1,nm1);
  for i=1:1:nm1
     for j=i-1:1:i+1
        if (i == j)
           J(i,j) = -2 -2 * alpha;
        elseif ( j > 0 && j < N )
           J(i,j) = alpha;
        end
     end
```

```
end

mm1 = M - 1;
nm2 = N - 2;
MNm1 = mm1*nm1;
deltay2 = deltay^2;
Q = repmat(0,MNm1,MNm1);
B = repmat(NaN,MNm1,1);
for i=1:1:mm1
   for j=i-1:1:i+1
      if (i == j)
         Q((j-1)*nm1+1:j*nm1,(i-1)*nm1+1:i*nm1) = J;
      elseif ( j > 0 && j < M )
         Q((j-1)*nm1+1:j*nm1,(i-1)*nm1+1:i*nm1) = eye(nm1);
      end
   end
   im1 = i - 1;
   ip1 = i + 1;
   if (i == 1)
      B(im1*nm1+1,1) = -U(2,1) -alpha*U(1,ip1) +f(X(2),Y(ip1))*deltay2;
      for k=2:1:nm2
         B(im1*nm1+k,1) = -U(k+1,1) + f(X(k+1),Y(ip1))*deltay2;
      end
      B(i*nm1,1) = -U(N,1) -alpha*U(np1,ip1) +f(X(N),Y(ip1))*deltay2;
   elseif (i == mm1)
      B(im1*nm1+1,1) = -U(2,mp1) -alpha*U(1,ip1) +f(X(2),Y(ip1))*deltay2;
      for k=2:1:nm2
         B(im1*nm1+k,1) = -U(k+1,mp1) + f(X(k+1),Y(ip1))*deltay2;
      end
      B(i*nm1,1) = -U(N,mp1) -alpha*U(np1,ip1) +f(X(N),Y(ip1))*deltay2;
   else
      B(im1*nm1+1,1) = -alpha*U(1,ip1) +f(X(2),Y(ip1))*deltay2;
      for k=2:1:nm2
         B(im1*nm1+k,1) = f(X(k+1),Y(ip1))*deltay2;
      end
      B(i*nm1,1) = -alpha*U(np1,ip1) +f(X(N),Y(ip1))*deltay2;
   end
end

% Solve the system Q UU = B with matlab library
UU = linsolve(Q,B);
% Solve the system Q UU = B with gauss()
% UU = gauss(Q,B,1);

% Transfer UU to U
for i=1:1:mm1
```

```
    im1 = i - 1;
    ip1 = i + 1;
    U(2:N,ip1) = UU(im1*nm1+1:i*nm1,1);
  end
end
```

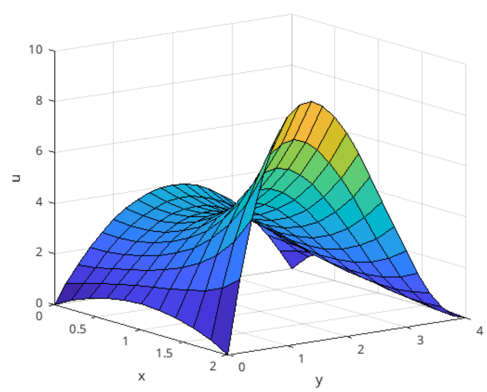
Remark 15.2.8
There is one issue with the code above when the mesh sizes are really small. The matrix Q may be very large. So, simple Gauss elimination is not recommended to solve $QU = B$. The matrix Q is block tridiagonal as are the matrices J and K. It is therefore really important to develop efficient and economical methods to solve very large linear system of this form. This is outside the scope of this manuscript. A good starting reference is [13]. ♠

Example 15.2.9
Use the previous finite difference scheme to approximate the solution to the following Dirichlet equation.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = xy;$$

on the domain $R = [0,2]\times[0,4]$ with the boundary conditions $u(x,0) = x(2-x)$ and $u(x,4) = x(2-x)^2$ for $0 \leq x \leq 2$, and $u(0,y) = y(4-y)$ and $u(2,y) = y(4-y)^2$ for $0 \leq y \leq 4$.

With the matlab code below, we got the following graph for the approximation of the solution u.



Code 15.2.10

```
f = @(x,y) x.*y;
gb = @(x) x.*(2-x);
gt = @(x) x.*(2-x).^2;
gl = @(y) y.*(4-y);
gr = @(y) y.*(4-y).^2;
a = 0;
b = 2;
c = 0;
d = 4;
```

```
N = 10;
M = 20;

[X,Y,U] = dirichletS1(f,gb,gt,gl,gr,N,M,a,b,c,d);
[XX,YY] = meshgrid(X,Y);

% We need to transpose the matrix U because meshgrid()
% transposes the coordinates.
surf(XX,YY,U')
xlabel('x')
ylabel('y')
zlabel('u')
```

♣

15.2.3 Hyperbolic Equations

We consider the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad , \quad 0 < x < L \text{ and } 0 < t < T \ , \tag{15.2.14}$$

with the boundary conditions

$$u(0,t) = h_0(t) \text{ and } u(L,t) = h_L(t) \quad , \quad 0 \leq t \leq T \ , \tag{15.2.15}$$

and the initial conditions

$$u(x,0) = g(x) \text{ and } \frac{\partial u}{\partial t}(x,0) = f(x) \quad , \quad 0 \leq x \leq L \ , \tag{15.2.16}$$

where $g : [0, L] \to \mathbb{R}$ is a continuous function satisfying $g(0) = h_0(0)$ and $g(L) = h_L(0)$, and $f : [0, L] \to \mathbb{R}$ is also continuous.

We develop a finite difference scheme for the wave equation given in (15.2.14), (15.2.15) and (15.2.16).

Given two integers $N \geq 2$ and $M \geq 1$, we set $\Delta x = L/N$, $\Delta t = T/M$, $x_i = i\Delta x$, $t_j = j\Delta t$ and $u_{i,j} = u(x_i, t_j)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$. From (15.1.6) for x and t, we get

$$\begin{split} \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta t)^2} - \frac{1}{4!}\left(\frac{\partial^4 u}{\partial t^4}(x_i, \zeta_{i,j}) + \frac{\partial^4 u}{\partial t^4}(x_i, \eta_{i,j})\right)(\Delta t)^2 \\ = c^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} - \frac{c^2}{4!}\left(\frac{\partial^4 u}{\partial x^4}(\mu_{i,j}, t_j) + \frac{\partial^4 u}{\partial x^4}(\nu_{i,j}, t_j)\right)(\Delta x)^2 \end{split} \tag{15.2.17}$$

for $\zeta_{i,j}, \eta_{i,j} \in]t_{j-1}, t_{j+1}[$ and $\mu_{i,j}, \nu_{i,j} \in]x_{i-1}, x_{i+1}[$. For Δt and Δx small, we have

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta t)^2} \approx c^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \ .$$

This suggests the following finite difference equation.

$$\frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta t)^2} = c^2\,\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} \tag{15.2.18}$$

for $0 < i < N$ and $0 < j < M$.

Since $w_{i,-1}$ is not defined for $0 < i < N$, (15.2.12) cannot be used for $j = 0$. Thus, the value of $w_{i,1}$ for $0 < i < N$ cannot be computed with (15.2.12). The initial condition of $\dfrac{\partial u}{\partial t}$ is useful here. The initial condition of $\dfrac{\partial u}{\partial t}$ may be evaluated with the formula (15.1.5). If we assume that u is defined for $t < 0$, we may write

$$\frac{\partial u}{\partial t}(x_i,0) = \frac{u_{i,1} - u_{i,-1}}{2\Delta t} - \frac{1}{12}\left(\frac{\partial^3 u}{\partial t^3}\left(x_i,\tilde{\zeta}_i\right) + \frac{\partial^3 u}{\partial t^3}\left(x_i,\tilde{\eta}_i\right)\right)(\Delta t)^2$$

for some $\tilde{\zeta}_i, \tilde{\eta}_i \in]t_{-1}, t_1[$. We choose this finite difference formula to approximate $\dfrac{\partial u}{\partial t}$ because, for Δt near 0, its local truncation error

$$-\frac{1}{12}\left(\frac{\partial^3 u}{\partial t^3}\left(x_i,\tilde{\zeta}_i\right) + \frac{\partial^3 u}{\partial t^3}\left(x_i,\tilde{\eta}_i\right)\right)(\Delta t)^2 = O\left((\Delta t)^2\right)$$

is comparable to the local truncation error

$$-\frac{1}{4!}\left(\frac{\partial^4 u}{\partial t^4}\left(x_i,\zeta_{i,j}\right) + \frac{\partial^4 u}{\partial t^4}\left(x_i,\eta_{i,j}\right)\right)(\Delta x)^2 = O\left((\Delta t)^2\right)$$

of the finite difference formula that has been used to approximate $\dfrac{\partial^2 u}{\partial t^2}$. We have for Δt small enough that

$$\frac{\partial u}{\partial t}(x_i,0) \approx \frac{u_{i,1} - u_{i,-1}}{2\Delta t} \quad , \quad 0 \le i \le N\ .$$

Thus

$$u_{i,-1} \approx u_{i,1} - 2\frac{\partial u}{\partial t}(x_i,0)\Delta t \quad , \quad 0 \le i \le N\ .$$

This suggests the following formula for $w_{i,-1}$.

$$w_{i,-1} = w_{i,1} - 2\,w'_{i,0}\Delta t \quad , \quad 0 \le i \le N\ ,$$

where

$$w'_{i,0} = \frac{\partial u}{\partial t}(x_i,0) = f(x_i) \quad , \quad 0 \le i \le N\ .$$

The initial condition on u imposes $w_{i,0} = g(x_i)$ for $0 \le i \le N$. The boundary conditions impose $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \le j \le M$.

We finally get the following finite difference scheme.

Algorithm 15.2.11

$$w_{i,j+1} - 2w_{i,j} + w_{i,j-1} - \alpha \left(w_{i+1,j} - 2w_{i,j} + w_{i-1,j} \right) = 0$$

for $0 < i < N$ and $0 \le j < M$, where $\alpha = \dfrac{c^2(\Delta t)^2}{(\Delta x)^2}$, $w_{0,j} = h_0(t_j)$ and $w_{N,j} = h_L(t_j)$ for $0 \le j \le M$, $w_{i,0} = g(x_i)$ for $0 \le i \le N$, and $w_{i,-1} = w_{i,1} - 2\,f(x_i)\Delta t$ for $0 \le i \le N$.

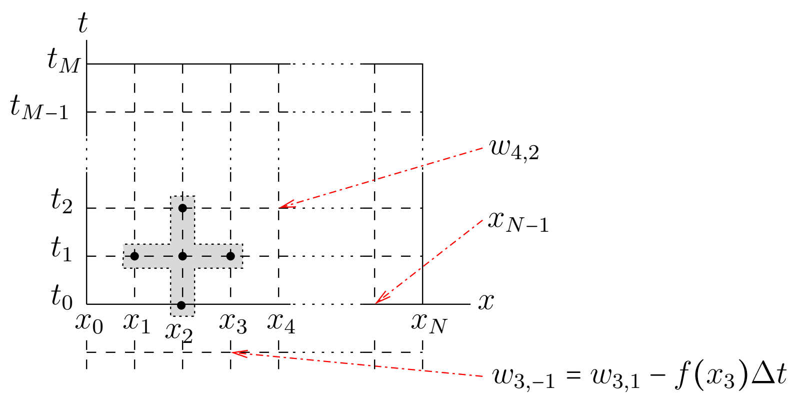


Figure 15.4: Schematic representation of the finite difference scheme given in Algorithm 15.2.11.

This scheme is illustrated in Figure 15.4.

As the previous schemes, it can be expressed as a linear system of the form $A\mathbf{w} = \mathbf{B}$. The (column) vector $\mathbf{w}$ is defined by

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} \quad \text{with} \quad \mathbf{w}_j = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{N-1,j} \end{pmatrix}$$

for $0 < j \le M$. The matrix A is a $(N-1)M \times (N-1)M$ matrix of the form

$$A = \begin{pmatrix} \mathrm{Id} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ J & \mathrm{Id} & 0 & 0 & \dots & 0 & 0 & 0 \\ \mathrm{Id} & J & \mathrm{Id} & 0 & \dots & 0 & 0 & 0 \\ 0 & \mathrm{Id} & J & \mathrm{Id} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathrm{Id} & J & \mathrm{Id} \end{pmatrix},$$

where Id is the $(N-1)\times(N-1)$ identity matrix and

$$J=\begin{pmatrix}-2+2\alpha & -\alpha & 0 & 0 & 0 & \dots & 0 & 0\\ -\alpha & -2+2\alpha & -\alpha & 0 & 0 & \dots & 0 & 0\\ 0 & -\alpha & -2+2\alpha & -\alpha & 0 & \dots & 0 & 0\\ 0 & 0 & -\alpha & -2+2\alpha & -\alpha & \dots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & 0 & \cdots & -\alpha & -2+2\alpha\end{pmatrix}$$

is a $(N-1)\times(N-1)$ matrix. The vector $\mathbf{B}$ is defined by

$$\mathbf{B}=\begin{pmatrix}\mathbf{B}_1\\ \mathbf{B}_2\\ \vdots\\ \mathbf{B}_M\end{pmatrix}, \quad \text{where } \mathbf{B}_1=\frac{1}{2}\begin{pmatrix}(2-2\alpha)w_{1,0}+\alpha w_{0,0}+\alpha w_{2,0}+2\,w'_{1,0}\Delta t\\ (2-2\alpha)w_{2,0}+\alpha w_{1,0}+\alpha w_{3,0}+2\,w'_{2,0}\Delta t\\ (2-2\alpha)w_{3,0}+\alpha w_{2,0}+\alpha w_{4,0}+2\,w'_{3,0}\Delta t\\ \vdots\\ (2-2\alpha)w_{N-1,0}+\alpha w_{N-2,0}+\alpha w_{N,0}+2\,w'_{N-1,0}\Delta t\end{pmatrix},$$

$$\mathbf{B}_2=\begin{pmatrix}-w_{1,0}+\alpha w_{0,1}\\ -w_{2,0}\\ -w_{3,0}\\ \vdots\\ -w_{N-1,0}+\alpha w_{N,1}\end{pmatrix} \quad \text{and} \quad \mathbf{B}_j=\begin{pmatrix}\alpha w_{0,j-1}\\ 0\\ 0\\ \vdots\\ \alpha w_{N,j-1}\end{pmatrix}$$

for $3\le j\le M$.

15.3 Convergence, Consistency and Stability

The presentation in this section is based on [20, 18, 19].

There are three questions that come to mind when using a finite difference scheme to numerically solve a partial differential equation.

1. Is there a solution to the system $A\mathbf{w}=\mathbf{B}$ associated to a finite difference scheme and, if so, is this solution unique?

2. Since computations cannot be performed exactly on computers (round off errors), and since the boundary and initial conditions are often approximations of the true values (experimental values) or cannot be entered exactly on computer (round off errors), is the finite difference scheme "stable?" Namely, if the computed value at one step of the finite difference scheme is slightly modified, will the computed value at the following step be close to the value that should have been found if the previous value had not been modified. If the method is not stable, we cannot hope to get reliable results.

3. Is the solution to the finite difference scheme close to the solution of the partial differential equation from which we have developed the finite difference scheme?

The first question is easy to answer positively because the matrices A obtained from the finite difference schemes that we have presented are invertible. As we will see when studying the stability of the finite difference scheme, the matrices A have non-zero eigenvalues and so a non-zero determinant. Hence there exists a unique solution to $A\mathbf{w} = \mathbf{B}$.

15.3.1 Uniform Theory

We assume that the **domain** for the partial differential equation is $R = \{(x,y) : a \leq x \leq b \text{ and } x \leq y \leq d\}$. The **boundary** of R, denoted ∂R, is the set of points (x,y) where conditions are imposed on u. The **interior** of R is defined as the set $R^o = R \setminus \partial R$. Be aware that the definition of boundary and interior of a set given here may not coincide with the normal definition of border and interior of a set in topology.

Once the step sizes $\Delta x = (b-a)/N$ and $\Delta y = (d-c)/M$ have been selected, we define the **domain for the finite difference scheme** as

$$R_\Delta = \{(x_i, y_j) : x_i = i\Delta x \text{ for } 0 \leq i \leq N, \text{ and } y_j = j\Delta y \text{ for } 1 \leq j \leq M\} \ . \tag{15.3.1}$$

The **boundary** of R_Δ, denoted ∂R_Δ, is the set of mesh points $(x_i, y_j) \in \partial R$. The **interior** of R_Δ is defined as the set $R_\Delta^o = R_\Delta \setminus \partial R$.

Example 15.3.1
For the heat and wave equations, y is replaced by t and the boundary of R is defined as the set $\partial R = \{(x,0) : 0 \leq x \leq L\} \cup \{(x,t) : x = 0 \text{ or } L, \text{ and } 0 \leq t \leq T\}$. The boundary of R_Δ is defined as the set

$$\partial R_\Delta = \{(x_i, 0) : x_i = i\Delta x \text{ for } 0 \leq i \leq N\} \cup \{(x, t_j) : x = 0 \text{ or } L, \text{ and } t_j = j\Delta t \text{ for } 1 \leq j \leq M\} \ .$$

For the Dirichlet equation, the boundaries of R and R_δ have the expected definition: $\partial R = \{(x,y) : y = c \text{ or } d, \text{ and } a \leq x \leq b\} \cup \{(x,y) : x = a \text{ or } b, \text{ and } c \leq y \leq d\}$ and

$$\begin{aligned}\partial R_\Delta = &\{(x_i, y) : y = c \text{ or } d, \text{ and } x_i = i\Delta x \text{ for } 0 \leq i \leq N\} \\ &\cup \{(x, y_j) : x = a \text{ or } b, \text{ and } y_j = j\Delta y \text{ for } 1 \leq j \leq M\} \ .\end{aligned}$$

♣

The partial differential equations that we are considering are of the form

$$P\left(u(x,y), \frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y), \frac{\partial^2 u}{\partial x^2}(x,y), \ldots\right) = F(x,y) \ , \tag{15.3.2}$$

where P is a linear mapping and $F : R \to \mathbb{R}$ is a given function. The finite difference schemes that we have deduced to numerically solve these partial differential equations are based on finite difference equations of the form

$$P_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) = F(x_i, y_j) \tag{15.3.3}$$

for all (i,j) such that $(x_i, y_j) \in R_\Delta^o$, where P_Δ is also a linear mapping. These schemes were deduced in Section 15.2 from the expressions that we got after substituting finite difference formulae for the partial derivatives at (x_i, y_j) into the heat, Dirichlet and wave equations.

Example 15.3.2
For the finite difference scheme in Algorithm 15.2.1, we have

$$P\left(u(x,y), \frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y), \frac{\partial^2 u}{\partial x^2}(x,y), \ldots\right) = \frac{\partial u}{\partial t} - c^2 \frac{\partial^2 u}{\partial x^2}$$

and

$$P_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) = \frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} \ . \tag{15.3.4}$$

♣

In the definition of P_Δ given in (15.3.3), we are referring to $(x_i, y_j) \in R^o_\Delta$. This is not really correct. As for the finite difference scheme in (15.3.4) above, the formula (15.3.3) is used to approximate the value of $u(x_i, y_{j+1})$. It is (x_i, y_{j+1}) which is really in R^o_Δ. The point (x_i, y_j) may be on the boundary as it is the case in (15.3.4) for $j = 0$. Nevertheless, we prefer to use the formulation above because it expresses more clearly that formula (15.3.3) is used to approximate a value of u at an interior point.

As we had to do for the wave equation, we may also have to approximate the boundary and/or initial conditions of u on ∂R_Δ. These conditions are given by a formula of the form

$$B\left(u(x,y), \frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y), \ldots\right) = G(x,y) \tag{15.3.5}$$

evaluated on ∂R. where B is a linear mapping and $G : \partial R \to \mathbb{R}$ is a given function. The approximation of the boundary or initial condition at each mesh points (x_i, y_j) of ∂R_Δ is given by a formula of the form

$$B_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) = G(x_i, y_j) \tag{15.3.6}$$

for all (i,j) such that $(x_i, y_j) \in \partial R_\Delta$, where B_Δ is a linear mapping. This formula is also deduced from the expressions that we got after substituting finite difference formulae for the partial derivatives at $(x_i, y_j) \in \partial R_\Delta$ into the boundary and initial conditions.

Example 15.3.3
For the heat equation with forcing, we have

$$B\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial t}(x,t), \ldots\right) = u(x,t)$$

for all $(x,t) \in \partial R$, $B_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) = w_{i,j}$ for all $(i.j)$ such that $(x_i, t_j) \in \partial R_\Delta$, and

$$G(x,t) = \begin{cases} h_0(t) & \text{for } x = 0 \text{ and } 0 \leq t \leq T \\ h_L(t) & \text{for } x = L \text{ and } 0 \leq t \leq T \\ g(x) & \text{for } t = 0 \text{ and } 0 \leq x \leq L \end{cases}$$

for all $(x,t) \in \partial R$.

For the Dirichlet equation, we have

$$B\left(u(x,y), \frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y), \ldots\right) = u(x,y)$$

for all $(x,y) \in \partial R$, $B_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) = w_{i,j}$ for all $(i.j)$ such that $(x_i, y_j) \in \partial R_\Delta$, and $G(x,y) = g(x,y)$ for all $(x,y) \in \partial R$.

For the wave equation, we may choose

$$B\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial t}(x,t), \ldots\right) = \begin{cases} \begin{pmatrix} u(x,t) \\ u(x,t) \end{pmatrix} & \text{for } x = 0 \text{ or } x = L, \text{ and } 0 \le t \le T \\ \begin{pmatrix} u(x,t) \\ \dfrac{\partial u}{\partial t}(x,t) \end{pmatrix} & \text{for } t = 0 \text{ and } 0 \le x \le L \end{cases}$$

for all $(x,t) \in \partial R$,

$$B_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) \begin{cases} \begin{pmatrix} w_{i,j} \\ w_{i,j} \end{pmatrix} & \begin{array}{l} \text{for } i = 0 \text{ or } i = N, \\ \text{and } 0 \le j \le M \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ -1/(2\Delta t) & 0 & 1/(2\Delta t) \end{pmatrix} \begin{pmatrix} w_{i,-1} \\ w_{i,0} \\ w_{i,1} \end{pmatrix} & \text{for } j = 0 \text{ and } 0 \le i \le N \end{cases}$$

for all $(i.j)$ such that $(x_i, y_j) \in \partial R_\Delta$, and

$$G(x,t) = \begin{cases} \begin{pmatrix} h_0(t) \\ h_0(t) \end{pmatrix} & \text{for } x = 0 \text{ and } 0 < t \le T \\ \begin{pmatrix} h_L(t) \\ h_L(t) \end{pmatrix} & \text{for } x = L \text{ and } 0 < t \le T \\ \begin{pmatrix} g(x) \\ f(x) \end{pmatrix} & \text{for } t = 0 \text{ , and } 0 \le x \le L \end{cases}$$

for all $(x,t) \in \partial R$. ♣

To answer the third question in the introduction of this section, we have to show that the following definition is satisfied.

Definition 15.3.4

The solution of a finite difference scheme associated to a finite difference equation $P_\Delta = F$ with conditions $B_\Delta = G$ as in (15.3.3) and (15.3.6) **converges** toward the solution of the partial differential equation given by $P = F$ with conditions $B = G$ as in (15.3.2) and (15.3.5) if

$$\max\left\{\left|w_{i,j} - u_{i,j}\right| : 0 \le i \le N \text{ and } 0 \le j \le M\right\} \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty \ ,$$

where $\{w_{i,j} : 0 \le i \le N \text{ and } 0 \le j \le M\}$ is the solution of the finite difference scheme and u is the solution of the partial differential equation. As before $u_{i,j} = u(x_i, y_j)$ for $0 \le i \le N$ and $0 \le j \le M$.

Remark 15.3.5
Be aware that the previous definition of convergence does not consider any round off error or perturbation. Therefore, a method may theoretically converge according to the previous definition but not give good results in practice. Nevertheless, we must at least verify that a method converges according to the previous definition before using it. To keep the presentation to a reasonable level of sophistication, we will not consider round off error in our presentation of the finite difference schemes except in some special cases like when we will discuss stability of finite difference schemes. ♠

Unfortunately, convergence is sometime difficult to prove. However, it may not be necessary to prove convergence directly to prove that a finite difference scheme is convergent as we will see later. To justify this approach, we will need the following concepts.

Definition 15.3.6

Given any sufficiently differentiable function $q : R \to \mathbb{R}$, the **local truncation error** of the linear mapping P_Δ in (15.3.3) is the expression

$$\begin{aligned}\tau_{i,j}(\Delta x, \Delta y, q) = P_\Delta \left(q(x_i, y_j), q(x_i, y_{j+1}), q(x_{i+1}, y_j), \ldots \right) \\ - P\left(q(x_i, y_j), \frac{\partial q}{\partial x}(x_i, y_j), \frac{\partial q}{\partial y}(x_i, y_j), \frac{\partial^2 q}{\partial x^2}(x_i, y_j), \ldots \right)\end{aligned}$$

for all (i,j) such that $(x_i, y_j) \in R_\Delta^o$.
We also define the **local error** for the linear mapping B_Δ in (15.3.6) as

$$\begin{aligned}\sigma_{i,j}(\Delta x . \Delta y, q) = B_\Delta \left(q(x_i, y_j), q(x_i, y_{j+1}), q(x_{i+1}, y_j), \ldots \right) \\ - B\left(q(x_i, y_j), \frac{\partial q}{\partial x}(x_i, y_j), \frac{\partial q}{\partial y}(x_i, y_j), \ldots \right) .\end{aligned}$$

for all (i,j) such that $(x_i, y_j) \in \partial R$.
A finite difference scheme determined by the linear mappings P_Δ and B_Δ as in (15.3.3) and (15.3.6) is **consistent** if

$$\max_{\substack{(i,j) \text{ such that} \\ (x_i,y_j) \in R_\Delta^o}} |\tau_{i,j}(\Delta x, \Delta y, q)| \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty$$

and

$$\max_{\substack{(i,j) \text{ such that} \\ (x_i,y_j) \in \partial R_\Delta}} \|\sigma_{i,j}(\Delta x, \Delta y, q)\| \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty$$

for all sufficiently differentiable function $q : R \to \mathbb{R}$. If there are constraints on the grids used in the two previous limits, namely on Δx and Δy, then the finite difference scheme is said to be **conditionally consistent**.

As mentioned previously, we are using the imprecise reference to $\tau_{i,j}$ for $(x_i, y_j) \in R_\Delta^o$ though (x_i, y_j) may not be in R_Δ^o. The formula (15.3.3) is used to approximate the value of $u(x_i, y_{j+1})$. It is (x_i, y_{j+1}) which is really in R_Δ^o. The point (x_i, y_j) may be on the boundary.

Remark 15.3.7 (Warning)
The expression $P\left(q(x_i,y_j), \dfrac{\partial q}{\partial x}(x_i,y_j), \dfrac{\partial q}{\partial y}(x_i,y_j), \ldots\right)$ in the definition of $\tau_{i,j}(\Delta x, \Delta y, q)$ may in fact be a linear mapping of the form

$$P\left(q(x_i,y_j), \frac{\partial q}{\partial x}(x_i,y_j), \frac{\partial q}{\partial y}(x_i,y_j), \ldots, q(x_i,y_{j+1}), \frac{\partial q}{\partial x}(x_i,y_{j+1}), \frac{\partial q}{\partial y}(x_i,y_{j+1}), \ldots\right) .$$

The Crank-Nicolson scheme is an example of this situation. For this reason, the expression $P\left(q(x_i,y_j), \dfrac{\partial q}{\partial x}(x_i,y_j), \dfrac{\partial q}{\partial y}(x_i,y_j), \ldots\right)$ should not be interpreted literally. For simplicity, we prefer to use the expression of the form $P\left(q(x_i,y_j), \dfrac{\partial q}{\partial x}(x_i,y_j), \dfrac{\partial q}{\partial y}(x_i,y_j), \ldots\right)$ to clearly refer to the interior point (x_i,y_j) where we try to approximate the value of the solution u. ♠

Definition 15.3.8

A finite difference scheme determined by the linear mappings P_Δ and B_Δ as in (15.3.3) and (15.3.6) is **stable** if, for all function $v : R_\Delta \to \mathbb{R}$, there exists a constant C_α such that

$$\begin{split} \max_{\substack{0\leq i\leq N\\ 0\leq j\leq M}} |v_{i,j}| \leq C_\alpha \Bigg(& \max_{\substack{(i,j) \text{ such that}\\ (x_i,y_j)\in R^o_\Delta}} |P_\Delta\left(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots\right)| \\ & + \max_{\substack{(i,j) \text{ such that}\\ (x_i,y_j)\in \partial R_\Delta}} \|B_\Delta\left(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots\right)\| \Bigg) , \end{split} \tag{15.3.7}$$

where $v_{i,j} = v(x_i,y_j)$ for all i and j.
The index α for C_α is to indicate that there may be a constraining relation on Δx and Δy that must be satisfied for (15.3.7) to be satisfied. If there is no constraining relation on Δx and Δy used in (15.3.7), then the finite difference scheme is said to be **unconditionally stable**. If there is a constraining relation, then the finite difference scheme is said to be **conditionally stable**.

We have used the norm notation for $\|\sigma_{i,j}\|$ and $\|B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)\|$ in the previous two definitions because, as we have seen in the previous example, $B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)$ may be a vector.

To satisfy the notion of stability introduced in the second question in the introduction to this section, our finite difference schemes will need to satisfy the previous definition. To understand why, we have to consider round off errors. Suppose that $v_{i,j}$ is the computed approximation of $w_{i,j}$ for all i and j. We may assume that $\{v_{i,j} : 0 \leq i \leq N \text{ and } 0 \leq j \leq M\}$ is the exact solution of

$$P_\Delta\left(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots\right) = F(x_i,y_j) + \delta_{i,j}(\Delta x, \Delta y)$$

for all (i,j) such that $(x_i,y_j) \in R^o_\Delta$, and

$$B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots) = G(x_i,y_j) + \tilde{\delta}_{i,j}(\Delta x, \Delta y)$$

for all (i,j) such that $(x_i,y_j)\in\partial R_\Delta$, where the $\delta_{i,j}(\Delta x,\Delta y)$ and $\tilde{\delta}_{i,j}(\Delta x,\Delta y)$ represent round off errors. Thus $\{v_{i,j}-w_{i,j}: 0\le i\le N \text{ and } 0\le j\le M\}$ satisfies

$$P_\Delta\left(v_{i,j}-w_{i,j}, v_{i,j+1}-w_{i,j+1}, v_{i+1,j}-w_{i+1.j},\ldots\right)=\delta_{i,j}(\Delta x,\Delta y)$$

for all (i,j) such that $(x_i,y_j)\in R^o_\Delta$, and

$$B_\Delta(v_{i,j}-w_{i,j}, v_{i,j+1}-w_{i,j+1}, v_{i+1,j}-w_{i+1,k},\ldots)=\tilde{\delta}_{i,j}(\Delta x,\Delta y)$$

for all (i,j) such that $(x_i,y_j)\in\partial R_\Delta$. If the finite difference scheme is stable, there exists a constant C_α such that

$$\max_{\substack{0\le i\le N\\0\le j\le M}}|v_{i,j}-w_{i,j}|\le C_\alpha\left(\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}}|\delta_{i,j}(\Delta x,\Delta y)|+\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in\partial R_\Delta}}\left\|\tilde{\delta}_{i,j}(\Delta x,\Delta y)\right\|\right)$$

for all (i,j) such that $(x_i,y_j)\in R_\Delta$. The error $|v_{i,j}-w_{i,j}|$ is proportional to the round off errors in our computations.

The following theorem is quite useful to prove the convergence of a finite difference scheme.

Theorem 15.3.9

Consider finite difference scheme determined by the linear mappings P_Δ and B_Δ as in (15.3.3) and (15.3.6). If this finite difference scheme is stable and consistent, then it is convergent.

Proof.
For every $(x_i,y_j)\in R^o_\Delta$, we have

$$\begin{aligned}
0&=F(x_i,y_j)-F(x_i,y_j)\\
&=P_\Delta\left(w_{i,j},w_{i,j+1},w_{i+1,j},\ldots\right)-P\left(u(x_i,y_j),\frac{\partial u}{\partial x}(x_i,y_j),\frac{\partial u}{\partial y}(x_i,y_j),\frac{\partial^2 u}{\partial x^2}(x_i,y_j),\ldots\right)\\
&=P_\Delta\left(w_{i,j},w_{i,j+1},w_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\qquad+P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\qquad\quad-P\left(u(x_i,y_j),\frac{\partial u}{\partial x}(x_i,y_j),\frac{\partial u}{\partial y}(x_i,y_j),\frac{\partial^2 u}{\partial x^2}(x_i,y_j),\ldots\right)\\
&=P_\Delta\left(w_{i,j},w_{i,j+1},w_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)+\tau_{i,j}(\Delta x,\Delta y,u)\ .
\end{aligned}$$

It follows from the linearity of P_Δ and the definition of the local truncation error that

$$\begin{aligned}
&P_\Delta\left(w_{i,j}-u(x_i,y_j),w_{i,j+1}-u(x_i,y_{j+1}),w_{i+1,j}-u(x_{i+1},y_j),\ldots\right)\\
&\quad=P_\Delta\left(w_{i,j},w_{i,j+1},w_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)=-\tau_{i,j}(\Delta x,\Delta y,u)
\end{aligned}$$

for all (i,j) such that $(x_i,y_j)\in R^o_\Delta$. Similarly, we have

$$B_\Delta\left(w_{i,j}-u(x_i,y_j),w_{i,j+1}-u(x_i,y_{j+1}),w_{i+1,j}-u(x_{i+1},y_j),\ldots\right)$$

$$= B_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) - B_\Delta\left(u(x_i,y_j), u(x_i,y_{j+1}), u(x_{i+1},y_j), \ldots\right) = -\sigma_{i,j}(\Delta x, \Delta y, u)$$

for all (i,j) such that $(x_i,y_j) \in \partial R_\Delta$. Since the finite difference is stable, there exists a constant C_α such that

$$\begin{aligned}
&\max_{\substack{0\le i\le N\\0\le j\le M}} |w_{i,j} - u(x_i,y_j)| \\
&\quad \le C_\alpha\Bigg(\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}} \left|P_\Delta\left(w_{i,j}-u(x_i,y_j), w_{i,j+1}-u(x_i,y_{j+1}), w_{i+1,j}-u(x_{i+1},y_j),\ldots\right)\right| \\
&\quad + \max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in \partial R_\Delta}} \left\|B_\Delta(w_{i,j}-u(x_i,y_j), w_{i,j+1}-u(x_i,y_{j+1}), w_{i+1,j}-u(x_{i+1},y_j),\ldots)\right\| \Bigg) \\
&\quad \le C_\alpha\Bigg(\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}} |\tau_{i,j}(\Delta x,\Delta y,u)| + \max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in \partial R_\Delta}} \|\sigma_{i,j}(\Delta x,\Delta y,u)\| \Bigg).
\end{aligned}$$

Finally, since the finite difference scheme is consistent,

$$\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}} |\tau_{i,j}(\Delta x,\Delta y,u)| \to 0 \quad \text{and} \quad \max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in \partial R_\Delta}} \|\sigma_{i,j}(\Delta x,\Delta y,u)\| \to 0$$

as $\min\{N,M\} \to \infty$ imply that

$$\max_{\substack{0\le i\le N\\0\le j\le M}} |w_{i,j} - u(x_i,y_j)| \to 0$$

as $\min\{N,M\} \to \infty$. ∎

Since it is generally easier to prove stability and consistency, the previous theorem gives us a method to prove convergence without having to prove it from the definition. This is the approach that we generally use later to prove convergence for some of the finite difference schemes that we have presented in the previous section.

Remark 15.3.10

1. Be aware that some finite difference schemes may not converge for all possible functions F and G but may be converging for some sub-classes of functions F and G.

2. There are finite difference scheme that may be converging but note stable according to the definition that we have given. This is because the definition of stability that we have given is more restrictive that is often necessary. Consult [20] for more information.

♠

We can give a more precise analysis of the effect of round off error on the numerical approximation of the solution. As before, suppose that $v_{i,j}$ is the computed approximation of $w_{i,j}$ for all i and j. We may assume that $\{v_{i,j} : 0 \le i \le N \text{ and } 0 \le j \le M\}$ is the exact solution of

$$P_\Delta\left(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots\right) = F(x_i,y_j) + \delta_{i,j}(\Delta x,\Delta y) \tag{15.3.8}$$

for all (i,j) such that $(x_i,y_j)\in R^o_\Delta$, and

$$B_\Delta(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots)=G(x_i,y_j)+\tilde{\delta}_{i,j}(\Delta x,\Delta y)$$

for all (i,j) such that $(x_i,y_j)\in\partial R_\Delta$, where the $\delta_{i,j}(\Delta x,\Delta y)$ and $\tilde{\delta}_{i,j}(\Delta x,\Delta y)$ represent round off errors. Let us assume that there exists $\delta:\mathbb{R}\to\mathbb{R}$ such that $|\delta_{i,j}(\Delta x,\Delta y)|\leq\delta(\Delta x,\Delta y)$ and $|\tilde{\delta}_{i,j}(\Delta x,\Delta y)|\leq\delta(\Delta x,\Delta y)$ for all i and j. So $\delta(\Delta x,\Delta y)$ is a bound on the round off errors. Proceeding as in the proof of Theorem 15.3.9 and using (15.3.8), we have that

$$\begin{aligned}
0 &= F(x_i,y_j)-F(x_i,y_j)=P_\Delta\left(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots\right)\\
&\qquad -P\left(u(x_i,y_j),\frac{\partial u}{\partial x}(x_i,y_j),\frac{\partial u}{\partial y}(x_i,y_j),\frac{\partial^2 u}{\partial x^2}(x_i,y_j),\ldots\right)-\delta_{i,j}(\Delta x,\Delta y)\\
&= P_\Delta\left(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\qquad +P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\qquad -P\left(u(x_i,y_j),\frac{\partial u}{\partial x}(x_i,y_j),\frac{\partial u}{\partial y}(x_i,y_j),\frac{\partial^2 u}{\partial x^2}(x_i,y_j),\ldots\right)-\delta_{i,j}(\Delta x,\Delta y)\\
&= P_\Delta\left(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\qquad +\tau_{i,j}(\Delta x,\Delta y,u)-\delta_{i,j}(\Delta x,\Delta y)
\end{aligned}$$

for all (i,j) such that $(x_i,y_j)\in R^o_\Delta$. It follows from the linearity of P_Δ that

$$\begin{aligned}
&P_\Delta\left(v_{i,j}-u(x_i,y_j),v_{i,j+1}-u(x_i,y_{j+1}),v_{i+1,j}-u(x_{i+1},y_j),\ldots\right)\\
&\quad=P_\Delta\left(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots\right)-P_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\quad=-\tau_{i,j}(\Delta x,\Delta y,u)+\delta_{i,j}(\Delta x,\Delta y)
\end{aligned}$$

for all (i,j) such that $(x_i,y_j)\in R^o_\Delta$. Similarly, we have

$$\begin{aligned}
&B_\Delta\left(v_{i,j}-u(x_i,y_j),v_{i,j+1}-u(x_i,y_{j+1}),v_{i+1,j}-u(x_{i+1},y_j),\ldots\right)\\
&\quad=B_\Delta\left(v_{i,j},v_{i,j+1},v_{i+1,j},\ldots\right)-B_\Delta\left(u(x_i,y_j),u(x_i,y_{j+1}),u(x_{i+1},y_j),\ldots\right)\\
&\quad=-\sigma_{i,j}(\Delta x,\Delta y,u)+\tilde{\delta}_{i,j}(\Delta x,\Delta y)
\end{aligned}$$

for all (i,j) such that $(x_i,y_j)\in\partial R_\Delta$. Since the finite difference is stable, there exist a constant C_α such that

$$\begin{aligned}
&\max_{\substack{0\leq i\leq N\\0\leq j\leq M}}|v_{i,j}-u(x_i,y_j)|\\
&\quad\leq C_\alpha\Bigg(\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}}\left|P_\Delta\left(v_{i,j}-u(x_i,y_j),v_{i,j+1}-u(x_i,y_{j+1}),v_{i+1,j}-u(x_{i+1},y_j),\ldots\right)\right|\\
&\quad+\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in\partial R_\Delta}}\left\|B_\Delta(v_{i,j}-u(x_i,y_j),v_{i,j+1}-u(x_i,y_{j+1}),v_{i+1,j}-u(x_{i+1},y_j),\ldots)\right\|\Bigg)\\
&\quad\leq C_\alpha\Bigg(\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}}|\tau_{i,j}(\Delta x,\Delta y,u)|+\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in\partial R_\Delta}}\|\sigma_{i,j}(\Delta x,\Delta y,u)\|+2\delta(\Delta x,\Delta y)\Bigg)\ .
\end{aligned}$$

If the finite difference scheme is consistent, we have that

$$\max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R_\Delta^o}} |\tau_{i,j}| \to 0 \quad \text{and} \quad \max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in \partial R_\Delta}} \|\sigma_{i,j}\| \to 0$$

as $\min\{N,M\} \to \infty$. Hence, the precision of the finite difference scheme is proportional to the round off error.

Note that there is no reason for round off errors to decrease as $\min\{N,M\} \to \infty$; namely, as $\max\{\Delta x, \Delta y\} \to 0$. In fact, round off errors may start to increase for $\max\{\Delta x, \Delta y\}$ small if the computation involve divisions by small numbers.

15.3.2 ℓ^2 Theory

The definitions of convergence, consistency and stability that we have presented in the previous section are the strongest ones to be given because they require uniform convergence on all the domain of the boundary value problem. Unfortunately, these definitions are too restrictive for many of the interesting finite difference schemes. Weaker definitions of convergence, consistency and stability are required.

The discussion in this section is basically for the heat and wave equation. For the Dirichlet equation, the previous notion of convergence, consistency and stability are fine. We prove in Section 15.6, using totally different techniques than those presented in this section, that Algorithm 15.2.6 for the Dirichlet equation satisfies the previous definitions of convergence, consistency and stability.

We only touch the subject of stability and convergence for finite difference schemes to solve partial differential equations. A good reference on the subject and one of the principal source of information for this section is [18].

The universal idea about that stability is to ensure that the errors in our computed values do not increase as the step sizes decrease, at least that the errors are bounded by a small value as the step sizes decrease.

We consider a partial differential equation

$$P\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial t}(x,t), \frac{\partial^2 u}{\partial x^2}(x,t), \ldots\right) = 0$$

on the domain $\mathbb{R} \times [0,T]$ and assume that the initial conditions are periodic with period 2π. More precisely, we assume that $u(x,0) = g(x)$ for a periodic function $g : \mathbb{R} \to \mathbb{R}$ of period 2π. For hyperbolic equation like the wave equation, we also assume that $\dfrac{\partial u}{\partial t}(x,0) = h(x)$ for a periodic function $h : \mathbb{R} \to \mathbb{R}$ of period 2π. Instead of boundary conditions, we assume that the solution $u(x,t)$ of the partial differential equation is periodic of period 2π with respect to x.

A problem given by a partial differential equation with only initial conditions like the problem above is called a **Cauchy problem**.

Suppose that we have a finite difference scheme of the form $P_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) = F(x_i, t_j)$ for all (i,j) such that

$$(x_i, t_j) \in R_\Delta^o = \{(x_i, t_j) : x_i = i\Delta x \text{ for } i \in \mathbb{Z} \text{ and } t_j = j\Delta t \text{ for } 0 < j \leq M\}$$

and $w_{i,j} \approx u_{i,j}$.

We consider the space $\ell^2(\mathbb{Z})$ of all functions $g : \mathbb{Z} \to \mathbb{C}$ such that

$$\|g\|_2^2 = \sum_{k\in\mathbb{Z}} |g(k)|^2 < \infty \ .$$

We assume that we can express this finite difference scheme as $g_{j+1} = Q_\alpha(g_j)$ for $j \geq 0$, where $g_j : \mathbb{Z} \to \mathbb{R}$ for $j \geq 0$ is defined by $g_j(i) = w_{i,j}$ for all i and j, and $Q_\alpha : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ is a bounded linear mapping. The index α in Q_α is to indicate that the linear operator may depend on a relation between Δx and Δt.

Example 15.3.11

For the heat equation without forcing, the Crank-Nicolson scheme given in Algorithm 15.2.2 is of the form $g_{j+1} = Q_\alpha(g_j)$; namely,

$$g_{j+1}(i) = Q_\alpha(g_j)(i) = \sum_{s\in\mathbb{Z}} q_{i,s}\, g_j(s) \ ,$$

where $q_{i,s}$ is the (i,s) component of the infinite matrix $Q_\alpha = -J^{-1}K$, where

$$J_{r,s} = \begin{cases} 1 + 2\alpha & \text{if} \quad r = s \\ -\alpha & \text{if} \quad s = r+1 \text{ or } s = r-1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$K_{r,s} = \begin{cases} -1 + 2\alpha & \text{if} \quad r = s \\ -\alpha & \text{if} \quad s = r+1 \text{ or } s = r-1 \\ 0 & \text{otherwise} \end{cases}$$

for $r, s \in \mathbb{Z}$. Recall that $\alpha = c\Delta t/(2(\Delta x)^2)$.

As we will see later, we do not have to worry about computing the inverse of the “infinite dimensional” matrix J. Note that the (r,s) component of the product of two infinite dimensional matrices A and B is defined by $\sum_{k\in\mathbb{Z}} A_{r,k}B_{k,s}$. ♣

We need new definitions for convergence, consistency and stability.

Definition 15.3.12

A finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ for all $j \geq 0$ is ℓ^2**-convergent** if, for all $t \in [0,T]$,

$$\|g_j - u_t\|_2 \to 0 \quad \text{as} \quad M \to \infty \text{ and } j\Delta t \to t \ ,$$

where $u_t(i) = u(i\Delta x, t)$ for all $i \in \mathbb{Z}$.

The previous definition is more general than what we may have expected. It does not only consider $t = j\Delta t$ but $j\Delta t \to t$.

Definition 15.3.13

Given any sufficiently differentiable function $q : \mathbb{R} \times [0,T] \to \mathbb{R}$, the **local truncation error** of the finite difference scheme $g_{j+1} = Q_\alpha(g_j)$ is the expression

$$\tau_t(\Delta x, \Delta t, q) = \frac{1}{\Delta t}\big(q_{t+\Delta t} - Q_\alpha(q_t)\big)$$

where $q_t(i) = q(i\Delta x, t)$ for all $i \in \mathbb{Z}$.
The finite difference scheme $g_{j+1} = Q_\alpha(g_j)$ is **consistent** if, for all t,

$$\sup_{0\le t\le T-\Delta t} \|\tau_t(\Delta x, \Delta t, q)\|_2 \to 0 \quad \text{as} \quad M \to \infty$$

for all sufficiently differentiable function $q : \mathbb{R} \times [0,T] \to \mathbb{R}$.

We also have a new definition of stability.

Definition 15.3.14

A finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ for all $j \ge 0$ is **ℓ^2-stable** if there exists a constant C_α such that $\|Q_\alpha^j\|_2 \le C$ for $0 \le j \le M$ and all M. The index α for C_α is to indicate that there may be a constraining relation on Δx and Δy that must be satisfied for $\|Q_\alpha^j\|_2 \le C$ to be satisfied for $0 \le j \le M$ and all M. If no constraining relation on Δx and Δy is imposed, then the finite difference scheme is said to be **unconditionally stable**. If there is a constraining relation, then the finite difference scheme is said to be **conditionally stable**.

This definition of stability ensures that the error of a computed value does not increase in ℓ^2 norm. This can be heuristically justified as it follows. Suppose that $\tilde{g}_j$ is the computed value obtained using the finite difference scheme. We may assume that

$$\tilde{g}_{j+1} = Q_\alpha(\tilde{g}_j) + \Delta t\,\delta_j \ , \tag{15.3.9}$$

where $\delta_j \in \ell^2(\mathbb{Z})$ represents the round off error. Let $r_j = \tilde{g}_j - g_j$ for $j \ge 0$. We have

$$r_{j+1} = \tilde{g}_{j+1} - g_{j+1} = \big(Q_\alpha(\tilde{g}_j) + \Delta t\,\delta_j\big) - Q_\alpha(g_j) = Q_\alpha\left(\tilde{g}_j - g_j\right) + \Delta t\,\delta_j = Q_\alpha\left(r_j\right) + \Delta t\,\delta_j$$

for $j \ge 0$. We get by induction that

$$r_j = Q_\alpha^j(r_0) + \Delta t \sum_{k=0}^{j-1} Q_\alpha^k\left(\delta_{j-1-k}\right)$$

for $j \ge 1$. If we assume that $\|\delta_j\|_2 \le \delta$ for all j, we get

$$\begin{aligned}\|r_j\|_2 &\le \|Q_\alpha^j\|_2\,\|r_0\|_2 + \Delta t\sum_{k=0}^{j-1}\|Q_\alpha^k\|_2\,\|\delta_{j-1-k}\|_2 \le \|Q_\alpha^j\|_2\,\|r_0\|_2 + \delta\Delta t\sum_{k=0}^{j-1}\|Q_\alpha^k\|_2\\ &\le C\|r_0\|_2 + \delta C(j\Delta t) \le C\|r_0\|_2 + \delta C T\end{aligned}$$

for $0 < j \leq M$. The error is bounded in ℓ^2 norm. In particular, if $r_0 = 0$, the error is bounded by a multiple of the round off error. This is the ideal case.

Remark 15.3.15
The reader certainly wanders why we may assume that the error in (15.3.9) is of the form $\Delta t\,\delta_j$. This can be motivated by the following example. If we consider the finite difference equation (15.2.5), then the approximate values $v_{i,j}$ of $w_{i,j}$ are given by the exact solution of

$$\frac{v_{i,j+1} - v_{i,j}}{\Delta t} - c^2\,\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta x)^2} = f(x_i, t_j) + \delta_{i,j}\ .$$

Thus

$$v_{i,j+1} = v_{i,j} + \alpha\,\left(v_{i+1,j} - 2v_{i,j} + v_{i-1,j}\right) = \Delta t\, f(x_i, t_j) + \Delta t\,\delta_{i,j}\ .$$

The function $\delta_j \in \ell^2(\mathbb{Z})$ is defined by $\delta_j(i) = \delta_{i,j}$ for $i \in \mathbb{Z}$. We can reach a similar conclusion with other finite difference schemes. ♠

As the reader may know, $\ell^2(\mathbb{Z})$ is a **Hilbert space**. The linear operator Q_α is therefore a linear operator from a Hilbert space into itself. The operators that we consider behave very like linear mapping on $\mathbb{R}^n$. Let E be a Hilbert space with the norm $\|\cdot\|$ defined by a scalar product $\langle\cdot,\cdot\rangle$, and let $P : E \to E$ be a bounded linear mapping. As in finite dimension, we have that $\|P^j\| \leq \|P\|^j$ and $\rho(P) \leq \|P\|$, where $\rho(P)$ is the **spectral radius** of P. Moreover, we have that $\rho(P) = \lim_{k\to\infty} \|P^k\|^{1/k}$ as in finite dimension. The **adjoint** of P is the bounded linear mapping $P^\ast$ such that $\langle Px, y\rangle = \langle x, P^\ast y\rangle$ for all $x, y \in E$. The operator $P^\ast$ is the equivalent of the transpose of a $n\times n$ matrix. A bounded linear operator P is **normal** if $PP^\ast = P^\ast P$ [3]. Again, as in finite dimension, if P is a normal operator, then $\rho(P) = \|P\|$.

The following criteria will be useful to determine if a finite difference scheme is stable.

Proposition 15.3.16 (Lax)

Consider a finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ for all $j \geq 0$. If there exists a constant C_α such that $\|Q_\alpha\|_2 \leq 1 + C_\alpha\Delta t$ for all Δt. then the finite difference scheme is ℓ^2-stable.

Proof.
We have

$$\left\|Q_\alpha^j\right\|_2 \leq \|Q_\alpha\|_2^j \leq (1 + C_\alpha\Delta t)^j \leq e^{jC_\alpha\Delta t} \leq e^{TC_\alpha}$$

for $0 < j \leq M$ and all $M > 0$. We have used the relation $e^x \geq 1 + x$ for all $x \in \mathbb{R}$. ∎

Proposition 15.3.17 (von Neumann)

If a finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ for all $j \geq 0$ is ℓ^2-stable, then there exists a constant C_α such that $\rho(Q_\alpha) \leq 1 + C_\alpha/M$ for all M.

[3] Symmetric operator (i.e. $P = P^\ast$) are obviously normal operators.

Proof.
Since the finite difference scheme is ℓ^2-stable, we have $\|Q_\alpha^j\|_2 \leq C_\alpha$ for $0 \leq j \leq M$. Hence,

$$\rho^j(Q_\alpha) = \rho(Q_\alpha^j) \leq \|Q_\alpha^j\|_2 \leq C_\alpha$$

for $0 \leq j \leq M$. For $j = M$, we get

$$\rho(Q_\alpha) \leq C_\alpha^{1/M} \leq 1 + \frac{C_\alpha}{M} .$$

The last inequality comes from the following observation. Consider $f(x) = e^{x \ln C_\alpha} - 1 - C_\alpha x$ for $0 \leq x \leq 1$. Since $f'(x) = \ln(C_\alpha) e^{x \ln C_\alpha} - C_\alpha \leq 0$ for $0 < C_\alpha \leq e$, we have $f(x) \leq f(0) = 0$ for $0 \leq x \leq 1$. Since f reaches it absolute maximum at $\tilde{x} = (\ln(C_\alpha) - \ln(\ln(C_\alpha)))/\ln(C_\alpha) \in [0,1]$ for $C_\alpha \geq e$, we have $f(x) \leq f(\tilde{x}) = -1 - C_\alpha + C_\alpha/\ln(C_\alpha) + C_\alpha \ln(\ln(C_\alpha))/\ln(C_\alpha) \leq 0$ for $0 \leq x \leq 1$. ∎

Remark 15.3.18
The conclusion of the previous proposition is often stated as there exists a constant D_α such that $\rho(Q_\alpha) \leq 1 + D_\alpha \Delta t$ for all Δt. The constant D_α is C_α/T in the proposition above. ♠

If Q_α is a normal operator, we have a strong criteria for ℓ^2 stability.

Proposition 15.3.19 (Lax)

Consider a finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ for all $j \geq 0$. If Q_α is a normal operator, then the finite difference scheme is ℓ^2-stable if and only if there exists a constant C_α such that $\rho(Q_\alpha) \leq 1 + C_\alpha/M$ for all M.

Proof.
Since Q_α is normal, we have that $\left\|Q_\alpha^2\right\|_2 = \|Q_\alpha\|^2$. The reader is asked to prove this result in Exercise 15.3. By induction, we have $\left\|Q_\alpha^{2^k}\right\|_2 = \|Q_\alpha\|^{2^k}$ for all $k \geq 0$.

From $\rho\left(Q_\alpha\right) = \lim_{k\to\infty} \|Q_\alpha^k\|^{1/k}$, we get

$$\rho\left(Q_\alpha\right) = \lim_{k\to\infty} \left\|Q_\alpha^{2^k}\right\|^{1/2^k} \lim_{k\to\infty} \left(\|Q_\alpha\|^{2^k}\right)^{1/2^k} = \|Q_\alpha\|_2$$

It follows from Proposition 15.3.16 that $\rho\left(Q_\alpha\right) \leq 1 + C_\alpha/M$ for a constant C_α and all M is a sufficient condition for the ℓ^2 stability of the finite difference scheme. It follows from Proposition 15.3.17 that $\rho\left(Q_\alpha\right) \leq 1 + C_\alpha/M$ for a constant C_α and all M is a necessary condition for the ℓ^2 stability of the finite difference scheme. ∎

The main result of this section is the following.

Theorem 15.3.20

Consider a finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ where $g_j \in \ell^2$. If this finite difference scheme is ℓ^2-stable and consistent, then it is ℓ^2-convergent.

A proof of this result is given in [18] (Theorem 6.22). They also prove that ℓ^2-convergence implies ℓ^2-stability. They provide many more criteria to determine if a finite difference scheme is ℓ^2-stable.

15.3.3 Stability Analysis with Fourier Transforms (von Neumann's Method)

We first review some concepts of functional analysis that will be useful to justify the von Neumann's method.

We consider the space $L^2([0,2\pi])$ of all integrable functions $f : [2,\pi] \to \mathbb{C}$ such that

$$\|f\|_2^2 = \frac{1}{2\pi}\int_0^{2\pi} |f(x)|^2 \,\mathrm{d}x < \infty \ .$$

We have used the same notation for the norm in $\ell^2(\mathbb{Z})$ and the norm in $L^2([0,2\pi])$. The reader should be able by the context to determine which norm is used.

It is well know in functional analysis that there is an isometry[4] between these two spaces defined by

$$\begin{aligned} \Phi : L^2[0,2\pi] &\to \ell^2(\mathbb{Z}) \\ f &\mapsto \hat{f} \end{aligned}$$

where

$$\hat{f}(k) = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-kx\,i}\,\mathrm{d}x$$

for $k \in \mathbb{Z}$ and $i \in \mathbb{C}$ is such that $i^2 = -1$. The function $\hat{f}$ is the Fourier transform of f. We could have considered only real value functions but then $\cos(k\pi)$ and $\sin(k\pi)$ would have to be used to define Φ and the notation becomes messy. We will therefore stick to complex valued functions for a while. The equation $\|f\|_2 = \|\hat{f}\|_2$ is known as Parseval equality.

The inverse Fourier transform is define by

$$\begin{aligned} \Phi^{-1} : \ell^2(\mathbb{Z}) &\to L^2[0,2\pi] \\ g &\mapsto \check{g} \end{aligned}$$

where

$$\check{g}(x) = \sum_{k\in\mathbb{Z}} g(k)e^{kx\,i}$$

[4] An isometry between two normed spaces X and Y is a one-to-one and onto mapping $F : X \to Y$ such that the norm of x is equal to the norm of $F(x)$ for all $x \in X$.

for $x \in [0, 2\pi]$. We also have that $\|g\|_2 = \|\check{g}\|_2$.

If we apply the inverse Fourier transform on both sides of $g_{j+1} = Q_\alpha(g_j)$, we get

$$\Phi^{-1}(g_{j+1}) = \Phi^{-1}(Q_\alpha(g_j)) = \left(\Phi^{-1} \circ Q_\alpha \circ \Phi\right)(\Phi^{-1}(g_j)) \ .$$

If we set $\check{Q}_\alpha = \Phi^{-1} \circ Q_\alpha \circ \Phi$, we get $\check{g}_{j+1} = \check{Q}_\alpha(\check{g}_j)$ with $\check{Q}_\alpha : L^2([0, 2\pi]) \to L^2([0, 2\pi])$ a linear mapping.

Since $\Phi : L^2([0, 2\pi]) \to \ell^2(\mathbb{Z})$ is an isometry with inverse Φ^{-1}, we have that they are both of induced norm 1. Hence,

$$\left\|\check{Q}_\alpha^j\right\|_2 = \left\|\left(\Phi^{-1} \circ Q_\alpha \circ \Phi\right)^j\right\|_2 = \left\|\Phi^{-1} \circ Q_\alpha^j \circ \Phi\right\|_2 = \left\|Q_\alpha^j\right\|_2$$

for all $j > 0$. We have proved that the finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ for $j \geq 0$ is ℓ^2-stable if there exists a constant C such that $\|\check{Q}_\alpha^j\|_2 \leq C$ for all $j \geq 0$.

To determine if a finite difference scheme of the form $g_{j+1} = Q_\alpha(g_j)$ with $g_j \in \ell^2(\mathbb{Z})$ is ℓ^2-stable, we have to prove that there exists a constant C_α such that $\|Q_\alpha^j\|_2 \leq C_\alpha$ or $\|\check{Q}_\alpha^j\|_2 \leq C_\alpha$ for all j. This may not be easy to do. Even if Q_α is normal, proving that the eigenvalues of Q_α are less or equal to 1 in absolute value may not be trivial.

We need a detailed description of the action of Q_α and $\check{Q}_\alpha$ to be able to determine the ℓ^2 stability of the finite difference scheme. As we have seen in a previous example, we can express $g_{j+1} = Q_\alpha(g_j)$ as

$$g_{j+1}(k) = Q_\alpha(g_j)(k) = \sum_{s \in \mathbb{Z}} q_{k,s}\, g_j(s) \ ,$$

where $q_{k,s}$ is the (k, s) component of the matrix Q_α. We first observe that for all our finite difference schemes, we have that $q_{k,s} = q_{k-1,s-1}$. Thus

$$g_{j+1}(k) = Q_\alpha(g_j)(k) = \sum_{s \in \mathbb{Z}} q_{k,s}\, g_j(s) = \sum_{s \in \mathbb{Z}} q_{0,s-k}\, g_j(s) = \sum_{s \in \mathbb{Z}} q_{0,s}\, g_j(s + k)$$

for all $k \in \mathbb{Z}$. So, for the rest of the discussion, we will assume that $g_{j+1} = Q_\alpha(g_j)$ is given by

$$g_{j+1}(k) = \sum_{s \in \mathbb{Z}} q_s\, g_j(s + k) \tag{15.3.10}$$

for all $k \in \mathbb{Z}$, where $q_s = q_{0,s}$. We also observe that for all our finite difference schemes, the sum in (15.3.10) is finite. There is only a finite number of q_s that are non-null.

We have that

$$\begin{aligned}
\check{g}_{j+1}(x) &= \Phi^{-1}\left(Q_\alpha(g_j)\right)(x) = \sum_{k \in \mathbb{Z}} \left(\sum_{s \in \mathbb{Z}} q_s\, g_j(s + k) \right) e^{kxi} = \sum_{s \in \mathbb{Z}} q_s \left(\sum_{k \in \mathbb{Z}} g_j(s + k) e^{kxi} \right) \\
&= \sum_{s \in \mathbb{Z}} q_s \left(\sum_{r \in \mathbb{Z}} g_j(r) e^{(r-s)xi} \right) = \sum_{s \in \mathbb{Z}} q_s \underbrace{\left(\sum_{r \in \mathbb{Z}} g_j(r) e^{rxi} \right)}_{=\check{g}_j(x)} e^{-sxi} = \left(\sum_{s \in \mathbb{Z}} q_s e^{-sxi} \right) \check{g}_j(x) \ .
\end{aligned}$$

Let $\tilde{Q}_\alpha(x) = \sum_{s\in\mathbb{Z}} q_s e^{-sxi}$ for $x \in \mathbb{R}$. The action of $\check{Q}_\alpha$ on a function in $f \in L^2([0,2\pi])$ is just the product $\tilde{Q}_\alpha f$.

Hence,

$$\|\check{Q}_\alpha\|_2 = \sup_{\substack{f\in L^2([0,2\pi]) \\ \|f\|_2=1}} \|\check{Q}_\alpha(f)\|_2 = \sup_{\substack{f\in L^2([0,2\pi]) \\ \|f\|_2=1}} \|\tilde{Q}_\alpha f\|_2 = \|\tilde{Q}_\alpha\|_2 \ ,$$

where the last norm is just the L^2-norm of the function $\tilde{Q}_\alpha$.

To use Proposition 15.3.16 to show that a finite difference scheme is ℓ^2-stable, we may simply show that

$$\|\tilde{Q}_\alpha\|_2 \le \|\tilde{Q}_\alpha\|_\infty \le 1 + C_\alpha \Delta t$$

for some constant C_α.

Example 15.3.21

For the finite difference scheme presented in Algorithm 15.2.1, we have (15.3.10) with $q_{-1} = \alpha$, $q_0 = 1 - 2\alpha$, $q_1 = \alpha$ and $q_s = 0$ otherwise, where $\alpha = \dfrac{c^2\Delta t}{(\Delta x)^2}$. Thus

$$\tilde{Q}_\alpha(x) = \alpha e^{-xi} + (1 - 2\alpha) + \alpha e^{xi} = 1 - 2\alpha(1 - \cos(x))$$

Since

$$\|\tilde{Q}_\alpha(x)\|_\infty = \sup_{x\in[0,2\pi]} |1 - 2\alpha(1 - \cos(x))| \le 1$$

for $2\alpha \le 1$, we have that the finite difference scheme in Algorithm 15.2.1 is ℓ^2-stable for $\dfrac{c^2\Delta t}{(\Delta x)^2} \le \dfrac{1}{2}$. ♣

It could be trickier to use the theory developed above to prove that an implicit finite difference scheme is ℓ^2-stable because we may not have an explicit formulation for Q_α. For instance, the matrix Q_α for the Crank-Nicolson finite difference scheme is given by $Q_\alpha = -J^{-1}K$, where J and K are defined in Example 15.3.11.

We do not need to know Q_α to find $\check{Q}_\alpha$. Consider an implicit finite difference scheme of the form $A_\alpha(g_{j+1}) = B_\alpha(g_j)$ for $g_j \in \ell^2(\mathbb{Z})$, where $A_\alpha, B_\alpha : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ are two bounded linear mapping. This relation can be rewritten explicitly as

$$\sum_{s\in\mathbb{Z}} a_{k,s} g_{j+1}(s) = \sum_{s\in\mathbb{Z}} b_{k,s}\, g_j(s) \tag{15.3.11}$$

for all $k \in \mathbb{Z}$, where $a_{k,s}$ is the (k,s) component of the infinite matrix A_α and $b_{k,s}$ is the (k,s) component of the infinite matrix B_α. For our finite difference schemes, we have $a_{k,s} = a_{k-1,s-1}$ and $b_{k,s} = b_{k-1,s-1}$ for all k and s. As we did above for Q_α, we can rewrite (15.3.11) as

$$\sum_{s\in\mathbb{Z}} a_s g_{j+1}(s+k) = \sum_{s\in\mathbb{Z}} b_s\, g_j(s+k) \tag{15.3.12}$$

for all $k \in \mathbb{Z}$, where $a_s = a_{0,s}$ and $b_s = b_{0,s}$.

Hence, proceeding as we have done for Q_α, we have

$$\begin{aligned}\Phi^{-1}\left(A_\alpha(g_{j+1})\right) = \Phi^{-1}\left(B_\alpha(g_j)\right) &\Rightarrow \sum_{k\in\mathbb{Z}}\left(\sum_{s\in\mathbb{Z}} a_s\, g_{j+1}(s+k)\right)e^{kxi} = \sum_{k\in\mathbb{Z}}\left(\sum_{s\in\mathbb{Z}} b_s\, g_j(s+k)\right)e^{kxi}\\ &\Rightarrow \sum_{s\in\mathbb{Z}} a_s \underbrace{\left(\sum_{r\in\mathbb{Z}} g_{j+1}(r)e^{rxi}\right)}_{=\check{g}_{j+1}(x)} e^{-sxi} = \sum_{s\in\mathbb{Z}} b_s \underbrace{\left(\sum_{r\in\mathbb{Z}} g_j(r)e^{rxi}\right)}_{=\check{g}_j(x)} e^{-sxi}\\ &\Rightarrow \left(\sum_{s\in\mathbb{Z}} a_s e^{-sxi}\right)\check{g}_{j+1}(x) = \left(\sum_{s\in\mathbb{Z}} b_s e^{-sxi}\right)\check{g}_j(x)\end{aligned}$$

for $x \in \mathbb{R}$. Hence,

$$\check{g}_{j+1}(x) = \left(\sum_{s\in\mathbb{Z}} a_s e^{-sxi}\right)^{-1}\left(\sum_{s\in\mathbb{Z}} b_s e^{-sxi}\right)\check{g}_j(x)$$

Let

$$\tilde{Q}_\alpha(x) = \left(\sum_{s\in\mathbb{Z}} a_s e^{-sxi}\right)^{-1}\left(\sum_{s\in\mathbb{Z}} b_s e^{-sxi}\right)$$

for $x \in \mathbb{R}$. The action of $\check{Q}_\alpha$ on a function in $f \in L^2([0,2\pi])$ is just the product $\tilde{Q}_\alpha f$.

Remark 15.3.22

The ℓ^2 theory above can be expanded to finite difference scheme that are more then one-step schemes.

Suppose that we have a finite difference scheme of the form

$$\sum_{s\in\mathbb{Z}} a_{k,s} g_{j+1}(s) = \sum_{s\in\mathbb{Z}} b_{k,s}\, g_j(s) + \sum_{s\in\mathbb{Z}} c_{k,s}\, g_{j-1}(s) \tag{15.3.13}$$

for all $k \in \mathbb{Z}$. Suppose that we assume, as we did before, that $a_{k,s} = a_{k-1,s-1}$, $b_{k,s} = b_{k-1,s-1}$ and $c_{k,s} = c_{k-1,s-1}$ for all k and s. Then (15.3.13) can be written as

$$\sum_{s\in\mathbb{Z}} a_s g_{j+1}(s+k) = \sum_{s\in\mathbb{Z}} b_s\, g_j(s+k) + \sum_{s\in\mathbb{Z}} c_s\, g_{j-1}(s+k) \tag{15.3.14}$$

for all $k \in \mathbb{Z}$, where $a_s = a_{0,s}$, $b_s = b_{0,s}$ and $c_s = c_{0,s}$.

Using the inverse Fourier transform as we did before, we get

$$\left(\sum_{s\in\mathbb{Z}} a_s e^{-sxi}\right)\check{g}_{j+1}(x) = \left(\sum_{s\in\mathbb{Z}} b_s e^{-sxi}\right)\check{g}_j(x) + \left(\sum_{s\in\mathbb{Z}} c_s e^{-sxi}\right)\check{g}_{j-1}(x)$$

for $x \in \mathbb{R}$. This is a finite difference equation for $\check{g}_j$. The characteristic polynomial of this finite difference equation is

$$R(x)\big(\lambda(x)\big)^2 + S(x)\lambda(x) + T(x) = 0\ , \tag{15.3.15}$$

where $R(x) = \sum_{s\in\mathbb{Z}} a_s e^{-sxi}$, $S(x) = \sum_{s\in\mathbb{Z}} b_s e^{-sxi}$ and $T(x) = \sum_{s\in\mathbb{Z}} c_s e^{-sxi}$. The solution of this finite difference equation is of the form

$$\check{g}_j(x) = C_1(x)\big(\lambda_1(x)\big)^j + C_2(x)\big(\lambda_2(x)\big)^j$$

if the characteristic polynomial at x has two distinct roots $\lambda_1(x)$ and $\lambda_2(x)$, or of the form

$$\breve{g}_j(x) = C_1(x)\big(\lambda_1(x)\big)^j + C_2(x)j\big(\lambda_1(x)\big)^j$$

if $\lambda_1(x) = \lambda_2(x)$.

Doing a stability analysis using this approach seems to be a formidable task. However, it is often very simple in practice as it is illustrated in Item 4 of Remark 15.7.4 for the finite difference scheme in Algorithm 15.2.11 used to numerically solve the wave equation. ♠

Example 15.3.23

For the Crank-Nicolson scheme presented in Algorithm 15.2.2, we have (15.3.12) with $a_{-1} = -\alpha$, $a_0 = 1 + 2\alpha$, $a_1 = -\alpha$, $b_{-1} = \alpha$, $b_0 = 1 - 2\alpha$, $b_1 = \alpha$, and $a_s = b_s = 0$ otherwise, where $\alpha = \dfrac{c^2\Delta t}{2(\Delta x)^2}$. Thus

$$\tilde{Q}_\alpha(x) = \frac{\alpha e^{-xi} + (1-2\alpha) + \alpha e^{xi}}{-\alpha e^{-xi} + (1+2\alpha) - \alpha e^{xi}} = \frac{1 - 2\alpha(1-\cos(x))}{1 + 2\alpha(1 - cos(x))} \ .$$

for all x. Since $\big\|\tilde{Q}_\alpha(x)\big\|_\infty \leq 1$ independently of the value of α, we have that the Crank-Nicolson scheme is unconstrained ℓ^2-stable. ♣

15.3.4 L^2 Stability

There is another approach to the theory of convergence, consistency and stability that we present briefly in this section. It is often presented as the von Neumann's method in many books. The L^2 notation in this section is not widely used but we use it to distinguish the definition of stability presented in this section from the definition of stability presented in other sections.

We still consider the Cauchy problem presented in Section 15.3.2. The main difference with the previous approach is that we now assume that the approximations $w_{i,j}$ of $u_{i,j} = u(x_i, t_j)$ are given by functions $h_j \in L^2([0, 2\pi])$ for $0 \leq j \leq M$. So, $w_{k,j} = h_j(x_k)$ for $0 \leq k \leq N$ and $0 \leq j \leq M$.

We now define the stability as it follows.

Definition 15.3.24

A finite difference scheme is **L^2-stable** if there exists a constant C_α such that $\|h_j\|_2 \leq C_\alpha\|h_0\|_2$ for $0 \leq j \leq M$ and all M.

To motivate the previous definition, we go back to our general form for a finite difference scheme

$$\sum_{r\in\mathbb{Z}} a_r w_{k+r,j+1} = \sum_{r\in\mathbb{Z}} b_r w_{k+r,j} \qquad (15.3.16)$$

where m is a non-negative integer. Do not forget that a_r and b_r may depend on a relation between Δx and Δt. Also, the two summations above are in fact finite for the finite difference schemes that we consider.

Example 15.3.25
As we have seen, the Crank-Nicholson scheme is of this form (15.3.16) with $a_{-1} = -\alpha$, $a_0 = 1+2\alpha$, $a_1 = -\alpha$, $b_{-1} = \alpha$, $b_0 = 1-2\alpha$, $b_1 = \alpha$ and $a_k = b_k = 0$ otherwise. ♣

We may express each h_j using Fourier series to get $h_j(x) = \sum_{k\in\mathbb{Z}} A_{k,j} e^{kxi}$, where i is the complex number such that $i^2 = -1$. We have that $g(x) = h_0(x) = \sum_{k\in\mathbb{Z}} A_{k,0} e^{kxi}$ with $A_{k,0} = \frac{1}{2\pi}\int_0^{2\pi} g(x) e^{-kxi}\,\mathrm{d}x$.

We expand (15.3.16) to the finite difference equation

$$\sum_{r\in\mathbb{Z}} a_r h_{j+1}(x + r\Delta x) = \sum_{r\in\mathbb{Z}} b_r h_j(x+r\Delta x) \ . \tag{15.3.17}$$

Using the Fourier series of h_j, we get

$$\begin{aligned}
&\sum_{r\in\mathbb{Z}} a_r \left(\sum_{k\in\mathbb{Z}} A_{k,j+1} e^{k(x+r\Delta x)i}\right) = \sum_{r\in\mathbb{Z}} b_r \left(\sum_{k\in\mathbb{Z}} A_{k,j} e^{k(x+r\Delta x)i}\right) \\
&\quad\Rightarrow \sum_{k\in\mathbb{Z}} \left(A_{k,j+1} \sum_{r\in\mathbb{Z}} a_r e^{kr\Delta x\, i}\right) e^{kx\,i} = \sum_{k\in\mathbb{Z}} \left(A_{k,j} \sum_{r\in\mathbb{Z}} b_r e^{kr\Delta x\, i}\right) e^{kx\,i} \\
&\quad\Rightarrow \sum_{k\in\mathbb{Z}} \left(A_{k,j+1} \sum_{r\in\mathbb{Z}} a_r e^{kr\Delta x\, i} - A_{k,j} \sum_{r\in\mathbb{Z}} b_r e^{kr\Delta x\, i}\right) e^{kx\,i} = 0 \ .
\end{aligned}$$

Thus, for all k,

$$A_{k,j+1}\alpha_k - A_{k,j}\beta_k = 0 \tag{15.3.18}$$

for $0 \le j < M$, where

$$\alpha_k = \sum_{r\in\mathbb{Z}} a_r e^{kr\Delta x\, i} \quad \text{and} \quad \beta_k = \sum_{r\in\mathbb{Z}} b_r e^{kr\Delta x\, i} \ .$$

We therefore have that

$$A_{k,j} = \left(\frac{\beta_k}{\alpha_k}\right)^j A_{k,0}$$

for $j \ge 0$.

Since $\alpha_{k+N} = \alpha_k$ and $\beta_{k+N} = \beta_k$ for all k because $\Delta x = 2\pi/N$, there is only a finite number of ratios $\lambda_k = \beta_k/\alpha_k$ to determine. We only need to compute λ_k for $0 \le k < N$.

If there exists a constant C_α such that $|\lambda_k|^j \le C_\alpha$ for $0 \le j \le M$ and $0 \le k < N$, then $|A_{k,j}|^2 \le C_\alpha^2 |A_{k,0}|^2$ for $0 \le j \le M$ and $k \in \mathbb{Z}$. We get from Parseval equality that

$$\|h_j\|_2^2 = \left\|\hat{h}_j\right\|_2^2 = \sum_{k\in\mathbb{Z}} |A_{k,j}|^2 \le C_\alpha^2 \sum_{k\in\mathbb{Z}} |A_{k,0}|^2 = C_\alpha^2 \|\hat{h}_0\|_2^2 = C_\alpha^2 \|h_0\|_2^2$$

for $0 \le j \le M$. We have shown that $\|h_j\|_2 \le C_\alpha \|h_0\|_2$ if $|\lambda_k|^j \le C$ for $0 \le j \le M$ and $0 \le k < N$. This last condition is satisfied if and only if $|\lambda_k| \le 1$ for $0 \le k < N$ because M can be as large as we want. Do not forget that there may be a restriction on N and M because $|\lambda_k|^j \le C$ may be true only if a relation between Δx and Δt is satisfied; a relation that is inherited from the dependence of a_r and b_r on the parameter α that we have defined for the finite difference schemes presented in Section 15.2.

Remark 15.3.26
We have considered one-step finite difference schemes in the previous discussion but this method can be generalized to two or more step schemes. If instead of (15.3.16) we have

$$\sum_{r\in\mathbb{Z}} a_r w_{i+r,j+1} = \sum_{r\in\mathbb{Z}} b_r w_{i+r,j} + \sum_{r\in\mathbb{Z}} c_r w_{i+r,j-1} \ , \tag{15.3.19}$$

then instead of (15.3.17) we get

$$\sum_{r\in\mathbb{Z}} a_r h_{j+1}(x + r\Delta x) = \sum_{r\in\mathbb{Z}} b_r h_j(x + r\Delta x) + \sum_{r\in\mathbb{Z}} c_r h_{j-1}(x + r\Delta x) \tag{15.3.20}$$

for $j \geq 0$. We have that $h_0(x) = g(x)$ and we may assume that $h_{-1}(x)$ is a given periodic function of period 2π. For instance, in the case of the wave equation in Section 15.2.3, we will have $h_{-1}(x) = h_1(x) - 2f(x)\Delta t$.

Using the Fourier series of the h_j, (15.3.20) yields

$$\begin{aligned}
&\sum_{r\in\mathbb{Z}} a_r \left(\sum_{k\in\mathbb{Z}} A_{k,j+1} e^{k(x+r\Delta x)i} \right) = \sum_{r\in\mathbb{Z}} b_r \left(\sum_{k\in\mathbb{Z}} A_{k,j} e^{k(x+r\Delta x)i} \right) + \sum_{r\in\mathbb{Z}} c_r \left(\sum_{k\in\mathbb{Z}} A_{k,j-1} e^{k(x+r\Delta x)\, i} \right) \\
&\quad \Rightarrow \sum_{k\in\mathbb{Z}} \left(A_{k,j+1} \sum_{r\in\mathbb{Z}} a_r e^{kr\Delta x\, i} - A_{k,j} \sum_{r\in\mathbb{Z}} b_r e^{kr\Delta x\, i} - A_{k,j-1} \sum_{r\in\mathbb{Z}} c_r e^{kr\Delta x\, i} \right) e^{kx\, i} = 0 \ .
\end{aligned}$$

Thus, for all k,

$$A_{k,j+1}\alpha_k - A_{k,j}\beta_k - A_{k,j-1}\gamma_k = 0 \tag{15.3.21}$$

for $0 \leq j < M$, where

$$\alpha_k = \sum_{r\in\mathbb{Z}} a_r e^{kr\Delta x\, i} \quad , \quad \beta_k = \sum_{r\in\mathbb{Z}} b_r e^{kr\Delta x\, i} \quad \text{and} \quad \gamma_k = \sum_{r\in\mathbb{Z}} c_r e^{kr\Delta x\, i} \ . \tag{15.3.22}$$

If the characteristic polynomial $\alpha_k\lambda^2 - \beta_k\lambda - \gamma_k$ has two distinct roots $\lambda_{k,1}$ and $\lambda_{k,2}$, then the general solution of (15.3.21) is of the form

$$A_{k,j} = C_{k,1}\lambda_{k,1}^j + C_{k,2}\lambda_{k,2}^j$$

for $0 \leq j < M$ and some constants $C_{k,1}$ and $C_{k,2}$. If $\lambda_{k,1} = \lambda_{k,2}$, then

$$A_{k,j} = C_{k,1}\lambda_{k,1}^j + C_{k,2}\, j\lambda_{k,1}^j$$

for $0 \leq j < M$ and some constants $C_{k,1}$ and $C_{k,2}$.

Since $\alpha_{k+N} = \alpha_k$, $\beta_{k+N} = \beta_k$ and $\gamma_{k+N} = \gamma_k$ for all k because $\Delta x = 2\pi/N$, there is only a finite number of roots to determine. We only need to compute $\lambda_{k,1}$ and $\lambda_{k,2}$ for $0 \leq k < N$. We could show that the finite difference scheme is "stable" when $|\lambda_{k,1}| \leq 1$ and $|\lambda_{k,2}| \leq 1$. If $|\lambda_{k,1}| = |\lambda_{k,2}| = 1$, then we also need $\lambda_{k,1} \neq \lambda_{k,2}$.

This technique is illustrated in Item 1 of Remark 15.7.4 for the finite difference scheme in Algorithm 15.2.11 used to numerically solve the wave equation.

We will not pursue on this subject. The generalization to higher dimension in the rest of this section can instead be used to handle finite difference schemes that are more than one-step schemes. ♠

Proceeding exactly as we have just done, the previous discussion can be generalized to the finite difference scheme of the form

$$\sum_{r\in\mathbb{Z}} J_r \mathbf{w}_{j+1} = \sum_{r\in\mathbb{Z}} K_r \mathbf{w}_j \ , \tag{15.3.23}$$

where J_r and K_r are $n\times n$ matrices, and $\mathbf{w}_j \in \mathbb{R}^n$ for $j\geq 0$.

Example 15.3.27
The Crank-Nicolson scheme is of the form (15.3.16) with $J_0 = -J$ and $K_0 = K$ with K and J defined in (15.2.6) and (15.2.9) respectively, and $J_r = K_r = 0$ otherwise. We also have that $n = N-1$. ♣

If we assume that $\mathbf{w}_j = h_j(x_k)$ for $h_j \in L^2([0,2\pi],\mathbb{R}^n)$ [5], we may expand (15.3.23) to the finite difference equation

$$\sum_{r\in\mathbb{Z}} J_r h_{j+1}(x+r\Delta x) = \sum_{r\in\mathbb{Z}} K_r h_j(x+r\Delta x) \ . \tag{15.3.24}$$

If we substitute the Fourier series $h_j(x) = \sum_{k\in\mathbb{Z}} \mathbf{A}_{k,j} e^{kx\,i}$, where $A_{k,j} \in \mathbb{R}^n$, in the previous equation, we find that

$$\mathbf{A}_{k,j+1} = Q_k \mathbf{A}_{k,j} \ , \tag{15.3.25}$$

where

$$Q_k = \left(\sum_{r=-m}^{m} K_r e^{kr\Delta x\, i}\right)^{-1} \left(\sum_{r=-m}^{m} J_r e^{kr\Delta x\, i}\right) \ .$$

By induction, we get from (15.3.25) that

$$\mathbf{A}_{k,j} = Q_k^j \mathbf{A}_{k,0} \quad , \quad j\geq 0 \ . \tag{15.3.26}$$

If there exists C_α such that $\|Q_k^j\|_2 \leq C_\alpha$ for $0\leq j\leq M$ and $k\in\mathbb{Z}$, we get from Parseval equality that

$$\|h_j\|_2^2 = \|\hat{h}_j\|_2^2 = \sum_{k\in\mathbb{Z}} \|\mathbf{A}_{k,j}\|_2^2 \leq C_\alpha^2 \sum_{k\in\mathbb{Z}} \|\mathbf{A}_{k,0}\|_2^2 = C_\alpha^2 \|\hat{h}_0\|_2^2 = C_\alpha^2 \|h_0\|_2^2$$

for $0\leq j\leq M$. We have shown that $\|h_j\|_2 \leq C_\alpha \|h_0\|_2$ if there exists C_α such that $\|Q_k^j\|_2 \leq C_\alpha$ for $0\leq j\leq M$ and $k\in\mathbb{Z}$. Again, because of the periodicity of Q_k, we only have to consider $0\leq k<N$. We have the more precise result that follows.

Proposition 15.3.28

A finite difference scheme of the form (15.3.23) is L^2**-stable** if and only if there exists C_α such that $\|Q_k^j\|_2 \leq C_\alpha$ for $0\leq j\leq M$, $k\in\mathbb{Z}$ and $N>0$.

Proof.
We have proved above that the condition $\|Q_k^j\|_2 \leq C$ for $0\leq j\leq M$ and $k\in\mathbb{Z}$ was sufficient.

[5] The functions $h:[0,2\pi]\to\mathbb{R}^n$ such that each component is in $L^2([0,2\pi])$.

We now prove that it is necessary. Suppose that $\|Q_{k_0}^{j_0}\|_2 > C_\alpha$ for some k_0 and j_0. Choose $\mathbf{w} \in \mathbb{R}^n$ such that $\|Q_{k_0}^{j_0}\mathbf{w}\|_2 > C_\alpha\|\mathbf{w}\|_2$. If we consider a boundary value problem such that $g(x) = \mathbf{w}e^{k_0 x\, i}$, then $h_0(x) = \mathbf{A}_{k_0,0}e^{k_0 x\, i}$ with $\mathbf{A}_{k_0,0} = \mathbf{w}$. From (15.3.26), we have $\mathbf{A}_{k_0,j} = Q_{k_0}^j\mathbf{w}$. Thus

$$\|h_{j_0}\|_2^2 = \|\hat{h}_{j_0}\|_2^2 \geq \|\mathbf{A}_{k_0,j_0}\|_2^2 = \|Q_{k_0}^{j_0}\mathbf{w}\|_2^2 > C_\alpha^2\|\mathbf{w}\|_2^2 = C_\alpha^2\|\mathbf{A}_{k_0,0}\|_2^2 = C_\alpha^2\|\hat{h}_0\|_2^2 = C_\alpha^2\|h_0\|_2^2 \ .$$

This contradicts $\|h_j\|_2 \leq C_\alpha\|h_0\|_2$ for $0 \leq j \leq M$ and all M. ∎

We have a version of the von Neumann criteria, Proposition 15.3.17, in the present context.

Proposition 15.3.29 (von Neumann)

If a finite difference scheme of the form (15.3.23) is L^2**-stable**, then there exists a constant C_α such that $\rho(Q_k) \leq 1 + C_\alpha/M$ for all $k \in \mathbb{Z}$, N and M.

Proof.
Since the finite difference scheme is L^2-stable, we get from Proposition 15.3.28 that $\|Q_k^j\|_2 \leq C_\alpha$ for $0 \leq j \leq M$ and $k \in \mathbb{Z}$. Hence, from Remark 3.1.12, we get

$$\rho^j(Q_k) = \rho(Q_k^j) \leq \|Q_k^j\|_2 \leq C_\alpha$$

for $0 \leq j \leq M$ and $k \in \mathbb{Z}$. As in the proof of Proposition 15.3.17, we get for $j = M$ that

$$\rho(Q_k) \leq C_\alpha^{1/M} \leq 1 + \frac{C_\alpha}{M} \ .$$ ∎

We can deduce from Proposition 15.3.28 a sufficient condition to determine L^2 stability.

Proposition 15.3.30

If the matrix Q_k is normal, then the finite difference scheme of the form (15.3.23) is L^2-stable if $\rho(Q_k) \leq 1$ for all $k \in \mathbb{Z}$, N and M.

Proof.
Since the matrix Q_k is normal, we have $\rho(Q_k) = \|Q_k\|_2$. Hence, $\|Q_k^j\|_2 \leq \|Q_k\|_2^j = \rho^j(Q_k) \leq 1$ for $0 \leq j \leq M$, $k \in \mathbb{Z}$ and $N > 0$. ∎

15.3.5 Matrix Method

There is yet another approach to determine the convergence and stability of a finite difference scheme. The method presented in this section is called the **Matrix Method** because it is based on the matrix representation of the finite difference schemes as presented in Section 15.2. We are back in finite dimension.

We only give a brief description of this method though it is widely used in engineering for historical reasons. We focus our discussion on explicit one-step finite difference schemes.

Suppose that $P_\Delta(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots) = F(x_i, t_j)$ for all (i,j) such that $(x_i, t_j) \in R_\Delta^o$ can be expressed in vector form as $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ for $0 \leq j < M$, where

1. Q_α is a $(N-1)\times(N-1)$ matrix,
2. $\mathbf{w}_j = \begin{pmatrix} w_{1,j} & w_{2,j} & w_{3,j} & \ldots & w_{N-1,j}\end{pmatrix}^\top$ and
3. $\mathbf{B}_j \equiv \mathbf{B}_j\left(F, \{w_{r,s} : x_{r,s} \in \partial R\}\right) \in \mathbb{R}^{N-1}$; namely, $\mathbf{B}_j$ is a vector valued function of F evaluated at the mesh points and the values $w_{i,j}$ on the boundary of the domain R which, for the purpose of this section, we assume are the values of the solution u on the boundary. Note that boundary also includes the initial values.

The vectors $\mathbf{w}_j$ for $0 < j \leq M$ represent all the values $w_{i,j}$ for (i,j) such that $(x_i, t_j) \in R_\Delta^o$.

As we have done before, the index α in Q_α is to indicate that the linear operator may depend on a relation between Δx and Δt.

Example 15.3.31
For the heat equation with forcing for instance, the finite difference scheme given in Algorithm 15.2.1 can be expressed in the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ where $Q_\alpha = -K$ for K given in (15.2.6) and

$$\mathbf{B}_j = \begin{pmatrix} \alpha w_{0,j} + f(x_1, t_j)\Delta t \\ f(x_2, t_j)\Delta t \\ \vdots \\ f(x_{N-2}, t_j)\Delta t \\ \alpha w_{N,j} + f(x_{N-1}, t_j)\Delta t \end{pmatrix} .$$

The Crank-Nicolson scheme given in Algorithm 15.2.2 can be expressed in the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ where $Q_\alpha = -J^{-1}K$ for K given in (15.2.6), J given in (15.2.9) and

$$\mathbf{B}_j = J^{-1} \begin{pmatrix} \alpha\left(w_{0,j+1} + w_{0,j}\right) + \big(f(x_1, t_j) + f(x_1, t_{j+1})\big)\Delta t/2 \\ \big(f(x_2, t_j) + f(x_2, t_{j+1})\big)\Delta t/2 \\ \vdots \\ \big(f(x_{N-2}, t_j) + f(x_{N-2}, t_{j+1})\big)\Delta t/2 \\ \alpha\left(w_{N,j+1} + w_{N,j}\right) + \big(f(x_{N-1}, t_j) + f(x_{N-1}, t_{j+1})\big)\Delta t/2 \end{pmatrix} .$$

♣

We consider the norm on $\mathbb{R}^{N-1}$ defined by

$$\|\mathbf{y}\|_N = \left(\sum_{i=1}^{N-1} y_i^2 \, \Delta x\right)^{1/2}$$

for $\mathbf{y} \in \mathbb{R}^{N-1}$. If $y_i = g(x_i)$ for all i, where $g : [0, L] \to \mathbb{R}$ is a continuous function, then $\|\mathbf{y}\|_N \to \left(\int_0^L g^2(x)\,\mathrm{d}x\right)^{1/2}$ as $N \to \infty$ by definition of the Riemann integral.

We can give new, and not so new, definitions of convergence, consistency and stability.

Definition 15.3.32

A finite difference scheme of the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ with $\mathbf{w}_j \in \mathbb{R}^{N-1}$ is **ℓ^2-convergent** if

$$\sup_{0\leq j\leq M} \|\mathbf{w}_j - \mathbf{u}_j\|_N \to 0 \quad \text{as} \quad \min\{N,M\} \to \infty \ ,$$

where $\mathbf{u}_j = \begin{pmatrix} u_{1,j} & u_{2,j} & u_{3,j} & \ldots & u_{N-1,j}\end{pmatrix}^\top$.

Definition 15.3.33

Given any sufficiently differentiable function $q : R \to \mathbb{R}$, the **local truncation error** of the finite difference scheme of the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ is the expression

$$\tau_j(\Delta x, \Delta t, q) = \frac{1}{\Delta t}\Big(\mathbf{q}_{j+1} - Q_\alpha \mathbf{q}_j - \mathbf{B}_j\big(F, \{\mathbf{q}(x_r, t_s) : x_{r,s} \in \partial R\}\big)\Big)$$

for $j \geq 0$, where $\mathbf{q}_{i,j} = q(i\Delta x . j\Delta t)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$.
A finite difference scheme of the form $\mathbf{w}_{j+1} = Q\mathbf{w}_j + \mathbf{B}_j$ with $\mathbf{w}_j \in \mathbb{R}^{N-1}$ is **ℓ^2-consistent** if

$$\|\tau_j(\Delta x, \Delta t, q)\|_N \to 0 \quad \text{as} \quad \max\{N,M\} \to \infty \ ,$$

for all function $q : R \to \mathbb{R}$.

Remark 15.3.34

It should be pointed out that consistency according to Definition 15.3.6 automatically implies consistency according to the previous definition because the i^{th} component of $\tau_j(\Delta x, \Delta t, q)$ is $\tau_{i,j}(\Delta x, \Delta t, q)$ as defined in Definition 15.3.6. Hence

$$\begin{aligned}\|\tau_j(\Delta x, \Delta t, q)\|_N^2 &= \Delta x \sum_{i=1}^{N-1} \left(\tau_{i,j}(\Delta x, \Delta t, q)\right)^2 \leq (N-1)\Delta x \max_{0<i<N} |\tau_{i,j}(\Delta x, \Delta t, q)|^2 \\ &\leq L \max_{\substack{(i,j)\text{ such that}\\ (x_i y_j)\in R^o}} |\tau_{i,j}(\Delta x, \Delta t, q)|^2 \ .\end{aligned}$$

♠

Definition 15.3.35

A finite difference scheme of the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$ with $\mathbf{w}_j \in \mathbb{R}^{N-1}$ is **ℓ^2-stable** if there exists a constant C_α such that $\|Q_\alpha^j\|_N \leq C_\alpha$ for all $N > 0$ and $j \geq 0$.

A word of caution about the previous definitions of convergence and stability. Because of the dependence of Q_α on α, a relation between Δx and Δt may need to be satisfied to ensure convergence and stability. So, N and M may not be totally independent of each other.

Definition 15.3.35 is a weaker definition of stability than Definition 15.3.8 in the sense that it ensures that round off errors do not increase in ℓ^2-norm instead of in the uniform norm as M increases.

It follows from Theorem 3.1.11 that $\rho(Q_\alpha) \leq \|Q_\alpha\|_N$ for all norms on the space of $(N-1)\times(N-1)$ matrices where $\rho(Q_\alpha)$ is the spectral radius of Q_α. So, if $\|Q_\alpha\|_N \leq 1$, then $\rho(Q_\alpha) \leq 1$; namely, all the eigenvalues of Q_α are less than or equal to 1 in absolute value. We have a partial converse to this statement. If Q_α is a normal matrix (i.e. $Q_\alpha Q_\alpha^\top = Q_\alpha^\top Q_\alpha$) and the induced norm on the $(N-1)\times(N-1)$ matrices is the Euclidean norm $\|\cdot\|_2$ on $\mathbb{R}^{N-1}$, then $\|Q_\alpha\|_2 = \rho(Q_\alpha)$. This is also true if the Euclidean norm is replaced by a multiple of the Euclidean norm as we have for $\|\cdot\|_N$. So, $\|Q_\alpha\|_N = \rho(Q_\alpha)$. Thus $\rho(Q_\alpha) \leq 1$ if and only if $\|Q_\alpha\|_N \leq 1$. We also get that $\|Q_\alpha^j\|_N \leq \|Q_\alpha\|_N^j \leq 1$.

We get the following result.

Proposition 15.3.36

A finite difference scheme of the form $\mathbf{w}_{j+1} = Q_\alpha \mathbf{w}_j + \mathbf{B}_j$, where $\mathbf{w}_j \in \mathbb{R}^{N-1}$ and Q_α is normal, is ℓ^2-stable according to Definition 15.3.35 if all the eigenvalues of Q_α are less than or equal to 1 in absolute value, independently of the value of N.

Remark 15.3.37

As we saw in Section 15.2, we may represent the finite difference scheme formed of (15.3.3) and (15.3.6) as a linear system $A\mathbf{w} = \mathbf{B}$.

There is yet another definition of stability that is frequently used in this situation. A linear system of the form $A\mathbf{w} = \mathbf{B}$ is **stable** if there exists a constant K_α such that $\|\mathbf{w}\| \leq K_\alpha \|A\mathbf{w}\|$ for all $\mathbf{w}$. This definition is reminiscent of the property that well-conditioned linear systems of equations have. If $\mathbf{w}_1$ and $\mathbf{w}_2$ are two solutions of $A\mathbf{w} = \mathbf{B}$, then $\|\mathbf{w}_1 - \mathbf{w}_2\| \leq K_\alpha \|A(\mathbf{w}_1 - \mathbf{w}_2)\|$. So, if the difference between $A\mathbf{w}_1$ and $A\mathbf{w}_2$ is small, then the difference between $\mathbf{w}_1$ and $\mathbf{w}_2$ should also be proportionally small. This is the property that we have associated to well-conditioned systems in Section 4.4. We will not treat this subject. ♠

Remark 15.3.38

There is a link between definition of stability given in Definition 15.3.24 and Definition 15.3.35.

Suppose that $w_{k,j} = g_j(k\Delta x)$ for a continuous function $g_j : [0, 2\pi] \to \mathbb{R}$ for all j. Then

$$\|\mathbf{w}_j\|_N^2 = \sum_{k=1}^{N-1} |w_{k,j}|^2 \Delta x = \sum_{k=1}^{N-1} |g_j(k\Delta x)|^2 \Delta x \to \int_0^{2\pi} |g_j|^2 \,\mathrm{d}x = \|g_j\|_2^2$$

as $N \to \infty$ and so $\Delta x \to 0$.

Given two real numbers $0 < S_1 < 1 < S_2$, there exists N_M large enough such that

$$S_1 \leq \frac{\|\mathbf{w}_j\|_N}{\|g_j\|_2} \leq S_2$$

for $0 \leq j \leq M$ and $N \geq N_M$. We assume that $\|g_j\|_2 > 0$ for $0 \leq j \leq M$ and leave the case $\|g_j\|_2 = 0$ for some j to the reader. Hence,

$$\|\mathbf{w}_j\|_N \leq \frac{C_\alpha S_2}{S_1}\|\mathbf{w}_0\|_N \iff \|g_j\|_2 \leq C_\alpha \|g_0\|_2$$

for $0 \leq j \leq M$ and $N \geq N_M$. ♠

15.3.6 Conclusion

We have seen several definitions of convergence and stability. Some of them were based on a Cauchy problem and so were ignoring the boundary conditions that some partial differential equations may have to satisfied. Which definitions should we use? This depends on the problem and on what we want to achieve.

Uniform approximation of the solution seems to be the ultimate goal to achieve but this is not possible for all finite difference schemes. Some finite difference schemes may have very desirable features other than uniform convergence. So ℓ^2 convergence and stability may be preferable.

There is also the issue of proving stability. Proving stability for uniform approximation is far from trivial when possible. Using Definition 15.3.35 to determine ℓ^2 stability may seem to be the next reasonable option but that requires a nice matrix (usually "near" diagonal) to be able to determine the eigenvalues of Q_α. Using Definitions 15.3.8 and 15.3.14 may be the best options. They may not be considering partial differential equation with boundary conditions but may be good enough to ensure stability. That is our ultimate goal to avoid propagation of round off errors.

The reader may have noticed that we did not mention consistency in this conclusion. The reason is simple. Must of the finite difference schemes (at least those that we have presented) are developed from finite difference formulae as those in Section 15.1 to ensure that the local truncation error is of order greater than one in Δx and Δt. This is enough to ensure consistency.

15.4 Preliminaries of Linear Algebra

We take a little pause to review some notions of linear algebra that will be needed later on.

Proposition 15.4.1

Consider the tri-diagonal matrix

$$Q = \begin{pmatrix} a & b & 0 & 0 & 0 & \ldots & 0 & 0 \\ c & a & b & 0 & 0 & \ldots & 0 & 0 \\ 0 & c & a & b & 0 & \ldots & 0 & 0 \\ 0 & 0 & c & a & b & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & c & a \end{pmatrix}$$

of dimension $n \times n$. The eigenvalues of Q are

$$\lambda_k = a + 2b\sqrt{\frac{c}{b}}\cos\left(\frac{k\pi}{n+1}\right) \quad , \quad 0 < k < n+1 \ .$$

A possible eigenvector associated to λ_k is

$$\mathbf{v}_k = \begin{pmatrix} (c/b)^{1/2}\sin\left(k\pi/(n+1)\right) \\ (c/b)^{2/2}\sin\left(2k\pi/(n+1)\right) \\ \vdots \\ (c/b)^{n/2}\sin\left(nk\pi/(n+1)\right) \end{pmatrix} .$$

Proof.
Let $\mathbf{v}$ be an eigenvector of Q associated to the eigenvalue λ. If we set $v_0 = v_{n+1} = 0$, the equation $A\mathbf{v} = \lambda\mathbf{v}$ can be written as

$$cv_{j-1} + (a-\lambda)v_j + bv_{j+1} = 0 \quad , \quad 1 \leq j \leq n . \tag{15.4.1}$$

To find the solution of this difference equation, we have to find the roots of the characteristic equation

$$b\rho^2 + (a-\lambda)\rho + c = 0 . \tag{15.4.2}$$

Let ρ_1 and ρ_2 be the roots of (15.4.2). Since $\rho_1\rho_2 = c/b \neq 0$, none of the roots is null.

If $\rho_1 = \rho_2$, the solution of (15.4.1) is of the form $v_j = \alpha\rho_1^j + \beta j\rho_1^j$ for $0 \leq j \leq n+1$. Since $v_0 = 0$ implies $\alpha = 0$, we get that $v_{n+1} = 0$ implies $\beta(n+1)\rho_1^{n+1} = 0$. It follows that $\beta = 0$. We find $v_j = 0$ for all j which is not possible for an eigenvector.

We may assume that ρ_1 and ρ_2 are two distinct and non null roots. In this case, the solution of (15.4.1) is of the form

$$v_j = \alpha\rho_1^j + \beta\rho_2^j \quad , \quad 0 \leq j \leq n+1 . \tag{15.4.3}$$

From $v_0 = 0$, we get $0 = \alpha + \beta$. Hence $\beta = -\alpha$. From $v_{n+1} = 0$, we get

$$0 = \alpha\rho_1^{n+1} + \beta\rho_2^{n+1} = \alpha\left(\rho_1^{n+1} - \rho_2^{n+1}\right) .$$

Thus $(\rho_1/\rho_2)^{n+1} = 1$. It follows that ρ_1/ρ_2 is a $(n+1)$ root of the unity; namely,

$$\frac{\rho_1}{\rho_2} = e^{2k\pi i/(n+1)} \quad , \quad 0 \leq k < n+1 . \tag{15.4.4}$$

Hence,

$$\frac{c}{b} = \rho_1\rho_2 = \left(\rho_2 e^{2k\pi i/(n+1)}\right)\rho_2 = \rho_2^2 e^{2k\pi i/(n+1)}$$

yields

$$\rho_2 = \sqrt{\frac{c}{b}}e^{-k\pi i/(n+1)} \quad , \quad 0 \leq k < n+1 .$$

We also get from (15.4.4) that

$$\rho_1 = \sqrt{\frac{c}{b}}e^{k\pi i/(n+1)} \quad , \quad 0 \leq k < n+1 .$$

We have to ignore $k = 0$ because this gives $\rho_1 = \rho_2 = \sqrt{c/b}$ which is impossible as we have shown before.

Finally,

$$-\frac{a-\lambda}{b} = \rho_1 + \rho_2 = \sqrt{\frac{c}{b}} e^{k\pi i/(n+1)} + \sqrt{\frac{c}{b}} e^{-k\pi i/(n+1)} = 2\sqrt{\frac{c}{b}} \cos\left(\frac{k\pi}{n+1}\right) \quad , \quad 0 < k < n+1 \ .$$

Thus, the eigenvalues of Q are

$$\lambda_k = a + 2b\sqrt{\frac{c}{b}} \cos\left(\frac{k\pi}{n+1}\right) \quad , \quad 0 < k < n+1 \ .$$

Since $\beta = \alpha$ in (15.4.3), the eigenvectors $\mathbf{v}_k$ of Q associated to the eigenvalue λ_k will have the components

$$v_{j,k} = \alpha\left(\left(\frac{c}{b}\right)^{j/2} e^{jk\pi i/(n+1)} - \left(\frac{c}{b}\right)^{j/2} e^{-jk\pi i/(n+1)}\right) = 2\alpha\, i \left(\frac{c}{b}\right)^{j/2} \sin\left(\frac{jk\pi}{n+1}\right) \quad , \quad 0 < j < n+1 \ .$$

Since α can be any non-null complex number, we may take $\alpha = -i/2$ to get real eigenvectors

$$\mathbf{v}_k = \begin{pmatrix} (c/b)^{1/2} \sin\left(k\pi/(n+1)\right) \\ (c/b)^{2/2} \sin\left(2k\pi/(n+1)\right) \\ \vdots \\ (c/b)^{n/2} \sin\left(nk\pi/(n+1)\right) \end{pmatrix} \ .$$ ∎

Proposition 15.4.2

Consider the matrix

$$Q = \begin{pmatrix} Q_{1,1} & Q_{1,2} & \dots & Q_{1,s} \\ Q_{2,1} & Q_{2,2} & \dots & Q_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{s,1} & Q_{s,2} & \dots & Q_{s,s} \end{pmatrix} ,$$

where each sub-matrix $Q_{i,j}$ is a matrix of dimension $n \times n$. Suppose that $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ in $\mathbb{R}^n$ are n linearly independent eigenvectors for each matrix $Q_{i,j}$. Let $\lambda_{i,j,k}$ be the eigenvalue of $Q_{i,j}$ associated to the eigenvector $\mathbf{v}_k$ for $1 \leq i,j \leq s$ and $1 \leq k \leq n$. Then the eigenvalues of the $s \times s$ matrices

$$P_k = \begin{pmatrix} \lambda_{1,1,k} & \lambda_{1,2,k} & \dots & \lambda_{1,s,k} \\ \lambda_{2,1,k} & \lambda_{2,2,k} & \dots & \lambda_{2,s,k} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{s,1,k} & \lambda_{s,2,k} & \dots & \lambda_{s,s,k} \end{pmatrix}$$

for $1 \leq k \leq n$ are eigenvalues of Q.

Proof.

Suppose that λ is an eigenvalue of P_k for some k fixed. We will show that there exist a_1, a_2,

$\ldots$, a_s in $\mathbb{R}$ such that

$$\mathbf{v} = \begin{pmatrix} a_1\mathbf{v}_k \\ a_2\mathbf{v}_k \\ \vdots \\ a_s\mathbf{v}_k \end{pmatrix}$$

is an eigenvector of Q associated to the eigenvalue λ.

The following statement about λ and the vector $\mathbf{v}$ above are equivalent:

i. $\mathbf{v}$ is an eigenvector of Q associated to the eigenvalue λ; namely, $Q\mathbf{v} = \lambda\mathbf{v}$.

ii.
$$\sum_{j=1}^{s} a_j Q_{i,j}\mathbf{v}_k = \sum_{j=1}^{s} a_j \lambda_{i,j,k}\mathbf{v}_k = \lambda\, a_i\, \mathbf{v}_k \quad , \quad 1 \le i \le s \ .$$

iii.
$$\sum_{\substack{j=1\\ j\neq i}}^{s} a_j \lambda_{i,j,k}\mathbf{v}_k + a_i\left(\lambda_{i,i,k} - \lambda\right)\mathbf{v}_k = 0 \quad , \quad 1 \le i \le s \ .$$

iv. $R\mathbf{a} = \mathbf{0}$, where

$$R = \begin{pmatrix} \lambda_{1,1,k} - \lambda & \lambda_{1,2,k} & \ldots & \lambda_{1,s,k} \\ \lambda_{2,1,k} & \lambda_{2,2.k} - \lambda & \ldots & \lambda_{2,s,k} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{s,1,k} & \lambda_{s,2,k} & \ldots & \lambda_{s,s,k} - \lambda \end{pmatrix} , \quad \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_s \end{pmatrix} \text{ and } \mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

Since λ is an eigenvalue of P_k. The matrix R is not invertible and, therefore, there exists a non-trivial solution $\mathbf{a}$ of $R\mathbf{a} = \mathbf{0}$. This solution yields the eigenvector $\mathbf{v}$ of Q associated to λ. ∎

15.5 Heat Equation

15.5.1 Algorithm 15.2.1

We study the uniform convergence of Algorithm 15.2.1 which is used to approximate the solution of the heat equation with forcing.

Proposition 15.5.1

The scheme in Algorithm 15.2.1 is consistent.

Proof.
Using the notation introduced in Section 15.3, we have that

$$P\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial y}(x,t), \frac{\partial^2 u}{\partial x^2}(x,t), \ldots\right) = \frac{\partial u}{\partial t}(x,t) - c^2\frac{\partial^2 u}{\partial x^2}(x,t)$$

and

$$P_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) = \frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2}$$

for the finite difference scheme in Algorithm 15.2.1.

The local truncation error of the finite difference scheme in Algorithm 15.2.1 is deduced from (15.2.4) with the function u replaced by the function q. We have

$$\begin{aligned}\tau_{i,j}(\Delta x, \Delta t, q) &= P\left(q(x_i,t_j), \frac{\partial q}{\partial x}(x_i,t_j), \frac{\partial q}{\partial y}(x_i,t_j), \frac{\partial^2 q}{\partial x^2}(x_i,t_j), \ldots\right) \\ &\qquad - P_\Delta\left(q(x_i,t_j), q(x_i,y_{j+1}), q(x_{i+1},t_j), \ldots\right) \\ &= -\frac{1}{2}\frac{\partial^2 q}{\partial t^2}\left(x_i, \rho_{i,j}\right)\Delta t + \frac{c^2}{4!}\left(\frac{\partial^4 q}{\partial x^4}\left(\zeta_{i,j}, t_j\right) + \frac{\partial^4 q}{\partial x^4}\left(\eta_{i,j}, t_j\right)\right)(\Delta x)^2\end{aligned}$$

for some $\rho_{i,j} \in]t_j, t_{j+1}[$, and $\zeta_{i,j}, \eta_{i,j} \in]x_{i-1}, x_{i+1}[$. If the partial derivatives of order up to four of q are continuous on the compact set

$$R = \{(x,t) : 0 \leq x \leq L \text{ and } 0 \leq t \leq T\} \ , \tag{15.5.1}$$

then there exists H such that

$$\max_{(x,t)\in R}\left|\frac{\partial^2 q}{\partial t^2}(x,t)\right| \leq H \text{ and } \max_{(x,t)\in R}\left|\frac{\partial^4 q}{\partial x^4}(x,t)\right| \leq H \ .$$

Hence,

$$|\tau_{i,j}(\Delta x, \Delta t, q)| \leq \frac{H}{2}\Delta t + \frac{c^2 H}{12}(\Delta x)^2 \tag{15.5.2}$$

for i and j. We conclude that

$$\max\left\{|\tau_{i,j}(\Delta x, \Delta t, q)| : 0 < i < N \text{ and } 0 \leq j < M\right\} \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty \tag{15.5.3}$$

since $\Delta x = L/N$ and $\Delta t = T/M$ converge to 0 as $\min\{N,N\}$ converges to infinity.

Since $B\left(u(x,y), \ldots\right) = u(x,y)$, $B_\Delta(w_{i,j}, \ldots) = w_{i,j}$ and $w_{i,j} = u(x_i,t_j)$ for (i,j) such that $(x_i,t_j) \in \partial R_\Delta$, we get from (15.5.3) that Definition 15.3.6 is satisfied. ∎

Proposition 15.5.2

The finite difference scheme in Algorithm 15.2.1 is stable as defined in Definition 15.3.8 if

$$\frac{c^2 \Delta t}{(\Delta x)^2} \leq \frac{1}{2} \ .$$

Proof.
Consider a function $v : R_\Delta \to \mathbb{R}$. Let $v_{i,j} = v(x_i,t_j)$ for all $(x_i,t_j) \in R_\Delta$ and let $f(x_i,t_j) = P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)$.

We have

$$v_{i,j+1} = v_{i,j} + \alpha\left(v_{i+1,j} - 2v_{i,j} + v_{i-1,j}\right) + f(x_i,t_j)\Delta t$$

$$= (1-2\alpha)v_{i,j} + \alpha v_{i+1,j} + \alpha v_{i-1,j} + f(x_i,t_j)\Delta t$$

for $0 < i < N$ and $0 \le j < M$, where $\alpha = \dfrac{c^2\Delta t}{(\Delta x)^2}$.

Let

$$v_j = \max\left\{ \max_{0\le i\le N} |v_{i,j}|, \max_{\substack{i=0,N \text{ and}\\ 0\le j\le M}} |v_{i,j}| \right\}$$

and $F = \max\limits_{\substack{0<i<N\\0\le j<M}} |f(x_i,y_j)|$.

If $\alpha < 1/2$, we get

$$\begin{aligned}|v_{i,j+1}| &\le (1-2\alpha)|v_{i,j}| + \alpha|v_{i+1,j}| + \alpha|v_{i-1,j}| + |f(x_i,y_i)|\Delta t\\ &\le (1-2\alpha)v_j + \alpha v_j + \alpha v_j + F\Delta t = v_j + F\Delta t\end{aligned}$$

for $0 < i < N$ and $0 \le j < M$. Thus $v_{j+1} \le v_j + F\Delta t$ for $0 \le j < M$. By induction, we get

$$v_j \le v_0 + (j\Delta t)F \le v_0 + TF$$

for $0 \le j \le M$. Hence,

$$|v_{i,j}| \le v_j \le v_0 + TF$$

for $0 \le j \le M$ and $0 \le i \le N$. Since $f(x_i,y_j) = P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)$ for (i,j) such that $(x_i,t_j) \in R^o_\Delta$ and $B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots) = v_{i,j}$ for the (i,j) such that $(x_i,t_j) \in \partial R_\Delta$. We can rewrite the previous inequality as

$$|v_{i,j}| \le C\left\{ \max_{\substack{(i,j) \text{ such that}\\ (x_i,t_j)\in\partial R_\Delta}} |B_\Delta(v_{i,0}, v_{i,1}, v_{i+1,0}, \ldots)| + \max_{\substack{(i,j) \text{ such that}\\ (x_i,t_j)\in R^o_\Delta}} |P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)| \right\}$$

for $0 \le j \le M$ and $0 \le i \le N$, and $C = \max\{1, T\}$. We get (15.3.7). ∎

In Question 15.3 of the exercise section below, the reader is asked to prove that the finite difference scheme in Algorithm 15.2.1 is ℓ^2-stable if $c^2\Delta t/(\Delta x)^2 \le 1/2$.

The next proposition follows from Theorem 15.3.9.

Proposition 15.5.3

The finite difference scheme in Algorithm 15.2.1 is convergent if $\dfrac{c^2\Delta t}{(\Delta x)^2} \le \dfrac{1}{2}$.

This scheme is fairly restrictive because Δt is forced to be very small if Δx is small. If $\Delta x < 10^{-2}$, then $\Delta t < 10^{-4}/(2c^2)$. Thus, a lot of computations are required to advance moderately in time. For this reason, this finite difference scheme is not recommended.

Though we have proved Proposition 15.5.3 about the convergence of the finite difference scheme in Algorithm 15.2.1. it is instructive to prove it again from the definition of convergence.

Proof.
Let $r_{i,j} = w_{i,j} - u(x_i, t_j)$ for $0 \le i \le N$ and $0 \le j \le M$, where u is the solution of (15.2.1) with the associated boundary and initial conditions given in (15.2.2) and (15.2.2) respectively. If we subtract

$$\frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{\Delta t} - c^2 \frac{u(x_{i+1},t_j) - 2u(x_i,t_j) + u(x_{i-1},t_j)}{(\Delta x)^2} = f(x_i,t_j) - \tau_{i,j}(\Delta x, \Delta t, u) \ ,$$

from

$$\frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} = f(x_i,t_j) \ ,$$

we get

$$r_{i,j+1} - r_{i,j} - \alpha\left(r_{i+1,j} - 2r_{i,j} + r_{i-1,j}\right) = \tau_{i,j}(\Delta x, \Delta t, u)\,\Delta t \ .$$

where $\alpha = \dfrac{c^2\Delta t}{(\Delta x)^2}$, Let $R_j = \max\limits_{0<i<N} |r_{i,j}|$. Since we assume that $w_{i,j} = u(x_i,t_j)$ for $i = 0$ and $i = N$, we have in fact that $R_j = \max\limits_{0\le i\le N} |r_{i,j}|$.

If we assume that $1 - 2\alpha > 0$ and use (15.5.2), we get

$$\begin{aligned}
|r_{i,j+1}| &= \left|(1-2\alpha)r_{i,j} + \alpha r_{i+1,j} + \alpha r_{i-1,j} + \tau_{i,j}(\Delta x,\Delta t,u)\,\Delta t\right| \\
&\le (1-2\alpha)|r_{i,j}| + \alpha|r_{i+1,j}| + \alpha|r_{i-1,j}| + |\tau_{i,j}(\Delta x,\Delta t,u)|\,\Delta t \\
&\le R_j + \left(\frac{H}{2}\Delta t + \frac{c^2H}{12}(\Delta x)^2\right)\Delta t
\end{aligned}$$

for $0 < i < N$. Thus,

$$R_{j+1} \le R_j + \left(\frac{H}{2}\Delta t + \frac{c^2H}{12}(\Delta x)^2\right)\Delta t \ .$$

It follows by induction that

$$R_j \le R_0 + \left(\frac{H}{2}\Delta t + \frac{c^2H}{12}(\Delta x)^2\right) j\Delta t = R_0 + \left(\frac{H}{2}\Delta t + \frac{c^2H}{12}(\Delta x)^2\right) t_j$$

for $0 \le j \le M$. Since we assume that $w_{i,0} = u(x_i,0) = g(x_i)$ for $0 \le i \le N$, we have that $R_0 = \mathbf{0}$. Thus,

$$\max_{\substack{0\le i\le N\\ 0\le j<M}} |w_{i,j} - u(x_i,t_j)| \le \max_{0\le j\le M} R_j \le \left(\frac{H}{2}\Delta t + \frac{c^2H}{12}(\Delta x)^2\right) L \to 0$$

as $\min(N,M) \to \infty$ as long as

$$\alpha = \frac{c^2\Delta t}{(\Delta x)^2} \le \frac{1}{2} \ . \tag{15.5.4}$$

We have proved Proposition 15.5.3. ∎

In fact, the condition (15.5.4) is necessary as can be seen from the following example.

Example 15.5.4
This example comes from [20]. We consider the heat equation

$$\frac{\partial u}{\partial t} = c^2\frac{\partial^2 u}{\partial x^2} \quad , \quad -\infty < x < \infty \text{ and } 0 < t < T \ ,$$

with the initial condition

$$u(x,0) = g(x) = \sum_{m=0}^{\infty} \beta_m \cos\left(\frac{2^m \pi x}{L}\right)$$

for $0 \leq x \leq L$. where we assume that $\displaystyle\sum_{m=1}^{\infty} |\beta_m|$ and $\displaystyle\sum_{m=1}^{\infty} \beta_m^2$ converge. This ensure that $g(x)$ is a differentiable function which satisfy $g(0) = g(L)$.

Since the initial condition is a periodic function of period L, we may assume that the solution $u(x,t)$ is periodic of period L with respect to x.

We consider the finite difference scheme

$$\frac{w_{i,j+1} - w_{i,j}}{\Delta t} - c^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} = 0 \tag{15.5.5}$$

with $\Delta x = L/N$ and $\Delta t = T/M$. The domain of the finite difference scheme is

$$R_\Delta = \{(x_i, y_j) : x_i = i\Delta x \text{ for } i \in \mathbb{Z}, \text{ and } y_j = j\Delta y \text{ for } 1 \leq j \leq M\}$$

with

$$\partial R_\Delta = \{(x_i, 0) : x_i = i\Delta x \text{ for } i \in \mathbb{Z}\} \ .$$

As is done to solve the heat equation using separation of variables, we seek solutions of (15.5.5) of the form $w_{i,j} = e^{at_j}\cos(bx_i)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$, and some constants a and b to be determined. If we substitute this expression of $w_{i,j}$ in (15.5.5), we get

$$\begin{aligned}
0 &= \frac{e^{at_{j+1}}\cos(bx_i) - e^{at_j}\cos(bx_i)}{\Delta t} - c^2 \frac{e^{at_j}\cos(bx_{i+1}) - 2e^{at_j}\cos(bx_i) + e^{at_j}\cos(bx_{i-1})}{(\Delta x)^2} \\
&= \frac{e^{at_j}e^{a\Delta t}\cos(bx_i) - e^{at_j}\cos(bx_i)}{\Delta t} \\
&\qquad - c^2 \frac{e^{at_j}\cos(bx_i)\cos(b\Delta x) - 2e^{at_j}\cos(bx_i) + e^{at_j}\cos(bx_i)\cos(b\Delta x)}{(\Delta x)^2} \\
&= \frac{e^{at_j}\cos(bx_i)}{\Delta t}\left(e^{a\Delta t} - 1 + 2\alpha(1 - \cos(b\Delta x))\right) = \frac{e^{at_j}\cos(bx_i)}{\Delta t}\left(e^{a\Delta t} - 1 + 4\alpha\sin^2\left(\frac{b\Delta x}{2}\right)\right)
\end{aligned}$$

for all i and j, where α is defined in (15.5.4). Thus, we must have that

$$e^{a\Delta t} = 1 - 4\alpha\sin^2\left(\frac{b\Delta x}{2}\right) \ .$$

We have found that

$$w_{i,j} = e^{at_j}\cos(bx_i) = \cos(bx_i)(e^{a\Delta t})^{t_j/\Delta t} = \cos(bx_i)\left(1 - 4\alpha\sin^2\left(\frac{b\Delta x}{2}\right)\right)^{t_j/\Delta t} \ .$$

Since the solution $u(x,t)$ is periodic of period L with respect to x, we must have that $w_{i+N,j} = w_{i,j}$ for all i. This is certainly true for $j = 0$ because we have that

$$w_{i+N,0} = g(x_{i+N}) = g(x_i + N\Delta x) = g(x_i + L) = g(x_i)$$

for all i. For this periodic condition to be satisfied, we must have that $b = b_n = 2n\pi/L$ for an integer n that we may assume positive.

Let

$$w_{i,j}^{[n]} = e^{at_j}\cos(b_n x_i) = \cos(b_n x_i)(e^{a\Delta t})^{t_j/\Delta t} = \cos(b_n x_i)\left(1 - 4\alpha \sin^2\left(\frac{b_n \Delta x}{2}\right)\right)^{t_j/\Delta t}$$

for $n > 0$. We seek a solution $w_{i,j}$ of the form

$$w_{i,j} = \sum_{n=0}^{\infty} \gamma_n w_{i,j}^{[n]}$$

for $i \in \mathbb{Z}$ and $0 \leq j \leq M$. The periodicity with respect to i is still satisfied.

The initial condition $w_{i,0} = g(x_i)$ for all i yields

$$\sum_{n=0}^{\infty} \gamma_n \cos(b_n x_i) = \sum_{m=0}^{\infty} \beta_m \cos\left(\frac{2^m \pi x_i}{L}\right) .$$

Thus,

$$\gamma_n = \begin{cases} \beta_m & \text{for } n = 2^{m-1} \\ 0 & \text{otherwise} \end{cases}$$

We have just matched the coefficients of two Fourier cosine series. We get

$$w_{i,j} = \sum_{m=1}^{\infty} \beta_m \cos\left(\frac{2^m \pi x_i}{L}\right)\left(1 - 4\alpha \sin^2\left(\frac{2^m \pi \Delta x}{2L}\right)\right)^{t_j/\Delta t} .$$

We choose a positive integer S such that $c^2T/L^2 \leq S < 2c^2T/L^2$, where we assume that $2c^2T/L^2 > 1$. If $N = 2^k$ and $M = 2^{2k}S$ for $k > 0$ arbitrary, then $\Delta x = L/2^k$ and $\Delta t = T/(S2^{2k})$. We have that

$$\alpha = \frac{c^2 \Delta t}{(\Delta x)^2} = \frac{c^2 T/(S2^{2k})}{L^2/2^{2k}} = \frac{Tc^2}{SL^2}$$

with $\dfrac{1}{2} < \dfrac{Tc^2}{SL^2} \leq 1$. We get

$$w_{i,j} = \sum_{m=1}^{\infty} \beta_m \cos\left(\frac{2^m i \pi}{2^k}\right)\left(1 - 4\alpha \sin^2\left(\frac{2^m \pi}{2^{k+1}}\right)\right)^{2^{2k} S t_j/T} .$$

For $m > k$, we have $\sin^2\left(\dfrac{2^m \pi}{2^{k+1}}\right) = \sin^2\left(2^{m-k-1}\pi\right) = 0$ because $m - k - 1 \geq 0$. Thus

$$1 - 4\alpha \sin^2\left(\frac{2^m \pi}{2^{k+1}}\right) = 1$$

for $m > k$.

For $m = k$, we have $\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right) = \sin^2\left(\frac{\pi}{2}\right) = 1$. Thus

$$-3 \leq 1 - 4\alpha = 1 - 4\alpha\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right) < -1$$

for $m = k$.

For $m = k-1$, we have $\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right) = \sin^2\left(\frac{\pi}{4}\right) = \frac{1}{2}$. Thus

$$-1 \leq 1 - 2\alpha = 1 - 4\alpha\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right) < 0$$

for $m = k-1$.

For $m < k-1$, we have $\sin^2\left(\frac{\pi}{2^{k+1-m}}\right) \leq \frac{1}{4}$ because $0 \leq \frac{2^m\pi}{2^{k+1}} = \frac{\pi}{2^{k+1-m}} \leq \frac{\pi}{8}$ for $k - m + 1 > 2$. Thus

$$0 \leq 1 - \alpha \leq 1 - 4\alpha\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right) < 1$$

for $m < k-1$.

We have that

$$\begin{aligned}|w_{i,j}| &\geq |\beta_k|\,|1-4\alpha|^{2^{2k}St_j/T} - \sum_{m=0}^{k-1}|\beta_m|\left|\cos\left(\frac{2^m i\pi}{2^k}\right)\right|\left|1 - 4\alpha\sin^2\left(\frac{2^m\pi}{2^{k+1}}\right)\right|^{2^{2k}St_j/T} - \sum_{m=k+1}^{\infty}|\beta_m| \\ &\geq |\beta_k|\,|1-4\alpha|^{2^{2k}St_j/T} - \sum_{m=0}^{k-1}|\beta_m| - \sum_{m=k+1}^{\infty}|\beta_m| \geq |\beta_k|\,|1-4\alpha|^{2^{2k}St_j/T} - \sum_{m=0}^{\infty}|\beta_m| \ .\end{aligned}$$

Let us now be a little more specific and assume for instance that $\beta_m = e^{-2^m} > 0$ [6]. We have that $\sum_{m=1}^{\infty}|\beta_m|$ and $\sum_{m=1}^{\infty}\beta_m^2$ converge as required.

Hence

$$|w_{i,j}| \geq e^{-2^k}\,|1-4\alpha|^{2^{2k}St_j/T} - \sum_{m=0}^{\infty}\beta_m = e^{-2^k + 2^{2k}St_j\ln|1-4\alpha|/T} - g(0) \geq e^{-2^k+2^{2k}St_j/T} - g(0) \to \infty$$

as $k \to \infty$ because $|1-4\alpha| > 1$. Thus $w_{i,j}$ cannot approach $u(x_i, y_j)$ as $\min\{N, M\} \to \infty$. ♣

15.5.2 Crank-Nicolson Scheme

We study the ℓ^2 convergence according to Definition 15.3.32 of Algorithm 15.2.2 which is used to approximate the solution of the heat equation with forcing.

As was mentioned in Remark 15.3.34, we will get ℓ^2 consistency according to Definition 15.3.33 if we prove consistency according to Definition 15.3.6. This is what we now do.

[6] Any sequence $\{\beta_m\}_{m=0}^{\infty}$ that preserves the convergence of $\sum_{m=1}^{\infty}|\beta_m|$ and $\sum_{m=1}^{\infty}\beta_m^2$, and such that $|\beta_k|e^{2^{2k}St_j/T} \to \infty$ as $k \to \infty$ can be used.

Proposition 15.5.5

The Crank-Nicolson scheme in Algorithm 15.2.2 is consistent.

Proof.
Using the notation introduced in Section 15.3, we have that

$$P\left(u(x,t),\frac{\partial u}{\partial x}(x,t),\frac{\partial u}{\partial y}(x,t),\frac{\partial^2 u}{\partial x^2}(x,t),\ldots\right)=\frac{\partial u}{\partial t}(x,t)-c^2\frac{\partial^2 u}{\partial x^2}(x,t)$$

and

$$\begin{aligned}P_\Delta\left(w_{i,j},w_{i,j+1},w_{i+1,j},\ldots\right)&=\frac{w_{i,j+1}-w_{i,j}}{\Delta t}\\&\quad-c^2\left(\frac{w_{i+1,j}-2w_{i,j}+w_{i-1,j}}{(\Delta x)^2}+\frac{w_{i+1,j+1}-2w_{i,j+1}+w_{i-1,j+1}}{(\Delta x)^2}\right)\end{aligned}$$

for the Crank-Nicolson scheme. For this scheme, the local truncation is

$$\begin{aligned}\tau_{i,j}(\Delta x,\Delta t,q)&=P_\Delta\left(q(x_i,t_j),q(x_i,t_{j+1}),q(x_{i+1},t_j),\ldots\right)\\&\quad-\frac{1}{2}\left(P\left(q(x_i,t_j),\frac{\partial q}{\partial x}(x_i,t_j),\frac{\partial q}{\partial y}(x_i,t_j),\frac{\partial^2 q}{\partial x^2}(x_i,t_j),\ldots\right)\right.\\&\qquad\left.+P\left(q(x_i,t_{j+1}),\frac{\partial q}{\partial x}(x_i,t_{j+1}),\frac{\partial q}{\partial y}(x_i,t_{j+1}),\frac{\partial^2 q}{\partial x^2}(x_i,t_{j+1}),\ldots\right)\right)\ .\end{aligned}$$

To find the local truncation error of the Crank-Nicolson scheme, we note that

$$\begin{aligned}&\frac{q(x_i,t_{j+1})-q(x_i,t_j)}{\Delta t}-\frac{1}{2}\left(\frac{\partial q}{\partial t}(x_i,t_j)+\frac{\partial q}{\partial t}(x_i,t_{j+1})\right)\\&=\frac{1}{2}\left(\frac{q(x_i,t_{j+1})-q(x_i,t_j)}{\Delta t}-\frac{\partial q}{\partial t}(x_i,t_j)\right)+\frac{1}{2}\left(\frac{q(x_i,t_{j+1})-q(x_i,t_j)}{\Delta t}-\frac{\partial q}{\partial t}(x_i,t_{j+1})\right)\\&=\frac{1}{2}\left(\frac{1}{2}\frac{\partial^2 q}{\partial t^2}(x_i,t_j)\Delta t+\frac{1}{6}\frac{\partial^3 q}{\partial t^3}(x_i,\xi_{i,j})(\Delta t)^2\right)+\frac{1}{2}\left(-\frac{1}{2}\frac{\partial^2 q}{\partial t^2}(x_i,t_{j+1})\Delta t+\frac{1}{6}\frac{\partial^3 q}{\partial t^3}(x_i,\tilde{\xi}_{i,j})(\Delta t)^2\right)\\&=\frac{1}{4}\left(\frac{\partial^2 q}{\partial t^2}(x_i,t_j)\Delta t-\frac{\partial^2 q}{\partial t^2}(x_i,t_{j+1})\Delta t\right)+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\xi_{i,j})(\Delta t)^2+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\tilde{\xi}_{i,j})(\Delta t)^2\\&=-\frac{1}{4}\frac{\partial^3 q}{\partial t^3}(x_i,\breve{\xi}_j)(\Delta t)^2+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\xi_{i,j})(\Delta t)^2+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\tilde{\xi}_{i,j})(\Delta t)^2\\&=\left(-\frac{1}{4}\frac{\partial^3 q}{\partial t^3}(x_i,\breve{\xi}_j)+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\xi_{i,j})+\frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\tilde{\xi}_{i,j})\right)(\Delta t)^2\end{aligned}$$

for some $\xi_{i,j},\tilde{\xi}_{i,j},\breve{\xi}_{i,j}\in]t_i,t_{i+1}[$, and

$$\begin{aligned}&\frac{1}{2}\left(\frac{q(x_{i+1},t_j)-2q(x_i,t_j)+q(x_{i-1},t_j)}{(\Delta x)^2}+\frac{q(x_{i+1},t_{j+1})-2q(x_i,t_{j+1})+q(x_{i-1},t_{j+1})}{(\Delta x)^2}\right)\\&\qquad-\frac{1}{2}\left(\frac{\partial^2 q}{\partial x^2}(x_i,t_j)+\frac{\partial^2 q}{\partial x^2}(x_i,t_{j+1})\right)\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2}\left(\frac{q(x_{i+1},t_j) - 2q(x_i,t_j) + q(x_{i-1},t_j)}{(\Delta x)^2} - \frac{\partial^2 q}{\partial x^2}(x_i,t_j)\right) \\
&\quad + \frac{1}{2}\left(\frac{q(x_{i+1},t_{j+1}) - 2q(x_i,t_{j+1}) + q(x_{i-1},t_{j+1})}{(\Delta x)^2} - \frac{\partial^2 q}{\partial x^2}(x_i,t_{j+1})\right) \\
&= \frac{1}{2}\left(\frac{1}{4!}\left(\frac{\partial^4 q}{\partial x^4}(\zeta_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\eta_{i,j},t_j)\right)(\Delta x)^2\right) \\
&\quad + \frac{1}{2}\left(\frac{1}{4!}\left(\frac{\partial^4 q}{\partial x^4}(\nu_{i,j},t_{j+1}) + \frac{\partial^4 q}{\partial x^4}(\mu_{i,j},t_{j+1})\right)(\Delta x)^2\right) \\
&= \frac{1}{48}\left(\frac{\partial^4 q}{\partial x^4}(\zeta_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\eta_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\nu_{i,j},t_{j+1}) + \frac{\partial^4 q}{\partial x^4}(\mu_{i,j},t_{j+1})\right)(\Delta x)^2
\end{aligned}$$

for $\zeta_{i,j}, \eta_{i,j}, \nu_{i,j}, \mu_{i,j} \in]x_{i-1}, x_{i+1}[$. Thus, the local truncation error of the finite difference equation (15.2.8) is

$$\begin{aligned}
\tau_{i,j}(\Delta x, \Delta t, q) &= P_\Delta\left(q(x_i,t_j), q(x_i,t_{j+1}), q(x_{i+1},t_j), \ldots\right) \\
&\quad - \frac{1}{2}\Bigg(P\left(q(x_i,t_j), \frac{\partial q}{\partial x}(x_i,t_j), \frac{\partial q}{\partial y}(x_i,t_j), \frac{\partial^2 q}{\partial x^2}(x_i,t_j), \ldots\right) \\
&\qquad + P\left(q(x_i,t_{j+1}), \frac{\partial q}{\partial x}(x_i,t_{j+1}), \frac{\partial q}{\partial y}(x_i,t_{j+1}), \frac{\partial^2 q}{\partial x^2}(x_i,t_{j+1}), \ldots\right)\Bigg) \\
&= \left(-\frac{1}{4}\frac{\partial^3 q}{\partial t^3}(x_i,\breve{\xi}_j) + \frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\xi_{i,j}) + \frac{1}{12}\frac{\partial^3 q}{\partial t^3}(x_i,\tilde{\xi}_{i,j})\right)(\Delta t)^2 \\
&\quad - \frac{c^2}{48}\left(\frac{\partial^4 q}{\partial x^4}(\zeta_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\eta_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\nu_{i,j},t_{j+1}) + \frac{\partial^4 q}{\partial x^4}(\mu_{i,j},t_{j+1})\right)(\Delta x)^2
\end{aligned}$$

for some $\xi_{i,j}, \tilde{\xi}_{i,j}, \breve{\xi}_{i,j} \in]t_i, t_{i+1}[$ and $\zeta_{i,j}, \eta_{i,j}, \nu_{i,j}, \mu_{i,j} \in]x_{i-1}, x_{i+1}[$. If the partial derivatives of order up to four of q are continuous on the compact set R defined in (15.5.1), then there exists H such that

$$\max_{(x,t)\in R}\left|\frac{\partial^3 q}{\partial t^3}(x,t)\right| \leq H \quad \text{and} \quad \max_{(x,t)\in R}\left|\frac{\partial^4 q}{\partial x^4}(x,t)\right| \leq H \ .$$

Hence,

$$|\tau_{i,j}(\Delta x, \Delta t, q)| \leq \frac{5H}{12}(\Delta t)^2 + \frac{c^2 H}{12}(\Delta x)^2 \tag{15.5.6}$$

for i and j. We conclude that

$$\max\left\{|\tau_{i,j}(\Delta x, \Delta t, q)| : 0 < i < N \text{ and } 0 \leq j < M\right\} \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty \tag{15.5.7}$$

since $\Delta x = L/N$ and $\Delta t = T/M$ converge to 0 as N and M converge to infinity.

Since $B\left(u(x,y), \ldots\right) = u(x,y)$, $B_\Delta(w_{i,j}, \ldots) = w_{i,j}$ and $w_{i,j} = u(x_i,t_j)$ for (i,j) such that $(x_i,t_j) \in \partial R_\Delta$, we get from (15.5.7) that Definition 15.3.6 is satisfied. ∎

A direct proof that the Crank-Nicolson scheme satisfies the stability condition in Definition 15.3.8 is not simple. The reader is asked to proof a limited version of this stability in Question 15.1 at the end of this chapter.

We have proved in Example 15.3.23 that the Crank-Nicolson scheme is ℓ^2-stable according to Definition 15.3.14. Hence, it follows from Theorem 15.3.20, that the Crank-Nicolson scheme is ℓ^2-convergent according to Definition 15.3.12.

We could stop here but, since we want to demonstrate how to use several definition of stability, we will prove that the Crank-Nicolson scheme is ℓ^2-stable according to Definition 15.3.35.

Proposition 15.5.6

The Crank-Nicolson scheme in Algorithm 15.2.2 is ℓ^2-stable without constraints on Δt and Δx.

Proof.
As we have seen in Section 15.3.5, the finite difference scheme given by Algorithm 15.2.1 can be expressed as $\mathbf{U}_{j+1} = Q_\beta \mathbf{U}_j + \mathbf{B}_j$ for $j \geq 0$, where $Q_\beta = -J^{-1}K$ for K given in (15.2.6) and J given in (15.2.9), and

$$\mathbf{B}_j = J^{-1}\begin{pmatrix} \alpha\left(w_{0,j+1} + w_{0,j}\right) + \big(f(x_1,t_j) + f(x_1,t_{j+1})\big)\Delta t/2 \\ \big(f(x_2,t_j) + f(x_2,t_{j+1})\big)\Delta t/2 \\ \vdots \\ \big(f(x_{N-2},t_j) + f(x_{N-2},t_{j+1})\big)\Delta t/2 \\ \alpha\left(w_{N,j+1} + w_{N,j}\right) + \big(f(x_{N-1},t_j) + f(x_{N-1},t_{j+1})\big)\Delta t/2 \end{pmatrix} .$$

The matrix K can be written as $K = \mathrm{Id} + \alpha A$, where

$$A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix}$$

is a $(N-1)\times(N-1)$ matrix. It follows from Proposition 15.4.1 that the eigenvalues of A are

$$\lambda_k = -2 + 2\cos(k\pi/N) = -4\sin^2(k\pi/(2N))$$

for $0 < k < N$. Thus, the eigenvalues of K are

$$1 + \alpha\lambda_k = 1 - 4\alpha\sin^2(k\pi/(2N))$$

for $0 < k < N$. Proceeding as we did for K, we find that the eigenvalues of J are $1 + 4\alpha\sin^2(k\pi/(2N))$ for $0 < k < N$.

It follows from Proposition 15.4.1 that the eigenvectors of J associated to the eigenvalue $1 + 4\alpha\sin^2(k\pi/(2N))$ are also the eigenvectors of K associated to the eigenvalue $1 - 4\alpha\sin^2(k\pi/(2N))$. Thus the eigenvalues of $J^{-1}K$ are

$$\frac{1 - 4\alpha\sin^2(k\pi/(2N))}{1 + 4\alpha\sin^2(k\pi/(2N))}$$

for $0 < k < N$. These values are all smaller than one in absolute value for every $\alpha > 0$. We have shown that $\|Q_\beta\|_N < 1$ for all N. ∎

Since we have not stated a theorem equivalent to Theorem 15.3.9 for ℓ^2-convergence according to Definition 15.3.32, though one exists [7], we will prove the following proposition.

Proposition 15.5.7

The Crank-Nicolson scheme given in Algorithm 15.2.2 is ℓ^2-convergent according to Definition 15.3.32 without any constrain on Δx and Δt.

Proof.
Let u be the solution of the partial differential equation

$$P\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial t}(x,t), \ldots\right) = \frac{\partial u}{\partial t}(x,t) - c^2\frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t)$$

on $R = [0,L]\times[0,T]$ with $u(x,0) = g(x)$ for $0 \leq x \leq L$, and $u(0,t) = h_0(t)$ and $u(L,t) = h_L(t)$ for $0 \leq t \leq T$.

Suppose that the scheme $\mathbf{w}_{j+1} = Q\mathbf{w}_j + \mathbf{B}_j$ is the matrix representation of the Crank-Nicolson scheme given in Algorithm 15.2.2. We have that $Q = -J^{-1}K$ for K given in (15.2.6) and J given in (15.2.9), and

$$\mathbf{B}_j = J^{-1}\begin{pmatrix} \alpha(w_{0,j+1} + w_{0,j}) + \big(f(x_1,t_j) + f(x_1,t_{j+1})\big)\Delta t/2 \\ \big(f(x_2,t_j) + f(x_2,t_{j+1})\big)\Delta t/2 \\ \vdots \\ \big(f(x_{N-2},t_j) + f(x_{N-2},t_{j+1})\big)\Delta t/2 \\ \alpha(w_{N,j+1} + w_{N,j}) + \big(f(x_{N-1},t_j) + f(x_{N-1},t_{j+1})\big)\Delta t/2 \end{pmatrix}$$

for $j \geq 0$.

In vector form, we get from

$$\begin{aligned}\tau_{i,j}(\Delta x,\Delta t,u) = P_\Delta\left(u(x_i,t_j), q(u_i,t_{j+1}), u(x_{i+1},t_j),\ldots\right) \\ - \frac{1}{2}\Bigg(P\left(u(x_i,t_j), \frac{\partial u}{\partial x}(x_i,t_j), \frac{\partial u}{\partial y}(x_i,t_j), \frac{\partial^2 u}{\partial x^2}(x_i,t_j),\ldots\right) \\ - P\left(u(x_i,t_{j+1}), \frac{\partial u}{\partial x}(x_i,t_{j+1}), \frac{\partial u}{\partial y}(x_i,t_{j+1}), \frac{\partial^2 u}{\partial x^2}(x_i,t_{j+1}),\ldots\right)\Bigg)\end{aligned}$$

that $\mathbf{u}_{j+1} = Q\mathbf{u}_j + \mathbf{b}_j + \Delta t\, J^{-1}\tau_j(\Delta x,\Delta t,u)$, where

$$\mathbf{u}_j = \begin{pmatrix} u(x_1,t_j) \\ u(x_2,t_j) \\ \vdots \\ u(x_{N-1},t_j) \end{pmatrix} \quad , \quad \tau_j(\Delta x,\Delta t,u) = \begin{pmatrix} \tau_{1,j}(\Delta x,\Delta t,u) \\ \tau_{2,j}(\Delta x,\Delta t,u) \\ \vdots \\ \tau_{N-1,j}(\Delta x,\Delta t,u) \end{pmatrix}$$

[7] The proof of Proposition 15.5.7 can be slightly modified to prove this result for a large class of finite difference schemes. The requirement is that the linear operator in front of τ_j be uniformly bounded for all N as it is the case for J^{-1} in the proof of Proposition 15.5.7

and

$$\mathbf{b}_j = J^{-1}\begin{pmatrix} \alpha\big(u(x_0,t_{j+1})+u(x_0,t_j)\big)+\big(f(x_1,t_j)+f(x_1,t_{j+1})\big)\Delta t/2 \\ \big(f(x_2,t_j)+f(x_2,t_{j+1})\big)\Delta t/2 \\ \vdots \\ \big(f(x_{N-2},t_j)+f(x_{N-2},t_{j+1})\big)\Delta t/2 \\ \alpha\big(u(x_N,t_{j+1})+u(x_N,t_j)\big)+\big(f(x_{N-1},t_j)+f(x_{N-1},t_{j+1})\big)\Delta t/2 \end{pmatrix}$$

for $j \geq 0$. Since we assume that $w_{0,j} = u(0,t_j)$ and $w_{N,j} = u(L,t_j)$ for all j, we have that $\mathbf{B}_j = \mathbf{b}_j$ for all j.

We have that $\mathbf{w}_{j+1} = Q\mathbf{w}_j + \mathbf{B}_j$ satisfies Definition 15.3.14 with $C = 1$ because $\|Q\|_N < 1$ as we have shown in the proof of the previous proposition. Moreover, we have also shown that

$$\|J^{-1}\|_N = \rho(J^{-1}) = \max_{0<k<N}\{\,1/(1+4\alpha\sin^2(k\,\pi/(2N)))\,\} < 1 \ .$$

Since

$$\mathbf{w}_{j+1} - \mathbf{u}_{j+1} = (Q\mathbf{w}_j + \mathbf{B}_j) - \big(Q\mathbf{u}_j + \mathbf{b}_j + \Delta t\, J^{-1}\tau_j\big) = Q\,(\mathbf{w}_j - \mathbf{u}_j) - \Delta t\, J^{-1}\tau_j$$

for $0 \leq j < M$, we get by induction that

$$\mathbf{w}_j - \mathbf{u}_j = Q^j\,(\mathbf{w}_0 - \mathbf{u}_0) - \Delta t\sum_{s=0}^{j-1} Q^s J^{-1}\tau_{j-1-s}$$

for $0 < j \leq M$. Hence

$$\begin{aligned}\|\mathbf{w}_j - \mathbf{u}_j\|_N &\leq \|Q^j\|_N\,\|\mathbf{w}_0 - \mathbf{u}_0\|_N + \Delta t\,\|J^{-1}\|_N\sum_{k=0}^{j-1}\|Q^s\|_N\,\|\tau_{j-1-s}\|_N \\ &\leq \|\mathbf{w}_0 - \mathbf{u}_0\|_N + j\Delta t\, L^{1/2}\,\tau(\Delta x,\Delta t,u) \\ &\leq \|\mathbf{w}_0 - \mathbf{u}_0\|_N + T\, L^{1/2}\,\tau(\Delta x,\Delta t,u) \qquad (15.5.8)\end{aligned}$$

for $0 < j \leq M$, where

$$\tau(\Delta x,\Delta t,u) \equiv \max_{\substack{(i,j)\text{ such that}\\(x_i,y_j)\in R^o_\Delta}} |\tau_{i,j}(\Delta x,\Delta t,u)| \leq \frac{5H}{12}(\Delta t)^2 + \frac{c^2H}{12}(\Delta x)^2$$

because of (15.5.6). It is here that the choice of the norm $\|\cdot\|_N$ is important because we have

$$\begin{aligned}\|\tau_j\|_N &= \left(\sum_{i=1}^{N-1}\left(\tau_{i,j}(\Delta x,\Delta t,u)\right)^2\,\Delta x\right)^{1/2} \leq \left(\sum_{i=1}^{N-1}\tau^2(\Delta x,\Delta t,u)\,\Delta x\right)^{1/2} \\ &= \tau(\Delta x,\Delta t,u)\,((N-1)(L/N))^{1/2} \leq L^{1/2}\,\tau(\Delta x,\Delta t,u)\ .\end{aligned}$$

Since the Crank-Nicolson scheme is consistent, we get from (15.5.8) that

$$\max_{0<j\leq M}\|\mathbf{w}_j - \mathbf{u}_j\|_N \leq T\,L^{1/2}\,\tau(\Delta x,\Delta t,u) \to 0$$

as $\min\{N,M\} \to \infty$ since we assume that $w_{i,j} = u(x_i,t_j)$ for all (i,j) such that $(x_i,t_j) \in \partial R_\Delta$. This implies that $\mathbf{w}_0 = \mathbf{u}_0$. ∎

Crank-Nicolson is a really good scheme because there is no constrains on Δx and Δt, and the convergence is quadratic; namely, order two in Δx and Δt.

15.6 Dirichlet Equation

We could study the consistence, stability and convergence of the finite difference scheme in Algorithm 15.2.6 as we did for the previous finite difference schemes for the heat equation with forcing. However, we will not do so. There is a more elegant approach to study the consistence, stability and convergence of the finite difference schemes to numerically solve elliptic equations.

We first consider the Dirichlet problem (15.2.10) with $f(x,y) = 0$ for all $(x,y) \in R$. In that particular case, it is called the Laplace equation. Our first result will be a maximum principle [8] for the following finite difference scheme used to numerically solve the Laplace equation.

$$Q_\Delta(w_{i,j}) \equiv \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta x)^2} + \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta y)^2} = 0$$

for $0 < i < N$ and $0 < j < M$, where $w_{i,0} = g(x_i, 0)$ and $w_{i,M} = g(x_i, b)$ for $0 \leq i \leq N$, and $w_{0,j} = g(0, y_j)$ and $w_{N,j} = g(a, y_j)$ for $0 \leq j \leq M$.

As we did in Section 15.3, we have

$$R = \{(x,y) : 0 \leq x \leq a \text{ and } 0 \leq y \leq b\}$$

and

$$R_\Delta = \{(x_i, y_j) : x_i = i\Delta x \text{ for } 0 \leq i \leq N, \text{ and } y_j = j\Delta y \text{ for } 1 \leq j \leq M\} ,$$

where $\Delta x = a/N$ and $\Delta y = b/M$. We also define

$$\begin{aligned} \partial R_\Delta = \{(x_i, y) : x_i = i\Delta x \text{ for } 0 \leq i \leq N, \text{ and } y = y_0 = 0 \text{ or } y = y_M = b\} \\ \cup \{(x, y_j) : y_j = j\Delta y \text{ for } 0 \leq i \leq M, \text{ and } x = x_0 = 0 \text{ or } x = x_N = a\} \end{aligned}$$

and $R_\Delta^o = R_\Delta \setminus \partial R_\Delta$.

Theorem 15.6.1

Suppose that $v_{i,j} = v(x_i, y_j)$ for all i and j, where $v : R_\Delta \to \mathbb{R}$. If $Q_\Delta(v_{i,j}) \geq 0$ for all (i,j) with $(x_i, y_j) \in R_\Delta^o$, then

$$\max_{(x_i,y_j)\in R_\Delta^o} v_{i,j} \leq \max_{(x_i,y_j)\in \partial R_\Delta} v_{i,j} \ .$$

Proof.
The proof is by contradiction. Suppose that there exist $(x_{i_0}, y_{j_0}) \in R_\Delta^o$, (i.e. $0 < i_0 < M$ and $0 < j_0 < N$) such that $v_{i_0,j_0} \geq v_{i,j}$ for all (i,j) with $(x_i, y_j) \in R_\Delta^o$, and $v_{i_0,j_0} > v_{i,j}$ for all (i,j) with $(x_i, y_j) \in \partial R_\Delta$. We get from

$$Q_\Delta\left(v_{i_0,j_0}\right) \equiv \frac{v_{i_0,j_0+1} - 2v_{i_0,j_0} + v_{i_0,j_0-1}}{(\Delta x)^2} + \frac{v_{i_0+1,j_0} - 2v_{i_0,j_0} + v_{i_0-1,j_0}}{(\Delta y)^2} \geq 0$$

[8] It is a well know result that the solution of the Laplace equation reach it maximum on the boundary of the domain R. The solution of $Q_\Delta(w) = 0$ has the same property.

that

$$2\left(\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}\right)v_{i_0,j_0} \leq \frac{1}{(\Delta x)^2}\left(v_{i_0,j_0+1}+v_{i_0,j_0-1}\right)\frac{1}{(\Delta x)^2}\left(v_{i_0+1,j_0}+v_{i_0-1,j_0}\right) \ .$$

Since $v_{i,j} \leq v_{i_0,j_0}$ for all (i,j), the only way to satisfy this inequality is if $v_{i_0,j_0+1} = v_{i_0,j_0-1} = v_{i_0+1,j_0} = v_{i_0-1,j_0} = v_{i_0,j_0}$.

We can then repeat the same procedure with v_{i_0,j_0+1}, v_{i_0,j_0-1}, v_{i_0+1,j_0} and v_{i_0-1,j_0}. Moving that way horizontally and vertically, we find that $v_{i,j} = v_{i_0,j_0}$ for all (i,j) [9], even those for $(x_i,y_j) \in \partial R_\Delta$. This contradicts our assumption that $v_{i_0,j_0} > v_{i,j}$ for all (i,j) with $(x_i,y_j) \in \partial R_\Delta$. ∎

Using an argument like the one in the proof of the previous theorem or simply applying the previous theorem to $-v_{i,j}$ instead of $v_{i,j}$, we get the following result.

Theorem 15.6.2

Suppose that $v_{i,j} = v(x_i,y_j)$ for all i and j, where $v : R_\Delta \to \mathbb{R}$. If $Q_\Delta(v_{i,j}) \leq 0$ for all (i,j) with $(x_i,y_j) \in R^o_\Delta$, then

$$\min_{(x_i,y_j)\in R^o_\Delta} v_{i,j} \geq \min_{(x_i,y_j)\in \partial R_\Delta} v_{i,j} \ .$$

It follows from the previous two theorems that the finite difference scheme in Algorithm 15.2.6 has a unique solution. Suppose that $\{w^{[k]}_{i,j} : 0 \leq i \leq N \text{ and } 0 \leq j \leq M\}$ for $k = 1$ and 2 are two solutions of the finite difference scheme in Algorithm 15.2.6. We then have that $\{w_{i,j} : 0 \leq i \leq N \text{ and } 0 \leq j \leq M\}$ with $w_{i,j} = w^{[1]}_{i,j} - w^{[2]}_{i,j}$ for all i and j is a solution of $\Delta_\Delta w_{i,j} = 0$ for all (i,j) such that $(x_i,y_j) \in R^o_\Delta$, and $w_{i,j} = 0$ for all (i,j) such that $(x_i,y_j) \in \partial R_\Delta$. It follows from Theorems 15.6.1 and 15.6.2 that

$$0 = \min_{(x_i,y_j)\in \partial R_\Delta} w_{i,j} \leq \min_{(x_i,y_j)\in R^o_\Delta} w_{i,j} \leq \max_{(x_i,y_j)\in R^o_\Delta} w_{i,j} \leq \max_{(x_i,y_j)\in \partial R_\Delta} w_{i,j} = 0 \ .$$

Thus $w^{[1]}_{i,j} - w^{[2]}_{i,j} = 0$ for $0 \leq i \leq N$ and $0 \leq j \leq M$.

15.6.1 Algorithm 15.2.6

Proposition 15.6.3

The finite difference scheme in Algorithm 15.2.6 is consistent.

Proof.
Using the notation introduced in Section 15.3, we have that

$$P\left(u(x,y), \frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y), \frac{\partial^2 u}{\partial x^2}(x,y), \ldots\right) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

[9]To be exact, we are missing the four corner points of R_Δ

and
$$P_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta y)^2}$$
for the finite difference scheme in Algorithm 15.2.6.

To find the local truncation error of the finite difference scheme in Algorithm 15.2.6, we need to use the formula in (15.1.6) twice, for the second other partial derivative of u with respect to x and the second other partial derivative of u with respect to y.

Given any sufficiently differentiable function $q : R \to \mathbb{R}$, let $q_{i,j} = q(x_i, y_j)$ for all $(x_i, y_j) \in R_\Delta$. We have
$$\frac{\partial^2 q}{\partial x^2}(x_i, y_j) = \frac{q_{i+1,j} - 2q_{i,j} + q_{i-1,j}}{(\Delta x)^2} - \frac{1}{4!}\left(\frac{\partial^4 q}{\partial x^4}\left(\zeta_{i,j}, y_j\right) + \frac{\partial^4 q}{\partial x^4}\left(\eta_{i,j}, y_j\right)\right)(\Delta x)^2$$
and
$$\frac{\partial^2 q}{\partial y^2}(x_i, y_j) = \frac{q_{i,j+1} - 2q_{i,j} + q_{i,j-1}}{(\Delta y)^2} - \frac{1}{4!}\left(\frac{\partial^4 q}{\partial y^4}\left(x_i, \mu_{i,j}\right) + \frac{\partial^4 q}{\partial y^4}\left(x_i, \nu_{i,j}\right)\right)(\Delta y)^2$$
for $\zeta_{i,j}, \eta_{i,j} \in]x_{i-1}, x_{i+1}[$ and $\mu_{i,j}, \nu_{i,j} \in]y_{j-1}, y_{j+1}[$. Hence
$$\begin{aligned}\tau_{i,j}(\Delta x, \Delta y, q) &= P\left(q(x_i, t_j), \frac{\partial q}{\partial x}(x_i, t_j), \frac{\partial q}{\partial y}(x_i, t_j), \frac{\partial^2 q}{\partial x^2}(x_i, t_j), \ldots\right) \\ &\qquad - P_\Delta\left(q(x_i, t_j), q(x_i, y_{j+1}), q(x_{i+1}, t_j), \ldots\right) \\ &= -\frac{1}{4!}\left(\frac{\partial^4 q}{\partial x^4}\left(\zeta_{i,j}, y_j\right) + \frac{\partial^4 q}{\partial x^4}\left(\eta_{i,j}, y_j\right)\right)(\Delta x)^2 - \frac{1}{4!}\left(\frac{\partial^4 q}{\partial y^4}\left(x_i, \mu_{i,j}\right) + \frac{\partial^4 q}{\partial y^4}\left(x_i, \nu_{i,j}\right)\right)(\Delta y)^2\end{aligned}$$
for $\zeta_{i,j}, \eta_{i,j} \in]x_{i-1}, x_{i+1}[$ and $\mu_{i,j}, \nu_{i,j} \in]y_{j-1}, y_{j+1}[$.

If the partial derivatives of order up to four of q are continuous on the compact set R, then there exists H such that
$$\max_{(x,y)\in R}\left|\frac{\partial^4 q}{\partial x^4}(x,t)\right| \le H \quad \text{and} \quad \max_{(x,y)\in R}\left|\frac{\partial^4 q}{\partial y^4}(x,t)\right| \le H\ .$$
Hence,
$$|\tau_{i,j}(\Delta x, \Delta y, q)| \le \frac{H}{12}\left((\Delta x)^2 + (\Delta y)^2\right) \tag{15.6.1}$$
for i and j. We conclude that
$$\max\left\{|\tau_{i,j}(\Delta x, \Delta y, q)| : 0 < i < N \text{ and } 0 < j < M\right\} \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty \tag{15.6.2}$$
since $\Delta x = L/N$ and $\Delta t = T/M$ converge to 0 as N and M converge to infinity.

Since $B\left(u(x,y), \ldots\right) = u(x,y)$, $B_\Delta(w_{i,j}, \ldots) = w_{i,j}$ and $w_{i,j} = u(x_i, t_j)$ for (i,j) such that $(x_i, t_j) \in \partial R_\Delta$, we get from (15.6.2) that Definition 15.3.6 is satisfied. ∎

The stability of the finite difference scheme in Algorithm 15.2.6 will be a consequence of the next theorem.

Theorem 15.6.4

Suppose that $v_{i,j} = v(x_i,y_j)$ for all i and j, where $v : R_\Delta \to \mathbb{R}$. Then

$$\max_{(x_i,y_j)\in R_\Delta} |v_{i,j}| \le \max_{(x_i,y_j)\in \partial R_\Delta} |v_{i,j}| + \frac{a^2}{2} \max_{(x_i,y_j)\in R_\Delta^o} |Q_\Delta(v_{i,j})| \ .$$

Proof.
We need to define some real-valued functions on R_Δ to prove this result. First

$$\begin{aligned} h : R_\Delta &\to \mathbb{R} \\ (x_i,y_j) &\mapsto x_i^2/2 \end{aligned}$$

We obviously have that $0 \le h(x_i,y_j) \le a^2/2$ for all $(x_i,y_j) \in R_\Delta$. Moreover, if we let $h_{i,j} = h(x_i,y_j)$ for all i and j, we have

$$\begin{aligned} Q_\Delta(h_{i,j}) &= \frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{(\Delta x)^2} + \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{(\Delta y)^2} \\ &= \frac{(x_i+\Delta x)^2 - 2x_i^2 + (x_i - \Delta x)^2}{2(\Delta x)^2} + \frac{x_i^2 - 2x_i^2 + x_i^2}{2(\Delta y)^2} = \frac{(\Delta x)^2 + (\Delta x)^2}{2(\Delta x)^2} = 1 \end{aligned}$$

for all (i,j) such that $(x_i,y_j) \in R_\Delta^o$.

let $K = \max_{(x_i,y_j)\in R_\Delta^o} |Q_\Delta(v_{i,j})|$. We define two additional functions.

$$\begin{aligned} g^\pm : R_\Delta &\to \mathbb{R} \\ (x_i,y_j) &\mapsto \pm v(x_i,y_j) + K h(x_i,y_j) = \pm v_{i,j} + K h_{i,j} \end{aligned}$$

Let $g^\pm_{i,j} = g^\pm(x_i,y_j) = \pm v_{i,j} + K h_{i,j}$ for all i and j. By linearity of the operator Q_Δ, we have that

$$Q_\Delta\left(g^\pm_{i,j}\right) = \pm Q_\Delta(v_{i,j}) + K Q_\Delta(h_{i,j}) = \pm Q_\Delta(v_{i,j}) + K \ge 0$$

for all (i,j) such that $(x_i,y_j) \in R_\Delta^o$. Therefore, we may apply Theorem 15.6.1 to $g^\pm$. For g^+, we get

$$\begin{aligned} v_{i,j} &\le \max_{(x_i,y_j)\in R_\Delta^o} v_{i,j} \le \max_{(x_i,y_j)\in R_\Delta^o} g^+_{i,j} \le \max_{(x_i,y_j)\in \partial R_\Delta} g^+_{i,j} \le \max_{(x_i,y_j)\in \partial R_\Delta} v_{i,j} + K \max_{(x_i,y_j)\in \partial R_\Delta} h_{i,j} \\ &\le \max_{(x_i,y_j)\in \partial R_\Delta} |v_{i,j}| + \frac{Ka^2}{2} \end{aligned}$$

for all (i,j) such that $(x_i,y_j) \in R_\Delta^o$. For g^-, we also get

$$\begin{aligned} -v_{i,j} &\le \max_{(x_i,y_j)\in R_\Delta^o} -v_{i,j} \le \max_{(x_i,y_j)\in R_\Delta^o} g^-_{i,j} \le \max_{(x_i,y_j)\in \partial R_\Delta} g^-_{i,j} \le \max_{(x_i,y_j)\in \partial R_\Delta} -v_{i,j} + K \max_{(x_i,y_j)\in \partial R_\Delta} h_{i,j} \\ &\le \max_{(x_i,y_j)\in \partial R_\Delta} |v_{i,j}| + \frac{Ka^2}{2} \end{aligned}$$

for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$. Thus

$$|v_{i,j}| \leq \max_{(x_i,y_j)\in\partial R_\Delta} |v_{i,j}| + \frac{Ka^2}{2}$$

for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$. This gives the conclusion of the theorem. ∎

Proposition 15.6.5

The finite difference scheme in Algorithm 15.2.6 is stable without constraints on Δx and Δy.

Proof.
For the finite difference scheme in Algorithm 15.2.6, we have that $P_\Delta(w_{i,j}, \ldots) = \Delta_\Delta w_{i,j}$ for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$, and $B_\Delta(w_{i,j}, \ldots) = w_{i,j}$ for all (i,j) such that $(x_i, y_j) \in \partial R_\Delta$. It then follows from the previous theorem that the condition (15.3.7) for the definition of stability in Definition 15.3.8 is satisfied with $C = \max\{1, a^2/2\}$. ∎

The following result is a consequence of Proposition 15.6.3, Proposition 15.6.5 and Theorem 15.3.9.

Proposition 15.6.6

The finite difference scheme in Algorithm 15.2.6 is convergent without any constrains on Δx and Δt.

We can also prove this proposition using Theorem 15.6.4.

Proof.
Suppose that $\{w_{i,j} : 0 \leq i \leq N \text{ and } 0 \leq j \leq M\}$ is the solution of $Q_\Delta(w_{i,j}) = f(x_i, y_j)$ for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$, and $w_{i,j} = g(x_i, y_j)$ for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$.

Suppose that u is the solution of the Dirichlet equation introduced in Section 15.2.2; namely, u is the solution of $\Delta u(x,y) = f(x,y)$ for $(x,y) \in R \setminus \partial R$ and $u(x,y) = g(x,y)$ for $(x,y) \in \partial R$.

We have from

$$\tau_{i,j}(\Delta x, \Delta y, u) = P\left(u(x_i,y_j), \frac{\partial u}{\partial x}(x_i,y_j), \ldots\right) - P_\Delta\left(u(x_i,y_j), u(x_{i+1},y_j), \ldots\right)$$

that

$$\begin{aligned} &\frac{u(x_{i+1},y_j) - 2u(x_i,y_j) + u(x_{i-1},y_j)}{(\Delta x)^2} + \frac{u(x_i,y_{j+1}) - 2u(x_i,y_j) + u(x_i,y_{j-1})}{(\Delta y)^2} \\ &\qquad = f(x_i,y_j) - \tau_{i,j}(\Delta x, \Delta y, u) \end{aligned} \tag{15.6.3}$$

for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$, where $\tau_{i,j}(\Delta x, \Delta y, u)$ is defined in the proof of Proposition 15.6.3.

Let $r_{i,j} = w_{i,j} - u(x_i, t_j)$ for $0 \leq i \leq N$ and $0 \leq j \leq M$. If we subtract (15.6.3) from $Q_\Delta(w_{i,j}) = f(x_i, y_j)$, we get

$$Q_\Delta(r_{i,j}) = \frac{r_{i+1,j} - 2r_{i,j} + r_{i-1,j}}{(\Delta x)^2} + \frac{r_{i,j+1} - 2r_{i,j} + r_{i,j-1}}{(\Delta y)^2} = \tau_{i,j}(\Delta x, \Delta y, u) \ .$$

for all (i,j) such that $(x_i, y_j) \in R^o_\Delta$.

Because $w_{i,j} = u(x_i, y_j) = g(x_i, y_j)$ for (i,j) such that $(x_i, y_j) \in \partial R_\Delta$, we have that $r_{i,j} = 0$ for (i,j) such that $(x_i, y_j) \in \partial R_\Delta$. Hence, if we apply Theorem 15.6.4 to the function

$$\begin{aligned} r : R_\Delta &\to \mathbb{R} \\ (x_i, y_j) &\mapsto r_{i,j} \end{aligned}$$

we get

$$\max_{(x_i,y_j)\in R_\Delta} |r_{i,j}| \leq \frac{a^2}{2} \max_{(x_i,y_j)\in R^o_\Delta} |Q_\Delta(r_{i,j})| = \frac{a^2}{2} \max_{(x_i,y_j)\in R^o_\Delta} |\tau_{i,j}(\Delta x, \Delta y, u)| \ .$$

It follows from (15.6.1) that

$$\max_{(x_i,y_j)\in R_\Delta} |w_{i,j} - u(x_i, y_j)| \leq \frac{a^2 H}{24} \left((\Delta x)^2 + (\Delta y)^2 \right) \to 0 \quad \text{as} \quad \min\{N, M\} \to \infty \ . \qquad \blacksquare$$

15.7 Wave Equation

The finite difference scheme in Algorithm 15.2.11 was developed to numerically solve the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad , \quad 0 < x < L \text{ and } 0 < t < T \tag{15.7.1}$$

with the boundary conditions

$$u(0,t) = u(L,t) = 0 \quad , \quad 0 < t < T \ , \tag{15.7.2}$$

and the initial conditions

$$u(x,0) = g(x) \text{ and } \frac{\partial u}{\partial t}(x,0) = f(x) \quad , \quad 0 \leq x \leq L \ . \tag{15.7.3}$$

As we will show below for the special case of the wave equation above, finite difference scheme are not ideal to numerically solve hyperbolic differential equations. Strict conditions on the step sizes are required to get convergent finite difference schemes. We will address this issue in the next section before studying the stability, consistency and convergence of the finite difference scheme in Algorithm 15.2.11.

15.7.1 The Role of the Domain of Dependence

This section uses some of the basic techniques to solve partial differential equations.

Let us start with the wave equation on the real line.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad , \quad -\infty < x < \infty \text{ and } t > 0 \tag{15.7.4}$$

with the initial conditions

$$u(x,0) = g(x) \text{ and } \frac{\partial u}{\partial t}(x,0) = f(x) \quad , \quad -\infty < x < \infty \ . \tag{15.7.5}$$

If we use the substitution $\xi = x + ct$ and $\eta = x - ct$, we get

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial t} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial t} = c\frac{\partial u}{\partial \xi} - c\frac{\partial u}{\partial \eta}$$

and

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2} &= \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right) = \frac{\partial}{\partial \xi}\left(c\frac{\partial u}{\partial \xi} - c\frac{\partial u}{\partial \eta}\right)\frac{\partial \xi}{\partial t} + \frac{\partial}{\partial \eta}\left(c\frac{\partial u}{\partial \xi} - c\frac{\partial u}{\partial \eta}\right)\frac{\partial \eta}{\partial t} \\ &= c\left(c\frac{\partial^2 u}{\partial \xi^2} - c\frac{\partial^2 u}{\partial \xi \partial \eta}\right) - c\left(c\frac{\partial^2 u}{\partial \eta \partial \xi} - c\frac{\partial^2 u}{\partial \eta^2}\right) = c^2\left(\frac{\partial^2 u}{\partial \xi^2} - 2\frac{\partial^2 u}{\partial \eta \partial \xi} + \frac{\partial^2 u}{\partial \eta^2}\right) \ .\end{aligned} \tag{15.7.6}$$

Similarly,

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{\partial u}{\partial \xi} + \frac{\partial u}{\partial \eta}$$

and

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial^2 u}{\partial \xi^2} + 2\frac{\partial^2 u}{\partial \eta \partial \xi} + \frac{\partial^2 u}{\partial \eta^2} \ . \tag{15.7.7}$$

If we substitute (15.7.6) and (15.7.6) in (15.7.4), and simplify the result, we get

$$\frac{\partial^2 u}{\partial \eta \partial \xi} = 0 \ . \tag{15.7.8}$$

Integrating this equation with respect to ξ yields $\dfrac{\partial u}{\partial \eta} = H(\eta)$ for some function $H : \mathbb{R} \to \mathbb{R}$. Integrating $\dfrac{\partial u}{\partial \eta} = H(\eta)$ with respect to η yields $u(\eta, \xi) = \int H(\eta)\,d\eta + G(\xi)$ for some function $G : \mathbb{R} \to \mathbb{R}$. If we define $F(\eta) = \int H(\eta)\,d\eta$, we get the solution

$$u(\eta, \xi) = F(\eta) + G(\xi)$$

for (15.7.8). In terms of x and t, we get the solution

$$u(x,t) = F(x - ct) + G(x + ct)$$

for (15.7.4). We now consider the initial conditions in (15.7.5). We have that f and g satisfy the equations $f(x) = F(x) + G(x)$ and $g(x) = -cF'(x) + cG'(x)$. If we assume that g is locally integrable, we may write

$$\int_0^x g(s)\,\mathrm{d}s = -cF(x) + cG(x) + cF(0) - cG(0) \ .$$

We end up with two linearly independent equations for F and G.

$$F(x) + G(x) = f(x) \tag{15.7.9}$$

$$F(x) - G(x) = F(0) - G(0) - \frac{1}{c}\int_0^x g(s)\,\mathrm{d}s \ . \tag{15.7.10}$$

Adding (15.7.9) and (15.7.10) and dividing by 2 yield

$$F(x) = \frac{1}{2}f(x) - \frac{1}{2c}\int_0^x g(s)\,\mathrm{d}s + \frac{1}{2}\left(F(0) - G(0)\right) \ .$$

Subtracting (15.7.10) from (15.7.9) and dividing by 2 yield

$$G(x) = \frac{1}{2}f(x) + \frac{1}{2c}\int_0^x g(s)\,\mathrm{d}s - \frac{1}{2}\left(F(0) - G(0)\right) \ .$$

We finally get the solution

$$u(x,t) = F(x-ct) + G(x+ct) = \frac{1}{2}\left(f(x-ct) + f(x+ct)\right) + \frac{1}{2c}\int_{x-ct}^{x+ct} g(s)\,\mathrm{d}s \tag{15.7.11}$$

for (15.7.4) and (15.7.5). The **domain of dependence** for $u(\tilde{x},\tilde{t})$ is the set $\{(x,t) : 0 \leq t \leq \tilde{t}$ and $ct + \tilde{x} - c\tilde{t} \leq x \leq -ct + \tilde{x} + c\tilde{t}\}$. This is the set of all points (x,t) where $f(x-ct)$, $f(x+ct)$ and $g(s)$ in (15.7.11) are evaluated to get the value of u at $(\tilde{x},\tilde{t})$. The domain of dependence for $u(\tilde{x},\tilde{t})$ is displayed in figure (15.5). The domain of dependence will play an important role in our analysis of finite difference schemes used to numerically solve the wave equation and hyperbolic equations in general.

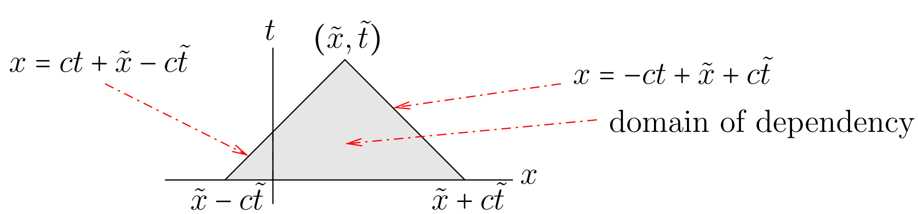


Figure 15.5: Domain of dependence for $u(\tilde{x},\tilde{t})$, the value of the solution u of the wave equation at $(\tilde{x},\tilde{t})$.

We now solve the wave equation (15.7.1) with the boundary conditions (15.7.2) and the initial conditions (15.7.3). To do so, we will use the method of separation of variables. The presentation will be a little more formal than usual. However, all the theoretical details could be filled in by the reader who has some knowledge of L^2-spaces and Fourier series.

If we substitute $u(x,t) = F(x)G(t)$ in (15.7.1), we get

$$F(x)\,\frac{\mathrm{d}^2G}{\mathrm{d}t^2}(t) = c^2\frac{\mathrm{d}^2F}{\mathrm{d}x^2}(x)\,G(t)\ .$$

Thus, after dividing both sides by $c^2\,F(x)G(t)$, we get

$$\frac{1}{c^2\,G(t)}\,\frac{\mathrm{d}^2G}{\mathrm{d}t^2}(t) = \frac{1}{F(x)}\,\frac{\mathrm{d}^2F}{\mathrm{d}x^2}(x) \quad , \quad t>0 \text{ and } 0<x<L\ .$$

Since the right hand side is independent of t and the left hand side is independent of x, we get

$$\frac{1}{c^2\,G(t)}\,\frac{\mathrm{d}^2G}{\mathrm{d}t^2}(t) = \frac{1}{F(x)}\,\frac{\mathrm{d}^2F}{\mathrm{d}x^2}(x) = k \quad , \quad t>0 \text{ and } 0<x<L$$

for some constant k. We end up with two ordinary differential equations.

$$\frac{\mathrm{d}^2F}{\mathrm{d}x^2}(x) - kF(x) = 0 \quad \text{and} \quad \frac{\mathrm{d}^2G}{\mathrm{d}t^2}(t) - c^2\,k\,G(t)) = 0\ . \tag{15.7.12}$$

From $u(0,t) = 0$, we get $F(0)G(t) = 0$. Since we assume that G is not trivial, we get $F(0) = 0$. Similarly, from $u(L,t) = 0$, we get $F(L)G(t) = 0$. Again, since we assume that G is not trivial, we get $F(L) = 0$. The boundary conditions for the first ordinary differential equation in (15.7.12) are $F(0) = F(L) = 0$.

We consider the boundary value problem

$$\frac{\mathrm{d}^2F}{\mathrm{d}x^2}(x) - kF(x) = 0 \quad \text{with} \quad F(0) = F(L) = 0\ . \tag{15.7.13}$$

The form of the general solution of (15.7.13) is determined by the roots of the characteristic equation $\lambda^2 - k = 0$.

If $k > 0$, the roots of the characteristic equation are $\pm\sqrt{k}$. Since the roots are real, the general solution of the ordinary differential equation is of the form

$$F(x) = Ae^{\sqrt{k}\,x} + Be^{-\sqrt{k}\,x}\ .$$

However, $F(0) = 0$ implies that $A+B = 0$ and $F(L) = 0$ implies $Ae^{L\sqrt{k}}+Be^{-L\sqrt{k}} = 0$. The only solution for these two equations is $A = B = 0$ because $e^{L\sqrt{k}} - e^{-L\sqrt{k}} = e^{L\sqrt{k}}\left(1 - e^{-2L\sqrt{k}}\right) \neq 0$ for $k \neq 0$. Therefore, the trivial solution is the only solution of the boundary value problem (15.7.13) for $k > 0$.

If $k = 0$, the general solution of the ordinary differential equation is $F(x) = Bx + A$. However, $F(0) = 0$ implies that $A = 0$. Hence $F(L) = 0$ implies that $BL = 0$. Thus $A = B = 0$. Again, the trivial solution is the only solution of the boundary value problem (15.7.13) for $k = 0$.

If $k < 0$, the roots of the characteristic equation are $\pm i\sqrt{-k}$. Since the roots are complex, the general solution of the ordinary differential equation is of the form

$$F(x) = A\cos\left(\sqrt{-k}\,x\right) + B\sin\left(\sqrt{-k}\,x\right) .$$

However, $F(0) = 0$ implies that $A = 0$. Hence $F(L) = 0$ implies that $B\sin\left(L\sqrt{-k}\right) = 0$. If we take $B = 0$, we get the trivial solution. We must therefore have $\sin\left(L\sqrt{-k}\right) = 0$ with $k \neq 0$. This implies that $k = k_n = -(n\pi/L)^2$ for n a positive integer. The boundary value problem (15.7.13) has non-trivial solutions only for $k = k_n = -(n\pi/L)^2 < 0$ with n a positive integer, and the solutions associated to k_n are of the form

$$F(x) = F_n(x) = B_n \sin\left(\frac{n\pi x}{L}\right) .$$

We only need to consider the second ordinary differential equation in (15.7.12) with $k = k_n = -(n\pi/L)^2$; namely,

$$\frac{\mathrm{d}^2 G}{\mathrm{d}t^2}(t) + \left(\frac{cn\pi}{L}\right)^2 G(t) = 0 ,$$

where n is a positive integer. For each n, this is a second order ordinary differential equation with the characteristic equation $\lambda^2 + (cn\pi/L)^2 = 0$. The two roots of this equation are the complex numbers $\lambda_\pm = \pm(cn\pi/L)i$. The general solution is therefore

$$G(t) = G_n(t) = C_n \cos\left(\frac{cn\pi t}{L}\right) + D_n \sin\left(\frac{cn\pi t}{L}\right) .$$

We have found that the functions

$$u_n(x,t) = F_n(x)G_n(t) = \left(a_n \cos\left(\frac{cn\pi t}{L}\right) + b_n \sin\left(\frac{cn\pi t}{L}\right)\right)\sin\left(\frac{n\pi\,x}{L}\right)$$

for $n > 0$ satisfy the wave equation (15.7.1) and the boundary conditions (15.7.2). The constant a_n is the product of the constants B_n and C_n, and b_n is the product of B_n and D_n.

To satisfy the initial conditions, we seek a solution of the form

$$u(x,t) = \sum_{n=1}^{\infty} u_n(x,t) = \sum_{n=1}^{\infty}\left(a_n \cos\left(\frac{cn\pi t}{L}\right) + b_n \sin\left(\frac{cn\pi t}{L}\right)\right)\sin\left(\frac{n\pi\,x}{L}\right) .$$

From $u(x,0) = f(x)$, we get

$$f(x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi x}{L}\right) \quad , \quad 0 < x < L .$$

This is the Fourier sine series of f. The coefficients of this series are given by

$$a_n = \frac{2}{L}\int_0^L f(x)\sin\left(\frac{n\pi x}{L}\right)\mathrm{d}x .$$

From $\displaystyle \frac{\partial u}{\partial t}(x,0) = g(x)$, we get

$$g(x) = \sum_{n=1}^{\infty} \frac{cn\pi b_n}{L} \sin\left(\frac{n\pi x}{L}\right) \quad , \quad 0 < x < L \ .$$

This is the Fourier sine series of g. The formula to compute the coefficients of this Fourier series yields

$$b_n = \frac{2}{cn\pi} \int_0^L g(x) \sin\left(\frac{n\pi x}{L}\right) \, dx \ .$$

The solution of the wave equation (15.7.1) with the boundary conditions (15.7.2) and the initial conditions (15.7.3) is therefore

$$\begin{aligned} u(x,t) &= \sum_{n=1}^{\infty} a_n \cos\left(\frac{cn\pi t}{L}\right) \sin\left(\frac{n\pi x}{L}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{cn\pi t}{L}\right) \sin\left(\frac{n\pi x}{L}\right) \\ &= \sum_{n=1}^{\infty} \frac{a_n}{2} \left(\sin\left(\frac{n\pi}{L}(x+ct)\right) + \sin\left(\frac{n\pi}{L}(x-ct)\right) \right) \\ &\quad + \sum_{n=1}^{\infty} \frac{b_n}{2} \left(\cos\left(\frac{n\pi}{L}(x-ct)\right) - \cos\left(\frac{n\pi}{L}(x+ct)\right) \right) \ . \end{aligned}$$

The solution is of the form

$$u(x,t) = \frac{1}{2}\left(f(x+ct) + f(x-ct)\right) + \frac{1}{2c} \int_{x-ct}^{x+ct} g(s) \, ds$$

because

$$\sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi s}{L}\right) = f(s)$$

and

$$\frac{d}{ds}\left(\sum_{n=1}^{\infty} b_n \cos\left(\frac{n\pi s}{L}\right) \right) = -\sum_{n=1}^{\infty} \frac{n\pi b_n}{L} \sin\left(\frac{n\pi s}{L}\right) = -\frac{1}{c} g(s) \ .$$

In the discussion above, it is obviously assumed that f and g, initially defined on the interval $[0,L]$, are extended to even and periodic functions of period $2L$ on the real line.

The domain of dependence of $u(\tilde{x},\tilde{t})$ in the region $R = \{(x,t) : 0 \leq x \leq L \text{ and } t \geq 0\}$ is a little more complex than for the wave equation on the real line but still depends on the two **characteristic lines** $x = ct + \tilde{x} - c\tilde{t}$ and $x = -ct + \tilde{x} + c\tilde{t}$ (if we consider all the possible reflections of these two lines through the lines $x = 0$ and $x = L$). This two characteristic lines play a crucial role in the convergence of finite difference schemes to numerically solve the wave equation.

If we consider the finite difference scheme in Algorithm 15.2.11, we may define the **numerical domain of dependence** of $w_{i,j}$ as the set of values $\{w_{r,s} : 0 \leq s \leq j \text{ and } s+i-j \leq r \leq -s+i+j\}$. This region is illustrated in Figure 15.6.

Suppose that g and f in the definition of the initial conditions for the wave equation change drastically at a point $a \in]0,L[$. For instance, $g(x) = f(x) = 0$ for $x < a$ and increase exponentially for $x > a$. Suppose that $(\tilde{x},\tilde{t})$ is such that $\tilde{x} < a < \tilde{x} + c\tilde{t}$. Let us consider grids

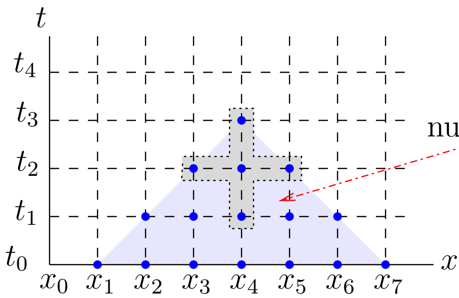


Figure 15.6: Domain of dependence for $w_{4,3}$ associated to the finite difference scheme in Algorithm 15.2.11.

such that $\rho = \Delta x/\Delta t$ is constant and satisfies $\rho < c$, and such that $(\tilde{x}, \tilde{t}) = (x_r, t_s)$ for some r and s. Namely, $(\tilde{x}, \tilde{t})$ is part of all the grids that we consider. We assume that ρ is small enough (or alternatively, that a is large enough) to have $x_{r+s} = (r+s)\Delta x < a$. This situation is completely summarized in Figure 15.7.

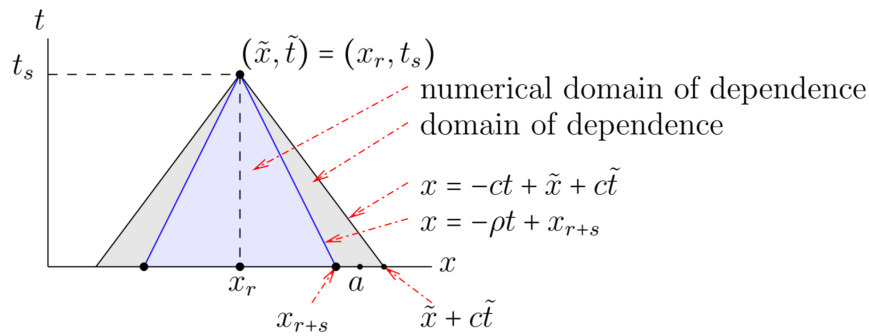


Figure 15.7: Comparison between the domain of dependence for the wave equation and the finite difference scheme in Algorithm 15.2.11 for $\Delta x/\Delta t < c$.

It is clear that $w_{r,s}$, the numerical approximation of $u(\tilde{x}, \tilde{t})$, will never take into consideration the values of $f(x)$ and $g(x)$ for $x > a$ as $(\Delta x, \Delta t) \to \mathbf{0}$ with $\Delta x/\Delta t < c$ [10]. But, the domain of dependence of $u(\tilde{x}, \tilde{t})$ tells us that the value of $u(\tilde{x}, \tilde{t})$ depends on the values of $g(x)$ and $f(x)$ for $x > a$. Thus, $w_{r,s}$ will generally not converge to $u(\tilde{x}, \tilde{t})$.

In conclusion, a necessary condition for our finite difference scheme to converge for the wave equation is that $\Delta x/\Delta t \geq c$. Thus, the numerical domain of dependence of the finite difference scheme in Algorithm 15.2.11 must include the domain of dependence of the wave equation. We will give a more rigorous justification later.

The reader may wonder if the conclusion that we have just stated is particular to the wave equation, and if our finite difference scheme may behave more nicely with other hyperbolic

[10] r and s will increase as Δx and Δt converge to 0 but we always assume that $x_r = \tilde{x}$ and $t_s = \tilde{t}$

differential equations. To add more strength to our argument, we will consider hyperbolic systems of linear first order partial differential equations.

Definition 15.7.1

A system of linear first order partial differential equations of the form $\mathbf{v}_t + A\mathbf{v}_x = 0$, where $\mathbf{v} : \mathbb{R}^2 \to \mathbb{R}^n$ and A is a $n \times n$ real matrix, is an **hyperbolic system** if A is a $n \times n$ real symmetric matrix[11]. The system of partial differential equations is **strictly hyperbolic** if all the real eigenvalues of A are distinct.

The wave equation

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$$

can be reduced to a strictly hyperbolic system of first order partial differential equations. Let

$$\mathbf{v} = \begin{pmatrix} u_x \\ u_t/c \end{pmatrix} \text{ and } A = \begin{pmatrix} 0 & c \\ c & 0 \end{pmatrix} .$$

Then

$$\mathbf{v}_t - A\mathbf{v}_x = \begin{pmatrix} u_{xt} \\ u_{tt}/c \end{pmatrix} - \begin{pmatrix} 0 & c \\ c & 0 \end{pmatrix} \begin{pmatrix} u_{xx} \\ u_{xt}/c \end{pmatrix} = \begin{pmatrix} u_{xt} - u_{xt} \\ u_{tt}/c - c u_{xx} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} ,$$

where the eigenvalues of A are $\pm c$.

One of the reasons to study hyperbolic systems of first order partial differential equations is that one can use the method of characteristic[12] to understand the dangers of solving numerically hyperbolic differential equations with finite difference schemes.

Consider the simple **advection equation**

$$c \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0 \quad , \quad -\infty < x < \infty \text{ and } y > 0 \tag{15.7.14}$$

with the initial condition $u(x,0) = g(x)$ for $x \in \mathbb{R}$. The **characteristic equations** for this partial differential equation are

$$x'(t) = c \ , \ y'(t) = 1 \text{ and } u'(t) = 0$$

with the initial conditions

$$x(0) = s \ , \ y(0) = 0 \text{ and } u(0) = g(s) \ .$$

The equation $x'(t) = c$ with $x(0) = s$ yields $x = ct + s$, the equation $y'(t) = 1$ with $y(0) = 0$ yields $y = t$ and the equation $u'(t) = 0$ with $u(0) = g(s)$ yields $u = g(s)$. We can solve $x = ct+s$ and $y = t$ in terms of s and t to get $t = y$ and $s = x - cy$. If we substitute this value of s in the expression for u, we get the solution $u = u(x,y) = g(x - cy)$. The function u is constant

[11]It is shown in linear algebra courses that real symmetric matrices have only real eigenvalues.

[12]The reader may consult an introductory book on partial differential equations like [32] to learn more about the method of characteristic to solve some partial differential equations.

along the characteristic lines $x - cy = a$ for $a \in \mathbb{R}$. The value of u at (x, y) is the value of u at the points below and to the left of x along the line $x - cy = a$.

Consider the following finite difference scheme to numerically solve (15.7.14) with $u(x, 0) = g(x)$ for $x \in \mathbb{R}$. Let $x_i = i\,\Delta x$ for $i \in \mathbb{Z}$ and $y_j = j\,\Delta y$ for $j \geq 0$. An approximation $w_{i,j}$ of $u(x_i, y_j)$ is provided by the solution of the finite difference equation

$$c\,\frac{w_{i+1,j} - w_{i,j}}{\Delta x} + \frac{w_{i,j+1} - w_{i,j}}{\Delta y} = 0 \tag{15.7.15}$$

for $i \in \mathbb{Z}$ and $j \geq 0$, where $w_{i,0} = g(x_i)$ for all i. This scheme is illustrated in Figure 15.8.

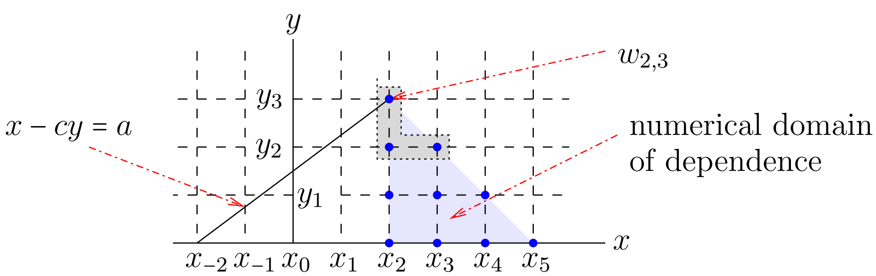


Figure 15.8: Schematic representation of the finite difference scheme given in (15.7.15) and the numerical domain of dependence for $w_{2,3}$.

We get

$$w_{i,j+1} = w_{i,j} - \frac{c\Delta y}{\Delta x}\left(w_{i+1,j} - w_{i,j}\right) .$$

Thus $w_{i,j+1}$ depends on the values $w_{i,j}$ and $w_{i+1,j}$. The first value is at the point (x_i, y_j) straight below (x_i, y_{j+1}) and the other value is at the point (x_{i+1}, y_j) below and to the right of (x_i, y_{j+1}). The numerical domain of dependence for $w_{2,3}$ is illustrated in Figure 15.8. The numerical domain of dependence of $w_{i,j+1}$ does not contain the characteristic line $x - cy = a$ through (x_i, y_{j+1}) that defines $u(x_i, y_{j+1})$. The finite difference scheme in (15.7.15) will generally not converge to the solution of the advection equation whatever the relation between Δy and Δx.

Let us try another finite difference scheme to numerically solve (15.7.14) with $u(x, 0) = g(x)$ for $x \in \mathbb{R}$. Let $x_i = i\,\Delta x$ for $i \in \mathbb{Z}$ and $y_j = j\,\Delta y$ for $j \geq 0$ as before. An approximation $w_{i,j}$ of $u(x_i, y_j)$ is provided by the solution of the finite difference equation

$$c\,\frac{w_{i,j} - w_{i-1,j}}{\Delta x} + \frac{w_{i,j+1} - w_{i,j}}{\Delta y} = 0 \tag{15.7.16}$$

for $i \in \mathbb{Z}$ and $j \geq 0$, where $w_{i,0} = g(x_i)$ for all i. This scheme is illustrated in Figure 15.9.

We get

$$w_{i,j+1} = w_{i,j} - \frac{c\Delta y}{\Delta x}\left(w_{i,j} - w_{i-1,j}\right) .$$

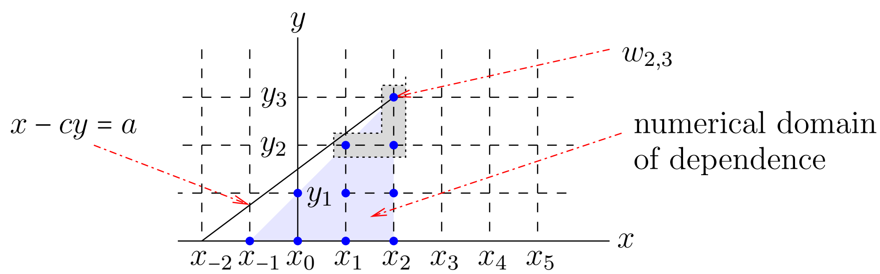


Figure 15.9: Schematic representation of the finite difference scheme given in (15.7.16) and the numerical domain of dependence for $w_{2,3}$ for $\Delta y/\Delta x > 1/c$.

Now, $w_{i,j+1}$ depends on the values $w_{i-1,j}$ and $w_{i,j}$. The first value is at the point (x_{i-1}, y_j) below and to the right of (x_i, y_{j+1}) and the other value is at the point (x_i, y_j) straight below the point (x_i, y_{j+1}). The numerical domain of dependence for $w_{2,3}$ is illustrated in Figure 15.9.

If we assume that $\Delta y/\Delta x > 1/c$, then the numerical domain of dependence of $w_{i,j+1}$ does not contain the characteristic line $x - cy = a$ through (x_i, y_{j+1}) that defines $u(x_i, y_{j+1})$ as can be seen in Figure 15.9. The finite difference scheme in (15.7.15) will generally not converge to the solution of the advection equation. We must therefore assume that $\Delta y/\Delta x \le 1/c$ if we hope to get a converging finite difference scheme. Under this condition, the characteristic line $x - cy = a$ through (x_i, y_{j+1}) is contained in the numerical domain of dependence of $w_{i,j+1}$ for the finite difference scheme given in (15.7.16) as shown in Figure 15.10.

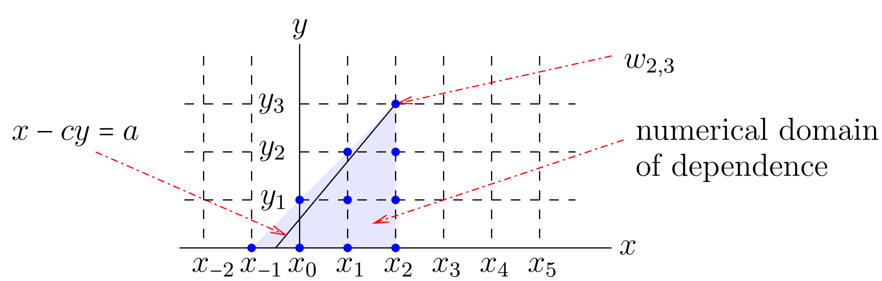


Figure 15.10: The numerical domain of dependence for $w_{2,3}$ associated to the finite difference scheme given in (15.7.16) for $\Delta y/\Delta x \le 1/c$.

In conclusion, given a difference scheme associated to an hyperbolic differential equation, we may say that a necessary condition for this scheme to converge is that its numerical domain of dependence contains the domain of dependence of the associated hyperbolic differential equation. This is known as the **Courant-Friedrichs-Lewy (CFL) condition**. This may be a very restrictive condition as we have seen in the few examples above. For this reason, finite difference schemes are not generally recommended to numerically solve hyperbolic differential equation. Instead, one may use the method of characteristics or finite element methods to numerically solve hyperbolic differential equations.

The issue associated to the domain of dependence when finite difference schemes are used to numerically solve hyperbolic differential equations is not present for elliptic or parabolic differential equations. We were able to find stable, consistent and convergent finite difference schemes for the heat equation with forcing (a parabolic differential equation) and the Dirichlet equation (an elliptic differential equation) that had no constraint on the step sizes.

Nevertheless, we will study the stability, consistency and convergence of the finite difference scheme given in Algorithm 15.2.11. This will provide a rigorous justification for the restriction $\Delta t/\Delta x \leq 1/c$ which is required to get a stable finite difference scheme.

15.7.2 Algorithm 15.2.11

We study the ℓ^2 convergence according to Definition 15.3.32 of Algorithm 15.2.11 which is used to approximate the solution of the wave equation.

As was mentioned in Remark 15.3.34, we will get ℓ^2 consistency according to Definition 15.3.33 if we prove consistency according to Definition 15.3.6.

Proposition 15.7.2

The scheme in Algorithm 15.2.11 is consistent according to Definition 15.3.6.

Proof.
Using the notation introduced in Section 15.3, we have that

$$P\left(u(x,t), \frac{\partial u}{\partial x}(x,t), \frac{\partial u}{\partial y}(x,t), \frac{\partial^2 u}{\partial x^2}(x,t), \ldots\right) = \frac{\partial^2 u}{\partial t^2}(x,t) - c^2 \frac{\partial^2 u}{\partial x^2}(x,t)$$

and

$$P_\Delta\left(w_{i,j}, w_{i,j+1}, w_{i+1,j}, \ldots\right) = \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta t)^2} - c^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2}$$

for the finite difference scheme in Algorithm 15.2.11.

The procedure to deduce the local truncation error for the finite difference equation in Algorithm 15.2.11 is almost identical to the procedure to deduce the local truncation error for the finite difference equation in Algorithm 15.2.6 given in the proof of Proposition 15.6.3. We find

$$\begin{aligned}
\tau_{i,j}(\Delta x, \Delta t, q) &= P\left(q(x_i,t_j), \frac{\partial q}{\partial x}(x_i,t_j), \frac{\partial q}{\partial y}(x_i,t_j), \frac{\partial^2 q}{\partial x^2}(x_i,t_j), \ldots\right) \\
&\qquad - P_\Delta\left(q(x_i,t_j), q(x_i,y_{j+1}), q(x_{i+1},t_j), \ldots\right) \\
&= -\frac{1}{4!}\left(\frac{\partial^4 q}{\partial t^4}(x_i,\zeta_{i,j}) + \frac{\partial^4 q}{\partial t^4}(x_i,\eta_{i,j})\right)(\Delta t)^2 + \frac{c^2}{4!}\left(\frac{\partial^4 q}{\partial x^4}(\mu_{i,j},t_j) + \frac{\partial^4 q}{\partial x^4}(\nu_{i,j},t_j)\right)(\Delta x)^2
\end{aligned}$$

for $\zeta_{i,j}, \eta_{i,j} \in]t_{j-1}, t_{j+1}[$ and $\mu_{i,j}, \nu_{i,j} \in]x_{i-1}, x_{i+1}[$. If the partial derivatives of order up to four are continuous on the compact set $R = \{(x,t) : 0 \leq x \leq L \text{ and } 0 \leq t \leq T\}$, then there exists H

such that
$$\max_{(x,t)\in R}\left|\frac{\partial^4 q}{\partial x^4}(x,t)\right| \leq H \quad \text{and} \quad \max_{(x,t)\in R}\left|\frac{\partial^4 q}{\partial t^4}(x,t)\right| \leq H \ .$$
Hence
$$\max\{\tau_{i,j} : 0 < i < N \text{ and } 0 < j < M\} \leq \frac{H}{12}(\Delta t)^2 + \frac{c^2 H}{12}(\Delta x)^2 \to 0 \qquad \text{as} \quad \min\{N, M\} \to \infty \tag{15.7.17}$$
since $\Delta x = L/N$ and $\Delta t = T/M$ converge to 0 as N and M converge to infinity.

Since $B(u(x,y),\ldots) = u(x,y)$, $B_\Delta(w_{i,j},\ldots) = w_{i,j}$ and $w_{i,j} = u(x_i, t_j)$ for (i,j) such that $(x_i, t_j) \in \partial R_\Delta$, we get from (15.7.17) that Definition 15.3.6 is satisfied. ∎

Proposition 15.7.3

The finite difference scheme in Algorithm 15.2.11 is ℓ^2-stable when
$$\frac{\Delta t}{\Delta x} \leq \frac{1}{c} \ .$$

Proof.
Since the definition of stability that we have presented were for one-step finite difference schemes, we have to use a little trick to prove ℓ^2 stability for Algorithm 15.2.11.

To use Definition 15.3.35, we have to rewrite the finite difference schemes in the $\mathbf{v}_{j+1} = \tilde{Q}_\alpha \mathbf{v}_j + \tilde{\mathbf{B}}_j$, where
$$\mathbf{v}_j = \begin{pmatrix} \mathbf{w}_{j-1} \\ \mathbf{w}_j \end{pmatrix} \ .$$
Namely,
$$\begin{pmatrix} w_j \\ w_{j+1} \end{pmatrix} = \begin{pmatrix} 0 & \mathrm{Id} \\ \mathrm{Id} & J \end{pmatrix} \begin{pmatrix} w_{j-1} \\ w_j \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{B}_j \end{pmatrix}$$
for $0 \leq j < M$, where J and B_j are defined at the end of Section 15.2.3.

Since the matrix $\tilde{Q}_k \equiv \begin{pmatrix} 0 & \mathrm{Id} \\ \mathrm{Id} & J \end{pmatrix}$ is symmetric, we may use Proposition 15.3.36. We need to find all the eigenvalues of $\tilde{Q}_\alpha$. From Proposition 15.4.1, we have that the eigenvalues of J are
$$\lambda_k = -2 + 2\alpha - 2\alpha\cos\left(\frac{k\pi}{N}\right) = -2 + 4\alpha\sin^2\left(\frac{k\pi}{2N}\right) \quad , \quad 0 < k < N \ .$$
Since the eigenvectors of J associated to λ_k are obviously also eigenvectors of Id and 0, we may use Proposition 15.4.2 to claim that the eigenvalues of $\begin{pmatrix} 0 & 1 \\ 1 & \lambda_i \end{pmatrix}$ for $0 < k < N$ are eigenvalues of $\tilde{Q}_\alpha$. Namely, the $2N-2$ distinct eigenvalues of $\tilde{Q}_\alpha$ are the roots of the characteristic polynomials
$$p_k(\lambda) = \lambda^2 - \lambda_k\lambda - 1 = \lambda^2 - 2\left(1 - 2\alpha\sin^2\left(\frac{k\pi}{2N}\right)\right)\lambda - 1$$

for $0 < k < N$. Let

$$\delta_k = 1 - 2\alpha \sin^2\left(\frac{k\pi}{2N}\right)$$

for $0 < k < N$. We have that $p_k(\lambda) = \lambda^2 - 2\delta_k\lambda + 1 = 0$ for

$$\lambda_\pm = \delta_k \pm \sqrt{\delta_k^2 - 1} \ .$$

We have $\delta_k < 1$ because $\alpha > 0$ and $0 < k < N$. When $\delta_k < -1$, we have $\operatorname{Re}\lambda_- < -1$ and thus $|\lambda_-| > 1$. Hence, we only need to consider $-1 \leq \delta_k < 1$. When $-1 < \delta_k < 1$, we get $\lambda_\pm = \delta_k \pm i\sqrt{1-\delta_k^2} \in \mathbb{C} \setminus \mathbb{R}$ and $|\lambda_-|^2 = |\lambda_+|^2 = \delta_k^2 + (1-\delta_k^2) = 1$. When $\delta_k = -1$, we get $\lambda_- = \lambda_+ = -1$. Thus $|\lambda_-| = |\lambda_+| \leq 1$ only when $-1 \leq \delta_k < 1$.

We have shown that

$$-1 \leq \delta_k = 1 - 2\alpha \sin^2\left(\frac{k\pi}{2N}\right) < 1$$

implies $|\lambda_\pm| \leq 1$. Since $\alpha > 0$, the second inequality is always true. From the first inequality, we get

$$\alpha \sin^2\left(\frac{k\pi}{2N}\right) \leq 1 \ .$$

This can be true for any k and N only if $\alpha \leq 1$; namely,

$$\alpha = \left(\frac{c\Delta t}{\Delta x}\right)^2 \leq 1 \ .$$

∎

Since we have ℓ^2 consistency and ℓ^2 stability, we may conclude that the finite difference scheme in Algorithm 15.2.11 is ℓ^2-convergent if (and only if) $\Delta t/\Delta x \leq 1/c$. This condition says that the numerical domain of dependence of the finite difference scheme must include the domain of dependence of the wave equation as we have seen in Section 15.7.1. This is called a **Courant-Friedrichs-Lewy (CFL) condition** .

Remark 15.7.4

1. We may also determine the L^2 stability as defined in Section 15.3.4 of the finite difference scheme in Algorithm (15.2.11). To use the formulae in Remark 15.3.26, we first have to use the substitute $z = 2\pi x/L$ to obtain a partial differential equation defined for $0 \leq z \leq 2\pi$ and $0 \leq t \leq T$. With this substitution, the wave equation becomes

$$\frac{\partial^2 u}{\partial t^2} - \left(\frac{2\pi c}{L}\right)^2 \frac{\partial^2 u}{\partial z^2} = 0$$

The only difference with the original partial differential equation is that c is replaced by $2\pi c/L$.

We now have $\alpha = \left(\frac{2\pi c\Delta t}{L\Delta z}\right)^2$ in the finite difference scheme in Algorithm 15.2.11. We also have $\Delta z = 2\pi/N$ and $\Delta t = T/M$.

We have (15.3.19) with $a_0 = 1$, $b_{-1} = b_1 = \alpha$, $b_0 = 2(1-\alpha)$, $c_0 = -1$ and all the other a_r, b_r and c_r null. Hence

$$\alpha_k = 1 \quad , \quad \beta_k = \alpha e^{-k\Delta z\, i} + 2(1-\alpha) + \alpha e^{k\Delta z\, i} \quad \text{and} \quad \gamma_k = -1 \ .$$

We have to find the roots of the characteristics polynomials

$$\begin{aligned} p_k(\lambda) &= \alpha_k\lambda^2 - \beta_k\lambda - \gamma_k = \lambda^2 - \left(\alpha e^{-k\Delta z\, i} + 2(1-\alpha) + \alpha e^{k\Delta z\, i}\right)\lambda + 1 \\ &= \lambda^2 - \Big(2 - 2\alpha\big(1+\cos(k\Delta z)\big)\Big)\lambda + 1 = \lambda^2 - \left(2 - 4\alpha\sin^2\left(\frac{k\Delta z}{2}\right)\right)\lambda + 1 \end{aligned} \tag{15.7.18}$$

for $0 \leq k < N$. Let

$$\delta_k = 1 - 2\alpha\sin^2\left(\frac{k\Delta z}{2}\right) \ .$$

We get $p_k(\lambda) = \lambda^2 - 2\delta_k\lambda + 1$ as in the proof of Proposition 15.7.3. Proceeding as was done in that proof, we get

$$\alpha = \left(\frac{2\pi c\Delta t}{L\Delta z}\right)^2 = \left(\frac{c\Delta t}{\Delta x}\right)^2 < 1 \ .$$

We have a strict inequality because $\alpha = 1$ is associated to a root of absolute value 1 but multiplicity 2 for p_k.

2. We did not really need to use the substitution $z = 2\pi x/L$ to reduce the problem to the interval $[0, 2\pi]$ as we have done above. We could have done all the work in Section 15.3.4 assuming periodic function of period L. The space $L^2([0,2\pi])$ is replaced by $L^2([0,L])$ with the norm $\|f\|_2 = \left(\frac{1}{L}\int_0^L |f(x)|^2\,\mathrm{d}x\right)^{1/2}$. The Fourier transform of f is defined by

$$\hat{f}(k) = \frac{1}{L}\int_0^L f(x)e^{-(2\pi kx/L)i}\,\mathrm{d}x$$

for $k \in \mathbb{Z}$. We seek a solution of (15.3.20) of the form

$$w_j = \sum_{k\in\mathbb{Z}} A_{k,j}e^{(2\pi kx/L)i}$$

for some $A_{k,j} \in \mathbb{R}$. We get (15.3.21) with

$$\alpha_k = \sum_{r=-m}^{m} a_r e^{(2\pi kr\Delta x/L)i} \quad , \quad \beta_k = \sum_{r=-m}^{m} b_r e^{(2\pi kr\Delta x/L)i} \quad \text{and} \quad \gamma_k = \sum_{r=-m}^{m} c_r e^{(2\pi kr\Delta x/L)i} \ .$$

The values of α_k, β_k and γ_k above are the same values defined in (15.3.22) because $\dfrac{2\pi kr\Delta x}{L} = \dfrac{2\pi kr}{N} = kr\Delta z$, where $z = 2\pi x/L$ is the substitution above.

3. We do not need to remember the formulae for α_k, β_k and γ_k to find the characteristic polynomials $\alpha_k\lambda^2 - \beta_k\lambda - \gamma_k$. These characteristic polynomials are given by the coefficients of $e^{(2\pi kx/L)i}$ after we substitute the expression of w_j above in (15.3.20). So, it

suffices to substitute $w_{n,j} = \lambda^j e^{(2\pi k x_n/L)i}$ in (15.3.19) to get, after some simplifications, the characteristic polynomial associated to k.

For instance, for the finite difference scheme in Algorithm 15.2.11 which is considered in the present section, if we substitute $w_{s,j} = \lambda^j e^{(2\pi k x_s/L)i}$ in (15.2.11) [13], we get

$$\begin{aligned}0 &= \lambda^{j+1}e^{(2\pi k x_s/L)i} - 2\lambda^j e^{(2\pi k x_s/L)i} + \lambda^{j-1}e^{(2\pi k x_s/L)i} \\ &\quad - \alpha\left(\lambda^j e^{(2\pi k(x_s+\Delta x)/L)i} - 2\lambda^j e^{(2\pi k x_s/L)i} + \lambda^j e^{(2\pi(x_s-\Delta x)/L)i}\right) .\end{aligned}$$

If we divide by $\lambda^{j-1}e^{(2\pi k x_s/L)i}$, we get

$$\begin{aligned}0 &= \lambda^2 - 2\lambda + 1 - \alpha\left(e^{(2\pi k\Delta x/L)i} - 2 + e^{-(2\pi k\Delta x/L)i}\right)\lambda \\ &= \lambda^2 - 2\left(1 - 2\alpha\sin^2\left(\frac{\pi k\Delta x}{L}\right)\right)\lambda + 1 . \end{aligned} \tag{15.7.19}$$

This is $p_k(\lambda) = 0$ with p_k defined in (15.7.18) because the substitution $z = 2\pi x/L$ yields $\Delta z = 2\pi\Delta x/L$.

4. Finally, we may study the ℓ^2 stability as presented in Section 15.3.2. More precisely, we may use the content of Remark 15.3.22. The finite difference scheme in Algorithm 15.2.11 is of the form (15.3.14) with $a_0 = 1$, $b_{-1} = b_1 = -\alpha$, $b_0 = -2(1-\alpha)$, $c_0 = 1$ and the other a_r, b_r and c_r are null.

   Note that we assume that we are in the context of Item 1 with $\alpha = \left(\frac{2\pi c\Delta t}{L\Delta z}\right)^2$ because the formulae in Remark 15.3.22 are for 2π periodic functions in their space variable. The characteristic polynomial in (15.3.15) is therefore

$$(\lambda(z))^2 - \left(\alpha e^{-zi} + 2(1-\alpha) + \alpha e^{zi}\right)\lambda(z) + 1 = (\lambda(z))^2 - 2\left(1 - 2\alpha\sin^2(z)\right)\lambda(z) + 1 .$$

   Let $\gamma(z) = 1 - 2\alpha\sin^2(z)$. We find

$$\lambda_\pm(z) = \gamma(z) \pm \sqrt{\gamma^2(z) - 1}$$

   for $z \in [0, 2p]$. To have stability, we need to have $|\lambda_\pm(z)| \le 1$ for all z. This is possible only if $\alpha \le 1$. We again get

$$\alpha = \left(\frac{2\pi c\Delta t}{L\Delta z}\right)^2 = \left(\frac{c\Delta t}{\Delta x}\right)^2 < 1 .$$

♠

15.8 Exercises

Question 15.1

Proof that the Crank-Nicolson scheme in Algorithm 15.2.2 is stable according to Definition 15.3.8 if we assume that $\dfrac{c^2\Delta t}{2(\Delta x)^2} \le 1$. This condition is not required but the proof is easier with it.

[13] Where the index i is replaced by s because i is presently used as the complex number such that $i^2 = -1$.

Question 15.2

Let E be a Hilbert space with the norm $\|\cdot\|$ defined by a scalar product $\langle\cdot,\cdot\rangle$, and let $P : E \to E$ be a normal operator.

a) Prove that $\|P^2\| = \|PP^*\|$.

b) Use (a) to prove that $\|P^2\| = \|P\|^2$.

Question 15.3

Prove that the finite difference scheme in Algorithm 15.2.1 is ℓ^2-stable according to Definition 15.3.35.

Hint: Proposition 15.3.36.

Chapter 16

Solutions to Selected Exercises

Chapter 1 : Computer Arithmetic

Question 1.1
The exact value is 139/660.

Three-digit chopping arithmetic:

$$\left(\frac{1}{3}-\frac{3}{11}\right)+\frac{3}{20}=(0.333-0.272)+0.150=0.0610+0.150=0.211\ .$$

The relative error is $\dfrac{|0.211-139/660|}{139/660}\approx 0.00187$.

Three-digit rounding arithmetic:

$$\left(\frac{1}{3}-\frac{3}{11}\right)+\frac{3}{20}=(0.333-0.273)+0.150=0.0600+0.150=0.210\ .$$

The relative error is $\dfrac{|0.210-139/660|}{139/660}\approx 0.00288$.

Question 1.2
The exact answer for the sum is $1.549767731166541\ldots$

If we sum for $i=1$ to $i=10$, we have

$$\begin{aligned}\sum_{i=1}^{10}\frac{1}{i^2} &= \left(\left(\left(\left(\left(\left(\left(\left(1+\frac{1}{4}\right)+\frac{1}{9}\right)+\frac{1}{16}\right)+\frac{1}{25}\right)+\frac{1}{36}\right)+\frac{1}{49}\right)+\frac{1}{64}\right)+\frac{1}{81}\right)+\frac{1}{100}\\
&= ((((((((1+0.25)+0.111)+0.0625)+0.04)+0.0277)+0.0204)+0.0156)\\
&\qquad +0.0123)+0.01\\
&= (((((((1.25+0.111)+0.0625)+0.04)+0.0277)+0.0204)+0.0156)+0.0123)+0.01\\
&= ((((((1.36+0.0625)+0.04)+0.0277)+0.0204)+0.0156)+0.0123)+0.01\\
&= (((((1.42+0.04)+0.0277)+0.0204)+0.0156)+0.0123)+0.01\end{aligned}$$

$$
\begin{aligned}
&= ((((1.46 + 0.0277) + 0.0204) + 0.0156) + 0.0123) + 0.01 \\
&= (((1.48 + 0.0204) + 0.0156) + 0.0123) + 0.01 \\
&= ((1.50 + 0.0156) + 0.0123) + 0.01 = (1.51 + 0.0123) + 0.01 = 1.52 + 0.01 = 1.53\ .
\end{aligned}
$$

The relative error is about $\dfrac{|1.53 - 1.549767731166541|}{1.549767731166541} \approx 0.012755$.

If we sum for $i = 10$ to $i = 1$, we have

$$
\begin{aligned}
\sum_{i=1}^{10} \frac{1}{i^2} &= 1 + \left(\frac{1}{4} + \left(\frac{1}{9} + \left(\frac{1}{16} + \left(\frac{1}{25} + \left(\frac{1}{36} + \left(\frac{1}{49} + \left(\frac{1}{64} + \left(\frac{1}{81} + \frac{1}{100}\right)\right)\right)\right)\right)\right)\right)\right) \\
&= 1 + (0.25 + (0.111 + (0.0625 + (0.04 + (0.0277 + (0.0204 + (0.0156 \\
&\qquad + (0.0123 + 0.01)))))))) \\
&= 1 + (0.25 + (0.111 + (0.0625 + (0.04 + (0.0277 + (0.0204 + (0.0156 + 0.0223))))))) \\
&= 1 + (0.25 + (0.111 + (0.0625 + (0.04 + (0.0277 + (0.0204 + 0.0379)))))) \\
&= 1 + (0.25 + (0.111 + (0.0625 + (0.04 + (0.0277 + 0.0583))))) \\
&= 1 + (0.25 + (0.111 + (0.0625 + (0.04 + 0.0860)))) \\
&= 1 + (0.25 + (0.111 + (0.0625 + 0.126))) \\
&= 1 + (0.25 + (0.111 + 0.188)) = 1 + (0.25 + 0.299) = 1 + 0.549 = 1.54\ .
\end{aligned}
$$

The relative error is about $\dfrac{|1.54 - 1.549767731166541|}{1.549767731166541} \approx 0.0063027$.

It is more accurate to compute the sum by starting with the smallest terms to avoid as much as possible the lost of significant digits associated to the addition of a (very) large number with a (very) small number.

Question 1.3

The numbers should be summed from the smallest to the largest. We do not want to add a large number to a small number. So we compute

$$
\begin{aligned}
&\left(\left(\left(\left(\frac{1}{5!} + \frac{1}{4!}\right) + \frac{1}{3!}\right) + \frac{1}{2!}\right) + \frac{1}{1!}\right) + \frac{1}{0!} = \left(\left(\left(\left(\frac{1}{120} + \frac{1}{24}\right) + \frac{1}{6}\right) + \frac{1}{2}\right) + 1\right) + 1 \\
&= ((((0.008333 + 0.04167) + 0.1667) + 0.5) + 1) + 1 = (((0.05000 + 0.1667) + 0.5) + 1) + 1 \\
&= ((0.2167 + 0.5) + 1) + 1 = (0.7167 + 1) + 1 = 1.717 + 1 = 2.717\ .
\end{aligned}
$$

The absolute error is $|e - 2.717| \approx 0.128183 \times 10^{-2}$ and the relative error is $\dfrac{|e - 2.717|}{e} \approx 0.4715583 \times 10^{-3}$. Since 4 is the largest value of r such that the relative error is smaller than 5×10^{-r}, there are 4 significant digits.

Question 1.4

Suppose that the mantissa of the normalized representation of the numbers has ten digits. Then, the ten-digit representation of $\cos(0.25)$ is 0.9689124217. Using 10-digit rounding arithmetic, we have that $1 - \cos(0.25) \approx 0.310875783 \times 10^{-1}$. The mantissa of the result has only nine digits, a lost of one digit.

This illustrates the importance of not subtracting two numbers that are almost equal.

Question 1.5
We have that $0.22345 \le x < 0.22355$ and $0.321445 \le y < 0.321455$. Hence,

$$0.695120623415408 \approx \frac{0.22345}{0.321455} < \frac{x}{y} < \frac{0.22355}{0.321445} \approx 0.695453343495777 \ .$$

Question 1.6
We have that

$$\frac{|x-\pi|}{|\pi|} < 5\times 10^{-4} \Rightarrow |x-\pi| < 5\pi\times 10^{-4} \Rightarrow -5\pi\times 10^{-4} < x-\pi < 5\pi\times 10^{-4} \ .$$

The interval is

$$3.140021857262998 \approx \pi + -5\pi\times 10^{-4} < x < \pi + 5\pi\times 10^{-4} \approx 3.143163449916588 \ .$$

Question 1.7
a) There are two possible formulae to compute the smallest root of the polynomial ax^2+bx+c, either $x_- = \dfrac{-b-\sqrt{b^2-4ac}}{2a}$ or $x_- = \dfrac{2c}{-b+\sqrt{b^2-4ac}}$. For the polynomial given in the question, since $b<0$ and $\sqrt{b^2-4ac}\approx b$, the operation $-b-\sqrt{b^2-4ac}$ is not suggested because we will subtract two numbers which are almost equal. We risk to lose a lot of significant digits. Therefore, to avoid this problem, we should choose $x_- = \dfrac{2c}{-b+\sqrt{b^2-4ac}}$.

b) Using 4-digit rounding arithmetic, we have $b^2 = 55230$, $4ac = 12$, $b^2-4ac = 55220$, $\sqrt{b^2-4ac} = 235$, $-b+\sqrt{b^2-4ac} = 470$, $2c = 6$ and finally

$$\tilde{x}_- = \frac{2c}{-b+\sqrt{b^2-4ac}} = \frac{6}{470} = 0.01277 \ .$$

c) Using $x_- = 0.012766651010\ldots$, we get the absolute error $\epsilon = |\tilde{x}_- - x_-| = 0.334899\times 10^{-5}$ and the relative error $\epsilon_r = \dfrac{\epsilon}{|x_-|} = 0.262323\times 10^{-3}$. The number of significant digits is 4 because it is the largest value of r such that $\epsilon_r < 5\times 10^{-r}$.

Question 1.8
If x and y are very large, x^2+y^2 can produce an overflow. To avoid overflow, we use one of the following equivalent expressions for $\sqrt{x^2+y^2}$.

$$\sqrt{x^2+y^2} = x\sqrt{1+\left(\frac{y}{x}\right)^2} \quad \text{or} \quad \sqrt{x^2+y^2} = y\sqrt{\left(\frac{x}{y}\right)^2+1} \ .$$

Hopefully, one of x/y or y/x will be small.

Question 1.9
There is a loss of significant digits because we subtract two almost equal numbers. We should use the relation

$$\ln(1+x) - \ln(x) = \ln\left(\frac{1+x}{x}\right)$$

when x is large.

Question 1.10

The problem with $1 - \cos(x)$ for x near 0 is the subtraction of almost equal numbers. One way to eliminate this subtraction of almost equal numbers is with the formula

$$1 - \cos(x) = (1 - \cos(x)) \left(\frac{1 + \cos(x)}{1 + \cos(x)} \right) = \frac{1 - \cos^2(x)}{1 + \cos(x)} = \frac{\sin^2(x)}{1 + \cos(x)} \ .$$

Note that we have not introduce any division by a really small number because $1 + \cos(x) \approx 2$ for x near 0.

Question 1.11

The problem with $\sqrt{x^4 + 4} - 2$ for x near 0 is the subtraction of almost equal numbers. If x is closed to 0, then x^4 is closer to 0 and $\sqrt{x^4 + 4} \approx \sqrt{4} = 2$. One way to eliminate this subtraction of almost equal numbers is with the formula

$$\sqrt{x^4 + 4} - 2 = (\sqrt{x^4 + 4} - 2) \left(\frac{\sqrt{x^4 + 4} + 2}{\sqrt{x^4 + 4} + 2} \right) = \frac{x^4}{\sqrt{x^4 + 4} + 2} \ .$$

Note that $\sqrt{x^4 + 4} + 2 \approx 4$ for x near 0 and so there is no risk of division by a really small number.

Question 1.12

$\tilde{x}$	x	absolute error $\lvert\tilde{x} - x\rvert$	relative error $\lvert\tilde{x} - x\rvert/\lvert x\rvert$	significant digits
18.66600092909	18.6666519729...	$0.65104387\ldots \times 10^{-3}$	$0.34877378\ldots \times 10^{-4}$	5
0.333329	0.3333332888...	$0.42888888\ldots \times 10^{-5}$	$0.12866668 \times 10^{-4}$	5
1.33382	1.33382044913...	$0.44913624 \times 10^{-6}$	$0.33672916 \times 10^{-6}$	7

Question 1.13

We seek a solution of the form $x_n = \lambda^n$. If we substitute this value of x_n into (1.7.1), we get

$$\lambda^n = 2\lambda^{n-1} + \lambda^{n-2} \ .$$

If $\lambda \neq 0$, we can divide by λ^{n-2} on both sides of the equality to get $\lambda^2 = 2\lambda + 1$; namely, $\lambda^2 - 2\lambda - 1 = 0$. The roots of this polynomial are $\lambda_\pm = 1 \pm \sqrt{2}$.

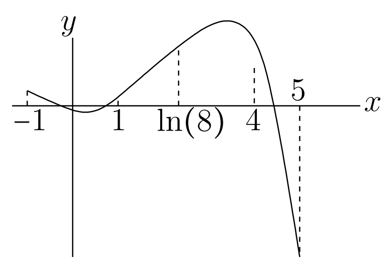
The general solution of (1.7.1) is

$$x_n = \alpha_1\,(1 + \sqrt{2})^n + \alpha_2\,(1 - \sqrt{2})^n$$

Since $1 + \sqrt{2} > 1 > |1 - \sqrt{2}|$, the term $(1 + \sqrt{2})^n$ will dominate as n increase. Thus, because of rounding errors, the formula for the solution of (1.7.1) will always eventually produce a sequence $\{x_n\}_{n=0}^\infty$ that will converge to ∞ in absolute value as n increase, even if the initials conditions are $x_0 = c$ and $x_1 = c(1 - \sqrt{2})$ for some constant c.

Chapter 2 : Iterative Methods for Nonlinear Equations of One Variable

Question 2.1
We will roughly sketch the graph of $f(x) = 4x^2 - e^x$. This is not easy because even the critical points are hard to find. However, it is easier to determine the intervals of concavity of the function. Since $f''(x) = 8 - e^x$, a potential point of inflection is $x = \ln(8)$. Since $f''(x) > 0$ for $x < \ln(8)$ and $f''(x) < 0$ for $x > \ln(8)$, we have that $\ln(8)$ is a point of inflection, f is concave up for $x < \ln(8)$ and concave down for $x > \ln(8)$. To sketch the graph of f below, we have computed the sign of f at -1, 0, 1, 4 and 5.



There is a unique root in each of the intervals $[-1, 0]$, $[0, 1]$ and $[4, 5]$. There is no other root.

Question 2.2
$\sqrt[3]{25}$ is the unique root of $f(x) = x^3 - 25$.

Since f is a continuous function and $f(2) = -17 < 0 < 2 = f(3)$, it follows from the Intermediate Value Theorem that $\sqrt[3]{25}$, the only root of f, is between 2 and 3.

According to Corollary 2.2.3, to get an approximation of $\sqrt[3]{25}$ with an accuracy of 10^{-4}, we need to select the number of iterations n such that

$$\frac{3-2}{2^n} < 10^{-4} \Rightarrow 10^4 < 2^n \Rightarrow 4\ln(10) < n\ln(2) \Rightarrow \frac{4\ln(10)}{\ln(2)} = 13.28771237\ldots < n \ .$$

Since n is an integer, we may take $n = 14$.

We get the following results from the Bisection Method

n	x_n	a_n	b_n	sign of $f(x_n)$	sign of $f(a_{n-1})$
0		2	3		
1	2.5	2.5	3	−	−
2	2.75	2.75	3	−	−
3	2.875	2.875	3	−	−
4	2.9375	2.875	2.9375	+	−
5	2.90625	2.90625	2.9375	−	−
6	2.921875	2.921875	2.9375	−	−
7	2.9296875	2.921875	2.9296875	+	−
8	2.9257812	2.921875	2.9257812	+	−
9	2.9238281	2.9238281	2.9257812	−	−
10	2.9248047	2.9238281	2.9248047	+	−
11	2.9243164	2.9238281	2.9243164	+	−
12	2.9240723	2.9238281	2.9240723	+	−
13	2.9239502	2.9239502	2.9240723	−	−
14	2.9240112	2.9240112	2.9240723	−	−

After 14 iterations, we find $\sqrt[3]{25} \approx 2.9240112$.

Question 2.3
We have seen in Corollary 2.2.3 that

$$|x_n - r| < |b_n - a_n| = \frac{b_0 - a_0}{2^n} \ .$$

Hence, the relative error satisfies

$$\frac{|x_n - r|}{|r|} < \frac{(b_0 - a_0)}{2^n\,|r|} \tag{16.1}$$

If n satisfies (2.11.1), we have that

$$\ln(2^n) = n\ln(2) \geq \ln(b_0 - a_0) - \ln(\epsilon) - \ln(a_0) = \ln\left(\frac{b_0 - a_0}{\epsilon\, a_0}\right) \Rightarrow 2^n \geq \frac{b_0 - a_0}{\epsilon\, a_0} \Rightarrow \epsilon\, a_0 \geq \frac{b_0 - a_0}{2^n} \ .$$

Hence, from (16.1), we get

$$\frac{|x_n - r|}{|r|} < \frac{\epsilon\, a_0}{|r|} \leq \epsilon$$

for n satisfying (2.11.1) because $r > a_0 > 0$ implies that $a_0/r < 1$.

Question 2.4
Since x_{n+1} is the middle point of the interval $[a_n, b_n]$, where one of the endpoints is x_n, we get

$$|x_{n+1} - x_n| = \frac{1}{2}(b_n - a_n) = \frac{1}{2}\left(\frac{b_0 - a_0}{2^n}\right) = \frac{b_0 - a_0}{2^{n+1}} \ ,$$

where we have used the property that the length of the interval $[a_n, b_n]$ is $(b_0 - a_0)/2^n$.

Question 2.5
As it is given, the bisection algorithm can never have a sequence $a_0 < a_1 < a_2 < \ldots$.

Let r be the root in $[a_0, b_0]$ approximated by the bisection algorithm. We suppose that $a_0 < a_1 < a_2 < \ldots$ and prove by contradiction that this is not possible.

Since we have an infinite sequence of distinct values $\{a_n\}_{n=0}^{\infty}$, none of a_n or b_n for $n \geq 0$ can be a root of f because of (2) and (4) in Algorithm 2.2.1. Moreover, since only a_n varies, we must have $b_n = b_0$ and $a_n < r$ for all n. But then $b_n - a_n \geq b_0 - r$ for all n. So $\{b_n - a_n\|_{n=0}^{\infty}$ cannot converge to 0. This is our contradiction.

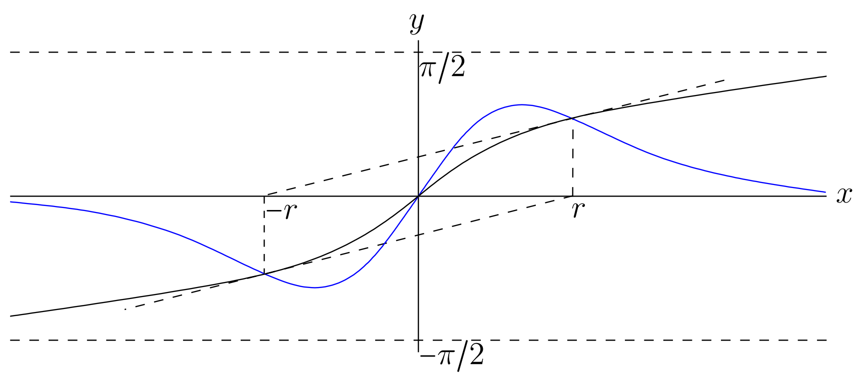
Question 2.8
We first find all points $r > 0$ such that

$$-r = r - \frac{f(r)}{f'(r)} = r - (1+r^2)\arctan(r) \ ;$$

namely, such that

$$\arctan(r) = \frac{2r}{1+r^2} \ .$$

In the figure below, we draw the curves $y = \arctan(x)$ (in black) and $y = 2x/(1+x^2)$ (in blue) on the same coordinate system. Note that the two functions are odd. There is only one intersection for $x > 0$ between these two curves. The x-coordinate of this point is the value of interest.



For $0 < x_0 < r$, we have $-r < x_0 - f(x_0)/f'(x_0) < 0$ and the Newton's method will converge to the root 0 of $\arctan(x)$.

For $x_0 > r$, we have $x_0 - f(x_0)/f'(x_0) < -r$ and the Newton's method will not converge to the root 0 of $\arctan(x)$.

For $x_0 = r$, we have $x_1 = x_0 - f(x_0)/f'(x_0) = -r$, $x_2 = x_1 - f(x_1)/f'(x_1) = r = x_0$, and so on.

We may use Newton's method to approximate r. Let

$$g(x) = \arctan(x) - \frac{2x}{1+x^2} \ .$$

Then

$$g'(x) = \frac{1}{1+x^2} - \frac{2}{1+x^2} + \frac{4x^2}{(1+x^2)^2} = \frac{-1+3x^2}{(1+x^2)^2} \ .$$

With $x_0 = 1$, the formula
$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$$
yields $r \approx 1.3917452002707$ after 5 iterations with an accuracy of at least 10^{-12}.

Question 2.9

We seek a function f such that
$$2x_n - Cx_n^2 = x_n - \frac{f(x_n)}{f'(x_n)} \ .$$
Thus,
$$\frac{f(x)}{f'(x)} = -x + Cx^2 \ .$$
This is the separable differential equation
$$\frac{f'(x)}{f(x)} = \frac{1}{-x + Cx^2} \ .$$
If we integrate both sides with respect to x, we get
$$\begin{aligned}\ln|f(x)| &= \int \frac{f'(x)}{f(x)}\,\mathrm{d}x = \int \frac{1}{-x+Cx^2}\,\mathrm{d}x = \int \frac{1}{x(Cx-1)}\,\mathrm{d}x \\ &= \int \left(-\frac{1}{x} + \frac{C}{Cx-1}\right)\mathrm{d}x = -\ln|x| + \ln|Cx-1| + D = \ln\left(\frac{|Cx-1|}{|x|}\right) + D \ .\end{aligned}$$
Taking the exponential on both sides yields
$$|f(x)| = e^D\,\frac{|Cx-1|}{|x|} \Rightarrow f(x) = E\,\frac{Cx-1}{x}$$
for $E \neq 0$ in $\mathbb{R}$.

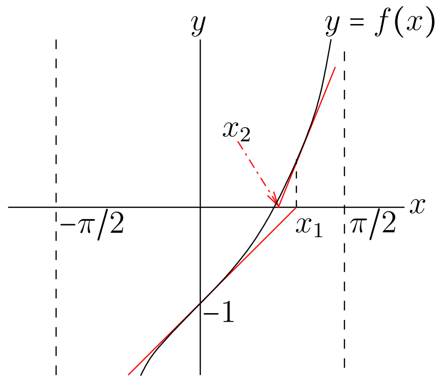
Question 2.10

This is the formula for the Newton's method applied to $f(x) = \tan(x) - 1$. Since $x_0 = 0$, the limit of the sequence above is a root of f between $-\pi/2$ and $\pi/2$. The easiest way to convince us that this is true is to use the graphical interpretation of the Newton's method as the intersection of the tangent line to the graph of f at $(x_n, f(x_n))$ with the x-axis to get x_{n+1}.

All tangents to the graph of f at $(x, f(x))$ for $-\pi/2 < x < \pi/2$ intersect the x-axis between $-\pi/2$ and $\pi/2$ because:

1. $f'(x) = \sec^2(x) > 1$ for $-\pi/2 < x < \pi/2$ with $x \neq 0$, and $f'(0) = 1$:
2. $f''(x) = 2\sec^2(x)\tan(x) \geq 0$ for $0 \leq x < \pi/2$, and $f''(x) \leq 0$ for $-\pi/2 < x \leq 0$;
3. $\displaystyle\lim_{x\to -pi/2^-} f(x) = -\infty$ and $\displaystyle\lim_{x\to pi/2^+} f(x) = \infty$.

Using curve sketching as seen in calculus, we get the following graph.



The solution of $f(x) = 0$ between $-\pi/2$ and $\pi/2$ is the value x between $-\pi/2$ and $\pi/2$ such that $\tan(x) = 1$; namely, $x = \pi/4$. The limit of the sequence above is therefore $\pi/4$.

Question 2.11
The formula for the Newton's method is

$$x_{n+1} = x_n - \frac{\tan(x_n)}{\sec^2(x_n)} = x_n - \sin(x_n)\cos(x_n)$$

for $n = 0, 1, 2, \ldots$ We get

n	x_{n-1}	x_n	$\lvert x_n - x_{n-1}\rvert$
1	5.000000000000	5.272010555445	$0.272010555445 \nless 10^{-8}$
2	5.272010555445	5.721895774546	$0.449885219101 \nless 10^{-8}$
3	5.721895774546	6.172506324972	$0.450610550427 \nless 10^{-8}$
4	6.172506324972	6.282283652706	$0.109777327733 \nless 10^{-8}$
5	6.282283652706	6.283185306691	$0.000901653985 \nless 10^{-8}$
6	6.283185306691	6.283185307180	$0.000000000489 < 10^{-8}$

Starting with $x_0 = 5$, it took 6 iterations to get the approximation $x_6 = 6.283185307180$ of the root of f with the requested accuracy.

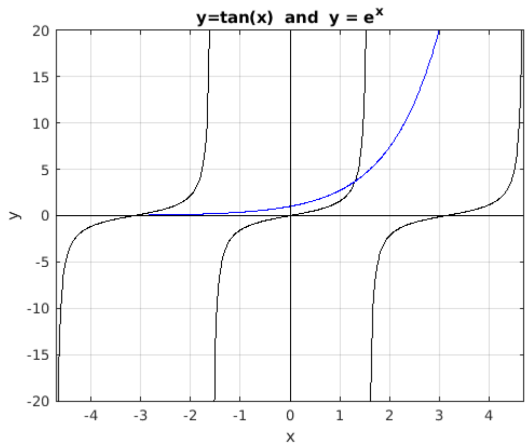
Question 2.12
The formula for the secant method is

$$x_{n+1} = x_n - \frac{e^{x_n} - \tan(x_n)}{\big((e^{x_n} - \tan(x_n)) - (e^{x_{n-1}} - \tan(x_{n-1}))\big)/(x_n - x_{n-1})}$$

for $n = 1, 2, 3, \ldots$ We have

n	x_n	x_{n+1}	$\lvert x_{n+1} - x_n\rvert$
0	1.300000000000	1.350000000000	$0.050000000000 \nless 10^{-8}$
1	1.350000000000	1.305052269533	$0.044947730467 \nless 10^{-8}$
2	1.305052269533	1.306071050733	$0.001018781201 \nless 10^{-8}$
3	1.306071050733	1.306328498317	$0.000257447584 \nless 10^{-8}$
4	1.306328498317	1.306326938521	$0.000001559796 \nless 10^{-8}$
5	1.306326938521	1.306326940423	$0.000000001902 < 10^{-8}$

The secant algorithm must be started with x_0 and x_1 really close to the first positive root of f, the root that we want to approximate, otherwise the secant algorithm will likely converge to a totally different root of f. The first positive root of f is the first intersection of $\tan(x)$ (in black) and e^x (in blue) in the following figure.



Starting with $x_0 = 1.3$ and $x_1 = 1.35$, it took 5 iterations to get the approximation $x_6 = 1.306326940423$ of the root of f with the requested accuracy.

To illustrate how unpredictable Newton's method can be, if we start with $x_0 = 4$ and $x_1 = 5$, it takes 13 iterations to get the approximation $x_{14} = -3.096412304914$ of the first negative root of f instead of the first positive root.

Question 2.13

a) The Taylor polynomial of f of degree one about x_n is $p(x) = f(x_n) + f'(x_n)(x - x_n)$. We have that $f(x) = p(x) + \dfrac{1}{2} f''(\xi_n)(x - x_n)^2$ for some ξ_n between x_n and x. If $x = r$, the root of f in the interval $[0, 1]$, we get

$$
\begin{aligned}
&0 = f(x_n) + f'(x_n)(r - x_n) + \frac{1}{2} f''(\xi_n)(r - x_n)^2 \\
&\Rightarrow -f(x_n) = f'(x_n)(r - x_n) + \frac{1}{2} f''(\xi_n)(r - x_n)^2 \Rightarrow -\frac{f(x_n)}{f'(x_n)} = r - x_n + \frac{f''(\xi_n)}{2f'(x_n)}(r - x_n)^2 \\
&\Rightarrow \underbrace{x_n - \frac{f(x_n)}{f'(x_n)}}_{=x_{n+1}} = r + \frac{f''(\xi_n)}{2f'(x_n)}(r - x_n)^2 \Rightarrow \underbrace{x_{n+1} - r}_{=e_{n+1}} = \frac{f''(\xi_n)}{2f'(x_n)} \underbrace{(r - x_n)^2}_{=e_n^2} .
\end{aligned}
$$

Thus,

$$
e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)} e_n^2 \tag{16.2}
$$

for some ξ_n between x_n and x.

b) You may assume that $x_n \geq 0$ for all n if $x_0 \geq 0$, because

$$
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n - e^{-x_n}}{1 + e^{x_n}} = \frac{(x_n + 1)e^{x_n}}{1 + e^{x_n}} > 0
$$

for $n = 1, 2, 3, \ldots$ by induction. Moreover, we have $f'(x) = 1 + e^{-x}$ and $f''(x) = -e^{-x}$. Hence,

$$\left|\frac{f''(\xi_n)}{f'(x_n)}\right| = \frac{e^{-\xi_n}}{1 + e^{-x_n}} \leq 1$$

for all non-negative numbers ξ_n and x_n.

We use induction to prove (2.11.2). From (16.2) with $n = 0$, we get

$$|e_1| = \left|\frac{f''(\xi_0)}{2f'(x_0)}\right| e_0^2 \leq \frac{e_0^2}{2} \ .$$

This is (2.11.2) for $n = 1$. Suppose that (2.11.2) is true for $n = k$. Then,

$$|e_{k+1}| = \left|\frac{f''(\xi_k)}{2f'(k_n)}\right| e_k^2 \leq \frac{e_k^2}{2} \leq \frac{1}{2}\left(2\left(\frac{x_0}{2}\right)^{2^k}\right)^2 = 2\left(\frac{x_0}{2}\right)^{2^{k+1}} \ .$$

The first equality comes from (16.2) with $n = k$ and the second inequality comes from the hypothesis of induction. We get that (2.11.2) is true for $n = k + 1$. This complete the proof by induction.

c) According to (b), we need to find n such that

$$|e_n| \leq 2\left(\frac{1-r}{2}\right)^{2^n} < 10^{-5} \ .$$

We do not know r but we know that r is between 0 and 1, so $|1-r| < 1$. It is therefore enough to find n such that $2\,(1/2)^{2^n} < 10^{-5}$. Thus, n satisfies

$$2^{2^n-1} > 10^5 \Rightarrow (2^n - 1)\ln(2) > 5\ln(10) \Rightarrow n > \frac{1}{\ln(2)} \ln\left(\frac{5\,\ln(10)}{\ln(2)} + 1\right) \approx 4.1383 \ .$$

We choose $n = 5$.

Question 2.16

a) Let $f(x) = g(x) - x$. A fixed point of g is a root of f and vice-versa. Since f is continuous and $f(0) = 1 > 0 > -1/2 = f(1)$, it follows from the Mean Value Theorem that $f(x) = 0$ for some $x \in]0,1[$. Since $f'(x) = g'(x) - 1 = -2x/(1+x^2)^2 - 1 < 0$ for $x \geq 0$, the function f is strictly decreasing on $[0,1]$. Thus, $f(x) = 0$ for a unique value of $x \in [0,1]$ and, therefore, $g(x) = x$ for a unique value of $x \in [0,1]$.

b) We show that the hypotheses of the Fixed Point Theorem are satisfied; namely, we show that $g([0,1]) \subset [0,1]$ and $|g(x) - g(y)| \leq K|x-y|$ for some $K < 1$ and all $x, y \in [0,1]$.

Since $g'(x) = -2x/(1+x^2)^2 < 0$ for $x > 0$, the function g is strictly decreasing on $]0,\infty[$. Hence, $1 = g(0) \geq g(x) \geq g(1) = 1/2$ for all $x \in [0,1]$. Thus, $g : [0,1] \to [0,1/2] \subset [0,1]$..

To prove that there exists a positive constant $K < 1$ such that $|g(x) - g(y)| \leq K|x-y|$ for $x, y \in [0,1]$, we show that the maximum of $G(x) = |g'(x)| = 2x/(1+x^2)^2$ on $[0,1]$ is less than 1 and use this maximum as our constant K as explained in Remark 2.4.4.

We use the maximum principle to find the maximum of G on $[0,1]$. Namely, since G is differentiable on $[0,1]$, the maximum of G is either at the endpoints of the interval or at one of the critical points of G in $[0,1]$ if there is one. The critical points of G are given by $G'(x) = 2(1-3x^2)/(1+x^2)^3 = 0$. There is only one critical point in $[0,1]$. It is $x = 1/\sqrt{3}$. Since $G(0) = 0$, $G(1) = 1/2$ and $G(1/\sqrt{3}) = 3\sqrt{3}/8 < 1$, we have that $G(x) = |g'(x)| \leq K = 3\sqrt{3}/8 < 1$ for all $x \in [0,1]$.

c) Let p be the fixed point of g in $[0,1]$. Since $p > 0$ and $g'(x) \neq 0$ for $x > 0$, we get that $g'(p) \neq 0$. Hence, we have only linear convergence according to Theorem 2.7.2.

Question 2.17

a) We show that g satisfies all the hypothesis of the Fixed Point Theorem on the interval $[1/3,1]$. Hence, from the Fixed Point Theorem, we will have that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ for $n \geq 0$ and $x_0 \in [1/3,1]$ converges to the unique fixed point p of g in the interval $[1/3,1]$.

1. Since $g'(x) = -2^{-x}\ln(2) < 0$ for all x, the function g is strictly decreasing on $[1/3,1]$. Thus,
$$1 > 2^{-1/3} = g(1/3) \geq g(x) \geq g(1) = 0.5$$
for all $x \in [1/3,1]$. Hence, $g : [1/3,1] \to [1/3,1]$.

2. Since $|g'(x)| = 2^{-x}\ln(2) < 2^{-1/3}\ln(2)$ for $x \in [1/3,1]$. We have that $|g(x)-g(y)| \leq K|x-y|$ for all $x, y \in [1/3,1]$ with $K = 2^{-1/3}\ln(2) < 1$ according to Remark 2.4.4.

b) Since $|x_n - p| \leq \dfrac{K^n}{1-K}|x_1 - x_0|$, we need to find n such that
$$\frac{K^n}{1-K}|x_1 - x_0| < 10^{-4} \ ;$$
namely,
$$\frac{1}{-\ln(K)}\ \ln\left(\frac{10^4|x_1 - x_0|}{1-K}\right) < n \ .$$
If $x_0 = 0.5$, we have $x_1 = 1/\sqrt{2}$. Thus, n must satisfy
$$n > \frac{1}{-\ln(2^{-1/3}\ln(2))}\ \ln\left(\frac{10^4|1/2 - 1/\sqrt{2}|}{1-2^{-1/3}\ln(2)}\right) \approx 14.115 \ .$$
Hence, $n = 15$ iterations is enough to reach the accuracy requested.

c) Starting with $x_0 = 0.5$, we compute $x_{n+1} = g(x_n)$ until $|x_{n+1} - x_n| < 10^{-4}$. The first time that this happen is for $n = 10$. We get $x_{11} \approx 0.64120525$ as an approximation of the fixed point of g in the interval $[1/3,1]$.

Question 2.18

a) For $x \neq 0$, we have
$$g(x) = 12 - \frac{20}{x} = x \iff x^2 - 12x + 20 = (x-10)(x-2) = 0 \iff x = 2 \quad \text{or} \quad x = 10 \ .$$

b) We show that g satisfies all the hypothesis of the Fixed Point Theorem on the interval $[9.5, 11.5]$. Hence, from the Fixed Point Theorem, we will have that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ for $n \geq 0$ and $x_0 \in [9.5, 11.5]$ converges to the unique fixed point of g in the interval $[9.5, 11.5]$.

1. Since $g'(x) = 20/x^2 > 0$ for all $x \in [9.5, 11.5]$, the function g is strictly increasing. Thus,
$$9.5 < \frac{188}{19} = g(9.5) \leq g(x) \leq g(11.5) = \frac{236}{23} < 10.5$$
for all $x \in [9.5, 10.5]$. Hence, $g : [9.5, 11.5] \to [9.5, 11.5]$.
2. Since $|g'(x)| = 20/x^2 < 20/9.5^2 = 80/361$ for $x \in [9.5, 11.5]$. We have that $|g(x) - g(y)| \leq K|x - y|$ for all $x, y \in [9.5, 11.5]$ with $K = 80/361 < 1$ according to Remark 2.4.4.

c) Since $|x_n - p| \leq \dfrac{K^n}{1-K}|x_1 - x_0|$, we need to find n such that
$$\frac{K^n}{1-K}|x_1 - x_0| < 10^{-7} \ ;$$
namely,
$$\frac{1}{-\ln(K)} \ln\left(\frac{10^7|x_1 - x_0|}{1-K}\right) < n \ .$$
If $x_0 = 9.5$, we have $x_1 = 188/19$. Thus, n must satisfy
$$n > \frac{1}{-\ln(80/361)} \ln\left(\frac{10^7|9.5 - 188/19|}{1 - 80/361}\right) \approx 10.2459 \ .$$
Hence, $n = 11$ iterations is enough to reach the accuracy requested.

d) Since $g'(x) = 20/x^2 > 0$ for $x \in [9.5, 11.5]$, we have that $g'(p) \neq 0$ at the fixed point $p \in [9.5, 11.5]$. Thus, the method is of order one; the order of the first non-null derivative of g at p.

e) We use the Steffensen's algorithm given in Algorithm 2.8.1. Let $\hat{x}_{-1} = x_0 = 9.5$ and
$$\hat{x}_{n+1} = \hat{x}_n - \frac{(g(\hat{x}_n) - \hat{x}_n)^2}{g(g(\hat{x}_n)) - 2g(\hat{x}_n) + \hat{x}_n}$$
for $n = -1,\ 0,\ 1,\ \ldots$ until $|\hat{x}_{n+1} - \hat{x}_n| < 10^{-7}$. We get $|\hat{x}_{n+1} - \hat{x}_n| < 10^{-7}$ for the first time with $n = 1$. We have $\hat{x}_2 = 10$ to 16 significant digits.

The Steffensen's Algorithm converge faster than the simple Fixed Point Method applied to g because it is of order two.

Question 2.19

a) Since f is a continuous function and $f(1) = e - 3 < 0 < f(2) = e^2 - 5$, it follows from the Intermediate Value Theorem that f has at least one root in the interval $[1, 2]$. To prove that it is unique, we note that $f'(x) = e^x - 2 > 0$ for all $x \in [1, 2]$. So the function is strictly increasing and can therefore cross the x-axis only once.

b) We have

$$f(x) = 0 \iff e^x - 2x - 1 = 0 \iff e^x = 2x + 1 \iff x = \ln(2x+1) = g(x)$$

for $x > -1/2$.

c) We show that g satisfies all the hypothesis of the Fixed Point Theorem on the interval $[1,2]$. Hence, from the Fixed Point Theorem, we will have that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ for $n \geq 0$ and $x_0 \in [1,2]$ converges to the unique fixed point of g in the interval $[1,2]$.

1. Since $g'(x) = 2/(1+2x) > 0$ for all $x \in [1,2]$, the function g is strictly increasing. Thus,
$$1 < \ln(3) = g(1) \leq g(x) \leq g(2) = \ln(5) < 2$$
for all $x \in [1,2]$. Hence, $g : [1,2] \to [1,2]$.

2. Since $|g'(x)| = 2/(2x+1) < 2/3$ for $x \in [1,2]$. We have that $|g(x) - g(y)| \leq K|x-y|$ for all $x, y \in [1,2]$ with $K = 2/3 < 1$ according to Remark 2.4.4.

d) Since $g'(x) \geq g'(2) = 2/5$ for $x \in [1,2]$, we certainly have that $g'(p) \neq 0$ at the fixed point p. So the method is of order one; the order of the first non-null derivative of g at p.

Question 2.20

a) $\sqrt[3]{25}$ is obviously the unique root of $f(x) = x^3 - 25$ because

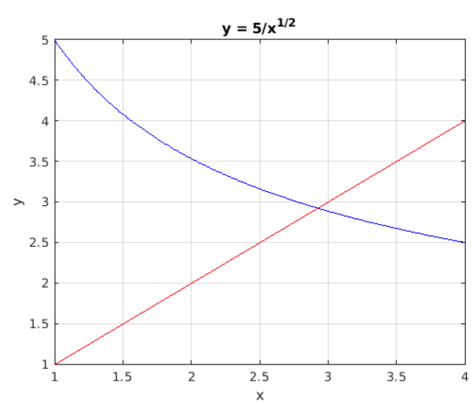
$$x^3 - 25 = 0 \iff x^3 = 25 \iff x = \sqrt[3]{25}\ .$$

b) If $p > 0$, we have

$$f(p) = p^3 - 25 = 0 \iff p^3 = 25 \iff p^2 = \frac{25}{p} \iff p = \frac{5}{\sqrt{p}} = g(p)\ .$$

Since $\sqrt[3]{25}$ is the only (positive) root of f, it is the fixed point $p > 0$ of g.

c) The graph of g (in blue) between 1 and 4 is sketched in the figure below. We have also included the line $y = x$ (in red).



The fixed point of g is given by the point of intersection of the graph of g with the line $y = x$. The fixed point $p = \sqrt[3]{25}$ is closed to 3.

We consider the interval $[2,4]$ that contains p. We show that g satisfies all the hypothesis of the Fixed Point Theorem on the interval $[2,4]$. Hence, from the Fixed Point Theorem, we will have that the sequence $\{x_n\}_{n=0}^{\infty}$ generated by $x_{n+1} = g(x_n)$ for $n \geq 0$ and $x_0 \in [2,4]$ converges to the unique fixed point $p = \sqrt[3]{25}$ of g in the interval $[2,4]$.

1. Since $\sqrt{x}$ is strictly increasing and positive for $x > 0$, we have that g is strictly decreasing for $x > 0$. Thus,
$$4 > 5/\sqrt{2} = g(2) \geq g(x) \geq g(4) = 5/2 > 2$$
for $x \in [2,4]$. Hence, $g : [2,4] \to [2,4]$.

2. Since $|g'(x)| = \dfrac{5}{2x^{3/2}} \leq \dfrac{5}{2^{5/2}}$ for $x \in [2,4]$. We have that $|g(x) - g(y)| \leq K|x-y|$ for all $x, y \in [1,2]$ with $K = \dfrac{5}{2^{5/2}} < 1$ according to Remark 2.4.4.

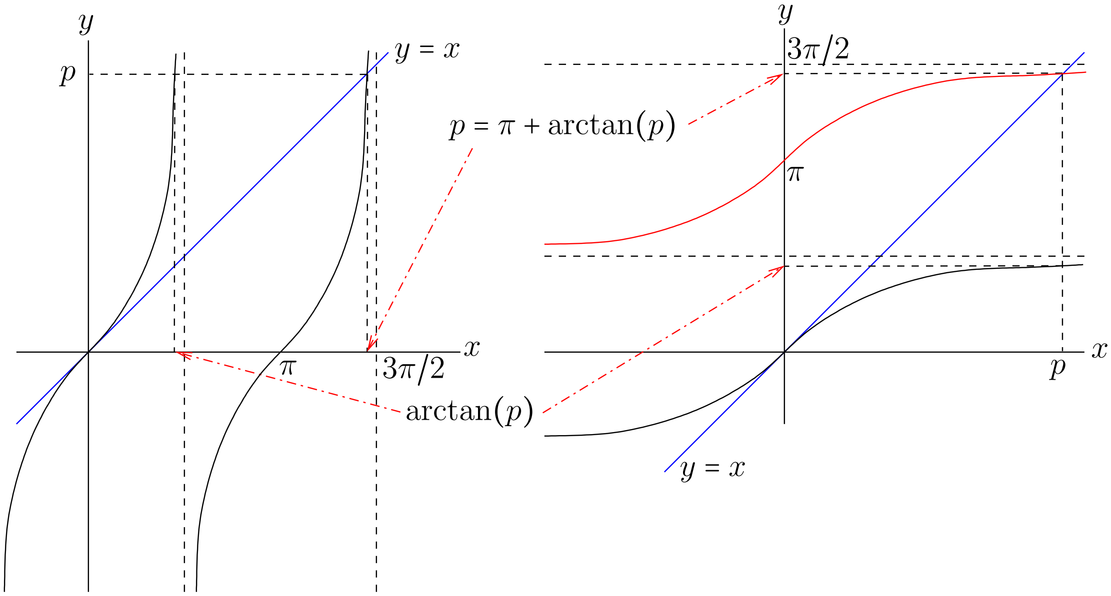
d) Since $|x_n - p| \leq \dfrac{K^n}{1-K}|x_1 - x_0|$, we need to find n such that
$$\frac{K^n}{1-K}|x_1 - x_0| < 10^{-5} \ ;$$
namely,
$$\frac{1}{-\ln(K)} \ln\left(\frac{10^5|x_1 - x_0|}{1-K}\right) < n \ .$$
If $x_0 = 3$, we have $x_1 = 5/\sqrt{3} \approx 2.8867513$. Thus, n must satisfy
$$n > \frac{1}{-\ln(5/2^{5/2})} \ln\left(\frac{10^5|3 - 5/\sqrt{3}|}{1 - 5/2^{5/2}}\right) \approx 105.6598 \ .$$
Hence, $n = 106$ iterations is enough to reach the accuracy requested.

e) Starting with $x_0 = 3$, we compute $x_n = g(x_{n-1})$ until $|x_n - x_{n-1}| < 10^{-5}$. The first time that this happen is for $n = 15$. We get $x_{15} \approx 2.924015$ as an approximation of the fixed point of g in the interval $[2,4]$.

As we can see, the value of n estimated in (d) is a very large overestimation of the number of iterations needed to reach an accuracy of 10^{-5} if we start with $x_0 = 3$. The formula used in (d) to estimate n will generally give a large overestimation. However, we have to keep in mind that the value n obtained in (d) is valid for all x_0, not just for $x_0 = 3$.

Question 2.22

a) The figure below shows the graph of $\tan(x)$ and g (red curve in the graph on the right).



b) We have $g'(x) = 1/(1+x^2)$. Hence g is strictly increasing on $[\pi, 3\pi/2]$ and $|g'(x)| \leq g'(\pi) = 1/(1+\pi^2) < 1$ on $[\pi, 3\pi/2]$. We can then say that:

1. $g : [\pi, 3\pi/2] \to [\pi, 3\pi/2]$ because
$$\pi < \pi + \arctan(\pi) = g(\pi) \leq g(x) \leq g(3\pi/2) = \pi + \arctan(3\pi/2) < 3\pi/2$$
for all $x \in [\pi, 3\pi/2]$ since g is increasing. We have used the fact that $0 < \arctan(x) < \pi/2$ for all $x > 0$.
2. With $K = 1/(1+\pi^2)$, we have
$$|g'(x)| \leq K < 1$$
for all $x \in [\pi.3\pi/2]$. Thus, $|g(x) - g(y)| \leq K|x-y|$ for all $x, y \in [\pi, 3\pi/2]$ according to Remark 2.4.4.

c) From the Fixed Point Theorem, we have that the sequence $\{x_i\}_{i=0}^{\infty}$ defined by $x_0 \in [\pi, 3\pi/2]$ and $x_{i+1} = g(x_i)$ converges to p. Since
$$|x_n - p| \leq \frac{K^n}{1-K}|x_1 - x_0| = \frac{1}{\pi^2(1+\pi^2)^{n-1}}|x_1 - x_0| \ ,$$
we need to find n such that
$$\frac{1}{\pi^2(1+\pi^2)^{n-1}}|x_1 - x_0| < 10^{-5} \ ;$$
namely,
$$\frac{\ln\left(10^5|x_1 - x_0|/\pi^2\right)}{\ln(1+\pi^2)} < n - 1 \ .$$

If $x_0 = 4$, we have $x_1 = g(x_0) = \pi + \arctan(4)$. Thus, n must satisfy

$$n > \frac{\ln\left(10^5|\pi + \arctan(4) - 4|/\pi^2\right)}{\ln(1+\pi^2)} + 1 \approx 4.54695 \ .$$

Hence, $n = 5$ iterations is enough to reach the accuracy requested.

Question 2.23

a) If $p > 0$ is the fixed point of g, then

$$p = \frac{p}{2} + \frac{a}{2p} \ .$$

If we multiply both sides of the equality by 2 and subtract p from both sides, we get $p = a/p$. Thus $p^2 = a$ or $p = \sqrt{a}$ since we assume that $a > 0$.

b) Suppose that $x > 0$. Since $(x - \sqrt{a})^2 > 0$, we get $x^2 - 2\sqrt{a}x + a > 0$. Thus $x^2 + a > 2\sqrt{a}x$ and, after division by $2x$ on both side of the inequality, we have

$$g(x) = \frac{x}{2} + \frac{a}{2x} > \sqrt{a} \ .$$

Therefore, $x_i = g(x_0) \geq \sqrt{a}$ if $x_0 > 0$. We may assume that $x_0 \geq \sqrt{a}$.

Since

$$g'(x) = \frac{1}{2} - \frac{a}{2x^2} > 0$$

for $x > \sqrt{a}$, we have that g is strictly increasing on $]\sqrt{a}, \infty[$.

We now show that g satisfies all the hypothesis of the Fixed Point Theorem on $[\sqrt{a}, m]$ for $m > \sqrt{a}$ arbitrary.

1. We have $g : [\sqrt{a}, m] \to [\sqrt{a}, m]$. Since g is strictly increasing and $m > \sqrt{a}$, we have that
$$\sqrt{a} = g(\sqrt{a}) \leq g(x) \leq g(m) = \frac{m}{2} + \frac{a}{2m^2} \leq \frac{m}{2} + \frac{m}{2} = m$$
for all $x \in [\sqrt{a}, m]$.

2. For all $x \in [\sqrt{a}, m]$, we have that
$$|g'(x)| = \left|\frac{1}{2} - \frac{a}{2x^2}\right| = \frac{1}{2} - \frac{a}{2x^2} \leq \frac{1}{2}$$
because $x^2 \geq a$. Hence, $|f(x) - f(y) \leq K|x-y|$ for all $x, y \in [\sqrt{a}, m]$ with $K = 1/2 < 1$ according to Remark 2.4.4.

For any $m > \sqrt{a}$, we have from the Fixed Point Theorem that the function g has an unique fixed point in $[\sqrt{a}, m]$ and the sequence $\{x_n\}_{n=0}^{\infty}$ converge to this fixed point for any $x_0 \in [\sqrt{a}, m]$. Thus, since $\sqrt{a}$ is a unique fixed point for $x > 0$, the sequence $\{x_n\}_{n=0}^{\infty}$ converges to the fixed point $\sqrt{a}$ whatever $x_0 \geq \sqrt{a}$.

Since $x_1 = g(x_0) \geq \sqrt{a}$ for all $x_0 > 0$, the sequence $\{x_n\}_{n=0}^{\infty}$ converges to the fixed point $\sqrt{a}$ whatever $x_0 > 0$ because the sequences $\{x_n\}_{n=0}^{\infty}$ and $\{x_n\}_{n=1}^{\infty}$ have the same limit.

Question 2.24

a) The function f is continuous on $[a,b]$ because it is differentiable on $[a,b]$. Moreover, f has opposite signs at a and b because $f(a)f(b) < 0$. It follows from the Intermediate Value Theorem that f must be null at a point in the interval $[a,b]$.

Since $f'(x) > 0$ for all $x \in [a,b]$, we have that f is a strictly increasing function on $[a,b]$. Thus, $f(a) < 0 < f(b)$ and f cannot intersect the x axis more than once.

b) Since f' is continuous on the closed interval $[a,b]$, it reaches its absolute maximum and absolute minimum at some points of the interval $[a,b]$. Let x_M and x_m in $[a,b]$ be such that

$$M = f(x_M) = \max\{f'(x) : a \leq x \leq b\} \quad \text{and} \quad m = f(x_m) = \min\{f'(x) : a \leq x \leq b\} \ .$$

We have that $0 < m < M$ since $f'(x) > 0$ for all $x \in [a,b]$.

We claim that the function $F(x) = x + \lambda f(x)$ with $\lambda = -1/M$ satisfies the Fixed Point Theorem on $[a,b]$.

1. Since
$$F'(x) = 1 + \lambda f'(x) = 1 - \frac{f'(x)}{M} \geq 0$$
for all $x \in [a,b]$, we have that F is never decreasing on $[a,b]$. Thus
$$a < a - f(a)/M = F(a) \leq F(x) \leq F(b) = b - f(b)/M < b$$
for all $x \in [a,b]$. Hence, $F : [a,b] \to [a,b]$. Recall that $f(a) < 0 < f(b)$.
2. Since
$$0 \leq F'(x) = 1 + \lambda f'(x) = 1 - \frac{f'(x)}{M} \leq 1 - \frac{m}{M}$$
for all $x \in [a,b]$, we have that $|F'(x)| \leq K = 1 - m/M < 1$ for all $x \in [a,b]$. It follows that $|F(x) - F(y)| \leq K|x-y|$ for all $x \in [a,b]$ with $K < 1$ according to Remark 2.4.4.

The hypotheses of the Fixed Point Theorem are satisfied by F on $[a,b]$. Obviously, if $F(p) = p$, then $p + \lambda f(p) = p$. Hence, $f(p) = 0$.

Question 2.25

We prove that g satisfies the hypothesis of the Fixed Point Theorem on $[a,b]$.

1. Choose any $x \in [a,b]$. From the Mean Value Theorem, there exists ξ between x and m (and so in $[a,b]$) such that $g(x) - g(m) = g'(\xi)(x-m)$. Since $|g'(\xi)| < 1$, we have
$$|g(x) - m| = |g(x) - g(m)| = |g'(\xi)(x-m)| = |g'(\xi)|\,|x-m| < |x-m|$$
for all $x \in [a,b]$. From $|x-m| \leq (b-a)/2$, we get $|g(x) - m| \leq (b-a)/2$ for all $x \in [a,b]$. This proves that $g(x) \in [a,b]$ for all $x \in [a,b]$.

2. Since $|g'|$ is a continuous function on the closed set $[a,b]$, $|g'|$ reaches its absolute maximum at a point $\nu \in [a,b]$. Hence $K = |g'(\nu)| < 1$ will satisfies $|g'(x)| \leq K < 1$ for all $x \in [a,b]$. Therefore, $|g(x) - g(y)| \leq K|x-y|$ for all $x, y \in [a,b]$ with $K < 1$ according to Remark 2.4.4.

Question 2.26
It is clear that if $\{x_n\}_{n=0}^{\infty}$ is a sequence defined by $x_{n+1} = g(x_n)$ for $n \geq 0$ such that $x_N = p$ for some N, then $x_n = p$ for all $n \geq N$ because $g(p) = p$. Thus, $\{x_n\}_{n=0}^{\infty}$ converges to p in a finite number of iterations.

Are there other sequences $\{x_n\}_{n=0}^{\infty}$ that converge to p? To show that there are no other sequence converging to p requires a little bit of work.

Choose K between 1 and $|g'(p)|$. Since $|g'|$ is continuous and $|g'(p)| > K$, there exists $\delta > 0$ such that $1 < K \leq |g'(x)|$ for all $x \in [p-\delta, p+\delta]$.

Since g is a continuous function and $|g'(x)| > 1$ for $x \in [p-\delta, p+\delta]$, then p is the unique fixed point of g in $[p-\delta, p+\delta]$

Given any $x \in [p-\delta, p[\cup]p, p+\delta]$, it follows from the Mean Value Theorem that there exists ξ between p and x, and so in $[p-\delta, p+\delta]$, such that

$$|g(x) - p| = |g(x) - g(p)| = |g'(\xi)(x-p)| = |g'(\xi)|\,|x-p| \geq K|x-p| > |x-p| \ .$$

Suppose that $\{x_n\}_{n=0}^{\infty}$ is a sequence defined by $x_{n+1} = g(x_n)$ for $n \geq 0$ such that $x_n \neq p$ for all n. Suppose also that this sequence also converges to the fixed point p of g. By definition of convergence, there exists $N > 0$ such that $|x_n - p| < \delta$ for all $n > N$. However, $x_n \in [p-\delta, p+\delta]$ for all $n > N$ implies that

$$|x_{n+1} - p| = |g(x_n) - p| > |x_n - p|$$

for all $n > N$ as we have shown above. It follows by induction that $|x_n - p| \geq |x_{N+1} - p| > 0$ for $n > N$. The distance between x_n and p does not go to zero. This is a contradiction that $\{x_n\}_{n=0}^{\infty}$ is a sequence converging to the fixed point p.

Question 2.27
We already have one of the two hypotheses required by the Fixed Point Theorem; namely, $|g'(x)| \leq \lambda < 1$ for all $x \in [x_0-\rho, x_0+\rho]$. It is left to show that $g : [x_0-\rho, x_0+\rho] \to [x_0-\rho, x_0+\rho]$.

Choose $x \in [x_0-\rho, x_0+\rho]$. Then,

$$\begin{aligned} |g(x) - x_0| &= |g(x) - g(x_0) + g(x_0) - x_0| \leq |g(x) - g(x_0)| + |g(x_0) - x_0| \\ &= |g'(\mu)(x-x_0)| + (1-\lambda)\rho \ , \end{aligned}$$

where we have used the Mean Value Theorem to find μ between x and x_0 such that $g(x) - g(x_0) = g'(\mu)(x-x_0)$. We have also used the definition of ρ. Hence, since $|g'(x)| \leq \lambda < 1$ for all $x \in [x_0-\rho, x_0+\rho]$, we have

$$|g(x) - x_0| \leq \lambda|x-x_0| + (1-\lambda)\rho \leq \lambda\rho + (1-\lambda)\rho = \rho$$

for all $x \in [x_0-\rho, x_0+\rho]$. Hence, $g(x) \in [x_0-\rho, x_0+\rho]$ for all $x \in [x_0-\rho, x_0+\rho]$.

Question 2.28
If $[a,b] = [0,1]$ and $f(x) = 0.5x + 1$, then $f'(x) = 0.5 < 1$ for all $x \in [0,1]$ but there is no fixed point of f in $[a,b]$.

A contraction satisfies all hypothesis of the Fixed Point Theorem but one. The condition $f : [a,b] \to [a,b]$ is not required for a contraction. The function F of the previous paragraph satisfies $f : [0,1] \to [1,1.5]$. The interval $[0,1]$ has been contracted but not mapped into itself.

$f(x) = 0.5x + 1$ for $x \in [1,3]$ does satisfy all the hypotheses of the Fixed Point Theorem (verify this). The fixed point is $2 \in [1,3]$.

Question 2.29
$\Leftarrow$) From Taylor's Theorem, Theorem 2.1.6, we have that $f(x) = p_{m-1}(x) + r_{m-1}(x)$, where

$$p_{m-1}(x) = \sum_{j=0}^{m-1} \frac{1}{j!} f^{(j)}(p)\,(x-p)^j \quad \text{and} \quad r_{m-1}(x) = \frac{1}{m!} f^{(m)}(\xi)\,(x-p)^m$$

for $\xi(x)$ in the interval with endpoints x and p. Since, $f^{(j)}(p) = 0$ for $0 \le j < m$, we have that $p_{m-1}(x) = 0$ for all x. Thus $f(x) = r_{m-1}(x) = q(x)\,(x-p)^m$ with $q(x) = \dfrac{1}{m!} f^{(m)}(\xi(x))$.

Since $\xi(x)$ is between x and p, we have that $\lim_{x\to p} \xi(x) = p$. Therefore, since $f^{(m)}$ is continuous, we have

$$\lim_{x\to p} q(x) = \frac{1}{m!} f^{(m)}(p) \neq 0 \ .$$

Note that q is a continuous function on $\mathbb{R} \setminus \{p\}$ because $q(x) = \dfrac{f(x)}{(x-p)^m}$ for $x \neq p$, where f and $(x-p)^m$ are continuous functions of x. The function q is also continuous at p because $\lim_{x\to p} \dfrac{f(x)}{(x-p)^m} = \dfrac{1}{m!} f^{(m)}(p) = q(p)$ according to l'Hospital Rule.

$\Rightarrow$) From $f(x) = (x-p)^m q(x)$, we have by induction that

$$f^{(k)}(x) = \sum_{j=0}^{k} \binom{k}{j} C_j (x-p)^{m-j} q^{(k-j)}(x) \tag{16.3}$$

for $0 \le k \le m$, where

$$C_j = \begin{cases} 1 & \text{if} \quad j = 0 \\ m(m-1)\ldots(m-j+1) & \text{if} \quad j > 0 \end{cases}$$

Hence, $f^{(k)}(p) = 0$ for $0 \le k < m$ because each term in (16.3) has a factor $(x-p)$. For $k = m$, we get

$$f^{(m)}(p) = \binom{m}{m} C_m q(p) = m!\, q(p) \neq 0 \ .$$

Question 2.30
According to Theorem 2.7.2, to get a convergence of order three, we need $F(r) = r$, $F'(r) = F''(r) = 0$ and $F'''(r) \neq 0$.

Since $f(r) = 0$, we get $F(r) = r - f(r)f'(r) = r$.

From $F'(r) = 0$, we get

$$0 = 1 - (f'(r))^2 - f(r)f''(r) = 1 - (f'(r))^2$$

because $f(r) = 0$. Hence $f'(r) = \pm 1$. From $F''(r) = 0$, we get

$$0 = -3f'(r)f''(r) - f(r)f'''(r) = -3f'(r)f''(r)$$

because $f(r) = 0$. Since $f'(r) \neq 0$, we get $f''(r) = 0$. From $F'''(r) \neq 0$, we get

$$0 \neq -3(f''(r))^2 - 4f'(r)f'''(r) - f(r)f^{(4)}(r) = -4f'(r)f'''(r)$$

because $f(r) = f''(r) = 0$. Since $f'(r) \neq 0$, we get $f'''(r) \neq 0$.

The conditions on f are $f(r) = f''(r) = 0$, $|f'(r)| = 1$ and $f'''(r) \neq 0$.

Question 2.31

To get a convergence of order exactly three, we need $F(r) = r$, $F'(r) = F''(r) = 0$ and $F'''(r) \neq 0$.

Since $f(r) = 0$, we get $F(r) = r + f(r)g(r) = r$.

From $F'(r) = 0$, we get

$$0 = 1 + f'(r)g(r) + f(r)g'(r) = 1 + f'(r)g(r)$$

because $f(r) = 0$. Hence $g(r) = -\dfrac{1}{f'(r)}$. From $F''(r) = 0$, we get

$$0 = f''(r)g(r) + 2f'(r)g'(r) + f(r)g''(r) = -\frac{f''(r)}{f'(r)} + 2f'(r)g'(r)$$

because $f(r) = 0$ and $g(r) = -\dfrac{1}{f'(r)}$. Hence $g'(r) = \dfrac{f''(r)}{2(f'(r))^2}$. From $F'''(r) \neq 0$, we get

$$\begin{aligned} 0 &\neq f'''(r)g(r) + 3f''(r)g'(r) + 3f'(r)g''(r) + f(r)g'''(r) \\ &= -\frac{f'''(x)}{f'(r)} + \frac{3(f''(r))^2}{2(f'(r))^2} + 3f'(r)g''(r) \end{aligned}$$

because $f(r) = 0$, $g(r) = -\dfrac{1}{f'(r)}$ and $g'(r) = \dfrac{f''(r)}{2(f'(r))^2}$. Hence $g''(r) \neq \dfrac{f'''(x)}{3(f'(r))^2} - \dfrac{(f''(r))^2}{2(f'(r))^3}$.

The conditions on g are $g(r) = -\dfrac{1}{f'(r)}$, $g'(r) = \dfrac{f''(r)}{2(f'(r))^2}$ and $g''(r) \neq \dfrac{f'''(x)}{3(f'(r))^2} - \dfrac{(f''(r))^2}{2(f'(r))^3}$.

Question 2.32

We have to find for which sequence $\{x_n\}$ above the following statement is true.

$$\lim_{n\to\infty} \frac{|x_{n+1}|}{|x_n|^2} = \lambda \neq 0 \text{ or } \infty \; .$$

Note that all sequences converge to 0. Therefore, the error e_n is $e_n = |x_n - 0| = |x_n|$ for all n.

We have:

a)
$$\lim_{n\to\infty} \frac{1/(n+1)^2}{(1/n^2)^2} = \lim_{n\to\infty} \frac{n^4}{(n+1)^2} = \lim_{n\to\infty} \frac{n^4}{n^2+2n+1} = \lim_{n\to\infty} \frac{n^2}{1+2/n+1/n^2} = \infty \ .$$

b)
$$\lim_{n\to\infty} \frac{1/2^{2^{(n+1)}}}{\left(1/2^{2^n}\right)^2} = \lim_{n\to\infty} \frac{2^{2^{(n+1)}}}{2^{2^{(n+1)}}} = 1 \ .$$

c)
$$\lim_{n\to\infty} \frac{1/\sqrt{n+1}}{(1/\sqrt{n})^2} = \lim_{n\to\infty} \frac{n}{\sqrt{n+1}} = \infty$$

because
$$\frac{n}{\sqrt{n+1}} \geq \frac{n}{\sqrt{2n}} = \frac{\sqrt{n}}{\sqrt{2}} \to \infty$$

as $n \to \infty$.

d)
$$\lim_{n\to\infty} \frac{1/(e^{n+1})}{(1/e^n)^2} = \lim_{n\to\infty} \frac{e^{2n}}{e^{n+1}} = \lim_{n\to\infty} e^{n-1} = \infty \ .$$

e) We have
$$\lim_{n\to\infty} \frac{1/(n+1)^{n+1}}{(1/n^n)^2} = \lim_{n\to\infty} \frac{n^{2n}}{(n+1)^{n+1}} = \infty$$

because
$$\frac{n^{2n}}{(n+1)^{n+1}} \geq \frac{n^{2n}}{(2n)^{n+1}} = \frac{n^{n-1}}{2^{n+1}} = \frac{1}{2^2}\left(\frac{n}{2}\right)^{n-1} \to \infty$$

as $n \to \infty$.

Only the sequence in (b) converges quadratically.

Question 2.33

a) Let $e_n = 10^{-k^n} - 0$. We have
$$\frac{|e_{n+1}|}{|e_n|^\alpha} = \frac{10^{-k^{n+1}}}{\left(10^{-k^n}\right)^\alpha} = 10^{-k^{n+1}+\alpha k^n} = 10^{(\alpha-k)k^n} \ .$$

Hence,
$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \begin{cases} 1 & \text{if} \quad \alpha = k \\ +\infty & \text{if} \quad \alpha > k \\ 0 & \text{if} \quad \alpha < k \end{cases}$$

The order of convergence is $\alpha = k$.

b) Let $e_n = 10^{-n^k} - 0$. We have
$$\frac{|e_{n+1}|}{|e_n|^\alpha} = \frac{10^{-(n+1)^k}}{\left(10^{-n^k}\right)^\alpha} = 10^{-(n+1)^k+\alpha n^k} = 10^{(\alpha-1)n^k - kn^{k-1} - \text{l.o.t.}} \ ,$$

where l.o.t. stands for the terms in n^j with $0 \le j < n-1$. Hence,

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \begin{cases} 0 & \text{if} \quad \alpha \le 1 \\ \infty & \text{if} \quad \alpha > 1 \end{cases}$$

There is nothing in between. Recall that according to the binomial theorem,

$$(n+1)^k = \sum_{i=0}^{k} \binom{k}{i} n^i = n^k + k\,n^{k-1} + \ldots + 1 \ , \ \text{where} \quad \binom{k}{i} = \frac{k!}{(k-i)!\,i!} \ .$$

Question 2.35
All the results of the computations below will be displayed using 15-digit rounding accuracy to compare with the exact solutions at the end.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = 1$ and a tolerance of 10^{-10}, we find $r_0 = 0.333333333308985$ as an approximation of a root of p.

The deflated polynomial q_1 such that $p(x) = (x - r_0)q_1(x)$ is
$q_1(x) = x^2 - 5.641592653691014x + 7.853981634573692$.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = 1$ and a tolerance of 10^{-10}, we find $r_1 = 2.500000000539526$ as an approximation of a root of q_1.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = r_1$ and a tolerance of 10^{-10}, we find $c_1 = 2.500000000539525$ as an approximation of a root of p.

The deflated polynomial q_2 such that $q_1(x) = (x - r_1)q_2(x)$ is
$q_2(x) = x - 3.141592653151490$.

we have that $r_2 = 3.141592653151490$ as an approximation of a root of q_2.

Finally, using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = r_2$ and a tolerance of 10^{-10}, we find $c_2 = 3.141592653151491$ as an approximation of a root of p.

If $c_0 = r_0$, the approximations of the roots of p are c_i for $i = 0$, 1 and 2. If we use 10-digit rounding accuracy for the c_i, we have that all the 10 digits of c_0 are correct, only the last digit of c_1 and c_2 is wrong.

Question 2.36
All the results of the computations below will be displayed using 15-digit rounding accuracy to compare with the exact solutions at the end.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = 1$ and a tolerance of 10^{-9}, we find $r_0 = -3.548232897979703$ as an approximation of a root of p.

The deflated polynomial q_1 such that $p(x) = (x - r_0)q_1(x)$ is
$q_1(x) = x^3 - 5.548232897979703x^2 + 7.686422494264843x - 11.273217161921718$.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = 1$ and a tolerance of 10^{-9}, we find $r_1 = 4.381113440995944$ as an approximation of a root of q_1.

Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = r_1$ and a tolerance of 10^{-9}, we find $c_1 = 4.381113440995943$ as an approximation of a root of p.

The deflated polynomial q_2 such that $q_1(x) = (x - r_1)q_2(x)$ is
$q_2(x) = x^2 - 1.167119456983760x + 2.573139754025407$.

The polynomial q_2 has two complex roots that can be found with the formula to find the roots of a quadratic polynomial. They are $r_2 = 0.583559728491880 + 1.494188006011255\,i$ and $r_3 = 0.583559728491880 - 1.494188006011255\,i$. Using Newton's Method with Horner's Algorithm, Code 2.9.3, with $x_0 = r_2$ and a tolerance of 10^{-9}, we find $c_2 = 0.583559728491880 + 1.494188006011255\,i$ as an approximation of a root of p. Similarly, with $x_0 = r_3$, we find $c_3 = 0.583559728491880 - 1.494188006011255\,i$ as an approximation of a root of p as expected since complex roots of a polynomial with real coefficients come in pairs of complex conjugate values.

If $c_0 = r_0$, the approximations of the roots of p are c_i for $0 \leq i \leq 3$. If we use 9-digit rounding accuracy for the c_i, we have that all the 9 digits are right.

Chapter 3 : Iterative Methods for Systems of Linear Equations

Question 3.1
We have four conditions to verify.

1. We obviously have that $\|\mathbf{x}\| = \sum_{i=1}^{n} 2^{-i}|x_i| \geq 0$ for all $\mathbf{x}$. The sum of non-negative terms is non-negative.

2. If $\|\mathbf{x}\| = 0$ then $\sum_{i=1}^{n} 2^{-i}|x_i| = 0$. Since a sum of non-negative terms is null if and only if each term is null, $2^{-i}|x_i| = 0$ for all i and therefore $x_i = 0$ for all i.

3. For $\lambda \in \mathbb{R}$, we have
$$\|\lambda\mathbf{x}\| = \left\| \begin{pmatrix} \lambda x_1 & \lambda x_2 & \ldots & \lambda x_n \end{pmatrix}^\top \right\| = \sum_{i=1}^{n} 2^{-i}|\lambda x_i| = \sum_{i=1}^{n} 2^{-i}|\lambda|\,|x_i| = |\lambda| \sum_{i=1}^{n} 2^{-i}|x_i| = |\lambda|\,\|\mathbf{x}\| \ .$$

4. For any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, we have
$$\begin{aligned}\|\mathbf{x}+\mathbf{y}\| &= \left\| \begin{pmatrix} x_1+y_1 & x_2+y_2 & \ldots & x_n+y_n \end{pmatrix}^\top \right\| = \sum_{i=1}^{n} 2^{-i}|x_i+y_i| \\ &\leq \sum_{i=1}^{n} 2^{-i}\left(|x_i|+|y_i|\right) = \sum_{i=1}^{n} 2^{-i}|x_i| + \sum_{i=1}^{n} 2^{-i}|y_i| = \|\mathbf{x}\| + \|\mathbf{y}\| \ .\end{aligned}$$

Question 3.2
Since $\mathbf{x} = \mathbf{x} - \mathbf{y} + \mathbf{y}$, we get $\|\mathbf{x}\| = \|\mathbf{x}-\mathbf{y}+\mathbf{y}\| \leq \|\mathbf{x}-\mathbf{y}\| + \|\mathbf{y}\|$ from the triangle inequality. Thus,
$$\|\mathbf{x}\| - \|\mathbf{y}\| \leq \|\mathbf{x}-\mathbf{y}\| \ . \tag{16.4}$$

Similarly, since $\mathbf{y} = \mathbf{y} - \mathbf{x} + \mathbf{x}$, we get $\|\mathbf{y}\| = \|\mathbf{y} - \mathbf{x} + \mathbf{x}\| \leq \|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x}\|$ from the triangle inequality. Thus,

$$\|\mathbf{x}\| - \|\mathbf{y}\| \geq -\|\mathbf{y} - \mathbf{x}\| \ . \tag{16.5}$$

We get (3.6.1) from (16.4) and (16.5).

Question 3.3

Since $\{\mathbf{x} : \|\mathbf{x}\| = 1\} \subset \{\mathbf{x} : \mathbf{x} \neq \mathbf{0}\}$, we have

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| = \max_{\|\mathbf{x}\|=1} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \leq \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \ .$$

To prove the converse inequality, let $\mathbf{x}$ be any non-zero vector. Since $\mathbf{y} = \|\mathbf{x}\|^{-1}\mathbf{x}$ is a vector of norm 1, we have $\|A\mathbf{y}\| \leq \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| = \|A\|$. Thus

$$\|A\| \geq \|A\mathbf{y}\| = \left\| A\left(\frac{1}{\|\mathbf{x}\|}\mathbf{x}\right)\right\| = \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

for all $\mathbf{x} \neq \mathbf{0}$. Hence,

$$\|A\| \geq \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \ .$$

Question 3.4

Let $\mathbf{x}$ be a vector of ℓ^1-norm 1; namely, $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i| = 1$. Then,

$$\begin{aligned}
\|A\mathbf{x}\|_1 &= \sum_{i=1}^{n}\left|\sum_{j=1}^{n} a_{i,j}x_j\right| \leq \sum_{i=1}^{n}\left(\sum_{j=1}^{n}|a_{i,j}||x_j|\right) = \sum_{j=1}^{n}\left(\sum_{i=1}^{n}|a_{i,j}|\right)|x_j| \\
&\leq \sum_{j=1}^{n}\left(\max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\}\right)|x_j| = \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\}\sum_{j=1}^{n}|x_j| = \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\} \ .
\end{aligned}$$

Since this is true for any vector $\mathbf{x}$ such that $\|\mathbf{x}\|_1 = 1$, we have

$$\|A\|_1 = \max_{\|\mathbf{x}\|_1=1} \|A\mathbf{x}\|_1 \leq \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\} \ .$$

To prove that

$$\|A\|_1 \geq \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\} \ , \tag{16.6}$$

we prove that there exists $\mathbf{x} \in \mathbb{R}^n$ of ℓ^1-norm 1 such that $\|A\mathbf{x}\|_1 = \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\}$. Let k be the index of the column of A such that

$$\sum_{i=0}^{n}|a_{i,k}| = \max_{0\leq j\leq n}\left\{\sum_{i=0}^{n}|a_{i,j}|\right\}$$

and let $\mathbf{x}$ be the vector defined by

$$x_i = \begin{cases} 1 & \text{if} \quad i = k \\ 0 & \text{if} \quad i \neq k \end{cases}$$

Then $\|\mathbf{x}\|_1 = 1$ and

$$\|A\mathbf{x}\|_1 = \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{i,j} x_j| = \sum_{i=1}^{n} |a_{i,k}| = \max_{0 \leq j \leq n} \left\{ \sum_{i=0}^{n} |a_{i,j}| \right\}$$

Thus, (16.6) is true.

Question 3.5
We have that

$$\|A\|_\infty = \max\{|4| + |-3| + |2|, |-1| + |0| + |5|, |2| + |6| + |-2|\} = 10$$

Hence, it follows from Theorem 3.1.8 that $\|A\|_\infty$, the maximum value of $\|A\mathbf{x}\|_\infty$ for $\|\mathbf{x}\|_\infty = 1$, is 10. Moreover, in the proof of Theorem 3.1.8, we have seen that the maximum is reached for the vector $\mathbf{x}$ defined by

$$x_j = \begin{cases} 1 & \text{if } a_{k,j} \geq 0 \\ -1 & \text{if } a_{k,j} < 0 \end{cases}$$

where $k = 3$ because the third row of A gives the value of $\displaystyle \max_{0 \leq i \leq n} \left\{ \sum_{j=0}^{n} |a_{i,j}| \right\}$. Hence, $\mathbf{x} = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix}^\top$ gives $\|A\mathbf{x}\|_\infty = \|A\|_\infty$.

Question 3.6
Let $A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ -1 & 3 \end{pmatrix}$. Since $AB = \begin{pmatrix} 2 & -3 \\ 0 & 3 \end{pmatrix}$ and $BA = \begin{pmatrix} 1 & -1 \\ 2 & 4 \end{pmatrix}$, we get $\|AB\|_\infty = 5$ but $\|BA\|_\infty = 6$. So, it is not true that $\|AB\|_\infty = \|BA\|_\infty$ for all matrices A and B. There is nothing special about the ℓ^∞-norm.

Question 3.7
Given $\mathbf{x} \neq \mathbf{0}$, let $\mathbf{y} = \|\mathbf{x}\|^{-1}\mathbf{x}$. By definition of the norm of A, we have that $\|A\mathbf{y}\| \leq \|A\|$ since $\|\mathbf{y}\| = 1$. Thus,

$$\|\mathbf{x}\|^{-1} \|A\mathbf{x}\| = \left\| A\left(\|\mathbf{x}\|^{-1}\mathbf{x}\right)\right\| = \|A\mathbf{y}\| \leq \|A\| \Rightarrow \|A\mathbf{x}\| \leq \|A\| \, \|\mathbf{x}\| \ .$$

$\|A\mathbf{x}\| \leq \|A\| \, \|\mathbf{x}\|$ is obviously true for $\mathbf{x} = \mathbf{0}$.

Suppose that there exists $C < \|A\|$ such that $\|A\mathbf{x}\| \leq C \|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{R}^n$. Since $\|A\mathbf{x}\| \leq C$ for all $\mathbf{x} \in \mathbb{R}^n$ such that $\|\mathbf{x}\| = 1$, we get that $\displaystyle \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| \leq C < \|A\|$. This is a contradiction of the definition of $\|A\|$.

Question 3.8
a) We first interchange the second and third row of the system of linear equations. This will give us a linear equation $A\mathbf{x} = \mathbf{b}$, where A is strictly diagonally dominant. We have

$$3x_1 - x_2 + x_3 = 1$$

$$x_1 + 3x_2 - x_3 = 1$$
$$2x_1 + x_2 - 4x_3 = 0$$

This is equivalent to $A\mathbf{x} = \mathbf{b}$, where $A = \begin{pmatrix} 3 & -1 & 1 \\ 1 & 3 & -1 \\ 2 & 1 & -4 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$. Since A is strictly diagonally dominant, the Gauss-Seigel Iterative Method will converge.

b) The Gauss-Seidel Iterative Method is given by the iterative system

$$x_{n+1,1} = \frac{x_{n,2} - x_{n,3} + 1}{3}$$
$$x_{n+1,2} = \frac{-x_{n+1,1} + x_{n,3} + 1}{3}$$
$$x_{n+1,3} = \frac{2x_{n,1} + x_{n,2}}{4}$$

for $n = 0, 1, 2, \ldots$ If we start with $\mathbf{x}_0 = \mathbf{0}$, the first time that we have $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < 10^{-5}$ is for $n = 10$. We get $\mathbf{x}_{10} \approx \begin{pmatrix} 0.35000 \\ 0.30000 \\ 0.25000 \end{pmatrix}$, where the values have been rounded to 5 significant digits. This is in fact the exact answer.

Question 3.9
Let

$$A = \begin{pmatrix} 1 & 0 & 0 & \sqrt{2}/2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \sqrt{2}/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & -\sqrt{2}/2 & 0 & -1 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{2}/2 & 0 & 0 & \sqrt{3}/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sqrt{3}/2 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10000 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix}.$$

Note that we have reordered the equations to get non-null elements on the diagonal. However, it was impossible to get a diagonally dominant matrix. Nevertheless, if we consider the form $\mathbf{x}_{n+1} = T\mathbf{x}_n + \mathbf{c}$ for each of the respective iterative methods, we can use Theorem 3.2.8 to show that these three methods converge.

1. For the Jacobi iterative method, $T = D^{-1}(L + U)$ and the eigenvalues of T are 0 (with multiplicity 6) and $\pm 0.75983568565\ldots$ Hence, the spectral radius of T is $r_J = \rho(T) = 0.7598356856\ldots < 1$.

2. For the Gauss-Seidel iterative method, $T = (D - L)^{-1}U$ and the eigenvalues of T are 0 (with multiplicity 7) and $0.5773502691\ldots$ Hence, the spectral radius of T is $r_{GS} = \rho(T) = 0.5773502691\ldots < 1$.

3. For the relaxation iterative method with $\omega = 1.2$, $T = (D - \omega L)^{-1}((1-\omega)D + \omega U)$ and the eignevalues of T are -0.2 (with multiplicity 6), $0.2964580434\ldots$ and $0.134926344\ldots$ The spectral radius of T is $r_R = \rho(T) = 0.2964580434\ldots < 1$.

Since $r_R < r_{GS} < r_J$, we expect that the relaxation method will be the method that converges the fastest to the equilibrium, followed by the Gauss-Seidel iterative method. The slowest method should be the Jacobi Iterative Method.

We start all three iterations with the vector $\mathbf{x}_0 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)^\top$.

a) The solution (rounded to seven significant digits) found with the Jacobi Iterative Method is

$$\mathbf{F} \approx \begin{pmatrix} 10000 & -13660.25 & 13660.25 & 19318.52 & -23660.25 & 0 & 27320.51 & -23660.25 \end{pmatrix}^\top$$

after 64 iterations.

b) The solution (rounded to seven significant digits) found with the Gauss-Seidel Iterative Method is

$$\mathbf{F} \approx \begin{pmatrix} 10000 & -13660.25 & 13660.25 & 19318.52 & -23660.25 & 0 & 27320.51 & -23660.25 \end{pmatrix}^\top$$

after 33 iterations.

c) For the relaxation iterative method, we use $\omega = 1.2$ which is between 0 and 2 as required. The solution (rounded to seven significant digits) found with the relaxation iterative method is

$$\mathbf{F} \approx \begin{pmatrix} 10000 & -13660.25 & 13660.25 & 19318.52 & -23660.25 & 0 & 27320.50 & -23660.25 \end{pmatrix}^\top$$

after 17 iterations.

Question 3.10

a) Since A is strictly diagonally dominant, it follows from Theorem 3.2.12 that both iterative methods converge.

For (b), (c) and (d), we use $\mathbf{x}_0 = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^\top$.

b) The iterative system associated to the Jacobi iterative method is

$$\begin{aligned}
x_{n+1,1} &= \frac{1}{4}\left(5 - x_{n,2} + x_{n,3} - x_{n,4}\right)\\
x_{n+1,2} &= \frac{1}{5}\left(2 - x_{n,1} - x_{n,3} + x_{n,4}\right)\\
x_{n+1,3} &= \frac{1}{6}\left(-14 - 2x_{n,1} + x_{n,2} - 2x_{n,4}\right)\\
x_{n+1,4} &= \frac{1}{5}\left(25 + x_{n,1} - x_{n,2} + 2x_{n,3}\right)
\end{aligned}$$

After 25 iterations, the Jacobi Iterative Method yields the following approximation of the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}.

$$\mathbf{x} \approx \begin{pmatrix} -0.755414 & 1.796178 & -2.892990 & 3.332482 \end{pmatrix}^\top .$$

c) The iterative system associated to the Gauss-Seidel iterative method is

$$\begin{aligned}
x_{n+1,1} &= \frac{1}{4}\left(5 - x_{n,2} + x_{n,3} - x_{n,4}\right) \\
x_{n+1,2} &= \frac{1}{5}\left(2 - x_{n+1,1} - x_{n,3} + x_{n,4}\right) \\
x_{n+1,3} &= \frac{1}{6}\left(-14 - 2x_{n+1,1} + x_{n+1,2} - 2x_{n,4}\right) \\
x_{n+1,4} &= \frac{1}{5}\left(25 + x_{n+1,1} - x_{n+1,2} + 2x_{n+1,3}\right)
\end{aligned}$$

After 10 iterations, the Gauss-Seidel iterative method yields the following approximation of the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}.

$$\mathbf{x} \approx \begin{pmatrix} -0.755414 & 1.796178 & -2.892994 & 3.332484 \end{pmatrix} .$$

d) The iterative system associated to the relaxation method is

$$\begin{aligned}
x_{n+1,1} &= x_{n,1} + \frac{\omega}{4}\left(5 - 4x_{n,1} - x_{n,2} + x_{n,3} - x_{n,4}\right) \\
x_{n+1,2} &= x_{n,2} + \frac{\omega}{5}\left(2 - x_{n+1,1} - 5x_{n,2} - x_{n,3} + x_{n,4}\right) \\
x_{n+1,3} &= x_{n,3} + \frac{\omega}{6}\left(-14 - 2x_{n+1,1} + x_{n+1,2} - 6x_{n,3} - 2x_{n,4}\right) \\
x_{n+1,4} &= x_{n,1} + \frac{\omega}{5}\left(25 + x_{n+1,1} - x_{n+1,2} + 2x_{n+1,3} - 5x_{n,4}\right)
\end{aligned}$$

To prove that this iterative method converge for $0 < \omega < 2$, we use the form $\mathbf{x}_{n+1} = T\mathbf{x}_n + \mathbf{c}$, where $T = (D - \omega L)^{-1}((1-\omega)D + \omega U)$ and $\mathbf{c} = \omega(D - \omega L)^{-1}\mathbf{b}$ with

$$U = \begin{pmatrix} 0 & -1 & 1 & -1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad , \quad L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 2 & 0 \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix} .$$

If $\omega = 9.6$, we have

$$T = \begin{pmatrix} 0.04 & -0.24 & 0.24 & -0.24 \\ -0.00768 & 0.08608 & -0.23808 & 0.23808 \\ -0.0140288 & 0.0905728 & -0.0748928 & -0.2051072 \\ 0.0037675008 & -0.0278274048 & 0.0630325248 & -0.1305525248 \end{pmatrix} .$$

The eigenvalues of T (rounded to five significant digits) are $-0.022692 \pm 0.16947\,i$, -0.031173 and -0.0028092. Since they are all smaller than 1 in absolute value, it follows from Theorem 3.2.8 that the relaxation iterative method with $\omega = 9.6$ converges because $\rho(T) < 1$. The general case for $0 < \omega < 2$ is left to the reader interested in very long algebraic computations.

The value of ω for which the convergence of the Relaxation Method seems to be the fastest is ω between 0.96 and 0.97 approximately. With $\omega = 9.6$, only 9 iterations are necessary to get the following approximation of the solution of $A\mathbf{x} = \mathbf{b}$ with an accuracy of 10^{-5}.

$$\mathbf{x} \approx \begin{pmatrix} -0.755412 & 1.796177 & -2.892995 & 3.332484 \end{pmatrix} .$$

Question 3.11

Suppose that $\mathbf{p}$ is a solution of $\mathbf{x} = T\mathbf{x} - \mathbf{c}$. Since $\rho(T) \geq 1$, there is at least one eigenvalue, say λ, such that $|\lambda| \geq 1$. Let $\mathbf{u}$ be an eigenvector associated to λ, and let $\mathbf{x}_0 = \mathbf{u} + \mathbf{p}$. We prove by induction that

$$\mathbf{x}_k - \mathbf{p} = \lambda^k \mathbf{u} \tag{16.7}$$

for $k \geq 0$. It is obvious that (16.7) is true for $k = 0$. Let's suppose that (16.7) is true for k, then

$$\mathbf{x}_{k+1} - \mathbf{p} = (T\mathbf{x}_k + \mathbf{c}) - (T\mathbf{p} + \mathbf{c}) = T(\mathbf{x}_k - \mathbf{p}) = T(\lambda^k \mathbf{u}) = \lambda^k T\mathbf{u} = \lambda^{k+1}\mathbf{u} .$$

Thus (16.7) is true for k replaced by $k+1$.

However, $\{\lambda^k \mathbf{u}\}_{k=0}^{\infty}$ does not converge to $\mathbf{0}$. If $|\lambda| > 1$, we have that $\|\lambda^k \mathbf{u}\| = |\lambda|^k \|\mathbf{u}\| \to \infty$ as $k \to \infty$. If $|\lambda| = 1$, we have that $\|\lambda^k \mathbf{u}\| = \|\mathbf{u}\| > 0$ for all k. Again, $\|\lambda^k \mathbf{u}\| \not\to 0$ as $k \to \infty$. If follows from (16.7) that $\{\mathbf{x}_k\}_{k=0}^{\infty}$ does not converge to $\mathbf{p}$ for $\mathbf{x}_0 = \mathbf{u} + \mathbf{p}$. That such a vector $\mathbf{x}_0$ exists was predicted by Theorem 3.2.8.

It is interesting to note that if there is no $\mathbf{p}$ such that $\mathbf{p} = T\mathbf{p} - \mathbf{c}$, then $\{\mathbf{x}_k\}_{k=0}^{\infty}$ defined by $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c}$ does not converge at all for all choices of $\mathbf{x}_0$. If $\{\mathbf{x}_k\}_{k=0}^{\infty}$ was to converge to a vector $\mathbf{q}$, then we would get

$$\mathbf{q} = \lim_{k\to\infty} \mathbf{x}_{k+1} = \lim_{k\to\infty} \left(T\mathbf{x}_k + \mathbf{c}\right) = T\Big(\lim_{k\to\infty} \mathbf{x}_k\Big) + \mathbf{c} = T\mathbf{q} + \mathbf{c}$$

by continuity. This would be a contradiction of our assumption that there is no $\mathbf{p}$ such that $\mathbf{p} = T\mathbf{p} - \mathbf{c}$. Thus, $\{\mathbf{x}_k\}_{k=0}^{\infty}$ doesn't converge for all $\mathbf{x}_0$.

Question 3.12

a) Jacobi Iterative Method is of the form $\mathbf{x}_{k+1} = T\mathbf{x}_\mathbf{k} + \mathbf{c}$, where $\mathbf{c} = D^{-1}\mathbf{b}$ and $T = D^{-1}(L + U) = D^{-1}U$ is a strictly upper-triangular matrix (only 0 on the diagonal). Since the only eigenvalues of T is 0, the spectral radius of T is $\rho(T) = 0 < 1$. Hence, the iterative method converges according to Theorem 3.2.8.

b) For this particular choice of A, we have $D = \mathrm{Id}_3$, $L = 0$ and $U = \begin{pmatrix} 0 & -3 & -5 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{pmatrix}$. Thus $T = U$. With $\mathbf{x}_0 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\top$, we get

$$\mathbf{x}_1 = \begin{pmatrix} 0 & -3 & -5 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} , \quad \mathbf{x}_2 = \begin{pmatrix} 0 & -3 & -5 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -7 \\ -4 \\ 1 \end{pmatrix} ,$$

$$\mathbf{x}_3 = \begin{pmatrix} 0 & -3 & -5 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -7 \\ -4 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ -4 \\ 1 \end{pmatrix} , \quad \mathbf{x}_4 = \begin{pmatrix} 0 & -3 & -5 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 8 \\ -4 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ -4 \\ 1 \end{pmatrix} .$$

The solution is $\begin{pmatrix} 8 & -4 & 1 \end{pmatrix}^\top$.

c) We have seen, in the context of the proof of Theorem 3.2.8, that

$$\mathbf{x}_k = T^k \mathbf{x}_0 + \sum_{i=0}^{k-1} T^i \mathbf{c}$$

for $k > 0$. However, it is easy to show that any $n\times n$ strictly upper-triangular matrix satisfies $T^n = 0$. Thus
$$\mathbf{x}_k = \sum_{i=0}^{n-1} T^i\mathbf{c}$$
for $k \geq n$. Thus, the Jacobi iterative method converges in n iterations to its limit.

Question 3.13

Since A is upper-triangular, the Gauss-Seidel iterative method is of the form $\mathbf{x}_{k+1} = T\mathbf{x}_\mathbf{k} + \mathbf{c}$, where $\mathbf{c} = (D-L)^{-1}\mathbf{b} = D^{-1}\mathbf{b}$ and $T = (D-L)^{-1}U = D^{-1}U$ is a strictly upper-triangular matrix (only 0 on the diagonal). Since the only eigenvalues of T is 0, the spectral radius of T is $\rho(T) = 0 < 1$. Hence, the Gauss-Seidel iterative method converges according to Theorem 3.2.8.

We have seen, in the context of the proof of Theorem 3.2.8, that
$$\mathbf{x}_k = T^k\mathbf{x}_0 + \sum_{i=0}^{k-1} T^i\mathbf{c}$$
for $k > 0$. However, it is easy to show that any $n\times n$ strictly upper-triangular matrix satisfies $T^n = 0$. Thus
$$\mathbf{x}_k = \sum_{i=0}^{n-1} T^i\mathbf{c}$$
for $k \geq n$. Thus, the Gauss-Seidel iterative method converges in n iterations to its limit.

Question 3.14

a) The relaxation method to approximate the solution $\mathbf{p}$ of $A\mathbf{x} = \mathbf{b}$ is of the form $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{c}$, where $T = (D-\omega L)^{-1}((1-\omega)D + \omega U)$ and $\mathbf{c} = \omega(D-\omega L)^{-1}\mathbf{b}$. Since
$$\begin{aligned} T &= (D-\omega L)^{-1}((1-\omega)D+\omega U) = \begin{pmatrix} 1 & 0 \\ -2\omega & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1-\omega & 0 \\ 0 & 3(1-\omega) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 2\omega/3 & 1/3 \end{pmatrix} \begin{pmatrix} 1-\omega & 0 \\ 0 & 3(1-\omega) \end{pmatrix} = \begin{pmatrix} 1-\omega & 0 \\ 2\omega(1-\omega)/3 & 1-\omega \end{pmatrix} \end{aligned}$$
$1-\omega$ is an eigenvalue of T of algebraic multiplicity two.

It follows from Theorem 3.2.8 that the sequence $\{\mathbf{x}_k\}_{k=0}^\infty$ generated by the relaxation method will converge for any $\mathbf{x}_0$ to the fixed point $\mathbf{p}$ of $F(\mathbf{x}) = T\mathbf{x} + \mathbf{c}$ (and so the solution of $A\mathbf{x} = \mathbf{b}$) if and only if $\rho(T)$, the spectral radius of T, is smaller than 1. Therefore, the relaxation method will converge to the fixed point $\mathbf{p}$ if and only if $\omega \in]0, 2[$.

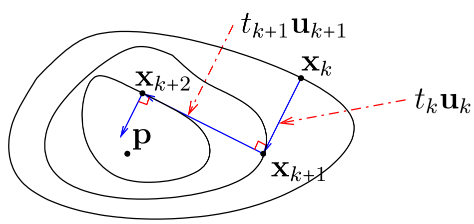
b) The optimal value of ω is the value for which $\rho(T)$ is the smallest. This happen for $\omega = 1$ when we get $\rho(T) = 0$. The relaxation method is then the Gauss-Seidel Iterative Method.

c) It is shown in the answer to Question 3.11 that the sequence $\{\mathbf{x}_k\}_{k=0}^\infty$ does not converge for all $\mathbf{x}_0$ if there is no solution to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$. Moreover, if there is a solution $\mathbf{p}$ to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$, then the sequence $\{\mathbf{x}_k\}_{k=0}^\infty$ does not converge to $\mathbf{p}$ if $\mathbf{x}_0 = \mathbf{p} + \mathbf{u}$ for $\mathbf{u}$ an eigenvector associated to the eigenvalue $1-\omega$.

Question 3.15
a) The gradient of g is
$$\nabla g(\mathbf{x}) = (A + A^\top)\mathbf{x} - 2\mathbf{b} \ .$$
Since A is symmetric,
$$\nabla g(\mathbf{x}_k) = 2A\mathbf{x}_k - 2\mathbf{b} = -2\mathbf{u}_k$$
by definition of $\mathbf{u}_k$ for this version of the steepest descent method.

b) Since $\mathbf{u}_k = \mathbf{b} - A\mathbf{x}_k$ and $\mathbf{x}_{k+1} = \mathbf{x} + t_k\mathbf{u}_k$, we get
$$t_k = \frac{\langle \mathbf{u}_k, \mathbf{b} - A\mathbf{x}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle} = \frac{\langle \mathbf{u}_k, \mathbf{u}_k\rangle}{\langle \mathbf{u}_k, A\mathbf{u}_k\rangle}$$
and
$$\langle \mathbf{u}_{k+1}, \mathbf{u}_k\rangle = \langle \mathbf{b} - A\mathbf{x}_{k+1}, \mathbf{u}_k\rangle = \langle \mathbf{b} - A\mathbf{x}_k - t_kA\mathbf{u}_k, \mathbf{u}_k\rangle = \langle \mathbf{u}_k, \mathbf{u}_k\rangle - t_k\,\langle A\mathbf{u}_k, \mathbf{u}_k\rangle = 0 \ .$$

c) The following figure contains some level curves of g and illustrates this version of the steepest descent method.



Question 3.17
Since A is strictly positive definite, We have that
$$\langle \mathbf{r}, \mathbf{e}\rangle = \left\langle \mathbf{b} - A\mathbf{x}, A^{-1}\mathbf{b} - \mathbf{x}\right\rangle = \left\langle A(A^{-1}\mathbf{b} - \mathbf{x}), A^{-1}\mathbf{b} - \mathbf{x}\right\rangle > 0$$
as long as $A^{-1}\mathbf{b} - \mathbf{x} \neq \mathbf{0}$; namely, as long as, $\mathbf{b} - A\mathbf{x} \neq \mathbf{0}$.

Chapter 4 : Algebraic Methods for Systems of Linear Equations

Question 4.4
From $\mathrm{Id} = A\,A^{-1}$, we get $1 = \|\,\mathrm{Id}\,\| = \|A\,A^{-1}\| \leq \|A\|\,\|A^{-1}\| = K(A)$.

Question 4.5
To system of equations $A\mathbf{x} = \mathbf{b}_p$ gives the equations
$$\begin{aligned} x_1 + 2x_2 &= 3.00001 \\ 1.00001x_1 + 2x_2 &= 3.00003 \end{aligned}$$

If we subtract 1.00001 times the first equation from the second equation, we get $-0.00002x_2 = -0.00001$. Hence $x_2 = 0.5$. If we substitute this value of x_2 in the first equation, we get $x_1 = 2.00001$.. The solution of the perturbed system is $\mathbf{x}_p = \begin{pmatrix} 2.00001 & 0.5 \end{pmatrix}^\top$.

The relative error is

$$\frac{\|\mathbf{x}_s - \mathbf{x}_p\|_\infty}{\|\mathbf{x}_s\|_\infty} = 1.00001 \ .$$

This is very large (an error of about 100%).

Since

$$A^{-1} = \begin{pmatrix} -10^5 & 10^5 \\ 50,000.5 & -50,000 \end{pmatrix} \ .$$

We find that the condition number is $K(A) = \|A\|_\infty \|A^{-1}\|_\infty = 3.00001 \times 2 \times 10^5 = 6.00002 \times 10^5$. The matrix A is ill-conditioned.

Question 4.6

The norm $\|A\|_\infty$ is given by the sum in absolute value of all the elements of the last row of A. Thus $\|A\|_\infty = n$.

The inverse of A is given by the matrix B defined by

$$b_{i,j} = \begin{cases} 0 & \text{if } j > i \\ 1 & \text{if } j = i \\ 2^{i-j-1} & \text{if } j < i \end{cases}$$

because $AB = \mathrm{Id}_n$ yields $b_{1,j} = \delta_{1,j}$ for $1 \le j \le n$, and $-\sum_{i=1}^{k-1} b_{i,j} + b_{k,j} = \delta_{k,j}$ for $1 \le j \le n$ and $2 \le k \le n$; namely, $b_{k,j} = \delta_{k,j} + \sum_{i=1}^{k-1} b_{i,j}$ for $1 \le j \le n$ and $2 \le k \le n$. In plain English, the first row of B is the first row of the identity matrix Id_n and the k^{th} row of B is the sum of the k^{th} row of Id_n with the previous $k-1$ rows of B for $2 \le k \le n$.

Again, the norm $\|B\|_\infty$ is given by the sum in absolute value of all the elements of the last row of B. Thus

$$\|B\|_\infty = 1 + \sum_{i=0}^{n-2} 2^i = 1 + \frac{1 - 2^{n-1}}{1-2} = 2^{n-1} \ .$$

The sum in the previous expression is given by the formula $\displaystyle\sum_{i=1}^{k} r^i = \frac{1 - r^{k+1}}{1-r}$ with $r = 2$ and $k = n-2$.

Hence, $K(A) = n 2^{n-1}$. For large matrices A (i.e. n large), the matrix is not well conditioned.

Question 4.7

We use the relation $\|\mathbf{q} - \mathbf{p}\|_1 \le K(A) \dfrac{\|\mathbf{b} - A\mathbf{q}\|_1}{\|A\|_1}$ to get $\|\mathbf{q} - \mathbf{p}\|_1 \dfrac{\|A\|_1}{\|\mathbf{b} - A\mathbf{q}\|_1} \le K(A)$.

Since $\|A\|_1 = \sum_{i=1}^{3} |a_{i,3}| = 4.21$, $\|\mathbf{q}-\mathbf{p}\|_1 = \|\begin{pmatrix} 0.03 & 0.02 & 0.07 \end{pmatrix}^\top\|_1 = 0.12$ and $\|\mathbf{b} - A\mathbf{q}\|_1 = \|\begin{pmatrix} 0.0007 & 0.002 & 0.35 \end{pmatrix}^\top\|_1 = 0.3527$, we get $K(A) \geq 0.12 \times 4.21/0.3527 \approx 1.43237879$. This is an indication that the system may be well conditioned. It does not prove that the system is well conditioned because the inequality is in the wrong direction.

The real condition number is about 169.3678. So, the matrix is ill conditioned.

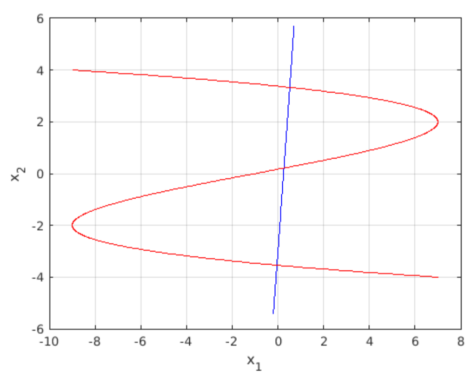
Chapter 5 : Iterative Methods for Systems of Nonlinear Equations

Question 5.2

a) We have

$$\begin{aligned} f(\mathbf{x}) = \mathbf{0} &\iff x_1^3 + 12x_1 - x_2 - 3 = 0 \quad \text{and} \quad 2x_1 + x_2^3 - 12x_2 + 2 = 0 \\ &\iff 12x_1 = x_2 - x_1^3 + 3 \quad \text{and} \quad 12x_2 = 2x_1 + x_2^3 + 2 \\ &\iff x_1 = \frac{x_2 - x_1^3 + 3}{12} \quad \text{and} \quad x_2 = \frac{2x_1 + x_2^3 + 2}{12} \iff \mathbf{x} = g(\mathbf{x}) \ . \end{aligned}$$

b) In the figure below, the level curve $f_1(\mathbf{x}) = 0$ is in blue and the level curve $f_2(\mathbf{x}) = 0$ is in red.



It follows from the figure above that there are at least three solutions of $f(\mathbf{x}) = \mathbf{0}$ corresponding to the points of intersection of the two level curves. We will focus on the solution in the set S given in (c).

c) We very the two hypotheses of the Fixed Point Theorem.

1. Since $0 < 1/6 = g_1(1,0) \leq g_1(x,y) \leq g_1(0,1) = 1/3 < 1$ and $0 < 1/6 = g_2(0,0) \leq g_2(x,y) \leq g_2(1,1) = 5/12 < 1$, we have that $g(S) \subset S$.

2. We have that

$$J_g(\mathbf{x}) = \begin{pmatrix} -x_1^2/4 & 1/12 \\ 1/6 & x_2^2/4 \end{pmatrix} \ .$$

Let

$$K = \|J_g(\mathbf{x}\|_\infty = \max_{0\le x_1,x_2\le 1}\{|-x^2/4| + 1/12, 1/6 + |y^2/4|\} = 5/12 \ .$$

We get from Remark 5.1.2 that $\|g(\mathbf{x}) - g(\mathbf{y})\|_\infty \le K\|\mathbf{x}-\mathbf{y}\|_\infty$ for all $\mathbf{x}$ and $\mathbf{y}$ in S with $K < 1$.

d) If we start with $\mathbf{x}_0 = \mathbf{0}$ and compute $\mathbf{x}_{n+1} = g(\mathbf{x}_n)$ for $n \ge 0$, we find that $\|\mathbf{x}_n - \mathbf{x}_{n-1}\|_\infty < 10^{-5}$ for the first time when $n = 6$. We get $\mathbf{x}_6 \approx \begin{pmatrix} 0.266078 & 0.211797\end{pmatrix}^\top$, where we have rounded the values to 6 significant digits.

e) We use the formula

$$\|\mathbf{x}_n - \mathbf{p}\| \le \frac{K^n}{1-K}\,\|\mathbf{x}_1 - \mathbf{x}_0\| < 10^{-5}$$

to determine the value of n. If $\mathbf{x}_0 = \mathbf{0}$, we get $\mathbf{x}_1 = \begin{pmatrix} 1/4 & 1/6\end{pmatrix}^\top$. Hence, with $K = 5/12$, we have

$$\begin{aligned}\frac{K^n}{1-K}\|\mathbf{x}_1 - \mathbf{x}_0\|_\infty < 10^{-5} &\Rightarrow \frac{(5/12)^n}{7/12}\left(\frac{1}{4}\right) < 10^{-5} \Rightarrow (5/12)^n < 10^{-5}\left(\frac{7}{3}\right)\\ &\Rightarrow n > \frac{-5\ln(10) + \ln(7/3)}{\ln(5/12)} \approx 12.18276 \ .\end{aligned}$$

So, $n = 13$ will be sufficient.

Question 5.3

We rewrite $f(\mathbf{x}) = \mathbf{0}$ as $g(\mathbf{x}) = \mathbf{x}$ with

$$g(\mathbf{x}) = \begin{pmatrix} (x_2+5)/4 \\ (1+\sqrt{x_1})^{1/3} - 1\end{pmatrix} \ .$$

Let $S = \{\mathbf{x} : 1 \le x_1 \le 2 \text{ and } 1/4 \le x_2 \le 3/4\}$. We have that $g : \mathbb{R}^2 \to \mathbb{R}^2$ satisfies the two hypotheses of the Fixed Point Theorem, Theorem 5.1.1.

1. Since $(x_2+5)/4 \le \big((3/4)+5\big)/4 = 23/16 < 2$, $(x_2+5)/4 \ge \big((1/4)+5\big)/4 = 21/16 > 1$, $(1+\sqrt{x_1})^{1/3} - 1 \le (1+\sqrt{2})^{1/3} - 1 = 0.3415\ldots < 3/4$ and $(1+\sqrt{x_1})^{1/3} - 1 \ge (1+\sqrt{1})^{1/3} - 1 = 0.2599\ldots > 1/4$, we have $g(S) \subset S$.

2. Instead of proving directly that there exists $0 < K < 1$ such that $\|g(\mathbf{x}) - g(\mathbf{y})\|_\infty \le K\|\mathbf{x} - \mathbf{y}\|_\infty$ for all $\mathbf{x}$ and $\mathbf{y}$ in S, we show that the Jacobian J_g of g satisfies $\max_{\mathbf{x}\in S}\|J_g(x)\|_\infty < 1$ and use Remark 5.1.2. Since

$$J_g(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial g_1}{\partial x_1}(\mathbf{x}) & \dfrac{\partial g_1}{\partial x_2}(\mathbf{x}) \\ \dfrac{\partial g_2}{\partial x_1}(\mathbf{x}) & \dfrac{\partial g_2}{\partial x_2}(\mathbf{x})\end{pmatrix} = \begin{pmatrix} 0 & \dfrac{1}{4} \\ \dfrac{1}{6\sqrt{x_1}\,(1+\sqrt{x_1})^{2/3}} & 0\end{pmatrix} ,$$

we get

$$\max_{\mathbf{x}\in S}\|J_g(\mathbf{x})\|_\infty = \max\left\{\frac{1}{4}, \frac{1}{6\,(2^{2/3})}\right\} = \frac{1}{4} < 1 \ .$$

Hence $\|g(\mathbf{x}) - g(\mathbf{y})\|_\infty \le K\|\mathbf{x}-\mathbf{y}\|_\infty$ for all $\mathbf{x}$ and $\mathbf{y}$ in S with $K = \max_{\mathbf{x}\in S}\|J_g(\mathbf{x})\|_\infty < 1$.

Starting with $\mathbf{x}_0 = \begin{pmatrix}1.5 & 0.5\end{pmatrix}^\top$, we compute $\mathbf{x}_{k+1} = g(\mathbf{x}_k)$ until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_\infty < 10^{-5}$. It takes 7 iterations to get the first approximation $\mathbf{x}_7 \approx \begin{pmatrix}1.32266994 & 0.29067777\end{pmatrix}^\top$ of the fixed point in S that satisfies the required accuracy.

Question 5.6
To use Newton's Method, we first need to compute the Jacobian of f.

$$J_f(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_1}{\partial x_2}(\mathbf{x}) & \dfrac{\partial f_1}{\partial x_3}(\mathbf{x}) \\ \dfrac{\partial f_2}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_2}{\partial x_2}(\mathbf{x}) & \dfrac{\partial f_2}{\partial x_3}(\mathbf{x}) \\ \dfrac{\partial f_3}{\partial x_1}(\mathbf{x}) & \dfrac{\partial f_3}{\partial x_2}(\mathbf{x}) & \dfrac{\partial f_3}{\partial x_3}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 3x_1^2 + 2x_1x_2 - x_3 & x_1^2 & -x_1 \\ e^{x_1} & e^{x_2} & -1 \\ -2x_3 & 2x_2 & -2x_1 \end{pmatrix}.$$

Starting with $\mathbf{x}_0 = \begin{pmatrix}1 & 1 & 1\end{pmatrix}^\top$, we compute

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J_f(\mathbf{x}_k))^{-1} f(\mathbf{x}_k)$$

until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < 10^{-6}$. It takes 12 iterations to get the first approximation $\mathbf{x}_{12} \approx \begin{pmatrix}-1.95629521 & -0.131795995 & 1.017901033\end{pmatrix}^\top$ of a solution of $f(\mathbf{x}) = \mathbf{0}$ that satisfies the required accuracy.

Chapter 6 : Polynomial Interpolation

Question 6.1
Let

$$q(x) = \sum_{j=0}^{n} \ell_j(x)$$

for all x. We first show that $q(x) = 1$ for all x. The polynomial $q(x) - 1$ is of degree at most n and has $n + 1$ distinct roots at x_0, x_1, ..., x_n because

$$\ell_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Since polynomial of degree n cannot have more than n roots, $q(x) - 1 = 0$ for all x.

Hence

$$\sum_{j=0}^{n} (f(x) - p(x_j))\,\ell_j(x) = f(x) \sum_{j=0}^{n} \ell_j(x) - \sum_{j=0}^{n} p(x_j)\,\ell_j(x) = f(x) - p(x)$$

because $p(x) = \sum_{j=0}^{n} p(x_j)\,\ell_j(x)$.

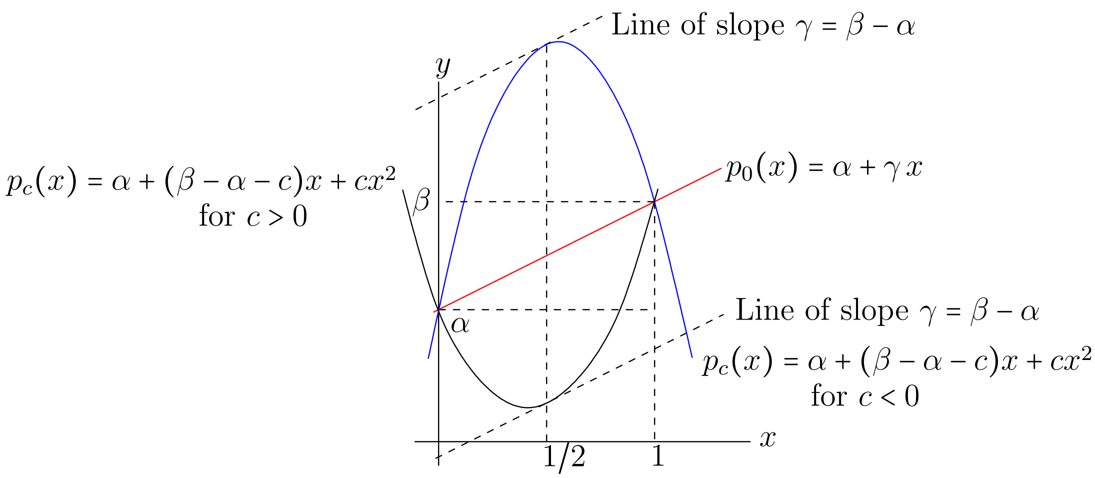
Question 6.2
Our theory of interpolation does not apply here because $p(\xi)$ is not fixed. We have to find

the coefficients of the polynomial $p(x) = a + b\,x + c\,x^2$ such that $p(0) = a = \alpha$, $p(1) = a + b + c = \beta$ and $p'(\xi) = b + 2c\,\xi = \gamma$. The first equation gives the value for a which is substituted into the other equations to give $b + c = \beta - \alpha$ and $b + 2c\,\xi = \gamma$. If we subtract the first equation from the second equation, we get the system $b + c = \beta - \alpha$ and $(2\xi - 1)\,c = \gamma - \beta + \alpha$.

If $\xi \neq 1/2$, then this system has a unique solution

$$c = \frac{\gamma - \beta + \alpha}{2\xi - 1}\ , \quad b = (\beta - \alpha) - \frac{\gamma - \beta + \alpha}{2\xi - 1} = \frac{-\gamma + 2\xi\,\beta - 2\xi\,\alpha}{2\xi - 1} \quad \text{and} \quad a = \alpha\ .$$

If $\xi = 1/2$ and $\gamma = \beta - \alpha$, then c is free, $b = \beta - \alpha - c$ and $a = \alpha$. There is an infinite number of solutions, and thus an infinite number of interpolating polynomial of degree at most 2. Note that $p'(1/2) = b + c = \beta - \alpha = \gamma$. Several interpolating polynomial are drawn in the following figure.



If $\xi = 1/2$ and $\gamma \neq \beta - \alpha$, there is no solution and thus no interpolating polynomial of degree at most 2.

Question 6.3
The polynomial r is of degree at most n because p and q are polynomial of degree at most $n - 1$. Moreover, $r(x_0) = p(x_0) = f(x_0)$, $r(x_n) = q(x_n) = f(x_n)$ and

$$\begin{aligned} r(x_j) &= \frac{x_j - x_n}{x_0 - x_n}\,p(x_j) + \frac{x_j - x_0}{x_n - x_0}\,q(x_j) = \frac{x_j - x_n}{x_0 - x_n}\,f(x_j) + \frac{x_j - x_0}{x_n - x_0}\,f(x_j) \\ &= \left(\frac{x_j - x_n}{x_0 - x_n} + \frac{x_j - x_0}{x_n - x_0}\right) f(x_j) = f(x_j) \end{aligned}$$

for $0 < j < n$. Hence, r is the interpolating polynomial of degree at most n at x_0, x_1, $\ldots$, x_n since this polynomial is unique.

Question 6.4
The Lagrange's form of the polynomial p is

$$p(x) = \sum_{i=0}^{n} f(x_i) \prod_{\substack{j=0 \\ i \neq j}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right)\ .$$

The coefficient of x^n in $f(x_i)\prod_{\substack{j=0\\ i\neq j}}^{n}\left(\frac{x-x_j}{x_i-x_j}\right)$ is $f(x_i)\prod_{\substack{j=0\\ i\neq j}}^{n}\left(\frac{1}{x_i-x_j}\right)=f(x_i)\ell_i$. The sum of these coefficients gives the coefficient of x^n in p.

If f is a polynomial of degree less than n, the interpolating polynomial p of f of degree at most n at x_0, x_1, ..., x_n is f itself by uniqueness. The coefficient of x^n in $p=f$ is therefore 0. Since the coefficient of x^n is $\sum_{i=0}^{n} f(x_i)\ell_i$, we have that $\sum_{i=0}^{n} f(x_i)\ell_i=0$.

Question 6.5

a) For $k\in\{0,1,2,\ldots,n\}$,

$$\prod_{\substack{j=0\\ i\neq j}}^{n}(x_k-x_j)=\underbrace{(x_k-x_0)(x_k-x_1)\ldots(x_k-x_{k-1})}_{k\text{ positive factors}}\ \underbrace{(x_k-x_{k+1})\ldots(x_k-x_n)}_{n-k\text{ negative factors}}\ .$$

Hence $\operatorname{sgn}(\ell_k)=(-1)^{n-k}$.

b) To prove (6.4.2), we consider $f(x)=x^n$. Since the interpolating polynomial p of $f(x)=x^n$ of degree at most n at the points x_0, x_1, ..., x_n is f itself by uniqueness, the coefficient of x^n in $f(x)=p(x)=x^n$ is $f[x_0,x_1,\ldots,x_n]=\sum_{j=0}^{n} x_i^n\ell_j=1$ according to (6.4.1).

To prove (6.4.3), we consider $f(x)=1$ for all x. The interpolating polynomial p of f of degree at most n at the points x_0, x_1, ..., x_n is $p(x)=f(x)=1$ for all x by uniqueness. Hence, the coefficient of x^n in $f(x)=p(x)=1$ is

$$f[x_0,x_1,\ldots,x_n]=\sum_{j=0}^{n}\ell_j=\begin{cases}1 & \text{if}\quad n=0\\ 0 & \text{if}\quad n>0\end{cases}$$

according to (6.4.1).

Question 6.6

The proof is by induction. (6.4.4) is true when $m=0$ because $\frac{1}{0!}\sum_{j=0}^{0}(-1)^{0-j}\binom{0}{j}f(j)=f(0)$ and $f[0]=f(0)$.

We assume that (6.4.4) is true for $m=k$ and show that it is then true for $m=k+1$.

Let $g(n)=f(n+1)$ for all n. We claim that $f[n,n+1,\ldots,n+r]=g[n-1,n,\ldots,n+r-1]$ for all n and $r\geq 0$. The proof of this claim is by induction on r. For $r=0$, we have $f[n]=f(n)$ and $g[n-1]=g(n-1)=f(n)$ for all n. Thus $f[n]=g[n-1]$ for all n. Suppose that $f[n,n+1,\ldots,n+r]=g[n-1,n,\ldots,n+r-1]$ is true for all n and some $r\geq 0$. Then

$$\begin{aligned}f[n,n+1,\ldots,n+r+1]&=\frac{f[n+1,n+2,\ldots,n+r+1]-f[n,n+1,\ldots,n+r]}{(n+r+1)-n}\\ &=\frac{g[n,n+1,\ldots,n+r]-g[n-1,n,\ldots,n+r-1]}{(n+r)-(n-1)}\\ &=g[n-1,n,\ldots,n+r]\end{aligned}$$

for all n, where the hypothesis of induction has been used for the second equality. Hence, $f[n, n+1, \ldots, n+r] = g[n-1, n, \ldots, n+r-1]$ is true for all n and r replaced by $r+1$.

We now go back to our main proof by induction. We have

$$
\begin{aligned}
f[0,1,2,\ldots,k+1] &= \frac{f[1,2,\ldots,k+1] - f[0,1,\ldots,k]}{k+1} = \frac{g[0,1,\ldots,k] - f[0,1,\ldots,k]}{k+1} \\
&= \frac{1}{k+1}\left(\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}g(j) - \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}f(j)\right) \\
&= \frac{1}{k+1}\left(\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}f(j+1) - \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}f(j)\right)
\end{aligned}
$$

The third equality is a consequence of the hypothesis of induction; namely, (6.4.4) with $m = k$ for both f and g.

If we replace j by $j-1$ in the first sum, we get

$$
\begin{aligned}
f[0,1,2,\ldots,k+1] &= \frac{1}{k+1}\left(\frac{1}{k!}\sum_{j=1}^{k+1}(-1)^{k-j+1}\binom{k}{j-1}f(j) - \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}f(j)\right) \\
&= \frac{1}{(k+1)!}\sum_{j=0}^{k+1}(-1)^{k-j+1}\left(\binom{k}{j-1} + \binom{k}{j}\right)f(j)
\end{aligned}
$$

where we have made used of the fact that $\binom{k}{-1} = 0$ and $\binom{k}{k+1} = 0$. Using the hint, we get

$$
f[0,1,2,\ldots,k+1] = \frac{1}{(k+1)!}\sum_{j=0}^{k+1}(-1)^{k-j+1}\binom{k+1}{j}f(j)
$$

which is (6.4.4) with $m = k+1$. This prove that (6.4.4) is true for all m by induction.

Question 6.7

The table of divided differences is

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$
1	$\boxed{1}$	$\boxed{-0.951625820}$	$\boxed{0.438432783}$
1.1	0.904837418	−0.820095985	0.383714117
1.3	0.740818221	−0.704981750	0.324799000
1.4	0.670320046	−0.607542050	0.275321725
1.6	0.548811636	−0.497413360	
1.8	0.449328964		

$f[\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot,\cdot]$
$\boxed{-0.136796667}$	$\boxed{0.0316107222}$	$\boxed{-0.00580682540}$
−0.1178302333	0.0269652619	
−0.0989545500		

The interpolating polynomial is

$$
p(x) \approx 1 - 0.951625820\,(x-1) + 0.438432783\,(x-1)(x-1.1)
$$

$$\begin{aligned}&- 0.136796667\,(x-1)(x-1.1)(x-1.3)\\&+ 0.0316107222\,(x-1)(x-1.1)(x-1.3)(x-1.4)\\&- 0.00580682540\,(x-1)(x-1.1)(x-1.3)(x-1.4)(x-1.6)\ ,\end{aligned}$$

where we have rounded the coefficients to 9 digits. The nested form of this polynomial is

$$\begin{aligned}p(x) \approx 1 + (x-1)\big(-0.951625820 + (x-1.1)\big(0.438432783 + (x-1.3)\big(-0.136796667\\ + (x-1.4)\big(0.0316107222 - 0.00580682540\,(x-1.6)\big)\big)\big)\big)\ .\end{aligned}$$

Hence,

$$\begin{aligned}f(1.35) \approx p(1.35) \approx 1 + 0.35\big(-0.951625820 + 0.25\big(0.438432783 + 0.05\big(-0.136796667\\ - 0.05\big(0.0316107222 - 0.00580682540 \times (-0.25)\big)\big)\big)\big) \approx 0.704688114\ .\end{aligned}$$

Question 6.8

a) We have $f(x) = e^{x/2}$, $f'(x) = e^{x/2}/2$ and $f''(x) = e^{x/2}/4$. The table of divided differences is

x	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$
0	1	$(e-1)/2$	$1/4$	$(e-2)/16$
2	e	$e/2$	$e/8$	
2	e	$e/2$		
2	e			

The interpolating polynomial is

$$\begin{aligned}p(x) &= 1 + \left(\frac{e-1}{2}\right)x + \frac{1}{4}\,x(x-2) + \left(\frac{e-2}{16}\right)x(x-2)^2\\&= 1 + x\left(\frac{e-1}{2} + (x-2)\left(\frac{1}{4} + \left(\frac{e-2}{16}\right)(x-2)\right)\right)\\&\approx 1 + x\,(0.8591409142 + (x-2)\,(0.25 + 0.04489261428\,(x-2)))\ .\end{aligned}$$

b)

$$f(1) \approx p(1) = 1 + (0.8591409142 - (0.25 - 0.04489261428)) \approx 1.65403352848\ .$$

c) It follows from Theorems 6.2.5 and 6.2.7 that, for each $x \in [0,2]$, there exists $\xi = \xi(x) \in [0,2]$ such that

$$|f(x) - p(x)| = \left|\frac{1}{4!}f^{(4)}(\xi)\,x(x-2)^3\right|\ .$$

Moreover

$$|f^{(4)}(x)| = \left|\frac{e^{x/2}}{2^4}\right| \leq \frac{e}{2^4}$$

for all $x \in [0,2]$. Hence,

$$|f(x) - p(x)| \leq \frac{e}{2^4\,4!}\left|x(x-2)^3\right| \leq \frac{e}{4!} \approx 0.113261743\ .$$

We have use the upper bound 2^4 for $|x(x-2)^3|$. A better (i.e. smaller) bound could be found by maximizing $g(x) = |x(x-2)^3| = x(2-x)^3$ on the interval $[0,2]$. Since $g'(x) =$

$(2-x)^2(2-4x)=0$, the critical points of g on $[0,2]$ are $x=1/2$ and $x=2$. It follows from the Extremum Theorem, Theorem 2.1.4, that the maximum of g on $[0,2]$ is the maximum of $g(0)=0$, $g(2)=0$ and $g(1/2)=3^3/2^4$. Hence,

$$|f(x)-p(x)| \leq \frac{e}{2^4\,4!}\left|x(x-2)^3\right| \leq \frac{3^3\,e}{2^8\,4!} \approx 0.01194557444 \ .$$

It is a better upper bound than the previous one. However, we do not usually maximize the polynomial part of the truncation error as we have just done because the roots of its derivative may be too difficult to find if the degree of this polynomial is high.

Question 6.9
a) The table of divided differences is

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot]$
0	e	$1-e$	$e-2$	$5/2-e$	$(e+e^{-1}-3)/2$
1	1	-1	$1/2$	$e^{-1}-1/2$	
1	1	-1	e^{-1}		
1	1	$e^{-1}-1$			
2	e^{-1}				

The interpolating polynomial is

$$\begin{aligned}
p(x) &= e+(1-e)x+(e-2)x(x-1)+\left(\frac{5}{2}-e\right)x(x-1)^2+\left(\frac{e+e^{-1}-3}{2}\right)x(x-1)^3 \\
&= e+x\Bigg((1-e)+(x-1)\Bigg((e-2)+(x-1)\Bigg(\left(\frac{5}{2}-e\right)+\left(\frac{e+1/e-3}{2}\right)(x-1)\Bigg)\Bigg)\Bigg) \ .
\end{aligned}$$

b)

$$f(1.1) \approx e+1.1\Bigg((1-e)+0.1\Bigg((e-2)+0.1\Bigg(\left(\frac{5}{2}-e\right)+0.1\left(\frac{e+1/e-3}{2}\right)\Bigg)\Bigg)\Bigg) \approx 0.9048291 \ .$$

c) It follows from Theorems 6.2.5 and 6.2.7 that, for each $x\in[0,2]$, there exists $\xi=\xi(x)\in[0,2]$ such that

$$|f(x)-p(x)| = \left|\frac{f^{(5)}(\xi)}{5!}\,x(x-1)^3(x-2)\right| \ .$$

Hence

$$|f(x)-p(x)| \leq \frac{e}{5!}\,|x(x-1)^3(x-2)| \leq \frac{e}{30}$$

for $0\leq x\leq 2$. We have use the conservative upper bound $|x(x-1)^3(x-2)|\leq 4$ for $0\leq x\leq 2$ provided by $|x|\leq 2$, $|(x-1)^3|\leq 1$ and $|x-2|\leq 2$ for $0\leq x\leq 2$.

Question 6.10

The table of divided differences is

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot,\cdot]$
1	**1.7165256995**	**−1.4444065708**	**0.14399171305**	**0.28047774588**	**−0.16268960195**	**0.054237061909**
1	1.7165256995	−1.4444065708	0.36837390975	0.052712303155	−0.086757715275	
1	1.7165256995	−1.1497074430	0.44217113417	−0.06874849823		
1.8	0.79675974510	−0.53066785517	0.34592323664			
2.4	0.47835903200	−0.32311391318				
2.4	0.47835903200					

The interpolating polynomial is

$$\begin{aligned} p(x) &\approx 1.7165256995 - 1.4444065708\,(x-1) + 0.14399171305\,(x-1)^2 \\ &\quad + 0.28047774588\,(x-1)^3 - 0.16268960195\,(x-1)^3(x-1.8) \\ &\quad + 0.054237061909\,(x-1)^3(x-1.8)(x-2.4)\ , \end{aligned}$$

where we have rounded the coefficients to 11 digits. The nested form of this polynomial is

$$\begin{aligned} p(x) &\approx 1.7165256995 + (x-1)\Big(-1.4444065708 + (x-1)\Big(14399171305 \\ &\quad + (x-1)\Big(0.28047774588 + (x-1.8)\Big(-0.16268960195 + 0.054237061909\,(x-2.4)\Big)\Big)\Big)\Big)\ . \end{aligned}$$

Hence,

$$\begin{aligned} f(1.75) \approx p(1.75) &\approx 1.7165256995 + 0.75\Big(-1.4444065708 + 0.75\Big(0.14399171305 \\ &\quad + 0.75\Big(0.28047774588 - 0.05\Big(-0.16268960195 + 0.054237061909\,(-0.65)\Big)\Big)\Big)\Big) \\ &\approx 0.83671803379\ . \end{aligned}$$

The absolute error is

$$|f(1.75) - p(1.75)| \approx |0.83673651441075 - 0.83671803379| \approx 0.0000184806\ .$$

The relative error is

$$\frac{|f(1.75) - p(1.75)|}{|f(1.75)|} \approx \frac{|0.83673651441075 - 0.8367180338|}{0.83673651441075} \approx 0.0000220865\ .$$

Since $k = 5$ is the largest positive integer such that $0.0000220865 < 5 \times 10^{-k}$, there are 5 significant digits.

Question 6.12
If the divided differences of order three are always egal to 1, then the divided differences of order fourth and higher are 0. Thus p is a polynomial of degree 3.

A table of divided differences of p at 0, 1, 2 and 3 (any other value greater than 2 would have been fine) will have the following table.

x_i	$p[x_i]$	$p[x_i, x_j]$	$p[x_i, x_j, x_k]$	$p[x_i, x_j, x_k, x_l]$
0	2	−1	2	1
1	1	3	c_1	
2	4	c_2		
3	c_3			

where c_1, c_2, and c_3 are constants. Since p is a polynomial of degree 3, we have

$$p(x) = 2 - x + 2x(x-1) + x(x-1)(x-2) = 2 - x - x^2 + x^3 \ .$$

Question 6.13

a) We have $f(x) = \cos\left(\dfrac{\pi}{2} - x\right)$, $f'(x) = \sin\left(\dfrac{\pi}{2} - x\right)$ and $f''(x) = -\cos\left(\dfrac{\pi}{2} - x\right)$.

The table of divided differences is

x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot,\cdot]$
0	0	0.90031632	−0.24600202	−0.13693866	0.02886361
0.78539816	0.70710678	0.70710678	−0.35355339	−0.09159981	
0.78539816	0.70710678	0.70710678	−0.42549571		
0.78539816	0.70710678	0.37292323			
1.57079633	1.00000000				

To save some space, we have only printed the numbers to 8 decimal places in the table above. However, computations were done with full Matlab accuracy.

The interpolating polynomial is

$$\begin{aligned} p(x) \approx\ & 0.900316316157\, x - 0.2460020203444\, x(x-\pi/4) \\ & - 0.136938657691\, x(x-\pi/4)^2 + 0.0288636058864\, x(x-\pi/4)^3 \ , \end{aligned}$$

where we have rounded the coefficients to 12 digits.

b) The nested form of this polynomial is

$$\begin{aligned} p(x) \approx \Bigg(& 0.900316316157 + \Bigg(-0.2460020203444 + \\ & \left(-0.136938657691 + 0.0288636058864 \left(x - \frac{\pi}{4}\right)\right)\left(x - \frac{\pi}{4}\right)\Bigg)\left(x - \frac{\pi}{4}\right)\Bigg) x \ . \end{aligned}$$

Hence,

$$\begin{aligned} f(\pi/8) \approx p(1.75) \approx \Bigg(& 0.900316316157 + \Bigg(-0.2460020203444 + \\ & \left(-0.136938657691 + 0.0288636058864 \left(\frac{-\pi}{8}\right)\right)\left(\frac{-\pi}{8}\right)\Bigg)\left(\frac{-\pi}{8}\right)\Bigg)\left(\frac{\pi}{8}\right) \end{aligned}$$

$$\approx 0.382510687216 \ .$$

c) It follows from Theorems 6.2.5 and 6.2.7 that, for each $x \in [0, \pi/2]$, there exists $\xi = \xi(x) \in [0, \pi/2]$ such that

$$|f(x) - p(x)| = \left| \frac{1}{5!} f^{(5)}(\xi)\, x \left(x - \frac{\pi}{4}\right)^3 \left(x - \frac{\pi}{2}\right) \right| \ .$$

However

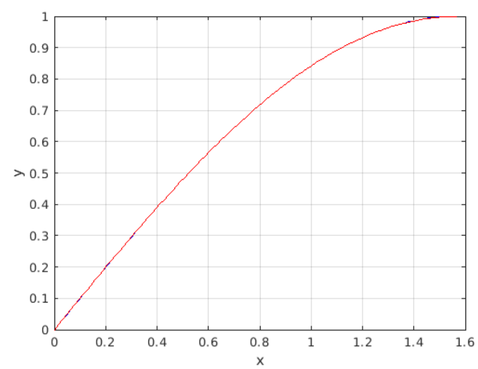
$$|f^{(5)}(x)| = \left| \sin\left(\frac{\pi}{2} - x\right)\right| \leq 1$$

for all x. Hence,

$$|f(x) - p(x)| \leq \frac{1}{5!} \left| x \left(x - \frac{\pi}{4}\right)^3 \left(x - \frac{\pi}{2}\right)\right| \leq \frac{1}{5!}\left(\frac{\pi}{4}\right)^5 \approx 0.0025$$

because $|x(x - \pi/2)| \leq (\pi/4)^2$ (the maximum is reached at $x = \pi/4$) and $|x - \pi/4| \leq \pi/4$ for $x \in [0, \pi/2]$.

d) The following figure contains the graph of p in blue and the graph of f in red. The graph of p was drawn first and is almost completely covered by the graph of f. The two graphs are basically indistinguishable at the level of the graph accuracy.



Chapter 7 : Splines

Question 7.1

We have

$$p_i(x) = \left(\left(\alpha_i(x - x_i) + \beta_i\right)(x - x_i) + \gamma_i\right)(x - x_i) + \delta_i$$

on $[x_i, x_{i+1}]$, where $x_0 = 0$, $x_1 = 1$, $x_2 = 3$. $x_3 = 4$, $x_4 = 5$ and $x_5 = 5.5$.

The solution of $A\mathbf{z} = \mathbf{b}$, where

$$A = \begin{pmatrix} 2\Delta x_0 & \Delta x_0 & 0 & 0 & 0 & 0 \\ \Delta x_0 & 2(\Delta x_1 - \Delta x_0) & \Delta x_1 & 0 & 0 & 0 \\ 0 & \Delta x_1 & 2(\Delta x_2 - \Delta x_1) & \Delta x_2 & 0 & 0 \\ 0 & 0 & \Delta x_2 & 2(\Delta x_3 - \Delta x_2) & \Delta x_3 & 0 \\ 0 & 0 & 0 & \Delta x_3 & 2(\Delta x_4 - \Delta x_3) & \Delta x_4 \\ 0 & 0 & 0 & 0 & \Delta x_4 & 2\Delta x_4 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 6 & 2 & 0 & 0 & 0 \\ 0 & 2 & 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 1 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} -6f'(x_0) + 6\,\dfrac{f(x_1) - f(x_0)}{x_1 - x_0} \\ 6\,\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - 6\,\dfrac{f(x_1) - f(x_0)}{x_1 - x_0} \\ 6\,\dfrac{f(x_3) - f(x_2)}{x_3 - x_2} - 6\,\dfrac{f(x_3) - f(x_1)}{x_2 - x_1} \\ 6\,\dfrac{f(x_4) - f(x_3)}{x_4 - x_3} - 6\,\dfrac{f(x_3) - f(x_2)}{x_3 - x_2} \\ 6\,\dfrac{f(x_5) - f(x_4)}{x_5 - x_4} - 6\,\dfrac{f(x_4) - f(x_3)}{x_4 - x_3} \\ 6f'(x_5) - 6\,\dfrac{f(x_5) - f(x_4)}{x_5 - x_4} \end{pmatrix} = \begin{pmatrix} 6 \\ -6 \\ -6 \\ 6 \\ -12 \\ 0 \end{pmatrix} ,$$

is

$$\mathbf{z} = \begin{pmatrix} 3.615279672578445 \\ -1.230559345156889 \\ -1.115961800818554 \\ 3.156889495225103 \\ -5.511596180081855 \\ 2.755798090040928 \end{pmatrix} .$$

The coefficients of p_i are given by

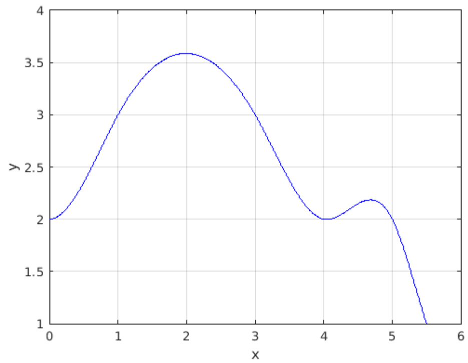
$$\delta_i = f(x_i) \ , \quad \gamma_i = -\frac{z_i \Delta x_i}{3} - \frac{z_{i+1}\Delta x_i}{6} + \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} \ , \quad \beta_i = \frac{z_i}{2} \quad \text{and} \quad \alpha_i = \frac{z_{i+1} - z_i}{6\Delta x_i}$$

for $i = 0, 2, \ldots, 5$.

The following table lists the values of the coefficients of p_i.

i	α_i	β_i	γ_i	δ_i
0	−0.807639836289222	1.807639836289222	0	2
1	0.009549795361528	−0.615279672578445	1.192360163710777	3
2	0.712141882673943	−0.557980900409277	−1.154160982264666	3
3	−1.444747612551160	1.578444747612551	−0.133697135061392	2
4	2.755798090040928	−2.755798090040928	−1.311050477489768	2

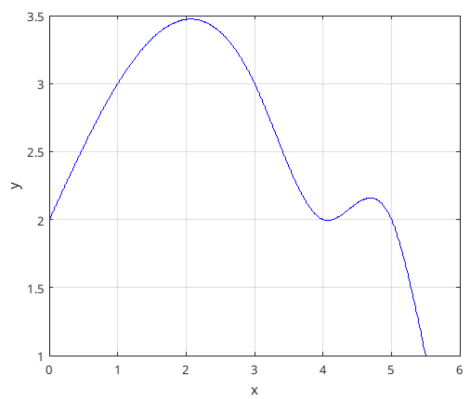
The graph of the clamped cubic spline polynomial is given below.



Question 7.2

The MATLAB code to generate the system $A\mathbf{z} = \mathbf{b}$ for the natural cubic spline interpolation is given in Code 16.0.1.

The MATLAB code used to produce the figure below is given in Code 16.0.2.



Code 16.0.1 (Natural Cubic Spline Interpolant - System)

This program computes the tridiagonal matrix A and the right hand side column vector $\mathbf{b}$ associated to the natural cubic spline interpolation.
Input: The nodes x_i for $0 \leq i \leq n$ (x(i+1) in the code below).
The values $f(x_i)$ for $0 \leq i \leq n$ (f(i+1) in the code below).
Output: The lower diagonal L, the diagonal D and the upper diagonal U of the tridiagonal matrix A.
The right hand side $\mathbf{b}$ of $A\mathbf{x} = \mathbf{b}$.

```
%   [L,D,U,b] = naturalsplinematrix(f,p)

function [L,D,U,b] = naturalsplinematrix(f,fp,p)
  N = length(p);
  L = repmat(NaN,1,N-3);
  U = repmat(NaN,1,N-3);
  D = repmat(NaN,1,N-2);
  b = repmat(NaN,1,N-2);
```

```
  dp = p(2)-p(1);
  if (dp == 0)
    return;
  end
  ratio = (f(2)-f(1))/dp;

  for n=1:N-2
    prevdp = dp;
    dp = p(n+2)-p(n+1);
    if (dp == 0)
      return;
    end
    prevratio = ratio;
    ratio = (f(n+2)-f(n+1))/dp;
    D(n) = 2*(dp+prevdp);
    if ( n < N - 2 )
        U(n) = dp;
        L(n) = dp;
    end
    b(n) = 6*(ratio - prevratio);
  end
end
```

Code 16.0.2 (**Code for Question 7.2**)

```
p = [ 0 , 1 , 3 , 4 , 5 , 5.5 ];
f = [ 2 , 3 , 3 , 2 , 2 , 1 ];
[L,D,U,b] = naturalsplinematrix(f,p)

z = tridmatrix(L,D,U,b);
z = [0,z,0]

x = 0:0.01:5.5;
[y,coeffs] = splinepoly(z,f,p,x);
coeffs
plot(x,y,'b')
grid on
xlabel('x')
ylabel('y')
```

Chapter 8 : Least Square Approximation (in L^2)

Question 8.1
Suppose that $\{P_k^{[i]}\}_{k=0}^{\infty}$ for $i = 1$ and 2 are two orthogonal families of monic polynomials

such that $\left\langle p, P_k^{[i]} \right\rangle = 0$ for all polynomial p of degree less than k. Given $k > 0$, we have that $P_k^{[1]} - P_k^{[2]}$ is a polynomial of degree $k-1$ because both polynomials are monic, such that

$$\left\langle p, P_k^{[1]} - P_k^{[2]} \right\rangle = \left\langle p, P_k^{[1]} \right\rangle - \left\langle p, P_k^{[2]} \right\rangle = 0$$

for all polynomial p of degree less than k. In particular,

$$\left\langle P_k^{[1]} - P_k^{[2]}, P_k^{[1]} - P_k^{[2]} \right\rangle = \int_a^b (P_k^{[1]} - P_k^{[2]})^2 w(x)\,\mathrm{d}x = 0 \ .$$

Since $P_k^{[1]} - P_k^{[2]}$ is continuous on $[a,b]$, we get that $P_k^{[1]}(x) = P_k^{[2]}(x)$ for all $x \in [a,b]$.

Question 8.2

We have from Theorem 8.2.3 that (8.4.1) is true for some constants A_k, B_k and C_k. We only have to prove that they have the form suggested in the question. We also have from this theorem that $A_k = \dfrac{a_{k+1,k+1}}{a_{k,k}}$ for $k \geq 0$ and $C_k = \dfrac{A_k}{A_{k-1}} = \dfrac{a_{k+1,k+1}a_{k-1,k-1}}{a_{k,k}^2}$ for $k > 0$. Because of our choice for $a_{-1,-1}$, we also have $C_0 = 0$ as required. Note that $A_k = \dfrac{a_{k+1,k+1}}{a_{k,k}}$ can be easily proved by comparing the coefficient of x^{k+1} on both sides of (8.4.1).

It remains only to prove that

$$B_k = \int_a^b x P_k^2(x) w(x)\,\mathrm{d}x = \frac{a_{k,k-1}}{a_{k,k}} - \frac{a_{k+1,k}}{a_{k+1,k+1}} \ .$$

Since

$$\begin{aligned} xP_k(x) &= \frac{a_{k,k}}{a_{k+1,k+1}} P_{k+1}(x) + \frac{1}{a_{k,k}} \left(a_{k,k-1} - \frac{a_{k,k}a_{k+1,k}}{a_{k+1,k+1}} \right) P_k(x) + q(x) \\ &= \frac{a_{k,k}}{a_{k+1,k+1}} P_{k+1}(x) + \left(\frac{a_{k,k-1}}{a_{k,k}} - \frac{a_{k+1,k}}{a_{k+1,k+1}} \right) P_k(x) + q(x) \ , \end{aligned}$$

where q is a polynomial of degree at most $k-1$, we have

$$\begin{aligned} B_k &= \int_a^b x P_k^2(x) w(x)\,\mathrm{d}x \\ &= \int_a^b \left(\frac{a_{k,k}}{a_{k+1,k+1}} P_{k+1}(x) + \left(\frac{a_{k,k-1}}{a_{k,k}} - \frac{a_{k+1,k}}{a_{k+1,k+1}} \right) P_k(x) + q(x) \right) P_k(x) w(x)\,\mathrm{d}x \\ &= \frac{a_{k,k}}{a_{k+1,k+1}} \underbrace{\int_a^b P_{k+1}(x) P_k(x) w(x)\,\mathrm{d}x}_{=0} + \left(\frac{a_{k,k-1}}{a_{k,k}} - \frac{a_{k+1,k}}{a_{k+1,k+1}} \right) \underbrace{\int_a^b P_k^2(x) w(x)\,\mathrm{d}x}_{=1} \\ &\qquad + \underbrace{\int_a^b q(x) P_k(x) w(x)\,\mathrm{d}x}_{=0} = \left(\frac{a_{k,k-1}}{a_{k,k}} - \frac{a_{k+1,k}}{a_{k+1,k+1}} \right) \ . \end{aligned}$$

Chapter 9 : Uniform Approximation

Question 9.1
The Taylor polynomial of f of degree $2n+1$ about the origin is

$$p(x) = x - \sum_{j=0}^{n} \frac{(-1)^j}{(2j+1)!}\, x^{2j+1} \ ,$$

where

$$f(x) - p(x) = -\frac{1}{(2n+2)!}\, f^{(2n+2)}(\xi) x^{2n+2}$$

for some ξ between 0 and x.

Since $f^{(2n+2)}(\xi)$ is either $\cos(\xi)$ or $\sin(\xi)$, we have $|f^{(2n+2)}(\xi)| \leq 1$. We need to find n such that

$$\left| \frac{1}{(2n+2)!}\, f^{(2n+2)}(\xi) x^{2n+2} \right| \leq \frac{1}{(2n+2)!} < 10^{-9}$$

for $|x| < 1$. We have $1/(2n+2)! = 1/14! < 10^{-9}$ for $n = 6$ and $1/(2n+2)! = 1/12! > 10^{-9}$ for $n = 5$. Thus

$$f(x) \approx x - \sum_{j=0}^{6} \frac{(-1)^j}{(2j+1)!}\, x^{2j+1} = \sum_{j=1}^{6} \frac{(-1)^{j+1}}{(2j+1)!}\, x^{2j+1}$$

with an accuracy of at least 10^{-9} for $|x| < 1$.

Question 9.3
We note that $f'(x) = (\cos(x) - \sin(x))e^x$, $f''(x) = -2\sin(x)e^x$ and $f^{(3)}(x) = -2(\sin(x) + \cos(x))e^x$.

The Taylor polynomial of f of degree two about the origin is

$$p_2(x) = f(0) + f'(0)\, x + \frac{f''(0)}{2}\, x^2 = 1 + x$$

because $f''(0) = 0$. So, there is no term in x^2. The truncation error is given by

$$f(x) - p_2(x) = \frac{f^{(3)}(\xi)}{3!}\, x^3$$

for $\xi \in [0, x]$.

Hence, $f(1/2) \approx p_2(1/2) = 3/2$. Moreover, we have

$$|f(1/2) - p_2(1/2)| = \frac{|f^{(3)}(\xi)|}{3!} \left(\frac{1}{2}\right)^3$$

for $\xi \in [0, 1/2]$. Since $f^{(4)}(x) = -4\cos(x)e^x < 0$ for $0 \leq x \leq 1/2$, we have that $f^{(3)}$ is decreasing on $[0, 1/2]$. Thus, $|f^{(3)}(x)|$ reaches its maximum value at one of the endpoints 0 or 1/2. Since $|f^{(3)}(0)| = 2$ and $|f^{(3)}(1/2)| \approx 4.47465624$ (rounded up after 8 digits), we have that $|f^{(3)}(\xi)| \leq |f^{(3)}(1/2)| \leq 4.47465624$. Hence

$$|f(1/2) - p_2(1/2)| \leq \frac{4.47465624}{3!} \left(\frac{1}{2}\right)^3 \approx 0.093222 \ .$$

This is an overestimate of the error which is

$$|f(1/2) - p_2(1/2)| = \left|\cos(1/2)e^{1/2} - \frac{3}{2}\right| \approx 0.05311096 \ .$$

Chapter 12 : Numerical Differentiation and Integration

Question 12.1
The polynomial interpolation of degree at most 2 at the points x_0, x_1 and x_2 is

$$\begin{aligned} f(x) &= f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) \\ &\qquad + f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2) \ . \end{aligned}$$

If we derive, we get

$$\begin{aligned} f'(x) &= f[x_0,x_1] + f[x_0,x_1,x_2]\big((x-x_0)+(x-x_1)\big) \\ &\qquad + f[x_0,x_1,x_2,x]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\ &\qquad + f[x_0,x_1,x_2,x,x]\,(x-x_0)(x-x_1)(x-x_2) \ , \end{aligned}$$

where we have used the formula $\dfrac{\mathrm{d}}{\mathrm{d}x} f[x_0,x_1,x_2,x] = f[x_0,x_1,x_2,x,x]$. Since $f[x_0,x_1,x_2,x] = \dfrac{1}{3!}\dfrac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)$ and $f[x_0,x_1,x_2,x,x] = \dfrac{1}{4!}\dfrac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)$ for some ξ and η in the smallest interval containing x_0, x_1, x_2 and x, we get

$$\begin{aligned} f'(x) &= f[x_0,x_1] + f[x_0,x_1,x_2]\big((x-x_0)+(x-x_1)\big) \\ &\qquad + \frac{1}{3!}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\ &\qquad + \frac{1}{4!}\frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\,(x-x_0)(x-x_1)(x-x_2) \ . \end{aligned}$$

Each of the x_i must be replaced by one of a, $a+h$ and $a+2h$ but there is no obligation to have $x_0 < x_1 < x_2$. We take $x_0 = a+2h$, $x_1 = a+h$ and $x_2 = a$. Hence, for $x = a$, the previous equation becomes

$$\begin{aligned} f'(a) &= f[a+2h,a+h] + f[a+2h,a+h,a]\big((-2h)+(-h)\big) + \frac{1}{3!}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\big((-2h)(-h)\big) \\ &= \frac{f(a+h)-f(a+2h)}{-h} + \left(\frac{\dfrac{f(a)-f(a+h)}{-h} - \dfrac{f(a+h)-f(a+2h)}{-h}}{-2h}\right)(-3h) \\ &\qquad + \frac{1}{3}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\,h^2 \\ &= \frac{-f(a+2h)+4f(a+h)-3f(a)}{2h} + \frac{1}{3}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\,h^2 \ . \end{aligned}$$

We get (12.9.1) with the truncation error $\dfrac{1}{3}\dfrac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\, h^2$.

Question 12.2

The polynomial interpolation of degree at most 2 of f at the three points x_0, x_1 and x_2 is

$$\begin{aligned} f(x) = f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) \\ + f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2)\ . \end{aligned}$$

If we derive once, we get

$$\begin{aligned} f'(x) = f[x_0,x_1] &+ f[x_0,x_1,x_2]\big((x-x_0)+(x-x_1)\big) \\ &+ f[x_0,x_1,x_2,x]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\ &+ f[x_0,x_1,x_2,x,x]\,(x-x_0)(x-x_1)(x-x_2)\ , \end{aligned}$$

where we have used the formula

$$\frac{\mathrm{d}}{\mathrm{d}x} f[x_0,x_1,x_2,x] = f[x_0,x_1,x_2,x,x]\ . \tag{16.8}$$

If we derive a second time, we get

$$\begin{aligned} f''(x) = 2\,f[x_0,x_1,x_2] &+ 2\,f[x_0,x_1,x_2,x]\big((x-x_0)+(x-x_1)+(x-x_2)\big) \\ &+ 2\,f[x_0,x_1,x_2,x,x]\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\ &+ 2\,f[x_0,x_1,x_2,x,x,x]\,(x-x_0)(x-x_1)(x-x_2)\ , \end{aligned}$$

where we have used (16.8) and the formula $\dfrac{\mathrm{d}}{\mathrm{d}x} f[x_0,x_1,x_2,x,x] = 2\,f[x_0,x_1,x_2,x,x,x]$. Since $f[x_0,x_1,x_2,x] = \dfrac{1}{3!}\dfrac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)$, $f[x_0,x_1,x_2,x,x] = \dfrac{1}{4!}\dfrac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)$ and $f[x_0,x_1,x_2,x,x,x] = \dfrac{1}{5!}\dfrac{\mathrm{d}^5 f}{\mathrm{d}x^5}(\nu)$ for some ξ, η and ν in the smallest interval containing x_0, x_1, x_2 and x, we get

$$\begin{aligned} f''(x) = 2\,f[x_0,x_1,x_2] &+ \frac{2}{3!}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\big((x-x_0)+(x-x_1)+(x-x_2)\big) \\ &+ \frac{2}{4!}\frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\big((x-x_1)(x-x_2)+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)\big) \\ &+ \frac{2}{5!}\frac{\mathrm{d}^5 f}{\mathrm{d}x^5}(\nu)\,(x-x_0)(x-x_1)(x-x_2)\ . \end{aligned}$$

We take $x_0 = a$, $x_1 = a+h$ and $x_2 = a+2h$. Hence, for $x = a$, we get

$$\begin{aligned} f''(a) &= 2\,f[a,a+h,a+2h] + \frac{2}{3!}\frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\big((-h)+(-2h)\big) + \frac{2}{4!}\frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\,(-h)(-2h) \\ &= 2\left(\frac{\dfrac{f(a+2h)-f(a+h)}{h} - \dfrac{f(a+h)-f(a)}{h}}{2h}\right) - \frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\,h + \frac{1}{3!}\frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\,h^2 \\ &= \frac{f(a)-2f(a+h)+f(a+2h)}{h^2} - \frac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\,h + \frac{1}{3!}\frac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\,h^2\ . \end{aligned}$$

We get (12.9.2) with the truncation error $-\dfrac{\mathrm{d}^3 f}{\mathrm{d}x^3}(\xi)\,h + \dfrac{1}{3!}\dfrac{\mathrm{d}^4 f}{\mathrm{d}x^4}(\eta)\,h^2$.

Question 12.3

Let $L_h^0(f) = L_h(f)$. We prove by induction on n that

$$L(f) = L_{h/2^k}^n(f) + \sum_{j=n+1}^{\infty} \frac{(2^{2n-1} - 2^{2j-1})\ldots(2^3 - 2^{2j-1})(2 - 2^{2j-1})}{(2^{2n-1} - 1)\ldots(2^3 - 1)(2 - 1)}\, a_j \left(\frac{h}{2^k}\right)^{2j-1} , \tag{16.9}$$

where

$$L_{h/2^k}^n(f) = \frac{2^{2n-1} L_{h/2^k}^{n-1}(f) - L_{h/2^{k-1}}^{n-1}(f)}{2^{2n-1} - 1}$$

and $k \geq n > 0$.

n = 1) If we replace h by $h/2$ in (12.9.3), we get

$$L(f) = L_{h/2}(f) + \sum_{j=1}^{\infty} a_j \left(\frac{h}{2}\right)^{2j-1} . \tag{16.10}$$

If we subtract (12.9.3) from 2 times (16.10) and divide by $2 - 1$, we get

$$\begin{aligned}
L(f) &= L_{h/2}^1(f) + 2\sum_{j=1}^{\infty} a_j \left(\frac{h}{2}\right)^{2j-1} - \sum_{j=1}^{\infty} a_j\, h^{2j-1} \\
&= L_{h/2}^1(f) + \sum_{j=1}^{\infty}\left(2a_j \left(\frac{h}{2}\right)^{2j-1} - 2^{2j-1} a_j \left(\frac{h}{2}\right)^{2j-1}\right) = L_{h/2}^1(f) + \sum_{j=2}^{\infty} \frac{2 - 2^{2j-1}}{2-1}\, a_j \left(\frac{h}{2}\right)^{2j-1} ,
\end{aligned}$$

where

$$L_{h/2}^1(f) = \frac{2L_{h/2}(f) - L_h(f)}{2-1} .$$

So (16.9) is true for $n = 1$ and $k = 1$. Replacing h by $h/2$ as many times as we want, we get that (16.9) is true for $n = 1$ and $k \geq 1$.

n = m) We suppose that (16.9) is true for $n = m$.

n = m + 1) By induction, we have

$$L(f) = L_{h/2^k}^m(f) + \sum_{j=m+1}^{\infty} \frac{(2^{2m-1} - 2^{2j-1})\ldots(2^3 - 2^{2j-1})(2 - 2^{2j-1})}{(2^{2m-1} - 1)\ldots(2^3 - 1)(2 - 1)}\, a_j \left(\frac{h}{2^k}\right)^{2j-1} \tag{16.11}$$

and

$$L(f) = L_{h/2^{k+1}}^m(f) + \sum_{j=m+1}^{\infty} \frac{(2^{2m-1} - 2^{2j-1})\ldots(2^3 - 2^{2j-1})(2 - 2^{2j-1})}{(2^{2m-1} - 1)\ldots(2^3 - 1)(2 - 1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{2j-1} \tag{16.12}$$

for $k \geq m$.

If we subtract (16.11) from 2^{2m+1} times (16.12) and divide by $2^{2m+1} - 1$, we get

$$L(f) = L_{h/2^{k+1}}^{m+1}(f)$$

$$
\begin{aligned}
&+ \frac{1}{2^{2m+1}-1}\Bigg(2^{2m+1}\sum_{j=m+1}^{\infty}\frac{(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k+1}}\right)^{2j-1}\\
&-\sum_{j=m+1}^{\infty}\frac{(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k}}\right)^{2j-1}\Bigg)\\
&= L^{m+1}_{h/2^{k+1}}(f) + \frac{1}{2^{2m+1}-1}\Bigg(2^{2m+1}\sum_{j=m+1}^{\infty}\frac{(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k+1}}\right)^{2j-1}\\
&-\sum_{j=m+1}^{\infty}2^{2j-1}\frac{(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k+1}}\right)^{2j-1}\Bigg)\\
&= L^{m+1}_{h/2^{k+1}}(f) + \sum_{j=m+1}^{\infty}\frac{(2^{2m+1}-2^{2j-1})(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m+1}-1)(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k+1}}\right)^{2j-1}\\
&= L^{m+1}_{h/2^{k+1}}(f) + \sum_{j=m+2}^{\infty}\frac{(2^{2m+1}-2^{2j-1})(2^{2m-1}-2^{2j-1})\ldots(2^3-2^{2j-1})(2-2^{2j-1})}{(2^{2m+1}-1)(2^{2m-1}-1)\ldots(2^3-1)(2-1)}\,a_j\left(\frac{h}{2^{k+1}}\right)^{2j-1}\ ,
\end{aligned}
$$

where

$$
L^{m+1}_{h/2^{k+1}}(f) = \frac{2^{2m+1}L^{m}_{h/2^{k+1}}(f) - L^{m}_{h/2^{k}}(f)}{2^{2m+1}-1}\ .
$$

This is (16.9) for $n = m+1$ if we substitute $k \geq n$ by $k \geq m$.

The general formula is

$$
L^{n}_{h/2^{k}}(f) = \frac{2^{2n-1}L^{n-1}_{h/2^{k}}(f) - L^{n-1}_{h/2^{k-1}}(f)}{2^{2n-1}-1}
$$

for $k \geq n > 0$ with a truncation error of $O(h^{2n+1})$.

Question 12.4
Let $L^0_h(f) = L_h(f)$. We prove by induction on n that

$$
L(f) = L^{n}_{h/2^{k}}(f) + \sum_{j=n+1}^{\infty}\frac{(2^{3n}-2^{3j})\ldots(2^6-2^3)(2^3-2^{3j})}{(2^{3n}-1)\ldots(2^6-1)(2^3-1)}\,a_j\left(\frac{h}{2^{k}}\right)^{3j}\ , \tag{16.13}
$$

where

$$
L^{n}_{h/2^{k}}(f) = \frac{2^{3n}L^{n-1}_{h/2^{k}}(f) - L^{n-1}_{h/2^{k-1}}(f)}{2^{3n}-1}
$$

and $k \geq n > 0$.

n = 1) If we replace h by $h/2$ in (12.9.4), we get

$$
L(f) = L_{h/2}(f) + \sum_{j=1}^{\infty} a_j\left(\frac{h}{2}\right)^{3j}\ . \tag{16.14}
$$

If we subtract (12.9.4) from 2^3 times (16.14) and divide by 2^3-1, we get

$$
L(f) = L^1_{h/2}(f) + \frac{1}{2^3-1}\left(2^3\sum_{j=1}^{\infty}a_j\left(\frac{h}{2}\right)^{3j} - \sum_{j=1}^{\infty}a_j\,h^{3j}\right)
$$

$$= L^1_{h/2}(f) + \frac{1}{2^3-1}\sum_{j=1}^{\infty}\left(2^3 a_j \left(\frac{h}{2}\right)^{3j} - 2^{3j} a_j \left(\frac{h}{2}\right)^{3j}\right) = L^1_{h/2}(f) + \sum_{j=2}^{\infty}\frac{2^3-2^{3j}}{2^3-1}\, a_j \left(\frac{h}{2}\right)^{3j} ,$$

where

$$L^1_{h/2}(f) = \frac{2^3 L_{h/2}(f) - L_h(f)}{2^3-1} .$$

So (16.13) is true for $n=1$ and $k=1$. Replacing h by $h/2$ as many times as we want, we get that (16.13) is true for $n=1$ and $k \geq 1$.

n = m) We suppose that (16.13) is true for $n=m$.

n = m + 1) By induction, we have

$$L(f) = L^m_{h/2^k}(f) + \sum_{j=m+1}^{\infty}\frac{(2^{2m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^k}\right)^{3j} \tag{16.15}$$

and

$$L(f) = L^m_{h/2^{k+1}}(f) + \sum_{j=m+1}^{\infty}\frac{(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j} \tag{16.16}$$

for $k \geq m$.

If we subtract (16.15) from 2^{3m+3} times (16.16) and divide by $2^{3m+3}-1$, we get

$$\begin{aligned}
L(f) &= L^{m+1}_{h/2^{k+1}}(f) \\
&\quad + \frac{1}{2^{3m+3}-1}\left(2^{3m+3}\sum_{j=m+1}^{\infty}\frac{(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j}\right. \\
&\quad \left. - \sum_{j=m+1}^{\infty}\frac{(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^k}\right)^{3j}\right) \\
&= L^{m+1}_{h/2^{k+1}}(f) + \frac{1}{2^{3m+3}-1}\left(2^{3m+3}\sum_{j=m+1}^{\infty}\frac{(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j}\right. \\
&\quad \left. - \sum_{j=m+1}^{\infty}2^{3j}\frac{(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j}\right) \\
&= L^{m+1}_{h/2^{k+1}}(f) + \sum_{j=m+1}^{\infty}\frac{(2^{3m+3}-2^{3j})(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m+3}-1)(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j} \\
&= L^{m+1}_{h/2^{k+1}}(f) + \sum_{j=m+2}^{\infty}\frac{(2^{3m+3}-2^{3j})(2^{3m}-2^{3j})\dots(2^6-2^{3j})(2^3-2^{3j})}{(2^{3m+3}-1)(2^{3m}-1)\dots(2^6-1)(2^3-1)}\, a_j \left(\frac{h}{2^{k+1}}\right)^{3j} ,
\end{aligned}$$

where

$$L^{m+1}_{h/2^{k+1}}(f) = \frac{2^{3m+3}L^m_{h/2^{k+1}}(f) - L^m_{h/2^k}(f)}{2^{3m+3}-1} .$$

This is (16.13) for $n = m+1$ if we substitute $k \geq n$ by $k \geq m$.

The general formula is

$$L^n_{h/2^k}(f) = \frac{2^{3n} L^{n-1}_{h/2^k}(f) - L^{n-1}_{h/2^{k-1}}(f)}{2^{3n}-1}$$

for $k \geq n > 0$ with a truncation error of $O(h^{3n+3})$.

Question 12.5
With $L_h(f) = (f(3+h) - f(3-h))/(2h)$, we get the following table.

h	$L_h(f)$		$L^1_h(f)$		$L^2_h(f)$		$L^3_h(f)$	$L^4_h(f)$	$\lvert L^i_h(f) - L^{i-1}_{2h}(f)\rvert$
0.8	0.16441388								
0.4	0.15472628	$\boxed{4.13687170}$	0.15149708						-0.0129169
0.2	0.15238451	$\boxed{4.03339716}$	0.15160392	$\boxed{16.53008728}$	0.15161104				$0.11396344 \times 10^{-03}$
0.1	0.15180391	$\boxed{4.00829909}$	0.15161038	$\boxed{16.13017524}$	0.15161081	$\boxed{65.68377022}$	0.15161081		$-0.23203700 \times 10^{-06}$
0.05	0.15165906		0.15161078		0.15161081		0.15161081	0.15161081	$0.93304309 \times 10^{-10}$

All the values in the table have been rounded. We stop the procedure as soon as $|L^i_h(f) - L^{i-1}_{2h}(f)|$ gets smaller than 10^{-7} and take $L^i_h(f)$ as our approximation of $f'(3)$. We have also included the ratios defined in (12.2.10) to ensure that the approximating values $L^i_h(h)$ can be trusted. We have that $f'(3) \approx L^4_{0.05}(f) \approx 0.15161081$ meets our criterion of accuracy.

Question 12.7
Let $a = 1$, $b = 3$, $h = (b-a)/2m = 1/m$ and $x_i = 1 + i\,h$ for $i = 0, 1, 2, \ldots, n = 2m$.

The local truncation error for the composite midpoint rule is $-\dfrac{f''(\xi)(b-a)}{6}\,h^2$ for some $\xi \in [a,b]$. We seek a small m for which this truncation error will be smaller in absolute value than 10^{-5}. We have

$$f''(x) = \frac{1}{x} + \frac{x}{4} - 10 \ .$$

We use the Extremum Theorem to find the maximum of $f''(x)|$ on $[1,3]$. We have that $x = 2$ is the only critical point of $f^{(3)}(x) = -1/x^2 + 1/4$ in the interval $[1,3]$. Since $f''(2) = -9 < f''(3) = -107/12 < f''(1) = -35/4$, we have that $-9 \leq f''(x) \leq -35/4$ for $1 \leq x \leq 3$. Thus, $|f''(x)| \leq 9$ for $1 \leq x \leq 3$. Hence,

$$\left| -\frac{f''(\xi)(b-a)}{6}\,h^2 \right| = \frac{|f''(\xi)|}{3m^2} \leq \frac{3}{m^2}$$

because $1 \leq \xi \leq 3$. We chose m that satisfies $\dfrac{3}{m^2} < 10^{-5}$; namely, $m > \sqrt{3 \times 10^5} \approx 547.72$.

With $m = 548$, we get

$$\int_1^3 \left(x\ln(x) + \frac{x^3}{24} - 5x^2 \right) \mathrm{d}x \approx \frac{1}{274} \sum_{i=1}^{548} \left(x_{2i-1} \ln(x_{2i-1}) + \frac{x^3_{2i-1}}{24} - 5x^2_{2i-1} \right) \approx -39.5562347658 \ .$$

Question 12.8
Let $a = 2$, $b = 4$, $h = (b-a)/2m = 1/m$ and $x_i = 2 + i\,h$ for $i = 0, 1, 2, \ldots, n = 2m$.

The local truncation error for the composite Simpson rule is $-\dfrac{h^4(b-a)}{180} f^{(4)}(\xi)$ for some $\xi \in [a,b]$. We seek a small m for which this truncation error will be smaller in absolute value than 10^{-5}. We have
$$f^{(4)}(x) = -\frac{80}{81}(x+1)^{-11/3} .$$
Hence,
$$\left|\frac{h^4(b-a)}{180} f^{(4)}(\xi)\right| = \left(\frac{1}{90\,m^4}\right)\left(\frac{80}{81}\right)(\xi+1)^{-11/3} \leq \frac{8}{3^6\,m^4}3^{-11/3} = \frac{8}{3^{29/3}\,m^4}$$
because $2 \leq \xi \leq 4$. We chose m that satisfies $\dfrac{8}{3^{29/3}\,m^4} < 10^{-5}$; namely, $m > \left(\dfrac{8}{3^{29/3}}10^5\right)^{1/4} \approx 2.102468339$.

With $m = 3$, we get
$$\begin{aligned}\int_2^4 (x+2)^{1/3}\,\mathrm{d}x &\approx \frac{1}{9}\left((3)^{1/3} + 2\sum_{i=1}^{2}\left(1+\left(2+\frac{2i}{3}\right)\right)^{1/3} + 4\sum_{i=0}^{2}\left(1+\left(2+\frac{2i+1}{3}\right)\right)^{1/3} + (5)^{1/3}\right)\\ &\approx 3.16734727452 .\end{aligned}$$

Question 12.9
Before answering this question, we note that $f'(x) = 2x\ln(x)+x$, $f''(x) = 2\ln(x)+3$, $f^{(3)}(x) = 2/x$ and $f^{(4)}(x) = -2/x^2$.

Moreover, a simple integration by parts gives
$$\int_1^3 x^2\ln(x)\,\mathrm{d}x = 9\ln(3) - \frac{26}{9} \approx 6.99862170912 .$$

We have $a = 1$ and $b = 3$ in the formulae for the truncation errors of the composite methods.

a) For the midpoint rule, we choose $n = 2m$ and $h = (b-a)/n = 1/m$ such that the truncation error $\left|\dfrac{f''(\eta)(b-a)}{6}h^2\right|$ for some $\eta \in [1,3]$ satisfies
$$\left|\frac{f''(\eta)(b-a)}{6}h^2\right| = \frac{1}{3}\left(\frac{1}{m}\right)^2|2\ln(\eta)+3| \leq \frac{1}{3m^2}|2\ln(3)+3| < 10^{-5} .$$
Thus,
$$m > \left(\frac{10^5}{3}(2\ln(3)+3)\right)^{1/2} \approx 416.22 .$$
We take $m = 417$. It follows that $h = 1/417$ and
$$\int_1^3 x^2\ln(x)\,\mathrm{d}x \approx \frac{2}{417}\sum_{j=1}^{417} f(x_{2j-1}) \approx 6.99861347 ,$$

where $x_j = 1 + j\,h = 1 + j/417$. The absolute error is about $0.82348266 \times 10^{-5}$.

b) For the trapezoidal rule, we choose n and $h = (b-a)/n = 2/n$ such that the truncation error $\left|\dfrac{f''(\eta)(b-a)}{12}\,h^2\right|$ for some $\eta \in [1,3]$ satisfies

$$\left|\frac{f''(\eta)(b-a)}{12}\,h^2\right| = \frac{1}{6}\left(\frac{2}{n}\right)^2 |2\ln(\eta)+3| \leq \frac{2}{3n^2}|2\ln(3)+3| < 10^{-5} \ .$$

Thus,

$$n > \left(\frac{2\times 10^5}{3}\,(2\ln(3)+3)\right)^{1/2} \approx 588.6269 \ .$$

We take $n = 589$. It follows that $h = 2/589$ and

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x \approx \frac{1}{589}\left(f(x_0) + 2\sum_{j=1}^{588} f(x_j) + f(x_{589})\right) \approx 6.99862996 \ ,$$

where $x_j = 1 + j\,h = 1 + 2j/589$. The absolute error is about $0.82551686 \times 10^{-5}$.

c) For the Simpson rule, we choose $n = 2m$ and $h = (b-a)/n = 1/m$ such that the truncation error $\left|\dfrac{f^{(4)}(\eta)(b-a)}{180}\,h^4\right|$ for some $\eta \in [1,3]$ satisfies

$$\left|\frac{f^{(4)}(\eta)(b-a)}{180}\,h^4\right| = \frac{1}{90}\left(\frac{1}{m}\right)^4 \left|\frac{-2}{\eta^2}\right| \leq \frac{1}{45\,m^4} < 10^{-5} \ .$$

Thus,

$$m > \left(\frac{10^5}{45}\right)^{1/4} \approx 6.86589 \ .$$

We take $m = 7$. It follows that $h = 1/7$ and

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x \approx \frac{1}{21}\left(f(x_0) + 2\sum_{j=1}^{6} f(x_{2j}) + 4\sum_{j=0}^{6} f(x_{2j+1}) + f(x_{14})\right) \approx 6.99861865 \ ,$$

where $x_j = 1 + j\,h = 1 + j/7$. The absolute error is about $0.30640240 \times 10^{-5}$.

Question 12.10

There are two ways to answer this question. We could use the formula

$$\int_a^b f(x)\,\mathrm{d}x = \frac{f(a) + 4f((a+b)/2) + f(b)}{6}\,(b-a) - \frac{f^{(4)}(\xi)\,(b-a)^5}{2880}$$

for some ξ between a and b. The truncation error will be 0 for all polynomials of degree less than 4 becuase $f^{(4)}(x) = 0$ for all polynomials f of degree less than 4.

The other way to answer the question is to proceed directly. By linearity of the integral, it is enough to show that Simpson's rule is exact for x^i with $i = 0$, 1, 2 and 3. For $f(x) = x^i$, Simpson's rule becomes

$$\int_a^b x^i\,\mathrm{d}x \approx \frac{b-a}{6}\left(a^i + 4\left(\frac{a+b}{2}\right)^i + b^i\right) \ . \tag{16.17}$$

For $i = 0$, the right hand side of (16.17) is

$$\frac{b-a}{6}\left(1+4+1\right) = b-a = \int_a^b \,\mathrm{d}x \ .$$

For $i = 1$, the right hand side of (16.17) is

$$\frac{b-a}{6}\left(a + 4\left(\frac{a+b}{2}\right) + b\right) = \frac{b-a}{6}\left(3a+3b\right) = \frac{1}{2}\left(b^2 - a^2\right) = \int_a^b x\,\mathrm{d}x \ .$$

For $i = 2$, the right hand side of (16.17) is

$$\frac{b-a}{6}\left(a^2 + 4\left(\frac{a+b}{2}\right)^2 + b^2\right) = \frac{b-a}{6}\left(2a^2+2ab+2b^2\right) = \frac{1}{3}\left(b^3 - a^3\right) = \int_a^b x^2\,\mathrm{d}x \ .$$

Finally, for $i = 3$, the right hand side of (16.17) is

$$\begin{aligned}\frac{b-a}{6}\left(a^3 + 4\left(\frac{a+b}{2}\right)^3 + b^3\right) &= \frac{b-a}{6}\left(a^3 + \frac{1}{2}\left(a^3+3a^2b+3ab^3+b^3\right) + b^3\right) \\ &= \frac{b-a}{12}\left(3a^3+3a^2b+3ab^2+3b^3\right) = \frac{1}{4}\left(b^4 - a^4\right) = \int_a^b x^3\,\mathrm{d}x \ .\end{aligned}$$

Thus, Simpson's rule is exact for polynomial of degree up to three.

However, for $i = 4$, the right hand side of (16.17) is

$$\begin{aligned}\frac{b-a}{6}\left(a^4 + 4\left(\frac{a+b}{2}\right)^4 + b^4\right) &= \frac{b-a}{6}\left(a^4 + \frac{1}{4}\left(a^4+4a^3b+6a^2b^2+4ab^4+b^4\right) + b^4\right) \\ &= \frac{b-a}{24}\left(5a^4+4a^3b+6a^2b^2+4ab^3+5b^4\right) \\ &= \frac{1}{24}\left(5b^5 - ab^4 + 2a^2b^3 - 2a^3b^2 + a^4b - 5a^5\right) \\ &\neq \frac{1}{5}\left(b^5 - a^5\right) = \int_a^b x^4\,\mathrm{d}x\end{aligned}$$

for almost all values of a and b.

Question 12.11

We use the composite trapezoidal rule to generate the first column of the table for Romberg integration.

For $n = 2$, we have $h = (4-2)/2 = 1$ and $x_j = 2 + j\,h = 2 + j$ for $0 \leq j \leq 2$. Hence,

$$\int_2^4 (1+x)^{1/3}\,\mathrm{d}x \approx \frac{1}{2}\left(3^{1/3} + 2\left(4^{1/3}\right) + 5^{1/3}\right) \approx 3.163513810460252 \ .$$

For $n = 4$, we have $h = (4-2)/4 = 1/2$ and $x_j = 2 + j\,h = 2 + j/2$ for $0 \leq j \leq 4$. Hence,

$$\int_2^4 (1+x)^{1/3}\,\mathrm{d}x \approx \frac{1}{4}\left(3^{1/3} + 2\left(3.5^{1/3} + 4^{1/3} + 4.5^{1/3}\right) + 5^{1/3}\right) \approx 3.166385960422699 \ .$$

For $n = 8$, we have $h = (4-2)/8 = 1/4$ and $x_j = 2 + j\,h = 2 + j/4$ for $0 \le j \le 8$. Hence,

$$\int_2^4 (1+x)^{1/3}\,\mathrm{d}x \approx \frac{1}{8}\left(3^{1/3} + 2\sum_{j=1}^{7}(1+2+j/4)^{1/3} + 5^{1/3}\right) \approx 3.167107453016058\ .$$

All values in the table below have been rounded to nine digits.

n	h	$L_h^0(f)$		$L_h^1(f)$	$L_h^2(f)$
2	1	3.16351381			
4	1/2	3.16638596	$\boxed{3.98084469}$	3.16734334	
8	1/4	3.16710745		3.16734795	3.16734826

Note that $L_{1/2}^1(f) = (4L_{1/2}^0(f) - L_1^0(f))/(4-1)$, $L_{1/4}^1(f) = (4L_{1/4}^0(f) - L_{1/2}^0(f))/(4-1)$ and $L_{1/4}^2(f) = (4^2 L_{1/4}^1(f) - L_{1/2}^1(f))/(4^2-1)$.

The requested approximation is 3.16734826 because $\left|L_{1/2}^1(f) - L_1^0(f)\right| \approx 0.0038295 > 10^{-5}$ and $\left|L_{1/4}^2(f) - L_{1/2}^1(f)\right| \approx 0,49139 \times 10^{-5} < 10^{-5}$.

Question 12.12

We have used the composite trapezoidal rule to generate the first column of the table for Romberg integration; namely, we have used

$$L_h^0(f) = \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right)\ ,$$

where $x_i = 1 + ih$ and $h = 2/n$.

h	$L_h^0(f)$		$L_h^1(f)$		$L_h^2(f)$		$L_h^3(f)$	$L_h^4(f)$	$\lvert L_h^i(f) - L_{2h}^{i-1}(f)\rvert$
2	2.31607401								
1	2.34724412	3.69506105	2.35763416						0.0415601
0.5	2.35567974	3.88972370	2.35849161	10.75596733	2.35854877				0.914612×10^{-3}
0.25	2.35784843	$\boxed{3.96766871}$	2.35857133	13.53301479	2.35857664	28.76692226	2.35857708		0.283121×10^{-4}
0.125	2.35839502		2.35857722		2.35857761		2.35857762	2.35857763	0.543938×10^{-6}

All the values in the table have been rounded.

We stopped the procedure when $|L_h^i(f) - L_{2h}^{i-1}(f)|$ got smaller than 10^{-5} and took $L_h^i(f)$ as our approximation of the integral. We have that

$$\int_3^5 (x-2)^{1/4}\,\mathrm{d}x \approx L_{0.125}^4 \approx 2.35857763$$

with the required accuracy.

We have also included the ratios defined in (12.2.10) to check if the values of $L_h^i(h)$ can be trusted. However, we have ignored this information.

Question 12.13

The first column of table used for Romberg integration is produced by the composite trapezoidal rule

$$L_h^0(f) = \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) ,$$

where $x_i = 1 + ih$ and $h = 2/n$. The table is given below.

h	$L_h^0(f)$		$L_h^1(f)$		$L_h^2(f)$
2	9.88751060				
1	7.71634402	$\boxed{4.03101596}$	6.99262183		
0.5	7.17772879	$\boxed{4.00899805}$	6.99819039	13.81887982	6.99856162
0.25	7.04337721	$\boxed{4.00237347}$	6.99859335	$\boxed{15.17336693}$	6.99862022
0.125	7.00980924		6.99861991		6.99862168

	$L_h^3(f)$	$L_h^4(f)$	$\lvert L_h^i(f) - L_{2h}^{i-1}(f)\rvert$
			2.8948888
			0.005939793
40.03582483	6.99862115		$0.59524745 \times 10^{-4}$
	6.99862170	6.99862171	$0.55889607 \times 10^{-5}$

All the values in the table have been rounded. We stopped the procedure when $|L_h^i(f) - L_{2h}^{i-1}(f)|$ got smaller than 10^{-7} and took $L_h^i(f)$ as our approximation of the integral. We have that

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x \approx L_{0.125}^4 \approx 6.99862171$$

meet our criteria of accuracy.

In question 12.9, we found that

$$\int_1^3 x^2 \ln(x)\,\mathrm{d}x = 9\ln(3) - \frac{26}{9} = 6.99862170912\ldots$$

So, the absolute error of our approximation 6.99862171 is about -0.88×10^{-9}. This is better than expected.

We have also included the ratios defined in (12.2.10) to check if the values of $L_h^i(h)$ can be trusted. However, we have ignored this information. The approximation of the value of the integral suggested by the ratios defined in (12.2.10) is $L_{0.25}^2(f) \approx 6.99862022$ associated to the ratio 15.17336693. The absolute error for this approximation is about 0.14891×10^{-5}. Despite the fact that the ratios were not respecting the rule of 4^i, we did the right thing by proceeding with the computations. The problem with the computations of the ratios defined in (12.2.10) is that they have to be done with very high precision because we are dividing by values closed to zero.

Question 12.14

The first column of the table associated to Romberg integration is given by the composite trapezoidal rule

$$L_h(f) = \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right) ,$$

where $h = (b-a)/n$ and $x_i = a + i\,h$ for $i = 1, 2, 3, \ldots, n$.

The second column of this table is

$$\begin{aligned} L^1_{h/2}(f) = \frac{4L_{h/2}(f) - L_h(f)}{4-1} &= \frac{1}{3}\Bigg(h\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + 2\sum_{j=0}^{n-1} f(x_j + h/2) + f(x_n)\right) \\ &\qquad - \frac{h}{2}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right)\Bigg) \\ &= \frac{h}{6}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + 4\sum_{j=0}^{n-1} f(x_j + h/2) + f(x_n)\right) \\ &= \frac{\tilde{h}}{3}\left(f(\tilde{x}_0) + 2\sum_{j=1}^{n-1} f(\tilde{x}_{2j}) + 4\sum_{j=0}^{n-1} f(\tilde{x}_{2j+1}) + f(\tilde{x}_n)\right) , \end{aligned}$$

where $\tilde{h} = (b-a)/(2n)$ and $\tilde{x}_j = a + j\tilde{h}$ for $0 \le j \le 2n$. This is the composite Simpson rule with m replaced by n, h by $\tilde{h}$ and x_j by $\tilde{x}_j$.

Question 12.15

Use the adaptive quadrature method presented in Section 12.6 to approximate

$$\int_3^5 (x-2)^{1/4}\,\mathrm{d}x$$

with an accuracy of 10^{-5}. For this purpose and to simplify the discussion, let $S(a,b,h)$ be the result of the composite Simpson's Rule, Theorems 12.3.4 and 12.4.4, for $\int_a^b (x-2)^{1/4}\,\mathrm{d}x$ with $m = (b-a)/(2h)$. The values displayed in the following computations have been rounded to 12 significant digits though the computations have been done with as many digits as possible.

level 0:

i	1	2	3	4	5
x_i	3	7/2	4	9/2	5

$h = 1$, $T = 10^{-5}$, $S_{[3,5]} = S(3,5,1) \approx 2.35763415765$,
$S_1 = S(3,4,0.5) \approx 1.10265579897$, $S_2 = S(4,5,0.5) \approx 1.25583580778$,
$\tilde{S}_{[3,5]} = S(3,5,0.5) \approx S_1 + S_2 = 2.35849160675$ and
$\tilde{R}_{[3,5]} \approx \dfrac{1}{15}\left|\tilde{S}_{[3,5]} - S_{[3,5]}\right| \approx 0.571633 \times 10^{-4} \not< 10^{-5}$.

Level 1:

i	1	2	3	4	5
x_i	3	13/4	7/2	15/4	4

$h = 0.5$, $T = 0.5 \times 10^{-5}$ (store for $[4,5]$), $S_{[3,4]} = S(3,4,0.5) \approx 1.10265579897$,
$S_1 = S(3,3.5,0.25) \approx 0.528013914455$, $S_2 = S(3.5,4,0.25) \approx 0.574711858524$,

$\tilde{S}_{[3,4]} = S(3,4,0.25) = S_1 + S_2 \approx 1.10272577298$ and
$\tilde{R}_{[3,4]} \approx \dfrac{1}{15}\left|\tilde{S}_{[3,4]} - S_{[3,4]}\right| \approx 0.466493 \times 10^{-5} < 0.5 \times 10^{-5}$.
We accept $\tilde{S}_{[3,4]} \approx 1.10272577298$ as an approximation of $\displaystyle\int_3^4 (x-2)^{1/4}\,dx$.

Level 1:

i	1	2	3	4	5
x_i	4	17/4	9/2	19/4	5

$h = 0.5$, $T = 0.5 \times 10^{-5}$, $S_{[4,5]} = S(4,5,0.5) \approx 1.25583580778$,
$S_1 = S(4,4.5,0.25) \approx 0.612135002521$, $S_2 = S(4.5,5,0.25) \approx 0.643710549703$,
$\tilde{S}_{[4,5]} = S(4,5,0.25) = S_1 + S_2 \approx 1.25584555222$ and
$\tilde{R}_{[4,5]} \approx \dfrac{1}{15}\left|\tilde{S}_{[4,5]} - S_{[4,5]}\right| \approx 0.649630 \times 10^{-6} < 0.5 \times 10^{-5}$.
We accept $\tilde{S}_{[4,5]} \approx 1.25584555222$ as an approximation of $\displaystyle\int_4^5 (x-2)^{1/4}\,dx$.

Level 0: We are done. We have found that

$$\begin{aligned}\int_3^5 (x-2)^{1/4}\,dx &= \int_3^4 (x-2)^{1/4} + \int_4^5 (x-2)^{1/4} \\ &\approx 1.10272577298 + 1.25584555222 = 2.35857132520\ .\end{aligned}$$

This approximation is meeting the required accuracy.

Question 12.16
Since (12.9.5) is a linear functional with respect to f and since all polynomial of degree at most 4 are linear combinations of the monomials x^i for $0 \leq i \leq 4$, it is enough to show that (12.9.5) is true for $f(x) = x^i$ with $0 \leq i \leq 4$ to prove that (12.9.5) is true for all polynomials of degree less than or equal to 4. For instance, for $i = 4$, we get

$$\int_0^1 x^4\,dx = \left.\frac{x^5}{5}\right|_{x=0}^1 = \frac{1}{5}$$

and

$$\frac{1}{90}\left(7(0^4) + 32\left(\frac{1}{4}\right)^4 + 12\left(\frac{1}{2}\right)^4 + 32\left(\frac{3}{4}\right)^4 + 7(1^4)\right) = \frac{1}{5}\ .$$

We leave to the reader the task of verifying the equality for the other monomials.

If we substitute $y = (b-a)x + a$ in $\displaystyle\int_a^b f(y)\,dy$, we get

$$\begin{aligned}\int_a^b f(y)\,dy &= \int_0^1 f\big((b-a)x+a\big)\,(b-a)\,dx \\ &= \frac{b-a}{90}\left(7\,f(a) + 32\,f\left(\frac{3a+b}{4}\right) + 12\,f\left(\frac{a+b}{2}\right) + 32\,f\left(\frac{a+3b}{4}\right) + 7\,f(b)\right).\end{aligned}$$

Question 12.17
For $f(x) = x^i$, the formula is

$$\int_0^1 x^i\,dx \approx A\left(x_0^i + x_1^i\right)\ .$$

For $i = 0$, we have $1 = \int_0^1 \mathrm{d}x = 2\,A$. Thus, $A = 1/2$. For $i = 1$, we have $\frac{1}{2} = \int_0^1 x\,\mathrm{d}x = A\,(x_0 + x_1) = \frac{1}{2}(x_0 - x_1)$. Thus

$$1 = x_0 + x_1 \ . \tag{16.18}$$

For $i = 2$, we have $\frac{1}{3} = \int_0^1 x^2\,\mathrm{d}x = A\,(x_0^2 + x_1^2) = \frac{1}{2}(x_0^2 + x_1^2)$. Thus,

$$\frac{2}{3} = x_0^2 + x_1^2 \ . \tag{16.19}$$

If we solve the system of equations given by (16.18) and (16.19) for x_0 and x_1, we get two solutions $x_0 = \frac{1}{2} + \frac{1}{2\sqrt{3}}$ and $x_1 = \frac{1}{2} - \frac{1}{2\sqrt{3}}$, and $x_0 = \frac{1}{2} - \frac{1}{2\sqrt{3}}$ and $x_1 = \frac{1}{2} + \frac{1}{2\sqrt{3}}$. Hence, we find that the formula

$$\int_0^1 x^i\,\mathrm{d}x \approx \frac{1}{2}\left(f\left(\frac{1}{2} + \frac{1}{2\sqrt{3}}\right) + f\left(\frac{1}{2} - \frac{1}{2\sqrt{3}}\right)\right)$$

is exact for polynomials of degree less or equal to 2.

Question 12.18

For $f(x) = x^i$, the formula is

$$\int_0^2 x^{i+1}\,\mathrm{d}x \approx A\,x^i\Big|_{x=0} + B\,x^i\Big|_{x=1} + C\,x^i\Big|_{x=2}$$

for $i \geq 0$. For $i = 0$, we get

$$2 = \int_0^2 x\,\mathrm{d}x = A + B + C \ . \tag{16.20}$$

For $i = 1$, we get

$$\frac{8}{3} = \int_0^2 x^2\,\mathrm{d}x = B + 2\,C \ . \tag{16.21}$$

For $i = 2$, we get

$$4 = \int_0^2 x^3\,\mathrm{d}x = B + 4\,C \ . \tag{16.22}$$

The system of linear equations formed of (16.20), (16.21) and (16.22) has a unique solution given by $A = 0$, $B = 4/3$ and $C = 2/3$.

For $i = 3$, we have

$$\frac{32}{5} = \int_0^2 x^4\,\mathrm{d}x = B + 8\,C \ .$$

Since this equation is not satisfied with $B = 4/3$ and $C = 2/3$, the formula is not valid for $i = 3$. Therefore, we can get a formula which is exact for polynomials of degree up to two with $A = 0$, $B = 4/3$ and $C = 2/3$. It is not possible to do better.

Question 12.19

We get $A + B = \int_0^{2\pi} \mathrm{d}x = 2\pi$ when $k = 0$ and $A\cos(0) + B\cos(\pi) = \int_0^{2\pi} \cos(x)\,\mathrm{d}x =$

$\sin(x)\Big|_{x=0}^{2\pi} = 0$ when $k = 1$. Thus $A + B = 2\pi$ and $A - B = 0$. We find $A = B = \pi$. The requested formula is

$$\int_0^{2\pi} f(x)\,\mathrm{d}x \approx \pi\, f(0) + \pi\, f(\pi)\ . \tag{16.23}$$

Since

$$\int_0^{2\pi} \cos((2k+1)x)\,\mathrm{d}x = \frac{1}{2k+1}\sin((2k+1)x)\Big|_{x=0}^{2\pi} = 0$$

and

$$\pi\cos((2k+1)x)\Big|_{x=0} + \pi\cos((2k+1)x)\Big|_{x=\pi} = \pi + \pi\cos((2k-1)\pi) = \pi - \pi = 0$$

for all $k \geq 0$, (16.23) is exact for $f(x) = \cos((2k+1)x)$ with $k \geq 0$. Moreover, since

$$\int_0^{2\pi} \sin(kx)\,\mathrm{d}x = -\frac{1}{k}\cos(kx)\Big|_{x=0}^{2\pi} = -\frac{1}{k}\left(\cos(2\pi k) - 1\right) = 0$$

and

$$\pi\sin(kx)\Big|_{x=0} + \pi\sin(kx)\Big|_{x=\pi} = 0 + \pi\sin(k\pi) = 0 - 0 = 0$$

for $k > 0$, (16.23) is exact for $f(x) = \sin(kx)$ with $k > 0$. It is also true for $f(x) = \sin(kx)$ with $k = 0$ because $f(x) = 0$ for all x in this case.

By linearity of the integral, (16.23) is exact for expressions of the form (12.9.6). It is not a really interesting relation because the integral of (12.9.6) in (16.23) is null and therefore does not require the formula $\pi f(0) + \pi f(\pi)$ to be computed.

Note: Formula (16.23) is not true for $f(x) = \cos(2k\,x)$ with $k \geq 1$. This can be shown directly. Another way to prove this is by contradiction. Suppose that the formula is true for $f(x) = \cos(2k\,x)$ with $k \geq 1$. Since all continuous functions on $[0, 2\pi]$ can be expressed as a Fourier series of $\cos(kx)$ and $\sin(kx)$ for $k \geq 0$, (16.23) will then be true for all continuous functions on $[0, 2\pi]$. But the formula is not true for $f(x) = x$.

Question 12.20

The polynomial interpolation of f at the points $x_0 = a + (b-a)/3 = (2a+b)/3$ and $x_1 = a + 2(b-a)/3 = (a+2b)/3$ is given by

$$f(x) = f[x_0] + f[x_0, x_1]\,(x - x_0) + f[x_0, x_1, x]\,(x - x_0)\,(x - x_1)\ .$$

Hence,

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b \left(f[x_0] + f[x_0,x_1]\,(x - x_0)\right)\mathrm{d}x + \int_a^b f[x_0, x_1, x]\,(x - x_0)\,(x - x_1)\,\mathrm{d}x\ .$$

Since $f[x_0] = f(x_0) = f\left(\dfrac{2a+b}{2}\right)$ and

$$f[x_0, x_1] = \frac{f\big((a+2b)/3\big) - f\big((2a+b)/3\big)}{(a+2b)/3 - (2a+b)/3} = \frac{3}{b-a}\left(f\left(\frac{a+2b}{3}\right) - f\left(\frac{2a+b}{3}\right)\right)\ ,$$

we get the formula

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \int_a^b \left(f[x_0] + f[x_0,x_1]\,(x-x_0)\right)\,\mathrm{d}x = \left(f(x_0)\,x + \frac{1}{2}\,f[x_),x_1]\,\left(x-x_0\right)^2\right)\Bigg|_{x=a}^{b} \\ &= f\left(\frac{2a+b}{3}\right)(b-a) + \frac{3}{2(b-a)}\left(f\left(\frac{a+2b}{3}\right) - f\left(\frac{2a+b}{3}\right)\right)\left(\left(b - \frac{2a+b}{3}\right)^2 - \left(a - \frac{2a+b}{3}\right)^2\right) \\ &= \frac{b-a}{2}\left(f\left(\frac{a+2b}{3}\right) + f\left(\frac{2a+b}{3}\right)\right)\ .\end{aligned}$$

We choose $A = B = (b-a)/2$.

For each $x \in [a,b]$, there exists $\xi \in [a,b]$ such that

$$|f[x_0,x_1,x]| = \left|\frac{1}{2}\,f''(\xi)\right| < \frac{M}{2}\ .$$

Thus,

$$\begin{aligned}\left|\int_a^b f[x_0,x_1,x]\,(x-x_0)\,(x-x_1)\,\mathrm{d}x\right| &\leq \int_a^b |f[x_0,x_1,x]|\,|(x-x_0)\,(x-x_1)|\,\mathrm{d}x \\ &\leq \frac{M}{2}\int_a^b |(x-x_0)\,(x-x_1)|\,\mathrm{d}x\ .\end{aligned}$$

To compute this integral, we split the interval of integration in three subintervals such that the sign of the integrand is constant on each subinterval. We can then eliminate the absolute value. To simplify the computation, we use integration by parts to get

$$\int (x-x_0)(x-x_1)\,\mathrm{d}x = \frac{(x-x_0)(x-x_1)^2}{2} - \frac{(x-x_1)^3}{6} + C_1$$

and

$$\int (x-x_0)(x-x_1)\,\mathrm{d}x = \frac{(x-x_1)(x-x_0)^2}{2} - \frac{(x-x_0)^3}{6} + C_2$$

for some constants C_1 and C_2. It is interesting to note that if we subtract the second integral from the first integral, we find that $C_2 - C_1 = (x_1-x_0)^3/6$.

$$\begin{aligned}&\int_a^b |(x-x_0)\,(x-x_1)|\,\mathrm{d}x \\ &\quad = \int_a^{x_0} (x-x_0)\,(x-x_1)\,\mathrm{d}x - \int_{x_0}^{x_1} (x-x_0)\,(x-x_1)\,\mathrm{d}x + \int_{x_1}^{b} (x-x_0)\,(x-x_1)\,\mathrm{d}x \\ &\quad = \left(\frac{(x-x_1)(x-x_0)^2}{2} - \frac{(x-x_0)^3}{6}\right)\Bigg|_{x=a}^{x_0} - \left(\frac{(x-x_1)(x-x_0)^2}{2} - \frac{(x-x_0)^3}{6}\right)\Bigg|_{x=x_0}^{x_1} \\ &\qquad + \left(\frac{(x-x_0)(x-x_1)^2}{2} - \frac{(x-x_1)^3}{6}\right)\Bigg|_{x=x_1}^{b} \\ &\quad = -\left(\frac{(a-x_1)(a-x_0)^2}{2} - \frac{(a-x_0)^3}{6}\right) + \frac{(x_1-x_0)^3}{6} + \left(\frac{(b-x_0)(b-x_1)^2}{2} - \frac{(b-x_1)^3}{6}\right)\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2}\left(\frac{2(a-b)}{3}\right)\left(\frac{a-b}{3}\right)^2 + \frac{1}{6}\left(\frac{a-b}{3}\right)^3 + \frac{1}{6}\left(\frac{b-a}{3}\right)^3 + \frac{1}{2}\left(\frac{2(b-a)}{3}\right)\left(\frac{b-a}{3}\right)^2 - \frac{1}{6}\left(\frac{b-a}{3}\right)^3 \\
&= \frac{11(b-a)^3}{162} \ .
\end{aligned}
$$

Hence,

$$
\left| \int_a^b f[x_0,x_1,x]\,(x-x_0)\,(x-x_1)\ \mathrm{d}x \right| \le \frac{11M(b-a)^3}{162} \ .
$$

Question 12.21

The polynomial interpolation of f (of degree at most 2) at the points $x_0 = a + (b-a)/4 = (3a+b)/4$, $x_1 = a + (b-a)/2 = (a+b)/2$ and $x_2 = a + 3(b-a)/4 = (a+3b)/4$ is given by

$$
\begin{aligned}
f(x) &= f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1) \\
&\quad + f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2) \ .
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\int_a^b f(x)\,\mathrm{d}x &= \int_a^b \left(f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1)\right)\,\mathrm{d}x \\
&\quad + \int_a^b f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x \ .
\end{aligned}
\tag{16.24}
$$

The integration formula is given by the first integral on the right side of (16.24). Since

$$
\begin{aligned}
f[x_0] &= f\left(\frac{3a+b}{4}\right) \ , \\
f[x_0,x_1] &= \frac{f(x_1)-f(x_0)}{x_1-x_0} = \left(\frac{1}{(b+a)/2-(3a+b)/4}\right)\left(f\left(\frac{b+a}{2}\right) - f\left(\frac{3a+b}{4}\right)\right) \\
&= \frac{4}{b-a}\left(f\left(\frac{b+a}{2}\right) - f\left(\frac{3a+b}{4}\right)\right) \ , \\
f[x_1,x_2] &= \frac{f(x_2)-f(x_1)}{x_2-x_1} = \left(\frac{1}{(a+3b)/4-(b+a)/2}\right)\left(f\left(\frac{a+3b}{4}\right) - f\left(\frac{b+a}{2}\right)\right) \\
&= \frac{4}{b-a}\left(f\left(\frac{a+3b}{4}\right) - f\left(\frac{b+a}{2}\right)\right)
\end{aligned}
$$

and

$$
\begin{aligned}
f[x_0,x_1,x_2] &= \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} \\
&= \left(\frac{1}{(a+3b)/4-(3a+b)/4}\right)\left(\frac{4}{b-a}\left(f\left(\frac{a+3b}{4}\right) - f\left(\frac{b+a}{2}\right)\right)\right. \\
&\qquad \left. - \frac{4}{b-a}\left(f\left(\frac{b+a}{2}\right) - f\left(\frac{3a+b}{4}\right)\right)\right) \\
&= \frac{8}{(b-a)^2}\left(f\left(\frac{a+3b}{4}\right) - 2f\left(\frac{b+a}{2}\right) + f\left(\frac{3a+b}{4}\right)\right) \ ,
\end{aligned}
$$

we get

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \int_a^b \big(f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1)\big)\,\mathrm{d}x \\ &= f\left(\frac{3a+b}{4}\right)\int_a^b \mathrm{d}x + \frac{4}{b-a}\left(f\left(\frac{b+a}{2}\right) - f\left(\frac{3a+b}{4}\right)\right)\int_a^b \left(x - \frac{3a+b}{4}\right)\mathrm{d}x \\ &\quad + \frac{8}{(b-a)^2}\left(f\left(\frac{a+3b}{4}\right) - 2\,f\left(\frac{b+a}{2}\right) + f\left(\frac{3a+b}{4}\right)\right)\int_a^b \left(x - \frac{3a+b}{4}\right)\left(x - \frac{a+b}{2}\right)\mathrm{d}x\ .\end{aligned}$$

Moreover, since

$$\int_a^b \left(x - \frac{3a+b}{4}\right)\mathrm{d}x = \frac{1}{2}\left(x - \frac{3a+b}{4}\right)^2\bigg|_{x=a}^{b} = \frac{1}{4}\,(b-a)^2$$

and

$$\begin{aligned}\int_a^b \left(x - \frac{3a+b}{4}\right)\left(x - \frac{a+b}{2}\right)\mathrm{d}x &= \int_a^b \left(x^2 - \frac{5a+3b}{4}\,x + \frac{3a^2+4ab+b^2}{8}\right)\mathrm{d}x \\ &= \left(\frac{x^3}{3} - \frac{5a+3b}{8}\,x^2 + \frac{3a^2+4ab+b^2}{8}\,x\right)\bigg|_{x=a}^{b} = \frac{1}{12}\,(b-a)^3\ ,\end{aligned}$$

we finally get

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \int_a^b \big(f[x_0] + f[x_0,x_1]\,(x-x_0) + f[x_0,x_1,x_2]\,(x-x_0)(x-x_1)\big)\,\mathrm{d}x \\ &= f\left(\frac{3a+b}{4}\right)(b-a) + \left(f\left(\frac{b+a}{2}\right) - f\left(\frac{3a+b}{4}\right)\right)(b-a) \\ &\quad + \frac{2}{3}\left(f\left(\frac{a+3b}{4}\right) - 2\,f\left(\frac{b+a}{2}\right) + f\left(\frac{3a+b}{4}\right)\right)(b-a) \\ &= \left(\frac{2}{3}\,f\left(\frac{a+3b}{4}\right) - \frac{1}{3}\,f\left(\frac{b+a}{2}\right) + \frac{2}{3}\,f\left(\frac{3a+b}{4}\right)\right)(b-a)\ .\end{aligned}$$

The truncation error is given by the second integral on the right hand side of (16.24). Since, for each $x \in [a,b]$ there exists $\xi \in [a,b]$ such that

$$|f[x_0,x_1,x_2,x]| = \left|\frac{1}{3!}\,f^{(3)}(\xi)\right| < \frac{M}{3!}\ ,$$

we get

$$\begin{aligned}&\left|\int_a^b \big(f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2)\big)\,\mathrm{d}x\right| \\ &\quad \leq \int_a^b |f[x_0,x_1,x_2,x]|\,|(x-x_0)(x-x_1)(x-x_2)|\,\mathrm{d}x \leq \frac{M}{6}\int_a^b |(x-x_0)(x-x_1)(x-x_2)|\,\mathrm{d}x\ .\end{aligned}$$

Because of the symmetry of the graph of $p(x) = |(x-x_0)(x-x_1)(x-x_2)|$ with respect to the vertical line $x = (a+b)/2$, we have

$$\int_a^b |(x-x_0)(x-x_1)(x-x_2)|\,\mathrm{d}x = 2\int_{x_1}^b |(x-x_0)(x-x_1)(x-x_2)|\,\mathrm{d}x$$

$$= -2\int_{x_1}^{x_2} (x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x + 2\int_{x_2}^{b} (x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x \ .$$

Using integration by parts with $u(x) = (x-x_0)(x-x_1)$ and $v'(x) = x - x_2$, we get $u'(x) = (x-x_0) + (x-x_1)$, $v(x) = (x-x_2)^2/2$ and

$$\begin{aligned}\int (x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x &= u(x)v(x) - \int u'(x)v(x)\,\mathrm{d}x \\ &= \frac{1}{2}(x-x_0)(x-x_1)(x-x_2)^2 - \frac{1}{2}\int (x-x_0)(x-x_2)^2\,\mathrm{d}x - \frac{1}{2}\int (x-x_1)(x-x_2)^2\,\mathrm{d}x \ .\end{aligned}$$

Again, using integration by parts with $u(x) = (x-x_0)$ and $v'(x) = (x-x_2)^2$ for the first integral on the right hand side of the equation above, and $u(x) = (x-x_0)$ and $v'(x) = (x-x_2)^2$ for the second integral on the right hand side of the equation above, we get

$$\begin{aligned}&\int (x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x \\ &\quad = \frac{1}{2}(x-x_0)(x-x_1)(x-x_2)^2 - \frac{1}{2}\left(\frac{1}{3}(x-x_0)(x-x_2)^3 - \frac{1}{12}(x-x_2)^4\right) \\ &\qquad\quad - \frac{1}{2}\left(\frac{1}{3}(x-x_1)(x-x_2)^3 - \frac{1}{12}(x-x_2)^4\right) + C \\ &\quad = \frac{1}{2}(x-x_0)(x-x_1)(x-x_2)^2 - \frac{1}{6}(x-x_0)(x-x_2)^3 - \frac{1}{6}(x-x_1)(x-x_2)^3 + \frac{1}{12}(x-x_2)^4 + C \ .\end{aligned}$$

Hence

$$\begin{aligned}\int_a^b |(x-x_0)(x-x_1)(x-x_2)|\,\mathrm{d}x &= -2\Bigg(\frac{1}{2}(x-x_0)(x-x_1)(x-x_2)^2 - \frac{1}{6}(x-x_0)(x-x_2)^3 \\ &\qquad - \frac{1}{6}(x-x_1)(x-x_2)^3 + \frac{1}{12}(x-x_2)^4\Bigg)\Bigg|_{x=x_1}^{x_2} + 2\Bigg(\frac{1}{2}(x-x_0)(x-x_1)(x-x_2)^2 \\ &\qquad - \frac{1}{6}(x-x_0)(x-x_2)^3 - \frac{1}{6}(x-x_1)(x-x_2)^3 + \frac{1}{12}(x-x_2)^4\Bigg)\Bigg|_{x=x_2}^{b} \\ &= 2\left(-\frac{1}{6}(x_1-x_0)(x_1-x_2)^3 + \frac{1}{12}(x_1-x_2)^4\right) + 2\Bigg(\frac{1}{2}(b-x_0)(b-x_1)(b-x_2)^2 \\ &\qquad - \frac{1}{6}(b-x_0)(b-x_2)^3 - \frac{1}{6}(b-x_1)(b-x_2)^3 + \frac{1}{12}(b-x_2)^4\Bigg) \\ &= 2\left(\frac{1}{4}\left(\frac{b-a}{4}\right)^4\right) + 2\left(\frac{9}{4}\left(\frac{b-a}{4}\right)^4\right) = 5\left(\frac{b-a}{4}\right)^4 \ .\end{aligned}$$

Therefore,

$$\left|\int_a^b \left(f[x_0,x_1,x_2,x]\,(x-x_0)(x-x_1)(x-x_2)\right)\,\mathrm{d}x\right| \leq \frac{5M}{6}\left(\frac{b-a}{4}\right)^4 \ .$$

Question 12.22

a) The polynomial interpolation of f at the points $x_0 = -2h$, $x_1 = -h$ and $x_2 = 0$ is given by

$$f(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)$$

$$+ f[x_0,x_1,x_2,x](x-x_0)(x-x_1)(x-x_2) \ .$$

Hence,

$$\begin{aligned}
\int_0^h f(x)\,\mathrm{d}x &= \int_0^h \Big(f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)\Big)\,\mathrm{d}x \\
&\qquad + \int_0^h f[x_0,x_1,x_2,x](x-x_0)(x-x_1)(x-x_2)\,\mathrm{d}x \\
&= \int_0^h \Big(f[-2h] + f[-2h,-h](x+2h) + f[-2h,-h,0](x+2h)(x+h)\Big)\,\mathrm{d}x \\
&\qquad + \int_0^h f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \\
&= f[-2h]\,x\Big|_0^h + f[-2h,-h]\left(\frac{x^2}{2} + 2hx\right)\Big|_0^h + f[-2h,-h,0]\left(\frac{x^3}{3} + \frac{3hx^2}{2} + 2h^2x\right)\Big|_0^h \\
&\qquad + \int_0^h f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \\
&= f[-2h]\,h + \frac{5}{2}f[-2h,-h]\,h^2 + \frac{23}{6}f[-2h,-h,0]\,h^3 \\
&\qquad + \int_0^h f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \ .
\end{aligned}$$

The formula to approximate the integral is

$$\begin{aligned}
\int_0^1 f(x)\,\mathrm{d}x &\approx f[-2h]\,h + \frac{5}{2}f[-2h,-h]\,h^2 + \frac{23}{6}f[-2h,-h,0]\,h^3 \\
&= f(-2h)\,h + \frac{5}{2}\left(\frac{f(-h)-f(-2h)}{-h-(-2h)}\right)h^2 + \frac{23}{6}\left(\frac{\dfrac{f(0)-f(-h)}{0-(-h)} - \dfrac{f(-h)-f(-2h)}{-h-(-2h)}}{0-(-2h)}\right)h^3 \\
&= \left(\frac{23}{12}\,f(0) - \frac{4}{3}\,f(-h) + \frac{5}{12}\,f(-2h)\right)h \ .
\end{aligned}$$

Since $(x+2h)(x+h)x \geq 0$ for all $x \in [0,h]$, we may use the Mean Value Theorem for Integrals to get $\eta \in [0,h]$ such that

$$\int_0^h f[-2,-1,0,x](x+2h)(x+h)x\,\mathrm{d}x = f[-2h,-h,0,\eta]\int_0^h (x+2h)(x+h)x\,\mathrm{d}x \ .$$

Moreover, from the theory on divided difference formulas, there exists $\xi \in [-2h,h]$ such that

$$\begin{aligned}
f[-2h,-h,0,\eta]\int_0^h (x+2h)(x+h)x\,\mathrm{d}x &= \frac{1}{3!}f^{(3)}(\xi)\int_0^h (x+2h)\,(x+h)\,x\,\mathrm{d}x \\
&= \frac{1}{3!}f^{(3)}(\xi)\left(\frac{9}{4}\right)h^4 = \frac{3}{8}\,f^{(3)}(\xi)\,h^4 \ .
\end{aligned}$$

Hence

$$\int_0^h f(x)\,\mathrm{d}x = \left(\frac{23}{12}\,f(0) - \frac{4}{3}\,f(-h) + \frac{5}{12}\,f(-2h)\right)h + \frac{3}{8}\,f^{(3)}(\xi)\,h^4 \tag{16.25}$$

for some $\xi \in [-2h, h]$.

b) To obtain a formula for $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ from (16.25), we use the substitution $x = a + t$ with $h = b - a$ to get

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \int_0^h f(a+t)\,\mathrm{d}t = \left(\frac{23}{12}\,f(a) - \frac{4}{3}\,f(a-h) + \frac{5}{12}\,f(a-2h)\right) h + \frac{3}{8}\,\frac{\mathrm{d}^3}{\mathrm{d}t^3} f(a+t)\bigg|_{t=\xi} h^4 \\ &= \left(\frac{23}{12}\,f(a) - \frac{4}{3}\,f(a-h) + \frac{5}{12}\,f(a-2h)\right) + \frac{3}{8}\,f^{(3)}(a+\xi)\,h^4 \ .\end{aligned}$$

c) The formula in (b) can be used to approximate the solution of the initial value problem given. If $y(a)$, $y(a-h)$ and $y(a-2h)$ are known, we can used them to approximate $y(a+h)$. We have

$$\int_a^{a+h} y'(x)\,\mathrm{d}x = y(x)\bigg|_{x=a}^{a+h} = y(a+h) - y(a) \ .$$

We also have

$$\begin{aligned}\int_a^{a+h} y'(x)\,\mathrm{d}x &= \int_a^{a+h} f(x, y(x))\,\mathrm{d}x \\ &\approx \left(\frac{23}{12}\,f(a, y(a)) - \frac{4}{3}\,f(a-h, y(a-h)) + \frac{5}{12}\,f(a-2h, y(a-2h))\right) \ .\end{aligned}$$

Hence,

$$y(a+h) \approx y(a) + h\left(\frac{23}{12}\,f(a, y(a)) - \frac{4}{3}\,f(a-h, y(a-h)) + \frac{5}{12}\,f(a-2h, y(a-2y))\right) \ .$$

Question 12.23

a) The polynomial interpolating of f at the points $x_0 = -2h$, $x_1 = -h$ and $x_2 = 0$ is

$$\begin{aligned}f(x) = f[x_0] &+ f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ f[x_0, x_1, x_2, x](x - x_0)(x - x_1)(x - x_2) \ .\end{aligned}$$

Hence,

$$\begin{aligned}\int_{-h}^h f(x)\,\mathrm{d}x &= \int_{-h}^h \big(f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)\big)\,\mathrm{d}x \\ &\quad + \int_{-h}^h f[x_0, x_1, x_2, x](x - x_0)(x - x_1)(x - x_2)\,\mathrm{d}x \\ &= \int_{-h}^h \big(f[-2h] + f[-2h, -h](x + 2h) + f[-2h, -h, 0](x + 2h)(x + h)\big)\,\mathrm{d}x \\ &\quad + \int_{-h}^h f[-2h, -h, 0, x](x + 2h)(x + h)x\,\mathrm{d}x \\ &= f[-2h]\,x\bigg|_{-h}^h + f[-2h, -h]\left(\frac{x^2}{2} + 2hx\right)\bigg|_{-h}^h + f[-2h, -h, 0]\left(\frac{x^3}{3} + \frac{3hx^2}{2} + 2h^2x\right)\bigg|_{-h}^h \\ &\quad + \int_{-h}^h f[-2h, -h, 0, x](x + 2h)(x + h)x\,\mathrm{d}x\end{aligned}$$

$$
\begin{aligned}
&= 2\,f[-2h]\,h + 4f[-2h,-h]\,h^2 + \frac{14}{3}f[-2h,-h,0]\,h^3 \\
&\qquad + \int_{-h}^{h} f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \ .
\end{aligned}
$$

The formula to approximate the integral is

$$
\begin{aligned}
\int_{-h}^{h} f(x)\,\mathrm{d}x &\approx 2\,f[-2h]\,h + 4f[-2h,-h]\,h^2 + \frac{14}{3}f[-2h,-h,0]\,h^3 \\
&= 2\,f(-2h)\,h + 4\left(\frac{f(-h)-f(-2h)}{-h-(-2h)}\right)h^2 + \frac{14}{3}\left(\frac{\dfrac{f(0)-f(-h)}{0-(-h)} - \dfrac{f(-h)-f(-2h)}{-h-(-2h)}}{0-(-2h)}\right)h^3 \\
&= \left(\frac{7}{3}\,f(0) - \frac{2}{3}\,f(-h) + \frac{1}{3}\,f(-2h)\right)h \ .
\end{aligned}
$$

Since $(x+2h)(x+h)x$ changes sign at $x=0$ on the interval $[-h,h]$, we may not directly use the Mean Value Theorem for Integrals to simplify the truncation error. However, from

$$
f[x_0,x_1,x_2,x_2,x] = \frac{f[x_0,x_1,x_2,x]-f[x_0,x_1,x_2,x_2]}{x-x_2} \ ,
$$

we get

$$
f[-2h,-h,0,x] = f[-2h,-h,0,0,x]x + f[-2h,-h,0,0] \ .
$$

Hence,

$$
\begin{aligned}
&\int_{-h}^{h} f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \\
&\quad = \int_{-h}^{h} \left(f[-2h,-h,0,0,x]x + f[-2h,-h,0,0]\right)(x+2h)(x+h)x\,\mathrm{d}x \\
&\quad = \int_{-h}^{h} f[-2h,-h,0,0,x](x+2h)(x+h)x^2\,\mathrm{d}x + f[-2h,-h,0,0]\int_{-h}^{h}(x+2h)(x+h)x\,\mathrm{d}x \ .
\end{aligned}
$$

The second integral can be easily computed. For the first integral, Since $(x+2h)(x+h)x^2$ is non-negative on the interval $[-h,h]$, we may use the Mean Value Theorem for Integrals to get $\eta \in [-h,h]$ such that

$$
\int_{-h}^{h} f[-2h,-h,0,0,x](x+2h)(x+h)x^2\,\mathrm{d}x = f[-2h,-h,0,0,\eta]\int_{-h}^{h}(x+2h)(x+h)x^2\,\mathrm{d}x \ .
$$

Hence,

$$
\begin{aligned}
&\int_{-h}^{h} f[-2h,-h,0,x](x+2h)(x+h)x\,\mathrm{d}x \\
&\quad = f[-2h,-h,0,0,\eta]\int_{-h}^{h}(x+2h)(x+h)x^2\,\mathrm{d}x + f[-2h,-h,0,0]\int_{-h}^{h}(x+2h)(x+h)x\,\mathrm{d}x \\
&\quad = \frac{26}{15}f[-2h,-h,0,0,\eta]\,h^5 + 2f[-2h,-h,0,0]\,h^4 \ .
\end{aligned}
$$

Moreover, from the theory on divided difference formulas, there exists $\xi_1, \xi_2 \in [-2h, h]$ such that $f[-2h, -h, 0, 0, \eta] = \dfrac{1}{4!} f^{(4)}(\xi_1)$ and $f[-2h, -h, 0, 0] = \dfrac{1}{3!} f^{(3)}(\xi_2)$. The truncation error is therefore

$$\begin{aligned}\int_{-h}^{h} f[-2h, -h, 0, x](x+2h)(x+h)x \, \mathrm{d}x &= \left(\frac{26}{15}\right) \frac{1}{4!} f^{(4)}(\xi_1)\, h^5 + 2\,\frac{1}{3!} f^{(3)}(\xi_2)\, h^4 \\ &= \frac{13}{180}\, f^{(4)}(\xi_1)\, h^5 + \frac{1}{3} f^{(3)}(\xi_2)\, h^4 \ .\end{aligned}$$

Hence,

$$\int_{-h}^{h} f(x)\, \mathrm{d}x = \left(\frac{7}{3} f(0) - \frac{2}{3} f(-h) + \frac{1}{3} f(-2h)\right) h + \frac{13}{180} f^{(4)}(\xi_1)\, h^5 + \frac{1}{3} f^{(3)}(\xi_2)\, h^4 \ . \qquad (16.26)$$

b) To obtain a formula for $\int_a^b f(x)\, \mathrm{d}x$ from (16.26), we use the substitution $x = \dfrac{b+a}{2} + t$ with $h = \dfrac{b-a}{2}$ to get

$$\begin{aligned}\int_a^b f(x)\, \mathrm{d}x &= \int_{-h}^{h} f\left(\frac{b+a}{2} + t\right) \mathrm{d}t \\ &= \left(\frac{7}{3} f\left(\frac{a+b}{2}\right) - \frac{2}{3} f(a) + \frac{1}{3} f\left(\frac{3a-b}{2}\right)\right)\left(\frac{b-a}{2}\right) \\ &\quad + \frac{13}{180} \frac{\mathrm{d}^4}{\mathrm{d}t^4} f\left(\frac{b+a}{2} + t\right)\Bigg|_{t=\xi_1} \left(\frac{b-a}{2}\right)^5 + \frac{1}{3} \frac{\mathrm{d}^3}{\mathrm{d}t^3} f\left(\frac{b+a}{2} + t\right)\Bigg|_{t=\xi_2} \left(\frac{b-a}{2}\right)^4 \\ &= \left(\frac{7}{6} f\left(\frac{b+a}{3}\right) - \frac{1}{3} f(a) + \frac{1}{6} f\left(\frac{3a-b}{2}\right)\right)(b-a) \\ &\quad + \frac{13}{5960} f^{(4)}\left(\frac{b+a}{2} + \xi_1\right)(b-a)^5 + \frac{1}{48} f^{(3)}\left(\frac{b+a}{2} + \xi_2\right)(b-a)^4 \ .\end{aligned}$$

Question 12.24

It suffices to show that there exist $c_1, c_2, \ldots, c_k$ such that (12.9.7) is true for $p(x) = x^m$ with $0 \leq m < k$. Namely,

$$\int_a^b x^m\, w(x)\, \mathrm{d}x = \sum_{j=1}^{k} c_j\, x_j^m$$

for $0 \leq m < k$. This can be rewritten as the system of linear equations $A\mathbf{c} = \mathbf{d}$, where

$$A = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_k \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \ldots & x_k^{k-1} \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \ldots \\ c_k \end{pmatrix} \quad \text{and} \quad \mathbf{d} = \begin{pmatrix} \int_a^b w(x)\, \mathrm{d}x \\ \int_a^b x\, w(t)\, \mathrm{d}x \\ \vdots \\ \int_a^b x^{k-1} w(x)\, \mathrm{d}x \end{pmatrix} .$$

This system has a unique solution because A is an invertible Vandermonde matrix. In fact, one can show that the determinant of the matrix A above is $\prod_{0<i<j\leq k} (x_j - x_i)$.

Question 12.25

Let p be a polynomial of degree less than $k+m$. By the Euclidean division algorithm, we get $p = Pq + r$, where q and r are polynomials of degree less than m. Hence,

$$\int_a^b p(x)\,w(x)\,dx = \int_a^b P(x)\,q(x)\,w(x)\,dx + \int_a^b r(x)\,w(x)\,dx = \int_a^b r(x)\,w(x)\,dx \tag{16.27}$$

because $\langle P, q\rangle = 0$ since q is of degree less than m. Moreover,

$$\sum_{j=1}^{k} c_j\,p(x_j) = \sum_{j=1}^{k} c_j\,P(x_j)\,q(x_j) + \sum_{j=1}^{k} c_j\,r(x_j) = \sum_{j=1}^{k} c_j\,r(x_j) \tag{16.28}$$

because $x_1, x_2, \ldots, x_k$ are the roots of P by hypothesis. Finally, because (12.9.7) is true for polynomials of degree less than k if the coefficients c_i are defined by (12.7.2), we have

$$\int_a^b r(x)\,w(x)\,dx = \sum_{j=1}^{k} c_j\,r(x_j) \tag{16.29}$$

since r is of degree less than $m \leq k$. It follows from (16.27), (16.28) and (16.29) that

$$\int_a^b p(t)\,w(t)\,dx = \sum_{j=1}^{k} c_j\,p(x_j)$$

for any polynomial of degree less than $k+m$.

We now prove that the quadrature formula is not true for all polynomial of degree $k+m$. Let $p(x) = x^m P(x) + r(x)$, where r is a polynomial of degree less than m. The equations (16.28) and (16.29) are still true if we replace $q(x)$ by x^m. So

$$\int_a^b r(x)\,w(x)\,dx = \sum_{j=1}^{k} c_j\,p(x_j)\ .$$

However,

$$\int_a^b p(x)\,w(x)\,dx = \underbrace{\int_a^b x^m P(x)\,w(x)\,dx}_{\neq 0} + \int_a^b r(x)\,w(x)\,dx \neq \int_a^b r(x)\,w(x)\,dx = \sum_{j=1}^{k} c_j\,p(x_j)\ .$$

Question 12.26

If f is q-time continuously differentiable on an open interval containing $[a,b]$, it follows from Taylor's Theorem, Theorem 2.1.6, that $f(x) = p(x) + r(x)$, where

$$p(x) = \sum_{j=0}^{q-1} \frac{f^{(j)}(a)}{j!}\,(x-a)^j \quad \text{and} \quad r(x) = \frac{f^{(q)}(\xi(x))}{q!}\,(x-a)^q$$

for some $\xi(x)$ between a in x. Thus,

$$\int_a^b f(x)w(x)\,dx = \int_a^b p(x)w(x)\,dx + \int_a^b r(x)w(x)\,dx = \sum_{j=1}^{k} b_j p(x_j) + \int_a^b r(x)w(x)\,dx$$

because p is a polynomial of degree at most $q-1$. Moreover

$$\begin{aligned}\int_a^b r(x)w(x)\,\mathrm{d}x &= \frac{1}{q!}\int_a^b f^{(q)}(\xi(x))\,(x-a)^q w(x)\,\mathrm{d}x = \frac{1}{q!} f^{(q)}(\xi(c)) \int_a^b (x-a)^q w(x)\,\mathrm{d}x \\ &= \frac{1}{q!} f^{(q)}(\xi(c))\,(b-a)^{q+1}\int_0^1 s^q w(a+(b-a)s)\,\mathrm{d}s\end{aligned}$$

for $c \in [a,b]$, where we have used the Mean Value Theorem for Integrals, Theorem 12.3.1, to get the second equality and the substitution $x = a+(b-a)s$ to get the last one. Finally, we also have

$$\sum_{j=1}^k b_j\, f(c_j) = \sum_{j=1}^k b_j\, p(c_j) + \sum_{j=1}^k b_j\, r(c_j) = \sum_{j=1}^k b_j\, p(c_j) + \frac{1}{q!}\sum_{j=1}^k b_j\, f^{(q)}(\xi(c_j))\,(c_j-a)^q \ .$$

Therefore,

$$\begin{aligned}&\left| \int_a^b f(x)\,w(x)\,\mathrm{d}x - \sum_{j=1}^k b_j\, f(c_j)\right| \\ &= \left|\frac{1}{q!} f^{(q)}(\psi(x))\,(b-a)^{q+1}\int_0^1 s^q w(a+(b-a)s)\,\mathrm{d}s - \frac{1}{q!}\sum_{j=1}^k b_j\, f^{(q)}(\xi(c_j))\,(c_j-a)^q\right| \\ &\leq \frac{1}{q!}\max_{a\leq x\leq b}|f^{(q)}(x)|\,(b-a)^{q+1}\int_0^1 s^q w(a+(b-a)s)\,\mathrm{d}s + \frac{1}{q!}\max_{a\leq x\leq b}|f^{(q)}(x)|\,(b-a)^q\sum_{j=1}^k |b_j| \\ &= K(b-a)^q \max_{a\leq x\leq b}|f^{(q)}(x)| \ ,\end{aligned}$$

where

$$K = \frac{1}{q!}\left((b-a)\int_0^1 s^q w(a+(b-a)s)\,\mathrm{d}s + \sum_{j=1}^k |b_j|\right) \ .$$

Question 12.28

To use Gauss-Legendre quadrature, we need to transform the intergral between 1 and 3 into an integral between -1 and 1. For this purpose, we use the change of variable $y = (x-M)/L$, where $M = 2$ is the middle of the interval $[1,3]$ and $L = 1$ is half the length of the interval $[1,3]$. Thus $y = x-2$ and, if we solve for x, we get $x = y+2$. The integral (12.9.8) becomes

$$\int_1^3 x^2\ln(x)\,\mathrm{d}x = \int_{-1}^1 (y+2)^2\ln(y+2)\,\mathrm{d}y \approx \sum_{j=1}^5 c_j(y_j+2)^2\,\ln(y_j+2) \approx 2.4040942246 \ ,$$

where y_i and c_i are given in the following table.

n	roots y_j	coefficients c_j
5	−0.9061798459	0.2369268851
	−0.5384693101	0.4786286705
	0.0	0.5688888889
	0.5384693101	0.4786286705
	0.9061798459	0.2369268851

Question 12.29
Because of the factor $\sqrt{(x-2)(3-x)}$ in the denominator of the integrand, Gauss-Chebyshev quadrature is a possible choice.

To transform the integral from an integral between 2 and 3 to an integral between −1 and 1, we use the substitution $t = (x-5/2)/(1/2)$. So $x = t/2+5/2$, $\mathrm{d}x = (1/2)\,\mathrm{d}t$ and

$$\int_2^3 \frac{\sin(x)}{\sqrt{(x-2)(3-x)}}\,\mathrm{d}x = \int_{-1}^1 \frac{\sin(t/2+5/2)}{\sqrt{1-t^2}}\,\mathrm{d}t \approx \frac{\pi}{3}\sum_{j=1}^3 \sin\left(\frac{1}{2}\cos\left(\frac{(2j-1)\pi}{6}\right)+\frac{5}{2}\right)$$
$$\approx 1.7644706129 \ .$$

Question 12.32
Because of the factor $\sqrt{x(1-x)}$ in the denominator of the integrand, Gauss-Chebyshev quadrature is a possible choice.

To transform the integral from an integral between 0 and 1 to an integral between −1 and 1, we use the substitution $x = (t+1)/2$. We get

$$\int_0^1 \frac{e^x}{\sqrt{x(1-x)}}\,\mathrm{d}x = \int_{-1}^1 \frac{e^{(t+1)/2}}{\sqrt{1-t^2}}\,\mathrm{d}t \approx \frac{\pi}{3}\sum_{i=1}^3 e^{(\cos((2i-1)\pi/6)+1)/2} \approx 5.50842622975 \ .$$

Question 12.34
Because of the factor $\sqrt{(1-x)(3+x)}$ in the denominator of the integrand, Gauss-Chebyshev quadrature is a possible choice.

Since we want the exact answer and the numerator of the integrand is a polynomial of degree 4, we have to take n equal to at least 3 as suggested by Theorem 12.7.5.

To transform the integral from an integral between −3 and 1 to an integral between −1 and 1, we use the substitution $t = (x+1)/2$. We get $x = 2t-1$ and

$$\int_{-3}^1 \frac{(1+x)^4}{\sqrt{(1-x)(3+x)}}\,\mathrm{d}x = 2\int_{-1}^1 \frac{(2t)^4}{\sqrt{(2-2t)(2+2t)}}\,\mathrm{d}x = 16\int_{-1}^1 \frac{t^4}{\sqrt{1-t^2}}\,\mathrm{d}x$$
$$= \frac{16\pi}{3}\sum_{j=1}^3 \cos^4\left(\frac{(2j-1)\pi}{6}\right) = \frac{16\pi}{3}\left(\left(-\frac{\sqrt{2}}{3}\right)^4+\left(\frac{\sqrt{2}}{3}\right)^4\right) = \frac{128\pi}{243} \ .$$

Question 12.36
We have to find polynomials P_k of degree exactly k such that the set $\{P_0, P_1, P_2, \ldots\}$ is an orthogonal set with respect to the scalar product

$$\langle g, h\rangle = \int_0^1 g(x)\,h(x)\,x\,\mathrm{d}x \ .$$

To generate the family of orthogonal polynomials, we use Theorem 8.2.3 with $\alpha_{k,k} = 1$ for all k and $w(x) = x$ for $0 \le x \le 1$. Let $P_0(x) = 1$ for all x. Since $A_0 = 1$ and $B_0 = \dfrac{\int_0^1 x^2\,\mathrm{d}x}{\int_0^1 x\,\mathrm{d}x} = \dfrac{2}{3}$, we get

$$P_1(x) = (x-B_0)P_0(x) = x - \frac{2}{3} \ .$$

Since $A_1 = 1$, $B_1 = \dfrac{\int_0^1 x^2(x-2/3)^2\,\mathrm{d}x}{\int_0^1 x(x-2/3)^2\,\mathrm{d}x} = \dfrac{8}{15}$ and $C_1 = \dfrac{\int_0^1 x(x-2/3)^2\,\mathrm{d}x}{\int_0^1 x\,\mathrm{d}x} = \dfrac{1}{18}$, we get

$$P_2(x) = (x - B_1)P_1(x) - C_1 P_0(x) = \left(x - \frac{8}{15}\right)\left(x - \frac{2}{3}\right) - \frac{1}{18} = x^2 - \frac{6}{5}x + \frac{3}{10} \ .$$

According to Theorem 12.7.5, we do not need higher degree polynomials since we want a Gaussian quadrature formula which is exact for polynomials of degree up to 3 only.

The roots of P_2 are $x_1 = (6-\sqrt{6})/10$ and $x_2 = (6+\sqrt{6})/10$.

The coefficients of (12.9.9) are

$$A = \int_0^1 \left(\frac{x - x_2}{x_1 - x_2}\right) x\,\mathrm{d}x = \left.\frac{x^3/3 - x_2 x^2/2}{x_1 - x_2}\right|_{x=0}^{1} = \frac{9-\sqrt{6}}{36}$$

and

$$B = \int_0^1 \left(\frac{x - x_1}{x_2 - x_1}\right) x\,\mathrm{d}x = \left.\frac{x^3/3 - x_1 x^2/2}{x_2 - x_1}\right|_{x=0}^{1} = \frac{9+\sqrt{6}}{36} \ .$$

It follows from Theorem 12.7.5 that (12.9.9) with the choice of x_1, x_2, A and B above is exact for polynomials of degree up to 3.

Question 12.37

We have to find polynomials P_k of degree exactly k such that the set $\{P_0, P_1, P_2, \ldots\}$ is an orthogonal set with respect to the scalar product

$$\langle g, h\rangle = \int_0^1 g(x)\,h(x)\,x^2\,\mathrm{d}x.$$

To generate the family of orthogonal polynomials, we use Theorem 8.2.3 with $\alpha_{k,k} = 1$ for all k and $w(x) = x^2$ for $0 \le x \le 1$. Let $P_0(x) = 1$ for all x. Since $A_0 = 1$ and $B_0 = \dfrac{\int_{-1}^1 x^3\,\mathrm{d}x}{\int_{-1}^1 x^2\,\mathrm{d}x} = 0$, we get

$$P_1(x) = (x - B_0)P_0(x) = x \ .$$

Since $A_1 = 1$, $B_1 = \dfrac{\int_{-1}^1 x^5\,\mathrm{d}x}{\int_{-1}^1 x^4\,\mathrm{d}x} = 0$ and $C_1 = \dfrac{\int_{-1}^1 x^4\,\mathrm{d}x}{\int_{-1}^1 x^2\,\mathrm{d}x} = \dfrac{3}{5}$, we get

$$P_2(x) = (x - B_1)P_1(x) - C_1 P_0(x) = x^2 - \frac{3}{5} \ .$$

According to Theorem 12.7.5, we do not need higher degree polynomials since we want a Gaussian quadrature formula which is exact for polynomials of degree up to 3 only.

The roots of P_2 are $x_1 = -\sqrt{3/5}$ and $x_2 = \sqrt{3/5}$.

The coefficients of (12.9.10) are

$$A = \int_{-1}^1 \frac{x - x_2}{x_1 - x_2}\,x^2\,\mathrm{d}x = \left.\frac{x^4/4 - x_2 x^3/3}{x_1 - x_2}\right|_{x=-1}^{1} = \frac{1}{3}$$

and

$$B = \int_{-1}^{1} \frac{x - x_1}{x_2 - x_1}\, x^2 \,\mathrm{d}x = \left.\frac{x^4/4 - x_1 x^3/3}{x_2 - x_1}\right|_{x=-1}^{1} = \frac{1}{3}\ .$$

It follows from Theorem 12.7.5 that (12.9.9) with the choice of x_1, x_2, A and B above is exact for polynomials of degree up to 3.

Question 12.38
Recall that the Gauss-Chebyshev quadrature with $n > 0$ is given by the the formula

$$\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b p(x)\,\mathrm{d}x\ , \tag{16.30}$$

where p is the interpolating polynomial of degree n of f at the $n+1$ Chebishev points adjusted to the interval $[a, b]$; namely, at the points

$$x_j = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(2j-1)\pi}{2(n+1)}\right)$$

for $j = 1, 2, \ldots, n+1$.

If we substitute $x = \dfrac{a+b}{2} + \dfrac{b-a}{2}\,t$ in the integrals in (16.30), we get

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\mathrm{d}t$$

and

$$\int_a^b p(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} p\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\mathrm{d}t\ .$$

$p\left(\dfrac{a+b}{2} + \dfrac{b-a}{2}\,t\right)$ is the interpolating polynomial of $f\left(\dfrac{a+b}{2} + \dfrac{b-a}{2}\,t\right)$ at the Chebyshev points $t_i = \cos\left(\dfrac{(2i-1)\pi}{2(n+1)}\right)$ for $1 \le i \le n+1$. From Proposition 9.2.6, we have that

$$\begin{aligned}
\left| f\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right) - p\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\right| &\le \frac{1}{2^n (n+1)!}\max_{-1\le t\le 1}\left|\frac{\mathrm{d}^{(n+1)}}{\mathrm{d}t^{(n+1)}} f\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\right| \\
&\le \frac{1}{2^n (n+1)!}\max_{-1\le t\le 1}\left| f^{(n+1)}\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\right| \left(\frac{b-a}{2}\right)^{n+1} \\
&\le \frac{1}{2^n (n+1)!}\max_{a\le x\le b}\left| f^{(n+1)}(x)\right| \left(\frac{b-a}{2}\right)^{n+1} \le \frac{M(b-a)^{n+1}}{2^{2n+1}(n+1)!}
\end{aligned}$$

for $-1 \le x \le 1$. Hence

$$\left|\int_a^b f(x)\,\mathrm{d}x - \int_a^b p(x)\,\mathrm{d}x\right| = \left|\frac{b-a}{2}\int_{-1}^{1}\left(f\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right) - p\left(\frac{a+b}{2} + \frac{b-a}{2}\,t\right)\right)\mathrm{d}t\right|$$

$$\leq \frac{b-a}{2}\int_{-1}^{1}\left|f\left(\frac{a+b}{2}+\frac{b-a}{2}\,t\right)-p\left(\frac{a+b}{2}+\frac{b-a}{2}\,t\right)\right|\,\mathrm{d}t$$
$$\leq \frac{b-a}{2}\int_{-1}^{1}\frac{M(b-a)^{n+1}}{2^{2n+1}(n+1)!}\,\mathrm{d}t = \frac{M(b-a)^{n+2}}{2^{2n+1}(n+1)!}\,\mathrm{d}t\ .$$

Question 12.39

Let q be any polynomial of degree less than n, then $f(x) = q(x)\prod_{j=1}^{n}(x-x_j)$ is a polynomial of degree less than $2n$. By hypothesis, we then have

$$\int_a^b f(x)\,w(x)\,\mathrm{d}x = \sum_{i=1}^{n} a_i\,f(x_i) = \sum_{i=1}^{n}\left(a_i\,q(x_i)\prod_{j=1}^{n}\underbrace{(x_i-x_j)}_{=0 \text{ for } j=i}\right) = 0\ .$$

Question 12.40

Consider the polynomial of degree $2n$ defined by $f(x) = \prod_{i=1}^{n}(x-x_i)^2$. If (12.9.11) is exact for polynomials of degree $2n$, we must have

$$\int_a^b f(x)\,w(x)\,\mathrm{d}x = \sum_{j=1}^{n} c_j\,f(x_j) = \sum_{j=1}^{n}\left(c_j\prod_{i=1}^{n}\underbrace{(x_j-x_i)^2}_{=0 \text{ for } j=i}\right) = 0\ .$$

But this is not possible because the integral cannot be null since f is a continuous function such that $f(x) > 0$ for all $x \in [a,b] \setminus \{x_1, x_2, \ldots, x_n\}$ and $w(x) > 0$ almost everywhere.

Chapter 13 : Initial Value Problems for Ordinary Differential Equations

Question 13.1

a) Let $f(t,y) = t^2\sin(y) + y$. The function f is obviously continuous.

According to the Mean Value Theorem, given any $y_1, y_2 \in \mathbb{R}$, there exists ξ between y_1 and y_2 such that

$$f(t,y_1) - f(t,y_2) = \frac{\partial f}{\partial y}(t,\xi)\,(y_1-y_2) = (t^2\cos(\xi)+1)\,(y_1-y_2)\ .$$

Since $|t^2\cos(y)+1| \leq 2$ for all $(t,y) \in D = \{(t,y) : 0 \leq t \leq 1 \text{ and } y \in \mathbb{R}\}$, we get

$$|f(t,y_1) - f(t,y_2)| \leq 2|y_1-y_2|$$

for all $(t,y) \in D$. Thus, f satisfies a Lipschitz condition with respect to y on D with Lipschitz constant $L = 2$.

It follows from Theorem 13.1.3 that the initial value problem is well posed.

Question 13.3

a) Let $f(t, y) = (y + t)/t = 1 + y/t$. The function f is obviously a continuous function for $(t, y) \in D = \{(t, y) : 1 \leq t \leq 2 \text{ and } y \in \mathbb{R}\}$. Moreover, f satisfies a Lipschitz condition with respect to y on D with Lipschitz constant $L = 1$ because

$$|f(t, y_1) - f(t, y_2)| = \left|\frac{y_1 - y_2}{t}\right| \leq |y_1 - y_2|$$

for $(t, y) \in D$. It follows from Theorem 13.1.3 that the initial value problem is well posed.

b) Let u_i be the computed value for w_i, where w_i is the approximation of $y(t_i)$ given by the Euler Method (Definition 13.2.1). Let $e_i = y(t_i) - u_i$. It follows from (13.2.6) that

$$|e_i| \leq \frac{1}{L}\left(\frac{Mh}{2} + \frac{\delta}{h}\right)\left(e^{L(t_i - t_0)} - 1\right) + |\delta_0|\, e^{L(t_i - t_0)} \leq \left(\frac{Mh}{2} + \frac{10^{-8}}{h}\right)\left(e^{t_i - t_0} - 1\right) + 10^{-8}\, e^{t_i - t_0} \ ,$$

where $M \geq |y''(t)|$ for $1 \leq t \leq 2$. To minimize the right side of this inequality with respect to h, we have to minimize $g(h) = \dfrac{Mh}{2} + \dfrac{10^{-8}}{h}$.

In general, it is not easy to find a possible value for M. Fortunately, for the present problem, we can. $y' = 1 + y/t$ is a linear ordinary differential equation of the form $y' + p(t)y = q(t)$ with $p(t) = -1/t$ and $q(t) = 1$. Its general solution is

$$\begin{aligned} y(t) &= e^{-\int p(t)\,\mathrm{d}t}\left(\int q(t)\, e^{\int p(t)\,\mathrm{d}t}\,\mathrm{d}t + C\right) = e^{-\int (-1/t)\,\mathrm{d}t}\left(\int e^{\int (-1/t)\,\mathrm{d}t}\,\mathrm{d}t + C\right) \\ &= t\left(\int \frac{1}{t}\,\mathrm{d}t + C\right) = t\,(\ln(t) + C)\ . \end{aligned}$$

The initial condition $y(1) = 0$ implies that $C = 0$. We get $y(t) = t\ln(t)$. For $1 \leq t \leq 2$, we have $|y''(t)| = |1/t| \leq 1$. Thus, we may take $M = 1$.

We have to minimize $g(h) = \dfrac{h}{2} + \dfrac{10^{-8}}{h}$ for $h > 0$. We deduce from the following information about g that it has a global minimum for $h > 0$ at $h = \sqrt{2}/10^4$.

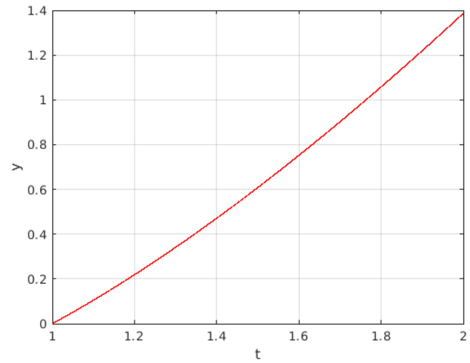
h	$0 < h < \sqrt{2}/10^4$	$\sqrt{2}/10^4$	$\sqrt{2}/10^4 < h$
$g'(h)$	$-$	0	$+$
	decreases	global min.	increases

Thus, $h = \sqrt{2}/10^4$ minimizes the error bound for the Euler Method.

c) We use the Euler Method with $t_0 = 1$, $t_f = 2$, $y_0 = 0$ and $f(t, y) = 1 + y/t$. Since $(t_f - t_0)/h = 10^4/\sqrt{2} \approx 7071.0678$ is an irrational number, we round up to the next integer to get $N = 7072$. Hence, $h = 1/7072$ and $t_i = t_0 + ih$ for $0 \leq i \leq 7072$. The approximation w_i of $y_i = y(t_i)$ is given by

$$\begin{aligned} w_0 &= 0 \\ w_{i+1} &= w_i + h(1 + w_i/t_i) \end{aligned}$$

for $i = 0, 1, \ldots, 7071$. The graph of the computed solution is basically indistinguishable from the graph of the solution.



The graph of the computed solution is in blue and the graph of the exact solution (drawn after the graph of the computed solution) is in red. The graph of the exact solution basically covers the graph of the computed solution.

d) The predicted error bound at $t = 2$ is given by

$$|e_i| \leq \left(\frac{h}{2} + \frac{10^{-8}}{h}\right)\left(e^{t_i - t_0} - 1\right) + 10^{-8}\, e^{t_i - t_0}$$

with $i = 7072$ and $h = 1/7072$. Hence $t_i = t_{7072} = 2$, $t_0 = 1$ and

$$|e_{7072}| \leq \left(\frac{1}{14144} + \frac{7072}{10^8}\right)(e-1) + \frac{e}{10^8} \approx 2.4303 \times 10^{-4} \ .$$

Since $u_{7072} \approx 1.3862237$ and $y_{7072} = y(2) \approx 1.3862944$, the actual error is $|e_{7072}| = |u_{7072} - y_{7072}| \approx 0.707 \times 10^{-4}$. Our predicted error bound is fairly conservative.

Question 13.4

Here is a need trick to solve (13.8.1). If $u = t - y$, the initial value problem (13.8.1) becomes

$$\begin{aligned} u' &= -u^2 \quad , \quad 2 \leq t \leq 3 \\ u(2) &= 1 \end{aligned}$$

This is a separable equation whose solution is $u(t) = \dfrac{1}{t-1}$ for $t \neq 1$. Thus, the solution of (13.8.1) is $y(t) = t - \dfrac{1}{t-1}$ for $t \neq 1$.

We have $t_0 = 2$, $t_f = t_{10} = 3$, $y_0 = 1$ and $f(t,y) = 1+(t-y)^2$. Since $h = (t_f - t_0)/N = 1/N = 0.1$, we get $N = 10$. Thus $t_j = t_0 + hi = 1 + 0.1i$ for $0 \leq i \leq 10$. The approximations w_i of $y_i = y(t_i)$ are given by $w_{i+1} = w_i + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$ for $i \geq 0$ with $w_0 = 1$, where $K_1 = 1 + (t_i - w_i)^2$, $K_2 = 1 + \big((t_i + 0.05) - (w_i + 0.05K_1)\big)^2$, $K_3 = 1 + \big((t_i + 0.05) - (w_i + 0.05K_2)\big)^2$ and $K_4 = 1 + \big(t_{i+1} - (w_i + 0.1K_3)\big)^2$.

The results are listed in the table below.

i	t_i	w_i	y_i	absolute error	relative error
0	2	1	1	0	0
1	2.1	1.1909088	1.1909091	$0.27724131 \times 10^{-6}$	$0.23279805 \times 10^{-6}$
2	2.2	1.3666663	1.3666667	$0.39550974 \times 10^{-6}$	$0.28939737 \times 10^{-6}$
3	2.3	1.5307688	1.5307692	$0.43652252 \times 10^{-6}$	$0.28516546 \times 10^{-6}$
4	2.4	1.6857138	1.6857143	$0.43960705 \times 10^{-6}$	$0.26078384 \times 10^{-6}$
5	2.5	1.8333329	1.8333333	$0.42439920 \times 10^{-6}$	$0.23149047 \times 10^{-6}$
6	2.6	1.9749996	1.9750000	$0.40094917 \times 10^{-6}$	$0.20301224 \times 10^{-6}$
7	2.7	2.1117643	2.1117647	$0.37446319 \times 10^{-6}$	$0.17732240 \times 10^{-6}$
8	2.8	2.2444441	2.2444444	$0.34762766 \times 10^{-6}$	$0.15488361 \times 10^{-6}$
9	2.9	2.3736839	2.3736842	$0.32179057 \times 10^{-6}$	$0.13556587 \times 10^{-6}$
10	3	2.4999997	2.5	$0.29758023 \times 10^{-6}$	$0.11903209 \times 10^{-6}$

Question 13.6

a) The Butcher array is

$$\begin{array}{r|cc} \alpha_1 = \beta & \beta_{1,1} = \beta & \beta_{1,2} = 0 \\ \alpha_2 = 1 + \beta & \beta_{2,1} = 1 & \beta_{2,2} = \beta \\ \hline & \gamma_1 = \beta + 1/2 & \gamma_2 = -\beta + 1/2 \end{array}$$

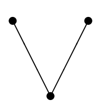
b) We answer this question using Tables 13.1 and 13.3.

i) Tree of order one: $\tau =$

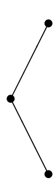
$\gamma(\tau) = 1$ and $\Psi(\tau) = \sum_{j=1}^{2} \gamma_j = 1$. So $\Psi(\tau) = 1/\gamma(\tau)$.

ii) Tree of order two: $\tau =$

$\gamma(\tau) = 2$ and $\Psi(\tau) = \sum_{j=1}^{2} \gamma_j \alpha_j = \left(\beta + \frac{1}{2}\right)\beta + \left(-\beta + \frac{1}{2}\right)(1+\beta) = \frac{1}{2}$. So $\Psi(\tau) = 1/\gamma(\tau)$.

iii) Trees of order three: $\tau_1 =$ 

$\gamma(\tau_1) = 3$ and $\Psi(\tau_1) = \sum_{j=1}^{2} \gamma_j \alpha_j^2 = \left(\beta + \frac{1}{2}\right)\beta^2 + \left(-\beta + \frac{1}{2}\right)(1+\beta)^2 = \frac{1}{2} - \beta^2$. So $\Psi(\tau_1) = 1/\gamma(\tau_1)$ only if $\frac{1}{2} - \beta^2 = \frac{1}{3}$; namely, only if $\beta = \pm\frac{1}{\sqrt{6}}$.

$\tau_2 =$ 

$\gamma(\tau_2) = 6$ and $\Psi(\tau_2) = \sum_{j_1=1}^{2}\left(\gamma_{j_1}\left(\sum_{j_2=1}^{2}\beta_{j_1,j_2}\alpha_{j_2}\right)\right) = \left(\beta+\frac{1}{2}\right)\beta^2 + \left(-\beta+\frac{1}{2}\right)(\beta+\beta(1+\beta)) =$ $\beta-\beta^2$. So $\Psi(\tau_2) = 1/\gamma(\tau_2)$ only if $\beta-\beta^2 = \frac{1}{6}$; namely, only if $\beta = \frac{1}{2}\left(1 \pm \frac{1}{\sqrt{3}}\right)$.

Since no value of β can simultaneously satisfy $\Psi(\tau_j) = 1/\gamma(\tau_j)$ for $j = 1$ and $j = 2$, it follows from Theorem 13.4.32 that the method is of order two. We note that the condition $\Psi(\tau) = 1/\gamma(\tau)$ for the trees of order one and two does not depend on β.

c) According to (13.4.13), the local truncation error is

$$\begin{aligned}\tau_{i+1}(h) &= \frac{h^2}{3!}\sum_{r(\tau)=3}\alpha(\tau)\left(1-\gamma(\tau)\,\Psi(\tau)\right)F(\tau) + O(h^3)\\ &= \frac{h^2}{6}\left(\left(1-3\left(\frac{1}{2}-\beta^2\right)\right)\{f\ f\} + \left(1-6(\beta-\beta^2)\right)\{\{f\}\}\right) + O(h^3)\ ,\end{aligned}$$

where the derivatives of f are evaluated at $\mathbf{y}(t_i)$.

d) For the initial value problem (13.8.2), we have that $f(\mathbf{y}) = A\mathbf{y}$ is a linear mapping. Thus $\{f\ f\} = 0$ and the local truncation error is now

$$\tau_{i+1}(h) = \frac{h^2}{6}\left(1-6(\beta-\beta^2)\right)\{\{f\}\} + O(h^3)\ .$$

It suffices to take one of the two possibles value for $\beta = \frac{1}{2}\left(1 \pm \frac{1}{\sqrt{3}}\right)$ to get a method of order (at least) three for this initial value problem.

e) We have

$$K_1 = A(\mathbf{w}_i + \beta h K_1) = A\mathbf{w}_i + \beta h A K_1 \Rightarrow (\mathrm{Id}-\beta hA)K_1 = A\mathbf{w}_i \Rightarrow K_1 = (\mathrm{Id}-\beta hA)^{-1}A\mathbf{w}_i$$

and

$$\begin{aligned}K_2 &= A(\mathbf{w}_i + hK_1 + \beta hK_2) = A\mathbf{w}_i + hAK_1 + \beta hAK_2 \Rightarrow (\mathrm{Id}-\beta hA)K_2 = A\mathbf{w}_i + hAK_1\\ &\Rightarrow K_2 = (\mathrm{Id}-\beta hA)^{-1}\left(A\mathbf{w}_i + +hAK_1\right) = (\mathrm{Id}-\beta hA)^{-1}\left(A\mathbf{w}_i + hA(\mathrm{Id}-\beta hA)^{-1}A\mathbf{w}_i\right)\\ &\quad= (\mathrm{Id}-\beta hA)^{-1}\left(A + hA^2(\mathrm{Id}-\beta hA)^{-1}\right)\mathbf{w}_i\end{aligned}$$

because $(\mathrm{Id}-\beta hA)^{-1}A = A(\mathrm{Id}-\beta hA)^{-1}$. Thus

$$\begin{aligned}\mathbf{w}_{i+1} &= \mathbf{w}_i + h\left(\frac{1}{2}+\beta\right)(\mathrm{Id}-\beta hA)^{-1}A\mathbf{w}_i + h\left(\frac{1}{2}-\beta\right)(\mathrm{Id}-\beta hA)^{-1}\left(A + hA^2(\mathrm{Id}-\beta hA)^{-1}\right)\mathbf{w}_i\\ &= R(hA,\beta)\mathbf{w}_i\ ,\end{aligned}$$

where

$$R(Z,\beta) = 1 + \left(\frac{1}{2}+\beta\right)(\mathrm{Id}-\beta Z)^{-1}Z + \left(\frac{1}{2}-\beta\right)(\mathrm{Id}-\beta Z)^{-1}\left(Z + Z^2(\mathrm{Id}-\beta Z)^{-1}\right)\ .$$

Question 13.7
We assume that $f(t,y)$ is twice continuously differentiable. So $y'''(t)$ is continuous. The local truncation error is given by

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - \frac{1}{2}\left(f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right) .$$

Since $y(t_{i+1}) = y(t_i) + y'(t_i)\,h + y''(t_i)\,\dfrac{h^2}{2} + y'''(\xi_i)\,\dfrac{h^3}{3!}$, $f(t_i, y(t_i)) = y'(t_i)$ and $f(t_{i+1}, y(t_{i+1})) = y'(t_{i+1}) = y'(t_i) + y''(t_i)h + y'''(\eta_i)\,\dfrac{h^2}{2}$ for some ξ_i and η_i, we get

$$\begin{aligned}\tau_{i+1}(h) = \frac{1}{h}&\left(y(t_i) + y'(t_i)\,h + y''(t_i)\,\frac{h^2}{2} + y'''(\xi_i)\,\frac{h^3}{3!} - y(t_i)\right) \\ &- \frac{1}{2}\left(y'(t_i) + y'(t_i) + y''(t_i)\,h + y'''(\eta_i)\,\frac{h^2}{2}\right) = y'''(\xi_i)\,\frac{h^2}{3!} - y'''(\eta_i)\,\frac{h^2}{4} .\end{aligned}$$

If $M = \max_{t_0 \le t \le t_f} |y'''(t)|$, we get

$$|\tau_{i+1}(h)| \le |y'''(\xi_i)|\frac{h^2}{3!} + |y'''(\eta_i)|\frac{h^2}{4} \le \frac{5}{12} M h^2$$

for $0 \le i \le N = (t_f - t_0)/h$. The method is of order 2 because $\tau_{i+1}(h) = O(h^2)$. The method is consistent because

$$\max_{0 \le i \le N} |\tau_{i+1}(h)| \le \frac{5}{12} M h^2 \to 0$$

as $h \to 0$.

Question 13.8
We have

$$\begin{aligned} w_{2i+2} &= w_{2i+1} + \frac{h}{2}\big(f(t_{2i+1}, w_{2i+1}) + f(t_{2i+2}, w_{2i+2})\big) \\ &= \left(w_{2i} + \frac{h}{2}\left(f(t_{2i}, w_{2i}) + f(t_{2i+1}, w_{2i+1})\right)\right) + \frac{h}{2}\big(f(t_{2i+1}, w_{2i+1}) + f(t_{2i+2}, w_{2i+2})\big) \\ &= w_{2i} + \frac{h}{2}\big(f(t_{2i}, w_{2i}) + 2f(t_{2i+1}, w_{2i+1}) + f(t_{2i+2}, w_{2i+2})\big) . \end{aligned}$$

Let $\tilde{h} = 2h$, $\tilde{t}_i = t_{2i}$ and $\tilde{w}_i = w_{2i}$. We get

$$\tilde{w}_{i+1} = \tilde{w}_i + \frac{\tilde{h}}{4}\left(K_1 + 2K_2 + K_3\right) , \tag{16.31}$$

where

$$K_1 = f(\tilde{t}_i, \tilde{w}_i) , \quad K_2 = f(t_{2i+1}, w_{2i+1}) = f\left(\tilde{t}_i + \frac{\tilde{h}}{2}, \tilde{w}_i + \frac{\tilde{h}}{4}(K_1 + K_2)\right)$$

and

$$K_3 = f(\tilde{t}_{i+1}, \tilde{w}_{i+1}) = f\left(\tilde{t}_i + \tilde{h}, \tilde{w}_i + \frac{\tilde{h}}{4}(K_1 + 2K_2 + K_3)\right) .$$

Note that
$$w_{2i+1} = w_{2i} + \frac{h}{2}\left(f(t_{2i}, w_{2i}) + f(t_{2i+1}, w_{2i+1})\right) = \tilde{w}_i + \frac{\tilde{h}}{4}(K_1 + K_2)$$
and $w_{2i+2} = \tilde{w}_{i+1}$.

The Butcher array of this Runge-Kutta Method is

$$\begin{array}{c|ccc} 0 & 0 & & \\ 1/2 & 1/4 & 1/4 & \\ 1 & 1/4 & 1/2 & 1/4 \\ \hline & 1/4 & 1/2 & 1/4 \end{array}$$

Since the trapezoidal method is of order two, we have

$$\frac{y_{2i+1} - y_{2i}}{h} - \frac{1}{2}\left(f(t_{2i}, y_{2i}) + f(t_{2i+1}, y_{2i+1})\right) = O(h^2)$$

and
$$\frac{y_{2i+2} - y_{2i+1}}{h} - \frac{1}{2}\left(f(t_{2i+1}, y_{2i+1}) + f(t_{2i+2}, y_{2i+2})\right) = O(h^2)\ .$$
The sum of these two equations yields

$$\frac{y_{2i+2} - y_{2i}}{h} - \frac{1}{2}\left(f(t_{2i}, y_{2i}) + 2f(t_{2i+1}, y_{2i+1}) + f(t_{2i+2}, y_{2i+2})\right) = O(h^2)\ .$$

With $\tilde{y}_i = y(\tilde{t}_i) = y(t_{2i}) = y_{2i}$, the previous equation becomes

$$\frac{\tilde{y}_{i+2} - \tilde{y}_i}{\tilde{h}} - \frac{1}{4}\left(\tilde{K}_1 + 2\tilde{K}_2 + \tilde{K}_3\right) = O(h^2)\ ,$$

where $\tilde{K}_1 = f(\tilde{t}_i, \tilde{y}_i)$, $\tilde{K}_2 = f(t_{2i+1}, y_{2i+1}) = f\left(\tilde{t}_i + \frac{\tilde{h}}{2}, \tilde{y}_1 + \frac{\tilde{h}}{4}(\tilde{K}_1 + \tilde{K}_2)\right)$ and $\tilde{K}_3 = f(t_{2i+2}, y_{2i+2}) = f\left(\tilde{t}_i + \tilde{h}, \tilde{y}_i + \frac{\tilde{h}}{4}(\tilde{K}_1 + 2\tilde{K}_2 + \tilde{K}_3)\right)$. This is (16.31) where $\tilde{w}_i$ has been replaced by $\tilde{y}_i$. Hence, the Runge-Kutta Method (16.31) is of order at least two.

Using the Taylor polynomial expansion (Theorem 2.1.6) of $y(t_{i+2})$, $y'(t_{i+1}) = f(t_{i+1}, y_{i+1})$ and $y'(t_{i+2}) = f(t_{i+2}, y_{i+2})$ about t_i, we can show that the order is in fact two.

Question 13.9

We use Theorem 13.4.11 to find the elements of the Butcher array of this 2-stage Runge-Kutta Method.

$$\beta_{1,1} = \int_0^{1/3} \frac{t - 2/3}{1/3 - 2/3}\,dt = -3\int_0^{1/3}\left(t - \frac{2}{3}\right)dt = \frac{1}{2}\ ,$$
$$\beta_{1,2} = \int_0^{1/3} \frac{t - 1/3}{2/3 - 1/3}\,dt = 3\int_0^{1/3}\left(t - \frac{1}{3}\right)dt = -\frac{1}{6}\ ,$$
$$\beta_{2,1} = \int_0^{2/3} \frac{t - 2/3}{1/3 - 2/3}\,dt = -3\int_0^{2/3}\left(t - \frac{2}{3}\right)dt = \frac{2}{3}\ ,$$

$$\beta_{2,2} = \int_0^{2/3} \frac{t-1/3}{2/3-1/3}\,\mathrm{d}t = 3\int_0^{2/3}\left(t-\frac{1}{3}\right)\mathrm{d}t = 0\ ,$$
$$\gamma_1 = \int_0^{1} \frac{t-2/3}{1/3-2/3}\,\mathrm{d}t = -3\int_0^{1}\left(t-\frac{2}{3}\right)\mathrm{d}t = \frac{1}{2}$$

and

$$\gamma_2 = \int_0^{1} \frac{t-1/3}{2/3-1/3}\,\mathrm{d}t = 3\int_0^{1}\left(t-\frac{1}{3}\right)\mathrm{d}t = \frac{1}{2}\ .$$

Hence, the Butcher array is

$$\begin{array}{c|cc} 1/3 & 1/2 & -1/6 \\ 2/3 & 2/3 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Let

$$q(t) = \left(t-\frac{1}{3}\right)\left(t-\frac{2}{3}\right)\ .$$

Since

$$\int_0^1 q(t)\,\mathrm{d}t = \left(\frac{t^3}{3}-\frac{t^2}{2}+\frac{2t}{9}\right)\bigg|_{t=0}^{1} = \frac{1}{18} \neq 0\ ,$$

it follows from Theorem 13.4.15 that the Runge-Kutta Method given by the Butcher array above is of order two.

If α_1 and α_2 are the roots of the Legendre polynomial $t^2-t+1/6$, then the 2-stage Runge-Kutta Method given by the collocation method is of order 4. More details are given in Examples 13.4.17 and 13.4.34.

Question 13.10

i) Let's suppose that the method is given by a collocation method. It follows from Remark 13.4.12 that

$$\int_0^{\alpha_i} q(t)\,\mathrm{d}t = \sum_{j=1}^k \beta_{i,j}\,q(a_j) \quad \text{and} \quad \int_0^1 q(t)\,\mathrm{d}t = \sum_{j=1}^k \gamma_j\,q(a_j) \tag{16.32}$$

for all polynomial of degree less than k and $1 \leq i \leq k$. If $q(t) = t^{n-1}$ with $1 \leq n \leq k$, then (16.32) yields

$$\sum_{j=1}^k \beta_{i,j}\,a_j^{n-1} = \int_0^{\alpha_i} t^{n-1}\,\mathrm{d}t = \frac{\alpha_i^n}{n} \quad \text{and} \quad \sum_{j=1}^k \gamma_j\,a_j^{n-1} = \int_0^1 t^{n-1}\,\mathrm{d}t = \frac{1}{n}$$

for $1 \leq i \leq k$. Thus, we get (13.8.3).

ii) Let's suppose that (13.8.3) is satisfied. If $q(t) = \sum_{m=0}^{k-1} a_m t^m$, then

$$\int_0^{\alpha_i} q(t)\,\mathrm{d}t = \sum_{m=0}^{k-1} a_m \int_0^{\alpha_i} t^m\,\mathrm{d}t = \sum_{m=0}^{k-1} a_m\left(\frac{\alpha_i^{m+1}}{m+1}\right) = \sum_{m=0}^{k-1} a_m\left(\sum_{j=1}^k \beta_{i,j}\alpha_j^m\right)$$

$$= \sum_{j=1}^{k} \beta_{i,j} \left(\sum_{m=0}^{k-1} a_m \alpha_j^m \right) = \sum_{j=1}^{k} \beta_{i,j} q(\alpha_j)$$

for $1 \leq i \leq k$ and

$$\begin{aligned}\int_0^1 q(t)\,\mathrm{d}t &= \sum_{m=0}^{k-1} a_m \int_0^1 t^m\,\mathrm{d}t = \sum_{m=0}^{k-1} a_m \left(\frac{1}{m+1}\right) = \sum_{m=0}^{k-1} a_m \left(\sum_{j=1}^{k} \gamma_j \alpha_j^m\right) \\ &= \sum_{j=1}^{k} \gamma_j \left(\sum_{m=0}^{k-1} a_m \alpha_j^m\right) = \sum_{j=1}^{k} \gamma_j q(\alpha_j)\ .\end{aligned}$$

Thus, (16.32) is satisfied.

Since $\alpha_i \neq \alpha_j$ for $i \neq j$, any polynomial q of degree less than k has a unique representation of the form $q(t) = \sum_{j=1}^{k} q(\alpha_j)\,\ell_j(t)$, where $\ell_j(t)$ is defined in Theorem 13.4.11. Hence,

$$\int_0^{\alpha_i} q(t)\,\mathrm{d}t = \sum_{j=1}^{k} q(\alpha_j) \int_0^{\alpha_i} \ell_j(t)\,\mathrm{d}t \quad \text{and} \quad \int_0^1 q(t)\,\mathrm{d}t = \sum_{j=1}^{k} q(\alpha_j) \int_0^1 \ell_j(t)\,\mathrm{d}t \tag{16.33}$$

for all polynomial of degree less than k and $1 \leq i \leq k$.

Combining (16.32) and (16.33), we get

$$\sum_{j=1}^{k} q(\alpha_j) \left(\beta_{i,j} - \int_0^{\alpha_i} \ell_j(t)\,\mathrm{d}t\right) = 0 \quad , \quad 1 \leq i \leq k\ , \tag{16.34}$$

and

$$\sum_{j=1}^{k} q(\alpha_j) \left(\gamma_j - \int_0^{1} \ell_j(t)\,\mathrm{d}t\right) = 0 \tag{16.35}$$

for all polynomial of degree less than k.

Let A be the $k \times k$ matrix with the components $a_{n+1,j} = \alpha_j^n$ for $0 \leq n < k$ and $1 \leq j \leq k$. Since the α_j are distinct, A is a non-singular matrix. In fact, A is a invertible Vandermonde matrix. For $1 \leq i \leq k$, Let $\mathbf{w}^{[i]}$ be the vector with components $w_j^{[i]} = \beta_{i,j} - \int_0^{\alpha_i} \ell_j(t)\,\mathrm{d}t$ for $1 \leq j \leq k$. Moreover, let $\mathbf{w}^{[k+1]}$ be the vector with the components $w_j^{[k+1]} = \gamma_j - \int_0^1 \ell_j(t)\,\mathrm{d}t$ for $1 \leq j \leq k$.

We have that (16.34) with $q(t) = t^n$ for $0 \leq n < k$ yields the linear equations $A\mathbf{w}^{[i]} = \mathbf{0}$ for $0 \leq i \leq k$. Similarly, (16.35) with $q(t) = t^n$ for $0 \leq n < k$ yields the linear equations $A\mathbf{w}^{[k+1]} = \mathbf{0}$. Since A is non-singular, the only solution of $A\mathbf{w} = \mathbf{0}$ is $\mathbf{w} = \mathbf{0}$. Thus, we get

$$\beta_{i,j} = \int_0^{\alpha_i} \ell_j(t)\,\mathrm{d}t \quad \text{and} \quad \gamma_j = \int_0^1 \ell_j(t)\,\mathrm{d}t$$

for $1 \leq i, j \leq k$.

Question 13.11
In the proof of Lemma 13.6.35, we showed that

$$r(z) = 1 + z\mathbf{c}^\top (\mathrm{Id} - zB)^{-1}\mathbf{u} ,$$

where

$$B = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,s} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{s,1} & \beta_{s,2} & \dots & \beta_{s,s} \end{pmatrix} , \quad \mathbf{c} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_s \end{pmatrix} \quad \text{and} \quad \mathbf{u} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} .$$

Since

$$(\mathrm{Id} - zB)^{-1} = \frac{1}{\det(\mathrm{Id} - zB)} \operatorname{adj}(\mathrm{Id} - zB) ,$$

we have that $\mathbf{c}^\top(\mathrm{Id} - zA)^{-1}\mathbf{u}$ is the quotient of two polynomials. The numerator is a polynomial of degree $k-1$ given by a linear combination of the components of $\operatorname{adj}(\mathrm{Id} - zB)$. The denominator is $\det(\mathrm{Id} - zB)$, a polynomial of degree k. Since B is lower-triangular, we have that $\det(\mathrm{Id} - zB) = \prod_{j=1}^{k}(1 - z\beta_{j,j})$.

Question 13.12
The Runge-Kutta method of this question is given by the collocation method associated to the nodes $\alpha_1 = (3-\sqrt{3})/6$ and $\alpha_2 = (3+\sqrt{3})/6$ which are the roots of the Legendre polynomial $q(t) = t^2 - t + 1/6$. See Example 13.4.17. It follows from Theorem 13.6.43 that the method is A-stable.

However, the question states that we cannot use this approach. According to Corollary 13.6.36, we have to prove that

$$\{z \in \mathbb{C} : |r(z)| < 1\} \supset \{z \in \mathbb{C} : \operatorname{Im} z < 0\} ,$$

where $r(z) = 1 + z\mathbf{c}^\top(\mathrm{Id} - zB)^{-1}\mathbf{u}$ with

$$B = \begin{pmatrix} 1/4 & (3 - 2\sqrt{3})/12 \\ (3 + 2\sqrt{3})/12 & 1/4 \end{pmatrix} , \quad \mathbf{c} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \quad \text{and} \quad \mathbf{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

We have

$$\mathrm{Id} - zB = \begin{pmatrix} 1 - z/4 & -z(3 - 2\sqrt{3})/12 \\ -z(3 + 2\sqrt{3})/12 & 1 - z/4 \end{pmatrix}$$

and so

$$(\mathrm{Id} - zB)^{-1} = \frac{1}{1 - z/2 + z^2/12} \begin{pmatrix} 1 - z/4 & z(3 - 2\sqrt{3})/12 \\ z(3 + 2\sqrt{3})/12 & 1 - z/4 \end{pmatrix} .$$

Thus

$$\begin{aligned} r(z) &= 1 + \frac{z}{1 - z/2 + z^2/12} \begin{pmatrix} 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 1 - z/4 & z(3 - 2\sqrt{3})/12 \\ z(3 + 2\sqrt{3})/12 & 1 - z/4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= 1 + \frac{z}{1 - z/2 + z^2/12} = \frac{12 + 6z + z^2}{12 - 6z + z^2} . \end{aligned}$$

The poles of $r(z)$ are the roots of the denominator $12 - 6z + z^2$; namely, $z_\pm = 3 \pm \sqrt{3}\,i$. Thus, there is no pole in the region $\{z \in \mathbb{C} : \operatorname{Re} z \le 0\}$. Moreover, if $z = si$, we get

$$|r(si)| = \left|\frac{12 + 6si - s^2}{12 - 6si - s^2}\right| = \left|\frac{12 + 6si - s^2}{\overline{12 + 6si - s^2}}\right| = 1$$

for all $s \in \mathbb{R}$. Thus, $|r(z)| = 1 \le 1$ on the imaginary axis. It follows from Lemma 13.6.40 that the Runge-Kutta Method is A-stable.

Question 13.16

a) The difference equation is

$$w_{i+1} = w_i + \frac{h^2}{12}\left(4(i+1) + 9i - (i-1)\right) = w_i + \frac{h^2}{12}\left(12i + 5\right)$$

for $i \in \mathbb{N}$.

b) First, we find the general solution of the linear difference equation $w_{i+1} = w_i$ for $i \in \mathbb{N}$. If we substitute $w_i = r^i$, we get $r^{i+1} = r^i$. The nontrivial solution is $r = 1$. Thus, the general solution of the linear difference equation is $w_i = C$, a constant, for all i.

We now seek a particular solution of the form $w_i = Ai^2 + Bi$ for the difference equation in (a). We get

$$A(i+1)^2 + B(i+1) = Ai^2 + Bi + \frac{h^2}{12}\left(12i + 5\right) \Rightarrow (2A - h^2)i + \left(A + B - \frac{5h^2}{12}\right) = 0$$

for all i. Thus, $2A - h^2 = 0$ and $A + B - 5h^2/12 = 0$. Solving for A and B, we find $A = h^2/2$ and $B = -h^2/12$. Hence, $w_i = \dfrac{h^2 i^2}{2} - \dfrac{h^2 i}{12}$ for all i.

The general solution of (a) is $w_i = \dfrac{h^2 i^2}{2} - \dfrac{h^2 i}{12} + C$ for all i.

c) With $w_0 = 0$, we get $C = 0$. Thus, $w_i = \dfrac{h^2 i^2}{2} - \dfrac{h^2 i}{12}$ for all i.

d) Since $t_i = hi$, we find $w_i = \dfrac{t_i^2}{2} - \dfrac{h t_i}{12}$.

The solution of the initial value problem (13.8.7) is $y(t) = t^2/2$. Hence,

$$\lim_{h\to 0} \max_{0\le i\le N} |y_i - w_i| = \lim_{h\to 0} \max_{0\le i\le N} |y(t_)) - w_i| = \lim_{h\to 0} \max_{0\le i\le N} \left|\frac{h t_i}{12}\right| \le \lim_{h\to 0} \left|\frac{5h}{12}\right| = 0\ .$$

This is not quite the definition of convergence as stated in Definition 13.6.1 since we have considered w_i instead of u_i, but it is the definition of convergence as given in Remark 13.6.3. See also (f) below.

e) The characteristic polynomial is $p(w) = \lambda^2 - \lambda$. It has two roots, $\lambda = 0$ and $\lambda = 1$ with the root $\lambda = 1$ being simple. So, the root condition is satisfied.

f) The local truncation error is given by

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - \frac{1}{12}\left(4f(t_{i+1}, y(t_{i+1})) + 9f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))\right)\ .$$

Since $y(t_{i+1}) = y(t_i) + y'(t_i)\,h + y''(\xi_i)\,h^2/2$, $f(t_i, y(t_i)) = y'(t_i)$, $f(t_{i+1}, y(t_{i+1})) = y'(t_{i+1}) = y'(t_i) + y''(\eta_i)h$ and $f(t_{i-1}, y(t_{i-1})) = y'(t_{i-1}) = y'(t_i) - y''(\nu_i)h$ for some ξ_i, η_i and ν_i, we get

$$\begin{aligned}\tau_{i+1}(h) &= \frac{1}{h}\left(y(t_i) + y'(t_i)\,h + y''(\xi_i)\,\frac{h^2}{2} - y(t_i)\right)\\ &\quad - \frac{1}{12}\left(4y'(t_i) + 4y''(\eta_i)\,h + 9y'(t_i) - y'(t_i) + y''(\nu_i)\,h\right) = \left(\frac{y''(\xi_i)}{2} - \frac{y''(\eta_i)}{3} - \frac{y''(\nu_i)}{12}\right)h\ .\end{aligned}$$

If $M = \max_{0\leq t\leq 5}|y''(t)|$, we get

$$|\tau_{i+1}(h)| \leq \frac{1}{12}Mh\ .$$

The method is of order 1 because $\tau_{i+1}(h) = O(h)$. The method is consistent because

$$\max_{0\leq i\leq n}|\tau_{i+1}(h)| \leq \frac{5}{12}Mh \to 0$$

as $h \to 0$.

Since the multistep method (13.8.8) is consistent and satisfies the root condition, we may affirm that it is convergent according to Theorem 13.6.26. See Remark 13.6.28.

Question 13.17

We use the method presented in Section 13.5.3 to answer this question.

a) We consider $p(w) = w^3 - 1$ and set $m = 3$. Using the substitution $w = v + 1$, we get

$$\begin{aligned}\frac{p(w)}{\ln(w)} &= \frac{w^3-1}{\ln(w)} = \frac{(v+1)^3-1}{\ln(v+1)} = (v^2+3v+3)\,\frac{v}{\ln(1+v)}\\ &= (v^2+3v+3)\left(1 + \frac{v}{2} - \frac{v^2}{12} + O(v^3)\right) = 3 + \frac{9v}{2} + \frac{9v^2}{4} + O(v^3) = \frac{3}{4} + \frac{9w^2}{4} + O((w-1)^3)\ .\end{aligned}$$

Thus $q(w) = \dfrac{3}{4} + \dfrac{9w^2}{4}$. The multistep method is

$$\begin{aligned}w_{i+1} &= w_{i-2} + h\left(\frac{3}{4}\,f(t_{i-2}, w_{i-2}) + \frac{9}{4}\,f(t_i, w_i)\right) \quad \text{for} \quad i = 2, 3, \ldots, N-1\\ w_i &= y_i \quad \text{for} \quad i = 0, 1, 2\end{aligned}$$

b) We consider $p(w) = w^3 - 1$ and set $m = 3$. Using the substitution $w = v + 1$, we get

$$\begin{aligned}\frac{p(w)}{\ln(w)} &= \frac{w^3-1}{\ln(w)} = \frac{(v+1)^3-1}{\ln(v+1)} = (v^2+3v+3)\,\frac{v}{\ln(1+v)}\\ &= (v^2+3v+3)\left(1 + \frac{v}{2} - \frac{v^2}{12} + \frac{v^3}{24} + O(v^4)\right) = 3 + \frac{9v}{2} + \frac{9v^2}{4} + \frac{3v^3}{8} + O(v^4)\\ &= \frac{3}{8} + \frac{9w}{8} + \frac{9w^2}{8} + \frac{3w^3}{8} + O((w-1)^3)\ .\end{aligned}$$

Thus $q(w) = \dfrac{3}{8} + \dfrac{9w}{8} + \dfrac{9w^2}{8} + \dfrac{3w^3}{8}$. The multistep method is

$$w_{i+1} = w_{i-2} + h\left(\frac{3}{8}\,f(t_{i-2}, w_{i-2}) + \frac{9}{8}\,f(t_{i-1}, w_{i-1}) + \frac{9}{8}\,f(t_i, w_i) + \frac{3}{8}\,f(t_{i+1}, w_{i+1})\right)$$

$$\begin{aligned} &\text{for} \quad i = 3, 4, \ldots, N-1 \\ w_i = y_i \quad &\text{for} \quad i = 0, 1, 2, 3 \end{aligned}$$

Question 13.18
We first prove by induction that

$$p_k(w) = 1 - (-1)^{k-1} \sum_{j=0}^{m} j^{k-1} a_j w^j \tag{16.36}$$

for all $k > 0$. This result is obviously true for $k = 1$. We assume that (16.36) is true for k. Then

$$p_{k+1}(w) = 1 - w p_k'(w) = 1 - w \left(-(-1)^{k-1} \sum_{j=0}^{m} j^k a_j w^{j-1} \right) = 1 - (-1)^k \sum_{j=0}^{m} j^k a_j w^j \ .$$

This is (16.36) with k replaced by $k+1$.

Similarly, we have by induction that

$$q_k(w) = (-1)^{k-1} \sum_{j=-1}^{m} j^{k-1} b_j w^j \tag{16.37}$$

for all $k > 0$. This result is obviously true for $k = 1$. We assume that (16.37) is true for k. Then

$$q_{k+1}(w) = -w q_k'(w) = -w \left((-1)^{k-1} \sum_{j=-1}^{m} j^k b_j w^{j-1} \right) = (-1)^k \sum_{j=-1}^{m} j^k b_j w^j \ .$$

This is (16.37) with k replaced by $k+1$.

It follows from (16.36) that

$$p_k(1) = 1 - (-1)^{k-1} \sum_{j=0}^{m} j^{k-1} a_j$$

for all $k > 0$. It also follows from (16.37) that

$$q_k(1) = (-1)^{k-1} \sum_{j=-1}^{m} j^{k-1} b_j$$

for all k. Hence, it follows from (13.5.13) and (13.5.14) that the multistep method is of order r if and only if $p_1(1) = 0$, $p_{k+1}(1) - k q_k(1) = 0$ for $1 \leq k \leq r$, and $p_{r+2}(1) - (r+1) q_{r+1}(1) \neq 0$.

Question 13.19
a) The local truncation error is given by

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - a_0 y(t_i) - a_1 y(t_{i-1})}{h} - \big(b_0 f(t_i, y(t_i)) + b_1 f(t_{i-1}, y(t_{i-1})) \big) \ .$$

Since there exist ξ_i, η_i and ν_i such that $y(t_{i+1}) = \sum_{k=0}^{5} \frac{1}{k!}\, y^{(k)}(t_i)\, h^k + \frac{1}{6!}\, y^{(6)}(\xi_i)\, h^6$, $y(t_{i-1}) = \sum_{k=0}^{5} \frac{(-1)^k}{k!}\, y^{(k)}(t_i)\, h^k + \frac{1}{6!}\, y^{(6)}(\eta_i)\, h^6$, $f(t_i, y(t_i)) = y'(t_i)$ and $f(t_{i-1}, y(t_{i-1})) = y'(t_{i-1}) = \sum_{k=0}^{4} \frac{(-1)^k}{k!}\, y^{(k+1)}(t_i)\, h^k - \frac{1}{5!}\, y^{(6)}(\mu_i)\, h^5$, we get

$$
\begin{aligned}
\tau_{i+1}(h) = \frac{1}{h}\Bigg(& \sum_{k=0}^{5} \frac{1}{k!}\, y^{(k)}(t_i)\, h^k + \frac{1}{6!}\, y^{(6)}(\xi_i)\, h^6 - a_0 y(t_i) \\
& - a_1 \left(\sum_{k=0}^{5} \frac{(-1)^k}{k!}\, y^{(k)}(t_i)\, h^k + \frac{1}{6!}\, y^{(6)}(\eta_i)\, h^6 \right) \Bigg) \\
& - \left(b_0 y'(t_i) + b_1 \left(\sum_{k=0}^{4} \frac{(-1)^k}{k!}\, y^{(k+1)}(t_i)\, h^k - \frac{1}{5!}\, y^{(6)}(\mu_i)\, h^5 \right) \right) \\
= {} & (1 - a_0 - a_1) y(t_i) h^{-1} + (1 + a_1 - b_0 - b_1) y'(t_i) \\
& + \sum_{k=2}^{5} \left(\frac{1}{k!} - \frac{(-1)^k a_1}{k!} - \frac{(-1)^{k-1} b_1}{(k-1)!} \right) y^{(k)}(t_i) h^{k-1} + O(h^5)
\end{aligned}
$$

if we assume that the derivatives of $y(t)$ are bounded on the interval $[t_0, t_N]$.

We get $a_0 = 1 - a_1$ from $1 - a_0 - a_1 = 0$ (we set the coefficient of $y(t_i)h^{-1}$ to 0). We get $b_0 = 1 + a_1 - b_1$ from $1 + a_1 - b_0 - b_1 = 0$ (we set the coefficient of $y(t_i)$ to 0). We get $b_1 = \frac{a_1}{2} - \frac{1}{2}$ from $\frac{1}{2} - \frac{a_1}{2} + b_1 = 0$ (we set the coefficient of $y''(t_i)h$ to 0). If we substitute this expression for b_1 in $\frac{1}{6} + \frac{a_1}{6} - \frac{b_1}{2} = 0$ (we set the coefficient of $(y^{(3)}(t_i)h^2$ to 0), we get $a_1 = 5$.

We have found that all the terms in h^k for $k < 3$ vanish if $a_0 = -4$, $a_1 = 5$. $b_0 = 4$ and $b_1 = 2$. With these values of a_1 and b_1, we have that $\frac{1}{24} - \frac{a_1}{24} + \frac{b_1}{6} = \frac{1}{6} \neq 0$ (the coefficient of $y^{(4)}h^3$). Hence,

$$
\tau_{i+1}(h) = \frac{1}{6}\, y^{(4)}(t_i)\, h^3 + O(h^4) \ . \tag{16.38}
$$

The method of highest order is

$$
w_{i+1} = -4w_i + 5w_{i-1} + h\left(4f(t_i, w_i) + 2f(t_{i-1}, w_{i-1})\right) \ .
$$

b) We have from (16.38) that $\tau_{i+1}(h) = O(h^3)$. So, the method is of order 3.

c) It follows from Dahlquist Second Barrier, Theorem 13.6.62, that this method cannot be A-stable.

Question 13.20

The stability polynomial for this multistep method is $p(\lambda) - z\, q(\lambda)$, where $p(\lambda) = -\lambda^{m+1} + \sum_{j=0}^{m} a_j\, \lambda^{m-j}$ is the characteristic polynomial and $q(\lambda) = \sum_{j=-1}^{m} b_j\, \lambda^{m-j}$.

Since the method is convergent, it satisfies the root condition according to Proposition 13.6.23; namely, all the roots of its characteristic polynomial have absolute values less than or equal to one and those equal to one are simple roots.

So, for $z = 0$, all the roots of the stability polynomial have absolute values less than or equal to one and those equal to one are simple roots since they are the roots of the characteristic polynomial. Thus, $z = 0$ is in the region of absolute stability or on its boundary.

It follows from Proposition 13.6.11 that $p(1) = 0$. So $\lambda = 1$ is a root of the stability polynomial when $z = 0$. Hence, $z = 0$ is on the boundary of the region of absolute stability.

Chapter 15 : Finite Difference Methods

Question 15.1

Consider a function $v : R_\Delta \to \mathbb{R}$. Let $v_{i,j} = v(x_i, t_j)$ for all $(x_i, t_j) \in R_\Delta$ and let $\frac{1}{2}\left(f(x_i, y_j) + f(x_i, y_{j+1})\right) = P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)$.

We have

$$\frac{v_{i,j+1} - v_{i,j}}{\Delta t} - \frac{c^2}{2}\left(\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\Delta x)^2} + \frac{v_{i+1,j+1} - 2v_{i,j+1} + v_{i-1,j+1}}{(\Delta x)^2}\right) = \frac{1}{2}\left(f(x_i, y_j) + f(x_i, y_{j+1})\right)$$

for $0 < i < N$ and $0 \le j < M$. Thus

$$(1+\alpha)v_{i,j+1} = (1-\alpha)v_{i,j} + \frac{\alpha}{2}\left(v_{i+1,j} + v_{i-1,j}\right) + \frac{\alpha}{2}\left(v_{i+1,j+1} + v_{i-1,j+1}\right) + \frac{1}{2}\left(f(x_i, y_j) + f(x_i, y_{j+1})\right)\Delta t$$

for $0 < i < N$ and $0 \le j < M$, where $\alpha = \dfrac{c^2 \Delta t}{(\Delta x)^2}$.

Let $v_j = \max_{0<i<N} |v_{i,j}|$ and $F = \max_{\substack{0<i<N \\ 0\le j<N}} \frac{1}{2}|f(x_i, y_j) + f(x_i, y_{j+1})|$. If $\alpha \le 1$, we get

$$\begin{aligned}
(1+\alpha)|v_{i,j+1}| &\le (1-\alpha)|v_{i,j}| + \frac{\alpha}{2}\left(|v_{i+1,j}| + |v_{i-1,j}|\right) + \frac{\alpha}{2}\left(|v_{i+1,j+1}| + |v_{i-1,j+1}|\right) \\
&\quad + \frac{1}{2}\left|f(x_i, y_j) + f(x_i, y_{j+1})\right|\Delta t \\
&\le (1-\alpha)v_j + v_j + v_{j+1} + F\Delta t \le v_j + v_{j+1} + F\Delta t
\end{aligned}$$

for $0 < i < N$ and $0 \le j < M$. Thus

$$(1+\alpha)|v_{j+1}| \le v_j + v_{j+1} + F\Delta t \Rightarrow v_{j+1} \le v_j + F\Delta t$$

for $0 \le j < M$. By induction, we get

$$v_j \le v_0 + (j\Delta t)F \le v_0 + TF$$

for $0 \le j \le M$. Hence,

$$|v_{i,j}| \le v_j \le v_0 + TF$$

for $0 \leq j \leq M$ and $0 < i < N$. Since $\dfrac{1}{2}\left(f(x_i,y_j) + f(x_i,y_{j+1})\right) = P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)$ for (i,j) such that $(x_i,t_j) \in R^o_\Delta$ and $B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots) = v_{i,j}$ for (i,j) such that $(x_i,t_j) \in \partial R_\Delta$. We can rewrite the previous inequality as

$$\begin{aligned}
|v_{i,j}| &\leq \max_{0<i<N} |B_\Delta(v_{i,0}, v_{i,1}, v_{i+1,0}, \ldots)| + T \max_{\substack{0<i<N\\0<j\leq M}} |P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)| \\
&\leq \max_{\substack{(i,j)\text{ such that}\\(x_i,t_j)\in\partial R_\Delta}} |B_\Delta(v_{i,0}, v_{i,1}, v_{i+1,0}, \ldots)| + T \max_{\substack{(i,j)\text{ such that}\\(x_i,t_j)\in R^o_\Delta}} |P_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots)|
\end{aligned}$$

for $0 \leq j \leq M$ and $0 < i < N$. Since $B_\Delta(v_{i,j}, v_{i,j+1}, v_{i+1,j}, \ldots) = v_{i,j}$ for (i,j) such that $(x_i,t_j) \in \partial R_\Delta$, we get (15.3.7) with $C = \max\{1, T\}$.

Question 15.2

a) Since $PP^* = P^*P$, we have that

$$\left\langle P^2x, P^2x\right\rangle = \langle Px, P^*PPx\rangle = \langle Px, PP^*Px\rangle = \langle P^*Px, P^*Px\rangle \ .$$

Thus

$$\|P^2\| = \sup_{\|x\|=1} \left\|P^2x\right\| = \sup_{\|x\|=1} \sqrt{\langle P^2x, P^2x\rangle} = \sup_{\|x\|=1} \sqrt{\langle P^*Px, P^*Px\rangle} = \sup_{\|x\|=1} \|P^*Px\| = \|P^*P\| \ .$$

b) We first prove that

$$\|P^*P\| = \sup_{\|x\|=\|y\|=1} \langle x, P^*Py\rangle \ . \tag{16.39}$$

Using Schwartz's inequality, we have

$$\langle x, P^*Py\rangle \leq \|x\| \ \|P^*Py\| \leq \|x\| \ \|P^*P\| \ \|y\| = \|P^*P\|$$

for all x and y such that $\|x\| = \|y\| = 1$. Thus

$$\sup_{\|x\|=\|y\|=1} \langle x, P^*Py\rangle \leq \|P^*P\| \ . \tag{16.40}$$

Moreover

$$\sup_{\|x\|=\|y\|=1} \langle x, P^*Py\rangle \geq \sup_{\substack{\|y\|=1\\x=\|P^*Py\|^{-1}P^*Py}} \langle x, P^*Py\rangle == \sup_{\|y\|=1} \|P^*Py\| = \|P^*P\| \ . \tag{16.41}$$

because $\|x\| = 1$ for $x = \|P^*Py\|^{-1} P^*Py$. Thus (16.39) follows from (16.40) and (16.41).

We have that

$$\|P^*P\| = \sup_{\|x\|=\|y\|=1} \langle x, P^*Py\rangle = \sup_{\|x\|=\|y\|=1} \langle Px, Py\rangle \geq \sup_{\|y\|=1} \langle Py, Py\rangle = \|P\|^2 \ .$$

We get from (a) that $\left\|P^2\right\| \geq \|P\|^2$. But we already know that $\left\|P^2\right\| \leq \|P\|^2$ by a property of bounded linear operators. Thus $\left\|P^2\right\| = \|P\|^2$.

Question 15.3

We use Proposition 15.3.36 to answer this question.

As we have seen in Example 15.3.31, the finite difference scheme given by Algorithm 15.2.1 can be expressed as $\mathbf{w}_{j+1} = Q\mathbf{w}_j + \mathbf{B}_j$ for $j \geq 0$, where $Q = -K$ for K given in (15.2.6) and

$$\mathbf{B}_j = \begin{pmatrix} \alpha w_{0,j} + f(x_1, t_j)\Delta t \\ f(x_2, t_j)\Delta t \\ \vdots \\ f(x_{N-2}, t_j)\Delta t \\ \alpha w_{N,j} + f(x_{N-1}, t_j)\Delta t \end{pmatrix} .$$

The matrix Q can be written as $Q = \mathrm{Id} + \alpha A$, where

$$A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix}$$

is a $(N-1) \times (N-1)$ matrix. It follows from Proposition 15.4.1 that the eigenvalues of A are

$$\lambda_i = -2 + 2\cos(k\pi/N) = -4\sin^2(k\pi/(2N)) \quad , \quad 0 < k < N \ .$$

Thus, the eigenvalues of Q are $1 + \alpha\lambda_k$ for $0 < k < N$. To get eigenvalues that are smaller or equal to 1 in absolute value, we need to have $|1 + \alpha\lambda_k| \leq 1$ for $0 < k < N$; namely, we need to have

$$-1 \leq 1 - 4\alpha\sin^2(k\pi/(2N)) \leq 1 \quad , \quad 0 < k < N \ .$$

The second inequality is always satisfied, so α must satisfy

$$-1 \leq 1 - 4\alpha\sin^2(k\pi/(2N)) \quad , \quad 0 < k < N \ .$$

This is equivalent to

$$0 < \alpha \leq \frac{1}{2\sin^2(k\pi/(2N))} \quad , \quad 0 < k < N \ .$$

However,

$$\frac{1}{2\sin^2(k\pi/(2N))} > \frac{1}{2}$$

for $0 < k < N$ and converges to $1/2$ for $k = N-1$ and $N \to \infty$.

Thus, the finite difference scheme is ℓ^2-stable if $0 < \alpha \leq 1/2$.

Bibliography

[1] L. Ammerall, **Computer Graphics for Java Programmers**, John Wiley & Sons, 1998.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, **Numerical Solution of Boundary Value Problems for Ordinary Differential Equations**, SIAM, 1995.

[3] D. Assaf, IV, and S. Gadbois, Definitions of Chaos, **The American Mathematical Monthly**, **99**, No. 9, p. 865.

[4] J. Banks, J. Brooks, G. Cairns, G. Davis and P. Stacey, On Deveney's Definition of Chaos, **The American Mathematical Monthly**, **99**, No. 4, pp. 332-334.

[5] R. G. Bartle, **The Elements of Real Analysis, $\mathbf{2}^{nd}$ Edition**, John Wiley & Sons, 1976.

[6] J. L. Buchanan and P. R. Turner, **Numerical Methods and Analysis**, McGraw-Hill, Inc., 1992.

[7] R. L. Burden and J. D. Faires, **Numerical Analysis,** 5^{th} **ed.**, PWS-KENT Publ. Comp., 1993.

[8] J. C. Butcher, **The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods**, Wiley, 1987.

[9] S. D. Conte, The numerical solution of linear boundary value problems, SIAM Review, **8**, 1966, pp. 309–321.

[10] S. D. Conte and C. de Boor, **Elementary Numerical Analysis**, McGraw-Hill, 1980.

[11] J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, **Math. Comp.**, vol. **19**, 1965, pp. 297–301.

[12] R. L. Devaney, **An Introduction to Chaotic Dynamical Systems**, The Benjamin/Cummings Publ. Co. Inc., 1986.

[13] G. H. Golub and C. F. van Loan, **Matrix Computation,** 3^{rd} **ed.**, John Hopkins Univ. Press, 1996

[14] E. Hairer, S. P. Norsett and G. Wanner, **Solving Ordinary Differential Equations I: Nonstiff Problems**, Springer-Verlag, 1991.

[15] E. Hairer and G. Wanner, **Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems**, Springer-Verlag, 1996.

[16] P. Henrici, Fast Fourier methods in computational complex analysis, SIAM Review, **21**, 1979, pp. 481–527.

[17] Morris W. Hirsch and Stephen Smale, **Differential Equations, Dynamical Systems, and Linear Algebra**, Academic Press, 1974.

[18] Wolfgang Hackbusch, **The Concept of Stability in Numerical Mathematics**, Springer-Verlag, 2014.

[19] A. Iserles, **A First Course in the Numerical Analysis of Differential Equations**, Cambridge Univ. Press, 1996.

[20] E. Issacson and H. B. Keller, **Analysis of Numerical Methods**, Dover Publ. Inc., 1994.

[21] D. Kincaid and W. Cheney, **Numerical Analysis, Mathematics of Scientific Computing, 3^{th} ed.**, Brookes/Cole, 2002.

[22] H. B. Keller, **Numerical Methods for Two-Point Boundary-Value Problems**, Dover Publ. Inc., 1992

[23] J. D. Lambert, **Computational Methods in Ordinary Differential Equations**, Wiley, 1973.

[24] J. D. Lambert, **Numerical Methods for Ordinary Differential Systems**, Wiley, 1991.

[25] J. M. Ortega, (1968). "The Newton-Kantorovich Theorem", **The American Mathematical Monthly**, **99**, No. 4, pp. 658–660.

[26] H.-O. Peitgen, J. Hartmut and S. Dietmar, **Chaos and Fractals: New Frontiers of Science**, Springer-Verlag, 1992.

[27] Walter Rudin, **Principle of Mathematical Analysis, 3^{rd} Edition**, McGraw-Hill, 1978.

[28] L. F. Shampine, **Numerical Solution of Ordinary Differential Equations**, Chapman & Hill, 1994.

[29] G. F. Simmons, **Differential Equations with Applications and Historical Notes**, McGraw-Hill, 1972.

[30] G. Strang, **Introduction to Applied Mathematics**, Wellesley-Cambridge Press, 1986.

[31] G. D. Smith, **Numerical Solution of Partial Differential Equations, 2nd Edition**, Oxford University Press, 1978.

[32] W. A. Strauss, **Partial Differential Equations: an Introduction**, John-Wiley & Sons, 1992.

[33] A. M. Stuart and A. R. Humphries, **Dynamical Systems and Numerical Analysis**, Cambridge Univ. Press, 1996.

Index