A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

EFOP-3.4.3-16-2016-00009

Introduction to Numerical Analysis

Ferenc Hartung

H-8200 Veszpr Egyetem u.10.
H-8201 Veszpr Pf. 158.
Telefon: (+36 88) 624-848
Internet: www.uni-pannon.hu

2018

Table of contents

Chapter 1
Introduction

In this chapter we discuss first the main objectives of numerical analysis and introduce some basic notions. We investigate different sources of errors in scientific computation, define the notion of stability of a mathematical problem or a numerical algorithm, the time and memory resources needed to perform the algorithm. We study the computer representation of integer and real numbers, and present some numerical problems due to the finite-digit arithmetic.

1.1. The Main Objective and Notions of Numerical Analysis

In scientific computations the first step is the mathematical modeling of the physical process. This is the task of the particular scientific discipline (physics, chemistry, biology, economics, etc.). The resulting model frequently contains parameters, constants, initial data which are typically determined by observations or measurements. If the mathematical model and its parameters are given, then we can use it to answer some questions related to the physical problem. We can ask qualitative questions (Does the problem have a unique solution? Does the solution have a limit as the time goes to infinity? Is the solution periodic? etc.), or we can ask quantitative questions (What is the value of the physical variable at a certain time? What is the approximate solution of the model?). The qualitative questions are discussed in the related mathematical discipline, but the quantitative questions are the main topics of numerical analysis. The main objective of numerical analysis is to give exact or approximate solutions of a mathematical problem using arithmetic operations (addition, subtraction, multiplication and division). See Figure 1.1 below for the schematic steps of the scientific computation of physical processes.

The numerical value of a physical quantity computed by a process described in Figure 1.1 is, in general, not equal to the real value of the physical quantity. The sources of error we get is divided into the following two main categories: *inherited error* and *computational error*. The mathematical modeling is frequently a simplification of the physical reality, so we generate an inherited error when we replace the physical problem by a mathematical model. This kind of error is called *modeling error*. An other subclass of the inherited error is what we get when we determine the parameters of the mathematical model by measurements, so we use an approximate parameter value instead of the true one. This is called *measurement error*.

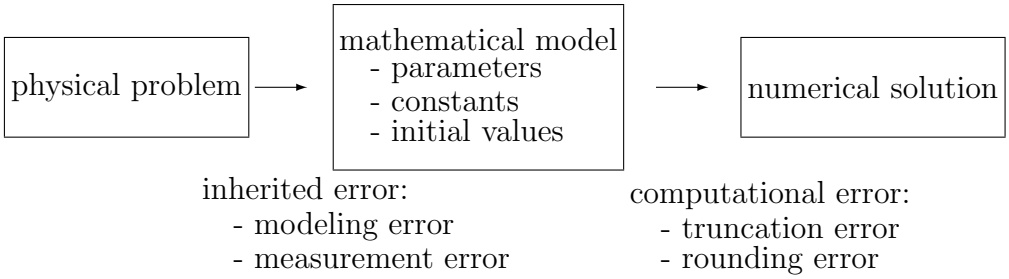

Figure 1.1: Scientific computations

The computational error is divided into two classes: *truncation error* and *rounding error*. We get a truncation error when we replace the exact value of a mathematical expression with an approximate formula.

Example 1.1. Suppose we need to compute the value of the function $f(x) = \sin x$ at a certain argument x. We can do it using arithmetic operations if instead of the function value $f(x)$ we compute, e.g., its Taylor-polynomial around 0 of degree 5: $T_5(x) = x - x^3/3! + x^5/5!$. The Taylor's theorem (Theorem 2.5 below) says that if $f(x)$ is replaced by $T_5(x)$, then the resulting error has the form $\frac{f^{(6)}(\xi)}{6!}x^6 = -\frac{\sin\xi}{6!}x^6$, where ξ is a number between 0 and x. This is the truncation error of the approximation, which is small if x is close to 0. □

The rounding error appears since real numbers can be stored in computers with finite digits accuracy. Therefore, we almost always generate a rounding error when we store a real number in a computer. Also, after computing each arithmetic operations, the computer rounds the result to a number which can be stored in the computer (see Sections 1.2–1.4).

When we specify a numerical algorithm, the first thing we have to investigate is the truncation error, since a numerical value is useful only if we know how large is the error of the approximation. The next notion we discuss related to a numerical algorithm is the *stability*. This notion is used in two meanings in numerics. We can talk about the *stability of a mathematical model* or about the *stability of a numerical method*. First we consider an example.

Example 1.2. Consider the linear system

$$8x + 917y = 1794$$
$$7x + 802y = 1569.$$

Its exact solution is $x = -5$ and $y = 2$. But if we change the coefficient of the variable x in the second equation to 7.01, then the solution of the corresponding system

$$8x + 917y = 1794$$
$$7.01x + 802y = 1569$$

is $x = -1.232562589$ and $y = 1.967132499$ (up to 9 decimal digits precision). We observe that 0.14% change in the size of a single coefficient results in 75.3% and 1.6% changes in the solutions, rspectively. □

We say that a mathematical problem is *correct* or *stable* or *well-conditioned*, if a "small" change in the parameters of the problem results only in "small" change in the solution of the problem. In the opposite case we say that the problem is *incorrect* or *ill-conditioned* or *unstable problem*. The linear system in the previous example is an incorrect mathematical problem.

We say that a numerical algorithm is *stable with respect to rounding errors* if the rounding errors do not influence the result of the computation significantly. If the computed result is significantly different from the true value, then we say that the *algorithm is unstable*. Next we present an example of an an unstable algorithm.

Example 1.3. Consider the following three recursive sequences:

$$\begin{aligned} x_n &= \frac{1}{3}x_{n-1}, \qquad x_0 = 1, \\ y_n &= 2y_{n-1} - \frac{5}{9}y_{n-2}, \qquad y_0 = 1, \quad y_1 = \frac{1}{3}, \\ z_n &= \frac{13}{3}z_{n-1} - \frac{4}{3}z_{n-2}, \qquad z_0 = 1, \quad z_1 = \frac{1}{3}. \end{aligned} \tag{1.1}$$

It is easy to show that all the three recursions generate the same sequence $x_n = y_n = z_n = \frac{1}{3^n}$, i.e., the three sequences are algebraically equivalent. But in practice, the numerical computations of the three recursions give different results. In Table 1.1 the first 18 terms of the sequences are displayed. The computations are performed using single precision floating point arithmetic in order to enlarge the effect of the rounding errors. We observe that the sequence x_n produce the numerical values of $1/3^n$, but the numerical values of y_n and z_n are different from it due to the accumulation of the rounding error. Both sequences has rounding error, but for the sequence z_n the error increases so rapidly, that in the 18th term it is of order 10^2. In this case the numerical values of z_n do not converge to 0. We experience that the sequence x_n is a stable method, but z_n is an unstable method to compute the values of $1/3^n$.

To check that the errors we observed in the previous computation are the consequence of the rounding errors, we repeated the generation of the three sequences but now using a double precision floating point arithmetic. We present here the error of the 18th terms: $|y_{18} - 1/3^{18}| = -2.5104e-13$ and $|z_{18} - 1/3^{18}| = 2.3804e-07$. We can observe that the magnitude of the errors are much smaller in this case. □

In case of an algorithm which terminates in a finite number of steps we are usually interested in the *time complexity* or the *cost* of an algorithm. By this we mean the number of steps, or more precisely, the *number of arithmetic operations* needed to perform the algorithm. Consider first an example.

Example 1.4. Evaluate numerically the polynomial $p(x) = 5x^4 - 8x^3 + 2x^2 + 4x - 10$ at the point x. Certainly, we can do it using the formula of p literally. It contains 4 additions/subtractions, 4

Table 1.1:

n	x_n	y_n	$\lvert y_n - 1/3^n\rvert$	z_n	$\lvert z_n - 1/3^n\rvert$
2	0.111111	0.111111	2.2352e-08	0.111111	4.4703e-08
3	0.037037	0.037037	4.0978e-08	0.037037	1.8254e-07
4	0.012346	0.012346	6.9849e-08	0.012346	7.3109e-07
5	0.004115	0.004115	1.1688e-07	0.004118	2.9248e-06
6	0.001372	0.001372	1.9465e-07	0.001383	1.1699e-05
7	0.000457	0.000458	3.2442e-07	0.000504	4.6795e-05
8	0.000152	0.000153	5.4071e-07	0.000340	1.8718e-04
9	0.000051	0.000052	9.0117e-07	0.000800	7.4872e-04
10	0.000017	0.000018	1.5019e-06	0.003012	2.9949e-03
11	0.000006	0.000008	2.5032e-06	0.011985	1.1980e-02
12	0.000002	0.000006	4.1721e-06	0.047920	4.7918e-02
13	0.000001	0.000008	6.9535e-06	0.191674	1.9167e-01
14	0.000000	0.000012	1.1589e-05	0.766693	7.6669e-01
15	0.000000	0.000019	1.9315e-05	3.066773	3.0668e+00
16	0.000000	0.000032	3.2192e-05	12.267091	1.2267e+01
17	0.000000	0.000054	5.3653e-05	49.068363	4.9068e+01
18	0.000000	0.000089	8.9422e-05	196.273453	1.9627e+02

multiplications and 3 exponentials. The exponentials mean 3+2+1=6 number of multiplications, i.e., altogether 10 multiplications are needed to apply the formula of p. But we can rewrite p as follows:

$$p(x) = 5x^4 - 8x^3 + 2x^2 + 4x - 10 = (((5x - 8)x + 2)x + 4)x - 10.$$

This form of the polynomial requires only 4 additions/subtractions and 4 multiplications. □

The previous method can be extended to polynomials of degree n:

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = ((\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_1)x + a_0$$

This formula requires only n additions/subtractions and n multiplications. This way of organizing a polynomial evaluation is called *Horner's method*. The method can be defined by the following algorithm.

Algorithm 1.5. Horner's method

INPUT: n - degree of the polynomial
$a_n, a_{n-1}, \ldots, a_0$ - coefficients of the polynomial
x - argument
OUTPUT: y - function value of the polynomial at the argument x

$y \leftarrow a_n$
for $i = n-1, \ldots, 0$ **do**
  $y \leftarrow yx + a_i$
end do
output(y)

The execution of a multiplication or division requires more time than that of an addition or subtraction. Therefore, in numerical analysis, we count the number of multiplications and divisions separately to the number of additions and subtractions.

It is also important to know the *space complexity* of an algorithm, which is the amount of the memory storage needed in the worst case at any point in the algorithm. When we work with an algorithm to solve a linear system with a 10×10 coefficient matrix, the storage cannot be a problem. But the same with a 10000×10000 dimensional matrix can be problematic. In case of algorithms working with a big amount of data, we prefer a method which requires less amount of memory space. For example, if in a matrix nonzero elements appear only in the main diagonal and in some diagonals above and below, then it is practical to use an algorithm which utilizes the special structure of the data, and does not store the unnecessary zeros in the matrix during the computation. We will see such methods in Section 3.5 below.

1.2. Computer Representation of Integers and Reals

Let I be a positive integer with a representation in base b number system with m number of digits:

$$I = (a_{m-1}a_{m-2}\dots a_1a_0)_b, \qquad \text{where} \quad a_i \in \{0, 1, \dots, b-1\}.$$

Its value is

$$I = a_{m-1}b^{m-1} + a_{m-2}b^{m-2} + \cdots + a_1b + a_0.$$

Therefore, the largest integer can be represented with m digits is $I_{\max}$ where all digits equal to $b-1$. Its numerical value is

$$I_{\max} = (b-1)(b^{m-1} + b^{m-2} + \cdots + b + 1) = b^m - 1.$$

Hence on m digits we can represent (store) integers from 0 up to $b^m - 1$, which is b^m number of different integers.

Suppose we use a base 2, other words, *binary* number system. Then on m bits we can store 2^m number of integers. We describe two methods to store negative integers. The first method is the *sign-magnitude representation*. Here we allocate a *sign bit* (typically the *most significant bit*, i.e., the left-most bit), which is 0 for positive integers and 1 for negative integers. Then on the rest of the $m-1$ bits we can store the magnitude or absolute value of the number. Then $I_{\max} = 2^{m-1} - 1$ is the largest, and $I_{\min} = -I_{\max}$ is the smallest integer which can be represented. In this system the integer 0 can be stored as an identically 0 bit sequence or as $100\dots0$.

Example 1.6. In Table 1.2 we listed all the integers which can be represented on $m = 3$ bits using the sign-magnitude binary representation. □

In practice the *two's-complement representation* is frequently used to store signed integers. Let I be an integer which we would like to represent on m bits. Instead of I we store the binary form of the number C defined by

$$C = \begin{cases} I, & \text{if } 0 \le I \le 2^{m-1} - 1, \\ 2^m + I, & \text{if } -2^{m-1} \le I < 0. \end{cases}$$

Table 1.2: Sign-magnitude binary representation on $m = 3$ bits

I	a binary code
0	000
1	001
2	010
3	011
0	100
-1	101
-2	110
-3	111

Here the largest and the smallest representable integer is $I_{\max} = 2^{m-1} - 1$ and $I_{\min} = -2^{m-1}$, respectively. Therefore, if $0 \leq I \leq 2^{m-1} - 1$, then $C < 2^{m-1}$, i.e., the first bit of C is 0. On the other hand, if $-2^{m-1} \leq I < 0$, then it is easy to see that $2^{m-1} \leq C \leq 2^m - 1$, i.e., the first bit of C is 1.

An important advantage of the two's-complement representation is that the subtraction can be obtained as an addition (see Exercise 4).

Example 1.7. Table 1.3 contains all the integers which can be represented on $m = 3$ bits using the two's-complement binary representation. □

Table 1.3: Two's-complement representation on $m = 3$ bits

I (in decimal)	I (in binary)	C, the stored binary
0	000	000
1	001	001
2	010	010
3	011	011
-1	-001	111
-2	-010	110
-3	-011	101
-4	-100	100

Next we discuss the representation of real numbers. We recall that the real number in base b number system

$$x = (x_{m-1}x_{m-2}\cdots x_0.x_{-1}x_{-2}\cdots)_b, \qquad x_i \in \{0, 1, \ldots, b-1\}$$

has the numerical value

$$x = x_{m-1}b^{m-1} + x_{m-2}b^{m-2} + \cdots x_1 b + x_0 + \frac{x_{-1}}{b} + \frac{x_{-2}}{b^2} + \cdots = \sum_{i=-\infty}^{m-1} x_i b^i.$$

Consider the real number 126.42. Different books define the normal form of this number as $1.2642 \cdot 10^2$ or $0.12642 \cdot 10^3$. In this lecture notes we use the first form as the normal form. Therefore, the *normal form* of a real number $x \neq 0$ in a base b number system is $x = \pm m \cdot b^k$, where $1 \leq m < b$. The number m is called the *mantissa*, and k the *exponent* of the number. In order to represent a real number, or in other words, a *floating*

point number we write it in a normal form in a base b number system, and we would like to store its signed mantissa and exponent. Computers use different number of bits to store these numbers. Here we present an IEEE specification[1] to represent floating point numbers on 32 bits (the so-called *single precision*), and on 64 bits (the *double precision*) using the binary number system. This representation is used in IBM PCs. Consider the binary normal form $x = (-1)^s m \cdot 2^k$, where $s \in \{0, 1\}$ and $m = (1.m_1m_2m_3\ldots)_2$. The value s is stored in the 1st bit. Instead of the exponent k, we store its shifted value, the nonnegative integer $e = k + 127$ on bits 2–9. In our definition of the binary normal form, a nonzero x has a mantissa of the form $m = (1.m_1m_2\ldots)_2$, i.e., it always starts with 1, which we do not store, we store the fractional digits of the mantissa rounded to 23 bits. These 23 bits are stored on bits 10–32 of the storage. This IEEE specification defines the representation of the number 0, and introduces two special symbols, `Inf` (to store infinity as a possible value) and `NaN` (not-a-number) in the following way:

number	s	e (bits 2–9)	mantissa bits (bits 10–32)
$+0$	0	00000000	every mantissa bit=0
-0	1	00000000	every mantissa bit=0
$+$`Inf`	0	11111111	at least one mantissa bit=0
$-$`Inf`	1	11111111	at least one mantissa bit=0
$+$`NaN`	0	11111111	every mantissa bit=1
$+$`NaN`	1	11111111	every mantissa bit=1

The symbol `Inf` can be used in programs as a result of a mathematical operation with value ∞, and the symbol `NaN` can be a result of a mathematical operation which is undefined (e.g., a division by 0 or a root of a negative number in real numbers). Both symbols can be positive or negative. The definition yields that the exponent $e = (11111111)_2 = 255$ is used exclusively for the special symbols `Inf` and `NaN`. For finite reals the possible values are $0 \le e \le 254$, hence the possible values of the exponent k are $-127 \le k \le 127$. Therefore, the smallest positive representable number corresponds to exponent $k = -127$ and mantissa $(1.00\ldots01)_2$. Hence its value is $x_{\min} = (1 + 1/2^{23})2^{-127} \approx 10^{-38}$. The largest real can be stored is $x_{\max} = (1.11\ldots1)_2 2^{127} = (2 - 2^{-23})2^{127} \approx 10^{38}$.

The representation on 64 bits is similar: the shifted exponent $e = k + 1023$ is stored on bits 2–12, the fractional part of the mantissa is stored on bits 13–64. Then the range of real numbers which can be stored in the computer is, approximately, from 10^{-308} to 10^{308}.

Example 1.8. Suppose we would like to store reals on 4 bits using a binary normal form. For example, we use the 1st bit as the sign bit, the shifted exponent $e = k + 1$ is stored on the 2nd bit, and the fractional part of the mantissa is stored on bits 3–4. (The symbols `Inf` and `NaN` are not defined now.) The nonnegative real numbers which can be represented by the above method are listed in Table 1.4, and are illustrated in Figure 1.2.

□

We can see that, using any floating point representation, we can store only finitely many reals on a computer. The reals which can be stored without an error in a certain floating point representation are called *machine numbers*. The machine number which is

[1] IEEE Binary Floating Point Arithmetic Standard, 754-1985.

Table 1.4: Nonnegative reals on 4 bits.

s	e	m	x
0	0	00	0
0	0	01	$(1.01)_2 \cdot 2^{-1} = (1+\frac{1}{4})\frac{1}{2} = \frac{5}{8}$
0	0	10	$(1.10)_2 \cdot 2^{-1} = (1+\frac{1}{2})\frac{1}{2} = \frac{3}{4} = \frac{6}{8}$
0	0	11	$(1.11)_2 \cdot 2^{-1} = (1+\frac{1}{2}+\frac{1}{4})\frac{1}{2} = \frac{7}{8}$
0	1	00	$(1.00)_2 \cdot 2^{0} = 1 = \frac{8}{8}$
0	1	01	$(1.01)_2 \cdot 2^{0} = 1+\frac{1}{4} = \frac{10}{8}$
0	1	10	$(1.10)_2 \cdot 2^{0} = 1+\frac{1}{2} = \frac{12}{8}$
0	1	11	$(1.11)_2 \cdot 2^{0} = 1+\frac{1}{2}+\frac{1}{4} = \frac{7}{4} = \frac{14}{8}$

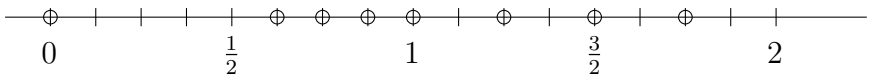

Figure 1.2: Nonnegative machine numbers on 4 bits.

stored in a computer instead of the real number x is denoted by $\mathrm{fl}(x)$. If $|x|$ is smaller than the smallest positive machine number, then, by definition, $\mathrm{fl}(x) = 0$, and if $|x|$ is larger than the largest positive machine number, then we define $\mathrm{fl}(x) = \texttt{Inf}$ for $x > 0$ and $\mathrm{fl}(x) = -\texttt{Inf}$ for $x < 0$. In the first case we talk about arithmetic *underflow*, and in the second case, about arithmetic *overflow*. The definition of $\mathrm{fl}(x)$ in the intermediate cases can be different in different computers. There are basically two methods: In the first case we take the binary normal form of x, consider its mantissa $m = (1.m_1m_2m_3\ldots)_2$, and we consider as many first several mantissa fractional bits as it is possible to store in the particular representation. We store them, and omit the rest of the mantissa bits. This method is called *chopping* of the mantissa. For example, using the single precision representation defined above, we store the first 23 mantissa fractional bits.

The other method, the *rounding*, is more frequently used to define the mantissa bits of the machine number $\mathrm{fl}(x)$. Here the mantissa of $\mathrm{fl}(x)$ is defined so that $\mathrm{fl}(x)$ be the nearest machine number to x. In case when x is exactly an average of two consecutive machine numbers, we could round down or up. The IEEE specification for single precision representation we defined above uses the following rule: Let the normal form of a positive real x be $x = m2^k$, where $m = (1.m_1m_2\ldots m_{23}m_{24}\ldots)_2$. Let $x' = (1.m_1m_2\ldots m_{23})_2 2^k$ and $x'' = \big((1.m_1m_2\ldots m_{23})_2 + 2^{-23}\big)2^k$. Then x' and x'' are consecutive machine numbers, $x' \le x \le x''$ and $x'' - x' = 2^{k-23}$. The specification defines

$$\mathrm{fl}(x) = \begin{cases} x', & \text{if } |x-x'| < \frac{1}{2}|x''-x'|, \\ x'', & \text{if } |x-x''| < \frac{1}{2}|x''-x'|, \\ x', & \text{if } |x-x'| = \frac{1}{2}|x''-x'| \text{ and } m_{23} = 0, \\ x'', & \text{if } |x-x'| = \frac{1}{2}|x''-x'| \text{ and } m_{23} = 1. \end{cases}$$

In the critical case, i.e., if $|x-x'| = \frac{1}{2}|x''-x'|$, approximately half of the cases we round down and in the other cases we round up. An other reason for this definition is that in

this critical case the last mantissa bit is always 0, so a division by 2 can be performed on $\mathrm{fl}(x)$ without an error. Using the rounding, the error is

$$|x - \mathrm{fl}(x)| \le \frac{1}{2}|x'' - x'| = \frac{1}{2}2^{-23}2^k.$$

If we compare it to the exact value we get

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \le \frac{|x - \mathrm{fl}(x)|}{(1.m_1m_2\ldots)_2 \cdot 2^k} \le \frac{1}{2}2^{-23}.$$

We can see that the first machine number which is larger than 1 is $1 + 2^{-23}$ in the single precision floating point arithmetic. Let ε_{m} denote the difference of the first machine number right to 1 and the number 1. This number is called *machine epsilon*. Therefore, ε_{m} is the smallest power of 2 (in a binary storage system) for which the computer evaluates the inequality $1 + \varepsilon_{\mathrm{m}} > 1$ to be true. The following theorem can be proved similarly to the consideration above for the single precision floating point representation.

Theorem 1.9. *Let* $0 < \mathrm{fl}(x) <$ `Inf`*, and suppose the floating point representation uses rounding. Then*

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \le \frac{1}{2}\varepsilon_m.$$

The proof of the next result is left for Exercise 5.

Theorem 1.10. *Suppose we use a number system with base b, and let t be the number of mantissa bits in the floating point representation. Then*

$$\varepsilon_m = \begin{cases} 2^{-t}, & \text{if } b = 2, \\ b^{1-t}, & \text{if } b \neq 2. \end{cases}$$

Now we define the notion of the error of an approximation and other related notions. Let x be a real number, and consider $\tilde{x}$ as its approximation. Then the *absolute error* or just simply the *error* of the approximation is the number $|x - \tilde{x}|$. Frequently the error without knowing the magnitude of the numbers does not mean too much. For example, 10000.1 can be considered as a good approximation of 10000, but in general, 1.1 is not considered as a good approximation of 1, but in both cases the errors are the same, 0.1. We may get more information if we compare the error to the exact value. The *relative error* is defined by

$$\frac{|x - \tilde{x}|}{|x|} \qquad (x \neq 0).$$

We say that in a base b number system $\tilde{x}$ is *exact in n digits* if

$$\frac{|x - \tilde{x}|}{|x|} \le \frac{1}{2}b^{1-n}.$$

We can see that the smaller is the relative error, the larger is the number of exact digits. In a decimal number system ($b = 10$) we can formulate the relation between the relative error and the number of exact digits in the following way: if the relative error decreases by a factor of 1/10, then the number of exact digits increases by 1.

Example 1.11. Let $x = 1657.3$ and $\tilde{x} = 1656.2$. Then the absolute error is $|x - \tilde{x}| = 1.1$, and the relative error is $|x - \tilde{x}|/x = 0.0006637$ (with 7 decimal digits precision). Since $|x - \tilde{x}|/x = 0.0006637 < 0.5 \cdot 10^{-2}$, the approximation is exact in 3 digits. On the other hand, if x is approximated by the value $\tilde{x} = 1656.9$, then $|x - \tilde{x}|/x = 0.0002413 < 0.5 \cdot 10^{-3}$, and hence the approximation is exact in 4 digits. □

The previous definition and Theorem 1.9 yield that the single precision floating point arithmetic is exact in 24 binary digits. Usually, we are interested in the exact number of digits in a decimal number system. In case of a single precision floating point arithmetic, we get it if we find the largest integer n for which

$$\frac{1}{2}2^{-23} \leq \frac{1}{2}10^{1-n}.$$

It can be computed that $n = 7$ is the number of exact digits for a single precision floating point arithmetic.

Example 1.12. Consider $x = 12.4$. First rewrite it in binary form: $12 = (1100)_2$. Find the binary form of its fractional part:

$$0.4 = (0.x_1x_2x_3\ldots)_2 = \frac{x_1}{2} + \frac{x_2}{2^2} + \frac{x_3}{2^3} + \cdots .$$

If we multiply 0.4 by 2, then its integer part gives x_1. $0.4 \cdot 2 = 0.8$, hence $x_1 = 0$. Consider the fractional part of this product, 0.8, and we repeat the previous procedure. $0.8 \cdot 2 = 1.6$, so $x_2 = 1$. The fractional part of the product is 0.6, which gives: $0.6 \cdot 2 = 1.2$, and therefore, $x_3 = 1$. The fractional part of the product is 0.2. We have $0.2 \cdot 2 = 0.4$, hence $x_4 = 0$, and we continue with 0.4. We can see that the digits 0011 will be repeated periodically infinitely many times, i.e., $0.4 = (0.01100110011001100110011001100110011\ldots)_2$. The binary normal form of x is

$$x = 12.4 = (1.10001100110011001100110011001100110011\ldots)_2 \cdot 2^3.$$

Rounding the mantissa to 23 bits (down) we get

$$\mathrm{fl}(x) = (1.10001100110011001100110)_2 \cdot 2^3.$$

Its numerical value (in a decimal form) is $\mathrm{fl}(x) = 12.3999996185302734375$. □

The arithmetic operations performed by a computer can be defined as

$$\begin{aligned} x \oplus y &:= \mathrm{fl}(\mathrm{fl}(x) + \mathrm{fl}(y)), \\ x \ominus y &:= \mathrm{fl}(\mathrm{fl}(x) - \mathrm{fl}(y)), \\ x \odot y &:= \mathrm{fl}(\mathrm{fl}(x) \cdot \mathrm{fl}(y)), \\ x \oslash y &:= \mathrm{fl}(\mathrm{fl}(x) / \mathrm{fl}(y)). \end{aligned}$$

Here we always take the machine representation of each operands and also the result of the arithmetic operation.

In later examples we will use the so-called *4-digit rounding arithmetic* or simply *4-digit arithmetic*. By this we mean a floating point arithmetic using a decimal number system with 4 stored mantissa digits (and suppose we can store any exponent). This means that, in every step of a calculation, the result is rounded to the first 4 *significant digits*, i.e., from the first nonzero digits for 4 digits, and this rounded number is used in the next arithmetic operation. We can enlarge the effect of rounding errors in such a way.

Example 1.13. Using a 4-digit arithmetic we get $1.043 + 32.25 = 33.29$, and similarly, $1.043 \cdot 32.25 = 33.64$ (after rounding). But $1.043 + 20340 = 20340$, since we rounded the exact value 20341.043 to for significant digits. □

Exercises

1. Convert the following decimal numbers to binary form:
$$57, \quad -243, \quad 0.25, \quad 35.27$$
2. Convert the binary numbers to decimal form:
$$(101101)_2, \quad (0.10011)_2, \quad (1010.01101)_2$$
3. Show that the two's-complement representation of a negative integer can be computed in the following way: Take the binary form of the absolute value of the number. Change all 0's to 1's and all 1's to 0's, and add 1 to the resulting number.
4. Let I_1 and I_2 be two positive integers with m bits. Show that $I_1 - I_2$ can be computed if we first consider the two's-complements representation C_2 of I_2, add I_1 to it, and finally, take the last m bits of the result.
5. Prove Theorem 1.10.
6. Write a computer code which gives back the machine epsilon of the particular computer.
7. Compute the exact number of digits of a machine number in case of a double precision floating point arithmetic.
8. Let $x = (x_0.x_1x_2\dots x_mx_{m+1}x_{m+2}\dots)\cdot 10^k$, $\tilde{x} = (x_0.x_1x_2\dots x_m\tilde{x}_{m+1}\tilde{x}_{m+2}\dots)\cdot 10^k$, i.e., x and $\tilde{x}$ has the same order of magnitude, and its first $m+1$ digits are the same. Show that, in this case, $\tilde{x}$ is an approximation of x with at least m number of exact digits.

1.3. Error Analysis

Let x and y be positive real numbers, and consider the numbers $\tilde{x}$ and $\tilde{y}$ as an approximation of x and y. Let $|x - \tilde{x}| \le \Delta_x$ and $|y - \tilde{y}| \le \Delta_y$ be the error bounds of the approximation. The relative error bounds are denoted by $\delta_x := \Delta_x/x$ and $\delta_y := \Delta_y/y$, respectively. In this section we examine the following question: We would like to perform an arithmetic operation (addition, subtraction, multiplication or division) on the real numbers x and y, but instead of it, we perform the operation on the numbers $\tilde{x}$ and $\tilde{y}$

(suppose without an error). We will consider this latter number as an "approximation" of the original one. We will examine the error and the relative error of this "approximation".

Consider first the addition. We are looking for error bounds Δ_{x+y} and δ_{x+y} such that

$$|x+y-(\tilde{x}+\tilde{y})| \le \Delta_{x+y} \quad \text{and} \quad \frac{|x+y-(\tilde{x}+\tilde{y})|}{x+y} \le \delta_{x+y}.$$

Theorem 1.14. *The numbers*

$$\Delta_{x+y} := \Delta_x + \Delta_y \qquad \textit{and} \qquad \delta_{x+y} := \max\{\delta_x, \delta_y\}$$

are absolute and relative error bounds of the addition, respectively.

Proof. Using the triangle inequality and the definitions of Δ_x and Δ_y, we get

$$|x+y-(\tilde{x}+\tilde{y})| \le |x-\tilde{x}| + |y-\tilde{y}| \le \Delta_x + \Delta_y.$$

This means that $\Delta_x + \Delta_y$ is an upper bound of the error of the addition.

Using the above relation, we obtain

$$\begin{aligned} \frac{|x+y-(\tilde{x}+\tilde{y})|}{x+y} &\le \frac{\Delta_x+\Delta_y}{x+y} \\ &= \frac{x}{x+y}\delta_x + \frac{y}{x+y}\delta_y \\ &\le \max\{\delta_x, \delta_y\}. \end{aligned}$$

Therefore, $\max\{\delta_x, \delta_y\}$ is a relative error bound of the addition. □

Clearly, the above theorem can be generalized for addition of several numbers: the error bounds will be added, and the relative error bound is the maximum value of the relative error bounds. We can reformulate this result as follows: the number of exact digits of the approximation of the sum is at least the smallest of the number of exact digits of the approximations of the operands. Certainly, the theorem gives the worst case estimate. In practice the errors can balance each other. For example, let $x = 1$, $y = 2$, $\tilde{x} = 1.1$ and $\tilde{y} = 1.8$. Then $x + y = 3$ and $\tilde{x} + \tilde{y} = 2.9$. Therefore, the error of the sum is only 0.1, smaller than the sum of the error of the terms, 0.3.

Theorem 1.15. *Let $x > y > 0$. The numbers*

$$\Delta_{x-y} := \Delta_x + \Delta_y \qquad \textit{and} \qquad \delta_{x-y} := \frac{x}{x-y}\delta_x + \frac{y}{x-y}\delta_y$$

are absolute and relative error bounds of the subtraction.

Proof. The inequalities

$$|x-y-(\tilde{x}-\tilde{y})| \le |x-\tilde{x}| + |y-\tilde{y}| \le \Delta_x + \Delta_y$$

imply the first statement. Consider

$$\frac{|x-y-(\tilde{x}-\tilde{y})|}{x-y} \leq \frac{\Delta_x+\Delta_y}{x-y} = \frac{x}{x-y}\delta_x + \frac{y}{x-y}\delta_y,$$

which gives the second statement. □

We can observe that if we subtract two nearly equal numbers, then the relative error can be magnified compared to the relative error of the terms. In other words, the number of exact digits can be significantly less that in the original numbers. This phenomenon is called *loss of significance*.

Example 1.16. Let $x = 12.47531$, $\tilde{x} = 12.47534$, $y = 12.47326$ and $\tilde{y} = 12.47325$. Then $\delta_x = 2.4 \cdot 10^{-6}$ and $\delta_y = 8 \cdot 10^{-7}$. On the other hand, $x - y = 0.00205$, $\tilde{x} - \tilde{y} = 0.00209$, and so $\delta_{x-y} = 0.0195$. We can check that $\tilde{x}$ and $\tilde{y}$ are exact in 6 digits, but $\tilde{x} - \tilde{y}$ is exact only in 2 digits. □

Theorem 1.17. *Let $x, y > 0$. The numbers*

$$\Delta_{x\cdot y} := x\Delta_y + y\Delta_x + \Delta_x\Delta_y, \qquad \textit{and} \qquad \delta_{x\cdot y} := \delta_x + \delta_y + \delta_x\delta_y$$

are absolute and relative error bounds of the multiplication, respectively.

Proof. The triangle-inequality and simple algebraic manipulations yield

$$\begin{aligned} |xy - \tilde{x}\tilde{y}| &= |xy - x\tilde{y} + x\tilde{y} - \tilde{x}\tilde{y}| \\ &\leq x|y-\tilde{y}| + |\tilde{y}||x-\tilde{x}| \\ &\leq x\Delta_y + |\tilde{y}|\Delta_x \\ &= x\Delta_y + |y + \tilde{y} - y|\Delta_x \\ &\leq x\Delta_y + y\Delta_x + \Delta_x\Delta_y, \end{aligned}$$

hence the first statement is proved. Therefore, we get

$$\frac{|xy - \tilde{x}\tilde{y}|}{xy} \leq \frac{x\Delta_y + y\Delta_x + \Delta_x\Delta_y}{xy} = \delta_x + \delta_y + \delta_x\delta_y,$$

which implies the second statement. □

Since, in general, Δ_x and Δ_y are much smaller than x and y, and so $\Delta_x\Delta_y$ is much smaller than $x\Delta_y$ and $y\Delta_x$, we have that $x\Delta_y + y\Delta_x$ is a good approximation of the absolute error of the multiplication. Similarly, $\delta_x + \delta_y$ is a good approximation of the relative error of the multiplication. Both results mean that the errors do not propagate rapidly in multiplication.

Theorem 1.18. *Suppose $x, y > 0$ and $\delta_y < 1$. Then the numbers*

$$\Delta_{x/y} := \frac{x\Delta_y + y\Delta_x}{y(y-\Delta_y)} \qquad \text{and} \qquad \delta_{x/y} := \frac{\delta_x + \delta_y}{1-\delta_y}$$

are absolute and relative error bounds of the division, respectively.

Proof. Elementary manipulations give

$$\left|\frac{x}{y} - \frac{\tilde{x}}{\tilde{y}}\right| = \frac{|x\tilde{y} - xy + xy - \tilde{x}y|}{y|\tilde{y}|} \le \frac{x\Delta_y + y\Delta_x}{y|\tilde{y}|} = \frac{x\Delta_y + y\Delta_x}{y|y-(y-\tilde{y})|}.$$

Assumption $\delta_y < 1$ implies $|y-\tilde{y}| \le \Delta_y < y$, hence $|y-(y-\tilde{y})| \ge y - |y-\tilde{y}| \ge y - \Delta_y > 0$ proves the first statement.

For the second part, consider

$$\frac{\left|\frac{x}{y} - \frac{\tilde{x}}{\tilde{y}}\right|}{\frac{x}{y}} = \frac{|x(\tilde{y}-y) - y(\tilde{x}-x)|}{x|\tilde{y}|} = \frac{\left|\frac{\tilde{y}-y}{y} - \frac{\tilde{x}-x}{x}\right|}{\left|1 - \frac{y-\tilde{y}}{y}\right|} \le \frac{\delta_x + \delta_y}{1-\delta_y}.$$

□

If δ_y is small, then the relative error bound of the division can be approximated well by $\delta_x + \delta_y$. Similarly, if Δ_y is much smaller than y, then $\frac{1}{y}\Delta_x + \frac{x}{y^2}\Delta_y$ is a good approximation of $\Delta_{x/y}$. If y is much smaller than x, or if y is close to 0, then Δ_y or Δ_x can be significantly magnified, so the absolute error can be much larger than the absolute error of the terms.

Exercises

1. Let $x = 3.50$, $y = 10.00$, $\tilde{x} = 3.47$, $\tilde{y} = 10.02$. Estimate the absolute and relative error of
$$3x + 7y, \quad \frac{1}{y}, \quad x^2, \quad y^3, \quad \frac{4xy}{x+y}$$
(without evaluating the expressions) assuming we replace x and y by $\tilde{x}$ and $\tilde{y}$. Then compute the expressions numerically and compute the absolute and relative errors exactly. Compare them with the estimates.
2. Let $\tilde{x}$ be an approximation of x, and $|x-\tilde{x}| \le \Delta_x$. Let $f\colon \mathbb{R} \to \mathbb{R}$ be a differentiable function satisfying $|f'(x)| \le M$ for all $x \in \mathbb{R}$. Let $y = f(x)$ and consider $\tilde{y} = f(\tilde{x})$ as an approximation of y. Estimate the absolute error of the approximation. (Hint: Use the Lagrange's Mean Value Theorem.)

1.4. The Consequences of the Floating Point Arithmetic

Example 1.19. Solve the equation

$$x^2 - 83.5x + 1.5 = 0$$

using 4-digit arithmetic in the computations.

Using the quadratic formula and the 4-digit arithmetic we get the numerical values

$$\tilde{x} = \frac{83.5 \pm \sqrt{83.5^2 - 4 \cdot 1.5}}{2} = \frac{83.5 \pm \sqrt{6972 - 6.000}}{2} = \frac{83.5 \pm 83.46}{2},$$

hence

$$\tilde{x}_1 = \frac{167.0}{2} = 83.50, \quad \text{and} \quad \tilde{x}_2 = \frac{0.040}{2} = 0.020.$$

We can check that the exact solutions (up to several digits precision) are $x_1 = 83.482032$ and $x_2 = 0.0179679$. Using the relative error bounds for each roots we get $\delta_1 = 0.0002152$ and $\delta_2 = 0.113096$. The first root is exact in 4 digits, but the second is only in 1 digits. So there is a significant difference between the order of the magnitudes of the relative errors. What is the reason of it? In the computation of the second root, we subtracted two close numbers. This is the point where we significantly lost the accuracy. □

Consider the second root of $ax^2 + bx + c = 0$:

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{1.2}$$

When b is negative and $4ac$ is much smaller than b^2, then we subtract two nearly equal numbers, and we observe the loss of significance. (This happened for the second root in Example 1.19.) To avoid this problem, consider

$$x_2 = \frac{b^2 - (b^2 - 4ac)}{2a(-b + \sqrt{b^2 - 4ac})} = \frac{2c}{-b + \sqrt{b^2 - 4ac}}. \tag{1.3}$$

This formula is algebraically equivalent to formula (1.2). But the difference is that here we do not subtract two close numbers (in the denominator we add two positive numbers). If b is positive, then for the first root we get

$$x_1 = \frac{2c}{-b - \sqrt{b^2 - 4ac}}. \tag{1.4}$$

Example 1.20. Compute the second root of the equation of Example 1.20 using 4-digit arithmetic and formula (1.3).

$$\tilde{x}_2 = \frac{2 \cdot 1.5}{83.5 + \sqrt{83.5^2 - 4 \cdot 1.5}} = \frac{3}{83.5 + 83.46} = \frac{3}{167.0} = 0.01796.$$

The relative error of x_2 is now $\delta_2 = 0.00044$, hence the exact number of digits is 4. □

Example 1.21. Suppose we need to evaluate the expression $\cos^2 x - \sin^2 x$. If $x = \frac{\pi}{4}$, then the exact value of this expression is 0, hence if x is close to $\frac{\pi}{4}$, then in the expression we need to subtract to nearly equal numbers, so we can face loss of significance. We can avoid it if, instead of the original formula, we evaluate its equivalent form, $\cos 2x$. □

In the previous examples we used algebraic manipulations to avoid the loss of significance. Now we show different techniques.

Example 1.22. Consider the function $f(x) = e^x - 1$. In the neighborhood of $x = 0$ we again need to subtract two nearly equal numbers, but here we cannot use an algebraic identity to avoid it. But here we can consider the Taylor series of the exponential function, and we get

$$f(x) = x + \frac{x}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots .$$

It is worth to take a finite approximation of this infinite series, and use it as an approximation of the function value $f(x)$. □

The next example shows a different problem.

Example 1.23. Evaluate the number $y = 20^{50}/50!$. The problem is the following: If we compute the numerator and the denominator separately first, then we run into the problem of overflowing the calculation if we use single precision floating point arithmetic. On the other hand, we know that $a^n/n! \to 0$ as $n \to \infty$, so the result must be a small number. We rearrange the computation as follows:

$$\frac{20^{50}}{50!} = \frac{20}{50} \cdot \frac{20}{49} \cdot \frac{20}{48} \cdots \frac{20}{1}.$$

Here in each steps the expressions we need to evaluate belong to the range which can be stored in the computer. This formula can be computed with a simple **for** cycle:

$y \leftarrow 20$
for $i = 2, \ldots, 50$ **do**
  $y \leftarrow y \cdot \frac{20}{i}$
end do
output(y)

The result is 3.701902 (with 7-digits precision). □

Example 1.24. Compute the sum

$$A = 10.00 + 0.002 + 0.002 + \cdots + 0.002 = 10.00 + \sum_{i=1}^{10} 0.002$$

using a 4-digit arithmetic. We perform the additions from left to right, so first we need to compute $10.00 + 0.002$. But with a 4-digit arithmetic the result is $10.00 + 0.002 = 10.002 = 10.00$ after the rounding. Adding the next number to it, because of the rounding to 4 digits, we get again $10.00 + 0.002 + 0.002 = 10.00$. Hence we get the numerical result $A = 10.00$.

Consider the same sum, but in another order:

$$B = 0.002 + 0.002 + \cdots + 0.002 + 10.00 = \sum_{i=1}^{10} 0.002 + 10.00.$$

First we need to compute $0.002 + 0.002 = 0.004$. The result is exact even if we use 4-digit arithmetic. Then we can continue: $0.002 + 0.002 + 0.002 = 0.006$ etc., and finally, $\sum_{i=1}^{10} 0.002 =$

0.02. Therefore, the numerical result will be $B = 10.02$. Here we have not observed any rounding error, since we could compute the result in each step exactly.

This example demonstrates that the addition using a floating point arithmetic is not a commutative operation numerically. □

A conclusion of the previous example is that in computing sums with several terms, it is advantageous to do the computation in an increasing order of the terms, since in that case we have a better chance for that the terms have similar order of magnitude, so the loss of significance has less chance.

Exercises

1. Investigate that in the next example what are the cases when we can observe the loss of significance. How can we avoid it?
   (a) $\ln x - 1$,
   (b) $\sqrt{x+4} - 2$,
   (c) $\sin x - x$,
   (d) $1 - \cos x$,
   (e) $(1 - \cos x)/\sin x$,
   (f) $(\cos x - e^{-x})/x$,
2. Compute the next expression using a 4-digit arithmetic $2.274 + 12.04 + 0.4233 + 0.1202 + 0.2204$, and then sort the terms in an increasing way, and repeat the calculation.

Chapter 2
Nonlinear Algebraic Equations and Systems

In this chapter we investigate numerical solution of scalar nonlinear algebraic equations and systems of nonlinear algebraic equations. We discuss the methods of bisection, false position, secant, Newton and quasi-Newton. We introduce the basic theory of fixed points, the notion of the speed of convergence, stopping criteria of iteration methods. We define the notion of vector and matrix norms, and discuss convergence of vector sequences.

2.1. Review of Calculus

In this section we summarize some basic results and notions from Calculus which will be needed in our later sections.

$C[a,b]$ will denote the set of continuous real valued functions defined on the interval $[a,b]$. $C^m[a,b]$ will denote the set of continuous real valued functions $f\colon\ [a,b] \to \mathbb{R}$, which are m-times continuously differentiable on the open interval (a,b).

Theorem 2.1. *Let $f \in C[a,b]$. Then f has its maximum and minimum on the interval $[a,b]$, i.e., there exist $c, d \in [a,b]$, such that*

$$f(c) = \max_{x\in[a,b]} f(x) \qquad \textit{and} \qquad f(d) = \min_{x\in[a,b]} f(x).$$

The open interval spanned by the numbers a and b is denoted by $\langle a,b\rangle$, i.e., $\langle a,b\rangle := (\min\{a,b\}, \max\{a,b\})$. In general, $\langle a_1, a_2, \ldots, a_n\rangle$ denotes the open interval spanned by the numbers a_1, a_2, ..., a_n, i.e.,

$$\langle a_1, a_2, \ldots, a_n\rangle := (\min\{a_1, a_2, \ldots, a_n\}, \max\{a_1, a_2, \ldots, a_n\}).$$

The next result, the so-called Intermediate Value Theorem, states that a continuous function takes any value in between two function values.

Theorem 2.2 (Intermediate Value Theorem). *Let $f \in C[a,b]$, $f(a) \neq f(b)$, and let $d \in \langle f(a), f(b)\rangle$. Then there exists $c \in (a,b)$ such that $f(c) = d$.*

Theorem 2.3 (Rolle's Theorem). *Let $f \in C^1[a,b]$ and $f(a) = f(b)$. Then there exists $\xi \in (a,b)$ such that $f'(\xi) = 0$.*

Theorem 2.4 (Lagrange's Mean Value Theorem). *Let $f \in C^1[a,b]$. Then there exists $\xi \in (a,b)$ such that $f(b) - f(a) = f'(\xi)(b-a)$.*

Theorem 2.5 (Taylor's Theorem). *Let $f \in C^{n+1}[a,b]$, and let $x_0 \in (a,b)$. Then for every $x \in (a,b)$ there exists $\xi = \xi(x) \in \langle x, x_0 \rangle$ such that*

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}.$$

The next result is called the Mean Value Theorem for integrals.

Theorem 2.6. *Let $f \in C[a,b]$, $g\colon [a,b] \to \mathbb{R}$ is integrable which has no sign change on $[a,b]$ (i.e., $g(x) \geq 0$ or $g(x) \leq 0$ holds for all $x \in [a,b]$). Then there exists $\xi \in (a,b)$ such that*

$$\int_a^b f(x)g(x)\,dx = f(\xi)\int_a^b g(x)\,dx.$$

The next result is called Cantor's Intersection Theorem.

Theorem 2.7. *Let $[a_n, b_n]$ $(n = 1, 2, \ldots)$ be a sequence of closed and bounded intervals, for which $[a_{n+1}, b_{n+1}] \subset [a_n, b_n]$ holds for all n and $(b_n - a_n) \to 0$ as $n \to \infty$. Then there exists a $c \in [a_1, b_1]$ such that $a_n \to c$ and $b_n \to c$ as $n \to \infty$.*

Theorem 2.8. *A monotone and bounded real sequence has a finite limit.*

We close this section with a result from algebra, which we state in the following form.

Theorem 2.9 (Fundamental Theorem of Algebra). *Any nth-degree polynomial*

$$p(x) = a_n x^n + \cdots a_1 x + a_0, \qquad a_j \in \mathbb{C}\ (j = 0, \ldots, n), \quad a_n \neq 0$$

has exactly n complex roots with counting multiplicities.

We will use the following consequence of the previous result. If a polynomial of the form $p(x) = a_n x^n + \cdots + a_1 x + a_0$ has $n+1$ different roots, then $p(x) = 0$ for all $x \in \mathbb{R}$.

2.2. Fixed-Point Iteration

Many numerical methods generate an infinite sequence whose limit gives the exact solution of the investigated problem. The sequence is frequently defined by a *recursion* or *iteration*. A recursion of the form $p_{k+1} = h(p_k, p_{k-1}, \ldots, p_{k-m+1})$ $(k \geq m-1)$ is called *m-order*

recursion or *m-step iteration*. An m-step iteration is well-defined if m number of initial values p_0, p_1, ..., p_{m-1} are given.

In this section we study the one-step iteration, the so-called *fixed-point iteration*. Given a function $g\colon I \to \mathbb{R}$, where $I \subset \mathbb{R}$. The recursive sequence $p_{k+1} = g(p_k)$ which corresponds to an initial value $p_0 \in I$ is called a *fixed-point iteration*.

Example 2.10. Consider the function $g(x) = -\frac{1}{8}x^3 + x + 1$. In Table 2.1 we computed the first few terms of the fixed-point iteration starting from the value $p_0 = 0.4$. We illustrate the sequence in Figure 2.1. Such picture is called *stair step diagram* or *Cobweb diagram*. From the starting point $(p_0, 0)$ we draw a vertical line segment to the graph of g. The second coordinate of the intersection gives p_1. From the point (p_0, p_1) we draw a horizontal line segment to the point (p_1, p_1) on the line $y = x$. Now we get the value $p_2 = g(p_1)$ as the second coordinate of the intersection of the vertical line starting from this point and the graph of g. Continuing this procedure we get the figure displayed in Figure 2.1. The line segments spiral and get closer and closer to the intersection of the graphs of the line $y = x$ and the function g. The coordinates of the intersection is $(2, 2)$. From Table 2.1 it can be seen that the sequence p_k converges to 2. □

Table 2.1: Fixed-point iteration, $g(x) = -\frac{1}{8}x^3 + x + 1$

k	p_k
0	0.40000000
1	1.39200000
2	2.05484646
3	1.97030004
4	2.01419169
5	1.99275275
6	2.00358428
7	1.99819822
8	2.00089846
9	1.99955017
10	2.00022477
11	1.99988758
12	2.00005620
13	1.99997190
14	2.00001405
15	1.99999297

In the previous example we observed that the fixed-point iteration converged to the first coordinate of the intersection of the graphs of the line $y = x$ and the function $y = g(x)$. The first (and also the second) coordinate of this point satisfies the equation $g(x) = x$. The number p is called the *fixed point* of the function g if it satisfies

$$g(p) = p.$$

Using this terminology in the previous example the fixed-point iteration converged to the fixed point of the function g. The next result shows that this is true for all convergent fixed-point iterations if the function g is continuous.

Theorem 2.11. *Let $g\colon [a, b] \to [a, b]$ (or $\mathbb{R} \to \mathbb{R}$) be a continuous function, $p_0 \in [a, b]$ be fixed, and consider the fixed-point iteration $p_{k+1} = g(p_k)$. If p_k is convergent and $p_k \to p$, then $p = g(p)$.*

Proof. Since $p_{k+1} = g(p_k)$ and $p_{k+1} \to p$ by the assumptions, the continuity of g yields $g(p_k) \to g(p)$ as $k \to \infty$, hence the statement follows. □

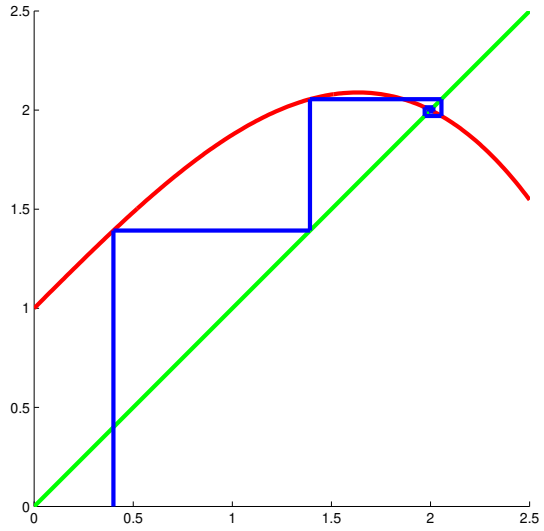

Figure 2.1: Fixed-point iteration

A fixed-point iteration is not always convergent, or the limit is not necessary finite. To see that it is enough to consider the function $g(x) = 2x$ and the initial value $p_0 = 1$. Then $p_k = 2^k$, and it converges to infinity. And if we consider $g(x) = -x$ and $p_0 = 1$, then the corresponding fixed-point sequence is $p_k = (-1)^k$, which is not convergent.

The next theorem gives sufficient conditions for the existence and uniqueness of the fixed point.

Theorem 2.12. *Let $g : [a, b] \to [a, b]$ be continuous. Then g has a fixed point in the interval $[a, b]$. Moreover, if g is differentiable on (a, b), and there exists a constant $0 \le c < 1$ such that $|g'(x)| \le c$ for all $x \in (a, b)$, then this fixed point is unique.*

Proof. Consider the function $f(x) = g(x) - x$. If $f(a) = 0$ or $f(b) = 0$, then a or b is a fixed point of g. Otherwise, $f(a) > 0$ and $f(b) < 0$. But then the continuity of f and the Intermediate Value Theorem imply that there exists a $p \in (a, b)$, such that $f(p) = 0$, i.e., $p = g(p)$.

For the proof of the uniqueness, suppose that g has two fixed points p and q. Then it follows from the Lagrange's Mean Value Theorem that there exists a $\xi \in (a, b)$ such that

$$|p - q| = |g(p) - g(q)| = |g'(\xi)||p - q| \le c|p - q|.$$

But this yields that $p = q$, i.e., the fixed point is unique. □

Theorem 2.13 (fixed-point theorem). *Let $g\colon [a, b] \to [a, b]$ be continuous, g is differentiable on (a, b), and suppose that there exists a constant $0 \le c < 1$ such that $|g'(x)| \le c$ for all $x \in (a, b)$. Let $p_0 \in [a, b]$ arbitrary, and $p_{k+1} = g(p_k)$ $(k \ge 0)$. Then the sequence p_k converges to the unique fixed point p of the function g,*

$$|p_k - p| \le c^k |p_0 - p|, \tag{2.1}$$

and

$$|p_k - p| \le \frac{c^k}{1-c}|p_1 - p_0|. \tag{2.2}$$

Proof. Theorem 2.12 implies that g has a unique fixed point p. Since $0 \le c < 1$ by our assumptions, the convergence $p_k \to p$ follows from (2.1). To show (2.1), we have from the assumptions and the Lagrange's Mean Value Theorem that

$$|p_k - p| = |g(p_{k-1}) - g(p)| = |g'(\xi)||p_{k-1} - p| \le c|p_{k-1} - p|.$$

Now mathematical induction gives relation (2.1) easily.

To prove (2.2), let $m > k$ be arbitrary. Then the triangle inequality, the Mean Value Theorem and our assumptions imply

$$\begin{aligned}
|p_k - p_m| &\le |p_k - p_{k+1}| + |p_{k+1} - p_{k+2}| + \cdots + |p_{m-1} - p_m| \\
&\le |g(p_{k-1}) - g(p_k)| + |g(p_k) - g(p_{k+1})| + \cdots + |g(p_{m-2}) - g(p_{m-1})| \\
&\le c|p_{k-1} - p_k| + c|p_k - p_{k+1}| + \cdots + c|p_{m-2} - p_{m-1}| \\
&\le (c^k + c^{k+1} + \cdots + c^{m-1})|p_0 - p_1| \\
&= c^k(1 + c + \cdots + c^{m-k-1})|p_1 - p_0| \\
&\le c^k \sum_{i=0}^{\infty} c^i |p_1 - p_0|.
\end{aligned}$$

Hence $|p_k - p_m| \le \frac{c^k}{1-c}|p_1 - p_0|$ holds for all $m > k$. Keeping k fixed and tending with m to ∞, we get (2.2). □

We remark that in the proof of the previous two theorems, the differentiability of g and the boundedness of the derivative is used only to get the estimate

$$|g(x) - g(y)| \le c|x - y|. \tag{2.3}$$

We say that the function $g\colon I \to \mathbb{R}$ is *Lipschitz continuous* on the interval I, or in other words, it has the *Lipschitz property* if there exists a constant $c \ge 0$ such that (2.3) holds for all $x, y \in I$. The constant c in (2.3) is called the *Lipschitz constant* of the function g.

Clearly, if g is Lipschitz continuous on I, then it is also continuous on I. From the Lagrange's Mean Value Theorem we get that if $g \in C^1[a,b]$, then g is Lipschitz continuous on $[a,b]$ with the Lipschitz constant $c := \max\{|g'(x)| : x \in [a,b]\}$. g is also Lipschitz continuous if it is only piecewise continuously differentiable. One example is the function $g(x) = |x|$. If g is Lipschitz continuous with a Lipschitz constant $0 \le c < 1$, then g is called a *contraction*. Theorem 2.13 can be stated in the following more general form.

Theorem 2.14 (contraction principle). *Let the function $g\colon [a,b] \to [a,b]$ be a contraction, $p_0 \in [a,b]$ be arbitrary, and $p_{k+1} = g(p_k)$ $(k \ge 0)$. Then the sequence p_k converges to the unique fixed point p of the function g, and relations (2.1) and (2.2) are satisfied.*

In numerics we frequently encounter with iterative methods which converge assuming the initial value is close enough to the exact solution of the problem, i.e., to the limit

of the sequence. We introduce the following notion. We say that the iteration $p_{k+1} = h(p_k, p_{k-1}, \dots, p_{k-m+1})$ *converges locally* to p if there exists a constant $\delta > 0$, such that for every initial value $p_0, p_1, \dots, p_{m-1} \in (p-\delta, p+\delta)$ the corresponding sequence p_k converges to p. If the iteration p_k converges to p for every initial value, then this iteration method is called *globally convergent*.

Theorem 2.15. *Let $g \in C^1[a,b]$, and let $p \in (a,b)$ be a fixed point of g. Suppose also that $|g'(p)| < 1$. Then the fixed-point iteration converges locally to p, i.e., there exists a $\delta > 0$ such that $p_{k+1} = g(p_k)$ converges to p for all $p_0 \in (p-\delta, p+\delta)$.*

Proof. Since g' is continuous and $|g'(p)| < 1$, there exists a $\delta > 0$ such that $[p-\delta, p+\delta] \subset (a,b)$ and $|g'(x)| < 1$ for $x \in [p-\delta, p+\delta]$. Let $c := \max\{|g'(x)| \colon x \in [p-\delta, p+\delta]\}$. Then $0 \le c < 1$.

We show that g maps the interval $[p-\delta, p+\delta]$ into itself. Let $p_0 \in [p-\delta, p+\delta]$. The Lagrange's Mean Value Theorem and the definition of c yield

$$|g(p_0) - p| = |g(p_0) - g(p)| \le c|p_0 - p| < |p_0 - p| < \delta,$$

i.e., $g(p_0) \in [p-\delta, p+\delta]$. Therefore, Theorem 2.13 can be applied for the function g restricting it to the interval $[p-\delta, p+\delta]$, which proves the result. □

Exercises

1. Let $g(x) = mx$, where $m \in \mathbb{R}$. Draw the stair step diagram of the fixed-point iteration corresponding to g and to any non-zero initial value for the parameter values $m = 0.5, 1, 1.5, -0.5, -1, -1.5$.
2. Rewrite the following equation as a fixed-point equation, and approximate its solution by a fixed-point iteration with a 4-digit accuracy.

   (a) $(x-2)^3 = x+1$, (b) $\frac{\cos x}{x} = 2$,

   (c) $x^3 + x - 1 = 0$, (d) $2x \sin x = 4 - 3x$.

3. Consider the equation $x^3 + x^2 + 3x - 5 = 0$. Show that the left hand side is monotone increasing, and has a root on the interval $[0,2]$. (It is easy to see that the exact root is $x = 1$.) Verify that the equation is equivalent to all of the following fixed-point problems.

   (a) $x = x^3 + x^2 + 4x - 5$, (b) $x = \sqrt[3]{5 - x^2 - 3x}$,

   (c) $x = \frac{5}{x^2+x+3}$, (d) $x = \frac{5-x^3}{x+3}$,

   (e) $x = \frac{2x^3+x^2+5}{3x^2+2x+3}$, (f) $x = \frac{5+7x-x^2-x^3}{10}$.

   Compute the first several terms of the associated fixed-point iteration using the starting value $p_0 = 0.5$, and determine if we get a convergent sequence from this starting value. Compare the speed of the convergence of the sequences.
4. Prove that the recursion $p_k = \frac{1}{2}p_{k-1} + \frac{1}{p_{k-1}}$ converges to $\sqrt{2}$, if $p_0 > \sqrt{2}$. What do we get if $0 < p_0 < \sqrt{2}$, or if $p_0 < 0$?
5. Prove that the sequence $p_k = \frac{1}{2}p_{k-1} + \frac{A}{2p_{k-1}}$ converges to $\sqrt{A}$, if $p_0 > 0$. What happens if $p_0 < 0$?

6. Let $g \in C^1(a,b)$, and let $p \in (a,b)$ be a fixed point of g, and $|g'(p)| > 1$. Show that the fixed-point iteration does not converge to p, if $p_0 \neq p$.
7. Consider $g(x) = \sqrt{1+x^2}$. Show that $|g'(x)| < 1$ for all $x \in \mathbb{R}$, but the fixed-point does not converge for any starting value p_0.
8. Let $f\colon [a,b] \to \mathbb{R}$ be continuous, and let $a = x_0 < x_1 < \cdots < x_n = b$ be mesh points such that f is linear on each interval $[x_i, x_{i+1}]$ $(i = 0, \ldots, n-1)$. Show that f is Lipschitz continuous.

2.3. Bisection Method

In this and in the next several sections we study the numerical solution of the scalar nonlinear algebraic equation $f(x) = 0$. One of the simplest algorithm to approximate its solution is the *bisection method*.

We suppose that $f\colon [a,b] \to \mathbb{R}$ is a continuous function of opposite sign at the end of the interval, i.e., $f(a)f(b) < 0$. Then the Intermediate Value Theorem yields that f has at least one root inside the interval $[a,b]$. We define a sequence of intervals: Let $[a_0,b_0] = [a,b]$, and let p_0 be the midpoint of the interval, i.e., $p_0 = (a_0+b_0)/2$. Then either $f(p_0) = 0$, or one of the intervals $[a_0,p_0]$ or $[p_0,b_0]$ has the property that the function f takes opposite sign at the end points of the interval. If f changes sign on the interval $[a_0,p_0]$, then we define $[a_1,b_1] = [a_0,p_0]$, otherwise let $[a_1,b_1] = [p_0,b_0]$. Continuing this procedure, either after finitely many steps, p_k is a root of the function f, or we define an infinite sequence of nested closed bounded intervals, so that a root of f is contained in each of the intervals. We have that the length of the kth interval $(b-a)/2^k$ tends to 0 as $k \to \infty$. But then the Cantors's nested intervals theorem shows that there exists $p \in [a,b]$ such that $a_k \to p$ and $b_k \to p$ as $k \to \infty$, and p is the only common point of the intervals. So, in particular, the sequence of midpoints, p_k also tends to p.

Suppose, e.g., that $f(a) < 0$ and $f(b) > 0$ (the other case can be treated similarly). Then for all k we have $f(a_k) < 0$ and $f(b_k) > 0$. Since $a_k \to p$ and $b_k \to p$, the continuity of f implies $f(p) \leq 0$ and $f(p) \geq 0$, hence $f(p) = 0$. Since $a_k \leq p \leq b_k$ is satisfied for all k, we get $|p_k - p| \leq (b_k - a_k)/2 = (b-a)/2^{k+1}$. We have proved the following result.

Theorem 2.16. *Let $f \in C[a,b]$ and $f(a)f(b) < 0$. Then the bisection sequence p_k converges to a root p of the function f, and*

$$|p_k - p| \leq \frac{b-a}{2^{k+1}}. \tag{2.4}$$

It follows from the estimate (2.4) that if we predefine a tolerance (error bound) $\varepsilon > 0$, then p_k is an approximation of p within this tolerance if its index k satisfies

$$k \geq \log_2 \frac{b-a}{\varepsilon} - 1. \tag{2.5}$$

Example 2.17. Consider the function $f(x) = e^x - 2\cos x$. Then we have $f(0) = -1$ and $f(1) > 0$, therefore f has a root in the interval $[0, 1]$, and the bisection method is applicable. (It is easy to check that f is strictly monotone increasing on $[0, 1]$, so it has a unique root inside the interval. Table 2.2 contains the result of the bisection method using tolerance value $\varepsilon = 10^{-5}$. Formula (2.5) yields that $k \geq \log_2 10^5 - 1 \approx 15.61$ steps are needed to obtain this accuracy. □

Table 2.2: bisection method, $f(x) = e^x - 2\cos x$, $[0, 1]$, $\varepsilon = 10^{-5}$

k	a_k	b_k	p_k	$f(p_k)$
0	0.00000000	1.00000000	0.50000000	-1.0644e-01
1	0.50000000	1.00000000	0.75000000	6.5362e-01
2	0.50000000	0.75000000	0.62500000	2.4632e-01
3	0.50000000	0.62500000	0.56250000	6.3206e-02
4	0.50000000	0.56250000	0.53125000	-2.3292e-02
5	0.53125000	0.56250000	0.54687500	1.9538e-02
6	0.53125000	0.54687500	0.53906250	-1.9818e-03
7	0.53906250	0.54687500	0.54296875	8.7517e-03
8	0.53906250	0.54296875	0.54101563	3.3784e-03
9	0.53906250	0.54101563	0.54003906	6.9670e-04
10	0.53906250	0.54003906	0.53955078	-6.4294e-04
11	0.53955078	0.54003906	0.53979492	2.6780e-05
12	0.53955078	0.53979492	0.53967285	-3.0810e-04
13	0.53967285	0.53979492	0.53973389	-1.4067e-04
14	0.53973389	0.53979492	0.53976440	-5.6946e-05
15	0.53976440	0.53979492	0.53977966	-1.5083e-05
16	0.53977966	0.53979492	0.53978729	5.8483e-06

Exercises

1. Show that the equation

   (a) $x^3 - 6x - 1 = 0$, $[a, b] = [-1, 1]$, (b) $x = e^{-2x}$, $[a, b] = [-1, 2]$,

   (c) $\tan x = x + 1$, $[a, b] = [-1, 1.5]$, (d) $e^{-\sin x} = x^2 - 1$, $[a, b] = [0, 2]$

   has a root in the interval $[a, b]$. Using the bisection method give an approximate solution within the tolerance $\varepsilon = 10^{-5}$.
2. Apply the bisection method for the function $f(x) = \frac{1}{x}$ on the interval $[-0.5, 3]$. What do you observe?

2.4. Method of False Position

The advantage of the bisection method is that it is easy to determine the number of steps needed to reach a given accuracy. But its weakness is that it does not take into account the shape of the functions when the next interval is selected in the sequence. This is the idea of the *method of false position* (also called *Regula Falsi*).

We assume the same conditions as in the bisection method. We suppose $f\colon [a, b] \to \mathbb{R}$ is a continuous function which has opposite sign at the end points of the interval, i.e.,

$f(a)f(b) < 0$. We define a sequence of nested intervals $[a_k, b_k]$ with a help of an inner point p_k, but it is no longer the midpoint of the intervals. First define $[a_0, b_0] = [a, b]$. At the kth step, let p_k be the intersection of the secant line of f corresponding to the points a_k and b_k (the line segment through the points $(a_k, f(a_k))$ and $(b_k, f(b_k))$) and the x-axis. Little calculation gives that

$$p_k = a_k - f(a_k)\frac{a_k - b_k}{f(a_k) - f(b_k)}. \tag{2.6}$$

The next interval $[a_{k+1}, b_{k+1}]$ will be either $[a_k, p_k]$ or $[p_k, b_k]$ where the function has a sign change. The method is defined in Algorithm 2.18.

Algorithm 2.18. method of false position

INPUT: f - is a function,
$[a, b]$ - is an interval, where $f(a)f(b) < 0$
TOL - is the tolerance,
$MAXIT$ - is the maximal iteration step,
OUTPUT: p - is the approximating root.

$i \leftarrow 1$ *(step counter)*
$q \leftarrow a$
while $i < MAXIT$ **do**
  $p \leftarrow a - f(a)(a - b)/(f(a) - f(b))$
  if $|p - q| < TOL$ **do**
    output(p)
    stop
  end do
  if $f(p)f(b) < 0$ **do**
    $a \leftarrow p$
  else if $f(a)f(p) < 0$ **do**
    $b \leftarrow p$
  else
    output(p)
    stop
  end do
  $i \leftarrow i + 1$
  $q \leftarrow p$
end do
output(Maximal iteration step is exceeded.)

When we implement the Algorithm 2.18 in a computer program, it is important to test whether $f(a)$ is equal to $f(b)$, since otherwise we divide by 0, and the program fails. Such technical details are not included in the algorithms we present in this lecture note, but those are important when we implement the algorithms.

We show the convergence of the method of false position under the condition when the function f is convex or concave.

Theorem 2.19. *Suppose the continuous function $f \in C[a,b]$ is convex or concave on $[a,b]$ and $f(a)f(b) < 0$. Then the method of false position converges to the unique root p of f.*

Proof. Suppose, e.g., that f is convex and $f(a) > 0$, $f(b) < 0$. The other cases can be argued similarly. Then the left subinterval contains the root p of f at each step, i.e., $a_{k+1} = a$ and $b_{k+1} = p_k$ for all k. Since the sequence p_k is monotone decreasing and a is a lower bound of the sequence, it converges to a limit $p \geq a$. We have $f(p_k) < 0$ for all k, therefore $f(p) \leq 0$. Since $f(a) > 0$, we get $p > a$. Taking the limit of Equation (2.6) as $k \to \infty$ we obtain

$$p = a - f(a)\frac{a-p}{f(a)-f(p)},$$

which implies that $f(p) = 0$. □

Example 2.20. Applying the method of false position to the problem of Example 2.17, we get the numerical values presented in Table 2.3. As in Example 2.17, we use the interval $[0,1]$ and $TOL = 10^{-5}$. We can observe that for this equation and using the given initial interval the method of false position converges much faster than the bisection method. □

Table 2.3: Method of false position, $f(x) = e^x - 2\cos x$, $[0,1]$, $TOL = 10^{-5}$

k	a_k	b_k	p_k	$f(p_k)$
0	0.00000000	1.00000000	0.37912145	-3.9698e-01
1	0.37912145	1.00000000	0.50026042	-1.0576e-01
2	0.50026042	1.00000000	0.53057677	-2.5118e-02
3	0.53057677	1.00000000	0.53766789	-5.8011e-03
4	0.53766789	1.00000000	0.53929982	-1.3311e-03
5	0.53929982	1.00000000	0.53967399	-3.0499e-04
6	0.53967399	1.00000000	0.53975970	-6.9856e-05
7	0.53975970	1.00000000	0.53977933	-1.5999e-05
8	0.53977933	1.00000000	0.53978383	-3.6640e-06

Example 2.21. We apply again the method of false position for the equation of Example 2.17 but now on the initial interval $[0,4]$. The numerical results are displayed in Table 2.4. (Only the first and last several steps are presented.) Now, the speed of the convergence is far slower than that of observed in the previous example. (And it becomes even slower if we further increase the right end point of the interval.) On the other hand, (2.5) yields that the bisection method with the initial interval $[0,4]$ has this accuracy in 18 steps, which is only two steps longer than in Example 2.17. □

Exercises

1. Apply the method of false position for the equations presented in Exercise 1 of Section 2.3.

Table 2.4: Method of false position, $f(x) = e^x - 2\cos x$, $[0, 4]$, $TOL = 10^{-5}$

k	a_k	b_k	p_k	$f(p_k)$
0	0.00000000	4.00000000	0.07029205	-9.2224e-01
1	0.07029205	4.00000000	0.13406612	-8.3858e-01
2	0.13406612	4.00000000	0.19119837	-7.5285e-01
3	0.19119837	4.00000000	0.24180834	-6.6826e-01
4	0.24180834	4.00000000	0.28620106	-5.8729e-01
⋮	⋮	⋮	⋮	⋮
47	0.53966897	4.00000000	0.53968870	-2.6464e-04
48	0.53968870	4.00000000	0.53970508	-2.1970e-04
49	0.53970508	4.00000000	0.53971868	-1.8240e-04
50	0.53971868	4.00000000	0.53972996	-1.5143e-04
51	0.53972996	4.00000000	0.53973934	-1.2572e-04

2. Let
$$f(x) = \begin{cases} \delta, & x \le 0.5 \\ 4(1+\delta)(x - x^2) - 1, & x \ge 0.5 \end{cases}$$
Apply the bisection method and the method of false position on the interval $[0, 1]$ to approximate the root of f if
(a) $\delta = 2$, (b) $\delta = 0.5$, (c) $\delta = 0.09$.

3. Work out the details of the proof of Theorem 2.19 for all the other cases.

2.5. Newton's Method

One general approach in numerical analysis is that we replace the problem by a "simpler" one which is "close" to the original problem, and we hope that the solution of the simpler problem approximate that of the original problem. Here our goal is to find the solution of the scalar equation $f(x) = 0$. We replace the function f by its first-order Taylor polynomial approximation, and we solve the resulting linear equation. Geometrically this means that the intersection of the tangent line with the x-axis gives an approximation of the root of the original nonlinear equation. The equation of the tangent line to the graph of f at p_0 is $y = f(p_0) + f'(p_0)(x - p_0)$, so its intersection with the x-axis is the solution of the linear equation $f(p_0) + f'(p_0)(x - p_0) = 0$, hence it is $x = p_0 - f(p_0)/f'(p_0)$ (assuming, of course, that $f'(p_0) \neq 0$). This number is denoted by p_1, and we repeat the procedure from this point. Then we get the recursive sequence defined by

$$p_{k+1} = p_k - \frac{f(p_k)}{f'(p_k)}. \tag{2.7}$$

The iterative method (2.7) is called *Newton–Raphson method* or shortly *Newton's method* or *Newton iteration*.

Example 2.22. We applied the Newton's method for the problem of Example 2.17, and we got the numerical results presented in Table 2.5. We observe that the sequence converges very fast to the root of the function. □

Table 2.5: Newton's method, $f(x) = e^x - 2\cos x$, $p_0 = 0$, $TOL = 10^{-5}$

k	p_k	$f(p_k)$
0	0.1000000000	-8.8484e-01
1	0.7781206411	7.5291e-01
2	0.5678850726	7.8450e-02
3	0.5402639121	1.3139e-03
4	0.5397853041	3.9302e-07
5	0.5397851608	3.5207e-14

The Newton's method is a one-step iteration with the function

$$g(x) := x - \frac{f(x)}{f'(x)}. \tag{2.8}$$

Computing the derivative of g we get

$$g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}. \tag{2.9}$$

Let p be a root f satisfying $f'(p) \neq 0$. Then $g'(p) = 0$, so Theorem 2.15 yields immediately the following result.

Theorem 2.23. *Let $f \in C^2[a,b]$, and let $p \in (a,b)$ be such that $f(p) = 0$ and $f'(p) \neq 0$. Then the Newton's method converges locally to p.*

Example 2.24. Consider the function $f(x) = 0.5 \arctan x$. It's only root is $p = 0$. We have that $f'(0) = 0.5$, so the Newton's method converges locally to $p = 0$, i.e., if p_0 is close enough to 0, then the Newton-iteration converges to 0. In Table 2.6 we present the first several terms of this sequence starting from $p_0 = 1.4$. (In the 15th step the program terminated with an error, since $f'(p_{14}) = 0$ on the computer.) We can see that the sequence p_k does not converge to 0 in this case. □

Exercises

1. Apply the Newton's method for the equations presented in Exercise 1 of Section 2.3.
2. Let $f(x) = 0.5 \arctan x$. Then f has the unique root $x = 0$. Let p_k be the Newton's iteration sequence. Show that there exists a number p^* such that
   (a) if $|p_0| < p^*$, then $p_k \to 0$,
   (b) if $|p_0| = p^*$, then the sequence p_k repeats p_0 and $-p_0$, and hence it is not convergent,
   (c) if $|p_0| > p^*$, then p_k alternates (i.e., $p_k p_{k+1} < 0$ for all k), and $|p_k| \to \infty$.
3. Give an iteration to approximate $\sqrt[k]{a}$.

Table 2.6: Newton's method, $f(x) = 0.5 \arctan x$, $p_0 = 1.4$

k	p_k	$f(p_k)$
0	1.4000000e+00	0.4752734
1	-1.4136186e+00	-0.4775591
2	1.4501293e+00	0.4835443
3	-1.5506260e+00	-0.4990071
4	1.8470541e+00	0.5372889
5	-2.8935624e+00	-0.6190257
6	8.7103258e+00	0.7282453
7	-1.0324977e+02	-0.7805557
8	1.6540564e+04	0.7853679
9	-4.2972148e+08	-0.7853982
10	2.9006412e+17	0.7853982
11	-1.3216239e+35	-0.7853982
12	2.7436939e+70	0.7853982
13	-1.1824729e+141	-0.7853982
14	2.1963537e+282	0.7853982

2.6. Secant Method

The Newton's method requires the computation (and hence the existence) of the derivative of f. But in practice, f' is not always known, (it is possible that f is not defined by a formula, it may be an output of an other numerical procedure which computes the value of f with a good precision). Or the computation of f' requires too much calculation, so we prefer not to evaluate it. The *secant method* does not require the computation of the derivative f'.

Let p_0 and p_1 be two different initial values of the sequence. Consider the secant line of f corresponding to the points p_0 and p_1, i.e., the line which connects the points $(p_0, f(p_0))$ and $(p_1, f(p_1))$. Its equation is

$$y = f(p_1) + \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_1).$$

The secant line intersects the x-axis at $x = p_1 - \frac{p_1 - p_0}{f(p_1) - f(p_0)} f(p_1)$. p_2 will denote this number. Then we consider the secant line corresponding to p_1 and p_2, and its intersection with the x-axis is denoted by p_3. Repeating this procedure we define the sequence p_k by the recursion

$$p_{k+1} = p_k - \frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})} f(p_k). \tag{2.10}$$

This is a two-step iteration, which defines the *secant method*.

Example 2.25. We used the secant method for the problem of Example 2.17. The numerical results can be seen in Table 2.7. Comparing it with the Table 2.5, we observe that the secant method converges to its limit slower than the Newton's method. □

For the proof of the secant method we need the following theorem.

Theorem 2.26. *Let $f \in C^2[a, b]$, and let $p \in (a, b)$ be such that $f(p) = 0$ and $f'(p) \neq 0$. Let p_k be the sequence defined by the secant method. Then for every k there exist $\xi_k \in$*

Table 2.7: secant method, $f(x) = e^x - 2\cos x$, $p_0 = 0$, $p_1 = 1$, $TOL = 10^{-5}$

k	p_k	$f(p_k)$
0	0.0000000000	-1.0000e+00
1	1.0000000000	1.6377e+00
2	0.3791214458	-3.9698e-01
3	0.5002604213	-1.0576e-01
4	0.5442561500	1.2301e-02
5	0.5396724494	-3.0921e-04
6	0.5397848464	-8.6246e-07
7	0.5397851608	6.0793e-11

$\langle p_k, p_{k-1}, p\rangle$ *and* $\eta_k \in \langle p_k, p_{k-1}\rangle$ *such that*

$$p_{k+1} - p = \frac{1}{2}\frac{f''(\xi_k)}{f'(\eta_k)}(p_k - p)(p_{k-1} - p). \tag{2.11}$$

Proof. Algebraic manipulations give

$$\begin{aligned} p_{k+1} - p &= p_k - p - \frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})} f(p_k) \\ &= \frac{(p_{k-1} - p)f(p_k) - (p_k - p)f(p_{k-1})}{f(p_k) - f(p_{k-1})} \\ &= \frac{(p_k - p)(p_{k-1} - p)}{f(p_k) - f(p_{k-1})} \left(\frac{f(p_k)}{p_k - p} - \frac{f(p_{k-1})}{p_{k-1} - p} \right) \\ &= (p_k - p)(p_{k-1} - p)\frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})} \frac{\frac{f(p_k)-f(p)}{p_k-p} - \frac{f(p_{k-1})-f(p)}{p_{k-1}-p}}{p_k - p_{k-1}}. \end{aligned}$$

Then the Lagrange Mean Value Theorem implies the existence of $\eta_k \in \langle p_k, p_{k-1}\rangle$ such that

$$\frac{f(p_k) - f(p_{k-1})}{p_k - p_{k-1}} = f'(\eta_k).$$

Now we have to show that there exists a $\xi_k \in \langle p_k, p_{k-1}, p\rangle$ such that

$$\frac{\frac{f(p_k)-f(p)}{p_k-p} - \frac{f(p_{k-1})-f(p)}{p_{k-1}-p}}{p_k - p_{k-1}} = \frac{f''(\xi_k)}{2}. \tag{2.12}$$

Its direct proof is left to Exercise 2. Here we prove relation (2.12) using a result and a notion will be discussed in Chapter 6. The left hand side of (2.12) is a second divided difference $f[p_{k-1}, p, p_k]$ of f corresponding to the points p_{k-1}, p and p_k (see Section 6.2). Corollary 6.17 yields that there exists a $\xi_k \in \langle p_k, p_{k-1}, p\rangle$ such that $f[p_{k-1}, p, p_k] = f''(\xi_k)/2$.□

Theorem 2.27. *Let $f \in C^2[a, b]$, and let $p \in (a, b)$ be such that $f(p) = 0$ and $f'(p) \neq 0$. Then the secant method converges locally to p.*

Proof. Let δ^* be such that $f'(x) \neq 0$ for $x \in [p-\delta^*, p+\delta^*]$. Such δ^* exists, since $f'(p) \neq 0$ and f' is continuous. Let

$$M := \frac{\max\{|f''(x)|\colon\ x \in [p-\delta^*, p+\delta^*]\}}{2\min\{|f'(x)|\colon\ x \in [p-\delta^*, p+\delta^*]\}}.$$

Select δ such that $\delta < \min\{\delta^*, \frac{1}{M}\}$, and let $\varepsilon := M\delta$. Then, by our conditions, $0 < \varepsilon < 1$. Let $p_0, p_1 \in (p-\delta, p+\delta)$ arbitrary but different numbers. Relation (2.11) and the definition of M yield that $|p_{k+1} - p| \leq M|p_k - p||p_{k-1} - p|$, and hence

$$M|p_{k+1} - p| \leq M|p_k - p|M|p_{k-1} - p| \tag{2.13}$$

for all k. With $k = 1$ we get $M|p_2 - p| \leq M|p_1 - p|M|p_0 - p| \leq (M\delta)^2 = \varepsilon^2 < \varepsilon$. Therefore $|p_2 - p| \leq \varepsilon/M = \delta$, and hence $p_2 \in (p-\delta, p+\delta)$. Similarly we can show that $p_k \in (p-\delta, p+\delta)$ for all k.

The definition of ε implies $M|p_0 - p| < \varepsilon$ and $M|p_1 - p| < \varepsilon$. Now we select a sequence q_k which satisfies $M|p_k - p| \leq \varepsilon^{q_k}$ for all k. We can define $q_0 = 1$ and $q_1 = 1$. Suppose the first k terms of the sequence q_k is already defined. Inequality (2.13) yields that relation $M|p_{k+1} - p| \leq \varepsilon^{q_k}\varepsilon^{q_{k-1}}$ must be satisfied. Hence $M|p_{k+1} - p| \leq \varepsilon^{q_{k+1}}$ holds, if q_{k+1} is defined by

$$q_{k+1} = q_k + q_{k-1}, \quad k \geq 1, \quad q_0 = 1, \quad q_1 = 1. \tag{2.14}$$

The sequence defined by (2.14) is the so-called *Fibonacci sequence*. We can show (see Exercise 3) that the general formula of q_k is

$$q_k = \frac{1}{\sqrt{5}}(r_0^{k+1} - r_1^{k+1}), \qquad k \geq 0, \tag{2.15}$$

where

$$r_0 = \frac{1+\sqrt{5}}{2} \approx 1.618, \quad \text{and} \quad r_1 = \frac{1-\sqrt{5}}{2} \approx -0.618.$$

But then $q_k \to \infty$ as $k \to \infty$. Now we get $p_k \to p$, since

$$|p_k - p| \leq \frac{1}{M}\varepsilon^{q_k} \to 0, \qquad \text{as } k \to \infty.$$

□

Exercises

1. Apply the secant method for the equations presented in Exercise 1 of Section 2.3.
2. Prove relation (2.12). (Hint: show that the expression
$$f[a,b,c] := \frac{\frac{f(c)-f(b)}{c-b} - \frac{f(b)-f(a)}{b-a}}{c-a}$$
is independent from the order of the numbers a, b, c. Therefore, we can assume that $a < b < c$. Take the first-order Taylor approximation of f around b together with the second-order error term. Then express the numerator of the right hand side. Finally, use Theorem 2.2 to show that $f[a,b,c] = f''(\xi)/2$ for some $\xi \in (a,c)$.)
3. Prove formula (2.15).

2.7. Order of Convergence

In the previous sections we observed that some sequence converges to a limit faster than other sequences. In this section we define the notion of order of convergence which can characterize the speed of the convergence.

Let p_k be a convergent sequence with limit p. We say that the *order of convergence* of the sequence p_k is α if $\alpha \geq 1$ and there exists a constant $c \geq 0$ such that

$$|p_{k+1} - p| \leq c|p_k - p|^\alpha \qquad \text{for all } k \geq 0, \tag{2.16}$$

and if $\alpha = 1$, then we also assume that $c < 1$.

If we want to be more precise, then in case when (2.16) holds, we could say that the order of convergence is *at least* α, since it is possible that (2.16) can be satisfied with an exponent bigger than α, too. For simplicity, we will omit "at least" in the sequel, but the notion should always be understood in this sense. If we want to emphasize that p_k satisfies (2.16) with some α, but it does not satisfy it with any exponent bigger than α, then we say that the order of convergence is *exactly* α.

If the order of convergence of a sequence is $\alpha = 1$, then we say that the convergence is *linear*, and if $\alpha = 2$, then we say that the convergence is *quadratic*.

Suppose p_k converges to p linearly. Then it is easy to see that

$$|p_k - p| \leq c^k |p_0 - p| \tag{2.17}$$

holds. For some cases, it is not easy to show a linear convergence of a numerical method using the definition (2.16). So we extend the previous definition in such a way that if a sequence satisfies relation (2.17) with a constant $0 \leq c < 1$, then we also say that the convergence is linear.

Suppose $p_k \to p$ with order α. If the finite limit

$$\lambda = \lim_{k\to\infty} \frac{p_{k+1} - p}{(p_k - p)^\alpha} \tag{2.18}$$

exists, then we call λ as the *asymptotic error constant*. It can be proved easily that if the limit (2.18) exists and it is finite, then p_k is convergent and its order of convergence is α. If p_k converges linearly and its asymptotic error constant is 0, then we speak about *superlinear* convergence.

Theorem 2.28. *Suppose p_k converges to p of order α with the asymptotic error constant $\lambda \neq 0$. Then*

(i) $\lim\limits_{k\to\infty} \dfrac{p_{k+1} - p}{(p_k - p)^\beta} = 0$ *for all $\beta < \alpha$, and*

(ii) $\lim\limits_{k\to\infty} \dfrac{|p_{k+1} - p|}{|p_k - p|^\beta} = \infty$ *for all $\beta > \alpha$.*

Proof. The statements follow from relation

$$\frac{|p_{k+1} - p|}{|p_k - p|^\beta} = \frac{|p_{k+1} - p|}{|p_k - p|^\alpha} \frac{1}{|p_k - p|^{\beta-\alpha}}.$$

□

It follows from the above theorem, that if a sequence p_k converges to p of order α, and the asymptotic error constant $\lambda \neq 0$, then the order of convergence is exaclty α.

Example 2.29. Consider again the Newton iteration of Example 2.22. In Table 2.8 we have listed in the last three columns the numerical values of the formula $|p_{k+1} - p|/|p_k - p|^\alpha$ for $\alpha = 1, 2$ and 3 using the value $p = 0.5397851608092811$. We can observe that for $\alpha = 1$ the sequence goes to 0. For $\alpha = 2$ the sequence remains bounded but it does not converge to 0, for $\alpha = 3$ it converges to ∞. (Certainly from the first 5 terms of a sequence we should not make conclusions about a limit of a sequence, but generation of more terms will confirm the above observations.) Therefore, the numerical evidence suggests that the order of convergence of this sequence is 2. □

Table 2.8: Order of convergence of the Newton iteration, $f(x) = e^x - 2\cos x$

			$\lvert p_k - p\rvert/\lvert p_{k-1} - p\rvert^\alpha$		
k	p_k	$f(p_k)$	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$
0	0.0000000000	-1.0000e+00			
1	1.0000000000	1.6377e+00	8.5259e-01	1.5795e+00	2.9262e+00
2	0.6279041258	2.5516e-01	1.9147e-01	4.1605e-01	9.0404e-01
3	0.5442066314	1.2164e-02	5.0176e-02	5.6941e-01	6.4619e+00
4	0.5397973257	3.3375e-05	2.7513e-03	6.2226e-01	1.4074e+02
5	0.5397851609	2.5388e-10	7.6071e-06	6.2533e-01	5.1404e+04

Theorem 2.30. *Suppose a sequence p_k satisfies inequality (2.16) with some $c \geq 0$ and $\alpha > 1$. Then p_k converges locally to p, and for every k*

$$|p_k - p| \leq c^{\frac{\alpha^k - 1}{\alpha - 1}} |p_0 - p|^{\alpha^k}. \tag{2.19}$$

Proof. Relation (2.19) can be easily proved with mathematical induction. Then it implies

$$|p_k - p| \leq c^{\frac{1}{1-\alpha}} \left(c^{\frac{1}{\alpha-1}} |p_0 - p| \right)^{\alpha^k}.$$

Hence if p_0 is such that $c^{\frac{1}{\alpha-1}} |p_0 - p| < 1$, then $p_k \to p$, i.e., p_k converges locally to p. □

Example 2.31. Suppose $p_k \to p$ and $q_k \to q$ linearly and quadratically, respectively, which satisfy (2.17) and (2.16) with $c = 1/2$, respectively. Moreover, we suppose $|p_0 - p| < 1$ and $|q_0 - q| < 1$. Then relations (2.17) and (2.19) yield that $|p_k - p| \leq (1/2)^k$ and $|q_k - q| \leq (1/2)^{2^k - 1}$. In Table 2.9 we listed these error bounds for $k = 1, 2, \ldots, 5$. We can see that the error decreases much faster in the quadratic case. □

Theorem 2.32. *Let $g \in C^m[a, b]$, $p \in (a, b)$ and $p = g(p)$. Consider the fixed-point iteration $p_{k+1} = g(p_k)$.*

(i) If $|g'(p)| < 1$, then the fixed-point iteration converges locally and linearly to p.

(ii) If $g'(p) = g''(p) = \ldots = g^{(m-1)}(p) = 0$, then the fixed-point iteration converges locally to p of order m with the asymptotic error constant $g^{(m)}(p)/m!$.

Table 2.9:

k	$(1/2)^k$	$(1/2)^{2^k-1}$
1	$5.0000 \cdot 10^{-1}$	$5.0000 \cdot 10^{-1}$
2	$2.5000 \cdot 10^{-1}$	$1.2500 \cdot 10^{-1}$
3	$1.2500 \cdot 10^{-1}$	$7.8125 \cdot 10^{-3}$
4	$6.2500 \cdot 10^{-2}$	$3.0518 \cdot 10^{-5}$
5	$3.1250 \cdot 10^{-2}$	$4.6566 \cdot 10^{-10}$
6	$1.5625 \cdot 10^{-2}$	$1.0842 \cdot 10^{-19}$

Proof. Statement (i) follows from the proof of Theorem 2.15.

For the proof of statement (ii), we consider the Taylor approximation of g around p of degree $(m-1)$:

$$g(p_k) = g(p) + g'(p)(p_k - p) + \cdots + \frac{g^{(m-1)}(p)}{(m-1)!}(p_k - p)^{m-1} + \frac{g^{(m)}(\xi_k)}{m!}(p_k - p)^m,$$

where $\xi_k \in \langle p_k, p\rangle$. Using that the first $m-1$ derivatives are equal to 0 at p, $g(p) = p$ and $g(p_k) = p_{k+1}$, we get

$$|p_{k+1} - p| = \frac{|g^{(m)}(\xi_k)|}{m!}|p_k - p|^m \le c|p_k - p|^m. \tag{2.20}$$

In the last estimate we used that $g \in C^m[a,b]$, i.e., $g^{(m)}$ is continuous, and therefore, it is bounded in a neighborhood of p. The limit (2.18) follows from these, since $\xi_k \to p$ as $k \to \infty$ by relation $|\xi_k - p| \le |p_k - p|$. Therefore we obtain

$$\lim_{k\to\infty} \frac{p_{k+1} - p}{(p_k - p)^m} = \lim_{k\to\infty} \frac{g^{(m)}(\xi_k)}{m!} = \frac{g^{(m)}(p)}{m!}.$$

□

It follows from the above theorem that the order of convergence of a fixed-point iteration is always a positive integer assuming that g is smooth enough. Theorem 2.36 below shows that it is not true, in general, in the case of multistep iterations.

We will need the notion of a multiple root. We say that $p \in (a,b)$ is a root of *multiplicity* m of $f \in C[a,b]$ if there exists a function $q \in C[a,b]$ such that $q(p) \neq 0$ and

$$f(x) = (x-p)^m q(x), \qquad x \in (a,b). \tag{2.21}$$

We can prove the next result easily.

Theorem 2.33. *Let $f \in C^m[a,b]$, $p \in (a,b)$.*

(i) Let p be a root of multiplicity m of f, and the function q in (2.21) is m times differentiable. Then

$$f(p) = f'(p) = f''(p) = \ldots = f^{(m-1)}(p) = 0, \quad \text{and } f^{(m)}(p) \neq 0. \tag{2.22}$$

(ii) If (2.22) holds, then p is a root of multiplicity m of f.

(iii) Suppose f is infinitely many times differentiable, f is expandable in a Taylor-series around p, and f satisfies relations (2.22). Then p is a root of order m of f, and the function q in (2.21) is also infinitely many times differentiable, and q is expandable in a Taylor-series around p.

The next theorem shows that if p is a simple root of f, then the Newton iteration is locally and quadratically convergent, and if p is a multiple root of f, then the order of convergence is linear.

Theorem 2.34. *Let $f \in C^2[a,b]$.*

(i) If $f(p) = 0$ and $f'(p) \neq 0$, then the Newton iteration converges locally to p, and the order of convergence is quadratic.

(ii) If $f(x) = (x-p)^m q(x)$, where $q \in C^2[a,b]$, $q(p) \neq 0$, $m > 1$, then the Newton iteration converges locally to p, and the order of convergence is linear.

Proof. Statement (i) follows form part (ii) of Theorem 2.32, since the Newton iteration is a fixed-point iteration with the function g defined in (2.8), and $g'(p) = 0$ by relation (2.9).

Since the function

$$g(x) := \begin{cases} x - \frac{f(x)}{f'(x)}, & x \neq p, \\ p & x = p \end{cases}$$

satisfies

$$g(x) = x - \frac{(x-p)q(x)}{mq(x) + (x-p)q'(x)},$$

it is continuously differentiable at p, and $g'(p) = 1 - \frac{1}{m}$. Therefore, part (ii) of Theorem 2.32 yields that the order of convergence is linear. □

Example 2.35. Find the root of $f(x) = x^3 + x^2 - 8x - 12$ by the Newton–Raphson method from the initial value $p_0 = 0$ and using tolerance 10^{-5}. It is easy to see that $x = -2$ is a double root, and $x = 3$ is a simple root of the polynomial. In Table 2.10 we can see the numerical values of the iteration corresponding to $p_0 = 0$, and in Table 2.11 corresponding to $p_0 = 2$. In the first case the sequence converges to -2, and in the second case it converges to 3. We can observe that in the first case the convergence is linear, but in the second case it is quadratic. □

Theorem 2.36. *If p is a simple root of f, then the secant method converges locally to p of order $\alpha = (1+\sqrt{5})/2 \approx 1.618$.*

Proof. We use the notations and results introduced in the proof of Theorem 2.27. By inequality (2.13) we have

$$|p_{k+1} - p| \leq M|p_k - p||p_{k-1} - p|.$$

Table 2.10: Newton iteration, $f(x) = x^3 + x^2 - 8x - 12$

k	p_k	$f(p_k)$	$\|p_k - p\|/\|p_{k-1} - p\|^\alpha$ $\alpha = 1$	$\alpha = 2$
0	0.0000000000	-1.2000e+01		
1	-1.5000000000	-1.1250e+00	2.5000e-01	1.2500e-01
2	-1.7647058824	-2.6379e-01	4.7059e-01	9.4118e-01
3	-1.8853313477	-6.4237e-02	4.8734e-01	2.0712e+00
4	-1.9433465411	-1.5866e-02	4.9406e-01	4.3086e+00
5	-1.9718365260	-3.9436e-03	4.9712e-01	8.7747e+00
6	-1.9859582600	-9.8308e-04	4.9858e-01	1.7703e+01
7	-1.9929890302	-2.4542e-04	4.9929e-01	3.5558e+01
8	-1.9964969780	-6.1313e-05	4.9965e-01	7.1267e+01
9	-1.9982491032	-1.5323e-05	4.9982e-01	1.4268e+02
10	-1.9991247050	-3.8300e-06	4.9991e-01	2.8552e+02
11	-1.9995623908	-9.5743e-07	4.9996e-01	5.7119e+02
12	-1.9997812050	-2.3935e-07	4.9998e-01	1.1425e+03
13	-1.9998906049	-5.9835e-08	4.9999e-01	2.2852e+03
14	-1.9999453030	-1.4959e-08	4.9999e-01	4.5705e+03
15	-1.9999726517	-3.7396e-09	5.0000e-01	9.1412e+03
16	-1.9999863259	-9.3491e-10	5.0000e-01	1.8283e+04
17	-1.9999931629	-2.3373e-10	5.0000e-01	3.6565e+04

Table 2.11: Newton iteration, $f(x) = x^3 + x^2 - 8x - 12$

k	p_k	$f(p_k)$	$\|p_k - p\|/\|p_{k-1} - p\|^\alpha$ $\alpha = 1$	$\alpha = 2$
0	2.0000000000	-1.6000e+01		
1	4.0000000000	3.6000e+01	1.0000e+00	1.0000e+00
2	3.2500000000	6.8906e+00	2.5000e-01	2.5000e-01
3	3.0217391304	5.4821e-01	8.6957e-02	3.4783e-01
4	3.0001866020	4.6654e-03	8.5837e-03	3.9485e-01
5	3.0000000139	3.4816e-07	7.4632e-05	3.9996e-01
6	3.0000000000	1.9400e-15	5.5721e-09	4.0011e-01

Then, applying estimate $|p_k - p| \le \frac{1}{M}\varepsilon^{q_k}$, we get

$$\begin{aligned}
|p_{k+1} - p| &\le |p_k - p|^{r_0} M |p_k - p|^{1-r_0} |p_{k-1} - p| \\
&\le |p_k - p|^{r_0} M \left(\frac{1}{M}\varepsilon^{q_k}\right)^{1-r_0} \frac{1}{M}\varepsilon^{q_{k-1}} \\
&= |p_k - p|^{r_0} M^{r_0-1} \varepsilon^{q_k + q_{k-1} - r_0 q_k} \\
&= |p_k - p|^{r_0} M^{r_0-1} \varepsilon^{q_{k+1} - r_0 q_k} \\
&= |p_k - p|^{r_0} M^{r_0-1} \varepsilon^{r_1^{k+1}}.
\end{aligned}$$

Note that the last step follows from (2.15) (with some calculations). Since $r_1^{k+1} \to 0$ as $k \to \infty$, we get that there exists a constant c such that $|p_{k+1} - p| \le c|p_k - p|^{r_0}$, and hence the order of convergence is $r_0 = \frac{1+\sqrt{5}}{2}$. □

We have seen that the Newton iteration is only linearly convergent in the case of a multiple root. It is possible to prove that the same holds for the secant method. Next we discuss how to accelerate the speed of the convergence in this case.

Let $f \in C^3[a,b]$, suppose $p \in (a,b)$ is a multiple root of f. More precisely, we assume that $f(x) = (x-p)^m q(x)$ with $m > 1$ and $q \in C^3[a,b]$. We define the function

$$\mu(x) = \begin{cases} \frac{f(x)}{f'(x)}, & \text{if } x \neq p, \\ 0, & \text{if } x = p. \end{cases}$$

We can see that

$$\mu(x) = \frac{(x-p)q(x)}{mq(x) + (x-p)q'(x)},$$

and hence $\mu \in C^2[a,b]$. Moreover, $\mu'(p) = \frac{1}{m}$, and so p is only a simple root of μ. Therefore if we use the Newton iteration for the function μ instead of f, we get a quadratic convergence. Then we get the sequence

$$p_{k+1} = p_k - \frac{\mu(p_k)}{\mu'(p_k)} = p_k - \frac{f(p_k)f'(p_k)}{(f'(p_k))^2 - f(p_k)f''(p_k)}. \tag{2.23}$$

Exercises

1. Show that the bisection method is linearly convergent.
2. Prove inequality (2.19).
3. Let $a > 0$. Show that
$$p_{k+1} = \frac{p_k(p_k^2 + 3a)}{3p_k^2 + a}$$
is a locally convergent sequence of order 3 to approximate $\sqrt{a}$.
4. Find the order of convergence of the sequence $p_k = \frac{1}{k}$. What is the order of convergence of $p_k = \frac{1}{k^n}$?
5. Show that $p_k = 10^{-2^k}$ goes to 0 quadratically.
6. Show that $x = 0$ is a double root of the function $\sin^2 x$.
7. Prove Theorem 2.33.
8. Consider the following iterations:
   (a) (Halley iteration) $p_{k+1} = p_k - \dfrac{1}{a_k}$, where $a_k = \dfrac{f'(p_k)}{f(p_k)} - \dfrac{1}{2}\dfrac{f''(p_k)}{f'(p_k)}$,
   (b) (Olver iteration) $p_{k+1} = p_k - \dfrac{f(p_k)}{f'(p_k)} - \dfrac{1}{2}\dfrac{f''(p_k)}{f'(p_k)}\left(\dfrac{f(p_k)}{f'(p_k)}\right)^2$,
   Determine the order of convergence of the methods. Apply these methods to the problems in Exercise 1 of Section 2.3.
9. Find the root of $f(x) = (x^2-5)^3$ using Newton iteration, secant method, iteration (2.23), and iteration
$$p_{k+1} = p_k - m\frac{f(p_k)}{f'(p_k)},$$
where m the multiplicity of the root. Compare the order of convergence of the sequences. What is the order of convergence of the last iteration?

10. Suppose we already determined a root x_1 of the function f. Then if we apply a numerical method to find a root of the function $g(x) = f(x)/(x - x_1)$, then we get another root of f (or x_1 again, if x_1 is a multiple root). This is the so-called *deflation method*. With this method determine all roots of the polynomials together with their multiplicities (using any approximation technique):

(a) $f(x) = x^3 - 3x^2 + 4$, (b) $f(x) = x^4 - 5x^3 + 9x^2 - 7x + 2$

2.8. Stopping Criteria of Iterations

In this chapter the numerical methods we discussed generate an infinite sequence p_k to find a root of the function f, and the limit p of the sequence is the exact value of the root. We approximate the limit of the sequence p by a term of the sequence p_k, where k is "large enough". So the question is how we determine the number of steps k for which p_k gives us a good approximation of p. Here we introduce three popular strategies. We predefine three tolerances $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ and $\varepsilon_3 > 0$. We consider the kth term p_k as an appropriate approximation of p if

$$\text{(i)}\ \ |p_k - p_{k-1}| < \varepsilon_1, \qquad \text{(ii)}\ \ \frac{|p_k - p_{k-1}|}{|p_k|} < \varepsilon_2, \quad \text{or} \quad \text{(iii)}\ \ |f(p_k)| < \varepsilon_3. \tag{2.24}$$

Condition (i) is a numerical analogue of the absolute error $|p_k - p|$ of the approximation. It assumes that if a new term of the sequence is closer to the previous one than the tolerance, then it is because both terms are already close to the limit. So we terminate the generation of the sequence.

Condition (ii) is the numerical analogue of the relative error $|p_k - p|/|p|$ of the approximation. As in the previous case, we examine the distance between consecutive terms but we take into account the order of magnitude of the terms.

Condition (iii) tests whether the function value at p_k is close to 0. If it is satisfied, we assume that it is because the term is close to a root of f, and we terminate the sequence.

In a computer code it is always recommended to count the number of iteration and stop computing the sequence if it is too large, i.e., larger than a predefined maximal iteration number. This way we avoid a possible infinite loop of the program, and also, we do not allow a convergence which is too slow.

The first two conditions can be applied for any iteration, but the third one is formulated for the problem of finding a root of a single variable function f. We remark that for other type of problems it is likely that we can formulate a similar condition which tests how well the approximate solution satisfies the investigated mathematical problem (see, e.g., Section 4.4 below).

We remark that the above reasoning is heuristic. We can find examples when a stopping condition (i), (ii) or (iii) in (2.24) holds, but the kth term of the sequence is not close to a root. Therefore, in practice, we usually use combination of stopping criteria.

Exercises

1. Suppose an iteration method generates the sequence $p_k = \sum_{i=1}^{k} \frac{1}{i}$, and suppose we use only the stopping criterion (i) defined in (2.24). What do we observe? Does the sequence converge? What do we get if we use stopping criterion (ii)?
2. Let $f(x) = x^8$, and suppose an iteration generates $p_k = 1/k$ to approximate the root of f. Suppose we use stopping condition (i) in (2.24) with $\varepsilon_1 = 10^{-8}$. What do we get as an approximate root? What do we get if we use only stopping condition (ii), and what if we use only condition (iii) with tolerances $\varepsilon_2 = 10^{-8}$ or $\varepsilon_3 = 10^{-8}$, respectively?

2.9. Review of Multivariable Calculus

In this section we review those notions, notations and results from multivariable calculus which we use in the rest of this chapter.

Theorem 2.37. *Let $E \subset \mathbb{R}^n$ be a closed and bounded set, $f\colon E \to \mathbb{R}$ be continuous. Then f has a maximum and a minimum on E, i.e., there exist $\mathbf{c}, \mathbf{d} \in E$ such that*

$$f(\mathbf{c}) = \max_{\mathbf{x}\in E} f(\mathbf{x}) \qquad \text{and} \qquad f(\mathbf{d}) = \min_{\mathbf{x}\in E} f(\mathbf{x}).$$

Let $E \subset \mathbb{R}^n$, and consider the function $f : E \to \mathbb{R}$ of n variables. The partial derivatives of the function $f = f(\mathbf{x}) = f(x_1, \dots, x_n)$ with respect to the variable x_i is denoted by $\frac{\partial f}{\partial x_i}$. If all the partial derivatives of f up to order m exist and are continuous, then we say that f is m times continuously partially differentiable, and we will denote it by $f \in C^m$. If $f \in C^1$, then f' denotes the *gradient vector* or shortly, the *gradient* of f, i.e.,

$$f'(\mathbf{x}) := \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T.$$

If $f \in C^2$, then $f''(\mathbf{x})$ is the so-called the *Hessian matrix* or shortly the *Hessian* defined by

$$f''(\mathbf{x}) := \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_2^2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_n \partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{pmatrix}$$

We will need the multivariable Taylor's formula later.

Theorem 2.38 (Taylor's formula). *Let $E \subset \mathbb{R}^n$ be an open set, $f\colon E \to \mathbb{R}$, $f \in C^{m+1}$, and let $\mathbf{a} \in E$. Then for every $\mathbf{x} \in E$ there exists a $\xi = \xi(\mathbf{x}) \in E$ such that $\xi = \mathbf{x}+t(\mathbf{a}-\mathbf{x})$ for some $t \in (0,1)$ (i.e., ξ lies on the line segment connecting $\mathbf{a}$ and $\mathbf{x}$), and*

$$\begin{aligned} f(x_1,\dots,x_n) = f(a_1,\dots,a_n) &+ \sum_{i=1}^{n} \frac{\partial f(a_1,\dots,a_n)}{\partial x_i}(x_i-a_i) \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f(a_1,\dots,a_n)}{\partial x_i\,\partial x_j}(x_i-a_i)(x_j-a_j) \\ &+ \dots + \frac{1}{m!}\sum_{i_1=1}^{n}\cdots\sum_{i_m=1}^{n} \frac{\partial^m f(a_1,\dots,a_n)}{\partial x_{i_1}\cdots\partial x_{i_m}}(x_{i_1}-a_{i_1})\cdots(x_{i_m}-a_{i_m}) \\ &+ \frac{1}{(m+1)!}\sum_{i_1=1}^{n}\cdots\sum_{i_{m+1}=1}^{n} \frac{\partial^{m+1} f(\xi_1,\dots,\xi_n)}{\partial x_{i_1}\cdots\partial x_{i_{m+1}}}(x_{i_1}-a_{i_1})\cdots(x_{i_{m+1}}-a_{i_{m+1}}). \end{aligned}$$

We will use the above Taylor's formula for the cases $m = 1$ or $m = 2$, hence we will approximate a function by a first-order or a second-order Taylor polynomial. We can easily check that using the notation of the gradient and the Hessian that for $f \in C^3$ the second-order Taylor approximation can be written as

$$f(\mathbf{x}) \approx f(\mathbf{a}) + f'(\mathbf{a})^T(\mathbf{x}-\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^T f''(\mathbf{a})(\mathbf{x}-\mathbf{a}).$$

This justifies the notations f' and f'' for the gradient and the Hessian. On the other hand, we know from calculus that for a C^2 function f' and f'' are the Fret derivative of the functions f and f', respectively. We do not need the formal definition of the Fret derivative, so we can use f' and f'' as the notations of the gradient and the Hessian.

Let $I \subset \mathbb{R}$, $g\colon I \to \mathbb{R}^n$, and we denote the component functions of g by g_i, i.e., we use the notation $g(t) = (g_1(t),\dots,g_n(t))^T$. We say that such g is differentiable if all its component functions are differentiable, and its derivative is

$$g'\colon I \to \mathbb{R}^n, \qquad g'(t) := (g_1'(t),\dots,g_n'(t))^T.$$

We say that g is continuously differentiable if its each component function is continuously differentiable.

We have the follow result.

Theorem 2.39 (chain rule). *Let $f\colon \mathbb{R}^n \to \mathbb{R}$, $f \in C^1$ and $g\colon \mathbb{R} \to \mathbb{R}^n$ be continuously differentiable. Then the composite function $f \circ g\colon \mathbb{R} \to \mathbb{R}$ is also continuously differentiable, and*

$$\frac{d}{dt} f(g(t)) = f'(g(t))^T g'(t).$$

We can get the following generalization of the Lagrange's Mean Value Theorem for multivariable functions from the chain rule.

Theorem 2.40 (Lagrange's Mean Value Theorem). *Let $E \subset \mathbb{R}^n$ be an open and convex set, $f\colon E \to \mathbb{R}$ be continuously differentiable with respect to all variables. Then for every $\mathbf{x}, \mathbf{y} \in E$ there exists $\xi \in (0,1)$ such that*

$$f(\mathbf{x}) - f(\mathbf{y}) = f'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))^T(\mathbf{x} - \mathbf{y}).$$

Proof. We define the single variable function $g(t) = f(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))$ for $t \in [0,1]$. Using the Lagrange's Mean Value Theorem of single variable functions and the chain rule, we get

$$f(\mathbf{x}) - f(\mathbf{y}) = g(1) - g(0) = g'(\xi) = f'(\mathbf{x} + \xi(\mathbf{y} - \mathbf{x}))^T(\mathbf{x} - \mathbf{y}).$$

□

Let $E \subset \mathbb{R}^n$ and $\mathbf{f}\colon E \to \mathbb{R}^n$. The component functions of $\mathbf{f}$ are denoted by f_i, i.e.,

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^T.$$

We say that $\mathbf{f}$ is m times continuously partially differentiable if its every component function is m times continuously differentiable, and it will be denoted by $\mathbf{f} \in C^m$. The *Jacobian matrix* or shortly, the *Jacobian* of the function $\mathbf{f} \in C^1$ is the $n \times n$ matrix defined by

$$\mathbf{f}'(\mathbf{x}) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Let $\mathbf{a} \in \mathbb{R}^n$ be fixed. If we approximate the component functions of $\mathbf{f}$ by its first-order Taylor polynomial around $\mathbf{a}$, then we get

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix} \approx \begin{pmatrix} f_1(\mathbf{a}) + f_1'(\mathbf{a})^T(\mathbf{x} - \mathbf{a}) \\ \vdots \\ f_n(\mathbf{a}) + f_n'(\mathbf{a})^T(\mathbf{x} - \mathbf{a}) \end{pmatrix} = \mathbf{f}(\mathbf{a}) + \mathbf{f}'(\mathbf{a})(\mathbf{x} - \mathbf{a}).$$

The expression $\mathbf{f}(\mathbf{a}) + \mathbf{f}'(\mathbf{a})(\mathbf{x} - \mathbf{a})$ is called the *linear approximation* of $\mathbf{f}$ around $\mathbf{a}$.

2.10. Vector and Matrix Norms and Convergence

The components of the vector $\mathbf{x} \in \mathbb{R}^n$ are denoted by $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. The function $\|\cdot\|\colon \mathbb{R}^n \to \mathbb{R}$ is called *vector norm* if

1. $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$, and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$,
2. $\|c\mathbf{x}\| = |c|\|\mathbf{x}\|$ for all $c \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,
3. (triangle inequality:) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Theorem 2.41. *For any vector norm $\|\cdot\|$ it follows that*

(i) $\Big|\|\mathbf{x}\| - \|\mathbf{y}\|\Big| \le \|\mathbf{x}-\mathbf{y}\|$,

(ii) $\|\cdot\|$ *is a continuous function on* $\mathbb{R}^n$.

Proof. The triangle inequality yields $\|\mathbf{x}\| = \|\mathbf{x}-\mathbf{y}+\mathbf{y}\| \le \|\mathbf{x}-\mathbf{y}\| + \|\mathbf{y}\|$. Hence we get $\|\mathbf{x}\| - \|\mathbf{y}\| \le \|\mathbf{x}-\mathbf{y}\|$. Similarly, $\|\mathbf{y}\| - \|\mathbf{x}\| \le \|\mathbf{x}-\mathbf{y}\|$ holds too, so part (i) follows. The continuity of $\|\cdot\|$ follows from part (i). □

Let $p \ge 1$, and define the so-called *p-norm*:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

It can be shown that $\|\cdot\|_p$ satisfies all the three requirements of the definition of a norm for all $p \ge 1$. The norm corresponding to $p = 2$, i.e., $\|\cdot\|_2$ is called *Euclidean norm*. Another special case is the 1-*norm*:

$$\|\mathbf{x}\|_1 := \sum_{i=1}^{n} |x_i|.$$

We will also use the following vector norm, the so-called *infinity norm* or *maximum norm*

$$\|\mathbf{x}\|_\infty := \max_{i=1,\dots,n} |x_i|.$$

It is left for the reader to show that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ satisfy the norm properties (Exercise 1). The Euclidean norm is clearly satisfies the 1st and 2nd norm properties, but for the proof of the triangle inequality we need the following estimate, which is important in its own right.

Theorem 2.42 (Cauchy–Bunyakovsky–Schwarz inequality). *For every* $x_1, \dots, x_n$, $y_1, \dots, y_n \in \mathbb{R}$ *it follows*

$$\left(\sum_{i=1}^{n} x_i y_i\right)^2 \le \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2,$$

where equality holds if and only if there exists $\lambda \in \mathbb{R}$ *such that* $y_i = \lambda x_i$ *for every* $i = 1, 2, \dots, n$.

Proof. Consider the second-order polynomial $p(t) := t^2 \sum_{i=1}^n x_i^2 - 2t \sum_{i=1}^n x_i y_i + \sum_{i=1}^n y_i^2$. Then $p(t) = \sum_{i=1}^n (t x_i - y_i)^2 \ge 0$ holds for all t, so p may not have two distinct real roots, i.e., its discriminant may not be positive:

$$4\left(\sum_{i=1}^{n} x_i y_i\right)^2 - 4\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2 \le 0.$$

This yields the Cauchy-Bunyakovsky-Schwarz inequality. p has one real root if and only if its discriminant is 0, i.e., the inequality holds with equality. On the other hand, $p(t) = 0$ holds for some $t = \lambda$ if and only if $y_i = \lambda x_i$ for all $i = 1, 2, \ldots, n$. □

Taking a square root for both sides of the Cauchy–Bunyakovsky–Schwarz inequality and using vector notation we get:

Corollary 2.43. *For all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ *it follows*

$$|\mathbf{x}^T\mathbf{y}| \leq \|\mathbf{x}\|_2\|\mathbf{y}\|_2,$$

where the equality is satisfied if and only if there exists $\lambda \in \mathbb{R}$ *such that* $\mathbf{y} = \lambda\mathbf{x}$.

Using the Cauchy–Bunyakovsky–Schwarz inequality we get

$$\begin{aligned}
\|\mathbf{x}+\mathbf{y}\|_2^2 &= \sum_{i=1}^{n}(x_i+y_i)^2 \\
&= \sum_{i=1}^{n} x_i^2 + 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\
&\leq \sum_{i=1}^{n} x_i^2 + 2\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2} + \sum_{i=1}^{n} y_i^2 \\
&= \left(\sqrt{\sum_{i=1}^{n} x_i^2} + \sqrt{\sum_{i=1}^{n} y_i^2}\right)^2 \\
&= \left(\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2\right)^2,
\end{aligned}$$

which shows that the Euclidean norm satisfies the triangle inequality.

With the application of the norm we can define the length of a vector, distance between two vectors and the notion of convergence of vector sequences. The expression $\|\mathbf{x}\|$ is called the *length* of the vector, which is the distance between $\mathbf{x}$ and $\mathbf{0}$. The *distance* between the vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as the real number $\|\mathbf{x}-\mathbf{y}\|$. Let $\mathbf{p}^{(k)}$ be a sequence of n-dimensional vectors, and let $\|\cdot\|$ be a vector norm on $\mathbb{R}^n$. We say that the sequence $\mathbf{p}^{(k)}$ *converges* to $\mathbf{p}$ if

$$\lim_{k\to\infty} \|\mathbf{p}^{(k)} - \mathbf{p}\| = 0.$$

It can be proved that the notion of the convergence of a vector sequence is independent of the selection of the vector norm, i.e., if a vector sequence is convergent in a norm, it is convergent to the same limit in any other norm. This property is called in mathematical analysis as the vector norms are equivalent in $\mathbb{R}^n$.

Theorem 2.44. *Let* $|\cdot|$ *and* $\|\cdot\|$ *be two vector norms, and* $\mathbf{p}^{(k)}$ *be a sequence in* $\mathbb{R}^n$. *Then* $\lim_{k\to\infty} |\mathbf{p}^{(k)} - \mathbf{p}| = 0$ *if and only if* $\lim_{k\to\infty} \|\mathbf{p}^{(k)} - \mathbf{p}\| = 0$.

Proof. It is enough to show that for any fixed vector norm $\|\cdot\|$, $\|\mathbf{p}^{(k)}-\mathbf{p}\|\to 0$ if and only if $\|\mathbf{p}^{(k)}-\mathbf{p}\|_1\to 0$. It holds if we show that there exist nonnegative constants m and M such that

$$m\|\mathbf{p}^{(k)}-\mathbf{p}\|_1\le\|\mathbf{p}^{(k)}-\mathbf{p}\|\le M\|\mathbf{p}^{(k)}-\mathbf{p}\|_1. \tag{2.25}$$

Let $E:=\{\mathbf{x}\in\mathbb{R}^n\colon\ \|\mathbf{x}\|_1=1\}$. Then E is a bounded and closed subset of $\mathbb{R}^n$, therefore Theorems 2.37 and 2.41 yield that the continuous function $\|\cdot\|$ takes its maximum and minimum on E. Let denote them by M and m, respectively. Let $\mathbf{x}=(\mathbf{p}^{(k)}-\mathbf{p})/\|\mathbf{p}^{(k)}-\mathbf{p}\|_1$. Then $\mathbf{x}\in E$, and hence $m\le\|\mathbf{x}\|\le M$, which yields (2.25) after multiplication by $\|\mathbf{p}^{(k)}-\mathbf{p}\|_1$. □

Theorem 2.45. *Let $p_i^{(k)}$ and p_i denote the ith components of the vectors $\mathbf{p}^{(k)}$ and $\mathbf{p}$, respectively. Then the sequence $\mathbf{p}^{(k)}$ converges to $\mathbf{p}$ if and only if $p_i^{(k)}\to p_i$ for all $i=1,2,\dots,n$ as $k\to\infty$.*

Proof. Theorem 2.44 yields that $\|\mathbf{p}^{(k)}-\mathbf{p}\|\to 0$ if and only if $\|\mathbf{p}^{(k)}-\mathbf{p}\|_1=\sum_{i=1}^n|\mathbf{p}_i^{(k)}-p_i|\to 0$, which is satisfied exactly when $p_i^{(k)}\to p_i$ for all $i=1,2,\dots,n$. □

The set of $n\times n$-dimensional real matrices is denoted by $\mathbb{R}^{n\times n}$. Let $\|\cdot\|$ be a vector norm on $\mathbb{R}^n$. The function $\|\cdot\|\colon\mathbb{R}^{n\times n}\to\mathbb{R}$ defined by the formula

$$\|\mathbf{A}\|:=\sup_{\mathbf{x}\neq\mathbf{0}}\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$$

is called the *matrix norm* generated by the vector norm $\|\cdot\|$. We note that both the vector and the matrix norms are denoted by the same symbol. It is possible to show that in the definition of the matrix norm sup can be replaced by max, i.e., there exists a vector $\mathbf{x}$ such that $\|\mathbf{A}\|=\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$. The following properties of the matrix norm can be proved easily:

Theorem 2.46. *For every $\mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times n}$ it follows*

(i) $\|\mathbf{A}\|\ge 0$, *and* $\|\mathbf{A}\|=0$ *if and only if* $\mathbf{A}=\mathbf{0}$,

(ii) $\|c\mathbf{A}\|=|c|\|\mathbf{A}\|$ *for all* $c\in\mathbb{R}$,

(iii) (triangle inequality:) $\|\mathbf{A}+\mathbf{B}\|\le\|\mathbf{A}\|+\|\mathbf{B}\|$,

(iv) $\|\mathbf{A}\mathbf{x}\|\le\|\mathbf{A}\|\|\mathbf{x}\|$, *for all* $\mathbf{x}\in\mathbb{R}^n$*-re,*

(v) $\|\mathbf{A}\mathbf{B}\|\le\|\mathbf{A}\|\|\mathbf{B}\|$,

(vi) $\|\mathbf{A}\|=\sup\{\|\mathbf{A}\mathbf{y}\|\colon\ \|\mathbf{y}\|=1\}$.

Proof. The proof of statements (i), (ii) and (iii) are left for the reader. Part (iv) follows from

$$\frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} \leq \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{Ay}\|}{\|\mathbf{y}\|} = \|\mathbf{A}\|.$$

Using (iv) we get

$$\frac{\|\mathbf{ABx}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\| \frac{\|\mathbf{Bx}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\|\|\mathbf{B}\|,$$

hence

$$\|\mathbf{AB}\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{ABx}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\|\|\mathbf{B}\|.$$

Finally, (vi) follows from

$$\frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} = \left\| \mathbf{A} \frac{\mathbf{x}}{\|\mathbf{x}\|} \right\|.$$

□

We note that the matrix norm could be defined in a more general way: a function $\|\cdot\|: \mathbb{R}^{n\times n} \to \mathbb{R}$ which satisfies parts (i)–(iii) of Theorem 2.46. Then there are matrix norms which are not generated by a vector norm. In this lecture note we will use only matrix norms generated by vector norms, so we use this notion in this restrictive sense.

We will need the notion of limits of matrix sequences later. We say that a matrix sequence $\mathbf{A}^{(k)}$ *converges* to a limit $\mathbf{A}$ if $\lim_{k\to\infty} \|\mathbf{A}^{(k)} - \mathbf{A}\| = 0$, where $\|\cdot\|$ is a matrix norm. The next theorem states that the limit of a matrix sequence is independentof the selection of the matrix norm, i.e., any matrix norms are equivalent.

Theorem 2.47. *Let $|\cdot|$ and $\|\cdot\|$ be two vector norms on $\mathbb{R}^n$, and we consider the corresponding matrix norms on $\mathbb{R}^{n\times n}$. Let $\mathbf{A}^{(k)}$ be a sequence in $\mathbb{R}^{n\times n}$. Then $\lim_{k\to\infty} |\mathbf{A}^{(k)} - \mathbf{A}| = 0$ if and only if $\lim_{k\to\infty} \|\mathbf{A}^{(k)} - \mathbf{A}\| = 0$.*

Proof. As in the proof of Theorem 2.44, it is enough to show that there exist nonnegative constants l and L such that

$$l|\mathbf{B}| \leq \|\mathbf{B}\| \leq L|\mathbf{B}|, \qquad \mathbf{B} \in \mathbb{R}^{n\times n}.$$

From the proof of Theorem 2.44 we know that there exist positive constants m and M such that

$$m|\mathbf{x}| \leq \|\mathbf{x}\| \leq M|\mathbf{x}|, \qquad x \in \mathbb{R}^n.$$

Then

$$\frac{m}{M}|\mathbf{B}| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{m|\mathbf{Bx}|}{M|\mathbf{x}|} \leq \|\mathbf{B}\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Bx}\|}{\|\mathbf{x}\|} \leq \sup_{\mathbf{x}\neq\mathbf{0}} \frac{M|\mathbf{Bx}|}{m|\mathbf{x}|} = \frac{M}{m}|\mathbf{B}|,$$

which completes the proof. □

For matrix norms we most frequently use the norms generated by the $\|\cdot\|_1$ and $\|\cdot\|_\infty$ vector norms. We have the following result for the computation of the corresponding matrix norms.

Theorem 2.48. *Let $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$. Then the matrix norms generated by the $\|\cdot\|_1$ and $\|\cdot\|_\infty$ vector norms satisfy*

$$\|\mathbf{A}\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|,$$

and

$$\|\mathbf{A}\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|.$$

Proof. We prove the first formula. The second formula is asked to be proved by the reader. Using the definition of the $\|\cdot\|_1$ vector norm and the triangle inequality we get

$$\begin{aligned}
\|\mathbf{Ax}\|_1 &= \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \\
&\le \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}x_j| \\
&= \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{ij}| \\
&\le \left(\max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}| \right) \sum_{j=1}^{n} |x_j| \\
&= \left(\max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}| \right) \|\mathbf{x}\|_1,
\end{aligned}$$

hence $\|\mathbf{A}\|_1 \le \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|$. Suppose $\max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}| = \sum_{i=1}^{n} |a_{ik}|$. We get the statement by multiplying $\mathbf{A}$ and $\mathbf{e}^{(k)} = (0,\dots,0,1,0,\dots,0)^T$, where $e_i^{(k)} = 0$ if $i \ne k$ and $e_k^{(k)} = 1$. Indeed, $\mathbf{Ae}^{(k)} = (a_{1k}, a_{2k}, \dots, a_{nk})^T$, therefore $\|\mathbf{Ae}^{(k)}\|_1 = \sum_{i=1}^{n} |a_{ik}|$. □

The following results generalize the properties of the convergence for the vector case. We formulate the statements without proofs.

Theorem 2.49.

1. *If the vector sequence $\mathbf{p}^{(k)}$ is convergent, then its limit is unique.*
2. *If $\mathbf{p}^{(k)} \to \mathbf{p}$ and $\mathbf{q}^{(k)} \to \mathbf{q}$, $\alpha, \beta \in \mathbb{R}$, then the sequence $\alpha\mathbf{p}^{(k)} + \beta\mathbf{q}^{(k)}$ is also convergent, and $\alpha\mathbf{p}^{(k)} + \beta\mathbf{q}^{(k)} \to \alpha\mathbf{p} + \beta\mathbf{q}$.*
3. *If $c_k \to c$ a real sequence and $\mathbf{p}^{(k)} \to \mathbf{p}$, then $c_k\mathbf{p}^{(k)} \to c\mathbf{p}$.*
4. *If $\mathbf{p}^{(k)} \to \mathbf{p}$, then $\mathbf{Ap}^{(k)} \to \mathbf{Ap}$ for all $\mathbf{A} \in \mathbb{R}^{n\times n}$.*
5. *(Cauchy's criterion for convergence) $\mathbf{p}^{(k)}$ is a convergent sequence if and only if it is a Cauchy sequence, i.e., for every $\varepsilon > 0$ there exists a $k_0 > 0$ such that $\|\mathbf{p}^{(k)} - \mathbf{p}^{(m)}\| < \varepsilon$ for all $k, m > k_0$.*

We can also generalize Theorems 2.44, 2.45 and 2.49 for matrices. Using vector and matrix norms we can extend the Lagrange's Mean Value Theorem for vector valued functions.

Theorem 2.50 (Lagrange's Mean Value Theorem). *Let $\|\cdot\|$ be a fixed vector norm on $\mathbb{R}^n$, and consider the generated matrix norm. Let $E \subset \mathbb{R}^n$ be an open and convex set, $\mathbf{f}\colon E \to \mathbb{R}^n$ be continuously partially differentiable, $\mathbf{x}, \mathbf{y} \in E$. Then*

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq \max_{t\in[0,1]} \|\mathbf{f}'(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))\| \cdot \|\mathbf{x} - \mathbf{y}\|.$$

Proof. We prove the statement only for the Euclidean norm $\|\cdot\| = \|\cdot\|_2$. Clearly, we can assume that $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{y})$. Let

$$\mathbf{h} := \frac{\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})}{\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_2}.$$

Then $\|\mathbf{h}\|_2 = 1$. Let $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^T$, $\mathbf{h} = (h_1, \dots, h_n)^T$. We define the real function

$$g(t) := \mathbf{h}^T \mathbf{f}(\mathbf{y} + t(\mathbf{x} - \mathbf{y})) = \sum_{i=1}^{n} h_i f_i(\mathbf{y} + t(\mathbf{x} - \mathbf{y})).$$

Then, using the Langrange's Mean Value Theorem for single variable functions and the chain rule, we get

$$\begin{aligned}
\mathbf{h}^T(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})) &= g(1) - g(0) \\
&= g'(\xi) \\
&= \sum_{i=1}^{n} h_i f_i'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))^T(\mathbf{x} - \mathbf{y}) \\
&= \mathbf{h}^T \mathbf{f}'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y})
\end{aligned}$$

for some $\xi \in (0, 1)$. Therefore the definition of $\mathbf{h}$, the vector form of the Cauchy-Bunyakovsky-Schwarz inequality, $\|\mathbf{h}\|_2 = 1$ and part (v) of Theorem 2.46 yield

$$\begin{aligned}
\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_2 &= \mathbf{h}^T(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})) \\
&= \mathbf{h}^T \mathbf{f}'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y}) \\
&\leq \|\mathbf{h}\|_2 \|f'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y})\|_2 \\
&\leq \|f'(\mathbf{y} + \xi(\mathbf{x} - \mathbf{y}))\|_2 \|\mathbf{x} - \mathbf{y}\|_2,
\end{aligned}$$

which concludes the proof. □

Exercises

1. Show that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ satisfy the properties of the norms.
2. Compute $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_\infty$, and $\|\mathbf{A}\|_1$ and $\|\mathbf{A}\|_\infty$ for

   (a) $\mathbf{x} = (3, -1, 0, 5)^T$, (b) $\mathbf{x} = (-3, -2, -1, 4, -1)^T$,

   and

   (c) $\mathbf{A} = \begin{pmatrix} -1 & 3 & -2 \\ 2 & -4 & 0 \\ 0 & 3 & 2 \end{pmatrix}$, (d) $\mathbf{A} = \begin{pmatrix} -1 & 2 & 4 \\ 2 & -3 & 5 \\ 7 & -2 & 3 \end{pmatrix}$.

3. Draw the graphs of the curves defined by

   (a) $\{\mathbf{x} \in \mathbb{R}^2 \colon \|\mathbf{x}\|_1 = 1\}$, (b) $\{\mathbf{x} \in \mathbb{R}^2 \colon \|\mathbf{x}\|_\infty = 1\}$.

4. Prove parts (i)–(iii) of Theorem 2.46.
5. Prove part (ii) of Theorem 2.48.
6. Prove Theorem 2.49.

2.11. Fixed-Point Iteration in n-dimension

We can generalize the notion of the fixed point and the fixed-point iteration for multivariable functions.

Example 2.51. Consider the system

$$\begin{aligned} 4x_1 - e^{x_1 x_2} - 3 &= 0 \\ x_1 - x_2^2 - 3x_2 - 1 &= 0. \end{aligned} \tag{2.26}$$

It is easy to check that $x_1 = 1$ and $x_2 = 0$ is a solution of the system. We rearrange (2.26) in the following way. We express x_1 from the first, and x_2 from the second equation:

$$\begin{aligned} x_1 &= \tfrac{1}{4}(e^{x_1 x_2} + 3) \\ x_2 &= \tfrac{1}{3}(x_1 - x_2^2 - 1) \end{aligned} \tag{2.27}$$

We can denote system (2.27) shorty as $\mathbf{x} = \mathbf{g}(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2)^T$ and

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(x_1, x_2) = \begin{pmatrix} \frac{1}{4}(e^{x_1 x_2} + 3) \\ \frac{1}{3}(x_1 - x_2^2 - 1) \end{pmatrix}. \tag{2.28}$$

We define an iteration to approximate the solutions of (2.27) as in the single variable case for $k = 0, 1, 2, \ldots$ by

$$\begin{aligned} p_1^{(k+1)} &= \tfrac{1}{4}(e^{p_1^{(k)} p_2^{(k)}} + 3) \\ p_2^{(k+1)} &= \tfrac{1}{3}\left(p_1^{(k)} - (p_2^{(k)})^2 - 1\right). \end{aligned} \tag{2.29}$$

We have listed the first several terms of the sequences $p_1^{(k)}$ and $p_2^{(k)}$ starting from the initial value $p_1^{(0)} = -2$ and $p_2^{(0)} = -2$. in Table 2.12. We can observe that the sequences converge to 1 and 0, respectively.

Table 2.12: Fixed-point iteration

k	$p_1^{(k)}$	$p_2^{(k)}$
0	-2.000000000	-2.000000000
1	14.399537510	-2.333333333
2	0.750000000	2.651697690
3	2.576641266	-2.427166879
4	0.750480717	-1.438165931
5	0.834956989	-0.772613509
6	0.881152644	-0.253991549
7	0.949867689	-0.061119687
8	0.985899367	-0.017955976
9	0.995613247	-0.004807684
10	0.998806211	-0.001469956
11	0.999633219	-0.000398650
12	0.999900394	-0.000122313

Defining the vector sequence $\mathbf{p}^{(k)} = (p_1^{(k)}, p_2^{(k)})^T$, iteration (2.29) can be written shortly as $\mathbf{p}^{(k+1)} = \mathbf{g}(\mathbf{p}^{(k)})$. □

Let $E \subset \mathbb{R}^n$, and consider a function $\mathbf{g}\colon E \to \mathbb{R}^n$. Similarly to the single variable case, we say that a vector $\mathbf{p} \in E$ is a *fixed point* of the function $\mathbf{g}$ if $\mathbf{p} = \mathbf{g}(\mathbf{p})$.

A function $\mathbf{g}\colon E \to \mathbb{R}^n$ is called a *contraction* on the set E using the vector norm $\|\cdot\|$ if there exists a constant $0 \le c < 1$ such that $\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \le c\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in E$. Note that a contraction is always a continuous function.

Theorem 2.52 (fixed-point theorem). *Let $E \subset \mathbb{R}^n$ be a closed set, $\mathbf{g}\colon E \to E$, and let $\mathbf{g}$ be a contraction on E using a vector norm $\|\cdot\|$. Then $\mathbf{g}$ has a unique fixed point $\mathbf{p} \in E$, and the fixed-point iteration $\mathbf{p}^{(k+1)} = \mathbf{g}(\mathbf{p}^{(k)})$ converges to $\mathbf{p}$ for all $\mathbf{p}^{(0)} \in E$. The order of convergence is (at least) linear.*

Proof. First we show that $\mathbf{p}^{(k)}$ is a Cauchy sequence. Let c be the Lipschitz constant of the function $\mathbf{g}$, and let $k > m$. Similarly to the single variable case, the definition of the sequence and the contraction property yield

$$\begin{aligned}
&\|\mathbf{p}^{(k)} - \mathbf{p}^{(m)}\| \\
&\le \|\mathbf{p}^{(k)} - \mathbf{p}^{(k-1)}\| + \|\mathbf{p}^{(k-1)} - \mathbf{p}^{(k-2)}\| + \cdots + \|\mathbf{p}^{(m+1)} - \mathbf{p}^{(m)}\| \\
&= \|\mathbf{g}(\mathbf{p}^{(k-1)}) - \mathbf{g}(\mathbf{p}^{(k-2)})\| + \|\mathbf{g}(\mathbf{p}^{(k-2)}) - \mathbf{g}(\mathbf{p}^{(k-3)})\| \\
&\qquad + \cdots + \|\mathbf{g}(\mathbf{p}^{(m)}) - \mathbf{g}(\mathbf{p}^{(m-1)})\| \\
&\le c(\|\mathbf{p}^{(k-1)} - \mathbf{p}^{(k-2)}\| + \|\mathbf{p}^{(k-2)} - \mathbf{p}^{(k-3)}\| + \cdots + \|\mathbf{p}^{(m)} - \mathbf{p}^{(m-1)}\|) \\
&\le (c^{k-1} + c^{k-2} + \cdots + c^m)\|\mathbf{p}^{(1)} - \mathbf{p}^{(0)}\| \\
&= c^m(c^{k-m-1} + c^{k-m-2} + \cdots + 1)\|\mathbf{p}^{(1)} - \mathbf{p}^{(0)}\| \\
&\le c^m \sum_{i=0}^{\infty} c^i \|\mathbf{p}^{(1)} - \mathbf{p}^{(0)}\|.
\end{aligned}$$

Therefore we get $\|\mathbf{p}^{(k)} - \mathbf{p}^{(m)}\| \to 0$ as $m \to \infty$, hence $\mathbf{p}^{(k)}$ is a Cauchy sequence. Part (v) of Theorem 2.49 implies that $\mathbf{p}^{(k)}$ converges to a vector $\mathbf{p}$. Using the continuity of $\mathbf{g}$ we get $\mathbf{p}^{(k+1)} = \mathbf{g}(\mathbf{p}^{(k)}) \to \mathbf{g}(\mathbf{p})$, and so $\mathbf{p} = \mathbf{g}(\mathbf{p})$, i.e., $\mathbf{p}$ is a fixed-point of $\mathbf{g}$.

The order of convergence is at least linear, since

$$\|\mathbf{p}^{(k+1)} - \mathbf{p}\| = \|\mathbf{g}(\mathbf{p}^{(k)}) - \mathbf{g}(\mathbf{p})\| \le c\|\mathbf{p}^{(k)} - \mathbf{p}\|.$$

Suppose that $\mathbf{p}$ and $\bar{\mathbf{p}}$ both are fixed points of $\mathbf{g}$. Using the contraction property of $\mathbf{g}$ we have $\|\mathbf{p} - \bar{\mathbf{p}}\| = \|\mathbf{g}(\mathbf{p}) - \mathbf{g}(\bar{\mathbf{p}})\| \le c\|\mathbf{p} - \bar{\mathbf{p}}\|$, and therefore, $\mathbf{p} = \bar{\mathbf{p}}$ follows. □

Theorem 2.53. *Let $E \subset \mathbb{R}^n$ be an open set, $\mathbf{g}\colon E \to \mathbb{R}^n$, $\mathbf{g} \in C^1$, and let $\mathbf{p}$ be a fixed point of $\mathbf{g}$. If $\|\mathbf{g}'(\mathbf{p})\| < 1$ in a matrix norm generated by a vector norm $\|\cdot\|$, then the fixed-point iteration $\mathbf{p}^{(k+1)} = \mathbf{g}(\mathbf{p}^{(k)})$ converges locally to $\mathbf{p}$.*

Proof. Since E is an open set, there exists a radius $\bar{\delta} > 0$ such that $\{\mathbf{x}\colon \|\mathbf{x} - \mathbf{p}\| < \bar{\delta}\} \subset E$. Fix a c such that $\|\mathbf{g}'(\mathbf{p})\| < c < 1$. The function $\mathbf{g}'$ is continuous at $\mathbf{p}$, therefore there exists $0 < \delta \le \bar{\delta}$ such that $\|\mathbf{g}'(\mathbf{x})\| \le c$ for all $\mathbf{x} \in V := \{\mathbf{x}\colon \|\mathbf{x} - \mathbf{p}\| \le \delta\}$. The Lagrange's Mean Value Theorem (Theorem 2.50) yields

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \le \max_{t\in(0,1)} \|\mathbf{g}'(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))\| \cdot \|\mathbf{x} - \mathbf{y}\| \le c\|\mathbf{x} - \mathbf{y}\|,$$

i.e., $\mathbf{g}$ is a contraction.

Now we show that the function $\mathbf{g}$ maps the set V into itself. Let $\mathbf{x} \in V$. The contraction property of $\mathbf{g}$ implies $\|\mathbf{g}(\mathbf{x}) - \mathbf{p}\| = \|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{p})\| \le c\|\mathbf{x} - \mathbf{p}\| < \delta$, hence $\mathbf{g}(\mathbf{x}) \in V$. If we restrict $\mathbf{g}$ to the set V, then this function satisfies the conditions of Theorem 2.52, therefore any fixed-point iteration with initial value from V converges to $\mathbf{p}$. □

Example 2.54. Compute the Jacobian matrix of the function $\mathbf{g}$ defined by (2.28) in Example 2.51:

$$\mathbf{g}'(\mathbf{x}) = \begin{pmatrix} \frac{1}{4}x_2 e^{x_1x_2} & \frac{1}{4}x_1 e^{x_1x_2} \\ \frac{1}{3} & -\frac{2}{3}x_2 \end{pmatrix}.$$

Its value at the fixed point of $\mathbf{g}$, i.e., at the point $(1,0)^T$ is

$$\mathbf{g}'(1,0) = \begin{pmatrix} 0 & \frac{1}{4} \\ \frac{1}{3} & 0 \end{pmatrix}.$$

Its 1-norm is $\|\mathbf{g}'(1,0)\|_1 = \frac{1}{3} < 1$, hence Theorem 2.53 yields that the fixed-point iteration converges locally to $(1,0)^T$. □

Theorem 2.55. *Let $E \subset \mathbb{R}^n$, $\mathbf{g}\colon E \to \mathbb{R}^n$, $\mathbf{g} \in C^2$, $\mathbf{g}(\mathbf{p}) = \mathbf{p}$, and $\mathbf{g}'(\mathbf{p}) = \mathbf{0}$. Then there exists a $\delta > 0$ such that the fixed-point iteration $\mathbf{p}^{(k+1)} = \mathbf{g}(\mathbf{p}^{(k)})$ converges to $\mathbf{p}$ if $\|\mathbf{p}^{(0)} - \mathbf{p}\|_\infty < \delta$. Moreover, there exists a constant c such that for all k it follows $\|\mathbf{p}^{(k+1)} - \mathbf{p}\|_\infty \le c\|\mathbf{p}^{(k)} - \mathbf{p}\|_\infty^2$, i.e., the iteration converges locally quadratically to $\mathbf{p}$.*

Proof. By the assumptions, $0 = \|\mathbf{g}'(\mathbf{p})\| < 1$, therefore, Theorem 2.53 yields that the fixed-point iteration is locally convergent.

Now we show that the order of convergence is quadratic. Consider the second-order Taylor approximation of the ith component function of $\mathbf{g}$ around $\mathbf{p} = (p_1, \ldots, p_n)^T$:

$$g_i(x_1, \ldots, x_n) = g_i(p_1, \ldots, p_n) + \sum_{j=1}^{n} \frac{\partial g_i(p_1, \ldots, p_n)}{\partial x_j}(x_j - p_j) + \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 g_i(\xi_1, \ldots, \xi_n)}{\partial x_j\, \partial x_l}(x_j - p_j)(x_l - p_l).$$

Applying this relation for $(x_1, \ldots, x_n)^T = (p_1^{(k)}, \cdots, p_n^{(k)})^T$, and using that $p_i = g_i(\mathbf{p})$ and $p_i^{(k+1)} = g_i(\mathbf{p}^{(k)})$, we get

$$p_i^{(k+1)} - p_i = \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 g_i(\xi_1, \ldots, \xi_n)}{\partial x_j\, \partial x_l}(p_j^{(k)} - p_j)(p_l^{(k)} - p_l).$$

Let M be such that $\left|\frac{\partial^2 g_i(x_1,\ldots,x_n)}{\partial x_j\, \partial x_l}\right| \le M$ for all $i, j, l = 1, \ldots, n$ in a neighborhood of $\mathbf{p}$ which contains all $\mathbf{p}^{(k)}$. The definition of M implies

$$|p_i^{(k+1)} - p_i| \le \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{n} M|p_j^{(k)} - p_j||p_l^{(k)} - p_l| \le \frac{n^2}{2} M\|\mathbf{p}^{(k)} - \mathbf{p}\|_\infty^2.$$

Since this holds for all $i = 1, \ldots, n$, we get

$$\|\mathbf{p}^{(k+1)} - \mathbf{p}\|_\infty \le \frac{n^2}{2} M\|\mathbf{p}^{(k)} - \mathbf{p}\|_\infty^2,$$

i.e., the order of convergence is quadratic. □

Exercises

1. Rewrite the following system as a fixed-point problem, and find an approximate solution by using the fixed-point iteration from the starting value $(0, 0)^T$:

   (a)
   $$\begin{aligned} -2x^2 + 6x - y^2 &= 4 \\ x^2 + y^3 - 5y &= 3 \end{aligned}$$

   (b)
   $$\begin{aligned} 8x + \cos x - y^3 &= -7 \\ x^2 + 4y &= 8 \end{aligned}$$

   (c)
   $$\begin{aligned} x^2 + 7x + y^2 - 4y &= 3 \\ 2x + y^3 + 4y &= -5 \end{aligned}$$

   (d)
   $$\begin{aligned} \cos x - 5y &= 3 \\ x^2 - 6x + y^2 - 2y &= 4 \end{aligned}$$

2. Compute the Jacobian matrix of the fixed-point functions we get in the problem mentioned above, and evaluate the norm of the Jacobian matrix at the fixed point obtained numerically.
3. Show that under the conditions of Theorem 2.55, the sequence $\mathbf{p}^{(k)}$ converges locally quadratically in any vector norm.

2.12. Newton's Method in n-dimension

Let $U \subset \mathbb{R}^n$ be an open set, $\mathbf{f}\colon U \to \mathbb{R}^n$, and consider the nonlinear system

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}.$$

Fix a vector $\mathbf{p}^{(k)} \in U$. As in the scalar case, we approximate $\mathbf{f}$ by its linear part $\mathbf{f}(\mathbf{p}^{(k)}) + \mathbf{f}'(\mathbf{p}^{(k)})(\mathbf{x} - \mathbf{p}^{(k)})$. Its root is $\bar{\mathbf{x}} = \mathbf{p}^{(k)} - (\mathbf{f}'(\mathbf{p}^{(k)}))^{-1}\mathbf{f}(\mathbf{p}^{(k)})$, assuming that $\mathbf{f}'(\mathbf{p}^{(k)})$ is invertible. Therefore we define the Newton's method by the iteration

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \left(\mathbf{f}'(\mathbf{p}^{(k)})\right)^{-1} \mathbf{f}(\mathbf{p}^{(k)}). \tag{2.30}$$

Theorem 2.56. *Let $\mathbf{f} \in C^2$, $\mathbf{f}(\mathbf{p}) = \mathbf{0}$ and suppose the matrix $\mathbf{f}'(\mathbf{p})$ is invertible. Then the Newton's iteration (2.30) converges locally quadratically to $\mathbf{p}$.*

Proof. The Newton's method is a fixed-point iteration with the iteration function

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} - (\mathbf{f}'(\mathbf{x}))^{-1}\mathbf{f}(\mathbf{x}).$$

Let $(\mathbf{f}'(\mathbf{x}))^{-1} = (b_{ij}(\mathbf{x}))_{n\times n}$. Then

$$\sum_{j=1}^{n} b_{ij}(\mathbf{x}) \frac{\partial f_j(\mathbf{x})}{\partial x_l} = \delta_{il} := \begin{cases} 1, & i = l, \\ 0, & i \neq l. \end{cases} \tag{2.31}$$

Consider the ith component of $\mathbf{g}$:

$$g_i(\mathbf{x}) = x_i - \sum_{j=1}^{n} b_{ij}(\mathbf{x}) f_j(\mathbf{x}).$$

Taking its partial derivative with respect to x_l we get

$$\frac{\partial g_i(\mathbf{x})}{\partial x_l} = \delta_{il} - \sum_{j=1}^{n} \left(\frac{\partial b_{ij}(\mathbf{x})}{\partial x_l} f_j(\mathbf{x}) + b_{ij}(\mathbf{x}) \frac{\partial f_j(\mathbf{x})}{\partial x_l} \right).$$

At the point $\mathbf{x} = \mathbf{p}$ we get, using relations $f_j(\mathbf{p}) = 0$ and (2.31), that

$$\frac{\partial g_i(\mathbf{p})}{\partial x_l} = \delta_{il} - \sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j(\mathbf{p})}{\partial x_l} = 0.$$

Therefore, $\mathbf{g}'(\mathbf{p}) = \mathbf{0}$, and hence Theorem 2.55 yields that the iteration is locally quadratically convergent. □

Applying formula (2.30) we need to compute the inverse of a matrix. Instead of it, in practice, we do the following: Introduce the notation $\mathbf{s}^{(k)} := \mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}$, and rearrange equation (2.30) into the form

$$\mathbf{f}'(\mathbf{p}^{(k)})\mathbf{s}^{(k)} = -\mathbf{f}(\mathbf{p}^{(k)}).$$

Table 2.13: Newton's method

k	$\mathbf{p}^{(k)}$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_\infty$
0	(-1.50000000000,-1.50000000000)^T	2.500000e+00
1	(-1.25000000000,-0.52120413480)^T	2.250000e+00
2	(0.53188386800,-0.10035922100)^T	4.681161e-01
3	(0.98873605300,-0.00042581408)^T	1.126395e-02
4	(0.99999868610,-0.00000037764)^T	1.313900e-06

We solve it for $\mathbf{s}^{(k)}$, and let $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \mathbf{s}^{(k)}$.

Example 2.57. Consider the system (2.26) of Example 2.51. We apply the Newton's method for this system starting from the initial value $(-1.5, -1.5)^T$. Table 2.13 lists the numerical result. We observe quick convergence to the true solution $\mathbf{p} = (1, 0)^T$. □

Exercises

1. Apply the Newton's method to solve the equations in Exercise 1 of Section 2.11.

2.13. Quasi-Newton Methods, Broyden's Method

The advantage of Newton's method is its fast speed of (local) convergence, but its disadvantage is that the computation of the Jacobian matrix is, in general, requires many arithmetic operations. Also, it requires matrix inversion or solution of a linear equation which is also computationally expensive. To avoid or reduce these problems we introduce *quasi-Newton methods* which are defined by

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \left(\mathbf{A}^{(k)}\right)^{-1} \mathbf{f}(\mathbf{p}^{(k)}). \tag{2.32}$$

Here the matrix $\mathbf{A}^{(k)}$ is an approximation of the Jacobian $\mathbf{f}'(\mathbf{p}^{(k)})$. Using different approximations, we get different classes of quasi-Newton methods.

One typical approach is to approximate the Jacobian matrix numerically. Let $\mathbf{e}^{(j)} = (0, \dots, 0, 1, 0, \dots, 0)^T$ be the jth standard unit vector, $h > 0$ be a small discretization constant, and define the components of $\mathbf{A}^{(k)}$ by the expressions

$$a_{ij}^{(k)} = \frac{f_i(\mathbf{p}^{(k)} + h\mathbf{e}^{(j)}) - f_i(\mathbf{p}^{(k)})}{h}, \qquad i, j = 1, \dots, n. \tag{2.33}$$

The resulting quasi-Newton method is a straightforward generalization of the secant method for the vector case.

Next we introduce an other popular selection of the matrices $\mathbf{A}^{(k)}$. This method is called *Broyden's method*. This is a different generalization of the secant method for the vector case.

For scalar equations the secan method replaces the nonlinear equation $f(x) = 0$ by a linear equation

$$f(p_k) + a_k(x - p_k) = 0,$$

where $a_k = (f(p_k) - f(p_{k-1}))/(p_k - p_{k-1})$. We replace k by $k+1$, and we rewrite the equation, we get that a_{k+1} solves the equation

$$a_{k+1}(p_{k+1} - p_k) = f(p_{k+1}) - f(p_k). \tag{2.34}$$

We will generalize this formula for the vector case.

Select an initial vector $\mathbf{p}^{(0)}$ and an initial matrix $\mathbf{A}^{(0)}$. For the selection of $\mathbf{A}^{(0)}$ we can use different strategies: it is possible to use the exact value $\mathbf{A}^{(0)} = \mathbf{f}'(\mathbf{p}^{(0)})$, or using the formula (2.33) we can compute an approximate derivative matrix at $\mathbf{p}^{(0)}$, or just select any invertible matrix $\mathbf{A}^{(0)}$.

Suppose $\mathbf{p}^{(k)}$ and $\mathbf{A}^{(k)}$ are already defined. Then we define $\mathbf{p}^{(k+1)}$ by formula (2.32). Similarly to equation (2.34), we require that $\mathbf{A}^{(k+1)}$ satisfies the so-called *secant equation*

$$\mathbf{A}^{(k+1)}(\mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}) = \mathbf{f}(\mathbf{p}^{(k+1)}) - \mathbf{f}(\mathbf{p}^{(k)}). \tag{2.35}$$

We introduce the following notations

$$\mathbf{y}^{(k)} := \mathbf{f}(\mathbf{p}^{(k+1)}) - \mathbf{f}(\mathbf{p}^{(k)}) \qquad \text{and} \qquad \mathbf{s}^{(k)} := \mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}.$$

Using these notations, equations (2.32) and (2.35) are equivalent to

$$\mathbf{A}^{(k)}\mathbf{s}^{(k)} = -\mathbf{f}(\mathbf{p}^{(k)}), \tag{2.36}$$

and

$$\mathbf{A}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}, \tag{2.37}$$

respectively. First we solve (2.36) for $\mathbf{s}^{(k)}$ (assuming that $\mathbf{A}^{(k)}$ is invertible), so the problem is reduced to the selection of a matrix $\mathbf{A}^{(k+1)}$ which satisfies equation (2.37). Unfortunately, this equation does not determine the matrix $\mathbf{A}^{(k+1)}$ uniquely, since this equation is equivalent to n number of scalar equations, but $\mathbf{A}^{(k+1)}$ is determined by n^2 number of components. Equation (2.37) requires that the linear operator $\mathbf{A}^{(k+1)}$ is defined on the one dimensional space spanned by the vector $\mathbf{s}^{(k)}$. But in the $n-1$ directions orthogonal to the vector $\mathbf{s}^{(k)}$ the linear map is undetermined. Since in the $k+1$-th step we "do not have new information" about the next linear operator, i.e., the next matrix, we define $\mathbf{A}^{(k+1)}$ so that its effect on this subspace be the same as the matrix $\mathbf{A}^{(k)}$. Therefore, in addition to equation (2.37), we require

$$\mathbf{A}^{(k+1)}\mathbf{z} = \mathbf{A}^{(k)}\mathbf{z}, \qquad \text{for all } \mathbf{z} \perp \mathbf{s}^{(k)}. \tag{2.38}$$

Equations (2.37) and (2.38) together determine the matrix $\mathbf{A}^{(k+1)}$ uniquely. It can be checked easily (see Exercise 2) that the matrix

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2} \tag{2.39}$$

satisfies both (2.37) and (2.38).

The recursion (2.32) requires the computation of $(\mathbf{A}^{(k)})^{-1}$. The next result is an efficient way to compute it.

Theorem 2.58 (Sherman–Morrison–Woodbury). *Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, $\mathbf{u}, \mathbf{v} \neq \mathbf{0}$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ be inverible. Then the matrix $\mathbf{A}+\mathbf{u}\mathbf{v}^T$ is inverible if and only if $1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u} \neq 0$, and then*

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

holds.

Proof. Let $\gamma \in \mathbb{R}$, and consider

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)(\mathbf{A}^{-1} - \gamma\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}) = \mathbf{I} + \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} - \gamma\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} - \gamma\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}.$$

Since $\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}$ is a scalar, we can rewrite the above relation as

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)(\mathbf{A}^{-1} - \gamma\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}) = \mathbf{I} + (1 - \gamma - \gamma\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u})\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1},$$

which proves the statement. □

A little computation and Theorem 2.58 give from (2.39)

$$\begin{aligned}
(\mathbf{A}^{(k+1)})^{-1} &= \left(\mathbf{A}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2}\right)^{-1} \\
&= (\mathbf{A}^{(k)})^{-1} - \frac{(\mathbf{A}^{(k)})^{-1}\left(\frac{\mathbf{y}^{(k)}-\mathbf{A}^{(k)}\mathbf{s}^{(k)}}{\|\mathbf{s}^{(k)}\|_2^2}\right)(\mathbf{s}^{(k)})^T(\mathbf{A}^{(k)})^{-1}}{1 + (\mathbf{s}^{(k)})^T(\mathbf{A}^{(k)})^{-1}\frac{\mathbf{y}^{(k)}-\mathbf{A}^{(k)}\mathbf{s}^{(k)}}{\|\mathbf{s}^{(k)}\|_2^2}} \\
&= (\mathbf{A}^{(k)})^{-1} - \frac{\left((\mathbf{A}^{(k)})^{-1}\mathbf{y}^{(k)} - \mathbf{s}^{(k)}\right)(\mathbf{s}^{(k)})^T(\mathbf{A}^{(k)})^{-1}}{(\mathbf{s}^{(k)})^T(\mathbf{A}^{(k)})^{-1}\mathbf{y}^{(k)}}. \qquad (2.40)
\end{aligned}$$

Using iteration (2.40), if $(\mathbf{A}^{(k)})^{-1}$ is known, then only matrix multiplication is needed to compute $(\mathbf{A}^{(k+1)})^{-1}$, so n^2 number of arithmetic operation is enough to generate the next matrix. On the other hand, in the next chapter we will show that the matrix inversion needs n^3 number of operation, so here we have an efficient computational method.

It can be shown that the Broyden's method converges locally to a root $\mathbf{p}$ of $\mathbf{f}$ if $\mathbf{A}^{(0)}$ is close enough to $\mathbf{f}'(\mathbf{p})$, and the order of convergence is superlinear, i.e.,

$$\lim_{k\to\infty} \frac{\|\mathbf{p}^{(k+1)} - \mathbf{p}\|}{\|\mathbf{p}^{(k)} - \mathbf{p}\|} = 0.$$

We do not prove this result here. A possible definition of the Broyden's method is formulated in the next algorithm.

Algorithm 2.59. Broyden's method

INPUT: $\mathbf{f}$ - function,
$\mathbf{p}^{(0)}$ - initial value,
h - step size for the approximation of $\mathbf{A}^{(0)}$,
$\|\cdot\|$ - vector norm,
TOL - tolerance,
$MAXIT$ - maximal iteration number,
OUTPUT: $\mathbf{p}$ - approximate root.

(computation of $\mathbf{A} = (a_{ij}) = A^{(0)}$*)*
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, n$ **do**
    $a_{ij} \leftarrow (f_i(\mathbf{p}^{(0)} + h\mathbf{e}^{(j)}) - f_i(\mathbf{p}^{(0)}))/h$
  end do
end do
$\mathbf{A} \leftarrow \mathbf{A}^{-1}$
$\mathbf{q} \leftarrow \mathbf{p}^{(0)}$
$k \leftarrow 1$ *(step size)*
while $k < MAXIT$ **do**
  $\mathbf{s} \leftarrow -\mathbf{A}\mathbf{f}(\mathbf{q})$
  $\mathbf{p} \leftarrow \mathbf{q} + \mathbf{s}$
  if $\|\mathbf{s}\| < TOL$ **do**
    output$(\mathbf{p})$
    stop
  end do
  $\mathbf{y} \leftarrow \mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{q})$
  $\mathbf{A} \leftarrow \mathbf{A} - \dfrac{(\mathbf{A}\mathbf{y} - \mathbf{s})\mathbf{s}^T\mathbf{A}}{\mathbf{s}^T\mathbf{A}\mathbf{y}}$
  $\mathbf{q} \leftarrow \mathbf{p}$
  $k \leftarrow k + 1$
end do
output(Maximal iteration is exceeded.)

Example 2.60. Consider again the system (2.26) examined in Examples 2.51 and 2.57. The numerical results of Algorithm 2.59 with $h = 0.001$ and $TOL = 10^{-5}$ is shown in Table 2.14. We observe that the convergence of this sequence is slower than that for the Newton's method in Example 2.57. The last column indicates that the speed of the convergence here is superlinear. □

Exercises

1. Apply Broyden's method to the systems listed in Exercise 1 of Section 2.11.
2. Show that the matrix $\mathbf{A}^{(k+1)}$ defined by (2.39) satisfies equations (2.37) and (2.38).

Table 2.14: Broyden's method

k	$\mathbf{p}^{(k)}$	$\|\mathbf{p}^{(k)}-\mathbf{p}\|_\infty$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_\infty}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_\infty}$
0	(-1.5000000000, -1.5000000000)^T	2.5000000000	
1	(-1.2490215360, -0.5215363883)^T	2.2490215360	0.8996086144
2	(-0.4968297655, -0.9366983828)^T	1.4968297660	0.6655471022
3	(-0.3045368940, -0.3621731989)^T	1.3045368940	0.8715332389
4	(0.5414891937, -0.0587408442)^T	0.4585108063	0.3514740046
5	(0.9527177435, -0.0515250779)^T	0.0515250779	0.1123748387
6	(1.0003263340, 0.0319681269)^T	0.0319681269	0.6204382061
7	(1.0000051000, -0.0040567750)^T	0.0040567750	0.1269006155
8	(1.0000069210, -0.0000347010)^T	0.0000347010	0.0085538489
9	(1.0000001100, 0.0000012682)^T	0.0000012682	0.0365458110
10	(1.0000000050, 0.0000000576)^T	0.0000000576	0.0453865979

Chapter 3
Linear Systems

In this chapter we discuss solution techniques of linear algebraic systems using direct methods and related problems of linear algebra. We introduce the Gaussian and Gauss-Jordan eliminations and their variants, and its application for the matrix inversion.

3.1. Review of Linear Algebra

In this section we review some notations, notions and statements of linear algebra. In the sequel, if we do not say otherwise, $\mathbf{A} = (a_{ij})$ is an $n \times n$ matrix, $\mathbf{x}$ is an n-dimensional column vector. The set of all real $n \times n$ dimensional matrices is denoted by $\mathbb{R}^{n\times n}$. Similarly, $\mathbb{C}^{n\times n}$ is the set of all $n \times n$ matrices with complex entries. The determinant of the matrix $\mathbf{A}$ is denoted by $\det(\mathbf{A})$, the $n \times n$ dimensional identity matrix is $\mathbf{I}$. The transpose of a matrix $\mathbf{A}$ or a vector $\mathbf{x}$ is denoted by $\mathbf{A}^T$ and $\mathbf{x}^T$, respectively. The diagonal matrix with elements $a_1, a_2, \ldots, a_n$ in the main diagonal is denoted by $\operatorname{diag}(a_1, a_2, \ldots, a_n)$.

The $n \times n$ matrix $\mathbf{A}^{-1}$ is called the *inverse* of the $n \times n$ matrix $\mathbf{A}$ if $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. A square matrix is *invertible* or *nonsingular* if its inverse exists. A square matrix $\mathbf{A}$ is called *singular* if it has no inverse.

The next theorem summarizes the basic properties of the determinant.

Theorem 3.1. *Let $\mathbf{A}, \mathbf{B}$ be $n \times n$ matrices. Then*

1. $\det(\mathbf{A}) = 0$ *if each element of a row (or column) in* $\mathbf{A}$ *is equal to 0;*
2. $\det(\mathbf{A}) = 0$ *if two rows (columns) of* $\mathbf{A}$ *are equal;*
3. $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$*;*
4. $\det(\mathbf{A}^T) = \det(\mathbf{A})$.
5. *If* $\mathbf{A}$ *is invertible, then* $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A})$.
6. *If* $\mathbf{B}$ *is obtained from* $\mathbf{A}$ *by multiplying one of its row (column) by a constant c, then* $\det(\mathbf{B}) = c \det(\mathbf{A})$.
7. *If* $\mathbf{B}$ *is obtained from* $\mathbf{A}$ *by swapping two rows (columns), then* $\det(\mathbf{B}) = -\det(\mathbf{A})$.
8. *If* $\mathbf{B}$ *is obtained from* $\mathbf{A}$ *by multiplying one of its row (column) by a constant c, and adding the result to another row (column), then* $\det(\mathbf{B}) = \det(\mathbf{A})$.

9. *Let* $\mathbf{A}_{ij}$ *denote the* $(n-1)\times(n-1)$ *matrix which we get from* $\mathbf{A}$ *by omitting its* i*th row and* j*th column. Then we have*

$$\det(\mathbf{A}) = \sum_{j=1}^{n}(-1)^{i+j}a_{ij}\det(\mathbf{A}_{ij}),$$

and

$$\det(\mathbf{A}) = \sum_{i=1}^{n}(-1)^{i+j}a_{ij}\det(\mathbf{A}_{ij}).$$

Theorem 3.2. *Let* $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{b}\in\mathbb{R}^n$. *The following statements are equivalent:*

1. $\det(\mathbf{A})\neq 0$,
2. *the matrix* $\mathbf{A}$ *is invertible,*
3. *the linear system* $\mathbf{Ax}=\mathbf{b}$ *has a unique solution for any vector* $\mathbf{b}$.

Theorem 3.3. *The linear system* $\mathbf{Ax}=\mathbf{0}$ *has nontrivial (nonzero) solution if and only if* $\mathbf{A}$ *is singular, i.e.,* $\det(\mathbf{A})=0$.

Theorem 3.4. *If* $\mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times n}$ *are both invertible, then* $\mathbf{AB}$ *is also invertible, and* $(\mathbf{AB})^{-1}=\mathbf{B}^{-1}\mathbf{A}^{-1}$.

The square matrix $\mathbf{A}$ is *upper (lower) triangular* if $a_{ij}=0$ for all $i>j$ $(i<j)$, i.e., all elements below (above) the main diagonal are 0.

Theorem 3.5. *For a triangular matrix* $\mathbf{A}\in\mathbb{R}^{n\times n}$ *it follows* $det(\mathbf{A})=a_{11}a_{22}\cdots a_{nn}$.

Theorem 3.6. *The product of lower (upper) triangular matrices is lower (upper) triangular. The inverse of a lower (upper) triangular matrix is lower (upper) triangular.*

A square matrix P is called *permutation matrix* if it is obtained from the identity matrix by interchanging its rows (or columns). Other words, in a permutation matrix each row and column contains exactly one 1, all the other elements are 0. The next theorem claims that the multiplication by a permutation matrix is equivalent to interchanging rows or columns of a matrix.

Theorem 3.7. *Let* $k_1,\ldots,k_n$ *be a permutation of the integers* $1,\ldots,n$, *and let* $\mathbf{P}\in\mathbb{R}^{n\times n}$ *be the permutation matrix which we get from the identity matrix by moving its 1st row to the* k_1*-th row, ..., the* n*th row to its* k_n*-th row. Let* $\mathbf{A}\in\mathbb{R}^{n\times n}$. *Then the matrix* $\mathbf{PA}$ *(*$\mathbf{AP}$*) can be obtained from* $\mathbf{A}$ *so that its 1st row (columns) is moved to the* k_1*-th row (column), ..., its* n*th row (columns) is moved to the* k_n*-th row (column).*

A square matrix $A \in \mathbb{R}^{n\times n}$ is called *row diagonally dominant* or simply *diagonally dominant* if

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|, \qquad i = 1, \dots, n.$$

Similarly, the matrix $\mathbf{A}$ is called *column diagonally dominant* if $\mathbf{A}^T$ is diagonally dominant, i.e.,

$$|a_{jj}| > \sum_{\substack{i=1\\ i\neq j}}^{n} |a_{ij}|, \qquad j = 1, \dots, n.$$

Theorem 3.8. *If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is diagonally dominant, then $\mathbf{A}$ is invertible.*

Proof. Suppose that $\mathbf{A}$ is not invertible. Then the linear system $\mathbf{Ax} = \mathbf{0}$ has a nontrivial solution $\mathbf{x} \neq \mathbf{0}$. Let k be such that $|x_k| = \max\{|x_i| \colon i = 1, \dots, n\}$. Then $x_k \neq 0$. Since $\sum_{j=1}^{n} a_{ij}x_j = 0$ for all $i = 1, \dots, n$, we get $a_{kk}x_k = -\sum_{j=1, j\neq k}^{n} a_{kj}x_j$. Then the triangle-inequality yields $|a_{kk}x_k| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}x_j|$, and so

$$|a_{kk}| \leq \sum_{\substack{j=1\\ j\neq k}}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} \leq \sum_{\substack{j=1\\ j\neq k}}^{n} |a_{kj}|,$$

which is a contradiction. □

The square matrix $\mathbf{A}$ is called *positive definite* (*negative definite*) if $\mathbf{A}$ is symmetric and $\mathbf{x}^T\mathbf{Ax} > 0$ ($\mathbf{x}^T\mathbf{Ax} < 0$, respectively) for all $\mathbf{x} \neq \mathbf{0}$. The matrix $\mathbf{A}$ is called *positive semi-definite* (*negative semi-definite*) if $\mathbf{A}$ is symmetric and $\mathbf{x}^T\mathbf{Ax} \geq 0$ ($\mathbf{x}^T\mathbf{Ax} \leq 0$, respectively) for all $\mathbf{x}$.

Theorem 3.9. *If the square matrix $\mathbf{A}$ is positive definite, then*

1. *$\mathbf{A}$ is invertible,*
2. *$a_{ii} > 0$ for $i = 1, \dots, n$.*

Theorem 3.10. *The symmetric matrix $\mathbf{A}$ is positive definite if and only if all of its upper left minors, the so-called principal minors are positive, i.e.,*

$$\det \begin{pmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} \end{pmatrix} > 0, \qquad i = 1, 2, \dots, n.$$

A square matrix $\mathbf{A}$ is *orthogonal* if $\mathbf{AA}^T = \mathbf{A}^T\mathbf{A} = \mathbf{I}$, i.e., $\mathbf{A}$ is invertible and $\mathbf{A}^{-1} = \mathbf{A}^T$.

Theorem 3.11. *The product of orthogonal matrices is orthogonal.*

The complex number $\lambda \in \mathbb{C}$ is an *eigenvalue* of the square matrix $\mathbf{A}$ if the linear system

$$\mathbf{Ax} = \lambda\mathbf{x}$$

has a nontrivial ($\mathbf{x} \neq \mathbf{0}$) solution. Its nontrivial solution $\mathbf{x}$ is called the *eigenvector* of the matrix $\mathbf{A}$ corresponding to the eigenvalue λ.

Theorem 3.12. *The $n \times n$ matrix $\mathbf{A}$ has n eigenvalues, which are solutions of the nth-degree algebraic equation*

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0,$$

the so-called characteristic equation.

Theorem 3.13. *Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of the $n \times n$ matrix $\mathbf{A}$. Then*

1. *$det(\mathbf{A}) = \lambda_1\lambda_2 \cdots \lambda_n$;*
2. *$\mathbf{A}$ is invertible if and only if $\lambda_i \neq 0$ for all $i = 1, 2, \ldots, n$;*
3. *if $\mathbf{A}$ is invertible, then the eigenvalues of $\mathbf{A}^{-1}$ are $1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_n$;*
4. *the eigenvalues of the matrix $\mathbf{A}^k$ are the numbers $\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k$.*

Theorem 3.14. *The eigenvalues of a triangular matrix $\mathbf{A}$ are the diagonal elements a_{11}, a_{22}, …, a_{nn}.*

Let $\mathbf{A}$ and $\mathbf{B}$ be square matrices of the same dimension. We say that $\mathbf{A}$ and $\mathbf{B}$ are *similar* if there exists an invertible matrix $\mathbf{P}$ such that $\mathbf{A} = \mathbf{P}^{-1}\mathbf{BP}$. We comment that then $\mathbf{B} = \mathbf{PAP}^{-1}$, so the similarity is a symmetric property. The linear map defined by $\mathbf{A} \mapsto \mathbf{P}^{-1}\mathbf{AP}$ is called *similarity transformation.*

Theorem 3.15. *Eigenvalues of similar matrices are identical.*

Proof. Let $\mathbf{A} = \mathbf{P}^{-1}\mathbf{BP}$. Then the properties of the determinant yield

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \det(\mathbf{P}^{-1}\mathbf{BP} - \lambda\mathbf{I}) = \det(\mathbf{P}^{-1})\det(\mathbf{B} - \lambda\mathbf{I})\det(\mathbf{P}) = \det(\mathbf{B} - \lambda\mathbf{I}),$$

which implies the statement. □

The number $\rho(\mathbf{A}) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathbf{A}\}$ is called the *spectral radius* of $\mathbf{A}$.

Theorem 3.16. *Let k be a positive integer, and let $\|\cdot\|$ be a matrix norm. Then*

1. *$\rho(\mathbf{A}^k) = (\rho(\mathbf{A}))^k$,*
2. *$\rho(\mathbf{A}) \leq \|\mathbf{A}\|$.*

Theorem 3.17. *For every square matrix* $\mathbf{A}$ *and a positive real* $\varepsilon > 0$ *there exists a matrix norm* $\|\cdot\|$ *such that* $\|\mathbf{A}\| \leq \rho(\mathbf{A}) + \varepsilon$.

Theorem 3.18. *For any square matrix* $\mathbf{A}$ *it follows* $\|\mathbf{A}\|_2 = \sqrt{\rho(\mathbf{A}^T\mathbf{A})}$. *If* $\mathbf{A}$ *is symmetric, then* $\|\mathbf{A}\|_2 = \rho(\mathbf{A})$.

Let $a_1, \ldots, a_n$ be complex numbers. The determinant

$$\det \begin{pmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^{n-1} \end{pmatrix} \tag{3.1}$$

is called *Vandermonde determinant.*

Theorem 3.19. *The Vandermonde determinant (3.1) is nonzero if and only if the numbers* $a_1, \ldots, a_n$ *are pairwise distinct.*

Exercises

1. Determine the possible values of the parameters α and β so that the matrix
$$\mathbf{A} = \begin{pmatrix} \alpha & 1 & 0 \\ \beta & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$
be
   (a) singular,
   (b) diagonally dominant,
   (c) symmetric,
   (d) positive definite.
2. Prove that if $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ positive definite matrices, then
   (a) $\mathbf{A}^T$,
   (b) $\mathbf{A} + \mathbf{B}$,
   (c) $\mathbf{A}^2$
   are also positive definite.
3. Prove Theorem 3.6.
4. Prove Theorem 3.7.
5. Prove Theorem 3.9.
6. Prove Theorem 3.11.
7. Prove Theorem 3.12.
8. Prove Theorem 3.14.
9. Prove Theorem 3.19. (Hint: In the determinant (3.1) substitute a_1 by x. Show that the resulting determinant is a polynomial of degree $n-1$ of x. Find $n-1$ distinct roots of this polynomial.)

10. Show that the value of the Vandermonde determinant (3.1) is

$$\prod_{i>j}(a_i - a_j).$$

(hint: Consider the proof of the previous problem.)

3.2. Triangular Systems

Example 3.20. Solve the linear system

$$\begin{array}{rcrcrcrcr} 2x_1 & - & x_2 & + & 3x_3 & + & x_4 & = & 3 \\ & & 3x_2 & - & x_3 & + & 2x_4 & = & 13 \\ & & & & 2x_3 & - & x_4 & = & -2 \\ & & & & & & 3x_4 & = & 12 \end{array}$$

Solving the fourth equation for x_4 we get $x_4 = 4$. Substituting it to the third equation we get $x_3 = (-2 + x_4)/2 = 1$. Then the second equation yields $x_2 = (13 + x_3 - 2x_4)/3 = 2$. Finally, from the first equation we have $x_1 = (3 + x_2 - 3x_3 - x_4)/2 = -1$. □

We can generalize the method used in the previous example to solve the upper triangular n-dimensional linear system $\mathbf{Ax} = \mathbf{b}$, i.e., a linear system of the form

$$\begin{array}{rcrcccrcl} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ & & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & a_{nn}x_n & = & b_n. \end{array} \tag{3.2}$$

We formulate the method of *backward substitution* in the following algorithm.

Algorithm 3.21. Backward substitution to solve a triangular system

INPUT: a_{ij}, $(i = 1, \dots, n, \quad j = 1, \dots, n)$, b_i, $(i = 1, \dots, n)$
OUTPUT: $x_1, \dots, x_n$

$x_n \leftarrow b_n/a_{nn}$
for $i = n-1, \dots, 1$ **do**
$\quad x_i \leftarrow (b_i - \sum_{j=i+1}^{n} a_{ij}x_j)/a_{ii}$
end do
output$(x_1, x_2, \dots, x_n)$

The method of backward substitution can be performed if an only if $a_{ii} \neq 0$ for all $i = 1, \ldots, n$. Since $\det(\mathbf{A}) = a_{11}a_{22}\cdots a_{nn}$, it follows that it can be performed if and only if the system (3.2) has a unique solution, i.e., $\det(\mathbf{A}) \neq 0$.

In order to determine the time complexity of the algorithm we count the number of arithmetic operations:

	multiplication/division	addition/subtraction
step 1:	1	0
step 2:	2	1
$\vdots$	$\vdots$	$\vdots$
step n:	n	$n-1$

Therefore, $1+2+\cdots+n = n(n+1)/2$ multiplications and divisions, and $1+2+\cdots+n-1 = (n-1)n/2$ additions and subtractions are needed to perform the algorithm. We introduce the notation $\mathcal{O}(n^k)$ for a polynomial of order at most k. With this notation we have that the number of multiplications/divisions is $n^2/2 + \mathcal{O}(n)$, and similarly, the number of additions/subtractions are needed for the algorithm is $n^2/2 + \mathcal{O}(n)$. This notation "hides" the lower order terms, which is useful, since the leading term determines the magnitude of the formula for large n.

Exercises

1. Solve the following triangular systems:

(a)

$$\begin{aligned} 3x_1 + x_2 - x_3 + 2x_4 &= -4 \\ 4x_2 - 2x_3 + x_4 &= 5 \\ 6x_3 - 2x_4 &= -7 \\ 2x_4 &= 4 \end{aligned}$$

(b)

$$\begin{aligned} 1.2x_1 + 2.1x_2 - 3.2x_3 + 2.0x_4 + 1.4x_5 &= 81.5 \\ 2.5x_2 - 1.1x_3 + 6.1x_4 - 3.0x_5 &= 159.7 \\ 2.6x_3 - 1.1x_4 &= 12.8 \\ 2.2x_4 + 4.1x_5 &= 46.9 \\ 1.3x_5 &= 6.5 \end{aligned}$$

3.3. Gaussian Elimination, Pivoting Strategies

Example 3.22. Consider the linear system

$$\begin{aligned} x_1 - 2x_2 - 2x_3 - 2x_4 &= -11 \\ 2x_1 - x_2 + 2x_3 + 4x_4 &= -8 \\ -x_1 + 2x_2 + 3x_3 - 4x_4 &= 27 \\ 2x_1 + x_2 + 4x_3 - 2x_4 &= 28 \end{aligned} \tag{3.3}$$

With the help of the first equation, the variable x_1 can be eliminated from the second, third and fourth equations. We multiply the first equation by 2, -1 and 2, respectively, and subtract it from the second, third and fourth equations, respectively:

$$\begin{aligned} x_1 - 2x_2 - 2x_3 - 2x_4 &= -11 \\ 3x_2 + 6x_3 + 8x_4 &= 14 \\ x_3 - 6x_4 &= 16 \\ -3x_2 - 6x_4 &= 6 \end{aligned} \tag{3.4}$$

The resulting system is equivalent to (3.3).

We associate the 4×5 dimensional matrix

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 2 & -1 & 2 & 4 & -8 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \end{pmatrix} \tag{3.5}$$

to the system (3.3). Here we augmented the 4×4 coefficient matrix with a fifth column which contains the elements from the right hand side of the system. We will call this matrix as the *augmented matrix*. In the augmented matrix we can do the above elimination by multiplying the first row by 2, -1 and 2, respectively, and we subtract it from the second, third and fourth row, respectively. Then we get

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & -3 & 0 & -6 & 6. \end{pmatrix}. \tag{3.6}$$

The variable x_2 is missing in the equation representing the third row, and we eliminate x_2 from the fourth row too with the help of the second row. We multiply the second row by -1, and subtract the result from the fourth row:

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & 0 & 6 & 2 & 20 \end{pmatrix}. \tag{3.7}$$

Finally, we multiply the third row by 6, and subtract it from the third row:

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & 0 & 0 & 38 & -76 \end{pmatrix}. \tag{3.8}$$

This augmented matrix describes the triangular system

$$\begin{array}{rcrcrcrcr} x_1 & - & 2x_2 & - & 2x_3 & - & 2x_4 & = & -11 \\ & & 3x_2 & + & 6x_3 & + & 8x_4 & = & 14 \\ & & & & x_3 & - & 6x_4 & = & 16 \\ & & & & & & 38x_4 & = & -76 \end{array}$$

Solving it with the backward substitution we get $x_1 = -3$, $x_2 = 2$, $x_3 = 4$ and $x_4 = -2$. The above elimination process is written shortly as

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 2 & -1 & 2 & 4 & -8 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \end{pmatrix} \sim \begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & -3 & 0 & -6 & 6. \end{pmatrix} \sim$$
$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & 0 & 6 & 2 & 20 \end{pmatrix} \sim \begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & 0 & 0 & 38 & -76 \end{pmatrix}.$$

□

Using the above method for the general n-dimensional linear system

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{array} \tag{3.9}$$

we get the *Gaussian elimination with backward substitution*. We put the coefficients and the right hand sides to the *augmented matrix*:

$$\tilde{\mathbf{A}}^{(0)} = (\mathbf{A}, \mathbf{b}) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{n,n+1} \end{pmatrix},$$

where $a_{i,n+1} := b_i$, $(i = 1, \dots, n)$. Starting from the matrix $\tilde{\mathbf{A}}^{(0)}$ we obtain the sequence of matrices $\tilde{\mathbf{A}}^{(1)}$, $\tilde{\mathbf{A}}^{(2)}$, …, $\tilde{\mathbf{A}}^{(n-1)}$ describing equivalent linear systems in the following way. Let

$$\tilde{\mathbf{A}}^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & a_{2,n+1}^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & a_{n,n+1}^{(1)} \end{pmatrix},$$

where $a_{ij}^{(1)} := a_{ij} - l_{i1}a_{1j}$, $\quad l_{i1} := \frac{a_{i1}}{a_{11}}$, $\quad i = 2, \dots, n$, $\quad j = 2, \dots, n+1$, (assuming $a_{11} \neq 0$). If the matrices $\tilde{\mathbf{A}}^{(1)}, \dots, \tilde{\mathbf{A}}^{(k-1)}$ are defined for some $k \leq n-1$, then let

$$\tilde{\mathbf{A}}^{(k)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,k} & a_{1,k+1} & \cdots & a_{1,n} & a_{1,n+1} \\ 0 & a_{22}^{(1)} & \cdots & a_{2,k}^{(1)} & a_{2,k+1}^{(1)} & \cdots & a_{2,n}^{(1)} & a_{2,n+1}^{(1)} \\ & & \ddots & & & & & \\ 0 & 0 & \cdots & a_{k,k}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & a_{k,n+1}^{(k-1)} \\ 0 & 0 & \cdots & 0 & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} & a_{k+1,n+1}^{(k)} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n,k+1}^{(k)} & \cdots & a_{n,n}^{(k)} & a_{n,n+1}^{(k)} \end{pmatrix},$$

where $a_{ij}^{(k)} := a_{ij}^{(k-1)} - l_{ik}a_{kj}^{(k-1)}$, $\quad l_{ik} := \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$, $i = k+1, \dots, n$, $\quad j = k+1, \dots, n+1$. We perform these *elimination steps* for $k = 1, \dots, n-1$. Finally, we solve the triangular system corresponding to the matrix $\tilde{\mathbf{A}}^{(n-1)}$ using the backward substitution method. The elements a_{11}, $a_{22}^{(1)}$, …, $a_{nn}^{(n-1)}$ in the main diagonal of the last matrix of the Gaussian elimination are called *pivot elements*. Clearly, we can perform the Gaussian elimination if and only if all the pivot elements are nonzero.

If we perform the steps of the Gaussian elimination only on the coefficient matrix, the resulting matrices will be denoted by $\mathbf{A}^{(0)} := \mathbf{A}$, $\mathbf{A}^{(1)}$, …, $\mathbf{A}^{(n-1)}$.

Algorithm 3.23. Gaussian elimination

INPUT: a_{ij}, $\quad (i = 1, \dots, n, \quad j = 1, \dots, n+1)$ - augmented matrix
OUTPUT: x_1, …, x_n

(elimination:)
for $k = 1, \dots, n-1$ **do**
  for $i = k+1, \dots, n$ **do**
    $l_{ik} \leftarrow a_{ik}/a_{kk}$

```
        for j = k+1, ..., n+1 do
            a_ij ← a_ij − l_ik a_kj
        end do
    end do
end do
(backward substitution:)
x_n ← a_{n,n+1}/a_nn
for i = n−1, ..., 1 do
    x_i ← (a_{i,n+1} − Σ_{j=i+1}^{n} a_ij x_j)/a_ii
end do
output(x_1, x_2, ..., x_n)
```

The above algorithm is formulated so that in each step the new value of an element overwrites the same element of the previous matrix. We note that the zeros in the matrix are not computed and even they are not stored. Therefore, after the last elimination steps the elements under the main diagonal have no meaning. They can be filled by zero directly if the whole matrix is needed.

Next we compute the number of arithmetic operations of the Gaussian elimination:

	multiplication/division	addition/subtraction
step 1	$(n-1)(n+1)$	$(n-1)n$
step 2	$(n-2)n$	$(n-2)(n-1)$
$\vdots$	$\vdots$	$\vdots$
step $n-1$	$1\cdot 3$	$1\cdot 2$
total:	$\sum_{i=1}^{n-1} i(i+2)$	$\sum_{i=1}^{n-1} i(i+1)$

Using the identity $1^2+2^2+\cdots+n^2=\frac{1}{6}n(n+1)(2n+1)$ we can easily check that the total number of multiplications and divisions needed for the elimination steps is $n^3/3+n^2/2-5n/6$, and the number of additions and subtractions is $(n^3-n)/3$. Together with the backward substitution, $n^3/3+n^2/2-5n/6+n^2/2+n/2=n^3/3+n^2-n/3=n^3/3+\mathcal{O}(n^2)$ number of multiplications and divisions, and $(n^3-n)/3+n^2/2-n/2=n^3/3+n^2/2-5n/6=n^3/3+\mathcal{O}(n^2)$ number of additions and subtractions are needed to perform the Gaussian elimination. Shortly we say that the time complexity of the Gaussian elimination is $n^3/3$ number of operations.

Example 3.24. Solve the system

$$\begin{array}{rcrcrcrcr} 2x_1 & - & x_2 & & & - & 3x_4 & = & 8 \\ 2x_1 & - & x_2 & + & x_3 & + & 5x_4 & = & 2 \\ -3x_1 & + & x_2 & + & x_3 & - & 2x_4 & = & -5 \\ 2x_1 & + & 4x_2 & & & - & x_4 & = & 21 \end{array}$$

by Gaussian elimination. After performing the first step of the elimination we get

$$\begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 2 & -1 & 1 & 5 & 2 \\ -3 & 1 & 1 & -2 & -5 \\ 2 & 4 & 0 & -1 & 21 \end{pmatrix} \sim \begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 0 & 0 & 1 & 8 & -6 \\ 0 & -1/2 & 1 & -13/2 & 7 \\ 0 & 5 & 0 & 2 & 13 \end{pmatrix}.$$

Since the pivot element of the second row is 0, the Algorithm 3.23 cannot be continued. On the other hand, the system has a unique solution: $x_1 = 4$, $x_2 = 3$, $x_3 = 2$ and $x_4 = -1$. But if we change the second and third rows of the previous augmented matrix, the corresponding linear system is the same, and the elimination can be continued:

$$\begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 0 & 0 & 1 & 8 & -6 \\ 0 & -1/2 & 1 & -13/2 & 7 \\ 0 & 5 & 0 & 2 & 13 \end{pmatrix} \sim \begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 0 & -1/2 & 1 & -13/2 & 7 \\ 0 & 0 & 1 & 8 & -6 \\ 0 & 5 & 0 & 2 & 13 \end{pmatrix} \sim$$
$$\begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 0 & -1/2 & 1 & -13/2 & 7 \\ 0 & 0 & 1 & 8 & -6 \\ 0 & 0 & 10 & -63 & 83 \end{pmatrix} \sim \begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 0 & -1/2 & 1 & -13/2 & 7 \\ 0 & 0 & 1 & 8 & -6 \\ 0 & 0 & 0 & -143 & 143 \end{pmatrix},$$

which yields the solution. □

Example 3.25. Solve the linear system

$$\begin{array}{rcrcr} 0.0002x_1 & - & 30.5x_2 & = & -60.99 \\ 5.060x_1 & - & 1.05x_2 & = & 250.9 \end{array}$$

using Gaussian elimination and 4-digit arithmetic. Following Algorithm 3.23, first we compute the factor $l_{21} = 5.060/0.0002 = 25300$ (rounding to 4 significant digits). Then by multiplying the first equation by l_{21} and subtracting it from the second row we get

$$\begin{pmatrix} 0.0002 & -30.5 & -60.99 \\ 5.06 & -1.05 & 250.9 \end{pmatrix} \sim \begin{pmatrix} 0.0002 & -30.5 & -60.99 \\ 0 & 771700 & 1543000 \end{pmatrix}.$$

We note that we do not compute the first element of the second row by Algorithm 3.23, it will be 0 without any calculation.) Solving it we get the numerical solutions $\tilde{x}_1 = -100.0$ and $\tilde{x}_2 = 1.999$. We can check that the exact solution of the system is $x_1 = 50$ and $x_2 = 2$. Therefore, the relative errors of the numerical solutions are 300% and 0.05%, respectively. Note that the relative error of the first variable is huge.

Repeat the calculation for the system where we interchange the two equations:

$$\begin{pmatrix} 5.06 & -1.05 & 250.9 \\ 0.0002 & -30.5 & -60.99 \end{pmatrix} \sim \begin{pmatrix} 5.06 & -1.05 & 250.9 \\ 0 & -30.5 & -61.0 \end{pmatrix}.$$

This gives the numerical values $x_1 = 50.00$ and $x_2 = 2.000$, which are identical to the exact solutions.

What is the difference in between the two computations? In the first case, in order to compute l_{21} we needed to divide by a small number (0.0002), which gave us the increase of the rounding error. In the second case we performed the division by 5.06 in the computation of l_{21}, and we did not observe any error in the final result. □

Partial Pivoting

The last two examples show that sometimes it is necessary, and in many cases it is useful to modify Algorithm 3.23. One of the most popular modification is the Gaussian elimination with *partial pivoting* (or *maximal column pivoting*). Here, before the kth step of the elimination, we select the element with the largest magnitude in the kth column in and under the main diagonal, i.e., let

$$|a_{lk}| = \max\{|a_{ik}|\colon\ i = k, \dots, n\}.$$

(An element with the largest magnitude is in the lth row. If there are several elements with the same largest magnitude, then l denotes the first possible row index.) We interchange the kth and lth rows, and then continue with the elimination. This will get around the problems of Examples 3.24 and 3.25. Indeed, if $a_{kk}^{(k-1)} = 0$, then after the row change a nonzero element is moved into this position (if there is a nonzero element below $a_{kk}^{(k-1)}$). Furthermore, the row change guarantees that the division will be performed by the element with a largest magnitude which helps to reduce the rounding error in the computation.

Theorem 3.26. *The next statements are equivalent:*

(i) *the linear system* $\mathbf{Ax} = \mathbf{b}$ *can be solved by Gaussian elimination with partial pivoting,*

(ii) $\det(\mathbf{A}) \neq 0$*,*

(iii) *the matrix* $\mathbf{A}$ *is invertible,*

(iv) *the linear system* $\mathbf{Ax} = \mathbf{b}$ *has a unique solution for all* $\mathbf{b}$*.*

Proof. It is known from linear algebra that statements (ii), (iii) and (iv) are equivalent (see Theorem 3.2). Now we show that (i) and (ii) are equivalent.

Suppose first that (i) holds. Let $\mathbf{A}^{(0)} := \mathbf{A}$, and let $\mathbf{A}^{(k)}$ be the coefficient matrix in the Gaussian elimination after the kth step. The properties of the determinants yield that $\det(\mathbf{A}^{(k)}) = \det(\mathbf{A}^{(k-1)})$ if there was no row change in the kth step, and $\det(\mathbf{A}^{(k)}) = -\det(\mathbf{A}^{(k-1)})$ if there was a row change. Since the Gaussian elimination can be performed by the assumption, the triangular system corresponding to the coefficient matrix $\mathbf{A}^{(n-1)}$ of the last step is solvable, therefore, $\det(\mathbf{A}^{(n-1)}) \neq 0$. But this implies $\det(\mathbf{A}) = \pm\det(\mathbf{A}^{(n-1)}) \neq 0$.

We show that if the Gaussian elimination with partial pivoting terminates before the kth step, then $\det(\mathbf{A}) = 0$. The kth step cannot be performed if and only if $a_{ik}^{(k-1)} = 0$ for all $i = k, \dots, n$, i.e.,

$$\mathbf{A}^{(k-1)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,k-1} & a_{1k} & a_{k,k+1} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2,k-1}^{(1)} & a_{2k}^{(1)} & a_{2,k+1}^{(1)} & \cdots & a_{2n}^{(1)} \\ & & \ddots & & & & & \\ 0 & 0 & \cdots & a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,k+1}^{(k-2)} & \cdots & a_{k-1,n}^{(k-2)} \\ 0 & 0 & \cdots & 0 & 0 & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & a_{n,k+1}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{pmatrix}.$$

Hence

$$\det(\mathbf{A}^{(k-1)}) = a_{11}a_{22}^{(1)}\cdots a_{k-1,k-1}^{(k-2)} \det\begin{pmatrix} 0 & a_{k,k+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n,k+1}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{pmatrix} = 0,$$

and so $\det(\mathbf{A}) = \pm\det(\mathbf{A}^{(k-1)}) = 0$. □

Example 3.27. Consider again the system examined in Example 3.24, and solve it using Gaussian elimination with partial pivoting. We get the following sequence of the augmented matrices:

$$\begin{pmatrix} 2 & -1 & 0 & -3 & 8 \\ 2 & -1 & 1 & 5 & 2 \\ -3 & 1 & 1 & -2 & -5 \\ 2 & 4 & 0 & -1 & 21 \end{pmatrix} \sim \begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 2 & -1 & 1 & 5 & 2 \\ 2 & -1 & 0 & -3 & 8 \\ 2 & 4 & 0 & -1 & 21 \end{pmatrix} \sim$$

$$\begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 0 & -1/3 & 5/3 & 11/3 & -4/3 \\ 0 & -1/3 & 2/3 & -13/3 & 14/3 \\ 0 & 14/3 & 2/3 & -7/3 & 53/3 \end{pmatrix} \sim \begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 0 & 14/3 & 2/3 & -7/3 & 53/3 \\ 0 & -1/3 & 2/3 & -13/3 & 14/3 \\ 0 & -1/3 & 5/3 & 11/3 & -4/3 \end{pmatrix} \sim$$

$$\begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 0 & 14/3 & 2/3 & -7/3 & 53/3 \\ 0 & 0 & 5/7 & -9/2 & 83/14 \\ 0 & 0 & 12/7 & 7/2 & -1/14 \end{pmatrix} \sim \begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 0 & 14/3 & 2/3 & -7/3 & 53/3 \\ 0 & 0 & 12/7 & 7/2 & -1/14 \\ 0 & 0 & 5/7 & -9/2 & 83/14 \end{pmatrix} \sim$$

$$\begin{pmatrix} -3 & 1 & 1 & -2 & -5 \\ 0 & 14/3 & 2/3 & -7/3 & 53/3 \\ 0 & 0 & 12/7 & 7/2 & -1/14 \\ 0 & 0 & 0 & -143/24 & 143/24 \end{pmatrix}$$

We can observe that there was a row change before the first and third elimination steps. The solution of the triangular system is $x_1 = 4$, $x_2 = 3$, $x_3 = 2$ and $x_4 = -1$. □

Suppose we perform the Gaussian elimination with partial pivoting on the coefficient matrix $\mathbf{A}$, and we collect the row changes performed during this algorithm. It is easy to see that if we perform all these row changes first on the matrix $\mathbf{A}$ without the elimination steps, then the Gaussian elimination can be performed on this matrix, and the numerical result will be the same as for the Gaussian elimination with partial pivoting performed for the original system. According to Theorem 3.7, the row change can be performed by multiplying the matrix $\mathbf{A}$ by a permutation matrix $\mathbf{P}$ from the left. Therefore, Theorem 3.26 has the following consequence.

Theorem 3.28. *If* $\det(\mathbf{A}) \neq 0$*, then there exists a permutation matrix* $\mathbf{P}$ *such that the linear system* $\mathbf{PAx} = \mathbf{Pb}$ *can be solved by Gaussian elimination (without row changes) for all vector* $\mathbf{b}$*.*

Complete Pivoting

To further reduce the effect of rounding we can use the following modification of the partial pivoting, which is called *complete pivoting* or *maximal pivoting*: before the kth step of the elimination we find the first row index l and column index m such that

$$|a_{lm}| = \max\{|a_{ij}|\colon\ i = k, \dots, n,\ j = k, \dots, n\}.$$

(That is the element with largest magnitude is located in the lth row and in the mth column.) Then we interchange the kth and lth rows and the kth and mth columns. We have to note that the first n columns of the augmented matrix of the system contains coefficients of the variables. At the beginning of the algorithm the first column contains the coefficients of x_1, the second one contains those of x_2, and so on, the nth column contains the coefficients of x_n. Therefore, when we interchange columns, we have to record the changes in the order of the variables too. Then we continue with the elimination step, as in the Gaussian elimination.

The disadvantage of this method is that it requires more comparisons than the partial pivoting, so it slows down the running of the algorithm.

Example 3.29. Consider again the system examined in Example 3.22, and here we solve it using Gaussian elimination with complete pivoting:

$$\begin{pmatrix} 1 & -2 & -2 & -2 & -11 \\ 2 & -1 & 2 & 4 & -8 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \\ \mathtt{x}_1 & \mathtt{x}_2 & \mathtt{x}_3 & \mathtt{x}_4 & \end{pmatrix} \sim \begin{pmatrix} 2 & -1 & 2 & 4 & -8 \\ 1 & -2 & -2 & -2 & -11 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \\ \mathtt{x}_1 & \mathtt{x}_2 & \mathtt{x}_3 & \mathtt{x}_4 & \end{pmatrix} \sim$$

$$\begin{pmatrix} 4 & -1 & 2 & 2 & -8 \\ -2 & -2 & -2 & 1 & -11 \\ -4 & 2 & 3 & -1 & 27 \\ -2 & 1 & 4 & -2 & 28 \\ \mathtt{x}_4 & \mathtt{x}_2 & \mathtt{x}_3 & \mathtt{x}_1 & \end{pmatrix} \sim \begin{pmatrix} 4 & -1 & 2 & 2 & -8 \\ 0 & -5/2 & -1 & 2 & -15 \\ 0 & 1 & 5 & 1 & 19 \\ 0 & 1/2 & 5 & -1 & 24 \\ \mathtt{x}_4 & \mathtt{x}_2 & \mathtt{x}_3 & \mathtt{x}_1 & \end{pmatrix} \sim$$

$$\begin{pmatrix} 4 & -1 & 2 & 2 & -8 \\ 0 & 1 & 5 & 1 & 19 \\ 0 & -5/2 & -1 & 2 & -15 \\ 0 & 1/2 & 5 & -1 & 24 \\ \mathtt{x}_4 & \mathtt{x}_2 & \mathtt{x}_3 & \mathtt{x}_1 & \end{pmatrix} \sim \begin{pmatrix} 4 & 2 & -1 & 2 & -8 \\ 0 & 5 & 1 & 1 & 19 \\ 0 & -1 & -5/2 & 2 & -15 \\ 0 & 5 & 1/2 & -1 & 24 \\ \mathtt{x}_4 & \mathtt{x}_3 & \mathtt{x}_2 & \mathtt{x}_1 & \end{pmatrix} \sim$$

$$\begin{pmatrix} 4 & 2 & -1 & 2 & -8 \\ 0 & 5 & 1 & 1 & 19 \\ 0 & 0 & -23/10 & 11/5 & -56/5 \\ 0 & 0 & -1/2 & -2 & 5 \\ \mathtt{x}_4 & \mathtt{x}_3 & \mathtt{x}_2 & \mathtt{x}_1 & \end{pmatrix} \sim \begin{pmatrix} 4 & 2 & -1 & 2 & -8 \\ 0 & 5 & 1 & 1 & 19 \\ 0 & 0 & -23/10 & 11/5 & -56/5 \\ 0 & 0 & 0 & -57/23 & 171/23 \\ \mathtt{x}_4 & \mathtt{x}_3 & \mathtt{x}_2 & \mathtt{x}_1 & \end{pmatrix}$$

In order to follow the effect of the column changes, we augmented the matrices with an extra row where we record the variable whose coefficients are listed in that particular column. Here before the first elimination step, we interchanged the first and fourth columns, since 4 was the element with the largest magnitude in the coefficients. (Another option would be to interchange the first and third rows and then the first and fourth columns; or to interchange the first and fourth rows and the first and third columns.) Before the second elimination step, we interchanged the second and third rows and the second and third columns. And before the third elimination step there were no row or column changes. Finally, we solved the triangular system. The fourth equation gave us the value of the variable x_1, and the third equation can be solved for x_2, the

second equation implied the value of x_3, and finally, from the first equation we got the solution for x_4. The result is again $x_1 = -3$, $x_2 = 2$, $x_3 = 4$ and $x_4 = -2$.

We comment that the advantage of the partial and complete pivoting appears when we do the computations using floating point arithmetic. □

Scaled Partial Pivoting

Numerical observations indicate that if the order of magnitude of the elements in the coefficient matrix is significantly different, then the effect of rounding can be large (see Example 3.25). Therefore, it is usual to multiply the rows of the system with a nonzero real to equalize the magnitude of the coefficients. If we combine it with the partial pivoting, we get a technique called *scaled partial pivoting*: We are looking for positive factors $d_1, \dots, d_n > 0$ so that the elements of the matrix $\mathbf{B} := \mathbf{DA}$ be of the same magnitude, where $\mathbf{D} = \operatorname{diag}(d_1, \dots, d_n)$. Then, instead of solving the linear system $\mathbf{Ax} = \mathbf{b}$, we solve the equivalent linear system $\mathbf{DAx} = \mathbf{Db}$ numerically. One simple strategy is to select $\mathbf{D}$ so that $\max\{|b_{ij}| : 1 \le j \le n\} \approx 1$ be satisfied for all $i = 1, \dots, n$. We can define $d_i := 1/s_i$ where $s_i := \max\{|a_{ij}| : 1 \le j \le n\}$. The problem here is that the division may introduce further rounding error in the calculation. To avoid it, let β be the base of the number representation on the computer, and let r_i be the smallest integer so that $\beta^{r_i} \ge s_i$, and define $b_{ij} := a_{ij}/\beta^{r_i}$ $(i, j = 1, \dots, n)$. Then the division will not contain rounding error, and $1/\beta < \max_{1 \le j \le n} |b_{ij}| \le 1$ holds for all $i = 1, \dots, n$.

The following result can be proved.

Theorem 3.30. *Suppose we perform a scaled partial pivoting on the coefficient matrix* $\mathbf{A}$ *with the matrix* $\mathbf{D} = \operatorname{diag}(d_1, \dots, d_n)$ *which do not introduce rounding errors (e.g., using* β *powers). Then if partial or complete pivoting on the matrix* $\mathbf{DA}$ *yields the same row (and column) changes as the same pivoting on the matrix* $\mathbf{A}$*, then the numerical solutions of the systems* $\mathbf{Ax} = \mathbf{b}$ *and* $\mathbf{DAx} = \mathbf{Db}$ *with Gaussian elimination using pivoting will be identical.*

The previous result shows that the scaling of the equations effects only the selection of the pivot elements, not the numerical result. So it is popular to use the scaling to select the pivot elements, but we do not perform the the scaling of the rows. This variant of the scaled pivoting is called *partial pivoting with implicit scaling*. The result is one of the most popular algorithms to solve linear systems.

Algorithm 3.31. Gaussian elimination with partial pivoting and implicit scaling

INPUT: a_{ij}, $(i = 1, \dots, n, \quad j = 1, \dots, n+1)$ - augmented matrix
OUTPUT: $x_1, \dots, x_n$

(computation of the scale factors:)
for $i = 1, \dots, n$ **do**

$s_i \leftarrow \max\limits_{1\le j\le n} |a_{ij}|$
end do
(elimination:)
for $k = 1, \dots, n-1$ **do**
let l be the smallest row index for which $\dfrac{|a_{lk}|}{s_l} = \max\limits_{k\le i\le n} \dfrac{|a_{ik}|}{s_i}$
interchange the kth and lth rows of the matrix $\mathbf{A}$
for $i = k+1, \dots, n$ **do**
$l_{ik} \leftarrow a_{ik}/a_{kk}$
for $j = k+1, \dots, n+1$ **do**
$a_{ij} \leftarrow a_{ij} - l_{ik}a_{kj}$
end do
end do
end do
(backward substitution:)
$x_n \leftarrow a_{n,n+1}/a_{nn}$
for $i = n-1, \dots, 1$ **do**
$x_i \leftarrow (a_{i,n+1} - \sum\limits_{j=i+1}^{n} a_{ij}x_j)/a_{ii}$
end do
output$(x_1, x_2, \dots, x_n)$

We note that in our methods many times we needed to interchange two rows of a matrix $\mathbf{A} = (a_{ij})$. This requires a lot of operation, therefore, instead of it we can do the following trick in programming: We store the elements of the matrix in a two-dimensional array $a[i, j]$. We define an array $m[i]$ with initial values $m[i] = i$, $(i = 1, \dots, n)$. If we interchange the kth and lth rows, we swap the kth and lth elements of the array $m[\cdot]$. When we have to refer to an element a_{ij} of the matrix $\mathbf{A}$, we can use the value $a[m[i], j]$.

Theorem 3.32. *If the matrix $\mathbf{A}$ is diagonally dominant, then the Gaussian elimination can be performed on the linear system $\mathbf{Ax} = \mathbf{b}$ without pivoting, and the method is stable with respect to the rounding errors.*

Proof. First we note that if the matrix $\mathbf{A}$ is diagonally dominant, then Theorem 3.8 implies that the linear system $\mathbf{Ax} = \mathbf{b}$ has a unique solution.

We show that each of the coefficient matrices $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, ..., $\mathbf{A}^{(n-1)}$ of the elimination steps is also diagonally dominant. Since $\mathbf{A}^{(0)} = \mathbf{A}$ is diagonally dominant, it follows $|a_{11}| > \sum_{j=2}^{n} |a_{1j}|$, and hence $a_{11} \neq 0$. Therefore, the matrix $\mathbf{A}^{(1)}$ is well-defined. We show that $\mathbf{A}^{(1)}$ is diagonally dominant. Since the first row of $\mathbf{A}^{(1)}$ is identical to that of $\mathbf{A}$, it is diagonally dominant. Let $1 < i \le n$. Using $a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}}a_{1j}$, $(j = 2, \dots, n)$, and

$a_{i1}^{(1)} = 0$, we get

$$\sum_{\substack{j=2\\ j\neq i}}^{n} |a_{ij}^{(1)}| = \sum_{\substack{j=2\\ j\neq i}}^{n} |a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}| \leq \sum_{\substack{j=2\\ j\neq i}}^{n} (|a_{ij}| + \frac{|a_{i1}|}{|a_{11}|}|a_{1j}|) = \sum_{\substack{j=2\\ j\neq i}}^{n} |a_{ij}| + \frac{|a_{i1}|}{|a_{11}|} \sum_{\substack{j=2\\ j\neq i}}^{n} |a_{1j}|.$$

Since the ith row of $\mathbf{A}$ is also diagonally dominant, it follows

$$\begin{aligned}
\sum_{\substack{j=2\\ j\neq i}}^{n} |a_{ij}^{(1)}| &< |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|}(|a_{11}| - |a_{1i}|) \\
&= |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|}|a_{1i}| \\
&\leq \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \\
&= |a_{ii}^{(1)}|.
\end{aligned}$$

This shows that all the rows of $\mathbf{A}^{(1)}$ are diagonally dominant, hence the matrix is diagonally dominant.

Similar argument shows that all matrices $\mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(n-1)}$ are diagonally dominant. The numerical stability is not shown here. □

We present the next result without its proof.

Theorem 3.33. *Let $\mathbf{A}$ be a symmetric $n \times n$ matrix, $\mathbf{b} \in \mathbb{R}^n$. Then $\mathbf{A}$ is positive definite if and only if the Gaussian elimination can be performed for the system $\mathbf{Ax} = \mathbf{b}$ without pivoting, and the pivot elements are all positive. Moreover, in this case the method is stable with respect to the rounding errors.*

Exercises

1. Solve the following linear systems using Gaussian elimination

   (i) without pivoting,

   (ii) with partial pivoting,

   (iii) with complete pivoting,

   (iv) with scaled partial pivoting:

   (a)
   $$\begin{array}{rcrcrcr} 2x_1 & + & 2x_2 & - & 2x_3 & = & -4 \\ -x_1 & + & 3x_2 & & & = & -11 \\ 4x_1 & + & 2x_2 & - & 3x_3 & = & -1 \end{array}$$

   (b)
   $$\begin{array}{rcrcrcrcr} -x_1 & - & 3x_2 & & & + & 2x_4 & = & 10 \\ -2x_1 & + & 3x_2 & & & + & x_4 & = & 8 \\ 4x_1 & + & x_2 & - & x_3 & - & 3x_4 & = & -21 \\ 2x_1 & + & x_2 & - & x_3 & + & 3x_4 & = & 7 \end{array}$$

2. Use 4-digit arithmetic in the calculations, and apply the question of the previous exercise for the following systems:

   (a)
   $$\begin{array}{rcrcrcr} 1.03x_1 & - & 1.1x_2 & + & 8x_3 & = & -9.06 \\ -4.1x_1 & + & 10.1x_2 & - & 6x_3 & = & 106.2 \\ 2.11x_1 & - & 4.2x_2 & + & 12x_3 & = & -40.22 \end{array}$$
   (exact solution: (-2, 10, 0.5)),

   (b)
   $$\begin{array}{rcrcrcr} x_1 & + & \frac{1}{2}x_2 & + & \frac{1}{3}x_3 & = & 20 \\ \frac{1}{2}x_1 & + & \frac{1}{3}x_2 & + & \frac{1}{4}x_3 & = & 14 \\ \frac{1}{3}x_1 & + & \frac{1}{4}x_2 & + & \frac{1}{5}x_3 & = & 11 \end{array}$$
   (exact solution: (6, -12, 60))

3. Prove Theorem 3.30.
4. Prove Theorem 3.33 (except the statement related to the stability).

3.4. Gauss–Jordan Elimination

A version of the Gaussian elimination is the *Gauss–Jordan elimination*, where we use the elimination steps of the Gaussian elimination to transform the coefficient matrix part of the augmented matrix to the identity matrix, i.e., the matrix $(\mathbf{A}, \mathbf{b})$ is converted to the form $(\mathbf{I}, \mathbf{b}^{(n-1)})$. Then the solution of the linear system is $\mathbf{x} = \mathbf{b}^{(n-1)}$.

Algorithm 3.34. Gauss–Jordan elimination

INPUT: a_{ij}, $(i = 1, \ldots, n, \quad j = 1, \ldots, n+1)$ - augmented coefficient matrix
OUTPUT: x_1, ..., x_n

(converting the coefficients to a diagonal form:)
for $k = 1, \ldots, n$ **do**
  for $i = 1, \ldots, n$ **do**
    if $i \neq k$ **do**
      $l_{ik} \leftarrow a_{ik}/a_{kk}$
      for $j = k+1, \ldots, n+1$ **do**
        $a_{ij} \leftarrow a_{ij} - l_{ik}a_{kj}$
      end do
    end do
  end do
end do
for $i = 1, \ldots, n$ **do**
  $x_i \leftarrow a_{i,n+1}/a_{ii}$
end do
output$(x_1, x_2, \ldots, x_n)$

It can be checked that the operation count of the Gauss-Jordan elimination is $n^3/2 + \mathcal{O}(n^2)$ number of multiplications and divisions and $n^3/2 + \mathcal{O}(n^2)$ number of additions and subtractions.

Example 3.35. We apply the Gauss–Jordan elimination to the linear system examined in Example 3.22:

$$\left(\begin{array}{rrrrr} 1 & -2 & -2 & -2 & -11 \\ 2 & -1 & 2 & 4 & -8 \\ -1 & 2 & -3 & -4 & 3 \\ -2 & 1 & 4 & -2 & 28 \end{array}\right) \sim \left(\begin{array}{rrrrr} 1 & -2 & -2 & -2 & -11 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & -5 & -6 & -8 \\ 0 & -3 & 0 & -6 & 6 \end{array}\right) \sim$$
$$\left(\begin{array}{rrrrr} 1 & 0 & 2 & 10/3 & -5/3 \\ 0 & 3 & 6 & 8 & 14 \\ 0 & 0 & -5 & -6 & -8 \\ 0 & 0 & 6 & 2 & 20 \end{array}\right) \sim \left(\begin{array}{rrrrr} 1 & 0 & 0 & 14/15 & -73/15 \\ 0 & 3 & 0 & 4/5 & 22/5 \\ 0 & 0 & -5 & -6 & -8 \\ 0 & 0 & 0 & -26/5 & 52/5 \end{array}\right) \sim$$
$$\left(\begin{array}{rrrrr} 1 & 0 & 0 & 0 & -3 \\ 0 & 3 & 0 & 0 & 6 \\ 0 & 0 & -5 & 0 & -20 \\ 0 & 0 & 0 & -26/5 & 52/5 \end{array}\right) \sim \left(\begin{array}{rrrrr} 1 & 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & -2 \end{array}\right)$$

The last column gives us the solution: $x_1 = -3$, $x_2 = 2$, $x_3 = 4$ and $x_4 = -2$. □

We can combine pivoting strategies together with the Gauss–Jordan elimination.

Example 3.36. Here we apply the Gauss–Jordan elimination with partial pivoting to the linear system examined in Example 3.22:

$$\left(\begin{array}{rrrrr} 1 & -2 & -2 & -2 & -11 \\ 2 & -1 & 2 & 4 & -8 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \end{array}\right) \sim \left(\begin{array}{rrrrr} 2 & -1 & 2 & 4 & -8 \\ 1 & -2 & -2 & -2 & -11 \\ -1 & 2 & 3 & -4 & 27 \\ -2 & 1 & 4 & -2 & 28 \end{array}\right) \sim$$
$$\left(\begin{array}{rrrrr} 2 & -1 & 2 & 4 & -8 \\ 0 & -3/2 & -3 & -4 & -7 \\ 0 & 3/2 & 4 & -2 & 23 \\ 0 & 0 & 6 & 2 & 20 \end{array}\right) \sim \left(\begin{array}{rrrrr} 2 & 0 & 4 & 20/3 & -10/3 \\ 0 & -3/2 & -3 & -4 & -7 \\ 0 & 0 & 1 & -6 & 16 \\ 0 & 0 & 6 & 2 & 20 \end{array}\right) \sim$$
$$\left(\begin{array}{rrrrr} 2 & 0 & 4 & 20/3 & -10/3 \\ 0 & -3/2 & -3 & -4 & -7 \\ 0 & 0 & 6 & 2 & 20 \\ 0 & 0 & 1 & -6 & 16 \end{array}\right) \sim \left(\begin{array}{rrrrr} 2 & 0 & 0 & 16/3 & -50/3 \\ 0 & -3/2 & 0 & -3 & 3 \\ 0 & 0 & 6 & 2 & 20 \\ 0 & 0 & 0 & -19/3 & 38/3 \end{array}\right) \sim$$
$$\left(\begin{array}{rrrrr} 2 & 0 & 0 & 0 & -6 \\ 0 & -3/2 & 0 & 0 & -3 \\ 0 & 0 & 6 & 0 & 24 \\ 0 & 0 & 0 & -19/3 & 38/3 \end{array}\right) \sim \left(\begin{array}{rrrrr} 1 & 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & -2 \end{array}\right)$$

Therefore, the solution is $x_1 = -3$, $x_2 = 2$, $x_3 = 4$ and $x_4 = -2$. □

Exercises

1. Solve the linear systems given in Exercises 1 and 2 of Section 3.3 with Gauss–Jordan elimination.
2. Prove that the number of arithmetic operation needed for the Gauss-Jordan elimination is $n^3/2 + n^2 - n/2$ multiplication and divisions.

3.5. Tridiagonal Linear Systems

We say that a square matrix (a_{ij}) is *tridiagonal* if $a_{ij} = 0$ for all $|i - j| > 1$, i.e., nonzero numbers can appear only in the main diagonal and in the next diagonal above and under it. Tridiagonal linear systems (i.e., a linear system with a tridiagonal coefficient matrix) appear frequently in applications, so it is an important class of linear systems. We will use the following notations:

$$\begin{pmatrix} d_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_1 & d_2 & c_2 & 0 & \cdots & 0 \\ 0 & a_2 & d_3 & c_3 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & a_{n-2} & d_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & a_{n-1} & d_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}. \tag{3.10}$$

It is practical to store the elements of a tridiagonal matrix in three vectors (a_i), (d_i) and (c_i), as it is used above. In this case only $3n - 2$ storage area is needed for the coefficients.

It is clear that applying the Gaussian elimination to the system (3.10) the elements a_i below the main diagonal will become 0, and the numbers c_i will not be changed during the elimination steps. We have to compute the new values of the variables d_i and b_i during the elimination. In the next algorithm we override the old values of the vectors (d_i) and (b_i) with the actual new ones.

Algorithm 3.37. Gaussian elimination for tridiagonal linear systems

INPUT: a_i, c_i $(i = 1, \ldots, n-1)$, d_i, b_i $(i = 1, \ldots, n)$
OUTPUT: $x_1, \ldots, x_n$

(elimination:)
for $i = 2, \ldots, n$ **do**
  $temp \leftarrow a_{i-1}/d_{i-1}$
  $d_i \leftarrow d_i - temp \cdot c_{i-1}$
  $b_i \leftarrow b_i - temp \cdot b_{i-1}$
end do
(backward substitution:)
$x_n \leftarrow b_n/d_n$
for $i = n-1, \ldots, 1$ **do**
  $x_i \leftarrow (b_i - c_i x_{i+1})/d_i$
end do
output$(x_1, x_2, \ldots, x_n)$

We can check that the method above requires $5n - 4$ number of multiplications and divisions. If we compare it with the number of operations of the Algorithm 3.23, which is $n^3/3$ multiplications and divisions, then we can see that for a tridiagonal system this special algorithm should be applied.

It follows from Theorem 3.32 that if the tridiagonal matrix $\mathbf{A}$ is also diagonally dominant, then Algorithm 3.37 can be performed (without pivoting).

Exercises

1. Solve the following tridiagonal linear systems:

$$\begin{array}{rcrcrcrcrcrcr}
x_1 & - & 0.5x_2 & & & & & & & & & = & 1.5 \\
0.5x_1 & + & 4x_2 & - & 0.5x_3 & & & & & & & = & -4.0 \\
 & & 0.5x_2 & + & 2x_3 & - & 0.5x_4 & & & & & = & 2.0 \\
 & & & & 0.5x_3 & + & 4x_4 & - & 0.5x_5 & & & = & -4.0 \\
 & & & & & & 0.5x_4 & + & 2x_5 & - & 0.5x_6 & = & 2.0 \\
 & & & & & & & & 0.5x_5 & + & x_6 & = & -0.5
\end{array}$$

2. Show that Algorithm 3.37 requires $5n - 4$ number of multiplications and divisions.
3. Formulate an algorithm similar to Algorithm 3.37 for a *band matrix* where the nonzero elements appear only in the main diagonal and in the next 2 diagonals above and below it, i.e., when $a_{ij} = 0$ for $|i - j| > 2$.

3.6. Simultaneous Linear Systems

Frequently we would like to solve so-called *simultaneous linear systems*, i.e., systems of the form $\mathbf{Ax} = \mathbf{b}^{(i)}$ for $i = 1, \ldots, m$, where the coefficient matrices are identical, but the right-hand-sides of the equations are different. We can shortly write the above system as $\mathbf{AX} = \mathbf{B}$, where the ith columns of the $n \times m$ dimensional matrix $\mathbf{B} = (\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(m)})$ is $\mathbf{b}^{(i)}$, and the ith column of the $n \times m$ dimensional matrix $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)})$ is $\mathbf{x}^{(i)}$, i.e., the solution of the system $\mathbf{Ax}^{(i)} = \mathbf{b}^{(i)}$. Since pivoting in the Gaussian or Gauss–Jordan elimination depends only on the coefficient matrix, it can be performed on the $n \times (n+m)$ dimensional augmented matrix. For example, if we perform the Gauss-Jordan elimination on the augmented matrix $(\mathbf{A}, \mathbf{B})$ we get a matrix of the form $(\mathbf{I}, \mathbf{X})$. Then the solution of the simultaneous linear system $\mathbf{X}$ appears in the last m columns of the augmented matrix.

Exercises

1. Show that the operation count of the Gaussian elimination on the augmented matrix $(\mathbf{A}, \mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)})$ is $n^3/3 + mn^2 - n/3$ number of multiplications and divisions.
2. Prove that the operation count of the Gauss–Jordan elimination on the augmented matrix $(\mathbf{A}, \mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)})$ is $n^3/2 + mn^2 - n/2$ number of multiplications and divisions.
3. Reformulate Algorithm 3.37 for solving simultaneous tridiagonal linear systems.
4. Prove that the system of linear systems $\mathbf{Ax}^{(i)} = \mathbf{b}^{(i)}$, $i = 1, 2, \ldots, m$ is equivalent to the matrix equation $\mathbf{AX} = \mathbf{B}$, where $\mathbf{X} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)})$ and $\mathbf{B} = (\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)})$.

3.7. Matrix Inversion and Determinants

The inverse matrix $\mathbf{A}^{-1}$ of a nonsingular square matrix $\mathbf{A}$ satisfies the matrix equation $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, so $\mathbf{A}^{-1}$ is the solution of the simultaneous linear system $\mathbf{A}\mathbf{X} = \mathbf{I}$. It can be shown that if such matrix $\mathbf{X}$ exists, then $\mathbf{X}\mathbf{A} = \mathbf{I}$ holds too, hence $\mathbf{X}$ is the inverse matrix of $\mathbf{A}$. We can use the Gauss-Jordan elimination to solve the simultaneous linear system. It can be checked that the number of operations needed to compute the matrix inverse with the Gauss–Jordan elimination is $\frac{3}{2}n^3 + \mathcal{O}(n^2)$ number of multiplications and divisions and $\frac{3}{2}n^3 + \mathcal{O}(n^2)$ number of additions and subtractions.

Example 3.38. Compute the inverse of the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 2 \\ -1 & 1 & 0 \\ -2 & 0 & -1 \end{pmatrix}.$$

We use the Gauss–Jordan elimination:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ -2 & 0 & -1 & 0 & 0 & 1 \end{array}\right) \sim \left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 & 0 \\ 0 & 0 & 3 & 2 & 0 & 1 \end{array}\right) \sim$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 & 0 \\ 0 & 0 & 3 & 2 & 0 & 1 \end{array}\right) \sim \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -1/3 & 0 & -2/3 \\ 0 & 1 & 0 & -1/3 & 1 & -2/3 \\ 0 & 0 & 3 & 2 & 0 & 1 \end{array}\right) \sim$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -1/3 & 0 & -2/3 \\ 0 & 1 & 0 & -1/3 & 1 & -2/3 \\ 0 & 0 & 1 & 2/3 & 0 & 1/3 \end{array}\right)$$

Hence

$$\mathbf{A}^{-1} = \frac{1}{3}\begin{pmatrix} -1 & 0 & -2 \\ -1 & 3 & -2 \\ 2 & 0 & 1 \end{pmatrix}.$$

□

Certainly, we can use pivoting techniques together with the Gauss-Jordan elimination for computing the inverse matrix if we wanted to reduce the rounding errors or to avoid division by zero.

According to Theorem 3.26 the Gaussian elimination with pivoting can be performed if and only if $\det(\mathbf{A}) \neq 0$. In the proof of the theorem we can see that $\det(\mathbf{A}) = (-1)^s \det(\mathbf{A}^{(n-1)})$, where s denotes the number of row changes. Therefore, the determinant is equal to the product of the pivot elements with an appropriate sign: $\det(\mathbf{A}) = (-1)^s a_{11} a_{22}^{(1)} \cdots a_{nn}^{(n-1)}$.

Example 3.39. Consider the coefficient matrix of Example 3.22, i.e., let

$$\mathbf{A} = \begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & -1 & 2 & 4 \\ -1 & 2 & 3 & -4 \\ -2 & 1 & 4 & -2 \end{pmatrix}.$$

Compute the determinant of $\mathbf{A}$. In Example 3.22 we performed the Gaussian elimination on $\mathbf{A}$ and got

$$\mathbf{A}^{(3)} = \begin{pmatrix} 1 & -2 & -2 & -2 \\ 0 & 3 & 6 & 8 \\ 0 & 0 & 1 & -6 \\ 0 & 0 & 0 & 38 \end{pmatrix}.$$

Therefore, $\det(\mathbf{A}) = \det(\mathbf{A}^{(3)}) = 1 \cdot 3 \cdot 1 \cdot 38 = 114$. □

Exercises

1. Compute the inverse of the matrices:

$$(a) \quad \begin{pmatrix} -1 & 1 & 2 \\ -2 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \qquad (b) \quad \begin{pmatrix} -3 & 1 & 2 \\ 0 & 3 & 1 \\ -2 & -1 & 1 \end{pmatrix} \qquad (c) \quad \begin{pmatrix} 1 & -1 & 0 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 2 & 2 & -1 \end{pmatrix}$$

2. Prove that the matrix inversion using Gauss–Jordan elimination requires $3n^3/2 - n/2$ number of multiplications and divisions.
3. Formulate an algorithm for matrix inversion using Gauss–Jordan elimination taking into account that in the problem $\mathbf{AX} = \mathbf{I}$ the matrix $\mathbf{I}$ has a special form, so multiplication by 0 should not be computed. Show that the resulting algorithm requires n^3 multiplications and divisions and $n^3 - 2n^2 + n$ additions and subtractions.
4. Compute the determinants of the matrices given in Exercise 1 using the Gaussian elimination.

Chapter 4
Iterative Techniques for Solving Linear Systems

In this chapter we first discuss the theory of linear fixed-point iteration, and then we apply it for the solution of linear systems (we define the Jacobi and Gauss–Seidel iterations). Finally, we introduce the condition number of matrices, and study perturbation of linear systems.

4.1. Linear Fixed-Point Iteration

In this section we investigate linear n dimensional fixed-point iterations of the form

$$\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c}, \qquad k = 0, 1, 2, \ldots. \tag{4.1}$$

First we consider the case when $\mathbf{c} = \mathbf{0}$. Then it is easy to see that $\mathbf{x}^{(k)} = \mathbf{T}^k\mathbf{x}^{(0)}$ for all $k = 1, 2, \ldots$.

Theorem 4.1. *The following statements are equivalent:*

(i) $\lim_{k\to\infty} \mathbf{T}^k = \mathbf{0}$ *(zero matrix), i.e.,* $\lim_{k\to\infty} \|\mathbf{T}^k\| = 0$ *for any matrix norm* $\|\cdot\|$*;*

(ii) $\lim_{k\to\infty} \mathbf{T}^k\mathbf{x} = \mathbf{0}$ *(zero vector) for all* $\mathbf{x} \in \mathbb{R}^n$*, i.e.,* $\lim_{k\to\infty} \|\mathbf{T}^k\mathbf{x}\| = 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *and for any vector norm* $\|\cdot\|$*;*

(iii) $\rho(\mathbf{T}) < 1$.

Proof. Statement (ii) follows from (i), since

$$\|\mathbf{T}^k\mathbf{x}\| \le \|\mathbf{T}^k\|\|\mathbf{x}\|$$

for all $x \in \mathbb{R}^n$ and for any norm $\|\cdot\|$.

Suppose (ii) holds. Let λ be an eigenvalue of $\mathbf{T}$, and let $\mathbf{v}$ be an eigenvector corresponding to λ. Then $\mathbf{T}^k\mathbf{v} = \lambda^k\mathbf{v}$, hence $\mathbf{T}^k\mathbf{v} \to \mathbf{0}$ (as $k \to \infty$) implies $|\lambda| < 1$, since $\mathbf{v} \neq \mathbf{0}$. Since λ was an arbitrary eigenvalue of $\mathbf{T}$, $\rho(\mathbf{T}) < 1$ is satisfied.

Now suppose (iii) holds. Theorem 3.17 implies that there exists a matrix norm $\|\cdot\|$ and $\varepsilon > 0$ such that $\|\mathbf{T}\| \le \rho(\mathbf{T}) + \varepsilon < 1$. Then

$$\|\mathbf{T}^k\| \le \|\mathbf{T}\|^k \le (\rho(\mathbf{T}) + \varepsilon)^k \to 0,$$

as $k \to \infty$. But then Theorem 2.47 yields $\|\mathbf{T}^k\| \to 0$ in any matrix norm $\|\cdot\|$, so (i) holds. □

The next theorem states that $\|\mathbf{T}\| < 1$ implies $\|\mathbf{T}^k\| \to 0$.

Theorem 4.2. *If $\|\mathbf{T}\| < 1$ in some matrix norm $\|\cdot\|$, then $\|\mathbf{T}^k\| \to 0$ as $k \to \infty$.*

Proof. The statement follows from $\|\mathbf{T}^k\| \le \|\mathbf{T}\|^k$. □

Next we investigate the convergence of the matrix *geometric series* or *Neumann-series* $\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots$, where $\mathbf{A}$ is a square matrix.

Theorem 4.3. *If $\rho(\mathbf{A}) < 1$, then the geometric series $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots$ is convergent, the matrix $\mathbf{I} - \mathbf{A}$ is invertible, and*

$$(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots .$$

Conversely, if the geometric series $\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots$ is convergent, then $\rho(\mathbf{A}) < 1$.

Proof. Let $\rho(\mathbf{A}) < 1$. Suppose that $\mathbf{I} - \mathbf{A}$ is not invertible. Then Theorem 3.3 yields that there exists a nonzero vector $\mathbf{x} \neq \mathbf{0}$ such that $(\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$. But then $\mathbf{A}\mathbf{x} = \mathbf{x}$, i.e., 1 is an eigenvalue of $\mathbf{A}$, which contradicts to the assumption that $\rho(\mathbf{A}) < 1$. Hence $\mathbf{I} - \mathbf{A}$ is invertible.

It is easy to check that

$$(\mathbf{I} - \mathbf{A})(\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^m) = \mathbf{I} - \mathbf{A}^{m+1}. \tag{4.2}$$

Therefore,

$$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^m = (\mathbf{I} - \mathbf{A})^{-1}(\mathbf{I} - \mathbf{A}^{m+1}),$$

and so, using that Theorem 4.1 implies $\mathbf{A}^{m+1} \to 0$, we get

$$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^m \to (\mathbf{I} - \mathbf{A})^{-1},$$

as $m \to \infty$.

Now suppose that the geometric series $\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots$ is convergent. Then it is easy to see that $\mathbf{A}^m \to \mathbf{0}$, and hence Theorem 4.1 yields $\rho(\mathbf{A}) < 1$. □

Corollary 4.4. *If $\|\mathbf{A}\| < 1$ in some matrix norm $\|\cdot\|$, then the matrix $\mathbf{I}-\mathbf{A}$ is invertible, the geometric series $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots$ is convergent, $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots = (\mathbf{I}-\mathbf{A})^{-1}$, and*

$$\|(\mathbf{I} - \mathbf{A})^{-1}\| \le \frac{1}{1 - \|\mathbf{A}\|}.$$

Proof. We have to prove only the last statement, the others follow immediately from Theorems 4.3 and 3.16. Using the continuity of the matrix norm, the triangle-inequality and the properties of the norm, we get

$$\begin{aligned}\|(\mathbf{I}-\mathbf{A})^{-1}\| &= \|\lim_{m\to\infty}(\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots+\mathbf{A}^m)\| \\ &= \lim_{m\to\infty}\|\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\cdots+\mathbf{A}^m\| \\ &\le \lim_{m\to\infty}(\|\mathbf{I}\|+\|\mathbf{A}\|+\|\mathbf{A}^2\|+\|\mathbf{A}^3\|+\cdots+\|\mathbf{A}^m\|) \\ &\le \lim_{m\to\infty}(1+\|\mathbf{A}\|+\|\mathbf{A}\|^2+\|\mathbf{A}\|^3+\cdots+\|\mathbf{A}\|^m) \\ &= \frac{1}{1-\|\mathbf{A}\|}.\end{aligned}$$

□

The last result has an important consequence: if $\mathbf{A}$ is nonsingular, then all matrices "close" to $\mathbf{A}$ are also nonsingular.

Theorem 4.5. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be* $n\times n$ *matrices. Let* $\mathbf{A}$ *be nonsingular, and*

$$\|\mathbf{A}-\mathbf{B}\| < \frac{1}{\|\mathbf{A}^{-1}\|}.$$

Then $\mathbf{B}$ *is also nonsingular, moreover,*

$$\|\mathbf{B}^{-1}\| \le \frac{\|\mathbf{A}^{-1}\|}{1-\|\mathbf{A}^{-1}\|\|\mathbf{A}-\mathbf{B}\|} \tag{4.3}$$

and

$$\|\mathbf{A}^{-1}-\mathbf{B}^{-1}\| \le \frac{\|\mathbf{A}^{-1}\|^2\|\mathbf{A}-\mathbf{B}\|}{1-\|\mathbf{A}^{-1}\|\|\mathbf{A}-\mathbf{B}\|}. \tag{4.4}$$

Proof. Consider the identities $\mathbf{B}=\mathbf{A}-(\mathbf{A}-\mathbf{B})=\mathbf{A}(\mathbf{I}-\mathbf{A}^{-1}(\mathbf{A}-\mathbf{B}))$. Using the assumption we get $\|\mathbf{A}^{-1}(\mathbf{A}-\mathbf{B})\|\le\|\mathbf{A}^{-1}\|\|\mathbf{A}-\mathbf{B}\|<1$, therefore, Corollary 4.4 yields that the matrix $\mathbf{I}-\mathbf{A}^{-1}(\mathbf{A}-\mathbf{B})$ is invertible. But then $\mathbf{B}^{-1}=(\mathbf{I}-\mathbf{A}^{-1}(\mathbf{A}-\mathbf{B}))^{-1}\mathbf{A}^{-1}$ also exists. From this, relation $\mathbf{A}^{-1}-\mathbf{B}^{-1}=\mathbf{A}^{-1}(\mathbf{B}-\mathbf{A})\mathbf{B}^{-1}$ and Corollary 4.4 imply estimates (4.3) and (4.4). □

Consider again the fixed-point problem (4.1). Now we consider the general case. It is easy to see that the kth term of the fixed-point iteration is

$$\mathbf{x}^{(k)}=\mathbf{T}^k\mathbf{x}^{(0)}+(\mathbf{T}^{k-1}+\mathbf{T}^{k-2}+\cdots+\mathbf{T}+\mathbf{I})\mathbf{c},\qquad k=1,2,\ldots.$$

Theorems 4.1 and 4.3 imply the following results.

Theorem 4.6. *Let* $\mathbf{c}\neq\mathbf{0}$*. The fixed-point equation* $\mathbf{x}=\mathbf{T}\mathbf{x}+\mathbf{c}$ *has a unique solution and the fixed-point iteration (4.1) converges to the unique solution of the equation for all* $\mathbf{x}^{(0)}$ *if and only if* $\rho(\mathbf{T})<1$.

Proof. Let $\rho(\mathbf{T}) < 1$. Then Theorem 4.3 yields that $\mathbf{I} - \mathbf{T}$ is invertible, hence the equation $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$ has a unique solution: $\mathbf{x} = (\mathbf{I} - \mathbf{T})^{-1}\mathbf{c}$. Theorems 4.1 and 4.3 imply that $\mathbf{T}^k\mathbf{x}^{(0)} \to 0$ for all $\mathbf{x}^{(0)} \in \mathbb{R}^n$, and $(\mathbf{T}^{k-1} + \mathbf{T}^{k-2} + \cdots + \mathbf{T} + \mathbf{I})\mathbf{c} \to (\mathbf{I} - \mathbf{T})^{-1}\mathbf{c}$ as $k \to \infty$.

Conversely, let $\mathbf{x}$ be the solution of $\mathbf{x} = \mathbf{T}\mathbf{x}+\mathbf{c}$, and suppose $\mathbf{x}^{(k)} \to \mathbf{x}$ as $k \to \infty$. Then subtracting the equations $\mathbf{x} = \mathbf{T}\mathbf{x}+\mathbf{c}$ and $\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)}+\mathbf{c}$ we get $\mathbf{x}-\mathbf{x}^{(k+1)} = \mathbf{T}(\mathbf{x}-\mathbf{x}^{(k)})$, and so

$$\mathbf{x} - \mathbf{x}^{(k+1)} = \mathbf{T}(\mathbf{x} - \mathbf{x}^{(k)}) = \cdots = \mathbf{T}^{k+1}(\mathbf{x} - \mathbf{x}^{(0)}). \tag{4.5}$$

Let $\mathbf{z}$ be an arbitrary vector, and $\mathbf{x}^{(0)} = \mathbf{x} - \mathbf{z}$. Then

$$\lim_{k\to\infty} \mathbf{T}^{k+1}\mathbf{z} = \lim_{k\to\infty} \mathbf{T}^{k+1}(\mathbf{x} - \mathbf{x}^{(0)}) = \lim_{k\to\infty} (\mathbf{x} - \mathbf{x}^{(k+1)}) = \mathbf{x} - \mathbf{x} = \mathbf{0}.$$

Theorem 4.1 yields $\rho(\mathbf{T}) < 1$. □

Corollary 4.7. *If $\|\mathbf{T}\| < 1$ in some matrix norm $\|\cdot\|$, then the iteration (4.1) is convergent for all initial values $\mathbf{x}^{(0)}$, and*

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \le \|\mathbf{T}\|^k \|\mathbf{x} - \mathbf{x}^{(0)}\|. \tag{4.6}$$

Estimate (4.6) implies that the smaller the $\|\mathbf{T}\|$ is, the faster the convergence of the sequence $\mathbf{x}^{(k)}$. Therefore, Theorem 3.17 yields that the smaller the $\rho(\mathbf{T})$ is, the faster the convergence (in a certain norm) of the sequence $\mathbf{x}^{(k)}$.

Next we investigate the effect of rounding error in the computation of the linear fixed-point iteration. Suppose that instead of the sequence (4.1) we generate the sequence

$$\mathbf{y}^{(k+1)} = \mathbf{T}\mathbf{y}^{(k)} + \mathbf{c} + \mathbf{w}^{(k+1)}, \qquad k = 0, 1, \ldots, \tag{4.7}$$

$$\mathbf{y}^{(0)} = \mathbf{x}^{(0)} + \mathbf{w}^{(0)}, \tag{4.8}$$

where the effect of the rounding error in the kth step is represented by $\mathbf{w}^{(k+1)}$, and $\mathbf{w}^{(0)}$ is the rounding error we get when we store the initial value of the sequence. We suppose that

$$\|\mathbf{w}^{(k)}\| \le \varepsilon, \qquad k = 0, 1, \ldots$$

holds in a certain vector norm. We compute the difference of equations (4.7) and (4.1):

$$\mathbf{y}^{(k+1)} - \mathbf{x}^{(k+1)} = \mathbf{T}(\mathbf{y}^{(k)} - \mathbf{x}^{(k)}) + \mathbf{w}^{(k+1)}.$$

Then

$$\begin{aligned}
\|\mathbf{y}^{(k+1)} - \mathbf{x}^{(k+1)}\| &\le \|\mathbf{T}(\mathbf{y}^{(k)} - \mathbf{x}^{(k)})\| + \|\mathbf{w}^{(k+1)}\| \\
&\le \|\mathbf{T}\|\|\mathbf{y}^{(k)} - \mathbf{x}^{(k)}\| + \varepsilon \\
&\vdots \\
&\le \|\mathbf{T}\|^{k+1}\|\mathbf{y}^{(0)} - \mathbf{x}^{(0)}\| + (\|\mathbf{T}\|^k + \cdots + \|\mathbf{T}\| + 1)\varepsilon \\
&\le (\|\mathbf{T}\|^{k+1} + \|\mathbf{T}\|^k + \cdots + \|\mathbf{T}\| + 1)\varepsilon.
\end{aligned}$$

If $\|\mathbf{T}\| < 1$, then the last expression can be estimated by the sum of the geometric series:

$$\|\mathbf{y}^{(k+1)} - \mathbf{x}^{(k+1)}\| \le \frac{1}{1-\|\mathbf{T}\|}\varepsilon.$$

This shows that the computation is stable with respect to the rounding errors, and the smaller the $\|\mathbf{T}\|$ is, the smaller the rounding error is.

Exercises

1. Compute the sum of the geometric series $\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots$ for

$$\text{(a)}\quad \mathbf{A} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \text{(b)}\quad \mathbf{A} = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/5 \end{pmatrix}.$$

2. Prove identity (4.2).
3. Work out the details of the proofs of (4.3) and (4.4).
4. Find all values of the parameter α for which the matrix sequence

$$\begin{pmatrix} 1 & 2 \\ \alpha & 0 \end{pmatrix}^k$$

converges to the zero matrix.

4.2. Jacobi Iteration

Example 4.8. Solve the linear system

$$\begin{array}{rcrcrcr} 5x_1 & + & 3x_2 & - & x_3 & = & -4 \\ 2x_1 & - & 10x_2 & + & x_3 & = & 25 \\ -3x_1 & + & 4x_2 & - & 12x_3 & = & -47. \end{array} \tag{4.9}$$

We express x_1 from the first, x_2 from the second and x_3 from the third equation:

$$\begin{aligned} x_1 &= (-4 - 3x_2 + x_3)/5 \\ x_2 &= (-25 + 2x_1 + x_3)/10 \\ x_3 &= (47 - 3x_1 + 4x_2)/12. \end{aligned} \tag{4.10}$$

System (4.10) is a three dimensional linear fixed-point equation, so we define the sequences

$$\begin{aligned} x_1^{(k+1)} &= (-4 - 3x_2^{(k)} + x_3^{(k)})/5 \\ x_2^{(k+1)} &= (-25 + 2x_1^{(k)} + x_3^{(k)})/10 \\ x_3^{(k+1)} &= (47 - 3x_1^{(k)} + 4x_2^{(k)})/12 \end{aligned} \tag{4.11}$$

for $k = 0, 1, 2, \ldots$. Table 4.1 lists the numerical results starting from the initial values $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$. We can observe that the sequences converge, and their limits are $x_1 = 1$,

$x_2 = -2$ and $x_3 = 3$, which are the solutions of the system (4.9). The iteration (4.11) can be written in a vector form as

$$\mathbf{x}^{(k+1)} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c}, \tag{4.12}$$

where

$$\mathbf{T} = \begin{pmatrix} 0 & -3/5 & 1/5 \\ 2/10 & 0 & 1/10 \\ -3/12 & 4/12 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{c} = \begin{pmatrix} -4/5 \\ -25/10 \\ 47/12 \end{pmatrix}.$$

Corollary 4.7 yields the convergence of the iteration (4.12) if the norm of $\mathbf{T}$ is less than 1 in some norm. Since $\|\mathbf{T}\|_\infty = \max\{4/5, 3/10, 7/12\} = 4/5 < 1$, we get that the iteration (4.11) is convergent. □

Table 4.1: Jacobi iteration

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	-0.800000	-2.500000	3.916667
2	1.483333	-2.268333	3.283333
3	1.217667	-1.875000	2.789722
4	0.882944	-1.977494	2.987250
⋮	⋮	⋮	⋮
14	0.999999	-1.999992	2.999990
15	0.999993	-2.000001	3.000003
16	1.000001	-2.000001	3.000001
17	1.000001	-2.000000	2.999999
18	1.000000	-2.000000	3.000000

Consider the linear system

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n. \end{array} \tag{4.13}$$

If $a_{ii} \neq 0$ for all $i = 1, \dots, n$, then the system (4.13) can be transformed into the form

$$x_i = -\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j + \frac{b_i}{a_{ii}}, \qquad i = 1, \dots, n, \tag{4.14}$$

and we can define the *Jacobi iteration* for $k = 0, 1, 2, \dots$ by

$$x_i^{(k+1)} = -\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \qquad i = 1, \dots, n. \tag{4.15}$$

If $a_{ii} = 0$ for some i, then we can try to interchange rows so that in the resulting matrix $a_{ii} \neq 0$ holds for all $i = 1, \dots, n$. We introduce the notations $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix},$$

and $\mathbf{D} = \operatorname{diag}(a_{11}, a_{22}, \ldots, a_{nn})$. $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular matrices (with zeros in the diagonal too). With this notation the linear system $\mathbf{Ax} = \mathbf{b}$ can be rewritten as $\mathbf{Dx} = -(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b}$. Then multiplying this equation by $\mathbf{D}^{-1}$ we get a linear system of the form (4.14). Therefore, the Jacobi iteration can be defined by (4.12), where $\mathbf{T} = \mathbf{T}_J := -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$ and $\mathbf{c} = \mathbf{D}^{-1}\mathbf{b}$.

Theorem 4.6 and Corollary 4.7 imply the following necessary and sufficient condition for the convergence of the Jacobi iteration.

Theorem 4.9. *The Jacobi iteration is convergent for all initial values if and only if* $\rho(\mathbf{T}_J) < 1$.

Corollary 4.10. *If* $\|\mathbf{T}_J\| < 1$ *in some matrix norm* $\|\cdot\|$, *then the Jacobi iteration is convergent for all initial values* $\mathbf{x}^{(0)}$.

In practice we can use the following sufficient condition.

Theorem 4.11. *If the matrix* $\mathbf{A}$ *is diagonally dominant, then the Jacobi iteration is convergent for all initial values* $\mathbf{x}^{(0)}$.

Proof. Since

$$\mathbf{T}_J = \begin{pmatrix} 0 & -a_{12}/a_{11} & -a_{13}/a_{11} & \cdots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & -a_{23}/a_{22} & \cdots & -a_{2n}/a_{22} \\ -a_{31}/a_{33} & -a_{32}/a_{33} & 0 & \cdots & -a_{3n}/a_{33} \\ \vdots & & & \ddots & \vdots \\ -a_{n1}/a_{nn} & -a_{n2}/a_{nn} & -a_{n3}/a_{nn} & \cdots & 0 \end{pmatrix},$$

using the diagonal dominance of $\mathbf{A}$, we get

$$\|\mathbf{T}_J\|_\infty = \max_{i=1,\ldots,n} \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{|a_{ij}|}{|a_{ii}|} \right\} < 1.$$

Hence Corollary 4.10 implies the statement. □

Exercises

1. Solve the following linear systems with Jacobi iteration:

$$\text{(a)} \quad \begin{array}{rcrcrcr} 6.2x_1 & + & 1.1x_2 & - & 3.4x_3 & = & 5.1 \\ -0.6x_1 & + & 2.9x_2 & + & 0.3x_3 & = & -7.2 \\ 1.1x_1 & - & 0.6x_2 & + & 4.4x_3 & = & 3.1 \end{array}$$

$$\text{(b)} \quad \begin{array}{rcrcrcrcr} -8x_1 & + & 3x_2 & - & 2x_3 & & & = & 6 \\ 2x_1 & + & 6x_2 & + & x_3 & - & 2x_4 & = & 3 \\ 3x_1 & - & 3x_2 & + & 10x_3 & + & x_4 & = & 5 \\ & & x_2 & - & 3x_3 & + & 7x_4 & = & -17 \end{array}$$

2. Show that the Jacobi iteration is convergent for all initial values if $\mathbf{A}$ is column diagonally dominant.

4.3. Gauss–Seidel Iteration

Example 4.12. Consider again the system (4.9), and its equivalent form (4.10). Define the iteration

$$\begin{aligned} x_1^{(k+1)} &= (-4 - 3x_2^{(k)} + x_3^{(k)})/5 \\ x_2^{(k+1)} &= (-25 + 2x_1^{(k+1)} + x_3^{(k)})/10 \\ x_3^{(k+1)} &= (47 - 3x_1^{(k+1)} + 4x_2^{(k+1)})/12. \end{aligned} \tag{4.16}$$

The difference between the iterations (4.11) and (4.16) is that here if a new value of a variable x_i is computed, then its is immediately used in the computation of the next iteration. The $k+1$-th value of x_1 is computed in the first equation, then in the computation of the new value of x_2 the updated value of $x_1^{(k+1)}$ is used in the second equation (which is hopefully a better approximation of x_1 than $x_1^{(k)}$) together with $x_3^{(k)}$, which has no a new value at this moment. In Table 4.2 we present the numerical results corresponding to the initial values $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$. We can observe that this method converges faster to the limits than the Jacobi iteration. □

Table 4.2: Gauss–Seidel iteration

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	-0.800000	-2.660000	3.230000
2	1.442000	-1.888600	2.926633
3	0.918487	-2.023639	3.012499
4	1.016683	-1.995413	2.997358
5	0.996720	-2.000920	3.000513
6	1.000655	-1.999818	2.999897
7	0.999870	-2.000036	3.000020
8	1.000026	-1.999993	2.999996
9	0.999995	-2.000001	3.000001
10	1.000001	-2.000000	3.000000
11	1.000000	-2.000000	3.000000

Now consider again the general linear system (4.13). Motivated by the example above, we define the *Gauss–Seidel iteration* to solve the system (4.13). For $k = 0, 1, 2, \ldots$ (if $a_{ii} \neq 0$ for all $i = 1, \ldots, n$) we define

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \qquad i = 1, \ldots, n. \tag{4.17}$$

Equation (4.17) can be rearranged to

$$\sum_{j=1}^{i} a_{ij} x_j^{(k+1)} = -\sum_{j=i+1}^{n} a_{ij} x_j^{(k)} + b_i, \qquad i = 1, \ldots, n,$$

i.e., using matrix notation,

$$(\mathbf{D} + \mathbf{L})\mathbf{x}^{(k+1)} = -\mathbf{U}\mathbf{x}^{(k)} + \mathbf{b},$$

where $\mathbf{L}$, $\mathbf{D}$, $\mathbf{U}$ is defined in the previous section. So the Gauss–Seidel iteration can be written in the form (4.12) with $\mathbf{T} = \mathbf{T}_G := -(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}$ and $\mathbf{c} = (\mathbf{D}+\mathbf{L})^{-1}\mathbf{b}$.

Theorem 4.6 and Corollary 4.7 imply immediately the next results.

Theorem 4.13. *The Gauss–Seidel iteration is convergent for any initial value if and only if $\rho(\mathbf{T}_G) < 1$.*

Corollary 4.14. *If $\|\mathbf{T}_G\| < 1$ in some matrix norm $\|\cdot\|$, then the Gauss–Seidel iteration is convergent for all initial values $\mathbf{x}^{(0)}$.*

Similarly to the Jacobi iteration, the Gauss–Seidel iteration is also convergent if the coefficient matrix is diagonally dominant.

Theorem 4.15. *If $\mathbf{A}$ is diagonally dominant, then the Gauss–Seidel iteration is convergent for all initial values $\mathbf{x}^{(0)}$.*

Proof. Let $\mathbf{x} = (x_1, \dots, x_n)^T$ be the solution of equation (4.13). Then we express x_i from the ith equation of (4.13), and subtracting it from (4.17), we get

$$x_i^{(k+1)} - x_i = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}}(x_j^{(k+1)} - x_j) - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}}(x_j^{(k)} - x_j).$$

Therefore,

$$|x_i^{(k+1)} - x_i| \le \sum_{j=1}^{i-1} \left|\frac{a_{ij}}{a_{ii}}\right| |x_j^{(k+1)} - x_j| + \sum_{j=i+1}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| |x_j^{(k)} - x_j|. \tag{4.18}$$

Let

$$\alpha_i := \sum_{j=1}^{i-1} \left|\frac{a_{ij}}{a_{ii}}\right| \qquad \text{and} \qquad \beta_i := \sum_{j=i+1}^{n} \left|\frac{a_{ij}}{a_{ii}}\right|.$$

With this notation inequality (4.18) yields

$$|x_i^{(k+1)} - x_i| \le \alpha_i \|\mathbf{x}^{(k+1)} - \mathbf{x}\|_\infty + \beta_i \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty$$

for $i = 1, \dots, n$. Let l be an index for which $|x_l^{(k+1)} - x_l| = \|\mathbf{x}^{(k+1)} - \mathbf{x}\|_\infty$. Then

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}\|_\infty \le \alpha_l \|\mathbf{x}^{(k+1)} - \mathbf{x}\|_\infty + \beta_l \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty.$$

The matrix $\mathbf{A}$ is diagonally dominant, therefore, $\alpha_l < 1$, and so

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}\|_\infty \le \frac{\beta_l}{1-\alpha_l} \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty.$$

Hence we obtain

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty \le \left(\max_{l=1,\dots,n} \frac{\beta_l}{1-\alpha_l}\right)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_\infty.$$

This guarantees that the Gauss–Seidel iteration is convergent, since the diagonal dominance yields

$$\frac{\beta_l}{1-\alpha_l} \le \alpha_l + \beta_l < 1$$

for all $l = 1, \dots, n$, and so

$$\max_{l=1,\dots,n} \frac{\beta_l}{1-\alpha_l} \le \max_{l=1,\dots,n} \{\alpha_l + \beta_l\} = \|\mathbf{T}_J\|_\infty < 1 \tag{4.19}$$

follows. □

Estimate (4.19) yields that the error estimate of the Gauss–Seidel iteration is better than that of for the Jacobi iteration. So, in general, the Gauss–Seidel iteration converges faster for the case of a diagonally dominant coefficient matrix. In the general case the Gauss–Seidel iteration is faster than the Jacobi iteration if $\rho(\mathbf{T}_G) < \rho(\mathbf{T}_J)$. But we do not know a general condition in terms of the coefficient matrix $\mathbf{A}$ to check this relation. We formulate one result below for a special case without proof.

Theorem 4.16 (Stein–Rosenberg). *Suppose $a_{ij} \le 0$ if $i \ne j$ and $a_{ii} > 0$ for all $i = 1, \dots, n$. Then exactly one of the statements holds:*

1. $0 \le \rho(\mathbf{T}_G) < \rho(\mathbf{T}_J) < 1$,
2. $1 < \rho(\mathbf{T}_J) < \rho(\mathbf{T}_G)$,
3. $\rho(\mathbf{T}_J) = \rho(\mathbf{T}_G) = 0$,
4. $\rho(\mathbf{T}_J) = \rho(\mathbf{T}_G) = 1$.

The theorem implies that for systems satisfying the conditions of the theorem the Jacobi iteration is convergent if and only if the Gauss–Seidel iteration is convergent, and the Gauss–Seidel iteration is faster. But in general we can find examples when the Jacobi iteration converges faster than the Gauss–Seidel iteration.

Exercises

1. Solve the systems given in Exercise 1 of the previous section using Gauss–Seidel iteration.
2. Show that both the Jacobi and the Gauss–Seidel iteration determine the exact root of the system in finitely many steps if $\mathbf{A}$ is upper triangular and $a_{ii} \ne 0$ for $i = 1, \dots, n$.

4.4. Error Bounds and Iterative Refinement

We can introduce stopping criteria for the Jacobi and the Gauss–Seidel iterations similar to nonlinear iterations. As we defined in Section 2.8, we can use the following stopping criteria or any combination of them:

$$\text{(i)}\ \ \|\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)}\|<\varepsilon,\quad \text{(ii)}\ \ \frac{\|\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k+1)}\|}<\varepsilon\quad\text{and}\quad \text{(iii)}\ \ \|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\|<\varepsilon.$$

Condition (iii) is a natural analogue of condition (iii) of Section 2.8 used for nonlinear equations. We investigate this criterion in this section.

The vector $\mathbf{r}:=\mathbf{b}-\mathbf{A}\tilde{\mathbf{x}}$ is called the *residual vector* of the approximate solution $\tilde{\mathbf{x}}$ of the linear system $\mathbf{A}\mathbf{x}=\mathbf{b}$. The stopping criterion (iii) relies on the hypothesis that if the norm of $\mathbf{r}$ is small, then $\tilde{\mathbf{x}}$ is a good approximation of the exact solution $\mathbf{x}$. The following example shows that this is not necessarily true in general.

Example 4.17. The exact solution of the linear system

$$\begin{pmatrix} 4 & 1 \\ 4.03 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} 5 \\ 5.03 \end{pmatrix} \tag{4.20}$$

is $\mathbf{x}=(1,\ 1)^T$. Consider $\tilde{\mathbf{x}}=(2,\ -3)^T$ as the "approximate" solution. The corresponding residual vector is $\mathbf{r}=\mathbf{b}-\mathbf{A}\tilde{\mathbf{x}}=(0,\ 0.03)^T$. Its infinity norm is $\|\mathbf{r}\|_\infty=0.03$, which is small, but $\tilde{\mathbf{x}}$ cannot be considered as a good approximation of the true solution. □

The next result gives conditions which imply that the smallness of the norm of $\|\mathbf{r}\|$ yields that the error of the approximation is also small.

Theorem 4.18. *Let* $\mathbf{A}$ *be a nonsingular square matrix,* $\mathbf{x}$ *be the exact solution of the system* $\mathbf{A}\mathbf{x}=\mathbf{b}$*, the vector* $\tilde{\mathbf{x}}$ *is an approximate solution, and* $\mathbf{r}:=\mathbf{b}-\mathbf{A}\tilde{\mathbf{x}}$*. Then*

$$\|\mathbf{x}-\tilde{\mathbf{x}}\|\le\|\mathbf{A}^{-1}\|\|\mathbf{r}\|, \tag{4.21}$$

and

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|}\le\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{4.22}$$

Proof. From the relations $\mathbf{A}\mathbf{x}=\mathbf{b}$ and $\mathbf{A}\tilde{\mathbf{x}}=\mathbf{b}-\mathbf{r}$ we have $\mathbf{A}(\mathbf{x}-\tilde{\mathbf{x}})=\mathbf{r}$, and hence $\mathbf{x}-\tilde{\mathbf{x}}=\mathbf{A}^{-1}\mathbf{r}$. This relation together with $\|\mathbf{A}^{-1}\mathbf{r}\|\le\|\mathbf{A}^{-1}\|\|\mathbf{r}\|$ implies (4.21).

Estimates (4.21) and $\|\mathbf{b}\|\le\|\mathbf{A}\|\|\mathbf{x}\|$ yield

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|}\le\frac{\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\|\mathbf{r}\|}{\|\mathbf{A}\|\|\mathbf{x}\|}\le\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

□

The previous result answers our previous question: if the residual vector is small in norm, then it implies the smallness of the error of the approximation only if the product $\|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ is not too big. The number $\mathrm{cond}(\mathbf{A}) := \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ is called the *condition number* of the matrix $\mathbf{A}$ relative to a norm $\|\cdot\|$. The condition number corresponding to the $\|\cdot\|_p$ norm is denoted by $\mathrm{cond}_p(\mathbf{A})$. If a condition number of the matrix $\mathbf{A}$ is "big", then it is called *ill-conditioned*, otherwise it is called *well-conditioned*. It is not defined exactly how big the condition number should be in order to call the matrix ill-conditioned. In practice, if the condition number is bigger than 100–1000, then we say that the matrix is ill-conditioned. Therefore, if the coefficient matrix is ill-conditioned then the stopping criterion (iii) is not reliable.

Example 4.19. Consider the coefficient martix $\mathbf{A}$ of Example 4.17. We can check that

$$\mathbf{A}^{-1} = \begin{pmatrix} -33.33 & 33.33 \\ 143.3 & -133.3 \end{pmatrix},$$

and so $\|\mathbf{A}\|_\infty = 5.03$, $\|\mathbf{A}^{-1}\|_\infty = 267.6$. Therefore, $\mathrm{cond}_\infty(\mathbf{A}) = 1346$, and Theorem 4.18 explains why $(2,-3)^T$ is not a good approximation of the true solution despite the fact that $\mathbf{r}$ is small in norm. □

Suppose we solve the linear system $\mathbf{Ax} = \mathbf{b}$ with Gaussian elimination and t-digit arithmetic. Let $\tilde{\mathbf{x}}$ be the numerical solution, which, in general, has rounding error. We compute the residual vector $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$, but using here $2t$-digit arithmetic (double precision) for the computation of $\mathbf{r}$. It can be shown that

$$\|\mathbf{r}\| \approx 10^{-t}\|\mathbf{A}\|\|\tilde{\mathbf{x}}\|.$$

We can use this relation to estimate the condition number of $\mathbf{A}$ in the following way: Consider the equation $\mathbf{Ay} = \mathbf{r}$, and let $\tilde{\mathbf{y}}$ be its numerical solution using t-digit arithmetic. Note that the linear system $\mathbf{Ay} = \mathbf{r}$ can be solved effectively if we store the l_{ij} factors and the row changes used in the first Gaussian elimination, and now we do the elimination steps only on the vector $\mathbf{r}$. (In Section 5.1 below we will show an effective method for solving several linear systems with the same coefficient matrix.) Then

$$\tilde{\mathbf{y}} \approx \mathbf{A}^{-1}\mathbf{r} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}) = \mathbf{A}^{-1}\mathbf{b} - \tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{x}},$$

so $\|\tilde{\mathbf{y}}\|$ is an estimate of the error $\|\mathbf{x} - \tilde{\mathbf{x}}\|$, and

$$\|\tilde{\mathbf{y}}\| \approx \|\mathbf{A}^{-1}\mathbf{r}\| \le \|\mathbf{A}^{-1}\|\|\mathbf{r}\| \approx \|\mathbf{A}^{-1}\|\|\mathbf{A}\|10^{-t}\|\tilde{\mathbf{x}}\| = 10^{-t}\mathrm{cond}(\mathbf{A})\|\tilde{\mathbf{x}}\|.$$

From this we get the formula

$$\mathrm{cond}(\mathbf{A}) \approx 10^t \frac{\|\tilde{\mathbf{y}}\|}{\|\tilde{\mathbf{x}}\|} \tag{4.23}$$

as an approximation of the condition number. Let $\tilde{\mathbf{r}} := \mathbf{r} - \mathbf{A}\tilde{\mathbf{y}}$ be the residual vector of $\tilde{\mathbf{y}}$. In general, $\|\tilde{\mathbf{r}}\|$ is much smaller than $\|\mathbf{r}\|$, therefore, if instead of $\tilde{\mathbf{x}}$ we consider $\bar{\mathbf{x}} := \tilde{\mathbf{x}} + \tilde{\mathbf{y}}$ as the approximation of $\mathbf{x}$, then for the residual vector corresponding to $\bar{\mathbf{x}}$ we have

$$\|\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}\| = \|\mathbf{b} - \mathbf{A}(\tilde{\mathbf{x}} + \tilde{\mathbf{y}})\| = \|\mathbf{r} - \mathbf{A}\tilde{\mathbf{y}}\| = \|\tilde{\mathbf{r}}\| \ll \|\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}\|,$$

i.e., $\bar{\mathbf{x}}$ is a much better approximation of $\mathbf{x}$ than $\tilde{\mathbf{x}}$. If we repeat this procedure we get the method of *iterative refinement*. This method gives a good approximation of the solution in a few steps even for ill-conditioned coefficient matrices.

Algorithm 4.20. Iterative refinement

INPUT: $\mathbf{A}$, $\mathbf{b}$
N - maximal iteration number
TOL - tolerance
t - number of digits of precision
OUTPUT $\mathbf{z}$ - approximate solution
$COND$ - estimate of $\text{cond}_\infty(\mathbf{A})$

We solve the system $\mathbf{Ax} = \mathbf{b}$ with Gaussian elimination
for $k = 1, 2, \ldots, N$ **do**
  We compute the residual vector $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ using double precision.
  We solve $\mathbf{Ay} = \mathbf{r}$ for $\mathbf{y}$
  $\mathbf{z} \leftarrow \mathbf{x} + \mathbf{y}$
  if $k = 1$ **do**
    $COND \leftarrow 10^t \frac{\|\mathbf{y}\|_\infty}{\|\mathbf{x}\|_\infty}$
    output$(COND)$
  end do
  if $\|\mathbf{y}\|_\infty < TOL$ **do**
    output$(\mathbf{z})$
    stop
  end do
  $\mathbf{x} \leftarrow \mathbf{z}$
end do
output(The maximal number of iteration is exceeded.)

Example 4.21. Consider again system (4.20). Its exact solution is $\mathbf{x} = (1,\ 1)^T$. Using Gaussian elimination with 4-digit arithmetic we get the approximate solution $\tilde{\mathbf{x}} = (0.9375,\ 1.2500)^T$. Its residual vector is (with double precision): $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} = (0,\ 0.001875)^T$, so $\|\mathbf{r}\|_\infty = 0.001875$.

Solving $\mathbf{Ay} = \mathbf{r}$ with Gaussian elimination (with 4-digit arithmetic) we get the approximate solution $\tilde{\mathbf{y}} = (0.0586,\ -0.2344)^T$. Hence (4.23) yields

$$\text{cond}_\infty(\mathbf{A}) \approx 10^4 \frac{\|\tilde{\mathbf{y}}\|_\infty}{\|\tilde{\mathbf{x}}\|_\infty} = 10^4 \frac{0.2344}{1.25} = 1875. \tag{4.24}$$

We have seen in Example 4.19 that $\text{cond}_\infty(\mathbf{A}) = 1346$, so (4.24) is an approximation of the condition number. The relative error of the approximate solution $\tilde{\mathbf{x}}$ is

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty} = 0.25,$$

which is relatively large (since $\mathbf{A}$ is ill-conditioned). Using Theorem 4.18 we get the error bound

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty} \le \text{cond}_\infty(\mathbf{A}) \frac{\|\mathbf{r}\|_\infty}{\|\mathbf{b}\|_\infty} = 0.5017$$

for the relative error. Using one step of the iterative refinement we get the approximate solution $\mathbf{x}^{(2)} = \mathbf{x} + \mathbf{y} = (0.9961, 1.016)^T$, which is close to the true solution. □

Exercises

1. Compute the condition numbers cond_∞ and cond_1 of the following matrices:

$$\text{(a)} \quad \begin{pmatrix} 1 & 2 \\ 4 & -1 \end{pmatrix}, \qquad \text{(b)} \quad \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}$$

2. Estimate the condition number $\mathrm{cond}_\infty(\mathbf{A})$ for

$$\mathbf{A} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}.$$

3. Using 4-digit arithmetic solve

$$0.009x_1 - 0.52x_2 = -5.191$$
$$9211x_1 + 21.1x_2 = 9422$$

with applying two steps of the iterative refinement. (The exact solution is: $(1,\ 10)$.)

4.5. Perturbation of Linear Systems

Consider the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{4.25}$$

Suppose that instead of (4.25) we solve the linear system

$$\mathbf{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{4.26}$$

where $\tilde{\mathbf{b}} := \mathbf{b} + \Delta\mathbf{b}$ is a perturbation of $\mathbf{b}$ by $\Delta\mathbf{b}$. Its exact solution is denoted by $\tilde{\mathbf{x}}$. The next result gives a relation between the solutions of the two problems.

Theorem 4.22. *Let $\mathbf{A}$ be a nonsingular square matrix, $\mathbf{x}$ and $\tilde{\mathbf{x}}$ be solutions of the linear systems (4.25) and (4.26), respectively. Then*

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \mathrm{cond}(\mathbf{A}) \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}.$$

Proof. Subtracting (4.25) and (4.26) we get $\mathbf{A}(\mathbf{x}-\tilde{\mathbf{x}}) = \mathbf{b}-\tilde{\mathbf{b}}$, hence $\mathbf{x}-\tilde{\mathbf{x}} = \mathbf{A}^{-1}(\mathbf{b}-\tilde{\mathbf{b}})$, therefore, $\|\mathbf{x}-\tilde{\mathbf{x}}\| \le \|\mathbf{A}^{-1}\|\|\mathbf{b}-\tilde{\mathbf{b}}\|$. Using this and the inequality $\|\mathbf{b}\| = \|\mathbf{A}\mathbf{x}\| \le \|\mathbf{A}\|\|\mathbf{x}\|$ it follows

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \frac{\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{A}\|\|\mathbf{x}\|} \le \mathrm{cond}(\mathbf{A}) \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}.$$

□

The theorem says that one order of magnitude increase in $\mathrm{cond}(\mathbf{A})$ can result in one order of magnitude increase in the relative error of the approximation, or in other words, a loss of one significant digit in the approximation.

Now we consider the general case, when we perturb both the coefficient matrix and the right-hand-side of the system. We consider the linear system

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{4.27}$$

where $\|\mathbf{b} - \tilde{\mathbf{b}}\|$ and $\|\mathbf{A} - \tilde{\mathbf{A}}\|$ are "small".

Theorem 4.23. *Let $\mathbf{A}$ be a nonsingular square matrix, and $\tilde{\mathbf{A}}$ be such that $\|\mathbf{A} - \tilde{\mathbf{A}}\| < 1/\|\mathbf{A}^{-1}\|$. Let $\mathbf{x}$ and $\tilde{\mathbf{x}}$ be the exact solutions of (4.25) and (4.27), respectively. Then*

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \frac{\mathrm{cond}(\mathbf{A})}{1 - \mathrm{cond}(\mathbf{A})\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|}{\|\mathbf{A}\|}} \left(\frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|}{\|\mathbf{A}\|} + \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|} \right).$$

Proof. First consider the relation $\tilde{\mathbf{A}} = \mathbf{A} - (\mathbf{A} - \tilde{\mathbf{A}}) = \mathbf{A}(\mathbf{I} - \mathbf{A}^{-1}(\mathbf{A} - \tilde{\mathbf{A}}))$. Since by our assumption $\|\mathbf{A}^{-1}(\mathbf{A} - \tilde{\mathbf{A}})\| \le \|\mathbf{A}^{-1}\|\|\mathbf{A} - \tilde{\mathbf{A}}\| < 1$, Corollary 4.4 yields that $\tilde{\mathbf{A}}$ is invertible, and

$$\begin{aligned}
\|(\tilde{\mathbf{A}})^{-1}\| &\le \|(\mathbf{I} - \mathbf{A}^{-1}(\mathbf{A} - \tilde{\mathbf{A}}))^{-1}\|\|\mathbf{A}^{-1}\| \\
&\le \frac{\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}(\mathbf{A} - \tilde{\mathbf{A}})\|} \\
&\le \frac{\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}\|\|\mathbf{A} - \tilde{\mathbf{A}}\|}.
\end{aligned}$$

From equations (4.26) and (4.25) we get

$$\mathbf{x} - \tilde{\mathbf{x}} = \mathbf{x} - (\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{b}} = (\tilde{\mathbf{A}})^{-1}(\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}) = (\tilde{\mathbf{A}})^{-1}(\mathbf{b} - \tilde{\mathbf{b}} - (\mathbf{A} - \tilde{\mathbf{A}})\mathbf{x}).$$

Therefore,

$$\begin{aligned}
\|\mathbf{x} - \tilde{\mathbf{x}}\| &\le \frac{\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}\|\|\mathbf{A} - \tilde{\mathbf{A}}\|}(\|\mathbf{b} - \tilde{\mathbf{b}}\| + \|\mathbf{A} - \tilde{\mathbf{A}}\|\|\mathbf{x}\|) \\
&= \frac{\|\mathbf{A}\|\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|}{\|\mathbf{A}\|}} \left(\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{A}\|} + \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|}{\|\mathbf{A}\|}\|\mathbf{x}\| \right).
\end{aligned}$$

Dividing both sides by $\|\mathbf{x}\|$ and using relation $\|\mathbf{b}\| \le \|\mathbf{A}\|\|\mathbf{x}\|$ we get

$$\begin{aligned}
\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} &\le \frac{\mathrm{cond}(\mathbf{A})}{1 - \mathrm{cond}(\mathbf{A})\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|}{\|\mathbf{A}\|}} \left(\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{A}\|\|\mathbf{x}\|} + \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|}{\|\mathbf{A}\|} \right) \\
&\le \frac{\mathrm{cond}(\mathbf{A})}{1 - \mathrm{cond}(\mathbf{A})\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|}{\|\mathbf{A}\|}} \left(\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|} + \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|}{\|\mathbf{A}\|} \right).
\end{aligned}$$

□

The following properties of the condition number can be proved easily.

Theorem 4.24. *Let $\|\cdot\|$ be a fixed matrix norm and* $\mathrm{cond}(\cdot)$ *be the corresponding condition number function. Then*

1. $\mathrm{cond}(\mathbf{A}) \geq 1$,
2. $\rho(\mathbf{A})\rho(\mathbf{A}^{-1}) \leq \mathrm{cond}(\mathbf{A})$

hold for all invertible matrices $\mathbf{A}$.

The number $\mathrm{cond}_*(\mathbf{A}) := \rho(\mathbf{A})\rho(\mathbf{A}^{-1})$ is called the *spectral condition number* of the matrix $\mathbf{A}$. According to the previous result, the spectral condition number of a matrix is always less than any other condition number. Its disadvantage is that it is difficult to compute, since it requires the computation of eigenvalues of matrices.

We present the next result without proof.

Theorem 4.25 (Gastinel). *Let $\|\cdot\|$ be a matrix norm, $\mathbf{A}$ be invertible. Then*

$$\frac{1}{\mathrm{cond}(\mathbf{A})} = \min\left\{\frac{\|\mathbf{A}-\mathbf{B}\|}{\|\mathbf{A}\|} :\ \mathbf{B} \text{ is singular}\right\}.$$

The theorem implies that if the condition number of $\mathbf{A}$ is big, then there is a singular matrix close to $\mathbf{A}$.

An example for an ill-conditioned matrix is the so-called *Hilbert-matrix*:

$$\mathbf{H}_n = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & & & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{pmatrix}.$$

In Table 4.3 we computed the spectral condition number of the Hilbert-matrix for several values of n. We can observe that the spectral condition number (and hence all conditions numbers) increase quickly as n increases.

Exercises

1. Compute the spectral condition number of the matrix
$$\begin{pmatrix} 1 & 4 \\ 2 & -1 \end{pmatrix}.$$
2. Prove Theorem 4.24.
3. Show that
$$\mathrm{cond}_*(\mathbf{A}) = \frac{\max\{|\lambda_1|, \ldots, |\lambda_n|\}}{\min\{|\lambda_1|, \ldots, |\lambda_n|\}},$$
where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the matrix $\mathbf{A}$.

Table 4.3: Spectral condition number of the Hilbert-matrix

n	$\mathrm{cond}_*(\mathbf{H}_n)$	n	$\mathrm{cond}_*(\mathbf{H}_n)$
3	$5.24 \cdot 10^2$	7	$7.45 \cdot 10^8$
4	$1.55 \cdot 10^4$	8	$1.53 \cdot 10^{10}$
5	$4.77 \cdot 10^5$	9	$4.93 \cdot 10^{11}$
6	$1.50 \cdot 10^6$	10	$1.60 \cdot 10^{13}$

Chapter 5
Matrix Factorization

We will investigate the matrix factorization problem: for a given square matrix $\mathbf{A}$ we are looking for special matrices $\mathbf{B}$ and $\mathbf{C}$ such that $\mathbf{A} = \mathbf{BC}$. First we study the LU factorization, and then the Cholesky factorization.

5.1. LU Factorization

Let $\mathbf{A}$ be an $n \times n$ matrix. The product $\mathbf{A} = \mathbf{LU}$ is called *LU factorization* of $\mathbf{A}$ or *Doolittle's method* if $\mathbf{L}$ is lower triangular with all entries 1 in the main diagonal, and $\mathbf{U}$ is upper triangular.

Theorem 5.1. *Let $\mathbf{A}$ be a nonsingular square matrix. If the LU factorization of $\mathbf{A}$ exists, then its is unique.*

Proof. Suppose $\mathbf{A} = \mathbf{L}_1\mathbf{U}_1 = \mathbf{L}_2\mathbf{U}_2$ are two LU factorizations of the matrix $\mathbf{A}$. Since $\det(\mathbf{A}) = \det(\mathbf{L}_1)\det(\mathbf{U}_1) = \det(\mathbf{L}_2)\det(\mathbf{U}_2) \neq 0$, therefore, $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{U}_1$ and $\mathbf{U}_2$ are nonsingular matrices. Hence $\mathbf{L}_2^{-1}\mathbf{L}_1 = \mathbf{U}_2\mathbf{U}_1^{-1}$. Using Theorem 3.6, the matrix $\mathbf{L}_2^{-1}\mathbf{L}_1$ is lower triangular, and the matrix $\mathbf{U}_2\mathbf{U}_1^{-1}$ is upper triangular. Therefore, both matrices are diagonal. It is easy to see that the main diagonal of $\mathbf{L}_2^{-1}\mathbf{L}_1$ consists of only 1 entry, hence $\mathbf{L}_2^{-1}\mathbf{L}_1 = \mathbf{U}_2\mathbf{U}_1^{-1} = \mathbf{I}$, which implies that $\mathbf{L}_1 = \mathbf{L}_2$ and $\mathbf{U}_1 = \mathbf{U}_2$. □

Consider the definition of the Gaussian elimination introduced in Section 3.3. Let $l_{i1} = a_{i1}/a_{11}$, $i = 2, 3, \ldots, n$, as in Section 3.3, and define the lower triangular matrix

$$\mathbf{L}_1 := \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & & \\ -l_{31} & & 1 & & \\ \vdots & & & \ddots & \\ -l_{n1} & & & & 1 \end{pmatrix},$$

where the missing elements are all equal to 0. It is easy to check whether the product $\mathbf{L}_1\mathbf{A}$ gives the matrix $\mathbf{A}^{(1)}$, the matrix obtained performing the first elimination step of the Gaussian elimination on the coefficient matrix: $\mathbf{A}^{(1)} = \mathbf{L}_1\mathbf{A}$. Similarly, let $l_{i2} = a_{i2}^{(1)}/a_{22}^{(1)}$, $i = 3, 4, \ldots, n$, and define the matrix

$$\mathbf{L}_2 := \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & -l_{32} & 1 & & \\ & \vdots & & \ddots & \\ & -l_{n2} & & & 1 \end{pmatrix},$$

where all elements in the main diagonal are 1, in the second column the elements under the diagonal are $-l_{32}$, $-l_{42}$, ..., $-l_{n2}$, and all the other elements are 0. Then $\mathbf{A}^{(2)} = \mathbf{L}_2\mathbf{A}^{(1)}$ holds. We define the lower triangular matrices $\mathbf{L}_3, \ldots, \mathbf{L}_{n-1}$ in a similar manner. Simple computation shows

$$\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1 = \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & & \\ -l_{31} & -l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -l_{n1} & -l_{n2} & \cdots & -l_{n,n-1} & 1 \end{pmatrix}, \tag{5.1}$$

and

$$\begin{aligned} \mathbf{L} &:= (\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1)^{-1} \\ &= \mathbf{L}_1^{-1}\cdots\mathbf{L}_{n-2}^{-1}\mathbf{L}_{n-1}^{-1} \\ &= \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & 0 & 1 & & \\ \vdots & 0 & \ddots & \ddots & \\ l_{n1} & 0 & \cdots & 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & 0 & 1 & & \\ 0 & \vdots & \ddots & \ddots & \\ 0 & 0 & \cdots & l_{n,n-1} & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix}. \end{aligned} \tag{5.2}$$

Let $\mathbf{U} := \mathbf{A}^{(n-1)}$, i.e., the upper triangular matrix which is the result of the Gaussian elimination. Then $\mathbf{U} = \mathbf{L}_{n-1}\cdots\mathbf{L}_1\mathbf{A}$, which gives $\mathbf{A} = \mathbf{L}\mathbf{U}$. We have proved the following result.

Theorem 5.2. *If the Gaussian elimination can be performed on a square matrix* $\mathbf{A}$*, then the LU factorization* $\mathbf{A} = \mathbf{L}\mathbf{U}$ *exists. Then* $\mathbf{U}$ *is the upper triangular matrix obtained by the Gaussian elimination, and* $\mathbf{L}$ *is defined by (5.2), where* l_{ij} *denote the factors used in the Gaussian elimination.*

Example 5.3. Consider the coefficient matrix of Example 3.22:

$$\mathbf{A} = \begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & -1 & 2 & 4 \\ -1 & 2 & 3 & -4 \\ -2 & 1 & 4 & -2 \end{pmatrix}.$$

As we saw in Example 3.22, the Gaussian elimination can be performed on $\mathbf{A}$, and $l_{21} = 2$, $l_{31} = -1$, $l_{41} = -2$, $l_{32} = 0$, $l_{42} = -1$ and $l_{43} = 6$. If we compute the LU factorization, then we write down the Gaussian elimination so that the factors l_{ij} can be written in place of the

elements which are eliminated (changed to 0):

$$\begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & -1 & 2 & 4 \\ -1 & 2 & 3 & -4 \\ -2 & 1 & 4 & -2 \end{pmatrix} \sim \begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & 3 & 6 & 8 \\ -1 & 0 & 1 & -6 \\ -2 & -3 & 0 & -6 \end{pmatrix} \sim$$
$$\begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & 3 & 6 & 8 \\ -1 & 0 & 1 & -6 \\ -2 & -1 & 6 & 2 \end{pmatrix} \sim \begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & 3 & 6 & 8 \\ -1 & 0 & 1 & -6 \\ -2 & -1 & 6 & 38 \end{pmatrix}.$$

Now in the last matrix the elements in the main diagonal and above are the elements of the matrix $\mathbf{U}$, and the elements below the main diagonal are the entries of $\mathbf{L}$. Therefore,

$$\begin{pmatrix} 1 & -2 & -2 & -2 \\ 2 & -1 & 2 & 4 \\ -1 & 2 & 3 & -4 \\ -2 & 1 & 4 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -2 & -1 & 6 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & -2 & -2 \\ 0 & 3 & 6 & 8 \\ 0 & 0 & 1 & -6 \\ 0 & 0 & 0 & 38 \end{pmatrix},$$

which can be checked by performing the product. □

The following results can be proved easily.

Theorem 5.4. *If all the principal minors of* $\mathbf{A}$ *are nonzero, then the Gaussian elimination can be performed without row changes, and so the LU factorization* $\mathbf{A} = \mathbf{LU}$ *exists.*

Theorem 5.5. *For any invertible square matrix* $\mathbf{A}$ *there exists a permutation matrix* $\mathbf{P}$ *such that the LU factorization* $\mathbf{PA} = \mathbf{LU}$ *exists.*

If an LU factorization $\mathbf{A} = \mathbf{LU}$ is known, then we can solve linear systems with the coefficient matrix $\mathbf{A}$ efficiently. Consider the system $\mathbf{Ax} = \mathbf{b}$. We introduce the new variable $\mathbf{y} = \mathbf{Ux}$. Then the original system is equivalent to

$$\mathbf{Ly} = \mathbf{b}$$
$$\mathbf{Ux} = \mathbf{y},$$

where both systems are triangular. We solve the first equation using a forward substitution method for $\mathbf{y}$, and then the second equation using the backward substitution method for $\mathbf{x}$. It is easy to check that $n^2 + \mathcal{O}(n)$ number of multiplications/divisions are needed to solve the two triangular systems, and to compute the LU factorization, $n^3/3 + \mathcal{O}(n^2)$ number of multiplications/divisions are needed. It is especially efficient if we solve several linear system with the same coefficient matrix.

Exercises

1. Compute the LU factorization of the following matrices:

(a) $\begin{pmatrix} 2 & 3 & -1 \\ -1 & -2 & -1 \\ 0 & 2 & 4 \end{pmatrix}$ (b) $\begin{pmatrix} 4 & -1 & 2 \\ -12 & 0 & -1 \\ 8 & -17 & 26 \end{pmatrix}$

(c) $\begin{pmatrix} 1 & 3 & -1 & 2 \\ -2 & -4 & 5 & -5 \\ 0 & 6 & 6 & -2 \\ 2 & 4 & -14 & 16 \end{pmatrix}$ (d) $\begin{pmatrix} 2 & -1 & 3 & -2 \\ -8 & 5 & -7 & 7 \\ 2 & -4 & -14 & 0 \\ -4 & 7 & 23 & 4 \end{pmatrix}$

2. Show that the matrix
$$\begin{pmatrix} 2 & 2 & 3 \\ 1 & 1 & 4 \\ 1 & 0 & 1 \end{pmatrix}$$
has no LU factorization.
3. Show that the matrix
$$\begin{pmatrix} 1 & 1 & -1 \\ 2 & 2 & 2 \\ 3 & 3 & -4 \end{pmatrix}$$
has infinitely many LU factorization. Do not we get a contradiction to Theorem 5.1?
4. Prove Theorem 5.4. (Hint: Use that during the elimination steps the principal minors of $\mathbf{A}^{(k-1)}$ and $\mathbf{A}^{(k)}$ are equal. Why?)
5. Prove Theorem 5.5.
6. Solve the linear systems given in Exercise 1 of Section 3.3 using LU factorization.

5.2. Cholesky Factorization

Let $\mathbf{A}$ be a symmetric matrix. The factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ of the matrix $\mathbf{A}$, where $\mathbf{L}$ is a lower triangular matrix, is called the *Cholesky factorization.*

We note that if the Cholesky factorization exists, it is not unique. The next theorem formulates a sufficient condition for the existence of the Cholesky factorization.

Theorem 5.6. *If $\mathbf{A}$ is positive definite, then the Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ exists, the matrix $\mathbf{L}$ is real, and we can select positive elements in the main diagonal of $\mathbf{L}$.*

Proof. We prove the statement using mathematical induction with respect to the dimension of the matrix $\mathbf{A}$. The statement is obvious for 1×1 matrices. Suppose the statement of the theorem holds for $(n-1) \times (n-1)$ matrices, and let $\mathbf{A}$ be an $n \times n$ matrix. We partition the matrix $\mathbf{A}$ in the following form:
$$\mathbf{A} = \begin{pmatrix} \mathbf{X} & \mathbf{y} \\ \mathbf{y}^T & a_{nn} \end{pmatrix},$$
where $\mathbf{X}$ is an $(n-1) \times (n-1)$ matrix, $\mathbf{y}$ is an $n-1$-dimensional column vector. Theorem 3.10 yields that $\mathbf{X}$ is positive definite. We are looking for the Cholesky factorization of $\mathbf{A}$ in the form
$$\mathbf{A} = \begin{pmatrix} \mathbf{X} & \mathbf{y} \\ \mathbf{y}^T & a_{nn} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{L}} & \mathbf{0} \\ \mathbf{c}^T & d \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{L}}^T & \mathbf{c} \\ \mathbf{0}^T & d \end{pmatrix}. \tag{5.3}$$
Here $\tilde{\mathbf{L}}$ is an $(n-1)\times(n-1)$ dimensional lower triangular matrix, $\mathbf{c}$ is an $n-1$-dimensional column vector, $d \in \mathbb{R}$. If we perform the matrix multiplication on the partitioned matrices, we get the relations
$$\mathbf{X} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T, \qquad \tilde{\mathbf{L}}\mathbf{c} = \mathbf{y} \quad \text{and} \quad \mathbf{c}^T\mathbf{c} + d^2 = a_{nn}.$$
By the induction hypothesis the equation $\mathbf{X} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$ has a lower triangular solution $\tilde{\mathbf{L}} \in \mathbb{R}^{(n-1)\times(n-1)}$, where in the main diagonal we can select positive elements. This yields

that $\tilde{\mathbf{L}}$ is nonsingular, so the equation $\tilde{\mathbf{L}}\mathbf{c} = \mathbf{y}$ has a unique solution $\mathbf{c}$. Let d be a (possibly complex) root of the equation $\mathbf{c}^T\mathbf{c} + d^2 = a_{nn}$. Then relation (5.3) holds. d can be selected to be a positive real if and only if $d^2 = a_{nn} - \mathbf{c}^T\mathbf{c} > 0$. Relation (5.3) implies $\det(\mathbf{A}) = \det(\tilde{\mathbf{L}})^2 d^2$. Since $\mathbf{A}$ is positive definite, it follows $\det(\mathbf{A}) > 0$ (see Theorem 3.10). This yields that d^2 is positive, hence d can be selected to be a positive real. □

Example 5.7. Find the Cholesky factorization of the matrix

$$\begin{pmatrix} 4 & -8 & 4 \\ -8 & 17 & -11 \\ 4 & -11 & 22 \end{pmatrix}.$$

We write

$$\begin{pmatrix} 4 & -8 & 4 \\ -8 & 17 & -11 \\ 4 & -11 & 22 \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix}$$

We consider first the equation for the first row first element: $4 = l_{11}^2$. This can be solved for l_{11}: the positive solution is $l_{11} = 2$. Then we consider the elements under the main diagonal of the first column: $-8 = l_{21}l_{11}$, $4 = l_{31}l_{11}$. These can be solved uniquely for l_{21} and l_{31}: $l_{21} = -4$, $l_{31} = 2$. Now we consider the element of the main diagonal of the second column: $17 = l_{21}^2 + l_{22}^2$. Its positive solution is $l_{22} = 1$. Then look at the element in the second column under the main diagonal: $-11 = l_{31}l_{21} + l_{32}l_{22}$. This can be solved as $l_{32} = -3$. Finally, the element in the third row and third column is $22 = l_{31}^2 + l_{32}^2 + l_{33}^2$. This gives $l_{33} = 3$. We have then

$$\begin{pmatrix} 4 & -8 & 4 \\ -8 & 17 & -11 \\ 4 & -11 & 22 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ -4 & 1 & 0 \\ 2 & -3 & 3 \end{pmatrix} \begin{pmatrix} 2 & -4 & 2 \\ 0 & 1 & -3 \\ 0 & 0 & 3 \end{pmatrix}.$$

□

We can generalize the method of the previous example:

Algorithm 5.8. Cholesky factorization

INPUT: $\mathbf{A}$
OUTPUT: $\mathbf{L}$

$l_{11} \leftarrow \sqrt{a_{11}}$
for $i = 2, \ldots, n$ **do**
  $l_{i1} \leftarrow a_{i1}/l_{11}$
end do
for $j = 2, \ldots, n-1$ **do**
  $l_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$
  for $i = j+1, \ldots, n$ **do**
    $l_{ij} \leftarrow \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}\right)/l_{jj}$
  end do
end do
$l_{nn} \leftarrow \sqrt{a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2}$
output$(l_{ij}, \quad i = 1, \ldots, n,\ j = 1, \ldots, i)$

The operation count of Algorithm 5.8 is $n^3/6+n^2/2-2n/3$ number of multiplications and divisions, and $n^3/6-n/6$ number of additions and subtractions, and n number of square roots.

Exercises

1. Compute the Cholesky factorization of the following matrices:

$$\text{(a)} \begin{pmatrix} 16 & -8 & -12 \\ -8 & 8 & 4 \\ -12 & 4 & 35 \end{pmatrix}, \qquad \text{(b)} \begin{pmatrix} 4 & -2 & -4 \\ -2 & 26 & 7 \\ -4 & 7 & 6 \end{pmatrix},$$

$$\text{(c)} \begin{pmatrix} 1 & -1 & -2 & 1 \\ -1 & 10 & 2 & 2 \\ -2 & 2 & 29 & 8 \\ 1 & 2 & 8 & 7 \end{pmatrix}, \qquad \text{(d)} \begin{pmatrix} 16 & -8 & 0 & -4 \\ -8 & 5 & 1 & 3 \\ 0 & 1 & 10 & -5 \\ -4 & 3 & -5 & 7 \end{pmatrix}.$$

2. Give an example for a matrix for which the Cholesky factorization is not unique.
3. Show that the matrix
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
has no Cholesky factorization.
4. Prove that the operation count of Algorithm 5.8 is $n^3/6+n^2/2-2n/3$ number of multiplications and divisions, and $n^3/6-n/6$ number of additions and subtractions, and n number of square roots.
5. Show without using Theorem 3.10 that the matrix $\mathbf{X}$ in the proof of Theorem 5.6 is positive definite.

Chapter 6

Interpolation

Given pairwise different points x_0, x_1, ..., $x_n \in [a, b]$, the so-called *mesh points* or *node points*, and corresponding function values y_0, y_1, ..., y_n. The basic problem of interpolation is to find a function g from a certain class of functions which *interpolates* the given data, i.e., satisfies relations

$$g(x_i) = y_i, \qquad i = 0, 1, \ldots, n.$$

The geometrical meaning of the problem is to find a function g of given property whose graph goes through the points (x_i, y_i) for all $i = 0, 1, \ldots, n$.

In this chapter we first study the case when g is assumed to be a polynomial of certain order. This problem is called Lagrange interpolation. In Section 6.4 we consider a more general problem, the Hermite interpolation, when we interpolate not only function values but also derivative values. Finally, we discuss the spline interpolation.

6.1. Lagrange Interpolation

Suppose we want to interpolate given data using a polynomial of degree m of the form $g(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_m x^m$. This formula contains $m + 1$ number of parameters. In the basic problem of interpolation the conditions define $n + 1$ number of equations. It is natural to expect that the problem has a unique solution if $m = n$. We reformulate the problem: We are looking for a polynomial L_n of degree at most n which satisfies

$$L_n(x_i) = y_i, \qquad i = 0, 1, \ldots, n. \tag{6.1}$$

This problem is called *Lagrange interpolation*. We show that this problem has a unique solution. The solution L_n of this problem is called *Lagrange interpolating polynomial*, or shortly, *Lagrange polynomial*. The proof for the existence is easy: we give its formula explicitly. For $k = 0, 1, \ldots, n$ we define the polynomial of degree n by

$$l_k(x) := \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}. \tag{6.2}$$

The polynomials $l_0, \ldots, l_n$ are called *Lagrange basis polynomials of degree n*. It follows from the definition that

$$l_k(x_i) = \begin{cases} 1, & \text{if } k = i, \\ 0, & \text{if } k \neq i. \end{cases}$$

It follows that the polynomial

$$L_n(x) := \sum_{k=0}^{n} y_k l_k(x)$$

is of degree at most n, and it solves the Lagrange interpolation problem (6.1).

Now we show that the Lagrange interpolation problem (6.1) has a unique solution. Suppose L_n and $\tilde{L}_n$ are polynomials of degree at most n, and both are solutions of problem (6.1). We define the function $P(x) := L_n(x) - \tilde{L}_n(x)$. Then P is a polynomial of degree at most n, and $P(x_i) = 0$ for all $i = 0, 1, \ldots, n$, i.e., P has $n+1$ different roots. But then the Fundamental theorem of algebra yields that P is identically equal to 0, i.e., $L_n = \tilde{L}_n$. We have proved the following theorem.

Theorem 6.1. *The Lagrange interpolating problem has a unique solution which can be given by*

$$L_n(x) = \sum_{k=0}^{n} y_k \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}. \tag{6.3}$$

Example 6.2. Consider the given data

x_i	-1	1	2	3
y_i	-3	1	3	29

Find the Lagrange polynomial which interpolates the data above. Since four data points are given, the Lagrange polynomial is of degree at most three. Using formula (6.3) we get

$$\begin{aligned} L_3(x) = & -3\frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)} + \frac{(x+1)(x-2)(x-3)}{(1+1)(1-2)(1-3)} \\ & + 3\frac{(x+1)(x-1)(x-3)}{(2+1)(2-1)(2-3)} + 29\frac{(x+1)(x-1)(x-2)}{(3+1)(3-1)(3-2)} \\ = & \, 3x^3 - 6x^2 - x + 5. \end{aligned}$$

□

The values y_i associated to mesh points x_i can be considered, in general, as values of a function f at the mesh points, i.e., $y_i = f(x_i)$. For example, f can be a physical quantity which is measured at finitely many points. Or f can be a solution of a mathematical model which we solve by a numerical method, so the value of f can be computed in finitely many points, and the obtained results are numerical approximations of the solution of the model. Or f can be a function with a known formula, but its computation requires too many arithmetic operations, so we compute it exactly only at a few points. In all these cases we would possibly like to evaluate the function f at a point x which is not a mesh point. It is common to compute an interpolation polynomial L_n associated to the given data, and we use $L_n(x)$ as an approximation of the function value $f(x)$. If x is located outside the interval determined by the mesh points, we speak about *extrapolation*. We use the terminology *interpolation* if x is located between two mesh points.

Example 6.3. Consider the function $f(x) = \cos x$ on the interval $[-\pi, \pi]$. Using the mesh points π, 0 and π, and the points $-\pi$, $-\pi/2$, 0, $\pi/2$ and π we have computed the associated Lagrange interpolating polynomials L_2 and L_4. The polynomials and the graph of the function f can be seen in Figure 6.1. We can observe that in the case of 5 mesh points we get a better approximation of f than using only 3 mesh points. It is also clear from the figure that outside the interval $[-\pi, \pi]$ the Lagrange polynomials are not close to the function f. □

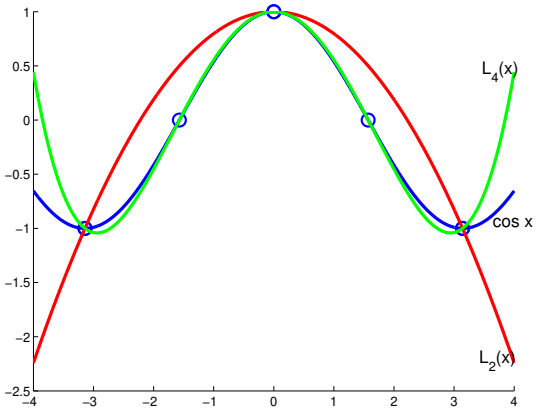

Figure 6.1: Lagrange interpolation of the function $\cos x$ using the mesh points $-\pi, 0, \pi$ and the mesh points $-\pi, -\pi/2, 0, \pi/2, \pi$, respectively

For the proof of Theorem 6.5 below we will need the following result.

Theorem 6.4 (Generalized Rolle's Theorem). *Let $f \in C^n[a,b]$, $a \le x_0 < x_1 \cdots < x_n \le b$, and suppose $f(x_0) = f(x_1) = \cdots = f(x_n) = 0$. Then there exists $\xi \in (x_0, x_n)$ such that $f^{(n)}(\xi) = 0$.*

Proof. Using the assumptions $f(x_0) = f(x_1) = 0$, Rolle's Theorem (Theorem 2.3) yields that there exists $\eta_1 \in (x_0, x_1)$ such that $f'(\eta_1) = 0$. Similarly, using Rolle's Theorem for the intervals $[x_1, x_2]$, ..., $[x_{n-1}, x_n]$ we get that there exist numbers $\eta_2 \in (x_1, x_2)$, ..., $\eta_n \in (x_{n-1}, x_n)$ such that $f'(\eta_2) = \cdots = f'(\eta_n) = 0$. Consider then the intervals $[\eta_1, \eta_2]$, ..., $[\eta_{n-1}, \eta_n]$. Since at the end points of the intervals we have $f'(\eta_i) = 0$, Rolle's Theorem implies that there exist numbers $\theta_2 \in (\eta_1, \eta_2)$, ..., $\theta_n \in (\eta_{n-1}, \eta_n)$ for which $f''(\theta_2) = \cdots = f''(\theta_n) = 0$. Applying again Rolle's Theorem we get that the third derivative of f has zeros at $n-2$ points, the fourth derivative of f vanishes at $n-3$ points, etc., $f^{(n)}$ is zero at a point ξ. □

Theorem 6.5. *Let $f \in C^{n+1}[a,b]$, $x_i \in [a,b]$ $(i = 0, \ldots, n)$ be pairwise distinct mesh points and $y_i = f(x_i)$ $(i = 0, \ldots, n)$. Let $L_n(x)$ be the corresponding Lagrange interpolating polynomial. Then for every $x \in [a,b]$ there exists $\xi = \xi(x) \in \langle x, x_0, x_1, \ldots, x_n \rangle$ such that*

$$f(x) = L_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n).$$

Proof. If $x = x_i$ for some i, then the statement is obviously satisfied. Fix a number $x \in (a, b)$ such that $x \neq x_i$ for all $i = 0, \dots, n$, and consider the function

$$g(t) := f(t) - L_n(t) - \frac{(t - x_0) \cdots (t - x_n)}{(x - x_0) \cdots (x - x_n)} (f(x) - L_n(x)).$$

Clearly, $g \in C^{n+1}$, and $g(x) = g(x_0) = g(x_1) = \cdots = g(x_n) = 0$. Then the generalized Rolle's Theorem (Theorem 6.4) yields that there exists a number $\xi \in \langle x, x_0, \dots, x_n \rangle$ such that $g^{(n+1)}(\xi) = 0$. Since L_n is a polynomial of degree at most n, its $(n+1)$-st order derivative is identically 0, so

$$g^{(n+1)}(t) = f^{(n+1)}(t) - \frac{(n+1)!}{(x - x_0) \cdots (x - x_n)} (f(x) - L_n(x)).$$

This gives the statement with $t = \xi$. □

Now we consider the case when the mesh points are equidistant, i.e., $x_i = x_0 + ih$. Theorem 6.5 yields that the truncation error of the interpolation can be estimated by

$$|f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0) \cdots (x - x_n)|, \tag{6.4}$$

where $M_{n+1} = \sup\{|f^{(n+1)}(t)| \colon t \in [x_0, x_n]\}$. Suppose $x \in (x_k, x_{k+1})$ for some $0 \leq k < n$. Then we have

$$|(x - x_k)(x - x_{k+1})| \leq \frac{h^2}{4},$$

and so

$$\begin{aligned}
\prod_{i=0}^{n} |x - x_i| &\leq \frac{h^2}{4} \prod_{i=0}^{k-1} (x - x_i) \prod_{i=k+2}^{n} (x_i - x) \\
&\leq \frac{h^2}{4} \prod_{i=0}^{k-1} (x_{k+1} - x_i) \prod_{i=k+2}^{n} (x_i - x_k) \\
&= \frac{h^{n+1}}{4} \prod_{i=0}^{k-1} (k + 1 - i) \prod_{i=k+2}^{n} (i - k) \\
&= \frac{h^{n+1}}{4} (k+1)!(n-k)! \\
&\leq \frac{h^{n+1}}{4} n!
\end{aligned}$$

(See Exercise 4.) This and (6.4) imply the next result.

Theorem 6.6. *Let $f \in C^{n+1}[a, b]$, $x_i = a + i(b - a)/n$ $(i = 0, \dots, n)$ and $y_i = f(x_i)$ $(i = 0, \dots, n)$. Let $x \in [a, b]$. Then*

$$|f(x) - L_n(x)| \leq \frac{M_{n+1}}{4(n+1)} \left(\frac{b - a}{n} \right)^{n+1},$$

where $M_{n+1} := \sup\{|f^{(n+1)}(x)| \colon x \in [a,b]\}$.

Example 6.7. Consider again Example 6.3. According to the previous theorem it follows for $x \in [-\pi, \pi]$

$$|f(x) - L_2(x)| \le \frac{1}{12}\pi^3 \approx 2.5839 \qquad \text{and} \qquad |f(x) - L_4(x)| \le \frac{1}{20}\left(\frac{\pi}{2}\right)^5 \approx 0.4782.$$

Certainly, Theorem 6.6 gives an upper estimate of the truncation error. Figure 6.1 shows that the actual error can be significantly smaller. □

The next result will be used in Chapter 7. We state the theorem without giving its proof.

Theorem 6.8. *Suppose* $f \in C^{n+2}[a,b]$, $a = x_0 < \cdots < x_n = b$, *and let*

$$\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\cdots(x - x_n)$$

be the truncation error of the Lagrange interpolation of degree n. *Then the function* $x \mapsto f^{(n+1)}(\xi(x))$ *can be extended continuously for* $x = x_i$, *and it is differentiable for all* $x \ne x_i$, *and*

$$\frac{d}{dx}f^{(n+1)}(\xi(x)) = \frac{1}{n+2}f^{(n+2)}(\eta(x)),$$

where $\eta(x) \in \langle x_0, \ldots, x_n, x\rangle$, *moreover,* $\dfrac{d}{dx}f^{(n+1)}(\xi(x))$ *can be extended continuously for* $x = x_i$ $(i = 0, 1, \ldots, n)$.

Newt we discuss the problem of interpolation for functions of two variables. We consider only the easiest case, we assume the function f is defined on a rectangular domain. Let $f \colon [a,b] \times [c,d] \to \mathbb{R}$, and consider the division of the intervals $[a,b]$ and $[c,d]$ by $a = x_0 < x_1 < \ldots < x_n = b$ and $c = y_0 < y_1 < \ldots < y_m = d$. Let $z_{ij} = f(x_i, y_j)$, $i = 0, \ldots, n$, $j = 0, \ldots, m$. We define the following two-variable polynomial to interpolate the given data:

$$L_{n,m}(x,y) := \sum_{i=0}^{n}\sum_{j=0}^{m} z_{ij} l_i(x)\tilde{l}_j(y), \tag{6.5}$$

where l_i and $\tilde{l}_j$ are the Lagrange basis polynomials of degree n and m, respectively, corresponding to the mesh points $a = x_0 < x_1 < \ldots < x_n = b$ and $c = y_0 < y_1 < \ldots < y_m = d$ defined by (6.2). The function $L_{n,m}$ satisfies $L_{n,m}(x_i, y_j) = z_{ij}$ for all i, j. If x is fixed, then $L_{n,m}(x, \cdot)$ is a polynomial of degree at most m. Conversely, if y is fixed, then $L_{n,m}(\cdot, y)$ is a polynomial of degree at most n. The problem above is called *two-dimensional Lagrange interpolation* or *bivariate Lagrange interpolation* or *Lagrange interpolation of two variables.*

Example 6.9. Consider the following given function values:

(x_i, y_j)	$(0,0)$	$(1,0)$	$(2,0)$	$(0,2)$	$(1,2)$	$(2,2)$
z_{ij}	2	-1	1	1	0	2

Applying formula (6.5) we get the two-variable polynomial

$$\begin{aligned} L_{2,1}(x,y) = 2\frac{(x-1)(x-2)}{(0-1)(0-2)}\frac{y-2}{0-2} - \frac{x(x-2)}{1(1-2)}\frac{y-2}{0-2} + \frac{x(x-1)}{2(2-1)}\frac{y-2}{0-2} \\ + \frac{(x-1)(x-2)}{(0-1)(0-2)}\frac{y}{2} + 0\frac{x(x-2)}{1(1-2)}\frac{y}{2} + 2\frac{x(x-1)}{2(2-1)}\frac{y}{2} \\ = -\frac{1}{2}x^2y + \frac{5}{2}x^2 + \frac{3}{2}xy - \frac{11}{2}x - \frac{1}{2}y + 2. \end{aligned}$$

This is of second order in x, and first order in y. The graph of the polynomial can be seen in Figure 6.2. □

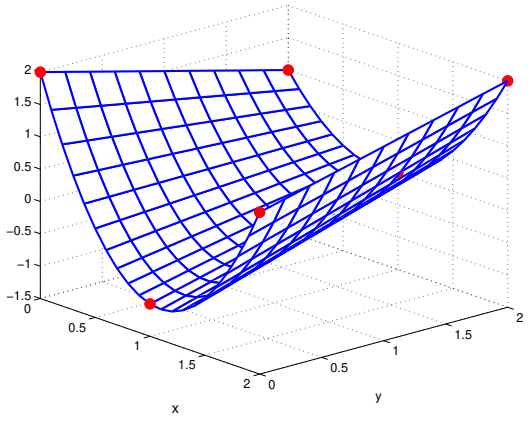

Figure 6.2: Bivariate Lagrange interpolation

Exercises

1. Compute and plot the graph of the Lagrange polynomials corresponding to the following data, and find the value of the Lagrange polynomial at $x = 1$:

(a)

x_i	-1	0	2	4
y_i	3	-2	4	-2

(b)

x_i	0.1	0.4	1.3	2.5	2.8
y_i	1.2	0.2	-2.2	3.1	1.3

(c)

x_i	-0.5	0.0	1.5	2.0	3.0	3.5
y_i	-0.5	1.5	3.5	2.0	2.5	6.5

2. Show, without giving the formula of the Lagrange polynomial, that the system (6.1) has a unique solution.

3. Let $l_i(x)$ $(i = 0, 1, \ldots, n)$ be defined by (6.2). Show that for all x

$$\sum_{i=0}^{n} l_i(x) = 1.$$

4. Prove that $(k+1)!(n-k)! \leq n!$ for all $k = 0, 1, \ldots, n-1$.

5. What is the smallest positive integer n for which the function $\cos x$ can be approximated by the Lagrange polynomial $L_n(x)$ for all $x \in [-\pi, \pi]$ with an error smaller than 0.001, assuming we use equidistant mesh points on the interval $[-\pi, \pi]$?

6. Give the two-dimensional Lagrange interpolating polynomial $L_{2,2}$ corresponding to the given data:

(x_i, y_j)	$(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$	$(2,0)$	$(2,1)$	$(2,2)$
z_{ij}	3	1	0	2	-1	0	2	3	1

6.2. Divided Differences

Given a function $f\colon [a, b] \to \mathbb{R}$ and pairwise different mesh points $x_i \in [a, b]$ $(i = 0, \ldots, n)$. Then the *zeroth divided difference* of the function f at the point x_0 is defined by $f[x_0] := f(x_0)$. The *first divided difference* of the function f at the points x_0, x_1 is the number

$$f[x_0, x_1] := \frac{f[x_1] - f[x_0]}{x_1 - x_0},$$

(i.e., $f[x_0, x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0}$). In general, the *nth divided difference* of the function f relative to the points $x_0, x_1, \ldots, x_n$ is defined by

$$f[x_0, x_1, \ldots, x_n] := \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}.$$

We note that we have not assumed the mesh points are ordered increasingly.

Theorem 6.10. *Let x_i $(i = 0, 1, \ldots, n)$ be pairwise different mesh points. Then*

$$f[x_0, x_1, \ldots, x_n] = \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}.$$

Proof. We prove the statement using mathematical induction with respect to n. For $n = 0$ the statement is obvious. (In this case in the denominator we have the "empty product", which, by definition, equals to 1.) Suppose the statement holds for n, and consider the $(n+1)$-st divided difference $f[x_0, x_1, \ldots, x_{n+1}]$. The definition of the divided

difference, the inductive hypothesis and some calculations yield

$$\begin{aligned}
f[x_0, x_1, \ldots, x_{n+1}] &= \frac{f[x_1, x_2, \ldots, x_{n+1}] - f[x_0, x_1, \ldots, x_n]}{x_{n+1} - x_0} \\
&= \frac{1}{x_{n+1} - x_0} \left\{ \sum_{i=1}^{n+1} \frac{f(x_i)}{(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_{n+1})} \right. \\
&\quad \left. - \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \right\} \\
&= \frac{1}{x_{n+1} - x_0} \left\{ \frac{f(x_{n+1})}{(x_{n+1} - x_1) \cdots (x_{n+1} - x_n)} - \frac{f(x_0)}{(x_0 - x_1) \cdots (x_0 - x_n)} \right. \\
&\quad + \sum_{i=1}^{n} \frac{f(x_i)}{(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \\
&\quad \left. \cdot \left(\frac{1}{x_i - x_{n+1}} - \frac{1}{x_i - x_0} \right) \right\} \\
&= \sum_{i=0}^{n+1} \frac{f(x_i)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_{n+1})},
\end{aligned}$$

which proves the statement. □

The previous result has some immediate consequences.

Corollary 6.11. *The divided differences are independent of the order of the mesh points.*

Corollary 6.12. *If the function f is continuous, then the divided differences depend continuously on the mesh points.*

Suppose f is differentiable. Then the function $x_1 \mapsto f[x_0, x_1]$ is continuous for $x_1 \neq x_0$. Now compute the limit $\lim_{x_1 \to x_0} f[x_0, x_1]$. Using the definition of the first divided difference and the differentiability of the function we get

$$\lim_{x_1 \to x_0} f[x_0, x_1] = \lim_{x_1 \to x_0} \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(x_0).$$

Therefore, we define the first divided difference relative to equal mesh points by

$$f[x_0, x_0] := f'(x_0).$$

With this definition the function $x_1 \mapsto f[x_0, x_1]$ is extended continuously for $x_1 = x_0$. Higher order divided differences with equal mesh points will be defined in Exercises 6 and 7 of the next section.

Exercises

1. Compute the following divided differences:

   (a) $f[x_0, x_1, x_2, x_3]$, where $x_i = i$, $f(x) = x^2$,

   (b) $f[x_0, x_1, x_2]$, where $x_i = 0.2i$, $f(x) = \sin x$,

   (c) $f[x_0, x_0]$, where $x_0 = 0$, $f(x) = \sin x$.

2. Let $f \in C^1[a, b]$, and $x_0, x_1 \in (a, b)$, $x_0 \neq x_1$. Show that there exists $\xi \in \langle x_0, x_1 \rangle$ such that
$$f[x_0, x_1] = f'(\xi).$$

3. Let $x_0 < x_1 < x_2 < x_3$ and
$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2).$$
Show that
$$a_0 = P[x_0], \quad a_1 = P[x_0, x_1], \quad a_2 = P[x_0, x_1, x_2], \quad \text{and} \quad a_3 = P[x_0, x_1, x_2, x_3].$$

6.3. Newton's Divided Difference Formula

The disadvantage of formula (6.3) is that if we add an additional mesh point, then the whole formula (6.3) must be recomputed. In this section we define a new formula for the Lagrange polynomial, and in this form it will be easy to add a new mesh point to the formula.

Suppose function values $y_i = f(x_i)$ are given for $i = 0, 1, \ldots, n$. First consider the relation

$$L_n(x) = L_0(x) + (L_1(x) - L_0(x)) + (L_2(x) - L_1(x)) + \cdots + (L_n(x) - L_{n-1}(x)).$$

By definition, $L_0(x) = f(x_0)$. Consider the difference $L_i(x) - L_{i-1}(x)$. It is a polynomial of degree at most i, and since L_i and L_{i-1} both satisfy the interpolating equations at x_0, …, x_{i-1}, we have $L_i(x_j) - L_{i-1}(x_j) = f(x_j) - f(x_j) = 0$ $(j = 0, 1, \ldots, i-1)$. But then the Fundamental Theorem of Algebra yields

$$L_i(x) - L_{i-1}(x) = a_i(x - x_0)(x - x_1) \cdots (x - x_{i-1}),$$

where $a_i \in \mathbb{R}$. If we substitute $x = x_i$ into this relation and use for $L_{i-1}(x_i)$ the formula (6.3), we get

$$f(x_i) - \sum_{k=0}^{i-1} f(x_k) \frac{(x_i - x_0) \cdots (x_i - x_{k-1})(x_i - x_{k+1}) \cdots (x_i - x_{i-1})}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_{i-1})} = a_i(x_i - x_0) \cdots (x_i - x_{i-1}).$$

So from this we get for a_i that

$$\begin{aligned} a_i &= \frac{f(x_i)}{(x_i-x_0)\cdots(x_i-x_{i-1})} - \frac{1}{(x_i-x_0)\cdots(x_i-x_{i-1})} \\ &\quad \cdot \sum_{k=0}^{i-1} f(x_k) \frac{(x_i-x_0)\cdots(x_i-x_{k-1})(x_i-x_{k+1})\cdots(x_i-x_{i-1})}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_{i-1})} \\ &= \sum_{k=0}^{i} \frac{f(x_k)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_i)} \\ &= f[x_0, x_1, \ldots, x_i]. \end{aligned}$$

Therefore, the Lagrange interpolating polynomial can be written as

$$\begin{aligned} L_n(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + \cdots \\ + \ f[x_0,x_1,\ldots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}). \end{aligned} \tag{6.6}$$

We have to emphasize that this is the same polynomial as (6.3), only it is given by a different formula. The polynomial given by (6.6) is called *Newton's divided difference formula* or shortly *Newton polynomial*.

The advantage of formula (6.6) compared to (6.3) can be seen immediately. It is easy to add a new mesh point to the formula, we have the simple correction term:

$$L_{n+1}(x) = L_n(x) + f[x_0, x_1, \ldots, x_{n+1}](x-x_0)\cdots(x-x_n).$$

Another advantage is that a polynomial of the form (6.6) can be easily evaluated using the Horner's method. Furthermore, the degree of the polynomial can be determined in this form easily. If, for example, $f[x_0, x_1, \ldots, x_n] \neq 0$, then the polynomial is of degree n. In Algorithm 6.13 we present the computation of the coefficients of the Newton polynomial, i.e., the values $a_i = f[x_0, \ldots, x_i]$. In Algorithm 6.14 we formulate a method to evaluate the Newton polynomial using Horner's method.

Algorithm 6.13. Computation of the coefficients of the Newton polynomial

INPUT: n - number of mesh points $-$ 1
x_i, $(i = 0, 1, \ldots, n)$ - mesh points
y_i, $(i = 0, 1, \ldots, n)$ - function values
OUTPUT: a_i, $(i = 0, 1, \ldots, n)$ - coefficients of the Newton polynomial, where a_i is the coefficient of the ith-order term

for $i = 0, 1, \ldots, n$ **do**
    $a_i \leftarrow y_i$
end do
for $j = 1, 2, \ldots, n$ **do**
    for $i = n, n-1, \ldots, j$ **do**
        $a_i \leftarrow (a_i - a_{i-1})/(x_i - x_{i-j})$
    end do
end do
output$(a_0, a_1, \ldots, a_n)$

Note that Algorithm 6.13 was organized so that only those divided differences are stored by the end of the algorithm which are needed for the Newton polynomial.

Algorithm 6.14. Evaluation of the Newton polynomial

INPUT: n - number of mesh points $-$ 1
x_i, $(i = 0, 1, \ldots, n)$ - mesh points
a_i, $(i = 0, 1, \ldots, n)$ - coefficients of the Newton polynomial
x - the value where we evaluate the Newton polynomial
OUTPUT: y - function value of the Newton polynomial at x

$y \leftarrow a_n$
for $i = n-1, n-2, \ldots, 0$ **do**
  $y \leftarrow y(x - x_i) + a_i$
end do
output(y)

When we do the computation of the divided differences by hand, it is recommended to list the values of the divided differences in a triangular table as it can be seen in Table 6.1. The numbers in the first two columns are the input data, the rest of the numbers must be computed: a number is obtained so that we take the difference of the number to the left and above, and it is divided by the difference of the appropriate mesh points x_k. The numbers in frames in the diagonal of the table give the coefficients of the Newton polynomial in (6.6).

Table 6.1: Computation of the divided differences by hand

x_0	$\boxed{f(x_0)}$				
x_1	$f(x_1)$	$\boxed{f[x_0, x_1]}$			
x_2	$f(x_2)$	$f[x_1, x_2]$	$\boxed{f[x_0, x_1, x_2]}$		
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$\cdots$	$\boxed{f[x_0, x_1, \ldots, x_n]}$

Example 6.15. Consider again Example 6.2. We compute $L_3(x)$ in Newton's divided difference form, and we evaluate $L_3(0)$. First we compute the table of divided differences:

$$\begin{array}{ccccc} -1 & -3 & & & \\ 1 & 1 & 2 & & \\ 2 & 3 & 2 & 0 & \\ 3 & 29 & 26 & 12 & 3 \end{array}$$

This yields that

$$L_3(x) = -3 + 2(x+1) + 3(x+1)(x-1)(x-2),$$

and so $L_3(0) = -3 + 2 \cdot 1 + 3 \cdot 1(-1)(-2) = 5$. We can simplify this formula of L_3 and we get the same form of the polynomial as in Example 6.2: $L_3(x) = 3x^3 - 6x^2 - x + 5$.

□

Next we study again the truncation error of the interpolation. In Section 6.1 we obtained that it has the form $\frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n)$. This is certainly the same for the Newton's divided difference form of the interpolating polynomial, but here we give a different form of the same truncation error.

Theorem 6.16. *Let $x_i \in (a, b)$ $(i = 0, \ldots, n)$ be pairwise different mesh points and $y_i = f(x_i)$ $(i = 0, \ldots, n)$. Let $L_n(x)$ be the corresponding nth degree Lagrange interpolating polynomial. Then*

$$f(x) = L_n(x) + f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n).$$

Proof. Fix $x \in (a, b)$ which is different from each mesh points. (If $x = x_i$ for some i, then the statement is clearly true.) Add x to the mesh points together with the function value $f(x)$. Let L_{n+1} be the Lagrange interpolating polynomial corresponding to the extended data set. Then we have

$$L_{n+1}(t) = L_n(t) + f[x_0, x_1, \ldots, x_n, x](t - x_0)\cdots(t - x_n).$$

Now substitution $t = x$ proves the statement, since $f(x) = L_{n+1}(x)$. □

This form of the truncation error has no practical importance, since in order to compute $f[x_0, \ldots, x_n, x]$ the exact value of $f(x)$ is needed. But its consequence is important. Comparing it to Theorem 6.5 we get the following result.

Corollary 6.17. *If $f \in C^n[a, b]$ and x_i $(i = 0, \ldots, n)$ are pairwise different mesh points, then there exists $\xi \in \langle x_0, x_1, \ldots, x_n \rangle$ such that*

$$f[x_0, x_1, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi).$$

Exercises

1. Repeat Exercise 1 of Section 6.1 using the Newton's divided difference form of the Lagrange interpolating polynomial.
2. Show that if P is a polynomial of degree n, then
$$P(x) = \sum_{i=0}^{n} P[x_0, \ldots, x_i] \prod_{k=0}^{i-1} (x - x_k).$$
3. Let $x_0, \ldots, x_n$ be pairwise different numbers. Show that if P is a polynomial of degree n, then $P[x_0, \ldots, x_m] = 0 \quad$ for all $m > n$.

4. Prove that if $f(x) = c_0 + c_1 x + \cdots + c_n x^n$, then $c_n = f[x_0, x_1, \ldots, x_n]$.
5. Prove that
$$f[x_0, x_1, \ldots, x_n] = \frac{\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} & f(x_0) \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} & f(x_1) \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} & f(x_n) \end{vmatrix}}{\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} & x_n^n \end{vmatrix}}.$$
6. Show that
$$\lim_{(x_1, x_2, \ldots, x_n) \to (x_0, x_0 \ldots, x_0)} f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(x_0)}{n!}.$$
(Hint: Use Corollary 6.17.)
7. Let $f \in C^2$. Define the following divided differences:
$$f[x_0, x_0, x_1] := \lim_{x_2 \to x_0} f[x_0, x_2, x_1], \quad f[x_0, x_1, x_0] := \lim_{x_2 \to x_0} f[x_0, x_1, x_2],$$
and
$$f[x_1, x_0, x_0] := \lim_{x_2 \to x_0} f[x_1, x_0, x_2], \qquad f[x_0, x_0, x_0] = \frac{f''(x_0)}{2}.$$
Show that the limits above exist, and the second divided differences satisfy:
   (a) $f[x_0, x_0, x_1] = \dfrac{f[x_0, x_1] - f[x_0, x_0]}{x_1 - x_0}$,
   (b) $f[x_1, x_0, x_0] = \dfrac{f[x_0, x_0] - f[x_1, x_0]}{x_0 - x_1}$,
   (c) $f[x_0, x_0, x_1] = f[x_0, x_1, x_0] = f[x_1, x_0, x_0]$,
   (d) $\lim\limits_{(x_1, x_2) \to (x_0, x_0)} f[x_0, x_1, x_2] = f[x_0, x_0, x_0]$,
   (e) There exists $\xi \in \langle x_0, x_1 \rangle$ such that $f[x_0, x_0, x_1] = f''(\xi)/2$.
8. Check that Algorithm 6.13 gives back the coefficients of the Newton polynomial.

6.4. Hermite Interpolation

In this section we generalize the basic problem of interpolation. Let f be a differentiable function, and given mesh points x_i $(i = 0, \ldots, n)$. The so-called *Hermite interpolation* asks to find a polynomial $g(x) = c_0 + c_1 x + \cdots + c_m x^m$ which interpolates not only the function values $y_i = f(x_i)$, but also the derivative values $y_i' := f'(x_i)$. Therefore, we are looking for a polynomial g of degree m which satisfies the interpolation conditions

$$g(x_i) = y_i, \qquad g'(x_i) = y_i', \qquad i = 0, 1, \ldots, n.$$

The geometrical meaning of this problem is that the graph of g goes through the given points (x_i, y_i) in a way that the tangent line of the graph at x_i has a slope equal to the value y_i'. In the formula of the polynomial g there are $m+1$ parameters, and the interpolation conditions specify $2(n+1)$ conditions. So we expect that the Hermite interpolation problem has a unique solution in the class of polynomials with degree at most $m = 2n+1$. The next theorem will prove this result. The solution of the Hermite interpolation problem is called *Hermite interpolating polynomial* or shortly *Hermite polynomial*, and it is denoted by H_{2n+1}.

In the next theorem we will use higher order divided differences where two consecutive mesh points can be equal: $f[x_0, x_0, x_1, x_1, \dots, x_n, x_n]$, where $x_0, \dots, x_n$ are pairwise different mesh points. Its definition is the usual recursion:

$$f[x_0, x_0, x_1, x_1, \dots, x_n, x_n] = \frac{f[x_0, x_1, x_1, \dots, x_n, x_n] - f[x_0, x_0, x_1, x_1, \dots, x_n]}{x_n - x_0}.$$

The divided difference with lower orders are defined in a similar manner until we get first divided differences with different or equal mesh points. Both are already defined in Section 6.2.

Theorem 6.18. *The Hermite interpolation problem has a unique solution in the class of polynomials with degree at most $(2n+1)$, which is given by*

$$\begin{aligned} H_{2n+1}(x) = {} & f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_1](x - x_0)^2 \\ & + f[x_0, x_0, x_1, x_1](x - x_0)^2(x - x_1) + f[x_0, x_0, x_1, x_1, x_2](x - x_0)^2(x - x_1)^2 \\ & + f[x_0, x_0, x_1, x_1, x_2, x_2](x - x_0)^2(x - x_1)^2(x - x_2) + \cdots \\ & + f[x_0, x_0, x_1, x_1, \dots, x_n, x_n](x - x_0)^2(x - x_1)^2 \cdots (x - x_{n-1})^2(x - x_n). \end{aligned} \tag{6.7}$$

Moreover, the truncation error is

$$f(x) - H_{2n+1}(x) = f[x_0, x_0, \dots, x_n, x_n, x](x - x_0)^2 \cdots (x - x_n)^2. \tag{6.8}$$

Proof. First we discuss the uniqueness of the Hermite polynomial. Suppose H_{2n+1} and $\tilde{H}_{2n+1}$ are polynomials of degree at most $(2n+1)$ which both satisfy the equations of the Hermite interpolation problem. Then $P := H_{2n+1} - \tilde{H}_{2n+1}$ is a polynomial of degree at most $(2n+1)$ which satisfies $P(x_i) = H_{2n+1}(x_i) - \tilde{H}_{2n+1}(x_i) = f(x_i) - f(x_i) = 0$ and $P'(x_i) = H'_{2n+1}(x_i) - \tilde{H}'_{2n+1}(x_i) = f'(x_i) - f'(x_i) = 0$, i.e., x_i is a double root of P for all $i = 0, 1, \dots, n$. Hence P has $2(n+1) = 2n+2$ number of roots, and hence the Fundamental Theorem of Algebra yields that P is identically equal to 0, since the degree of P is at most $(2n+1)$. This implies that if the solution of the Hermite interpolation problem exists, it has to be unique.

Now we show that the polynomial H_{2n+1} defined by (6.7) is a solution of the Hermite interpolation problem, and satisfies the error formula (6.9) too. Direct computation gives that $H_{2n+1}(x_0) = f(x_0)$ and $H'_{2n+1}(x_0) = f[x_0, x_0] = f'(x_0)$. Next we show that $H_{2n+1}(x_1) = f(x_1)$ and $H'_{2n+1}(x_1) = f'(x_1)$ hold too. To prove this, select numbers $\tilde{x}_i$ close to x_i so that the numbers $\{x_i, \tilde{x}_i \colon i = 0, 1, \dots, n\}$ be pairwise different, and let L_{2n+1}

be the Lagrange polynomial interpolating the function values of f at these mesh points. Then

$$\begin{aligned} L_{2n+1}(x) = {} & f[x_0] + f[x_0, x_0'](x - x_0) + f[x_0, x_0', x_1](x - x_0)(x - x_0') \\ & + \; f[x_0, x_0', x_1, x_1'](x - x_0)(x - x_0')(x - x_1) + \cdots \\ & + \; f[x_0, x_0', x_1, x_1', \ldots, x_n, x_n'](x - x_0)(x - x_0') \cdots (x - x_{n-1}) \\ & \quad \cdot (x - x_{n-1}')(x - x_n), \end{aligned}$$

and

$$f(x) = L_{2n+1}(x) + f[x_0, x_0', \ldots, x_n, x_n', x](x - x_0)(x - x_0') \cdots (x - x_n)(x - x_n').$$

The definition of L_{2n+1} and H_{2n+1} and the continuity of the divided difference (see Exercise 3) yield for all x that

$$L_{2n+1}(x) \to H_{2n+1}(x) \quad \text{as } (x_0', x_1', \ldots, x_n') \to (x_0, x_1, \ldots, x_n), \tag{6.9}$$

and so

$$f(x) = H_{2n+1}(x) + f[x_0, x_0, x_1, x_1, \ldots, x_n, x_n, x](x - x_0)^2(x - x_1)^2 \cdots (x - x_n)^2.$$

This proves relation (6.8). It follows from the uniqueness of the Lagrange polynomial that if we interchange x_0, x_0' and x_1, x_1', then the interpolating polynomial remains the same, so

$$\begin{aligned} L_{2n+1}(x) = {} & f[x_1] + f[x_1, x_1'](x - x_1) + f[x_1, x_1', x_0](x - x_1)(x - x_1') \\ & + f[x_1, x_1', x_0, x_0'](x - x_1)(x - x_1')(x - x_0) + \cdots \\ & + f[x_1, x_1', x_0, x_0', x_2, x_2' \ldots, x_n, x_n'](x - x_1)(x - x_1')(x - x_0)(x - x_0') \\ & \quad \cdot (x - x_2)(x - x_2') \cdots \; (x - x_{n-1})(x - x_{n-1}')(x - x_n). \end{aligned}$$

But then taking the limit $(x_0', x_1', \ldots, x_n') \to (x_0, x_1, \ldots, x_n)$ of both sides, and using relation (6.9), we get

$$\begin{aligned} H_{2n+1}(x) = {} & f[x_1] + f[x_1, x_1](x - x_1) + f[x_1, x_1, x_0](x - x_1)^2 \\ & + \; f[x_1, x_1, x_0, x_0](x - x_1)^2(x - x_0) + f[x_1, x_1, x_0, x_0, x_2](x - x_1)^2(x - x_0)^2 \\ & + \; f[x_1, x_1, x_0, x_0, x_2, x_2](x - x_1)^2(x - x_0)^2(x - x_2) + \cdots \\ & + \; f[x_1, x_1, x_0, x_0, x_2, x_2, \ldots, x_n, x_n](x - x_1)^2(x - x_0)^2(x - x_2)^2 \\ & \quad \cdots (x - x_{n-1})^2(x - x_n). \end{aligned}$$

But from this form it is clear that $H_{2n+1}(x_1) = f(x_1)$ and $H_{2n+1}'(x_1) = f'(x_1)$. In a similar manner we can show that $H_{2n+1}(x_i) = f(x_i)$ and $H_{2n+1}'(x_i) = f'(x_i)$ hold for $i = 2, 3, \ldots, n$. □

Theorem 6.19. *Let $f \in C^{2n+2}$. Then there exists $\xi \in \langle x_0, x_1, \ldots, x_n, x \rangle$ such that*

$$f(x) - H_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x - x_0)^2 \cdots (x - x_n)^2.$$

Proof. The proof is similar to that of Theorem 6.5. Let x be a number different from all mesh points, and define the function

$$g(z) := f(z) - H_{2n+1}(z) - \frac{(z-x_0)^2 \cdots (z-x_n)^2}{(x-x_0)^2 \cdots (x-x_n)^2}(f(x) - H_{2n+1}(x)).$$

Clearly, $g \in C^{2n+2}$, and $x_0, \ldots, x_n$ are all double roots, and x is a simple root of g. Therefore, the generalized Rolle's Theorem (Theorem 6.4) implies that there exists $\xi \in \langle x_0, x_1, \ldots, x_n, x\rangle$ such that $g^{(2n+2)}(\xi) = 0$. This yields the statement of the theorem. □

Comparing relations (6.8) and Theorem 6.19 we get the next result.

Corollary 6.20. *Suppose $f \in C^{2n+2}$, and $x, x_0, \ldots, x_n$ are pairwise different numbers. Then there exists $\xi \in \langle x_0, x_1, \ldots, x_n, x\rangle$ such that*

$$f[x_0, x_0, \ldots, x_n, x_n, x] = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}.$$

Table 6.2: Table of divided differences for the Hermite polynomial

x_0	$\boxed{f(x_0)}$				
x_0	$f(x_0)$	$\boxed{f[x_0, x_0]}$			
x_1	$f(x_1)$	$f[x_0, x_1]$	$\boxed{f[x_0, x_0, x_1]}$		
x_1	$f(x_1)$	$f[x_1, x_1]$	$f[x_0, x_1, x_1]$	$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-1}, x_{n-1}, x_n]$	$\cdots$	
x_n	$f(x_n)$	$f[x_n, x_n]$	$f[x_{n-1}, x_n, x_n]$	$\cdots$	$\boxed{f[x_0, x_0, x_1, x_1 \ldots, x_n, x_n]}$

When we compute the divided differences required in formula (6.8), we list the numbers in a triangular table (see Table 6.2). This is similar to Table 6.1. The difference is that we list all mesh points and the corresponding function values twice, and in the third column the first divided differences corresponding to equal mesh points are the given derivative values. The rest of the numbers in the table are computed in a similar way as in Table 6.1. The framed numbers are used in formula (6.8) as the coefficients.

Example 6.21. Consider the following data:

x_i	-1	1	2
y_i	2	4	11
y_i'	3	-5	30

Find the corresponding Hermite interpolating polynomial. We fill out the following table of divided differences:

$$\begin{array}{cccccccc}
-1 & 2 & & & & & & \\
-1 & 2 & \boxed{3} & & & & & \\
1 & 4 & 1 & -1 & & & & \\
1 & 4 & \boxed{-5} & -3 & -1 & & & \\
2 & 11 & 7 & 12 & 5 & 2 & & \\
2 & 11 & \boxed{30} & 23 & 11 & 2 & 0 &
\end{array}$$

In the third column the framed numbers are the input derivative values. Therefore, the Hermite polynomial is

$$H_5(x) = 2 + 3(x+1) - (x+1)^2 - (x+1)^2(x-1) + 2(x+1)^2(x-1)^2 = 2x^4 - x^3 - 6x^2 + 2x + 7,$$

so H_5 is a polynomial of degree 4. □

Exercises

1. Compute the Hermite interpolating polynomials corresponding to the following data:

(a)

x_i	-2	-1	0	1
y_i	4	1	14	-35
y_i'	-1	-2	43	-394

(b)

x_i	-1	0	2	3
y_i	1	2	64	-19
y_i'	3	-1	111	-301

2. Prove that if P is a polynomial of degree at most $(2n+2)$, x_i $(i = 0, 1, \ldots, n)$ are pairwise different mesh points, and H_{2n+1} is the Hermite polynomial corresponding to P and the mesh points, then $P(x) = H_{2n+1}(x)$ for all x.

3. Let $f \in C^1$. Prove that

$$\lim_{(x_0', x_1', \ldots, x_n') \to (x_0, x_1, \ldots, x_n)} f[x_0, x_0', x_1, x_1', \ldots, x_n, x_n'] = f[x_0, x_0, x_1, x_1, \ldots, x_n, x_n]$$

and

$$\lim_{(x_0', \ldots, x_{n-1}') \to (x_0, \ldots, x_{n-1})} f[x_0, x_0', x_1, x_1', \ldots, x_{n-1}, x_{n-1}', x_n] = f[x_0, x_0, x_1, x_1, \ldots, x_{n-1}, x_{n-1}, x_n].$$

4. Let $i_0, i_1, \ldots, i_n$ be a rearrangement of the finite sequence $0, 1, \ldots, n$. Show that

$$f[x_0, x_0, x_1, x_1, \ldots, x_n, x_n] = f[x_{i_0}, x_{i_0}, x_{i_1}, x_{i_1}, \ldots, x_{i_n}, x_{i_n}].$$

5. The Hermite interpolation problem can be formulated in a general form: at the ith mesh point the first k_i derivatives of a function is given, which we are to interpolate. We can generalize the method of this section. As an illustration we consider the following problem: given two mesh points x_0 and x_1, and a function $f \in C^3$. We are looking for a polynomial of minimal degree for which

$$H(x_0) = f(x_0), \quad H'(x_0) = f'(x_0), \quad H''(x_0) = f''(x_0), \quad \text{and} \quad H(x_1) = f(x_1).$$

(Here $k_0 = 2$ and $k_1 = 0$.) Show that the solution of this problem is the polynomial of degree at most 3

$$H(x) := f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_0](x - x_0)^2 + f[x_0, x_0, x_0, x_1](x - x_0)^3.$$

6.5. Spline Interpolation

Let $a = x_0 < x_1 < \ldots < x_n = b$ be a division of the interval $[a, b]$. The continuous function $S\colon [a, b] \to \mathbb{R}$ is a *spline function of degree* k corresponding to the mesh $\{x_0, \ldots, x_n\}$ if $S \in C^{k-1}[a, b]$, and the restriction of S to each interval $[x_i, x_{i+1}]$ is a polynomial of degree at most k. The first, second and third order spline functions are called *linear, quadratic* and *cubic spline functions*, respectively.

The simplest method of the interpolation is when linear splines are used to interpolate the given data. Geometrically this means that we connect the given data points (x_i, y_i) by line segments. The error of the linear spline interpolation is discussed in Exercise 2.

The main disadvantage of the linear spline interpolation is that the interpolating function is not smooth, i.e., it is not differentiable. In case of cubic spline interpolation the interpolating function is twice continuously differentiable, which is smooth enough in practice. For the rest of this section we investigate cubic spline interpolation.

Suppose given pairwise different mesh points $a = x_0 < x_1 < \ldots < x_n = b$ and corresponding function values $y_0, y_1, \ldots, y_n$. We are looking for a cubic spline function S which interpolates the given data, i.e., it satisfies

$$S(x_i) = y_i, \qquad i = 0, 1, \ldots, n.$$

The restriction of S to the interval $[x_i, x_{i+1}]$ is denoted by S_i $(i = 0, 1, \ldots, n-1)$. Since S interpolates the points (x_i, y_i), and it is twice continuously differentiable, therefore, the functions S_i satisfy the following relations:

$$S_i(x_i) = y_i, \qquad i = 0, 1, \ldots, n-1, \tag{6.10}$$
$$S_i(x_{i+1}) = y_{i+1}, \qquad i = 0, 1, \ldots, n-1, \tag{6.11}$$
$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \qquad i = 0, 1, \ldots, n-2, \tag{6.12}$$
$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), \qquad i = 0, 1, \ldots, n-2. \tag{6.13}$$

Since the polynomials S_i are defined by 4 parameters, so S is determined by $4n$ parameters. The number of conditions in (6.10)–(6.13) is only $4n - 2$, therefore, this problem has no unique solution yet. Hence we expect that two additional conditions can be given, and then we hope to have a unique solution. Frequently used conditions are the following

$$S_0''(x_0) = 0 \qquad \text{and} \qquad S_{n-1}''(x_n) = 0. \tag{6.14}$$

A cubic spline function defined by conditions (6.10)–(6.14) is called *natural spline* function. Next we show that the above problem has a unique natural spline solution. Consider the functions S_i in the form:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,$$

where a_i, b_i, c_i and d_i $(i = 0, 1, \ldots, n-1)$ are parameters to be determined. Then

$$S_i'(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2,$$
$$S_i''(x) = 2c_i + 6d_i(x - x_i).$$

These equations imply

$$a_i = S_i(x_i) = y_i, \quad b_i = S_i'(x_i) \quad \text{and} \quad c_i = S_i''(x_i)/2, \quad i = 0, 1, \ldots, n-1. \tag{6.15}$$

With the help of relation (6.15) we define the constants a_n, b_n and c_n (which will be used later):

$$a_n := y_n, \qquad b_n := S'(x_n) \qquad \text{and} \qquad c_n := S''(x_n)/2. \tag{6.16}$$

(The derivatives in (6.16) denote left sided derivatives.) Substituting $x = x_{i+1}$ into the formula of S_i, and using equation (6.11) and relation $a_i = y_i$, we get

$$y_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = y_{i+1}.$$

Introduce the notations $\Delta x_i := x_{i+1} - x_i$ and $\Delta y_i := y_{i+1} - y_i$. Then

$$b_i \Delta x_i + c_i(\Delta x_i)^2 + d_i(\Delta x_i)^3 = \Delta y_i, \qquad i = 0, 1, \ldots, n-1. \tag{6.17}$$

Condition (6.12) and relation $b_{i+1} = S_{i+1}'(x_{i+1})$ yield

$$b_i + 2c_i \Delta x_i + 3d_i(\Delta x_i)^2 = b_{i+1} \tag{6.18}$$

for $i = 0, 1, \ldots, n-2$. Using the definition of b_n we get that (6.18) holds for $i = n-1$ too. Similarly, from equation (6.13) and the definition of c_n it follows

$$2c_i + 6d_i \Delta x_i = 2c_{i+1}, \qquad i = 0, 1, \ldots, n-1,$$

hence

$$d_i = \frac{c_{i+1} - c_i}{3\Delta x_i}, \qquad i = 0, 1, \ldots, n-1. \tag{6.19}$$

Substituting it back to equations (6.17) and (6.18) we get

$$b_i \Delta x_i + c_i(\Delta x_i)^2 + \frac{c_{i+1} - c_i}{3}(\Delta x_i)^2 = \Delta y_i, \qquad i = 0, 1, \ldots, n-1, \tag{6.20}$$

$$b_i + 2c_i \Delta x_i + (c_{i+1} - c_i)\Delta x_i = b_{i+1}, \qquad i = 0, 1, \ldots, n-1. \tag{6.21}$$

From the first equation we express

$$b_i = \frac{\Delta y_i}{\Delta x_i} - \frac{2c_i + c_{i+1}}{3}\Delta x_i,$$

and substituting it into the second equation for $i = 0, 1, \ldots, n-2$ we get

$$c_i \Delta x_i + 2c_{i+1}(\Delta x_i + \Delta x_{i+1}) + c_{i+2}\Delta x_{i+1} = 3\frac{\Delta y_{i+1}}{\Delta x_{i+1}} - 3\frac{\Delta y_i}{\Delta x_i}, \quad i = 0, 1, \ldots, n-2. \tag{6.22}$$

Note that in the derivation of equation (6.22) we have not used condition (6.14), so it holds for any cubic spline interpolation.

Equation (6.22) determines a system of $n-1$ linear equations for c_i. We add equations $c_0 = 0$ and $c_n = 0$ following from condition (6.14) into it, so we get a $n+1$-dimensional linear system of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x} = (c_0, c_1, \ldots, c_n)^T$,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \Delta x_0 & 2(\Delta x_0 + \Delta x_1) & \Delta x_1 & 0 & 0 & \cdots & 0 \\ 0 & \Delta x_1 & 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & \cdots & & & \Delta x_{n-2} & 2(\Delta x_{n-2} + \Delta x_{n-1}) & \Delta x_{n-1} \\ 0 & \cdots & & & 0 & 0 & 1 \end{pmatrix}$$

is a tridiagonal matrix and

$$\mathbf{b} = \begin{pmatrix} 0 \\ 3\frac{\Delta y_1}{\Delta x_1} - 3\frac{\Delta y_0}{\Delta x_0} \\ \vdots \\ 3\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - 3\frac{\Delta y_{n-2}}{\Delta x_{n-2}} \\ 0 \end{pmatrix}.$$

Since $\mathbf{A}$ is diagonally dominant, the system $\mathbf{Ax} = \mathbf{b}$ has a unique solution. Then with the help of c_i, we can compute the coefficients d_i and b_i. Therefore, the problem has a unique solution. We note that, in practice, the tridiagonal system $\mathbf{Ax} = \mathbf{b}$ can be solved efficiently by the special Gaussian elimination defined in Algorithm 3.37. We have proved the following result.

Theorem 6.22. *The problem of natural cubic spline interpolation has a unique solution.*

Example 6.23. Find the natural cubic spline interpolation of the following given data:

x_i	0.0	1.0	1.5	2.0	3.0	4.0
y_i	0.5	0.1	2.5	-1.0	-0.5	0.0

Using the notations introduced before the linear system of the coefficients c_i is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 2 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 3 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 15.6 \\ -35.4 \\ 22.5 \\ 0 \\ 0 \end{pmatrix}.$$

Solving it and substituting back c_i into (6.19) and (6.20) we get the coefficients d_i and b_i. The resulting natural spline function is:

$$\begin{aligned} S_0(x) &= 0.5 - 3.4141079x + 3.0141079x^3, \\ S_1(x) &= 0.1 + 5.6282158(x-1) + 9.04232365(x-1)^2 - 21.3975104(x-1)^3, \\ S_2(x) &= 2.5 - 1.3775934(x-1.5) - 23.0539419(x-1.5)^2 + 23.6182573(x-1.5)^3, \\ S_3(x) &= -1.0 - 6.7178423(x-2) + 12.3734440(x-2)^2 - 5.1556017(x-2)^3, \\ S_4(x) &= -0.5 + 2.5622407(x-3) - 3.0933610(x-3)^2 + 1.0311203(x-3)^3. \end{aligned}$$

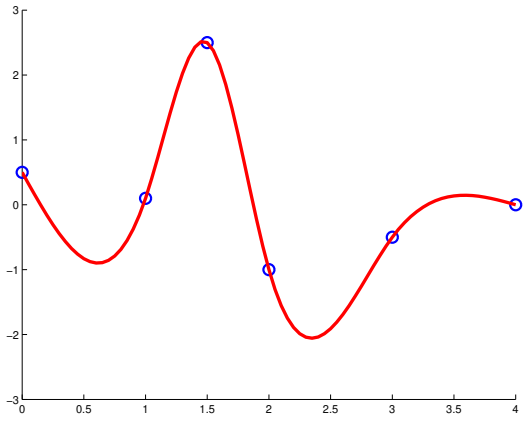

Figure 6.3: Natural spline interpolation

The graph of this function can be seen in Figure 6.3. □

Instead of condition (6.14) we can specify other boundary conditions for S. Now we investigate condition

$$S'(x_0) = y'_0 \qquad \text{and} \qquad S'(x_n) = y'_n, \tag{6.23}$$

where y'_0 and y'_n are given numbers. This means that we know (specify) the slope of the tangent line of S at the end points of the interval. A cubic spline which satisfy conditions (6.23) is called *clamped spline* function. In this case equations (6.22) hold. We need to add two equations in order to get a well-posed linear system. Using relations $b_0 = S'(x_0) = y'_0$, equation (6.20) implies

$$y'_0 \Delta x_0 + c_0(\Delta x_0)^2 + \frac{c_1 - c_0}{3}(\Delta x_0)^2 = \Delta y_0,$$

hence

$$2c_0 \Delta x_0 + c_1 \Delta x_0 = 3\frac{\Delta y_0}{\Delta x_0} - 3y'_0. \tag{6.24}$$

Expressing b_{n-1} from equation (6.20) and substituting it into (6.21), and using relation $b_n = y'_n$ we get

$$\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{2c_{n-1} + c_n}{3}\Delta x_{n-1} + \Delta x_{n-1}(c_{n-1} + c_n) = y'_n,$$

hence

$$c_{n-1}\Delta x_{n-1} + 2c_n \Delta x_{n-1} = 3y'_n - 3\frac{\Delta y_{n-1}}{\Delta x_{n-1}}. \tag{6.25}$$

If in the system $\mathbf{Ax} = \mathbf{b}$ of the natural spline interpolation we replace the first equation with equation (6.24) and the last equation with (6.25), then it is easy to see that the coefficient matrix remains to be diagonally dominant, therefore, the modified system has a unique solution. So the cubic spline interpolation problem together with conditions (6.23) has a unique clamped spline function solution.

The natural cubic spline interpolating functions have the following minimal property, which means that the spline interpolating functions are the smoothest among all possible interpolating functions.

Theorem 6.24. *Let $a = x_0 < x_1 < \ldots < x_n = b$ be mesh points and $y_0, y_1, \ldots, y_n$ be function values, and let S be the natural cubic spline interpolating function associated to the given data. Then*

$$\int_a^b (S''(x))^2\,dx \le \int_a^b (f''(x))^2\,dx \tag{6.26}$$

for every $f \in C^2[a,b]$, which also interpolates the given data, i.e., $f(x_i) = y_i$ for $i = 0, 1, \ldots, n$.

Proof. Introduce the function $g(x) := f(x) - S(x)$. Then $f''(x) = S''(x) + g''(x)$, and so

$$\int_a^b (f''(x))^2\,dx = \int_a^b (S''(x))^2\,dx + 2\int_a^b S''(x)g''(x)\,dx + \int_a^b (g''(x))^2\,dx.$$

Since $\int_a^b (g''(x))^2\,dx \ge 0$, the statement of the theorem follows if we show

$$\int_a^b S''(x)g''(x)\,dx = 0.$$

Dividing the integral into the sum of integral over the intervals of consecutive mesh points, and using integration by parts we get

$$\begin{aligned}\int_a^b S''(x)g''(x)\,dx &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} S''(x)g''(x)\,dx \\ &= \sum_{i=1}^n [S''(x)g'(x)]_{x_{i-1}}^{x_i} - \sum_{i=1}^n \int_{x_{i-1}}^{x_i} S'''(x)g'(x)\,dx \\ &= S''(b)g'(b) - S''(a)g'(a) - \sum_{i=1}^n \int_{x_{i-1}}^{x_i} S'''(x)g'(x)\,dx.\end{aligned}$$

Since S is a natural spline function, we have $S''(a) = S''(b) = 0$. Since S is a third order polynomial over the intervals $[x_{i-1}, x_i]$, its second derivative is constant, which can be factored out in front of the integral. But $\int_{x_{i-1}}^{x_i} g'(x)\,dx = g(x_i) - g(x_{i-1}) = 0$, since $g(x_i) = 0$ for $i = 0, 1, \ldots, n$. This completes the proof. □

The next theorem investigates the error of the clamped cubic spline interpolation. We present the result without proof.

Theorem 6.25. *Let $f \in C^4[a,b]$, $a = x_0 < x_1 < \ldots < x_n = b$ mesh points, $y_i = f(x_i)$, $i = 0, 1, \ldots, n$ function values, and $y_0' = f'(a)$ and $y_n' = f'(b)$ derivative values, and let S*

be the corresponding clamped cubic spline function. Then for $x \in [a,b]$ it follows

$$|f(x) - S(x)| \le \frac{5}{384} M_4 h^4,$$
$$|f'(x) - S'(x)| \le \left(\frac{\sqrt{3}}{216} + \frac{1}{24} \right) M_4 h^3,$$
$$|f''(x) - S''(x)| \le \left(\frac{1}{12} + \frac{h}{3k} \right) M_4 h^2,$$

where $M_4 := \max\{|f^{(4)}(x)| : x \in [a,b]\}$, $h := \max\{x_{i+1} - x_i : i = 0, 1, \ldots, n-1\}$, $k := \min\{x_{i+1} - x_i : i = 0, 1, \ldots, n-1\}$.

We note that the error of the natural cubic spline interpolating function can be given similarly.

Exercises

1. Find the formula of the linear spline function interpolating the data (x_i, y_i), $i = 0, 1, \ldots, n$ on the interval $[x_i, x_{i+1}]$.
2. Given a continuous function $f : [a,b] \to \mathbb{R}$, and let S_h be a linear spline interpolating function of the function f corresponding to equidistant mesh of the interval $[a,b]$ with step size h.
   (a) Show that $\max\{|f(x) - S_h(x)| : x \in [a,b]\} \to 0$, as $h \to 0$.
   (b) Let $f \in C^1[a,b]$. Show that
   $$|f(x) - S_h(x)| \le M_1 h, \qquad x \in [a,b],$$
   where $M_1 := \max\{|f'(x)| : x \in [a,b]\}$.
3. Compute and draw the graph of the natural cubic spline function corresponding to the data given in Exercise 1 of Section 6.1.
4. Show that for a cubic spline interpolation any of the conditions
   $$S'(x_0) = f'(x_0) \qquad \text{or} \qquad S'(x_n) = f'(x_n)$$
   determines the cubic spline interpolation function uniquely.
5. Show that if S is the clamped cubic spline corresponding to given mesh points $a = x_0 < x_1 < \ldots < x_n = b$, function values $y_0, y_1, \ldots, y_n$, and derivative values y'_0 and y'_n, then S satisfies inequality (6.26) for all functions $f \in C^2[a,b]$ which satisfy $f(x_i) = y_i$ for all i, $f'(a) = y'_0$ and $f'(b) = y'_n$.

Chapter 7
Numerical Differentiation and Integration

In this chapter first we study several methods for numerical differentiation, and consider the Richardson's extrapolation method to obtain higher order methods. Next we define Newton-Cotes formulas and the Gaussian quadrature to approximate definite integrals.

7.1. Numerical differentiation

In this section we present two methods to derive numerical approximation formulas for the derivative, and we derive some basic approximation formulas.

The derivative of a function is defined by the limit

$$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h}.$$

Therefore, if $|h|$ is small, then the difference quotient $\frac{f(x_0+h)-f(x_0)}{h}$ is close to the value of the derivative. But we need more: we need to know the truncation error of the approximation. Next we derive this formula in two different ways, and we will derive the formula of the truncation error too.

Suppose $f \in C^3[a,b]$ and $x_0 \in (a,b)$. The idea of the first method is the following: We approximate the function f in a neighbourhood of x_0 by a Lagrange polynomial $L_n(x)$. We use $L_n'(x_0)$ as an approximation of $f'(x_0)$. We will call this method as *Lagrange's method*. Consider a simple case: let $n = 1$, $x_1 = x_0 + h \in (a,b)$ (and $x_0 \neq x_1$), consider the first-order Lagrange polynomial interpolation of f corresponding to the mesh points x_0 and x_1:

$$\begin{aligned} f(x) &= L_1(x) + E_1(x) \\ &= \frac{f(x_0)(x-x_0-h)}{-h} + \frac{f(x_0+h)(x-x_0)}{h} + \frac{f''(\xi(x))}{2}(x-x_0)(x-x_0-h). \end{aligned}$$

Taking the derivative of both sides we get

$$\begin{aligned} f'(x) = \frac{f(x_0+h)-f(x_0)}{h} &+ \frac{f''(\xi(x))}{2}(2(x-x_0)+h) \\ &+ \frac{d}{dx}\Big(f''(\xi(x))\Big)\frac{(x-x_0)(x-x_0-h)}{2}. \end{aligned} \tag{7.1}$$

Theorem 6.8 yields that the function $f''(\xi(x))$ is differentiable for $x \neq x_0, x_0 + h$, but the derivative cannot be computed explicitly. On the other hand, taking the limit $x \to x_0$ in (7.1) we get

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi), \tag{7.2}$$

where $\xi \in \langle x_0, x_0 + h \rangle$. Therefore, if we use the approximation formula

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}, \tag{7.3}$$

the truncation error of the approximation has the form $-\frac{h}{2} f''(\xi)$. Formula (7.3) is called *first-order forward difference* formula if $h > 0$, and *first-order backward difference* formula if $h < 0$. In these formulas the mesh point $x_0 + h$ is located right and left to x_0, in the respective cases. Formula (7.2) shows that approximation (7.3) is first-order in h. Formula (7.3) is also called *two-point difference formula*, since it uses two mesh points.

The same formula can be derived (under weaker conditions) in the following way: Let $f \in C^2[a, b]$, and consider the first-order Taylor expansion of f around x_0:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi(x))}{2}(x - x_0)^2.$$

Substitution $x = x_0 + h$ gives

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(\xi)}{2} h^2,$$

hence

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi),$$

where $\xi = \xi(x_0 + h)$.

Example 7.1. Consider the function $f(x) = e^{x^2+x}$. We have $f'(x) = e^{x^2+x}(2x + 1)$, so $f'(0) = 1$. We compute an approximate value of $f'(0)$ using the first-order forward ($h > 0$) and backward ($h < 0$) difference formula, i.e., formula (7.3). In Table 7.1 we printed the approximate values and their errors for different values of h. The numerical results show that if the step size h decreases by one order of magnitude, then the corresponding error also decreases by one order of magnitude. □

Table 7.1: First-order difference formula, $f(x) = e^{x^2+x}$, $x_0 = 0$

$\lvert h \rvert$	forward difference	error	backward difference	error
0.100	1.1627807	1.6278e-01	0.8606881	1.3931e-01
0.010	1.0151177	1.5118e-02	0.9851156	1.4884e-02
0.001	1.0015012	1.5012e-03	0.9985012	1.4988e-03

The previous two methods are appropriate to derive higher order, so more precise formulas. Suppose $f \in C^{n+1}$, and consider an approximation of f by a Lagrange polynomial of degree n:

$$f(x) = \sum_{k=0}^{n} f(x_k) l_k(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1) \cdots (x - x_n), \tag{7.4}$$

where $l_k(x)$ are the Lagrange basis polynomials of degree n defined by (6.2). Differentiating (7.4) and using substitution $x = x_i$ we get

$$f'(x_i) = \sum_{j=0}^{n} f(x_j) l_j'(x_i) + \frac{f^{(n+1)}(\xi(x_i))}{(n+1)!} \prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j), \tag{7.5}$$

which is called $n+1$-*point difference formula* to approximate $f'(x_i)$. We apply relation (7.5) for equidistant mesh points, so we assume $x_j = x_0 + jh$, where $h > 0$. It can be shown that the error term in (7.5) is of nth-order in h, and then the resulting formula will also be called *difference formula of order* n.

Consider the case when $n = 2$, i.e., we study three-point formulas. Consider the mesh points $x_0, x_0 + h, x_0 + 2h$. Then

$$\begin{aligned} l_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-x_1)(x-x_2)}{2h^2}, \\ l_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-x_0)(x-x_2)}{-h^2}, \\ l_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-x_0)(x-x_1)}{2h^2}, \end{aligned}$$

therefore,

$$\begin{aligned} l_0'(x) &= \frac{2x-x_1-x_2}{2h^2}, \\ l_1'(x) &= \frac{2x-x_0-x_2}{-h^2}, \\ l_2'(x) &= \frac{2x-x_0-x_1}{2h^2}. \end{aligned}$$

We apply them with $x = x_0$, $x = x_0 + h$ and $x = x_0 + 2h$, so relation (7.5) yields

$$f'(x_0) = \frac{1}{h}\left(-\frac{3}{2}f(x_0) + 2f(x_0+h) - \frac{1}{2}f(x_0+2h)\right) + \frac{h^2}{3}f'''(\xi_0), \tag{7.6}$$

$$f'(x_0+h) = \frac{1}{h}\left(-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0+2h)\right) - \frac{h^2}{6}f'''(\xi_1), \tag{7.7}$$

$$f'(x_0+2h) = \frac{1}{h}\left(\frac{1}{2}f(x_0) - 2f(x_0+h) + \frac{3}{2}f(x_0+2h)\right) + \frac{h^2}{3}f'''(\xi_2). \tag{7.8}$$

The substitutions $x_0 \leftarrow x_0 - 2h$ and $h \leftarrow -h$ give that (7.8) can be written in the form (7.6), and (7.7) has the form

$$f'(x_0) = \frac{1}{h}\left(-\frac{1}{2}f(x_0-h) + \frac{1}{2}f(x_0+h)\right) - \frac{h^2}{6}f'''(\xi_1). \tag{7.9}$$

Relation (7.9) is called *three-point midpoint formula* or *second-order central difference formula*. (It is also called *centered difference*.) Formula (7.6) is called *three-point endpoint*

formula. It is also called *second-order forward difference* formula if $h > 0$, and *second-order backward difference* formula if $h < 0$.

Example 7.2. We approximate the derivative of the function $f(x) = e^{x^2+x}$ at $x = 0$ with second-order difference formulas (formulas (7.6) and (7.9)). The results can be seen in Table 7.2 for different values of h. The numerical results demonstrate that the truncation error of the formulas is second-order in h. □

Table 7.2: Second-order difference formulas, $f(x) = e^{x^2+x}$, $x_0 = 0$

$\|h\|$	forward	error	backward	error	central	error
0.100	0.9693157	3.0684e-02	0.9820952	1.7905e-02	1.0117344	1.1734e-02
0.010	0.9997603	2.3968e-04	0.9997728	2.2718e-04	1.0001167	1.1667e-04
0.001	0.9999977	2.3396e-06	0.9999977	2.3271e-06	1.0000012	1.1667e-06

Without proofs we present 5-point central and one-sided formulas, i.e., *fourth-order difference formulas*:

$$f'(x_0) = \frac{1}{12h}\Big(-25f(x_0) + 48f(x_0+h) - 36f(x_0+2h) + 16f(x_0+3h) - 3f(x_0+4h)\Big) + \frac{h^4}{5}f^{(5)}(\xi_0), \tag{7.10}$$

$$f'(x_0) = \frac{1}{12h}\Big(f(x_0-2h) - 8f(x_0-h) + 8f(x_0+h) - f(x_0+2h)\Big) + \frac{h^4}{30}f^{(5)}(\xi_1). \tag{7.11}$$

Formula (7.10) is one-sided, and (7.11) is central difference.

Example 7.3. We apply formulas (7.10) and (7.11) to approximate the first derivative of $f(x) = e^{x^2+x}$ at $x = 0$. Table 7.3 shows the numerical results. □

Table 7.3: Fourth-order difference formulas, $f(x) = e^{x^2+x}$, $x_0 = 0$

$\|h\|$	forward	error	backward	error	central	error
0.100	0.9967110	3.2890e-03	0.9991793	8.2070e-04	0.9997248	2.7523e-04
0.010	0.9999998	1.7345e-07	0.9999998	1.5136e-07	1.0000000	2.7005e-08
0.001	1.0000000	1.6311e-11	1.0000000	1.6090e-11	1.0000000	2.7000e-12

Next we use the *Taylor's method* to derive approximation formulas for higher order derivatives. Let $f \in C^4$, and consider the third-order Taylor polynomial expansion of f at x_0 with the fourth-order error term:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 + \frac{f'''(x_0)}{6}(x-x_0)^3 + \frac{f^{(4)}(\xi)}{24}(x-x_0)^4.$$

If we substitute $x = x_0 - h$ and $x = x_0 + h$ into this relation, we get

$$f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{f''(x_0)}{2}h^2 - \frac{f'''(x_0)}{6}h^3 + \frac{f^{(4)}(\xi_1)}{24}h^4$$

and

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2}h^2 + \frac{f'''(x_0)}{6}h^3 + \frac{f^{(4)}(\xi_2)}{24}h^4.$$

Adding the two equations we get

$$f(x_0 - h) + f(x_0 + h) = 2f(x_0) + f''(x_0)h^2 + \frac{f^{(4)}(\xi_1) + f^{(4)}(\xi_2)}{24}h^4,$$

which yields

$$f''(x_0) = \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2} + \frac{f^{(4)}(\xi_1) + f^{(4)}(\xi_2)}{24}h^4.$$

Therefore, the approximation formula

$$f''(x_0) \approx \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2} \tag{7.12}$$

has an error of order h^2. We can rewrite the error term $\frac{f^{(4)}(\xi_1)+f^{(4)}(\xi_2)}{24}h^4$ in a simpler form. We have by our assumptions that $f^{(4)}$ is continuous, therefore, Theorem 2.2 yields that there exists a point ξ in between ξ_1 and ξ_2 such that

$$f^{(4)}(\xi) = \frac{f^{(4)}(\xi_1) + f^{(4)}(\xi_2)}{2}.$$

Hence

$$f''(x_0) = \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2} + \frac{f^{(4)}(\xi)}{12}h^2. \tag{7.13}$$

Example 7.4. We computed the approximation of the second-order derivative of $f(x) = e^{x^2+x}$ at $x = 0$ using formula (7.12) and different step sizes. The numerical results can be seen in Table 7.4. Note that the exact derivative value is $f''(0) = 3$. □

Table 7.4: Approximation of the second-order derivative, $f(x) = e^{x^2+x}$, $x_0 = 0$

h	approximation	error
0.100	3.0209256	2.0926e-02
0.010	3.0002083	2.0834e-04
0.001	3.0000021	2.0833e-06

The numerical differentiation is an unstable problem. To illustrate it we consider a function $f(x)$ and its perturbation of the form

$$g(x) = f(x) + \frac{1}{n}\sin(n^2 x).$$

If we compute an approximation of g' instead of f' using any difference formula obtained above, then there is a small change in the function values used in the difference formula if n is large. But the difference between the exact value of the derivatives is large, since $g'(x) = f'(x) + n\cos(n^2x)$.

Next we investigate the effect of the rounding in numerical differentiation. Consider the simplest difference formula, the first-order difference (7.2). Suppose that here, instead of the exact function values $f(x_0)$ and $f(x_0+h)$, we use their approximate values f_0 and f_1, where

$$f(x_0) = f_0 + e_0 \qquad \text{and} \qquad f(x_0+h) = f_1 + e_1.$$

Then

$$f'(x_0) \approx \frac{f_1 - f_0}{h},$$

and the resulting error is

$$\begin{aligned} f'(x_0) - \frac{f_1 - f_0}{h} &= f'(x_0) - \frac{f(x_0+h) - f(x_0)}{h} + \frac{f(x_0+h) - f(x_0)}{h} - \frac{f_1 - f_0}{h} \\ &= -\frac{h}{2}f''(\xi) + \frac{e_1 - e_0}{h}. \end{aligned} \tag{7.14}$$

Relation (7.14) shows that the error consists of two parts: the truncation error and the rounding error. If the step size h is small, then the rounding error will be small, but the rounding error can go to ∞ as $h \to 0$.

Example 7.5. Consider the function $f(x) = e^x$. We compute the approximation of $f'(1) = e$ using first-order forward difference formula. In order to enlarge the effect of the rounding, we used 6- and 4-digit arithmetic in the computation. We can see in Table 7.5 that in case of the 4-digit arithmetic, when we decreased the step size to 0.001 from 0.01, the error of the approximation increased. The reason is, clearly, the increase of the rounding error, since here we subtracted two numbers which are close to each other, and also divided by a small number. □

Table 7.5: Effect of rounding in first-order forward difference, $f(x) = e^x$, $x_0 = 1$

	6-digit arithmetic		4-digit arithmetic	
h	approximation	error	approximation	error
0.100	2.8589000	1.4062e-01	2.8600000	1.4172e-01
0.010	2.7320000	1.3718e-02	2.8000000	8.1718e-02
0.001	2.7200000	1.7182e-03	3.0000000	2.8172e-01

The formulas derived in this section can be applied to approximate partial derivatives. We list some formulas next.

$$\frac{\partial f(x_0,y_0)}{\partial x} \approx \frac{f(x_0+h,y_0)-f(x_0,y_0)}{h}, \tag{7.15}$$

$$\frac{\partial f(x_0,y_0)}{\partial y} \approx \frac{f(x_0,y_0+h)-f(x_0,y_0)}{h}, \tag{7.16}$$

$$\frac{\partial^2 f(x_0,y_0)}{\partial x^2} \approx \frac{f(x_0+h,y_0)-2f(x_0,y_0)+f(x_0-h,y_0)}{h^2} \tag{7.17}$$

$$\frac{\partial^2 f(x_0,y_0)}{\partial y^2} \approx \frac{f(x_0,y_0+h)-2f(x_0,y_0)+f(x_0,y_0-h)}{h^2} \tag{7.18}$$

$$\frac{\partial^2 f(x_0,y_0)}{\partial x\,\partial y} \approx \frac{f(x_0+h,y_0+h)-f(x_0+h,y_0)-f(x_0,y_0+h)+f(x_0,y_0)}{h^2} \tag{7.19}$$

$$\frac{\partial^2 f(x_0,y_0)}{\partial x^2} \approx \frac{f(x_0+2h,y_0)-2f(x_0+h,y_0)+f(x_0,y_0)}{h^2} \tag{7.20}$$

Exercises

1. Compute an approximation of $f'(x_0)$ using first-order forward and backward difference formulas with $h = 0.1$ and 0.01 if

   (a) $f(x) = x^4 - 6x^2 + 3x, \quad x_0 = 1,$ (b) $f(x) = e^x \sin x, \quad x_0 = 0,$

   (c) $f(x) = \cos x^2, \quad x_0 = 1,$ (d) $f(x) = x \ln x, \quad x_0 = 1.$

2. Apply second-order difference formulas in the previous exercise.
3. Approximate $f''(x_0)$ for the functions given in Exercise 1.
4. Derive formulas (7.6) and (7.9) using Taylor's method.
5. Prove relations (7.10) and (7.11).
6. Derive the following approximation formulas:

$$f'''(x_0) \approx \frac{1}{2h^3}\Big(f(x_0+2h)-2f(x_0+h)+2f(x_0-h)-f(x_0-2h)\Big),$$

$$f^{(4)}(x_0) \approx \frac{1}{h^4}\Big(f(x_0+2h)-4f(x_0+h)+6f(x_0)-4f(x_0-h)+f(x_0+2h)\Big)$$

7. Derive formulas (7.15)–(7.20) using

   (a) approximation formulas formulated for single variable functions,

   (b) two-variable Lagrange's method,

   (c) two-variable Taylor's method.

   Compute the truncation errors.

7.2. Richardson's extrapolation

Suppose given a value M, and let $K(h)$ be its approximation, where h denotes the discretization parameter of the approximation method. We also suppose that the truncation error of the approximation is known, and it has a special form, the error can be given by an even-order Taylor polynomial (or possibly Taylor series) approximation of the form

$$M = K(h) + a_2h^2 + a_4h^4 + a_6h^6 + \cdots + a_{2m}h^{2m} + b(h), \tag{7.21}$$

where $|b(h)| \le Bh^{2m+2}$ with some constant $B > 0$. The error here is second-order in h. Now we present a general method to generate higher order approximation formulas using $K(h)$. Consider relation (7.21) corresponding to parameter $h/2$:

$$M = K(h/2) + a_2\frac{h^2}{4} + a_4\frac{h^4}{16} + a_6\frac{h^6}{64} + \cdots + a_{2m}\frac{h^{2m}}{2^{2m}} + b(h/2). \tag{7.22}$$

Multiplying both sides of (7.22) by 4, and subtracting equation (7.21) from it, the second-order term in h cancels out, and solving it for M we get

$$\begin{aligned} M = {} & \frac{4K(h/2) - K(h)}{3} - \frac{1}{4}a_4h^4 - \frac{5}{16}a_6h^6 \\ & - \cdots - \frac{2^{2m-2}-1}{2^{2m-2}\cdot 3}a_{2m}h^{2m} + \frac{4b(h/2) - b(h)}{3}. \end{aligned} \tag{7.23}$$

This relation can be written in the form

$$M = K^{(1)}(h) + a_4^{(1)}h^4 + a_6^{(1)}h^6 + \cdots + a_{2m}^{(1)}h^{2m} + b^{(1)}(h), \tag{7.24}$$

where

$$K^{(1)}(h) := \frac{4K(h/2) - K(h)}{3}, \quad b^{(1)}(h) := \frac{4b(h/2) - b(h)}{3}, \quad a_{2i}^{(1)} := \frac{1 - 4^{i-1}}{4^{i-1}\cdot 3}a_{2i},$$

$i = 2, \ldots, m$. Relation (7.24) yields that formula $K^{(1)}(h)$ approximates M with a fourth-order error in h. The previous method can be repeated: we use (7.24) with $h/2$, multiply it by 16, subtract from it equation (7.24), and then solve it for M. Then the fourth-order error term cancels out, and we get relation

$$M = K^{(2)}(h) + a_6^{(2)}h^6 + \cdots + a_{2m}^{(2)}h^{2m} + b^{(2)}(h), \tag{7.25}$$

where

$$\begin{aligned} K^{(2)}(h) &:= \frac{16K^{(1)}(h/2) - K^{(1)}(h)}{15}, \quad b^{(2)}(h) := \frac{16b^{(1)}(h/2) - b^{(1)}(h)}{15}, \\ a_{2i}^{(2)} &:= \frac{1 - 4^{i-2}}{4^{i-2}\cdot 15}a_{2i}^{(1)}, \quad i = 3, \ldots, m. \end{aligned}$$

Relation (7.25) means that $K^{(2)}(h)$ approximates M with a sixth-order error in h. The generation of new approximation formulas can be continued as

$$K^{(i+1)}(h) := K^{(i)}(h/2) + \frac{K^{(i)}(h/2) - K^{(i)}(h)}{4^{i+1} - 1}, \qquad i = 0, 1, \ldots, m-1, \tag{7.26}$$

where $K^{(0)}(h) := K(h)$. This procedure to generate higher order approximation formulas is called *Richardson's extrapolation*. A similar procedure can be applied also in the case when the Taylor expansion of the truncation error contains all powers of h (see Exercises 2 and 3), but later we will use the case presented in this section.

Example 7.6. In the previous section we saw that the central difference formula (7.9) is second-order in h. Using Taylor's method we get a more precise form of the truncation error. Suppose that $f \in C^{2m+3}$, and consider the following Taylor's expansion:

$$f(x_0+h) = f(x_0) + f'(x_0)h + \cdots + \frac{f^{(2m+2)}(x_0)}{(2m+2)!}h^{2m+2} + \frac{f^{(2m+3)}(\xi_1)}{(2m+3)!}h^{2m+3}.$$

We apply the previous relation with $-h$ instead of h, subtracting the two equations, and solving it for $f'(x_0)$ we get:

$$\begin{aligned} f'(x_0) = {} & \frac{f(x_0+h)-f(x_0-h)}{2h} - \frac{f'''(x_0)}{3!}h^2 - \frac{f^{(5)}(x_0)}{5!}h^4 \\ & - \cdots - \frac{f^{(2m+1)}(x_0)}{(2m+1)!}h^{2m} - \frac{f^{(2m+3)}(\xi_1)+f^{(2m+3)}(\xi_2)}{(2m+3)!}h^{2m+2}. \end{aligned}$$

Hence we have that the central difference satisfies relation (7.21). Therefore, we get a higher order formula using Richardson's extrapolation. We have that formula

$$\begin{aligned} K^{(1)}(h) &= \frac{4\dfrac{f(x_0+h/2)-f(x_0+h/2)}{h} - \dfrac{f(x_0+h)-f(x_0-h)}{2h}}{3} \\ &= \frac{f(x_0-h) - 8f(x_0-h/2) + 8f(x_0+h/2) - f(x_0+h)}{6h} \end{aligned}$$

has fourth-order error in h. We note that this formula is equivalent to (7.11). □

Exercises

1. Derive a sixth-order approximation formula for the first derivative of a function starting from the central difference formula (7.9) using the Richardson's extrapolation. Apply the formula for approximating the first derivative of $f(x) = e^x \sin x$ at $x = 0$ using step size $h = 0.25$.
2. Reformulate the Richardson's extrapolation for the case when the Taylor expansion of the truncation error contains all powers of h, i.e.,
$$M = K(h) + a_1 h + a_2 h^2 + \cdots + a_m h^m + b(x),$$
where $|b(h)| \le Bh^{m+1}$ with some $B > 0$.
3. Reformulate the Richardson's extrapolation for the general case when
$$M = K(h) + a_1 h^{\alpha_1} + a_2 h^{\alpha_2} + \cdots + a_m h^{\alpha_m} + b(x),$$
where $1 \le \alpha_1 < \alpha_2 < \cdots < \alpha_m$ are integers, and $|b(h)| \le Bh^{\alpha_m+1}$ with some $B > 0$.
4. Derive a third-order approximation of the first derivative using Richardson's extrapolation starting from the first-order difference formula.

7.3. Newton–Cotes Formulas

Let $f \in C[a,b]$. The definite integral, similarly to the derivative, is defined by a limit. The definition using Riemann's sum is the following: consider a finite partition of the interval $[a,b]$ using the mesh points $a = x_0 < x_1 < \cdots < x_n = b$, and in each subinterval $[x_{i-1}, x_i]$ select a point ξ_i. Then the integral $\int_a^b f(x)\,dx$ is a limit of the Riemann's sum $\sum_{i=1}^n f(\xi_i)(x_i - x_{i-1})$ as the norm of the partition, $\max\{x_i - x_{i-1}\colon i = 1,\dots,n\}$ goes to zero. Such a Riemann's sum is for example

$$\int_a^b f(x)\,dx \approx \frac{b-a}{n}\left(f\left(\frac{x_0+x_1}{2}\right) + f\left(\frac{x_1+x_2}{2}\right) + \cdots + f\left(\frac{x_{n-1}+x_n}{2}\right)\right), \quad (7.27)$$

where $x_i = a + i(b-a)/n$, $i = 0, 1, \dots, n$. This formula is called *midpoint rule* or *rectangle rule*. (See Exercises 5 and 6.)

Similarly to the numerical differentiation, we can use the Lagrange's method to derive approximation formulas for definite integrals. Consider a partition of the interval $[a,b]$ (typically with equidistant mesh points), and let L_n be the Lagrange interpolating polynomial of the function f corresponding to the given mesh. Consider $\int_a^b L_n(x)\,dx$ as an approximation of $\int_a^b f(x)\,dx$. We suppose that $f \in C^{n+1}[a,b]$. Then Theorem 6.5 yields the error of the approximation:

$$\begin{aligned}\int_a^b f(x)\,dx = &\sum_{k=0}^n f(x_k)\int_a^b l_k(x)\,dx \\ &+ \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)\,dx,\end{aligned} \quad (7.28)$$

where $l_k(x)$ (corresponding to the mesh points) is the Lagrange basis polynomial of degree n defined by (6.2). Here we get an approximation formula of the form

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^n c_k f(x_k), \quad (7.29)$$

where the weights c_k are defined by

$$c_k = \int_a^b l_k(x)\,dx. \quad (7.30)$$

Approximation formulas of the form (7.29) are called *quadrature formulas*. Those quadrature formulas when the weights c_k are defined by the integrals (7.30) are called *Newton–Cotes formulas*. If the end points of the interval a and b belong to the mesh points, then formulas (7.29)–(7.30) are called *closed Newton–Cotes formulas*, and if all mesh points belong to the open interval (a,b), then they are called *open Newton–Cotes formulas*.

We say that the *degree of precision* of a quadrature formula is n if the formula gives back the exact value of the definite integral for all polynomials with degree at most n, and there exists a polynomial of degree $n+1$ for which the quadrature formula is not exact. Therefore, the degree of precision of the $(n+1)$-point Newton–Cotes formula (7.29)–(7.30)

is at least n, since in this case the Lagrange polynomial L_n is identical to the function f. It is possible to show that for even n the $(n+1)$-point Newton–Cotes formulas are exact for polynomials with degree $n+1$ too.

Next we consider the closed Newton–Cotes formula for $n=1$. Let $x_0=a$, $x_1=b$ and $h=b-a$. Then

$$L_1(x) = f(x_0)\frac{x-x_1}{x_0-x_1} + f(x_1)\frac{x-x_0}{x_1-x_0},$$

so

$$\begin{aligned}\int_{x_0}^{x_1} L_1(x)\,dx &= f(x_0)\int_{x_0}^{x_1}\frac{x-x_1}{x_0-x_1}dx + f(x_1)\int_{x_0}^{x_1}\frac{x-x_0}{x_1-x_0}dx\\ &= \left[f(x_0)\frac{(x-x_1)^2}{2(x_0-x_1)} + f(x_1)\frac{(x-x_0)^2}{2(x_1-x_0)}\right]_{x_0}^{x_1}\\ &= \frac{h}{2}(f(x_0)+f(x_1)).\end{aligned}$$

The error of this formula, according to (7.28), is

$$\int_{x_0}^{x_1} f(x)\,dx - \frac{h}{2}(f(x_0)+f(x_1)) = \int_{x_0}^{x_1}\frac{f''(\xi(x))}{2}(x-x_0)(x-x_1)\,dx.$$

To simplify the formula of the error term we use that $(x-x_0)(x-x_1)<0$ for $x\in(x_0,x_1)$, and hence Theorem 2.6 can be used. Therefore, there exists $\eta\in(x_0,x_1)$ such that

$$\int_{x_0}^{x_1}\frac{f''(\xi(x))}{2}(x-x_0)(x-x_1)\,dx = \frac{f''(\eta)}{2}\int_{x_0}^{x_1}(x-x_0)(x-x_1)\,dx.$$

Hence

$$\begin{aligned}\int_{x_0}^{x_1} f(x)\,dx - \frac{h}{2}(f(x_0)+f(x_1)) &= \frac{f''(\eta)}{2}\int_{x_0}^{x_1}(x-x_0)^2 - h(x-x_0)\,dx\\ &= \frac{f''(\eta)}{2}\left[\frac{(x-x_0)^3}{3} - h\frac{(x-x_0)^2}{2}\right]_{x_0}^{x_1}\\ &= -\frac{h^3}{12}f''(\eta).\end{aligned}$$

We obtained the so-called *trapezoidal rule*:

$$\int_a^b f(x)\,dx = \frac{h}{2}(f(a)+f(b)) - \frac{h^3}{12}f''(\xi), \qquad \xi\in(a,b). \tag{7.31}$$

The name of the formula comes from the fact that $\frac{h}{2}(f(a)+f(b))$ gives back the area of the region bounded by the secant line of the function corresponding to the points a and b, the x-axis, and the vertical lines $x=a$ and $x=b$.

The trapezoidal rule gives a good approximation of the integral if the length of the interval is small. If we have a large interval, then we divide it into n subintervals of equal

length by the mesh points $x_i = a + ih$ $(i = 0, 1, \dots, n)$, where $h = (b-a)/n$, and we apply the trapezoidal rule for each subintervals:

$$\begin{aligned}\int_a^b f(x)\,dx &= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx \\ &= \sum_{i=1}^{n} \frac{h}{2}(f(x_{i-1}) + f(x_i)) - \frac{h^3}{12}\sum_{i=1}^{n} f''(\xi_i) \\ &= \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right) - \frac{nh^3}{12}\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i).\end{aligned}$$

We suppose that $f \in C^2[a,b]$. Then it follows from Theorem 2.2 that the average value $\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)$ can be replaced by a single function value of the form $f''(\xi)$. Therefore, using $hn = b - a$, we get

$$\int_a^b f(x)\,dx = \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right) - \frac{(b-a)h^2}{12} f''(\xi), \qquad \xi \in (a,b). \quad (7.32)$$

This formula is called *composite trapezoidal rule*.

Example 7.7. We compute approximate values of the integral $\int_0^1 x^2 e^x\,dx$ using the basic or composite trapezoidal rule with $h = 1$, $h = 0.5$ and $h = 0.25$, respectively. It can be checked that the exact value of the integral is $\int_0^1 x^2 e^x\,dx = e - 2 = 0.7182818$ (with 7 digits precision). For the first case we have

$$\int_0^1 x^2 e^x\,dx \approx \frac{1}{2}(0 + e) = 1.3591409,$$

where we computed the numerical values with 7 digits precision. The error in this case is 0.6408591. With $h = 0.5$ the composite trapezoidal rule gives

$$\int_0^1 x^2 e^x\,dx \approx \frac{0.5}{2}(0 + 0.5^2 e^{0.5} + e) = 0.8856606.$$

Hence its error is 0.1673788. Finally, for $h = 0.25$ we get

$$\int_0^1 x^2 e^x\,dx \approx \frac{0.25}{2}(0 + 0.25^2 e^{0.25} + 0.5^2 e^{0.5} + 0.75^2 e^{0.75} + e) = 0.7605963,$$

so its error is 0.0423145. We can observe that if the step size reduces to its half, then the corresponding error in the approximation reduces to its quarter, which indicates that the error in h is quadratic. □

Consider formula (7.28) for $n = 2$ and using equidistant mesh points, i.e., $x_0 = a$,

$x_1 = x_0 + h$, $x_2 = b$, $h = (b-a)/2$.

$$\begin{aligned}
\int_{x_0}^{x_2} L_2(x)\,dx &= f(x_0)\int_{x_0}^{x_2}\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}dx + f(x_1)\int_{x_0}^{x_2}\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}dx \\
&\quad + f(x_2)\int_{x_0}^{x_2}\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}dx \\
&= \frac{f(x_0)}{2h^2}\int_{x_0}^{x_2}(x-x_2+h)(x-x_2)\,dx - \frac{f(x_1)}{h^2}\int_{x_0}^{x_2}(x-x_0)(x-x_0-2h)\,dx \\
&\quad + \frac{f(x_2)}{2h^2}\int_{x_0}^{x_2}(x-x_0)(x-x_0-h)\,dx \\
&= \frac{f(x_0)}{2h^2}\left[\frac{(x-x_2)^3}{3}+h\frac{(x-x_2)^2}{2}\right]_{x_0}^{x_2} - \frac{f(x_1)}{h^2}\left[\frac{(x-x_0)^3}{3}-2h\frac{(x-x_0)^2}{2}\right]_{x_0}^{x_2} \\
&\quad + \frac{f(x_2)}{2h^2}\left[\frac{(x-x_0)^3}{3}-h\frac{(x-x_0)^2}{2}\right]_{x_0}^{x_2} \\
&= \frac{h}{3}(f(x_0)+4f(x_1)+f(x_2)).
\end{aligned}$$

The truncation error is

$$\int_{x_0}^{x_2}\frac{f'''(\xi(x))}{6}(x-x_0)(x-x_1)(x-x_2)\,dx.$$

Now there is a difference compared to the previous case: the function $(x-x_0)(x-x_1)(x-x_2)$ has opposite signs on the intervals (x_0,x_1) and (x_1,x_2), so Theorem 2.6 is not applicable on (x_0,x_2). We have a different method to simplify the formula for the error term. Let

$$\begin{aligned}
p(x) &:= \int_{x_0}^{x}(t-x_0)(t-x_1)(t-x_2)\,dt \\
&= \int_{x_0}^{x}(t-x_1+h)(t-x_1)(t-x_1-h)\,dt \\
&= \left[\frac{(t-x_1)^4}{4}-h^2\frac{(t-x_1)^2}{2}\right]_{x_0}^{x} \\
&= \frac{(x-x_1)^4}{4}-\frac{h^2(x-x_1)^2}{2}+\frac{h^4}{4} \\
&= \frac{1}{4}((x-x_1)^2-h^2)^2.
\end{aligned}$$

Then $p(x_0) = p(x_2) = 0$, so integration by parts gives

$$\int_{x_0}^{x_2}\frac{f'''(\xi(x))}{6}(x-x_0)(x-x_1)(x-x_2)\,dx = -\int_{x_0}^{x_2}\frac{d}{dx}\frac{f'''(\xi(x))}{6}p(x)\,dx.$$

p is a nonnegative function, hence applying Theorems 2.6 and 6.8, we get

$$\int_{x_0}^{x_2}\frac{f'''(\xi(x))}{6}(x-x_0)(x-x_1)(x-x_2)\,dx = -\frac{f^{(4)}(\eta)}{24}\int_{x_0}^{x_2}p(x)\,dx = -\frac{h^5}{90}f^{(4)}(\eta).$$

We have proved the relation

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}(f(x_0)+4f(x_1)+f(x_2)) - \frac{h^5}{90}f^{(4)}(\eta), \qquad \eta\in(x_0,x_2), \tag{7.33}$$

which is called *Simpson's rule*.

This error formula yields that the Simpson's rule is precise for third-order polynomials, since then $f^{(4)}$ is identically equal to 0. On the other hand, the order of approximation in h is five. Similar higher order of precision can be shown for all Newton–Cotes formulas with even n.

Similarly to the composite trapezoidal rule, we can derive the *composite Simpson's rule*: We divide the interval $[a,b]$ into $2n$ equal parts, so let $h=(b-a)/2n$. Then

$$\int_a^b f(x)\,dx = \frac{h}{3}\left(f(x_0)+4\sum_{i=1}^{n} f(x_{2i-1})+2\sum_{i=1}^{n-1} f(x_{2i})+f(x_{2n})\right) - \frac{(b-a)h^4}{180}f^{(4)}(\xi), \qquad \xi\in(a,b). \tag{7.34}$$

Example 7.8. Compute the approximate values of $\int_0^1 x^2e^x\,dx$ using (composite) Simpson's formula with $h=0.5$, $h=0.25$ and $h=0.125$. First we get

$$\int_0^1 x^2e^x\,dx \approx \frac{0.5}{3}(0+4\cdot 0.5^2e^{0.5}+e) = 0.7278339.$$

The error is 0.0095520. For $h=0.25$ we apply the composite Simpson's formula:

$$\int_0^1 x^2e^x\,dx \approx \frac{0.25}{3}(0+4\cdot 0.25^2e^{0.25}+2\cdot 0.5^2e^{0.5}+4\cdot 0.75^2e^{0.75}+e) = 0.7189082.$$

Its error is 0.0006264. Finally, for $h=0.125$ we get

$$\int_0^1 x^2e^x\,dx \approx \frac{0.125}{3}\Big(0+4\cdot 0.125^2e^{0.125}+2\cdot 0.25^2e^{0.25}+4\cdot 0.375^2e^{0.375}+2\cdot 0.5^2e^{0.5} + 4\cdot 0.625^2e^{0.625}+2\cdot 0.75^2e^{0.75}+4\cdot 0.875^2e^{0.875}+e\Big) = 0.7183215,$$

which has the error 0.0000396. □

Next we present some other closed Newton–Cotes formulas.

Simpson's $\frac{3}{8}$ formula:

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}\Big(f(x_0)+3f(x_1)+3f(x_2)+f(x_3)\Big) - \frac{3h^5}{80}f^{(4)}(\xi) \tag{7.35}$$

$n=4$:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}\Big(7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)\Big) - \frac{8h^7}{945}f^{(6)}(\xi) \tag{7.36}$$

Finally, we present some open Newton–Cotes formulas:

$$\int_{x_{-1}}^{x_1} f(x)\,dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi), \tag{7.37}$$

$$\int_{x_{-1}}^{x_2} f(x)\,dx = \frac{3h}{2}\Big(f(x_0)+f(x_1)\Big) + \frac{3h^3}{4}f''(\xi), \tag{7.38}$$

$$\int_{x_{-1}}^{x_3} f(x)\,dx = \frac{4h}{3}\Big(2f(x_0)-f(x_1)+2f(x_2)\Big) + \frac{14h^5}{45}f^{(4)}(\xi), \tag{7.39}$$

$$\int_{x_{-1}}^{x_4} f(x)\,dx = \frac{5h}{24}\Big(11f(x_0)+f(x_1)+f(x_2)+11f(x_3)\Big) + \frac{95h^5}{144}f^{(4)}(\xi). \tag{7.40}$$

We close this section with the investigation of the numerical stability of the integration.

Theorem 7.9. *Let $\sum_{i=1}^n c_i f(x_i)$ be a quadrature formula which is exact for constant functions and each coefficient c_i is positive. Let y_i be an approximate value of the exact function value $f(x_i)$, and suppose $|y_i - f(x_i)| \le \varepsilon$. Then*

$$\left|\sum_{i=1}^n c_i f(x_i) - \sum_{i=1}^n c_i y_i\right| \le \varepsilon(b-a).$$

Proof. According to the assumptions, $(b-a) = \int_a^b 1\,dx = \sum_{i=1}^n c_i$, therefore,

$$\left|\sum_{i=1}^n c_i f(x_i) - \sum_{i=1}^n c_i y_i\right| \le \sum_{i=1}^n c_i |f(x_i)-y_i| \le \varepsilon \sum_{i=1}^n c_i = \varepsilon(b-a).$$

□

We note that all quadrature formulas we presented in this section were exact for constant functions, and most of them had positive weights. Therefore, all such formulas are stable for the rounding error.

Exercises

1. Compute approximate values of the integrals using the trapezoidal rule with step sizes $h = 0.5, 0.25, 0.125$, respectively:
   (a) $\int_0^1 \sin^3 x\,dx$,
   (b) $\int_1^2 \ln(x+1)\,dx$,
   (c) $\int_1^2 e^{1/x}\,dx$.
2. Repeat Exercise 1 using the Simpson's rule.
3. Repeat Exercise 1 using formulas (7.35)-(7.36).

4. Repeat Exercise 1 using formulas Newton–Cotes Formulas (7.37)-(7.40).
5. Prove that the midpoint formula (7.27) gives back the sum of the areas under tangent lines at the midpoints of the intervals $[x_i, x_{i+1}]$.
6. Show that the midpoint formula is a Newton–Cotes formula, and derive its error term.
7. Derive formulas (7.35)-(7.36) (without computing the error terms).
8. Derive formulas (7.37)-(7.40) (without computing the error terms).

7.4. Gaussian Quadrature

In the previous section we have seen that the Newton–Cotes formulas give back the exact value of the integral for polynomials with certain degree. Now we would like to derive quadrature formulas with similar property. Consider the general quadrature formula

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} c_i f(x_i).$$

We have the following statement:

Theorem 7.10. *A quadrature formula*

$$Q(f) := \sum_{i=1}^{n} c_i f(x_i) \tag{7.41}$$

is exact for polynomials $p(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_0$ of degree at most m if and only if it is exact for the monomials x^i for all $i = 0, 1, \ldots, m$.

Proof. If Q is exact for all polynomials with degree at most m, it certainly implies that it is exact for all monomials x^i for all $i = 0, 1, \ldots, m$.

Suppose now that Q is exact for the monomials x^i for all $i = 0, 1, \ldots, m$. Then the linearity of the integral and the quadrature formula Q yield that

$$\begin{aligned}
&\int_a^b a_m x^m + a_{m-1}x^{m-1} + \cdots + a_0\,dx \\
&\qquad = a_m \int_a^b x^m\,dx + a_{m-1}\int_a^b x^{m-1}\,dx + \cdots + a_0 \int_a^b 1\,dx \\
&\qquad = a_m Q(x^m) + a_{m-1}Q(x^{m-1}) + \cdots + a_0 Q(1) \\
&\qquad = Q(a_m x^m + a_{m-1}x^{m-1} + \cdots + a_0).
\end{aligned}$$

□

The quadrature formula Q defined by (7.41) contains $2n$ number of parameters, c_i, x_i $(i = 1, 2, \ldots, n)$. The previous theorem indicates that such a quadrature formula can be exact for polynomials with degree at most $2n - 1$, since it also contains $2n$ coefficients. Then Theorem 7.10 yields that a quadrature formula Q is exact for polynomials of degree

at most $2n-1$ if and only if the following $2n$ number of equations hold:

$$\begin{aligned}
\int_a^b 1\,dx &= \sum_{i=1}^n c_i \\
\int_a^b x\,dx &= \sum_{i=1}^n c_i x_i \\
\int_a^b x^2\,dx &= \sum_{i=1}^n c_i x_i^2 \\
&\vdots \quad \vdots \\
\int_a^b x^{2n-1}\,dx &= \sum_{i=1}^n c_i x_i^{2n-1}
\end{aligned} \tag{7.42}$$

The quadrature formula of the form (7.41) where the parameters are the solutions of the nonlinear system (7.42) is called n-point *Gaussian quadrature* formula.

Consider the special case when $n = 2$ and $[a, b] = [-1, 1]$. Then system (7.42) is equivalent to the system

$$\begin{aligned}
2 &= c_1 + c_2 \\
0 &= c_1 x_1 + c_2 x_2 \\
\frac{2}{3} &= c_1 x_1^2 + c_2 x_2^2 \\
0 &= c_1 x_1^3 + c_2 x_2^3.
\end{aligned}$$

It can be checked that this system has a unique solution (apart from the order): $c_1 = c_2 = 1$ and $x_1 = -\frac{\sqrt{3}}{3}$, $x_2 = \frac{\sqrt{3}}{3}$. So the two-point Gaussian quadrature formula is

$$\int_{-1}^1 f(x)\,dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right). \tag{7.43}$$

Example 7.11. We compute the approximation of the integral of $f(x) = e^x$ on the interval $[-1, 1]$. The Gaussian formula (7.43) yields

$$\int_{-1}^1 e^x\,dx \approx e^{-\frac{\sqrt{3}}{3}} + e^{\frac{\sqrt{3}}{3}} = 2.3426961.$$

Comparing it with the exact value $e - 1/e = 2.350424$ we get that the error of the approximation is 0.0077062, which is small, compared to the simplicity of the formula.

□

We need the notion of orthogonal functions. The functions f and g are called *orthogonal* on the interval $[a, b]$ if

$$\int_a^b f(x)g(x)\,dx = 0.$$

We show that there exists a sequence of functions $(P_i)_{i=0,1,\dots}$ which are pairwise orthogonal on the interval $[-1,1]$, and P_i is a polynomial of degree i. Let $P_0(x) := 1$ and $P_1(x) := x$. Then P_0 and P_1 are orthogonal on $[-1,1]$. We are looking for P_2 in the form $P_2(x) = x^2 + a_{2,1}P_1(x) + a_{2,0}P_0(x)$. Then the requested orthogonality yields

$$\begin{aligned} 0 &= \int_{-1}^{1} P_2(x)P_0(x)\,dx \\ &= \int_{-1}^{1} x^2P_0(x)\,dx + a_{2,1}\int_{-1}^{1} P_1(x)P_0(x)\,dx + a_{2,0}\int_{-1}^{1} P_0^2(x)\,dx \\ &= \int_{-1}^{1} x^2P_0(x)\,dx + a_{2,0}\int_{-1}^{1} P_0^2(x)\,dx, \end{aligned}$$

which gives

$$a_{2,0} = -\frac{\int_{-1}^{1} x^2P_0(x)\,dx}{\int_{-1}^{1} P_0^2(x)\,dx}.$$

Similarly,

$$\begin{aligned} 0 &= \int_{-1}^{1} P_2(x)P_1(x)\,dx \\ &= \int_{-1}^{1} x^2P_1(x)\,dx + a_{2,1}\int_{-1}^{1} P_1^2(x)\,dx + a_{2,0}\int_{-1}^{1} P_0(x)P_1(x)\,dx \\ &= \int_{-1}^{1} x^2P_1(x)\,dx + a_{2,1}\int_{-1}^{1} P_1^2(x)\,dx, \end{aligned}$$

so

$$a_{2,1} = -\frac{\int_{-1}^{1} x^2P_1(x)\,dx}{\int_{-1}^{1} P_1^2(x)\,dx}.$$

We found a unique P_2 of this form. We can continue this procedure. If $P_0,\dots,P_i$ are already defined, then we are looking for P_{i+1} in the form

$$P_{i+1}(x) = x^{i+1} + a_{i+1,i}P_i(x) + \cdots + a_{i+1,0}P_0(x). \tag{7.44}$$

Then, similarly to the previous computation, we get

$$a_{i+1,j} = -\frac{\int_{-1}^{1} x^{i+1}P_j(x)\,dx}{\int_{-1}^{1} P_j^2(x)\,dx}, \qquad j = 0,1,\dots,i, \tag{7.45}$$

so P_{i+1} can be defined uniquely. This method is called *Gram–Schmidt orthogonalization*, and the resulting polynomial P_i is called *Legendre polynomial* of degree i. The formulas

of the first five Legendre polynomials are:

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= x^2 - \frac{1}{3}, \\ P_3(x) &= x^3 - \frac{3}{5}x, \\ P_4(x) &= x^4 - \frac{6}{7}x^2 + \frac{3}{35}. \end{aligned}$$

It can be shown that the Legendre polynomials satisfy the recursion

$$P_{n+1}(x) = xP_n(x) - \frac{n^2}{4n^2-1}P_{n-1}(x). \tag{7.46}$$

The next theorem summarizes the most important properties of the Legendre polynomials.

Theorem 7.12. *Let P_i be the ith Legendre polynomial. Then*

1. *P_i is orthogonal to any polynomial with degree at most $i-1$.*
2. *P_i is even if i is even, and it is odd if i is odd.*
3. *P_i has i distinct real roots in the interval $(-1,1)$, and they are symmetric to the origin.*
4. *If $(p_i)_{i=0,1,\dots}$ is a sequence of polynomials of degree (exactly) i, which are pairwise orthogonal, then $p_i(x) = c_iP_i(x)$ for all i for some constant $c_i \neq 0$.*

The next theorem shows that the mesh points of the n-point Gaussian quadrature formula defined on the interval $[-1,1]$ are the roots of the nth-order Legendre polynomial P_n.

Theorem 7.13. *Let $x_1, x_2, \dots, x_n$ be the roots of the nth Legendre polynomial P_n, and let*

$$c_i = \int_{-1}^{1} \frac{(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}\,dx. \tag{7.47}$$

Then, for any polynomial p of degree at most $2n-1$, it follows

$$\int_{-1}^{1} p(x)\,dx = \sum_{i=1}^{n} c_i p(x_i).$$

The next result gives the truncation error of the Gaussian quadrature.

Theorem 7.14. *Let $f \in C^{2n}[-1,1]$. Then there exists $\xi \in (-1,1)$ such that the n-point Gaussian quadrature formula satisfies*

$$\int_{-1}^{1} f(x)\,dx = \sum_{k=1}^{n} c_k f(x_k) + \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} P_n^2(x)\,dx.$$

It can be shown that the error term in the previous theorem has the form

$$\frac{\pi f^{(2n)}(\xi)}{4^n (2n)!},$$

which gives that if $f^{(2n)}$ is bounded for all n with a bound independent of n, then the error of the Gaussian quadrature goes to 0 exponentially. Note that the error in the Newton–Cotes formulas tends to 0 only with polynomial speed if $n \to \infty$.

Table 7.6 presents the roots of the first several Legendre polynomials and the corresponding coefficients.

Table 7.6: The parameters of the Gaussian quadrature formulas

n	x_i	c_i
2	0.5773502692	1.0000000000
	-0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	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	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	-0.5384693101	0.4786286705
	-0.9061798459	0.2369268850

The Gaussian quadrature formulas can be applied to the case when the interval is $[-1,1]$. But in case of an arbitrary interval $[a,b]$, the new variable $x = ((b-a)t+a+b)/2$ transforms the computation of the integral to the interval $[-1,1]$:

$$\int_a^b f(x)\,dx = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{(b-a)t+a+b}{2}\right) dt.$$

Example 7.15. Approximate the integral $\int_0^1 x^2 e^x\,dx$ using the two-point Gaussian quadrature:

$$\begin{aligned}\int_0^1 x^2 e^x\,dx &= \frac{1}{2}\int_{-1}^{1} \left(\frac{t+1}{2}\right)^2 e^{(t+1)/2}\,dt \\ &\approx \frac{1}{2}\left(\left(\frac{-\sqrt{3}/3+1}{2}\right)^2 e^{(-\sqrt{3}/3+1)/2} + \left(\frac{\sqrt{3}/3+1}{2}\right)^2 e^{(\sqrt{3}/3+1)/2}\right) \\ &= 0.7119418.\end{aligned}$$

The error of this approximation is 0.0063400. □

Exercises

1. Apply the 2-point Gaussian quadrature to the integrals given in Exercise 1 of the previous section.
2. Apply the 3-, 4- and 5-point Gaussian quadrature formulas to the integrals given in Exercise 1 of the previous section.

Chapter 8
Minimization of Functions

In this chapter we investigate the minimization of single and several variable real functions. We study only the minimization, since a function $f(x)$ takes its maximum at a point where the corresponding function $-f(x)$ takes its minimum, so finding a maximum of a function can be reduced to minimization.

We classify minimization algorithms into three groups: methods which do not use derivatives, methods which use only first and which use also second derivatives of a function. In the first class we study the golden section search, the simplex and the Nelder–Mead methods. In the second class we consider the gradient method, and in the third class we define the Newton's method. The quasi-Newton methods can be considered as algorithm in the third class where not the exact values of the derivatives, but their approximate values are used.

8.1. Review of Calculus

Theorem 8.1. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be partially differentiable with respect to all variables. Then if f has a local extremum at the point $\mathbf{a} \in \mathbb{R}^n$, then $\frac{\partial f(\mathbf{a})}{\partial x_i} = 0$ holds for all $i = 1, \dots, n$.*

If $f \in C^2$ and $f'(\mathbf{a}) = \mathbf{0}$ for some $\mathbf{a} \in \mathbb{R}^n$, moreover, the Hessian matrix $f''(\mathbf{a})$ is positive (negative) definite, then f has a local minimum (maximum) at the point $\mathbf{a}$.

For two-variable functions we have the following special case of the previous result.

Theorem 8.2. *Let $f\colon \mathbb{R}^2 \to \mathbb{R}$, $f \in C^2$. Then if f has a local extremum at the point (a, b), then*

$$\frac{\partial f}{\partial x}(a,b) = 0, \qquad \frac{\partial f}{\partial y}(a,b) = 0 \tag{8.1}$$

holds.

On the other hand, if relation (8.1) holds at a point (a, b), and

$$D(a,b) := \frac{\partial^2 f}{\partial x^2}(a,b) \cdot \frac{\partial^2 f}{\partial y^2}(a,b) - \left(\frac{\partial^2 f}{\partial x\, \partial y}(a,b)\right)^2 > 0,$$

then f has a local extremum point at (a, b). Moreover, f has a local maximum at (a, b) if $\frac{\partial^2 f}{\partial x^2}(a,b) < 0$, and it has a local minimum at (a, b) if $\frac{\partial^2 f}{\partial x^2}(a,b) > 0$. If $D(a, b) < 0$, then f has no extremum at (a, b).

8.2. Golden Section Search Method

Let $f\colon [a,b] \to \mathbb{R}$ be continuous, and suppose that it is a *unimodal function*, i.e., it has a unique minimum point in the interval $[a,b]$. This holds if, e.g., the function is convex on $[a,b]$, but it is not necessary (see, e.g. the second and third functions in Figure 8.1). Let p be the (unique) minimum point of f.

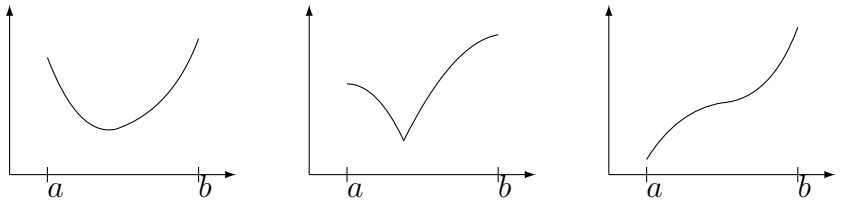

Figure 8.1: Unimodal functions

The *golden section search method* is similar to the bisection method in the sense that we define a sequence of nested intervals which all contains the minimum point p of f: Let $a < y < x < b$. If $f(x) > f(y)$, then $p \in [a,x]$, otherwise $p \in [y,b]$ holds. (See Figure 8.2.) Then we repeat the procedure with the interval $[a,x]$ or $[y,b]$.

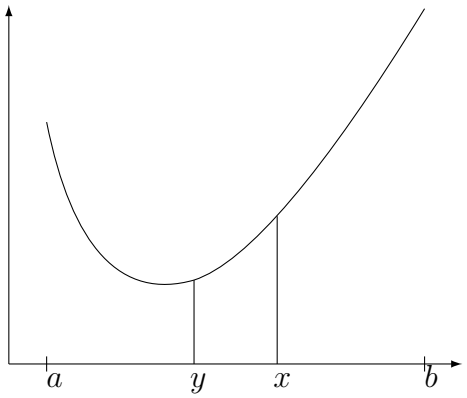

Figure 8.2:

We define the points x and y so that the length of the intervals $[a,x]$ and $[y,b]$ be the same: $x - a = b - y = r(b-a)$ for some $0 < r < 1$. Then

$$x = a + r(b-a), \qquad y = a + (1-r)(b-a) \tag{8.2}$$

hold. The assumption $x > y$ implies that $0.5 < r < 1$ must be satisfied. We denote the next interval by $[a',b']$. We specify the next mesh points x' and y' by the rule (8.2), and comparing the functions values $f(x')$ and $f(y')$ we determine the next interval. We have not defined the ratio r yet. In case of the golden section search method, r is defined so that one of the new mesh points x' and y' should coincide with one of the previous mesh points in order in each steps we should evaluate only one new function value.

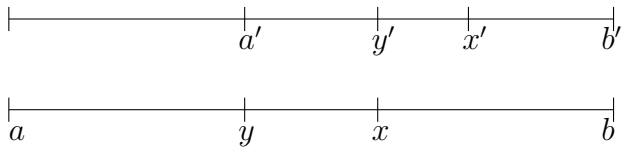

Figure 8.3:

Figure 8.3 demonstrates the situation when in the next step the minimum point is located in the right interval $[y, b]$. Then we require that $y' = x$ be a mesh point in the next step. Then the following relations are satisfied:

$$\begin{aligned} a + r(b-a) &= y' \\ &= a' + (1-r)(b'-a') \\ &= y + (1-r)(b-y) \\ &= a + (1-r)(b-a) + (1-r)(b-a-(1-r)(b-a)), \end{aligned}$$

and so

$$r = 1 - r + (1-r)(1-(1-r)),$$

which yields equation

$$r^2 + r - 1 = 0 \tag{8.3}$$

for the ratio r. Its positive solution is $r = (\sqrt{5}-1)/2 \approx 0.61834$. This is the ratio of the *golden section*, since r satisfies the equation

$$\frac{r}{1-r} = \frac{1}{r}.$$

In the opposite case when the minimum point is located in the interval $[a, x]$, and we select x' and y' so that $x' = y$ be satisfied. It can be shown easily (see Exercise 3) that this yields the same equation (8.3).

Algorithm 8.3. Golden section search method

INPUT: $f(x)$ - function to minimze
$[a, b]$ - interval
ε - tolerance
OUTPUT: p - approximation of the minimum point

$r \leftarrow (\sqrt{5}-1)/2$
$x \leftarrow a + r(b-a)$
$y \leftarrow a + (1-r)(b-a)$
$fx \leftarrow f(x)$
$fy \leftarrow f(y)$
while $(b-a) > \varepsilon$ **do**
  if $fx > fy$ **do**
    $b \leftarrow x$

$x \leftarrow y$
$fx \leftarrow fy$
$y \leftarrow a + (1 - r)(b - a)$
$fy \leftarrow f(y)$
else do
$a \leftarrow y$
$y \leftarrow x$
$fy \leftarrow fx$
$x \leftarrow a + r(b - a)$
$fx \leftarrow f(x)$
end do
end do
output$((a + b)/2)$

The next result can be shown.

Theorem 8.4. *Let $f \in C[a, b]$ be a unimodal function. Then the golden section search method converges to the minimum point of the function f.*

It is easy to compute that the length of the interval after n steps is $(b - a)r^n$. Hence to reach ε tolerance in Algorithm 8.3

$$n \geq \frac{\log \frac{\varepsilon}{b-a}}{\log r} \tag{8.4}$$

steps are required.

Example 8.5. Find the minimum point of the function $f(x) = x^2 - 0.8x + 1$. It can be easily checked that its minimum point is $p = 0.4$. We applied Algorithm 8.3 with the starting interval $[-1, 2]$ and tolerance $\varepsilon = 0.005$. Formula (8.4) yields that $n \geq 13.29337586$ steps are needed to reach the required precision. The corresponding numerical results can be seen in Table 8.1. Therefore, the minimum point is located in the interval $[0.3977741449, 0.4013328688]$. The Algorithm 8.3 is formulated so that its output is the midpoint of the last interval, i.e., 0.3995535068. □

Exercises

1. Approximate the minimum point of the following functions using the golden section search method on the given interval:

   (a) $f(x) = x^3 - 3x + 1$, $x \in [-1, 2]$, (b) $f(x) = |\cos x|$, $x \in [0, 2]$,

   (c) $f(x) = 1 - 10xe^{-x}$, $x \in [0, 2]$, (d) $f(x) = \cos(x^2 - x)$, $x \in [1, 3]$.

2. Apply the golden section search method for the function $f(x) = -1/x^2$ on the interval $[-1, 1]$. What do you observe?

Table 8.1: Golden section search method, $f(x) = x^2 - 0.8x + 1$

k	$[a_k, b_k]$	y_k	x_k
0	[-1.0000000000, 2.0000000000]	0.1458980338	0.8541019662
1	[-1.0000000000, 0.8541019662]	-0.2917960675	0.1458980338
2	[-0.2917960675, 0.8541019662]	0.1458980338	0.4164078650
3	[0.1458980338, 0.8541019662]	0.4164078650	0.5835921350
4	[0.1458980338, 0.5835921350]	0.3130823038	0.4164078650
5	[0.3130823038, 0.5835921350]	0.4164078650	0.4802665738
6	[0.3130823038, 0.4802665738]	0.3769410125	0.4164078650
7	[0.3769410125, 0.4802665738]	0.4164078650	0.4407997213
8	[0.3769410125, 0.4407997213]	0.4013328688	0.4164078650
9	[0.3769410125, 0.4164078650]	0.3920160087	0.4013328688
10	[0.3920160087, 0.4164078650]	0.4013328688	0.4070910050
11	[0.3920160087, 0.4070910050]	0.3977741449	0.4013328688
12	[0.3977741449, 0.4070910050]	0.4013328688	0.4035322811
13	[0.3977741449, 0.4035322811]	0.3999735572	0.4013328688
14	[0.3977741449, 0.4013328688]	0.3991334565	0.3999735572

3. Prove that if $[a', b'] = [a, x]$ is selected in golden section search then $x' = y$ is satisfied if r is a solution of equation (8.3).
4. Prove Theorem 8.4.
5. Check formula (8.4).

8.3. Simplex Method

An n-dimensional *simplex* is a convex hull of $n+1$ number of n-dimensional vectors, i.e., the closed set

$$\{\alpha_0 \mathbf{x}^{(0)} + \cdots + \alpha_n \mathbf{x}^{(n)} \colon\ 0 \le \alpha_i \le 1, \quad \alpha_0 + \cdots + \alpha_n \le 1\},$$

where the vectors $\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \ldots, \mathbf{x}_n - \mathbf{x}_0$ are linearly independent. The vectors $\mathbf{x}^{(0)}$,...,$\mathbf{x}^{(n)}$ are called the vertices of the simplex. The 1-dimensional simplexes are the line segments, the 2-dimensional simplexes are the triangles, and the 3-dimensional simplexes are the tetrahedrons.

The *simplex method* is used to approximate the minimum point of a function of n variables. Consider a starting n-dimensional simplex. First we find the "worst" vertex, i.e., the vertex where the function takes the largest function value. Let this point be the vector $\mathbf{x}^{(j)}$. Then we reflect the simplex over the center of the best n vertices, i.e., to the point

$$\mathbf{x}_c := \frac{1}{n} \sum_{\substack{i=0 \\ i \ne j}}^{n} \mathbf{x}^{(i)}.$$

The reflected point is given by the formula

$$\mathbf{x}_r = 2\mathbf{x}_c - \mathbf{x}^{(j)}.$$

If $f(\mathbf{x}_r)$ is not smaller than the largest function value of the previous step, i.e., $f(\mathbf{x}^{(j)})$, then we discard the reflection, and instead of it, we shrink the simplex to half of its size from its "best" vertex: let $\mathbf{x}^{(k)}$ be the best vertex, i.e., the vertex where the function takes the smallest function value. Then we recompute all the other vertices by the formula

$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(k)} + \frac{1}{2}(\mathbf{x}^{(i)} - \mathbf{x}^{(k)}), \quad i = 0, 1, \dots, k-1, k+1, \dots, n.$$

We repeat the previous steps for the resulting (reflected or shrinked) simplex.

We can define several different stopping criteria to this method, or we can use combinations of these methods. For example, we can stop the method when the simplex becomes smaller than a predefined tolerance size. The size of the simplex can be defined, e.g., as the length of its longest edge, i.e., by the number $\max\{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| : \ i, j = 0, \dots, n\}$. Another option is that we apply the stopping criterion $|f_{k+1} - f_k| < \varepsilon$, where f_k denotes the function value at the center of the kth simplex. A third criterion can be the following: Let $\bar{f}$ be the average of the function values at the vertices, and σ be its standard deviation, i.e.,

$$\bar{f} := \frac{1}{n+1}\sum_{i=0}^{n} f(\mathbf{x}^{(i)}), \qquad \sigma := \sqrt{\frac{1}{n+1}\sum_{i=0}^{n}(f(\mathbf{x}^{(i)}) - \bar{f})^2}.$$

We interrupt the iteration when σ becomes smaller than a tolerance. The center of the simplex can be used as an approximation of the minimum point. Finally, we can apply conditions (i) or (ii) of Section 4.4 for the sequence of the center points to set up a stopping criterion.

Example 8.6. Find the minimum point of the function $f(x, y) = (x^2 - 2y)^2 + 2(x-1)^2$. It is easy to see that the (global) minimum point of the function is $(1, 0.5)$, and the minimal function value is 0. We use the simplex method to approximate the minimum point. We use the starting simplex corresponding to the vertices $(-2, 4)$, $(-1, 4)$ and $(-1.5, 5)$. The numerical values of the first 25 steps of the method can be seen in Table 8.2. The center of the 25th simplex is $(0.9063, 0.3542)$, which is a good approximation of the exact minimum point. The corresponding function value is 0.0303 which is close to the true minimum 0. In Figure 8.4 the contour lines (level curves) of the function and the sequence of the simplexes (triangles) can be seen. The blue dot represents the exact minimum point. □

A variant of the simplex method is the *Nelder–Mead method*. Here we reflect, expand or contract the simplex in the following way. Suppose that in each steps the vertices are indexed so that $f(\mathbf{x}^{(0)}) \le f(\mathbf{x}^{(1)}) \le \dots \le f(\mathbf{x}^{(n)})$. Then $\mathbf{x}^{(n)}$ is the "worst" vertex, so we reflect it over the center of the remaining points, i.e., over the point

$$\mathbf{x}_c = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}^{(i)}.$$

The reflected point is $\mathbf{x}_r = 2\mathbf{x}_c - \mathbf{x}^{(n)}$. We evaluate the function value $f(\mathbf{x}_r)$. We distinguish three cases: (i) $f(\mathbf{x}^{(0)}) < f(\mathbf{x}_r) < f(\mathbf{x}^{(n-1)})$, (ii) $f(\mathbf{x}_r) \le f(\mathbf{x}^{(0)})$, so $\mathbf{x}_r$ would be the new best vertex, and (iii) $f(\mathbf{x}_r) \ge f(\mathbf{x}^{(n-1)})$, i.e., $\mathbf{x}_r$ would be the new worst vertex.

Table 8.2: Simplex method, $f(x,y) = (x^2 - 2y)^2 + 2(x-1)^2$

k	$\mathbf{x}^{(k,1)}$	$\mathbf{x}^{(k,2)}$	$\mathbf{x}^{(k,3)}$	$f(\mathbf{x}^{(k,1)})$	$f(\mathbf{x}^{(k,2)})$	$f(\mathbf{x}^{(k,3)})$
0	(-1.000, 4.000)	(-2.000, 4.000)	(-1.500, 5.000)	57.000	34.000	72.563
1	(-2.000, 4.000)	(-1.000, 4.000)	(-1.500, 3.000)	34.000	57.000	26.563
2	(-1.500, 3.000)	(-2.000, 4.000)	(-2.500, 3.000)	26.563	34.000	24.563
3	(-2.500, 3.000)	(-1.500, 3.000)	(-2.000, 2.000)	24.563	26.563	18.000
4	(-2.000, 2.000)	(-2.250, 2.500)	(-1.750, 2.500)	18.000	21.129	18.879
5	(-2.000, 2.000)	(-1.750, 2.500)	(-1.500, 2.000)	18.000	18.879	15.563
6	(-1.500, 2.000)	(-2.000, 2.000)	(-1.750, 1.500)	15.563	18.000	15.129
7	(-1.750, 1.500)	(-1.500, 2.000)	(-1.250, 1.500)	15.129	15.563	12.191
8	(-1.250, 1.500)	(-1.750, 1.500)	(-1.500, 1.000)	12.191	15.129	12.563
9	(-1.250, 1.500)	(-1.500, 1.000)	(-1.000, 1.000)	12.191	12.563	9.000
10	(-1.000, 1.000)	(-1.250, 1.500)	(-0.750, 1.500)	9.000	12.191	12.066
11	(-1.000, 1.000)	(-0.750, 1.500)	(-0.500, 1.000)	9.000	12.066	7.563
12	(-0.500, 1.000)	(-1.000, 1.000)	(-0.750, 0.500)	7.563	9.000	6.316
13	(-0.750, 0.500)	(-0.500, 1.000)	(-0.250, 0.500)	6.316	7.563	4.004
14	(-0.250, 0.500)	(-0.750, 0.500)	(-0.500, 0.000)	4.004	6.316	4.563
15	(-0.250, 0.500)	(-0.500, 0.000)	(0.000, 0.000)	4.004	4.563	2.000
16	(0.000, 0.000)	(-0.250, 0.500)	(0.250, 0.500)	2.000	4.004	2.004
17	(0.000, 0.000)	(0.250, 0.500)	(0.500, 0.000)	2.000	2.004	0.563
18	(0.500, 0.000)	(0.250, 0.000)	(0.375, 0.250)	0.563	1.129	0.910
19	(0.500, 0.000)	(0.375, 0.250)	(0.625, 0.250)	0.563	0.910	0.293
20	(0.625, 0.250)	(0.500, 0.000)	(0.750, 0.000)	0.293	0.563	0.441
21	(0.625, 0.250)	(0.750, 0.000)	(0.875, 0.250)	0.293	0.441	0.102
22	(0.875, 0.250)	(0.750, 0.250)	(0.813, 0.125)	0.102	0.129	0.239
23	(0.875, 0.250)	(0.750, 0.250)	(0.813, 0.375)	0.102	0.129	0.078
24	(0.813, 0.375)	(0.875, 0.250)	(0.938, 0.375)	0.078	0.102	0.024
25	(0.938, 0.375)	(0.875, 0.375)	(0.906, 0.313)	0.024	0.031	0.056

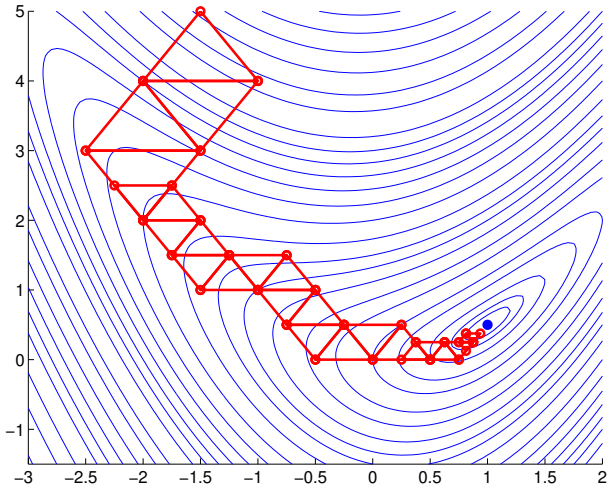

Figure 8.4: Simplex method.

In case (i) we replace $\mathbf{x}^{(n)}$ by $\mathbf{x}_r$ (i.e., we accept the reflection), and continue the iteration.

In case (ii) we expand the simplex in the direction of $\mathbf{x}_r$ hoping that we get an even

better point. Let

$$\mathbf{x}_e := \mathbf{x}_c + \alpha(\mathbf{x}_r - \mathbf{x}_c),$$

where $\alpha > 1$ is a fixed constant (a parameter of the method). If $f(\mathbf{x}_e) < f(\mathbf{x}^{(0)})$ holds, then the expansion is considered to be successful, and we replace $\mathbf{x}^{(n)}$ by $\mathbf{x}_e$. Otherwise we replace $\mathbf{x}^{(n)}$ by $\mathbf{x}_r$, i.e., the reflection is performed but we do not expand the simplex.

In case (iii) we think that the reflection is too far from $\mathbf{x}^{(n)}$, so we try to contract the simplex. Let

$$\mathbf{x}_z := \begin{cases} \mathbf{x}_c - \beta(\mathbf{x}_r - \mathbf{x}_c), & \text{if } f(\mathbf{x}^{(n)}) < f(\mathbf{x}_r), \\ \mathbf{x}_c + \beta(\mathbf{x}_r - \mathbf{x}_c), & \text{if } f(\mathbf{x}^{(n)}) \geq f(\mathbf{x}_r), \end{cases}$$

where $0 < \beta < 1$ is another parameter. If $f(\mathbf{x}_z) < \min\{f(\mathbf{x}^{(n)}),\ f(\mathbf{x}_r)\}$, then $\mathbf{x}^{(n)}$ is replaced by $\mathbf{x}_z$. Otherwise we shrink the simplex to its half size from its best point:

$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(0)} + \frac{1}{2}(\mathbf{x}^{(i)} - \mathbf{x}^{(0)}), \quad i = 1, \ldots, n.$$

Example 8.7. We apply the Nelder–Mead method with parameters $\alpha = 1.4$ and $\beta = 0.7$ for the function $f(x, y) = (x^2 - 2y)^2 + 2(x - 1)^2$ considered in Example 8.6. We start from the same initial simplex $(-2, 4)$, $(-1, 4)$ and $(-1.5, 5)$. The first 17 terms of the resulting sequence of vertices can be seen in Table 8.3 and in Figure 8.5. The center of the 17th triangle is $(1.0071, 0.5929)$, and the corresponding function value is 0.0295. We can observe that for this example the Nelder–Mead method converges faster to the minimum point than the simplex method. □

Table 8.3: Nelder–Mead method, $f(x, y) = (x^2 - 2y)^2 + 2(x - 1)^2$, $\alpha = 1.4$, $\beta = 0.7$

k	$\mathbf{x}^{(k,1)}$	$\mathbf{x}^{(k,2)}$	$\mathbf{x}^{(k,3)}$	$f(\mathbf{x}^{(k,1)})$	$f(\mathbf{x}^{(k,2)})$	$f(\mathbf{x}^{(k,3)})$
0	(-1.000, 4.000)	(-2.000, 4.000)	(-1.500, 5.000)	57.000	34.000	72.563
1	(-2.000, 4.000)	(-1.000, 4.000)	(-1.500, 2.600)	34.000	57.000	21.203
2	(-1.500, 2.600)	(-2.000, 4.000)	(-2.500, 2.600)	21.203	34.000	25.603
3	(-1.500, 2.600)	(-2.500, 2.600)	(-2.000, 1.200)	21.203	25.603	20.560
4	(-2.000, 1.200)	(-1.500, 2.600)	(-0.700, 0.920)	20.560	21.203	7.602
5	(-0.700, 0.920)	(-2.000, 1.200)	(-1.200,-0.480)	7.602	20.560	15.440
6	(-0.700, 0.920)	(-1.200,-0.480)	(0.520,-1.152)	7.602	15.440	7.088
7	(0.520,-1.152)	(-0.700, 0.920)	(1.464, 0.394)	7.088	7.602	2.270
8	(1.464, 0.394)	(0.520,-1.152)	(-0.192, 0.530)	2.270	7.088	3.891
9	(1.464, 0.394)	(-0.192, 0.530)	(0.555,-0.668)	2.270	3.891	3.097
10	(1.464, 0.394)	(0.555,-0.668)	(0.168, 0.330)	2.270	3.097	1.783
11	(0.168, 0.330)	(1.464, 0.394)	(0.999, 1.083)	1.783	2.270	1.362
12	(0.999, 1.083)	(0.168, 0.330)	(1.200, 0.487)	1.362	1.783	0.296
13	(1.200, 0.487)	(0.999, 1.083)	(0.448, 0.467)	0.296	1.362	1.147
14	(1.200, 0.487)	(0.448, 0.467)	(0.648,-0.129)	0.296	1.147	0.707
15	(1.200, 0.487)	(0.648,-0.129)	(0.591, 0.380)	0.296	0.707	0.505
16	(1.200, 0.487)	(0.591, 0.380)	(1.068, 0.828)	0.296	0.505	0.274
17	(1.068, 0.828)	(1.200, 0.487)	(0.754, 0.464)	0.274	0.296	0.251

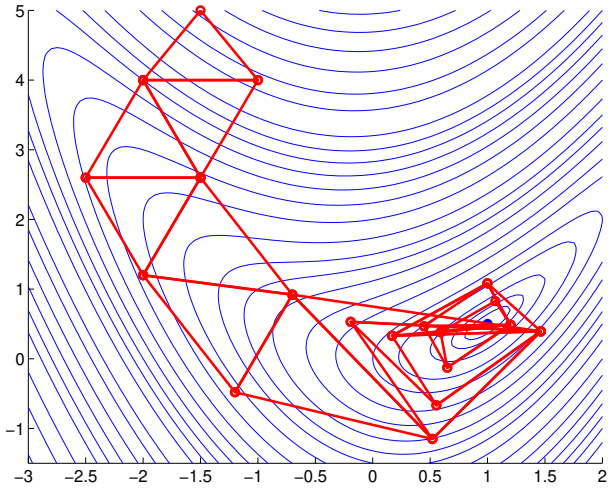

Figure 8.5: Nelder–Mead method with $\alpha = 1.8$ and $\beta = 0.6$.

Exercises

1. Find the minimum point of the functions

   (a) $f(x, y) = x^2 + 5y^2$, (b) $f(x, y) = x^2 + (x + y - 2)^2$,

   (c) $f(x, y) = 3x^2 + e^{(x-y)^2}$, (d) $f(x, y) = x^2 + \cos^2(x - y)$

   with the Nelder–Mead method. Use the method with different parameter values α and β (including $\alpha = 1 = \beta$, i.e., the simplex method).
2. Apply the Nelder–Mead method with some parameter values $\alpha > 1$ and $0 < \beta < 1$ for the function $f(x) = x^2 - y^2$ using the initial simplex vertices $[0, 1]$, $[0, -1]$, $[1, 0]$. What do you observe? What do you observe if you use the simplex method for the same problem?
3. Formulate the simplex method for functions of one variable, and apply it for the problems given in Exercise 1 of Section 8.2.
4. Consider the following method for minimization of real functions of two variables: let f be a function of two variables, $(p_1^{(0)}, p_2^{(0)})$ be a given initial point. Minimize the function of one variable $t \mapsto f(p_1^{(0)} + t, p_2^{(0)})$ (for example, with the simplex method defined in the previous exercise). Let t_1 be the minimum point, and define $(p_1^{(1)}, p_2^{(1)}) := (p_1^{(0)} + t_1, p_2^{(0)})$. Then minimize the function of single variable $t \mapsto f(p_1^{(1)}, p_2^{(1)} + t)$. Let t_2 be its minimum point, and then we repeat the method above starting from the point $(p_1^{(2)}, p_2^{(2)}) := (p_1^{(1)}, p_2^{(1)} + t_2)$. So repeatedly, minimizing the function along with x- and y-axes we get the next element of the sequence. Apply this method for the functions defined in Exercise 1. Compare the speed of the convergence with that of the Nelder–Mead method.

8.4. Gradient Method

Consider a function $f\colon \mathbb{R}^n \to \mathbb{R}$. It is known from calculus that at a point $\mathbf{p}$ the most rapid decrease of the function f is in the direction of the vector $-f'(\mathbf{p})$:

Theorem 8.8. *Let $f \in C^1$. Then the directional derivatives*

$$\lim_{t\to 0+} \frac{f(\mathbf{p}+t\mathbf{u})-f(\mathbf{p})}{t}, \qquad \|\mathbf{u}\|_2 = 1$$

has a minimum for the direction $\mathbf{u} = -f'(\mathbf{p})/\|f'(\mathbf{p})\|_2$.

A direction $\mathbf{u}$ is called a *descent* of a function f at the point $\mathbf{p}$ if there exists $\delta > 0$ such that $f(\mathbf{p}+t\mathbf{u}) < f(\mathbf{p})$ for all $0 < t < \delta$, i.e., the function decreases at the point $\mathbf{p}$ in the direction of $\mathbf{u}$. Theorem 8.8 can be interpreted so that the steepest descent of f at the point $\mathbf{p}$ is in the direction $-f'(\mathbf{p})$.

The *gradient method* is based on the previous observation that starting from a point $\mathbf{p}^{(0)}$ we should step forward in the direction of the negative gradient vector. This method is also called the *steepest descent method*. We define it as follows:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \alpha_k f'(\mathbf{p}^{(k)}), \tag{8.5}$$

where the scaling parameter α_k determines the step size. The gradient method (8.5) has several variants. The simplest case is when the step size is constant. Let $h > 0$ be fixed, and use the factor $\alpha_k = h/\|f'(\mathbf{p}^{(k)})\|_2$. Then the distance between the consecutive points is constant h. Then, in general, the method cannot approximate the exact minimum point better than h.

Another variant is that we select α_k so that

$$\phi_k(\alpha_k) = \min_{t\in\mathbb{R}} \phi_k(t)$$

be satisfied, where

$$\phi_k(t) := f\Big(\mathbf{p}^{(k)} - tf'(\mathbf{p}^{(k)})\Big). \tag{8.6}$$

Then in each step we have to minimize a function of a single variable along with the direction of the negative gradient. This version of the gradient method is called *optimal gradient method*.

Using the optimal gradient method we step forward from a point in the direction of the negative gradient into a point where the line is tangent to the contour line (level curve) of the function f. This implies that the consecutive directions are perpendicular to each other. (See Exercise 3.)

It can be shown that the optimal gradient method is locally linearly convergent. But the asymptotic error constant can be close to 1, so the convergence can be slow.

Example 8.9. We consider again the function $f(x,y) = (x^2-2y)^2 + 2(x-1)^2$ examined in Examples 8.6 and 8.7 and we use the gradient method to find its minimum point. First we use the gradient method with the scaling factor $\alpha_k = 0.3/\|f'(\mathbf{p}^{(k)})\|_2$, i.e., with the constant step size 0.3. The first 21 terms of the sequence can be seen in Figure 8.6 starting from the initial point $(-1,4)$ (red circles) and from the initial point $(0.5,3.5)$ (green circles). The sequences approximate the minimum point $(1,0.5)$ (blue dot) slowly, and oscillates around it. Note that, as it is known in calculus, the gradient vector is always perpendicular to the contour line through that point, so the gradient method steps in a direction perpendicular to the contour line.

Next we apply the optimal gradient method from the initial points $(-1, 4)$ (red circles) and $(0.5, 3.5)$ (green circles), respectively. We plotted the first 3 and 12 terms of the corresponding sequences in Figure 8.7. The first sequence gets very close to the minimizer (blue dot) in two steps, and then approaches further to the minimum point. The second sequence enters quickly into the "valley" of the contour lines containing the minimum point, but there it zigzags slowly towards the minimum point. □

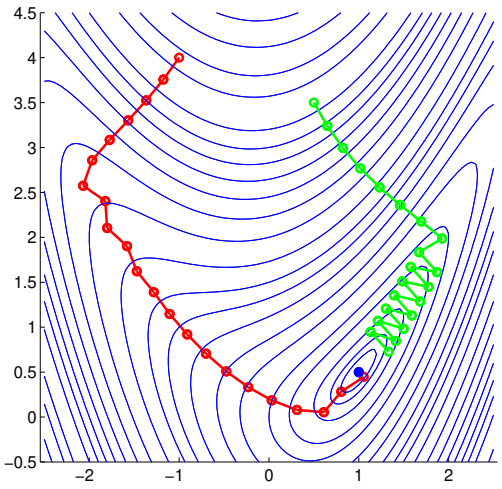

Figure 8.6: Gradient method with constant step size.

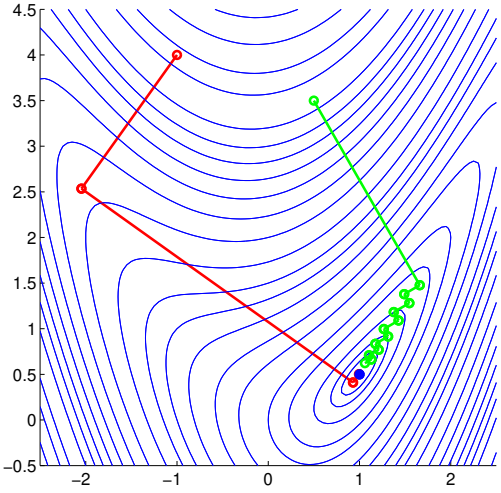

Figure 8.7: Optimal gradient method.

If we cannot or do not want to compute the gradient vector exactly, then we can use the following variant of the method (8.5):

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \alpha_k \mathbf{v}^{(k)}, \tag{8.7}$$

where the ith component of the vector $\mathbf{v}^{(k)}$ is defined by

$$v_i^{(k)} = \frac{1}{h}\Big(f(\mathbf{p}^{(k)} + h\mathbf{e}^{(i)}) - f(\mathbf{p}^{(k)})\Big), \qquad i = 1, \dots, n,$$

and here $\mathbf{e}^{(i)}$ is the ith unit vector.

Exercises

1. Apply the gradient method for the functions given in Exercise 1 of Section 8.3. Select any initial point, and use the constant step size $\alpha_k = h/\|f'(\mathbf{p}^{(k)})\|_2$ with some $h > 0$, and also use the optimal gradient method.
2. Repeat the previous problem using the scale $\alpha_k = h$ with some $h > 0$.
3. Compute the derivative of the function ϕ_k defined by (8.6). Using the value of the derivative at $t = \alpha_k$ show that the vectors $\mathbf{p}^{(k+2)} - \mathbf{p}^{(k+1)}$ and $\mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}$ are orthogonal.

8.5. Solving Linear Systems with Gradient Method

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$, and consider the quadratic function

$$g\colon \mathbb{R}^n \to \mathbb{R}, \qquad g(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c. \tag{8.8}$$

Using the notations $\mathbf{A} = (a_{ij})$, $\mathbf{x} = (x_1, \dots, x_n)^T$, $\mathbf{b} = (b_1, \dots, b_n)^T$ we have the following form of g:

$$g(x_1, \dots, x_n) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j - \sum_{i=1}^{n} b_ix_i + c.$$

Compute the partial derivative $\frac{\partial g}{\partial x_i}$. Since $a_{ij} = a_{ji}$, we get

$$\frac{\partial g}{\partial x_i}(x_1, \dots, x_n) = \frac{1}{2}\sum_{j=1}^{n}(a_{ij}x_j + a_{ji}x_j) - b_i = \sum_{j=1}^{n} a_{ij}x_j - b_i.$$

Therefore, in a vectorial form we have

$$g'(\mathbf{x}) = \left(\frac{\partial g}{\partial x_1}(\mathbf{x}), \dots, \frac{\partial g}{\partial x_n}(\mathbf{x})\right)^T = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{8.9}$$

Hence if $\mathbf{A}$ is invertible, then g has exactly one critical point, which is the solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Let $\bar{\mathbf{x}}$ be the critical point of g, and $\mathbf{x} = \bar{\mathbf{x}} + \Delta\mathbf{x}$.

$$\begin{aligned} g(\bar{\mathbf{x}} + \Delta\mathbf{x}) &= \frac{1}{2}(\bar{\mathbf{x}} + \Delta\mathbf{x})^T\mathbf{A}(\bar{\mathbf{x}} + \Delta\mathbf{x}) - \mathbf{b}^T(\bar{\mathbf{x}} + \Delta\mathbf{x}) + c \\ &= \frac{1}{2}\bar{\mathbf{x}}^T\mathbf{A}\bar{\mathbf{x}} + \frac{1}{2}\bar{\mathbf{x}}^T\mathbf{A}\Delta\mathbf{x} + \frac{1}{2}(\Delta\mathbf{x})^T\mathbf{A}\bar{\mathbf{x}} + \frac{1}{2}(\Delta\mathbf{x})^T\mathbf{A}\Delta\mathbf{x} \\ &\quad - \mathbf{b}^T\bar{\mathbf{x}} - \mathbf{b}^T\Delta\mathbf{x} + c. \end{aligned}$$

So using the relations $\mathbf{A} = \mathbf{A}^T$, $\bar{\mathbf{x}}^T\mathbf{A}\Delta\mathbf{x} = (\Delta\mathbf{x})^T\mathbf{A}\bar{\mathbf{x}}$, $\mathbf{b}^T\Delta\mathbf{x} = (\Delta\mathbf{x})^T\mathbf{b}$ and $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$, we get

$$\begin{aligned} g(\bar{\mathbf{x}} + \Delta\mathbf{x}) &= \frac{1}{2}\bar{\mathbf{x}}^T\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}^T\bar{\mathbf{x}} + (\Delta\mathbf{x})^T(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}) + \frac{1}{2}(\Delta\mathbf{x})^T\mathbf{A}\Delta\mathbf{x} + c \\ &= g(\bar{\mathbf{x}}) + \frac{1}{2}(\Delta\mathbf{x})^T\mathbf{A}\Delta\mathbf{x}. \end{aligned}$$

Therefore,

$$g(\bar{\mathbf{x}} + \Delta\mathbf{x}) - g(\bar{\mathbf{x}}) = \frac{1}{2}(\Delta\mathbf{x})^T\mathbf{A}\Delta\mathbf{x}. \tag{8.10}$$

If $\mathbf{A}$ is positive definite, then $g(\bar{\mathbf{x}} + \Delta\mathbf{x}) - g(\bar{\mathbf{x}}) > 0$ for all vectors $\Delta\mathbf{x} \neq \mathbf{0}$, hence $\bar{x}$ minimizes the function g. Similarly, if $\mathbf{A}$ is negative definite, then it follows from equation (8.10) that g has a maximum at $\bar{x}$. All positive or negative definite matrices are invertible by Theorem 3.9. Hence we proved the following result.

Theorem 8.10. *Let $\mathbf{A}$ be symmetric. Then the gradient vector of the quadratic function $g(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$ is $g'(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. If $\mathbf{A}$ is positive (negative) definite, then g has a global minimum (maximum), which is taken at the point $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.*

The proof of the previous result yields easily:

Corollary 8.11. *If a quadratic function has a local minimum (maximum) at a point, then there the function has also global minimum (maximum).*

If $\mathbf{A}$ is a symmetric positive definite matrix, then Theorem 8.10 yields that the linear system $\mathbf{Ax} = \mathbf{b}$ can be solved that we define the quadratic function g by (8.8), and we minimize it by the optimal gradient method. Therefore, we define the iteration

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \alpha_k \mathbf{v}^{(k)},$$

where

$$\mathbf{v}^{(k)} = g'(\mathbf{p}^{(k)}) = \mathbf{A}\mathbf{p}^{(k)} - \mathbf{b}.$$

α_k is selected so that it be the minimum point of the scalar function $\phi_k(t) := g(\mathbf{p}^{(k)} - t\mathbf{v}^{(k)})$. The function ϕ_k is a quadratic polynomial, since

$$\begin{aligned}\phi_k(t) &= \frac{1}{2}\left(\mathbf{p}^{(k)} - t\mathbf{v}^{(k)}\right)^T \mathbf{A}\left(\mathbf{p}^{(k)} - t\mathbf{v}^{(k)}\right) - \mathbf{b}^T\left(\mathbf{p}^{(k)} - t\mathbf{v}^{(k)}\right) + c \\ &= t^2\frac{1}{2}\left(\mathbf{v}^{(k)}\right)^T \mathbf{A}\mathbf{v}^{(k)} - t\left(\mathbf{v}^{(k)}\right)^T\left(\mathbf{A}\mathbf{p}^{(k)} - \mathbf{b}\right) + c - \mathbf{b}^T\mathbf{p}^{(k)}.\end{aligned}$$

Therefore, its minimum point α_k can be given explicitly as

$$\alpha_k = \frac{\left(\mathbf{v}^{(k)}\right)^T\left(\mathbf{A}\mathbf{p}^{(k)} - \mathbf{b}\right)}{\left(\mathbf{v}^{(k)}\right)^T \mathbf{A}\mathbf{v}^{(k)}}.$$

Introducing the residual vector $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{p}^{(k)}$, the method can be summarized in the following way:

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{p}^{(k)} \tag{8.11}$$

$$\alpha_k = \frac{\left(\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)}}{\left(\mathbf{r}^{(k)}\right)^T \mathbf{A}\mathbf{r}^{(k)}} \tag{8.12}$$

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \alpha_k \mathbf{r}^{(k)}. \tag{8.13}$$

Example 8.12. Consider the linear system

$$\begin{array}{rcrcrcl} 4x_1 & + & 2x_2 & - & x_3 & = & 0 \\ 2x_1 & + & 5x_2 & & & = & 8 \\ -x_1 & & & + & 3x_3 & = & 1. \end{array}$$

We applied the optimal gradient method (8.11)-(8.13) with the initial point $\mathbf{p}^{(0)} = (3,3,3)^T$. Note that the method is applicable since the coefficient matrix of the linear system is symmetric and positive definite. The first 13 terms of the sequence $\mathbf{p}^{(k)}$ are listed in Table 8.4 together with the error of the approximation. Note, the true solution is $(-1,2,0)$. □

Table 8.4: Solving the linear system with gradient method

k	$\mathbf{p}^{(k)}$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_2$
0	(3.00000000, 3.00000000, 3.00000000)	5.09901951
1	(0.43469388, 0.77673469, 2.14489796)	2.85575065
2	(0.03799038, 1.89938726, 0.41611180)	1.12280719
3	(-0.59954375, 1.61568290, 0.37817223)	0.67162421
4	(-0.75093609, 1.98854968, 0.13393796)	0.28302529
5	(-0.90321440, 1.90857051, 0.10622765)	0.17032651
6	(-0.93575911, 1.99605148, 0.03257991)	0.07213829
7	(-0.97504377, 1.97631917, 0.02650106)	0.04342696
8	(-0.98365956, 1.99904876, 0.00839916)	0.01839730
9	(-0.99365117, 1.99398134, 0.00679190)	0.01107528
10	(-0.99583018, 1.99975420, 0.00213698)	0.00469196
11	(-0.99837993, 1.99846385, 0.00173029)	0.00282459
12	(-0.99893668, 1.99993749, 0.00054530)	0.00119662
13	(-0.99958687, 1.99960829, 0.00044139)	0.00072037

Exercises

1. Show that any quadratic function

$$g(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{a}_{ij}x_i x_j + \sum_{i=1}^{n} \tilde{b}_i x_i + c$$

can be written in the form (8.8). How can $g'(\mathbf{x})$ and $g''(\mathbf{x})$ be given using a matrix notation?

2. Prove Corollary 8.11.

3. Check the derivation of formulas (8.11)-(8.13).

4. Apply the gradient method for finding the minimum point of the functions:

(a) $f(x,y) = 2x^2 - 12x + 3y^2 + 30y$, (b) $f(x,y) = 2x^2 - 4xy + 3y^2 - 2y$

5. Solve the following linear systems with gradient method:

(a)
$$\begin{array}{rcrcrcl} 4x_1 & - & 3x_2 & = & 4 \\ -3x_1 & + & 3x_2 & = & 3 \end{array}$$

(b)
$$\begin{array}{rcrcrcl} 6x_1 & + & 3x_2 & - & 2x_3 & = & 6 \\ 3x_1 & + & 5x_2 & - & x_3 & = & -4 \\ -2x_1 & - & x_2 & + & 3x_3 & = & -2 \end{array}$$

6. Let $f(x,y) = \frac{1}{2}x^2 + \frac{9}{2}y^2$. Show that the optimal gradient method started from the initial point $\mathbf{p}^{(0)} = (9,1)^T$ generates the sequence

$$\mathbf{p}^{(k)} = \begin{pmatrix} 9 \\ (-1)^k \end{pmatrix} 0.8^k.$$

What is the asymptotic error constant of this sequence? Give a function and initial value such that the asymptotic error constant of the sequence generated by the optimal gradient method is a predefined constant $0 < \alpha < 1$.

8.6. Newton's Method for Minimization

Consider a function $f\colon \mathbb{R}^n \to \mathbb{R}$, and fix a vector $\mathbf{p}^{(0)}$. If $f \in C^3$, then in a neighbourhood of $\mathbf{p}^{(0)}$ the function f can be approximated by its second-order Taylor polynomial

$$g(\mathbf{x}) := f(\mathbf{p}^{(0)}) + f'(\mathbf{p}^{(0)})^T(\mathbf{x} - \mathbf{p}^{(0)}) + \frac{1}{2}(\mathbf{x} - \mathbf{p}^{(0)})^T f''(\mathbf{p}^{(0)})(\mathbf{x} - \mathbf{p}^{(0)}), \tag{8.14}$$

where $f'(\mathbf{p}^{(0)})$ is the gradient vector of f, and $f''(\mathbf{p}^{(0)})$ is the Hessian matrix of f at $\mathbf{p}^{(0)}$. Suppose that $f''(\mathbf{p}^{(0)})$ is positive definite. Then, by Theorem 8.10, g has a global minimum at the point

$$\mathbf{p}^{(1)} = \mathbf{p}^{(0)} - \left(f''(\mathbf{p}^{(0)})\right)^{-1} f'(\mathbf{p}^{(0)}).$$

Then we consider $\mathbf{p}^{(1)}$ as an approximation of the minimum point of f, and we repeat the previous process from the point $\mathbf{p}^{(1)}$. We can define the iteration:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \left(f''(\mathbf{p}^{(k)})\right)^{-1} f'(\mathbf{p}^{(k)}), \tag{8.15}$$

which is called *Newton's method for minimization*. It is easy to see that it is equivalent to the Newton's method for solving the nonlinear system $f'(\mathbf{x}) = \mathbf{0}$. Therefore, we get the following result immediately.

Theorem 8.13. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$, $f \in C^3$, $f'(\mathbf{p}) = \mathbf{0}$ and $\mathbf{f}''(\mathbf{p})$ be positive definite. Then f has a local minimum at $\mathbf{p}$, and the Newton's iteration (8.15) locally quadratically converges to $\mathbf{p}$.*

Proof. We apply Theorem 8.1 to obtain that f has a local minimum at $\mathbf{p}$. Since iteration (8.15) is equivalent to solving the system $f'(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} = \mathbf{p}$ using Newton's method, Theorem 2.56 yields the local quadratic convergence of iteration (8.15) to $\mathbf{p}$. □

Example 8.14. We apply Newton's method for the function $f(x, y) = (x^2 - 2y)^2 + 2(x - 1)^2$ of Examples 8.6, 8.7. and 8.9. The first 5 terms of the sequence starting from $(-1, 4)^T$ can be seen in Table 8.5. We observe quick convergence to the minimum point $(1, 0.5)^T$. The numerical results indicate that the order of convergence is quadratic. We note that the Newton's iteration starting from $(1, 3)^T$ gives back the exact minimum point in one step. □

Table 8.5: Newton's method, $f(x, y) = (x^2 - 2y)^2 + 2(x - 1)^2$

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)} - \mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)} - \mathbf{p}\|_2^2}$
0	(-1.00000000, 4.00000000)	57.00000000	4.03112887	
1	(-1.33333333, 0.83333333)	10.90123457	2.35702260	0.14504754
2	(0.76666667,-1.91111111)	19.55698889	2.42237512	0.43602752
3	(0.80979667, 0.32695523)	0.07235807	0.25714159	0.04382173
4	(0.99964684, 0.48162536)	0.00129935	0.01837803	0.27794212
5	(0.99998771, 0.49998766)	0.00000000	0.00001742	0.05156519

Example 8.15. Consider the function $f(x,y) = 0.1(x^2-2y)^4 + (x-1)^2$. It is easy to see that the minimum point of this function is also $(1, 0.5)^T$. It can be checked that the Hessian of the function at the minimum point is $f''(1, 0.5) = \mathbf{0}$, so it is not positive definite. Despite of it, the Newton's method converges for this function starting from $(-1, 4)^T$, as it can be seen in Table 8.6. But the convergence in this case is only linear.

□

Table 8.6: Newton's method, $f(x,y) = 0.1(x^2-2y)^4 + (x-1)^2$

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_2}$
0	(-1.00000000, 4.00000000)	244.10000000	4.03112887	
1	(-1.01468429, 2.84801762)	51.47734819	3.09388745	0.76749902
2	(-1.06550085, 2.12183854)	13.60182932	2.62614813	0.84881825
3	(-1.25304590, 1.80360379)	6.79822461	2.60299802	0.99118476
4	(-2.19917836, 2.64963726)	10.23933318	3.85430701	1.48071838
5	(1.13216300,-4.75372475)	1355.09401353	5.25538684	1.36351018
6	(1.13190045,-2.95581491)	267.68684927	3.45833116	0.65805454
7	(1.13102026,-1.75800646)	52.89017856	2.26180447	0.65401616
8	(1.12811546,-0.96208855)	10.46057564	1.46769088	0.64890263
9	(1.11900871,-0.43955842)	2.07752857	0.94706552	0.64527588
10	(1.09458417,-0.11167347)	0.41720946	0.61894313	0.65353781
11	(1.05056809, 0.07705747)	0.08386326	0.42595483	0.68819704
12	(1.01290080, 0.19574848)	0.01637137	0.30452490	0.71492300
13	(1.00119582, 0.28963767)	0.00320655	0.21036572	0.69079974
14	(1.00003517, 0.35899525)	0.00063312	0.14100475	0.67028386
15	(1.00000031, 0.40597370)	0.00012506	0.09402630	0.66683071
16	(1.00000000, 0.43731559)	0.00002470	0.06268441	0.66666888
17	(1.00000000, 0.45821040)	0.00000488	0.04178960	0.66666668
18	(1.00000000, 0.47214026)	0.00000096	0.02785974	0.66666667
19	(1.00000000, 0.48142684)	0.00000019	0.01857316	0.66666667
20	(1.00000000, 0.48761789)	0.00000004	0.01238211	0.66666667

Exercises

1. Apply the Newton's method for minimization for the functions defined in Exercise 1 of Section 8.3.
2. Show that for quadratic functions where the Hessian is positive definite, the Newton's method gives back the minimum point of the function exactly in one step.
3. Show that if the conditions of Theorem 8.13 hold and $\mathbf{p}^{(0)}$ is close enough to $\mathbf{p}$, then the sequence (8.15) is defined for all k, i.e., $f''(\mathbf{p}^{(k)})$ is invertible.

8.7. Quasi-Newton Method for Minimization

Similarly to the previous section, we approximate the function $f\colon \mathbb{R}^n \to \mathbb{R}$ in a neighbourhood of $\mathbf{p}^{(k)}$ by the quadratic function

$$g(\mathbf{x}) := f(\mathbf{p}^{(k)}) + \left(\mathbf{v}^{(k)}\right)^T (\mathbf{x} - \mathbf{p}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{p}^{(k)})^T \mathbf{A}^{(k)} (\mathbf{x} - \mathbf{p}^{(k)}). \tag{8.16}$$

If $\mathbf{v}^{(k)} \approx f'(\mathbf{p}^{(k)})$ and $\mathbf{A}^{(k)} \approx f''(\mathbf{p}^{(k)})$, then (8.16) approximates the second-order Taylor polynomial of f around $\mathbf{p}^{(k)}$, so it can be considered as an approximation of f in a small neighbourhood of $\mathbf{p}^{(k)}$. We hope that the minimum point of g will approximate that of f. If $\mathbf{A}^{(k)}$ is positive definite, then Theorem 8.10 yields that the minimum point of g is

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \left(\mathbf{A}^{(k)}\right)^{-1} \mathbf{v}^{(k)}. \tag{8.17}$$

Such iterations are called *quasi-Newton methods for minimization*.

We can define $\mathbf{A}^{(k)}$ and $\mathbf{v}^{(k)}$ as a numerical approximation of the Hessian matrix $f''(\mathbf{p}^{(k)})$ and the gradient vector $f'(\mathbf{p}^{(k)})$: $\mathbf{A}^{(k)} = (a_{ij}^{(k)})$ and $\mathbf{v}^{(k)} = (v_1^{(k)}, \dots, v_n^{(k)})^T$, where

$$a_{ij}^{(k)} = \frac{1}{h^2}\left(f(\mathbf{p}^{(k)} + h\mathbf{e}^{(i)} + h\mathbf{e}^{(j)}) - f(\mathbf{p}^{(k)} + h\mathbf{e}^{(i)}) - f(\mathbf{p}^{(k)} + h\mathbf{e}^{(j)}) + f(\mathbf{p}^{(k)})\right) \tag{8.18}$$

and

$$v_i^{(k)} = \frac{1}{h}\Big(f(\mathbf{p}^{(k)} + h\mathbf{e}^{(i)}) - f(\mathbf{p}^{(k)})\Big),$$

$i, j = 1, \dots, n$ ($\mathbf{e}^{(i)}$ is the ith unit vector, $h > 0$ is fixed small step size). Here we used the first-order forward difference formula to approximate the first partial derivatives of f, and formulas (7.19)–(7.20) to approximate the second partial derivatives. This way we do not need to now the exact values of the gradient vector and the Hessian matrix, but in each step of the iteration we need to perform n^2 number of function evaluations.

Next we consider the case when in (8.17) we have the exact gradient value $\mathbf{v}^{(k)} = f'(\mathbf{p}^{(k)})$, and hence we examine quasi-Newton methods of the form

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \left(\mathbf{A}^{(k)}\right)^{-1} f'(\mathbf{p}^{(k)}). \tag{8.19}$$

Here we assume that we can evaluate the gradient vector of the function, so the question is only how to approximate the Hessian matrix. One possibility is to use Broyden's method defined in Section 2.13 to approximate solutions of the system $f'(\mathbf{x}) = \mathbf{0}$:

$$\mathbf{A}^{(k)}\mathbf{s}^{(k)} = -f'(\mathbf{p}^{(k)}), \tag{8.20}$$

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \mathbf{s}^{(k)}, \tag{8.21}$$

$$\mathbf{y}^{(k)} = f'(\mathbf{p}^{(k+1)}) - f'(\mathbf{p}^{(k)}), \tag{8.22}$$

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2}. \tag{8.23}$$

Example 8.16. We apply Broyden's method defined by (8.20)–(8.23) for minimizing the function $f(x, y) = (x^2 - 2y)^2 + 2(x - 1)^2$. We start the sequence from the initial point $(2, 2)^T$, and the matrix $\mathbf{A}^{(0)}$ is defined as a second-order difference approximation (8.18) of the Hessian matrix $f''(2, 2)$ using step size $h = 0.05$. The first 10 elements of the sequence can be seen in Table 8.7. □

The problem with the iteration (8.23) is that since $\mathbf{A}^{(k)}$ is an approximation of the Hessian $f''(\mathbf{p})$, it is natural to require that $\mathbf{A}^{(k)}$ be positive definite for all k. It is also needed to argue that the quadratic function (8.16) has a minimum for all k. The numerical

Table 8.7: Broyden's method for minimization, $f(x,y) = (x^2 - 2y)^2 + 2(x-1)^2$

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_2}$
0	(2.00000000, 2.00000000)	2.00000e+00	1.80277564	
1	(1.28952043, 0.56127886)	4.59574e-01	0.29593441	0.16415488
2	(1.35039835, 0.89916410)	2.46195e-01	0.53114121	1.79479368
3	(1.24875073, 0.73204681)	1.32833e-01	0.34018032	0.64047058
4	(1.12570322, 0.59780553)	3.67287e-02	0.15927091	0.46819553
5	(1.05911935, 0.54518730)	7.97359e-03	0.07441095	0.46719737
6	(0.99939685, 0.49649610)	3.43894e-05	0.00355544	0.04778109
7	(1.01133354, 0.50962433)	2.69479e-04	0.01486866	4.18194987
8	(1.00464762, 0.50384065)	4.58758e-05	0.00602918	0.40549562
9	(1.00047293, 0.50036811)	4.91375e-07	0.00059931	0.09940111
10	(1.00008014, 0.50006497)	1.37638e-08	0.00010316	0.17213595

experience also gives that those quasi-Newton methods of the form (8.19) are the most efficient where $\mathbf{A}^{(k)}$ is a positive definite approximation of the Hessian. But the matrix sequence $\mathbf{A}^{(k)}$ generated by the Broyden's method is not even symmetric.

Our first goal is to modify the Broyden's method so that it should generate a symmetric matrix for all k. Suppose $\mathbf{A}^{(k)}$ is symmetric, and let

$$\mathbf{B}^{(k+1,1)} = \mathbf{A}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2}$$

be the matrix computed by the Broyden iteration. It can be shown (see Exercise 2) that the closest symmetric matrix to $\mathbf{A}$ (in some sense) is the matrix $\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$. Therefore, it is natural to modify $\mathbf{B}^{(k+1,1)}$ in the following way

$$\begin{aligned}\mathbf{B}^{(k+1,2)} &= \frac{1}{2}\left(\mathbf{B}^{(k+1,1)} + \mathbf{B}^{(k+1,1)^T}\right) \\ &= \mathbf{A}^{(k)} + \frac{1}{2}\frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T + \mathbf{s}^{(k)}(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2}.\end{aligned} \tag{8.24}$$

But now the problem is that the matrix $\mathbf{B}^{(k+1,2)}$ does not satisfy the secant equation $\mathbf{A}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$ which was the motivation of the Broyden's method. We correct it by applying relation (8.23) again: let

$$\mathbf{B}^{(k+1,3)} = \mathbf{B}^{(k+1,2)} + \frac{(\mathbf{y}^{(k)} - \mathbf{B}^{(k+1,2)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2}. \tag{8.25}$$

This is again a non-symmetric matrix, so we repeat the above procedure again: define the matrices $\mathbf{B}^{(k+1,2i)}$ and $\mathbf{B}^{(k+1,2i+1)}$ from the previous term of the sequence using formulas (8.24) and (8.25), respectively, for $i = 2, 3, \ldots$. It can be shown that the matrix sequence $\mathbf{B}^{(k+1,i)}$ converges to the symmetric matrix

$$\begin{aligned}\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} &+ \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{s}^{(k)})^T + \mathbf{s}^{(k)}(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})^T}{\|\mathbf{s}^{(k)}\|_2^2} \\ &- \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})^T\mathbf{s}^{(k)}}{\|\mathbf{s}^{(k)}\|_2^4}\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T.\end{aligned} \tag{8.26}$$

This is a correction iteration which preserves the symmetric property of the matrix, and also it satisfies the secant equation $\mathbf{A}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$. This iteration is called *Powell-symmetric-Broyden update*, or shortly, *PSB update*. The following result can be shown:

Theorem 8.17. *Let $f \in C^3$, $f'(\mathbf{p}) = 0$, $f''(\mathbf{p})$ be positive definite. Then there exist $\varepsilon, \delta > 0$ such that the iteration (8.20)–(8.22), (8.26) is defined for all k, and it converges superlinearly to $\mathbf{p}$ if $\|\mathbf{p}^{(0)} - \mathbf{p}\|_2 < \varepsilon$ and $\|\mathbf{A}^{(0)} - f''(\mathbf{p})\|_2 < \delta$.*

Example 8.18. Here we apply the quasi-Newton method (8.19) with the PSB update for the function $f(x, y) = (x^2-2y)^2+2(x-1)^2$. We started the computation from the same initial value that was used in Example 8.16. The corresponding numerical values can be seen in Table 8.8. The approximation here is better than that of for the Broyden's method. □

Table 8.8: Quasi-Newton method (8.19) with the PSB update

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)} - \mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_2}$
0	(2.00000000, 2.00000000)	2.00000e+00	1.80277564	
1	(1.28952043, 0.56127886)	4.59574e-01	0.29593441	0.16415488
2	(1.25102079, 0.70409379)	1.50630e-01	0.32352080	1.09321792
3	(1.19910219, 0.73444653)	8.02473e-02	0.30758228	0.95073416
4	(1.14966546, 0.69907469)	5.06393e-02	0.24905919	0.80973192
5	(1.00399514, 0.50473229)	3.40491e-05	0.00619320	0.02486638
6	(0.99975498, 0.49938607)	6.64526e-07	0.00066102	0.10673251
7	(1.00003118, 0.49997474)	1.46839e-08	0.00004012	0.06070113
8	(1.00001593, 0.50000889)	7.05953e-10	0.00001824	0.45466117
9	(1.00000627, 0.50000724)	8.24492e-11	0.00000958	0.52515860
10	(1.00000015, 0.50000024)	7.49020e-14	0.00000028	0.02901243

The PSB update does not satisfy the goal formulated earlier that $\mathbf{A}^{(k)}$ be positive definite for all k if $\mathbf{A}^{(0)}$ is positive definite. According to Theorem 5.6, if a matrix $\mathbf{A}$ is positive definite, then it has a Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is non-singular. Otherwise, if a matrix $\mathbf{A}$ has the form $\mathbf{A} = \mathbf{M}\mathbf{M}^T$ where $\mathbf{M}$ is non-singular, then $\mathbf{A}$ is positive definite, since $\mathbf{x}^T\mathbf{M}\mathbf{M}^T\mathbf{x} = \|\mathbf{M}^T\mathbf{x}\|_2^2 \geq 0$, and here equality holds if and only if $\mathbf{M}^T\mathbf{x} = \mathbf{0}$, and hence $\mathbf{x} = \mathbf{0}$.

Let $\mathbf{A}^{(k)} = \mathbf{M}^{(k)}(\mathbf{M}^{(k)})^T$ where $\mathbf{M}^{(k)}$ is invertible (but not necessary lower triangular). We look for the next Hessian approximation $\mathbf{A}^{(k+1)}$ in the form $\mathbf{A}^{(k+1)} = \mathbf{M}^{(k+1)}(\mathbf{M}^{(k+1)})^T$ where we require that $\mathbf{A}^{(k+1)}$ satisfies the secant equation $\mathbf{A}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$. Then it implies $(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)} = (\mathbf{s}^{(k)})^T\mathbf{A}^{(k+1)}\mathbf{s}^{(k)}$, hence if $\mathbf{A}^{(k+1)}$ is positive definite, then the inequality

$$(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)} > 0 \tag{8.27}$$

holds. We show that the secant equation has a positive definite solution assuming (8.27) holds.

We introduce the notation $\mathbf{v}^{(k)} := (\mathbf{M}^{(k+1)})^T\mathbf{s}^{(k)}$. Then the secant equation has the form

$$(\mathbf{M}^{(k+1)})^T\mathbf{s}^{(k)} = \mathbf{v}^{(k)}, \tag{8.28}$$

$$\mathbf{M}^{(k+1)}\mathbf{v}^{(k)} = \mathbf{y}^{(k)}. \tag{8.29}$$

We would like to compute the matrix $\mathbf{M}^{(k+1)}$ by updating the matrix $\mathbf{M}^{(k)}$. Therefore, using the derivation of the Broyden's method and using (8.29), it is natural to look for the matrix $\mathbf{M}^{(k+1)}$ in the form

$$\mathbf{M}^{(k+1)} = \mathbf{M}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{M}^{(k)}\mathbf{v}^{(k)})(\mathbf{v}^{(k)})^T}{\|\mathbf{v}^{(k)}\|_2^2}. \tag{8.30}$$

Then $\mathbf{M}^{(k+1)}$ satisfies equation (8.29), and its difference from the matrix $\mathbf{M}^{(k)}$ is the smallest in the sense that for all $\mathbf{z} \perp \mathbf{v}^{(k)}$ it follows $\mathbf{M}^{(k+1)}\mathbf{z} = \mathbf{M}^{(k)}\mathbf{z}$. Substituting $\mathbf{M}^{(k+1)}$ back to equation (8.28) we get

$$\begin{aligned}
\mathbf{v}^{(k)} &= (\mathbf{M}^{(k)})^T\mathbf{s}^{(k)} + \frac{\left((\mathbf{y}^{(k)} - \mathbf{M}^{(k)}\mathbf{v}^{(k)})(\mathbf{v}^{(k)})^T\right)^T}{\|\mathbf{v}^{(k)}\|_2^2}\mathbf{s}^{(k)} \\
&= (\mathbf{M}^{(k)})^T\mathbf{s}^{(k)} + \frac{\mathbf{v}^{(k)}(\mathbf{y}^{(k)} - \mathbf{M}^{(k)}\mathbf{v}^{(k)})^T}{\|\mathbf{v}^{(k)}\|_2^2}\mathbf{s}^{(k)} \\
&= (\mathbf{M}^{(k)})^T\mathbf{s}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{M}^{(k)}\mathbf{v}^{(k)})^T\mathbf{s}^{(k)}}{\|\mathbf{v}^{(k)}\|_2^2}\mathbf{v}^{(k)}.
\end{aligned}$$

It yields $(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)} = \alpha\mathbf{v}^{(k)}$, where

$$\begin{aligned}
\alpha &= 1 - \frac{(\mathbf{y}^{(k)} - \mathbf{M}^{(k)}\mathbf{v}^{(k)})^T\mathbf{s}^{(k)}}{\|\mathbf{v}^{(k)}\|_2^2} \\
&= 1 - \frac{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}{\|\mathbf{v}^{(k)}\|_2^2} + \frac{(\mathbf{v}^{(k)})^T(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)}}{\|\mathbf{v}^{(k)}\|_2^2} \\
&= 1 - \alpha^2\frac{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{M}^{(k)}(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)}} + \alpha,
\end{aligned}$$

and so

$$\alpha^2 = \frac{(\mathbf{s}^{(k)})^T\mathbf{M}^{(k)}(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)}}{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}} = \frac{(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}\mathbf{s}^{(k)}}{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}. \tag{8.31}$$

We have that the numerator is positive since $\mathbf{A}^{(k)}$ is positive definite, therefore, α can be obtained from equation (8.31), and

$$\mathbf{v}^{(k)} = \frac{1}{\alpha}(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)} = \left(\frac{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}\mathbf{s}^{(k)}}\right)^{1/2}(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)}.$$

Substituting it back to equation (8.30) we get

$$\begin{aligned}
\mathbf{M}^{(k+1)} &= \mathbf{M}^{(k)} + \frac{(\mathbf{y}^{(k)} - \frac{1}{\alpha}\mathbf{M}^{(k)}(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)})\frac{1}{\alpha}(\mathbf{s}^{(k)})^T\mathbf{M}^{(k)}}{\frac{1}{\alpha^2}\|(\mathbf{M}^{(k)})^T\mathbf{s}^{(k)}\|_2^2} \\
&= \mathbf{M}^{(k)} + \alpha\frac{\mathbf{y}^{(k)}(\mathbf{s}^{(k)})^T\mathbf{M}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}\mathbf{s}^{(k)}} - \frac{\mathbf{A}^{(k)}\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T\mathbf{M}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}\mathbf{s}^{(k)}}.
\end{aligned}$$

Little computation gives (see Exercise 4) that

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \frac{\mathbf{y}^{(k)}(\mathbf{y}^{(k)})^T}{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}} - \frac{\mathbf{A}^{(k)}\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{A}^{(k)}\mathbf{s}^{(k)}}. \tag{8.32}$$

We have to show that the iteration generates a positive definite matrix. Since $\mathbf{A}^{(k+1)} = \mathbf{M}^{(k+1)}(\mathbf{M}^{(k+1)})^T$, it is enough to show that $\mathbf{M}^{(k+1)}$ is invertible. By our assumption, the matrix $\mathbf{M}^{(k)}$ is positive definite, and hence it is invertible. If we assume that (8.27) holds, then the invertibility of $\mathbf{M}^{(k+1)}$ follows easily from (8.30) and Theorem 2.58. The details are left to the reader (Exercise 5).

The formula (8.32) was introduced by Broyden, Flecher, Goldfarb and Shanno in 1970, therefore, it is called *BFGS update*. This is the best known iteration for the approximation of the Hessian. The initial value of the iteration can be the matrix $f''(\mathbf{p}^{(0)})$ or its numerical approximation by the second-order difference formula (8.18). If $\mathbf{p}^{(0)}$ is close enough to $\mathbf{p}$ and $f''(\mathbf{p})$ is positive definite, then $f''(\mathbf{p}^{(0)})$ and so $\mathbf{A}^{(0)}$ is also positive definite.

Finally, consider condition (8.27). Applying Lagrange's Mean Value Theorem (Theorem 2.40), relations (8.21) and (8.22), we get

$$\begin{aligned}(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)} &= \left(f'(\mathbf{p}^{(k+1)}) - f'(\mathbf{p}^{(k)})\right)^T(\mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}) \\ &= \sum_{i=1}^n \left(\frac{\partial f_i(\mathbf{p}^{(k+1)})}{\partial x_i} - \frac{\partial f_i(\mathbf{p}^{(k)})}{\partial x_i}\right)(p_i^{(k+1)} - p_i^{(k)}) \\ &= \sum_{i=1}^n \left(\sum_{j=1}^n \frac{\partial^2 f_i(\xi^{(k,i)})}{\partial x_i\,\partial x_j}(p_j^{(k+1)} - p_j^{(k)})\right)(p_i^{(k+1)} - p_i^{(k)}).\end{aligned}$$

If the iterates $\mathbf{p}^{(k)}$ are close enough to $\mathbf{p}$ during the iteration, then the vectors $\xi^{(k,i)}$ are also close to $\mathbf{p}$, and hence the continuity of f'' yields

$$\begin{aligned}(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)} &\approx \sum_{i=1}^n \left(\sum_{j=1}^n \frac{\partial^2 f_i(\mathbf{p})}{\partial x_i\,\partial x_j}(p_j^{(k+1)} - p_j^{(k)})\right)(p_i^{(k+1)} - p_i^{(k)}) \\ &= (\mathbf{p}^{(k+1)} - \mathbf{p}^{(k)})^T f''(\mathbf{p})(\mathbf{p}^{(k+1)} - \mathbf{p}^{(k)}),\end{aligned}$$

which is positive, since $f''(\mathbf{p})$ is positive definite. Therefore, this condition is automatically satisfied for large k if the sequence converges to $\mathbf{p}$. Clearly, if (8.27) does not hold, then iteration (8.32) can be defined, but in this case $\mathbf{A}^{(k+1)}$ is only positive semidefinite, not positive definite.

The following result can be proved.

Theorem 8.19. *Let $f \in C^3$, $f'(\mathbf{p}) = \mathbf{0}$, and $f''(\mathbf{p})$ be positive definite. Then there exist $\varepsilon, \delta > 0$ such that the iteration (8.20)–(8.22), (8.32) is defined for all k, and it converges superlinearly to $\mathbf{p}$, assuming $\|\mathbf{p}^{(0)} - \mathbf{p}\|_2 < \varepsilon$ and $\|\mathbf{A}^{(0)} - f''(\mathbf{p})\|_2 < \delta$.*

Example 8.20. We applied the quasi-Newton method (8.19) with the BFGS update for the function $f(x, y) = (x^2 - 2y)^2 + 2(x-1)^2$. We used the same initial condition as in Example 8.16. The numerical results are listed in Table 8.9. We have got a very precise approximation in 8 steps. □

Table 8.9: Quasi-Newton method (8.19) with the BFGS update

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)}-\mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_2}$
0	(2.00000000, 2.00000000)	2.00000e+00	1.80277564	
1	(1.28952043, 0.56127886)	4.59574e-01	0.29593441	0.16415488
2	(1.23976784, 0.70438005)	1.31429e-01	0.31505527	1.06461181
3	(1.02721672, 0.49403232)	5.98519e-03	0.02786330	0.08843939
4	(1.00995636, 0.51197836)	2.13820e-04	0.01557595	0.55901316
5	(0.99954439, 0.49921815)	8.41172e-07	0.00090492	0.05809714
6	(1.00000534, 0.50000495)	5.76547e-11	0.00000728	0.00804964
7	(1.00000005, 0.50000002)	9.15800e-15	0.00000005	0.00708494
8	(1.00000000, 0.50000000)	8.60000e-19	0.00000000	0.01827989

It can be proved by mathematical induction that the inverses $\mathbf{B}^{(k)} := (\mathbf{A}^{(k)})^{-1}$ of the matrices $\mathbf{A}^{(k)}$ generated by the BFGS update satisfy the recursion

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \left(1 + \frac{(\mathbf{y}^{(k)})^T\mathbf{B}^{(k)}\mathbf{y}^{(k)}}{(\mathbf{s}^{(k)})^T\mathbf{y}^{(k)}}\right)\frac{\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T}{(\mathbf{s}^{(k)})^T\mathbf{y}^{(k)}} - \frac{\mathbf{s}^{(k)}(\mathbf{y}^{(k)})^T\mathbf{B}^{(k)} + \mathbf{B}^{(k)}\mathbf{y}^{(k)}(\mathbf{s}^{(k)})^T}{(\mathbf{s}^{(k)})^T\mathbf{y}^{(k)}}. \tag{8.33}$$

Using this formula, (8.20) can be replaced by

$$\mathbf{s}^{(k)} = -\mathbf{B}^{(k)}f'(\mathbf{p}^{(k)}), \tag{8.34}$$

so during the iteration we do not need to compute matrix inverses or solving linear systems.

Similarly to the derivation of the BFGS update, we can obtain the definition of the *DFP update*. Again, we are looking for the approximation of the Hessian in the form $\mathbf{A}^{(k+1)} = \mathbf{M}^{(k+1)}(\mathbf{M}^{(k+1)})^T$, but instead of the iterates (8.28)–(8.29) we use the equivalent iteration

$$(\mathbf{M}^{(k+1)})^{-1}\mathbf{y}^{(k)} = \mathbf{v}^{(k)}$$
$$\left(\mathbf{M}^{(k+1)^T}\right)^{-1}\mathbf{v}^{(k)} = \mathbf{s}^{(k)}.$$

Its solution is considered in the form

$$\left(\mathbf{M}^{(k+1)}\right)^{-1} = \left(\mathbf{M}^{(k)}\right)^{-1} + \frac{\left(\mathbf{s}^{(k)} - (\mathbf{M}^{(k)})^{-1}\mathbf{v}^{(k)}\right)(\mathbf{v}^{(k)})^T}{\|\mathbf{v}^{(k)}\|_2^2}.$$

Then we get

$$\mathbf{v}^{(k)} = \left(\frac{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}{(\mathbf{y}^{(k)})^T(\mathbf{A}^{(k)})^{-1}\mathbf{y}^{(k)}}\right)^{1/2}(\mathbf{M}^{(k)})^{-1}\mathbf{y}^{(k)},$$

assuming (8.27) holds. From this and Theorem 2.58 we get

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})(\mathbf{y}^{(k)})^T + \mathbf{y}^{(k)}(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})^T}{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}} - \frac{(\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{s}^{(k)})^T\mathbf{s}^{(k)}}{((\mathbf{y}^{(k)})^T\mathbf{s}^{(k)})^2}\mathbf{y}^{(k)}(\mathbf{y}^{(k)})^T. \tag{8.35}$$

This formula is called the *DFP update*, since it was established by Davidon (1959) and Flecher, Powell (1963). This iteration satisfies a result analogous to Theorem 8.19.

It can be checked that the inverse of the matrix $\mathbf{A}^{(k)}$ generated by the DFP update can be computed by the recursion:

$$(\mathbf{A}^{(k+1)})^{-1} = (\mathbf{A}^{(k)})^{-1} + \frac{\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T}{(\mathbf{s}^{(k)})^T\mathbf{y}^{(k)}} - \frac{(\mathbf{A}^{(k)})^{-1}\mathbf{y}^{(k)}(\mathbf{y}^{(k)})^T(\mathbf{A}^{(k)})^{-1}}{(\mathbf{y}^{(k)})^T(\mathbf{A}^{(k)})^{-1}\mathbf{y}^{(k)}}. \tag{8.36}$$

Example 8.21. Here we used the DFP update in the problem investigated in Examples 8.16 and 8.20. This method converges with a speed similar to the BFGS update. The numerical results can be seen in Table 8.10. □

Table 8.10: Quasi-Newton method (8.19) with DFP update

k	$\mathbf{p}^{(k)}$	$f(\mathbf{p}^{(k)})$	$\|\mathbf{p}^{(k)}-\mathbf{p}\|_2$	$\frac{\|\mathbf{p}^{(k)}-\mathbf{p}\|_2}{\|\mathbf{p}^{(k-1)}-\mathbf{p}\|_2}$
0	(2.00000000, 2.00000000)	2.00000e+00	1.80277564	
1	(1.28952043, 0.56127886)	4.59574e-01	0.29593441	0.16415488
2	(1.25682024, 0.70394625)	1.61396e-01	0.32794924	1.10818219
3	(1.09891338, 0.59229507)	2.00977e-02	0.13528576	0.41252041
4	(1.01148073, 0.50204318)	6.24877e-04	0.01166112	0.08619621
5	(1.00103666, 0.50022718)	4.77384e-06	0.00106126	0.09100838
6	(1.00001771, 0.50001111)	8.01068e-10	0.00002090	0.01969409
7	(0.99999976, 0.49999958)	2.45621e-13	0.00000049	0.02332123
8	(1.00000001, 0.50000002)	4.22000e-16	0.00000002	0.03601757

Exercises

1. Apply the quasi-Newton methods introduced in this section to the problems of Exercise 1 of Section 8.3.
2. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$. Define
$$\|\mathbf{A}\|_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2},$$
which is the so-called *Frobenius norm* of the matrix $\mathbf{A}$. (This is not a matrix norm generated by a vector norm.) Prove that the unique solution of the minimization problem
$$\min\{\|\mathbf{B}-\mathbf{A}\|_F\colon\ \mathbf{B}\in\mathbb{R}^{n\times n},\ \mathbf{B}\text{ symmetric}\}$$
is the matrix $\mathbf{B} = \frac{1}{2}(\mathbf{A}+\mathbf{A}^T)$.
3. Show that the matrix defined by (8.26) is symmetric and it satisfies the secant equation $\mathbf{A}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$.
4. Check the derivation of formula (8.32).
5. Prove that the matrix $\mathbf{M}^{(k+1)}$ is invertible if relation (8.27) holds.
6. Show recursion (8.33).
7. Work out the details for the derivation of the DFP update.
8. Prove recursion (8.36).

Chapter 9
Method of Least Squares

Suppose that a physical process can be described by a real function g, where we know or assume the formula of the function but we do not know the values of some parameters in the formula. We put the parameters into a vector $\mathbf{a}$, and the notation $g(x;\mathbf{a})$ will emphasize the dependence of g on the parameters $\mathbf{a}$. Suppose we have measurements y_i $(i = 0, 1, \ldots, n)$ of the function values at the mesh points x_i. For example, we know or assume that g is a quadratic polynomial, then g is determined by its three coefficients. If we have more than 3 measurements, then, in general, there is no unique parabola whose graph goes through all the measurement points, since due to measurement error, the data points are typically not located on the graph of g. Therefore, our goal is to find the parameter values for which the corresponding function g differs from the measurements with the "smallest error". This problem is called *curve fitting*. It is not obvious how to measure the error of the curve fitting. Depending on its definition, we get different mathematical problems.

It is possible to measure the error of the curve fitting using the formulas

$$F_1(\mathbf{a}) := \max\{|g(x_i;\mathbf{a}) - y_i| :\ i = 0, 1, \ldots, n\}$$

or

$$F_2(\mathbf{a}) := \sum_{i=0}^{n} |g(x_i;\mathbf{a}) - y_i|.$$

Both looks natural, since if these errors are small, then the difference between $g(x_i)$ and the measurements y_i will be small at every singular point. The problem is that if we wanted to minimize $F_1(\mathbf{a})$ or $F_2(\mathbf{a})$ with respect to $\mathbf{a}$, then it is difficult to compute, since none of the above functions are differentiable. This technicality can be eliminated if we consider the error formula

$$F(\mathbf{a}) := \sum_{i=0}^{n} (g(x_i;\mathbf{a}) - y_i)^2,$$

the so-called *least square error*. Here the mathematical problem is to minimize $F(\mathbf{a})$, and consider the graph of the function $g(x;\bar{\mathbf{a}})$ corresponding to the minimum point $\bar{\mathbf{a}}$ of $\mathbf{F}(\mathbf{a})$ as the best fitted curve to the data points. This is called the *method of least squares*.

In this chapter we investigate some basic cases of the method of least squares. We study the curve fitting first for lines, and next for polynomial functions. Finally, we consider this method for some other special nonlinear functions using the method of linearization.

9.1. Line Fitting

Given data points (x_i, y_i), $i = 0, 1, \ldots, n$, where at least some of the mesh points x_i are different. We are looking for a linear function of the form $g(x) = ax + b$ which minimizes the least square error

$$F(a, b) := \sum_{i=0}^{n} (ax_i + b - y_i)^2. \tag{9.1}$$

The function F is continuously partially differentiable with respect to a and b, and

$$\begin{aligned} \frac{\partial F}{\partial a}(a, b) &= 2 \sum_{i=0}^{n} (ax_i + b - y_i) x_i, \\ \frac{\partial F}{\partial b}(a, b) &= 2 \sum_{i=0}^{n} (ax_i + b - y_i). \end{aligned} \tag{9.2}$$

Making the partial derivatives in (9.2) equal to 0, and rearranging the system we get the so-called *Gaussian normal equations*:

$$\begin{aligned} a \sum_{i=0}^{n} x_i^2 + b \sum_{i=0}^{n} x_i &= \sum_{i=0}^{n} x_i y_i, \\ a \sum_{i=0}^{n} x_i + b(n+1) &= \sum_{i=0}^{n} y_i. \end{aligned} \tag{9.3}$$

It is worth to mention that the coefficient of b in the second equation is $n + 1$, which is the number of data points. This is a linear system for solving a and b. This system is solvable if the determinant of its coefficient matrix

$$d := \det \begin{pmatrix} \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & n+1 \end{pmatrix} = (n+1) \sum_{i=0}^{n} x_i^2 - \left(\sum_{i=0}^{n} x_i \right)^2$$

is nonzero. The Cauchy–Bunyakovsky–Schwarz inequality (Theorem 2.42) yields

$$\left(\sum_{i=0}^{n} x_i \right)^2 = \left(\sum_{i=0}^{n} 1 \cdot x_i \right)^2 \leq \sum_{i=0}^{n} 1 \sum_{i=0}^{n} x_i^2 = (n+1) \sum_{i=0}^{n} x_i^2,$$

therefore, $d \geq 0$ holds. If we assume that there are at least two distinct mesh points x_i, then Theorem 2.42 implies that the strict inequality $d > 0$ holds. Hence system (9.3) has a unique solution which can be given in the following form:

$$\begin{aligned} \bar{a} &= \frac{(n+1)\left(\sum_{i=0}^{n} x_i y_i\right) - \left(\sum_{i=0}^{n} x_i\right)\left(\sum_{i=0}^{n} y_i\right)}{(n+1)\left(\sum_{i=0}^{n} x_i^2\right) - \left(\sum_{i=0}^{n} x_i\right)^2}, \\ \bar{b} &= \frac{\left(\sum_{i=0}^{n} x_i^2\right)\left(\sum_{i=0}^{n} y_i\right) - \left(\sum_{i=0}^{n} x_i y_i\right)\left(\sum_{i=0}^{n} x_i\right)}{(n+1)\left(\sum_{i=0}^{n} x_i^2\right) - \left(\sum_{i=0}^{n} x_i\right)^2}. \end{aligned}$$

According to Theorem 8.2, the function F has a local extremum at $(\bar{a}, \bar{b})$ if

$$D(\bar{a}, \bar{b}) := \frac{\partial^2 F}{\partial a^2}(\bar{a}, \bar{b}) \cdot \frac{\partial^2 F}{\partial b^2}(\bar{a}, \bar{b}) - \left(\frac{\partial^2 F}{\partial a \, \partial b}(\bar{a}, \bar{b}) \right)^2 > 0.$$

It is easy to compute that

$$\frac{\partial^2 F}{\partial a^2}(\bar{a}, \bar{b}) = 2 \sum_{i=0}^{n} x_i^2, \quad \frac{\partial^2 F}{\partial b^2}(\bar{a}, \bar{b}) = 2(n+1), \quad \frac{\partial^2 F}{\partial a \, \partial b}(\bar{a}, \bar{b}) = 2 \sum_{i=0}^{n} x_i.$$

Hence

$$D(\bar{a}, \bar{b}) = 4(n+1) \sum_{i=0}^{n} x_i^2 - 4 \left(\sum_{i=0}^{n} x_i \right)^2 = 4d,$$

which we know that it is positive. Since $\frac{\partial^2 F}{\partial a^2}(\bar{a}, \bar{b}) > 0$, Theorem 8.2 yields that F has a local minimum at $(\bar{a}, \bar{b})$, and hence Corollary 8.11 implies that it is also a global minimum. We have proved the following result.

Theorem 9.1. *Given data points (x_i, y_i) $(i = 0, 1, \ldots, n)$ such that there exist i and j with $x_i \neq x_j$. Then the problem*

$$\min_{(a,b) \in \mathbb{R}^2} \sum_{i=0}^{n} (a x_i + b - y_i)^2$$

has a unique solution, which satisfies the Gaussian normal equations (9.3).

Example 9.2. Given the following data:

x_i	-1.0	1.0	2.5	3.0	4.0	4.5	6.0
y_i	0.0	1.2	1.9	2.5	3.1	3.2	4.5

Find a line of best fit to the data points. In case we do the calculation by hand, we copy the data to the first two columns of the Table 9.1. Then we fill out the third and fourth columns of the table, and finally, in the last line, we compute the sum of the numbers located above in that column. This last line is used to write down the Gaussian normal equations (9.3):

$$\begin{array}{rcrcl} 67.25a & + & 20.0b & = & 67.25 \\ 20.0a & + & 7b & = & 16.4. \end{array}$$

Its solution is $a = 0.630243$ and $b = 0.542163$. The graph of the corresponding line $y = 0.630243x + 0.542163$ and the given data points can be seen in Figure 9.1. The error of the fitting is

$$\sum_{i=0}^{6} (0.630243 x_i + 0.542163 - y_i)^2 = 0.124691.$$

□

Table 9.1: Line fitting

x_i	y_i	x_i^2	x_iy_i
-1.0	0.0	1.00	0.00
1.0	1.2	1.00	1.20
2.5	1.9	6.25	4.75
3.0	2.5	9.00	7.50
4.0	3.1	16.00	12.40
4.5	3.2	20.25	14.40
6.0	4.5	36.00	27.00
20.0	16.4	89.50	67.25

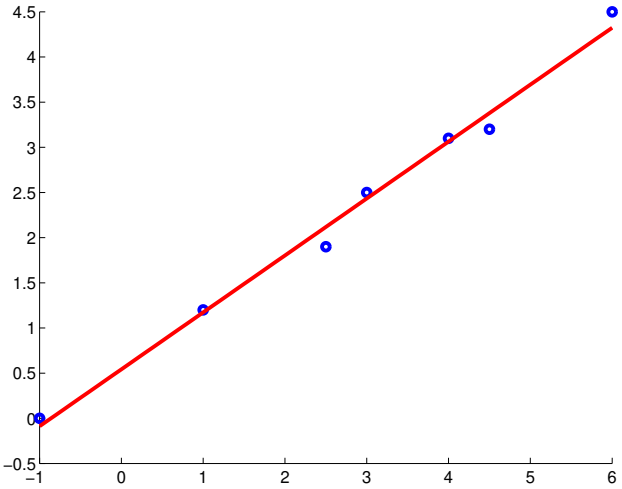

Figure 9.1: Line fitting: $y = 0.630243x + 0.542163$

Exercises

1. Find the line of best fit to the following data, and compute the error of the fitting:

(a)

x_i	0.0	1.0	1.5	2.0	3.0
y_i	-1.8	1.3	2.5	3.9	8.3

(b)

x_i	-1.0	1.0	2.0	3.0	4.0	5.0	6.0
y_i	4.2	2.1	1.3	2.1	2.8	-2.1	-3.0

(c)

x_i	-1.0	1.0	3.0	5.0	9.0	10.0	13.0
y_i	-0.1	3.4	7.3	15.1	29.1	35.6	56.3

9.2. Polynomial Curve Fitting

In this section we study the problem of polynomial curve fitting. Given data points (x_i, y_i) $(i = 0, 1, \ldots, n)$. We find a polynomial of degree m of best fit to the data points, i.e., we are looking for parameters a_m, a_{m-1}, …, a_0 which minimize the least square error function

$$F(a_m, a_{m-1}, \ldots, a_1, a_0) := \sum_{i=0}^{n} (a_m x_i^m + a_{m-1} x_i^{m-1} + \cdots + a_1 x_i + a_0 - y_i)^2,$$

a function of $m+1$ variables. If $n \leq m$, then there is a polynomial of degree m which interpolates the given data (the minimal value of F is 0). So the coefficients can be obtained by polynomial interpolation. Therefore, we assume for the rest of this section that $m < n$, and in this case F can be positive at every point.

Using Theorem 8.2 we get that F can have an extremum at a point where all partial derivatives are equal to 0. Easy computation gives

$$\begin{aligned}
\frac{\partial F}{\partial a_m}(a_m, a_{m-1}, \dots, a_0) &= 2\sum_{i=0}^{m}(a_m x_i^m + a_{m-1}x_i^{m-1} + \dots + a_0 - y_i)x_i^m,\\
\frac{\partial F}{\partial a_{m-1}}(a_m, a_{m-1}, \dots, a_0) &= 2\sum_{i=0}^{m}(a_m x_i^m + a_{m-1}x_i^{m-1} + \dots + a_0 - y_i)x_i^{m-1},\\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
\frac{\partial F}{\partial a_0}(a_m, a_{m-1}, \dots, a_0) &= 2\sum_{i=0}^{m}(a_m x_i^m + a_{m-1}x_i^{m-1} + \dots + a_0 - y_i).
\end{aligned}$$

Making the partial derivatives equal to 0 and rearranging the resulting system, we get the *normal equations*

$$\begin{aligned}
a_m\sum_{i=0}^{n}x_i^{2m} &+ a_{m-1}\sum_{i=0}^{n}x_i^{2m-1} + \dots + a_1\sum_{i=0}^{n}x_i^{m+1} + a_0\sum_{i=0}^{n}x_i^m &&= \sum_{i=0}^{n}x_i^m y_i\\
a_m\sum_{i=0}^{n}x_i^{2m-1} &+ a_{m-1}\sum_{i=0}^{n}x_i^{2m-2} + \dots + a_1\sum_{i=0}^{n}x_i^m + a_0\sum_{i=0}^{n}x_i^{m-1} &&= \sum_{i=0}^{n}x_i^{m-1} y_i\\
&\vdots &&\vdots\\
a_m\sum_{i=0}^{n}x_i^{m+1} &+ a_{m-1}\sum_{i=0}^{n}x_i^m + \dots + a_1\sum_{i=0}^{n}x_i^2 + a_0\sum_{i=0}^{n}x_i &&= \sum_{i=0}^{n}x_i y_i\\
a_m\sum_{i=0}^{n}x_i^m &+ a_{m-1}\sum_{i=0}^{n}x_i^{m-1} + \dots + a_1\sum_{i=0}^{n}x_i + a_0(n+1) &&= \sum_{i=0}^{n}y_i
\end{aligned} \tag{9.4}$$

We prove that the linear system (9.4) has a unique solution. For this it is enough to show that the coefficient matrix

$$\mathbf{A} := \begin{pmatrix}
\sum_{i=0}^{n}x_i^{2m} & \sum_{i=0}^{n}x_i^{2m-1} & \cdots & \sum_{i=0}^{n}x_i^{m+1} & \sum_{i=0}^{n}x_i^m\\
\sum_{i=0}^{n}x_i^{2m-1} & \sum_{i=0}^{n}x_i^{2m-2} & \cdots & \sum_{i=0}^{n}x_i^m & \sum_{i=0}^{n}x_i^{m-1}\\
\vdots & \vdots & & \vdots & \vdots\\
\sum_{i=0}^{n}x_i^m & \sum_{i=0}^{n}x_i^{m-1} & \cdots & \sum_{i=0}^{n}x_i & \sum_{i=0}^{n}1
\end{pmatrix}$$

is invertible. It is enough to show by Theorem 3.9 that $\mathbf{A}$ is positive definite. The jk-th element of the matrix $\mathbf{A}$ is given by formula $\sum_{i=0}^{n}x_i^{2m+2-j-k}$, where $j,k = 1,2,\dots,m+1$.

Let $\mathbf{z} = (z_1, z_2, \dots, z_{m+1}) \in \mathbb{R}^{m+1}$. Simple calculations give

$$\begin{aligned}\mathbf{z}^T\mathbf{A}\mathbf{z} &= \sum_{j=1}^{m+1}\sum_{k=1}^{m+1}\sum_{i=0}^{n} x_i^{2m+2-j-k} z_j z_k \\ &= \sum_{i=0}^{n}\sum_{j=1}^{m+1}\sum_{k=1}^{m+1} x_i^{m+1-j} z_j x_i^{m+1-k} z_k \\ &= \sum_{i=0}^{n}\left(\sum_{j=1}^{m+1} x_i^{m+1-j} z_j\right)^2.\end{aligned}$$

Suppose that $\mathbf{z}^T\mathbf{A}\mathbf{z} = 0$. Then we have that $\sum_{j=1}^{m+1} x_i^{m+1-j} z_j = 0$ for all $i = 0, 1, \dots, n$. So if there are $m+1$ distinct mesh points, then the polynomial $p(x) := \sum_{j=1}^{m+1} z_j x^{m+1-j}$ of degree at most m has $m+1$ distinct roots. Therefore, the Fundamental theorem of algebra (Theorem 2.9) yields that p must be identically equal to 0, i.e., $z_j = 0$ for all $j = 1, 2, \dots, m+1$. Hence we get that $\mathbf{A}$ is positive definite, and so system (9.4) has a unique solution denoted by $\bar{\mathbf{a}}$. Since

$$\frac{\partial^2 F}{\partial a_j\, \partial a_k}(\bar{\mathbf{a}}) = 2\sum_{i=0}^{n} x_i^{j+k},$$

we get $F''(\bar{\mathbf{a}}) = 2\mathbf{A}$. Therefore, it follows from Theorem 8.1 that F has a local minimum at $\bar{\mathbf{a}}$, and since F is a quadratic function, it is also a global minimum. We can summarize our result in the next theorem.

Theorem 9.3. *Let $m < n$, and given data point (x_i, y_i) $(i = 0, 1, \dots, n)$ such that there exist at least $m+1$ distinct mesh points x_i. Then the problem*

$$\min_{(a_m,\dots,a_0)\in\mathbb{R}^{m+1}} \sum_{i=0}^{n} (a_m x_i^m + a_{m-1}x_i^{m-1} + \cdots + a_1 x_i + a_0 - y_i)^2$$

has a unique solution which satisfies the normal equations (9.4).

Example 9.4. Find a parabola of best fit to the data

x_i	-1.0	-0.5	0.0	1.0	2.0	3.0	3.5
y_i	1.6	1.7	1.9	1.5	0.6	-0.1	-1.0

We list the data in the first two columns of Table 9.2, and fill out the rest of the columns. In the last line we compute the sum of the numbers in the respective columns, and we use these numbers in the normal equations (9.4):

$$\begin{aligned} 249.1250a &+ 77.750b &+ 27.50c &= -7.225 \\ 77.750a &+ 27.50b &+ 8.0c &= -3.55 \\ 27.50a &+ 8.0b &+ 7c &= 6.2. \end{aligned}$$

Its solution is $a = -0.196021$, $b = -0.084748$ and $c = 1.752653$. The graph of the corresponding parabola and the given data point can be seen in Figure 9.2. The error of the fitting is

$$\sum_{i=0}^{6}(-0.196021x_i^2 - 0.084748x_i + 1.752653 - y_i)^2 = 0.0964456.$$

□

Table 9.2: Parabola fitting

x_i	y_i	x_i^4	x_i^3	x_i^2	$x_i^2 y_i$	$x_i y_i$
-1.0	1.4	1.0000	-1.000	1.00	1.400	-1.40
0.0	1.9	0.0000	0.000	0.00	0.000	0.00
0.5	1.6	0.0625	0.125	0.25	0.400	0.80
1.0	1.7	1.0000	1.000	1.00	1.700	1.70
2.0	0.2	16.0000	8.000	4.00	0.800	0.40
2.5	-0.1	39.0625	15.625	6.25	-0.625	-0.25
3.0	-2.0	81.0000	27.000	9.00	-18.000	-6.00
8.0	4.7	138.1250	50.750	21.50	-14.325	-4.75

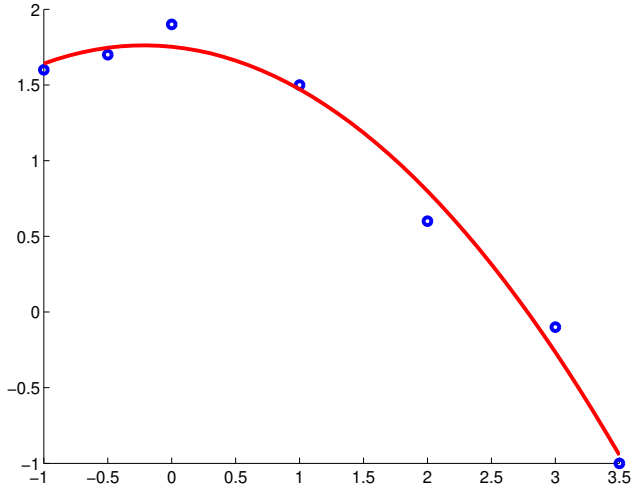

Figure 9.2: Parabola fitting: $y = -0.196021x^2 - 0.084748x + 1.752653$

Exercises

1. Find a parabola of best fit to the given data, and compute the error of the fitting:

(a)

x_i	-2.0	-1.0	1.0	2.0	3.0
y_i	-2.1	1.4	0.5	-2.5	-7.2

(b)

x_i	1.0	2.0	3.0	4.0	5.0	6.0
y_i	2.5	1.2	-2.0	3.9	6.2	8.3

9.3. Special Nonlinear Curve Fitting

The method of the previous sections can be extended easily to nonlinear functions where the unknown parameters appear linearly in the formula, because in this case the resulting normal equations will be linear systems. But in the general case the normal equations can be nonlinear too. Consider an example. Suppose we would like to fit an exponential function of the form be^{ax} to given data (x_i, y_i) $(i = 0, 1, \ldots, n)$. The least square error in this case will define the function

$$F(a, b) = \sum_{i=0}^{n} (be^{ax_i} - y_i)^2,$$

whose critical points are the solutions of the nonlinear system

$$2\sum_{i=0}^{n}(be^{ax_i}-y_i)be^{ax_i}x_i=0$$
$$2\sum_{i=0}^{n}(be^{ax_i}-y_i)e^{ax_i}=0.$$

We cannot solve this system analytically, and it is not easy to analyse whether this system has a unique solution or several solutions, or in the latter case, which solution minimizes the error function. Certainly, we can solve the system numerically, or we can minimize F by a numerical method.

But now we define the method of *linearization* for this special example. We observe that if we take the natural logarithm of both sides of the equation $y=be^{ax}$, then we get the relation $\ln y=\ln b+ax$, where $\ln y$ depends linearly on x. We introduce the new variables: $X:=x$, $Y:=\ln y$, $A:=a$ and $B:=\ln b$. So we can fit a line of the form $Y=AX+B$ to the data points $(x_i,\ln y_i)$. Let $\bar{A}$ and $\bar{B}$ be the solution of this linear fitting. Then the function $\bar{b}e^{\bar{a}x}$ can be considered as the best fit to the points (x_i,y_i), where $\bar{a}=\bar{A}$, $\bar{b}=e^{\bar{B}}$. Note that this linearization does not give us the solution of the original nonlinear fitting problem. But its solution can be computed easily, so it is used frequently in practice.

Example 9.5. Fit an exponential function be^{ax} to the data

x_i	0.0	1.0	1.5	2.0	3.0	4.0
y_i	0.3	0.7	0.9	1.2	1.8	2.7

using linearization. The linearized data can be seen in Table 9.3. The corresponding Gaussian normal equations are

$$\begin{aligned}32.25A &+ 11.5B &&= 5.586294\\ 11.5A &+ 6B &&= 0.097352,\end{aligned}$$

which gives $A=0.528951$ and $B=-0.997597$. So the solution of the linearized fitting is $0.368765^{0.528951x}$. Its graph and the data point can be seen in Figure 9.3. The error of the linear fitting is

$$\sum_{i=0}^{5}(0.528951x_i-0.997597-\ln y_i)^2=0.095396,$$

and the error of the nonlinear fitting is

$$\sum_{i=0}^{5}(0.368765^{0.528951x_i}-y_i)^2=0.165543.$$

□

Example 9.6. Fit a power function of the form bx^a to the given data

x_i	0.5	1.0	1.5	2.5	3.0
y_i	0.7	1.1	1.6	2.1	2.3

Table 9.3: Fitting an exponential function be^{ax}

x_i	y_i	$\ln y_i$	x_i^2	$x_i \ln y_i$
0.0	0.3	-1.203973	0.00	0.000000
1.0	0.7	-0.356675	1.00	-0.356675
1.5	0.9	-0.105361	2.25	-0.158041
2.0	1.2	0.182322	4.00	0.364643
3.0	1.8	0.587787	9.00	1.763360
4.0	2.7	0.993252	16.00	3.973007
11.5		0.097352	32.25	5.586294

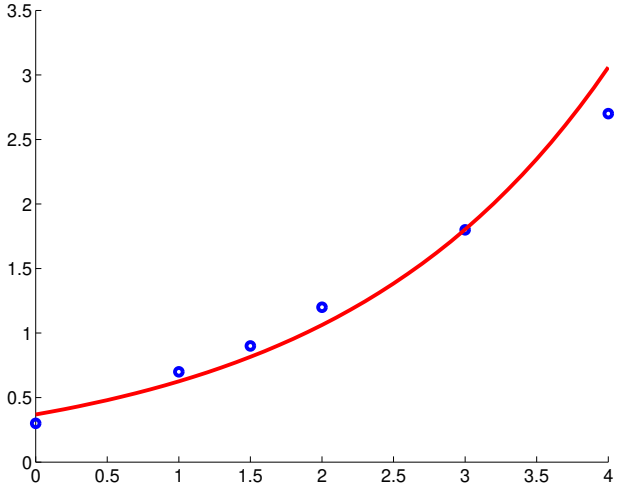

Figure 9.3: Fitting an exponential function be^{ax}: $y = 0.368765^{0.528951x}$

Here we can also use the method of linearization: consider the relation $\ln y = a \ln x + \ln b$ which follows from the equation $y = bx^a$. Then $\ln y$ depends linearly on $\ln x$. Therefore, we fit a line to the data points $(\ln x_i, \ln y_i)$. The computation is shown in Table 9.4, the corresponding Gaussian normal equations are:

$$\begin{array}{rcrcr} 2.691393A & + & 1.727221B & = & 2.032673 \\ 1.727221A & + & 5B & = & 1.783485. \end{array}$$

Its solution is $A = 0.676257$, $B = 0.123088$, and hence the original parameters are $a = A = 0.676257$ and $b = e^B = e^{0.123088} = 1.130984$. The error of the linear fitting is

$$\sum_{i=0}^{4} (0.676257 \ln x_i + 0.123088 - \ln y_i)^2 = 0.007279,$$

and the error of the original nonlinear fitting is

$$\sum_{i=0}^{4} (1.130984 x_i^{0.676257} - y_i)^2 = 0.019616.$$

□

Exercises

1. Fit an exponential function be^{ax} to the given data, and compute the error of the fitting:

Table 9.4: Fitting of a power function bx^a

x_i	y_i	$\ln x_i$	$\ln y_i$	$(\ln x_i)^2$	$\ln x_i \ln y_i$
0.5	0.7	-0.693147	-0.356675	0.480453	0.247228
1.0	1.1	0.000000	0.095310	0.000000	0.000000
1.5	1.6	0.405465	0.470004	0.164402	0.190570
2.5	2.1	0.916291	0.741937	0.839589	0.679830
3.0	2.3	1.098612	0.832909	1.206949	0.915044
		1.727221	1.783485	2.691393	2.032673

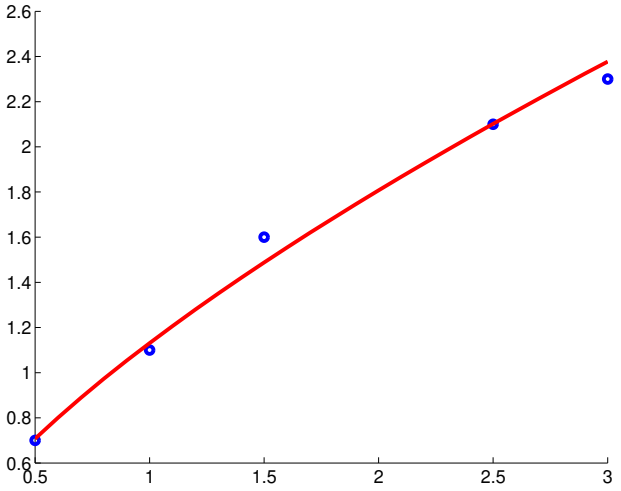

Figure 9.4: Fitting of a power function bx^a: $y = 1.130984x^{0.676257}$

(a)

x_i	-2.0	-1.0	1.0	2.0	3.0
y_i	0.6	0.9	1.6	2.3	2.9

(b)

x_i	1.0	1.5	2.0	2.5	3.0	3.5
y_i	1.3	1.6	1.9	2.2	3.0	4.1

2. Fit a power function bx^a for the given data, and compute the error of the fitting:

(a)

x_i	1.0	3.0	4.0	5.0	6.0	9.0
y_i	1.6	1.9	2.2	2.3	3.4	4.9

(b)

x_i	1.0	2.0	3.0	4.0	5.0
y_i	0.7	2.8	7.5	14.8	25.6

3. Solve the previous exercises using numerical minimization of the nonlinear least square error by Newton's method.

Chapter 10
Ordinary Differential Equations

In this chapter we study numerical solution techniques of ordinary differential equations (ODEs). We define the Euler's, Taylor's and Runge–Kutta methods.

10.1. Review of Differential Equations

In this chapter we investigate approximate solutions of the initial value problem (IVP)

$$y' = f(t, y), \qquad y(t_0) = y_0 \tag{10.1}$$

on a finite time interval $[t_0, T]$. For simplicity we study the case when $y = y(t)$ is a real function, i.e., we assume that

$$f\colon [t_0, T] \times \mathbb{R} \to \mathbb{R}, \qquad y_0 \in \mathbb{R}.$$

The methods we define can be generalized to the system case: then the unknown variable $\mathbf{y} = \mathbf{y}(t)$ denotes a vector of m dimension, and the system has the form

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \qquad \mathbf{y}(t_0) = \mathbf{y}^{(0)}, \tag{10.2}$$

where

$$\mathbf{f}\colon [t_0, T] \times \mathbb{R}^m \to \mathbb{R}^m, \qquad \mathbf{y}^{(0)} \in \mathbb{R}^m.$$

We introduce the following definition: The function $f : [t_0, T] \times \mathbb{R} \to \mathbb{R}$ is called *Lipschitz continuous* in its second variable with the *Lipschitz constant* L if

$$|f(t, y) - f(t, \tilde{y})| \le L|y - \tilde{y}| \qquad \text{for all } t \in [t_0, T] \text{ and } y, \tilde{y} \in \mathbb{R}. \tag{10.3}$$

This notion can be easily generalized to the system case if instead of the absolute value we use a vector norm in the previous definition.

It is known from the theory of ODEs that the existence of solution of the IVPs (10.1) or (10.2) follows if the functions f or $\mathbf{f}$ are continuous. To get the uniqueness of the solutions, we have to assume also the Lipschitz continuity of f or $\mathbf{f}$ in its second variable. Therefore, we have the following result (formulated for the scalar case):

Theorem 10.1. *Suppose that $f : [t_0, T] \times \mathbb{R} \to \mathbb{R}$ is continuous and it is Lipschitz continuous in its second variable. Then the IVP (10.1) has a unique solution on the interval $[0, T]$ for all initial value $y_0 \in \mathbb{R}$.*

We note that the Lipschitz continuity of f in Theorem 10.1 and also in later results, i.e., the assumption that inequality (10.3) holds for all $y, \bar{y} \in \mathbb{R}$ is a strong condition on f. Instead of it we could assume the so-called *local Lipschitz continuity*: for every interval $[a, b]$ for which $y_0 \in (a, b)$ there exists a constant $L > 0$ (which depends on $[a, b]$) such that (10.3) holds for all $t \in [t_0, T]$, $y, \bar{y} \in [a, b]$. This property holds for most of the functions which are important in applications. For example, it is enough to assume that f be continuously differentiable with respect to its second derivative. Then it implies that f is locally Lipschitz continuous in its second variable (see Exercise 3). But from the local Lipschitz continuity it does not follow that the solution of the IVP (10.1) exists on $[t_0, T]$. It follows only that there exists a $0 < \bar{T} \leq T$ such that the IVP (10.1) has a unique solution on the interval $[t_0, \bar{T}]$ (see Exercise 4). To avoid this technical problem we will assume in later results that f is globally Lipschitz continuous in its second variable, i.e., (10.3) holds.

It is known that the scalar mth-order IVP

$$y^{(m)} = f(t, y, y', \dots, y^{(m-1)}), \quad y(t_0) = y_0,\ y'(t_0) = y_1, \dots,\ y^{(m-1)}(t_0) = y_{m-1}$$

is equivalent to an IVP of the form (10.2), where

$$\mathbf{y} = (y, y', \dots, y^{(m-1)})^T, \quad \text{and} \quad \mathbf{y}^{(0)} = (y_0, y_1, \dots, y_{m-1})^T.$$

So for simplicity, later we will study only scalar IVPs of the form (10.1), but most of the results can be generalized to the system case and to mth-order IVPs too.

Exercises

1. Reformulate the following higher order scalar IVPs as an equivalent system of the form (10.2):

   (a) $y'' + 5y' = e^{2t-1}$, $\quad y(0) = 3, \quad y'(0) = -1,$

   (b) $y'' - t^2 y' + ty = 0$, $\quad y(1) = 1, \quad y'(1) = 0,$

   (c) $y''' + 4y'' - 2y' + 5y = t^3$, $\quad y(-1) = 2, \quad y'(-1) = -3.$

2. Show that the IVP $y' = \sqrt{|y|}$, $\ y(0) = 0$ has two solutions $y(t) = 0$ and $y(t) = t^2/4$. Show that the function $f(y) = \sqrt{|y|}$ is not Lipschitz continuous in y.

3. Prove that if the function $f\colon [t_0, T] \times \mathbb{R} \to \mathbb{R}$ is continuously differentiable in its second variable, then f is locally Lipschitz continuous in its second variable.

4. Show that the IVP $y' = y^2$, $y(0) = 1$ has no solution on the interval $[0, T]$ for $T \geq 1$. Show that the function $g(y) = y^2$ is not globally Lipschitz continuous in y, but it is locally Lipschitz continuous.

10.2. Euler's Method

Consider the IVP (10.1). In this section we investigate the simplest numerical approximation method for solving ODEs, the *Euler's method*. Given a finite interval $[t_0, T]$, and equidistant mesh points $t_0 < t_1 < \cdots < t_n = T$, where $h = (T - t_0)/n$, and $t_i = t_0 + ih$ $(i = 0, \ldots, n-1)$. Note that the Euler's method can be easily extended for non-equidistant mesh points, but for simplicity, here we study only the case of the equidistant mesh. The function values $y(t_i)$ are approximated by the so-called *Euler sequence* z_i defined by

$$z_{i+1} = z_i + hf(t_i, z_i), \quad (i = 0, 1, 2, \ldots, n-1), \qquad z_0 = y_0. \tag{10.4}$$

We show three different methods to derive the formula of the Euler's method, and then we investigate the truncation error of the approximation. We assume that the function f is continuous.

Method (i): Suppose that $y(t)$ is the solution of the IVP (10.1). Since $y(t)$ satisfies the initial condition, we have that $y(t_0) = y_0$, and hence z_0 is the exact solution value at t_0. We estimate $y(t)$ by its first-order Taylor polynomial around t_0: $y(t) \approx y(t_0)+y'(t_0)(t-t_0)$. Then at $t = t_1$ we get

$$y(t_1) \approx y(t_0) + y'(t_0)h. \tag{10.5}$$

This formula involves the derivative of the solution at t_0, but equation (10.1) yields $y'(t_0) = f(t_0, y(t_0))$. Since $y(t_0) = y_0 = z_0$, we can compute $y'(t_0)$ with the help of t_0 and z_0: $y'(t_0) = f(t_0, z_0)$. Hence relation (10.5) implies $y(t_1) \approx z_1 := z_0+hf(t_0, z_0)$. Therefore, z_1 approximates the value of the solution at t_1. Suppose now that z_i approximates $y(t_i)$. Then following the previous idea, $y(t_{i+1}) \approx y(t_i) + y'(t_i)h$, and since $y(t_i) \approx z_i$ and $y'(t_i) = f(t_i, y(t_i)) \approx f(t_i, z_i)$, we get $y(t_{i+1}) \approx z_{i+1}$, where z_{i+1} is defined by formula (10.4).

Method (ii): The solution satisfies relation $y'(t_i) = f(t_i, y(t_i))$. Applying the first-order difference formula we get

$$y'(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{h},$$

and therefore,

$$\frac{y(t_{i+1}) - y(t_i)}{h} \approx f(t_i, y(t_i)).$$

Rearranging this equation we get $y(t_{i+1}) \approx y(t_i) + hf(t_i, y(t_i))$. Assuming that $y(t_i) \approx z_i$, the expression z_{i+1} defined by (10.4) satisfies $y(t_{i+1}) \approx z_{i+1}$.

Method (iii): Integrating both sides of the equation $y'(t) = f(t, y(t))$ from t_i to t_{i+1} we get

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(s, y(s))\, ds,$$

and hence

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(s, y(s))\, ds. \tag{10.6}$$

We do not know $y(s)$, and therefore, we cannot integrate $f(s, y(s))$ exactly. We use the following simple approximation formula for the definite integral:

$$\int_a^b g(s)\,ds \approx g(a)(b-a). \tag{10.7}$$

This formula can be applied here, since it uses only a function value at the left end point of the interval, which is assumed to be known. With this formula we have $\int_{t_i}^{t_{i+1}} f(s, y(s))\,ds \approx hf(t_i, y(t_i))$, and hence

$$y(t_{i+1}) \approx y(t_i) + hf(t_i, y(t_i)),$$

which gives again formula (10.4).

The geometric interpretation of method (i) is the following: we take the point (t_i, z_i) obtained in the ith step, and consider the tangent line to the solution which goes through this point, and we move to the point on the tangent line with first coordinate t_{i+1}.

Example 10.2. Consider the IVP

$$y' = 2y - 10t^2 + 2t, \qquad y(0) = 1. \tag{10.8}$$

We can easily check that the exact solution of the problem is $y(t) = 5t^2 + 4t + 2 - e^{2t}$. Fix a step size $h > 0$, and consider the equidistant mesh points $t_i = ih$. The Euler sequence is defined by the recursion

$$z_{i+1} = z_i + h\Big(2z_i - 10t_i^2 + 2t_i\Big), \qquad i = 0, 1, 2, \ldots, \qquad z_0 = 1.$$

We printed the first several terms of the sequence and the error of the approximation $e_i = |y(t_i) - z_i|$ in Table 10.1 corresponding to step sizes $h = 0.2$, 0.1 and 0.05. We can observe that the error decreases as h decreases. Moreover, the numerical values indicate that the error is linear in h: when the step size is reduced to its half, the error also reduces approximately to its half. □

Table 10.1: Euler's method

		$h = 0.2$			$h = 0.1$			$h = 0.05$		
t_i	$y(t_i)$	i	z_i	e_i	i	z_i	e_i	i	z_i	e_i
0.0	1.0000	0	1.0000	0.0000	0	1.0000	0.0000	0	1.0000	0.0000
0.2	1.0652	1	1.1000	0.0348	2	1.0830	0.0178	4	1.0742	0.0090
0.4	1.0614	2	1.1340	0.0726	4	1.0986	0.0372	8	1.0802	0.0188
0.6	0.9899	3	1.1034	0.1135	6	1.0481	0.0583	12	1.0194	0.0295
0.8	0.8518	4	1.0097	0.1579	8	0.9329	0.0811	16	0.8930	0.0411
1.0	0.6487	5	0.8547	0.2060	10	0.7547	0.1060	20	0.7025	0.0538

Next we investigate the convergence of the Euler's method. We need the following definition: The *local truncation error* of the Euler's method at the ith mesh point is defined by the number

$$\tau_{i+1} := \frac{y(t_{i+1}) - y(t_i)}{h} - f(t_i, y(t_i)), \qquad (i = 0, 1, \ldots, n-1), \tag{10.9}$$

where $y(t)$ is the solution of the IVP (10.1).

Rearranging equation (10.9) we have

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \tau_{i+1}h. \tag{10.10}$$

This yields that the error at the $(i+1)$st step is $\tau_{i+1}h$, if at the ith step the error is 0, i.e., we made the step from the exact solution value.

Consider the first Taylor approximation of $y(t)$ around t_i:

$$y(t) = y(t_i) + y'(t_i)(t - t_i) + \frac{1}{2}y''(\xi)(t - t_i)^2.$$

From this relation, equation $y'(t_i) = f(t_i, y(t_i))$ and relation (10.10) it follows that the local truncation error of the Euler's method equals to

$$\tau_{i+1} = \frac{h}{2}y''(\xi), \tag{10.11}$$

where $\xi \in (t_i, t_{i+1})$.

We need the following result.

Theorem 10.3. *Let a, b be positive reals, $x_0, x_1, x_2, \ldots, x_n$ be a finite sequence of reals, for which $x_0 \geq -b/a$ and*

$$x_{i+1} \leq (1+a)x_i + b, \qquad i = 0, 1, \ldots, n-1.$$

Then

$$x_i \leq e^{ia}\left(\frac{b}{a} + x_0\right) - \frac{b}{a}$$

holds for $i = 0, 1, \ldots, n$.

Proof. Applying the conditions and simple manipulations we get

$$\begin{aligned}
x_i &\leq (1+a)x_{i-1} + b \\
&\leq (1+a)((1+a)x_{i-2} + b) + b \\
&\vdots \\
&\leq (1+a)((1+a)(\cdots((1+a)x_0 + b)\cdots) + b) + b \\
&= (1+a)^i x_0 + (1 + (1+a) + (1+a)^2 + \cdots + (1+a)^{i-1})b \\
&= (1+a)^i x_0 + \frac{(1+a)^i - 1}{a}b \\
&= (1+a)^i\left(\frac{b}{a} + x_0\right) - \frac{b}{a}.
\end{aligned} \tag{10.12}$$

It follows from $1 + x \leq e^x$ that $(1+x)^i \leq e^{ix}$, which, together with (10.12), implies the statement of the theorem. □

Theorem 10.4. *Let $f\colon [t_0,T]\times\mathbb{R}\to\mathbb{R}$ be continuous and Lipschitz continuous in its second variable with the Lipschitz constant L, and let $z_0,z_1,\dots,z_n$ be the Euler sequence, and $\tau=\max\{|\tau_{i+1}|\colon i=0,1,\dots,n-1\}$. Then*

$$|y(t_i)-z_i|\le\left(e^{L(T-t_0)}-1\right)\frac{\tau}{L},\qquad (i=0,1,\dots,n). \tag{10.13}$$

Proof. Subtracting equations (10.10) and (10.4) we get

$$y(t_{i+1})-z_{i+1}=y(t_i)-z_i+h\Big(f(t_i,y(t_i))-f(t_i,z_i)\Big)+\tau_{i+1}h.$$

Then the triangle inequality, Lipschitz continuity of f, the definition of τ yields

$$\begin{aligned}|y(t_{i+1})-z_{i+1}|&\le|y(t_i)-z_i|+h\Big|f(t_i,y(t_i))-f(t_i,z_i)\Big|+|\tau_{i+1}|h\\&\le|y(t_i)-z_i|+Lh|y(t_i)-z_i|+|\tau_{i+1}|h\\&\le(1+Lh)|y(t_i)-z_i|+\tau h.\end{aligned}$$

Using Theorem 10.3 with $x_i=|y(t_i)-z_i|$, $a=Lh$, $b=\tau h$, the last inequality and relations $x_0=0$ and $nh=t_n-t_0=T-t_0$ imply (10.13). □

The previous theorem gives the error estimate

$$|y(t_i)-z_i|\le K_1\tau,\qquad i=0,1,\dots,n, \tag{10.14}$$

where K_1 is a constant. Hence the error of the Euler sequence is small if the local truncation error is small. Formula (10.11) implies that τ_{i+1} can be estimated by

$$|\tau_{i+1}|\le\frac{M_2}{2}h,\qquad i=0,1,\dots,n-1 \tag{10.15}$$

where $M_2=\max\{|y''(t)|\colon t\in[t_0,T]\}$ (assuming that the solution is twice differentiable). This means that if h is small, then the error is small.

The solution is differentiable and satisfies equation $y'(t)=f(t,y(t))$. So if we assume that f is continuously partially differentiable with respect to both variables, then y is twice continuously differentiable, and

$$y''(t)=\frac{\partial f}{\partial t}(t,y(t))+\frac{\partial f}{\partial y}(t,y(t))y'(t).$$

Here we can use again (10.1) to substitute $y'(t)$:

$$y''(t)=\frac{\partial f}{\partial t}(t,y(t))+\frac{\partial f}{\partial y}(t,y(t))f(t,y(t)). \tag{10.16}$$

Therefore, if f and the partial derivatives of f are bounded, then (10.16) gives an explicit estimate of M_2.

Summarizing the above considerations, we get the next result.

Theorem 10.5. *Let $f\colon [t_0,T]\times\mathbb{R}\to\mathbb{R}$ be continuous and Lipschitz continuity in its second variable, and continuously partially differentiable with respect to both variables. Then the Euler's method converges linearly to the solution of the IVP (10.1), i.e., there exists a constant $K>0$ such that*

$$|y(t_i)-z_i|\le Kh,\qquad i=0,1,\dots,n.$$

Exercises

1. Compute the first 10 terms of the Euler sequence and compute the error of the approximation (using the given exact solution) for the following IVPs:

   (a) $ty'-y=2t,\quad y(1)=1,\quad h=0.1,\quad$ the solution: $y(t)=2t\ln t+t,$

   (b) $y'-2y=6,\quad y(0)=2,\quad h=0.1,\quad y(t)=-3+5e^{2t},$

   (c) $y'-\frac{2}{t}y=1,\quad y(1)=1,\quad h=0.2,\quad y(t)=2t^2-t,$

   (d) $y'=\frac{t}{1+y},\quad y(1)=2,\quad h=0.1,\quad y(t)=\sqrt{t^2+8}-1.$

2. Formulate the Euler's method for systems of differential equations.

3. Solve the following system of differential equations using Euler's method, and give the error of the approximation (using the given solution):

   (a) $\left.\begin{array}{rcl} y_1' &=& 2y_1-3y_2,\\ y_2' &=& -y_1+4y_2,\end{array}\right\}\quad t\in[0,2],\quad y_1(0)=1,\quad y_2(0)=-5,$
   $h=0.1,\quad y_1(t)=-3e^{t}+4e^{5t},\quad y_2(t)=-4e^{5t}-e^{t}.$

   (b) $\left.\begin{array}{rcl} y_1' &=& 2y_1-3y_2,\\ y_2' &=& 3y_1+2y_2,\end{array}\right\}\quad t\in[0,1],\quad y_1(0)=1,\quad y_2(0)=0,$
   $h=0.1,\quad y_1(t)=e^{2t}\cos 3t,\quad y_2(t)=e^{2t}\sin 3t.$

4. Give the equivalent system of differential equations for the following scalar differential equations. Compute the approximate solution of the system using Euler's method, and give the error of the approximation (using the given solution).

   (a) $y''-3y'+2y=2,\quad t\in[0,1]\qquad y(0)=1,\ y'(0)=-1,\quad h=0.1,\quad y(t)=1+e^{t}-e^{2t},$

   (b) $y''-2y'+5y=0,\quad t\in[0,2],\qquad y(1)=1,\quad y'(0)=3,\quad h=0.2,\quad y(t)=e^{t}\sin 2t+e^{t}\cos 2t.$

5. Let $t_i=t_0+ih$ be an equidistant mesh of the interval $[t_0,T]$, $\{z_i\}$ be the corresponding Euler sequence, and $z(t;h)$ be the linear spline function which interpolates the values z_i: $z(t_i;h)=z_i$, $i=0,1,\dots,n$. Prove that

$$\sup_{t\in[t_0,T]}|y(t)-z(t;h)|\to 0,\qquad \text{as } h\to 0.$$

10.3. Effect of Rounding in the Euler's Method

In practice in the application of the Euler's (or any other) method the rounding error can effect the numerical result of the computation. First, when we store the initial value y_0 in the computer, there can occur a rounding error when we replace the number with a machine number. In each step of the computation, we may also observe rounding error in the output. Let z_i denote the exact value of the Euler sequence, and w_i be the numerically computed value of the sequence. Furthermore, let w_0 be the machine number stored instead of y_0. Define $\delta_0 := y_0 - w_0$, and let δ_i be the rounding error in the ith step. Then we have that

$$w_{i+1} = w_i + hf(t_i, w_i) + \delta_{i+1}, \qquad i = 0, 1, 2, \ldots, n-1. \tag{10.17}$$

Subtracting equations (10.17) and (10.4) we get

$$w_{i+1} - z_{i+1} = w_i - z_i + h(f(t_i, w_i) - f(t_i, z_i)) + \delta_{i+1}.$$

Suppose f is Lipschitz continuous in its second variable with the Lipschitz constant L. Let $\delta := \max\{|\delta_1|, |\delta_2|, \ldots, |\delta_n|\}$. Then the triangle inequality yields

$$\begin{aligned} |w_{i+1} - z_{i+1}| &\le |w_i - z_i| + h|f(t_i, w_i) - f(t_i, z_i)| + |\delta_{i+1}| \\ &\le |w_i - z_i| + hL|w_i - z_i| + \delta, \qquad i = 0, 1, 2, \ldots. \end{aligned}$$

Hence Theorem 10.3 gives the next result.

Theorem 10.6. *Let $f\colon [t_0, T] \times \mathbb{R} \to \mathbb{R}$ be continuous, Lipschitz continuous in its second variable with the Lipschitz constant L, and be continuously partially differentiable with respect to both variables. Then*

$$|y(t_i) - w_i| \le \frac{e^{L(T-t_0)} - 1}{L} \left(\frac{hM_2}{2} + \frac{\delta}{h} \right) + |\delta_0| e^{L(T-t_0)}, \qquad i = 0, 1, \ldots, n,$$

where $M_2 := \max\{|y''(t)|\colon\ t \in [t_0, T]\}$ and $\delta := \max\{|\delta_1|, |\delta_2|, \ldots, |\delta_n|\}$.

The factor $\frac{hM_2}{2} + \frac{\delta}{h}$ in Theorem 10.6 is no longer linear in h, moreover

$$\lim_{h \to 0+} \left(\frac{hM_2}{2} + \frac{\delta}{h} \right) = \infty.$$

Hence if h is too small, then the effect of rounding in the Euler's method can be significant. If the step size is much bigger than the rounding error, then the effect of the rounding is small in the output.

Exercises

1. Work out the details of Theorem 10.6.
2. Draw the graph of the function $g(h) = \frac{hM_2}{2} + \frac{\delta}{h}$ which appears in Theorem 10.6. What is its minimum point?
3. Using the optimal step size obtained in the previous exercise compute for the problem of Example 10.2 assuming $\delta = 0.00001$.

10.4. Taylor's Method

The results of Section 10.2 can be repeated for more general methods. Motivated by the Euler's method, we define the following general one step method to approximate the solutions of the IVP (10.1):

$$z_{i+1} = z_i + hF(t_i, z_i; h), \qquad i = 0, 1, \ldots, n-1, \qquad z_0 = y_0, \tag{10.18}$$

where $F\colon [t_0, T] \times \mathbb{R} \times [0, H] \to \mathbb{R}$ for some $H > 0$. (For the Euler's method $F(t, z; h) = f(t, z)$.) In this section we formulate the methods for the case of equidistant mesh points, but the methods can be generalized for the case of non-uniform mesh points too, i.e., for $z_{i+1} = z_i + h_i F(t_i, z_i; h_i)$.

Similarly to the Euler's method, we define the *local truncation error* for the method (10.18) at the ith mesh point by

$$\tau_{i+1} := \frac{y(t_{i+1}) - y(t_i)}{h} - F(t_i, y(t_i); h), \qquad i = 0, 1, \ldots, n-1, \tag{10.19}$$

where $y(t)$ is the exact solution of the IVP (10.1).

Clearly, Theorem 10.4 can be extended to the general one step method (10.18) if F is continuous and Lipschitz continuous in its second variable. The computations after Theorem 10.4 can also be generalized, and inequality (10.14) holds too. If we also assume that (10.15) holds too (it is not automatic), then it yields a result similar to Theorem 10.5. We can prove the following result.

Theorem 10.7. *Let $F\colon [t_0, T]\times\mathbb{R}\times[0, H] \to \mathbb{R}$ be continuous and Lipschitz continuous in its second variable, and be continuously differentiable with respect to its first two variables. Suppose the local truncation error of (10.18) is of order α, i.e., there exists a constant $K_2 > 0$ such*

$$|\tau_{i+1}| \le K_2 h^\alpha$$

for all $i = 0, 1, \ldots, n-1$. Then the approximate solution (10.18) converges to the exact solution of the IVP (10.1) in order α, i.e., there exists a constant $K > 0$ such that

$$|y(t_i) - z_i| \le K h^\alpha, \qquad i = 0, 1, \ldots, n.$$

How can we select F so that the conditions of Theorem 10.7 be satisfied? It is natural from method (i) presented for the Euler's method to consider higher order Taylor polynomial approximation of the solution (assuming it is sufficiently many times differentiable):

$$\begin{aligned} y(t) = y(t_i) + y'(t_i)(t - t_i) + \frac{1}{2}y''(t_i)(t - t_i)^2 + \ldots + \frac{1}{\alpha!}y^{(\alpha)}(t_i)(t - t_i)^\alpha \\ + \frac{1}{(\alpha+1)!}y^{(\alpha+1)}(\xi_i)(t - t_i)^{\alpha+1}, \end{aligned}$$

where $\xi_i \in \langle t, t_i\rangle$. How can we compute higher order derivatives of y? We know that $y'(t) = f(t, y(t))$. Computing the derivatives of both sides we get relation (10.16). If we compute the derivatives of the right hand side of (10.16) and using relation $y'(t) =$

$f(t,y(t))$ we get an expression for $y'''(t)$ in terms of t, $y(t)$, f and the partial derivatives of f. We introduce the notation

$$f^{(i)}(t,y(t)) := \frac{d^i}{d\,t^i}\Big(f(t,y(t))\Big), \tag{10.20}$$

(i.e., $f^{(i)}(t,y(t))$ denotes the ith derivative of the composite function $f(t,y(t))$ with respect to t). $f^{(i)}(t,z)$ denotes the formula which we get when in the formula of $f^{(i)}(t,y(t))$ we replace $y(t)$ with z. Using this notation we get $y^{(i)}(t) = f^{(i-1)}(t,y(t))$, and hence

$$y(t_{i+1}) = y(t_i) + f(t_i,y(t_i))h + \frac{1}{2}f^{(1)}(t_i,y(t_i))h^2 + \ldots + \frac{1}{\alpha!}f^{(\alpha-1)}(t_i,y(t_i))h^\alpha + \frac{1}{(\alpha+1)!}f^{(\alpha)}(\xi_i,y(\xi_i))h^{\alpha+1}.$$

Suppose $f \in C^\alpha$, and define F by

$$F(t,z;h) := f(t,z) + \frac{1}{2}f^{(1)}(t,z)h + \ldots + \frac{1}{\alpha!}f^{(\alpha-1)}(t,z)h^{\alpha-1}. \tag{10.21}$$

Then

$$\tau_{i+1} = \frac{1}{(\alpha+1)!}f^{(\alpha)}(\xi_i,y(\xi_i))h^\alpha,$$

and hence the local truncation error is of order α in h. The method defined by (10.18) and (10.21) is called *Taylor's method* of order α.

Example 10.8. Consider again the problem of Example (10.8), and apply the second-order Taylor's method for it. First compute $f^{(1)}$:

$$\begin{aligned} f^{(1)}(t,y(t)) &= \frac{d}{dt}\left(2y(t) - 10t^2 + 2t\right) = 2y'(t) - 20t + 2 \\ &= (4y(t) - 20t^2 + 4t) - 20t + 2 = 4y(t) - 20t^2 - 16t + 2. \end{aligned}$$

Hence the numerical method is defined by

$$z_{i+1} = z_i + h\Big(2z_i - 10t_i^2 + 2t_i\Big) + \frac{h^2}{2}\Big(4z_i - 20t_i^2 - 16t_i + 2\Big), \qquad i = 0,1,2,\ldots, \qquad z_0 = 1.$$

In Table 10.2 we listed the numerical values of first few terms of this method and the error of the approximation corresponding to step sizes $h = 0.2$ and 0.1. We can see that when the step size reduces to its half, then the error reduces to its quarter, which demonstrates that the method is of order 2. Comparing to the errors presented in Table 10.1 we can see that the errors here are better than that in the Euler's method.

Next we apply the third-order Taylor's method for the same problem. Simple calculations yield

$$f^{(2)}(t,y(t)) = \frac{d}{dt}\left(4y(t) - 20t^2 - 16t + 2\right) = 4y'(t) - 40t - 16 = 8y(t) - 40t^2 - 32t - 16.$$

Hence the third-order Taylor's method is defined by:

$$z_{i+1} = z_i + h\Big(2z_i - 10t_i^2 + 2t_i\Big) + \frac{h^2}{2}\Big(4z_i - 20t_i^2 - 16t_i + 2\Big) + \frac{h^3}{6}(8z_i - 40t_i^2 - 32t_i - 16),$$

for $i = 0,1,2,\ldots$, and $z_0 = 1$. The numerical results can be seen in Table 10.3. We observe smaller error then in the previous example. □

Table 10.2: Second-order Taylor's method

		$h = 0.2$			$h = 0.1$		
t_i	$y(t_i)$	i	z_i	$\lvert y(t_i) - z_i \rvert$	i	z_i	$\lvert y(t_i) - z_i \rvert$
0.0	1.00000	0	1.00000	0.0000e-01	0	1.00000	0.0000e-01
0.2	1.50818	1	1.52000	1.1825e-02	2	1.51160	3.4247e-03
0.4	2.17446	2	2.20960	3.5141e-02	4	2.18467	1.0206e-02
0.6	2.87988	3	2.95821	7.8325e-02	6	2.90270	2.2813e-02
0.8	3.44697	4	3.60215	1.5518e-01	8	3.49229	4.5325e-02
1.0	3.61094	5	3.89918	2.8823e-01	10	3.69537	8.4425e-02

Table 10.3: Third-order Taylor's method

		$h = 0.2$			$h = 0.1$		
t_i	$y(t_i)$	i	z_i	$\lvert y(t_i) - z_i \rvert$	i	z_i	$\lvert y(t_i) - z_i \rvert$
0.0	1.00000	0	1.00000	0.0000e-01	0	1.00000	0.0000e-01
0.2	1.50818	1	1.50933	1.1580e-03	2	1.50834	1.6959e-04
0.4	2.17446	2	2.17791	3.4538e-03	4	2.17497	5.0596e-04
0.6	2.87988	3	2.88761	7.7257e-03	6	2.88102	1.1321e-03
0.8	3.44697	4	3.46233	1.5361e-02	8	3.44922	2.2518e-03
1.0	3.61094	5	3.63958	2.8634e-02	10	3.61514	4.1989e-03

Exercises

1. Solve the IVPs presented in Exercise 1 of Section 10.2 using the second- and third-order Taylor' method.
2. Formulate and apply the fourth- and fifth-order Taylor's method for the IVP (10.8).

10.5. Runge–Kutta Method

The difficulty in the application of the Taylor's method is the computation of the derivatives $f^{(i)}$. Here we can get complicated formulas which can require a lot of computational time, which may result in the accumulation of the rounding errors too. The *Runge–Kutta methods* will preserve the high convergence rates of the Taylor's method, but reduce the computational complexity. The idea is presented for the second-order case:

Let $f \in C^2$, consider the formula of the second-order Taylor's method

$$F(t, z; h) = f(t, z) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t, z) + \frac{\partial f}{\partial y}(t, z) f(t, z)\right).$$

Here, as usual, $\frac{\partial f}{\partial y}$ denotes the partial derivative of f with respect to its second variable. Compare this formula to the following Taylor formula:

$$f(t + a, z + b) = f(t, z) + \frac{\partial f}{\partial t}(t, z)a + \frac{\partial f}{\partial y}(t, z)b + E(t, z, a, b),$$

where the error is of second order

$$E(t,z,a,b)=\frac{1}{2}\left(\frac{\partial^2 f}{\partial t^2}(\xi,\eta)a^2+2\frac{\partial^2 f}{\partial t\,\partial y}(\xi,\eta)ab+\frac{\partial^2 f}{\partial y^2}(\xi,\eta)b^2\right) \tag{10.22}$$

for some $\xi\in\langle t,t+a\rangle$ and $\eta\in\langle z,z+b\rangle$. If we use the parameters $a=h/2$ and $b=f(t,z)h/2$, we get

$$f\left(t+\frac{h}{2},z+\frac{h}{2}f(t,z)\right)=F(t,z;h)+E\left(t,z,\frac{h}{2},\frac{h}{2}f(t,z)\right),$$

so the essential part of $f\left(t+\frac{h}{2},z+\frac{h}{2}f(t,z)\right)$ coincides with $F(t,z;h)$. But the significant difference is that it is much simpler to evaluate $f\left(t+\frac{h}{2},z+\frac{h}{2}f(t,z)\right)$ than $F(t,z;h)$. This motivates to define the approximation sequence

$$z_{i+1}=z_i+hf\left(t_i+\frac{h}{2},z_i+\frac{h}{2}f(t_i,z_i)\right),\quad i=0,1,2,\dots,\qquad z_0=y_0. \tag{10.23}$$

This is called the *midpoint method*. Let τ_{i+1} and $\bar\tau_{i+1}$ be the local truncation error of the midpoint and the second-order Taylor's methods, respectively. Then

$$\begin{aligned}\tau_{i+1}&=\frac{y(t_{i+1})-y(t_i)}{h}-f\left(t_i+\frac{h}{2},y(t_i)+\frac{h}{2}f(t_i,y(t_i))\right)\\&=\frac{y(t_{i+1})-y(t_i)}{h}-F(t_i,y(t_i);h)-E\left(t_i,y(t_i),\frac{h}{2},\frac{h}{2}f(t_i,y(t_i))\right)\\&=\bar\tau_{i+1}-E\left(t_i,y(t_i),\frac{h}{2},\frac{h}{2}f(t_i,y(t_i))\right).\end{aligned}$$

We know from the previous section that $|\bar\tau_{i+1}|\le\bar K h^2$, and (10.22) and $f\in C^2$ imply that there exists $\tilde K$ such that $\left|E\big(t_i,y(t_i),\frac{h}{2},\frac{h}{2}f(t_i,y(t_i))\big)\right|\le\tilde K h^2$. But then $|\tau_{i+1}|\le(\bar K+\tilde K)h^2$ holds, and therefore, the method (10.23) converges quadratically, assuming that the Lipschitz continuity needed in Theorem 10.7 also holds. This is clearly satisfied if f is Lipschitz continuous in its second variable. (See Exercise 2.)

Now we define F in the following way:

$$\begin{aligned}F(t,z;h)&:=\sum_{j=1}^{p}\gamma_j G_j(t,z;h),\\G_1(t,z;h)&:=f(t,z),\\G_j(t,z;h)&:=f\left(t+\alpha_j h,z+h\sum_{k=1}^{j-1}\beta_{jk}G_k(t,z;h)\right),\qquad j=2,3,\dots,p.\end{aligned} \tag{10.24}$$

The class of methods defined by formulas (10.18) and (10.24) is called *(explicit) Runge–Kutta methods*. The goal is to select the parameters so that we get high order local truncation errors.

Consider now the case when $p = 2$. Then

$$F(t, z; h) = \gamma_1 f\,(t, z) + \gamma_2 f(t + \alpha_1 h, z + \beta_{11} h f(t, z))\,.$$

(If $\gamma_1 = 0$, $\gamma_2 = 1$, $\alpha_1 = \beta_{11} = 1/2$, then we get back the midpoint method.) We try to select parameters so that we get third-order local truncation error. We apply the second Taylor formula for the right hand side:

$$\begin{aligned} F(t, z; h) = (\gamma_1 + \gamma_2) f(t, z) &+ h\gamma_2 \Big(\alpha_1 \frac{\partial f}{\partial t}(t, z) + \beta_{11} f(t, z) \frac{\partial f}{\partial y}(t, z) \Big) \\ &+ \frac{h^2}{2} \gamma_2 \Big(\alpha_1^2 \frac{\partial^2 f}{\partial t^2}(t, z) + 2\alpha_1 \beta_{11} f(t, z) \frac{\partial^2 f}{\partial t\, \partial y}(t, z) \\ &+ \beta_{11}^2 (f(t, z))^2 \frac{\partial^2 f}{\partial y^2}(t, z) \Big) + E(t, z, \alpha_1 h, \beta_{11} h f(t, z)), \end{aligned} \tag{10.25}$$

where E is a third-order error term. Compare it to the formula of the third-order Taylor's method

$$\begin{aligned} \tilde{F}(t, z; h) = f(t, z) &+ \frac{h}{2} \left(\frac{\partial f}{\partial t}(t, z) + \frac{\partial f}{\partial y}(t, z) f(t, z) \right) \\ &+ \frac{h^2}{6} \Big(\frac{\partial^2 f}{\partial t^2}(t, z) + 2 f(t, z) \frac{\partial^2 f}{\partial t\, \partial y}(t, z) \\ &+ (f(t, z))^2 \frac{\partial^2 f}{\partial y^2}(t, z) + \frac{\partial f}{\partial t}(t, z) \frac{\partial f}{\partial y}(t, z) + \left(\frac{\partial f}{\partial y}(t, z) \right)^2 f(t, z) \Big). \end{aligned} \tag{10.26}$$

We can see that all the terms of F with at most second order appear in the formula of $\tilde{F}$. But the opposite case is not true: the terms $\frac{\partial f}{\partial t}(t, z) \frac{\partial f}{\partial y}(t, z)$ and $\left(\frac{\partial f}{\partial y}(t, z) \right)^2 f(t, z)$ which appear in (10.26) have no corresponding term in (10.25). This means that we cannot replace all second-order terms of the Taylor's method with the second-order terms of F, so the local truncation error can only be quadratic. But we try to identify as many terms of (10.25) and (10.26) as possible. Therefore, we assume

$$\gamma_1 + \gamma_2 = 1, \qquad \gamma_2 \alpha_1 = \frac{1}{2}, \qquad \gamma_2 \beta_{11} = \frac{1}{2}, \tag{10.27}$$

and

$$\frac{\gamma_2}{2} \alpha_1^2 = \frac{1}{6}, \qquad \gamma_2 \alpha_2 \beta_{11} = \frac{1}{3}, \qquad \frac{\gamma_2}{2} \beta_{11}^2 = \frac{1}{6}. \tag{10.28}$$

For example, $\gamma_1 = \gamma_2 = 1/2$ and $\alpha_1 = \beta_{11} = 1$ satisfy (10.27), but not (10.28). But since all the first-order terms are identified, we get a second-order method. The corresponding method

$$z_{i+1} = z_i + \frac{h}{2} \Big(f(t_i, z_i) + f(t_{i+1}, z_i + h f(t_i, z_i)) \Big), \qquad i = 0, 1, 2, \ldots, \qquad z_0 = y_0 \tag{10.29}$$

is called *modified Euler method*.

If we use the parameter values $\gamma_1 = 1/4$, $\gamma_2 = 3/4$ and $\alpha_1 = \beta_{11} = 2/3$, then both (10.27) and (10.28) are satisfied. The corresponding method, the so-called *Heun's method* is defined by

$$z_{i+1} = z_i + \frac{h}{4}\left(f(t_i, z_i) + 3f\left(t_i + \frac{2h}{3}, z_i + \frac{2}{3}hf(t_i, z_i)\right)\right), \quad i = 0, 1, 2, \ldots,$$
$$z_0 = y_0. \tag{10.30}$$

Both methods are so-called second-order Runge–Kutta methods (since their local truncation error is of second order).

The geometric meaning of the modified Euler method is the following: Suppose the point (t_i, z_i) is given in the ith step of the method. If we used the Euler's method, then we would take one step along with a line through this point with slope $f(t_i, z_i)$, and we would move to the point (t_{i+1}, w_{i+1}) where $w_{i+1} := z_i + hf(t_i, z_i)$. The slope of the tangent line to the graph of the exact solution at this point is $f(t_{i+1}, w_{i+1})$. We compute the average of the two slopes, and move one step along with a line of such averaged slope starting from the point $f(t_i, z_i)$. See Figure 10.1.

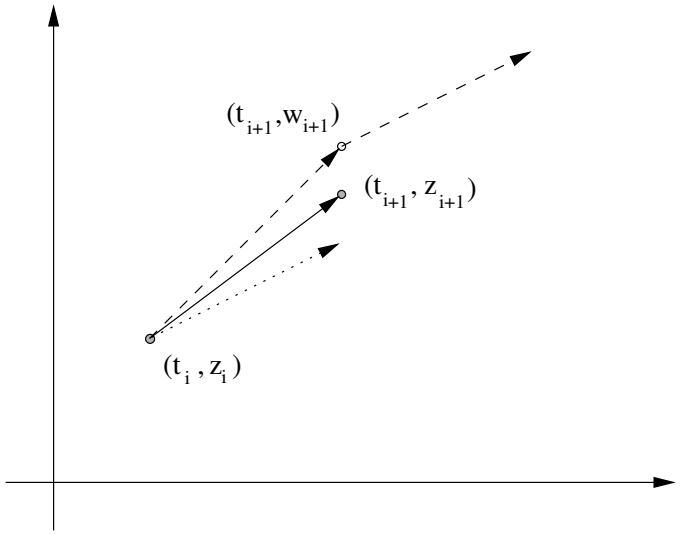

Figure 10.1: Geometric interpretation of the modified Euler method

Following the idea presented above, we can define several other Runge-Kutta methods. It can be shown that for different parameter values p the corresponding methods of the form can have at most the order of the local truncation error given in the following table:

p	1	2	3	4	5	6	7	8	9	10
maximal order of the method	1	2	3	4	4	5	6	6	7	7

One of the most popular ODE approximation method of the form (10.24) is the "clas-

sical" Runge–Kutta method:

$$\begin{aligned} z_0 &= y_0, \\ w_{i,1} &= f(t_i, z_i), \\ w_{i,2} &= f\left(t_i + \frac{h}{2}, z_i + \frac{h}{2} w_{i,1}\right), \\ w_{i,3} &= f\left(t_i + \frac{h}{2}, z_i + \frac{h}{2} w_{i,2}\right), \\ w_{i,4} &= f\left(t_{i+1}, z_i + h w_{i,3}\right), \\ z_{i+1} &= z_i + \frac{h}{6}(w_{i,1} + 2w_{i,2} + 2w_{i,3} + w_{i,4}), \qquad i = 0, 1, 2, \ldots. \end{aligned} \tag{10.31}$$

It can be shown that this method has a fourth-order local truncation error (if $f \in C^5$). The derivation of the method and the proof of its order is not presented here.

Example 10.9. For the IVP (10.8) we applied the modified Euler, Heun and the classical fourth-order Runge–Kutta methods using step size $h = 0.1$. The numerical results are presented in Table 10.4. □

Table 10.4: Runge–Kutta methods

		modified Euler		Heun		classical	
t_i	$y(t_i)$	z_i	$\lvert y(t_i) - z_i\rvert$	z_i	$\lvert y(t_i) - z_i\rvert$	z_i	$\lvert y(t_i) - z_i\rvert$
0.0	1.0000	1.0000	0.0000e-01	1.0000	0.0000e-01	1.0000	0.0000e-01
0.2	1.5082	1.5005	7.6753e-03	1.5042	3.9753e-03	1.5082	1.1773e-05
0.4	2.1745	2.1570	1.7415e-02	2.1663	8.2078e-03	2.1744	2.6024e-05
0.6	2.8799	2.8505	2.9398e-02	2.8679	1.1995e-02	2.8798	4.2338e-05
0.8	3.4470	3.4035	4.3486e-02	3.4331	1.3882e-02	3.4469	5.9304e-05
1.0	3.6109	3.5521	5.8862e-02	3.5998	1.1100e-02	3.6109	7.3610e-05

Exercises

1. Solve the IVPs presented in Exercise 1 of Section 10.2 using the midpoint, modified Euler, Heun and the classical fourth-order Runge–Kutta methods.
2. Prove that if f is Lipschitz continuous in its second variable, then the function
$$F(t, z; h) = \frac{1}{2} f\left(t + \frac{h}{2}, z + \frac{h}{2} f(t, z)\right)$$
of the midpoint method is also Lipschitz continuous in its second variable.
3. Similarly to the method (iii) of the Euler's method, derive formula (10.29).
4. Show that the midpoint method, the modified Euler and Heun method gives back the same approximation for all step sizes for the IVP
$$y' = 2 - t - y, \qquad y(0) = 1.$$
5. Find a geometric interpretation to the classical fourth-order Runge–Kutta method.
6. Show that if f depends only on t, then the classical fourth-order Runge–Kutta method reduces to the Simpson's rule.

References

[1] K. E. Atkinson, An Introduction to Numerical Analysis, Wiley, New York, 1978.

[2] R. L. Burden, J. D. Faires, Numerical Analysis, Brooks/Cole, Cengage Learning, 2011.

[3] J. E. Dennis Jr., R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Classics in Applied Mathematics (Book 16), Society for Industrial and Applied Mathematics, 1987.

[4] E. Isaacson, H. B. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

[5] A. Ralston, P. Rabinowitz, A First Course in Numerical Analysis, Dover Books on Mathematics, Dover Publications, 2001.

[6] L. Ridgway Scott, Numerical Analysis, Princeton University Press, 2011.

[7] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York, 1980.

Index