Chapter 1

Conditioning, Stability and Finite Arithmetic

A problem relates solutions to data. Conditioning is related to the perturbation behaviour of the problem (that is, how small changes in the inputs affects the results). Stability is related to the perturbation behavior of an algorithm used to solve the problem. This chapter is entirely devoted to these three issues.

1.1 Conditioning of a problem

A problem $\mathcal{P}$ has inputs $x \in X$ and outputs $y \in Y$ where X and Y are some, normed, spaces for data x and solutions y, respectively. In an abstract manner, the problem $\mathcal{P}$ may be seen as a function $f : X \mapsto Y$.

Example 1.1 *In the simplest, yet useful, model for the radioactive decay of a radioactive material, the number $N(t)$ of radioactive atoms inside a sample of this material is a function of time t as*

$$N(t) = N_0 e^{-t/\tau}, \quad t \geq 0$$

In this equation, $N_0 = N(0)$ is the initial (that is, when the sample was created) number of radioactive atoms. The time constant $\tau > 0$ is a given parameter and it is characteristic of each material. The time needed to halve the initial amount of radioactive atoms is called the half-time $t_{1/2}$ and can be evaluated from the previous equation. Requiring that $N(t_{1/2}) = N_0/2$, we get

$$N(t_{1/2}) = \frac{N_0}{2} \quad \Leftrightarrow \quad N_0 e^{-t_{1/2}/\tau} = \frac{N_0}{2} \quad \Leftrightarrow \quad \frac{-t_{1/2}}{\tau} = \ln(1/2)$$

which gives $t_{1/2} = \tau \ln(2)$. The following table shows half-times for some materials.

Material	Uranium-238	Carbon-14	Phosphorus-32	Technetium-99m
$t_{1/2}$	4.51×10^9 years	5.73×10^3 years	14.3 days	6 hours

Let X and Y be some normed spaces. Consider a problem $\mathcal{P}$ with input data $x_0 \in X$. The corresponding output is $y_0 = f(x_0) \in Y$. Now, consider, instead of x_0, a small, allowable, perturbation of the data x_0, say $x_0 + \delta x$. The output changes to $y_0 + \delta y = f(x_0 + \delta x)$, see Figure 1.1.
We say that the problem $\mathcal{P}$ with input x_0 is *well-conditioned* if all small, allowable, perturbations δx lead to small perturbations δy. Otherwise, if there is at least one small perturbation δx which leads to a large perturbation δy we say that the problem $\mathcal{P}$ with input x_0 is *ill-conditioned*.

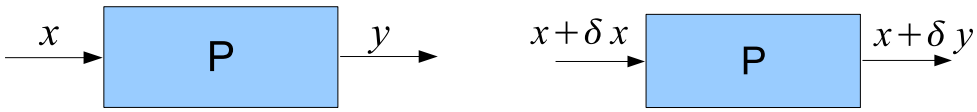


Figure 1.1: A problem $\mathcal{P}$ may be seen as a black box where output y is related to input x throughout a function f. On the right, the problem P with perturbed input $x + \delta x$ and the corresponding perturbed output $y + \delta y$.

One of the most useful, thought not the unique, measure for the conditioning of a problem $\mathcal{P}$ at x_0 is the *relative condition number* K. Denoting by $||\cdot||$ a norm, it is defined as

$$K = \sup_{\delta x} \frac{\frac{||\,\delta y\,||}{||\,y_0\,||}}{\frac{||\,\delta x\,||}{||\,x_0\,||}}$$

where the supremum is taken over all the allowable, small (infinitesimal from a mathematical point of view), perturbations δx. We say that the problem is well-conditioned if K is small, for example less than, about, 10^2; the problem is ill-conditioned if K is large, for example greater than 10^6.

Remark 1.1 *The interesting for the conditioning of a problem is strictly based on the possibility to find a stable algorithm to solve the problem on a computer. Generally, if the problem is well-conditioned, it is possible to find, among all the possible algorithms that can be used to solve the problem, a stable one. If the problem is ill-conditioned, it is better to rewrite the problem in a well-defined manner before attempt to solve it on a computer.*

Let us give some examples.

Example 1.2 (Evaluation of a function) *Consider the computation of $y_0 = f(x_0)$ where f is a differentiable given function. Using Taylor expansion, we have*

$$f(x_0 + \delta x) = f(x_0) + f'(x_0) \cdot \delta x + o(\,\delta x\,) \qquad \Rightarrow \qquad \delta y = f(x_0 + \delta x) - f(x_0) \approx f'(x_0) \cdot \delta x$$

and so, recalling that $y_0 = f(x_0)$, we find

$$\frac{\delta y}{y_0} = \frac{x_0 \cdot f'(x_0)}{f(x_0)} \cdot \frac{\delta x}{x_0} \qquad \Rightarrow \qquad K = \left|\, \frac{x_0 \cdot f'(x_0)}{f(x_0)} \,\right|$$

As an example, consider $f(x) = \sqrt{x+1} - \sqrt{x}$, $x \geq 0$. Since the first derivative of f may be rewritten as

$$f'(x) = \frac{-f(x)}{2\sqrt{x \cdot (x+1)}}$$

we obtain

$$K = \frac{|x_0|}{2\sqrt{x_0 \cdot (x_0+1)}}$$

and so the problem is well-conditioned for all $x_0 \geq 0$ since $K \leq 1/2$ with $K \approx 1/2$ for $x_0 \to +\infty$.

Example 1.3 (Root finding: multiple roots) *Consider the computation of the roots of the polynomial*

$$p(x) = x^2 - 4x + \alpha$$

as a function of the data α. Since we can rewrite $p(x) = (x-2)^2 + \alpha - 4$, its roots are

$$x_1 = 2 + \sqrt{4-\alpha}, \qquad x_2 = 2 - \sqrt{4-\alpha}.$$

Consider x_1 and x_2 as functions of α. The condition numbers at α_0 are

$$K_1 = \frac{|\,\alpha_0\,|}{2 \cdot \sqrt{4-\alpha_0}\,(\,2 + \sqrt{4-\alpha_0}\,)}, \qquad K_2 = \frac{|\,\alpha_0\,|}{2 \cdot \sqrt{4-\alpha_0}\,|\,2 - \sqrt{4-\alpha_0}\,|}$$

So, the computation of both roots is an ill-conditioned problem for $\alpha_0 \approx 4$, that is near the double root $x_1 = x_2 = 2$. For $\alpha_0 \ll 4$ both x_1 and x_2 are well-conditioned. Note also that

$$\lim_{\alpha_0 \to 0} K_2(\alpha_0) = 1$$

and so also the computation of x_2 for $\alpha_0 \approx 0$ is a well-conditioned problem.

Example 1.4 (Root finding: Wilkinson polynomial) *The computation of the roots of a polynomial is often an ill-conditioned problem even in the case where there isn't multiple roots. Consider the polynomial of degree $n = 4$ with roots $x_i = i$, $i = 1, 2, 3, 4$ defined as*

$$\begin{aligned} p(x) &= x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = \prod_{k=1}^{4} (x - k) \\ &= x^4 - 10x^3 + 35x^2 - 50x + 24 \end{aligned}$$

It is possible to shown that the condition number $K_{i,j}$ of the i-th root with respect to an infinitesimal perturbation on the single coefficient a_j is

$$K_{i,j} = \left| \frac{a_j \cdot x_i^{j-1}}{p'(x_i)} \right|$$

Values of $K_{i,j}$ are shown on the oncoming table

perturbation on	$a_3 = -10$	$a_2 = 35$	$a_1 = -50$	$a_0 = 24$
$K_{1,j}$ *(root $x_1 = 1$)*	*1.7*	*5.8*	*8.3*	*4.0*
$K_{2,j}$ *(root $x_2 = 2$)*	*20.0*	*35.0*	*25.0*	*6.0*
$K_{3,j}$ *(root $x_3 = 3$)*	*45.0*	*52.5*	*25.0*	*4.0*
$K_{4,j}$ *(root $x_4 = 4$)*	*26.7*	*23.3*	*8.3*	*1.0*

So, root $x_3 = 3$ is the most sensitive root when we perturb coefficient $a_2 = 35$. Furthermore, note that the computation of $x_1 = 1$ is a well-conditioned problem against the variation of each one of the coefficients of the polynomial.

It is also possible to investigate the conditioning of a problem directly, without using the relative condition number, as shown in the following examples.

Example 1.5 (Solution of a linear system) *Consider the computation of the solution $\mathbf{x} = [x_1 \ x_2]^T$ of the family of linear systems $A_\epsilon \mathbf{x} = \mathbf{b}$ depending on the parameter $\epsilon \neq 0$ given by*

$$A_\epsilon = \begin{bmatrix} 1 & 1-\epsilon \\ 1+\epsilon & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The solution may be written as

$$\mathbf{x} = A_\epsilon^{-1} \cdot \mathbf{b} = \frac{1}{\epsilon^2} \cdot \begin{bmatrix} 1 & -1+\epsilon \\ -1-\epsilon & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/\epsilon \\ -1/\epsilon \end{bmatrix}$$

Setting $\epsilon = 0.01$, we obtain $\mathbf{x} = [100 \ 100]^T$. Now, consider a small perturbation on b, for example let $\hat{\mathbf{b}} = [0.99 \ 1]^T$. The corresponding solution of the perturbed system $A_\epsilon \mathbf{x} = \hat{\mathbf{b}}$ is $\hat{\mathbf{x}} = [0 \ 1]^T$ which is completely different from the previous one. Thus, in front of a small relative change (we have $| \delta b_1 | / |b_1| = 0.01 = 1\%$) we have a great variation of both x_1 and x_2. For example, we have for x_1

$$\frac{| x_1 - \hat{x}_1 |}{x_1} = \frac{| 100 - 0 |}{100} = 1 = 100\%.$$

Thus, for this value of ϵ the linear system is ill conditioned.
On the other hand, setting $\epsilon = 1$, we obtain $\mathbf{x} = [1\ 1]^T$ and, for the same perturbation of $\mathbf{b}$ as before, we get $\hat{\mathbf{x}} = [0.99\ 0.98]^T$ with a relative error on x_1 equal to 1%. So, for this value of ϵ, the system is well conditioned.

For a linear system of order two, it is also possible to see if it is well or ill conditioned using a picture. Let

$$\begin{cases} a_{11}x + a_{12}y &= b_1 \\ a_{21}x + a_{22}y &= b_2 \end{cases}$$

be the linear system. The corresponding solution is the intersection point of the two lines $a_{11}x + a_{12}y = b_1$ and $a_{21}x + a_{22}y = b_2$ in the (O, x, y) plane (see figure 1.2). As we can see

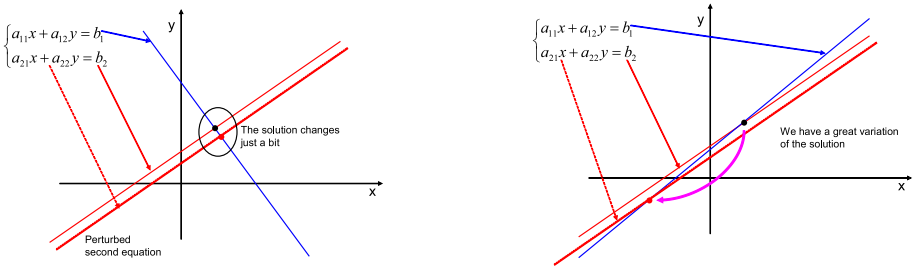


Figure 1.2: A well conditioned linear system (left) and an ill-conditioned linear system (right).

on the left, if the two lines are near orthogonal a small change in the coefficients of one line (or on both lines) does not change the intersection point (and thus the solution of the linear system). So, in this case, the linear system is well conditioned. On the other hand, if the two lines are near to be parallel, even a small changes in the coefficients of one line gives a large variation on the intersection point. Thus, the linear system is ill conditioned. □

Example 1.6 (Computation of the eigenvalues) *The computation of the eigenvalues of a non symmetric matrix is often an ill-conditioned problem. To see this, consider the matrices A and its perturbed version $\hat{A}$ defined as*

$$A = \begin{bmatrix} 101 & 110 \\ -90 & -98 \end{bmatrix} \qquad \hat{A} = \begin{bmatrix} 100 & 110 \\ -90 & -98 \end{bmatrix}$$

Note that the only difference among A and $\hat{A}$ is that $a_{11} = 101$ and $\hat{a}_{11} = 100$. That is, a small change, of the order of 1%. However, the eigenvalues of the two matrices are

$$\begin{array}{llll} \lambda_1 = 1 & \lambda_2 = 2 & \text{for matrix } A \\ \hat{\lambda}_1 \approx 1 + 10i & \hat{\lambda}_2 \approx 1 - 10i & \text{for matrix } \hat{A} \end{array}$$

So, we have a large change in the eigenvalues a front of a small change in the matrix. Thus, according to our definition, the problem is ill-conditioned.

As a note, which we do not prove, the computation of the eigenvalues of a symmetric matrix is a well-conditioned problem. □

1.2 Floating point system

The floating point system is the set containing real numbers defined as

$$\mathbb{F}(\beta, t, L, U) = \{\ 0\ \} + \left\{\ x \in \mathbb{R} \mid x = (-1)^s \cdot \beta^p \sum_{k=1}^{t} a_k\, \beta^{-k}\ \right\} \tag{1.1}$$

where

- β, the base, is an integer with $\beta \geq 2$. Common used bases are $\beta = 10$ (this is the base we use to count), $\beta = 2$ and $\beta = 16$.
- L and U are two integer numbers. Typically we have $L < 0 < U$. The scaling factor p is an integer satisfying $L \leq p \leq U$.
- t is a positive integer representing the number of figures a_k, $k = 1, \ldots, t$ of each floating point number. The unique representation of each floating point number requires $a_1 > 0$. Let us show what happens if this is not the case. Consider, as an example, the number $x = 1$ and $\mathbb{F}(10, 5, -6, 6)$. Then, the number $x = 1$ has different representations: 0.1×10^1, 0.01×10^2, 0.001×10^3 and many others.
- $s = 0$ for positive numbers and $s = -1$ for negative numbers.

Each element of $\mathbb{F}$ is said a floating point number (or a machine number).

Theorem 1.1 *The set of floating point system $\mathbb{F}(\beta, t, L, U)$ has the following properties.*

(a) $\mathbb{F} \subset \mathbb{R}$.

(b) if $x \in \mathbb{F}$ *then also* $-x \in \mathbb{F}$.

(c) $\mathbb{F}$ *has* $1 + 2 \cdot (U - L + 1) \cdot (\beta - 1) \cdot \beta^{t-1}$ *numbers.*

(d) The lower and the larger positive floating point numbers are, respectively, x_{min} *and* x_{max} *defined as*

$$x_{min} = \beta^{L-1}, \qquad x_{max} = \beta^U \cdot \left(1 - \beta^{-t} \right)$$

Proof. The first three and x_{min} are trivial. Consider x_{max}. We have

$$\begin{aligned} x_{max} &= \beta^U \cdot \sum_{k=1}^{t} (\beta - 1)\, \beta^{-k} = \beta^U \cdot (\beta - 1) \cdot \sum_{k=1}^{t} \left(\beta^{-1} \right)^k \\ &= \beta^U \cdot (\beta - 1) \cdot \left[\frac{1 - \left(\beta^{-1} \right)^{t+1}}{1 - \beta^{-1}} - 1 \right] \\ &= \beta^U \cdot (\beta - 1) \cdot \frac{1 - \beta^{-t-1} - 1 + \beta^{-1}}{1 - \beta^{-1}} \\ &\overset{(1)}{=} \beta^U \cdot (\beta - 1) \cdot \frac{1 - \beta^{-t}}{\beta - 1} \end{aligned}$$

where in (1) we have multiplied numerator and denominator by β. □

Example 1.7 *Let us explicitly write* $\mathbb{F}(10, 1, -1, 2)$. *It is* $\beta = 10$, $t = 1$, $L = -1$, $U = 2$. *Thus, for the positive floating point numbers, we have the 36 numbers shown in Table 1.1.*

$p = -1$	$p = 0$	$p = 1$	$p = 2$
$0.1 \cdot 10^{-1} = 0.01$	$0.1 \cdot 10^0 = 0.1$	$0.1 \cdot 10^1 = 1$	$0.1 \cdot 10^2 = 10$
$0.2 \cdot 10^{-1} = 0.02$	$0.2 \cdot 10^0 = 0.2$	$0.2 \cdot 10^1 = 2$	$0.2 \cdot 10^2 = 20$
$0.3 \cdot 10^{-1} = 0.03$	$0.3 \cdot 10^0 = 0.3$	$0.3 \cdot 10^1 = 3$	$0.3 \cdot 10^2 = 30$
$0.4 \cdot 10^{-1} = 0.04$	$0.4 \cdot 10^0 = 0.4$	$0.4 \cdot 10^1 = 4$	$0.4 \cdot 10^2 = 40$
$0.5 \cdot 10^{-1} = 0.05$	$0.5 \cdot 10^0 = 0.5$	$0.5 \cdot 10^1 = 5$	$0.5 \cdot 10^2 = 50$
$0.6 \cdot 10^{-1} = 0.06$	$0.6 \cdot 10^0 = 0.6$	$0.6 \cdot 10^1 = 6$	$0.6 \cdot 10^2 = 60$
$0.7 \cdot 10^{-1} = 0.07$	$0.7 \cdot 10^0 = 0.7$	$0.7 \cdot 10^1 = 7$	$0.7 \cdot 10^2 = 70$
$0.8 \cdot 10^{-1} = 0.08$	$0.8 \cdot 10^0 = 0.8$	$0.8 \cdot 10^1 = 8$	$0.8 \cdot 10^2 = 80$
$0.9 \cdot 10^{-1} = 0.09$	$0.9 \cdot 10^0 = 0.9$	$0.9 \cdot 10^1 = 9$	$0.9 \cdot 10^2 = 90$

Table 1.1: The numbers in $\mathbb{F}(10, 1, -1, 2)$.

Considering also the negative ones and the zero we have

$$1 + 2 \cdot (U - L + 1) \cdot (\beta - 1) \cdot \beta^{t-1} = 1 + 2 \cdot [\, 2 - (-1) + 1 \,] \cdot (10 - 1) \cdot 10^{1-1} = 73$$

floating point numbers. Also, we have

$$x_{min} = 10^{L-1} = 10^{-1-1} = 0.01, \qquad x_{max} = 10^{U} \cdot (1 - 10^{-t}) = 10^{2} \cdot (1 - 10^{-1}) = 90$$

It is interesting to note that the difference between two consecutive numbers is not a constant. It is if they have the same value of p. □

1.2.1 Representation of real numbers in $\mathbb{F}$

Let us consider, for semplicity, only positive numers. The positive real number x may be written, using the base β, as

$$x = \beta^p \sum_{k=1}^{+\infty} a_k \, \beta^{-k}$$

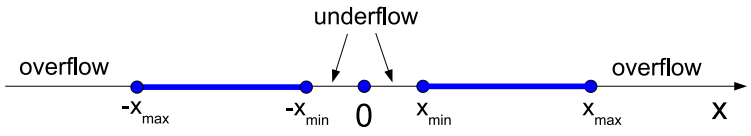
for some integer p.When this number has to be represented using a floating point number in the set $\mathbb{F}(\beta, t, L, U)$, one of the following cases may occur.

- If $p < L$ the number is less then the smallest representable floating point number. An *underflow* occurs. In this case, the real number x is usually represented by the floating point number 0. Moreover, since in this way we completely loose the numebr x, some message is also given to us but the computation proceeds.
- If $L \leq p \leq U$ the number x can be represented in $\mathbb{F}$. There are, however, two cases
  - $a_k = 0$ for $k \geq t$. In this case, $x \in \mathbb{F}$ and so it can be exactly represented in $\mathbb{F}$.
  - $a_k \neq 0$ for at least one $k > t$. The number $x \notin \mathbb{F}$. In this case, the better we can do is to represent the number x with the floating point number fl(x) (read: "the float of x") defined as

$$\mathrm{fl}(x) = \begin{cases} \beta^p \ \sum_{k=1}^{t} a_k \, \beta^{-k} & \text{if} \quad a_t \in \{0, \ldots, \frac{\beta}{2} - 1\} \\ \\ \beta^p \ \sum_{k=1}^{t} a_k \, \beta^{-k} + \beta^{-t} & \text{if} \quad a_t \in \{\frac{\beta}{2}, \ldots, \beta - 1\} \end{cases}$$

  The representation of $\mathrm{fl}(x)$ instead of x leads to an error called *rounding error*. We will see more on this in a moment.
- $p > U$. The real number x is beyond the upper limit of the floating point system $\mathbb{F}$. An *overflow* occurs and, usually, the computation stops with an error message.

Taking into account also negative numbers, the floating point numbers $\mathbb{F}$ are inside the blue intervals of the following figure.



Remark 1.2 (denormalized numbers) *Consider $\mathbb{F}(\beta, t, L, U)$. We have said that the first figure a_1 of each floating point number has to fulfill the condition $a_1 > 0$ in order to avoid multiple representations.*

However if, and only if, $p = L$ it is usual to remove this condition allowing a_1 to be equal to zero. The real numbers obtained for $p = L$, $a_1 = 0$ and $a_k \neq 0$ for at least one $k = 2, \ldots, t$, are considered as new floating point numbers of $\mathbb{F}$. We call them denormalized floating point numbers. The other numbers of $\mathbb{F}$ for which $a_1 > 0$ (regardless of L) are called normalized floating point numbers.

Let's turn back on the error that we have when rounding the real number x into the floating point number $\mathrm{fl}(x)$. The following theorem holds.

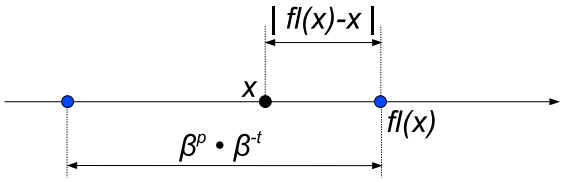
Theorem 1.2 *Let*

$$x = \beta^p \sum_{k=1}^{+\infty} a_k \beta^{-k}$$

be a positive, real number with $a_1 \neq 0$. Then, assuming that there is non overflow, using the floating point system $\mathbb{F}(\beta, t, L, U)$, the following inequality holds

$$\left| \frac{fl(x) - x}{x} \right| \leq \frac{\beta^{1-t}}{2} \tag{1.2}$$

Proof. Clearly, if $x \in \mathbb{F}$, we have $\mathrm{fl}(x) = x$ and thus $\mid \mathrm{fl}(x) - x \mid = 0$. So, the inequality is trivially fulfilled. Otherwise, the number x lies between two consecutive floating point numbers (blue circles in the next figure). The representative of x in $\mathbb{F}$ is the nearest to x of this two floating point numbers. Noting that two consecutive floating point numbers with the same p differ one from the other by $\beta^p \cdot \beta^{-t} = \beta^{p-t}$, we have $\mid \mathrm{fl}(x) - x \mid \leq \beta^{p-t}/2$.



Thus, recalling that $x > 0$ and so $|x| = x$, we have

$$\left| \frac{\mathrm{fl}(x) - x}{x} \right| = \frac{\mid \mathrm{fl}(x) - x \mid}{x} \overset{(1)}{\leq} \frac{\mid \mathrm{fl}(x) - x \mid}{\beta^p \cdot \beta^{-1}} \leq \frac{\frac{1}{2}\, \beta^p \cdot \beta^{-t}}{\beta^p \cdot \beta^{-1}} = \frac{\beta^{1-t}}{2}$$

where inequality (1) holds since (recall that $a_k \in \{0, 1, \cdots, \beta - 1\}$ and $a_1 > 0$)

$$\begin{aligned} x = \beta^p \sum_{k=1}^{+\infty} a_k \beta^{-k} &= \beta^p \left(a_1 \cdot \beta^{-1} + a_2 \cdot \beta^{-2} + a_3 \cdot \beta^{-3} + \cdots \right) \\ &\geq \beta^p \left(1 \cdot \beta^{-1} + 0 \cdot \beta^{-2} + 0 \cdot \beta^{-3} + \cdots \right) \\ &= \beta^p \cdot \beta^{-1} \end{aligned}$$

This ends the proof. □

Definition 1.1 (machine precision) *Let $\mathbb{F}(\beta, t, L, U)$ be a floating point system. The number eps defined as*

$$eps = \frac{\beta^{1-t}}{2} \tag{1.3}$$

is called the machine precision of the floating point system $\mathbb{F}$.

Note that the number 1 belongs to any floating point system since

$$1 = \beta^1 \cdot \beta^{-1} = \beta^1 \cdot \sum_{k=1}^{t} a_k \beta^{-k}$$

with $a_1 = 1$ and $a_k = 0$, $k = 2, \ldots, t$. The next floating point number is

$$x_+ = \beta^1 \cdot \left(1 \cdot \beta^{-1} + 0 \cdot \beta^{-2} + \cdots + 0 \cdot \beta^{-t+1} + 1 \cdot \beta^{-t} \right) = \beta^1 \cdot \left(1 \cdot \beta^{-1} + 1 \cdot \beta^{-t} \right)$$

which differs from 1 by $x_+ - 1 = \beta^{1-t} = 2\,\mathrm{eps}$. So, the real number $x = 1 + \mathrm{eps}$ lies exactly in the middle between 1 and x_+; thus, it is rounded to $\mathrm{fl}(1 + \mathrm{eps}) = x_+$. Note also that each

real number x satisfying $1 < x < 1 + \text{eps}$ is rounded to the floating number 1. So, eps is the smallest number that we have to add to the floating point number 1 in order to have a floating point number greater than 1.

From equation (1.2), for some $\bar{\epsilon}$ with $0 \leq \bar{\epsilon} \leq \text{eps}$, we can write

$$\left| \frac{\text{fl}(x) - x}{x} \right| = \bar{\epsilon} \quad \Leftrightarrow \quad \frac{\text{fl}(x) - x}{x} = \pm\bar{\epsilon} \quad \Leftrightarrow \quad \text{fl}(x) = x \pm \bar{\epsilon}\, x = x(1 \pm \bar{\epsilon})$$

Taking into account the sign, i.e. assuming $\epsilon \in [-\text{eps}, \text{eps}]$, $|\epsilon| = \bar{\epsilon}$, we have the following equation

$$\text{fl}(x) = x\,(1 + \epsilon), \quad \epsilon \in [-\text{eps}, \text{eps}] \tag{1.4}$$

This equation says that $\text{fl}(x)$ can be seen as a small perturbation of x.

1.2.2 Floating point arithmetic

The main aim of this section is to show some of the problems which arises when working with floating point numbers $\mathbb{F}(\beta, t, L, U)$. We define the four arithmetic operations between $x, y \in \mathbb{F}$ as

$$\begin{array}{ll} \text{(a)} & x \oplus y = \text{fl}(x + y) \\ \text{(b)} & x \ominus y = \text{fl}(x - y) \\ \text{(c)} & x \otimes y = \text{fl}(x \times y) \\ \text{(d)} & x \oslash y = \text{fl}(x - y) \end{array}$$

So, each floating point operation require two steps: (i) execute the operation in $\mathbb{R}$; (ii) represent the obtained result in $\mathbb{F}$. As an example, consider $x \oplus y$.

(i) We first compute $x + y$ as an operation between the real numbers x and y.

(ii) We represent the result $x + y$ in $\mathbb{F}$ (considering, if the case, over and under flow).

Example 1.8 *Consider* $\mathbb{F}(10, 1, -1, 2)$ *and the three floating point numbers* $x = 0.1$, $y = 0.2$, $z = 0.7$. *Then, we have*

$$x \oplus y = \mathit{fl}(x + y) = \mathit{fl}(0.1 + 0.2) = \mathit{fl}(0.3) = 0.3$$

since $0.3 \in \mathbb{F}$. *Also, we have*

$$x \oslash z = \mathit{fl}(x/y) = \mathit{fl}(0.1/0.7) = \mathit{fl}(\ 0.14285714285714\cdots\ ) = 0.1$$

Finally, $1 \oslash (x \otimes x)$ *gives an overflow; first, we compute*

$$x \otimes x = \mathit{fl}(x \times x) = \mathit{fl}(0.1 \times 0.1) = \mathit{fl}(0.01) = 0.01$$

next, we compute $1 \oslash 0.01 = \mathit{fl}(1/0.01) = \mathit{fl}(100)$*; since* 100 *is greater then the maximum representable floating point number in* $\mathbb{F}$*, an overflow is produced.* □

It is interesting to point out that most of the common properties of the operations with real numbers are not still valid in $\mathbb{F}$. For example, given positive floating point numbers x and y, we may have

$$x \oplus y = x$$

if y is less then half of the distance between x and the next floating point number x_+. Indeed, let

$$x = \beta^p \sum_{k=1}^{t} a_k \beta^{-k}$$

Then, we have

$$x_+ = \beta^p \left(\sum_{k=1}^{t} a_k \beta^{-k} + \beta^{-t} \right) = x + \beta^p \cdot \beta^{-t}$$

and so we get $x_+ - x = \beta^{p-t} = 2\,\beta^{p-1} \cdot \text{eps}$. Then, we have

$$x \oplus y = \begin{cases} x & \text{if} \quad y < \beta^{p-1} \cdot \text{eps} \\ \\ x_+ & \text{if} \quad \beta^{p-1} \cdot \text{eps} \leq y \leq \beta^{p-1} \cdot \text{eps} \end{cases}$$

Also, associative and distributive laws may, even not necessarily, fail. Consider the following examples.

Example 1.9 *Consider again* $\mathbb{F}(10, 1, -1, 2)$ *and the three floating point numbers* $x = 0.1$, $y = 2$, $z = 80$. *Using exact arithmetic, it is known that* $(x \times y) \times z = x \times (y \times z) = 16$. *Using floating point arithmetic, we have*

$$x \otimes y = fl(x \times y) = fl(0.1 \times 2) = fl(0.2) = 0.2$$

and

$$(x \otimes y) \otimes z = fl(0.2 \times 80) = fl(16) = 20$$

This is the best result we can have with our floating point system since $fl(x \times y \times z) = fl(16) = 20$. *On the other hand,* $x \otimes (y \otimes z)$ *returns an overflow since* $y \times z = 160$ *which is greater then the maximum representable number in* $\mathbb{F}$. *So, the executing order of the operations may be important.*

Example 1.10 (Smearing effect) *Consider the floating point system* $\mathbb{F}(10, 3, -2, 2)$ *and the three floating point numbers* $x = 0.123$, $y = 45.6$, $z = -45.5$. *The computation of* $x + y + z = 0.223$ *may be done in two ways.*

(i) We compute $w = x \oplus y$ *and then* $w \oplus z$. *We have*

$$w = x \oplus y \quad = \quad fl(0.123 + 45.6) = fl(45.723) = 45.7$$

and

$$w \oplus y \quad = \quad fl(45.7 - 45.5) = 0.200$$

(ii) We compute $u = y \oplus z$ *and then* $x \oplus u$. *We have*

$$u = y \oplus z = fl(45.6 - 45.5) = 0.100$$

and

$$x \oplus u = fl(0.123 + 0.100) = 0.223$$

So, in the first case the absolute value of the error is $0.10 = 10\%$ *whereas in the second case we have no error. Looking closely to the example, we may see that in the first case we add* x *and* y *first. These two numbers are quite different in size and so we loose some of the digits of* x *when performing the sum. The next sum* $w \oplus z$ *is done correctly but using the data* w *which has already a great error. So, the two ending zeros of the final result* 0.200 *are not correct. This is not the case when we rearrange the computation as shown in the second case (ii). Here, when we compute* $u = 0.100$ *the two ending zeros are correct and so* u *is correct to three decimal places. As a consequence, no error appears when we compute* $x \oplus u$ *and the final result is correct.*

Example 1.11 *Let* $f(x) = \sqrt{1+x} - \sqrt{x}$. *Consider the computation of* $f(49)$. *In exact arithmetic, we have* $f(49) = \sqrt{50} - \sqrt{49} = 0.07106781186548...$. *Using* $\mathbb{F}(10, 1, -1, 2)$ *and assuming that* $\sqrt{\xi}$ *is computed in a floating point system as* $fl(\sqrt{\xi})$, *we obtain*

$$fl(\sqrt{50}) = fl(7.07106781186548) = 7 \qquad \textit{and} \qquad fl(\sqrt{49}) = fl(7) = 7.$$

and so $f(49) = 7 - 7 = 0$ *with an absolute value of the relative error*

$$\frac{|fl(f(49)) - f(49)|}{|f(49)|} |0.07106781186548... - 0| / 0.07106781186548... = 1 = 100\%$$

A different way to evaluate $f(49)$ leads to a better result. Noting that

$$f(x) = \sqrt{1+x} - \sqrt{x} = \frac{\left(\sqrt{1+x} - \sqrt{x}\right) \cdot \left(\sqrt{1+x} + \sqrt{x}\right)}{\sqrt{1+x} + \sqrt{x}} = \frac{1}{\sqrt{1+x} + \sqrt{x}}$$

we obtain in $\mathbb{F}(10, 1, -1, 2)$

$$\begin{aligned} f(49) &= fl\left(\frac{1}{fl\left(\, fl\left(\sqrt{50}\right) + fl\left(\sqrt{49}\right)\right)}\right) = fl\left(\frac{1}{fl(\, 7+7)}\right) = fl\left(\frac{1}{fl(\, 14)}\right) \\ &= fl\left(\frac{1}{10}\right) = fl(0.1) = 0.1 \end{aligned}$$

with an absolute value of the error $|0.07106781186548... - 0.1|/0.07106781186548... \approx 0.41 = 41\%$. A bit of comment. The result is clearly better in the second way even if it is still unsatisfactory due to the high error. However, this simple example shows that sometimes the result of the evaluation of a function can be improved simply revritting the function in a different way. Things become even more interesting if we consider that arithmetic operations are usually done using one more figure (that is, $t+1$ figures); in this way, the result of the previous evaluation becomes

$$f(49) = fl\left(\frac{1}{fl\left(\sqrt{50}\right) + fl\left(\sqrt{49}\right)}\right) = fl\left(\frac{1}{7+7}\right) = fl(0.07142857142857...) = 0.07$$

which is the best possible result using $\mathbb{F}(10, 1, -1, 2)$.

As we can see from the previous examples, operations on floating point numbers is not an easy task. Things become more and more complicated when we have massive algorithms where thousand to millions or more of floating point operations are done. In such cases, it is useful the following (informal)

Definition 1.2 (Stability of an algorithm) *An algorithm is <u>stable</u> if and only if small errors in the data and in the floating point operations does not grow up too much. Otherwise, the algorithm is <u>instable</u>.*

Let's turn back on the computation, in some floating point system, of $f(x) = \sqrt{x+1} - \sqrt{x}$. We may see the formula as a simple algorithm where some operations are done in the input x in order to obtain the result $f(x)$. Since, as we have seen, the result is greatly affected by the rounding errors, we say that this formula is unstable. Otherwise, the other formula $f(x) = 1/(\sqrt{x+1} + \sqrt{x})$ is a stable since rounding errors that occur during the algorithm do not grow too much. So, for the same (well-conditioned) problem we may have both stable and unstable algorithms. Recall that, for a given problem, it is not always simple to find a stable one. This is the case of the next example.

Example 1.12 *Consider the computation of the positive integrals*

$$I_n = \frac{1}{e}\int_0^1 x^n e^x dx, \quad n \in \mathbb{N} = \{0, 1, 2, \cdots\}$$

It is easy to see that $I_0 = 1 - e^{-1} = 0.6321205588285577....$ Moreover, integrating by parts, we get the recursive relation

$$I_n = \frac{1}{e}\left\{ \left[\, x^n e^x \,\right]_0^1 - \int_0^1 n x^{n-1} e^x \right\} = 1 - n I_{n-1}.$$

Finally, it is easy to check that $\lim_{n\to+\infty} I_n = 0$ since we have (recall that $1 \le e^x \le e$, $x \in [0, 1]$)

$$0 \le \frac{1}{e}\int_0^1 x^n e^x dx \le \frac{1}{e} \cdot e \int_0^1 x^n dx = \frac{1}{n+1}$$

Now, consider the computation of I_n for some given $n > 1$ with the following two algorithms:

ALGORITHM 1

set $I_0 = 0.6321205588285577$
FOR k=1:n
$I_k = 1 - kI_{k-1}$
END

ALGORITHM 2

choose some N *with* $N \gg n$
set $I_N = 0$
FOR k=N:-1:n
$I_{k-1} = (1 - I_k)/k$
END

The first algorithm is unstable whereas the second one is stable. This can be easily seen in the following figure

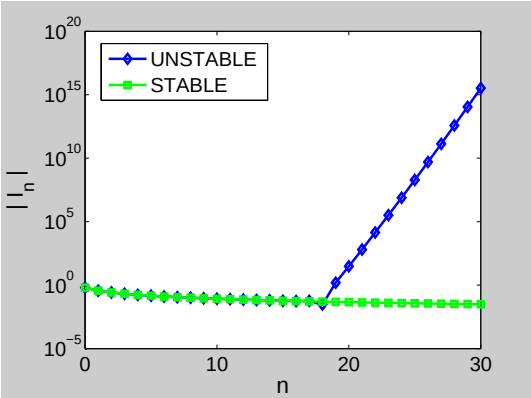


Figure 1.3: Computation of $I_n = \int_0^1 x^n e^x dx$ *for different values of n using a stable and an unstable algorithm. For the stable algorithm we have used* $N = 2n$.

We can explain the different behaviour of the two algorithms. Consider the first one. Due to the rounding process, the starting integral I_0 *that is actually stored inside the computer is not* I_0 *but, instead,* $\hat{I}_0 := fl(I_0)$. *We know that* $fl(I_0) = I_0(1 + \epsilon)$ *with* $\epsilon \in (-eps, eps)$. *Assume, for simplicity, that all the other operations are exact. So, we have,*

$$\hat{I}_1 := 1 - I_0(1 + \epsilon) = (1 - I_0) - \epsilon I_0 = I_1 - \epsilon I_0$$

$$\hat{I}_2 := 1 - 2 \cdot \hat{I}_1 = 1 - 2 \cdot (I_1 - \epsilon I_0) = (1 - 2I_1) + 2 \cdot 1 \cdot \epsilon I_0 = I_2 + 2! \cdot \epsilon I_0$$

$$\hat{I}_3 := 1 - 3 \cdot \hat{I}_2 = 1 - 3 \cdot (I_2 + 2! \cdot \epsilon I_0) = (1 - 3I_2) - 3 \cdot 2! \cdot \epsilon I_0 = I_3 - 3! \cdot \epsilon I_0$$

Proceeding in this way, we find

$$\hat{I}_n = I_n + (-1)^n n! \cdot \epsilon I_0$$

and so the error is $| \hat{I}_n - I_n | = n! \cdot \epsilon I_0$ *which we may expect to grow as a function of n.*

Consider now algorithm 2. Suppose that the only error is inside I_N *and that all operations are done exactly. Thus, we have* $\hat{I}_N = I_N(1 + \epsilon)$ *for some* $\epsilon \in (-eps, eps)$. *We get*

$$\hat{I}_{N-1} = \frac{1 - I_N(1 + \epsilon)}{N} = \frac{1 - I_N}{N} - \frac{I_N \cdot \epsilon}{N} = I_{N-1} - \frac{I_N \cdot \epsilon}{N}$$

$$\hat{I}_{N-2} = \frac{1 - \left(I_{N-1} - \frac{I_N \cdot \epsilon}{N} \right)}{N - 1} = \frac{1 - I_{N-1}}{N - 1} - \frac{I_N \cdot \epsilon}{N \cdot (N - 1)} = I_{N-2} - \frac{I_N \cdot \epsilon}{N \cdot (N - 1)}$$

In this way we find

$$\hat{I}_n = \hat{I}_{N-(N-n)} = I_n - \frac{I_N \cdot \epsilon}{N \cdot (N-1) \cdots (n+1)}$$

So, the initial error ϵI_N is shut down by the product of the denominator provided that $N \cdot (N-1) \cdots n$ is large.

1.2.3 The floating point system on a computer: IEEE 754 standard

The storing of informations on a computers is based on electronic devices which have only two stable states.

Thus, it is quite natural to use the base $\beta = 2$ for the floating point system. We show the main idea of the currently adopted floating point system, called IEEE 754 (Institute of Electrical and Electronics Engineers). Typically, there are two different values for t:

- $t = 24$ for the single precision;
- $t = 53$ for the double precision.

The next figure shows a single precision pattern. As we can see, there are 4 Bytes (one Byte is a set of 8 consecutive bit) for storing floating point numbers using single precision. The red one is used for the sign, the green ones for the exponent end the blue ones for the mantissa.

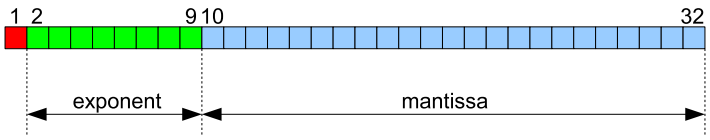


Figure 1.4: A single precision pattern for the representation of the floating point numbers inside a computer. Note that the number of bits for the mantissa is 23 since the first bit is always equal to 1 and thus does not need to be stored (hidden bit).

In real floating point systems, as one can image, thing are more complex that this since some sequence of bits are used to represent other things rather than numbers. Just to give an idea, the result of a division 0/0 return a NaN (NotaNumber) in Matlab. A NaN has its own representation [1].

Still, the main ideas are the one presented here.

1.3 Exercises

A difficult exercise has a star near its number. Some other exercises can be found in the simulations.

1. A problem has input $x = 1$. The corresponding output is $y = 10$. When $x = 1 + 10^{-3}$ the corresponding output becomes $y = 100$. Is the problem well or ill conditioned? Give an estimation of the condition number K. [Answer: ill-conditioned with $K \approx 9000$.]

2. Let $\mathbb{F}(10, 3, -2, 3)$. Considering only positive normalized numbers, compute x_{min}, x_{max} and the machine precision eps.

3. Let $\mathbb{F}(\beta, t, L, U)$. Is it true that the product of $x \in \mathbb{F}$ and $y \in \mathbb{F}$ is always an element of $\mathbb{F}$? Does the answer change if we avoid under and over flow? [Hint: maybe it is useful to consider $\mathbb{F}(10, 1, -1, 2)$.]

[1] If one is interested in, just open Octave (not Matlab), type "format bit" in the command Window, return, 0/0 and return; the following representation of the 0/0 operations appears on the screen and looks as eleven 1 followed by a long number of 0.

4. Prove that $\mathbb{F}(\beta, t, L, U)$ has $2\,\beta^{t-1} - 1$ denormalized numbers. The smallest positive one is $\beta^L\,\beta^{-t} = \beta^{L-t}$.

5. Consider $\mathbb{F}(10, 1, -1, 2)$. Is it possible to compute $x \times (y + z)$ without overflow if we have $x = 0.1$, $y = z = 80$? If yes, how can the operation be done and what result does it produce? [Answer: $x \otimes (y \oplus z)$ gives an overflow. Write $x \times (y + z) = x \times y + x \times z$ and compute $x \otimes y$, $x \otimes z$ and then $(\ x \otimes y\ ) \ \oplus\ (\ x \otimes z\ )$. We obtain 20.]

6. We have to evaluate the function $f(x) = 1 - \cos(x)$ for $x \approx 0$. Rewrite the function in order to obtain a good evaluation in $\mathbb{F}(10, 1, -1, 2)$. [Hint: it is $f(x) = 2\sin^2(x/2)$.]

7. ($\star$) We have to evaluate the function

$$f(x) = \frac{1}{x} - \frac{1}{\sin(x)}$$

for $x \approx 0$. Rewrite the function in order to obtain a good evaluation in $\mathbb{F}(10, 1, -1, 2)$. [Hint: common denominator and then Mac-Laurin expansion for $\sin(x) - x$.]

8. ($\star$) Consider the computation of e^x both for positive and negative x using Taylor expansion as

$$e^x \approx \sum_{k=0}^{n} \frac{x^k}{k!}$$

Using Matlab or Octave, write a program to evaluate the previous sum for some given values of n and x (both positive and negative near zero and far away in both directions). Give a possible explanation of the results.

Chapter 2

Roots of Equations

This chapter is devoted to the determination of roots of equations. So we begin with the following definition.

Definition 2.1 *Let f be a function of the (real or complex) variable x. The roots of the equation*

$$f(x) = 0$$

are the numbers ξ for which $f(\xi) = 0$. Each root of the equation $f(x) = 0$ is said to be a zero of the function f.

The roots of the equation $f(x) = 0$ are the intersections of the graph $y = f(x)$ with the real axis, i.e., the line $y = 0$. In the same manner, the roots of the equation $f(x) = g(x)$ are the abscissas of the intersection points of the two graphs $y = f(x)$ and $y = g(x)$.

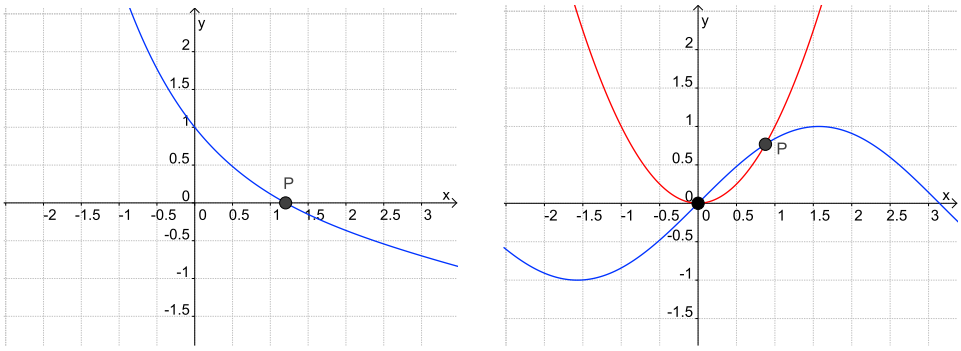


Figure 2.1: Left: The equation $e^{-x} - 0.25 \cdot x = 0$ has only one root $\xi \in (1,\ 1.5)$ since the corresponding graph $y = e^{-x} - 0.25 \cdot x$ intersect the x axis only in one point $P = (\xi, 0)$. Right: the equation $\sin(x) - x^2 = 0$ has two roots: $\xi_1 = 0$ and $\xi_2 \in (0.5,\ 1)$ since the graphs $y = \sin(x)$ and $y = x^2$ have two intersection points $O = (0, 0)$ and $P = (\xi_2, f(\xi_2))$.

The computation of real roots of the equation $f(x) = 0$ follows two main steps

(a) **roots separation** : for each root ξ_k, we find an interval $[a_k,\ b_k]$ such that $\xi_k \in [a_k,\ b_k]$ and no one of the other roots belongs to $[a_k,\ b_k]$.

(b) **roots approximation** : we approximate some, or even all, of the roots.

The first step may be done sketching the graph of the function f. It is also useful the following theorem.

Theorem 2.1 (zeros of a continuous function) *Let f be a continuous function (at least) in the interval $[a,\ b]$ with $f(a) \cdot f(b) < 0$. Then, f has almost one zero in the interval $[a,\ b]$. Furthermore, if the function f is strictly monotone in $[a,\ b]$, then the zero is unique.*

Example 2.1 *Consider the equation*

$$x \log(x) - 1 = 0$$

The function $f(x) = x \log(x) - 1$ *is continuous and defined for* $x > 0$*. Its graph is shown in Figure 2.2.*

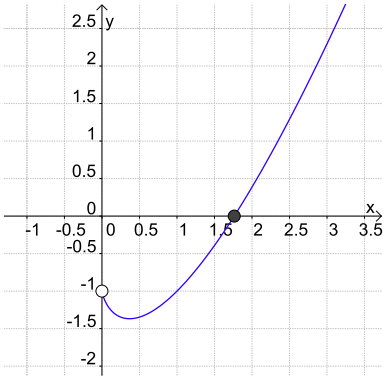


Figure 2.2: The graph of the function $f(x) = x \log(x) - 1$ shows a zero $\xi \in [1,\ 2]$.

We have

$$\begin{aligned} f(1) &= 1 \cdot \log(1) - 1 = 1 \cdot 0 - 1 = -1 < 0 \\ f(e) &= e \cdot \log(e) - 1 = e \cdot 1 - 1 > 0 \end{aligned}$$

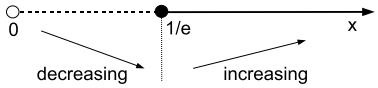
Thus, $f(1) \cdot f(e) < 0$ *and theorem 2.1 guarantees that there is almost one root in the interval* $[1,\ e]$*. This root is the only one in* $[1, e]$ *because* f *is strictly increasing in this interval since*

$$f'(x) = \log(x) + 1 > 0 \quad \forall\, x \in [1,\ e].$$

Indeed, this equation has no roots other than ξ*. To prove this, note that*

$$\lim_{x \to 0^+} f(x) = -1, \qquad \lim_{x \to +\infty} f(x) = +\infty$$

and the sign of the first derivative f' *is*



So, starting from -1*,* f *decreases in* $(0,\ 1/e]$*, reaches a minimum at* $x_m = 1/e$ *where* $f(1/e) = -1/e - 1 < 0$ *and then increases going to* $+\infty$ *as* $x \to +\infty$*. Thus, there is only one root* $\xi > x_m = 1/e$*.* □

2.1 Convergent sequences

A sequence x_k is a convergent sequence if exists $\xi \in \mathbb{R}$ such that

$$\lim_{k \to +\infty} x_k = \xi.$$

Definition 2.2 *Let* x_k *be a sequence that converges to* ξ*. The error* e_k *at step* k *is*

$$e_k := \xi - x_k.$$

Definition 2.3 *Let x_k be a sequence that converges to ξ. If there are two positive constants p and c such that*

$$\lim_{k\to+\infty} \frac{|e_{k+1}|}{|e_k|^p} = c \tag{2.1}$$

we say that the sequence has order of convergence p with asymptotic error constant c. Moreover, we say that the convergence is

- *linear if $p = 1$. In this case, in order to have convergence, we must have $c < 1$.*
- *superlinear if $p > 1$. More specifically, if $p = 2$, we say that the convergence is quadratic.*

It is usual to plot $\log_{10}(|e_k|)$ as a function of k. The behaviour of this semilogarithmic plot is, at least for large values of the iteration indexes k, linear if $p = 1$ and parabolic if $p > 1$.

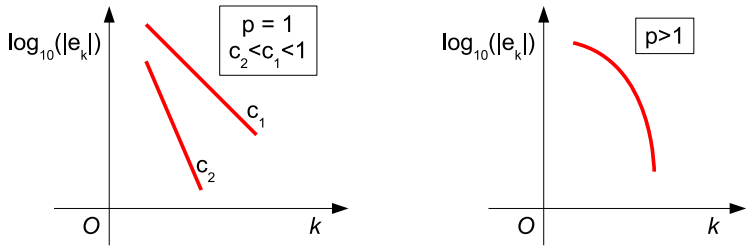


Figure 2.3: Different behaviour of $\log_{10}(|e_k|)$ versus the iteration number k for linear convergent sequences (left) and more than linear (right).

Let's prove it just for $p = 1$ since for $p > 1$ it is more difficult. We have

$$|e_k| \approx c \cdot |e_{k-1}| \approx c\,(c \cdot |e_{k-2}|) = c^2 \cdot |e_{k-2}| \approx \cdots \approx c^k \cdot |e_0|$$

and taking logarithm of both sides

$$\log_{10}(|e_k|) \approx k \log_{10}(c) + \log_{10}(|e_0|)$$

This is the equation of a line with negative slope since $\log_{10}(c) < 0$ due to $0 < c < 1$.
It is also very interesting to see, using numbers, the different behaviour of the errors for two convergent sequences of order $p = 1$ and $p = 2$. We assume, for simplicity, for both sequences, $c = 0.5$ and $|e_0| = 1$. Roughly, using $|e_{k+1}| \approx c \cdot |e_k|$ for the linear case and $|e_{k+1}| \approx c \cdot |e_k|^2$ for the quadratic case, we have

k	0	1	2	3	4	5
$p = 1$	1	0.5	0.25	0.125	0.0625	0.0313
$p = 2$	1	0.5	0.125	$7.8 \cdot 10^{-3}$	$3.1 \cdot 10^{-5}$	$4.7 \cdot 10^{-10}$

Table 2.1: Behaviour of $|e_k|$ for different values of k for linear and more than linear convergent sequences.

Thus, the error in the quadratic case drops very quickly; moreover, looking only to the exponents, assuming $|e_k| \approx 10^{-n}$ than $|e_{k+1}| \approx 10^{-2n}$. That is, we double the number of correct digits per step. Completely different is the behaviour of the error in the linear case where the error drops slowly (in this case, since $c = 0.5$, it halves at each step).

Remark 2.1 *For a linear convergent sequence it is possible to find an approximation of the index k for which it is $|e_k| < \epsilon \cdot |e_0|$ for some given $\epsilon > 0$. Indeed, recalling that $|e_k| \approx c^k \cdot |e_0|$, we have*

$$|e_k| < \epsilon \cdot |e_0| \quad \Leftrightarrow \quad c^k \cdot |e_0| \approx \epsilon \cdot |e_0| \quad \Leftrightarrow \quad k \approx \frac{\log_{10}(\epsilon)}{\log_{10}(c)}$$

2.2 Bisection method

Let ξ be the unique zero in the interval $[a,\, b]$ of the function f which we assume continuous at least in $[a, b]$. Assume $\xi \neq a$ and $\xi \neq b$.

Starting from $I_0 := [a_0, b_0] = [a, b]$, the bisection method constructs a sequence of nested intervals $I_k = [a_k, b_k]$ containing the root:

$$I_0 \supset I_1 \supset I_2 \supset I_3 \supset \cdots \supset I_k \supset I_{k+1} \cdots \qquad \text{with} \quad \xi \in I_k \; \forall\, k$$

The k-th step, $k = 0, 1, \ldots$, of the bisection method is

---

1. compute $x_k = (a_k + b_k)/2$. Note that $x_k \in I_k$.
2. compute $f(x_k)$
3. choose <u>one</u> of the following cases

3.1. $f(x_k) = 0$, i.e., x_k is a root of f. Since $x_k \in [a, b]$ by construction and ξ is the unique root inside $[a, b]$, then it must be $\xi = x_k$. We have find the root and the iterative process stops.

3.2. $f(a_k) \cdot f(x_k) < 0$, i.e. $f(a_k)$ and $f(x_k)$ have opposite signs. Thus $\xi \in [a_k, x_k]$. So, we set $I_{k+1} = [a_{k+1},\ b_{k+1}] = [a_k,\ x_k]$. That is, $a_{k+1} = a_k$ and $b_{k+1} = x_k$ (see Figure 2.4 on the left).

3.3. $f(a_k) \cdot f(x_k) > 0$, i.e. $f(a_k)$ and $f(x_k)$ have the same signs. Thus $\xi \in [x_k, b_k]$. So, we set $I_{k+1} = [a_{k+1},\ b_{k+1}] = [x_k,\ b_k]$. That is, $a_{k+1} = x_k$ and $b_{k+1} = b_k$ (see Figure 2.4 on the right).

Let us denote by $|I_k| = b_k - a_k$ the length of the interval I_k. Then, in cases 3.2 and 3.3 we have

$$|I_{k+1}| = \frac{|I_k|}{2} \overset{(1)}{=} \frac{|I_0|}{2^{k+1}}$$

where (1) follows from mathematical induction. So, after k-th step is complete, the error $e_k = x_k - \xi$ satisfies the inequality

$$|e_k| \leq |I_{k+1}| = \frac{|I_0|}{2^{k+1}}$$

as it is clear from Figure 2.5

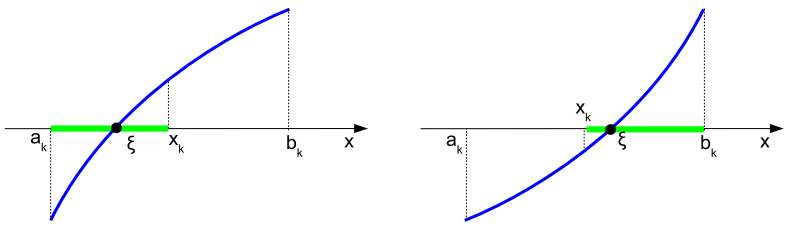


Figure 2.4: The single step of the bisection method. On the left: $I_{k+1} = [a_k,\ x_k]$ since $f(a_k) \cdot f(x_k) < 0$. Right: $I_{k+1} = [x_k,\ b_k]$ since $f(x_k) \cdot f(b_k) < 0$.

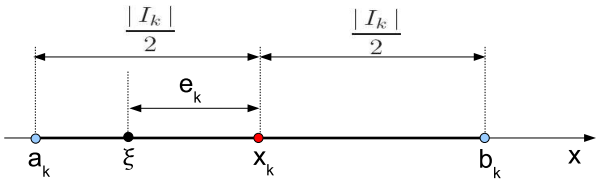


Figure 2.5: The error in the k-th step of the bisection method satisfies the inequality $\mid e_k \mid < \mid I_k \mid /2$ (we have assumed that the root ξ belongs to the first half of the interval $[a_k, b_k]$).

From the latter equation it is simple to compute the number of iterations of the bisection method that have to be performed in order to obtain $\mid e_k \mid < \varepsilon$ for some given $\varepsilon > 0$. Indeed, we have

$$\mid e_k \mid < \varepsilon \qquad \Leftrightarrow \qquad \frac{b-a}{2^{k+1}} < \varepsilon \qquad \Leftrightarrow \qquad 2^{k+1} > \frac{b-a}{\varepsilon}$$

and finally, taking the logarithm in the latter inequality, we get

$$\log\left(2^{k+1}\right) > \log\left(\frac{b-a}{\varepsilon}\right) \qquad \Leftrightarrow \qquad k > \frac{\log\left(\frac{b-a}{\varepsilon}\right)}{\log(2)} - 1$$

So, to obtain $\mid e_k \mid < \varepsilon$ it is necessary to perform at least $k_{\min}$ iterations with

$$k_{\min} = \left\lceil \frac{\log\left(\frac{b-a}{\varepsilon}\right)}{\log(2)} - 1 \right\rceil . \tag{2.2}$$

where $\lceil a \rceil$ is the smallest integer greater or equal to a.

Example 2.2 *The computation of the first positive zero of the equation*

$$x - \tan\left(\frac{x}{2}\right) = 0$$

within the tolerance $\varepsilon = 1.E-5 = 10^{-5}$ *and with starting interval* $[a, b] = [2.0, 2.5]$ *requires, at least,*

$$k_{min} = \left\lceil \frac{\log\left(\frac{2.5-2.0}{10^{-5}}\right)}{\log(2)} - 1 \right\rceil = \lceil 14.61 \rceil = 15 \text{ iterations}$$

which is exactly what it is shown in Figure 2.6 on the right. It is interesting to note, from the same figure, that the error does not decrease monotonically i.e., we can have $\mid e_{k+1} \mid > \mid e_k \mid$ *for some indexes* k.

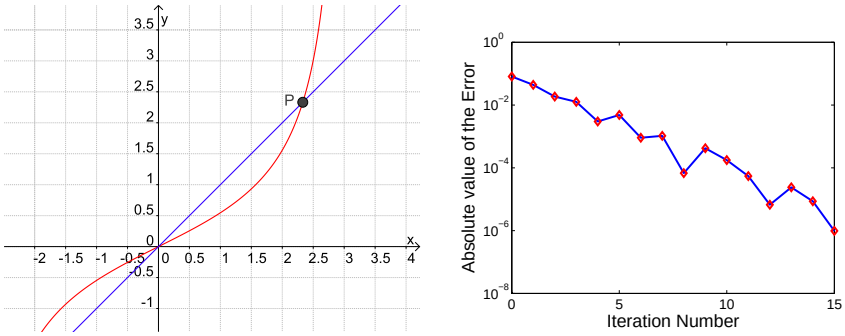


Figure 2.6: Approximation of the first positive zero of the equation $x - \tan(x/2) = 0$.

Exercise 2.1 (♡) *Compute the number of iterations needed to compute the zero $\xi \in [0,\ 1]$ within the tolerance $tol = 1E-9$ for each of the following equations*

$$(a)\ x^3 + x - 1 = 0 \qquad (b)\ \sin(x) - x + 0.1 = 0$$

Does the number of iterations changes for (a) and (b)? Why? (Answer: 29 iterations. No, the number of iterations is the same because it is related only to the amplitude of the initial interval $[a,\ b]$ containing the zero and to the required tolerance tol.)

Exercise 2.2 (♡) *Using the bisection method, compute the number of iterations needed to compute the zero $\xi \in [0,\ 1]$ within the tolerance $tol = 1E-3$ for the equation $1000x = 999$. Does the absolute value of the error $|e_k|$ monotonically decrease in this case or does it not? Explain.*

Exercise 2.3 (♡) *Consider the approximation, using the bisection method, of the positive zero $\xi = 1$ of the equation $f(x) = 0$ where $f(x) = x^2 - 1$. Assume the starting interval $I_0 = [a_0,\ b_0] = [0,\ 1.5]$.*

(a) Compute x_0, x_1 and the absolute value of the corresponding errors e_0 and e_1. Which one is larger? Sketch a graph of f near ξ and use it to explain the behavior or the two errors.

(b) Can we apply the bisection method to compute the positive root of the equation $f^2(x) = 0$ using the same starting interval as before? Explain.

Exercise 2.4 (♡) *How many iterations does the bisection method need to find the root $\xi = 0$ of $x^3 = 0$ within the tolerance $tol = 10^{-9}$ starting from the interval $[a_0, b_0] = [-1, 3]$?*

Solution. The bisection method works as follows.

1. First step ($k = 0$). We have $[a_0, b_0] = [-1, 3]$ and thus $x_0 = (a_0 + b_0)/2 = 1$. Since the root $\xi \in [a_0, x_0]$ we set $a_1 = a_0 = -1$ and $b_1 = x_0 = 1$.
2. Second step (k=1). We have $[a_1, b_1] = [-1, 1]$ and thus $x_1 = (a_1 + b_1)/2 = 0$. Thus, we have find the root and the algorithm halts.

Exercise 2.5 (⋆) *Compute the number of iterations k needed to reduce the absolute value of the initial error e_0 to $|e_k| = 10^{-k}|e_0|$ where k is some given positive integer. How many iterations are needed to gain an extra decimal digit, i.e. to reduce the error to $|e_k| = 10^{-k-1}|e_0|$?*

2.3 Fixed point iterations

Let us start with some theoretical background.

Definition 2.4 *The function $\phi(x),\ x \in [a, b]$ has the fixed point $\alpha \in [a, b]$ if $\alpha = \phi(\alpha)$.*

So, fixed points of the function ϕ are, if any, the roots of the equation $x = \phi(x)$. Graphically, they are abscissas of the intersection points of the graphs $y = x$ and $y = \phi(x)$.

Example 2.3 *The function $\phi(x) = x^2 + 1$ does not have any fixed point since the equation $x = x^2 + 1$ has no real roots.*
The function $\phi(x) = x^2$ has two fixed points since the equation $x = x^2$ has roots $\alpha_1 = -1$ and $\alpha_2 = 0$.

Existence and uniqueness of the fixed point are stated by the oncoming theorem.

Theorem 2.2 (Existence and uniqueness of fixed points) *Let ϕ be a continuous function on the interval $[a,\ b]$ with $\phi([a,\ b]) \subseteq [a,\ b]$. Then ϕ has at least one fixed point in $[a,\ b]$. Moreover, the fixed point is unique if ϕ is differentiable on $(a,\ b)$ and fulfills*

$$|\ \phi'(x)\ | \leq k < 1 \quad \forall x \in (a, b).$$

Proof. Let us prove the existence of at least one fixed point in $[a, b]$. First note that, since $\phi([a, b]) \subseteq [a, b]$, we have $\phi(a) \geq a$ and $\phi(b) \leq b$. Accordingly to this, we split the proof in two cases.

(a) If $a = \phi(a)$ or $b = \phi(b)$ then $\alpha = a$ or $\alpha = b$ is a fixed point for ϕ.

(b) If, otherwise, $\phi(a) > a$ and $\phi(b) < b$, let introduce the function $g(x) := \phi(x) - x$. This function is continuous since it is the difference of two continuous functions and satisfies

$$g(a) = \phi(a) - a > 0 \qquad g(b) = \phi(b) - b < 0$$

The theorem of the zeros of a continuous function assures that there exists almost an $\alpha \in [a, b]$ such that

$$g(\alpha) = 0 \quad \Leftrightarrow \quad \phi(\alpha) - \alpha = 0 \quad \Leftrightarrow \quad \alpha = \phi(\alpha)$$

and so α is a fixed point for ϕ.

Now, assuming also that $|\phi'(x)| \leq k < 1$ in $[a, b]$, we prove the uniqueness of the fixed point. The proof is by contradiction. Suppose that there are two fixed points $\alpha_1 \in [a, b]$ and $\alpha_2 \in [a, b]$. Thus, we have

$$\begin{aligned} | \alpha_1 - \alpha_2 | &\overset{(1)}{=} | \phi(\alpha_1) - \phi(\alpha_2) | \overset{(2)}{=} | \phi'(\xi) \cdot (\alpha_1 - \alpha_2) | = | \phi'(\xi) | \cdot | \alpha_1 - \alpha_2 | \\ &\overset{(3)}{\leq} k | \alpha_1 - \alpha_2 | \overset{(3)}{<} | \alpha_1 - \alpha_2 | \end{aligned}$$

where (1) is due to the fact that $\alpha_1 = \phi(\alpha_1)$ and $\alpha_2 = \phi(\alpha_2)$ since they are fixed points; (2) follows from the the mean value theorem, with $\xi \in [a, b]$; (3) comes from the boundedness of $| \phi'(x) | \leq k < 1$. The previous inequality is a contradiction. Thus, there is only one fixed point in the interval $[a, b]$ and the proof is complete. □

Exercise 2.6 *Give an example of a function f which has the three fixed points $\alpha_1 = -1$, $\alpha_2 = 0$ and $\alpha_3 = 1$.*
[Hint: consider, for example, the function $f(x) = x^3 + a_2x^2 + a_1x + a_0$. Then, to find coefficients a_k, $k = 1, 2, 3$, we impose the three equations $f(\alpha_k) = \alpha_k$, $k = 1, 2, 3$.]

Exercise 2.7 (♡) *Is it possible to find a differentiable function f which has a unique fixed point α with $| f'(\alpha) | > 2$? If possible, write down the function f; if not possible, prove it.*

2.3.1 Fixed point iterations

To introduce the fixed point method, the first step is to rewrite the equation $f(x) = 0$ in the form $x = \phi(x)$ for some function ϕ. The function ϕ is not unique. For example, consider the equation $x^2 - 1 = 0$. We can rewrite it as

$$\text{(a) } x = x^2 + x - 1 =: \phi(x), \qquad \text{(b) } x = \frac{1}{x} =: \phi(x), \qquad \text{(c) } x = \frac{-x^2 + 4x + 1}{4} =: \phi(x)$$

and in many other manners.

Next, let $\alpha \in [a, b]$ be the unique fixed point in the interval $[a, b]$ of $x = \phi(x)$. Given an initial estimate $x_0 \in [a, b]$ of the fixed point α, we consider the following iterative scheme for the computation of α:

$$\begin{cases} x_0 & \text{given initial estimate of } \alpha \\ x_{k+1} = \phi(x_k), & k = 0, 1, 2, \ldots \end{cases}$$

The following theorem provides whether the previous iterations x_k converges to the fixed point α of $x = \phi(x)$.

Theorem 2.3 (Convergence of the iterations) *Let ϕ be a continuous function on $[a, b]$, differentiable in (a, b) with*

(i) $\phi([a,\,b]) \subseteq [a,\,b]$;

(ii) $|\,\phi'(x)\,| \leq K < 1 \quad \forall\, x \in (a,b)$

Then, the sequence

$$x_{k+1} = \phi(x_k), \quad k = 0, 1, 2, \ldots$$

converges to the unique fixed point $\alpha \in [a,\,b]$ *for any choice of* $x_0 \in [a,\,b]$.

Proof. We divide the proof in three steps.

(a) All the values x_k, $k = 0, 1, 2, \ldots$ belongs to the interval $[a,\,b]$. This is clearly true for x_0 by assumption. Now, assumed that $x_k \in [a,\,b]$ we have, using (i), that $x_{k+1} = \phi(x_k) \in [a,\,b]$. Thus, by mathematical induction, $x_k \in [a,\,b]$ for all k.

(b) Let $e_k = x_k - \alpha$ be the error of the k-th iterate. The following inequality holds for each $k = 1, 2, 3, \ldots$

$$\begin{aligned} |\,e_k\,| &= |\,x_k - \alpha\,| \overset{(1)}{=} |\,\phi(x_{k-1}) - \phi(\alpha)\,| \overset{(2)}{=} |\,\phi'(\xi_k)\cdot(x_{k-1}-\alpha)\,| \\ &= |\,\phi'(\xi_k)\,|\cdot|\,x_{k-1}-\alpha\,| \overset{(3)}{\leq} k\,|\,e_{k-1}\,| \end{aligned}$$

where (1) follows from the definition of fixed point iterations using also the fact that α is a fixed point for ϕ; (2) comes from the mean value theorem where ξ_k is a point between x_{k-1} and α (and so, $\xi_k \in [a,\,b]$); (3) follows immediately from (ii).

(c) Using the previous equation $|e_k| \leq k\,|e_{k-1}|$ we can relate $|e_k|$ to $|e_0|$ since

$$\begin{aligned} |\,e_k\,| &\leq k\,|\,e_{k-1}\,| \leq k\,(\,k\,|\,e_{k-2}\,|\,) = k^2\,|\,e_{k-2}\,| \\ &\leq k^2\,(\,k\,|\,e_{k-3}\,|\,) = k^3\,|\,e_{k-3}\,| \\ &\leq \ldots \leq K^k\,|\,e_0\,| \end{aligned}$$

Recalling that $0 \leq K < 1$, we have

$$0 \leq \lim_{k\to+\infty} |\,e_k\,| \leq \lim_{k\to+\infty} K^k\,|\,e_0\,| = |\,e_0\,|\cdot\lim_{k\to+\infty} K^k = |\,e_0\,|\cdot 0 = 0$$

and so the iterates x_k converge to α.

This ends the proof. □

It is also interesting the following result, which we do not prove.

Theorem 2.4 (Ostrowski) *Let* ϕ *be a differentiable function in* $[a,\,b]$ *with fixed point* $\alpha \in [a,\,b]$. *If* $|\,\phi'(\alpha)\,| < 1$, *then exists* $\delta > 0$ *such that the fixed point iterations* $x_{k+1} = \phi(x_k)$ *converge to* α *for each* x_0 *with* $|x_0 - \alpha| < \delta$.

The fixed point iterations has the remarkable geometric interpretation shown in Figures 2.7 and 2.8. This geometric interpretation is an extremely valuable tool to study the behavior of fixed point iterations. For example, once we have sketched the graph of ϕ near the fixed point α, we can see if the iterations approximate monotonically α from below (as in Figure 2.7 (a)) or, alternatively, from below and from above (as in the case shown in Figure 2.7 (b)) or in some other manner. Furthermore, we can see if the fixed point iterations are convergent or divergent without the needed to compute $\phi'(\alpha)$.

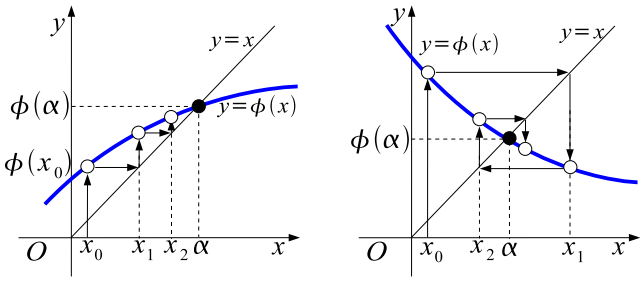


Figure 2.7: The fixed point iterations converge to the fixed point α if $|\,\phi'(\alpha)\,| < 1$. On the left, $0 < \phi'(\alpha) < 1$: the iterations converge to α in a monotone fashion (increasing or decreasing accordingly to the position of x_0 with respect to α). On the right, $-1 < \phi'(\alpha) < 0$: the iterations converge to α with values alternately above and below α.

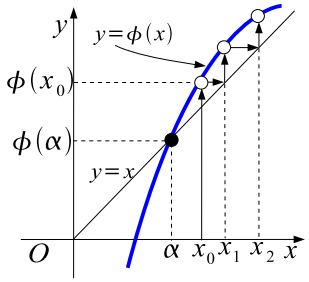


Figure 2.8: The fixed point iterations diverge from the fixed point α if $|\,\phi'(\alpha)\,| > 1$ (in this case we have $\phi'(\alpha) > 0$).

Example 2.4 *If $|\,\phi'(\alpha)\,| = 1$ the fixed point iteration $x_{k+1} = \phi(x_k)$ may, or may not, converge to the fixed point. The functions*

$$(a)\ \phi(x) = x^3 - 3x^2 + 4x - 1 \qquad (b)\ \phi(x) = -x^3 + 3x^2 - 2x + 1$$

have both the fixed point $\alpha = 1$ with $|\,\phi'(\alpha)\,| = 1$. For example for case (a) we have,

$$\phi(\alpha) = \phi(1) = 1^3 - 3 \cdot 1^2 + 4 \cdot 1 - 1 = 1 = \alpha$$

and, since $\phi'(x) = 3x^2 - 6x + 4$,

$$\phi'(\alpha) = \phi'(1) = 3 \cdot 1^2 - 6 \cdot 1 + 4 = 1.$$

Nevertheless, the behavior of the fixed point iterations are quite different. Indeed, the geometrically interpretation suggests that we have divergence for case (a) and convergence for case (b) as shown in Fig. 2.9.

Now, we state a result about the order of convergence of a fixed point iterations.

Theorem 2.5 *Let $\phi \in C^p(\,(\alpha - \delta, \alpha + \delta)\,)$ for suitable $\delta > 0$ and integer $p \geq 1$ of the fixed point α of ϕ. If*

$$\phi'(\alpha) = \phi''(\alpha) = \cdots = \phi^{(p-1)}(\alpha) = 0 \qquad \text{and} \qquad \phi^{(p)}(\alpha) \neq 0$$

then the fixed point iterations $x_{k+1} = \phi(x_k)$ has order of convergence p and

$$\lim_{k \to +\infty} \frac{|\ e_{k+1}\ |}{|\ e_k\ |^p} = \frac{|\ \phi^{(p)}(\alpha)\ |}{p!}.$$

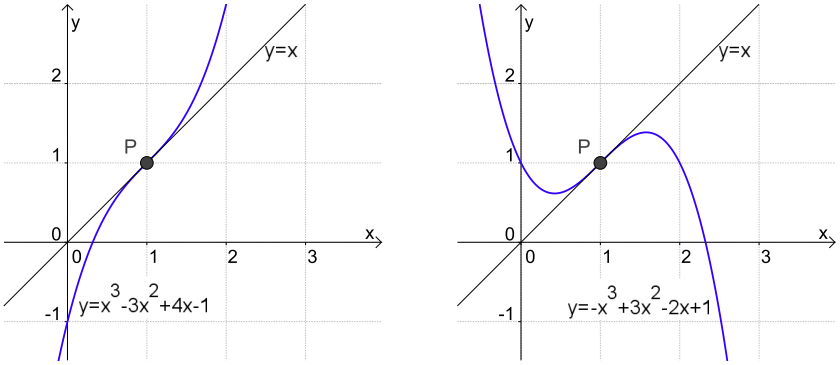


Figure 2.9: Two different behavior of the fixed point iterations when $|\,\phi'(\alpha)\,| = 1$. On the left, iterations x_k diverge from α for each $x_0 \neq \alpha$ chosen near α. On the right, iterations x_k converge to α for each x_0 chosen near α.

Proof. Using Taylor expansion, recalling that, by definition, $\phi^{(0)}(\alpha)$, we get

$$\begin{aligned} e_{k+1} &= x_{k+1} - \alpha = \phi(x_k) - \alpha \\ &= \sum_{j=0}^{p-1} \frac{\phi^{(j)}(\alpha)\cdot(x_k-\alpha)^j}{j!} + \frac{\phi^{(p)}(\xi_k)\cdot(x_k-\alpha)^p}{p!} - \alpha \\ &= \phi^{(0)}(\alpha) + \sum_{j=1}^{p-1} \frac{\phi^{(j)}(\alpha)\cdot(x_k-\alpha)^j}{j!} + \frac{\phi^{(p)}(\xi_k)\cdot(\,e_k\,)^p}{p!} - \alpha \\ &\overset{(1)}{=} \frac{\phi^{(p)}(\xi_k)\cdot(\,e_k\,)^p}{p!} \end{aligned}$$

where ξ_k is a suitable point between α and x_k and (1) follows from $\phi^{(j)}(\alpha) = 0$, $j = 1,\ldots,p-1$ and $\phi(\alpha) = \alpha$ since α is a fixed point. Providing that x_k converges to the fixed point α, we also have that $\xi_k \to \alpha$ which completes the proof due to the continuity of $\phi^{(p)}$. □

2.3.2 Termination of the fixed point iterations

It is common to terminate the convergent fixed point iterations

$$\begin{cases} x_0 & \text{given initial estimate of the fixed point } \alpha \\ x_{k+1} = \phi(x_k), & k = 0,1,2,\ldots \end{cases}$$

when $|\,x_{k+1} - x_k\,| < \varepsilon$ for some given tolerance $\varepsilon > 0$.
Let us see how good is this stopping criteria. We have

$$x_{k+1} - \alpha = \phi(x_k) - \phi(\alpha) = \phi'(\xi_k)(x_k - \alpha)$$

for some ξ_k in the interval of endpoints α and x_k. Since it is

$$x_k - \alpha = (x_{k+1} - \alpha) - (x_{k+1} - x_k) \quad \Rightarrow \quad x_{k+1} - \alpha = x_k - \alpha + x_{k+1} - x_k$$

and denoting the error at the k-ih iteration by $e_k = x_k - \alpha$ we obtain

$$x_k - \alpha + x_{k+1} - x_k = \phi'(\xi_k)(x_k - \alpha) \quad \Rightarrow \quad e_k + x_{k+1} - x_k = \phi'(\xi_k)\,e_k$$

and finally, assuming that $\phi'(x) \neq 0$ near α and taking the absolute values,

$$|\,e_k\,| = \frac{1}{|\,1 - \phi'(\xi_k)\,|}\cdot|\,x_{k+1} - x_k\,| \tag{2.3}$$

So, if $\phi'(\alpha) \approx 0$ (and, thus, $\phi'(x) \approx 0$ near α by continuity) the difference between two consecutive iterates is a reliable estimator of the error. Note that this is the case if $\phi'(\alpha) = 0$. If, otherwise, $\phi'(\alpha) \approx 1$, eq. (2.3) is not useful to estimate the error.

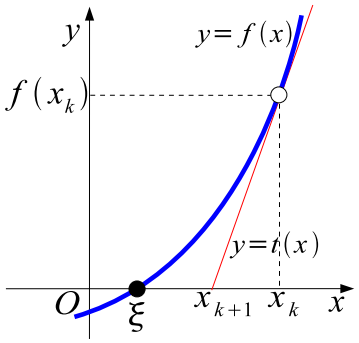
Remark 2.2 *Equation (2.3) may be used to estimate the asymptotic error constant without computing errors. This is interesting since in real cases we do not know α (indeed, we are searching α). Let p be the order of the fixed point method. Then, we get roughly*

$$C \approx \frac{|e_{k+1}|}{|e_k|^p} = \frac{\frac{1}{|\,1-\phi'(\xi_{k+1})\,|}\cdot|\,x_{k+2}-x_{k+1}\,|}{\left[\frac{1}{|\,1-\phi'(\xi_k)\,|}\cdot|\,x_{k+1}-x_k\,|\right]^p} \approx \frac{|\,x_{k+2}-x_{k+1}\,|}{|\,x_{k+1}-x_k\,|^p}$$

if we assume $\phi'(\xi_{k+1}) \approx \phi'(\xi_k)$ at least for large k. This is not a strong assumption when we are near the fixed point α; in this case, $x_{k+1} \approx x_k$ and so, due to the continuity of ϕ', $\phi'(x_{k+1}) \approx \phi'(x_k)$.

2.4 The Newton and the secant method

Let us derive the method of Newton from geometric considerations. Consider the graph of the function f near the root ξ shown in the following figure.



Let x_k be an estimation of ξ. Consider the tangent to f at x_k

$$t(x) = f'(x_k)\cdot(x-x_k) + f(x_k)$$

The next estimation of ξ is computed as the point x_{k+1} where $t(x) = 0$; we get

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

This equation is the single step of the Newton method. So, starting from a given initial guess x_0 of the root ξ, the Newton method repeats the previous step until x_k becomes near enough to the root ξ

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

Theorem 2.6 (Local convergence of the Newton method) *Let $\mathcal{I} = (a, b)$ be an open interval containing the only root ξ of $f \in C^m(\mathcal{I})$ with $m \geq 2$. Then, there is $\delta > 0$ such that the Newton iterations*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

converges to ξ for each $x_0 \in (\xi-\delta, \xi+\delta) \cap \mathcal{I}$.
Moreover, setting

$$\lim_{k\to+\infty} \frac{|\,e_{k+1}\,|}{|\,e_k\,|^p} = C$$

we have two different cases for the behaviour of the error:

- *if $f'(\xi) \neq 0$ we have*

$$p \geq 2, \qquad C = \left|\frac{f''(\xi)}{2\,f'(\xi)}\right|$$

  More precisely, the order is $p = 2$ if also $f''(\xi) \neq 0$, otherwise it is $p \geq 3$.

- *if $f'(\xi) = \cdots = f^{(m-1)}(\xi) = 0$ and $f^{(m)}(\xi) \neq 0$ we have*

$$p = 1, \qquad C = 1 - \frac{1}{m}$$

  In this case, we can resume $p = 2$ considering the modified Newton method

$$x_{k+1} = x_k - m\cdot\frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

for which it is also

$$C = \frac{1}{m \cdot (m+1)} \left| \frac{f^{(m+1)}(\xi)}{f^{(m)}(\xi)} \right|.$$

Proof. Let us prove only the first part of the Theorem. We can see the Newton method as a fixed point iteration with iteration function

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

and fixed point ξ. We have

$$\phi'(x) = 1 - \frac{f'(x) \cdot f'(x) - f(x) \cdot f''(x)}{[\, f'(x)\,]^2} = \frac{f(x) \cdot f''(x)}{[\, f'(x)\,]^2}$$

and

$$\phi''(x) = \frac{[f'(x) \cdot f''(x) + f(x) \cdot f'''(x)\,] \cdot [\, f'(x)\,]^2 - f(x) \cdot f''(x) \cdot 2 \cdot f'(x) \cdot f''(x)}{[\, f'(x)\,]^4}$$

Since $f(\xi) = 0$ and $f'(\xi) \neq 0$, we obtain $\phi'(\xi) = 0$, $\phi''(\xi) = f''(\xi)/f'(\xi)$. Thus, the method has a local convergence due to Theorem of Ostrowski. Also, the order of convergence is at least $p = 2$ and, providing $f''(\xi) \neq 0$, is exactly $p = 2$ with an asympotic error constant given by

$$\left| \frac{\phi^{(2)}(\xi)}{2\,!} \right| = \left| \frac{f''(\xi)}{2\, f'(\xi)} \right|$$

Thus the proof is complete. □

Remark 2.3 (Stopping criteria for Newton) *The Newton method is a fixed point iteration $x = \phi(x) := x - f(x)/f'(x)$. If $f'(\xi) \neq 0$ we have $\phi'(\xi) = 0$ and the iterative process can be stopped when $|x_{k+1} - x_k| < \varepsilon$ for some given $\varepsilon > 0$. Indeed, this is an excellent estimation of the error (see subsection 2.3.2). On the other hand the previous criteria may be inaccurate if the root ξ has multiplicity greater then 1. Indeed, in this case $f'(\xi) = 0$ and it can be shown that $\phi'(\xi) = 1 - 1/m$.*

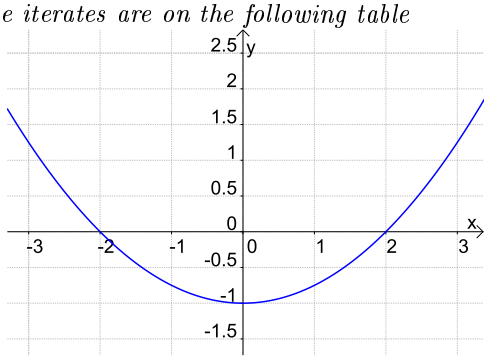
Example 2.5 *Consider the function $f(x) = 0.25x^2 - 1$. Let us apply the Newton method starting from $x_0 = 3$ with a stopping criteria $\mid x_{k+1} - x_k \mid < \varepsilon = 10^{-9}$. From the graph of f we can see that iterates are monotonically decreasing and converge to the root $\alpha = 2$. This is a single root since, again, we see from the graph that $f'(2) > 0$; alternatively, we can prove this using the first derivative of f:*

$$f'(x) = 0.25 \cdot 2 \cdot x = 0.5x \qquad \Rightarrow \qquad f'(2) = 0.5 \cdot 2 = 1 \neq 0.$$

Moreover, since $f''(\alpha) = 0.5 \neq 0$, we expect that the Newton method has order $p = 2$ with error constant

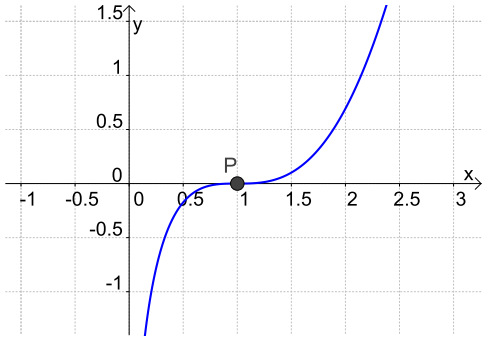
$$C = \left| \frac{f''(\alpha)}{2f'(\alpha)} \right| = \left| \frac{0.5}{2 \cdot 1} \right| = 0.25.$$

The iterates are on the following table



k	x_k
0	3.000000000000000e+000
1	2.166666666666667e+000
2	2.006410256410256e+000
3	2.000010240026215e+000
4	2.000000000026214e+000
5	2.000000000000000e+000

Example 2.6 *Consider the function* $f(x) = (x-1)^2 \log(x)$. *This equation has only one root,* $\xi = 1$ *with multiplicity* $m = 3$. *Its graph is shown in the following figure among with the iterates starting from* $x_0 = 1.5$. *The stopping criteria is* $|x_{k+1} - x_k| < \varepsilon$ *with* $\varepsilon = 10^{-3}$.



k	x_k	k	x_k
0	1.5000	9	1.0118
1	1.3228	10	1.0078
2	1.2104	11	1.0052
3	1.1381	12	1.0035
4	1.0911	13	1.0023
5	1.0603	14	1.0015
6	1.0400		
7	1.0266		
8	1.0177		

We make some comments. The iterates are monotonically decreasing as one expects due to the geometrical interpretation of the Newton method on the graph of $y = f(x)$ *given. Note, also, that we have*

$$\phi'(1) \approx \frac{|\, e_{14} \,|}{|\, e_{13} \,|} = \frac{|\, 1.000 - 1.0023 \,|}{|\, 1.000 - 1.0015 \,|} = 0.66635$$

a value that is quite close to the theoretical one $1 - 1/m = 1 - 1/3 = 2/3$. *Finally, we have*

$$|\, x_{14} - x_{13} \,| = 8 \times 10^{-4} < \epsilon$$

but $|e_{14}| = 1.5 \times 10^{-3} > \epsilon$ *due to the fact that* $\phi'(1) = 2/3 \approx 1$ *(and so the used stopping criteria is not so satisfactory).*

It is also interesting to plot the absolute value of the error $e_k = \alpha - x_k$, $k = 0, 1, \cdots$ *and the absolute value of* $s_k := x_{k+1} - x_k$, $k = 0, 1, \cdots$.

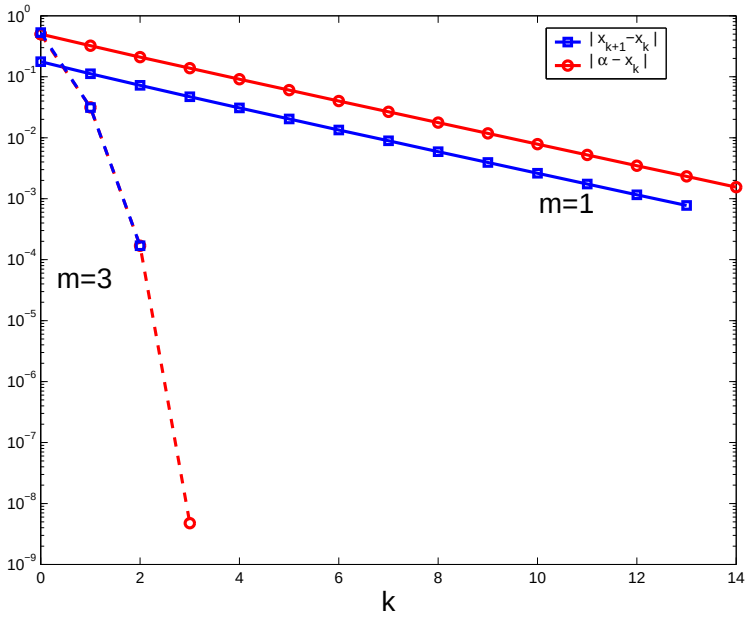


Figure 2.10: The Convergence process for the Newton method applied to approximate the root $\alpha = 1$ of the equation $(x-1)^2 \log(x) = 0$. In dashed lines we have the behaviour of he modified Newton method.

2.4.1 The secant method

The secant method follows from Newton approximating $f'(x_k)$ as

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

Thus, given two (possibly suitable) starting points x_0 and x_1, we have the iterations

$$x_{k+1} = x_k - \frac{f(x_k) \cdot (x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \quad k = 0, 1, 2, \ldots$$

Each step of this method has a simple geometrical interpretation: x_{k+1} is the intersection point with the x axis of the line for the two points $(x_{k-1}, f(x_{k-1}))$ and $(x_k, f(x_k))$.

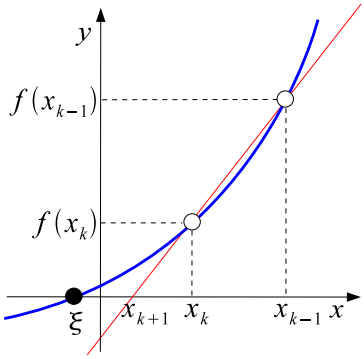
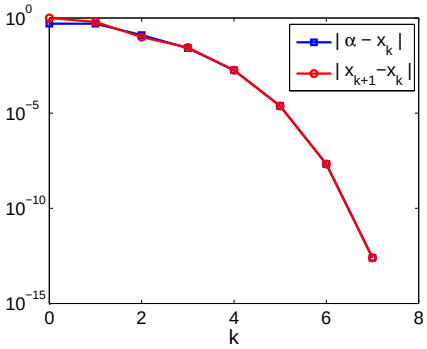


Figure 2.11: The single step of the secant method.

For the convergence of the secant method we have the following theorem.

Theorem 2.7 *Let α be a simple root of the equation $f(x) = 0$ with $f \in C^2(\mathcal{I})$ in some open interval $\mathcal{I}$ containing the root.*
Then exists a ball $B(x_0, \delta)$ such that the secant method converges to α for each couple $x_0, x_1 \in B(x_0, \delta)$. Moreover, the order of convergence is $p = (1 + \sqrt{5})/2 \approx 1.618$.

Example 2.7 *The secant method for the computation of the root $\alpha = 1$ of the equation $f(x) = x^2 - 1$ gives the following results*



k	x_k
0	0.50000000000000
1	1.50000000000000
2	0.87500000000000
3	0.97368421052632
4	1.00177935943061
5	0.99997629657723
6	0.99999997893004
7	1.00000000000025
8	1.00000000000000

Note that both error and difference between two consecutive iterates have the following behaviour. So, as for Newton method with a single root, it is a good choice to stop the iterations of the secant method when $|x_{k+1} - x_k| < \varepsilon$. This is not surprising: looking back to the geometrical derivation of the secant method, we can see that it resembles the Newton method.

Example 2.8 *We compute x_2 of the secant method starting from $x_0 = 3$, $x_1 = 2$ for $f(x) = x^2 - 1$. Instead of remember the formula, recall the derivation. The line through points $(x_0, f(x_0)) = (3, 8)$ and $(x_1, f(x_1)) = (2, 3)$ is*

$$y = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \cdot (x - x_0) + f(x_0) = \frac{3 - 8}{2 - 3}(x - 3) + 8 = 5x - 7$$

and thus x_2 is the solution of $5x_2 - 7 = 0$ or $x_2 = 7/5 = 1.4$.

2.5 Exercises

Exercise 2.8 *Find how many roots have the following equations. Give also an interval that contains the small positive root of each equation.*

$$\text{(a) } x = \cos(x), \qquad \text{(b) } x\tan(x) = 1, \qquad \text{(c) } |\ln(x)| - e^{-x} = 0$$

(Hint for (b): rewrite the equation as $\tan(x) = 1/x$).

Exercise 2.9 *Consider the function*

$$f(x) = \begin{cases} -|x-1| & , \quad x < 1 \\ (x-1)^2 & , \quad x \geq 1 \end{cases}$$

(a) Sketch the graph of $y = f(x)$, find and separate the roots.

(b) For which values of the starting point x_0 we may expect that the Newton method converges to the greater root? Can we find an x_0 for which the Newton method converges in less than 2 iterations? Explain your answer.

(c) Setting $x_0 = 2$, find the first five iterates of the Newton method. Estimates the order of convergence from $|e_4|/|e_3|$. Is this correct with our theory? Why? How, if possible, can we improve the rate of convergence?

(d) Set now $x_0 = -10,000$. Find the fourth iterate of the Newton method.

Exercise 2.10 *Consider the function $f(x) = x^3 - x$.*

(a) Sketch a graph of f and separate the roots.

(b) Study the behaviour of Newton method for different starting points. Can we find a point x_0 such that $x_2 = x_0$? What happens if we choose x_0 such that $x_1 = -1/\sqrt{3}$?

(c) Does the following fixed point iterations $x_{k+1} = x_k^3$ converges to any of the root of the function f?

Is it possible to find λ such that

$$x_{k+1} = \lambda x_k^3 - (\lambda - 1)x_k$$

has an order three of convergence?

Exercise 2.11 *Consider the function $f(x) = (x-\alpha)^m$ where m in an integer greater then 1 and $\alpha \in \mathbb{R}$. Prove or disprove with a counterexample that the modified Newton method converges to the (unique) root α of the function f in only one iteration.*

Exercise 2.12 (♡) *Consider the fixed point iterations*

$$x_{k+1} = \frac{x_k}{4} + \frac{3}{4}$$

(a) Prove that there is the unique fixed point $\alpha = 1$.

(b) Starting from $x_0 = 2$, compute the first two iterations of the method and the corresponding absolute value of the errors $e_k = |\alpha - x_k|$.

(c) Compute the order of the method and the asymptotic error constant.

(d) Using Matlab, plot the behavior of the $\log_{10}(|e_k|)$ as a function of k. Does the behavior of the plot agree with the theory?

Answer. (a) We have $\phi(x) = x/4 + 3/4$. Fixed points are solutions of $x = \phi(x)$. So, we get

$$x = \frac{x}{4} + \frac{3}{4} \quad \Leftrightarrow \quad \frac{3x}{4} = \frac{3}{4} \quad \Leftrightarrow \quad x = 1$$

So, there is only the fixed point $\alpha = 1$.
(b) We have

$$\begin{aligned} x_1 &= \frac{x_0}{4} + \frac{3}{4} = \frac{2}{4} + \frac{3}{4} = \frac{3}{2}, \quad e_1 = |\alpha - x_1| = \left|\frac{3}{2} - 1\right| = \frac{1}{2}; \\ x_2 &= \frac{x_1}{4} + \frac{3}{4} = \frac{3/2}{4} + \frac{3}{4} = \frac{9}{8}, \quad e_2 = |\alpha - x_2| = \left|\frac{9}{8} - 1\right| = \frac{1}{8} \end{aligned}$$

(c) Since $\phi'(x) = 1/4$ we have $\phi'(\alpha) = 1/4$; thus, it is $|\phi'(\alpha)| < 1$ and the fixed point iterations are convergent to α. Moreover, since $\phi'(\alpha) \neq 0$, the method has order $p = 1$ and an asymptotic error constant $C = |\phi'(\alpha)|/1! = 1/4$.
(d) Since the fixed point method has order $p = 1$, we expect a linear graph for $\log_{10}(|e_k|)$ (at least for large k). Figure 2.12 on the right for the plot shows an excellent agreement with the theory.

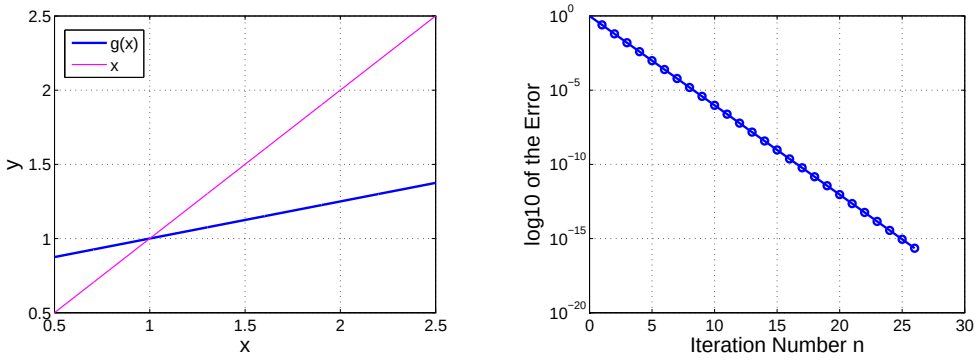


Figure 2.12: Left: plots of $\phi(x) = x/4 + 3/4$ and $y = x$; there is only one intersection point and, thus, only one fixed point.

Exercise 2.13 (♡) *Consider the function $f(x) = (x-1)^4$ which has the only root $\xi = 1$ with multiplicity $m = 4$.*

(a) Write the Newton iteration explicitly for this function.

(b) Using point (a), find the ratio $|e_{k+1}|/|e_k|$, $k = 0, 1, \ldots$. From this ratio, find the asymptotic error constant and the order of convergence p. Finally, sketch the $\log_{10}(|e_k|)$ graph.

(c) Starting from $x_0 = 2$, find iterations x_1 and x_2. Is the sequence x_k of the Newton method monotone? If yes, is an increasing or a decreasing one? Explain.

(d) Give a method of order $p = 2$ to find the root ξ of the equation $f(x) = (x-1)^4$.

Answer. (a, b, c) Since $f'(x) = 4(x-1)^3$, we have

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{(x_k - 1)^4}{4(x_k - 1)^3} = \frac{3x_k + 1}{4}, \; k = 0, 1, \ldots$$

and so the error e_{k+1} is

$$e_{k+1} = \xi - x_{k+1} = 1 - \frac{3x_k + 1}{4} = \frac{3}{4} \cdot (1 - x_k) = \frac{3}{4} e_k, \quad k = 0, 1, \ldots$$

and thus the required ratio is

$$\frac{|e_{k+1}|}{|e_k|} = \frac{3}{4}$$

From this last equation it follows that the Newton method has order of convergence $p = 1$ (as already known, since the root has multiplicity greater then 1) and asymptotic error constant $C = 3/4$. We expect $C < 1$ since we know drawing the graph of $f(x) = x^2 - 1$ that Newton method is convergent to the root ξ (and, moreover, that the sequence x_k converges to ξ from above in a monotone fashion). This, accordingly to the theory, imply that $C < 1$. Finally, the $\log_{10}(|e_k|)$ plot gives a line since the order of the method is $p = 1$. In this case it is also easy to write the equation of the line. We have

$$\frac{|e_k|}{|e_{k-1}|} = \frac{3}{4} \quad \Rightarrow \quad |e_k| = \frac{3}{4} \cdot |e_{k-1}| = \ldots = \left(\frac{3}{4}\right)^k \cdot |e_0|$$

and taking logarithm

$$\log_{10}(|e_k|) = k \cdot \log_{10}(3/4) + \log_{10}(|e_0|)$$

The required iterations are

$$\begin{aligned} x_0 &= 2 \\ x_1 &= \frac{3x_0 + 1}{4} = \frac{3 \cdot 2 + 1}{4} = \frac{7}{4} \\ x_2 &= \frac{3x_1 + 1}{4} = \frac{3 \cdot (7/4) + 1}{4} = \frac{25}{16} \end{aligned}$$

(d) A possible method of order $p = 2$ is the modified Newton method

$$x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)} = 1, \quad k = 0, 1, \ldots$$

and so we have convergence in just one iteration.

Chapter 3

Direct methods for linear systems

The present chapter is devoted to the solution of a linear system $A\mathbf{x} = \mathbf{b}$ where A is an $m \times n$ real matrix, $\mathbf{b}$ is a column vector of length m of real numbers and $\mathbf{x}$ is an n dimension column vector of unknowns. Writing explicitly, we have

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m-1,1} & a_{m-1,2} & \cdots & a_{m-1,n} \\ a_{m1} & a_{m,2} & \cdots & a_{mn} \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_{m-1} \\ b_m \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_{n-1} \\ x_n \end{pmatrix}$$

We show some direct methods, that is methods which, in exact arithmetic, give the correct answer in a finite number of operations.

3.1 Linear Algebra

In this section we recall some useful issues of linear algebra.

3.1.1 Vectors and vector norms

Consider the (column or row) vector space $\mathbb{R}^n$ where n is an integer greater or equal to 1.

The *weight* of a vector $\mathbf{x} \in \mathbb{R}^n$ is measured by its norm $\|\mathbf{x}\|$.

Definition 3.1 *A norm is a function from* $\mathbb{R}^n$ *to* $\mathbb{R}_+ := \{x \in \mathbb{R} \mid x \geq 0\}$ *which satisfies the conditions*

1. $\|\mathbf{x}\| \geq 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *with* $\|\mathbf{x}\| = 0$ *if and only if* $\mathbf{x} = \mathbf{0}$.
2. $\|\alpha \cdot \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$ *for each* $\alpha \in \mathbb{R}$, *for each* $\mathbf{x} \in \mathbb{R}^n$.
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ *for each* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Let x_k be the k-th component of the vector $\mathbf{x}$ so that we can write for the row case

$$\mathbf{x} = [x_1, x_2, \cdots, x_n], \quad x_k \in \mathbb{R}, k = 1, \ldots, n$$

Same notations hold for a column vector. We use only the following vector norms:

$$\|\mathbf{x}\|_1 = \sum_{k=1}^{n} |x_k| \ \text{(1-norm)} \qquad \|\mathbf{x}\|_2 = \left(\sum_{k=1}^{n} |x_k|^2 \right)^{1/2} \ \text{(2-norm)}$$

$$\|\mathbf{x}\|_\infty = \max_{1 \leq k \leq n} |x_k| \ \text{(} \infty\text{-norm)}$$

The 2-norm is also known as Euclidean norm; the ∞-norm as the maximum or uniform norm.

Example 3.1 *Let* $\mathbf{x} = [1\,,\ -2\,,\ 2]$. *Then, we have*

$$\|\mathbf{x}\|_1 = |1| + |-2| + |2| = 5, \qquad \|\mathbf{x}\|_2 = \left(|1|^2 + |-2|^2 + |2|^2 \right)^{1/2} = 3,$$

$$\|\mathbf{x}\|_\infty = \max\{\, |1|\,, |-2|\,, |2|\,\} = 2$$

We obtain the same results if $\mathbf{x}$ *is a column vector.*

3.1.2 Eigenvectors and eigenvalues

Let's now consider the vector space $\mathbb{R}^{n\times n}$ of square matrices of order n.

Definition 3.2 *The number* $\lambda \in \mathbb{C}$ *is an eigenvalue for* $A \in \mathbb{R}^{n\times n}$ *if there is a column vector (called an eigenvector)* $\mathbf{x} \neq \mathbf{0}$ *such that* $A\mathbf{x} = \lambda\mathbf{x}$.

All the eigenvalues of A are solutions of the algebraic equation

$$P(\lambda) = 0 \quad \text{where} \quad P(\lambda) = |\, A - \lambda I_n \,|$$

where I_n is the identity matrix of order n (i.e., the matrix with all zeros but ones in the main diagonal).The algebraic multiplicity $\mu(\lambda)$ of an eigenvalue λ is the multiplicity of this root in the equation $P(\lambda) = 0$. The spectrum of A is the set $\sigma(A)$ containing all the eigenvalues.

Definition 3.3 *Let* $A \in \mathbb{R}^{n\times n}$. *The maximum modulus of the eigenvalues is the spectral radius* $\rho(A)$ *of the matrix* A

$$\rho(A) := \max_{1\le k\le n} |\lambda_k|.$$

Example 3.2 *The eigenvalues of the upper triangular matrix*

$$U = \begin{pmatrix} 1 & \alpha & \beta \\ 0 & 1 & \gamma \\ 0 & 0 & -3 \end{pmatrix} \qquad \textit{where } \alpha, \beta, \gamma \in \mathbb{R}$$

are elements of the main diagonal and so $\lambda_1 = 1$ *with* $\mu(\lambda_1) = 2$ *and* $\lambda_2 = -3$ *with* $\mu(\lambda_2) = 1$. *Thus,* $\sigma(A) = \{1, -3\}$ *and* $\rho(A) = \max\{|1|, |1|, |-3|\} = 3$. *We remark that eigenvalues of a diagonal matrix as well as upper or lower triangular matrices are elements of the main diagonal.*

Eigenvalues of the matrix $A \in \mathbb{R}^{n\times n}$ satisfy the

Theorem 3.1 (Gershgorin) *Let* $A \in \mathbb{R}^{n\times n}$. *Let* $\mathcal{D}_i$, $i = 1, \cdots, n$ *be the disks in the complex plane defined as*

$$\mathcal{D}_i := \left\{ z \in \mathbb{C} : |z - a_{ii}| \le r_i, \ \ \textit{where } \ r_i = \sum_{\substack{k=1 \\ k\neq i}}^{n} |a_{ik}| \right\}$$

Then, all the eigenvalues of A *are inside* $\mathcal{D}$ *defined as*

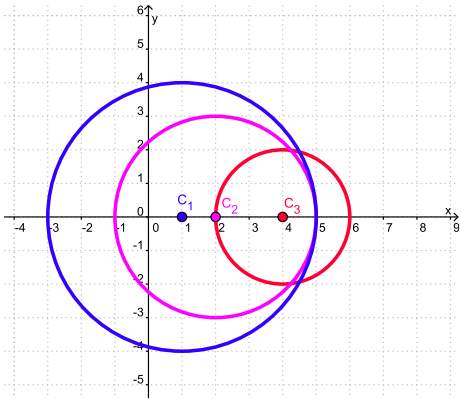
$$\mathcal{D} = \bigcup_{1\le i\le n} \mathcal{D}_i$$

i.e., the union of disks $\mathcal{D}_i$, $i = 1, \ldots, n$.

Example 3.3 *The eigenvalues of the matrix*

$$A = \begin{pmatrix} 1 & -4 & 0 \\ 3 & 2 & 0 \\ 1 & 1 & 4 \end{pmatrix}$$

are $\lambda_1 = 4$, $\lambda_{2,3} = (-3 \pm i\sqrt{47})/2 \approx 1.5 \pm i3.4278$ *and can be obtained as the roots of* $P(\lambda) = (\lambda - 4)(\lambda^2 - 3\lambda + 14)$. *The three Gershgorin disks are shown in the next Figure.*



3.1.3 Matrix norms

We consider on this space only norms which fulfill the following properties 1 to 5

1. $\|A\| \geq 0$ for all $A \in \mathbb{R}^{n\times n}$ with $\|A\| = 0$ if and only if $A = 0 \in \mathbb{R}^{n\times n}$.
2. $\|\alpha \cdot A\| = |\alpha| \cdot \|A\|$ for each $\alpha \in \mathbb{R}$, for each $A \in \mathbb{R}^{n\times n}$.
3. $\|A + B\| \leq \|A\| + \|B\|$ for each $A, B \in \mathbb{R}^{n\times n}$.
4. $\| A \cdot B \| \leq \| A \| \cdot \| B \|$ for each $A, B \in \mathbb{R}^{n\times n}$

We also require the compatibility condition with a vector norm, that is

5. $\|A\mathbf{x}\| \leq \|A\| \cdot \| \mathbf{x} \|$ for each $A \in \mathbb{R}^{n\times n}$ and for each $\mathbf{x} \in \mathbb{R}^n$.

Definition 3.4 *The norm of the matrix* $A \in \mathbb{R}^{n\times n}$ *induced by the vector norm* $\|\mathbf{x}\|$ *is defined by*

$$\| A \| := \max_{\|\mathbf{x}\|=1} \|A \cdot \mathbf{x}\|$$

Theorem 3.2 *Let* $A \in \mathbb{R}^{n\times n}$. *The matrix norms induced by the* 1, 2 *and* ∞ *vector norms are*

$$\begin{aligned} \| A \|_1 &= \max_{1\leq j\leq n} \sum_{k=1}^{n} |a_{kj}| \quad (\textit{ maximum column sum }) \\ \| A \|_2 &= \sqrt{\rho(A^T \cdot A)} \\ \| A \|_\infty &= \max_{1\leq i\leq n} \sum_{k=1}^{n} |a_{ik}| \quad (\textit{ maximum row sum }) \end{aligned}$$

Another quite important norm, since it is simple to compute, is the Frobenius norm. It is defined as

$$\|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} | \, a_{ij} \, |^2 \right)^2$$

This norm is not induced by any vector norm.

Example 3.4 *Consider the matrix, taken from G. Zilli, "Calcolo Numerico"*

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 1 & 3 & -1 \\ 0 & 1 & 1 \end{pmatrix}$$

We have

$$\begin{aligned} \| A \|_1 &= \max\{|4| + |1| + |0|,\ | -1| + |3| + |1|,\ |1| + | -1| + |1|\} = 5 \\ \| A \|_\infty &= \max\{| -4| + | -1| + |1|,\ |1| + |3| + | -1|,\ |0| + |1| + |1|\} = 6 \\ \| A \|_F &= \left[4^2 + 1^2 + 0^2 + (-1)^2 + 3^2 + 1^2 + 1^2 + (-1)^2 + 1^2 \right]^{1/2} = \sqrt{31} \\ &\approx 5.5678 \end{aligned}$$

Moreover, since

$$A^T \cdot A = \begin{pmatrix} 4 & 1 & 0 \\ -1 & 3 & 1 \\ 1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & -1 & 1 \\ 1 & 3 & -1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 17 & -1 & 3 \\ -1 & 11 & -3 \\ 3 & -3 & 3 \end{pmatrix}$$

Using Matlab, the eigenvalues of $A^T \cdot A$ are $\lambda_1 = 18$, $\lambda_2 = 11.4244$, $\lambda_3 = 1.5756$. Thus, it is $\rho(A^T \cdot A) = 18$ and so finally we have

$$\|A\|_2 = \sqrt{\rho(A^T \cdot A)} = \sqrt{18} = 4.2426$$

Norms of matrices can help us to delimit the complex plane where the eigenvalues of a matrix belong to. Indeed, the following theorem holds.

Theorem 3.3 *For any induced matrix norm, we have*

$$\rho(A) \leq \|A\|$$

Proof. Let $\mathbf{y}$ be an eigenvector associated to the eigenvalue λ. Assume also that $\|\mathbf{y}\| = 1$. We have

$$|\lambda| = |\lambda| \cdot \|\mathbf{y}\| = \|\lambda \cdot \mathbf{y}\| = \|A \cdot \mathbf{y}\| \leq \max_{\|\mathbf{x}\|=1} \|A \cdot \mathbf{x}\| = \|A\|$$

Thus, all eigenvalues are, in modulus, less than $\|A\|$. So, also the greatest in modulus eigenvalue, that is $\rho(A)$, fulfill this inequality. □

Example 3.5 *Consider again the matrix of Example 3.4. We may write*

$$\rho(A) \leq \min\{\| A \|_1, \| A \|_2, \| A \|_\infty\} = \| A \|_2 < 4.3.$$

So, all the eigenvalues of the matrix A have modulus less than 4.3. Say in another way, they lie in the disk of the complex plane of radius $r = 4.3$ and center $0 = 0 + i0$.

Remark 3.1 *In Matlab or Octave environment, a matrix norm can be computed using the function* `norm`*:* `norm(A,1)`*,* `norm(A,2)`*,* `norm(A,'inf')`*,* `norm(A,'fro')` *returns, respectively, 1, 2, ∞ and Frobenius norms of the matrix A. See the help command for more details.*

Remark 3.2 *In Matlab or Octave environment, the (real and complex) eigenvalues of the matrix $A \in \mathbb{R}^{n \times n}$ can be obtained using the function* `eig(A)`*. For each matrix A,* `eig` *selects the most suitable algorithm to compute the eigenvalues of A.*

Remark 3.3 (Solution of an algebraic equation) *Consider the polynomial*

$$p(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

where a_k, $k = 0, \ldots, n-1$ are real numbers. It is known that there are no close formulas for $n \geq 5$ to compute its roots. Moreover, even for $n = 2$ the formulas are numerically not stable.

That said, it is interesting to find alternative ways to compute the roots of a polynomial. There are many such ways. We give just one. It can be proved that the companion matrix

$$A = \begin{pmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \dots & -a_1 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

has as its eigenvalues the roots of p. So, the function `eig()` *can be used to find the roots of p. As an example, consider $p(x) = x^3 + x^2 - 2$ whose roots are 1 and $-1 \pm i$. Polynomial p has $a_2 = 1$, $a_1 = 0$, $a_0 = -2$ and so the companion matrix is*

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

The function `eig(A)` *under Matlab or Octave, gives the roots of p with 16 correct figures.*

3.1.4 Positive definite matrices

A very special set of matrices is the set of symmetric positive definite matrices.

Definition 3.5 *A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is said to be positive definite if $\mathbf{x}^T A \mathbf{x} > 0$ for every non zero $\mathbf{x} \in \mathbb{R}^n$.*

If A is positive definite we write $A > 0$. It can be proved the following theorem.

Theorem 3.4 *A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is positive definite if and only if has all the eigenvalues positive (i.e, $\lambda_k > 0$, $k = 1, \dots, n$).*

Example 3.6 *Consider the symmetric matrix A of order $n = 100$ with all zeros elements but $a_{ii} = 3$, $i = 1, \dots, n$, $l_{i,i-1} = 1$, $i = 2, \dots, n$, $l_{i,i+1} = 1$, $i = 1, \dots, n-1$. This matrix is positive definite since, by Gershgoring theorem, all eigenvalues are inside the disk of center $(3, 0)$ and radius $r = 2$ (more precisely, since A is symmetric, the eigenvalues are real and so the rely on the real line inside the disk). This disk does not contain the point $(0, 0)$ and so it is completely on the right hand side of the imaginary axis. Thus, all the eigenvalues are positive and, due to theorem 3.4, A is positive definite.*

Now we present some algorithms for the solution of square, non singular, linear systems $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^{n \times 1}$. There are many algorithm to compute $\mathbf{x}$. Among them, we show a very interesting one known as the Gauss algorithm (or Gaussian elimination) which is both fast and, for all practical cases, accurate. Let's start with two easy cases, the upper and lower triangular linear systems.

3.2 Solution of triangular systems

We recall that a matrix A is said

- LOWER TRIANGULAR: all the elements above the main diagonal are zeros, i.e. $a_{ij} = 0$, $j > i$. For example, these matrices are all lower triangular regardless the values of l_{ij}:

$$\begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix}, \quad \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}, \quad \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix}$$

- UPPER TRIANGULAR: all the elements below the main diagonal are zeros, i.e. $a_{ij} = 0$, $j < i$. For example, these matrices are all upper triangular regardless the values of u_{ij}:

$$\begin{pmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{pmatrix}, \quad \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}, \quad \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

Usually, we denote with L and U the lower triangular matrices and the upper triangular matrices, respectively.

It is easy to solve a triangular linear system. Let's see the lower triangular ones. For these systems, we first solve the first equation $l_{11}x_1 = b_1$ which gives $x_1 = b_1/l_{11}$. Then we solve the second equation $l_{21}x_1 + l_{22}x_2 = b_2$ with respect to x_2 since we know x_1 from the previous step: we get $x_2 = (b_2 - l_{21}x_1)/l_{22}$. From the third equation $l_{31}x_1 + l_{32}x_2 + l_{33}x_3 = b_3$, since we have already computed x_1 and x_2, we find x_3 as $x_3 = (b_3 - l_{31}x_1 - l_{32}x_2)/l_{33}$. Proceeding in this way, it is easy to write the following FORWARD SUBSTITUTION ALGORITHM

$$\begin{aligned} x_1 &= \frac{b_1}{l_{11}} \\ x_i &= \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right), \quad i = 2, \ldots, n \end{aligned}$$

It needs n^2 arithmetic operations ($+$, $-$, $\cdot$, $/$). Indeed, to find x_i, $i > 1$, we need $2i - 1$ operations

- $i - 1$ multiplications to compute all terms $l_{ij} \cdot x_j$ of the sum;
- $i - 2$ addition to add together terms $l_{ij}x_j$ (and so compute $\sum_{j=1}^{i-1} l_{ij}x_j$);
- 1 subtraction to compute $b_i - \sum_{j=1}^{i-1} l_{ij}x_j$;
- 1 division of the previous result by l_{ii}.

Adding we get for i-th step $(i-1)+(i-2)+1+1 = 2i-1$. Note that this formula works also for $i = 1$, where we need only one operation (a /). So, the number of operations required by the algorithm is the sum of the number of operations required to find each x_i, $i = 1, \ldots, n$:

$$N_{\text{op}}(n) = \sum_{i=1}^{n} (2i - 1) = 2 \sum_{i=1}^{n} i - \sum_{i=1}^{n} 1 = 2 \, \frac{n(n+1)}{2} - n = n^2.$$

Exactly in the same manner, we can solve an upper triangular linear system. However, in this case, we start from the last equation and we go back until we reach the first one. The BACKWARD SUBSTITUTION ALGORITHM is

$$\begin{aligned} x_n &= \frac{b_n}{l_{nn}} \\ x_i &= \frac{1}{l_{ii}} \left(b_i - \sum_{j=i+1}^{n} l_{ij} x_j \right), \quad i = n-1, \ldots, 1 \end{aligned}$$

and requires the same amount of arithmetic operations as the forward substitution method.

It can be proved that both algorithms are stable (that is, they do not amplify too much rounding errors). However, if the condition number of the system is high, the obtained solution may be not accurate (see section 3.6). In the following figure 3.1, we show the absolute value of the error, component by component, of the solution of an upper triangular linear system $U\mathbf{x} = \mathbf{b}$ of dimension $n = 20$. The linear system has random elements with uniform distribution; $\mathbf{b}$ is choose in order to have the all 1's solution. As we can see, the solution of the linear system with high condition number has a larger error.

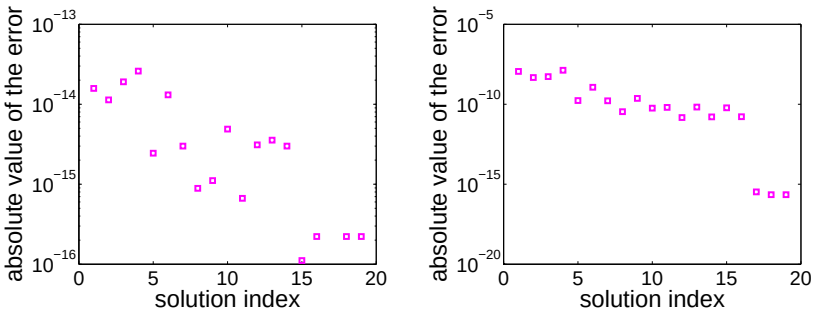


Figure 3.1: Error behaviour for the solution of an upper triangular linear system. On the left, we have a low condition number ($K(U) = 400$); on the right we have a high condition number ($K(U) = 3 \times 10^8$).

3.3 Gaussian elimination and LU factorization

Consider the linear system $A\mathbf{x} = \mathbf{b}$ where A is a square matrix of order n. Recall that if L is a non singular matrix then $A\mathbf{x} = \mathbf{b}$ has the same solutions of $LA\mathbf{x} = L\mathbf{b}$. We say that the two linear systems are equivalent.

Setting $A^{(1)} = A$ and $\mathbf{b}^{(1)} = \mathbf{b}$, the Gaussian Elimination (GE in the forward) reduces the given linear system $A^{(1)}\mathbf{x} = \mathbf{b}^{(1)}$ to another upper triangular equivalent one $A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$ using $n-1$ steps

$$\begin{array}{lll}
\text{Step 1:} & A^{(1)}\mathbf{x} = \mathbf{b}^{(1)} & \Leftrightarrow \quad \underbrace{L_1 A^{(1)}}_{:=A^{(2)}}\mathbf{x} = \underbrace{L_1\mathbf{b}^{(1)}}_{:=\mathbf{b}^{(2)}} \\
\text{Step 2:} & A^{(2)}\mathbf{x} = \mathbf{b}^{(2)} & \Leftrightarrow \quad \underbrace{L_2 A^{(2)}}_{:=A^{(3)}}\mathbf{x} = \underbrace{L_2\mathbf{b}^{(2)}}_{:=\mathbf{b}^{(3)}} \\
\cdots\cdots & & \cdots\cdots \\
\text{Step k:} & A^{(k)}\mathbf{x} = \mathbf{b}^{(k)} & \Leftrightarrow \quad \underbrace{L_k A^{(k)}}_{:=A^{(k+1)}}\mathbf{x} = \underbrace{L_k\mathbf{b}^{(k)}}_{:=\mathbf{b}^{(k+1)}} \\
\cdots\cdots & & \cdots\cdots \\
\text{Step n-1:} & A^{(n-1)}\mathbf{x} = \mathbf{b}^{(n-1)} & \Leftrightarrow \quad \underbrace{L_{n-1} A^{(n-1)}}_{:=A^{(n)}}\mathbf{x} = \underbrace{L_{n-1}\mathbf{b}^{(n)}}_{:=\mathbf{b}^{(n)}}
\end{array}$$

Let's see with some details the k-th step, $k = 1, \ldots, n-1$. At the beginning of this step the matrix $A^{(k)}$ looks like

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ \\ 0 & A_{22}^{(k)} \end{pmatrix}$$

where $A_{11}^{(k)}$ is an upper triangular square matrix of order $k-1$ (assuming $A_{11}^{(1)}$, $A_{12}^{(1)}$ empty and $A_{22}^{(1)} = A$). The aim of the k-th step is to zero the elements below the main diagonal in the k-th column of $A^{(k)}$. Assuming $a_{kk}^{(k)} \neq 0$, this is done by the following elementary row operations

$$\begin{array}{rcll}
l_{ik} & = & a_{ik}^{(k)} / a_{kk}^{(k)} & \\
\mathbf{r}_i^{(k+1)} & = & \mathbf{r}_i^{(k)} - l_{ik}\mathbf{r}_k^{(k)} & , \quad i = k+1, \cdots, n \\
b_i^{(k+1)} & = & b_i^{(k)} - l_{ik} b_k^{(k)} &
\end{array} \tag{3.1}$$

where $\mathbf{r}_i^{(k)}$ and $\mathbf{r}_k^{(k)}$ are the i-th and the k-th rows of $A_{22}^{(k)}$, respectively. See figure 3.2 for a picture of what's going on on matrix $A^{(k)}$ and on vector $\mathbf{b}^{(k)}$.

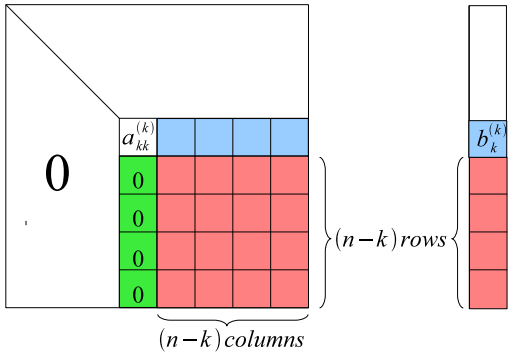


Figure 3.2: At the k-th step of Gaussian elimination, all $(n-k)^2$ elements inside the red area of matrix $A^{(k)}$ and all $n-k$ elements inside the red area of $\mathbf{b}^{(k)}$ have to be updated using the corresponding elements inside blue areas. Note that elements inside the green area of $A^{(k)}$ are set to zero by construction.

It is not difficult to prove that $A^{(k+1)} = L_k \cdot A^{(k)}$ where $L_k \in R^{n\times n}$ has always zero elements regardless in the main diagonal where there are all ones and in the elements below the diagonal in the k-th column which are $L_k(i,k) = -l_{ik}$. Since L_k is non singular (indeed, $|L_k| = 1$), the linear system $A^{(k+1)}\mathbf{x} = \mathbf{b}^{(k+1)} := L_k \cdot \mathbf{b}^{(k)}$ is equivalent to $A^{(k)}\mathbf{x} = \mathbf{b}^{(k)}$.

At the end of the $n-1$ step the matrix $A^{(n)}$ is upper triangular and the final linear system $A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$ can be solved using backward substitution. Furthermore, writing down explicitly all steps, we have

$$\underbrace{L_{n-1}\cdot L_{n-2}\cdot L_2\cdot L_1\cdot A}_{A^{(n)}}\cdot\mathbf{x} = \underbrace{L_{n-1}\cdot L_{n-2}\cdot L_2\cdot L_1\mathbf{b}}_{\mathbf{b}^{(n)}}$$

and so, defining $U := A^{(n)}$, we get

$$L_{n-1}\cdot L_{n-2}\cdot L_2\cdot L_1\cdot A = U \quad \text{or} \quad A = \underbrace{L_1^{-1}\cdot L_2^{-1}\cdots L_{n-2}^{-1}\cdot L_{n-1}^{-1}}_{:=L}\cdot U = LU$$

It can be shown with some matrix algebra that L is the lower triangular matrix with all ones in the main diagonal and, for $k = 1,\ldots,n-1$, has $L(i,k) = l_{ik}$, $i = 2,\ldots,n$. The previous factorization $A = LU$ is known as the *LU factorization* of A.

Gaussian elimination and LU factorization of A require $a_{kk}^{(k)} \neq 0$ for $k = 1,\ldots,n-1$. It is not clear whether this is the case just looking at A. The following theorem help us.

Theorem 3.5 *Let $A \in \mathbb{R}^{n\times n}$. Denote by A_k, $k = 1,\ldots,n-1$, the square matrices of order k built taking the first k upper left rows and columns. Then, there exists a unique LU factorization of A and Gaussian Elimination completes the final steps if and only all matrices A_k, $k = 1,\ldots,n-1$ are nonsingular. Moreover, if some A_k is singular, the LU factorization may exist but, if so, it is not unique.*

The theorem is interesting but, mostly for large n, hypotheses are too expensive to check. So, from a computational point of view, when solving $A\mathbf{x} = \mathbf{b}$ using Gaussian elimination, we simply apply the algorithm and looks whether it works or not.

Example 3.7 *Let's solve and compute the LU factorization of A for the linear system $A\mathbf{x} = \mathbf{b}$ where*

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 4 & 0 & -2 \\ -2 & 1 & 5 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 2 \\ -2 \\ 15 \end{pmatrix}$$

For this very small linear system it is easy to check hypothesis of theorem 3.5. We have

$$\begin{aligned} |A_1| &= |(1)| = 1 \\ |A_2| &= \left|\begin{pmatrix} 3 & 1 \\ 4 & 0 \end{pmatrix}\right| = -4 \\ |A_3| &= \left|\begin{pmatrix} 3 & 1 & -1 \\ 4 & 0 & -2 \\ -2 & 1 & 5 \end{pmatrix}\right| = -14 \end{aligned}$$

and, since all three determinant are different from zero, the Gaussian eliminations works until the end and the LU factorization exists and it is unique. Since $n = 3$, there are $n - 1 = 2$ steps of the Gaussian algorithm.

- STEP 1. *It is $k = 1$, $a_{11}^{(1)} = 3$. Let's update rows from $(k + 1) = 2$ to $n = 3$ (that is, rows with indexes $i = 2$ e $i = 3$).*
  - *Let $i = 2$. We have*

$$\begin{aligned} l_{21} &= a_{21}^{(1)}/a_{11}^{(1)} = 4/3 \\ \mathbf{r}_2^{(2)} &= \mathbf{r}_2^{(1)} - l_{21}\mathbf{r}_1^{(1)} = (4\ \ 0\ \ -2) - \frac{4}{3}\cdot(3\ \ 1\ \ -1) = \left(0\ \ -\frac{4}{3}\ \ -\frac{2}{3}\right) \\ b_2^{(2)} &= b_2^{(1)} - l_{21}b_1^{(1)} = -2 - \frac{4}{3}\cdot 2 = -\frac{14}{3} \end{aligned}$$

  - *Let $i = 3$. We have*

$$\begin{aligned} l_{31} &= a_{31}^{(1)}/a_{11}^{(1)} = -2/3 \\ \mathbf{r}_3^{(2)} &= \mathbf{r}_3^{(1)} - l_{31}\mathbf{r}_1^{(1)} = (-2\ \ 1\ \ 5) - \left(-\frac{2}{3}\right)\cdot(3\ \ 1\ \ -1) = \left(0\ \ \frac{5}{3}\ \ \frac{13}{3}\right) \\ b_3^{(2)} &= b_3^{(1)} - l_{31}b_1^{(1)} = 15 - \left(-\frac{2}{3}\right)\cdot 2 = \frac{49}{3} \end{aligned}$$

  So, matrix L_1 is

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ -l_{31} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -4/3 & 1 & 0 \\ 2/3 & 0 & 1 \end{pmatrix}$$

  As a check, we can obtain $A^{(2)}$ and $\mathbf{b}^{(2)}$, already computed, as

$$\begin{aligned} A^{(2)} &= L_1A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -4/3 & 1 & 0 \\ 2/3 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} 3 & 1 & -1 \\ 4 & 0 & -2 \\ -2 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 3 & 1 & -1 \\ 0 & -4/3 & -2/3 \\ 0 & 5/3 & 13/3 \end{pmatrix} \\ \mathbf{b}^{(2)} &= L_1\mathbf{b}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -4/3 & 1 & 0 \\ 2/3 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} 2 \\ -2 \\ 15 \end{pmatrix} = \begin{pmatrix} 2 \\ -14/3 \\ 49/3 \end{pmatrix} \end{aligned}$$

  and so the two linear systems $A^{(1)}\mathbf{x} = \mathbf{b}^{(1)}$ and $A^{(2)}\mathbf{x} = \mathbf{b}^{(2)}$ have the same solutions.

- STEP 2. *It is $k = 2$, $a_{22}^{(2)} = -4/3$. Let's update rows of $A^{(2)}$ from $k + 1 = 3$ to $n = 3$ (that is row with index $i = 3$).*
  - *Let i=3. We have*

$$\begin{aligned} l_{32} &= a_{32}^{(2)}/a_{22}^{(2)} = -5/4 \\ \mathbf{r}_3^{(3)} &= \mathbf{r}_3^{(2)} - l_{32}\mathbf{r}_2^{(2)} = (0\ \ 5/3\ \ 13/3) - (-5/4)\cdot(0\ \ -4/3\ \ -2/3) = (0\ \ 0\ \ 7/2) \\ b_3^{(3)} &= b_3^{(2)} - l_{32}b_2^{(2)} = 49/3 - (-5/4)\cdot(-14/3) = 21/2 \end{aligned}$$

So, matrix L_2 is

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5/4 & 1 \end{pmatrix}$$

and $A^{(3)}$ and $\mathbf{b}^{(3)}$ are

$$A^{(3)} = \begin{pmatrix} 3 & 1 & -1 \\ 0 & -4/3 & -2/3 \\ 0 & 0 & 7/2 \end{pmatrix}, \qquad \mathbf{b}^{(3)} = \begin{pmatrix} 2 \\ -14/3 \\ 21/2 \end{pmatrix}$$

Again, we may check that $A^{(3)} = L_2 \cdot A^{(2)}$ and $\mathbf{b}^{(3)} = L_2 \cdot \mathbf{b}^{(2)}$.

The linear system $A^{(3)}\mathbf{x} = \mathbf{b}^{(3)}$ produced by the Gaussian algorithm has thus

$$A^{(3)} = \begin{pmatrix} 3 & 1 & -1 \\ 0 & -4/3 & -2/3 \\ 0 & 0 & 7/2 \end{pmatrix}, \qquad \mathbf{b}^{(3)} = \begin{pmatrix} 2 \\ -14/3 \\ 21/2 \end{pmatrix}$$

which can be easily solved using backward substitution method giving

$$\begin{aligned} \frac{7}{2}x_3 = \frac{21}{2} &\quad \Rightarrow \quad x_3 = 3 \\ -\frac{4}{3}x_2 - \frac{2}{3}x_3 = -\frac{14}{3} &\quad \Rightarrow \quad x_2 = 2 \\ 3x_1 + x_2 - x_3 = 2 &\quad \Rightarrow \quad x_1 = 1 \end{aligned}$$

The LU factorization is easy to write now:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 4/3 & 1 & 0 \\ -2/3 & -5/4 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 3 & 1 & -1 \\ 0 & -4/3 & -2/3 \\ 0 & 0 & 7/2 \end{pmatrix}$$

The product $L \cdot U$ gives A as we can check taking the product.

Remark 3.4 *The Gauss Algorithm can be used to compute directly the solution of the linear system $A\mathbf{x} = \mathbf{b}$ without computing the LU factorization. For example, consider*

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

As usual, we have

$$(A|\mathbf{b}) = \left(\begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 2 & 1 & 0 & 2 \\ 3 & 2 & 1 & 3 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right)$$

The first three columns of the last 3×4 matrix is U. Moreover, since we have done the same operations on A and $\mathbf{b}$, the linear system

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

is equivalent to the original one. Thus, we can solve it using the back substitution algorithm; we find $x_3 = 0$, $x_2 = 0$, $x_1 = 1$.

3.3.1 Computational cost

Gaussian elimination uses $n-1$ steps to produce the final upper triangular linear system. So, it's cost is the sum of the costs of each one of these steps. Let's find the cost of the k-th step, $k = 1, \ldots, n-1$. Looking at equation (3.1), since the index i goes from $k+1$ to n, there are $n-k$ equally costing steps to do. So, let's see the cost of the i-th step.

- First, we have to compute one division in order to find $l_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}$.
- Second, we update all $n-k$ elements of the i-th row of matrix $A^{(k)}$ accordingly to the equation $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}$, $j = k+1, \ldots, n$. Each one of these updates require two operations, a multiplication and a subtraction. Note that no operations are needed for $a_{ik}^{(k+1)}$ since it is set to zero by construction.
- Third, we compute $b_i^{(k+1)} = b_i^{(k)} - l_{ik}b_k^{(k)}$ which needs again two operations.

Thus, the i-th step requires $1 + 2(n-k) + 2 = 2(n-k) + 3$ operations to complete. Finally, the total amount of operations for the k-th step to be completed is $(n-k)[2(n-k)+3]$. Now, the cost of Gaussian elimination can be obtained summing the costs of each step:

$$N_{\mathrm{op}}(n) = \sum_{k=1}^{n-1} \{\, (n-k)[2(n-k)+3] \,\} \overset{(h=n-k)}{=} \sum_{h=1}^{n-1} h(2h+3) = \frac{n(4n^2+3n-7)}{6}$$

For large values of n, $N_{\mathrm{op}}(n) \approx 2n^3/3$. This is the work required to obtain the upper triangular linear system $A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$. To solve this linear system we need n^2 more operations, which is, for large n, a small amount compared to the previous ones.

3.3.2 Applications of the LU factorization

Using the LU factorization we are able to solve some interesting, and not so easy, problems.

- The computation of the determinant of the matrix A is a difficult task. However, if we know the LU factorization of A, due to Binet, we have

  $$|A| = |L \cdot U| = |L| \cdot |U| = |U| = \prod_{i=1}^{n} u_{ii}$$

  since $|L| = 1$. Then, it is also easy to get, for example, $|A^2| = |A|^2$ and, assuming A non singular, $|A^{-1}| = 1/|A|$, $|A^{-3}| = 1/|A|^3$.

- Sometimes we have to solve a large number of linear systems each one with the same, non singular, matrix A and different right hand sides: $A\mathbf{x} = \mathbf{b}_i$, $i = 1, \ldots, p$ with $p \gg 1$. To solve one of these linear systems we need about $2n^3/3$ operations. So, to solve all we need about $2pn^3/3$ operations. It is possible to reduce this number of operations just taking into account that all linear system have the same matrix A. The idea is the following. First, we find the LU factorization of the matrix A with a cost of $2n^3/3$ operations. Second, we solve, one at a time, all the p linear systems as

  $$A\mathbf{x}_i = \mathbf{b}_i \quad \Leftrightarrow \quad LU\mathbf{x}_i = \mathbf{b}_i \quad \Leftrightarrow \quad \begin{array}{lll} 1. & \text{solve for } \mathbf{y}_i & L\mathbf{y}_i = \mathbf{b}_i \\ 2. & \text{solve for } \mathbf{x}_i & U\mathbf{x}_i = \mathbf{y}_i \end{array}$$

  where $\mathbf{x}_i$, $i = 1, \ldots, p$ is the solution of the i-th linear system. To solve $A\mathbf{x}_i = \mathbf{b}_i$, we need only $2n^2$ operations: n^2 for $L\mathbf{y}_i = \mathbf{b}_i$ plus n^2 for $U\mathbf{y}_i = \mathbf{b}_i$. To solve all p linear systems we need $2pn^2$ operations. Taking into account the number of operations for the LU factorization, the overall amount of operations is $2n^3/3 + 2pn^2$. The ratio

  $$\frac{2n^3/3 + p2n^2}{2pn^3/3} = \frac{1}{p} + \frac{3}{n} \ll 1$$

  says that using the LU factorization there is a great saving of the number of operations (and thus of the computational time).

- Consider a non singular matrix $A \in \mathbb{R}^{n\times n}$. From the equation $A\cdot A^{-1} = I_n$, the inverse $A^{-1} = (\mathbf{c}_1|\ldots|\mathbf{c}_n)$, where $\mathbf{c}_j$ is the j-th column of A^{-1}, can be computed solving n linear systems $A\mathbf{c}_j = \mathbf{e}_j$, $j = 1\ldots,n$ where $\mathbf{e}_j$ is a column vector with all zeros but a one in the j-th position. This work can be done as stated in the previous point.

Example 3.8 *The LU factorization of the matrix A gives $|U| = -3$. Then, we have*

$$\begin{aligned} |A| &= |L\cdot U| = |L|\cdot|U| = 1\cdot(-3) = -3 \\ |A^2| &= |A|^2 = (-3)^2 = 9 \\ |A^{-4}| &= 1/|A|^4 = 1/(-3)^4 = 1/81 \end{aligned}$$

Example 3.9 *Using the LU factorization of the matrix A, compute the inverse of the matrix*

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ -3 & -2 & 0 \end{pmatrix}$$

We first compute the LU factorization of A. Since the matrix has order $n = 3$, we have $n-1 = 2$ steps. Let $A^{(1)} = A$.

- *First step: we zero the elements below $a_{11}^{(1)} = 1$. We get easily $l_{21} = a_{21}^{(1)}/a_{11}^{(1)} = 2/1 = 2$ and $l_{31} = a_{31}^{(1)}/a_{11}^{(1)} = -3/1 = -3$ and so*

$$A^{(1)} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ -3 & -2 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 4 & 3 \end{pmatrix} := A^{(2)}, \quad \textit{with} \quad L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

- *Second step: we zero the elements below $a_{22}^{(2)} = 1$. We have $l_{31} = a_{31}^{(2)}/a_{22}^{(2)} = 4/1 = 4$ and so*

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 4 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -5 \end{pmatrix} := A^{(3)}, \quad \textit{with} \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{pmatrix}$$

Thus, the matrices L and U are

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 4 & 1 \end{pmatrix}, \quad U = A^{(3)} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -5 \end{pmatrix}$$

Since $|A| = |U| = -5 \neq 0$, matrix A is non singular and thus it has A^{-1}. Let's denote the inverse of A as

$$A^{-1} = \left(\begin{array}{c|c|c} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{array}\right) = (\mathbf{c}_1 \mid \mathbf{c}_2 \mid \mathbf{c}_3)$$

where $\mathbf{c}_j$, $j = 1,2,3$ is the j-th column of A^{-1}. From $A\cdot A^{-1} = I_3$, the identity matrix of order 3, we have

$$\begin{aligned} A\cdot A^{-1} = A\cdot(\mathbf{c}_1 \mid \mathbf{c}_2 \mid \mathbf{c}_3) &= (A\cdot\mathbf{c}_1 \mid A\cdot\mathbf{c}_2 \mid A\cdot\mathbf{c}_3) \\ &= (\ \mathbf{e}_1 \ \mid \ \mathbf{e}_2 \ \mid \ \mathbf{e}_3\ ) = I_3 \end{aligned}$$

where $\mathbf{e}_j$, $j = 1,2,3$ is the vector with all 0 but 1 in the j-th position (starting from above). Thus, we must have $A\cdot\mathbf{c}_j = \mathbf{e}_j$, $j = 1,2,3$. These are three linear system with the same matrix A but different right hand sides. So, we can use the LU factorization to solve them efficiently.

- *Solve* $A \cdot \mathbf{c}_1 = \mathbf{e}_1$. *We have*

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ -3 & -2 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

which, recalling that $A = L \cdot U$, *is solved in two steps.*

1. *Setting* $\mathbf{y}_1 = (y_1\ ,\ y_2\ ,\ y_3)^T$, *we solve the lower triangular linear system* $L\mathbf{y}_1 = \mathbf{e}_1$*:*

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 4 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{cases} y_1 & = & 1 \\ 2y_1 + y_2 & = & 0 \\ -3y_1 + 4y_2 + y_3 & = & 0 \end{cases}$$

So, we get $y_1 = 1$, $y_2 = -2$, $y_3 = 11$.

2. *We solve the upper triangular linear system* $U\mathbf{c}_1 = \mathbf{y}_1$*:*

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -5 \end{pmatrix} = \begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 11 \end{pmatrix} \quad \Leftrightarrow \quad \begin{cases} c_{11} + 2c_{21} + c_{31} & = & 1 \\ c_{21} + 2c_{31} & = & -2 \\ -5c_{31} & = & 11 \end{cases}$$

and thus we get $c_{31} = -11/5$, $c_{21} = 12/5$, $c_{11} = -8/5$. *That is, the first column of* A^{-1} *is* $\mathbf{c}_1 = (-11/5\ ,\ 12/5\ ,\ -8/5)^T$

- *Solve* $A \cdot \mathbf{c}_2 = \mathbf{e}_2$. *We have*

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ -3 & -2 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_{12} \\ c_{22} \\ c_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Proceeding in the same way as in the previous step, we find $\mathbf{c}_2 = (2/5\ ,\ -3/5\ ,\ 4/5)^T$.

- *Solve* $A \cdot \mathbf{c}_3 = \mathbf{e}_3$. *We have*

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ -3 & -2 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_{13} \\ c_{23} \\ c_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Proceeding in the same way as in the first step, we find $\mathbf{c}_2 = (-3/5\ ,\ 2/5\ ,\ -1/5)^T$.

So, the inverse of A *is*

$$A^{-1} = \begin{pmatrix} -8/5 & 2/5 & -3/5 \\ 12/5 & -3/5 & 2/5 \\ -11/5 & 4/5 & -1/5 \end{pmatrix}$$

3.3.3 Drawbacks and pivoting strategy

There are some problems with Gaussian Elimination. The first one: Gaussian Elimination is not able to reach the end if, for some k, we have $a_{kk}^{(k)} = 0$. The second one: Gaussian Elimination is not stable.

Let's see how to solve the first one. Since the order of the equations is not important, if $a_{kk}^{(k)} = 0$ for some $k = 1, \ldots, n-1$, then we can exchange k-th row with one of the following rows (with index $i > k$). If A is a non singular matrix, then such i-th row exists for sure. Indeed, writing $A^{(k)}$ as

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}$$

with $A_{11}^{(k)}$ upper triangular, we have $|A| = |A^{(k)}| = |A_{11}^{(k)}| \cdot |A_{22}^{(k)}|$. Now, it is $|A_{11}^{(k)}| \neq 0$ because all the previous pivotal elements are different from zero; thus, it must be $|A_{22}^{(k)}| \neq 0$ which cannot be the case if all elements below $a_{kk}^{(k)}$ are zeros.

Example 3.10 *The Gauss Algorithm does not work on the matrix*

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

since $a_{11} = 0$. *We have to exchange row* 1 *with row* 2. *This can be done by left multiply* A *(and the right hand side* $\mathbf{b}$ *of the linear system* $A\mathbf{x} = \mathbf{b}$*) by a permutation matrix* P*:*

$$P \cdot A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

It is easy to see that PA *admits an LU factorization with* $L = I_2$ *and* $U = PA$.

The second problem has no solution. It is possible, however, to modify Gaussian elimination in order to avoid a large grow of the rounding errors for almost all practical linear systems. There are to ways.

- PARTIAL PIVOTING: at k-th step of the Gaussian elimination, we search for the element of maximum modulus among $a_{i,k}^{(k)}$, $i = k, \ldots, n$. Let $a_{ik}^{(k)}$ be this one. Then, we exchange the k-th row wit i-th row. Doing so, at each step the element of maximum modulus is the pivotal element. We need $n - k$ comparisons.

- TOTAL PIVOTING: we search in the matrix $A_{22}^{(k)}$ for the element with the largest absolute value. Then, exchanging rows and columns, we make it as the pivotal element. We need about $(n - k)^2$ comparisons.

Total pivoting, usually, gives better results; however, it is more computationally expensive. Moreover, partial pivoting gives, for all kind of linear systems, good results (i.e., does not propagate too much rounding errors). Just to have an idea, consider the linear system $A\mathbf{x} = \mathbf{b}$. Matrix $A \in \mathbb{R}^{n \times n}$, $n = 100$ has uniform random entries; $\mathbf{b}$ is chosen to have the all ones solution. That stated, we solve the system with and without partial pivoting. Figure 3.3 shows the results. On the left, for a linear system with $K(A) = 700$, we plot the absolute value of the error, component by component. We can see that partial pivoting gives a better solution. On the right of the same figure, we plot the norm of the errors, versus the condition number $K(A)$, for a large number of simulations of the same kind of linear systems. Again, partial pivoting gives better result regardless the value of $K(A)$.

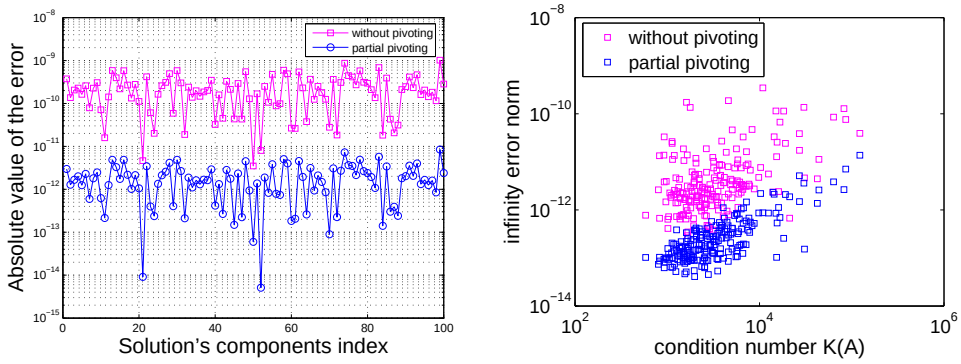


Figure 3.3: Behaviour of the absolute value of the error (left) and of the norm of the error (right) for a large number of simulations.

Finally, it is possible to see the better behaviour of partial pivoting with a numerical example.

Example 3.11 (from the book of Comincioli) *Consider the linear system* $A\mathbf{x} = \mathbf{b}$ *with*

$$A = \begin{pmatrix} 0.005 & 1 \\ 1 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$$

Setting $\mathbf{x} = (\ x_1\ x_2\ )^T$, *we find in exact arithmetic*

$$x_1 = \frac{100}{199} = 0.50251256281407... \qquad x_2 = \frac{99}{199} = 0.49748743718593...$$

Let's see what happens if we work using $\mathbb{F}(10, 2, , -4, 4)$. *We have*

$$(A|\mathbf{b}) = \left(\begin{array}{cc|c} 0.005 & 1 & 0.5 \\ 1 & 1 & 1 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 0.005 & 1 & 0.5 \\ 0 & -200 & -99 \end{array}\right)$$

since, using the floating point arithmetic we have,

$$[1 \otimes (-200)] \oplus 1 = -200 \oplus 1 = fl(-199) = -200$$

because we only have two decimal places; furthermore, we get

$$[0.5 \otimes (-200)] \oplus 1 = -100 \oplus 1 = -99$$

Proceeding with the back substitution algorithm, we find

$$x_2 = -99 \oslash (-199) = fl(99/199) = fl(0.4975...) = 0.50$$

and thus

$$x_1 = [0.5 \ominus (1 \otimes x_2)] \oslash 0.005 = [0.5 \ominus 0.5] \oslash 0.005 = 0 \oslash 0.005 = 0$$

So, we have completely miss x_1.

Let's do the same by a first exchange of the rows. Thus, we solve the linear system $PA\mathbf{x} = P\mathbf{b}$ *instead of the original one where* P *is the permutation matrix used to exchange the rows.*

$$\left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0.005 & 1 & 0.5 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & 1 & 0.5 \end{array}\right)$$

Proceeding as before, we find $x_2 = 0.5$ *and* $x_2 = 0.5$ *which is the best possible answer using the floating point arithmetic we have. Note that exchanging the rows have made the pivotal element of the first column as greater as possible.*

3.3.4 The $PA = LU$ factorization

The solution of $A\mathbf{x} = \mathbf{b}$ with partial pivoting may require, at the k-th step, to exchange the row $\mathbf{r}_k^{(k)}$ with the row $\mathbf{r}_i^{(k)}$, $i > k$. Now, let P by the (non singular) matrix obtained from the identity matrix where only rows k and i are exchanged (i.e., the k-th row of P is the i-th row of I_n and the $i-$th row of P is the k-th of I_n). Then, the matrix $P \cdot A$ has only rows k and i exchanged. For example, to exchange rows 2 and 4 of some matrix A we write

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 1 & 0 \\ 0 & \mathbf{1} & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{pmatrix}$$

That said, the k-th step of the Gaussian elimination is change as follows:

- find the element $a_{ik}^{(k)}$ of greatest absolute value below $a_{kk}^{(k)}$ in the k-th column of $A^{(k)}$.
- exchange rows k and i of matrix $A^{(k)}$ (and, of course, of right hand side $\mathbf{b}^{(k)}$) using a permutation matrix P_k:

$$A^{(k)}\mathbf{x} = \mathbf{b}^{(k)} \quad \Leftrightarrow \quad P_k A^{(k)}\mathbf{x} = P_k\mathbf{b}^{(k)}$$

- zero the elements in the k-th column below the element in the main diagonal using the matrix L_k.

Thus, the k-th step is now

$$L_k \cdot P_k \cdot A^{(k)}\mathbf{x} = L_k \cdot P_k \cdot \mathbf{b}^{(k)}$$

and the overall Gaussian elimination looks like

$$\underbrace{L_{n-1}P_{n-1} \cdot L_{n-2}P_{n-2} \cdot L_2P_2 \cdot L_1P_1 \cdot A}_{U}\mathbf{x} = L_{n-1}P_{n-1} \cdot L_{n-2}P_{n-2} \cdot L_2P_2 \cdot L_1P_1 \cdot \mathbf{b}$$

where U is an upper triangular linear system. Thus, we have

$$U = L_{n-1}P_{n-1} \cdot L_{n-2}P_{n-2} \cdot L_2P_2 \cdot L_1P_1 \cdot A$$

With some amount of linear algebra, this equation can be rewritten as $PA = LU$ where $P = P_{n-1}P_{n-2} \dots P_2P_1$ and L and U are the usual matrices obtained from Gaussian elimination applied to the matrix PA instead of A.

3.4 Cholesky factorization

The Cholesky factorization is based on the following theorem.

Theorem 3.6 (Cholesky factorization) *Let A be a symmetric, positive definite matrix of order n. Then, there is a unique lower triangular matrix L with positive entries on the main diagonal such that $A = L \cdot L^T$.*

We skip the proof of the theorem which can be done, for example, by mathematical induction on n. Instead, we give one possible algorithm to compute L column by column. Just writing down the product $A = L \cdot L^T$, we get

for $j = 1 : n$
  $l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$
  for $i = j + 1 : n$
    $l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} \cdot l_{jk}\right) / l_{jj}$
  end
end

where the result of a sum is zero if the ending value of its index is smaller then the corresponding starting value. This algorithm needs about $n^3/3$ arithmetic operations ($+$, $-$, $\times$, $/$) plus n square roots.

It can be proved that Cholesky factorization is a stable algorithm, that is it does not propagate much rounding errors. Recall that this may not be the case for LU factorization, even if we use pivoting strategy.

Finally, it is interesting to note that if the factorization algorithm fails because some l_{ii} cannot be computed, then the matrix A is not positive define. Say in a different way, the previous algorithm may be used to check whether a matrix is positive define.

Let's see an example which is also useful to understand in which way equations of the algorithm come from.

Example 3.12 *Consider the symmetric matrix*

$$A = \begin{pmatrix} 3 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

Note that A is positive define since, using the Gershgoring theorem, it has all the eigenvalues positive. Thus, the Cholesky factorization exists. Now, since A has order $n = 4$, the matrix L is

$$L = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix}$$

where l_{11}, l_{22}, l_{33} and l_{44} are positive numbers. Let's compute l_{ij}.

- *Compute the first column of L ($j=1$). Writing $L \cdot L^T = A$ we get*

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \cdot \begin{pmatrix} l_{11} & l_{21} & l_{31} & l_{41} \\ 0 & l_{22} & l_{32} & l_{42} \\ 0 & 0 & l_{33} & l_{43} \\ 0 & 0 & 0 & l_{44} \end{pmatrix} = \begin{pmatrix} 3 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 3 \end{pmatrix} \tag{3.2}$$

from which, equating column $j=1$ of $L \cdot L^T$ and A, it follows

$$\begin{aligned} l_{11} \cdot l_{11} &= a_{11} \quad \Rightarrow \quad l_{11} = \sqrt{a_{11}} = \sqrt{3} \\ l_{21} \cdot l_{11} + l_{22} \cdot 0 &= a_{21} \quad \Rightarrow \quad l_{21} = a_{21}/l_{11} = 1/\sqrt{3} \\ l_{31} \cdot l_{11} + l_{32} \cdot 0 + l_{33} \cdot 0 &= a_{31} \quad \Rightarrow \quad l_{31} = a_{31}/l_{11} = 0/\sqrt{3} = 0 \\ l_{41} \cdot l_{11} + l_{42} \cdot 0 + l_{43} \cdot 0 + l_{44} \cdot 0 &= a_{41} \quad \Rightarrow \quad l_{41} = a_{41}/l_{11} = 0/\sqrt{3} = 0 \end{aligned}$$

So, at the end of this step we have computed the first column of L which now looks as

$$L = \begin{pmatrix} \sqrt{3} & 0 & 0 & 0 \\ 1/\sqrt{3} & l_{22} & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & l_{42} & l_{43} & l_{44} \end{pmatrix}$$

- *Compute the second column of L (j=2). Consider again equation (3.2). Equating column $j=2$ of the matrices $L \cdot L^T$ and A, considering that A is symmetric (and so we can equate only elements with row index greater or equal to the column index $j=2$), we get*

$$\begin{aligned} l_{21} \cdot l_{21} + l_{22} \cdot l_{22} &= a_{22} \quad \Rightarrow \quad l_{22} = \sqrt{a_{22} - l_{21} \cdot l_{21}} \\ l_{31} \cdot l_{21} + l_{32} \cdot l_{22} + l_{33} \cdot 0 &= a_{32} \quad \Rightarrow \quad l_{32} = \left(a_{32} - l_{31} \cdot l_{21}\right)/l_{22} \\ l_{41} \cdot l_{21} + l_{42} \cdot l_{22} + l_{43} \cdot 0 + l_{44} \cdot 0 &= a_{42} \quad \Rightarrow \quad l_{42} = \left(a_{42} - l_{41} \cdot l_{21}\right)/l_{22} \end{aligned}$$

Recalling that, $l_{21} = 1/\sqrt{3}$, $l_{31} = 0$ and $l_{41} = 0$, we easily find from the three previous equations $l_{22} = \sqrt{8/3}$, $l_{32} = \sqrt{3/8}$ and $l_{42} = 0$. So, at the end of this step we have computed the first and the second columns of L which now looks as

$$L = \begin{pmatrix} \sqrt{3} & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & \sqrt{\frac{8}{3}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{8}} & l_{33} & 0 \\ 0 & 0 & l_{43} & l_{44} \end{pmatrix}$$

- *Compute the third column of L (j=3). Consider once more equation (3.2). Equating column $j=3$ of matrices $L \cdot L^T$ and A, considering that A is symmetric (and so we can equate only elements with row index greater or equal to the column index $j=3$), we get*

$$\begin{aligned} l_{31} \cdot l_{31} + l_{32} \cdot l_{32} + l_{33} \cdot l_{33} &= a_{33} \quad \Rightarrow \quad l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \\ l_{41} \cdot l_{31} + l_{42} \cdot l_{32} + l_{43} \cdot l_{33} + l_{44} \cdot 0 &= a_{43} \quad \Rightarrow \quad l_{43} = \left(a_{43} - l_{41} \cdot l_{31} - l_{42} \cdot l_{33}\right)/l_{33} \end{aligned}$$

Recalling the already computed values of L, we find from the two previous equations $l_{33} = \sqrt{21/8}$, $l_{43} = \sqrt{8/21}$. So, at the end of this step we have computed the first second and third columns of L which now looks as

$$L = \begin{pmatrix} \sqrt{3} & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & \sqrt{\frac{8}{3}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{8}} & \sqrt{\frac{21}{8}} & 0 \\ 0 & 0 & \sqrt{\frac{8}{21}} & l_{44} \end{pmatrix}$$

- *Compute the fourth column of L ($j = 4$). From equation (3.2) again, equating columns $j = 4$ of matrices $L \cdot L^T$ and A we get*

$$l_{41} \cdot l_{41} + l_{42} \cdot l_{42} + l_{43} \cdot l_{43} + l_{44} \cdot l_{44} = a_{44} \quad \Rightarrow \quad l_{44} = \sqrt{a_{44} - l_{41}^2 - l_{42}^2 - l_{43}^2}$$

which gives $l_{44} = \sqrt{55/21}$. So, finally, we have

$$L = \begin{pmatrix} \sqrt{3} & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & \sqrt{\frac{8}{3}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{8}} & \sqrt{\frac{21}{8}} & 0 \\ 0 & 0 & \sqrt{\frac{8}{21}} & \sqrt{\frac{55}{21}} \end{pmatrix}$$

3.5 Conditioning of a linear system

We start with a useful definition.

Definition 3.6 *For any induced matrix norm, the condition number of the square, non singular, matrix A is*

$$K(A) = \|A\| \cdot \left\|A^{-1}\right\|$$

If A is singular, we set $K(A) = +\infty$.

The condition number fulfills some useful properties.

Theorem 3.7 *Let A be a non singular matrix of order n. Then, we have*

(a) $K(A) \geq 1$ for all matrices A.

(b) $K(A^{-1}) = K(A)$ and $K(\alpha\, A) = K(A)$ for each $\alpha \neq 0$.

Proof. Let prove only (a). We have

$$1 = \|I_n\| = \left\|A \cdot A^{-1}\right\| \leq \|A\| \cdot \left\|A^{-1}\right\| = K(A).$$

where I_n is the identity matrix of order n. □

Theorem 3.8 *Let A be an orthogonal matrix of order n. Then $K_2(A) = 1$.*

Proof. We have

$$\|A\|_2 = \sqrt{\rho(\, A \cdot A^T\,)} = \sqrt{\rho(\, A \cdot A^{-1}\,)} = \sqrt{\rho(\, I_n\,)} = 1$$

because $A^T = A^{-1}$ since A is orthogonal. Also, it is

$$\left\|A^{-1}\right\|_2 = \left\|A^T\right\|_2 = \sqrt{\rho(\, A^T \cdot (\, A^T\,)^T\,)} = \sqrt{\rho(\, A^{-1} \cdot A\,)} = 1$$

So, we have $K_2(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = 1 \cdot 1 = 1$ and the proof is complete. □

Theorem 3.9 *Let A be a symmetric, positive define matrix of order n. Then*

$$K_2(A) = \frac{\lambda_{max}}{\lambda_{min}}.$$

Example 3.13 (Hilbert matrices) *A Hilbert matrix H of order n is define as*

$$h_{ij} = \frac{1}{i+j-1}, \quad i,j = 1,\dots,n$$

This matrices are symmetric since $h_{ji} = h_{ij}$. It can also be proved that they are positive definite. Their condition number is high even for small n:

n	5	10	15
$K(H)$	$5 \cdot 10^5$	$2 \cdot 10^{13}$	$8 \cdot 10^{17}$

Now consider the linear system $A\mathbf{x} = \mathbf{b}$. A small change $\delta\mathbf{b}$ in the right hand side produces a small change $\delta\mathbf{x}$ in the solution: $A(\mathbf{x}+\delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$. That is, $A\delta\mathbf{x} = \delta\mathbf{b}$. Let's find a relation between $\delta\mathbf{b}$ and $\delta\mathbf{x}$. We have, with respect to some compatibility vector norm,

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} = \frac{\|A^{-1}\cdot\delta\mathbf{b}\|}{\|\mathbf{x}\|} \overset{(1)}{\leq} \frac{\|A^{-1}\|\cdot\|\delta\mathbf{b}\|}{\|\mathbf{b}\|/\|A\|} = \underbrace{\|A\|\cdot\|A^{-1}\|}_{K(A)} \cdot \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

where $K(A) = \|A\|\cdot\|A^{-1}\|$ is, by definition, the condition number of the matrix A. For the denominator of inequality (1), note that $\|\mathbf{b}\| = \|A\cdot\mathbf{x}\| \leq \|A\|\cdot\|\mathbf{x}\|$ and so $\|\mathbf{x}\| \geq \|A\|/\|\mathbf{b}\|$. If $K(A)$ is large we say that the linear system (and the matrix A) is ill-conditioned. In this case, small $\delta\mathbf{b}$ may, even if not necessarily, produce large $\delta\mathbf{x}$. That is, small perturbation may completely change the solution. If $K(A)$ is small, the linear system (and the matrix A) is well-conditioned; small perturbations on $\mathbf{b}$ does not change too much $\mathbf{x}$.
As a final result, that we give without proof, we have the following theorem.

Theorem 3.10 *Let A be a non singular matrix of order n. Let $\mathbf{x}+\delta\mathbf{x}$ be the solution*

$$(A+\delta A)(\mathbf{x}+\delta\mathbf{x}) = \mathbf{b}+\delta\mathbf{b}$$

Assuming also that $\mathbf{b} \neq \mathbf{0}$ and that, for some compatible matrix norm, $\|\delta A\| < 1/\left\|A^{-1}\right\|$, we have

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{K(A)}{1 - K(A)\frac{\|\,\delta A\,\|}{\|\,A\,\|}} \left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\,\delta A\,\|}{\|\,A\,\|} \right)$$

3.5.1 Error bound in the Gaussian Algorithm

Consider the solution of the linear system $A\mathbf{x} = \mathbf{b}$ with the Gaussian Algorithm. Denote by $\mathbf{x}$ the exact solution and by $\hat{\mathbf{x}}$ the actually computed solution by the algorithm.

Typically, the Gaussian Algorithm gives a small norm for the residual vector $\hat{\mathbf{r}} = \mathbf{b} - A\hat{\mathbf{x}}$. Recall that for the exact solution $\mathbf{x}$ it is $\mathbf{r} = \mathbf{b} - A\,\mathbf{x} = \mathbf{0}$. So, it seems a good idea to take $\|\,\hat{\mathbf{r}}\,\|$ as a measure of the goodness of the solution $\hat{\mathbf{x}}$. Let's see if this is indeed the case.

Writing $A\hat{\mathbf{x}} = \mathbf{b} - \hat{\mathbf{r}}$ we can say that $\hat{\mathbf{x}}$ is the solution of our linear system where the right hand side is just a little perturbation of $\mathbf{b}$ by the term $\delta\mathbf{b} = -\hat{\mathbf{r}}$. Using the previous analysis, the error norm $\|\mathbf{e}\| = \|\mathbf{x} - \hat{\mathbf{x}}\|$ satisfies

$$\frac{\|\,\mathbf{e}\,\|}{\|\,\mathbf{x}\,\|} \leq K(A)\cdot\frac{\|\,\hat{\mathbf{r}}\,\|}{\|\,\mathbf{b}\,\|}$$

So, if $K(A)$ is large (that is, the linear system is ill-conditioned) we *may* have a large error even if the relative residual $\|\,\hat{\mathbf{r}}\,\|\,/\,\|\,\mathbf{b}\,\|$ is small. On the other hand, if the linear system is well-conditioned the test on the residual works pretty well.

Example 3.14 *Consider the linear system $H\mathbf{x} = \mathbf{b}$ where H is the Hilbert matrix of order $n = 10$. Let choose $\mathbf{b}$ in order to have the exact solution $\mathbf{x}$ of all ones. Using Matlab, we find $\|\,\hat{\mathbf{r}}\,\|\,/\,\|\,\mathbf{b}\,\| \approx 10^{-16}$ and $\|\,\mathbf{e}\,\|\,/\,\|\,\mathbf{x}\,\| \approx 10^{-4}$ which is much greater then the residual. This is due to the high $K(A) \approx 10^{13}$.*

3.6 Appendix

In this appendix we can see a complete Gaussian elimination working on a system $A\mathbf{x} = \mathbf{b}$ of order $n = 4$. We have $n - 1 = 4 - 1 = 3$ steps.

- Step 1: $A^{(1)}\mathbf{x} = \mathbf{b}^{(1)} \Leftrightarrow L_1 \cdot A^{(1)}(\mathbf{x}) = L_1 \cdot \mathbf{b}^{(1)}$ where L_1 zeros all elements below $a_{11}^{(1)}$ by elementary rows operations using

$$\left.\begin{array}{rcl} l_{21} & = & a_{21}^{(1)}/a_{11}^{(1)} \\ l_{31} & = & a_{31}^{(1)}/a_{11}^{(1)} \\ l_{41} & = & a_{41}^{(1)}/a_{11}^{(1)} \end{array}\right\} \quad \Rightarrow \quad L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{41} & 0 & 0 & 1 \end{pmatrix}$$

Let's compute $A^{(2)} = L_1 \cdot A^{(1)}$:

$$\begin{aligned} L_1 \cdot A^{(1)} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{41} & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{34}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ -l_{21} \cdot a_{11}^{(1)} + a_{21}^{(1)} & -l_{21} \cdot a_{12}^{(1)} + a_{22}^{(1)} & -l_{21} \cdot a_{13}^{(1)} + a_{23}^{(1)} & -l_{21} \cdot a_{14}^{(1)} + a_{24}^{(1)} \\ -l_{31} \cdot a_{11}^{(1)} + a_{31}^{(1)} & -l_{31} \cdot a_{12}^{(1)} + a_{32}^{(1)} & -l_{31} \cdot a_{13}^{(1)} + a_{33}^{(1)} & -l_{31} \cdot a_{14}^{(1)} + a_{34}^{(1)} \\ -l_{41} \cdot a_{11}^{(1)} + a_{41}^{(1)} & -l_{41} \cdot a_{12}^{(1)} + a_{42}^{(1)} & -l_{41} \cdot a_{13}^{(1)} + a_{43}^{(1)} & -l_{41} \cdot a_{14}^{(1)} + a_{44}^{(1)} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & -l_{21} \cdot a_{12}^{(1)} + a_{22}^{(1)} & -l_{21} \cdot a_{13}^{(1)} + a_{23}^{(1)} & -l_{21} \cdot a_{14}^{(1)} + a_{24}^{(1)} \\ 0 & -l_{31} \cdot a_{12}^{(1)} + a_{32}^{(1)} & -l_{31} \cdot a_{13}^{(1)} + a_{33}^{(1)} & -l_{31} \cdot a_{14}^{(1)} + a_{34}^{(1)} \\ 0 & -l_{41} \cdot a_{12}^{(1)} + a_{42}^{(1)} & -l_{41} \cdot a_{13}^{(1)} + a_{43}^{(1)} & -l_{41} \cdot a_{14}^{(1)} + a_{44}^{(1)} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & a_{14}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{pmatrix} = A^{(2)} \end{aligned}$$

where $a_{1,j}^{(2)} = a_{1,j}^{(1)}$, $j = 1, \ldots, 4$ and $a_{i,j}^{(2)} = a_{ij}^{(1)} - l_{ij} \cdot a_{1j}^{(1)}$, $i, j = 2, \ldots, 4$,

- Step 2: $A^{(2)}\mathbf{x} = \mathbf{b}^{(2)} \Leftrightarrow L_2 \cdot A^{(2)}(\mathbf{x}) = L_2 \cdot \mathbf{b}^{(2)}$ where L_2 zeros all elements below $a_{22}^{(2)}$ by elementary rows operations using

$$\left.\begin{array}{rcl} l_{32} & = & a_{32}^{(1)}/a_{22}^{(2)} \\ l_{42} & = & a_{42}^{(2)}/a_{22}^{(2)} \end{array}\right\} \quad \Rightarrow \quad L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -l_{32} & 1 & 0 \\ 0 & -l_{42} & 0 & 1 \end{pmatrix}$$

Let's compute $A^{(3)} = L_2 \cdot A^{(2)}$:

$$\begin{aligned} L_2 \cdot A^{(2)} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -l_{32} & 1 & 0 \\ 0 & -l_{42} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & a_{14}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & a_{14}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{34}^{(2)} \\ 0 & -l_{32} \cdot a_{22}^{(2)} + a_{32}^{(2)} & -l_{32} \cdot a_{23}^{(2)} + a_{33}^{(2)} & -l_{32} \cdot a_{24}^{(2)} + a_{34}^{(2)} \\ 0 & -l_{42} \cdot a_{22}^{(2)} + a_{42}^{(2)} & -l_{42} \cdot a_{23}^{(2)} + a_{43}^{(2)} & -l_{42} \cdot a_{24}^{(2)} + a_{44}^{(2)} \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & a_{14}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{34}^{(2)} \\ 0 & 0 & -l_{32}\cdot a_{23}^{(2)} + a_{33}^{(2)} & -l_{32}\cdot a_{24}^{(2)} + a_{34}^{(2)} \\ 0 & 0 & -l_{42}\cdot a_{23}^{(2)} + a_{43}^{(2)} & -l_{42}\cdot a_{24}^{(2)} + a_{44}^{(2)} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}^{(3)} & a_{12}^{(3)} & a_{13}^{(3)} & a_{14}^{(3)} \\ 0 & a_{22}^{(3)} & a_{23}^{(3)} & a_{24}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{pmatrix} = A^{(3)}$$

- Step 3: $A^{(3)}\mathbf{x} = \mathbf{b}^{(3)} \Leftrightarrow L_3 \cdot A^{(3)}(\mathbf{x}) = L_3 \cdot \mathbf{b}^{(3)}$ where L_3 zeros all elements below $a_{33}^{(3)}$ by elementary rows operations using

$$l_{43} = a_{43}^{(3)}/a_{33}^{(3)} \qquad \Rightarrow \qquad L_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -l_{43} & 1 \end{pmatrix}$$

Let's compute $A^{(4)} = L_3 \cdot A^{(3)}$:

$$L_3 \cdot A^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -l_{43} & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11}^{(3)} & a_{12}^{(3)} & a_{13}^{(3)} & a_{14}^{(3)} \\ 0 & a_{22}^{(3)} & a_{23}^{(3)} & a_{24}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}^{(3)} & a_{12}^{(3)} & a_{13}^{(3)} & a_{14}^{(3)} \\ 0 & a_{22}^{(3)} & a_{23}^{(3)} & a_{24}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & -l_{43}\cdot a_{33}^{(3)} + a_{43}^{(3)} & -l_{43}\cdot a_{34}^{(3)} + a_{44}^{(3)} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}^{(3)} & a_{12}^{(3)} & a_{13}^{(3)} & a_{14}^{(3)} \\ 0 & a_{22}^{(3)} & a_{23}^{(3)} & a_{24}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & 0 & -l_{43}\cdot a_{34}^{(3)} + a_{44}^{(3)} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}^{(4)} & a_{12}^{(4)} & a_{13}^{(4)} & a_{14}^{(4)} \\ 0 & a_{22}^{(4)} & a_{23}^{(4)} & a_{24}^{(4)} \\ 0 & 0 & a_{33}^{(4)} & a_{34}^{(4)} \\ 0 & 0 & 0 & a_{44}^{(4)} \end{pmatrix} = A^{(4)}$$

At the end of the three step, we have an upper triangular linear system $A^{(4)}\mathbf{x} = \mathbf{b}^{(4)}$ equivalent to the first one. Moreover, looking at the LU factorization, defining $U = A^{(4)}$, we have

$$\underbrace{L_3 \cdot L_2 \cdot L_1 \cdot A}_{U}\mathbf{x} = L_3 \cdot L_2 \cdot L_1 \cdot \mathbf{b}$$

and thus

$$L_3 \cdot L_2 \cdot L_1 \cdot A = U \quad \Leftrightarrow \quad A = \underbrace{L_1^{-1} \cdot L_2^{-1} \cdot L_3^{-1}}_{L} \cdot U$$

It is easy to check that

$$L = L_1^{-1} \cdot L_2^{-1} \cdot L_3^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & 0 & 1 & 0 \\ l_{41} & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & l_{32} & 1 & 0 \\ 0 & l_{42} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & l_{43} & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix}$$

3.7 Exercises

A star near the number of an exercise denote a difficult one.

1. Find the spectrum $\sigma(A)$, the spectral radius $\rho(A)$, the $\|A\|_1$ and $\|A\|_\infty$ of the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 2 \\ 0 & -2 & 0 \end{pmatrix}$$

2. Find, under the condition $\|\mathbf{x}\|_2 = 1$, the maximum value of the function $f(\mathbf{x}) = \|A\mathbf{x}\|_2$ where A is

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

[Answer: the maximum is just $\|A\|_2$ and so it is $\|A\|_2 = \left(\ \rho(A^T \cdot A)\ \right)^{1/2} = 3$.]

3. Prove that all eigenvalues of the matrix

$$A = \begin{pmatrix} 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \\ 1/5 & 1/6 & 1/7 \end{pmatrix}$$

have modulus less then 5/6. [Answer: for each eigenvalues λ of the matrix A we have $|\lambda| \leq \rho(A) \leq \|A\|_\infty = \max\left(\{47/60\ ,\ 37/60\ ,\ 107/210\}\right) = 47/60 < 5/6$.]

4. Prove that if A is symmetric and positive definite then all his eigenvalues are positive. [Hint: take $\mathbf{x}$ as an eigenvector of A, i.e., $A\mathbf{x} = \lambda\mathbf{x}$. So $\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T(\lambda\mathbf{x}) = \lambda\|\mathbf{x}\|^2$ and so on.]

5. Find the number of operations required to compute $\sum_{k=1}^{n} l_k \cdot r_k^2$. [Hint: note that $l_k \cdot r_k^2 = l_k \cdot r_k \cdot r_k$.]

6. It is known that the LU factorization (L has all ones in the main diagonal) of the matrix A has $|U| = -2$. Find, if possible, $|A^{-2}|$.

7. Compute the LU factorization (without pivoting) for the matrix

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

Using the LU factorization, compute $|A|$ and $|A^{-1}|$. [Answer:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 5/2 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}$$

as we can check doing the product. Then, we get $|A| = |LU| = |L| \cdot |U| = 2 \cdot (5/2) \cdot 1 = 5$ since $|L| = 1$. Then, we have $|A^{-1}| = 1/|A| = 1/5$]

8. A large linear system $A\mathbf{x} = \mathbf{b}$ with $n = 10^3$ equations and $K(A) = 10^6$ is solved using Gaussian elimination. Knowing that the true solution $\mathbf{x}$ has $\|\mathbf{x}\|_\infty = 100$ and the final relative residual is $\|\mathbf{r}\|_\infty / \|\mathbf{b}\|_\infty \approx 10^{-12}$, is it true that each component of the actually computed solution has at least three decimal places corrects after the decimal point?

9. Compute the condition number $K_2(A)$ of the matrix

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^{-4} \end{pmatrix}$$

[Answer: A is symmetric and positive definite with eigenvalues $\lambda_1 = 10^{-4}$, $\lambda_2 = 1$, $\lambda_3 = 10^4$; thus, the minimum eigenvalue is $\lambda_{min} = 10^{-4}$ and the maximum eigenvalue is $\lambda_{max} = 10^4$. Using theorem 3.9 we have $K_2(A) = \lambda_{max}/\lambda_{min} = 10^4/10^{-4} = 10^8$.]

10. A matrix $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant by rows if

$$|\, a_{ii} \,| > \sum_{\substack{k=1 \\ k\neq i}}^{n} |\, a_{ik} \,| \quad i = 1, \ldots, n$$

In the same manner, A is strictly diagonally dominant by columns if

$$|\, a_{ii} \,| > \sum_{\substack{k=1 \\ k\neq i}}^{n} |\, a_{ki} \,| \quad i = 1, \ldots, n$$

Prove that a symmetric and strictly diagonally dominant by row (or by column) matrix is positive definite.

11. Give some examples of well conditioned matrices and some other examples of ill conditioned matrices.

12. Give an exam ple of a square matrix of order $n = 3$ where it is possible to do the first step of the Gaussian elimination but it is not possible to do the second step.

13. The square matrix A has dimension $n = 1000$ with all zeros but $a_{ii} = 4$, $i = 1, \ldots, n$, $a_{i,i-1} = -2$, $i = 2, \ldots, n$, $a_{i,i+1} = -1$, $i = 1, \ldots, n-1$. Prove that this matrix is non singular. Then, using Matlab or Octave, plot the eigenvalues in the complex plane. [Hint: a matrix is singular if and only if it has the eigenvalue 0. What can we say about the region of the complex plane where eigenvalues of A are?]

14. The real, symmetric and positive definite matrix A has spectral radius $\rho(A) = 1/3$. Which of the following are true?

    (a) A has a negative eigenvalue.

    (b) A may be singular.

    (c) A may have the eigenvalue $1/10 + i/10$.

    (d) A has all the eigenvalues in the $(0, 1/3]$ interval of the real axis.

    (e) A has a Cholesky factorization.

15. ($\star$) Write a Matlab or Octave function for the Gaussian elimination able to solve the linear system $A\mathbf{x} = \mathbf{b}$ with and without partial pivoting. Then, use this function inside a script to find figures like figure (3.3).

16. Use Cholesky factorization to find if the following matrix is or is not positive definite

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \end{pmatrix}$$

Check your answer using the function eig() of Matlab or Octave.

Chapter 4

Classical iterative methods

Nowadays large linear systems where most of the coefficients are zero are frequently encountered in practice. The corresponding matrices, where about 1% to 10% of the elements are non zero, are called sparse matrices. For these large linear systems the Gaussian algorithm (or, that is the same, the LU factorization) suffers of a technical problem. To understand what's going on, first, we have to say that the elements of the linear system are stored in a memory inside the computer. There are, basically, three levels of memory: the hard disk, the RAM memory and the CACHE memory. The fasted one is the CACHE, at lest 10 times faster then the RAM which is at least 10000 times faster then the hard disk. Thus, for a fast solution we need to store all the linear system inside the CACHE memory. However, the CACHE memory is very expensive and so we just have a very small amount of it, say about 2 MB to 8 MB.

So, if the linear system is large but has very little elements different from zero there are techniques to store all these non zero elements inside the CACHE. What may happens using Gaussian elimination or LU factorization, see the figure below, is the creation of a lot of new non-zero elements (we say that that we have the *fill-in* phenomena and the matrix A at the end of the Gaussian algorithm is no more sparse).

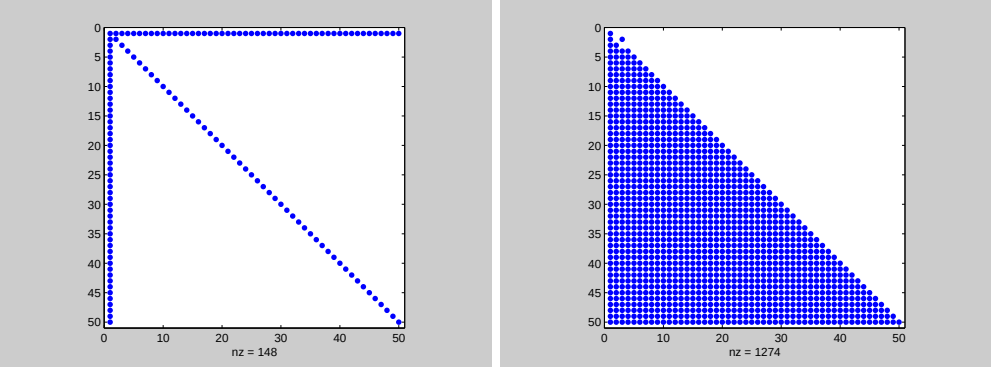


Figure 4.1: The non zero elements of A (left) and of L (right).

If the number of these new non-zero elements is large, they cannot be stored inside the CACHE memory and the overall algorithm slows down of almost one order of magnitude (this means that the amount of time needed to solve the linear system increases to about 10 times).

The aim of iterative methods is to preserve the original sparsity of the matrix and to get fast solution of the linear system.

4.1 Iterative methods

The basic idea of an iterative method is the following. First rearrange the linear system $A\mathbf{x} = \mathbf{b}$ as a fixed point equation $\mathbf{x} = E\mathbf{x} + \mathbf{q}$ for some matrix E and vector $\mathbf{q}$. Second, find the fixed point $\mathbf{x}$ of this latter equation as the limit for $k \to +\infty$ of the sequence $\mathbf{x}^{(k)}$ given by $\mathbf{x}^{(k+1)} = E\mathbf{x}^{(k)} + \mathbf{q}$, $k = 0, 1, \ldots$ where $\mathbf{x}^{(0)}$ is some starting point. First of all, let's see the conditions for which the sequence converge to the fixed point. Denoting the error of the k-th iteration $\mathbf{x}^{(k)}$ by

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}, \quad k = 0, 1, \ldots$$

we have

$$\begin{aligned} \mathbf{e}^{(k)} &= \mathbf{x} - \mathbf{x}^{(k)} = (E\mathbf{x} + \mathbf{q}) - \left(E\mathbf{x}^{(k-1)} + \mathbf{q} \right) = E\left(\mathbf{x} - \mathbf{x}^{(k-1)} \right) \\ &= E\mathbf{e}^{(k-1)} \end{aligned}$$

and so, using mathematical induction, $\mathbf{e}^{(k)} = E^k \mathbf{e}^{(0)}$. For the behaviour of the sequence $\mathbf{e}^{(k)}$ we need the following lemma for which we skip the proof.

Lemma 4.1 *Let E be a square matrix of order n. We have*

$$\lim_{k \to +\infty} E^k = O_n \quad \Leftrightarrow \quad \rho(E) < 1.$$

where O_n is the all zeros matrix of order n.

Now, we can state the main result for the convergence of the sequence $\mathbf{x}^{(k)}$.

Theorem 4.1 *The sequence $\mathbf{x}^{(k+1)} = E\mathbf{x}^{(k)} + \mathbf{q}$, $k = 0, 1, \ldots$ converges to the fixed point $\mathbf{x}$ of the equation $\mathbf{x} = E\mathbf{x} + \mathbf{q}$ for each $\mathbf{x}^{(0)}$ if and only if $\rho(E) < 1$.*

Proof.— We note that $\mathbf{x}^{(k)}$ converges to $\mathbf{x}$ if and only if $\mathbf{e}^{(k)}$ converges to $\mathbf{0}$. Then, assuming $E^k \to 0_n$, using the lemma, we have

$$\lim_{k \to +\infty} \mathbf{e}^{(k)} = \lim_{k \to +\infty} E^k \mathbf{e}^{(0)} = O_n \cdot \mathbf{e}^{(0)} = \mathbf{0}$$

On the other hand, if $\mathbf{e}^{(k)} \to \mathbf{0}$ for each $\mathbf{x}^{(0)}$, then it must be $E^k \to 0_n$. Indeed, if this is not the case, then there is at least one element of the limit of $E^{(k)}$ different from 0. Let $E_{ij}^{(k)} \to l_{ij} \neq 0$. Then, choosing $\mathbf{x}^{(0)}$ in order to have an error $\mathbf{e}^{(0)}$ with all zeros but a 1 in the position j, the limit of the error has at least $\mathbf{e}_i$ different from zero. This is a contradiction. So, the limit of $E^{(k)}$ is the O_n matrix and the theorem is proved. □

In most cases it is not easy to compute $\rho(E)$ in order to apply the previous theorem. Thus, it is interesting to note that if we find $\|E\| < 1$ for some induced norm, then it is also $\rho(E) < 1$. Note, however, that this is only a sufficient condition.

Now consider a convergent sequence. We can gain a deeper understanding of the convergence behaviour if E is a diagonalizable matrix with all real eigenvalues and with one, say λ_1, greater, in absolute value, of the absolute value of all other eigenvalues. Let $B = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ be the base of eigenvectors with $\mathbf{u}_1$ associated to λ_1. Thus, writing

$$\mathbf{e}^{(0)} = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i$$

and recalling that $\mathbf{u}_i$ is also an eigenvector for E^k associated to λ_i^k, we get, assuming $\alpha_1 \neq 0$,

$$\begin{aligned} \mathbf{e}^{(k)} &= E^k \mathbf{e}^{(0)} = \sum_{i=1}^{n} \alpha_i E^k \mathbf{u}_i = \sum_{i=1}^{n} \alpha_i \lambda_i^k \mathbf{u}_i \\ &= \lambda_1^k \cdot \left[\alpha_1 \mathbf{u}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{u}_i \right] \\ &\approx \alpha_1 \lambda_1^k \mathbf{u}_1 \end{aligned}$$

since, at least for large k, we have $(|\lambda_i|/|\lambda_1|)^k \ll 1$. From the latter equation, we have

$$\frac{\|\mathbf{e}^{(k)}\|}{\|\mathbf{e}^{(k-1)}\|} \approx \frac{\|\alpha_1 \lambda_1^k \mathbf{u}_1\|}{\|\alpha_1 \lambda_1^{k-1} \mathbf{u}_1\|} = |\lambda_1| = \rho(E)$$

since λ_1 has, among all eigenvalues, the larger absolute value. So, the order of convergence is 1 with asymptotic error constant equal to $\rho(E)$. This also means that we expect a linear plot for $\log_{10}\left(\|\mathbf{e}^{(k)}\|\right)$ as a function of k.

4.2 Classical methods

Let's see some simple, classical, ways to construct a fixed point iteration for a given linear system $A\mathbf{x} = \mathbf{b}$. Consider the splitting of the matrix A given by

$$A = L + D + U$$

where D is the diagonal of A, L is the strictly lower triangular part of A (i.e, the elements below the main diagonal) and U is the strictly upper triangular part of A (i.e, the elements above the main diagonal). That given, we have the following methods.

4.2.1 Jacobi method

If all elements on the main diagonal of A are different from 0, we can write

$$\begin{aligned} A\mathbf{x} = \mathbf{b} \quad &\Leftrightarrow \quad (L + D + U)\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad D\mathbf{x} = -(L + U)\mathbf{x} + \mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x} = -D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b} \end{aligned}$$

Setting $E_J = -D^{-1}(L + U)$ and $\mathbf{q}_J = D^{-1}\mathbf{b}$, the fixed point equation $\mathbf{x} = E_J\mathbf{x} + \mathbf{q}_J$ gives the fixed point iteration

$$\mathbf{x}^{(k+1)} = E_J\mathbf{x}^{(k)} + \mathbf{q}_J, \quad k = 0, 1, \ldots$$

known as the Jacobi method. The matrix E_J is the Jacobi iteration matrix.

4.2.2 Gauss-Seidel method

If all elements on the main diagonal of A are different from 0, we can write

$$\begin{aligned} A\mathbf{x} = \mathbf{b} \quad &\Leftrightarrow \quad (D + L + U)\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad (D + L)\mathbf{x} = -U\mathbf{x} + \mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x} = -(D + L)^{-1}U\mathbf{x} + (D + L)^{-1}\mathbf{b} \end{aligned}$$

Setting $E_S = -(D + L)^{-1}U$ and $\mathbf{q}_S = (D + L)^{-1}\mathbf{b}$, the fixed point equation $\mathbf{x} = E_S\mathbf{x} + \mathbf{q}_S$ gives the fixed point iteration

$$\mathbf{x}^{(k+1)} = E_S\mathbf{x}^{(k)} + \mathbf{q}_S, \quad k = 0, 1, \ldots$$

known as the Gauss-Seidel method. The matrix E_S is the Gauss-Seidel iteration matrix.

For hand writing exercises or for not Matlab-like programming, it is useful to write explicitly all equations of the previous methods. Let's see how to do with an example.

Example 4.1 *Consider the linear system $A\mathbf{x} = \mathbf{b}$ with*

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The Jacobi method may be written as $D\mathbf{x}^{(k+1)} = -(L + U)\mathbf{x}^{(k)} + \mathbf{b}$ or, explicitly,

$$\begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & -a_{12} & -a_{13} \\ -a_{21} & 0 & -a_{23} \\ -a_{31} & -a_{32} & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Solving this diagonal linear system with respect to $\mathbf{x}^{(k+1)}$*, we get*

$$\begin{cases} a_{11}x_1^{(k+1)} &= b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \\ a_{22}x_2^{(k+1)} &= b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} \\ a_{33}x_3^{(k+1)} &= b_3 - a_{31}x_2^{(k)} - a_{32}x_3^{(k)} \end{cases} \Leftrightarrow \begin{cases} x_1^{(k+1)} &= \left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right]/a_{11} \\ x_2^{(k+1)} &= \left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}\right]/a_{22} \\ x_3^{(k+1)} &= \left[b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right]/a_{33} \end{cases}$$

For the Gauss-Seidel method we have $(L+D)\mathbf{x}^{(k+1)} = -U\mathbf{x}^{(k)} + \mathbf{q}$ *or, explicitly,*

$$\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & -a_{12} & -a_{13} \\ 0 & 0 & -a_{23} \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Solving this lower triangular linear system with respect to $\mathbf{x}^{(k+1)}$*, we get*

$$\begin{cases} x_1^{(k+1)} &= \left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right]/a_{11} \\ x_2^{(k+1)} &= \left[b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)}\right]/a_{22} \\ x_3^{(k+1)} &= \left[b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}\right]/a_{33} \end{cases}$$

So, for the Jacobi method we compute each $x_i^{(k+1)}$ *just using* $\mathbf{x}^{(k)}$ *whereas for Gauss-Seidel we use, if available, components* $x_j^{(k+1)}$ *for* $1 \le j < i$ *and components* $x_j^{(k)}$ *for* $i < j \le n$*.*

Obviously, the order of the equations does not change the solution of a linear system. However, this order may affect the convergence of Jacobi and Gauss-Seidel methods. Let's see an example.

Remark 4.1 *The order of the equation in the linear system is important. Consider, for example, the Jacobi method for the following linear system of matrix* A*. We have*

$$A = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix} \quad \Rightarrow \quad B_J = \begin{pmatrix} 0 & -\frac{\beta}{\alpha} \\ -\frac{\beta}{\alpha} & 0 \end{pmatrix} \quad \Rightarrow \quad \rho(B_J) = \frac{|\beta|}{|\alpha|}$$

and thus the Jacobi method converges if $|\beta| < |\alpha|$*. That stated, let's exchange the two rows. For the new matrix* $\tilde{A}$*, we have*

$$\tilde{A} = \begin{pmatrix} \beta & \alpha \\ \alpha & \beta \end{pmatrix} \quad \Rightarrow \quad \tilde{B}_J = \begin{pmatrix} 0 & -\frac{\alpha}{\beta} \\ -\frac{\alpha}{\beta} & 0 \end{pmatrix} \quad \Rightarrow \quad \rho(\tilde{B}_J) = \frac{|\alpha|}{|\beta|}$$

and so, since now $\rho(\tilde{B}_J) > 1$*, Jacobi iterations do not converge.*

Remark 4.2 *The two linear systems* $A\mathbf{x} = \mathbf{b}$ *and* $PA\mathbf{x} = P\mathbf{b}$ *have the same solution for each non singular matrix* P *(preconditioning matrix). However, when applying an iterative method, the iteration matrices may have different spectral radii and, as a consequence, a different rate of convergence. As an example, useful also to see the behaviour of the Gauss-Seidel method to a real problem, consider the matrix "sherman1.mtx" of the Matrix Market collection (free from the Web). As we can see on the left of figure 4.2, it is a very sparse matrix: it has dimension* $n = 1000$ *but only* 3750 *non zero elements.*

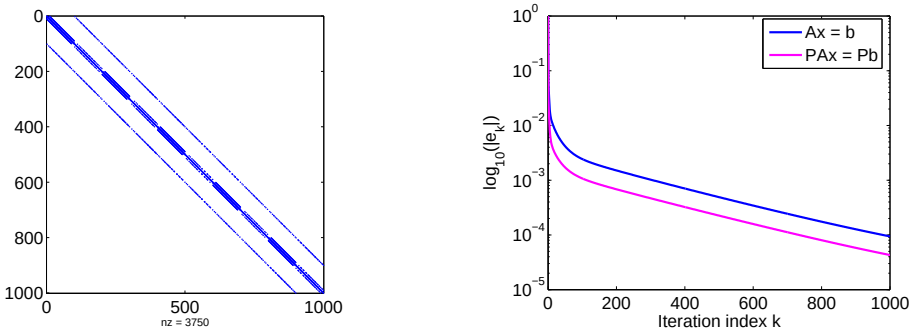


Figure 4.2: Convergence behaviour for the linear system with matrix given by sherman1.mtx. The right hand side is build in order to have the all 1 *solution.*

The matrix P is chosen as a diagonal matrix where entry $p_{ii} = 1/\sum_{k=1}^{n} |a_{ik}|$. As figure 4.2 shows, there is some advantage in solving $PA\mathbf{x} = P\mathbf{b}$ instead of $A\mathbf{x} = \mathbf{b}$. The problem of how to choose the best matrix P (i.e., the matrix for which the corresponding iteration matrix has the smallest spectral radius) is a challenging problem.

4.2.3 Conditions for the convergence of Jacobi and Gauss-Seidel

Given the linear system $A\mathbf{x} = \mathbf{b}$ it is not always easy to find the iteration matrix E_J or E_S and to compute the corresponding spectral radius. Thus, it is interesting to give some sufficient conditions for the convergence of the Jacobi and Gauss-Seidel methods just looking on the matrix A. There are such many of these conditions. We just give some.

Theorem 4.2 *If the matrix A of the linear system $A\mathbf{x} = \mathbf{b}$ is strictly diagonally dominant by rows or by column both Jacobi and Gauss-Seidel methods are convergent.*

Theorem 4.3 *If the matrix A of the linear system $A\mathbf{x} = \mathbf{b}$ is symmetric and positive definite then the Gauss-Seidel method is convergent.*

Theorem 4.4 *If the matrix A of the linear system $A\mathbf{x} = \mathbf{b}$ is a tridiagonal matrix, then Jacobi and the Gauss-Seidel methods are both divergent or both convergent. Moreover, if they are convergent, then $\rho(E_S) = \rho^2(E_J)$.*

For a general linear system $A\mathbf{x} = \mathbf{b}$, there is no relationship between the convergence behaviour of Jacobi and Gauss-Seidel methods. That is, both may not converge, or one may but not the other, or both can converge with different rates. Let's see some examples.

Example 4.2 *Let $a > 0$; the linear system with matrix A given by*

$$A = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}$$

has

$$E_J = \begin{pmatrix} 0 & -a \\ -a & 0 \end{pmatrix}, \quad E_S = -(D+L)^{-1}U = \begin{pmatrix} 1 & 0 \\ -a & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & a \\ 0 & -a^2 \end{pmatrix}$$

and so $\rho(E_J) = a$ and $\rho(E_S) = a^2$. Thus, both methods are convergent if $a < 1$ with Gauss-Seidel faster since $\rho(E_S) = a^2 < a = \rho(E_J)$, both are divergent if $a > 1$.

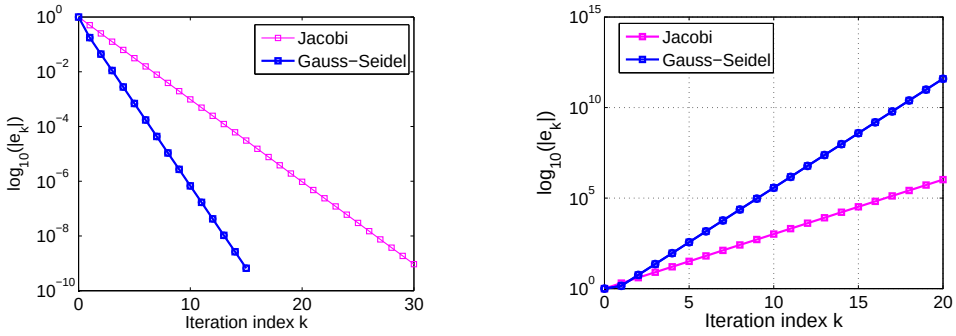


Figure 4.3: Convergence behaviour for the linear system with $a = 0.5$ (left) and $a = 2$ (right).

Example 4.3 *Using a Matlab program, we have find the following matrices with the corresponding spectral radii of the Jacobi and Gauss-Seidel methods.*

$$A = \begin{pmatrix} 34 & 20 & 11 \\ 17 & -5 & -16 \\ -8 & -13 & -20 \end{pmatrix}, \quad \rho(E_J) = 0.53, \quad \rho(E_S) = 0.87$$

$$A = \begin{pmatrix} -22 & -6 & -10 \\ 2 & -14 & 3 \\ 2 & -2 & -5 \end{pmatrix}, \quad \rho(E_J) = 0.55, \quad \rho(E_S) = 0.15$$

$$A = \begin{pmatrix} -11 & -6 & -11 \\ 4 & -10 & -11 \\ -9 & 0 & -8 \end{pmatrix}, \quad \rho(E_J) = 1.21, \quad \rho(E_S) = 0.82$$

$$A = \begin{pmatrix} 3 & -8 & 14 \\ -13 & -16 & -11 \\ -18 & -10 & 40 \end{pmatrix}, \quad \rho(E_J) = 0.57, \quad \rho(E_S) = 1.42$$

$$A = \begin{pmatrix} 3 & -2 & 18 \\ 9 & 15 & 10 \\ -6 & -3 & 6 \end{pmatrix}, \quad \rho(E_J) = 2.60, \quad \rho(E_S) = 4.84$$

As we can see, for a general matrix there is non relationship between the convergence of the Jacobi and Gauss-Seidel methods.

4.3 How to stop the iterations

There are two main stopping criteria for the iterative method $\mathbf{x}_{k+1} = E\mathbf{x}_k + \mathbf{q}$, $k = 0, 1, \ldots$

(a) Let $\mathbf{r_k} = \mathbf{b} - A\mathbf{x}_k$ be the residual vector at the k-th iteration. Given some small positive ϵ, the iterative method stops when $\|\mathbf{r}_k\| < \epsilon\|\mathbf{b}\|$ for some given norm. That is, the iterations stop when the residual is low enough with respect to $\mathbf{b}$. To see the goodness of this stopping criteria, note than if we write $A\mathbf{x}_k = \mathbf{b} - \mathbf{r}_k$ then $\mathbf{x}_k$ can be seen as the correct solution to the perturbed linear system $A\mathbf{x}_k = \mathbf{b} + \delta\mathbf{b}$ where $\delta\mathbf{b} = -\mathbf{r_k}$. So, we can relate the error $\mathbf{e}_k = \mathbf{x} - \mathbf{x}_k$ to the residual as

$$\frac{\| \mathbf{e}_k \|}{\| \mathbf{x} \|} \leq K(A) \cdot \frac{\| \mathbf{r}_k \|}{\| \mathbf{b} \|}$$

When the stopping criteria is fulfilled, we have $\|\mathbf{e}_k\|/\|\mathbf{x}\| \leq K(A)\epsilon$. So, if $K(A)$ is small, then the stopping criteria guarantees also a small relative error. However, if $K(A)$ is large, then the product $K(A)\epsilon$ is large and, as a consequence, the stopping criteria is not able to guarantee a small relative error.

(b) Given some small positive ϵ, the iterative method stops when $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon \|\mathbf{b}\|$. Taking into account that $\|\mathbf{e}^{(k+1)}\| \approx \rho(E)\|\mathbf{e}^{(k)}\|$, we can write

$$\begin{aligned} \left\|\mathbf{e}^{(k)}\right\| &= \|\mathbf{x} - \mathbf{x}_k\| = \|(\mathbf{x} - \mathbf{x}_{k+1}) - (\mathbf{x}_k - \mathbf{x}_{k+1})\| \\ &\leq \|\mathbf{x} - \mathbf{x}_{k+1}\| + \|\mathbf{x}_k - \mathbf{x}_{k+1}\| \\ &= \left\|\mathbf{e}^{(k+1)}\right\| + \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \\ &\approx \rho(E) \left\|\mathbf{e}^{(k)}\right\| + \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \end{aligned}$$

and thus

$$\left\|\mathbf{e}^{(k)}\right\| \leq \frac{1}{1-\rho(E)} \cdot (\|\mathbf{x}_{k+1} - \mathbf{x}_k\|)$$

When the stopping criteria is fulfilled, we have $\|\mathbf{e}^{(k)}\|/\|\mathbf{b}\| \leq \epsilon/[1-\rho(E)]$. As a consequence, if $\rho(E) \ll 1$ the stopping criteria guarantees a small relative error; otherwise, if $\rho(E) \approx 1$ the stopping criteria is not able to guarantee a small relative error.

4.4 exercises

A star ($\star$) near the number of an exercise denote a difficult one.

1. Is it true that if the fixed point iterations $\mathbf{x}^{(k)} = E\mathbf{x}^{(k)} + \mathbf{q}$, $k = 0, 1, \ldots$ converges for all $\mathbf{x}^{(0)}$ then also $\mathbf{x}^{(k)} = E^2\mathbf{x}^{(k)} + \mathbf{q}$, $k = 0, 1, \ldots$ does it? Explain.

2. Given the iteration matrix

$$E = \begin{pmatrix} 1/20 & 0 \\ 1 & -1/10 \end{pmatrix}$$

estimate the number of iterations needed to have $\|\mathbf{e}^{(k)}\| \leq \epsilon\|\mathbf{e}^{(0)}\|$ for $\epsilon = 10^{-9}$. [Answer: 9]

3. Starting from $\mathbf{x}^{(0)} = (0, 0)^T$, find the behaviour of the error for Jacobi and Gauss-Seidel for the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

4. Find the best stopping criteria for $\mathbf{x}^{(k+1)} = E\mathbf{x}^{(k)} + \mathbf{q}$ if the iteration matrix E is

$$E = \begin{pmatrix} 0.999 & 0 \\ 1 & 0.998 \end{pmatrix}$$

5. Consider the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$A = \begin{pmatrix} 3 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

   (a) Just looking to the matrix A, find if the Jacobi and the Gauss-Seidel methods are convergent.

   (b) Write explicitly the three iteration equations for both Jacobi and Gauss-Seidel.

   (c) Find the eigenvalues of the iteration matrix of Jacobi and the corresponding spectral radius.

   (d) Repeat point (c) for Gauss-Seidel. Check your answer using Matlab or Octave. Which method converges faster?

   (e) Give an estimation of the number of iterations needed to have $\|\mathbf{e}^{(k)}\| < \epsilon\|\mathbf{e}^{(0)}\|$ for $\epsilon = 10^{-6}$.

(f) Check answers of points (c), (d), (e) using Matlab or Octave.

6. The linear system $A\mathbf{x} = \mathbf{b}$ has the matrix A symmetric and positive definite. What can we say for the spectral radius of the Gauss-Seidel iteration matrix? [Answer: the Gaussian method is convergent and so $\rho(E_S) < 1$.]

7. Given the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 1 & \alpha \\ 1 & \alpha^2 \end{pmatrix}$$

find $\alpha \in \mathbb{R}$ such that the iteration matrix of the Gauss-Seidel method has spectral radius equal to $1/2$.

8. Prove that $\lambda = 0$ is always an eigenvalue of the Gauss-Seidel iteration matrix. [Hint: since U has the first column of all zeros, then also $E_S = -(D+L)^{-1}U$ does.]

9. Compute how many iterations does the Jacobi method need in order to have a norm of the error $\|\mathbf{e}^{(k)}\| < 10^{-9}$ for the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$A = \begin{pmatrix} 1/10 & 0 & 0 \\ 0 & 1/20 & 0 \\ 0 & 0 & -1/10 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

starting from $\mathbf{x}^{(0)} = (0\ ,\ 0\ ,\ 0)^T$. [Answer: just one iteration.]

10. Prove that is λ is an eigenvalue of the matrix E then λ^k is an eigenvalue for E^k. [Answer: let $\mathbf{v}$ be an eigenvector associated to λ. Then, we have $E^k\mathbf{v} = E^{k-1}(E\mathbf{v}) = E^{k-1}(\lambda\mathbf{v}) = \lambda E^{k-1}\mathbf{v} = \lambda E^{k-2}(E\mathbf{v}) = \cdots = \lambda^k\mathbf{v}$ and thus λ^k is an eigenvalue of E^k.]

11. ($\star$) Given the linear system $A\mathbf{x} = \mathbf{b}$ of order $n = 2$, find, if possible, a matrix A for which the Gauss-Seidel method converges but Jacobi does not converge.

Chapter 5

Interpolation and Approximation

5.1 Introduction

The interpolation problem is the following.

Problem 5.1 *Let $\mathcal{F}$ be a family of functions depending on $n+1$ parameters. We search for a function $\tilde{f} \in \mathcal{F}$ such that*

$$\tilde{f}(x_i) = y_i, \quad i = 0, \ldots, n \quad \textit{(interpolation conditions)}$$

where (x_i, y_i), $i = 0, \ldots, n$ are $n+1$ given points. We call x_i nodes.

Interpolation conditions give a system, maybe non linear, of $n+1$ equations in $n+1$ unknowns parameters. The solution of this system gives the parameters and thus the function $\tilde{f}$. Some families $\mathcal{F}$ are the following.

- **Polynomial functions** of degree (at most) n (polynomial interpolation)

  $$\tilde{f}(x) = p_n(x) = a_0 + a_1 x + \ldots + a_n x^n$$

  We have $n+1$ parameters a_k, $k = 0, \ldots, n$.

- **Trigonometric functions**

  $$\tilde{f}(x) = \frac{c_0}{2} + \sum_{k=1}^{m} [\, c_k \cos(kx) + b_k \sin(kx) \,]$$

  We have $n = 2m+1$ parameters c_k, $k = 0, \ldots, m$ and b_k, $k = 1, \ldots, m$.

- **Rational functions**

  $$\tilde{f}(x) = \frac{a_0 + a_1 x + \ldots + a_p x^p}{b_0 + b_1 x + \ldots + b_q x^q}$$

  We have $n = p + q + 1$ parameters a_k, $k = 0, \ldots, p$ and b_k, $k = 0, \ldots, q$.

- **Exponential functions**

  $$\tilde{f}(x) = a_0 e^{\lambda_0 x} + \ldots + a_m e^{\lambda_m x}$$

  We have $n = 2m+2$ parameters a_k, $k = 0, \ldots, m$ and λ_k, $k = 0, \ldots, m$.

5.2 Polynomial interpolation

The following theorem holds.

Theorem 5.1 *Let (x_i, y_i), $i = 0, \ldots, n$ be a set of $n+1$ points with different nodes x_i (i.e., $x_i \neq x_j$ for $i \neq j$). Then, there is one and only one polynomial $p_n(x)$ of degree n such that $p(x_i) = y_i$, $i = 0, \ldots, n$.*

Proof. Let $p_n(x) = a_0 + a_1 x + \ldots + a_n x^n$. The $n+1$ interpolation conditions $p(x_i) = y_i$, $i = 0, \ldots, n$ gives the linear system of $n+1$ equations in $n+1$ unknowns a_i, $i = 0, \ldots, n$

$$\left\{\begin{array}{rcl} p(x_0) & = & y_0 \\ p(x_1) & = & y_1 \\ \cdots & = & \cdots \\ p(x_i) & = & y_i \\ \cdots & = & \cdots \\ p(x_n) & = & y_n \end{array}\right. \Leftrightarrow \left\{\begin{array}{rcl} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n & = & y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n & = & y_1 \\ \cdots & = & \cdots \\ a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_n x_i^n & = & y_i \\ \cdots & = & \cdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n & = & y_n \end{array}\right.$$

or using matrices, $V_n \mathbf{a} = \mathbf{y}$ where

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_i & x_i^2 & \ldots & x_i^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_i \\ \ldots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \ldots \\ y_i \\ \ldots \\ y_n \end{pmatrix}$$

The matrix of the linear system is a Vandermonde matrix. For this kind of matrices, it can be proved that the determinant is given by

$$|V_n| = \prod_{i=0}^{n-1} \left(\prod_{j=i+1}^{n} (x_j - x_i) \right)$$

Due to the assumption of distinct nodes, the determinant in different form zero and so the linear system has only one solution. This means that there is only one polynomial that fulfills the interpolation conditions. □

From a numerical point of view, we have to take in mind that the Vandermonde matrix V_n is ill-conditioned (and so is the corresponding linear system) as we can see from the tables

n	1	3	7	15
$K(V_n)$	6	10^3	10^9	10^{23}

m	-2	-1	0	1	2
$K(V_3)$	10^6	$2 \cdot 10^3$	10^3	10^5	10^8

On the left, the condition number $K_2(V_n)$ is taken for different values of n choosing as nodes $x_i = i + 1$, $i = 0, \ldots, n$. On the right, the condition number of $K_2(V_3)$ is computed for the same number of nodes but with different spaces between two consecutive nodes: $x_i = (i+1) \cdot 10^m$, $i = 0, \ldots, n$ ed $m = -2, -1, 0, 1, 2$. As we can see, again the condition number may be very high. Thus we have to take care when the interpolation polynomial is computed in this way. Moreover, luckily, there are other more stable ways to find the interpolating polynomial.

5.2.1 Lagrangian interpolation

We begin with the definition of Lagrangian polynomials.

Definition 5.1 *Given the set of $n+1$ distinct nodes x_i, $i = 0, \ldots, n$, the k-th Lagrangian polynomial l_k is defined as*

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, \ldots, n, \quad k = 0, \ldots, n$$

As we can see, Lagrangian polynomials depend only on nodes and have the property that

$$l_k(x_i) = \begin{cases} 0 & , \quad i \neq k \\ 1 & , \quad i = k \end{cases}$$

For example, Lagrangian polynomials of degree $n = 2$ associated to nodes x_0, x_1, x_2 are

$$l_0(x) = \prod_{\substack{j=0 \\ j\neq 0}}^{2} \frac{x - x_j}{x_0 - x_j} = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2}$$

$$l_1(x) = \prod_{\substack{j=0 \\ j\neq 1}}^{2} \frac{x - x_j}{x_1 - x_j} = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2}$$

$$l_2(x) = \prod_{\substack{j=0 \\ j\neq 2}}^{2} \frac{x - x_j}{x_2 - x_j} = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1}$$

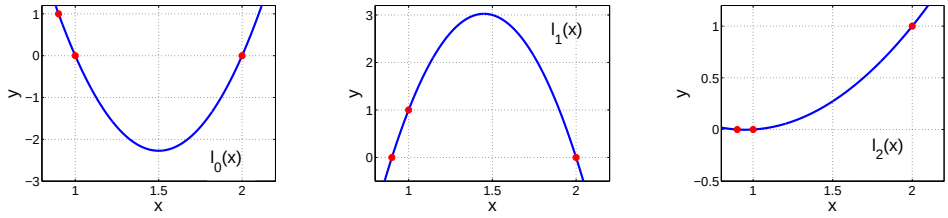


Table 5.1: Lagrangian polynomials of degree $n = 2$ associated to the nodes $x_0 = 0.9$, $x_1 = 1.0$, $x_2 = 2.0$.

Lagrange polynomials lead to a simple expression for the interpolation polynomial. Let (x_i, y_i), $i = 0, \ldots, n$ be a set of $n+1$ points with different nodes x_i. Then, the interpolation polynomial p_n can be write as

$$p_n(x) = \sum_{k=0}^{n} y_k l_k(x)$$

which is known as the Lagrange expression for the interpolation polynomials. Indeed, p_n fulfills the interpolation conditions $p_n(x_i) = x_i$, $i = 0, \ldots, n$ since

$$\begin{aligned} p_n(x_i) &= \sum_{k=0}^{n} y_k l_k(x_i) = y_i l_i(x_i) + \sum_{\substack{k=0 \\ k\neq i}}^{n} y_k l_k(x_i) \\ &= y_i \cdot 1 + \sum_{\substack{k=0 \\ k\neq i}}^{n} y_k \cdot 0 = y_i, \quad i = 0, \ldots, n \end{aligned}$$

Now, since the polynomial that fulfills interpolation conditions is unique, p_n is exactly the interpolation polynomial.

Using the Lagrange expression for the interpolation polynomial, it is easy to prove the following theorem.

Theorem 5.2 *The Lagrangian polynomials associated to the distinct nodes x_i, $i = 0, \ldots, n$ satisfy*

$$\sum_{k=0}^{n} l_k(x) = 1.$$

Proof. Consider the $n+1$ points $(x_i, 1)$, $i = 0, \ldots, n$. Of course, the polynomial through these points is $p_n(x) = 1$ and so $1 = \sum_{k=0}^{n} 1 \cdot l_k(x) = \sum_{k=0}^{n} l_k(x)$. □
Thus, the sum of all the Lagrange polynomial is 1 for all $x \in \mathbb{R}$.

5.2.2 Interpolation error

Sometimes, it is useful to have, at least in a given interval $[a,b]$, a *polynomial* alias p_n for a given function f. One possible choice for this polynomial is the interpolation polynomial through the points $(x_i, y_i = f(x_i))$, $i = 0, \ldots, n$ where x_i are chosen in $[a,b]$. The reason to operate in in this way is simple: instead of do operations using f, we do the same operations using p_n. For example, for $\bar{x} \in [a,b]$, instead of computing $f(\bar{x})$ we compute $p_n(\bar{x})$ and we assume this last value as an approximation for $f(\bar{x})$: $f(\bar{x}) \approx p_n(\bar{x})$. We do the same for more complicated operations such as derivatives and integrals:

$$f'(\bar{x}) \approx p_n'(\bar{x}), \qquad \int_a^b f(x)dx \approx \int_a^b p_n(x)\,dx$$

Doing so we have a clear advantage since operations with polynomials are very simple to work out. On the other hand, we have to take under control the error $E_n(x) = f(x) - p_n(x)$, $x \in [a,b]$.

Theorem 5.3 *Let $f \in C^{n+1}([a,b])$. If $x_i \in [a,b]$, $i = 0, \ldots, n$ are distinct nodes, then*

$$E_n(x) = \left(\prod_{i=0}^{n} (x - x_i) \right) \cdot \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

for a suitable point ξ dependent on x, $\xi \in (a,b)$.

We skip the proof of the theorem. Since we do not know ξ, it is useful to give an upper bound for $|E_n(x)|$, $x \in [a,b]$. We have obviously

$$|E_n(x)| \leq \left| \prod_{i=0}^{n} (x - x_i) \right| \cdot \frac{M_{n+1}}{(n+1)!} \quad \text{where} \quad M_{n+1} = \max_{x \in [a,b]} \left| f^{(n+1)}(x) \right|$$

This upper bound may be very large compared to the actually error and so less useful as the following example shows.

Example 5.1 *Consider $f(x) = \sqrt{x}$, $x \in [1,9]$. Let's choose the interpolating polynomial p_2 of degree $n = 2$ with nodes $x_0 = 1$, $x_1 = 4$, $x_2 = 9$ given by*

$$\begin{aligned} p_2(x) &= \sum_{k=0}^{2} y_k l_k(x) = y_0 \cdot l_0(x) + y_1 \cdot l_1(x) + y_2 \cdot l_2(x) \\ &= 1 \cdot \frac{(x-4)\cdot(x-9)}{(1-4)\cdot(1-9)} + 2 \cdot \frac{(x-1)\cdot(x-9)}{(4-1)\cdot(4-9)} + 3 \cdot \frac{(x-1)\cdot(x-4)}{(9-1)\cdot(9-4)} \\ &= -\frac{1}{60} \cdot x^2 + \frac{5}{12} \cdot x + \frac{3}{5} \end{aligned}$$

since $y_0 = f(x_0) = \sqrt{1} = 1$, $y_1 = f(x_1) = \sqrt{4} = 2$, $y_2 = f(x_2) = \sqrt{9} = 3$. The comparison between f and p_2 with the error E_2 is shown in figure 5.1. As we can see, the error is small: $|E_2(x)| \leq 0.05$ for $x \in [1,9]$. For example, for $\bar{x} = 4.1$ we get $f(\bar{x}) = 2.02$ and $p_2(\bar{x}) = 2.03$ with an error of 0.01. However, the upper bound is very poor. Since $f^{(3)}(x) = 3x^{-5/2}/8$ and so $M_3 = f^{(3)}(1) = 3/8$, we get

$$|E_2(\bar{x})| \leq |(4.1-1)(4.1-4)(4.1-9)| \cdot \frac{3/8}{3!} \approx 2$$

which is about two order of magnitude greater than the actually error. The bound performs better if both the interval $[a,b]$ and the $n+1$ derivative of f are small. Taking, for example, $x_0 = 3.61$, $x_1 = 4$ and $x_2 = 4.41$, we get the new upper bound

$$|E_2(\bar{x})| \leq |(4.1-3.61)(4.1-4)(4.41-4.1)| \cdot \frac{3/8}{3!} \approx 10^{-3}$$

The corresponding interpolating polynomial is able to guarantee at least three correct digits after the decimal point.

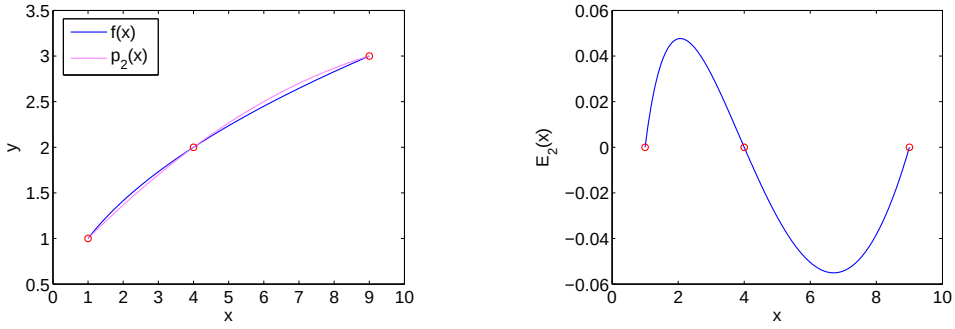


Figure 5.1: Left: the function $f(x) = \sqrt{x}$ and the interpolation polynomial $p_2(x)$ of degree two associated to the nodes $x_0 = 1$, $x_1 = 4$ and $x_2 = 9$ we get $p_2(x) =$. Right: the error $E_2(x) = f(x) - p_2(x)$.

Chebyshev nodes

A proper choice of the nodes x_i, $i = 0, \ldots, n$ inside $[a, b]$ can lead to a better approximation property of the interpolation polynomial. Looking back to the expression of the upper bound for $|E_n(x)|$, the idea is to choose the nodes in order to minimize the maximum of the absolute value of

$$\omega(x) = \prod_{i=0}^{n} (x - x_i)$$

for $x \in [a, b]$. The searching for these nodes is not an easy task. A near optimal set of nodes are the Chebyshev nodes given by

$$t_i = \cos\left(\frac{2i+1}{n+1} \cdot \frac{\pi}{2}\right), \qquad x_i = \frac{b+a}{2} + \frac{b-a}{2} \cdot t_i, \qquad i = 0, \cdots, n$$

where t_i, $i = 0, \ldots, n$ are the Chebyshev nodes of the interval $[-1, 1]$. The following theorem holds for Chebyshev nodes.

Theorem 5.4 (Bernstein) *Let x_i, $i = 0, \ldots, n$ be the Chebyshev nodes of the interval $[a, b]$. Then, for each function $f \in C^1([a, b])$ we have*

$$\lim_{n \to +\infty} \max_{x \in [a,b]} | E_n(x) | = 0.$$

That is, as one may hope, when the number of nodes increases the corresponding error decreases. This is a feature of the Chebyshev nodes. For example, for equally spaced nodes the error may not decrease when the number of nodes increases as the following example shows.

Example 5.2 (Runge) *Consider, in the interval $[-5, 5]$, the Runge function*

$$f(x) = \frac{1}{1 + x^2}$$

This is the function used by Runge to show that the error may not decrease if the number of equally spaced nodes increases. See figures 5.2 and 5.3 on the left. The nodes x_i, $i = 0, \ldots, n$ start with $x_0 = -5$ and are equally spaced in $[-5, 5]$. We can see, in both figures, two peaks at the endpoints of the interval. Moreover, the amplitude of these peaks increases with the number of nodes. So, also the error increases with the number of points. It can be proved that $\max_{x \in [-5,5]} |E_n(x)|$ goes toward $+\infty$ as n goes to $+\infty$ (this is a difficult result coming from complex analysis). Consider now the same figures on the right where Chebyshev nodes are taken into account. We can see that there are no more peaks and when the number of points increases the interpolation polynomial behaves better. Finally, in figure 5.2 we can see

the behaviour of the error as a function of the number of nodes. Clearly, for equally spaced nodes the error increases whereas it decreases for Chebyshev nodes.

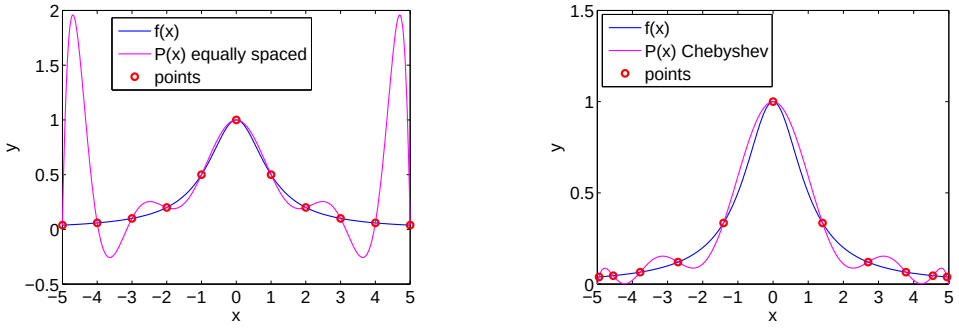


Figure 5.2: Interpolation polynomial using 15 nodes both equally spaced and Chebyshev.

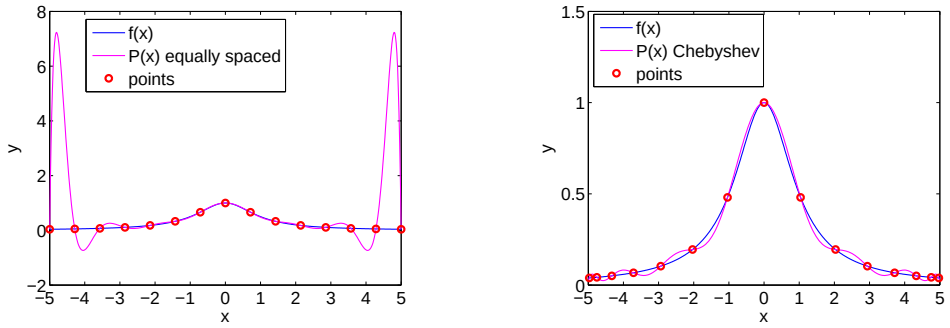


Figure 5.3: Interpolation polynomial using 15 nodes both equally spaced and Chebyshev.

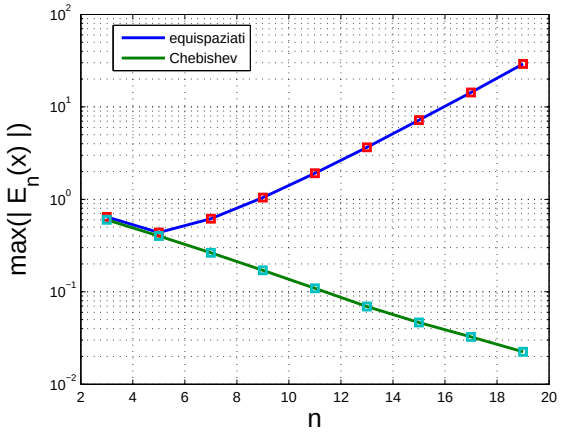


Figure 5.4: The interpolation error increases with n for equally spaced nodes and decreases for Chebyshev nodes.

5.3 Newton expression of the interpolating polynomial

We first give the definition of divided differences.

Definition 5.2 *Let x_i, $i = 0, \ldots, n$ be $n+1$ distinct nodes and f a function known (at lest) in this nodes. We define*

$$\begin{aligned}
f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad \textit{(order 1)} \\
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \quad \textit{(order 2)} \\
f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} \quad \textit{(order 3)} \\
\ldots &= \ldots \\
f[x_0, x_1, x_2, \ldots, x_{n-1}, x_n] &= \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} \quad \textit{(order n)}
\end{aligned}$$

Trivially, we have $f[x_0, x_1] = f[x_1, x_0]$; that is, the order of the points does not change the divided difference of order 1. This is a general rule.

Theorem 5.5 *Divided differences do not depend on the order on which the points are taken.*

Proof. — Omitted. □
Thus, if we consider three points, we have

$$f[x_0, x_1, x_2] = f[x_0, x_2, x_1] = f[x_1, x_0, x_2] = f[x_1, x_2, x_0] = f[x_2, x_0, x_1] = f[x_2, x_1, x_0]$$

as we can work out explicitly. It is usual to present divided differences in a table. Let's see how tables with two, three and four points look. From these, its clear how they behave for a different number of points.

$$\begin{array}{l|ll}
x_0 & f(x_0) & \\
x_1 & f(x_1) & f[x_0, x_1]
\end{array}$$

$$\begin{array}{l|lll}
x_0 & f(x_0) & & \\
x_1 & f(x_1) & f[x_0, x_1] & \\
x_2 & f(x_2) & f[x_1, x_2] & f[x_0, x_1, x_2]
\end{array}$$

$$\begin{array}{l|llll}
x_0 & f(x_0) & & & \\
x_1 & f(x_1) & f[x_0, x_1] & & \\
x_2 & f(x_2) & f[x_1, x_2] & f[x_0, x_1, x_2] & \\
x_3 & f(x_3) & f[x_2, x_3] & f[x_1, x_2, x_3] & f[x_0, x_1, x_2, x_3]
\end{array}$$

Remark 5.1 *Looking at previous tables, it is clear that the second table is the first one plus the last row. The same is true for the third one: it is the second plus the last row. So, if we already have a given table, it is easy to add a point and construct the new table. Just append the point at the end, regardless its value with respect to other nodes, and construct the last line of the new table.*

Example 5.3 *The divided differences table of points $(-1, 0)$, $(0, 1)$ and $(1, 3)$ is*

$$\begin{array}{r|lll}
\mathbf{-1} & \mathbf{0} & & \\
\mathbf{0} & \mathbf{1} & 1 & \\
\mathbf{1} & \mathbf{3} & 2 & 1/2
\end{array}$$

since we have (look at the previous table in the middle)

$$\begin{aligned}
f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1 - 0}{0 - (-1)} = 1 \\
f[x_1, x_2] &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{3 - 1}{1 - 0} = 2 \\
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{2 - 1}{1 - (-1)} = \frac{1}{2}
\end{aligned}$$

Divided differences allow a different expression to the interpolating polynomial.

Theorem 5.6 (Newton expression) *The interpolating polynomial p throughout points $(x_i, f(x_i))$, $i = 0, \ldots, n$, with distinct nodes, may be written as*

$$\begin{aligned} p(x) &= f(x_0) + \\ &+ f[x_0, x_1](x - x_0) + \\ &+ f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\ &+ \ldots \\ &+ f[x_0, x_1, x_2, \ldots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

Proof. — Omitted. □

Example 5.4 *Let's find the interpolating polynomial p throughout points $(y_i = f(x_i))$*

$$(x_0, y_0) = (-1, 1), \quad (x_1, y_1) = (0, 2), \quad (x_2, y_2) = (1, 0), \quad (x_3, y_3) = (2, 1)$$

The divided differences table is

$$\begin{array}{r|llll} -1 & \mathbf{1} & & & \\ 0 & 2 & \mathbf{1} & & \\ 1 & 0 & -2 & \mathbf{-3/2} & \\ 2 & 1 & 1 & 3/2 & \mathbf{1} \end{array}$$

Reading the numbers in bold, we get the interpolating polynomial

$$\begin{aligned} p(x) &= 1 + 1 \cdot (\, x - (-1)\,) - \frac{3}{2}(\, x - (-1)\,)(x - 0) + 1 \cdot (\, x - (-1)\,)(x - 0)(x - 1) \\ &= 1 + (x + 1) - \frac{3}{2}\, x(x + 1) + x(x + 1)(x - 1) \end{aligned}$$

It is easy to see that interpolation conditions $p(x_i) = y_i$, $i = 0, \ldots, 3$ are fulfilled.

Consider the polynomial p interpolating a function f at distinct nodes x_i, $i = 0 \ldots, n$. We may take p as an alias of f doing operations with p instead of f. For example, we write

$$f(\xi) \approx p(\xi) \qquad f'(\xi) \approx p'(\xi) \qquad \int_a^b f(x)\, dx \approx \int_a^b p(x)\, dx$$

where ξ, a, b are points within the minimum nodes and the maximum nodes. Previous relations are not exact in general but, the right hand side, may give a reasonable estimate of the exact value in the left hand side.

Example 5.5 *Consider the polynomial interpolating $f(x) = e^x$ at nodes $x_0 = 0$, $x_1 = 0.5$, $x_2 = 1$. We have the table*

$$\begin{array}{l|lll} 0 & 1 & & \\ 0.5 & \sqrt{e} & 2\,(\,\sqrt{e} - 1\,) & \\ 1 & e & 2\,(\,e - \sqrt{e}\,) & 2\,(\,e + 1 - 2\sqrt{e}\,) \end{array}$$

and so the interpolating polynomial is

$$p(x) = 1 + 2\,\left(\,\sqrt{e} - 1\,\right) x + 2\left(\,e + 1 - 2\sqrt{e}\,\right) x(x - 0.5)$$

Taking, for example, $\xi = 0.75$ we have $f(\xi) = f'(\xi) = 2.12$ and $p(\xi) = 2.13$, $p'(\xi) = 2.14$; thus, there is a good agreement.

5.4 Least square approximation

Given the set of $m+1$ points (x_i, y_i), $i = 0, \ldots, m$ in the plane, we search for a polynomial $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ of degree $n < m$ (indeed, in real problems we have $n \ll m$) such that is minimum

$$d = \sum_{i=0}^{m} [\, p(x_i) - y_i \,]^2$$

So, we minimize the sum of squares of vertical displacements between $p(x_i)$ and y_i. From this, we call p the *approximating polynomial in the least square sense*. Its coefficients are solution of the linear system $A^T A\mathbf{a} = A^T\mathbf{y}$ (called normal equations) where

$$A = \begin{pmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ 1 & x_2 & \ldots & x_2^n \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_m & \ldots & x_m^n \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \ldots \\ y_m \end{pmatrix}$$

and $\mathbf{a} = (\, a_0,\ a_1, \ldots, a_n \,)^T$. Note that A has $m+1$ rows and $n+1$ columns. The matrix $A^T A$ is, obviously, symmetric and, it can be proved, positive define. Thus, we can use Cholesky to solve the system of the normal equations. However, it may happen that $A^T A$ is very ill conditioned; in such cases, it is better to use other methods to find the polynomial coefficients such as the QR factorization or the SVD (Singular Value Decomposition).

5.4.1 Regression line

We call *regression line* the approximating polynomial of degree $n = 1$. Setting $p(x) = a_1 x + a_0$, we get easily

$$A^T A = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_0 & x_1 & x_2 & \ldots & x_m \end{pmatrix} \cdot \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \\ \ldots & \ldots \\ 1 & x_m \end{pmatrix} = \begin{pmatrix} m+1 & \sum_{i=0}^{m} x_i \\ \sum_{i=0}^{m} x_i & \sum_{i=0}^{m} x_i^2 \end{pmatrix}$$

$$A^T\mathbf{y} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_0 & x_1 & x_2 & \ldots & x_m \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \ldots \\ y_m \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{m} y_i \\ \sum_{i=0}^{m} x_i y_i \end{pmatrix}$$

Stating the linear system in the usual way, we have

$$\begin{cases} (m+1)\ a_0 + \sum_{i=0}^{m} x_i\ a_1 = \sum_{i=0}^{m} y_i \\ \sum_{i=0}^{m} x_i\ a_0 + \sum_{i=0}^{m} x_i^2\ a_1 = \sum_{i=0}^{m} x_i\, y_i \end{cases}$$

The first equation of the system states that barycenter $G = (x_G, y_G)$ of the set of given points defined as

$$x_G = \frac{\sum_{i=0}^{m} x_i}{m+1} \qquad y_G = \frac{\sum_{i=0}^{m} y_i}{m+1}$$

lies on the regression line. To prove, just divide it by $m+1$.

Remark 5.2 (Non linear models) *Given $m+1$ points (x_i, y_i), $i = 0, \ldots, m$, a regression line may also be seen as a first order model to describe the relationship between abscissas and ordinate of points. The regression line works well if data are "somehow" around a*

line. Sometimes, however, this linear relationship is obviously wrong as we may check again plotting the points or by some other, maybe theoretical, aspects.

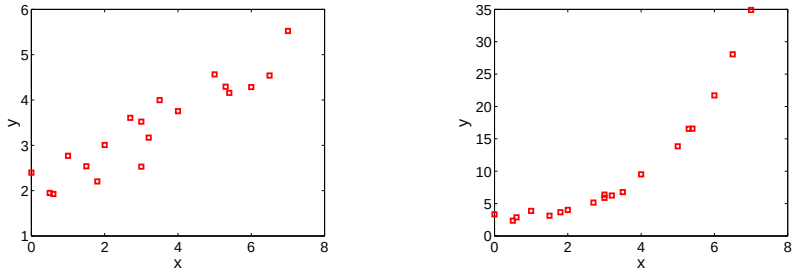


Figure 5.5: For data on the left a linear model $y = a_1 x + a_0$ seems good but for data on the right it seems better to search for another type of model.

We consider just two types of model depending on parameters K and λ:

$$(i)\ \ y = K\, e^{\lambda\, x}, \qquad (ii)\ \ y = K\, x^{\lambda}$$

To see how to find parameters, consider case (i). Taking logarithms of both sides we get

$$\ln(y) = \lambda\, x + \ln(K)$$

which is a linear relation between the new variables $Y = \ln(y)$ and $X = x$. In the same way, for case (ii) we have

$$\ln(y) = \lambda \ln(x) + \ln(K)$$

which is a linear relation between the new variables $Y = \ln(y)$ and $X = \ln(x)$. As an example, consider a model of type (i) for the points given in the table

x_i	1	2	3	4	5	6	7	8	9	10
y_i	7.79	6.07	4.72	3.68	2.87	2.23	1.73	1.35	1.05	0.82

First, we compute the new table

$X_i = x_i$	1	2	3	4	5	6	7	8	9	10
$Y_i = \ln(y_i)$	2.05	1.80	1.55	1.30	1.05	0.80	0.55	0.30	0.05	-0.20

Second, using Matlab, we get the regression line $Y = -0.25X + 2.30$. So, it is $\lambda = a_1 = -0.25$ and $\ln(K) = a_0 = 2.30$ or $K = e^{2.30} = 9.97 \approx 10$. The model is $y = 10 \cdot e^{-0.25x}$.

5.5 Exercises

1. Write the Lagrangian polynomials l_0 and l_1 associated to the distinct nodes x_0 and x_1. Plot, in the same window, the graphs of the two polynomial.

2. Compute the three Lagrange polynomial associated to the nodes $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ and prove that their sum gives 1. Then, give the Lagrange expression for the interpolation polynomial through the points $(x_0, 1)$, $(x_1, -1)$, $(x_2, 2)$.

3. Compute the Newton expression for the interpolation polynomial associated to the points $(0, 2)$, $(1, 3)$, $(2, 1)$. Then add the point $(3, -1)$ and find the new interpolation polynomial without recomputing all the difference table. Check that the given answers are correct!

4. Let $f(x) = x - \sqrt{x}$. Given the set of nodes $x_i = i^2$, $i = 0, 1, 2, 3$, compute, if possible, the number d defined as

$$d = | f[x_0, x_1, x_2, x_3] - f[x_0, x_3] - f[x_3, x_2, x_1, x_0] |$$

5. Find the regression line for the four points of Exercise 3. Show that the barycenter of these points is on the regression line.

Chapter 6

Numerical Integration

Numerical integration is also known as *numerical quadrature*. The problem is to compute, with a prescribed degree of accuracy, the definite integral

$$I = \int_a^b f(x)dx$$

where f is a continuous function, at lest in the interval $[a, b]$, with $a < b$. The main idea, is to replace, in the interval $[a, b]$, the function f with an interpolation polynomial $p_n(x)$ of degree n and to compute the integral of $p_n(x)$ as an approximation of I:

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx =: I_n$$

Consider a set of $n+1$ distinct points in $[a, b]$, called nodes. Using the Lagrange expression for the interpolation polynomial, we find a first example of *quadrature formula*

$$I_n = \int_a^b \left(\sum_{i=0}^n f(x_i) l_i(x) \right) dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^n A_i^{(n)} f(x_i)$$

where we have defined

$$A_i^{(n)} := \int_a^b l_i(x)dx, \quad i = 0, \ldots, n$$

More generally, we give the definition

Definition 6.1 *Consider the computation of the integral*

$$I = \int_a^b \omega(x) f(x)dx$$

where $\omega(x)$, $x \in [a, b]$ is a given positive function. Consider a set of distinct nodes x_i, $i = 0, \ldots, n$ in $[a, b]$. A quadrature formula is any sum of the kind

$$I_n = \sum_{i=0}^n \alpha_i f(x_i) \tag{6.1}$$

using to approximate the integral I. Numbers α_i are called weights and x_i nodes of the quadrature formula.

Note that in the quadrature formula does not appear the function $\omega(x)$. Clearly, the purpose of any quadrature formula is to give good (possible exact) approximations of I. A way to quantify the goodness of a quadrature formula is throughout the following definition.

Definition 6.2 (Degree of precision) *A quadrature formula has degree of precision $s \in \mathbb{N}$ if*

- *is correct for each polynomial of degree $n \leq s$;*
- *there is at least one polynomial of degree $n = s+1$ for which the formula is not correct (that is, $I_n \neq I$).*

It is possible to prove the following

Theorem 6.1 *The quadrature formula (6.1) has at most degree of precision $s = 2n + 1$ (obtained for the Gaussian quadrature formulas).*

Let's see how a way, not the best one indeed, to compute the coefficients and the nodes in order to have the maximum degree of precision.

Example 6.1 *Consider the quadrature formula*

$$\int_0^1 \sqrt{x} f(x)dx \approx \sum_{i=0}^{1} \alpha_i f(x_i)$$

Since $n = 1$, the maximum degree of precision is $s = 2 \cdot 1 + 1 = 3$. So, we can choose weights and nodes in order to have a formula which gives the correct result for each polynomial of degree at most 3. Since a quadrature formula is exact for each polynomial of degree at most n if and only if it is exact for x^i, $i = 0, \ldots, n$, we impose the conditions

$$\begin{aligned}
\int_0^1 \sqrt{x} \cdot 1 \, dx &= \alpha_0 + \alpha_1 \\
\int_0^1 \sqrt{x} \cdot x \, dx &= \alpha_0 x_0 + \alpha_1 x_1 \\
\int_0^1 \sqrt{x} \cdot x^2 \, dx &= \alpha_0 x_0^2 + \alpha_1 x_1^2 \\
\int_0^1 \sqrt{x} \cdot x^3 \, dx &= \alpha_0 x_0^3 + \alpha_1 x_1^3
\end{aligned}$$

The solution of this non linear system gives

$$\begin{aligned}
x_0 &= \tfrac{5}{9} - \tfrac{2\sqrt{70}}{63} & \alpha_0 &= \tfrac{1}{3} - \tfrac{\sqrt{70}}{150} \\
x_1 &= \tfrac{5}{9} + \tfrac{2\sqrt{70}}{63} & \alpha_1 &= \tfrac{1}{3} + \tfrac{\sqrt{70}}{150}
\end{aligned}$$

The quadrature formula is then

$$\int_0^1 \sqrt{x} f(x)dx \approx \left(\frac{1}{3} - \frac{\sqrt{70}}{150}\right) f\left(\frac{5}{9} - \frac{2\sqrt{70}}{63}\right) + \left(\frac{1}{3} + \frac{\sqrt{70}}{150}\right) f\left(\frac{5}{9} + \frac{2\sqrt{70}}{63}\right)$$

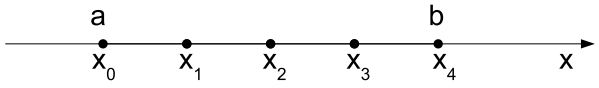
which gives the correct result for each $f(x) = p_n(x)$, $n \leq 3$.

6.1 Newton Cotes formulas (closed form)

We have these formulas if the $n + 1$ distinct nodes are equally spaced in $[a, b]$ as

$$x_i = a + ih, \quad i = 0, \ldots, n \quad \text{with } h = \frac{b-a}{n}$$

We can see an example of 5 nodes (and so $n = 4$) in the following figure. In this case, $h = (b - a)/4$.



Using the Lagrange expression for the interpolating polynomial, it is possible rewrite the numbers A_i^n as

$$A_i^n = nhC_i^{(n)} \quad \text{where} \quad C_i^{(n)} = \frac{1}{n}\int_0^n \frac{\prod_{r=0,r\neq i}^{n}(s-r)}{\prod_{r=0,r\neq i}^{n}(i-r)}ds$$

where $C_i^{(n)}$ are called the *Cotes numbers*. The idea of the proof is just change the integration variable as $x = x_0 + hs$ in the integral of $l_i(x)$.

Theorem 6.2 *The Cotes numbers fulfill the relation*

$$\sum_{i=0}^{n} C_i^{(n)} = 1.$$

Proof. Just take $f(x) = 1$, $x \in [a,b]$. In this case, we have

$$b-a = \int_a^b 1\,dx = \int_a^b \sum_{i=0}^{n} l_i(x)dx = \sum_{i=0}^{n}\int_a^b l_i(x)dx = \sum_{i=0}^{n} nhC_i^{(n)} = (b-a)\sum_{i=0}^{n} C_i^{(n)}$$

since $nh = b-a$. The proof is complete. □

Note that the Cotes numbers do not depend on the function nor on the integration interval. So, they may be computed just once.

Example 6.2 (Trapezoidal rule) *Consider the case $n = 1$ and so $x_0 = a$, $x_1 = b$, $h = b-a$. We have*

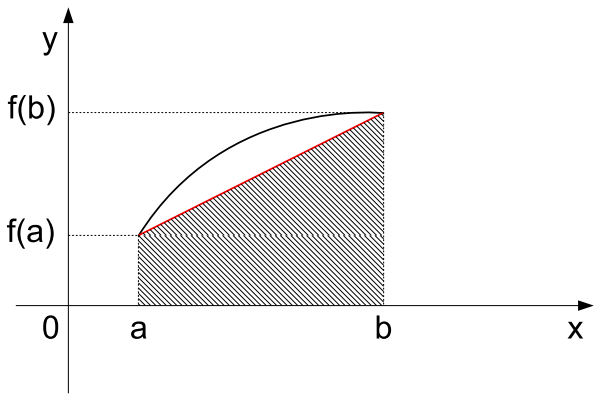
$$C_0^{(1)} = \int_0^1 \frac{s-1}{0-1}\,ds = \left[-\frac{s^2}{2}+s\right]_0^1 = \frac{1}{2}$$

$$C_1^{(1)} = \int_0^1 \frac{s-0}{1-0}\,ds = \left[\frac{s^2}{2}\right]_0^1 = \frac{1}{2}$$

and so the corresponding quadrature formula is

$$I_1 = \sum_{i=0}^{1} A_i^{(1)} f(x_i) = (b-a)\left[\frac{1}{2}f(x_0) + \frac{1}{2}f(x_1)\right] = \frac{(b-a)[\,f(a)+f(b)\,]}{2}$$

So I_1 is the area of the trapezoid in the next figure.



Example 6.3 (Cavallieri-Simpson rule) *Consider the case $n = 2$ and so $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$ and $h = (b-a)/2$. We have*

$$\begin{aligned} C_0^{(2)} &= \frac{1}{2}\int_0^2 \frac{(s-1)(s-2)}{(0-1)(0-2)}\,ds = \frac{1}{6} \\ C_1^{(2)} &= \frac{1}{2}\int_0^2 \frac{(s-0)(s-2)}{(1-0)(1-2)}\,ds = \frac{4}{6} \\ C_2^{(2)} &= \frac{1}{2}\int_0^2 \frac{(s-0)(s-1)}{(2-0)(2-1)}\,ds = \frac{1}{6} \end{aligned}$$

and so we have the Cavalieri-Simpson quadrature formula

$$I_2 = \sum_{i=0}^{2} A_i^{(2)} f(x_i) = \frac{b-a}{2}\left[\frac{1}{3}f(x_0) + \frac{4}{3}f(x_1) + \frac{1}{3}f(x_2)\right]$$

The previous formulas, as any quadrature formula, are not correct in the general case, i.e. we have $I = I_n + E_n$ where E_n is the error. Just as an example, we have

$$I = \int_0^1 x^3 dx = \frac{1}{4} \neq \frac{1}{2} = \frac{(1-0)[0^3+1^3]}{2} = I_1$$

where we have used the trapezoidal formula. We have the following theorems.

Theorem 6.3 *Let $f \in C^2([a,b])$. The error of the Trapezoidal formula is*

$$E_1 = -\frac{(b-a)^3}{12} f''(\xi)$$

where ξ is a suitable point in $[a,b]$.

Theorem 6.4 *Let $f \in C^4([a,b])$. The error of the Cavalieri-Simpson formula is*

$$E_2 = -\frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

where ξ is a suitable point in $[a,b]$.

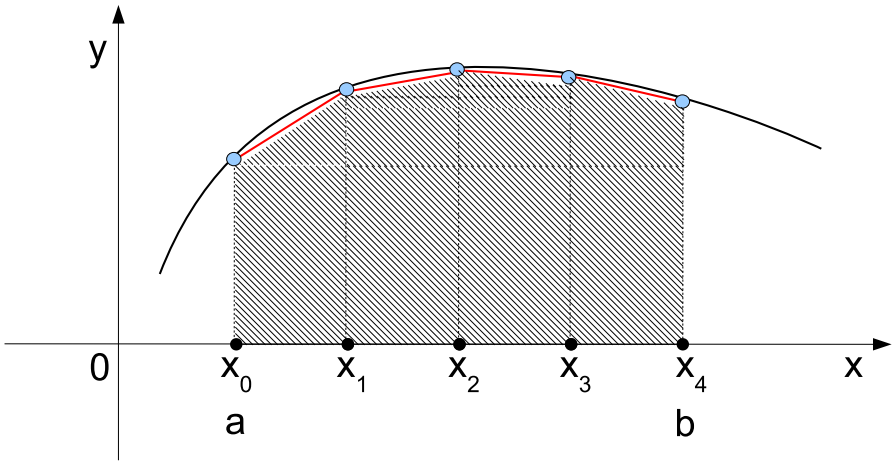
From these theorems we see that trapezoidal formula has degree of precision $s = 1$ whereas Cavalieri-Simpson has $s = 3$ (with just one more node).

6.2 Composite Newton-Cotes formulas

It is not a good idea to have an interpolation polynomial of high degree due to its oscillatory behaviour at the endpoints. It is better to divide the integration interval $[a,b]$ into m intervals and to apply a Newton-Cotes formula (with a low degree interpolation polynomial) to each of these intervals. Proceeding in this way we obtain the *composite Newton-Cotes formulas.*

6.2.1 Composite trapezoidal formula

The idea is shown in the following figure where $m = 4$ since $[a,b]$ is divided into 4 intervals. At each interval we apply the trapezoidal rule.



In the general case, each interval has length $h = (b-a)/m$ and so we get, using the additivity of the integral,

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{k=0}^{m-1}\int_{x_k}^{x_{k+1}} f(x)dx \approx \sum_{k=0}^{m-1}\frac{h[\,f(x_k)+f(x_{k+1})\,]}{2}\\ &= \frac{b-a}{m}\left[\frac{1}{2}f(x_0)+\sum_{k=1}^{m-1}f(x_k)+\frac{1}{2}f(x_m)\right]\end{aligned}$$

since each internal point, i.e, for $k = 1$ to $k = m-1$, is counted twice. The global error is the sum of the errors we have in each interval; denoting as $E_{1,k}$ $k = 0, \ldots, m-1$ the error in the interval $\mathcal{I}_k = [x_k, x_{k+1}]$, we get

$$\begin{aligned}E_m &= \sum_{k=0}^{m-1} E_{1,k} = \sum_{k=0}^{m-1} -\frac{h^3}{12}f''(\xi_k) = -\frac{(b-a)^3}{12m^3}\,\frac{\sum_{k=0}^{m-1} f''(\xi_k)}{m}\cdot m\\ &\overset{(*)}{=} -\frac{(b-a)^3}{12m^2}f''(\xi)\end{aligned}$$

where ξ is some suitable point inside $[a,b]$. To state equality $(*)$ we need the continuity of $f''(x)$ inside $[a,b]$; then, we apply the all values theorem since we have

$$\min_{x\in[a,b]} f''(x) \le \frac{\sum_{k=0}^{m-1} f''(\xi_k)}{m} \le \max_{x\in[a,b]} f''(x)$$

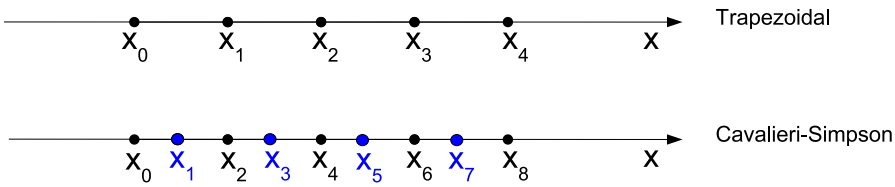
Remark 6.1 *Assuming that $f''(x)$ does not change to much in $[a,b]$, we have the following behaviour of the error.*

$$\frac{E_{2m}}{E_m} = \frac{-\frac{(b-a)^3}{12(2m)^3}f''(\xi_{2m})}{-\frac{(b-a)^3}{12m^3}f''(\xi_m)} = \frac{f''(\xi_{2m})}{4\cdot f''(\xi_m)} \approx \frac{1}{4}.$$

Thus, doubling the number of points, the error E_{2m} reduces to about $E_m/4$.

6.2.2 Composite Cavalieri-Simpson formula

Proceeding in the same way, we find the *composite Cavalieri-Simpson formula.* In this case, however, each interval has also a point in the middle. The comparison between the two rules is given in the figure (in both cases we have $m = 4$).



Thus, the intervals are now $\mathcal{I}_k = [x_{2k}, x_{2k+2}]$, $k = 0, 1 \ldots, m-1$. So, recalling the Cavalieri-Simpson rule, we obtain

$$\int_a^b f(x)dx \approx \frac{b-a}{2m}\left[\frac{1}{3}f(x_0) + \frac{2}{3}\sum_{\substack{\text{internal}\\ \text{even}}} f(x_k) + \frac{4}{3}\sum_{\substack{\text{internal}\\ \text{odd}}} f(x_k) + \frac{1}{3}f(x_{2m})\right]$$

In the same way the error is

$$E_m = -\frac{(b-a)^5}{2880m^4} f^{(4)}(\xi)$$

since each one of the m intervals has length $(b-a)/m$; again, ξ is a suitable point in $[a, b]$. As in the previous remark, if $f^{(4)}(x)$ does not change much in $[a, b]$, we have

$$\frac{E_{2m}}{E_m} \approx \frac{1}{16}.$$

6.3 Richardson extrapolation

Consider the composite trapezoidal rule. Assuming that the second derivative does not change much in the integration interval, we know that $E_{2m}/E_m \approx 1/4$. Defining Q_m and Q_{2m} approximations with m and $2m$ intervals, we can write

$$\begin{cases} Q_m &= I &+ 4E_{2m} \\ Q_{2m} &= I &+ E_{2m} \end{cases}$$

Solving for I we get the Richardson extrapolation formula (for the trapezoidal)

$$I = \frac{4Q_{2m} - Q_m}{3}$$

In the same way but using composite Cavalieri-Simpson formula, for which it is $E_{2m}/E_m \approx 1/16$, we find the Richardson extrapolation formula (for Cavalieri-Simpson)

$$I = \frac{16Q_{2m} - Q_m}{15}$$

6.3.1 Romberg method

Setting $m = 2^p$, $p = 0, 1, 2, \ldots$ it is possible to obtain a table (a subinterval is of the kind $[x_k, x_{k+1}]$)

p	number of subintervals	0	1	2	3	4	...
0	1	A_0					
1	2	A_1	B_1				
2	4	A_2	B_2	C_2			
3	8	A_3	B_3	C_3	D_4		
4	16	A_4	B_4	C_4	D_4	E_4	
...	...	...	...	...	...	...	...

where A_p are obtained using the composite trapezoidal formula and

$$\begin{aligned} B_p &= \frac{4A_p - A_{p-1}}{4^1 - 1}, \quad p \geq 1 \\ C_p &= \frac{16B_p - B_{p-1}}{4^2 - 1}, \quad p \geq 2 \\ D_p &= \frac{64C_p - C_{p-1}}{4^3 - 1}, \quad p \geq 3 \end{aligned}$$

and so on. It is possible to prove that the best approximation of the integral is on the right lower side of the table.

Exercise 6.1 *Consider the definite integral*

$$I = \int_1^2 x \, \ln(x) \, dx.$$

1. *Compute by hands the exact value of the integral I.*
2. *Using Cavalieri-Simpson formula, compute the approximation I_1 of the integral; give the maximum possible error of the approximation and the real error $E_1 = |I - I_1|$.*
3. *Using the composite Cavalieri-Simpson with $m = 2$, compute the approximation I_2 of the integral; give the maximum possible error of the approximation and the real error $E_2 = |I - I_2|$.*
4. *Using the Richardson extrapolation formula, find the best possible approximation of the integral and the corresponding error $E_R = |I - I_R|$.*
5. *Find and justify the ratios*
$$\frac{E_{max,1}}{E_{max,2}}, \qquad \frac{E_1}{E_2}.$$
6. *Find the best possible approximation of the integral using Romberg and $m = 4$ subdivision of the interval.*
7. *Find an estimation of the number m of intervals needed to have for sure an absolute value of the error less then 10^{-6}.*

Answer. *Using the integration by parts, we find*

$$\begin{aligned} \int_1^2 x \, \ln(x) \, dx &= \left[\frac{x^2}{2} \cdot \ln(x)\right]_1^2 - \int_1^2 \frac{x^2}{2} \cdot \frac{1}{x} \, dx = \frac{2^2}{2} \cdot \ln(2) - \frac{1^2}{2} \cdot \ln(1) - \frac{1}{2}\left[\frac{x^2}{2}\right]_1^2 \\ &= 2\ln(2) - \frac{1}{2} \cdot \left[\frac{2^2}{2} - \frac{1^2}{2}\right] = 2\ln(2) - \frac{3}{4} \approx 0.6362943611 \end{aligned}$$

since $\ln(1) = 0$. *The Cavalieri-Simpson gives*

$$I_1 = \frac{h}{3}\left[\, f(x_0) + 4f(x_1) + f(x_2) \,\right], \qquad h = \frac{b-a}{2} = \frac{2-1}{2} = \frac{1}{2}$$

with nodes $x_k = x_0 + k\,h$ *and* $f_k = f(x_k) = x_k \cdot \ln(x_k)$, $k = 0, 1, 2$ *given by*

k	0	1	2
x_k	1	$\frac{3}{2}$	2
f_k	0	$\frac{3}{2} \cdot \ln\left(\frac{3}{2}\right)$	$2 \cdot \ln(2)$

since $x_0 = a$, $x_2 = b$ *e* $x_1 = (a+b)/2$. *Thus, we obtain*

$$I_1 = \frac{1/2}{3} \cdot \left[0 + 4 \cdot \frac{3}{2} \cdot \ln\left(\frac{3}{2}\right) + 2 \cdot \ln(2)\right] \approx 0.6365141683.$$

The maximum (absolute value of the) error is

$$E_{max,1} = \frac{h^5}{90} \cdot \max_{\xi \in [a,b]} |f^{(4)}(\xi)| = \frac{(1/2)^5}{90} \cdot \frac{2}{1^3} \approx 6.9 \cdot 10^{-4}$$

since $f^{(4)}(x) = 2/x^3$ *is a decreasing function in* $[1,2]$ *and so has its maximum for* $\xi = 1$. *The real error is*

$$E_1 = |I - I_1| \approx |0.6362943611 - 0.6365141683| \approx 2.2 \cdot 10^{-4}.$$

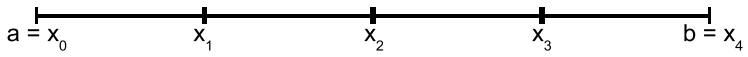
Now, take into account the composite Cavalieri-Simpson formula with $m = 2$. *The distance between consecutive nodes is*

$$h = \frac{1}{2} \cdot \frac{b-a}{m} = \frac{2-1}{2 \cdot 2} = \frac{1}{4}.$$

Setting $x_0 = a$, *nodes* $x_k = x_0 + k \cdot h$ *and values* $f(x_k)$, $k = 0, \cdots, 2m$ *are*

k	0	1	2	3	4
x_k	1	$\frac{5}{4}$	$\frac{6}{4}$	$\frac{7}{4}$	2
f_k	0	$\frac{5}{4} \cdot \ln\left(\frac{5}{4}\right)$	$\frac{3}{2} \cdot \ln\left(\frac{3}{2}\right)$	$\frac{7}{4} \cdot \ln\left(\frac{7}{4}\right)$	$2 \cdot \ln(2)$

The following figure give also a picture of the nodes.



Thus, we get

$$\begin{aligned} I_2 &= \frac{h}{3}\left[f_0 + 4\left(f_1 + f_3\right) + 2f_2 + f_4\right] \\ &= \frac{1/4}{3}\left\{0 + 4 \cdot \left[\frac{5}{4} \cdot \ln\left(\frac{5}{4}\right) + \frac{7}{4} \cdot \ln\left(\frac{7}{4}\right)\right] + 2 \cdot \frac{3}{2} \cdot \ln\left(\frac{3}{2}\right) + 2\ln(2)\right\} \\ &\approx 0.6363098298. \end{aligned}$$

The maximum error we can have is

$$E_{max,2} = \frac{(b-a)^5}{2880 \cdot m^4} \cdot \max_{\xi \in [a,b]} |f^{(4)}(\xi)| = \frac{(2-1)^5}{2880 \cdot 2^4} \cdot \frac{2}{1^3} \approx 4.3 \cdot 10^{-5}$$

whereas the real error we have is

$$E_2 = |I - I_2| \approx |0.6362943611 - 0.6363098298| \approx 1.5 \cdot 10^{-5}.$$

Since I_1 *and* I_2 *are computed using the composite Cavalieri-Simpson formula with* $m = 1$ *and* $m = 2$ *we can find a better approximation using Richardson extrapolation:*

$$I_R = \frac{16 \cdot I_2 - I_1}{15} = \frac{16 \cdot 0.6363098298 - 0.6365141683}{15} \approx 0.6362962072$$

with an error

$$E_R = |I - I_R| = |0.6362943611 - 0.6362962072| \approx 1.8 \cdot 10^{-6}.$$

The ratios between maximum errors and real errors are

$$\frac{E_{max,1}}{E_{max,2}} = \frac{6.9 \cdot 10^{-4}}{4.3 \cdot 10^{-5}} \approx 16.0, \qquad \frac{E_1}{E_2} = \frac{2.2 \cdot 10^{-4}}{1.5 \cdot 10^{-5}} \approx 14.7$$

as expected, since the ratio between maximum error is exactly 16 *whereas the other ratio depends also on the ratio of the derivative in two distinct points. Finally, Romberg method gives*

m			
1	A_0		
2	A_1	B_1	
4	A_2	B_2	C_2

where $B_1 = I_1 = 0.6365141683$ *and* $B_2 = I_2 = 0.6363098298$. *So the best approximation for the integral is*

$$I_{RG} = C_2 = \frac{16B_2 - B_1}{15} = I_R$$

For the last question, the maximum absolute value of the error using m *intervals is*

$$E_{max,m} = \frac{(b-a)^5}{2880\, m^4} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|$$

So, requiring that $E_{max,m} < \epsilon$ *with* $\epsilon = 10^{-6}$ *we find*

$$m > \sqrt[4]{\frac{(b-a)^5}{2880\,\epsilon} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|} = \sqrt[4]{\frac{(2-1)^5}{2880 \cdot 10^{-6}} \cdot \frac{2}{1^3}} \approx 5.1$$

Thus, we need $m = 6$.

Chapter 7

Simulations

The present chapter present some exercises which may be useful for the preparation of the exam.

7.1 Simulation 1

Multiple choice questions

Read carefully the text of each question and mark the box with the best answer.

1. How many floating point numbers has $\mathbb{F}(2, 3, -1, 1)$?

   ☐ 21 ☐ 24 ☐ 25 ☐ 27

2. The machine precision for $\mathbb{F}(10, 2, -3, 4)$ is

   ☐ 0.001 ☐ 0.01 ☐ 0.05 ☐ 0.1

3. The maximum number in $\mathbb{F}(10, 2, -3, 4)$ is

   ☐ 9.9 ☐ 99 ☐ 990 ☐ 9900

4. Let $x = 0.5$, $y = 1$ and $z = 10$ be floating point numbers of $\mathbb{F}(10, 1, -1, 2)$. Which is the result of $(x \oplus y) \otimes z$?

   ☐ 10 ☐ 15 ☐ 20 ☐ overflow

5. The problem of finding the solution of a non singular linear system $A\mathbf{x} = \mathbf{b}$ is always a well conditioned problem.

   ☐ True ☐ False

6. The absolute value of the error in the bisection method is always non increasing, i.e., $|e_{k+1}| \leq |e_k|$ for each $k \geq 0$

   ☐ True ☐ False

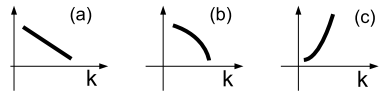
7. How many fixed points has the function $f(x) = x^3$?

   ☐ 0 ☐ 1 ☐ 2 ☐ 3

8. The fixed point iterations $x_{k+1} = x_k^2$ with $x_0 = 0.5$ goes toward the fixed point α equals to

   ☐ 0 ☐ 0.5 ☐ 1 ☐ $+\infty$

9. Consider the method $x_{k+1} = \phi(x_k)$ with fixed point $\alpha = 1$. Assume that x_k goes toward α and $\phi'(\alpha) = 0$. Then, the plot of $\log_{10}(|e_k|)$, choosing from the figure below, may be



☐ (a) ☐ (b) ☐ (c) ☐ (a) or (c)

10. The Newton method has the absolute value of the error $|e_5| = 10^{-6}$ for the computation of the root of $e^x + x = 0$. Assuming a unitary asymptotic error constant, we expect $|e_6|$ equals to about

☐ 10^{-6} ☐ 10^{-8} ☐ 10^{-10} ☐ 10^{-12}

11. The Newton method for the solution ξ of equation $f(x) = 0$ gives $|x_6 - x_5| = 10^{-3}$ and $|x_7 - x_6| = 2 \cdot 10^{-6}$. The estimation of the absolute value of the error $|e_7|$ is

☐ 10^{-12} ☐ $4 \cdot 10^{-12}$ ☐ $8 \cdot 10^{-12}$

12. The number of iterations required to the Newton method to compute the root of $3x - 2 = 0$ with an error not greater then 10^{-6} and starting at $x_0 = 2.0$ are

☐ 1 ☐ 14 ☐ 35 ☐ 68

13. The equation $f(x) = 0$ has a unique root ξ in $(0, 1)$. Assume that $f'(x) > 0$ and $f''(x) < 0$ in $[0, 1]$. Starting from $x_0 = 0$, the sequence x_k produced by the Newton method fulfills

☐ $x_k < x_{k+1}$ ☐ $x_k > x_{k+1}$ ☐ $x_k > \xi$

14. The spectral radius of the matrix

$$\begin{pmatrix} 2 & 2 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & -4 \end{pmatrix}$$

is

☐ -4 ☐ 2 ☐ 4 ☐ 24

15. The number of arithmetic operations required to the backward substitution algorithm to solve an upper triangular linear system is about

☐ n ☐ n^2 ☐ n^3 ☐ $\dfrac{2n^3}{3}$

16. The matrix U of the LU factorization of the matrix A

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

is

☐ $\begin{pmatrix} 2 & 1 \\ 0 & 1/2 \end{pmatrix}$ ☐ $\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$

☐ $\begin{pmatrix} 2 & 1 \\ 0 & 3/2 \end{pmatrix}$ ☐ $\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$

17. Each square matrix A may be written as $A = L \cdot U$ where L is lower triangular with unitary elements on the main diagonal and U is upper triangular

☐ True ☐ False

18. The matrix A has an LU factorization with the determinant of U equal to $|U| = -2$. The determinant of A^3 is

☐ -8 ☐ $-1/8$ ☐ 3

☐ we cannot compute since we not have A

19. Consider the linear system $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

☐ $\|A\|_F = 10$ ☐ GS converges
☐ A is singular ☐ $\rho(A) < 1$

20. The condition number $K_2(A)$ of the matrix A of the previous item is

☐ 1 ☐ 2 ☐ 3 ☐ 4

21. The following Matlab code

```
1.   v = 2:3:10;
2.   A = [ v; v + 1 ];
```

☐ is wrong in line 2 ☐ $A = \begin{pmatrix} 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$

☐ $A = \begin{pmatrix} 2 & 5 & 8 \\ 3 & 5 & 8 \end{pmatrix}$ ☐ $A = \begin{pmatrix} 2 & 5 & 8 \\ 2 & 5 & 9 \end{pmatrix}$

22. Consider the following Matlab code

```
1.   v = linspace(2,10,5);
2.   w = length( size( v' * v ) );
```

The variable `w` is

☐ 2 ☐ 3 ☐ [2 2] ☐ $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$

23. Consider the following Matlab code

```
1.   S = 0;
2.   for k=1:4
3.      S = S + k;
4.      S = 2 * S;
5.   end
```

At the end of the loop, the variable `S` is equal to

☐ 0 ☐ 6 ☐ 26 ☐ 52

24. Which is `B` at the end of the Matlab code

```
1.   A = [1 2 3; 4 5 6];
2.   B = A( :, 1:2 ).^2;
```

□ $\begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$ □ $\begin{pmatrix} 1 & 4 \\ 16 & 25 \end{pmatrix}$

□ $\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$ □ $\begin{pmatrix} 9 & 12 \\ 24 & 33 \end{pmatrix}$

25. Which is the value of `n` at the end of the Matlab code

```
1.   n = 0;
2.   while( n<=3 )
3.      n = n + 2;
4.      if n >= 4
5.         n = n-1;
6.      end
7.   end
```

□ 3 □ 4 □ 5 □ 6

Open questions

Write clearly all the answers in the exam's booklet.

1. (**10 points**) Describe the bisection method. Then, apply it to the function $f(x) = x-1$ with starting interval $[a_0, b_0] = [0,\ 1.9]$. Compute x_0, x_1 and the corresponding errors and justify the obtained result.

2. (**10 points**) Consider the Newton method for the approximation of the root $\xi = 1$ of the function $f(x) = x^2 - 1$. Assume the starting point to be $x_0 = 2$.

   (a) Compute the first two iterations of the Newton method and the corresponding absolute value of the errors.

   (b) Sketch a *qualitative* graph of $\log_{10}(\,|e_k|\,)$ as a function of the iteration number k. Justify your answer.

   (c) What happens if we choose as new starting point $x_0 = 0$? Justify your answer.

   (d) Write a fixed point method of order $p = 2$ for the computation of the root ξ. Justify your answer.

3. (**20 points**) Consider the upper triangular linear system $U\mathbf{x} = \mathbf{b}$.

   (a) Show, with all the mathematical details, the backward substitution method.

   (b) Show, with all details, the computational cost of the method.

   (c) Write the Matlab code to solve such a linear system using the for loop.

4. (**20 points**) Consider the linear system $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix}$$

(a) (**10 points**) Prove that the matrix A is positive definite, compute the Cholesky factorization and solve the linear system. Check out that the obtained solution is correct.

(b) (**2 points**) Prove that the Jacobi method converges to the solution starting from $\mathbf{x_0} = [0\ \ 0\ \ 0]^T$.

(c) (**8 points**) Compute the first two iterations of the Gauss-Seidel method starting from $\mathbf{x_0} = [0\ \ 0\ \ 0]^T$.

7.2 Simulation 2

Multiple choice questions

Read carefully the text of each question and mark the box with the best answer.

1. The maximum floating point number x_{max} and the machine precision `eps` of the floating point system $\mathbb{F}(10, 2, -1, 1)$ are

$\square$ $x_{max} = 9.9$ `eps` $= 0.05$
$\square$ $x_{max} = 9.0$ `eps` $= 0.10$
$\boxtimes$ $x_{max} = 9.9$ `eps` $= 0.05$
$\square$ $x_{max} = 9.0$ `eps` $= 0.10$

Answer. The floating point system has $\beta = 10$, $t = 2$, $L = -1$, $U = 1$. Thus, the maximum number and the machine precision are

$$x_{max} = \beta^U \cdot \left(1 - \beta^{-t}\right) = 10^1 \cdot \left(1 - 10^-2\right) = 9.9$$

$$\text{eps} = \frac{\beta^{1-t}}{2} = \frac{10^{1-2}}{2} = 0.05$$

2. Consider the fixed point iterations given by $x_{k+1} = x_k/2 + 1$. Let α be the unique fixed point. Starting at $x_0 = 1$, the absolute value of the error $e_2 = \alpha - x_2$ of the second iteration x_2 is

$\boxtimes$ 0.25 $\square$ 0.50 $\square$ 1.0 $\square$ 1.5

Answer. The iteration function is $\phi(x) = \frac{x}{2} + 1$; its fixed points are solutions of $x = \phi(x)$. We get $x = 2$ and so ϕ has the unique fixed point $\alpha = 2$. Starting from $x_0 = 1$, the first two iterations are

$$x_1 = \phi(x_0) = \frac{x_0}{2} + 1 = \frac{1}{2} + 1 = \frac{3}{2}.$$

$$x_2 = \phi(x_1) = \frac{x_1}{2} + 1 = \frac{3/2}{2} + 1 = \frac{7}{4}.$$

So, we have

$$| e_2 | = | \alpha - x_2 | = | 2 - \frac{7}{4} | = \frac{1}{4}.$$

3. The order of convergence of the fixed point method $x_{k+1} = 2 - 2x_k + x_k^2$ when $x_0 = 0.5$ is

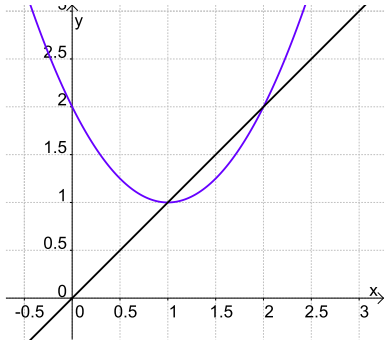
$\square$ 1 $\boxtimes$ 2 $\square$ 3 $\square$ 4

Answer. The iteration function is $\phi(x) = 2 - 2x + x^2$. The corresponding fixed points are solutions of $x = \phi(x)$. We have

$$x = \phi(x) \quad \Leftrightarrow \quad x = 2 - 2x + x^2 \quad \Leftrightarrow \quad x^2 - 3x + 2 = 0$$

which has two solutions $x_1 = 1$ and $x_2 = 2$. Thus, the function ϕ has two fixed points: $\alpha_1 = 1$ and $\alpha_2 = 2$. So, first of all, we have to find toward which one of the two go the

fixed point iterations when we start at $x_0 = 0.5$. To this aim, it is useful the geometric interpretation.



From the figure, since $x_0 = 0.5$. we see that the iterations go toward $\alpha = 1$. So, to find the order we have to look derivatives of ϕ in $\alpha_1 = 1$. We have

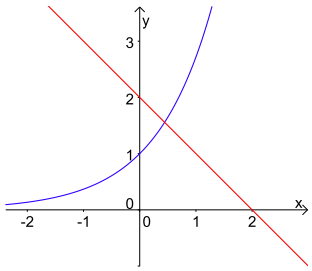
$$\begin{aligned} \phi'(x) = 2x - 2 \quad &\Rightarrow \quad \phi'(1) = 0 \\ \phi''(x) = 2 \quad &\Rightarrow \quad \phi''(1) \neq 0 \end{aligned}$$

Thus, the order of the method is $p = 2$ since the first non zero derivative (evaluated in $\alpha_1 = 1$) of ϕ has order $p = 2$.

4. The order of convergence of the Newton method for the solution of the non linear equation $e^x = 2 - x$ is

$$\square\ 0.5 \qquad \square\ 1 \qquad \boxtimes\ 2 \qquad \square\ \text{more then 2}$$

Answer. The equation has only one root since graphs $y = e^x$ and $y = 2 - x$ intersects just once. Moreover, the root is positive.



Setting $f(x) = e^x + x - 2$, we have $f'(x) = e^x + 1$ and $f''(x) = e^x$. Thus, we have

- $f'(\xi) = e^{\xi} + 1 > 0$. So, we have $f'(\xi) \neq 0$: this means that the root is simple (or, it has multiplicity 1) and so the order p of the method satisfies $p \geq 2$;
- $f''(\xi) > 0$. So, we have $f''(\xi) \neq 0$: as a consequence, the order ix exactly $p = 2$.

So, the Newton method for approximating the root ξ

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{e^{x_k} + 1}{e^{x_k}} = x_k - 1 - e^{-x_k}$$

has order of convergence $p = 2$ providing the starting point x_0 is sufficiently near the root ξ. We can see this behavior taking $x_0 = 1.0$; we have, for the errors

k	0	1	2	3	4
$\lvert e_k \rvert$	$5.5 \cdot 10^{-1}$	$9.5 \cdot 10^{-2}$	$2.8 \cdot 10^{-3}$	$2.3 \cdot 10^{-6}$	$1.6 \cdot 10^{-12}$

5. The order of convergence of the Newton method is always less or equal to 2

   ☐ True ☒ False

   Answer. It's false. For example, if $f(x) = 0$ has the root ξ with $f'(\xi) \neq 0$ (and so $p \geq 2$) and $f''(\xi) = 0$ the order is $p \geq 3$. Consider $f(x) = x^3 + x$ which has the unique root $\xi = 0$. Starting from $x_0 = 1$, the behavior of the errors are

k	0	1	2	3	4	5
$\lvert e_k \rvert$	1.0	0.5	0.14	$5.5 \cdot 10^{-3}$	$3.3 \cdot 10^{-7}$	$7.7 \cdot 10^{-20}$

6. The Hilbert matrices are an example of well conditioned matrices

   ☐ True ☒ False

   Answer. It's false. The Hilbert matrices, as well as the Vandermonde matrices, are examples of ill conditioned matrices. For example, the Hilbert matrix H_5 of order $n = 5$ has a condition number $K_2(H_5) \approx 5 \cdot 10^5$. An example of well conditioned matrix is the identity matrix which has a condition number equal to 1, the less possible value.

7. Let

$$A = \begin{pmatrix} 10 & 0 \\ 0 & 0.01 \end{pmatrix}$$

   The condition number $K_2(A)$ of the matrix A is

   ☐ 0.01 ☐ 10 ☐ 100 ☒ 1000

   Answer. The matrix A is a diagonal matrix with positive entries in the main diagonal. So, is a positive definite matrix, since all the eigenvalues are positive. For a positive definite matrix, we know that $K_2(A) = \lambda_{max}/\lambda_{min}$. In our case, we have $\lambda_{max} = 10$ and $\lambda_{min} = 0.01$. So, $K_2(A) = 10/0.01 = 1000$.

8. The LU factorization of the matrix A gives $|U| = 4$. The determinant of A^{-2} is

   ☐ 16 ☒ $\frac{1}{16}$ ☐ $\frac{1}{4}$ ☐ 4

   Answer. We have, using the Binet formula and the relation $\mid A^{-1} \mid = 1/\mid A \mid$,

$$\mid A^{-2} \mid = \mid (A^{-1})^2 \mid = \mid A^{-1} \mid^2 = \left(\frac{1}{\mid A \mid} \right)^2 = \frac{1}{\mid A \mid^2} = \frac{1}{\mid U \mid^2} = \frac{1}{4^2} = \frac{1}{16}$$

   since, from the LU factorization of $A = LU$, we have

   - L is a lower triangular matrix with ones on the main diagonal. So, its determinant, which is the product of all the elements in the main diagonal, is $\mid L \mid = 1$.
   - U is an upper triangular matrix. The determinant of U is again the product of the elements in the main diagonal but now this values are not known a priori (they depends on the matrix A)

   So, again, from Binet, we have

$$\mid A \mid = \mid LU \mid = \mid L \mid \cdot \mid U \mid = 1 \cdot \mid U \mid = \mid U \mid$$

9. The L matrix of the LU-factorization of the matrix A given by

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 3 & 2 & 4 \end{pmatrix}$$

is

$$\boxtimes \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix} \quad \square \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & -2 & 1 \end{pmatrix}$$

$$\square \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -2 & 1 \end{pmatrix} \quad \square \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 2 & 1 \end{pmatrix}$$

Answer. Using the Gauss algorithm we find

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 3 & 2 & 4 \end{pmatrix} \overset{\substack{(-2) \\ (-3)}}{\longrightarrow} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -2 \\ 0 & 2 & 1 \end{pmatrix} \overset{(-2)}{\longrightarrow} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -2 \\ 0 & 0 & 5 \end{pmatrix}$$

So, recalling that L is a lower triangular matrix and has as entries the multiplicators (the elements above the arrows) changed in sign, and ones in the main diagonal, we get

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

10. Given the splitting $A = D - E - F$ where E is strictly lower triangular, F is strictly upper triangular and D is diagonal, the iteration matrix for the Jacobi method is

$$\boxtimes \ D^{-1}(E+F) \qquad \square \ D(E+F)$$

$$\square \ -D^{-1}(E+F) \qquad \square \ -D(E+F)$$

Answer. From the theory, providing that the entries of the diagonal of D are all non singular, we know that the iteration matrix of the Jacobi method is $B_J = D^{-1}(E+F)$. Indeed, just write

$$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad (D - E - F)\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad D\mathbf{x} = (E+F)\mathbf{x} + \mathbf{b} \quad \Leftrightarrow$$

Providing that D is invertible, we get

$$\mathbf{x} = D^{-1}(E+F)\mathbf{x} + D^{-1}\mathbf{b} \quad \text{and so we get} \quad \mathbf{x}_{k+1} = D^{-1}(E+F)\mathbf{x}_k + D^{-1}\mathbf{b}$$

where $B_J = D^{-1}(E+F)$ is the iteration matrix. Exactly in the same way we can find the iteration matrix of the Gauss-Seidel method: just start from $(D-E)\mathbf{x} = F\mathbf{x} + \mathbf{b}$.

11. Starting from $\mathbf{x}_0 = (0, 0)^T$, the norm of the residual $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1$ after the first Gauss-Seidel iteration for the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

is

$$\square \ \frac{\sqrt{130}}{6} \qquad \boxtimes \ \frac{7}{6} \qquad 0 \qquad \left[-\frac{7}{6}, \ 0 \right]^T$$

Answer. Setting $\mathbf{x} = [x_1 \ x_2]^T$, the linear system is

$$\begin{cases} 2x_1 + x_2 = 3 \\ -x_1 + 3x_2 = 2 \end{cases} \quad \Rightarrow \quad \begin{cases} x_1 = \frac{3 - x_2}{2} \\ \\ x_2 = \frac{2 + x_1}{3} \end{cases}$$

Denoting with a superscript the index of the iteration, the Gauss-Seidel iterations are

$$x_1^{(k+1)} = \frac{3 - x_2^{(k)}}{2}$$

$$x_2^{(k+1)} = \frac{2 + x_1^{(k+1)}}{3}$$

So, starting from $\mathbf{x}_0 = [0\ 0]^T$ (that is, $x_1^{(0)} = 0$ and $x_2^{(0)} = 0$) we get for $\mathbf{x}_1 = [x_1^{(1)}\ x_2^{(1)}]^T$

$$x_1^{(1)} = \frac{3 - x_2^{(0)}}{2} = \frac{3-0}{2} = \frac{3}{2}$$

$$x_2^{(1)} = \frac{2 + x_1^{(1)}}{3} = \frac{2 + 3/2}{3} = \frac{7}{6}$$

The corresponding residual vector $\mathbf{r}_1$ is

$$\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1^{(1)} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix} \cdot \begin{pmatrix} 3/2 \\ 7/6 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 25/6 \\ 2 \end{pmatrix} = \begin{pmatrix} -7/6 \\ 0 \end{pmatrix}$$

Finally, the infinity norm of the residual $\mathbf{r}_1$ is (we obtain the same result using other norms)

$$\| \mathbf{r}_1 \|_\infty = \max \left\{ \left| -\frac{7}{6} \right|, \ | 0 | \right\} = \frac{7}{6}.$$

12. The Lagrange polynomials depends only on the nodes of the points (x_i, y_i), $i = 0, \ldots, n$

☒ True ☐ False

Answer. It is true: if x_i, $i = 0, \ldots, n$ are the nodes, then the Lagrange polynomials are

$$l_i(x) = \frac{\prod_{k=0, k\neq i}^{n} (x - x_k)}{\prod_{k=0, k\neq i}^{n} (x_i - x_k)}, \quad i = 0, 1, \ldots, n$$

13. The sum of all the Lagrange polynomials depends on the values of the function f in the interpolating points

☐ True ☒ False

Answer. It is false: from the theory, it is known that the sum of all the Lagrange polynomials is (the constant function) 1. To prove, just take $f(x) = 1$, a polynomial of degree zero. Thus,

$$f(x) = 1 = \sum_{i=0}^{n} f(x_i) l_i(x) = \sum_{i=0}^{n} 1 \cdot l_i(x) = \sum_{i=0}^{n} l_i(x)$$

14. If we want to approximate a function in an interval $[a, b]$ using equally spaced nodes, a higher degree interpolating polynomial always works better then a lower degree one

☐ True ☒ False

Answer. It is false: just remember the Runge example where the error (at the endpoints of the interval) increases with the degree of the interpolating polynomial.

15. The regression line for the set of points

x_i	-1	0	1	2
y_i	0	2	3	3

is

☒ $y = x + 1.5$ ☐ $y = 1.5x + 1$

☐ $y = x + 1$ $y = 1.5x + 1.5$

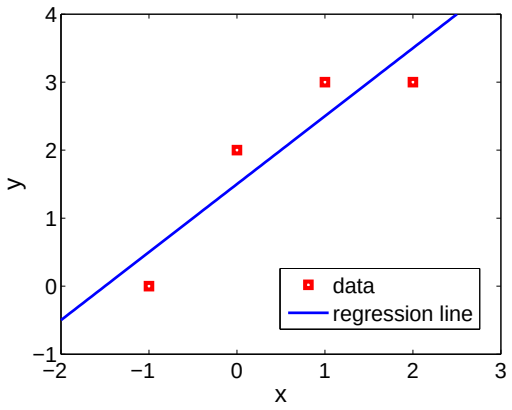
Answer. Coefficients a_0 and a_1 of the regression line $y = a_0 + a_1 x$ are solution of the linear system

$$\begin{pmatrix} m+1 & \sum_{k=0}^{m} x_k \\ \sum_{k=0}^{m} x_k & \sum_{k=0}^{m} x_k^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{m} y_k \\ \sum_{k=0}^{m} x_k y_k \end{pmatrix}$$

where $m+1$ is the number of points. In this case $m+1=4$ and so, using the data in the table, we have

$$\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 8 \\ 9 \end{pmatrix} \quad \Leftrightarrow \quad \begin{cases} 4a_0 + 2a_1 = 8 \\ 2a_0 + 6a_1 = 9 \end{cases}$$

We find $a_0 = 3/2$ and $a_1 = 1$. We can see the regression line in the next figure.



16. The quadrature formula

$$\int_0^1 \sqrt{x} f(x) \approx \sum_{i=0}^{3} \alpha_i f(x_i)$$

has the maximum degree of precision. Then the number

$$d = \left| \int_0^1 \left(\sqrt{x} + x\sqrt{x} \right) dx \; - \sum_{i=0}^{3} \alpha_i x_i \right|$$

is equal to

☐ 0 ☐ 1/3 ☒ 2/3 ☐ 1

Answer. The maximum degree of precisionof the quadrature formula is $s = 2 \cdot 3 + 1 = 7$. Since the integral in the quadrature formula is of the type

$$\int_0^1 \omega(x)\, f(x) dx$$

with $\omega(x) = \sqrt{x}$, the fist step to do is to rewrite the integral inside the expression of d in this way. We may note that

$$\sqrt{x} + x\sqrt{x} = \sqrt{x}\,(1+x) \quad \Rightarrow \quad f(x) = 1 + x$$

So, f is a polynomial of degree $n = 1 < 7 = s$. Thus, the quadrature formula gives the exact result for this function f, i.e.,

$$\int_0^1 \left(\sqrt{x} + x\sqrt{x} \right) dx \; = \sum_{i=0}^{3} \alpha_i \,(1 + \; x_i \,)$$

Looking again to d, we have

$$d \;\; = \;\; \left| \int_0^1 \left(\sqrt{x} + x\sqrt{x} \right) dx \; - \sum_{i=0}^{3} \alpha_i x_i \right|$$

$$= \left| \int_0^1 \left(\sqrt{x} + x\sqrt{x} \right) dx - \sum_{i=0}^{3} \alpha_i \left(1 + x_i - 1 \right) \right|$$
$$= \left| \left\{ \int_0^1 \left(\sqrt{x} + x\sqrt{x} \right) dx - \sum_{i=0}^{3} \alpha_i \left(1 + x_i \right) \right\} - \sum_{i=0}^{3} \alpha_i \right| = \left| \sum_{i=0}^{3} \alpha_i \right|$$

Let's compute the sum of the weights. Taking in the quadrature formula $f(x) = 1$ (a polynomial of degree $n = 0 < s$ and so the formula is correct) we have

$$\frac{2}{3} = \left[\frac{x^{3/2}}{3/2} \right]_0^1 = \int_0^1 \sqrt{x}\, dx = \sum_{i=0}^{3} \alpha_i$$

So, we have $d = 2/3$.

17. Given a positive n, the sum of all Cotes numbers $C_i^{(n)}$, $i = 0, \ldots, n$ is

☒ 1 ☐ n ☐ $\sqrt{n}$ ☐ $\frac{n}{2}$

Answer. From the theory, we know that the sum of all the Cotes numbers is 1. Recall that we have given the following definition of Cotes numbers

$$C_i^{(n)} = \frac{1}{n} \int_0^n \frac{\prod_{r=0, r\neq i}^{n} (s - r)}{\prod_{r=0, r\neq i}^{n} (i - r)} ds, \quad i = 0, \ldots, n$$

18. The error for the computation of

$$\int_0^{100} \left(x^3 + 13254x \right) dx$$

using the Cavalieri-Simpson formula is

☐ 10^{-3} ☐ 10^{-2} ☐ 10^{-1} ☒ 0

Answer. The error E in the Cavalieri-Simpson is related to $f^{(4)}(x)$ throughout the equation

$$E = -\frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

where, in our case, $a = 0$, $b = 100$, $\xi \in [0, 100]$ and $f(x) = x^3 + 13254x$. Since $f^{(4)}(x) = 0$ for all x, it is $f^{(4)}(\xi) = 0$ and so the error is zero.

19. The second derivative of f does not change much in the integration interval. Then, using the composite trapezoidal rule we expect that the ratio of the errors E_{2m}/E_m is

☐ 4 ☐ 1/4 ☒ near 1/4

Answer. From the theory, we know that

$$\frac{E_{2m}}{E_m} = \frac{-\frac{(b-a)^3}{12(2m)^3} f''(\xi_{2m})}{-\frac{(b-a)^3}{12m^3} f''(\xi_m)} = \frac{f''(\xi_{2m})}{4 \cdot f''(\xi_m)} \approx \frac{1}{4}.$$

since, if $f''(x)$ does not change much in the integration interval, it is $f''(\xi_m) \approx f''(\xi_{2m})$.

20. The Cavalieri-Simpson approximation of the integral

$$\int_0^1 \sqrt{x}\, dx$$

is

$$\boxtimes\ \frac{2\sqrt{2}+1}{6} \qquad \square\ \frac{2}{3} \qquad \square\ \frac{1}{6} \qquad \square\ 1$$

Answer. We have

$$x_0 = a = 0, \qquad x_1 = \frac{a+b}{2} = \frac{1}{2}, \qquad x_2 = b = 1$$

and so the Cavalieri-Simpson rule gives for the integral I

$$\begin{aligned} I_{CS} &= \frac{b-a}{2}\left[\frac{1}{3}f(x_0)+\frac{4}{3}f(x_1)+\frac{1}{3}f(x_2)\right] \\ &= \frac{1-0}{2}\left[\frac{1}{3}\sqrt{0}+\frac{4}{3}\sqrt{\frac{1}{2}}+\frac{1}{3}\sqrt{1}\right] = \frac{\sqrt{2}}{3}+\frac{1}{6} = \frac{2\sqrt{2}+1}{6} \end{aligned}$$

So, we have $I_{CS} = 0.638$ which may be compared with the correct value $I = 2/3 = 0.667$.

21. The divided difference $f[x_0, x_1, x_2]$ of the following table

$$\begin{array}{r|l} x_0 = -1 & 2 \\ x_1 = 0 & 3 \\ x_2 = 1 & 6 \end{array}$$

is

$$\boxtimes\ 1 \qquad \square\ 3 \qquad \square\ -1 \qquad \square\ 36$$

Answer. Completing the table, we find

$$\begin{array}{r|lll} x_0 = -1 & 2 & & \\ x_1 = 0 & 3 & f[x_0,x_1] & \\ x_2 = 1 & 6 & f[x_1,x_2] & f[x_0,x_1,x_2] \end{array} \qquad \text{or} \qquad \begin{array}{r|lll} x_0 = -1 & 2 & & \\ x_1 = 0 & 3 & 1 & \\ x_2 = 1 & 6 & 3 & 1 \end{array}$$

since we have

$$\begin{aligned} f[x_0,x_1] &= \frac{f(x_1)-f(x_0)}{x_1-x_0} = \frac{3-2}{0-(-1)} = 1 \\ f[x_1,x_2] &= \frac{f(x_2)-f(x_1)}{x_2-x_1} = \frac{6-3}{1-0} = 3 \\ f[x_0,x_1,x_2] &= \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = \frac{3-1}{1-(-1)} = 1 \end{aligned}$$

22. Let $p(x)$ be the Newton expression of the interpolating polynomial for the points (x_i, y_i), $i = 0, \ldots, n$. If we add a new point (x_{n+1}, y_{n+1}) with $x_0 < x_{n+1} < x_1$ we have to recompute all the divided difference table

$$\square\ \text{True} \qquad \boxtimes\ \text{False}$$

Answer. It is false: the difference divided does not depends on the order of points. So, we can add the point (x_{n+1}, y_{n+1}) at the previous table (the one we have already constructed with points (x_i, y_i), $i = 0, \ldots, n$) and compute just the last row of the new table.

23. The composite trapezoidal formula gives the results of the following table

A_0	A_1	A_2
1	0.875	0.844

The best approximation for the integral is then

$$\square\ 0.844 \qquad \boxtimes\ 0.833 \qquad \square\ 0.906 \qquad \square\ 0.875$$

Answer. We can apply the Romberg method to obtain

m			
1	A_0		
2	A_1	B_1	
4	A_2	B_2	C_2

where

$$B_1 = \frac{4A_1 - A_0}{3} = \frac{4 \cdot 0.875 - 1}{3} = 0.833$$

$$B_2 = \frac{4A_2 - A_1}{3} = \frac{4 \cdot 0.844 - 0.875}{3} = 0.8340.833$$

$$C_2 = \frac{16B_2 - B_1}{15} = \frac{16 \cdot 0.834 - 0.833}{15} = 0.834$$

So, the best approximation for the integral is 0.834.

24. Which is the value of `n` at the end of the Matlab code

```
1.  toll = 1E2;
2.  n = 5;
3.  while( 10^n > toll & n >= 2 )
4.     n = n - 2;
5.  end
```

☐ 0 ☒ 1 ☐ 2 ☐ 3

25. Consider the following Matlab code

```
1.  S = 5;

2.  for k=1:3
3.     if k>=3
4.        S = S*k;
5.     else
6.        S = S-k;
7.     end
5.  end
```

At the end of the loop, the variable `S` is equal to

☐ 1 ☐ 4 ☒ 6 ☐ 9

26. After the execution of the following Matlab code, the variable `r` is equal to

```
1.  A = diag( diag( [1 2; 3 4] ) );
2.  r = eig( A );
```

☒ $[1\ 4]^T$ ☐ $[1\ 2]^T$ ☐ 4 ☐ 1

27. After the execution of the following Matlab code, the variable `v` is equal to

```
1.   v = [ sum( 3:4:14 )  length( 1:4 )  ];
2.   v = v.^2;
```

☒ [441 16] ☐ 7056 ☐ 4 ☐ [21 4]

28. Given the Matlab code

```
1.   v = [ 1 2 3; 4 5 6; 6 7 8 ];
2.   v = v(2,[2 3]);
```

gives

☒ [5 6] ☐ [4 5 6] ☐ $[5\ 8]^T$ ☐ $[2\ 5]^T$

29. To plot a function with the command `plot(x,y)`, the vector `x` and `y` must have the same size

☒ True
☐ False
☐ It depends on the function

30. The command `clear all` makes the command Window clear but does not clear the variables in the Workspace

☐ True ☒ False

Open questions

Write clearly all the answers in the exam's booklet.

1. Prove that the condition number $K(A)$ fulfills $K(A) \geq 1$ for each matrix A. Give an example of well conditioned matrix and one of an ill conditioned matrix.

   Answer. From the theory, we have

   $$1 = \|I_n\| = \left\|A \cdot A^{-1}\right\| \leq \|A\| \cdot \left\|A^{-1}\right\| = K(A).$$

   where I_n is the identity matrix of order n. The Hilbert matrices are ill conditioned whereas the identity matrix is well conditioned.

2. Consider the iterative method $\mathbf{x}_{k+1} = B\mathbf{x}_k + \mathbf{f}$ to solve the linear system $A\mathbf{x} = \mathbf{b}$. Prove the relationship $\mathbf{e}_k = B\mathbf{e}_{k-1}$, $k = 1, 2, \ldots$ where $\mathbf{e}_k = \mathbf{x} - \mathbf{x}_k$ is the error at the k-th step. Give necessary and sufficient conditions on the iteration matrix B in order to have a convergent sequence for each starting point $\mathbf{x}_0$. Write the iteration matrix for the Jacobi method.

   Answer. The iterative method has solution $\mathbf{x}$ given by $\mathbf{x} = B\mathbf{x}+\mathbf{f}$ Defining $\mathbf{e}_k = \mathbf{x}-\mathbf{x}_k$, $k = 0, 1, \ldots$, we get

   $$\mathbf{x} - \mathbf{x}_k = B\mathbf{x} + \mathbf{f} - (\ B\mathbf{x}_k + \mathbf{f}\ ) \quad \Leftrightarrow \quad \mathbf{e}_k = B\mathbf{e}_{k-1}$$

   From this relation, iterating we get $\mathbf{e}_k = B^k\mathbf{e}_0$ and so

   $$\lim_{k\to+\infty} \mathbf{e}_k = \mathbf{0}\ \forall\ \mathbf{e}_0 \quad \Leftrightarrow \quad \lim_{k\to+\infty} B^k = 0 \quad \Leftrightarrow \quad \rho(B) < 1.$$

   Thus, the iteration method converges for all $\mathbf{x}_0$ if and only if $\rho(B) < 1$. For the Jacobi method we use the splitting of A given by $A = D - E - F$ where $-E$ is the strictly lower triangular part of A, $-F$ is strictly upper triangular part of A and D is the diagonal of A. Thus, we have

   $$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad (D-E-F)\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad D\mathbf{x} = (E+F)\mathbf{x}+\mathbf{b} \quad \Leftrightarrow \quad \mathbf{x} = D^{-1}(E+F)\mathbf{x}+D^{-1}\mathbf{b}$$

where the latter step requires a non singular D (and so the main diagonal of A has all elements different from zero). From the latter equation, we build the Jacobi iterative method setting

$$\mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{f} \qquad \text{where} \qquad B_J = D^{-1}(E+F), \quad \mathbf{f} = D^{-1}\mathbf{b}$$

3. Given the set of points $(\ x_i,\ y_i\ )$, $i = 0, \ldots, 3$ in the following table

x_i	-2	0	1	2
y_i	1	1	2	3

write the Newton expression of the interpolation polynomial. Compute the minimum value of the function $S(m,q)$

$$S(m,q) = \sum_{k=0}^{3} [\ y_i - mx_i - q\ ]^2$$

and give the values of m and q for which this minimum is reached.

Answer. We need the table of divided differences first.

$$\begin{array}{llllll} x_0 = -2 & f(x_0) = 1 & & & \\ x_1 = 0 & f(x_1) = 1 & f[x_0,x_1] = 0 & & \\ x_2 = 1 & f(x_2) = 2 & f[x_1,x_2] = 1 & f[x_0,x_1,x_2] = 1/3 & \\ x_3 = 2 & f(x_3) = 3 & f[x_2,x_3] = 1 & f[x_1,x_2,x_3] = 0 & f[x_0,x_1,x_2,x_3] = -1/12 \end{array}$$

So, we have

$$\begin{aligned} P(x) &= f(x_0) + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) \\ &+ f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2) \\ &= 1 + \frac{1}{3}(x+2)x - \frac{1}{12}(x+2)x(x-1) \end{aligned}$$

The minimum value of the sum $S(m,q)$ is reached for m and q associated to the regression line. So, we get

$$\begin{pmatrix} m+1 & \sum_{k=0}^{m} x_k \\ \sum_{k=0}^{m} x_k & \sum_{k=0}^{m} x_k^2 \end{pmatrix} \begin{pmatrix} q \\ m \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{m} y_k \\ \sum_{k=0}^{m} x_k y_k \end{pmatrix}$$

We have $m+1 = 4$ points; the linear system gives the solution $m = 17/35$ and $q = 57/35$. The minimum value of the sum is thus

$$S(17/35,\ 57/35) = \sum_{k=0}^{3} \left[\ y_i - \frac{17}{35}x_i - \frac{57}{35}\ \right]^2 \approx 0.6857$$

4. Show the composite trapezoidal rule using m intervals . Recalling that the error for the trapezoidal rule is

$$E = -\frac{h^3}{12}f''(\xi)$$

where h is the amplitude of the integration interval and ξ is a suitable point inside the integration interval, find the expression for the error in the composite trapezoidal formula.

Answer. See the theory.

5. Write a Matlab code for the computation of the sum of elements of the vector `x` using just a for loop.

Answer. We may write, for example,

```
1.  function s = sumvett( x )
2.  %SUMVETT sum of elements of vector
3.  n = length(x);
4.  s = 0;
5.  for k = 1:n
6.       s = s+ x(k);
7.  end
```

7.3 Exam of August 2014

SECTION A: Multiple choice questions

Read **carefully** the text of each question and write in the booklet the letter which corresponds to the best answer.

1. Consider the floating point system $\mathbb{F}(2, 4, -2, 2)$. The maximum floating point number x_{max} and the number of elements[1] N_{el} of $\mathbb{F}$ are

   A $x_{max} = 3.75$ $N_{el} = 80$ B $x_{max} = 3.75$ $N_{el} = 81$

   C $x_{max} = 4.00$ $N_{el} = 80$ D $x_{max} = 4.00$ $N_{el} = 81$

2. Noting that 1 and 2 are two consecutive numbers in $\mathbb{F}(10, 1, -1, 1)$, the representation of $x = 1.68$ in $\mathbb{F}$ using rounding is

   A 1.6 B 1.7 C 2 D overflow

3. The bisection method is used to approximate the root ξ of the equation $x - 1 = 0$ starting from the interval $[\, a_0,\ b_0\,] = [\, 0.8,\ 1.6\,]$. Denoting by x_0 and x_1 the first two iterates, the error $e_1 = \xi - x_1$ is

   A 0 B 0.1 C 0.2 D 0.4

4. How many fixed points has the function $f(x) = 2 - x^2$?

   A 0 B 1 C 2 D 4

5. Consider the fixed point iterations

   $$\begin{cases} x_0 &= 1.5 \\ x_{k+1} &= x_k^2 - 2x_k + 2 \end{cases}$$

   The order p of convergence of the iterates x_k, $k = 0, 1, \ldots$ is

   A 1 B 2 C 3 D does not converge

6. The number of iterations needed by the Newton method to find the approximation x_k of the root ξ of the equation $2x - 3 = 0$ starting from $x_0 = 2$ and with an absolute value of the error $|\xi - x_k| < 10^{-6}$ is

   A 1 B 4 C 9 D infinity

7. The Newton method for the approximation of the root $\xi = 1$ of the equation $(x - 1)^2 \ln(x) = 0$ has order of convergence p equal to

   A 1 B 2 C 3 D more then 3

[1] That is, how many elements has the set $\mathbb{F}$. For example, $\mathbb{F} = \{-1, 0, 1\}$ has 3 elements.

8. The Newton method for the approximation of the root ξ of the equation $xe^x = 1$ has order of convergence p equal to

   A 1 B 2 C 3 D more then 3

9. Find the condition number $K_2(A)$ of

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

   A 0.1 B 1 C 10 D $\sqrt{101}/10$

10. Let $A = L\,U$ the LU-factorization of the non singular matrix A. If L and U are known, the smallest number of operations needed to solve all linear systems

$$A\mathbf{x} = \mathbf{b}_k, \quad k = 1, \ldots, M$$

    is about

    A n^2 B $M\,n^2$ C $2M\,n^2$ D none of previous answers are correct

11. Let A be the matrix

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}$$

    Is it possible to find the Cholesky factorization $A = H\,H^T$ of A?

    A Yes B No

12. The Gauss-Seidel iteration matrix B_{GS} for the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

    is

    A $\begin{pmatrix} 0 & 1/2 \\ 0 & -1/4 \end{pmatrix}$ B $\begin{pmatrix} 0 & -1/2 \\ 0 & 1/4 \end{pmatrix}$ C $\begin{pmatrix} 0 & 1/2 \\ 0 & 0 \end{pmatrix}$ D $\begin{pmatrix} 0 & -1/2 \\ 0 & 0 \end{pmatrix}$

13. The number of arithmetic operations required for the computation of the sum S given by

$$S = \sum_{k=1}^{n} a_k \cdot b_k$$

    is about

    A n B $2n - 1$ C n^2 D $k\,n$

14. The matrix A has the LU-factorization with $|U| = -1$ (determinant of U) and L a lower triangular matrix with all ones in the main diagonal. The determinant of the matrix A^{51} is

    A -1 B 1 C -51 D 51

15. The Jacobi method is used to solve the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 4 & 1 \\ -1 & 2 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

    starting from $\mathbf{x}_0 = (\ 0 \quad 0\ )^T$. The 2-norm of the residual $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1$ of the first iterate $\mathbf{x}_1$ is

    A 1 B 5/2 C $\pm 5/2$ D $2\sqrt{2}$

16. The iterative method $\mathbf{x}_{k+1} = B\,\mathbf{x}_k + \mathbf{f}$, $k = 0, 1, \ldots$ with

$$B = \begin{pmatrix} 1/2 & 1 \\ 0 & -1/4 \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} 3/2 \\ -1/4 \end{pmatrix}$$

is convergent for each starting point $\mathbf{x}_0$

A True B False

17. The Lagrange polynomials associated to the three nodes $x_0 = -1$, $x_1 = 0$, $x_2 = 1$ are

$$l_0(x) = \frac{1}{2}(x^2 - x) \quad l_1(x) = 1 - x^2 \quad l_2(x) = \frac{1}{2}(x^2 + x)$$

A True B False

18. Let $f(x) = x^4 - x^2 + 5$. Consider $n = 4$ equally spaced nodes x_i, $i = 0, \ldots, n$ in the interval $[-1, 1]$. Let $p(x)$ be the corresponding interpolating polynomial, i.e. $p(x_i) = f(x_i)$, $i = 0, \ldots, n$. Then, the interpolation error in $x_0 = 0.5$ defined as $E = f(x_0) - p(x_0)$ is

A 0 B 0.10 C 0.12 D impossible to compute

19. Consider the points in the following table

$x_0 = 0$	1
$x_2 = 2$	3
$x_3 = 3$	-1

The divided difference $f[\, x_0,\, x_1,\, x_2\,]$ is

A -4 B $-5/3$ C $-2/3$ D 1

20. If we want to approximate an arbitrary function in an interval $[a, b]$ using equally spaced nodes, a higher degree interpolating polynomial always gives a better approximation then a lower degree one

A True B False

21. The sum of all Lagrange polynomials depends on the values of the function f in the interpolating points

A True B False

22. The absolute value of the error in the Cavalieri-Simpson formula for the computation of $\int_a^b f(x)dx$ is

$$E = \frac{|\, b - a \,|^3}{12} \,|\, f''(\xi) \,| \qquad \text{where } \xi \in [a, b]$$

A True B False

23. The composite trapezoidal formula with two equally length intervals (i.e., $m = 2$) is used to compute

$$\int_{-1}^{1} (\, 1 - x^4 \,)\, dx$$

The approximation given by he trapezoidal rule is

A -2 B -1 C 1 D 2

24. The maximum degree of precision of the quadrature formula

$$\int_a^b \omega(x)\, f(x)dx \approx \sum_{i=0}^{n} \alpha_i f(x_i)$$

where $\omega(x) > 0$ in $[a, b]$ is

A n B $n+1$ C $2n$ D $2n+1$

25. The Cavalieri-Simpson approximation of the integral

$$\int_0^1 \sqrt{x}\, dx$$

is (Cotes numbers are $C_0^{(2)} = 1/6$, $C_1^{(2)} = 4/6$, $C_2^{(2)} = 1/6$)

A $\frac{2\sqrt{2}+1}{6}$ B $\frac{2}{3}$ C $\frac{1}{6}$ D 1

26. To plot a function with the command `plot(x,y)`, the vector `x` and `y` must have the same size

A True B False C It depends on the function

27. Which of the following MatLab instructions is correct to extract the second row from the matrix A and store it in the vector v?

A `v = A( 2, : )` B `v = A( :, 2 )` C `A( :, 2 ) = v` D `v = A( 2; : )`

28. Which is `c` at the end of the Matlab code

```
1.  a = 1:3:8;
2.  b = linspace( 0, 2, 3 );
3.  c = ( a .* b ) * [ 1; 1; 1 ];
```

A 12 B 14 C 16 D 18

29. Consider the following Matlab code

```
1.  S = 5;
2.  for k=1:4
3.     if k>=3
4.        S = S + k;
5.     else
6.        S = S - k;
7.     end
5.  end
```

At the end of the loop, the variable `S` is equal to

A -5 B 0 C 5 D 9

30. Which is the value of `n` at the end of the Matlab code

```
1.  toll = 1E-6;
2.  n = 1;
3.  while( 10^n > toll & n <= 1 )
4.     n = n - 2;
5.  end
```

A -3 B -4 C -6 D -7

SECTION B: Open questions

Justify, in the exam's booklet, carefully all you answers, which must be short and clear.

1. Consider the Newton method for the approximation of the root ξ of the equation $f(x) = 0$.

   (a) Using the geometric interpretation, find the equation for the iterates given by

   $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \cdots$$

   (b) Show a graph of a function f where iterations x_k go toward the root ξ in a monotonically fashion with $x_k < x_{k+1}$, $k = 0, 1, \ldots$.

   (c) Consider equations (i) $\ln(x) = 0$ and (ii) $(x-1)\ln(x) = 0$. Sketch, for each one, a qualitative $\log_{10}(\mid e_k \mid)$ plot.

   (d) Give an example where the Newton method does not converge.

2. Consider the linear system $A\mathbf{x} = \mathbf{b}$ where

   $$A = \begin{pmatrix} 2 & 1 \\ 2 & 4 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

   (a) Just looking to A, explain why Jacobi and Gauss-Seidel are convergent methods.

   (b) Setting $\mathbf{x}^{(k)} = [x_1^{(k)}\ x_2^{(k)}]^T$, write $x_1^{(k+1)}$ and $x_2^{(k+1)}$ for Jacobi and Gauss-Seidel methods and find, for both methods, $\mathbf{x}^{(1)}$ starting from $\mathbf{x}^{(0)} = [0\ 0]^T$.

   (c) Let B_J be the iteration matrix of the Jacobi method. Using the relation

   $$\| \mathbf{e}_k \|_\infty \leq \rho(B_J) \| \mathbf{e}_{k-1} \|_\infty$$

   estimate the number k of iterations needed to have $\| \mathbf{e}_k \|_\infty / \| \mathbf{e}_0 \|_\infty \leq 10^{-3}$.

3. Consider the set of points in the following table

x_i	-1	0	1	2
y_i	-4	0	1	4

   (a) Write the Newton expression (assume $x_0 = -1$, $x_1 = 0$, $x_2 = 1$, $x_3 = 2$)

   $$\begin{aligned} P(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \end{aligned}$$

   for the interpolating polynomial. Show that P satisfies the interpolation conditions $P(x_i) = y_i$ for the given set of points.

   (b) Find the regression line $y = a_1 x + a_0$ and draw it among with points in the same plane.

4. Consider the composite trapezoidal formula for the computation of

   $$I = \int_a^b f(x)\, dx$$

(a) Write the formula for $m = 1$ (trapezoidal formula). Give the geometrical interpretation of the formula. Let I_T be the approximation of I given by the trapezoidal formula. Sketch three graphs of **three** different functions where $I_T < I$, $I_T = I$, $I_T > I$.

(b) Starting from the equation of I_T, find the composite trapezoidal formula when the interval $[a,\ b]$ is divided into $m > 1$ intervals.

(c) Given the error in the composite trapezoidal formula using m intervals as

$$E_{CT}^{(m)} = -\frac{(b-a)^3}{12m^2} f''(\xi)$$

where $\xi \in [a, b]$ is a suitable point, show that $E_{CT}^{(2m)}/E_{CT}^{(m)} \approx 1/4$ and use this result to prove the Richardson extrapolation formula.

7.4 Exam of September 2014

SECTION A: Multiple choice questions

Read **carefully** the text of each question and write in the booklet the letter which corresponds to the best answer.

1. Consider the floating point system $\mathbb{F}(10, 2, -3, 3)$. The minimum normalized positive floating point number x_{min} and the machine precision ϵ of $\mathbb{F}$ are

   A $x_{min} = 10^{-4}$ $\epsilon = 0.05$ B $x_{min} = 10^{-4}$ $\epsilon = 0.5$

   C $x_{min} = 10^{-5}$ $\epsilon = 0.05$ D $x_{min} = 10^{-5}$ $\epsilon = 0.5$

2. Consider the computation of $f(x) = \sqrt{x^2+1} - x$ for large, positive, values of $x \in \mathbb{F}(10, 2, -4, 4)$. Which of the following expressions are stable?

   (i) $f(x) = \sqrt{1+x^2} - x$ (ii) $f(x) = \dfrac{1}{\sqrt{1+x^2}+x}$ (iii) $f(x) = \sqrt{1+x^2} - \sqrt{x^2}$

   A (i) B (ii) C (iii) D (i) and (iii)

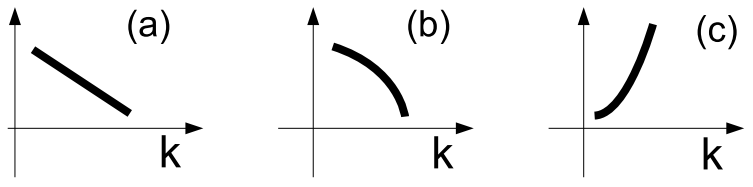
3. The bisection method is used to approximate the root $\xi = 1$ of the equation $x^2 - 1 = 0$ starting from the interval $[\ a_0,\ b_0\ ] = [\ 0.6,\ 1.6\ ]$. Denoting by x_0 and x_1 the first two iterates, the absolute value of the error $e_1 = \xi - x_1$ is

   A 0 B 0.10 C 0.15 D 0.20

4. How many fixed points has the function $f(x) = 1 - x^3$?

   A 0 B 1 C 2 D 4

5. Consider the method $x_{k+1} = \phi(x_k)$ with fixed point $\alpha = 1$. Assume that x_k goes toward α and $\phi'(\alpha) = 0$. Then, the plot of $\log_{10}(|e_k|)$, choosing from the figure below, may be



A (a) B (b) C (a) or (d) D (c) or (d)

6. The order of convergence of the Newton method for the solution of the non linear equation $e^x = 3 - 2x$ is

   A 1 B 2 C 3 D the method does not converge

7. The Newton method has the absolute value of the error $|e_5| = 10^{-6}$ for the computation of the root of $e^x + x = 0$. Assuming a unitary asymptotic error constant, we expect $|e_6|$ equals to about

   A 10^{-6} B 10^{-8} C 10^{-10} D 10^{-12}

8. The equation $f(x) = 0$ has a unique root ξ in $(0, 1)$. Assume that $f'(x) < 0$ and $f''(x) < 0$ in $[0, 1]$. Starting from $x_0 = 1$, the sequence x_k produced by the Newton method fulfills

   A $x_k < x_{k+1}$ B $x_k > x_{k+1}$ C $x_k < \xi,\ k = 1, 2, \ldots$ D none of the previous

9. Find the spectral radius of the matrix

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & -1 & 0 \\ 0 & 0 & -4 \end{pmatrix}$$

   A -1 B 2 C 4 D 8

10. Find the condition number $K_2(A)$ of

$$A = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

   A 0.01 B 0.1 C 10 D 1000

11. Find the lower triangular matrix L of the LU factorization of the following matrix A (recall: L has the main diagonal of all ones)

$$A = \begin{pmatrix} 4 & 1 \\ 3 & 3 \end{pmatrix}$$

   A $\begin{pmatrix} 1 & 0 \\ 1/3 & 1 \end{pmatrix}$ B $\begin{pmatrix} 1 & 0 \\ -1/4 & 1 \end{pmatrix}$ C $\begin{pmatrix} 1 & 0 \\ 1/4 & 1 \end{pmatrix}$ D $\begin{pmatrix} 1 & 0 \\ -1/3 & 1 \end{pmatrix}$

12. Starting from $\mathbf{x}_0 = [\ 0\ ,\ 0\ ]^T$, the norm of the residual $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1$ after the first Gauss-Seidel iteration for the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

   is

   A $\frac{\sqrt{130}}{6}$ B $\frac{7}{6}$ C 0 D $\left[-\frac{7}{6}\ ,\ 0 \right]^T$

13. The iterative method $\mathbf{x}_{k+1} = B\,\mathbf{x}_k + \mathbf{f}$, $k = 0, 1, \ldots$ with

$$B = \begin{pmatrix} 1/3 & 1 \\ 0 & -1/9 \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} 3/2 \\ -1/4 \end{pmatrix}$$

   is convergent for each starting point $\mathbf{x}_0 \in \mathbb{R}^2$

   A True B False

14. The Gauss-Seidel method converges if the matrix A is

$$B = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 9 \end{pmatrix}$$

A True B False

15. The Lagrange polynomials depends only on the nodes of the points (x_i, y_i), $i = 0, \ldots, n$

A True B False

16. The Lagrange polynomials associated to the three nodes $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ are

$$l_0(x) = \frac{1}{2}(x^2 - 3x + 2) \quad l_1(x) = 2x - x^2 \quad l_2(x) = \frac{1}{2}(x^2 - x)$$

A True B False

17. Chebyshev nodes associated to the interval $[-5, 5]$ are nearer in the middle of the interval then at the endpoints

A True B False

18. Find the divided difference $f[x_0, x_1, x_2]$ of the following table

$x_0 = 1$	2
$x_1 = 2$	3
$x_2 = 3$	6

A 1 B 3 C -1 D -3

19. Given the set of points $x_k = k$, $k = 0, 1, 2, 3$ and the function $f(x) = e^{P(x)}$ where $P(x)$ is a given polynomial of degree $n = 5$, the number ξ defined as

$$\xi = \left(f[x_0, x_1, x_2, x_3] - f[x_1, x_0, x_3, x_2] - 1 \right)^3$$

is equal to

A 1 B 0 C -1 D impossible to compute

20. Find the regression line for the set of points

x_i	-1	0	1	2
y_i	0	2	3	3

A $y = x + 1.5$ B $y = 1.5x + 1$ C $y = x + 1$ D $y = 1.5x + 1.5$

21. The absolute value of the error $E_{CS} = I - I_{CS}$ for the computation of

$$I = \int_0^2 x^4 \, dx$$

using the Cavalieri-Simpson formula (which gives the result I_{CS}) is

A $\frac{2}{15}$ B $\frac{3}{15}$ C $\frac{4}{15}$ D 0

22. The fourth derivative of f does not change much in the integration interval. Then, using the composite Cavalieri-Simpson formula, we expect that the ratio E_{2m}/E_m of the absolute value of errors using $2m$ and m intervals is

A $\frac{1}{4}$ B near $\frac{1}{4}$ C near $\frac{1}{16}$ D $\frac{1}{16}$

23. The composite trapezoidal formula with $m = 2$ intervals applied to the computation of the integral

$$I = \int_0^2 \sqrt{x}\, dx$$

gives

A $\frac{\sqrt{2}}{3}$ B 1 C $1 + \frac{1}{\sqrt{2}}$ D $\sqrt{2}$

24. Find the maximum degree of precision of the quadrature formula

$$\int_0^1 e^{-x} f(x)dx \approx \sum_{i=0}^{n} \alpha_i f(x_i)$$

A n B $2n-1$ C $2n$ D $2n+1$

25. The number of arithmetic operations needed to compute the inner product[2] between two vectors of length n is about

A n B $2n-1$ C $3n$ D none of the previous answers

26. Which is the value of `w` at the end of the Matlab code

```
1.   n = 3;
2.   v = 1:n;
3    w = zeros( n, 1 );
4.   for k = n:-1:1
5.      w(k) = v(n-k+1)^2;
6.   end
```

A $[\,1\ 4\ 9\,]^T$ B $[\,9\ 4\ 1\,]^T$ C $[\,0\ 0\ 0\,]^T$ D $[\,1\ 2\ 3\,]^T$

27. After the execution of the following Matlab code, the variable `r` is equal to

```
1.   A = diag( diag( [1 2; 3 4] ) );
2.   r = eig( A );
```

A $[1\ \ 4]^T$ B $[1\ \ 2]^T$ C 4 D 1

28. The command `clear all` makes the command Window clear but does not clear the variables in the Workspace

A True B False

29. Which is the value of `n` at the end of the Matlab code

```
1.   n = 0;
2.   while( n<=3 )
3.      n = n + 2;
4.      if n >= 4
5.         n = n-1;
6.      end
7.   end
```

[2] The inner product of vectors $\mathbf{x} = [\,x_1, \ldots, x_n\,]$ and $\mathbf{y} = [\,y_1, \ldots, y_n\,]$ is $\mathbf{x} \bullet \mathbf{y} = \sum_{k=1}^{n} x_k y_k$

A 3 B 4 C 5 D 6

30. Consider the following Matlab code

```
1.   v = linspace(2,10,5);
2.   w = length( size( v' * v ) );
```

The variable `w` is

A 2 B 3 C [2 2] D $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$

SECTION B: Open questions

Justify, in the exam's booklet, carefully all you answers, which must be short and clear.

1. The fixed point method is used to approximate the root of the equation $f(x) = 0$. The method looks as $x_{k+1} = \phi(x_k)$, $k = 0, 1, \ldots$ where x_0 is a given starting point.
   (a) Show the geometric interpretation of the method; sketch two graphs to show a convergent and a divergent fixed point iteration.
   (b) Consider $\phi(x) = -\frac{x}{2} + \frac{3}{2}$. Find the asymptotic error constant; using this value and starting at $x_0 = 2$, give an estimation of the number of iterations needed to have $|x_k - \alpha| \leq 10^{-3} \cdot |x_0 - \alpha|$.
   (c) Setting now $\phi(x) = \gamma x^2 + \delta x$, find the values of γ and δ in order to have a fixed point iterations that converge toward the fixed point $\alpha = 1$ with an order of convergence $p \geq 2$.
2. Consider the upper triangular linear system $U\mathbf{x} = \mathbf{b}$.
   (a) Show, with all the mathematical details, the backward substitution method.
   (b) Show, with all details, the computational cost of the method.
3. Give the definition of the condition number $K(A)$ of a matrix A. Prove that the condition number $K(A)$ fulfills $K(A) \geq 1$ for each matrix A. Give an example of a well conditioned matrix and one of an ill conditioned matrix.
4. The composite trapezoidal formula using m intervals for the approximation of the integral
$$I = \int_a^b f(x)\, dx$$
has an absolute value of the error given by
$$E_T = \frac{(b-a)^3}{12m^2} \mid f''(\xi) \mid$$
where ξ is a suitable point in $[a, b]$. Find the number of intervals m to have an absolute value of the error less than 10^{-6} for the computation of
$$\int_1^3 \ln(x)\, dx$$