People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Abou Bekr Belkaid University - Tlemcen

Faculty of Technology

Department of Mechanical Engineering

Lecture notes

Numerical methods for engineers: Lecture notes and exercises

Dr. MOKKEDEM Fatima Zahra

2023/2024

University of Tlemcen
Faculty of Technology
Department of Mecanical engineering
B.P. 230 - 13000 Chetouane - Tlemcen - Algeria.

fatimazahra.mokkedem@univ-tlemcen.dz

Preface

This book, entitled "Numerical methods for engineers: Lecture notes and exercises", is an introduction to the different numerical methods usually used to solve problems related to ordinary differential equations. Prior knowledge of single-variable calculus is highly required to better understand its content. The mastery of at least one programming language, like MATLAB, is recommanded to apply numerical methods on different examples easily and quickly.

Since this manuscript is addressed to engineers, it ignores some details of numerical theory like the proofs of theoretical results. However, it focuses on numerical formulas and different approximations that may be used. The content of this book goes from easy to more complicated techniques, all by explaining where they come from, and why they are used. We also emphasize how each formula works, what is the error obtained from its application and how to reduce this error according to our goals. In the end, the reader can choose the most convenient technique related to his purposes and his data.

For the best of the reader knowledge, each chapter of these notes contains many examples, some times figures also, and ends with a list of exercises with hints of them resolutions.

In the end of this book, the reader must be able to:

- Know and define the different numerical methods studied.
- Apply all the numerical methods explained (or not) in this book.
- Distinguish the differences between the numerical methods studied in the same chapter.
- List the advantages and disadvantages of each numerical method.
- Choose the number of subdivisions necessary to reduce and/or fix the error.

At the end of this course, the learner will be able to choose the most suitable numerical method in relation to his objective as well as the most practical step or number of iterations to reach the desired error.

Contents

Chapter 1

Solving nonlinear equations $f(x) = 0$

In this chapter, we learn how to find numerically the roots to equations of type $f(x) = 0$. Of course, we suppose that our goal can not be achieved analytically. Moreover, we should keep in mind that numerical methods in this chapter do not give the exact solutions but only approach it as much as needed.

1.1 Preliminary

Definition 1.1. A function f defined on its domain of definition D_f is said to be *linear* if:

$$\forall (x, y) \in D_f^2, \quad \forall \lambda \in \mathbb{R}, \quad f(\lambda x + y) = \lambda f(x) + f(y). \tag{1.1}$$

Example 1.1. The integral is a linear function.

Definition 1.2. A function which do not satisfy condition (1.1) is called *nonlinear*.

Example 1.2. The functions $f(x) = e^x + 1$, $f(x) = \sin(x) + \cos(x)$, $f(x) = x^4 + 13x - 2$,... are nonlinear.

Definition 1.3. An equation of type $f(x) = 0$ with f a nonlinear function is called a *nonlinear equation*.

Important 1. *Before solving a nonlinear equation, we should make sure that its solution (root) exists! Noting that x_r is said to be a solution (root) to equation $f(x) = 0$ if $f(x_r) = 0$. From a geometric point of view, x_r is the intersection of the graph of $f(x)$ and the absissa axe $y = 0$.*

Theorem 1.1. Let f be a continuous function on $[a, b]$ and let $f(a)f(b) < 0$. Then, there exists at least a root $x_r \in [a, b]$ to $f(x_r) = 0$.

Remark 1.1.

- If $f(a)f(b) > 0$, then $f(x) = 0$ may have no solution, one solution or more than one solution in $[a, b]$, see Figure 1.1.
- If $f(a)f(b) < 0$, then $f(x) = 0$ may have one solution or more than one solution in $[a, b]$, see Figure 1.2.

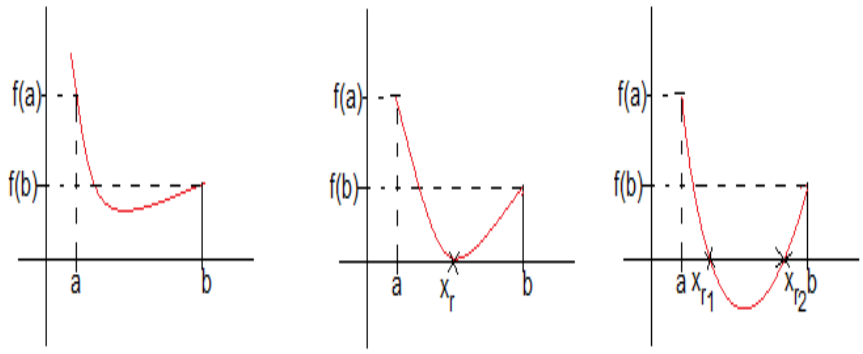


Figure 1.1: *First case* $f(a)f(b) > 0$

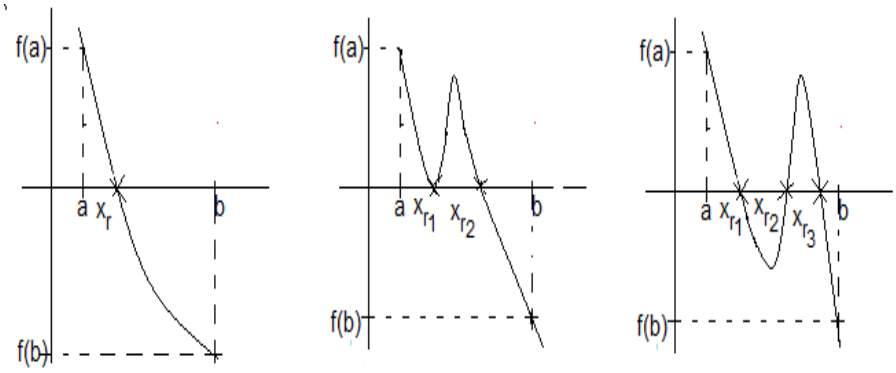


Figure 1.2: *Second case* $f(a)f(b) < 0$

To apply numerical methods, we should make sure that $f(x)$ has exactly one solution in $[a, b]$. Hence, we should take small interval $[a, b]$ (as small as possible) in which we:

Theorem 1.2. Let f be a continuous function on $[a, b]$ and let $f(a)f(b) < 0$. Suppose that f is strictly monotonic on $[a, b]$. Then, there exists a unique root $x_r \in [a, b]$ to $f(x) = 0$.

Recall 1.1. If f is derivable with $f'(x) < 0$ or $f'(x) > 0$ on $[a, b]$, then f is strictly monotonic on $[a, b]$.

Exercise 1.1. Separate (locate) the roots of $f(x) = x^4 + 4x + 2 = 0$.

Solution 1.1. Separate the roots of an equation means propose small intervals such that inside each interval of them we find exactly one solution. The most common way to do it is "the variation table".

Here we have $f(x) = x^4 + 4x + 2 = 0$. Then $f'(x) = 4x^3 + 4$. $f'(x) = 0$ implies $x = -1$. Hence

x	$-\infty$	-1	$+\infty$
$f'(x)$	$-$	0	$+$
$f(x)$	$+\infty$	$\searrow_{-1}\nearrow$	$+\infty$

From the previous table of variation, there exists $x_1 \in]-\infty, -1[$ a solution to $f(x) = 0$ and there exists $x_2 \in]-1, +\infty[$ another solution to $f(x) = 0$. This separates the two solutions of this equation. However, the proposed intervals are so long. We can reduce the lengh of them by choosing arbitrary values of x and verifying the sign of $f(x)$. For example, for $x = -2$, we have $f(-2) > 0$. This together with the fact that $f(-1) = -1 < 0$ implies that the first root $x_1 \in]-2, -1[$. Observe that $]-2, -1[$ is smaller (then better) than $]-\infty, 0[$. Similarly, we may take $x_2 \in]-1, 0[$ instead of $]-1, +\infty[$.

In the sequel, we suppose that f is continuous on $[a, b]$ and that a unique solution x_r to $f(x) = 0$ exists in $[a, b]$. We also suppose that we can not find the value of x_r analytically (by hand). Hence we **approach** it using *numerical methods*, also called *iterative methods*.

1.2 Bisection method

1.2.1 Algorithm

This method is based on the fact that $f(a)f(b) < 0$. It starts by dividing $[a, b]$ on two subintervals $[a, x_1]$ and $[x_1, b]$ with $x_1 = \frac{a+b}{2}$ is the center of $[a, b]$.

The sign of $f(a)f(x_1)$ determines in which subinterval the $x-$root exists:

– If $f(a)f(x_1) = 0$, then $f(x_1) = 0$. Hence x_1 is exactly the desired $x-$root.

– If $f(a)f(x_1) < 0$, then $x-$root belongs to $[a, x_1]$ instead of $[a, b]$.

– If $f(a)f(x_1) > 0$, then $x-$root belongs to $[x_1, b]$ instead of $[a, b]$.

Once we detremine the right subinterval containing the $x-$root, we repeat the procedure by looking for x_2, its center, and so on.

Example 1.3. Apply bisection method to approach the solution of equation $f(x) = xe^x - 1 = 0$ on $[0, 1]$ with *tolerance* of 10^{-3}.

Solution 1.2. Firstly, we see that $f(x)$ is continuous on $[0, 1]$. Also $f(0) < 0$ and $f(1) > 0$ which implies $f(0)f(1) < 0$. Moreover $f'(x) = (x + 1)e^x > 0$ on $[0, 1]$, hence f is strictly increasing on $[0, 1]$. Regarding the above three arguments, there exists a unique $x-$root to $f(x) = 0$ in $[0, 1]$.

1st iteration: $x_1 = \frac{0+1}{2} = 0.5$ and $f(x_1) = -0.17563$.

Hence $f(0)f(0.5) > 0$. Then $x-$root belongs to $[0.5, 1]$.

2nd iteration: $x_2 = \frac{0.5+1}{2} = 0.75$ and $f(x_2) = 0.58775$.

Hence $f(0.5)f(0.75) < 0$. Then $x-$root belongs to $[0.5, 0.75]$.

3rd iteration: $x_3 = \frac{0.5+0.75}{2} = 0.625$ and $f(x_3) = 0.16765$.

Hence $f(0.5)f(0.625) < 0$. Then $x-$root belongs to $[0.5, 0.625]$.

$\vdots$

13th iteration: $x_{13} = 0.567260741$ and $f(x_{13}) = 0.000324573$.

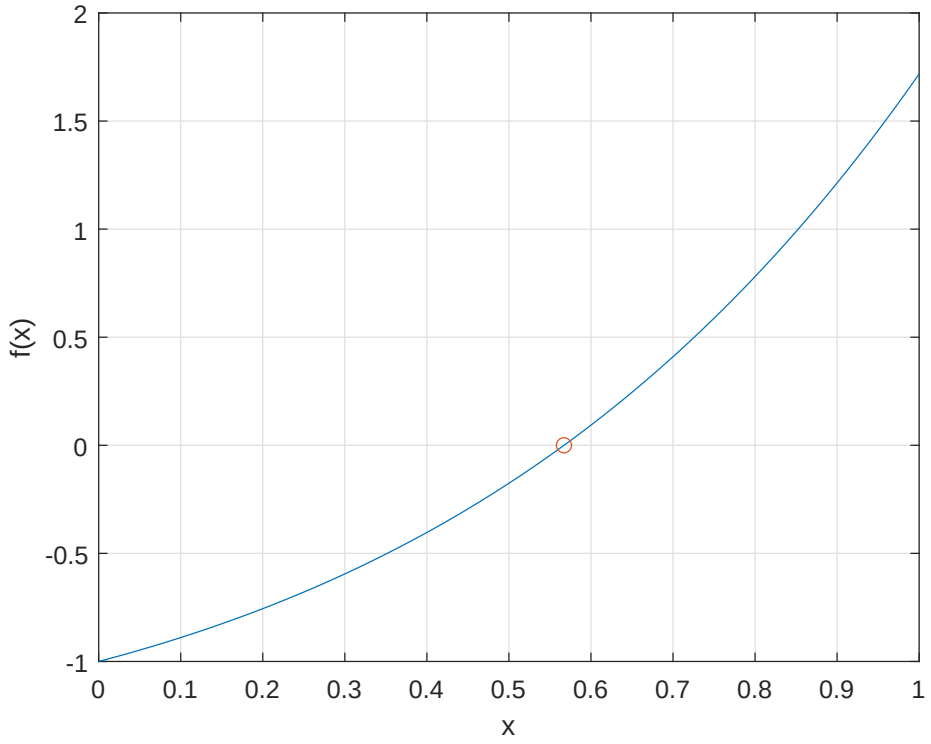


Figure 1.3: *Graph of* $f(x) = xe^x - 1$ *on* $[0, 1]$ *(Example 1.3)*

Observe that after 13^{th} iteration, there exist three zeros after the comma in $f(x_{13})$. Here we say that we have approached the solution with *three exact decimals* or that the *approximate error is less than* 10^{-3}.

If we need more exactitude, we should do more iterations !

1.2.2 Error and order of convergence

– At each iteration k, we can define the distance between the exact root x_r and the approximate value x_k by *the error at iteration* k, we note:

$$E_k = |x_r - x_k|.$$

– Since each interval is the half of its previous one, we can prove that

$$E_k \leq \frac{b-a}{2^{k+1}}.$$

– If we need a tolerance of 10^{-p}, then we should suppose that

$$E_k \leq \frac{b-a}{2^{k+1}} \leq 10^{-p}.$$

This will cost us

$$\boxed{\text{tolerance of } 10^{-p} \implies k \geq \frac{\ln(10^p(b-a))}{\ln(2)} \quad \text{iteration.}} \tag{1.2}$$

(Make sure that k is an integer value!).

– More we make iterations more the error becomes smaller. This implies that this method goes directly to the desired solution. This is called *convergence of the method*. However, bisection method is very "slow", we say that it is a *first-order method* or a *linear convergence method*.

1.2.3 Advantages and disadvantages

Bisection method is always convergent but it is a first-order one (slow).

1.3 Fixed point method

Looking for a solution to $f(x) = 0$ is exactly looking for the intersection of the graph of $f(x)$ with the abscissa axe $y = 0$. In this method we first need to rewrite $f(x) = 0$ into the form $g(x) = x$, then we turn to find the intersection between the graph of $g(x)$ and the line $y = x$. This leads to find a *fixed point* of g (a point x_r such that the value of $g(x_r)$ is fixed by the value x_r its self, $g(x_r) = x_r$). This is the origin of this method's name.

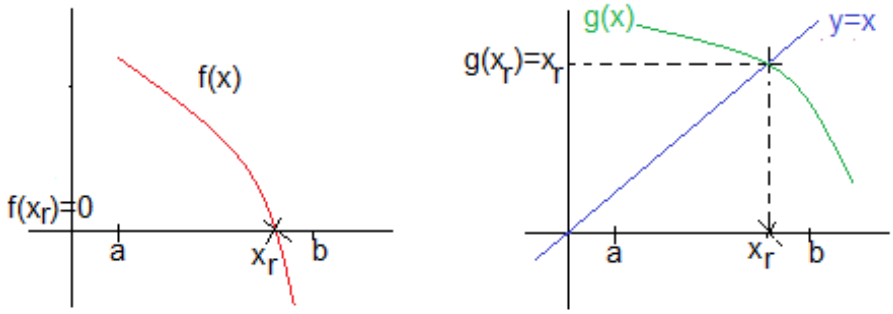


Figure 1.4: *Geometric explication of fixed point method*

Exercise 1.2. Rewrite the equation $x^2 - x - \ln(x) = 0$ into the form $g(x) = x$.

Solution 1.3. Here we have many propositions, like:

$$g_1(x) = x^2 - 3\ln(x), \quad \text{or} \quad g_2(x) = \sqrt{x + 3\ln(x)} \quad \text{or} \quad g_3(x) = e^{(x^2-x)/3}.$$

Question 1.1. Which one of the previous g functions is the "best"?

Answer 1. *The "best" function is the function which insures the convergence of the method. In other words, it is the function that satisfies the followings:*

Theorem 1.3. Let g be a continuous function on $[a, b]$ and let g' be its derivative. If

$$\forall x \in [a, b], \quad \exists L \leq 1 \quad \text{such that} \quad |g'(x)| \leq L \tag{1.3}$$

then g admets a unique fixed point x_r on $[a, b]$.

Remark 1.2. The above theorem gives a sufficient condition of convergence not a necessary one. In other words, this is not the only way to verify the convergence of fixed point method but it is the most common one.

Remark 1.3. The choice of interval $[a, b]$ is very important in equation (1.3). Observe that if $[a, b]$ is very large, then such a constant L may not exist for any g function, inversely, if $[a, b]$ is sufficiently small, then the constant L may exist. For this reason, the more we reduce the lengh of interval $[a, b]$, the more we have chance for convergence.

1.3.1 Algorithm

After rewriting $f(x) = 0$ into the form $g(x) = x$ and after making sure that condition (1.3) is verified, it only left to apply the following algorithm:

$$\boxed{\begin{cases} x_0 \quad \text{given or chosen in} \quad [a, b] \\ x_{k+1} = g(x_k), \quad k \geq 0. \end{cases}} \tag{1.4}$$

Example 1.4. Apply fixed point method to approach the solution to equation $f(x) = xe^x - 1 = 0$ on $[0, 1]$ with $x_0 = 0$ and *tolerance* of 10^{-3}.

Solution 1.4. We already checked the existence and unicity of the solution to this equation in example 1.3. Now we put

$$f(x) = 0 \implies xe^x = 1 \implies x = e^{-x} = g(x).$$

We have $|g'(x)| = |-e^{-x}| \leq 1$ on $[0, 1]$. Hence we can apply algorithm (1.4) with $x_0 = 0$.

We get

$$\begin{aligned}
x_1 &= g(x_0) = g(0) = e^{-0} = 1,\\
x_2 &= g(x_1) = g(1) = e^{-1} = 0.367879441,\\
&\vdots\\
x_{12} &= g(x_{11}) = 0.566414733,\\
x_{13} &= g(x_{12}) = 0.567556637,\\
x_{14} &= g(x_{13}) = 0.566908912.
\end{aligned}$$

Finally x_{13} is an approximate solution of $f(x) = 0$ with an approximate error less than 10^{-3}.

1.3.2 Error and order of convergence

- At each iteration k we have

$$E_k = |x_r - x_k| \leq \frac{L^k}{1-L}(b-a)$$

  with L the constant from (1.3) and $L \neq 1$.
- If we need an exactitude of p exact decimals, then we should suppose that

$$E_k \leq \frac{L^k}{1-L}(b-a) \leq 10^{-p}.$$

  This will cost us

$$\boxed{\text{tolerance of } 10^{-p} \implies k \geq \frac{\ln\left(\frac{(1-L)10^{-p}}{b-a}\right)}{\ln(k)} \qquad \text{iteration.}} \tag{1.5}$$

  Make sure that k is an integer !
- The order of convergence of this method depends on the quantity $g'(x_r)$:
  - If $g'(x_r) \neq 0$, then the convergence is of order one (*linear convergence*=slow).
  - If $g'(x_r) = 0$, then the convergence is of order two (*quadratic convergence*=quick).

  Remark 1.4. Although the above condition seems simple, it is not helpful because it depends on x_r value which is unknown before calculations.

1.3.3 Advantages and disadvantages

In general, this method is of order two (quick), however, this is not always the case. Moreover, the convergence of this method is not always guaranteed, the choice of function g as well as the

lengh of interval $[a, b]$ is of great unfluence on its convergence.

1.4 Newton-Raphson method

1.4.1 Algorithm

For this method we need to apply the following algorithm:

$$\boxed{\begin{cases} x_0 \quad \text{given or chosen in} \quad [a,b] \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}, \quad k \geq 0. \end{cases}} \tag{1.6}$$

We stop the algorithm when $x_{k+1} \cong x_k$ with the desired precision.

Example 1.5. Apply Newton-Raphson method to approach the solution to equation $f(x) = xe^x - 1 = 0$ on $[0, 1]$ with *tolerance* of 10^{-3}. Start with $x_0 = 0$.

Solution 1.5. We have $f(x) = xe^x - 1 = 0$. Then $f'(x) = (x + 1)e^x \neq 0$ on $[0, 1]$. From (1.6) we have:

$$\begin{cases} x_0 \quad \text{given or chosen in} \quad [a,b] \\ x_{k+1} = x_k - \dfrac{x_k e^{x_k} - 1}{(x_k + 1)e^{x_k}}, \quad k \geq 0. \end{cases}$$

Then

$$\begin{aligned} x_0 &= 0, \\ x_1 &= 1, \\ x_2 &= 0.68393972, \\ x_3 &= 0.577454476, \\ x_4 &= 0.567229737, \quad \longleftarrow \text{a precision of } 10^{-1} \\ x_5 &= 0.567143296, \quad \longleftarrow \text{a precision of } 10^{-3} \\ x_6 &= 0.567143290 \quad \longleftarrow \text{a precision of } 10^{-8}. \end{aligned}$$

1.4.2 Error and order of convergence

- Observing that for $g(x) = x - \frac{f(x)}{f'(x)}$, Newton-Raphson method is nothing but a particular fixed point method.
- If the second derivative of function f exists, then direct computation gives

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

Putting $x = x_r$ (the desired solution), we get $g'(x_r) = \frac{f(x_r)f''(x_r)}{(f'(x_r))^2} = 0$. Hence this method is of order two (quadratic convergent method).

– Since $f(x_r) = 0$ and since f is continuous, in a small neighberhood of x_r, the value of $f(x)$ must be very small, hence

$$|g'(x)| = \left|\frac{f(x)f''(x)}{(f'(x))^2}\right| \leq 1$$

is guaranteed if the interval $[a, b]$ is small enough or at least if x_0 is close enough to x_r.

1.4.3 Conditions of convergence

From all the previous discussion, we conclude that some conditions must be fulfilled to guarantee the application and the convergence of this method. We cite:

– The function f is continuous and two times differentiable on $[a, b]$.
– $f(a)f(b) < 0$.
– $f'(x) \neq 0$ on $[a, b]$.
– $f''(x)$ does not change its sign within $[a, b]$ or simply say $f''(x) \neq 0$ on $[a, b]$. This condition insures that f does not change its concavity inside $[a, b]$. See Section 3.4 to understand why.
– $f(x_0)f''(x_0) > 0$. This condition is verified when x_0 is close enough to x_r which is preferred

or

– $\left|\frac{f(a)}{f'(a)}\right| < b - a$ and $\left|\frac{f(b)}{f'(b)}\right| < b - a$. This condition insures that Newton-Raphson method will converge at every initial arbitrary point x_0 that belongs to $[a, b]$.

1.4.4 Geometric point of view

Let x_0 be the initial point. In the first iteration we draw the tangent line of the function f at point $(x_0, f(x_0))$. This tangent line has the equation:

$$y = f(x_0) + f'(x_0)(x - x_0).$$

The intersection of this line with the abscissa axe $y = 0$ is the point $(x_1, f(x_1))$ with:

$$0 = f(x_0) + f'(x_0)(x_1 - x_0)$$

or equivalently

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Observe that $\left|\frac{f(x_0)}{f'(x_0)}\right|$ is exactly the distance between x_0 and x_1. We need this distance to be small than $b - a$ to insure that x_1 is located inside $[a, b]$ and hence is near to the $x-$root.

In the second iteration, we draw the tangent line of f at point $(x_1, f(x_1))$. The intersection of this tangent with the abscissa axe $y = 0$ is the point $(x_2, f(x_2))$ with:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

and $\left|\frac{f(x_1)}{f'(x_1)}\right|$ is the distance between x_1 and x_2. To guarantee that this distance gets smaller and smaller, we need f to not change its concavity.

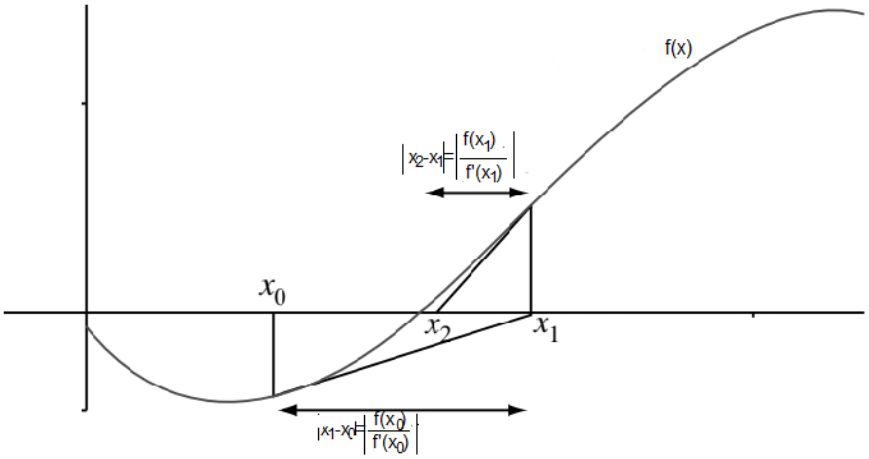


Figure 1.5: *Geometric explication of Newton-Raphson method*

1.4.5 Advantages and disadvantages

This method is not always applicable nor always convergent. These two goals need some hypothesis to be achieved. However, when this is the case, this method is always of quadratic order (quick).

1.5 Exercises

Exercise 1.3. Locate possible roots of the following equations:

$$g(x) = e^{-x} \quad \text{and} \quad g(x) = (x-2)^2 + x - \frac{e^x}{\pi}.$$

Hint: See Exercise 1.1. One can use the geometric way, see Important 1.

Exercise 1.4. Let the function $F(x) = 2x^3 - x - 2$. We wish to solve the equation $F(x) = 0$ in $[1, 2]$.

1. Prove that $F(x) = 0$ admits a unique solution x-root in $[1, 2]$.

2. If using bisection method, how many iterations are required to guarantee that the error is less than 10^{-3}?
3. Apply the first five iterations of bisection method.
4. We want to reformulate $F(x) = 0$ to $g(x) = x$. Check that the following choices of g really corresond to the equation $F(x) = 0$:
   (a) $g(x) = 2x^3 - 2$.
   (b) $g(x) = \frac{2}{2x^2-1}$.
   (c) $g(x) = (1 + \frac{x}{2})^{\frac{1}{3}}$.

   Check whether or not the iterative processes $x_{k+1} = g(x_k)$ converge.
5. For the convergent process, how many iterations we need to get an error less or equal 10^{-4}?
6. Estimate x-root with $x_0 = 1$.
7. Verify all necessary criteria (conditions) of Newton-Raphson method if one takes $x_0 = 1.5$.
8. Apply Newton-Raphson method to approach x-root.

   Hint: 1. Use Theorem 1.2. 2. Use inequality (1.2)**. 4. Use Theorem 1.3. 5. Use inequality** (1.5)**. 7. Use Section 1.4.3.**

Exercise 1.5. We want to find all the real solutions of $x^2 = \ln(1 + x)$.

1. Prove that the equation $f(x) = x^2 - \ln(1+x) = 0$ has two solutions, the first one is trivial $x_1^* = 0$ but the second one, denoted x_2^*, is still unknown.
2. Locate x_2^* in an interval of length $\frac{1}{4}$.
3. How many iterations are required, by bisection method, to guarantee that the error is less than 10^{-4}?
4. Apply the first three iterations of bisection method.
5. Propose a value of initial condition x_0 which guarantee the convergence of Newton-Raphson method.
6. Give the algorithm of Newton-Raphson method on this function and apply the first two iterations. What is the precision that you got? Conclude.
7. Let the following iterative processes:

$$x_{n+1} = \sqrt{\ln(1 + x_n)} \quad \text{and} \quad x_{n+1} = e^{x_n^2} - 1.$$

   Check whether they converge or diverge. For the convergent process, indicate the number of iterations required to approach x_2^* with a tolerance of 10^{-4}. How do you explain this number of iterations?

Exercise 1.6. We wish to calculate $\sqrt[4]{\frac{1}{3}}$ by finding the roots of an application f from $\mathbb{R}$ to $\mathbb{R}$.

1. Write this application.
2. Locate the two roots of f. In particular, prove that there exists a unique root in interval $[0, 1]$.
3. Apply Newton-Raphson method to find this root (we want a tolerance of 10^{-6}).

Hint: We want to find $x = \sqrt[4]{\frac{1}{3}}$ that is $x^4 = \frac{1}{3}$, hence we need to solve $f(x) = x^4 - \frac{1}{3} = 0$.

Exercise 1.7. Solve $10e^{x-2} + \sin(3x) - 3 = 0$ using a method of your choice on the interval $[0, 1]$. If you choose a method which might converge as well as diverge on the given interval, secure convergence of the method by verifying all necessary conditions.

Chapter 2

Solving linear systems of form AX=b

In this chapter we need to find the values of $(x_1, x_2, \cdots, x_n)$ such that the following system of equations is satisfied:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n. \end{cases} \tag{2.1}$$

Of course, a_{ij} and b_i (for $i, j = 1, \cdots, n$) are all supposed to be known.

System (2.1) can be written in the matrix form $AX = b$ by putting

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

In our lecture, we suppose that the linear system $AX = b$ has a unique solution for all vectors $b \in \mathbb{R}^n$. This is guaranteed if and only if $det(A) \neq 0$.

2.1 Direct methods for solving linear systems

A numerical method is said to be "direct" if it gives, after a finit number of iterations, the "exact" solution of system (2.1).

This kind of methods is used when $n \leq 100$ and A is a full matrix (i.e. it does not contain a lot of zeros).

2.1.1 Gauss elimination

The goal of Gauss elimination is to change the couple (A, b) with a new couple (C, d) such that C is a upper triangular matrix then solve system $CX = d$.

Definition 2.1. A square matrix C is said to be *upper triangular* if all the elements under its diagonal are equal to zero.

2.1.1.1 Naive Gaussian elimination

To transform $AX = b$ to $CX = d$ we need to apply the following algorithm:

Step 0: Construct the augmented matrix $(A|b)$.

Step 1: Define the "pivot" as a_{11} and transform all the numbers under the pivot to zero value. This is done as follows: for k from 2 to n: Multiply the first row by $a_{k1}/pivot$ and subtract the result from the k^{th} row.

Step 2: Define the "pivot" as a_{22} and transform all the numbers under the pivot to zero value. This is done as follows: for k from 3 to n: Multiply the second row by $a_{k2}/pivot$ and subtract the result from the k^{th} row.

$\vdots$

Step $n-1$: Define the "pivot" as $a_{n-1,n-1}$ and transform the number under the pivot to zero value. This is done as follows: Multiply the $n-1^{\text{th}}$ row by $a_{n,n-1}/pivot$ and subtract the result from the n^{th} row.

Example 2.1. Use Gaussian elimination to solve system

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 26 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = -19 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -34. \end{cases}$$

Solution 2.1.

Step 0: The augmented matrix is:

$$(A|b) = \left(\begin{array}{cccc|c} \boxed{6} & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \\ 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -34 \end{array}\right) \quad \text{and} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

Step 1: The pivot is $a_{11} = 6$, all the values under the pivot must be null.

$$\left(\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & \boxed{-4} & 2 & 2 & -6 \\ 0 & -12 & 8 & 1 & -27 \\ 0 & 2 & 3 & -14 & -18 \end{array}\right)$$

The second row was accomplished by multiplying the first row by $12/6$ and subtracting the result from the second row; the third row was accomplished by multiplying the first row by $3/6$ and subtracting the result from the third row and the fourth row was accomplished by multiplying the first row by $-6/6$ and subtracting the result from the fourth row.

Step 2: The pivot is $a_{22} = -4$, all the values under the pivot must be null.

$$\left(\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & \boxed{2} & -5 & -9 \\ 0 & 0 & 4 & -13 & -21 \end{array}\right)$$

the third row was accomplished by multiplying the second row by $-12/-4$ and subtracting the result from the third row and the fourth row was accomplished by multiplying the second row by $2/-4$ and subtracting the result from the fourth row.

Step 3: The pivot is $a_{33} = 2$, the value under the pivot must be null.

$$\left(\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 0 & -3 & -3 \end{array}\right)$$

The fourth row was accomplished by multiplying the third row by $4/2$ and subtracting the result from the fourth row.

Step 4: Now, we solve the new system of equations:

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 16 \\ -4x_2 + 2x_3 + 2x_4 = -6 \\ 2x_3 - 5x_4 = -9 \\ -3x_4 = -3. \end{cases}$$

Clearly, the last equation is the easiest, it gives $x_4 = 1$. Then the third one implies $x_3 = -2$ and so on. We call this the ***"back substitution"***. Finally, the solution vector

$X = (x_1, x_2, x_3, x_4)^T = (3, 1, -2, 1)^T$.

Remark 2.1.

- If at a step k, the pivot $a_{kk} = 0$, it becomes impossible to divide on zero ! Hence we need to change the row of the pivot with any other row below it in which the pivot is not null.
- If at a step k, the pivot a_{kk} is very small and very close to zero, then the algorithm may not converge to the exact solution. It is better to change the row of the pivot with any other row below it in which the pivot is far from zero value.

2.1.1.2 Partial pivoting Gaussian elimination

The goal of this method is to make sure that the pivot is neither null nor close to zero. Recall that at step k, the pivot is supposed to be the value in a_{kk} position. So we first choose the element that has the biggest absolute value among a_{kk} and all values under it. We denote it a_{pk} with $p \geq k$. Then we interchange row k with row p. Finally we proceed with the elimination as shown previously.

Example 2.2. Use partial pivoting Gaussian elimination to solve system

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 10 \\ -x_2 + 2x_1 + 3x_3 - 4x_4 = -7 \\ 11x_4 - 2x_3 + 4x_2 + 8x_1 = 54 \\ x_3 - 4x_4 + 9x_1 - x_2 = -6. \end{cases}$$

Solution 2.2.

Step 0: The augmented matrix is:

$$(A|b) = \left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 10 \\ 2 & -1 & 3 & -4 & -7 \\ 8 & 4 & -2 & 11 & 54 \\ 9 & -1 & 1 & -4 & -6 \end{array}\right) \quad \text{and} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

Step 1: The pivot must be the number of biggest absolute value in colomn 1, so a_{11} must be 9. To this end, we interchange row 1 with row 4 including b results:

$$\left(\begin{array}{cccc|c} \boxed{9} & -1 & 1 & -4 & \boxed{\boxed{-6}} \\ 2 & -1 & 3 & -4 & -7 \\ 8 & 4 & -2 & 11 & 54 \\ \boxed{1} & 1 & 1 & 1 & \boxed{10} \end{array}\right)$$

After this we proceed with the elimination that gives:

$$\left(\begin{array}{cccc|c} 9 & -1 & 1 & -4 & -6 \\ 0 & -0.77778 & 2.7778 & -3.1111 & -5.6667 \\ 0 & 4.8889 & -2.8889 & 14.556 & 59.333 \\ 0 & 1.1111 & 0.88889 & 1.4444 & 10.667 \end{array}\right)$$

Step 2: The pivot must be the number of biggest absolute value in colomn 2 except a_{12}, so a_{22} must be 4.8889. To this end, we interchange row 2 with row 3:

$$\left(\begin{array}{cccc|c} 9 & -1 & 1 & -4 & -6 \\ 0 & \boxed{4.8889} & -2.8889 & 14.556 & \boxed{\boxed{59.333}} \\ 0 & -0.77778 & 2.7778 & -3.1111 & -5.6667 \\ 0 & 1.1111 & 0.88889 & 1.4444 & 10.667 \end{array}\right)$$

After this we proceed with the elimination that gives:

$$\left(\begin{array}{cccc|c} 9 & -1 & 1 & -4 & -6 \\ 0 & 4.8889 & -2.8889 & 14.556 & 59.333 \\ 0 & 0 & \boxed{2.3182} & -0.79545 & 3.7727 \\ 0 & 0 & 1.5455 & -1.8636 & -2.8182 \end{array}\right)$$

Step 3: The pivot must be the number of biggest absolute value in colomn 3 except a_{13} and a_{23}, so a_{33} must be 2.3182. To this end, we do not need to interchange row 3 with row 4. We proceed directly with the elimination that gives:

$$\left(\begin{array}{cccc|c} 9 & -1 & 1 & -4 & -6 \\ 0 & 4.8889 & -2.8889 & 14.556 & 59.333 \\ 0 & 0 & 2.3182 & -0.79545 & 3.7727 \\ 0 & 0 & 0 & -1.3333 & -5.3333 \end{array}\right)$$

Step 4: Finally we solve the new system of equations by **"back substitution"** to get $X = (x_1, x_2, x_3, x_4)^T = (1, 2, 3, 4)^T$.

2.1.1.3 Total pivoting Gaussian elimination

Recall that at step k, the pivot is supposed to be the value in a_{kk} position. In this method, we choose the pivot to be the number of biggest absolute value among all elements of k colomn and also colomns after it. To achieve this goal we need to interchange rows and also colomns of matrix A. This operation is more complicated than partial pivoting Gaussian elimination because interchanging **colomns** requires interchanging components of X **vector** too !

Example 2.3. Use total pivoting Gaussian elimination to solve system

$$\begin{cases} 9x_1 - x_3 + 2x_4 - 8x_2 = 13 \\ x_3 + 7x_2 + 2x_4 - 13x_1 = -24 \\ 5x_1 + 8x_2 + x_3 + 5x_4 = -83 \\ 7x_3 - x_4 + 11x_1 - 17x_2 = 48. \end{cases}$$

Solution 2.3.

Step 0: The augmented matrix is:

$$(A|b) = \left(\begin{array}{cccc|c} 9 & -8 & -1 & 2 & 13 \\ -13 & 7 & 1 & 2 & -24 \\ 5 & 8 & 1 & 5 & -83 \\ 11 & -17 & 7 & -1 & 48 \end{array}\right) \quad \text{and} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

Step 1: The pivot must be the number of biggest absolute value in matrix A, so a_{11} must be -17. To this end, we interchange row 1 with row 4 and colomn 1 with colomn 2:

$$(A|b) = \left(\begin{array}{cccc|c} \boxed{-17} & 11 & 7 & -1 & \boxed{\boxed{48}} \\ 7 & -13 & 1 & 2 & -24 \\ 8 & 5 & 1 & 5 & -83 \\ -8 & 9 & -1 & 2 & 13 \end{array}\right) \quad \text{and} \quad X = \begin{pmatrix} \boxed{\boxed{x_2}} \\ \boxed{\boxed{x_1}} \\ x_3 \\ x_4 \end{pmatrix}.$$

Observe that the unknown vector starts now by x_2 instead of x_1 ! This is due to interchanging colomns 1 and 2. After this we proceed with the elimination that gives:

$$\left(\begin{array}{cccc|c} -17 & 11 & 7 & -1 & 48 \\ 0 & -8.4706 & 3.8824 & 1.5882 & -4.2353 \\ 0 & 10.176 & 4.2941 & 4.5294 & -60.412 \\ 0 & 3.8235 & -4.2941 & 2.4706 & -9.5882 \end{array}\right)$$

Step 2: The pivot must be the number of biggest absolute value in matrix A except row 1, so a_{22} must be 10.176. To this end, we interchange rows 2 and 3:

$$\left(\begin{array}{cccc|c} -17 & 11 & 7 & -1 & 48 \\ 0 & \boxed{10.176} & 4.2941 & 4.5294 & \boxed{\boxed{-60.412}} \\ 0 & -8.4706 & 3.8824 & 1.5882 & \boxed{\boxed{-4.2353}} \\ 0 & 3.8235 & -4.2941 & 2.4706 & -9.5882 \end{array}\right)$$

Observe that interchaging rows do not affect the unknown vector which is still $X = (x_2, x_1, x_3, x_4)^T$. However it affects vector b ! After this we proceed with the elimination:

$$\left(\begin{array}{cccc|c} -17 & 11 & 7 & -1 & 48 \\ 0 & 10.176 & 4.2941 & 4.5294 & -60.412 \\ 0 & 0 & \boxed{7.4569} & 5.3585 & -54.523 \\ 0 & 0 & -5.9076 & 0.76874 & 13.111 \end{array}\right)$$

Step 3: The pivot must be the number of biggest absolute value in last two rows and colomns of matrix A, so a_{33} must be 7.4569. To this end, we do not need to make any changes. We proceed with the elimination that gives:

$$\left(\begin{array}{cccc|c} -17 & 11 & 7 & -1 & 48 \\ 0 & 10.176 & 4.2941 & 4.5294 & -60.412 \\ 0 & 0 & 7.4569 & 5.3585 & -54.523 \\ 0 & 0 & 0 & 5.0139 & -30.084 \end{array}\right)$$

Step 4: Finally we solve the new system of equations by **"back substitution"** to get $(x_2, x_1, x_3, x_4) = (-2, -5, -3, -6)$ which is equivalent to $X = (x_1, x_2, x_3, x_4)^T = (-5, -2, -3, -6)^T$.

2.1.2 LU decomposition

The goal of *LU decomposition* (or said *LU factorisation*) is to change the matrix A with a product of two matrices L and U such that L is lower triangular and U is upper triangular, see Definition 2.1 and

Definition 2.2. A square matrix L is said to be *lower triangular* if all the elements up its diagonal are equal to zero.

In this manner, system $AX = b$ becomes $LUX = b$ with L and U particular matrices. This system can be solved in two easy steps: First we solve $LY = b$ then we deduce X from $UX = Y$.

Remark 2.2. Equation $A = LU$ has an infinity of solutions. Here we focus on two of them:

Crout method: in which we put $U_{ii} = 1$ for all $i = 1, \cdots, n$.

Doolittle method: in which we put $L_{ii} = 1$ for all $i = 1, \cdots, n$.

The rest of components of L and U is obtained by computing alternately one row of U and one colomn of L using the following rule:

$$\boxed{U_{ij} = \frac{A_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj}}{L_{ii}} \quad \text{and} \quad L_{ji} = \frac{A_{ji} - \sum_{k=1}^{i-1} L_{jk}U_{ki}}{U_{ii}}.} \tag{2.2}$$

Example 2.4. Use LU factorisation in Crout sens to solve system:

$$\begin{cases} x_1 - x_2 + 2x_3 + x_4 = 1 \\ 3x_1 + 2x_2 + x_3 + 4x_4 = 1 \\ 5x_1 + 8x_2 + 6x_3 + 3x_4 = 1 \\ 4x_1 + 2x_2 + 5x_3 + 3x_4 = -1. \end{cases}$$

Solution 2.4. The matrix A of the above system is:

$$A = \begin{pmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 4 \\ 5 & 8 & 6 & 3 \\ 4 & 2 & 5 & 3 \end{pmatrix}.$$

Since we are using CROUT method, the purpose is to find the matrices:

$$L = \begin{pmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} \boxed{1} & U_{12} & U_{13} & U_{14} \\ 0 & \boxed{1} & U_{23} & U_{24} \\ 0 & 0 & \boxed{1} & U_{34} \\ 0 & 0 & 0 & \boxed{1} \end{pmatrix}.$$

We start our calculus from the first colomn of L. According to equation (2.2):

$$L_{11} = \tfrac{A_{11}}{U_{11}} = 1, \qquad L_{21} = \frac{A_{21}}{U_{11}} = 3,$$
$$L_{31} = \tfrac{A_{31}}{U_{11}} = 5, \qquad L_{41} = \frac{A_{41}}{U_{11}} = 4.$$

Then, we calculate the elements in first row of U:

$$U_{12} = \frac{A_{12}}{L_{11}} = -1, \qquad U_{13} = \frac{A_{13}}{L_{11}} = 2, \qquad U_{14} = \frac{A_{14}}{L_{11}} = 1.$$

Next, we turn to second colomn of L:

$$L_{22} = \tfrac{A_{22} - L_{21}U_{12}}{U_{22}} = 5, \qquad L_{23} = \tfrac{A_{23} - L_{21}U_{13}}{U_{33}} = 13,$$
$$L_{24} = \tfrac{A_{24} - L_{21}U_{14}}{U_{44}} = 6 \quad .$$

After that, we compute the second row of U:

$$U_{23} = \frac{A_{23} - L_{21}U_{13}}{L_{22}} = -1, \qquad U_{24} = \frac{A_{24} - L_{21}U_{14}}{L_{22}} = 0.2.$$

Now, we turn to the third colomn of L:

$$L_{33} = \frac{A_{33} - L_{31}U_{13} - L_{32}U_{23}}{U_{33}} = 9, \qquad L_{43} = \frac{A_{43} - L_{41}U_{13} - L_{42}U_{23}}{U_{33}} = 3.$$

It remains to compute

$$U_{34} = \frac{A_{34} - L_{31}U_{14} - L_{32}U_{24}}{U_{33}} = -0.5111$$

and

$$L_{44} = \frac{A_{44} - L_{41}U_{14} - L_{42}U_{24} - L_{43}U_{34}}{U_{33}} = -0.6667.$$

Summarizing, we have

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 5 & 0 & 0 \\ 5 & 13 & 9 & 0 \\ 4 & 6 & 3 & -0.6667 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & -1 & 2 & 1 \\ 0 & 1 & -1 & 0.2 \\ 0 & 0 & 1 & -0.5111 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Put $LY = b$, i.e.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 5 & 0 & 0 \\ 5 & 13 & 9 & 0 \\ 4 & 6 & 3 & -0.6667 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

gives $(y_1, y_2, y_3, y_4) = (1, -0.4, 0.1333, 4.5)$. Finally, $UX = Y$, i.e.

$$\begin{pmatrix} 1 & -1 & 2 & 1 \\ 0 & 1 & -1 & 0.2 \\ 0 & 0 & 1 & -0.5111 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ -0.4 \\ 0.1333 \\ 4.5 \end{pmatrix}$$

implies $X = (x_1, x_2, x_3, x_4)^T = (-7.2333, 1.1333, 2.4333, 4.5)^T$.

Example 2.5. Use LU factorisation in DOOLITTLE sens to solve the same system in Example 2.4.

Solution 2.5. Since we are using Doolittle method now, the purpose is to find the matrices:

$$L = \begin{pmatrix} \boxed{1} & 0 & 0 & 0 \\ L_{21} & \boxed{1} & 0 & 0 \\ L_{31} & L_{32} & \boxed{1} & 0 \\ L_{41} & L_{42} & L_{43} & \boxed{1} \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ 0 & U_{22} & U_{23} & U_{24} \\ 0 & 0 & U_{33} & U_{34} \\ 0 & 0 & 0 & U_{44} \end{pmatrix}.$$

We start our calculus from the first row of U and we alternate between colomns of L and rows of U exactly like we did in example 2.4. Calculus shall give:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 5 & 2.6 & 1 & 0 \\ 4 & 1.2 & 0.3333 & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & -1 & 2 & 1 \\ 0 & 5 & -5 & 1 \\ 0 & 0 & 9 & -4.6 \\ 0 & 0 & 0 & -0.6667 \end{pmatrix}.$$

Put $LY = b$ gives $(y_1, y_2, y_3, y_4) = (1, -2, 1.2, -3)$. Finally, $UX = Y$ implies the same result $X = (x_1, x_2, x_3, x_4)^T = (-7.2333, 1.1333, 2.4333, 4.5)^T$.

Remark 2.3. Contrary to Gauss elimination, LU factorisation do not need to know the values of b vector in the begining.

Important 2. *If system $AX = b$ is well posed ($det(A) \neq 0$ hence matrix A is invertible), then system $AX = b$ admits a **unique** solution. Hence, whatever the (direct) method we use, the result of $X-$vector must be the **same**.*

2.2 Iterative methods for solving linear systems

A numerical method is said to be "iterative" if it "converges" to the exact solution of system (2.1).

This kind of methods is used when $n \geq 100$ or A is not a full matrix (i.e. it contains a lot of zeros).

Let $AX = b$ to be solved with

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Let's decompose A into three matrices D, E and S of the form:

$$D=\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & 0 & \cdots & 0 & 0 \\ 0 & 0 & a_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix},\ E=-\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & 0 & 0 & \cdots & 0 & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 & 0 \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn-1} & 0 \end{pmatrix}$$

and

$$S=-\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n-1} & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n-1} & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n-1} & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1n} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

2.2.1 Jacobi's method 1830

From the above decomposition of matrix A, if D is invertible, then:

$$\begin{aligned} AX=b &\Rightarrow (D-E-S)X=b \\ &\Rightarrow DX=(E+S)X+b \\ &\Rightarrow X=D^{-1}(E+S)X+D^{-1}b. \end{aligned}$$

2.2.1.1 Algorithm

The algorithm of Jacobi's method is the following:

$$\begin{cases} X_0 \quad \text{given}, \\ X_{k+1}=D^{-1}(E+S)X_n+D^{-1}b, \quad k\geq 0. \end{cases}$$

According to the special forms of $D,\ E$ and S, this algorithm is equivalent to

$$\boxed{\begin{cases} X_0 \quad \text{given}, \\ (x_i)_{k+1}=\dfrac{1}{A_{ii}}\left[b_i-\displaystyle\sum_{j<i}A_{ij}(x_j)_k-\sum_{j>i}A_{ij}(x_j)_k\right], \quad i,j=1,\cdots,n, \quad k\geq 0 \end{cases}}$$

where $X_k=(x_1,x_2,\cdots,x_n)_k^T=(x_{1_k},x_{2_k},\cdots,x_{n_k})^T$.

Remark 2.4. A square matrix is invertible if and only if its determinant is not null. In particular D is invertible if and only if all a_{ii} are not null ($i = 1, \cdots, n$).

2.2.1.2 Convergence

Theorem 2.1. The algorithm of Jacobi is convergent for any given X_0 to the solution of $AX = b$ if and only if: all eigenvalues of matrix $E + S$ have a modulus less or equal one. Or if matrix A is dominant, i.e.:

$$\text{for any } i = 1, \cdots, n: \quad |a_{ii}| > \sum_{j=1,\ j\neq i}^{n} |a_{ij}|.$$
$$\text{Or: for any } j = 1, \cdots, n: \quad |a_{jj}| > \sum_{i=1,\ i\neq j}^{n} |a_{ij}|.$$

Example 2.6. Solve the following system using Jacobi algorithm and $X_0 = (0, 0, 0, 0)$:

$$\begin{cases} 16x_1 + 6x_2 + 2x_3 + 5x_4 = 19 \\ 4x_1 + x_2 + 18x_3 + 2x_4 = 12 \\ x_1 + 2x_2 + 2x_3 + 14x_4 = 1 \\ 3x_1 + 10x_2 + 5x_3 + x_4 = 1. \end{cases}$$

Solution 2.6. The above system is equivalent to $AX = b$ with

$$A = \begin{pmatrix} 16 & 6 & 2 & 5 \\ 4 & 1 & 18 & 2 \\ 1 & 2 & 2 & 14 \\ 3 & 10 & 5 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 19 \\ 12 \\ 1 \\ 1 \end{pmatrix}.$$

Observe that the conditions in Theorem 2.1 are not satisfied. However, if we permute rows 2, 3 and 4, this problem will be solved:

$$A = \begin{pmatrix} \boxed{16} & 6 & 2 & 5 \\ 3 & \boxed{10} & 5 & 1 \\ 4 & 1 & \boxed{18} & 2 \\ 1 & 2 & 2 & \boxed{14} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 19 \\ \boxed{\boxed{1}} \\ \boxed{12} \\ \boxed{\boxed{1}} \end{pmatrix}.$$

According to Jacobi's algorithm, we have:

$$\begin{aligned}
(x_1)_{k+1} &= \frac{1}{A_{11}}\left[b_1 - \sum_{j>1} A_{1j}(x_j)_k\right] = \frac{1}{16}\left[19 - 6(x_2)_k - 2(x_3)_k - 5(x_4)_k\right] \\
(x_2)_{k+1} &= \frac{1}{A_{22}}\left[b_2 - A_{21}(x_1)_k - \sum_{j>2} A_{2j}(x_j)_k\right] = \frac{1}{10}\left[1 - 3(x_1)_k - 5(x_3)_k - (x_4)_k\right] \\
(x_3)_{k+1} &= \frac{1}{A_{33}}\left[b_3 - \sum_{j<3} A_{3j}(x_j)_k - A_{34}(x_4)_k\right] = \frac{1}{18}\left[12 - 4(x_1)_k - (x_2)_k - 2(x_4)_k\right] \\
(x_4)_{k+1} &= \frac{1}{A_{44}}\left[b_4 - \sum_{j<4} A_{4j}(x_j)_k\right] = \frac{1}{14}\left[1 - (x_1)_k - 2(x_2)_k - 2(x_3)_k\right].
\end{aligned}$$

For $k = 0$, we have $X_0 = (0, 0, 0, 0)^T$. Then $X_1 = \left(\frac{19}{16}, \frac{1}{10}, \frac{2}{3}, \frac{1}{14}\right)^T$. Next $X_2 = (1.0443, -0.5967, 0.3893, -0.1229)^T$. After many iterations (29 iteration) we get

$$X_{28} \approx X_{29} = (1.3269, -0.4974, 0.4005, -0.0095)^T.$$

Note that this is not the exact solution of the given system, it is just an approximation of it! For example, by replacing this result in the first equation, we get

$$16(1.3269) + 6(-0.4974) + 2(0.4005) + 5(-0.0095) = 18.9995 \approx 19.$$

2.2.2 Gauss-Seidel's method 1846

From the decomposition of matrix A, if $D - E$ is invertible, then:

$$\begin{aligned}
AX = b &\Rightarrow (D - E - S)X = b \\
&\Rightarrow (D - E)X = SX + b \\
&\Rightarrow X = (D - E)^{-1}SX + (D - E)^{-1}b.
\end{aligned}$$

The algorithm of Gauss-Seidel method is the following:

$$\begin{cases} X_0 \quad \text{given}, \\ X_{k+1} = (D - E)^{-1}SX_n + (D - E)^{-1}b, \quad k \geq 0. \end{cases}$$

According to the special forms of D, E and S, this algorithm is equivalent to

$$\boxed{\begin{cases} X_0 \quad \text{given}, \\ (x_i)_{k+1} = \dfrac{1}{A_{ii}}\left[b_i - \displaystyle\sum_{j<i} A_{ij}(x_j)_{k+1} - \sum_{j>i} A_{ij}(x_j)_k\right], \quad i, j = 1, \cdots, n, \quad k \geq 0 \end{cases}}$$

where $X_k = (x_1, x_2, \cdots, x_n)_k^T = (x_{1_k}, x_{2_k}, \cdots, x_{n_k})^T$.

Remark 2.5. $D - E$ is invertible if and only if no a_{ii} is null for all $i = 1, \cdots, n$ (same condition in Remark 2.4).

The only difference between this algorithm and the one of Jacobi is in the term: $\sum_{j<i} A_{ij}(x_j)_{k+1}$. Although this algorithm seems to be more difficult than the one of Jacobi, it is more useful in programmation and quicker in convergence. Note that Theorem 2.1 is still valid here.

Example 2.7. Solve the system in Example 2.6 using Gauss-Seidel algorithm and $X_0 = 0_4$. Recall:

$$\begin{cases} 16x_1 + 6x_2 + 2x_3 + 5x_4 = 19 \\ 4x_1 + x_2 + 18x_3 + 2x_4 = 12 \\ x_1 + 2x_2 + 2x_3 + 14x_4 = 1 \\ 3x_1 + 10x_2 + 5x_3 + x_4 = 1. \end{cases}$$

Solution 2.7. The above system is equivalent to $AX = b$ with

$$A = \begin{pmatrix} 16 & 6 & 2 & 5 \\ 3 & 10 & 5 & 1 \\ 4 & 1 & 18 & 2 \\ 1 & 2 & 2 & 14 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 19 \\ 1 \\ 12 \\ 1 \end{pmatrix}.$$

According to Gauss-Siedel algorithm, we have:

$$\begin{aligned}
(x_1)_{k+1} &= \frac{1}{A_{11}}\left[b_1 - \sum_{j>1} A_{1j}(x_j)_k\right] = \frac{1}{16}\left[19 - 6(x_2)_k - 2(x_3)_k - 5(x_4)_k\right] \\
(x_2)_{k+1} &= \frac{1}{A_{22}}\left[b_2 - A_{21}(x_1)_{k+1} - \sum_{j>2} A_{2j}(x_j)_k\right] = \frac{1}{10}\left[1 - 3(x_1)_{k+1} - 5(x_3)_k - (x_4)_k\right] \\
(x_3)_{k+1} &= \frac{1}{A_{33}}\left[b_3 - \sum_{j<3} A_{3j}(x_j)_{k+1} - A_{34}(x_4)_k\right] = \frac{1}{18}\left[12 - 4(x_1)_{k+1} - (x_2)_{k+1} - 2(x_4)_k\right] \\
(x_4)_{k+1} &= \frac{1}{A_{44}}\left[b_4 - \sum_{j<4} A_{4j}(x_j)_{k+1}\right] = \frac{1}{14}\left[1 - (x_1)_{k+1} - 2(x_2)_{k+1} - 2(x_3)_{k+1}\right].
\end{aligned}$$

For $k = 0$, we have $X_0 = (0, 0, 0, 0)^T$. Then $X_1 = (1.1875, -0.2563, 0.4170, -0.0364)^T$. Next $X_2 = (1.2428, -0.47777, 0.4211, -0.0093)^T$. After 10 iteration we get

$$X_9 \approx X_{10} = (1.3269, -0.4974, 0.4005, -0.0095)^T.$$

Observe that the same approximate solution was found after 29 iterations of Jacobi algorithm but only after 10 iterations of Gauss-Siedel algorithm!

2.3 Exercises

Exercise 2.1. We wish to solve the following linear systems:

$$1.\begin{cases} x_1 - x_2 + 2x_3 + x_4 = 1 \\ 3x_1 + 2x_2 + x_3 + 4x_4 = 1 \\ 5x_1 + 8x_2 + 6x_3 + 3x_4 = 1 \\ 4x_1 + 2x_2 + 5x_3 + 3x_4 = -1. \end{cases} \qquad 2.\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 26 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = -19 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -34. \end{cases}$$

$$3.\begin{cases} x_1 + 2x_2 + x_3 - x_4 = 5 \\ 3x_1 + 6x_2 + 4x_3 + 4x_4 = 16 \\ 4x_1 + 4x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + x_3 + 5x_4 = 15. \end{cases}$$

1. Solve the above systems by naive Gaussian ellimination.
2. Solve the above systems by total pivoting Gaussian ellimination.
3. Solve the above systems by LU factorisation in CROUT sens.
4. Solve the above systems by LU factorisation in DOOLITTLE sens.

Hint:

$$1. \Longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -7.2333 \\ 1.1333 \\ 2.4333 \\ 4.5 \end{pmatrix} \qquad 2. \Longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ -2 \\ 1 \end{pmatrix} \qquad 3. \Longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ -12 \\ 22 \\ -3 \end{pmatrix}.$$

Exercise 2.2. Let the linear system of dimension 3: $Ax = b$ where α and β are two real numbers:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & \alpha & 1 \\ 3 & 3 & 6 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} \beta \\ 2 \\ 0 \end{pmatrix}.$$

1. Give the decomposition LU of A.
2. Under which condition on α this decomposition exists?
3. For each value of α and β, this system admits a unique solution? Find it.

Exercise 2.3. Consider the following matrix A and vector b:

$$A = \begin{pmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 23 \\ 16 \end{pmatrix}.$$

1. Solve the system $Ax = b$ by the Gaussian method.
2. Give the decomposition LU of matrix A then solve the system $Ax = b$ by LU factorisation.
3. Prove that matrix A is symetric positive definite. Solve the system $Ax = b$ by the method of Choleski. (additional question)
4. Check the convergence of iterative methods.
5. Solve the system $Ax = b$ using the iterations of the Jacobi method, start with $x_0 = (1, 0, 0)$.
6. Solve the system $Ax = b$ using the iterations of the Gauss-Seidel method, start with $x_0 = (1, 0, 0)$.

Hint: The solution is $(x_1, x_2, x_3)^T = (1, 5, 2)^T$**.**

Exercise 2.4. Take the following linear system:

$$\begin{cases} 2x_1 + x_2 + x_3 = 8 \\ x_1 + 2x_2 + x_3 = 9 \\ x_1 + x_2 + 4x_3 = 19. \end{cases}$$

1. Prove that matrix A is symetric positive definite. Solve the system $Ax = b$ by the method of Choleski. (additional question)
2. Check the convergence of iterative methods then find the solution by Gauss-Seidel method by starting from the null vector.

Hint: The solution is $(x_1, x_2, x_3)^T = (1, 2, 4)^T$**.**

Exercise 2.5. Take the following linear system:

$$\begin{cases} 5x_1 - x_2 - x_3 - x_4 = -4 \\ -x_1 + 10x_2 - x_3 - x_4 = 12 \\ -x_1 - x_2 + 5x_3 - x_4 = 8 \\ -x_1 - x_2 - x_3 + 10x_4 = 34. \end{cases}$$

1. Check the convergence of iterative methods then find the solution by Jacobi method and by Gauss-Seidel method by starting from the null vector.

Hint: The solution is $(x_1, x_2, x_3, x_4)^T = (1, 2, 3, 4)^T$**.**

Chapter 3

Polynomial interpolation

3.1 Introduction

Let f be a function of independent variable x and suppose that the explicite expression of $f(x)$ is unknown. Instead, a data set of values $(x_i, f(x_i))$ is given:

x_0	x_1	x_2	$\cdots$	x_n
$f(x_0)$	$f(x_1)$	$f(x_2)$	$\cdots$	$f(x_n)$

Problem 1.

- *Suppose that we need to know the value of* $f(\tilde{x})$ *with* $\tilde{x} \in [x_0, x_n]$ *but* $\tilde{x} \neq x_i$*, for any* $i = 0, \cdots, n$.
- *Suppose that we need to integrate or derive* f *inside* $[x_0, x_n]$.
- *Suppose that we need to know the maximum or minumum value of* f *inside* $[x_0, x_n]$.

All the above operations (and others) can not be directly done since the explicite expression of $f(x)$ is unknown.

In this Chapter, we **approximate** f by the **polynomial** that respects the following ***interpolation conditions***:

1. For $n+1$ giving nodes, the polynomial is of degree less or equal n:

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

Remark 3.1. This gives $n+1$ unknown coefficient $(a_0, \cdots, a_n)$ which matches very well the $n+1$ known data points $(x_0, \cdots, x_n)$.

2. The polynomial passes exactly through the points of the data set:

$$P_n(x_i) = f(x_i), \quad \forall i = 0, \cdots, n.$$

Remark 3.2. Of course, we do not want to lose any of the information we have. Mathematically, under this condition, the distance between the polynomial $P_n(x)$ and the function $f(x)$ at the data points x_i is **null**.

3. The distance between the polynomial $P_n(x)$ and the function $f(x)$ between the data points x_i is **bounded**.

From the interpolation conditions 1. and 2. we have a system of linear equations of type:

$$\begin{cases} P_n(x_0) = a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = f(x_0), \\ P_n(x_1) = a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n = f(x_1), \\ \cdots \\ P_n(x_n) = a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = f(x_n). \end{cases}$$

Although the above system is formed by $n+1$ equation with $n+1$ unknown coefficient, once n exceeds 3, it becomes hard to be solved by hand.

Example 3.1. Use the interpolation conditions to find the interpolating polynomial of the following data:

x_i	0	1	3	5	6
$f(x_i)$	1	2	2	-1	-2

Solution 3.1. We have 5 points so we look for a polynomial of degree less or equal 4:

$$P_4(x) = a_0 + a_1x + a_2x^2 + a_3a^3 + a_4x^4.$$

Since $P_4(x_i) = f(x_i)$ for all $i = 0, \cdots, 4$ then

$$\begin{cases} P_4(x_0) = P_4(0) = a_0 = f(0) = 1, \\ P_4(x_1) = P_4(1) = a_0 + a_1 + a_2 + a_3 + a_4 = f(1) = 2, \\ P_4(x_2) = P_4(3) = a_0 + 3a_1 + 9a_2 + 27a_3 + 81a_4 = f(3) = 2, \\ P_4(x_3) = P_4(5) = a_0 + 5a_1 + 25a_2 + 125a_3 + 625a_4 = f(5) = -1, \\ P_4(x_4) = P_4(6) = a_0 + 6a_1 + 36a_2 + 216a_3 + 1296a_4 = f(6) = -2. \end{cases}$$

Equivalently, we have:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 5 & 25 & 125 & 625 \\ 1 & 6 & 36 & 216 & 1296 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ -1 \\ -2 \end{pmatrix}.$$

After calculations we find: (we can use Chapter 2: Solving Linear systems $AX = b$)

$$P_4(x) = \frac{1}{360}\left(7x^4 - 66x^3 + 53x^2 + 366x + 360\right).$$

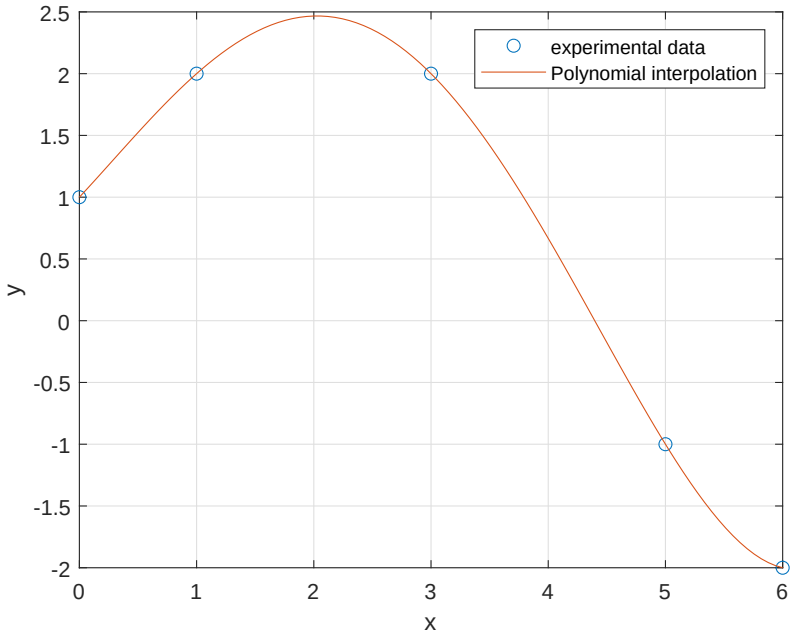


Figure 3.1: *Polynomial interpolation (Example 3.1)*

Question 3.1. How to make sure that the final result is true?

Answer 2. *Using condition 2. it suffices to substitute any value of given x_i in the final result and verify that the value of $P_n(x_i)$ is exactly equal to the given value $f(x_i)$.*

Question 3.2. How many interpolating polynomial one can find?

Answer 3. *The interpolating polynomial, if it exists, is **unique**.*

Question 3.3. Is there any other method to find the interpolating polynomial?

3.2 Lagrange interpolation method

This method is composed of two steps. In step 1. we calculate the Lagrangian components

$$\boxed{L(x_i) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}.}$$

Later in step 2. we formulate the interpolating polynomial as follows:

$$\boxed{P_n(x) = \sum_{i=0}^{n} f(x_i)L(x_i).}$$

Example 3.2. From the data in Example 3.1, find the interpolating polynomial by using Lagrange method.

Solution 3.2. Since we have 5 points we need to find 5 Lagrangian components:

$$\begin{aligned}
L(x_0) = L(0) &= \frac{(x-1)(x-3)(x-5)(x-6)}{(0-1)(0-3)(0-5)(0-6)},\\
L(x_1) = L(1) &= \frac{(x-0)(x-3)(x-5)(x-6)}{(1-0)(1-3)(1-5)(1-6)},\\
L(x_2) = L(3) &= \frac{(x-0)(x-1)(x-5)(x-6)}{(3-0)(3-1)(3-5)(3-6)},\\
L(x_3) = L(5) &= \frac{(x-0)(x-1)(x-3)(x-6)}{(5-0)(5-1)(5-3)(5-6)},\\
L(x_4) = L(6) &= \frac{(x-0)(x-1)(x-3)(x-5)}{(6-0)(6-1)(6-3)(6-5)}.
\end{aligned}$$

Then

$$P_4(x) = \sum_{i=0}^{4} f(x_i)L(x_i) = L(0) - L(5) + 2(L(1) + L(3) - L(5)).$$

After computations, we find the same polynomial:

$$P_4(x) = \frac{1}{360}\left(7x^4 - 66x^3 + 53x^2 + 366x + 360\right).$$

Remark 3.3. The Lagrangian components depend only on x values. Hence, same Lagrangian components $L(x_i)$ can be used to interpolate an infinity of functions that share same nodes x_i.

3.3 Newton divided difference interpolation method

This method is composed of two steps. In step 1. we calculate the divided differences as follows:

First divided difference: $f[x_0, x_1] := \frac{f(x_1)-f(x_0)}{x_1-x_0}$. In general,

$$f[x_i, x_{i+1}] := \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, \quad i = 0, \cdots, n-2.$$

Observe that in the denominator we have $x_{i+1} - x_i$.

Second divided difference: $f[x_0, x_1, x_2] := \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$. In general,

$$f[x_i, x_{i+1}, x_{i+2}] := \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}, \quad i = 0, \cdots, n-1.$$

Observe that in the denominator we have $x_{i+2} - x_i$. So one value (x_{i+1}) is neglected.

Third divided difference: $f[x_0, x_1, x_2, x_3] := \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$. In general,

$$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}] := \frac{f[x_{i+1}, x_{i+2}, x_{i+3}] - f[x_i, x_{i+1}, x_{i+2}]}{x_{i+3} - x_i}, \quad i = 0, \cdots, n-3.$$

Observe that in the denominator we have $x_{i+3} - x_i$. So two values (x_{i+1} and x_{i+2}) are neglected.

...

$\mathbf{n^{th}}$ divided difference: $f[x_0, x_1, \cdot, x_n] := \frac{f[x_1,\cdot,x_n]-f[x_0,x_1,\cdot,x_{n-1}]}{x_n-x_0}$.

Observe that in the denominator we have $x_n - x_0$. So all other values are neglected.

Those values forme the *divided difference's table*, see the next example. Later in step 2. we construct the interpolating polynomial as follows:

$$\boxed{\begin{aligned} P_n(x) =& f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &+ f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned}}$$

Example 3.3. From the data set of Example 3.1, find the interpolating polynomial by using Newton divided differences formula.

Solution 3.3. We start by fulling up the table of divided differences:

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \cdots, x_{i+3}]$	$f[x_i, \cdots, x_{i+3}, x_{i+4}]$
0	1				
1	2	$\frac{2-1}{1-0} = 1$			
3	2	$\frac{2-2}{3-1} = 0$	$\frac{0-1}{3-0} = \frac{-1}{3}$		
5	-1	$\frac{-1-2}{5-3} = \frac{-3}{2}$	$\frac{\frac{-3}{2}-0}{5-1} = \frac{-3}{8}$	$\frac{\frac{-3}{8}-\frac{-1}{3}}{5-0} = \frac{-1}{120}$	
6	-2	$\frac{-2-(-1)}{6-5} = -1$	$\frac{-1-\frac{-3}{2}}{6-3} = \frac{1}{6}$	$\frac{\frac{1}{6}-\frac{-3}{8}}{6-1} = \frac{13}{120}$	$\frac{\frac{13}{120}-\frac{-1}{120}}{6-0} = \frac{7}{360}$

Then we apply Newton's formula:

$$\begin{aligned} P_4(x) &= 1 + 1(x-0) - \frac{1}{3}(x-0)(x-1) - \frac{1}{120}(x-0)(x-1)(x-3) \\ &\quad + \frac{7}{360}(x-0)(x-1)(x-3)(x-5) \end{aligned}$$

to find

$$P_4(x) = \frac{1}{360}\left(7x^4 - 66x^3 + 53x^2 + 366x + 360\right).$$

3.4 Newton finite difference interpolation method

In particular, if the distance between x_i and x_{i+1} (called the *step*) is constant, then the calculus can be simplified as follows:

First finite difference: $\Delta f(x_0) := f(x_1) - f(x_0)$. In general,

$$\Delta f(x_i) := f(x_{i+1}) - f(x_i), \quad i = 0, \cdots, n-1.$$

Observe that there is no fraction here.

Second finite difference: $\Delta^2 f(x_0) := \Delta f(x_1) - \Delta f(x_0)$. In general,

$$\Delta^2 f(x_i) := \Delta f(x_{i+1}) - \Delta f(x_i), \quad i = 0, \cdots, n-2.$$

n^{th} finite difference: $\Delta^n f(x_0) := \Delta^{n-1} f(x_1) - \Delta^{n-1} f(x_0)$.

Those values forme the *finite difference's table*, see the next example.

Later in step 2. we construct the interpolating polynomial as follows:

$$\boxed{\begin{aligned} P_n(x) =& f(x_0) + \frac{\Delta f(x_0)}{1!\ h}(x - x_0) + \frac{\Delta f(x_0)}{2!\ h^2}(x - x_0)(x - x_1) + \cdots \\ &+ \frac{\Delta^n f(x_0)}{n!\ h^n}(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}}$$

where $h = x_{i+1} - x_i$ is the constant step between the nodes of data set.

Remark 3.4. Here we can not copy the Data set of example 1.3 because the step there is not constant.

Example 3.4. Find the interpolating polynomial of the following data set by using Newton finite difference interpolation formula:

x_i	1	3	5	7
$f(x_i)$	2	4	0	2

Solution 3.4. We start by fulling up the table of finite differences:

x_i	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$
1	2			
3	4	$4-2=2$		
5	0	$0-4=-4$	$-4-2=-6$	
7	2	$2-0=2$	$2-(-4)=6$	$6-(-6)=12$

Then we apply Newton's formula (here $h = 2$):

$$\begin{aligned} P_3(x) &= 2 + \frac{2}{1!\ 2}(x-1) - \frac{6}{2!\ 2^2}(x-1)(x-3) + \frac{12}{3!\ 2^3}(x-1)(x-3)(x-5) \\ &= \frac{1}{4}\left(x^3 - 12x^2 + 39x - 20\right). \end{aligned}$$

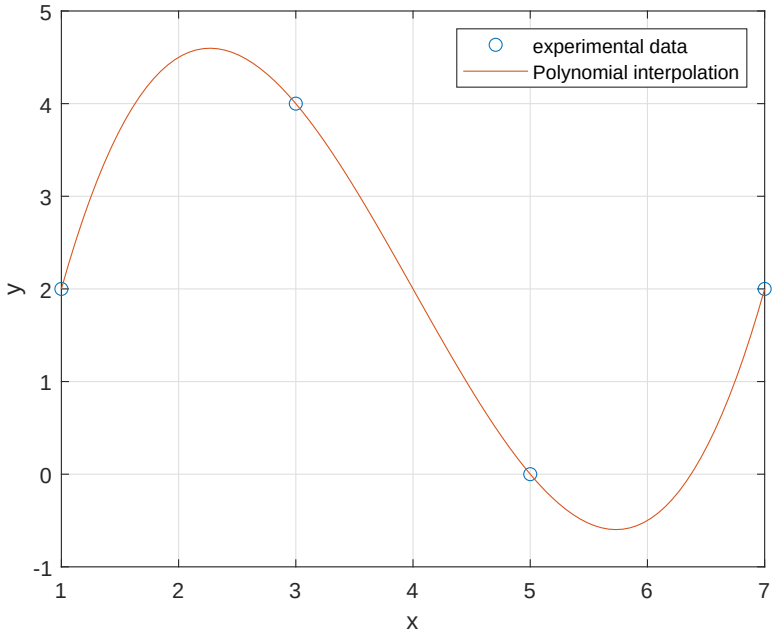


Figure 3.2: *Polynomial interpolation with constant step (Example 3.4)*

Exercise 3.1. Find the advantages and disadvantages of each of the previous methods.

3.5 Error for polynomial interpolation

Definition 3.1. For any $x \in [x_0, x_n]$, we define by *the error at* x the distance between the exact value $f(x)$ and the approached value $P_n(x)$:

$$\forall x \in [x_0, x_n], \quad E(x) = |f(x) - P_n(x)|.$$

Recall 3.1. From interpolation conditions we know that, for any x_i, $i = 0, \cdots n$, $E(x_i) = 0$ and that the error is bounded elsewhere.

Theorem 3.1. Suppose that the function f is $(n+1)$ times continuously differentiable over

$[x_0, x_n]$. Then, for any given $y \in [x_0, x_n]$, the error at point y is given by:

$$|E(y)| = \frac{\left|f^{(n+1)}(\xi)\right|}{(n+1)!} \left|\prod_{i=0}^{n}(y - x_i)\right| \quad \text{for some} \quad \xi \in]x_0, x_n[. \tag{3.1}$$

Remark 3.5. Since we do not know the exact value of ξ, we can not apply equation (3.1) in its previous form. We shall replace the term $|f^{(n+1)}(\xi)|$ by its maximum absolute value over $[x_0, x_n]$. This gives a computable estimation of $|E(y)|$:

$$\boxed{|E(y)| \leq \frac{\max\limits_{x\in[x_0,x_n]}\left|f^{(n+1)}(x)\right|}{(n+1)!} \left|\prod_{i=0}^{n}(y - x_i)\right|.} \tag{3.2}$$

Example 3.5. Consider $f(x) = \sqrt{x}$. Take three nodes $x_1 = 100,\ x_2 = 121$ and $x_3 = 144$. Find an estimate for the error of interpolation at $y = 115$.

Solution 3.5. We shall apply equation (3.2) with $n + 1 = 3$ points. We have

$$f(x) = \sqrt{x} \implies f'(x) = \frac{1}{2\sqrt{2}} \implies f''(x) = \frac{-1}{4x^{\frac{3}{2}}} \implies f'''(x) = \frac{3}{8x^{\frac{5}{2}}}.$$

So $\max\limits_{x\in[100,144]}\left|f'''(x)\right| = 0.00000375$. Then

$$|E(115)| \leq \frac{0.00000375}{3!} \left|(115 - 100)(115 - 121)(115 - 144)\right| \leq 0.00163125.$$

Remark 3.6. By replacing the term $\left|\prod\limits_{i=0}^{n}(y - x_i)\right|$ by its maximum absolute value over $[x_0, x_n]$, we get *the maximum error possible on* $[x_0, x_n]$:

$$\boxed{\max_{x\in[x_0,x_n]}|E(x)| \leq \frac{\max\limits_{x\in[x_0,x_n]}\left|f^{(n+1)}(x)\right|}{(n+1)!} \max_{x\in[x_0,x_n]}\left|\prod_{i=0}^{n}(x - x_i)\right|.} \tag{3.3}$$

Example 3.6. For the example 3.5 find an estimate for the maximum error for interpolation possible on $[100, 144]$.

Solution 3.6. Here we shall apply equation (3.3). We have:

$$\begin{aligned}\prod_{i=0}^{n}(x - x_i) &= (x - 100)(x - 121)(x - 144)\\ &= x^3 - 365x^2 + 43924x - 1742400.\end{aligned}$$

The table of variation of this equation is the following:

x	100	108.96058763	134.37274570	144
derivative	+	0	−	0	+	
$\prod_{i=0}^{n}(x-x_i)$	$0 \nearrow 3780.059149 \searrow -4425.244334 \nearrow 0$			

Then

$$\max_{x\in[x_0,x_n]}\Big|\prod_{i=0}^{n}(x-x_i)\Big| = 4425.244334.$$

Hence

$$\max_{x\in[x_0,x_n]}|E(x)| \leq \frac{0.00000375}{3!}\ 4425.244334 = 0.0027657777.$$

Remark 3.7. Observe that if n exceeds 3, the table of variation of $\prod_{i=0}^{n}(x-x_i)$ becomes very hard to be determined. In the particular case of equally spaced nodes x_i, we have

$$\max_{x\in[x_0,x_n]}\Big|\prod_{i=0}^{n}(x-x_i)\Big| \leq \frac{1}{4}h^{n+1}n!$$

with $h = x_{i+1} - x_i$ is the constant step. Substituting this result in equation (3.3), we get:

$$\boxed{\max_{x\in[x_0,x_n]}|E(x)| \leq \frac{h^{n+1}}{4(n+1)}\ \max_{x\in[x_0,x_n]}\Big|f^{(n+1)}(x)\Big|.} \tag{3.4}$$

Remark 3.8. It is worthy to mention that all the above formulas depend on $f^{(n+1)}$ which is not known unless f itself is known (and $(n+1)$-times differentiable). Whereas if f is well known then no need to approximate it by a polynomial !

3.6 Exercises

Exercise 3.2. By recording the temperatures of a plate in relation to its length in a refrigeration installation, the following table was obtained:

Length (x)	0	2	4	8	10
Temperature (y)	-1	1	0	2	5

1. Find the Lagrange interpolating polynomial for the above data on $[0, 10]$. Is it necessary to calculate $L(4)$? Why?
2. Find the interpolating polynomial in Newton form of the above data on $[0, 10]$. Is it the same polynomial found in question 1.? Why?
3. Can we use the table of finite differences to find the interpolating polynomial? Why?
4. Add the point $(6, 1)$ and find the interpolating polynomial of the temperature on $[0, 10]$ by all possible methods. What do you notice?

5. We repeated the same experiment for another refrigeration installation and we obtained the following table:

Length (x)	0	2	4	6	8	10
Temperature (y)	-2	-1	0	1.5	2.1	3

Find the interpolating polynomial on $[0, 10]$ by a method of your choice. Justify your choice.

This exercise helps you to solve the exercise 1.1.

Exercise 3.3.

1. Find the Lagrange interpolating polynomials of functions f, g and h using the following values:

x_i	-1	2	4	5
$f(x_i)$	-2	43	213	376
$g(x_i)$	104	83	21	6
$h(x_i)$	4	0	-2	0

What do you notice?

2. Add the point $f(6) = 400$ and reconstruct the interpolating polynomial of f. What do you notice?
3. Change the point $(5, g(5) = 6)$ by the point $(7, g(7) = 6)$ and reconstruct the interpolating polynomial of g. What do you notice?
4. Redo the exercise by Newton's method. What do you notice?

This exercise also helps you to solve the exercise 1.1.

Exercise 3.4. Round answers to eight decimal places

Consider the function $f(x) = e^{-\frac{x}{10}}$ on the interval $[0, 3]$ by the following table:

x	0	1	2	3
$f(x)$	1	0.904837	0.818731	0.740818

1. Approximate the value of f at the point $x = 1.5$ using the Lagrange interpolating polynomial.
2. Using the exact value $f(1.5)$ and the approximated value $P(1.5)$ found in question 1., calculate the exact error of interpolation at the point $x = 1.5$.
3. Using the error form for interpolation, determine the error at the point $x = 1.5$.
4. Determine the maximum error bound when the polynomial is used to approximate $f(x)$ for $x \in [0, 3]$.

5. Find the polynomial of interpolation of $f(x)$ using both methods of Newton.

Hint: 2. Calculate $|f(1.5) - P(1.5)|$**. 3. Use Inequality** (3.2)**. 4. Use Inequality** (3.3)**, the polynomial** $2x^3 - 9x^2 + 11x - 3$ **vanishes at points** $x_1 = 0.381966012,\ x_2 = 1.5$ **and** $x_3 = 2.618033989$**.**

Exercise 3.5.

1. Complete the following table of divided differences:

x	$f(x)$				
	-1				
1		0			
	-11	-10	-5		
3	-61				
4		-144		-9	

2. Choose the correct answer and justify your choice: The interpolating polynomial of $f(x)$ is:
   (a) $P(x) = -x^4 + x^3 + 2x^2 + x - 2$.
   (b) $P(x) = -x^4 + x^3 - x^2 + x - 1$.
   (c) $P(x) = x^4 + x^3 - x^2 - x - 1$.
3. Can we find an error bound for the polynomial interpolation? Why?

Hint: 2. Use Answer 2 of Question 3.1. 3. Remember Remark 3.8.

Exercise 3.6.

1. Determine the interpolating polynomial $P_3(x)$ of degee at the most 3 of the function $f(x) = \sin(\frac{\pi(x+1)}{3})$ on the interval $[-1, 5]$ with a step size $h = 2$ using the method of divided differences.
2. Find the worst case (maximum) estimate for the error that is valid for x throught the interval $[-1, 5]$, i.e., find a constante M such that:

$$\max_{x\in[-1,5]} |E(x)| = \max_{x\in[-1,5]} |f(x) - P_3(x)| \leq M.$$

Hint: 2. You can use inequality (3.3) **or inequality** (3.4)**.**

Exercise 3.7. Application

1. Suppose that a function $f(\cdot)$ defined on an interval $[a, b]$ is known on one point $x = \frac{a+b}{2}$.
   (a) Interpolate the function $f(\cdot)$ by a polynomial $P(\cdot)$.
   (b) Approximate $\int_a^b f(x)\ dx$ by $\int_a^b P(x)\ dx$.

2. Suppose now that the function f is known on two points $x_0 = a$ and $x_1 = b$. Redo question 1.

3. Redo question 1. if the function f is known on three points $x_0 = a,\ x_1 = \frac{a+b}{2}$ and $x_2 = b$.

Hint: This exercise introduce the numerical integration process, see Chapter 5.

Chapter 4
Least squares approximation

4.1 Problem formulation

The polynomial interpolation is the simplest way to fit a curve to experimental data (x_i, y_i), hence it is the key to solve many problems like Problem 1. However, it has some disadvantages.

Problem 2.

- *If the number of interpolation points $(n+1)$ is high, then the degree of the interpolation polynomial (n) is also high. As a result, the interpolation polynomial $(P_n(x))$ is not necessarily convergent to the unknown function (f) generating the data and the interpolation error risks to be very important. This does not fullfill the interpolaion condition 3.*
- *Since the data set (interpolation points) are experimental then some degree of error is made when measuring them. Hence it is not "right" to construct a curve that goes "exactly" throw every data point. However this is the essential interpolaion condition 2.!*

In this Chapter instead of supposing that

$$P_n(x_i) = y_i, \quad \text{for all} \quad i = 1, \cdots, n+1$$

we will minimize the sum of the squares of the distances between the polynomial $P_n(x_i)$ and the data $y_i = f(x_i)$ over all the points x_i:

$$S = \sum_{i=1}^{n+1} (P_n(x_i) - y_i)^2.$$

We call this technique *the approximation of a data set of points with a polynomial P_n in least squares sens.*

To avoid the first disadvantage cited above and to make calculus shorter, we can ignore the interpolation condition 1. In other words, we can approximate the data points with any polynomial of any degree. For example, we can aproximate a set of $n+1$ points by a straight line $y = ax + b$ which is a polynomial of degree 1.

Moreover, by observing the "shape of data points" in a graph, we can approximate them by any function following the sens of data. This means that not only polynomials can fit the experimental data. The only condition is that the sum of the squares of the distances between the approximation function (denoted $g(x_i)$) and the data y_i:

$$S = \sum_{i=1}^{n+1} (g(x_i) - y_i)^2$$

is minimized.

4.2 Construction of the line of best fit

Let (x_i, y_i) for $i = 1, \cdots, n$ be a set of experimental data points. We want to approximate this data by the straight line $g(x) = a_0 + a_1 x$ which minimizes the quantity:

$$S = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 = \sum_{i=1}^{n} (a_0 + a_1 x_i - y_i)^2.$$

This amounts to find where the partial derivatives $\dfrac{\partial S}{\partial a_0}$ and $\dfrac{\partial S}{\partial a_1}$ vanish:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\displaystyle\sum_{i=1}^{n} (a_0 + a_1 x_i - y_i) = 0, \\ \dfrac{\partial S}{\partial a_1} = 2\displaystyle\sum_{i=1}^{n} x_i (a_0 + a_1 x_i - y_i) = 0. \end{cases}$$

This implies that

$$\begin{cases} a_0 \boxed{\displaystyle\sum_{i=1}^{n} 1} + a_1 \displaystyle\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i, \\ a_0 \displaystyle\sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i. \end{cases}$$

Observe that $\displaystyle\sum_{i=1}^{n} 1 = n$, hence:

$$\begin{pmatrix} \boxed{n} & \displaystyle\sum_{i=1}^{n} x_i \\ \displaystyle\sum_{i=1}^{n} x_i & \displaystyle\sum_{i=1}^{n} x_i^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{i=1}^{n} y_i \\ \displaystyle\sum_{i=1}^{n} x_i y_i \end{pmatrix}.$$

The resolution of this system conducts to the desired result.

Example 4.1. Let

x_i	10	20	30	40	50	60
y_i	0.33	0.8	1.31	1.61	2.01	2.26

Using the least square method, approximate the above data by a straight line.

Solution 4.1. We want to approximate this data of 6 points by the straight line $g(x) = a_0 + a_1 x$ which minimizes the quantity:

$$S = \sum_{i=1}^{6} (a_0 + a_1 x_i - y_i)^2.$$

Putting the partial derivatives $\dfrac{\partial S}{\partial a_0}$ and $\dfrac{\partial S}{\partial a_1}$ null gives:

$$\begin{cases} 6\ a_0 + a_1 \displaystyle\sum_{i=1}^{6} x_i = \sum_{i=1}^{6} y_i, \\ a_0 \displaystyle\sum_{i=1}^{6} x_i + a_1 \sum_{i=1}^{6} x_i^2 = \sum_{i=1}^{6} x_i y_i. \end{cases}$$

This implies that

$$\begin{cases} 6\ a_0 + 210 a_1 = 8.32, \\ 210 a_0 + 9100 a_1 = 359.1. \end{cases}$$

Solving this system gives:

$$a_0 = 0.0287 \quad \text{and} \quad a_1 = 0.0388.$$

Hence the line of best fit is

$$g(x) = 0.0287 + 0.0388\ x.$$

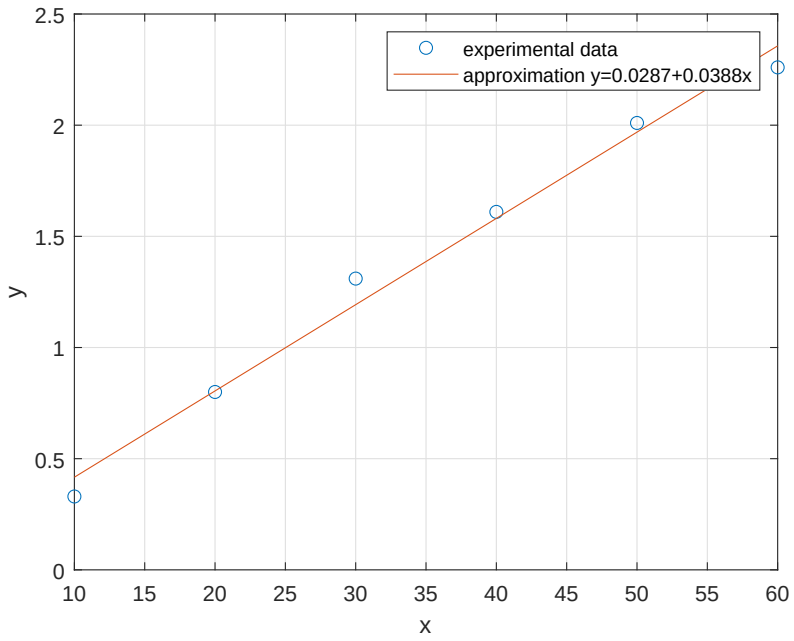


Figure 4.1: *Linear approximation (Example 4.1)*

4.3 Construction of the polynomial of best fit

Let (x_i, y_i) for $i = 1, \cdots, n$ be a set of experimental data points. We want to approximate this data by the polynomial of degree k:

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

which minimizes the quantity:

$$S = \sum_{i=1}^{n} (a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_k x_i^k - y_i)^2.$$

This amounts to find where all the partial derivatives $\dfrac{\partial S}{\partial a_j}$ for $j = 1, \cdots, k$ vanish:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\displaystyle\sum_{i=1}^{n} (a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_k x_i^k - y_i) = 0, \\ \dfrac{\partial S}{\partial a_1} = 2\displaystyle\sum_{i=1}^{n} x_i (a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_k x_i^k - y_i) = 0, \\ \vdots \\ \dfrac{\partial S}{\partial a_k} = 2\displaystyle\sum_{i=1}^{n} x_i^k (a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_k x_i^k - y_i) = 0. \end{cases}$$

This implies that

$$\begin{cases} a_0 \boxed{\displaystyle\sum_{i=1}^{n} 1} + a_1 \displaystyle\sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 + \cdots + a_k \sum_{i=1}^{n} x_i^k = \sum_{i=1}^{n} y_i, \\ a_0 \displaystyle\sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + a_2 \sum_{i=1}^{n} x_i^3 + \cdots + a_k \sum_{i=1}^{n} x_i^{k+1} = \sum_{i=1}^{n} x_i y_i, \\ \vdots \\ a_0 \displaystyle\sum_{i=1}^{n} x_i^k + a_1 \sum_{i=1}^{n} x_i^{k+1} + a_2 \sum_{i=1}^{n} x_i^{k+2} + \cdots + a_k \sum_{i=1}^{n} x_i^{2k} = \sum_{i=1}^{n} x_i^k y_i. \end{cases}$$

Or equivalently:

$$\begin{pmatrix} \boxed{n} & \displaystyle\sum_{i=1}^{n} x_i & \displaystyle\sum_{i=1}^{n} x_i^2 & \cdots & \displaystyle\sum_{i=1}^{n} x_i^k \\ \displaystyle\sum_{i=1}^{n} x_i & \displaystyle\sum_{i=1}^{n} x_i^2 & \displaystyle\sum_{i=1}^{n} x_i^3 & \cdots & \displaystyle\sum_{i=1}^{n} x_i^{k+1} \\ \vdots & & \ddots & & \vdots \\ \displaystyle\sum_{i=1}^{n} x_i^k & \displaystyle\sum_{i=1}^{n} x_i^{k+1} & \displaystyle\sum_{i=1}^{n} x_i^{k+2} & \cdots & \displaystyle\sum_{i=1}^{n} x_i^{2k} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{i=1}^{n} y_i \\ \displaystyle\sum_{i=1}^{n} x_i y_i \\ \vdots \\ \displaystyle\sum_{i=1}^{n} x_i^k y_i \end{pmatrix}.$$

The resolution of this system conducts to the desired result.

Example 4.2. Consider the following time series data:

x_i	0	1	2	3	4	5
y_i	2.1	7.7	13.6	27.2	40.9	61.1

Using the least square method, approximate the above data by a polynomial of degree 2.

Solution 4.2. We want to approximate the above data of 6 points by the polynomial of degree 2: $g(x) = a_0 + a_1x + a_2x^2$ by minimizing the quantity:

$$S = \sum_{i=1}^{6}(a_0 + a_1x_i + a_2x_i^2 - y_i)^2.$$

Putting the three partial derivatives $\dfrac{\partial S}{\partial a_0}$, $\dfrac{\partial S}{\partial a_1}$ and $\dfrac{\partial S}{\partial a_2}$ null gives:

$$\begin{pmatrix} 6 & \sum\limits_{i=1}^{6} x_i & \sum\limits_{i=1}^{6} x_i^2 \\ \sum\limits_{i=1}^{6} x_i & \sum\limits_{i=1}^{6} x_i^2 & \sum\limits_{i=1}^{6} x_i^3 \\ \sum\limits_{i=1}^{6} x_i^2 & \sum\limits_{i=1}^{6} x_i^3 & \sum\limits_{i=1}^{6} x_i^4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \sum\limits_{i=1}^{n} y_i \\ \sum\limits_{i=1}^{n} x_iy_i \\ \sum\limits_{i=1}^{n} x_i^2y_i \end{pmatrix}.$$

This implies that

$$\begin{pmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{pmatrix}.$$

After calculation, we get

$$a_0 = 2.48, \quad a_1 = 2.36 \quad \text{and} \quad a_2 = 1.86.$$

Hence the best approximation with a polynomial of degree 2 is made by

$$g(x) = 2.48 + 2.36\ x + 1.86\ x^2.$$

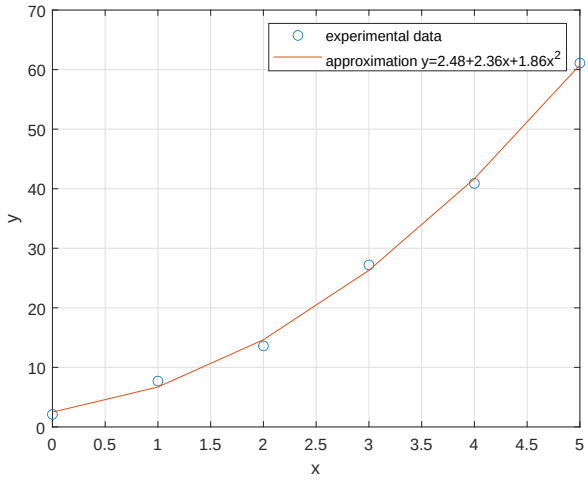


Figure 4.2: *Polynomial approximation (Example 4.2)*

4.4 Construction of the function of best fit

According to the least squares approach, it is not necessary to approximate the experimental data with a polynomial. However, any function that follows the sens of the data can be used.

Example 4.3. Let

x_i	0.5	0.75	1	1.5	2	2.25	2.75	3
y_i	-1.19	-0.45	-0.07	0.71	1.16	1.44	1.72	1.84

1. Plot these points on an orthogonal plane and observe their appearance.
2. Using the least squares method, approximate the above data by the function $g(x) = a\ln(x)$ with $a \in \mathbb{R}$ to be determined.

Solution 4.3. We start by plotting the experimental data:

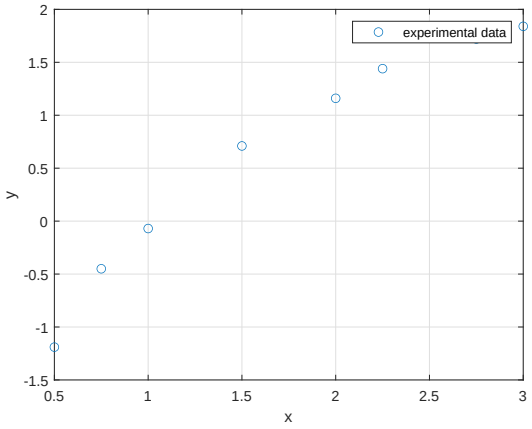


Figure 4.3: *Experimental Data (Example 4.3)*

We observe that the shape of these points is like the ln function except that the graph here is shifted to the left. So we shall approximate those data with the ln function multiplied by a weight a. In other words, we need to find the function $g(x) = a\ln(x)$ which minimizes the quantity:

$$S = \sum_{i=1}^{8}(a\ln(x_i) - y_i)^2, \quad \text{here 8 is the number of given points.}$$

This amounts to find where the partial derivative $\dfrac{\partial S}{\partial a}$ vanishes:

$$\frac{\partial S}{\partial a} = 2\sum_{i=1}^{8}\ln(x_i)(a\ln(x_i) - y_i) = 0.$$

By simplifying this equality, we get

$$a = \frac{\sum\limits_{i=1}^{8} y_i\ln(x_i)}{\sum\limits_{i=1}^{8}(\ln(x_i))^2}.$$

After calculations, we find

$$a = 1.7.$$

As a result, the best non-linear approximation of the above data is the function

$$g(x) = 1.7\ln(x).$$

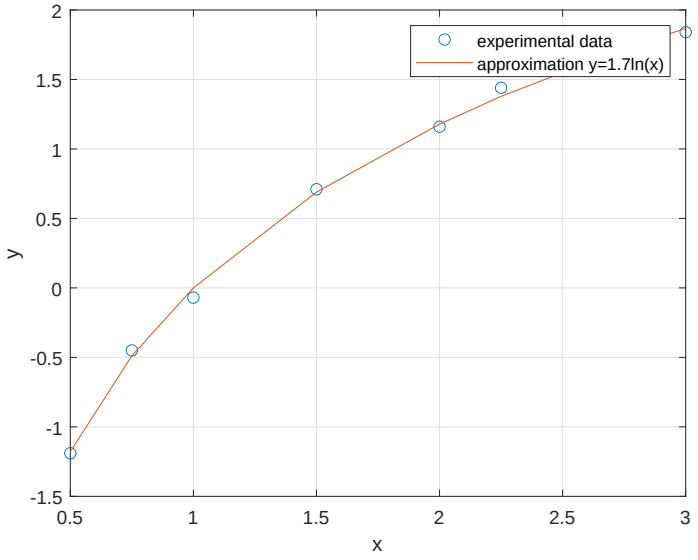


Figure 4.4: *Non-linear approximation (Example 4.3)*

Example 4.4. Let

x_i	0.2	0.5	1	1.5	2	3
y_i	0.3	0.5	0.8	1	1.2	1.3

and put $g(x) = a\frac{x}{x+1} + b(1 - e^{-x})$. Find a and b such that $g(x)$ be the approximation of best fit in the sens of least squares.

Solution 4.4. We need to find the function $g(x) = a\frac{x}{x+1} + b(1 - e^{-x})$ which minimizes the quantity:

$$S = \sum_{i=1}^{6} \left(a\frac{x_i}{x_i + 1} + b(1 - e^{-x_i}) - y_i \right)^2.$$

This amounts to find where the partial derivatives $\dfrac{\partial S}{\partial a}$ and $\dfrac{\partial S}{\partial b}$ vanish:

$$\begin{cases} \dfrac{\partial S}{\partial a} = 2\displaystyle\sum_{i=1}^{6} \frac{x_i}{x_i + 1} \left(a\frac{x_i}{x_i + 1} + b(1 - e^{-x_i}) - y_i \right) = 0, \\ \dfrac{\partial S}{\partial b} = 2\displaystyle\sum_{i=1}^{6} (1 - e^{-x_i}) \left(a\frac{x_i}{x_i + 1} + b(1 - e^{-x_i}) - y_i \right) = 0. \end{cases}$$

By simplifying the above equalities, we get

$$\begin{cases} a\displaystyle\sum_{i=1}^{6} \left(\frac{x_i}{x_i + 1} \right)^2 + b\sum_{i=1}^{6} \frac{x_i}{x_i + 1}(1 - e^{-x_i}) = \sum_{i=1}^{6} y_i \left(\frac{x_i}{x_i + 1} \right), \\ a\displaystyle\sum_{i=1}^{6} \frac{x_i}{x_i + 1}(1 - e^{-x_i}) + b\sum_{i=1}^{6} (1 - e^{-x_i})^2 = \sum_{i=1}^{6} y_i(1 - e^{-x_i}). \end{cases}$$

After calculations, we find

$$\begin{cases} 1.7558\ a + 2.2327\ b = 2.9917, \\ 2.2327\ a + 2.8413\ b = 3.8066. \end{cases}$$

This implies

$$a = 0.3745 \quad \text{and} \quad b = 1.0455.$$

As a result, the best non-linear approximation of the above data is the function

$$g(x) = 0.3745\frac{x}{x + 1} + 1.0455(1 - e^{-x}).$$

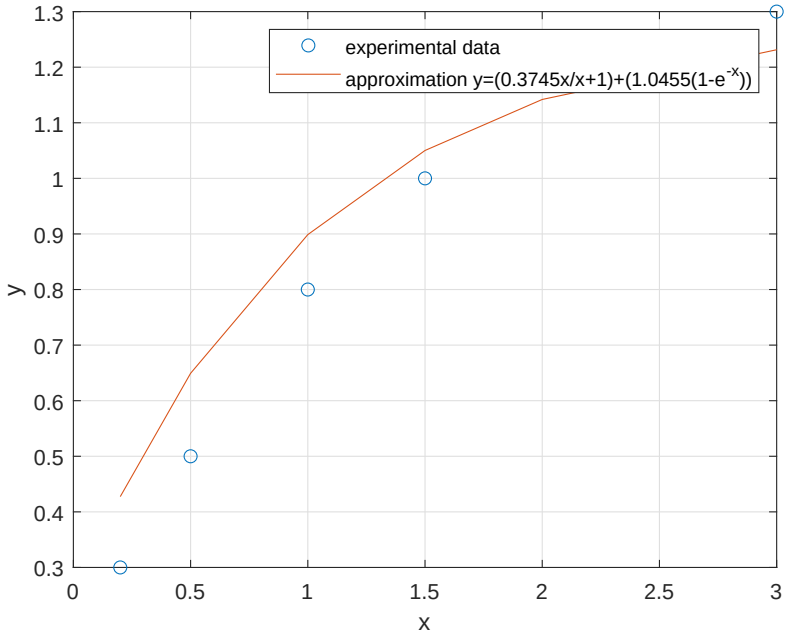


Figure 4.5: *Non-linear approximation (Example 4.4)*

Example 4.5. Let

x_i	0	1	2	3	4
y_i	1.5	2.5	3.5	5	7.5

Find the exponential function $g(x) = ce^{ax}$ which fits the above data in the least square sens.

Solution 4.5. We need to minimize the quantity

$$S = \sum_{i=1}^{5} (ce^{ax_i} - y_i)^2.$$

So, we look where the derivatives $\dfrac{\partial S}{\partial a}$ and $\dfrac{\partial S}{\partial c}$ vanish:

$$\begin{cases} \dfrac{\partial S}{\partial a} = 2\displaystyle\sum_{i=1}^{5} cx_i e^{ax_i} \left(ce^{ax_i} - y_i\right) = 0, \\ \dfrac{\partial S}{\partial c} = 2\displaystyle\sum_{i=1}^{5} e^{ax_i} \left(ce^{ax_i} - y_i\right) = 0. \end{cases}$$

By simplifying the above equalities, we get

$$\begin{cases} \displaystyle\sum_{i=1}^{5} cx_i e^{2ax_i} = \sum_{i=1}^{5} x_i y_i e^{ax_i}, \\ \displaystyle\sum_{i=1}^{5} ce^{2ax_i} = \sum_{i=1}^{5} y_i e^{ax_i}. \end{cases}$$

Clearly, the simplification of those equations is no more possible and the culculations are not easy to do. We can use the Chapter 1: Solving nonlinear equations $f(x) = 0$.

Here we will proceed in another way. Instead of minimizing the distances between ce^{ax_i} and y_i, we will minimize the distances between $\ln(ce^{ax_i})$ and $\ln(y_i)$. Put

$$\begin{aligned} T &= \sum_{i=1}^{5} \left(\ln\left(ce^{ax_i}\right) - \ln(y_i)\right)^2 \\ &= \sum_{i=1}^{5} \left(\ln(c) + ax_i - \ln(y_i)\right)^2, \quad (c, y_i) > (0, 0). \end{aligned}$$

Put $\dfrac{\partial T}{\partial a} = 0$ and $\dfrac{\partial T}{\partial c} = 0$ to get:

$$\begin{cases} \dfrac{\partial T}{\partial a} = 2\displaystyle\sum_{i=1}^{5} x_i \left(\ln(c) + ax_i - \ln(y_i)\right) = 0, \\ \dfrac{\partial T}{\partial c} = 2\dfrac{1}{c}\displaystyle\sum_{i=1}^{5} \left(\ln(c) + ax_i - \ln(y_i)\right) = 0, \qquad (c, y_i) > (0, 0). \end{cases}$$

Simplify the above result to find

$$\begin{cases} \ln(c)\displaystyle\sum_{i=1}^{5} x_i + a\sum_{i=1}^{5} x_i^2 = \sum_{i=1}^{6} x_i \ln(y_i), \\ \ln(c)\displaystyle\sum_{i=1}^{5} 1 + a\sum_{i=1}^{6} x_i = \sum_{i=1}^{5} \ln(y_i). \end{cases}$$

In numbers:

$$\begin{cases} 10\ln(c) + 30a = 16.29, \\ 5\ln(c) + 10a = 6.2. \end{cases}$$

After calculations, we obtain

$$\ln(c) = 0.462 \implies c = 1.5872 \quad \text{and} \quad a = 0.389.$$

As a result, the best non-linear approximation of the above data is the function

$$g(x) = 1.5872\ e^{0.389x}.$$

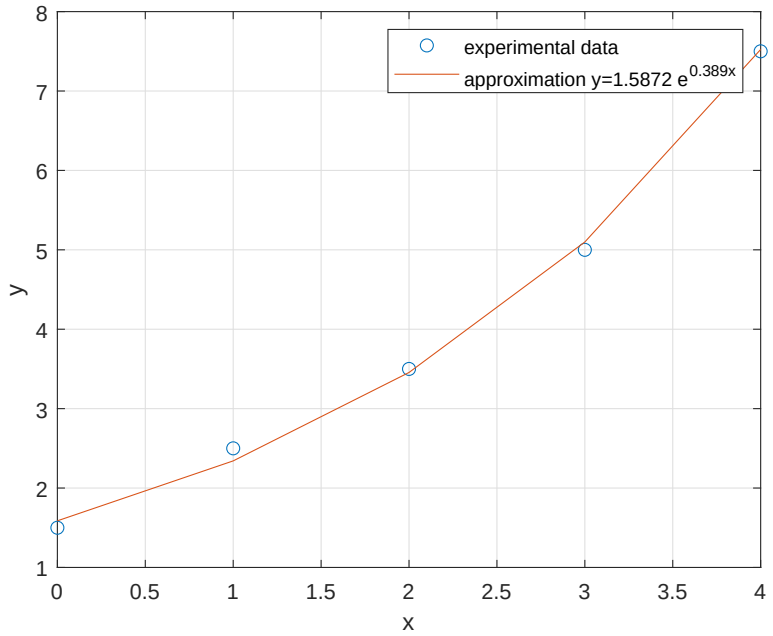


Figure 4.6: *Non-linear approximation (Example 4.5)*

Remark 4.1. When we deal with nonlinear functions, the direct calculations seem very difficult. In some cases, a simple change of variables makes the computations much easier: see example 4.5 for

- minimization of **exponential** quantity $S = \sum_{i=1}^{n} \left(ae^{bx} - y_i\right)^2$ which turns to minimize $T = \sum_{i=1}^{n} \left(\ln(a) + bx_i - \ln(y_i)\right)^2$. Also
- minimising the **power** quantity $S = \sum_{i=1}^{n} \left(ax_i^b - y_i\right)^2$ is equivalent to minimize $T = \sum_{i=1}^{n} \left(\ln(a) + b\ln(x_i) - \ln(y_i)\right)^2$ with $(a, x_i, y_i) > (0, 0, 0)$; and
- minimizing the **saturation-growth rate** quantity $S = \sum_{i=1}^{n} \left(\frac{ax_i}{b+x_i} - y_i\right)^2$ is equivalent to minimize $T = \sum_{i=1}^{n} \left(\frac{1}{a} + \frac{b}{a}x_i - y_i\right)^2$.

4.5 Exercises

1. Use least squares regression to fit a straight line to

x_i	-2	-1	0	1	2
$f(x_i)$	2	1	0	1	2

Plot the data points and the regression line to see how well the line represents the points.

2. Fit the following data to a parabolic model $y = ax^2 + bx$ using least squares

x_i	1	2	3	4	5	6	7	8
$f(x_i)$	2.5	7	38	55	61	77	83	145

3. Find the least squares parabola $y = ax^2 + bx + c$ that fits to the following data set:

x_i	0	1	2	3	4	5
$f(x_i)$	2.1	7.7	13.6	27.2	40.9	61.1

4. Consider the experimental points $f(0) = 1$, $f(1) = 3$ and $f(2) = 7$. Find the function $g(x) = a\sqrt{|x-1|} + bx^2$ that fits to the above data set.

5. Fit the following data points with the power model $y = ax^b$. Use the resulting power equation to predict y at $x = 7$.

x_i	0	2	4	6	9	11	12	15	17	19
$f(x_i)$	5	6	7	6	9	8	7	10	12	12

6. Fit an exponential model to

x_i	0.4	0.8	1.2	1.6	2	2.3
$f(x_i)$	800	975	1500	1950	2900	3600

7. The following data represents the scientific model $y = \frac{ax}{b-x}$. Use the method of least squares to find coefficients a and b of the equation.

x_i	18	22	26	28	30	36	46
$f(x_i)$	3.6	3.8	3.9	4	4.1	4.2	4.3

8. The data tabulated below can be modeled by $y = \left(\frac{a+\sqrt{x}}{b\sqrt{x}}\right)^2$. Use a transformation to linearize this equation and then employ linear regression to determine a and b. Based on your analysis predict y at $x = 1.6$.

x_i	0.5	1	2	3	4
$f(x_i)$	10.4	5.8	3.3	2.4	2

9. Let the set of points $(0,2)$, $(1,1)$, $(2,2/3)$, $(4,1/3)$ and $(10,1/10)$. Propose a function that best fits these points.

10. Consider the experimental data:

x_i	0.2	0.5	1	1.5	2	3
$f(x_i)$	0.3	0.5	0.8	1	1.2	1.3

We decide to adopt the following model

$$g(x) = a\frac{x}{x+1} + b(1 - e^{-x}).$$

Using least squares approximation, find a and b then plot the data points and the function g in the same plane.

Chapter 5

Numerical integration

5.1 Introduction

Geometrically, the definite integral of a continuous function $f(x)$ on a finite interval $[a, b]$, denoted by $\int_a^b f(x)dx$, is the area of the closed geometric shape bounded by the abscissa axes, the straight lines $x = a$ and $x = b$, and the graph of the function $f(x)$:

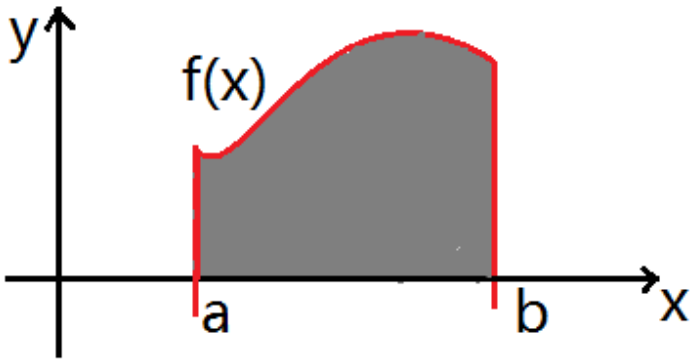


Figure 5.1: *Definite integrals geometrically*

Theoretically,

$$I := \int_a^b f(x)dx = F(b) - F(a)$$

with $F(\cdot)$ is the primitive of $f(\cdot)$.

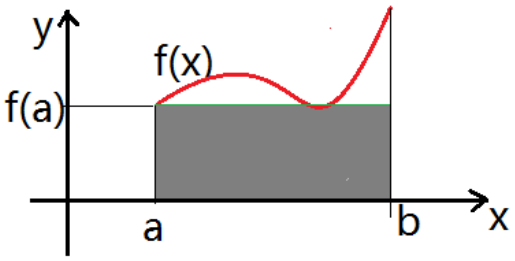
Problem 3. *Sometimes it is not easy to find the expression of $F(\cdot)$. Moreover, in practice, we usually do not have the expression of function $f(\cdot)$, we only have some couples of points $(x_i, f(x_i))_{i=1,\dots,n}$ measured by physical experiments. Hence the notion of primitive loses its sense and numerical methods become highly recommended.*

5.2 Rectangular rule

5.2.1 Left rectangle approximation

In this method, we draw a rectangle which its width is $b-a$ and its height is given by the height of the function f at the left hand point a: $f(a)$. Hence

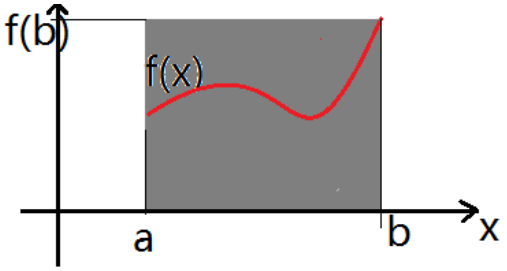
$$I := \int_a^b f(x)dx \approx I_l := (b-a)f(a).$$



5.2.2 Right rectangle approximation

In same way, if we draw a rectangle which its width is $b-a$ and its height is given by the height of the function f at the right hand point b: $f(b)$, we get

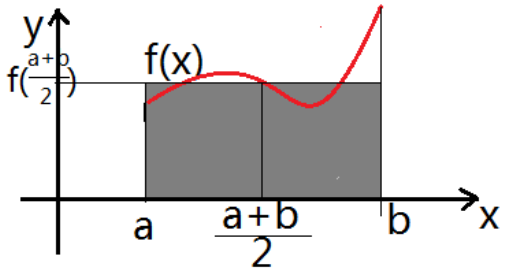
$$I := \int_a^b f(x)dx \approx I_r := (b-a)f(b).$$



5.2.3 Midpoint rectangle approximation

Similarly, if we draw a rectangle which its width is $b-a$ and its height is given by the height of the function f at the midpoint $\frac{a+b}{2}$: $f(\frac{a+b}{2})$, we get

$$I := \int_a^b f(x)dx \approx I_m := (b-a)f\left(\frac{a+b}{2}\right).$$



Remark 5.1. Generally, the error in midpoint rule

$$|E_m| = |I - I_m| \leq \frac{(b-a)^3}{24} \max_{x\in[a,b]} |f''(x)|, \quad f \in \mathcal{C}^2([a,b],\mathbb{R})$$

is smaller than the error in left rectangle rule and in right rectangle rule

$$|E_l| = |I - I_l| = |E_r| = |I - I_r| \leq \frac{(b-a)^2}{2} \max_{x\in[a,b]} |f'(x)|, \quad f \in \mathcal{C}^1([a,b],\mathbb{R}).$$

Accordingly, the midpoint approximation is better than the left rectangle approximation and the right rectangle approximation. Hence in our course, we shall focus on midpoint rule only.

5.2.4 Best approximation by midpoint rule

To get a better approximation (a smaller error $|E_m| = |I - I_m|$), we start by dividing the interval $[a, b]$ on n subinterval with same length $h = \frac{b-a}{n}$:

$$\begin{aligned}[a,b] &= [a, a+h] \cup [a+h, a+2h] \cup [a+2h, a+3h] \cup \cdots \cup [b-h, b] \\ &= [x_0, x_1] \cup [x_1, x_2] \cup [x_2, x_3] \cup \cdots \cup [x_{n-1}, x_n].\end{aligned}$$

Then, we apply on each subinterval $[x_i, x_{i+1}]$ the midpoint rule (Section 5.2.3).

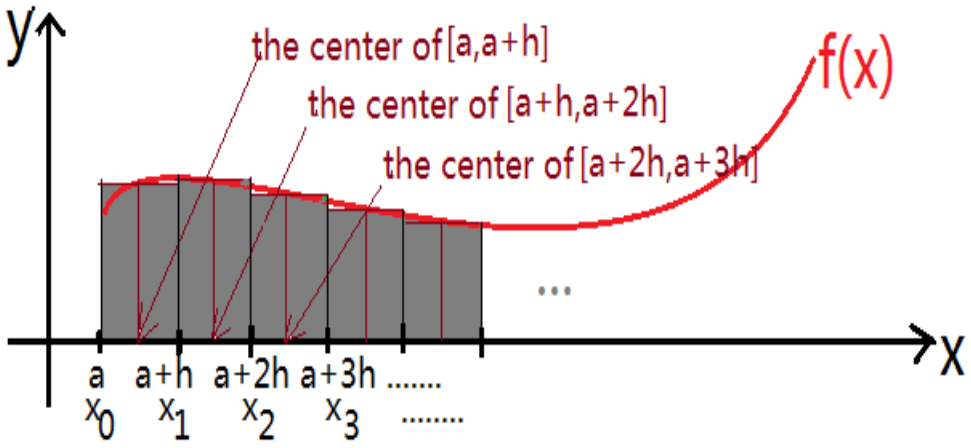


Figure 5.2: *Best approximation by midpoint rule*

In this manner we get:

$$\boxed{I := \int_a^b f(x)dx \approx I_m := \sum_{i=0}^{n-1} hf\left(\frac{x_i + x_{i+1}}{2}\right)} \tag{5.1}$$

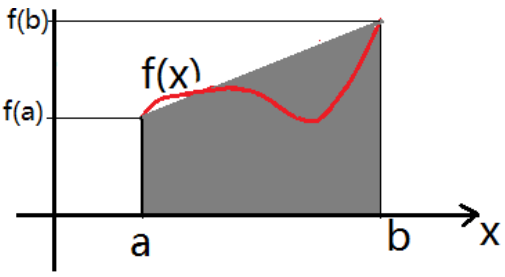
and

$$\boxed{|E_m| = |I - I_m| \leq \frac{(b-a)^3}{24\, n^2} \max_{x\in[a,b]} |f''(x)|, \quad f \in \mathcal{C}^2([a,b], \mathbb{R})}. \tag{5.2}$$

5.3 Trapezoidal rule

Here we draw the line going throw the points $(a, f(a))$ and $(b, f(b))$. Accordingly, we get a trapezoid with bases $f(a)$ and $f(b)$ and height $b-a$. Its area is considered as an approximation of I:

$$I = \int_a^b f(x)dx \approx I_t := \frac{b-a}{2}\left(f(a) + f(b)\right).$$



To get a better approximation (a smaller error $|E_t| = |I - I_t|$), we start by dividing the interval $[a, b]$ on n subinterval with same length $h = \frac{b-a}{n}$:

$$\begin{aligned}[a,b] &= [a, a+h] \cup [a+h, a+2h] \cup [a+2h, a+3h] \cup \cdots \cup [b-h, b] \\ &= [x_0, x_1] \cup [x_1, x_2] \cup [x_2, x_3] \cup \cdots \cup [x_{n-1}, x_n].\end{aligned}$$

Then, we apply on each subinterval $[x_i, x_{i+1}]$ the trapezoidal rule.

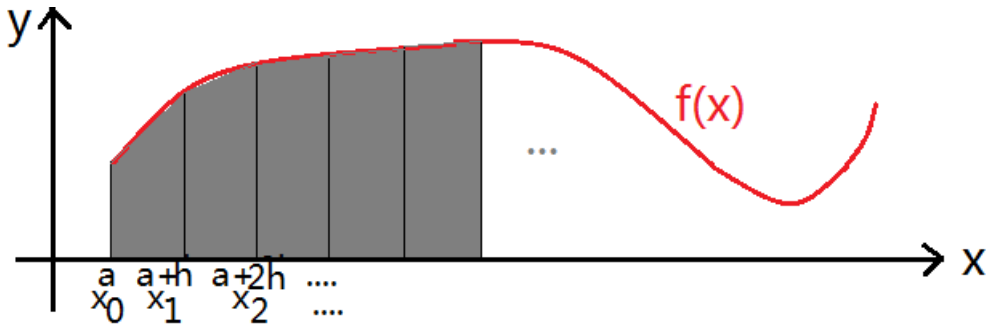


Figure 5.3: *Best approximation by trapezoidal rule*

In this manner we get:

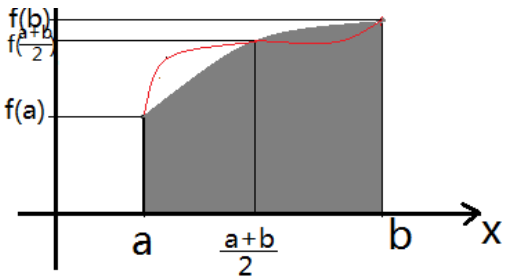
$$\boxed{I := \int_a^b f(x)dx \approx I_t := \frac{h}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right)} \tag{5.3}$$

and

$$\boxed{|E_t| = |I - I_t| \leq \frac{(b-a)^3}{12\ n^2} \max_{x\in[a,b]} |f''(x)|, \quad f \in \mathcal{C}^2([a,b], \mathbb{R})}. \tag{5.4}$$

5.4 Simpson's rule

Now we suppose that $[a, b]$ is divided on **two subintervals** of same lenght and that f is known in a, b and $\frac{a+b}{2}$. Using the polynomial interpolation (see Chapter 3), we can approximate f by a polynomial of degree 2, then we may integrate it instead of f on $[a, b]$.
As a result:



$$I = \int_a^b f(x)dx \approx I_s := \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

To get a better approximation (a smaller error $|E_s| = |I - I_s|$), we start by dividing the interval $[a, b]$ on n subinterval with same length $h = \frac{b-a}{n}$:

$$\begin{aligned}[a, b] &= [a, a + h] \cup [a + h, a + 2h] \cup [a + 2h, a + 3h] \cup \cdots \cup [b - h, b] \\ &= [x_0, x_1] \cup [x_1, x_2] \cup [x_2, x_3] \cup \cdots \cup [x_{n-1}, x_n].\end{aligned}$$

Important 3. ***We have to make sure that*** n ***(the number of subintervals) is an even number*** $(2, 4, 6, \dots)$

Then, we apply on each two subintervals $[x_i, x_{i+1}] \cup [x_{i+1}, x_{i+2}]$ the Simpson's rule.

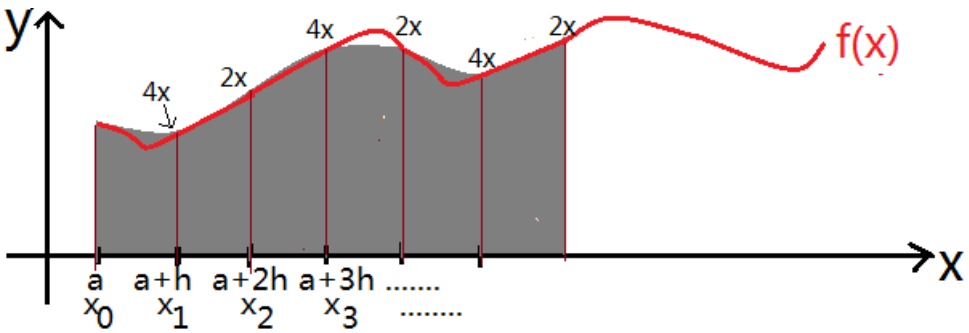


Figure 5.4: *Best approximation by Simpson's rule*

In this manner we get:

$$\boxed{I := \int_a^b f(x)dx \approx I_s := \frac{h}{3}\left(f(a) + f(b) + 2\sum_{i=1}^{\frac{n}{2}-2} f(x_{2i}) + 4\sum_{i=0}^{\frac{n}{2}-1} f(x_{2i+1})\right)} \tag{5.5}$$

and

$$\boxed{|E_s| = |I - I_s| \leq \frac{(b-a)^5}{180\ n^4} \max_{x\in[a,b]} |f^{(4)}(x)|, \quad f \in \mathcal{C}^4([a, b], \mathbb{R})}. \tag{5.6}$$

5.5 Gaussian quadrature

Suppose that the function $f(\cdot)$ is known in m noed: $x_1 = a, x_2, \cdots, x_m = b$. To approximate $I = \int_a^b f(x)dx$ with Gaussian rule, we first need to make the following changing of variables:

$$\boxed{x = \frac{b-a}{2}t + \frac{b+a}{2}.} \tag{5.7}$$

Hence

$$a = -1, \quad b = 1, \quad dx = \frac{b-a}{2}dt \quad \text{and} \quad f(x) = f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right).$$

Accordingly:

$$I = \int_a^b f(x)dx = \int_{-1}^{1} \frac{b-a}{2} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) dt = \int_{-1}^{1} g(t)dt.$$

Next, we can approximate the wanted integral by the formula

$$\boxed{I = \int_a^b f(x)dx = \int_{-1}^{1} g(t)dt \approx I_g = \sum_{i=1}^{k} \omega_i g(t_i)} \tag{5.8}$$

with $g(t) = \frac{b-a}{2} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)$, k chosen from 1 to $2m-1$ and ω_i and t_i taken from the next table:

k	i	t_i	ω_i
1	1	$t_1 = 0$	$\omega_1 = 2$
2	1	$t_1 = \frac{-1}{\sqrt{3}} = -0.5773502691896257$	$\omega_1 = 1$
	2	$t_2 = \frac{1}{\sqrt{3}} = 0.5773502691896257$	$\omega_2 = 1$
3	1	$t_1 = \frac{-\sqrt{3}}{\sqrt{5}} = -0.7745966692414834$	$\omega_1 = \frac{5}{9} = 0.5555555555555556$
	2	$t_2 = 0$	$\omega_2 = \frac{8}{9}$
	3	$t_3 = \frac{\sqrt{3}}{\sqrt{5}} = 0.7745966692414834$	$\omega_3 = \frac{5}{9} = 0.5555555555555556$
4	1	$t_1 = -0.3399810435848563$	$\omega_1 = 0.6521451548625464$
	2	$t_2 = 0.3399810435848563$	$\omega_2 = 0.6521451548625464$
	3	$t_3 = -0.8611363115940526$	$\omega_3 = 0.3478548451374476$
	4	$t_4 = 0.8611363115940526$	$\omega_4 = 0.3478548451374476$
5	...	...	...

Table 5.1: *Gaussian quadrature noeds and weights*

Important 4. ***Before proceeding with Table 5.1, we have to make sure that the integral on*** $[a, b]$ ***is transformed by*** (5.7) ***to an integral on*** $[-1, 1]$**.**

The error in the Gaussian quadrature rule is given by:

$$\boxed{|E_g| = |I - I_g| \leq \frac{(b-a)^{2m+1}(m!)^4}{[(2m)!]^3(2m+1)} \max_{x\in[a,b]} |f^{(2m)}(x)|.} \tag{5.9}$$

5.6 Order of precision

An integration method is said to be of order k if its error is null for all function of type polynomial of degree k. Hence, by observing the error formulas in Remark 5.1 and inequalities (5.2), (5.4), (5.6) and (5.9), we conclude that:

– The right rectangle approximation and the left rectangle approximation are of order 0. This is why we did not focus on them in our lecture.

- The trapezoidal rule is of order 1.
- The midpoint rule is of order 1 too but it is better than the trapezoidal one (24 in the denominator is better than 12).
- The Simpson's rule is of order 3 but it is applied only when the number of subintervals is even.
- The Gaussian quadrature formula is of order $2m-1$ with m is the number of noeds. However, it is applied only on $[-1,1]$, hence the changing of variables (5.7) is required.

5.7 Exercises

Exercise 5.1. Solve the exercise in Radian

Consider the function $f(x) = e^{(\sin(x))}$.

1. By employing the midpoint rule and the trapezoidal rule:
   - Compute numerically the integral $\int_0^5 f(x)dx$ by dividing the interval $[0,5]$ into $n=5$ equally sized intervals.
   - Compute the absolute error of approximation.
   - Estimate the minimum number of subintervals needed to approximate the above integral with an error magnitude of less than 0.01.
2. Redo the exercise using the Simpson's rule by taking $n=4$.
3. Apply the Gaussian method to approximate $\int_0^5 f(x)dx$ by taking $n=2$ then $n=3$.

Hint: 1. Use inequalities (5.2) **and** (5.4)**. 2. Use inequality** (5.6)**.**

Exercise 5.2.

1. We propose to approximate the integral

$$I = \int_2^4 \frac{1}{x-1}\, dx = \log(3) = 1.098612289$$

by integration methods of midpoint rectangular rule, trapezoidal rule and Simpson's rule for different numbers of subintervals n of $[2,4]$. Give these approximations by filling in the table below:

n	midpoint rectangular rule	trapezoidal rule	Simpson's rule
1			
2			
3			
4			

2. Estimate the minimum number of subintervals needed to approximate the integral I with an error less than 10^{-8} using the above three methods.
3. Use the Gaussian rule with $n = 2$ then with $n = 3$ to estimate I.

Hint: This exercise helps you rank integration methods according to their speed and accuracy.

Exercise 5.3.

Consider the experiment measurements:

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$f(x)$	1	0.99	0.96	0.914	0.852	0.779	0.698	0.613	0.527	0.445

1. Estimate the integral $I = \int_0^{0.9} f(x)dx$ using the trapezoidal rule.
2. Can we estimate I using the midpoint rule? Justify your answer.
3. Can we estimate I using the Simpson's rule? Justify your answer.
4. Add the point $(1, 0.368)$. Estimate $\int_0^1 f(x)dx$ using all the above mentioned methods.
5. Can we find an upper bound for the error in estimating I on the interval $[a, b] = [0, 1]$?

Hint: 5. We do not know the explicit formula of function f (hence of its derivatives), so we need to approximate it first either by using the polynomial interpolation (see Chapter 3) or by using the least squares method (see Chapter 4).

Exercise 5.4.

A rocket is launched vertically from the ground and the acceleration γ is measured during the first 80 seconds:

t(in s)	0	10	20	30	40	50	60	70	80
γ(in m/s^2)	30	31.63	33.44	35.47	37.75	40.33	43.29	46.70	50.67

Estimate the speed V of the rocket at time $t = 80s$ using the trapezoidal rule then using the Simpson's rule.

Hint: $V(t) = V(0) + \int_0^t \gamma(s)ds$**.**

Exercise 5.5.

Estimate the minimum number of subintervals needed to approximate the integral $\int_{-\pi}^{\pi} \cos(x)dx$ with an error magnitude of less than 10^{-3} using the Simpson's rule.

Hint: Use inequality (5.6) **and remember Important 3.**

Exercise 5.6.

Consider the integral $I = \int_0^1 f(x)dx$ with $f(x) = \sqrt{1 + 2x}$.

1. Calculate the exact value of I.
2. Calculate the approximated value of I using the Gaussian rule with three ordinates ($n = 3$).
3. Find an upper bound of the error in estimating I using the Gaussian rule with $n = 3$.

Hint: 3. Use inequality (5.9) **with** $\max_{x\in[0,1]} |f^{(6)}(x)| = 945$**.**

Exercise 5.7.

Let the integral $I = \int_1^2 \frac{1}{x} dx$.

1. Evaluate numerically the above integral using the trapezoidal rule with step size $h = \frac{1}{3}$.
2. Calculate the exact value of I.
3. – Why is the numerical value estimated in Question 1. great than $\ln(2)$?
   – Is this right for any step size h chosen?
   – Propose another function such that the value of estimated integral using trapezoidal rule is always great than the exact value of the integral.
4. Determine a value of h such that the Simpson's rule will approximate I with an error of no more than 10^{-4}.

Hint: 3. Check the convexity of f **either by verifying the sign of** f'' **or by drawing the graph of** f **on** $[1, 2]$**. 4. Recall that** $h = \frac{b-a}{n}$**.**

Exercise 5.8.

Let the integral $I = \displaystyle\int_0^{\pi} \frac{\sin(x)}{x} dx$. Approximate I using the trapezoidal rule with an error less than 10^{-2}.

Hint: We have $\max_{x\in[0,\pi]} \left|\left(\frac{\sin(x)}{x}\right)''\right| \leq \frac{1}{3}$ **and** $\lim_{x\to 0} \frac{\sin(x)}{x} = 1$**.**

Chapter 6
Solving differential equations: Cauchy problems

We call "a problem with initial condition", "an initial value problem" or "a Cauchy problem" every differential equation of type:

$$\begin{cases} y'(t) = f(t, y(t)), \quad \forall t \in [a, b], \\ y(a) = y_0 \end{cases} \tag{6.1}$$

where $f : [a, b] \times \mathbb{R} \to \mathbb{R}$ is a nonlinear function and $y(a) = y_0$ is a given "initial condition".

A common sufficient condition for problem (6.1) to admit a unique solution $y(\cdot)$ is that $f(t, y)$ is $k-$Lipschitz continuous with respect to the second variable, that is to say, there exists a constant $k \geq 0$ such that:

$$\forall t \in [a, b], \quad \forall (y_1, y_2) \in \mathbb{R}^2, \qquad |f(t, y_1) - f(t, y_2)| \leq k|y_1 - y_2|.$$

In this chapter, we suppose that problem (6.1) admits a unique solution $y(\cdot) \in \mathbb{R}$. However, we suppose that (for some raison) we can not determine the explicit formula of that solution. Hence, we approach it (the solution of problem (6.1)) using some numerical methods.

First, we divide the interval $[a, b]$ into n subintervals with equal-length h $\left(h = \frac{b-a}{n}\right)$. We denote

$$t_0 = a, \quad t_1 = a + h, \quad t_2 = a + 2h, \quad \cdots, \quad t_n = b.$$

According to the initial condition, we know that $y_0 := y(t_0)$ is given. The purpose in this chapter is to estimate $y_1 := y(t_1)$ from y_0, then to estimate $y_2 := y(t_2)$ from y_1, and so on. This approach is called "a one step method".

6.1 Euler's method

By integrating $y'(t) = f(t, y(t))$ from t_i to t_{i+1}, for any i from 0 to $n-1$, we get

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))dt.$$

Applying left rectangle approximation (see Chapter 5-Section 5.2.1), we get

$$y(t_{i+1}) - y(t_i) = (t_{i+1} - t_i)\, f(t_i, y(t_i))$$

simply denoted:

$$y_{i+1} - y_i = hf(t_i, y_i).$$

This is called **"Euler's explicit formula"**:

$$\boxed{\begin{cases} y_{i+1} = y_i + hf(t_i, y_i), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.2}$$

Similarly, by applying right rectangle approximation (see Chapter 5-Section 5.2.2), we get

$$y(t_{i+1}) - y(t_i) = (t_{i+1} - t_i)\, f(t_{i+1}, y(t_{i+1}))$$

simply denoted:

$$y_{i+1} - y_i = hf(t_{i+1}, y_{i+1}).$$

This is called **"Euler's implicit formula"**:

$$\boxed{\begin{cases} y_{i+1} = y_i + hf(t_{i+1}, \boxed{y_{i+1}}), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.3}$$

Remark 6.1. Observe that in Euler's implicit formula, the term y_{i+1}, which is unknown, figures in right hand side as well as left hand side of the algorithm. This requires a simplification of the formula before using it.

Example 6.1. Let

$$\begin{cases} y'(t) = t + y(t), \quad \forall t \in [0, 1], \\ y(0) = 1. \end{cases}$$

Use Euler's explicit formula and Euler's implicit formula to approximate $y(t)$ on $[0, 1]$ with $h = 0.1$.

Knowing that the exact solution of the above problem is $y(t) = 2e^t - t - 1$, compare the approximated results with the exact ones.

Solution 6.1. From the question we know that we have to fullfill the following table (y_i-exact is calculated by the formula $y(t) = 2e^t - t - 1$):

t_i	0	0.1	0.2	0.3	0.4	0.5
y_i-approximated	1					
y_i-exact	1	1.110342	1.242806	1.399718	1.583649	1.797443

t_i	0.6	0.7	0.8	0.9	1
y_i-approximated					
y_i-exact	2.044238	2.327505	2.651082	3.019206	3.436564

We start by applying Euler's explicit formula (6.2). We have:

$$\begin{aligned} y_{i+1} &= y_i + hf(t_i, y_i) \\ &= y_i + 0.1\,(t_i + y_i) \\ &= 1.1\ y_i + 0.1\ t_i. \end{aligned}$$

Hence

t_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
y_i-approximated	1	1.1	1.22	1.362	1.5282	1.72102	1.943122	2.197434

t_i	0.8	0.9	1
y_i-approximated	2.487178	2.815895	3.187485

Next, we apply Euler's implicit formula (6.3). We have:

$$\begin{aligned} y_{i+1} &= y_i + hf(t_{i+1}, y_{i+1}) \\ &= y_i + 0.1\,(t_{i+1} + y_{i+1}) \\ &= y_i + 0.1\ t_{i+1} + 0.1\ y_{i+1}. \end{aligned}$$

Hence

$$(1 - 0.1)\ y_{i+1} = y_i + 0.1\ t_{i+1}$$

which implies

$$y_{i+1} = \frac{y_i + 0.1\ t_{i+1}}{0.9}.$$

Accordingly, we get

t_i	0	0.1	0.2	0.3	0.4	0.5
y_i-approximated	1	1.122222	1.269136	1.443484	1.648316	1.887018

t_i	0.6	0.7	0.8	0.9	1
y_i-approximated	2.163353	2.481503	2.846115	3.262350	3.735944

Observe that in both approaches, the error between the exact value and the approximated value at $t = 1$ is not negligible !

Remark 6.2. Euler methods are of order one, i.e., the error is important.

6.2 Crank Nicolson's method

By integrating $y'(t) = f(t, y(t))$ from t_i to t_{i+1}, for any i from 0 to $n-1$, we get

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))dt.$$

Applying trapezoidal rule (see Chapter 5-Section 5.3), we get

$$y(t_{i+1}) - y(t_i) = \frac{t_{i+1} - t_i}{2} \left(f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right)$$

simply denoted:

$$y_{i+1} - y_i = \frac{h}{2} \left(f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right).$$

This is called **"Crank Nicolson's formula"**:

$$\boxed{\begin{cases} y_{i+1} = y_i + \dfrac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, \boxed{y_{i+1}})\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.4}$$

This algorithm is of order two (better than Euler's methods). However, it is an implicit formula. One way to get an algorithm of order two with implicit formula is the following:

6.3 Runge Kutta second order formula

$$\boxed{\begin{cases} k = y_i + hf(t_i, y_i), \\ y_{i+1} = y_i + \dfrac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, \boxed{\mathrm{k}})\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.5}$$

This is called **"Runge Kutta second order formula"** or **"modified Euler formula"**.

There are other ways to get implicit second order formulas. For example, the first-order Taylor polynomial of $y(t_{i+1})$ gives:

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + hy'(\tau), \quad \tau \in [t_i, t_{i+1}].$$

If we choose τ to be the midpoint of $[t_i, t_{i+1}]$: $\tau = \frac{t_i + t_{i+1}}{2} = t_i + \frac{h}{2}$, we get

$$y(t_{i+1}) = y(t_i) + hy'\left(t_i + \frac{h}{2}\right).$$

Observe (from (6.1)) that

$$y'\left(t_i + \frac{h}{2}\right) = f\left(t_i + \frac{h}{2}, y\left(t_i + \frac{h}{2}\right)\right).$$

Then

$$y(t_{i+1}) = y(t_i) + hf\left(t_i + \frac{h}{2}, y\left(t_i + \frac{h}{2}\right)\right).$$

To simplify this expression, we approximate $y\left(t_i + \frac{h}{2}\right)$ by the Euler's explicit method as follows:

$$y\left(t_i + \frac{h}{2}\right) = y(t_i) + \frac{h}{2} f(t_i, y(t_i)).$$

As a result, we have:

$$\boxed{\begin{cases} k = \dfrac{h}{2} f(t_i, y_i), \\ y_{i+1} = y_i + hf\left(t_i + \dfrac{h}{2}, y_i + k\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.6}$$

This is called **"Runge Kutta second order formula"** or **"Lax-Wendroff formula"** or **"Midpoint formula"**.

Another version of Runge Kutta second order formula is the following:

$$\boxed{\begin{cases} k_1 = hf(t_i, y_i), \\ k_2 = hf\left(t_{i+1}, y_i + k_1\right), \\ y_{i+1} = y_i + \dfrac{1}{2}\left(k_1 + k_2\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.7}$$

6.4 Runge Kutta fourth order formula

There exists another Runge Kutta's formula which is of order four hence it is more precise and much used in applications:

$$\boxed{\begin{cases} k_1 = hf(t_i, y_i), \\ k_2 = hf\left(t_i + \dfrac{h}{2}, y_i + \dfrac{k_1}{2}\right), \\ k_3 = hf\left(t_i + \dfrac{h}{2}, y_i + \dfrac{k_2}{2}\right), \\ k_4 = hf\left(t_i + h, y_i + k_3\right), \\ y_{i+1} = y_i + \dfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}} \tag{6.8}$$

6.5 Exercises

Exercise 6.1.

Consider the following initial value problem:

$$\begin{cases} \dot{x}(t) = t + 1 - x(t), \quad \forall t \in [0, 1], \\ x(0) = 1. \end{cases} \tag{6.9}$$

Find $x(1)$ by using Euler (explicit and implicit) methods, then Runge-Kutta formulas (of order 2 and 4). Take $h = 0.2$.

Knowing that the exact solution is $x(t) = e^{-t} + t$, compare the exact value of $x(1)$ with the approximated ones. Comment on the results.

Exercise 6.2.

Given:

$$\begin{cases} \dot{x}(t) = (t+1)^3 + \dfrac{2x(t)}{t+1}, \quad \forall t \in [0, 0.04], \\ x(0) = \dfrac{3}{2}. \end{cases}$$

1. Show that the above equation admits a uniqe solution.
2. Determine the values of x_i at points t_i by the Runge-Kutta fourth order method with $h = 0.01$.
3. Determine by the interpolation method of Newton (finite differences) the interpolating polynomial of $x(t)$. Deduce $x(0.026)$.
4. Calculate $\int_0^{0.04} x(t)dt$ by the trapezoidal method.

Hint: 1. Prove that $f(t,x)$ is Lipschitz continuous. 3. See Chapter 3. 4. See Chapter 5.

Exercise 6.3.

Let the following Cauchy problem:

$$\begin{cases} x''(t) - \dfrac{1}{4}x(t) = 0, \quad \forall t \in [0, 0.2], \\ x(0) = 1 \quad \text{and} \quad x'(0) = -0.5. \end{cases}$$

1. Suppose that $y(t) = (x(t), \dot{x}(t))^T$. Rewrite the above equation in the form:

$$\begin{cases} \dot{y}(t) = F(t, y(t)), \quad \forall t \in [0, 0.2], \\ y(0) = (1, -0.5)^T. \end{cases}$$

2. Apply the Euler (explicit and implicit) methods, then the Runge-Kutta formulas (of order 2 and 4) to find the approximate value of y for $t = 0.2$ in steps of $h = 0.1$.

Hint: From $y(t) = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$ **we have** $y'(t) = \begin{pmatrix} x'(t) \\ x''(t) \end{pmatrix} = \begin{pmatrix} x'(t) \\ \dfrac{1}{4}x(t) \end{pmatrix}$**. Hence**

$$F(t, y(t)) = F\left(t, \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}\right) = \begin{pmatrix} x'(t) \\ \dfrac{1}{4}x(t) \end{pmatrix}.$$

In this manner, the Euler explicit algorithm becomes:

$$y_{i+1} = y_i + hF(t_i, y_i)$$

equivalently

$$\begin{pmatrix} x_{i+1} \\ x'_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ x'_i \end{pmatrix} + 0.1 \begin{pmatrix} x'_i \\ \dfrac{1}{4}x_i \end{pmatrix}.$$

Using the initial condition $\begin{pmatrix} x_0 \\ x'_0 \end{pmatrix} = \begin{pmatrix} 1 \\ -0.5 \end{pmatrix}$**, one can easily fullfill the following table:**

t_i	0	0.1	0.2
x_i	1		
x'_i	-0.5		

Exercise 6.4.

Let the following initial value problem:

$$\begin{cases} x''(t) = -x(t) + 2t, \quad \forall t \in [0, 0.4], \\ x(0) = 0 \quad \text{and} \quad x'(0) = -1. \end{cases}$$

Apply the Runge-Kutta second order (modified Euler) formula to find the approximate value of x for $t = 0.4$. Take $h = 0.2$.

Hint: Put $y(t) = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$ **then** $y'(t) = \begin{pmatrix} x'(t) \\ x''(t) \end{pmatrix} = \begin{pmatrix} x'(t) \\ -x(t) + 2t \end{pmatrix}$**. Hence**

$$F(t, y(t)) = F\left(t, \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}\right) = \begin{pmatrix} x'(t) \\ -x(t) + 2t \end{pmatrix}.$$

In this manner, the modified Euler's formula becomes:

$$\begin{cases} k = y_i + hF(t_i, y_i), \\ y_{i+1} = y_i + \dfrac{h}{2}\left(F(t_i, y_i) + F(t_{i+1}, k)\right), \\ t_{i+1} = t_i + h, \qquad i = 0, \cdots, n-1. \end{cases}$$

equivalently

$$\begin{cases} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} x_i \\ x_i' \end{pmatrix} + 0.2 \begin{pmatrix} x_i' \\ -x_i + 2t \end{pmatrix} \\ \begin{pmatrix} x_{i+1} \\ x_{i+1}' \end{pmatrix} = \begin{pmatrix} x_i \\ x_i' \end{pmatrix} + \dfrac{0.2}{2} \begin{pmatrix} x_i' + k_2 \\ -x_i + 2t + (-k_1 + 2t_{i+1}) \end{pmatrix} \end{cases}$$

Simplify more the above equations then make the necessary calculus.

Exercise 6.5.

Solve

$$\begin{cases} x'''(t) = 3x(t), \quad \forall t \in [0, 1], \\ x(0) = 0, \quad x'(0) = -1 \quad \text{and} \quad x''(0) = -1 \end{cases}$$

by the modified Euler's method and obtain x at $t = 0.2,\ 0.4,\ 0.6,\ 0.8$ and 1.

Hint: Put $y(t) = \begin{pmatrix} x(t) \\ x'(t) \\ x''(t) \end{pmatrix}$ **and continue like the previous exercise.**

Exercise 6.6.

Choose one of the methods you know to solve one of the following problems:

$$\begin{cases} x'(t) = t^2 + x(t), \quad \forall t \in [0, 0.3], \\ x(0) = 1, \qquad h = 0.1. \end{cases} \qquad \begin{cases} x'(t) = -x(t) + 2t, \quad \forall t \in [0, 1], \\ x(0) = 1, \qquad h = 0.2. \end{cases}$$

$$\begin{cases} x'(t) = -x(t) + e^{-t} + e^{t}, \quad \forall t \in [0, 1], \\ x(0) = \dfrac{1}{2}, \qquad h = 0.1. \end{cases} \qquad \begin{cases} x'(t) = e^{-t} - 2x(t), \quad \forall t \in [0, 1], \\ x(0) = 1, \qquad h = 0.2. \end{cases}$$

$$\begin{cases} x'(t) = \sin(t) - x(t), \quad \forall t \in [0, 1], \\ x(0) = 1, \qquad h = 0.1. \end{cases} \qquad \begin{cases} x'(t) = 36x(t) - 37e^{-t}, \quad \forall t \in [0, 1], \\ x(0) = 1, \qquad h = 0.2. \end{cases}$$

Bibliography

[1] Alaoui, M. A., Bertelle, C., Méthodes numériques appliquées cours, exercices corrigés et mise en œuvre en JAVA. 2002

[2] Chapra, S. C., & Canale, R. P., *Numerical methods for engineers* (Vol. 1221), New York: Mcgraw-hill, 2011.

[3] Elarbi, M., *Programming numerical methods in matlab*, MechTutor.com, 2018.

[4] Epperson, J. F., *An introduction to numerical methods and analysis*, John Wiley & Sons, 2021.

[5] Meftah K., Analyse numérique et programmation Math5, Méthodes numériques appliquées. Février 2015, Université de Tlemcen.