



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

ZIANE ACHOUR UNIVERSITY – DJELFA
FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CIVIL ENGINEERING



Numerical Methods

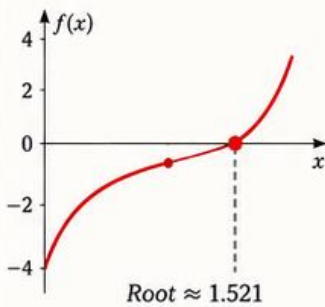
Course and solved exercises



For 2nd Year Civil Engineering Bachelor's Students

1 Solving Nonlinear Equations

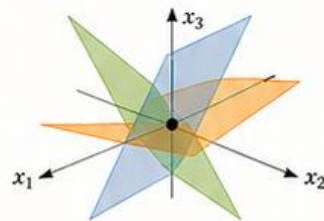
$$f(x) = x^3 - x - 2 = 0$$



2 Solution of Systems of Linear Equations

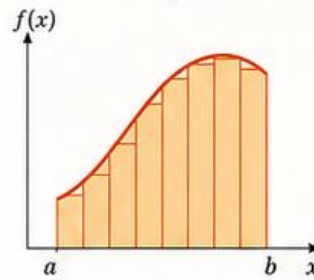
$$[A]\{x\} = \{b\}$$

$$\begin{bmatrix} 3 & 1 & -1 \\ 2 & -2 & 4 \\ -1 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix}$$



3 Numerical Integration

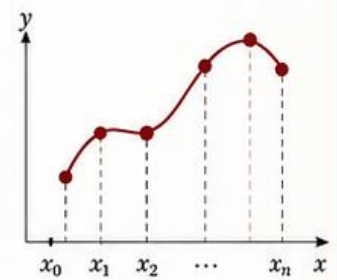
$$I = \int_a^b f(x) dx$$



Trapezoidal / Simpson Rules

4 Interpolation Methods

Find $P_n(x)$ such that
 $P_n(x_i) = f(x_i)$

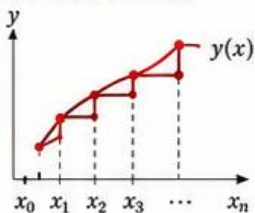


Lagrange / Newton Interpolation

5 Differential Equations

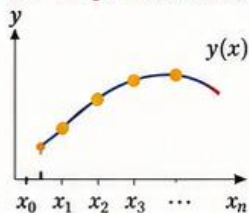
$$y' = f(x, y), y(x_0) = y_0$$

5.1 Euler's Method



$$y_{i+1} = y_i + h f(x_i, y_i)$$

5.2 Runge-Kutta Method



Higher accuracy
(e.g., RK4)

6 Eigenvalues and Eigenvectors

$$Av = \lambda v$$

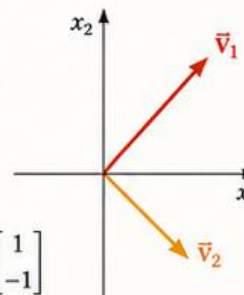
$$A = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$$

Eigenvalues:

$$\lambda_1 = 5, \lambda_2 = 2$$

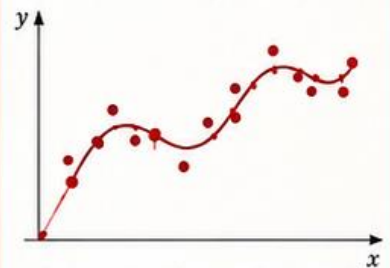
Eigenvectors:

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



7 Function Approximation

Approximate $f(x) \approx \hat{f}(x)$



Polynomials / Least Squares / Series Approximation

Prépare par :
Guesmi Mohamed



$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$
$$\int_a^b f(x) dx$$
$$y' = f(x, y)$$

2025/2026

Table of Contents

Table of Contents.....	2
General Introduction.....	5
Chapter 1: Solving Nonlinear Equations	7
1.1 Introduction.....	7
1.2 Bisection Method.....	7
1.2.1 Application condition.....	7
1.2.2 Calculation procedure	7
1.2.3 Stopping Condition	8
1.2.4 Number of iterations	8
1.2.5 Advantages and Disadvantages.....	8
1.3 Method of Successive Approximations(fixed-point).....	9
1.3.1 Definition	9
1.3.2 Calculation procedure	9
1.3.3 Stopping Condition	9
1.3.4 Advantages and disadvantages.....	10
1.4 Newton's Method	10
1.4.1 Conditions of application	10
1.4.2 Calculation procedure	11
1.4.3 Stopping Condition	11
1.4.4 Advantages and Disadvantages.....	11
1.5 Conclusion	12
Serie of Exercises 01	12
Solutions of Serie 01.....	13
Chapter 2 : Systems of Linear Equations.....	20
2.1 Introduction.....	20
2.2 Matrix Analysis	20
2.2.1 Matrix format.....	20
2.1.2 Types of matrices	21
2.1.3 Matrix operation.....	21
2.3. Direct methods.....	23
2.3.1 Gauss Method	23
2.3.2 Gauss_ Jordan Method.....	24
2.3.3 Cholesky Method	25
2.3.4 Comparison of Direct Methods	27
2.4. Iterative Methods	28
2.4.1 Convergence Conditions	28
2.4.1 Jacobi Method.....	29
2.4.2 Gauss-Seidel Method	32
2.5 Conclusion	33

Serie of Exercises 02	34
Solutions of Serie 02.....	35
Chapter 3 : Numerical Integration	43
3.1 Introduction.....	43
3.2 Principle of numerical integration	43
3.3 Rectangle Method	44
3.3 Trapezoidal Method.....	46
3.3. Simpson's 1/3 method	48
3.4 Simpson's 3/8 Rule Method ²	50
3.7 Gauss Quadrature Method	51
3.7.2 General Principle.....	51
3.7.3 Gaussian Quadrature Formula.....	52
3.8 Conclusion	53
Serie of Exercises 03	53
Solutions of Serie 03:.....	55
Chapter 4: Polynomial interpolation.....	60
3.1 Introduction.....	60
4.2 Lagrange Interpolation Method	60
4.2.1 Definition	60
4.2.2 Lagrange Basis Polynomials	60
4.2.3 Lagrange Interpolating Polynomial.....	60
4.2.4 Calculation Procedure:.....	61
4.2.5 Advantages and Limitations.....	62
4.3 Newton Interpolation Method.....	62
4.2.1 Definition	62
4.3.2 General Newton form with divided differences	62
4.3.3 Newton Forward and Backward Interpolation	65
4.4 Estimation of the interpolation error	68
4.5 Conclusion	68
Serie of Exercises 04	69
Solutions of Serie 04.....	70
Chapter 5: Differential equations.....	77
5.1 Introduction.....	77
5.2 Euler's method	77
5.2.1 Definition	77
5.2.2 Computational Procedure.....	77
5.2.3 Advantages and Limitations of the Euler Method.....	80
5.3 Runge–Kutta Method (Fourth-Order RK4)	80
5.3.1 Computational Procedure of the RK4 Method.....	80
5.3.2 Advantages and Limitations of the RK4 Method.....	82
5.4 Conclusion	82
Serie of Exercises 05	82

Solutions of Serie 05.....	83
Chapter 6: Eigenvalues and Eigenvectors.....	88
6.1 Introduction.....	88
6.2 Mathematical Definition	88
6.3 Physical Interpretation in Civil Engineering.....	89
6.4 Methods to Compute Eigenvalues	89
6.4.1 Analytical Method (Small Matrices).....	89
6.4.2 Numerical Methods for Eigenvalue Problems	91
6.5 Applications in Civil Engineering	92
6.5.1 Structural Vibration Analysis.....	92
6.5.2 Finite Element Method (FEM).....	92
6.5.3 Buckling of Columns	92
6.5.4 Seismic Engineering	92
6.6 Conclusion	92
Chapter 7: Function Approximation	94
7.1 Introduction.....	94
7.2 Taylor Series Approximation.....	94
7.2.1 Definition	94
7.2.2 Error in Taylor Approximation	95
7.2.3 Common Taylor Expansions.....	96
7.2.4 Advantages and Limitations of Taylor Series	96
7.3 Least Squares Methods	97
7.3.1 Introduction.....	97
7.3.2 Basic Idea of Least Squares	97
7.3.3 Procedure of the Least Squares Method.....	98
7.3.4 Advantages and Limitations.....	99
7.4 Conclusion	100
General conclusion	101
References and Resources	103

General Introduction

Numerical methods constitute one of the fundamental pillars of modern scientific and engineering computation. In many practical situations, engineering and physical problems cannot be solved using exact analytical methods because of the complexity of the governing equations, irregular geometries, nonlinear behavior, or large amounts of data involved. As a result, numerical techniques provide approximate but highly accurate solutions that enable engineers and scientists to analyze, simulate, and optimize real-world systems efficiently. These methods transform mathematical problems into computational procedures that can be implemented using calculators or computer algorithms, making them indispensable in modern engineering practice.

In civil engineering, numerical methods play a critical role in solving problems related to structural analysis, hydraulics, transportation systems, soil mechanics, environmental engineering, and construction management. Engineers frequently encounter systems of equations, interpolation problems, differential equations, optimization tasks, and integration processes that cannot be solved manually or by closed-form mathematical formulas. Numerical methods provide practical techniques to obtain approximate solutions with acceptable accuracy while reducing computational time and effort. Consequently, the study of numerical methods has become essential for engineering students and professionals who aim to solve increasingly complex engineering problems.

The development of computers has significantly increased the importance of numerical analysis in engineering and scientific research. Modern engineering software such as MATLAB, ANSYS, SAP2000, ETABS, and finite element analysis tools rely heavily on numerical algorithms to perform simulations and calculations. Numerical methods allow engineers to model the behavior of structures under different loading conditions, estimate fluid flow behavior, predict material performance, and evaluate system stability. Without numerical computation, many modern engineering achievements such as high-rise buildings, bridges, dams, transportation networks, and earthquake-resistant structures would be extremely difficult to design and analyze.

One of the most important applications of numerical methods is solving nonlinear equations. Many engineering systems are governed by nonlinear mathematical models where exact analytical solutions do not exist. Techniques such as the Bisection Method, Fixed-Point Iteration Method, and Newton–Raphson Method provide systematic procedures to approximate the roots of nonlinear equations. Each method has its own characteristics in terms of convergence speed, stability, and computational efficiency. Understanding these methods enables engineers to select the most appropriate technique according to the nature of the problem and the desired accuracy.

Another major field addressed by numerical methods is the solution of systems of linear equations. In civil engineering applications, systems of equations arise naturally in structural analysis, finite element methods, network analysis, and fluid mechanics. Direct methods such as Gaussian Elimination, Gauss–Jordan Elimination, and LU decomposition are commonly used to obtain exact numerical solutions, while iterative techniques such as Jacobi and Gauss–Seidel methods are suitable

for large systems where computational efficiency is important. Matrix analysis also provides the mathematical framework necessary for organizing and solving these systems effectively.

Numerical integration and interpolation are also essential topics in numerical analysis. Numerical integration techniques, including the Rectangle Rule, Trapezoidal Rule, Simpson's Rules, and Gaussian Quadrature, allow engineers to approximate definite integrals when analytical integration is impossible or difficult. These methods are particularly important in applications involving area calculations, load distributions, probability analysis, and differential equations. Interpolation methods such as Lagrange and Newton interpolation provide efficient ways to estimate unknown values between known data points, which is especially useful in experimental analysis, data fitting, and engineering measurements.

The numerical solution of differential equations represents another important application of numerical methods. Many physical and engineering phenomena such as heat transfer, fluid flow, structural vibration, and population growth are modeled using ordinary or partial differential equations. Methods such as Euler's Method and the Runge–Kutta Method enable approximate solutions to these equations when exact solutions are difficult to obtain. These techniques form the basis of many advanced simulation tools used in engineering analysis and scientific research.

In addition, numerical methods are widely used in approximation theory, eigenvalue problems, optimization, and computational modeling. Eigenvalues and eigenvectors are fundamental in vibration analysis, structural stability, seismic engineering, and finite element analysis. Approximation techniques such as Taylor series and least squares methods help engineers simplify complex functions and fit mathematical models to experimental data. These tools improve prediction accuracy and support decision-making in engineering design and analysis.

The primary objective of studying numerical methods is not only to understand the mathematical procedures behind the algorithms but also to develop the ability to analyze errors, convergence conditions, stability, and computational efficiency. Since numerical methods produce approximate solutions, it is important to evaluate the precision and reliability of the obtained results. Concepts such as truncation error, round-off error, convergence rate, and stopping criteria are therefore essential components of numerical analysis.

This course aims to provide students with a solid foundation in the principles and applications of numerical methods in engineering. Through theoretical explanations, solved examples, and practical exercises, students will learn how to formulate engineering problems mathematically and solve them using appropriate numerical techniques. The knowledge acquired from this course will help students develop analytical thinking, computational skills, and problem-solving abilities necessary for advanced engineering studies and professional practice.

Overall, numerical methods bridge the gap between mathematical theory and engineering applications. They provide engineers with powerful tools to solve complex problems efficiently and accurately in situations where analytical solutions are not feasible. As engineering systems continue to become more sophisticated, the importance of numerical methods in scientific computation and technological development will continue to grow.

Chapter 1: Solving Nonlinear Equations

1.1 Introduction

1.1 Bisection Method

1.2 Method of Successive Approximations

1.3 Newton's Method

1.4 Conclusion

1.1 Introduction

Nonlinear equations appear frequently in engineering and scientific problems, especially in civil engineering applications such as structural analysis, hydraulics, geotechnics, and material modeling. Unlike linear equations, nonlinear equations generally do not admit exact analytical solutions. As a result, numerical methods are required to approximate their solutions with acceptable accuracy. These methods are based on iterative procedures that progressively approach the exact root of the equation. The choice of a numerical method depends on factors such as convergence speed, simplicity, reliability, and the nature of the function involved. In this chapter, several classical numerical methods for solving nonlinear equations are presented and analyzed. Emphasis is placed on their principles, application conditions, procedures, and practical advantages and limitations.

1.2 Bisection Method

The term *bisection* means dividing into two equal parts. The Bisection Method, also known as the *Dichotomy Method*, is one of the simplest and most reliable numerical techniques for solving nonlinear equations. This method is based on the Intermediate Value Theorem and consists of repeatedly reducing the interval that contains the root.

1.2.1 Application condition

According to the Intermediate Value Theorem, if a function $f(x)$ is continuous on the interval $[a, b]$ and satisfies the following condition

$$f(a) \times f(b) < 0$$

then the function admits at least one root within the interval $[a, b]$. This condition ensures that the function changes sign, indicating the presence of a solution.

1.2.2 Calculation procedure

The procedure starts by computing the midpoint of the interval:

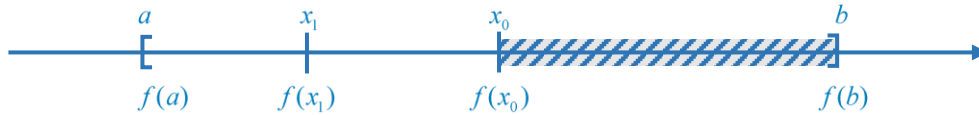
$$x_0 = \frac{a + b}{2}$$

- If $f(x_0) f(a) < 0$, the root lies in the interval $[a, x_0]$. The new interval is therefore $[a, x_0]$, and the next approximation is

$$x_1 = \frac{a + x_0}{2}$$

- Otherwise, the root lies in the interval $[x_0, b]$, and the interval $[a, x_0]$ is deleted.

This process is repeated iteratively, generating a sequence $x_0, x_1, x_2, \dots$, that converges progressively toward the exact solution.



1.2.3 Stopping Condition

The iterative process is stopped when the following condition is verified

$$|x_i - x_{i-1}| \leq \varepsilon$$

where ε represents the required precision.

1.3.4 Number of iterations

The minimum number of iterations required to achieve the desired accuracy is given by:

$$n \geq \left\lceil \frac{\ln((b-a)/\varepsilon)}{\ln(2)} \right\rceil, \quad n \in \mathbb{N},$$

Where ε is the required precision, n is the number of iterations.

Example: Solve $f(x) = x^3 - 4x - 9 = 0$ using the bisection method in the interval $[2,3]$, with a desired tolerance of $\varepsilon = 0.01$.

Solution

-Application condition: Check if $f(2) \cdot f(3) < 0$:

$$f(2) = 2^3 - 4(2) - 9 = -9, \quad f(3) = 3^3 - 4(3) - 9 = 6$$

Since $f(2) \cdot f(3) = -9(6) < 0$, there is a root in $[2, 3]$.

-iteration number: $n \geq \lceil \log((b-a)/\varepsilon)/\log(2) \rceil = \lceil \log((3-2)/0.01)/\log(2) \rceil = 6.64 \rightarrow n = 7$

-Iterative Process: We calculate the midpoint x_i of $[a, b]$ in each iteration and evaluate $f(x_i)$. The interval is updated based on the sign of $f(x_i)$.

Iteration	a	b	xi	f(a)	f(b)	f(xi)	error
0	2.000	3.000	2.500	-9.000	6.000	-3.375	-0.500
1	2.500	3.000	2.750	-3.375	6.000	0.797	-0.250
2	2.500	2.750	2.625	-3.375	0.797	-1.412	-0.125
3	2.625	2.750	2.688	-1.412	0.797	-0.339	-0.063
4	2.688	2.750	2.719	-0.339	0.797	0.221	-0.031
5	2.688	2.719	2.703	-0.339	0.221	-0.061	-0.016
6	2.703	2.719	2.711	-0.061	0.221	0.079	-0.008

-Stopping condition

The root is approximately $x \approx 2.711$ with $|x_7 - a| = 0.008 < \varepsilon = 0.01$

1.2.5 Advantages and Disadvantages

The Bisection Method guarantees convergence as long as the application condition is satisfied, regardless of the shape of the function. Its simplicity and robustness make it suitable for introductory numerical analysis. However, the method converges relatively slowly and strongly depends on the initial interval selection, which may limit its efficiency in practice.

1.3 Method of Successive Approximations(fixed-point)

1.3.1 Definition

To solve the equation $f(x) = 0$, we must transform it in the form:

$$x = g(x)$$

where the function $g(x)$ must satisfy the following two conditions over the interval $[a, b]$:

$$\left. \begin{array}{l} g(x) \in [a, b] \\ |g'(x)| < 1 \end{array} \right\} \forall x \in [a, b]$$

1.3.2 Calculation procedure

An initial approximation is chose n, commonly as

$$x_0 = \frac{(a + b)}{2}$$

The solution is then approached using the iterative relation:

$$x_{i+1} = g(x_i), \quad i = 0, 1, 2, \dots, n$$

This generates a sequence:

$$\begin{aligned} x_1 &= g(x_0) \\ x_2 &= g(x_1) \\ &\vdots \\ x_n &= g(x_{n-1}) \end{aligned}$$

which converges toward the fixed point if the convergence conditions are satisfied.

1.3.3 Stopping Condition

The iterative process is stopped when

$$|x_n - x_{n-1}| \leq \varepsilon$$

where ε denotes the prescribed accuracy.

Example: find the solution of the following equation by the method of successive approximations in the interval $[1, 2]$ with a precision of two decimal.

$$f(x) = x^3 - x - 1 = 0$$

Solution: Either the following function forms for $g(x)$:

$$\begin{aligned} 1. x &= x^3 - 1; & 2. x &= \frac{1}{x^2 - 1}; \\ 3. x &= \frac{x+1}{x^2}; & 4. x &= (x+1)^{\frac{1}{3}} \end{aligned}$$

Question: Which function $g(x)$ satisfies both conditions in the interval $[1,2]$?

Answer: after try the conditions the three first form we find that forth form verified the conditions as follow:

$$1-g(x) \in [1,2], \forall x \in [1,2] \quad *$$

$$g(1) = (1+1)^{\frac{1}{3}} = \sqrt[3]{1+1} = 1.26 \in [1,2]; \quad g(2) = (2+1)^{\frac{1}{3}} = \sqrt[3]{2+1} = 1.44 \in [1,2]$$

$$g'(x) = \frac{1}{3}(x+1)^{-2/3}; \quad g'(1) = 0.209; \quad g'(2) = 0.160$$

x	1	2
$g'(x)$	0.209	0.160

$$2- g'(x) = \frac{1}{3}(x+1)^{-2/3}$$

$|g'(1)| = 0.2 < 1, |g'(2)| = 0.16 < 1$, Then $g(x) = (x+1)^{\frac{1}{3}}$ verified the conditions of method

Calculation Procedure:

We take the initial solution $x_0 = (a+b)/2 = (1+2)/2 = 1.5$, using to fill up the table

i	x_i	$g(x_i)$	error
0	1.500	1.357	0.143
1	1.357	1.331	0.026
2	1.331	1.326	0.005

Stopping Condition: $|x_2 - x_1| = 0.05 < \epsilon = 0.01$

the solution quickly converges towards the solution $x_2 = \mathbf{1.326}$.

1.3.4 Advantages and disadvantages

This method is simple to implement and can be applied to a wide variety of nonlinear problems. However, its convergence is generally slow and highly sensitive to the choice of the function $g(x)$. An unsuitable transformation may lead to divergence.

1.4 Newton's Method

Newton's Method, also known as the Newton–Raphson Method, is a powerful and widely used numerical technique for solving nonlinear equations. It is based on linearizing the function using its first derivative and constructing successive tangent approximations to the curve. When the initial guess is sufficiently close to the exact root and the derivative does not vanish, Newton's Method exhibits very fast (quadratic) convergence, making it highly efficient in engineering applications.

1.4.1 Conditions of application

Newton's Method can be applied under the following conditions:

1. The function satisfies $f(a) \times f(b) < 0$, ensuring the existence of at least one root in the interval
2. The function $f(x)$ is **twice differentiable** on the interval $[a, b]$, and both $f'(x)$ and $f''(x)$ do not change sign on this interval.
3. The initial approximation x_0 is chosen such that $f(x_0)f''(x_0) > 0$ which helps guarantee convergence toward the root.

1.4.2 Calculation procedure

If all the above conditions are satisfied, the iterative formula of Newton's Method is given by:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

1.4.3 Stopping Condition

The iterative process is stopped when the desired accuracy is achieved, that is:

$$|x_{n+1} - x_n| \leq \varepsilon$$

where ε denotes the prescribed tolerance.

Example: $f(x) = x^3 + 4x^2 - 10$, using Newton method, find the approximate solution of $f(x)$ in the interval $[1, 2]$ and an error 0.0001.

Solution:

-Conditions of application

1. $f(1) = -5; f(2) = 14; f(1) \times f(2) < 0$

2. $f'(x) = 3x^2 + 8x; f''(x) = 6x + 8$

The two derivatives are not zero and are always positive $\forall x \in [1, 2]$

3. $x_0 = (1 + 2)/2 = 1.5; f(x_0) = f(1.5) = 2.375; f''(x_0) = f''(1.5) = 17$
 $f(x_0) \times f''(x_0) = 2.375 \times 17 > 0$

-Calculation procedure: $x_0 = (a + b)/2 = (1 + 2)/2 = 1.5$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1.5 - \frac{2.375}{18.75} = 1.3733$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1.37333 - \frac{f(1.37333)}{f'(1.37333)} = 1.36526$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 1.36523$$

i	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}	error
0	1.50000	2.37500	18.75000	1.37333	0.12667
1	1.37333	0.13435	16.64480	1.36526	0.00807
2	1.36526	0.00053	16.51392	1.36523	0.00003

So, the approximate solution is $x_2 = 1.36523$ where the error $= 0.00003 < \varepsilon = 0.0001$.

1.4.4 Advantages and Disadvantages

Newton's Method offers the significant advantage of very rapid convergence when the initial approximation is chosen sufficiently close to the exact root. In such cases, the method converges quadratically, which makes it highly efficient compared to many other numerical techniques. However, this method also presents some limitations. It requires the computation of the first derivative of the function, which may not always be simple or available. In addition, the convergence of the method strongly depends on the choice of the initial guess and the satisfaction of the applicability conditions; a poor initial estimate may lead to slow convergence or even divergence.

1.5 Conclusion

In this chapter, several classical numerical methods for solving nonlinear equations have been presented. The Bisection Method offers guaranteed convergence and simplicity but suffers from slow performance. The Method of Successive Approximations provides a flexible framework but depends strongly on the chosen transformation function. Newton's Method stands out for its rapid convergence and high efficiency, although it requires derivative evaluation and a good initial guess. Understanding the principles, conditions, and limitations of these methods allows engineers to select the most appropriate technique for a given nonlinear problem.

Serie of Exercises 01

Exercise 1

Find the root this polynomial $f(x) = x^3 - 4x - 1 = 0$ on the interval $[2,3]$ using the bisection method with accuracy $\varepsilon = 10^{-2}$.

Exercise 2

Solve the equation $f(x) = \ln(x) + x - 2 = 0$ on the interval $[1,2]$ using the bisection method with precision $\varepsilon = 10^{-3}$.

Exercise 3

Consider the nonlinear equation

$$x^3 + x - 3 = 0$$

on the interval $[1,2]$.

1. Determine whether this equation can be solved using the fixed point method on the interval $[1,2]$.
2. If applicable, find an approximate solution of the equation in the interval $[1,2]$ using the fixed point method, with an accuracy of $\varepsilon = 10^{-2}$.

Exercise 4

Given the equation on the interval $[0, 1]$

$$f(x) = e^{-x} - x = 0$$

1. Reformulate the equation as a fixed-point problem $x = g(x)$.
2. Check whether the chosen function $g(x)$ converges to the solution.
3. If the fixed-point method applicable, start from $x_0 = 0.5$ and compute an approximation of the solution with tolerance $\varepsilon = 10^{-3}$.

Exercise 5

Considering the following equation on the interval $[0, 1]$:

$$f(x) = e^x - 3x = 0$$

1. Verify that the Newton–Raphson method is applicable.
2. Use the Newton–Raphson method to find an approximate solution starting with error $\varepsilon = 10^{-3}$.

Exercise 6

Find the approximate solution of this equation $f(x) = \ln(x) + x^2 - 3 = 0$ using Newton Method in the interval $[0, 1.5]$ with precision $\varepsilon = 10^{-4}$.

Solutions of Serie 01**Solution to exercise 1**

polynomial $f(x) = x^3 - 4x - 1 = 0$, $[2,3]$, $\varepsilon = 10^{-2}$.

1.Applicability of Bisection method

$$f(2) = 8 - 8 - 1 = -1, \quad f(3) = 27 - 12 - 1 = 14$$

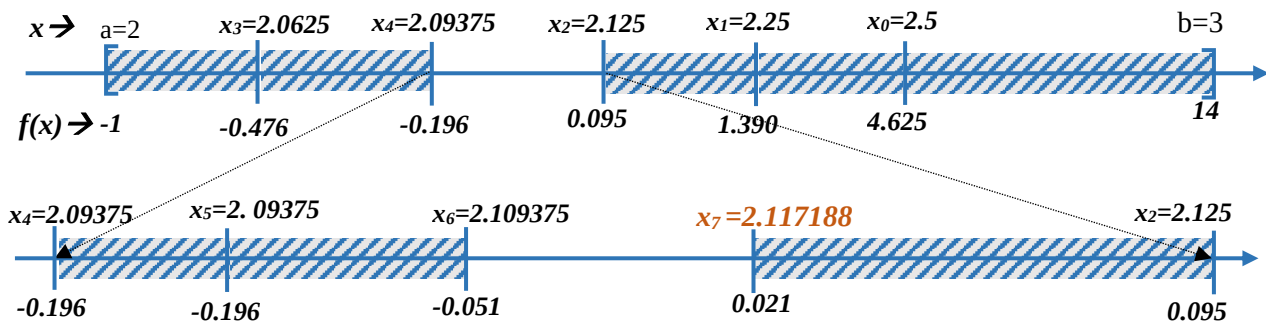
$$f(b)f(a) = -1(14) = -14 < 0 \Rightarrow \text{Root exists in } [2,3].$$

2.Iteration number

$$n \geq \left\lceil \frac{\ln((b-a)/\varepsilon)}{\ln(2)} \right\rceil = \left\lceil \frac{\ln((3-2)/0.01)}{\ln(2)} \right\rceil = 6.64 \Rightarrow n = 7$$

3.Iterative process

Start by $x_0 = \frac{a+b}{2} = \frac{3+2}{2} = 2.5$.



<i>Iter</i>	<i>a</i>	<i>b</i>	$x_i = \frac{a+b}{2}$	<i>f(a)</i>	<i>f(b)</i>	<i>f(x_i)</i>	<i>Error</i>
1	2	3	2.5	-1	14	4.625	0.5
2	2	2.5	2.25	-1	4.625	1.390625	0.25
3	2	2.25	2.125	-1	1.390625	0.095703	0.125
4	2	2.125	2.0625	-1	0.095703	-0.47632	0.0625
5	2.0625	2.125	2.09375	-0.47632	0.095703	-0.19644	0.03125
6	2.09375	2.125	2.109375	-0.19644	0.095703	-0.05191	0.015625
7	2.109375	2.125	2.117188	-0.05191	0.095703	0.021507	0.007813

4.Stopping criterion

$$|x_{n+1} - x_n| = |x_7 - x_6| = 0.007813 < \varepsilon = 0.01$$

Solution to exercise 2

$f(x) = \ln(x) + x - 2 = 0$, $[1,2]$, $\varepsilon = 10^{-3}$, Bisection method

1.Applicability of Bisection method

$$f(1) = \ln(1) + 1 - 2 = -1, \quad f(2) = \ln(2) + 2 - 2 = 0.693$$

$$f(b)f(a) = -1(0.693) = -0.693 < 0 \Rightarrow \text{Root exists in } [1,2].$$

2.Iteration number

$$n \geq \left\lceil \frac{\ln((b-a)/\varepsilon)}{\ln(2)} \right\rceil = \left\lceil \frac{\ln((2-1)/0.001)}{\ln(2)} \right\rceil = 9.96 \Rightarrow n = 10$$

3. Iterative process

Start by $x_0 = \frac{a+b}{2} = \frac{2+1}{2} = 1.5$

<i>Iter</i>	<i>a</i>	<i>b</i>	$x_i = \frac{a+b}{2}$	<i>f(a)</i>	<i>f(b)</i>	<i>f(x_i)</i>	<i>Error</i>
1	1.00000	2.00000	1.500000	-1.00000	0.69315	-0.09453	0.50000
2	1.50000	2.00000	1.750000	-0.09453	0.69315	0.30962	0.25000
3	1.50000	1.75000	1.625000	-0.09453	0.30962	0.11051	0.12500
4	1.50000	1.62500	1.562500	-0.09453	0.11051	0.00879	0.06250
5	1.50000	1.56250	1.531250	-0.09453	0.00879	-0.04267	0.03125
6	1.53125	1.56250	1.546875	-0.04267	0.00879	-0.01689	0.01562
7	1.54688	1.56250	1.554688	-0.01689	0.00879	-0.00404	0.00781
8	1.55469	1.56250	1.558594	-0.00404	0.00879	0.00238	0.00391
9	1.55469	1.55859	1.556641	-0.00404	0.00238	-0.00083	0.00195
10	1.55664	1.55859	1.557617	-0.00083	0.00238	0.00077	0.00098

4. Stopping criterion

$$|x_{n+1} - x_n| = |x_{10} - x_9| = 0.00098 < \varepsilon = 0.001$$

Solution to exercise 3

We consider the equation $x^3 + x - 3 = 0$ on $[1,2]$

1. Applicability of the fixed point method

We rewrite the equation in the form $x = g(x)$.

A convenient choice is:

$$x = \sqrt[3]{3-x} \Rightarrow g(x) = \sqrt[3]{3-x}$$

a) Condition 1: Stability of the interval

For $x \in [1,2]$,

We compute $g(1)$ and $g(2)$

$$g(1) = 1.259 \in [1,2], g(2) = 1 \in [1,2],$$

And we check that $g'(x)$ didn't change its signal on the interval $[1,2]$

$$g'(x) = -\frac{1}{3(3-x)^{2/3}}$$

So it obvious the g' remain negative mean $g(x)$ is strictly decreasing on $[1,2]$.

Thus

$$g(x) \in [1,2] \quad \forall x \in [a, b]$$

b) Condition 2: Contraction condition

We compute the derivative:

$$g'(x) = -\frac{1}{3(3-x)^{2/3}}$$

We compute $|g'(1)|$ and $|g'(2)|$

$$|g'(1)| = 0.209 < 1, \quad |g'(2)| = 0.333 < 1$$

And check the second derivative

$$|g''(x)| = \frac{2}{9(3-x)^{5/3}}$$

the $|g''|$ remain positive and $g'(x)$ is strictly increasing on $[1,2]$

the function g is a contraction on $[1,2]$

$$|g'(x)| < 1 \quad \forall x \in [1,2]$$

Therefore, the fixed point method converges to a unique solution in $[1,2]$.

2. Numerical solution by fixed point iteration

We use the iteration formula:

$$x_{n+1} = g(x_n) = \sqrt[3]{3-x_n}$$

with initial guess: $x_0 = 1.5$, and tolerance $\varepsilon = 10^{-2}$.

Iteration	x_n	x_{n+1}	Error
0	1.5000	1.1447	0.3553
1	1.1447	1.2311	0.0864
2	1.2311	1.2090	0.0221
3	1.2090	1.2149	0.0059

Stopping criterion:

$$|x_3 - x_2| = 0.0059 < \varepsilon = 0.01$$

Approximate solution: $x_3 \approx \mathbf{1.2149}$

At 24 iteration the method reach nearly the exact solution $x \approx \mathbf{1.21341}$ with zero error

Solution to exercise 4

$$f(x) = e^{-x} - x = 0 \quad x_0 = 0.5 \quad \varepsilon = 10^{-3}. \quad \text{Fixed-point method}$$

1. Function of fixed point problem

$$g(x) = x = e^{-x}$$

2. Convergence Check

a) Condition 1: Stability of the interval

For $x \in [1,2]$, We compute $g(1)$ and $g(2)$

$$g(0) = e^0 = 1 \in [0,1], \quad g(1) = e^{-1} = 0.367 \in [0,1]$$

And we check that $g'(x)$ didn't change its signal on the interval $[0,1]$

$$g'(x) = -e^{-x}$$

Since $e^{-x} > 0$, so it obvious the g' remain negative for all $x \in [0,1]$, which mean $g(x)$ is strictly decreasing on $[0,1]$.

Thus

$$g(x) \in [0,1] \quad \forall x \in [0,1]$$

b) Condition2: Contraction condition

We compute the derivative:

$$g'(x) = -e^{-x}$$

We compute $|g'(0)|$ and $|g'(1)|$

$$|g'(0)| = 1 \leq 1, \quad |g'(1)| = 0.367 \leq 1$$

And check the second derivative

$$|g''(x)| = e^{-x}$$

the $|g''|$ remain positive and $g'(x)$ is strictly increasing on $[0,1]$

the function g is a contraction on $[0,1]$

$$|g'(x)| < 1 \quad \forall x \in [a, b]$$

Therefore, the fixed point method converges to a unique solution in $[0,1]$.

3. Numerical solution by fixed point iteration

We use the iteration formula:

$$x_{n+1} = g(x_n) = e^{-x_n}$$

with initial guess: $x_0 = 0.5$ and tolerance $\varepsilon = 10^{-3}$

Iteration	x_n	x_{n+1}	Error
1	0.5	0.6065	0.1065
2	0.6065	0.5452	0.0613
3	0.5452	0.5797	0.0345
4	0.5797	0.5601	0.0196
5	0.5601	0.5712	0.0111
6	0.5712	0.5649	0.0063
7	0.5649	0.5684	0.0036
8	0.5684	0.5664	0.0020
9	0.5664	0.5676	0.0012
10	0.5676	0.5669	0.0007

Stopping criterion:

$$|x_{10} - x_9| = 0.0007 < \varepsilon = 0.001$$

Approximate solution: **$x_{10} \approx 0.5669$**

$$f(x_{10}) \approx 0.00038 \rightarrow 0$$

Solution to exercise 5

$$f(x) = e^x - 3x = 0, \quad x \in [0,1], \quad \varepsilon = 10^{-3}$$

1. Verify applicability of Newton–Raphson

First condition :

$$f(0) = 1 - 0 = 1, \quad f(1) = e^1 - 3 \approx -0.2817$$

$$f(0)f(1) = -0.2817 < 0$$

Second condition :

$$f'(x) = e^x - 3, \quad f''(x) = e^x$$

The derivatives exist and are continuous on $[0,1]$

1. Sign of $f''(x)$ on $[0,1]$

The exponential function satisfies: $e^x > 0 \quad \forall x \in R$. Therefore, for all $x \in (0,1) : f''(x) = e^x > 0$.

Hence $f''(x)$ is strictly positive on $(0,1)$.

2. Sign of $f'(x)$ on $[0,1]$

The exponential function is strictly increasing, and:

$$e^0 = 1, \quad e^1 = e \approx 2.718.$$

Thus, for any $x \in (0,1)$,

$$0 < x < 1 \\ e^0 < e^x < e^1 = 2.71 < 3 \quad \rightarrow \quad 1 < e^x < e = 2.71 < 3.$$

Subtracting 3 from all sides:

$$-2 < e^x - 3 < 0$$

Hence:

$$f'(x) = e^x - 3 < 0 \quad \forall x \in (0,1).$$

Therefore,

$$f'(x) \text{ is strictly negative on } (0,1).$$

third condition

Choose $x_0 = \frac{0+1}{2} = 0.5$

$$f(x_0) = e^{0.5} - 1.5 = 0.1487, \quad f''(x_0) = e^{0.5} = 1.64 \\ f(x_0) f''(x_0) > 0$$

Newton–Raphson method is applicable.

2. Calculation procedure

Newton formula:

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

$$x_1 = 0.5 - \frac{0.1487}{-1.3513} = 0.6100$$

$$x_2 = 0.6100 - \frac{0.0100}{-1.1600} = 0.618$$

$$x_3 = 0.6186 - \frac{0.0002}{-1.1440} = 0.6188$$

i	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}	error
1	0.5000	0.1487	-1.3513	0.6100	0.1100
2	0.6100	0.0100	-1.1600	0.6186	0.0086
3	0.6186	0.0002	-1.1440	0.6188	0.0002

Stopping

criterion:

$$|x_4 - x_3| = 0.0002 < \varepsilon = 0.001$$

Approximate solution: $x_4 \approx 0.6188$

Solution to exercise 6

$$f(x) = \ln(x) + x^2 - 3, \quad x \in [1,2], \quad \varepsilon = 10^{-4}$$

Conditions of application

First condition :

$$f(1) = \ln(1) + 1 - 3 = -2, \quad f(2) = \ln(2) + 4 - 3 \approx 1.693$$

$$f(1)f(2) < 0$$

Second condition :

$$f'(x) = \frac{1}{x} + 2x, \quad f''(x) = -\frac{1}{x^2} + 2$$

The derivatives exist and are continuous on $[1, 2]$

1. Sign of $f'(x)$ on $[1,2]$

For all $x \in [1,2]$: $x > 0$, hence $\frac{1}{x} > 0$ and $2x > 0$. Therefore: $f'(x) = \frac{1}{x} + 2x > 0 \quad \forall x \in [1,2]$.

Thus, $f'(x)$ is strictly positive on $[1,2]$.

2. Sign of $f''(x)$ on $[1,2]$

We estimate the term $\frac{1}{x^2}$ on $[1,2]$:

$$1 \leq x \leq 2 \quad \Rightarrow \quad \frac{1}{4} \leq \frac{1}{x^2} \leq 1.$$

Multiplying by -1:

$$-1 \leq -\frac{1}{x^2} \leq -\frac{1}{4}.$$

Adding 2:

$$1 \leq 2 - \frac{1}{x^2} \leq \frac{7}{4}.$$

Thus:

$$f''(x) = 2 - \frac{1}{x^2} > 0 \quad \forall x \in [1,2].$$

Hence,

$$f''(x) \text{ is strictly positive on } [1,2].$$

Third condition

Choose $x_0 = \frac{1+2}{2} = 1.5$

$$f(x_0) = \ln(1.5) + 2.25 - 3 \approx -0.3445; \quad f''(x_0) = -\frac{1}{(1.5)^2} + 2 \approx 1.556$$

$$f(x_0)f''(x_0) < 0$$

Newton–Raphson method is applicable.

2. Calculation procedure

Newton formula:

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

$$x_1 = 1.5 - \frac{-0.3445}{\frac{1}{1.5} + 2(1.5)} = 1.5933$$

$$x_2 = 1.5933 - \frac{0.0036}{3.8145} = 1.5924$$

$$x_3 = 1.5924 - \frac{0.00002}{3.8133} = 1.5924$$

Iteration table

i	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}	$error$
1	1.50000	-0.34453	3.66667	1.59396	0.09396
2	1.59396	0.00695	3.81529	1.59214	0.00182
3	1.59214	0.00000	3.81237	1.59214	0.00000

Stopping criterion:

$$|x_4 - x_3| = 0.00000 < \varepsilon = 0.0001$$

Approximate solution: $x_4 \approx 1.59214$

Chapter 2 : Systems of Linear Equations

2.1 Introduction

2.2 Matrix Analysis.

2.3 Direct Methods

2.3.1 Gaussian Elimination

2.3.2 Jordan Method

2.3.3 Cholesky Method

2.4 Iterative Methods

2.4.1 Jacobi

2.4.2 Gauss et Seidel

2.5 Conclusion

2.1 Introduction

Systems of linear equations are widely used in engineering, applied mathematics, and physics. In civil engineering, they appear frequently in structural analysis, numerical modeling, fluid flow, and geotechnical problems. Engineers rely on these systems to compute important quantities such as displacements, internal forces, stresses, reactions, and flow variables.

To handle large and complex systems efficiently, linear equations are often expressed using matrices and vectors. This representation makes it easier to organize the data and apply systematic solution techniques. As engineering problems increase in size and complexity, manual solutions become impractical, and numerical methods are required.

This chapter introduces the basic concepts of matrix analysis and presents common techniques used to solve systems of linear equations. Both direct and iterative methods are discussed, with examples to show how each method works in practice. Understanding these methods helps engineering students choose the appropriate approach for accurate and efficient problem solving in real applications.

2.2 Matrix Analysis

2.2.1 Matrix format

In practice, engineers frequently solve systems of equations to determine **displacements, deformations, internal forces, stresses, reactions, and flow variables**. These systems can be written in the general algebraic form:

[illegible]

This system can be written compactly in matrix form as:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

The goal is to find the vector $\{x\}$

2.1.2 Types of matrices

A **matrix** is a rectangular array of numbers arranged in **n rows and m columns**, denoted by $A_{n \times m}$. The most known types of matrices in engineering are :

- **Row matrix (Row vector):** A matrix with a single row ($1 \times m$).
- **Column matrix (Column vector):** A matrix with a single column ($n \times 1$).
- **Square matrix:** A matrix where $n = m$.
- **Diagonal matrix:** A square matrix where all non-diagonal elements are zero.
- **Unit (Identity) matrix:** A diagonal matrix with ones on the diagonal, denoted by **I**.
- **Triangular matrix:**

- **Upper triangular:** all elements below the diagonal are zero like this $A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 4 \\ 0 & 0 & 5 \end{bmatrix}$
- **Lower triangular:** all elements above the diagonal are zero like this $B = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 5 & 0 \\ 8 & 2 & 3 \end{bmatrix}$

- **Transpose matrix:** Obtained by interchanging rows and columns, denoted A^t .

$$A = \begin{bmatrix} -1 & 1 & 2 \\ 3 & 0 & 1 \\ 5 & -7 & 4 \end{bmatrix}; A^t = \begin{bmatrix} -1 & 3 & 5 \\ 1 & 0 & -7 \\ 2 & 1 & 4 \end{bmatrix}$$

Note: a symmetric matrix is a matrix that coincides with its transpose.

2.1.3 Matrix operation

a) Addition of two matrices

Two matrices can be added only if they have the **same dimensions**. The addition is performed element by element.

$$C = A + B \Rightarrow c_{ij} = a_{ij} + b_{ij}$$

Example

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 1 & 4 \\ 5 & 2 & 0 \end{bmatrix}; B = \begin{bmatrix} 1 & 2 & -1 \\ 3 & 5 & 1 \\ 0 & 1 & 0 \end{bmatrix}; A + B = \begin{bmatrix} 2 & 1 & 1 \\ 6 & 6 & 5 \\ 5 & 3 & 0 \end{bmatrix}$$

b) Multiplication by a scalar

Each element of the matrix is multiplied by a scalar k :

$$\lambda A = \lambda a_{ij}$$

This operation is commonly used in load scaling and normalization.

c) Product of Two Matrices

The matrix product $C = A \times B$ is defined by:

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

Condition:

The number of columns of matrix A must equal the number of rows of matrix B , Therefore, Each element of the resulting matrix C is the dot product of the i_{th} row of A and the j_{th} column of B .

Important note: Matrix multiplication is not commutative, meaning $AB \neq BA$ in general.

Example

Find the matrix $C = A \times B$ where $A = \begin{bmatrix} 1 & 2 & -1 \\ 1 & 0 & 3 \\ -2 & 1 & 4 \end{bmatrix}$; $B = \begin{bmatrix} 4 & 2 & -1 & 3 \\ -1 & 1 & 0 & 0 \\ 2 & -1 & 1 & 3 \end{bmatrix}$;

$A \times B$

$$= \begin{bmatrix} 1(4) + 2(-1) + (-1)2 & 1(2) + 2(1) + (-1)(-1) & 1(-1) + (-1)1 & 1(3) + 2(1) + (-1)3 \\ 1(4) + 0(-1) + 3(2) & 1(2) + 0(1) + 3(-1) & 1(-1) + 3(1) & 1(3) + 3(3) \\ -2(4) + 1(-1) + 4(2) & -2(2) + 1(1) + 4(-1) & -2(-1) + 3(4) & -2(3) + 1 + 4(3) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 5 & -2 & 2 \\ 10 & -1 & 2 & 12 \\ -1 & -7 & 6 & 7 \end{bmatrix}$$

d) Invertible Matrix

A square matrix **A** is said to be **invertible (non-singular)** if its determinant is non-zero.

$$\det(A) \neq 0.$$

The inverse matrix A^{-1} satisfies:

$$AA^{-1} = A^{-1}A = I$$

Properties:

$$(AB)^{-1} = B^{-1} \times A^{-1}$$

$$(A^{-1})^T = (A^T)^{-1}$$

Invertibility is essential for solving linear systems using **direct methods**.

e) Determinant of a Matrix

The **determinant** of a square matrix is a scalar value that indicates whether the matrix is invertible and represents the **area or volume scaling** of the corresponding linear transformation.

- Case $n = m = 1$: $A = [a]$; $\det(A) = a$
- Case $n = m = 2$: $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$; $\det(A) = a_{11}a_{22} - a_{21}a_{12}$
- Case $n=m>3$: The determinant is defined recursively by cofactor expansion:

$$\det(A) = \sum_{j=1}^n a_{kj} C_{ij}$$

Where $C_{ij} = (-1)^{i+j} \det(M^{ij})$ is the **minor matrix** obtained by deleting row k and column j .

Note: The determinant can be computed using **any row or any column**, and all results are identical.

Example

Compute the determinant of $A = \begin{bmatrix} -1 & 1 \\ 2 & 3 \end{bmatrix}$; $B = \begin{bmatrix} 1 & 2 & 1 \\ 3 & -1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$; $C = \begin{bmatrix} 1 & -1 & 0 \\ 2 & -1 & 3 \\ 1 & 0 & 1 \end{bmatrix}$

$$\det(A) = \begin{vmatrix} -1 & 1 \\ 2 & 3 \end{vmatrix} = -5$$

$$\det(B) = \begin{vmatrix} + & - & + \\ 1 & 2 & 1 \\ 3 & -1 & 1 \\ 2 & 1 & 0 \end{vmatrix} = +1 \begin{vmatrix} -1 & 1 \\ 1 & 0 \end{vmatrix} - 2 \begin{vmatrix} 3 & 1 \\ 2 & 0 \end{vmatrix} + 1 \begin{vmatrix} 3 & -1 \\ 2 & 1 \end{vmatrix} = 8$$

$$\det(C) = - \begin{vmatrix} + & - & 0 \\ 1 & -1 & 0 \\ 2 & -1 & 3 \\ + & 1 & 0 & 1 \end{vmatrix} = +1 \begin{vmatrix} -1 & 3 \\ 0 & 1 \end{vmatrix} - 2 \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} + 1 \begin{vmatrix} -1 & 0 \\ -1 & 3 \end{vmatrix} = -2$$

f) Inverse of a Matrix

The inverse of a matrix A can be computed using the adjugate (cofactor) method:

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$$

where the adjugate matrix is the transpose of the cofactor matrix:

$$\text{adj}(A) = (A^*)^T$$

$A^* = a_{ij}^*$ is the cofactor matrix. $a_{ij}^* = (-1)^{i+j} \det(M_{ij})$

$$A^* = \begin{bmatrix} a_{11}^* & a_{12}^* & \cdots \\ a_{21}^* & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} (-1)^{1+1} \det(M_{11}) & (-1)^{1+2} \det(M_{12}) & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

This method is primarily used for small matrices, while larger systems are solved using Gaussian elimination or iterative methods.

2.3. Direct methods

Direct methods are numerical techniques used to solve systems of linear equations by performing a finite sequence of algebraic operations. These methods lead to the exact solution of the system (apart from rounding errors due to numerical computation). In civil engineering, direct methods are widely used in structural analysis, stiffness matrix methods, finite element formulations, and material modeling when the size of the system is moderate.

The most commonly used direct methods are the Gauss (Gaussian Elimination) method and the Gauss–Jordan method.

2.3.1 Gauss Method

a) Definition

The Gauss method, also known as **Gaussian Elimination**, consists of eliminating all the elements **below the main diagonal** of the coefficient matrix. The objective is to transform the system into an **upper triangular form**, which can then be solved easily by back substitution.

Consider the following system of linear equations written in matrix form:

$$[A]\{x\} = \{b\} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

In Gauss method we need form the augmented matrix by appending the elements of vector $\{B\}$ to the right of matrix $[A]$, means:

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right]$$

Step 1(Triangulation) : convert A matrix into **triangular form**

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right] \begin{array}{l} L_2 = L_2 - c_1 L_1 \\ L_3 = L_3 - c_2 L_1 \end{array} \Rightarrow \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & 0 & a'_{33} & b'_3 \end{array} \right]$$

Step 1(Resolution by back substitution) :

$$\left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{array} \right] \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b'_2 \\ b'_3 \end{pmatrix} \Rightarrow \begin{cases} \dots\dots \\ a'_{22}x_2 + a'_{23} \frac{b'_3}{a'_{33}} = b'_2 \\ x_3 = b'_3/a'_{33} \end{cases}$$

Example

Solve the following system.

$$\begin{cases} 2x_1 + x_2 + 2x_3 = 10 \\ 6x_1 + 4x_2 = 26 \\ 8x_1 + 5x_2 + x_3 = 35 \end{cases}$$

Solution

Writing in matrix form, that is: $[A](x) = [B]$.

$$\begin{bmatrix} 2 & 1 & 2 \\ 6 & 4 & 0 \\ 8 & 5 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ 35 \end{pmatrix}$$

the augmented matrix

$$\left[\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 6 & 4 & 0 & 26 \\ 8 & 5 & 1 & 35 \end{array} \right]$$

Step 1(Triangulation) :

$$\left[\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 6 & 4 & 0 & 26 \\ 8 & 5 & 1 & 35 \end{array} \right] \begin{array}{l} L_2 - 3L_1 \\ L_3 - 4L_1 \end{array} \Rightarrow \left[\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & 1 & -6 & -4 \\ 0 & 1 & -7 & -5 \end{array} \right] \begin{array}{l} \\ \\ L_3 - L_2 \end{array} \Rightarrow \left[\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & 1 & -6 & -4 \\ 0 & 0 & -1 & -1 \end{array} \right]$$

Step 1(Resolution by back substitution) :

$$\begin{cases} -x_3 = -1 \\ x_2 - 6x_3 = -4 \\ 2x_1 + x_2 + 2x_3 = 10 \end{cases} \Rightarrow \begin{cases} x_3 = 1 \\ x_2 = 2 \\ x_1 = 3 \end{cases}$$

2.3.2 Gauss_ Jordan Method

a) Definition

The Gauss–Jordan method is an extension of the Gauss method. Instead of transforming the matrix into an upper triangular form, this method reduces the coefficient matrix directly to the **identity matrix**. Once this form is reached, the solution is obtained immediately without back substitution.

$$[A|b] \Rightarrow [I|x]$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right] \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & x_2 \\ 0 & 0 & 1 & x_3 \end{array} \right]$$

Example

Solve the following system of equations.

$$\begin{cases} x_1 - x_2 + 2x_3 = 5 \\ 3x_1 + 2x_2 + x_3 = 10 \\ 2x_1 - 3x_2 - 2x_3 = -10 \end{cases}$$

Solution

Writing in matrix form $[A](x) = [B]$:

$$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 2 & -3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ -10 \end{bmatrix}$$

Converting the augmented matrix

$$\left[\begin{array}{ccc|c} 1 & -1 & 2 & 5 \\ 3 & 2 & 1 & 10 \\ 2 & -3 & -2 & -10 \end{array} \right] \begin{array}{l} L_2 \leftarrow L_2 - 3L_1 \\ L_3 \leftarrow L_3 - 2L_1 \end{array} \Rightarrow \left[\begin{array}{ccc|c} 1 & -1 & 2 & 5 \\ 0 & 5 & -5 & -5 \\ 0 & -1 & -6 & -20 \end{array} \right] L_2 \leftarrow L_2/5 \Rightarrow$$

$$\left[\begin{array}{ccc|c} 1 & -1 & 2 & 5 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & -7 & -21 \end{array} \right] L_2 \leftarrow L_2/-7 \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 1 & 4 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 3 \end{array} \right] \begin{array}{l} L_1 \leftarrow L_1 - L_3 \\ L_2 \leftarrow L_2 + L_3 \end{array} \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right]$$

$$= \left[\begin{array}{ccc|c} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & x_2 \\ 0 & 0 & 1 & x_3 \end{array} \right]$$

Thus, the solution is:

$$\begin{cases} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{cases}$$

2.3.3 Cholesky Method

a) Definition

The Cholesky method is a **direct method** used to solve systems of linear equations. It is based on decomposing the coefficient matrix A into the product of two triangular matrices. In this formulation, the matrix A is written as:

$$[A] = [L][U]$$

where:

- $[L]$ is a lower triangular matrix with all diagonal elements equal to 1,
- $[U]$ is an upper triangular matrix.

This decomposition is also known as LU decomposition. Once the matrix A is decomposed, the system $Ax = b$ can be solved efficiently by solving two simpler triangular systems using forward and backward substitution.

This method is widely used in civil engineering applications such as structural analysis, finite element methods, and stiffness matrix formulations, where large systems of linear equations are frequently encountered.

b) General Form for a 3×3 Matrix

Let A be a 3×3 matrix :

Step 1 (LU Decomposition): In this step, all the unknown elements of matrices L and U are determined by equating the product $[L][U]$ to matrix $[A]$. This is done by comparing corresponding elements row by row and column by column.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

Step 2 (Resolution): Once the decomposition is completed, the solution of the system $Ax = b$ is obtained in two stages:

1. Forward substitution:

$$[L]\{y\} = \{b\} \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

2. Backward substitution:

$$[U]\{x\} = \{y\} \Rightarrow \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix}$$

Example

Solve the following system of equations.

$$\begin{cases} x_1 + 2x_2 + x_3 - 1 = 0 \\ 2x_1 + x_2 + 3x_3 + 2 = 0 \\ x_1 + 2x_2 + 4x_3 + 3 = 0 \end{cases}$$

Solution

Writing in matrix form, that is: $[A](x) = [B]$.

The initial equation represents a system of linear equations in matrix form:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix}$$

Step 1(LU Decomposition):

We seek matrices L and M such that:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ 0 & M_{22} & M_{23} \\ 0 & 0 & M_{33} \end{bmatrix}$$

From the first row of matrix A:

$$M_{11} = 1, \quad M_{12} = 2, \quad M_{13} = 1$$

From the second row:

$$\begin{aligned} L_{21}M_{11} &= 2 \Rightarrow L_{21} = 2 \\ L_{21}M_{12} + M_{22} &= 1 \Rightarrow M_{22} = -3 \\ L_{21}M_{13} + M_{23} &= 3 \Rightarrow M_{23} = 1 \end{aligned}$$

From the third row:

$$\begin{aligned} L_{31}M_{11} &= 1 \Rightarrow L_{31} = 1 \\ L_{31}M_{12} + L_{32}M_{22} &= 2 \Rightarrow L_{32} = 0 \\ L_{31}M_{13} + L_{32}M_{23} + M_{33} &= 4 \Rightarrow M_{33} = 3 \end{aligned}$$

Thus,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}; M = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

Step 2(Resolution):

Forward Substitution

$$[L]y = b \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix} \Rightarrow \begin{cases} y_1 = 1 \\ 2y_1 + y_2 = -2 \\ y_1 + y_3 = -3 \end{cases} \Rightarrow \begin{cases} y_1 = 1 \\ y_2 = -4 \\ y_3 = -4 \end{cases}$$

Backward Substitution

$$[M]\{x\} = \{y\} \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & -3 & 1 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -4 \\ -4 \end{bmatrix} \Rightarrow \begin{cases} 3x_3 = -4 \\ -3x_2 + x_3 = -4 \\ x_1 + 2x_2 + x_3 = 1 \end{cases} \Rightarrow \begin{cases} x_3 = -\frac{4}{3} \approx -1.3333 \\ x_2 = \frac{16}{9} \approx 1.7778 \\ x_1 = -\frac{11}{9} \approx -1.2222 \end{cases}$$

2.3.4 Comparison of Direct Methods

All three methods are **direct methods** used to solve systems of linear equations. However, they differ in procedure, efficiency, and practical applicability.

The **Gauss method** transforms the augmented matrix into an **upper triangular form** by eliminating the elements below the main diagonal. Once the triangular system is obtained, the solution is found using **backward substitution**. This method is simple and efficient for **small to medium-sized systems**. It is commonly used in hand calculations and introductory numerical analysis courses. However, numerical errors may accumulate if pivoting is not applied, especially for ill-conditioned systems.

The **Gauss–Jordan method** is an extension of the Gauss method. It eliminates both the elements **below and above** the main diagonal, transforming the coefficient matrix directly into the **identity matrix**. As a result, the solution is obtained without the need for backward substitution. Although this method is conceptually straightforward, it requires **more arithmetic operations** than the Gauss method. Therefore, it is **less efficient for large systems** and is mainly used for educational purposes or for computing the **inverse of a matrix**.

The **Cholesky method** decomposes the matrix AAA into the product of two triangular matrices. This decomposition allows the system to be solved in two simple steps using forward and backward substitution. Compared to Gauss and Gauss–Jordan methods, Cholesky is **computationally more efficient** and better suited for **large-scale engineering problems**. Once the decomposition is performed, multiple systems with different right-hand sides can be solved easily without repeating the entire process.

For clarity and ease of comparison, the fundamental principles, advantages, and limitations of these methods are summarized in the following table.

Method	Main Idea	Computational Cost	Advantages	Limitations	Typical Applications
--------	-----------	--------------------	------------	-------------	----------------------

Numerical methods

Gauss	Reduce to upper triangular matrix	Moderate	Simple, widely used	Needs pivoting, back substitution required	Small to medium systems
Gauss–Jordan	Reduce to identity matrix	High	Direct solution, no back substitution	Computationally expensive	Matrix inversion, teaching
Cholesky (LU)	Decompose into triangular matrices	Low (efficient)	Fast, stable, reusable decomposition	Requires non-singular matrix	Structural analysis, FEM

In practice, **Gauss elimination** is suitable for small systems, **Gauss–Jordan** is mainly used for theoretical purposes, while **Cholesky (LU) decomposition** is the preferred method in civil engineering and numerical modeling due to its efficiency, stability, and suitability for large systems.

2.4. Iterative Methods

Iterative methods are numerical techniques used to solve systems of linear equations by generating a sequence of approximate solutions. The solution process begins with an initial guess and proceeds through repeated calculations, called iterations, where each step produces a new approximation based on the previous one. This process continues until the solution converges and the difference between successive approximations becomes sufficiently small.

Unlike direct methods, which compute the solution in a finite number of steps, iterative methods approach the exact solution progressively. In this chapter, two classical iterative techniques are studied in detail: the Jacobi method and the Gauss–Seidel method.

2.4.1 Convergence Conditions

The convergence of iterative methods such as the Jacobi and Gauss–Seidel methods depends mainly on the properties of the coefficient matrix A . These methods do not always converge for arbitrary systems of linear equations. However, they are guaranteed to converge if at least one of the following sufficient conditions is satisfied.

1. First Condition: if the matrix A is **diagonally dominant**, then both the Jacobi method and the Gauss–Seidel method converge to the unique solution of the system. Diagonal dominance ensures that the influence of each unknown on its own equation is stronger than the combined influence of the other unknowns, which stabilizes the iterative process.

A square matrix $A = [a_{ij}]$ is said to be diagonally dominant if, for every row i , the absolute value of the diagonal element is greater than the sum of the absolute values of the other elements in the same row, that is:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|.$$

This property ensures that each equation in the system is mainly influenced by its corresponding unknown, which favors convergence of iterative methods.

Example

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & -6 & 2 \\ 1 & -2 & 7 \end{bmatrix} \quad |2| > |0| + |1| = 1, \quad |-6| = 6 > |1| + |2| = 3, \quad |7| > |1| + |-2| = 3.$$

A is not **symmetric**. diagonally dominant

2.Second condition: if the matrix A is **symmetric positive definite**, the Gauss–Seidel method is guaranteed to converge. In this case, the Jacobi method may converge, but convergence is not always assured. The Gauss–Seidel method generally benefits from faster and more stable convergence due to the immediate use of updated values during the iteration.

A matrix A is called **positive definite** if all its leading principal minors are positive. In practice, for real symmetric matrices, this property guarantees that the associated quadratic form is strictly positive for all nonzero vectors.

$$\Delta_1 > 0, \quad \Delta_2 > 0, \quad \Delta_3 > 0, \dots$$

Positive definite matrices frequently arise in engineering applications such as structural stiffness matrices and energy-based formulations, making iterative methods particularly effective.

Example

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

- $a_{ij} = a_{ji}$ A is not symmetric

$$\Delta_1 = |2| = 2 > 0,$$

$$\Delta_2 = \det \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix} = 6 > 0,$$

$$\Delta_3 = \det(A) = \begin{vmatrix} 2 & 0 & 1 \\ 1 & 3 & 1 \\ 2 & 1 & 1 \end{vmatrix} = -1 > 0. \text{ The third minor negative}$$

- A is not Positive definite

2.4.1 Jacobi Method

The Jacobi method is an iterative technique for solving a system of n linear equations with n unknowns. It is based on solving each equation for its corresponding unknown while using values from the previous iteration for the remaining unknowns.

Consider the general system:

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

Procedure

Each equation is rearranged to isolate one unknown:

$$\begin{aligned} x_1 &= \frac{1}{a_{11}} [b_1 - (a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n)] \\ x_2 &= \frac{1}{a_{22}} [b_2 - (a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n)] \\ &\vdots \end{aligned}$$

$$x_n = \frac{1}{a_{nn}} [b_n - (a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n,n-1}x_{n-1})]$$

Hence, the system of equations can be written in the following general form:

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \right]$$

This leads to the general Jacobi iteration formula:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right]$$

Here:

- k is the iteration index.
- All new values $x_i^{(k+1)}$ are computed using only values from iteration k .

Step 1 (Initialization $k = 0$)

$$x_i^{(0)} = 0 \quad (x_1^{(0)} = x_2^{(0)} = \dots = x_n^{(0)} = 0)$$

Step2 Iterative Computation

At the new approximation $k + 1$, the solution is computed using the values obtained at iteration k .

k	x_1	x_2	$x_{\dots}$	x_n
0	0	0	...	0
1	$x_1^{(1)}$	$x_2^{(1)}$	...	$x_n^{(1)}$
2	$x_1^{(2)}$	$x_2^{(2)}$	..	$x_n^{(2)}$
3	$x_1^{(3)}$	$x_2^{(3)}$	...	$x_n^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Step3 Stopping Condition

The iteration process stops when the difference between successive approximations satisfies:

$$|x_i^{(k+1)} - x_i^{(k)}| < \varepsilon \quad \text{for all variables } x_i, \quad i = 1, 2, \dots, n$$

where ε is a given tolerance.

Example

Solve the following system using the Jacobi method with a tolerance $\varepsilon=0.005$

$$\begin{cases} 8x_1 + x_2 - x_3 = 8 \\ x_1 - 7x_2 + 2x_3 = -4 \\ 2x_1 + x_2 + 9x_3 = 12 \end{cases}$$

Solution

Extract each unknown from each the equation as follow :

$$x_1 = \frac{1}{8}(8 - x_2 + x_3)$$

$$x_2 = \frac{1}{7}(4 + x_1 + 2x_3)$$

$$x_3 = \frac{1}{9}(12 - 2x_1 - x_2)$$

- **Initialization** For $k=1$: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$

- Iterations proceed as follows:

Iteration $k = 1$

$$x_1^{(2)} = 1 - 0.125(0) + 0.125(0) = 1$$

$$x_2^{(2)} = 0.571 - 0.142(0) + 0.285(0) = 0.571$$

$$x_3^{(2)} = 1.333 - 0.222(0) + 0.111(0) = 1.333$$

Iteration $k = 2$

$$x_1^{(2)} = 1 - 0.125(0.571) + 0.125(1.333) = 1.095$$

$$x_2^{(2)} = 0.571 - 0.142(1) + 0.285(1.333) = 1.092$$

$$x_3^{(2)} = 1.333 - 0.222(1) + 0.111(0.571) = 1.047$$

Iteration $k = 3$

$$x_1^{(3)} = 1 - 0.125(0.571) + 0.125(1.333) = 1.095$$

$$x_2^{(3)} = 0.571 - 0.142(1) + 0.285(1.333) = 1.092$$

$$x_3^{(3)} = 1.333 - 0.222(1) + 0.111(0.571) = 1.047$$

Lets continue and resume in following table

k	x_1	x_2	x_3
1	1.000	0.571	1.333
2	1.095	1.095	1.048
3	0.955	1.026	0.969
4	0.993	0.990	1.000
5	1.002	0.998	1.004
6	1.001	1.001	1.001
Error= $ x_i^{(6)} - x_i^{(5)} $	0.001	0.003	0.003

Verification:

The stopping condition is satisfied at iteration $k = 6$. Thus, the approximate solution is:

$$x_1^{(6)} = x_2^{(6)} = x_3^{(6)} = 1.001$$

This example demonstrates the convergence behavior of the Jacobi method and highlights its suitability for large systems where direct methods are computationally expensive.

2.4.2 Gauss-Seidel Method

Definition

The Gauss–Seidel method is an iterative technique for solving systems of linear equations and can be viewed as an improvement of the Jacobi method. Its main principle is that, within each iteration, the latest available values of the unknowns are immediately used to update the remaining unknowns. In other words, as soon as a new approximation for a variable is computed, it replaces the previous value and is used in the calculation of subsequent variables in the same iteration. This approach often leads to faster convergence compared to the Jacobi method, especially for diagonally dominant or well-conditioned systems.

Procedure

the general iteration formula is same like Jacobi:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right]$$

Step 1 (Initialization $k = 0$)

$$x_i^{(0)} = 0 \quad (x_1^{(0)} = x_2^{(0)} = \dots = x_n^{(0)} = 0)$$

Step2 Iterative Computation

It consists of **injecting all the new values $x_i^{(k+1)}$ found in the same iteration**, i.e., each new value $x_i^{(k+1)}$ replaces the old value $x_i^{(k)}$ immediately in the same iteration.

$$k \neq 0 \quad \begin{cases} x_1^{(k+1)} = 1 - 3x_2^{(k)} + x_3^{(k)} \\ x_2^{(k+1)} = 2 + 3x_1^{(k+1)} - x_3^{(k)} \\ x_3^{(k+1)} = 5 - x_2^{(k+1)} + x_1^{(k+1)} \end{cases}$$

k	x_1	x_2	$x_{\dots}$	x_n
0	0	0	...	0
1	$x_1^{(1)}$	$x_2^{(1)}$	...	$x_n^{(1)}$
2	$x_1^{(2)}$	$x_2^{(2)}$	..	$x_n^{(2)}$
3	$x_1^{(3)}$	$x_2^{(3)}$	...	$x_n^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Example

Solving the same previous system using the Gauss–Seidel method with a tolerance $\varepsilon=0.005$

$$\begin{cases} 8x_1 + x_2 - x_3 = 8 \\ x_1 - 7x_2 + 2x_3 = -4 \\ 2x_1 + x_2 + 9x_3 = 12 \end{cases}$$

Solution

Extracting iterative form

$$\begin{aligned}x_1^{(k+1)} &= 1 - 0.125x_2^{(k)} + 0.125x_3^{(k)} \\x_2^{(k+1)} &= 0.571 + 0.142x_1^{(k)} + 0.285x_3^{(k)} \\x_3^{(k+1)} &= 1.333 - 0.222x_1^{(k)} + 0.111x_2^{(k)}\end{aligned}$$

- Initialization For $k=1$: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$
- Iterations proceed as follows:

Iteration $k=1$

$$\begin{aligned}x_1 &= 1 \\x_2 &= 0.571 + 0.142(1) + 0.285(0) = 0.713 \\x_3 &= 1.333 + 0.222(1) - 0.111(0.714) = 1.032\end{aligned}$$

Iteration $k=2$

$$\begin{aligned}x_1 &= 1 - 0.125(0.714) + 0.125(1.032) = 1.039 \\x_2 &= 0.571 + 0.142(1.039) + 0.285(1.032) = 1.012 \\x_3 &= 1.333 + 0.222(1.039) - 0.111(1.012) = 0.997\end{aligned}$$

Iteration $k=3$

$$\begin{aligned}x_1 &= 1 - 0.125(1.012) + 0.125(0.997) = 0.998 \\x_2 &= 0.571 + 0.142(0.998) + 0.285(0.997) = 0.996 \\x_3 &= 1.333 - 0.222(0.998) - 0.111(0.996) = 1.001\end{aligned}$$

All the iterations resumed in this table

k	x_1	x_2	x_3
1	1	0.714	1.032
2	1.039	1.012	0.997
3	0.997	0.996	1.001
4	1.001	0.9998	1.001
Error= $ x_i^{(6)} - x_i^{(5)} $	0.0036	0.0038	0

Stopping Test

$$\begin{aligned}|1.0006 - 0.997| &= 0.0036 < \varepsilon = 0.005 \\|0.9998 - 0.996| &= 0.0038 < \varepsilon = 0.005 \\|1.001 - 1.001| &= 0 < \varepsilon = 0.005\end{aligned}$$

Remark on Convergence

Compared to the Jacobi method, the Gauss–Seidel method generally converges faster because it uses the most recently updated values of the unknowns within the same iteration. In contrast, the Jacobi method updates all variables using only values from the previous iteration. As a result, Gauss–Seidel usually requires fewer iterations to reach the same level of accuracy, especially for diagonally dominant systems. However, both methods may fail to converge if the system does not satisfy appropriate convergence conditions.

2.5 Conclusion

This chapter presented the main numerical methods used to solve systems of linear equations, which are fundamental in civil engineering applications such as structural analysis, finite element modeling, geotechnical problems, and hydraulic networks. Matrix analysis provides the mathematical basis for representing equilibrium and compatibility equations in engineering systems.

Direct methods, including Gaussian elimination, Gauss–Jordan, and Cholesky decomposition, offer reliable solutions for small to medium-sized systems. In particular, the Cholesky method is well suited for symmetric and positive-definite matrices commonly encountered in structural stiffness matrices. Iterative methods, such as Jacobi and Gauss–Seidel, are more appropriate for large and sparse systems arising from discretized engineering models, with Gauss–Seidel generally converging faster. Overall, the appropriate choice of method depends on the problem size and matrix properties, making these techniques essential tools for efficient and accurate civil engineering analysis.

Serie of Exercises 02

Exercise 1

Given the linear system:

$$\begin{cases} 2x + y + z = 5 \\ 4x + 3y + 3z = 11 \\ -2x + y + z = 1 \end{cases}$$

Write the system in matrix form $[A]\{x\} = \{b\}$.

Verify whether the matrix A is invertible.

find of the solution using the inverse matrix of A.

Exercise 2 (Gaussian elimination)

Solving those systems with Gaussian elimination

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 2x_1 + x_2 - x_3 = 3. \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases} \quad 2. \begin{cases} x_1 + 2x_2 - x_3 = -20 \\ 2x_1 - x_2 + 3x_3 = 0 \\ -x_1 + 3x_2 + 2x_3 = -2 \end{cases}$$

Exercise 3 (Gauss–Jordan method)

Solve the following systems using the Gauss–Jordan method

$$1. \begin{cases} x_1 + x_2 + x_3 = 6 \\ 2x_1 + x_2 - x_3 = 3 \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases} \quad 2. \begin{cases} x_1 + x_2 + x_3 = 4 \\ 2x_1 - x_2 + x_3 = 8 \\ -x_1 + 3x_2 + 2x_3 = 1 \end{cases}$$

Exercise 4 (Cholesky Method)

Solve the following systems using Cholesky Method

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

Exercise 5

Consider the following three matrices:

$$A_1 = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 1 \\ 0 & 1 & 3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 3 & -1 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

For each matrix :

Check whether the matrix is symmetric.

Determine whether the matrix is positive definite.

Check whether the matrix is diagonally dominant.

Exercise 6 (Jacobi Iterative Method)

Consider the linear system:

$$\begin{cases} 10x_1 - x_2 + 2x_3 = 6 \\ -x_1 + 11x_2 - x_3 = 25 \\ 2x_1 - x_2 + 10x_3 = -11 \end{cases}$$

1. Verify the convergence condition for the Jacobi method.
2. Solve the system using the Jacobi iterative method, with a precision of $\varepsilon = 0.01$

Exercise 7 (Gauss–Seidel Iterative Method)

Using the same system in exercise 6:

With Gauss–Seidel method, find the approximate solution with the same precision $\varepsilon = 0.01$.

Compare the convergence speed with the Jacobi method.

Exercise 8 (Gaussian Elimination with Pivoting)

Solve the system:

$$\begin{cases} 0.001x + y = 1 \\ x + y = 2 \end{cases}$$

Explain why pivoting is necessary.

Solve the system using Gaussian elimination with partial pivoting.

Ans: $x = 1, y = 1$

Exercise 9 (Comparison of Direct and Iterative Methods)

Given the system:

$$\begin{cases} 4x + y = 9 \\ x + 3y = 5 \end{cases}$$

1. Solve the system using Gaussian elimination.
2. Solve the system using the Jacobi method.
3. Comment on the results.

Ans : $x = 2, y = 1$

Exercise 10 (Practical Engineering Application)

In structural analysis, the nodal displacements x, y, z of a structure satisfy:

$$\begin{cases} 12x - 3y = 9 \\ -3x + 8y - 2z = -5 \\ -2y + 6z = 4 \end{cases}$$

1. Write the system in matrix form.

Solve the system using an appropriate numerical method.

Ans : $x = 1, y = 1, z = 1$

Solutions of Serie 02

Solution to Exercise 1

Given linear system

$$\begin{cases} 2x + y + z = 5 \\ 4x + 3y + 2z = 11 \\ -2x + y + z = 1 \end{cases}$$

2. Matrix form $[A]\{x\}=\{b\}$

$$\underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 2 \\ -2 & 1 & 1 \end{bmatrix}}_{[A]} \underbrace{\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}}_{\{x\}} = \underbrace{\begin{Bmatrix} 5 \\ 11 \\ 1 \end{Bmatrix}}_{\{b\}}$$

2. Verification of invertibility of matrix A

A matrix is invertible if and only if $\det(A) \neq 0$

$$\det(A) = \begin{vmatrix} 2 & 1 & 1 \\ 4 & 3 & 2 \\ -2 & 1 & 1 \end{vmatrix}$$

Expand along the first row:

$$\begin{aligned} \det(A) &= 2 \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} - 1 \begin{vmatrix} 4 & 2 \\ -2 & 1 \end{vmatrix} + 1 \begin{vmatrix} 4 & 3 \\ -2 & 1 \end{vmatrix} \\ &= 2(3 \cdot 1 - 2 \cdot 1) - 1(4 \cdot 1 - 2(-2)) + 1(4 \cdot 1 - 3(-2)) = \\ &= 2(1) - 1(4 + 4) + 1(4 + 6) = 2 - 8 + 10 = 4 \end{aligned}$$

$\boxed{\det(A) = 4 \neq 0}$ The matrix A is invertible

3. Inverse of matrix A

Cofactor matrix $A_{ij}^* = (-1)^{i+j} M_{ij}$

$$M_{11} = \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} = \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix}; \quad A_{11}^* = (-1)^{1+1} M_{11} = (1) \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} = \boxed{1};$$

$A_{12}^* = (-1)^{1+2} M_{12} = -1 \begin{vmatrix} 4 & 2 \\ -2 & 1 \end{vmatrix} = \boxed{-8}$; $A_{13}^* = (-1)^{1+3} M_{13} = 1 \begin{vmatrix} 4 & 3 \\ -2 & 1 \end{vmatrix} = \boxed{10}$; do the same the rest of the elements

$$A^* = \begin{bmatrix} 1 & -8 & 10 \\ 0 & 4 & -4 \\ -1 & 0 & 2 \end{bmatrix}$$

Adjoint (adjugate) matrix which is the transpose of matrix A^*

$$\text{adj}(A) = A^t = \begin{bmatrix} 1 & 0 & -1 \\ -8 & 4 & 0 \\ 10 & -4 & 2 \end{bmatrix}$$

Inverse matrix of A $\boxed{A^{-1} = \frac{(A^*)^t}{\det(A)}}$

$$A^{-1} = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ -8 & 4 & 0 \\ 10 & -4 & 2 \end{bmatrix}$$

3. Solve the system using the inverse

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ -8 & 4 & 0 \\ 10 & -4 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 11 \\ 1 \end{bmatrix}$$

$$x = \frac{1}{4}(1 \cdot 5 + 0 \cdot 11 - 1 \cdot 1) = \frac{1}{4}(4) = 1$$

$$y = \frac{1}{4}(-8 \cdot 5 + 4 \cdot 11 + 0 \cdot 1) = \frac{1}{4}(-40 + 44) = 1$$

$$z = \frac{1}{4}(10 \cdot 5 - 4 \cdot 11 + 2 \cdot 1) = \frac{1}{4}(50 - 44 + 2) = 2$$

$$\Rightarrow \begin{cases} x = 1 \\ y = 1 \\ z = 2 \end{cases}$$

Solution to Exercise 2

System (1)

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 2x_1 + x_2 - x_3 = 3 \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases}$$

From the Matrix form extract the augmented matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 10 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 1 & -1 & 3 \\ 3 & 2 & 1 & 10 \end{bmatrix}$$

Step 1: Triangulation

$$\begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 1 & -1 & 3 \\ 3 & 2 & 1 & 10 \end{bmatrix} \begin{matrix} L_2 \leftarrow L_2 - 2L_1 \\ L_3 \leftarrow L_3 - 3L_1 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -9 \\ 0 & -1 & -2 & -8 \end{bmatrix} \begin{matrix} \\ \\ L_3 \leftarrow L_3 - L_2 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -9 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Step 2: Back substitution

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ 1 \end{bmatrix} \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 6 \\ -x_2 - 3x_3 = -9 \\ x_3 = 1 \end{cases} \Rightarrow \begin{cases} x_1 = -1 \\ x_2 = 6 \\ x_3 = 1 \end{cases}$$

System (2)

$$\begin{cases} x_1 + 2x_2 - x_3 = -20 \\ 2x_1 - x_2 + 3x_3 = 0 \\ -x_1 + 3x_2 + 2x_3 = -2 \end{cases}$$

From the Matrix form extract the augmented matrix

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & -1 & 3 \\ -1 & 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -20 \\ 0 \\ -2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & -1 & -20 \\ 2 & -1 & 3 & 0 \\ -1 & 3 & 2 & -2 \end{bmatrix}$$

Step 1: Triangulation

$$\begin{bmatrix} 1 & 2 & -1 & -20 \\ 2 & -1 & 3 & 0 \\ -1 & 3 & 2 & -2 \end{bmatrix} \begin{matrix} \\ L_2 \leftarrow L_2 - 2L_1 \\ L_3 \leftarrow L_3 + L_1 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 2 & -1 & -20 \\ 0 & -5 & 5 & 40 \\ 0 & 5 & 1 & -22 \end{bmatrix} \begin{matrix} \\ \\ L_3 \leftarrow L_3 + L_2 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 2 & -1 & -20 \\ 0 & -5 & 5 & 40 \\ 0 & 0 & 6 & 18 \end{bmatrix}$$

Step 2: Back substitution

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & -5 & 5 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -20 \\ 40 \\ 18 \end{bmatrix} \Rightarrow \begin{cases} x_1 + 2x_2 - x_3 = -20 \\ -5x_2 + 5x_3 = 40 \\ 6x_3 = 18 \end{cases} \Rightarrow \begin{cases} x_3 = -7 \\ x_2 = -5 \\ x_1 = 3 \end{cases}$$

Solution to Exercise 3

System (1)

$$1. \begin{cases} x_1 + x_2 + x_3 = 6 \\ 2x_1 + x_2 - x_3 = 3 \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases}$$

From the Matrix form extract the augmented matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 10 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 1 & -1 & 3 \\ 3 & 2 & 1 & 10 \end{bmatrix} = [A|b]$$

Gauss–Jordan method based on transformation the augmented matrix from $[A|b]$ to $[I|x]$

$$\begin{aligned} & \begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 1 & -1 & 3 \\ 3 & 2 & 1 & 10 \end{bmatrix} \begin{matrix} L_2 \leftarrow L_2 - 2L_1 \\ L_3 \leftarrow L_3 - 3L_1 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -9 \\ 0 & -1 & -2 & -8 \end{bmatrix} \begin{matrix} L_3 \leftarrow L_3 - L_2 \\ L_1 \leftarrow L_1 - L_3 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -9 \\ 0 & 0 & 1 & 1 \end{bmatrix} \\ & \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -9 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} L_1 \leftarrow L_1 - L_3 \\ L_2 \leftarrow L_2 + 3L_3 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 5 \\ 0 & -1 & 0 & -6 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} L_1 \leftarrow L_1 + L_2 \\ L_2 \leftarrow L_2 / -1 \end{matrix} \Rightarrow \\ & \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & -6 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} L_2 \leftarrow L_2 / -1 \end{matrix} \\ & \Rightarrow \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 1 \end{bmatrix} = [I|x] \Rightarrow \begin{cases} x_1 = -1 \\ x_2 = 6 \\ x_3 = 1 \end{cases} \end{aligned}$$

System (2)

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

Augmented matrix:

$$[A | b] = \begin{bmatrix} 1 & 1 & 1 & 4 \\ 2 & -1 & 1 & 8 \\ -1 & 3 & 2 & 1 \end{bmatrix}$$

$$\begin{aligned} & \begin{bmatrix} 1 & 1 & 1 & 4 \\ 2 & -1 & 1 & 8 \\ -1 & 3 & 2 & 1 \end{bmatrix} \begin{matrix} L_2 \leftarrow L_2 - 2L_1 \\ L_3 \leftarrow L_3 + L_1 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 4 \\ 0 & -3 & -1 & 0 \\ 0 & 4 & 3 & 5 \end{bmatrix} \begin{matrix} L_3 \leftarrow L_3 + (4/3)L_2 \\ L_1 \leftarrow L_1 - L_3 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 4 \\ 0 & -3 & -1 & 0 \\ 0 & 0 & 5/3 & 5 \end{bmatrix} \begin{matrix} 3L_3/5 \\ L_2 \leftarrow L_2 / -3 \end{matrix} \\ & \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 4 \\ 0 & -3 & -1 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{matrix} L_1 \leftarrow L_1 - L_3 \\ L_2 \leftarrow L_2 + L_3 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & -3 & 0 & 3 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{matrix} L_1 \leftarrow L_1 - L_2 \\ L_2 \leftarrow L_2 / -3 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 \end{bmatrix} \\ & \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 \end{bmatrix} = [A|x] \\ & \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 \end{bmatrix} = [A|x] \Rightarrow \begin{cases} x_1 = 2 \\ x_2 = -1 \\ x_3 = 3 \end{cases} \end{aligned}$$

Solution to Exercise 4

System (1)

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

Write in matrix form

$$[A]x = [B], \quad \text{where} \quad A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ -1 & 3 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 4 \\ 8 \\ 1 \end{bmatrix}$$

Step 1(LU Decomposition):

$$A = LU \Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ -1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

From the first row of A:

$$U_{11} = 1, \quad U_{12} = 1, \quad U_{13} = 1$$

From the second row:

$$L_{21} \cdot U_{11} = 2 \Rightarrow \boxed{L_{21} = 2}$$

$$L_{21}U_{12} + U_{22} = -1 \Rightarrow 2 * 1 + U_{22} = -1 \Rightarrow \boxed{U_{22} = -3}$$

$$L_{21}U_{13} + U_{23} = 1 \Rightarrow 2 * 1 + U_{23} = 1 \Rightarrow \boxed{U_{23} = -1}$$

From the third row:

$$L_{31}U_{11} = -1 \Rightarrow \boxed{L_{31} = -1}$$

$$L_{31}U_{12} + L_{32}U_{22} = 3 \Rightarrow (-1) * 1 + L_{32} * (-3) = 3 \Rightarrow \boxed{L_{32} = -4/3}$$

$$L_{31}U_{13} + L_{32}U_{23} + U_{33} = 2 \Rightarrow (-1) * 1 + (-4/3) * (-1) + U_{33} = 2 \Rightarrow \boxed{U_{33} = 5/3}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -4/3 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -3 & -1 \\ 0 & 0 & 5/3 \end{bmatrix}$$

Step 2: Resolution

Solve $[L]\{y\} = \{B\}$ (Forward substitution)

$$Ly = B \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -4/3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 1 \end{bmatrix} \Rightarrow \begin{cases} \boxed{y_1 = 4} \\ 2 * 4 + y_2 = 8 \Rightarrow \boxed{y_2 = 0} \\ -4 - 0 + y_3 = 1 \Rightarrow \boxed{y_3 = 5} \end{cases} \Rightarrow \{y\} = \begin{pmatrix} 4 \\ 0 \\ 5 \end{pmatrix}$$

Solve $[U]\{x\} = \{y\}$ (Backward substitution)

$$[U]\{x\} = \{y\} \Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & -3 & -1 \\ 0 & 0 & 5/3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 5 \end{bmatrix} \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 4 \Rightarrow x_1 - 1 + 3 = 4 \Rightarrow \boxed{x_1 = 2} \\ -3x_2 - 1 * 3 = 0 \Rightarrow -3x_2 - 3 = 0 \Rightarrow \boxed{x_2 = -1} \\ (5/3)x_3 = 5 \Rightarrow \boxed{x_3 = 3} \end{cases} \Rightarrow \{x\} = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}$$

Solution to Exercise 5

1. Symmetry check

The matrix is symmetric since $A = A^T$

$$A_1 = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 1 \\ 0 & 1 & 3 \end{bmatrix} \quad A_1^T = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 1 \\ 0 & 1 & 3 \end{bmatrix} = A_1 \quad A_1 \text{ is symmetric}$$

2. Positive definiteness (Sylvester's criterion)

Compute leading principal minors for A_1 :

$$\Delta_1 = 4 > 0$$

$$\Delta_2 = \begin{vmatrix} 4 & 2 \\ 2 & 5 \end{vmatrix} = 20 - 4 = 16 > 0$$

$$\Delta_3 = \det(A_1) = 4(15 - 1) - 2(6 - 0) = 56 - 12 = 44 > 0$$

All leading principal minors are positive ,so A_1 is positive definite.

3. Diagonal dominance

Row by row: $|a_{ii}| \geq \sum_{j \neq i}^n |a_{ij}|$

Row 1 : $|a_{11}| = 4 > |a_{12}| + |a_{13}| = 2 + 0 = 2$

Row 3 : $|a_{22}| = 5 > |a_{21}| + |a_{23}| = 2 + 1 = 3$

Row 4 : $|a_{33}| = 3 > |a_{31}| + |a_{32}| = 0 + 1 = 1$

A_1 is strictly diagonally dominant

Short answers

A_2 : Symmetric but NOT positive definite and NOT diagonally dominant

A_3 : NOT Symmetric but NOT positive definite and diagonally dominant

Solution to Exercise 6

Matrix form :

$$A = \begin{bmatrix} 10 & -1 & 2 \\ -1 & 11 & -1 \\ 2 & -1 & 10 \end{bmatrix}, \quad \{X\} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \{B\} = \begin{bmatrix} 6 \\ 25 \\ -11 \end{bmatrix}, \quad \text{requested precision } \varepsilon = 0.01$$

1. Verify convergence condition

Sufficient condition: matrix is strictly diagonally dominant by rows :

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

Check each row:

Row 1: $|10| > |-1| + |2| = 3$

Row 2: $|11| > |-1| + |-1| = 2$

Row 3: $|10| > |2| + |-1| = 3$

Jacobi method will converge.

2. Iterative solution

Jacobi iteration contracted formula for row i:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

Jacobi iteration formula for each equation :

$$x_1^{(k+1)} = \frac{1}{10} (6 + x_2^{(k)} - 2x_3^{(k)})$$

$$x_2^{(k+1)} = \frac{1}{11} (25 + x_1^{(k)} + x_3^{(k)})$$

$$x_3^{(k+1)} = \frac{1}{10} (-11 - 2x_1^{(k)} + x_2^{(k)})$$

Iteration $k = 0$ (initial guess)

$$x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$$

Iteration $k = 1$

$$x_1^{(1)} = \frac{1}{10} (6 + 0 - 0) = 0.6$$

$$x_2^{(1)} = \frac{1}{11} (25 + 0 + 0) = 2.273$$

$$x_3^{(1)} = \frac{1}{10}(-11 - 0 + 0) = -1.1$$

Iteration $k = 2$

$$x_1^{(2)} = \frac{1}{10}(6 + 2.2727 - 2(-1.1)) = \frac{1}{10}(6 + 2.2727 + 2.2) = \frac{10.4727}{10} = 1.047$$

$$x_2^{(2)} = \frac{1}{11}(25 + 0.6 + (-1.1)) = \frac{24.5}{11} = 2.227$$

$$x_3^{(2)} = \frac{1}{10}(-11 - 2 * 0.6 + 2.2727) = \frac{-11 - 1.2 + 2.2727}{10} = \frac{-9.9273}{10} = -0.993$$

Iteration $k = 3$

$$x_1^{(3)} = \frac{1}{10}(6 + 2.227 - 2*(-0.993)) = \frac{6 + 2.227 + 1.986}{10} = \frac{10.213}{10} = 1.021$$

$$x_2^{(3)} = \frac{1}{11}(25 + 1.047 + (-0.993)) = \frac{25.054}{11} = 2.277$$

$$x_3^{(3)} = \frac{1}{10}(-11 - 2*1.047 + 2.227) = \frac{-11 - 2.094 + 2.227}{10} = \frac{-10.867}{10} = -1.087$$

The rest iterations is completed in this table

Iteration	x_1	x_2	x_3
1	0.600	2.273	-1.100
2	1.047	2.227	-0.993
3	1.021	2.278	-1.087
4	1.045	2.267	-1.077
5	1.042	2.270	-1.082

Check precision

$$\begin{aligned} \Delta &= \max(|x_1^{(5)} - x_1^{(4)}|, |x_2^{(5)} - x_2^{(4)}|, |x_3^{(5)} - x_3^{(4)}|) \\ &= \max(|1.042 - 1.045|, |2.270 - 2.267|, |-1.082 + 1.077|) \\ &= \max(0.003, 0.003, 0.005) = 0.005 < \varepsilon = 0.01 \end{aligned}$$

Final solution with precision 0.01 is **$x_1 \approx 1.042$, $x_2 \approx 2.270$, $x_3 \approx -1.082$**

Solution to exercise 7

Gauss–Seidel iteration formula for each equation :

$$x_1^{(k+1)} = \frac{1}{10}(6 + x_2^{(k)} - 2x_3^{(k)})$$

$$x_2^{(k+1)} = \frac{1}{11}(25 + x_1^{(k+1)} + x_3^{(k)})$$

$$x_3^{(k+1)} = \frac{1}{10}(-11 - 2x_1^{(k+1)} + x_2^{(k+1)})$$

Iteration $k = 0$ (initial guess)

$$x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$$

Iteration $k = 1$

$$x_1^{(1)} = \frac{1}{10}(6 + 0 - 2 * 0) = 0.6000$$

$$x_2^{(1)} = \frac{1}{11}(25 + 0.6000 + 0) = 2.3273$$

$$x_3^{(1)} = \frac{1}{10}(-11 - 2 * 0.6000 + 2.3273) = -0.9873$$

Iteration $k = 2$

$$x_1^{(2)} = \frac{1}{10}(6 + 2.3273 - 2 * (-0.9873)) = 1.0302$$

$$x_2^{(2)} = \frac{1}{11}(25 + 1.0302 + (-0.9873)) = 2.2766$$

$$x_3^{(2)} = \frac{1}{10}(-11 - 2 * 1.0302 + 2.2766) = -1.0784$$

Iteration $k = 3$

$$x_1^{(3)} = \frac{1}{10}(6 + 2.2766 - 2 * (-1.0784)) = 1.0433$$

$$x_2^{(3)} = \frac{1}{11}(25 + 1.0433 + (-1.0784)) = 2.2695$$

$$x_3^{(3)} = \frac{1}{10}(-11 - 2 * 1.0433 + 2.2695) = -1.0817$$

Iteration $k = 4$

$$x_1^{(4)} = \frac{1}{10}(6 + 2.2695 - 2 * (-1.0817)) \approx 1.0433$$

$$x_2^{(4)} = \frac{1}{11}(25 + 1.0433 + (-1.0817)) \approx 2.2692$$

$$x_3^{(4)} = \frac{1}{10}(-11 - 2 * 1.0433 + 2.2692) \approx -1.0817$$

Iteration	x_1	x_2	x_3
1	0.60000	2.32730	-0.98727
2	1.03020	2.27660	-1.07840
3	1.04330	2.26950	-1.08170
4	1.04330	2.26920	-1.08170

$$\Delta = \max(|x_1^{(4)} - x_1^{(3)}|, |x_2^{(4)} - x_2^{(3)}|, |x_3^{(4)} - x_3^{(3)}|)$$

$$= \max(0.0000, 0.0002, 0.0000) = 0.0002 < \varepsilon = 0.01$$

Chapter 3 : Numerical Integration

- 3.1 Introduction
- 3.2 Principle of numerical integration
- 3.3 Rectangle Method
- 3.4 Trapezoidal Method
- 3.5 Simpson's 1/3 Rule
- 3.6 Simpson's 3/8 Rule
- 3.7 Gauss Quadrature Method
- 3.8 Conclusion

3.1 Introduction

Numerical (or digital) integration is widely used in engineering and scientific applications when the exact analytical integration of a function is difficult or impossible. This situation often occurs when the function has a complex mathematical expression, when it is known only at discrete points obtained from experiments or measurements, or when an analytical antiderivative does not exist. In such cases, numerical integration provides an approximate value of the definite integral over a given interval

3.2 Principle of numerical integration

Numerical integration is based on approximating the original function $f(x)$ by a polynomial function $P_n(x)$ of degree n . Instead of integrating the exact function, the polynomial approximation is integrated over the interval. The integral can be written as:

$$\int_a^b f(x)dx = \left(\int_a^b P_n(x)dx \right) + E$$

Where $P_n(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$ is a polynomial of degree n , E represents the error introduced by the approximation.

The polynomial $P_n(x)$ is obtained using interpolation techniques. These techniques construct a polynomial that passes through a set of known data points. One of the most commonly used interpolation methods is the Lagrange interpolation method. By applying these interpolation formulas, a family of numerical integration methods known as Newton–Cotes formulas is derived. In this chapter, we focus on Newton–Cotes formulas corresponding to different polynomial degrees as shown in following table:

Degree n	Polynomial Approximation	Numerical Method
$n = 0$	Constant	Rectangle Method
$n = 1$	Linear	Trapezoidal Method

n = 2	Quadratic	Simpson's (1/3) Method
n = 3	Cubic	Simpson's (3/8) Method

In general, increasing the degree of the polynomial improves the accuracy of the approximation. However, higher-degree polynomials may also increase computational effort and numerical instability. Therefore, a balance between accuracy and simplicity is usually considered in practical applications.

3.3 Rectangle Method

The Rectangle Method is one of the simplest numerical integration techniques. It is used to approximate the value of a definite integral when the exact analytical solution is difficult or impossible to obtain. The method is based on approximating the area under a curve by the area of one or several rectangles.

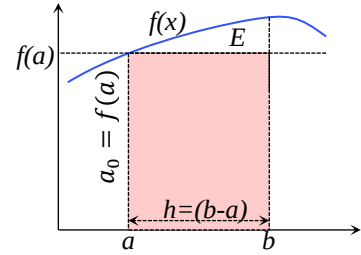
We aim to approximate the integral

$$\int_a^b f(x) dx$$

which represents the area under the curve $f(x)$ between $x = a$ and $x = b$.

a) Simple Rectangle Method

In the simplest form of the rectangle method, the function $f(x)$ is approximated by a constant polynomial of degree $n=0$. This means the function is replaced by a horizontal line, and the area is approximated using one single rectangle.



The approximation is written as:

$$\begin{aligned} I_r &= \int_a^b f(x) dx = \left(\int_a^b P_0(x) dx \right) + E = \left(\int_a^b (a_0) dx \right) + E \\ &= \left(\int_a^b f(a) dx \right) + E = (f(a)[x]_a^b) + E = h f(a) + E_r \end{aligned}$$

with $h = b - a$, and E represents the approximation error. This method is very simple but generally not very accurate.

b) Compound Rectangle Method

To improve accuracy, the interval $[a, b]$ is divided into n equal subintervals (see the following figure). Each subinterval corresponds to one rectangle.

the width of each rectangle is:

$$h = \frac{b - a}{n}$$

the height of each rectangle is $f(x_i)$, where points x_i are defined as:

$$x_i = a + i h, \quad i = 0, 1, \dots, n$$

The compound rectangle approximation is given by:

$$I_r = \int_a^b f(x) dx \approx \left(h \sum_{i=0}^{n-1} f(x_i) \right) + E_r$$

where E_r is the total approximation error.

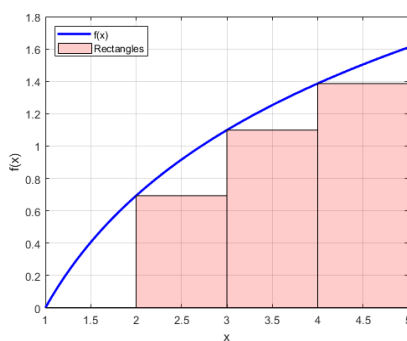
Variants of the Rectangle Method

The height of each rectangle depends on the point where the function is evaluated in each subinterval. This leads to three common variants: (see the following figure):

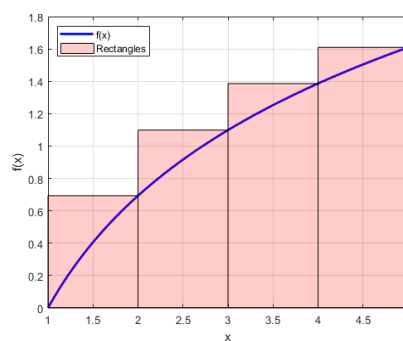
1. *Left Rectangle Method* : Height taken at the left endpoint $x_i = a + i h$, this variant is underestimates if $f(x)$ is increasing and overestimates if $f(x)$ is decreasing.

2. *Right Rectangle Method* : Height taken at the right endpoint $x_i = a + i h$, this variant is overestimates if $f(x)$ is increasing and underestimates if $f(x)$ is decreasing.

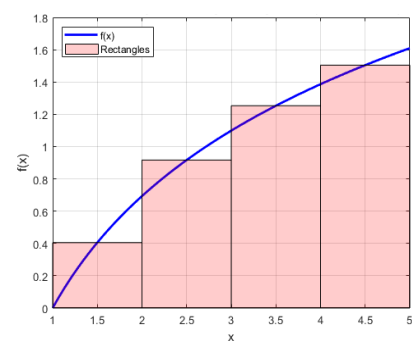
3. *Midpoint Rectangle Method* : Height taken at the middle of each subinterval $x_i = a + (i + 1/2)h$, generally more accurate than left or right methods



a) Left Rectangle Method



b) Right Rectangle Method



c) Midpoint Rectangle Method

Figure : Rectangle method variants.

Example

Approximate the integral

$$\int_0^1 x^2 dx$$

using Left Rectangle Method with $n=4$.

Solution

Step 1: Compute the step size

$$h = \frac{b - a}{n} = \frac{1 - 0}{4} = 0.25$$

Step 2: Determine the points

$$x_i = a + i h = 0.25i, \quad i = 0, 1, 2, 3$$

$$\begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \\ 0 & 0.25 & 0.5 & 0.75 \end{array}$$

Step 3: Compute the approximation

$$I_r = \int_0^1 x^2 dx \approx \left(h \sum_{i=0}^{n-1} x_i^2 \right) + E_r$$

$$I_r = 0.25[(0)^2 + (0.25)^2 + (0.5)^2 + (0.75)^2] + E_r$$

$$= 0.25(0 + 0.0625 + 0.25 + 0.5625) = 0.21875 + E_r$$

Exact value:

$$I_{exact} = \int_0^1 x^2 dx = \frac{1}{3} \approx 0.333$$

Error

$$Error = |I_r - I_{exact}| = |0.21875 - 0.333| = 0.11458$$

3.3 Trapezoidal Method

The trapezoidal method is a numerical integration technique that approximates a definite integral by replacing the function $f(x)$ with a **linear polynomial** (degree $n=1$). The area under the curve is approximated by the area of one or more **trapezoids**.

a) Simple Trapezoidal Method

In the simple trapezoidal method, the interval $[a, b]$ is approximated using one trapezoid. The linear polynomial passes through the points $(a, f(a))$ and $(b, f(b))$. The approximate value of the integral corresponds to the area that lies under the curve of the polynomial.

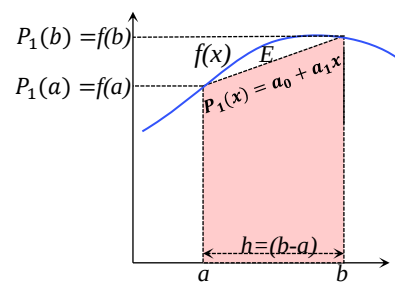
The approximation formula is:

$$\begin{aligned} I_t &= \int_a^b f(x) dx = \left(\int_a^b P_1(x) dx \right) + E \\ &= \left(\int_a^b (a_0 + a_1 x) dx \right) + E \\ &= \left[a_0 x + a_1 \frac{x^2}{2} \right]_a^b + E \\ &= \left(a_0 b + a_1 \frac{b^2}{2} \right) - \left(a_0 a + a_1 \frac{a^2}{2} \right) + E \\ &= \left(a_0(b - a) + \frac{a_1}{2}(b^2 - a^2) \right) + E \\ &= \left(a_0(b - a) + \frac{a_1}{2}(b - a)(b + a) \right) + E \\ &= \frac{(b - a)}{2} (2a_0 + a_1(b + a)) + E \\ &= \frac{(b - a)}{2} [(a_0 + a_1 a) + (a_0 + a_1 b)] + E \end{aligned}$$

The formula of Simple Trapezoidal Method becomes:

$$I_t = \frac{h}{2} [f(a) + f(b)] + E_t$$

with $h = b - a$, E denotes the error.



Example : Approximate the integral $\int_0^{\pi/2} \sin x dx$ Using one trapezoid:

$$\int_0^{\pi/2} \sin x \, dx = \frac{\frac{\pi}{2} - 0}{2} [\sin 0 + \sin \frac{\pi}{2}] = 0,78539$$

b)Compound trapezoid method:

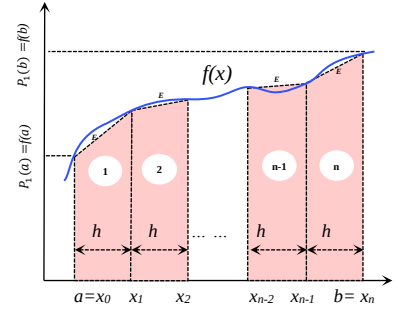
To improve the accuracy, the interval $[a, b]$ is divided into n equal subintervals. Each subinterval contributes one trapezoid. The width of each interval is denoted by

$$h = \frac{b - a}{n}$$

The x-coordinates of points are

$$x_i = a + i h = 0.25i, \quad i = 0,1,2,3$$

The compound trapezoidal approximation is:



$$I_t = \int_a^b f(x) dx = \frac{h}{2} [f(x_0) + 2(f(x_1) + f(x_2) + \dots + f(x_{n-1})) + f(x_n)] + E_t$$

Trapezoid method error

The error is given by the formula:

$$E_t \leq \frac{(b - a)^3}{12n^2} \max_{x \in [a, b]} |f''(x)|$$

This shows that the trapezoidal method has second-order accuracy, and the error decreases proportionally to $1/n^2$.

Example

Compute this $\int_0^{\pi/2} \sin x \, dx$ by the compound trapezoid method with four intervals.

Solution

$$n = 4, \quad h = \frac{\pi/2 - 0}{4} = \frac{\pi}{8}$$

$$x_i = a + i h = \frac{\pi}{8}i, \quad i = 0, \dots, 4$$

x_0	x_1	x_2	x_3	x_4
0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$

So the integral will be :

$$\begin{aligned} I_t &= \int_0^{\pi/2} \sin x \, dx = \frac{\pi}{16} \left[\sin 0 + 2 \sin \frac{\pi}{8} + 2 \sin \frac{\pi}{4} + 2 \sin \frac{3\pi}{8} + \sin \frac{\pi}{2} \right] + E \\ &= \frac{\pi}{16} [0 + 2(0.382683) + 2(0.707107) + 2(0.923880) + 1] + E = \mathbf{0.98175} \end{aligned}$$

Exact value of the integral

$$I_{exact} = \int_0^{\pi/2} \sin x \, dx = [-\cos(x)]_0^{\pi/2} = (-\cos \pi/2) - (-\cos 0) = 0 + 1 = 1$$

Method	Result	Absolute error	Relative error
Simple Trapezoidal Method (1 interval)	0,725	0.275	27.5%
Compound trapezoid method (4 intervals)	0.98175	0.01825	1.83%

3.3. Simpson's 1/3 method

Simpson's 1/3 method approximates the integrand using a quadratic polynomial (degree $n=2$) that passes through three equally spaced points: $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$.

The polynomial $P_2(x)$ satisfies:

$$(P_2(x_0) = f(x_0); \quad P_2(x_1) = f(x_1); \quad P_2(x_2) = f(x_2))$$

$$I_{S1/3} = \int_{x_0}^{x_2} f(x) dx = \left(\int_{x_0}^{x_2} P_2(x) dx \right) + E = \left(\int_{x_0}^{x_2} (a_0 + a_1 x + a_2 x^2) dx \right) + E$$

Where $2h = x_2 - x_0$

Instead of solving for a_0, a_1, a_2 explicitly, we use Lagrange interpolation:

$$P_2(x) = f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Substitute the equally spaced point $x_1 = x_0 + h$, $x_2 = x_0 + 2h$

the denominators become:

$$\begin{aligned} (x_0 - x_1)(x_0 - x_2) &= (-h)(-2h) = 2h^2 \\ (x_1 - x_0)(x_1 - x_2) &= (h)(-h) = -h^2 \\ (x_2 - x_0)(x_2 - x_1) &= (2h)(h) = 2h^2 \end{aligned}$$

Now integrate $P_2(x)$ from x_0 to x_2 :

$$\int_{x_0}^{x_2} P_2(x) dx = f(x_0) \frac{1}{2h^2} \int_{x_0}^{x_2} (x - x_1)(x - x_2) dx + f(x_1) \frac{-1}{h^2} \int_{x_0}^{x_2} (x - x_0)(x - x_2) dx + f(x_2) \frac{1}{2h^2} \int_{x_0}^{x_2} (x - x_0)(x - x_1) dx$$

After expanding and integrating each integral (straightforward polynomial integration):

$$\begin{aligned} \int_{x_0}^{x_2} (x - x_1)(x - x_2) dx &= \frac{2h^3}{3} \\ \int_{x_0}^{x_2} (x - x_0)(x - x_2) dx &= -\frac{4h^3}{3} \\ \int_{x_0}^{x_2} (x - x_0)(x - x_1) dx &= \frac{2h^3}{3} \end{aligned}$$

Substitute those three values into the global expression:

$$\int_{x_0}^{x_2} P_2(x) dx = f(x_0) \frac{1}{2h^2} \cdot \frac{2h^3}{3} + f(x_1) \frac{-1}{h^2} \cdot \left(-\frac{4h^3}{3} \right) + f(x_2) \frac{1}{2h^2} \cdot \frac{2h^3}{3}$$

Simplifying:

$$\begin{aligned}\int_{x_0}^{x_2} P_2(x) dx &= \frac{h}{3} f(x_0) + \frac{4h}{3} f(x_1) + \frac{h}{3} f(x_2) \\ &= \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)]\end{aligned}$$

$$I_{s1/3} = \int_{x_0}^{x_2} f(x) dx = \left(\int_{x_0}^{x_2} P_2(x) dx \right) + E = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + E$$

Note: the factor 1/3 that multiplies the (h) is at the origin of the name of the method.

a) Simple Simpson's method 1/3

means using one interval

$$I_{s1/3} = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + E_{s13}$$

Example: calculate this $\int_0^{\pi/2} \sin x dx$ by Simpson's method 1/3.

Solution:

$$\int_0^{\pi/2} \sin x dx = \frac{\pi}{12} \left[\sin 0 + 4 \sin \frac{\pi}{4} + \sin \frac{\pi}{2} \right] = 1,002$$

b) Compound Simpson 1/3:

The compound Simpson's 1/3 method is an extension of the simple Simpson's 1/3 formula, used to approximate the integral of a function over a larger interval $[a, b]$ with higher accuracy. Instead of applying the method over a single interval, we divide the input interval into n equal subintervals. Each subinterval is further divided into two parts of width h , so that:

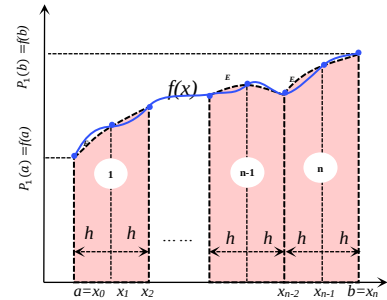
$$h = \frac{b - a}{2n}$$

the x-coordinates points are

$$x_i = a + i h, \quad i = 0, 1, 2, \dots, 2n$$

Applying Simpson's 1/3 to each pair of subintervals and summing over the entire interval gives the compound formula:

$$I_{s1/3} = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 2f(x_{2n-2}) + 4f(x_{2n})] + E_{s13}$$



Remark:

- The first and last term without multiplication.
- Pairs terms are multiplied by 2.
- odd terms are multiplied by 4.

Simpson's 1/3 error

The error is given by the formula:

$$E_{s13} \leq \frac{(b - a)^5}{180n^4} \left[\max_{x \in [a, b]} |f^{(4)}(x)| \right]$$

Note: This method is accurate with error $E = 0$, if the fourth derivative is zero.

Example: Find this integral $\int_0^{\pi/2} \sin x \, dx$ using Simpson 1/3 with $n=2$ intervals.

Solution : $h = \frac{b-a}{2n} = \frac{\pi/2-0}{2(2)} = \frac{\pi}{8}$

$x_i = a + i h = \frac{\pi}{8}i, \quad i=1, \dots, 2n=0, 1, 2, 3, 4$

x_0	x_1	x_2	x_3	x_4
0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$

$$\begin{aligned}
 I_{s1/3} &= \int_0^{\pi/2} \sin x \, dx = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)] + E \\
 &= \frac{\pi}{24} \left[\sin 0 + 4 \sin \frac{\pi}{8} + 2 \sin \frac{\pi}{4} + 4 \sin \frac{3\pi}{8} + \sin \frac{\pi}{2} \right] + E \\
 &= \frac{\pi}{24} [0 + 4(0.382683) + 2(0.707107) + 4(0.923880) + 1] + E \\
 &= \frac{\pi}{24} (7.64390) \approx \boxed{1.00013}
 \end{aligned}$$

$$\text{Error} = I_{s1/3} - I_{\text{exact}} = 1.3e - 4$$

This demonstrates the high accuracy of the compound Simpson's 1/3 method even with just two intervals.

3.4 Simpson's 3/8 Rule Method²

Simpson's 3/8 rule is a numerical integration technique based on approximating the integrand by a cubic polynomial (degree $n=3$) over an interval $[x_0, x_3]$. The interpolating polynomial passes through four equally spaced points: $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$, $(x_3, f(x_3))$,

a) Simple Simpson's 3/8 method

Approximating the integrand by a cubic polynomial and one interval leads this formula :

$$I_{s3/8} = \int_{x_0}^{x_3} f(x) dx = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] + E_{s3/8}$$

b) Compound Simpson's 3/8 method:

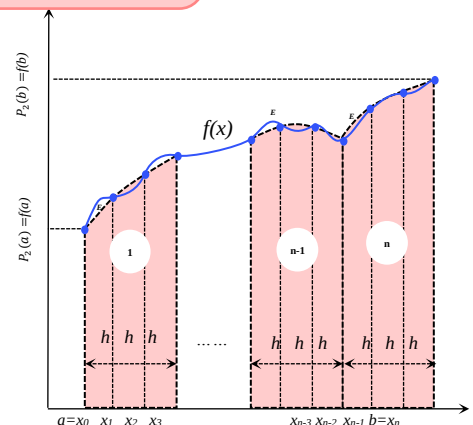
by dividing the interval $[a, b]$ into n subintervals, in each subinterval is divided into three equal parts of width h .

Therefore

$$h = \frac{b-a}{3n}$$

the x-coordinates points are

$$x_i = a + i h, \quad i = 0, 1, 2, \dots, 3n$$



Applying the Simpson's 3/8 rule on each group of three subintervals and summing over the entire interval gives the compound Simpson's 3/8 formula:

$$I_{s3/8} = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + f(x_4) + \dots + 2f(x_{3n-3}) + 3f(x_{3n-2}) + 3f(x_{3n-1}) + f(x_{3n})] + E_{s3/8}$$

Example: Find the $\int_0^{\pi/2} \sin x \, dx$ by Simpson 3/8 with $n=1$ interval.

Solution : $h = \frac{b-a}{3n} = \frac{\pi/2-0}{3(1)} = \frac{\pi}{6}$

$x_i = a + i h = \frac{\pi}{6} i$ where $i=0, 3n=0,3$

x_0	x_1	x_2	x_3
0	$\pi/6$	$\pi/3$	$\pi/2$

$$I_{s3/8} = \int_0^{\pi/2} \sin x \, dx = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] + E$$

$$= \frac{\pi}{16} \left[\sin 0 + 3 \sin \frac{\pi}{6} + 3 \sin \frac{\pi}{3} + \sin \frac{\pi}{2} \right] + E = \frac{\pi}{16} \left[0 + \frac{3}{2} + \frac{3\sqrt{3}}{2} + 1 \right] + E$$

$$= \frac{\pi}{16} \left(\frac{5 + 3\sqrt{3}}{2} \right) + E \approx \boxed{1.00026}$$

$$Error = I_{s3/8} - I_{exact} = 2.6e - 4$$

3.7 Gauss Quadrature Method

3.7.1 Definition

Gaussian quadrature is a numerical method used to approximate the value of a definite integral. Unlike classical numerical integration methods that use equally spaced points, the Gaussian method evaluates the integrand at specially chosen points called *Gauss points*, each associated with a specific weight. These points and weights are selected so that the approximation is exact for polynomials of the highest possible degree.

For an integral over a given interval, the method first transforms the interval into a standard interval, usually $[-1,1]$, and then computes the integral as a weighted sum of the function values at the Gauss points. Gaussian quadrature provides very high accuracy using only a small number of evaluation points, which makes it widely used in engineering applications, especially in the finite element method.

An important property of Gaussian quadrature is that it can be easily extended to evaluate double and triple integrals by using tensor products of Gauss points, which makes it particularly suitable for multidimensional problems in engineering, unlike most classical methods.

3.7.2 General Principle

The integral over a general interval $[a, b]$:

$$\int_a^b f(x) \, dx$$

is transformed into an integral over a standard interval $[-1,1]$:

$$\int_{-1}^1 g(\xi) d\xi$$

This transformation simplifies the application of Gauss points and weights.

Change of Variables for the Transformation :

In order to transform the integration interval from $[a,b]$ to the standard interval $[-1,1]$, we introduce the following change of variables:

$$x = \frac{b-a}{2} \xi + \frac{a+b}{2}; \quad dx = \frac{b-a}{2} d\xi$$

Thus

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}\xi + \frac{a+b}{2}\right) d\xi$$

3.7.3 Gaussian Quadrature Formula

The Gauss quadrature approximation is written as:

$$\int_{-1}^1 g(\xi) d\xi \approx \sum_{i=1}^n w_i g(\xi_i)$$

where:

- ξ_i are the Gauss integration points
- w_i are the weights
- n is the number of Gauss points

The final approximation over $[a,b]$ becomes:

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^n w_i f(x_i)$$

with: $x_i = \frac{b-a}{2} \xi_i + \frac{a+b}{2}$

Gauss–Legendre Points and Weights (Standard interval $[-1,1]$)

Points ξ_i	Weights w_i
One Point Gaussian Quadrature(Exact for polynomials of degree ≤ 1)	
0	2
Two Point Gaussian Quadrature(Exact for polynomials of degree ≤ 3)	
$\left[-\frac{1}{\sqrt{3}}; +\frac{1}{\sqrt{3}}\right]$	$[1; 1]$
Three Point Gaussian Quadrature(Exact for polynomials of degree ≤ 5)	
$\left[-\sqrt{\frac{3}{5}}; 0; +\sqrt{\frac{3}{5}}\right]$	$\left[\frac{5}{9}; \frac{8}{9}; \frac{5}{9}\right]$
Four Point Gaussian Quadrature(Exact for polynomials of degree ≤ 7)	
-0.8611363116	0.3478548451
-0.3399810436	0.6521451549
+0.3399810436	0.6521451549
+0.8611363116	0.3478548451
Five Point Gaussian Quadrature(Exact for polynomials of degree ≤ 9)	
-0.9061798459	0.236926885
-0.5384693101	0.478628671

0	0.568888889
+0.5384693101	0.478628671
+0.9061798459	0.236926885

Example

Approximate this integral using the **two-point Gauss quadrature**.

$$I = \int_0^6 (6 + x \sin x) dx$$

Step 1: Interval Transformation (change variables) :

$$x = 3\xi + 3, \quad dx = 3 d\xi$$

Step 2: Gauss 2-Points and Weights

$$\xi_1 = -\frac{1}{\sqrt{3}}, \quad \xi_2 = \frac{1}{\sqrt{3}}, \quad w_1 = w_2 = 1$$

Step 3: Corresponding x-Values

$$x_1 = 3\left(-\frac{1}{\sqrt{3}}\right) + 3 = 1.2679$$

$$x_2 = 3\left(\frac{1}{\sqrt{3}}\right) + 3 = 4.7321$$

Step 4: Function Evaluation

$$f(x_1) = 6 + 1.2679 \sin(1.2679) = 7.2096$$

$$f(x_2) = 6 + 4.7321 \sin(4.7321) = 2.7217$$

Step 5: Gauss Approximation

$$I_G = \frac{6-0}{2} [f(x_1) + f(x_2)] = 3(7.2096 + 2.7217)$$

$$\boxed{I_G = 29.7939} \quad \text{check it}$$

Exact (Analytical) Solution

$$I_{Ana} = \int_0^6 (6 + x \sin x) dx = [6x + \sin x - x \cos x]_0^6 = 29.95953 - 0 = 29.95953$$

$$Error = |I_{Ana} - I_G| = 0.1653$$

3.8 Conclusion

Numerical integration methods provide practical tools for approximating definite integrals when analytical solutions are difficult or impossible to obtain. The rectangle and trapezoidal methods are simple to implement but may lead to relatively large errors, especially for rapidly varying functions. Simpson's 1/3 and Simpson's 3/8 methods improve accuracy by approximating the integrand with higher-degree polynomials. These Simpson methods converge faster and yield very accurate results for smooth functions. Increasing the number of subintervals generally reduces the approximation error for all methods. The choice of method depends on the required accuracy, computational effort, and smoothness of the function. In practice, Simpson's 1/3 method is often preferred due to its balance between accuracy and efficiency. Overall, numerical integration plays a fundamental role in engineering and scientific computations.

Serie of Exercises 03

Exercise 1 (compound left rectangle method)

Approximate the following integral using the compound left rectangle method with $n=5$ subintervals

$$I = \int_0^1 (1+x^2) dx$$

Exercise 2 (Compound Rectangle Method (Midpoint))

Approximate the following integral, using the compound midpoint rectangle method with $n=5$.

$$I = \int_0^2 x^2 dx$$

Exercise 3 (Compound Trapezoidal Method)

Approximate the following integral using the compound trapezoidal method with $n=5$ and precision 0.0001.

$$I = \int_0^1 e^x dx.$$

Exercise 4

Using Compound Trapezoidal Method, find the approximate value of this integral with $n=3$ and precision 0.0001.

$$I = \int_1^2 \ln x dx$$

Exercise 5 (Compound Simpson's 1/3 Method)

Approximate $I = \int_{6.5}^{8.5} \sin(x) x^3 dx$, using Simpson's 1/3 with $n=2$ (4 subintervals) and 3 decimals.

Exercise 6

Evaluate the previous integral using Simpson's 3/8 with $n=2$ (4 subintervals) and 4 decimals.

Exercise 7

Compute those integral by four following methods with precision 0.0001

$$1. I = \int_{2.5}^{5.5} \cos(x) + \frac{5}{x} dx \quad 2. I = \int_0^2 x^2 e^x dx$$

1. Composite Rectangle method with $n=6$
 2. Composite Trapezoidal method with $n=5$
 3. Simpson's 1/3 rule with $n=2$ ($\Rightarrow$ 4 subintervals)
 4. Simpson's 3/8 rule with $n=2$ ($\Rightarrow$ 6 subintervals)
- Then compare the results with the analytic value.

Exercise 8 (structural engineering)

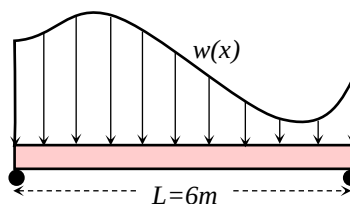
Find the bending moment at a section of a beam by integrating of the load intensity along the beam:

$$M = \int_0^L w(x) dx$$

Where the beam is simply supported of length $L=6$ m carries a non-uniform distributed load given by: $w(x) = 6 + x \sin x$ (kN/m)

Use the following method :

1. Composite Rectangle method with $n=6$
2. Composite Trapezoidal method with $n=5$
3. Simpson's 1/3 rule with $n=2$ ($\Rightarrow$ 4 subintervals)
4. Simpson's 3/8 rule with $n=2$ ($\Rightarrow$ 6 subintervals)



Solutions of Serie 03:

Solution to exercise 1

$$I = \int_0^1 (1+x^2) dx$$

Width of the subintervals : $h = \frac{b-a}{n} = \frac{1-0}{5} = 0.2$

x and y -coordinates of the points : $x_i = a + ih = 0.2i, \quad i = 0,1,2,3,4$

Subinterval i	0	1	2	3	4
x_i	0.0	0.2	0.4	0.6	0.8
$y = f(x_i)$	1.000	1.040	1.160	1.360	1.640

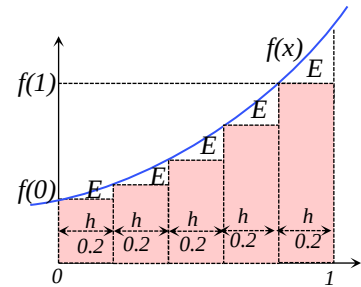
Approximation formula of compound left rectangle method

$$I_r = h \sum_{i=0}^4 f(x_i) + E$$

$$I_r \approx 0.2(1 + 1.04 + 1.16 + 1.36 + 1.64) = \boxed{1.24} + E.$$

Exact value $I = \int_0^1 (1+x^2) dx = \left[x + \frac{x^3}{3} \right]_0^1 = 1 + \frac{1}{3} = 1.3333$

Error $E = |1.3333 - 1.24| = \mathbf{0.0933}$



Solution to exercise 2

$$I = \int_0^2 x^2 dx$$

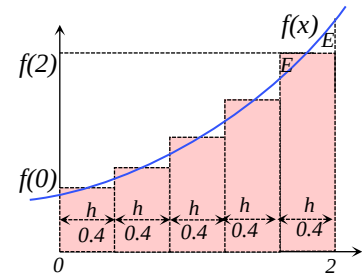
$h = \frac{2-0}{5} = 0.4$; $x_i^* = a + \left(i + \frac{1}{2}\right)h = 0.4(i + 0.5) \quad i = 0,1,2,3,4$

Subinterval i	0	1	2	3	4
x_i	0.2	0.6	1	1.4	1.8
$y = f(x_i)$	0.04	0.360	1	1.960	3.240

$$I_r \approx h \sum f(x_i^*) = 0.4(0.04 + 0.36 + 1 + 1.96 + 3.24) = 0.4(6.6) = \boxed{2.64}$$

Exact value $I = \int_0^2 x^2 dx = \left[\frac{x^3}{3} \right]_0^2 = \frac{2^3}{3} = 2.6667$

Error $E = |2.6667 - 2.64| = \mathbf{0.0267}$



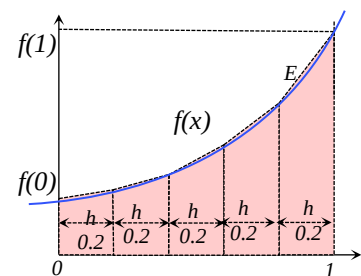
Solution to exercise 3

$$I = \int_0^1 e^x dx$$

$h = \frac{1-0}{5} = 0.2$

x and y -coordinates of the points : $x_i = a + ih = 0.2i, \quad i = 0,1,2,3,4$

Subinterval i	0	1	2	3	4	5
-----------------	---	---	---	---	---	---



x_i	0	0.2	0.4	0.6	0.8	1
$y = f(x_i)$	1	1.2214	1.4918	1.8221	2.2255	2.7183

Approximation formula of compound trapezoidal method

$$I_t = \frac{h}{2} \left[f_0 + 2 \sum_{i=1}^4 f_i + f_5 \right] + E$$

$$I_t = \frac{0.2}{2} (1 + 2(1.2214 + 1.4918 + 1.8221 + 2.2255) + 2.7183) + E$$

$$= 0.1(1 + 2(6.7609) + 2.7183) + E = \boxed{1.7140} + E$$

Exact value $I = \int_0^1 e^x dx = [e^x]_0^1 = e - 1 = 1.7183$

Error $E = |1.7183 - 1.7140| = \mathbf{0.0043}$

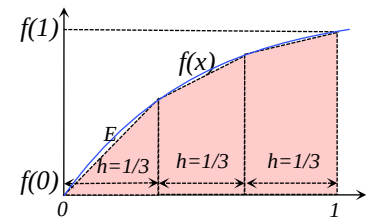
Solution to exercise 4

$$I = \int_1^2 \ln x dx$$

$$h = \frac{2-1}{3} = \frac{1}{3}$$

x and y -coordinates of the points : $x_i = a + ih = 1 + \frac{i}{3}$, $i = 0, 1, 2, 3$

Subinterval i	0	1	2	3
x_i	1	4/3	5/3	2
$y = f(x_i)$	0	0.2877	0.5108	0.6931



Approximation formula of compound trapezoidal method

$$I_t = \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^2 f(x_i) + f(x_3) \right] + E$$

$$I_t = \frac{1}{6} (0 + 2(0.2877 + 0.5108) + 0.6931) + E = \boxed{0.3817} + E$$

Exact value $I = \int_1^2 \ln(x) dx = [x \ln x - x]_1^2 = 2 \ln 2 - 1 = 0.3863$

Error $E = |0.3863 - 0.3817| = \mathbf{0.0046}$

Solution to exercise 5

$$I = \int_{6.5}^{8.5} \sin(x) x^3 dx$$

$$h = \frac{b-a}{2n} = \frac{8.5-6.5}{4} = 0.5$$

x and y -coordinates of the points : $x_i = a + ih = 6.5 + \frac{i}{2}$, $i = [0, 2n] = 0, 1, 2, 3, 4$

Subinterval i	0	1	2	3	4
x_i	6.5	7	7.5	8	8.5
$y = f(x_i)$	59.077	225.346	395.719	506.551	490.371

Approximation formula of compound Simpson's 1/3 method

$$\begin{aligned}
 I_{s1/3} &= \frac{h}{3} [f_0 + 4f_1 + 2f_2 + 4f_3 + f_4] + E \\
 &= 0.166667[59.077 + 4(225.346) + 2(395.719) + 4(506.551) + 490.371] + E \\
 &= \boxed{711.413} + E
 \end{aligned}$$

Exact value (analytical)

We integrate by parts. A known antiderivative is:

$$\int x^3 \sin x \, dx = -x^3 \cos x + 3x^2 \sin x + 6x \cos x - 6 \sin x + C$$

$$I_{exact} = [-x^3 \cos x + 3x^2 \sin x + 6x \cos x - 6 \sin x]_{6.5}^{8.5} = 507.28911 - (-204.1327) = 711.42183$$

$$\text{Error } E = |711.42183 - 711.413| = \mathbf{0.0088}$$

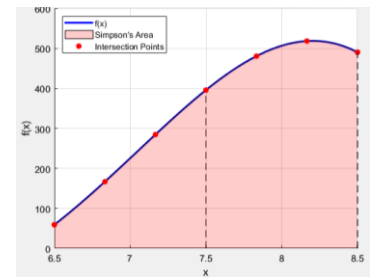
Solution to exercise 6

We repeat the approximation of $\int_{6.5}^{8.5} x^3 \sin x \, dx$ using Simpson's 3/8 rule with $n=2$ and precision 0.0001

this means 6 subintervals (because each Simpson 3/8 has 3 subintervals).

$$h = \frac{b-a}{3n} = \frac{8.5-6.5}{6} = \frac{2}{6} = \frac{1}{3} \approx 0.3333$$

x and y -coordinates of the points : $x_i = a + ih = 6.5 + \frac{i}{3}$, $i = [0, 3n] = 0,1,2,3,4,5,6$



Subinterval i	0	1	2	3	4	5	6
x_i	6.5	6.8333333	7.1666667	7.5	7.8333333	8.1666667	8.5
$y = f(x_i)$	59.0773	166.8186	284.5145	395.7187	480.5596	518.2607	490.3709

Approximation formula of compound Simpson's 3/8 formula For 6 subintervals:

$$\begin{aligned}
 I_{s3/8} &= \frac{3h}{8} [f_0 + 3(f_1 + f_2 + f_4 + f_5) + 2f_3 + f_6] + E \\
 &= \frac{3}{8} \left(\frac{1}{3} \right) [59.07733 + 3(166.81863 + 284.51446 + 480.55957 + 518.26075) + 2(395.71874) \\
 &\quad + 490.37090] + E \\
 &= 711.4182449 + E = \mathbf{711.4183} + E
 \end{aligned}$$

$$\text{Error } E = |711.4183 - 711.413| = \mathbf{0.0053}$$

Solution to exercise 7

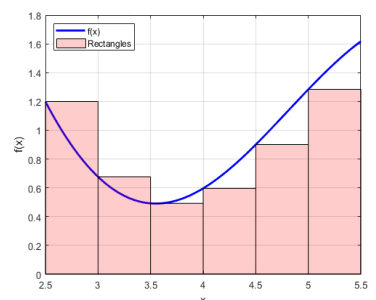
Integral 1:

$$I = \int_{2.5}^{5.5} \cos(x) + \frac{5}{x} \, dx$$

1. Left Rectangle Method $n = 6$

$$h = \frac{b-a}{n} = \frac{5.5-2.5}{6} = \frac{3}{6} = \mathbf{0.5}$$

x -coordinates of the points : $x_i = a + ih = 2.5 + 0.5i$, $i = 0,1,2,3,4,5$



Subinterval i	0	1	2	3	4	5
x_i	2.5	3	3.5	4	4.5	5
$y = f(x_i)$	1.19886	0.67667	0.49211	0.59636	0.90032	1.28366

$$I_R \approx h \sum_{i=0}^5 f(x_i) = 0.5 \times (1.19886 + 0.67667 + 0.49211 + 0.59636 + 0.90032 + 1.28366)$$

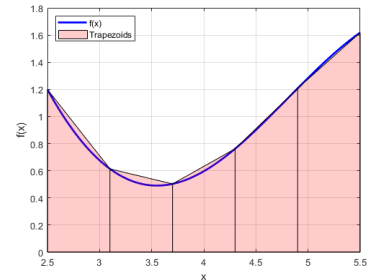
$$I_R \approx 2.57399$$

2. Trapezoidal Method $n = 5$

$$h = \frac{b-a}{n} = \frac{5.5-2.5}{5} = \frac{3}{5} = 0.6$$

x -coordinates of the points : $x_i = a + ih = 2.5 + 0.75i$, $i = 0, 1, 2, 3, 4$

Subinterval i	0	1	2	3	4	5
x_i	2.5	3.1	3.7	4.3	4.9	5.5
$y = f(x_i)$	1.19886	0.61377	0.50325	0.76199	1.20692	1.61776



$$I_t \approx \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^4 f(x_i) + f(x_5) \right] \approx 0.3 [1.19886 + 2(0.61377 + 0.50325 + 0.76199 + 1.20692) + 1.61776]$$

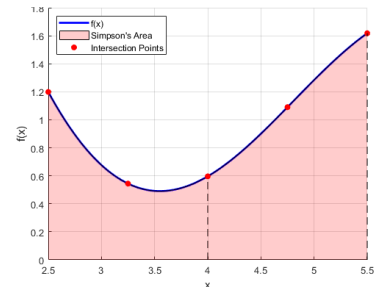
$$I_t \approx 2.69654$$

2. Simpson's 1/3 Method $n = 2$

$$h = \frac{b-a}{2n} = \frac{5.5-2.5}{4} = \frac{3}{4} = 0.75$$

x -coordinates of the points : $x_i = a + ih = 2.5 + 0.75i$, $i = 0, 1, 2, 3, 4$

Subinterval i	0	1	2	3	4
x_i	2.5	3.25	4	4.75	5.5
$y = f(x_i)$	1.19886	0.54433	0.59636	1.09023	1.61776



$$I_{s1/3} \approx \frac{h}{3} [f_0 + 4f_1 + 2f_2 + 4f_3 + f_4]$$

$$= \frac{0.75}{3} [1.19886 + 4 * 0.54433 + 2 * 0.59636 + 4 * 1.09023 + 1.61776]$$

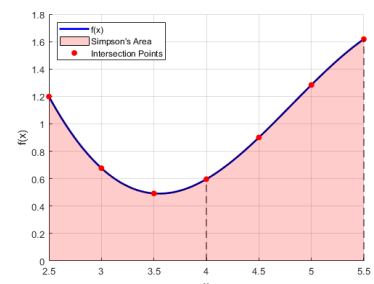
$$I_{s1/3} \approx 2.63690$$

2. Simpson's 3/8 Method $n = 2$

$$h = \frac{b-a}{3n} = \frac{5.5-2.5}{6} = \frac{3}{6} = 0.5$$

x -coordinates of the points : $x_i = a + ih = 2.5 + 0.5i$, $i = 0, 1, 2, 3, 4, 5, 6$

Subinterval i	0	1	2	3	4	5	6
x_i	2.5	3	3.5	4	4.5	5	5.5
$y = f(x_i)$	1.19886	0.67667	0.49211	0.59636	0.90032	1.28366	1.61776



$$I_{s3/8} \approx \frac{3h}{8} [f_0 + 3(f_1 + f_2 + f_4 + f_5) + 2f_3 + f_6]$$

$$= 0.1875[1.19886 + 3(0.67667 + 0.49211 + 0.90032 + 1.28366) + 2 * 0.59636 + 1.61776]$$

$$I_{s3/8} \approx 2.63768$$

5. Analytic solution

$$I_{ana} = \int_{2.5}^{5.5} (\cos x + \frac{5}{x}) dx = [\sin x + 5 \ln x]_{2.5}^{5.5} = (7.81820 - 5.17993) = 2.63827433$$

$$\approx 2.63827$$

Method	I_r (n = 6)	I_t (n = 5)	$I_{s1/3}$ (n=2)	$I_{s3/8}$ (n=2)	Exact
Value	2.57399	2.69654	2.63690	2.63768	2.63827
Absolute error	0.06428	-0.05827	0.00138	0.00059	
Relative error(%)	2.44	2.21	0.05	0.02	

Integral 2:

$$I = \int_0^2 x^2 e^x dx$$

method	analytic	Rectangle n=6	Trapezoidal n=4	Simpson's 1/3 n=2	Simpson's 3/8 n=1
value	12.7781	8.3942	14.6612	12.7844	13.0631

Solution to exercise 7

Exact (Analytical) Solution

$$\int_0^6 (6 + x \sin x) dx = [6x + \sin x - x \cos x]_0^6 = 29.95953 - 0 = 29.95953$$

Method	I_r (n=6)	I_t (n=4)	$I_{s1/3}$ (n=2)	$I_{s3/8}$ (n=2)	Exact
Value	31.26160	31.02371	29.77983	29.88185	29.95953
Absolute error	1.30210	1.06421	-0.17967	-0.07765	
Relative error(%)	4.35	3.55	0.60	0.26	

Chapter 4: Polynomial interpolation

4.1 Introduction

4.2 Lagrange Method

4.3 Newton Method

4.3.2 General Newton form with divided differences

4.3.3 Newton Forward and Backward Interpolation

4.4 Estimation of the interpolation error

4.5 Conclusion

3.1 Introduction

Interpolation, a common method in mathematics and science, estimates values between known data points. It constructs a function or curve passing through these points to predict intermediate values. Useful when limited data is present, it creates accurate approximations of missing data. Interpolation is applied to various fields (science, engineering, computer science), providing continuous and complete information, facilitating decision-making and analysis. In short, essential for estimating values between data points, improving understanding of hidden relationships and trends.

4.2 Lagrange Interpolation Method

4.2.1 Definition

The Lagrange Method is the most clever construction of the interpolating polynomial directly, without solving a system of equations. It is based on special basis polynomials called Lagrange basis polynomials.

4.2.2 Lagrange Basis Polynomials

For $n + 1$ points $(x_0, y_0), \dots, (x_n, y_n)$, define:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

$$L_i(x) = \frac{((x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n))}{((x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n))}$$

Properties: $L_i(x_j) = 1$ for $j = i$; $L_i(x_j) = 0$ for $j \neq i$

4.2.3 Lagrange Interpolating Polynomial

The interpolating polynomial is:

$$P_n(x) = \sum_{i=0}^n y_i L_i(x)$$

4.2.4 Calculation Procedure:

- Data points (x_i, y_i) , where x_i represents the input values and y_i the corresponding output values.
- Construction of Lagrange polynomials:

The general formula for the Lagrange polynomial $L_i(x)$ is:

$$L_i(x) = \frac{((x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n))}{((x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n))}$$

- Construction of the interpolation polynomial:

$$P_n(x) = y_0 * L_0(x) + y_1 * L_1(x) + \dots + y_n * L_n(x).$$

- Estimating intermediate values: Once the interpolation polynomial $P(x)$ is constructed, you can use it to estimate the output values (y) for input values (x) between known data points.

Example 1: Lagrange Interpolation (Quadratic Polynomial)

Given the following data points:

$$(1, 2), (2, 3), (4, 1)$$

Construct the interpolating polynomial using the Lagrange method.

Solution

Step 1: Identify the Data

We have: $x_0 = 1, y_0 = 2$; $x_1 = 2, y_1 = 3$; $x_2 = 4, y_2 = 1$

Since we have 3 points, the interpolating polynomial is of degree at most 2.

Step 2: Write the Lagrange Basis Polynomials

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}; \quad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}; \quad L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Step 3: Substitute the Numerical Values

$$\begin{aligned} L_0(x) &= \frac{(x - 2)(x - 4)}{(1 - 2)(1 - 4)} = \frac{(x - 2)(x - 4)}{3} \\ L_1(x) &= \frac{(x - 1)(x - 4)}{(2 - 1)(2 - 4)} = -\frac{(x - 1)(x - 4)}{2} \\ L_2(x) &= \frac{(x - 1)(x - 2)}{(4 - 1)(4 - 2)} = \frac{(x - 1)(x - 2)}{6} \end{aligned}$$

Step 4: Construction of the Interpolation Polynomial

$$\begin{aligned} P(x) &= y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) = 2L_0(x) + 3L_1(x) + 1L_2(x) \\ &= \frac{2(x - 2)(x - 4)}{3} - \frac{3(x - 1)(x - 4)}{2} + \frac{(x - 1)(x - 2)}{6} \\ &= \frac{2x^2 - 12x + 16}{3} - \frac{3x^2 - 15x + 12}{2} + \frac{x^2 - 3x + 2}{6} \end{aligned}$$

The common denominator is 6.

$$\begin{aligned} P(x) &= \left(\frac{2}{6}\right) \frac{2x^2 - 12x + 16}{3} - \left(\frac{3}{6}\right) \frac{3x^2 - 15x + 12}{2} + \frac{x^2 - 3x + 2}{6} \\ &= \frac{(4x^2 - 24x + 32) + (-9x^2 + 45x - 36) + (x^2 - 3x + 2)}{6} \end{aligned}$$

$$= \frac{(4 - 9 + 1)x^2 + (-24 + 45 - 3)x + (32 - 36 + 2)}{6} = \frac{-4x^2 + 18x - 2}{6}$$

$$\boxed{P(x) = -\frac{2}{3}x^2 + 3x - \frac{1}{3}}$$

We verify the first point of the input data :

$$P(1) = -\frac{2}{3} + 3 - \frac{1}{3} = 2\checkmark$$

4.2.5 Advantages and Limitations

Advantages

- Explicit formula
- Easy to implement for small data sets

Limitations

- Computationally expensive for large n
- Entire polynomial must be recomputed if a new point is added

4.3 Newton Interpolation Method

4.2.1 Definition

Newton's interpolation method is an alternative approach for constructing the interpolating polynomial through a given set of data points. It provides a more efficient formulation than the Lagrange method, especially when dealing with a large number of points. One of its main advantages is that the polynomial can be updated easily when new data points are added, without recomputing the entire expression. The method is based on the concept of divided differences, which allows the polynomial coefficients to be computed recursively. Because of its computational efficiency and flexibility, Newton's method is widely used in numerical analysis and scientific computing.

Newton's method can be divided into two main types, depending on the distribution of the interpolation points:

- General Newton form with divided differences, which works for any set of points, including points with variable spacing.
- Forward and backward differences, which are particularly suitable for equally spaced points and take advantage of difference tables for efficient computation.

4.3.2 General Newton form with divided differences

Divided differences constitute the core of Newton's interpolation method. They provide a systematic way to compute the coefficients of the interpolating polynomial directly from the given data points. Defined recursively, divided differences are commonly arranged in a triangular table, which improves both clarity and computational efficiency. Each level of divided differences corresponds to the slope of an interpolating polynomial of higher degree. This organized formulation makes Newton's method particularly effective for theoretical understanding as well as for practical numerical implementation.

- Zero-order divided difference: $f[x_i] = \frac{f(x_i)}{x_i}$
- First-order divided difference: $f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$

- Second-order divided difference: $f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$
-
- Higher-order divided differences: $f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$

These values are usually organized in a divided difference table as follows.

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	...	$f[x_i, \dots, x_{i+k}]$
x_0	$f[x_0] = y_0$	$f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$			
x_1	$f[x_1] = y_1$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
x_2	$f[x_2] = y_2$	$= \frac{y_2 - y_1}{x_2 - x_1}$	$f[x_1, x_2, x_3]$	...	$f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]$
$\vdots$	$\vdots$	$f[x_2, x_3]$	$\vdots$	...	$x_{i+k} - x_i$
$\vdots$	$\vdots$	$= \frac{y_3 - y_2}{x_3 - x_2}$	$f[x_{n-2}, x_{n-1}, x_n]$		
$\vdots$	$\vdots$	$\vdots$			
x_n	$f[x_n] = y_n$	$f[x_{n-1}, x_n]$			

The Newton polynomial is written as:

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, \dots, x_n] \prod_{k=0}^{n-1} (x - x_k)$$

Example 1: Newton Interpolation (Quadratic Polynomial)

Given the data: (1,1), (2,4), (3,9), Construct the interpolating polynomial using Newton's method.

Step 1: Write the Data $x_0 = 1, f(x_0) = 1$; $x_1 = 2, f(x_1) = 4$; $x_2 = 3, f(x_2) = 9$

Step 2: Compute Divided Differences

First order: $f[x_0, x_1] = \frac{4-1}{2-1} = 3$; $f[x_1, x_2] = \frac{9-4}{3-2} = 5$

Second order: $f[x_0, x_1, x_2] = \frac{5-3}{3-1} = 1$.

For clarity, the computation of divided differences is summarized in the triangular table below.

x_i	Zero order $y_i = f[x_i]$	1 st order $f[x_i, x_{i+1}]$	2 nd order $f[x_i, x_{i+1}, x_{i+2}]$
$x_0 = 1$	$y_0 = 1$	$f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{4-1}{2-1} = 3$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{5-3}{3-1} = 1$
$x_1 = 2$	$y_1 = 4$	$f[x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{9-4}{3-2} = 5$	
$x_2 = 3$	$y_2 = 9$		

Step 3: Write Newton's Polynomial

$$\begin{aligned}
 P(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
 &= 1 + 3(x - 1) + 1(x - 1)(x - 2) \\
 &= 1 + 3x - 3 + x^2 - 3x + 2 = x^2
 \end{aligned}$$

$$P(x) = x^2$$

Remark:

Since the given data points satisfy $y = x^2$, which is a polynomial of degree 2, the Newton interpolation using three points reproduces the exact function. In this case, the interpolation is exact, not an approximation.

Example 2: Comparison of Lagrange and Newton

Show that the Lagrange and Newton methods give the same polynomial for the data:

$$(x_0, y_0) = (1, 2), \quad (x_1, y_1) = (2, 5), \quad (x_2, y_2) = (4, 17)$$

Lagrange Interpolation

$$P(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) = 2L_0(x) + 5L_1(x) + 17L_2(x)$$

with:

$$L_0(x) = \frac{(x-2)(x-4)}{(1-2)(1-4)} = \frac{x^2 - 6x + 8}{3}, \quad L_1(x) = \frac{(x-1)(x-4)}{(2-1)(2-4)} = -\frac{x^2 - 5x + 4}{2},$$

$$L_2(x) = \frac{(x-1)(x-2)}{(4-1)(4-2)} = \frac{x^2 - 3x + 2}{6}$$

$$P(x) = 2\left(\frac{x^2 - 6x + 8}{3}\right) - 5\left(\frac{x^2 - 5x + 4}{2}\right) + 17\left(\frac{x^2 - 3x + 2}{6}\right)$$

After simplification, we obtain:

$$\begin{aligned} P(x) &= \left(\frac{2}{2}\right) 2\left(\frac{x^2 - 6x + 8}{3}\right) - \left(\frac{3}{3}\right) 5\left(\frac{x^2 - 5x + 4}{2}\right) + 17\left(\frac{x^2 - 3x + 2}{6}\right) \\ &= \left(\frac{4x^2 - 24x + 32}{6}\right) + \left(\frac{-15x^2 + 75x - 60}{6}\right) + \left(\frac{17x^2 - 51x + 34}{6}\right) \\ &= \frac{(4 - 15 + 17)x^2 + (-24 + 75 - 51)x + (32 - 60 + 34)}{6} = \frac{6x^2 + 6}{6} \end{aligned}$$

$$P(x) = x^2 + 1$$

Newton Interpolation

First, compute the divided differences:

x_i	Zero order $y_i = f[x_i]$	1 st order $f[x_i, x_{i+1}]$	2 nd order $f[x_i, x_{i+1}, x_{i+2}]$
1	$f[x_0] = 2$	$f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{5 - 2}{2 - 1} = 3$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{6 - 3}{4 - 1} = 1$
2	$f[x_1] = 5$		
4	$f[x_2] = 17$	$f[x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{17 - 5}{4 - 2} = 6$	

The Newton polynomial is:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) = 2 + 3(x - 1) + 1(x - 1)(x - 2)$$

$$= 2 + 3x - 3 + x^2 - 3x + 2 = x^2 + 1$$

$$P(x) = x^2 + 1$$

Thus, both Lagrange and Newton forms represent the same interpolating polynomial, written differently.

Advantage of Newton's Form : Suppose a new data point (5,26) is added. In Lagrange form, the entire polynomial must be recomputed. But in Newton form, we only need one new divided difference term to add:

$$+f[x_0, x_1, x_2, x_3](x-1)(x-2)(x-4)$$

Newton's form allows efficient updating of the interpolating polynomial without recomputing previous terms.

Adding to the previous table the term

x_i	Zero order $y_i = f[x_i]$	1 st order $f[x_i, x_{i+1}]$	2 nd order $f[x_i, x_{i+1}, x_{i+2}]$	3 rd order $f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
1	$f[x_0] = 2$	$f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{5-2}{2-1} = 3$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{6-3}{4-1} = 1$	$f[x_0, x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_0} = \frac{9-6}{5-1} = 1$
2	$f[x_1] = 5$	$f[x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{17-5}{4-2} = 6$	$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{9-6}{5-2} = 1$	$f[x_1, x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_1} = \frac{26-9}{5-1} = 7$
4	$f[x_2] = 17$	$f[x_2, x_3] = \frac{y_3 - y_2}{x_3 - x_2} = \frac{26-17}{5-4} = 9$		
5	$f[x_3] = 26$			

Since: $f[x_0, x_1, x_2, x_3] = 0$, The new term to be added is: $+0 [(x-1)(x-2)(x-4)] = 0$

And the updated polynomial is same : $P(x) = x^2 + 1$.

4.3.3 Newton Forward and Backward Interpolation

Newton's forward and backward interpolation formulas are used when the interpolation points are equally spaced, that is

$$x_0, \quad x_1 = x_0 + h, \quad x_2 = x_0 + 2h, \quad \dots, \quad x_n = x_0 + nh.$$

In this case, the interpolation process can be simplified by using finite differences instead of divided differences.

The forward interpolation method is particularly suitable when the interpolation point is located near the beginning of the data table, while the backward interpolation method is more appropriate when the interpolation point lies near the end of the table. Both methods rely on successive differences of function values and are especially convenient for tabulated data and repetitive numerical calculations. Newton's forward and backward interpolation methods provide efficient and practical solutions when dealing with uniformly distributed interpolation points.

a) Newton Forward Interpolation Polynomial

Let $f(x_i) = y_i$ for $i = 0, 1, \dots, n$ real numbers. The following recurrence relations are called finite differences of order k of function f :

- 1st order finite differences: $\Delta f_i = f_{i+1} - f_i$ for $i = 0, 1, \dots, n-1$
- 2nd order finite differences: $\Delta^2 f(x_i) = \Delta f_i - \Delta f_{i+1}$ for $i = 0, 1, \dots, n-2$
- ...
- k^{th} order finite differences : $\Delta^k f(x_i) = \Delta^{k-1} f_i - \Delta^{k-1} f_{i+1}$ for $i = 0, 1, \dots, n-k$

These differences are usually presented in a triangular table for clarity.

x_i	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$	...	$\Delta^n f(x_i)$
-------	----------	-----------------	-------------------	-------------------	-----	-------------------

x_0	$f(x_0)$	$\Delta f(x_0)$	$\Delta^2 f(x_0)$	$\Delta^3 f(x_0)$		
x_1	$f(x_1)$	$\Delta f(x_1)$	$\Delta^2 f(x_1)$			
x_2	$f(x_2)$	$\Delta f(x_2)$	$\Delta^2 f(x_2)$			
x_3	$f(x_3)$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\Delta^n f(x_0)$
$\vdots$	$\vdots$				$\dots$	
x_{n-2}	$f(x_{n-2})$	$\Delta f(x_{n-3})$	$\Delta^2 f(x_{n-4})$	$\Delta^3 f(x_{n-5})$		
x_{n-1}	$f(x_{n-1})$	$\Delta f(x_{n-2})$	$\Delta^2 f(x_{n-3})$			
x_n	$f(x_n)$	$\Delta f(x_{n-1})$				

Considering that the function $f(x)$ is known in those points (x_i, y_i) : $f(x_i) = y_i, i = 0, 1, \dots, n$,
With $x_i = x_{i-1} + ih, \quad h > 0$.

then the forward Newton polynomial is:

$$P_n(x) = f(x_0) + \frac{(x - x_0)}{1! h} \Delta f(x_0) + \frac{(x - x_0)(x - x_1)}{2! h^2} \Delta^2 f(x_0) + \dots + \frac{1}{n! h^n} \left(\prod_{k=0}^{n-1} (x - x_k) \right) \Delta^n f(x_0)$$

Example

Given the following equally spaced data:

x_i	0	1	2	3
$f(x_i)$	1	3	7	13

Use Newton's forward interpolation formula to:

Construct the interpolating polynomial

Approximate the value of $f(1.5)$

Solution

1. Construction of the polynomial using Newton's forward interpolation formula

Step 1: Identify the Step Size : $h = x_1 - x_0 = 1$.

Step 2: Construct the Forward Difference Table

x_i	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$
$x_0 = 0$	$f(x_0) = 1$	$\Delta f(x_0) = 3 - 1 = 2$	$\Delta^2 f(x_0) = 4 - 2 = 2$	$\Delta^3 f(x_0) = 2 - 2 = 0$
$x_1 = 1$	$f(x_1) = 3$	$\Delta f(x_1) = 7 - 3 = 4$	$\Delta^2 f(x_1) = 6 - 4 = 2$	
$x_2 = 2$	$f(x_2) = 7$	$\Delta f(x_2) = 13 - 7 = 6$		
$x_3 = 3$	$f(x_3) = 13$			

Step 3: Write the Interpolating Polynomial

$$P_3(x) = f(x_0) + \frac{(x - x_0)}{h} \Delta f(x_0) + \frac{(x - x_0)(x - x_1)}{2! h^2} \Delta^2 f(x_0)$$

Substitute numerical values:

$$P_3(x) = 1 + (x - 0) \cdot 2 + \frac{(x - 0)(x - 1)}{2} \cdot 2 = 1 + 2x + (x)(x - 1) = 1 + 2x + x^2 - x$$

$$\boxed{P_3(x) = x^2 + x + 1}$$

2. Approximate $f(1.5)$: $f(1.5) \approx P_3(1.5)$

$$P_3(1.5) = (1.5)^2 + 1.5 + 1 = 2.25 + 1.5 + 1 = 4.75$$

b) Newton Backward Interpolation Polynomial

Similarly, if we want to interpolate near the last point, we can use backward differences:

$$\nabla f_i = f_i - f_{i-1}$$

Let $f(x_i) = y_i$ for $i = 0, 1, \dots, n$ be known real values of the function f at the interpolation points x_i . The following recurrence relations are called backward finite differences of order k of the function f :

- 1st order backward finite differences: $\nabla f_i = f_i - f_{i-1}$ for $i = 1, 2, \dots, n$
- 2nd order backward finite differences: $\nabla^2 f(x_i) = \nabla f_i - \nabla f_{i-1}$ for $i = 2, 3, \dots, n$
-
- k^{th} order backward finite differences: $\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1})$ for $i = k, k + 1, \dots, n$

These backward finite differences are usually arranged in a triangular table for clarity and ease of computation.

The interpolation points are assumed to be equally spaced, such that:

$$x_i = x_{i-1} + h, \quad h > 0, \quad i = 1, 2, \dots, n$$

Under these assumptions, the Newton backward interpolation polynomial is given by:

$$P_n(x) = f(x_n) + \frac{(x - x_n)}{1!h} \nabla f(x_n) + \frac{(x - x_n)(x - x_{n-1})}{2!h^2} \nabla^2 f(x_n) + \dots + \frac{1}{n!h^n} \left(\prod_{k=0}^{n-1} (x - x_{n-k}) \right) \nabla^n f(x_n)$$

Example

We bring the same previous example.

x_i	0	1	2	3
$f(x_i)$	1	3	7	13

Use Newton's backward interpolation formula to:

- Construct the interpolating polynomial
- Approximate the value of $f(2.7)$.

Solution

1. Construction of the polynomial using Newton's backward interpolation formula

Step 1: Identify the Step Size $h = x_1 - x_0 = 1$

The interpolation point $x=2.7$ is close to the last data point $x_3 = 3$, so the backward method is appropriate.

Step 2: Construct the Backward Difference Table

x_i	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$
$x_0 = 0$	$f(x_0) = 1$			
$x_1 = 1$	$f(x_1) = 3$			
$x_2 = 2$	$f(x_2) = 7$	$\Delta f(x_1) = 7 - 3 = 4$	$\Delta^2 f(x_1) = 6 - 4 = 2$	Doesn't exist
$x_3 = 3$	$f(x_3) = 13$	$\Delta f(x_2) = 13 - 7 = 6$		

Step 3: Write the Interpolating Polynomial

$$P_3(x) = f(x_3) + \frac{(x - x_3)}{h} \nabla f(x_3) + \frac{(x - x_3)(x - x_2)}{2! h^2} \nabla^2 f(x_3)$$

Substitute the numerical values:

$$\begin{aligned}
 P_3(x) &= 13 + (x - 3) \cdot 6 + \frac{(x - 3)(x - 2)}{2} \cdot 2 = 13 + 6(x - 3) + (x - 3)(x - 2) = \\
 &= 13 + 6x - 18 + x^2 - 5x + 6 \\
 &\quad \boxed{P_3(x) = x^2 + x + 1}
 \end{aligned}$$

2. Approximate $f(2.7)$: $\boxed{f(2.7) = P_3(x) \approx 10.99}$

4.4 Estimation of the interpolation error

In practice, interpolation does not always reproduce the exact function. The interpolation error measures the difference between the true function $f(x)$ and the interpolating polynomial $P_n(x)$:

$$E_n(x) = f(x) - P_n(x)$$

This error depends on:

- The smoothness of the function
- The degree of the interpolating polynomial
- The choice and distribution of interpolation points

The both Lagrange and Newton methods, there exists $c \in [x_0, x_n]$ such that the error given by :

$$E_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

And $|E_n(x)| \leq \frac{M}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right|$ where $M = \max_{x \in [x_0, x_n]} |f^{(n+1)}(x)|$

From the error formula, several important observations can be made:

- The magnitude of the error depends on the higher-order derivatives of $f(x)$; therefore, if these derivatives are large, the interpolation error increases.
- The error is exactly zero at the interpolation points, since the interpolating polynomial matches the function values at these nodes.
- The distribution of the interpolation points strongly influences the error, as poorly spaced points may lead to large oscillations of the interpolating polynomial.
- Hence, the error depends on the function and the placement of the points, not on the interpolation formula itself.

4.5 Conclusion

Polynomial interpolation is a fundamental numerical tool for approximating unknown values from discrete data, a situation frequently encountered in civil engineering applications such as material testing, structural monitoring, geotechnical measurements, and experimental data analysis. In this chapter, the Lagrange and Newton interpolation methods were presented as two equivalent but complementary approaches for constructing interpolating polynomials.

The Lagrange method offers a direct and explicit formulation that is well suited for small data sets and theoretical understanding. However, its main limitation lies in the need to recompute the entire polynomial when new data points are introduced. In contrast, Newton's interpolation method provides a more flexible and computationally efficient framework, particularly through its divided-

difference formulation, which accommodates non-uniformly spaced data and allows easy polynomial updating.

For equally spaced data, the Newton forward and backward interpolation formulas further simplify the computation by using finite differences, making them especially practical for tabulated experimental results and repetitive engineering calculations. The discussion of interpolation error highlights the importance of function smoothness and point distribution, emphasizing that numerical accuracy depends not only on the method but also on the quality and placement of data.

Overall, mastering these interpolation techniques equips civil engineering students with essential numerical skills for data approximation, analysis, and modeling, forming a solid foundation for more advanced topics in numerical methods, structural analysis, and computational engineering.

Serie of Exercises 04

Exercise 1(Lagrange Interpolation)

Let the function be defined by: $f(x) = \frac{1}{1+x}$

- Using Lagrange method to determine the interpolating polynomial of degree 2 that interpolates the function $f(x)$ at the points $x_0 = 0, x_1 = 1, x_2 = 3$
- Use the polynomial obtained in question (1) to estimate the value of $f(1.5)$.
- Give an upper bound of the interpolation error committed at $x = 1.5$.

Exercise 2 (Lagrange Interpolation)

Let the following experimental data be obtained from a laboratory test:

x_i	0	2	5
$f(x_i)$	1	4	26

- Determine the interpolating polynomial of degree 2 using the Lagrange method.
- Estimate the value of $f(3)$.
- Verify that the polynomial interpolates the given data exactly

Exercise 3 (Newton Interpolation-Divided Differences)

Consider the function values:

x_i	0	1	3
$f(x_i)$	1	0	2

- Determine the Newton interpolating polynomial.
- Evaluate the approximation of $f(2)$.

Exercise 4 (Newton Forward/Backward Interpolation)

The deflection of a beam is measured at equally spaced positions:

x_i	1	2	3	4
$f(x_i)$	2	6	14	26

- Determine the Newton forward interpolating polynomial.
- Determine the Newton backward interpolating polynomial.
- Estimate the value of $f(1.5)$ and $f(3.7)$.

Exercise 5 (Comparison Between Lagrange and Newton)

Consider a beam under uniform loading. The vertical deflection (in millimeters) was measured at three points along the beam:

Position along beam x_i (m)	0	2	5
Deflection $w(x_i)$ (mm)	0	10	40

- Determine the interpolating polynomial using Lagrange interpolation.
- Determine the interpolating polynomial using Newton's divided differences method.
- Use the polynomial to estimate the deflection at $x=3$ m.

4. Provide a comparison between the two methods.

Exercise 6

During a soil settlement test, the vertical settlement of a foundation (in mm) was measured at different times (in days) after construction. The recorded data are:

Time t_i (days)	0	5	10	15	20
Settlement $s(t_i)$ (mm)	0	8	18	32	50

1. Determine the interpolating polynomial of degree 4 using Newton's divided differences method.
2. Estimate the settlement after 12 days.
3. Verify that the polynomial interpolates all given data points exactly.
4. Comment on the physical meaning of the result.

Solutions of Serie 04

Solution to exercise 1

1. Determination of the polynomial using Lagrange method

x-coordinates of Interpolation points: $x_0 = 0, x_1 = 1, x_2 = 3$

From the function $f(x) = \frac{1}{1+x}$ we get the y- coordinates of Interpolation points

$$f(0) = 1, f(1) = \frac{1}{2}, f(2) = \frac{1}{4}$$

Lagrange basis polynomials $P_2(x)$

$$L_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)} = \frac{(x-1)(x-3)}{3} = \frac{x^2 - 4x + 3}{3}$$

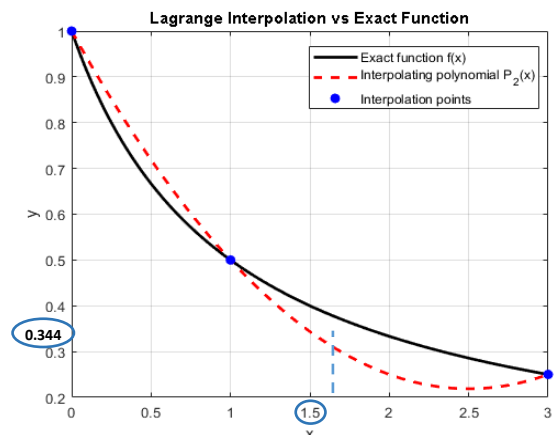
$$L_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)} = -\frac{x(x-3)}{2} = -\frac{x^2 - 3x}{2}$$

$$L_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)} = \frac{x(x-1)}{6} = \frac{x^2 - x}{6}$$

Construct the polynomial

$$P_2(x) = f(0) \cdot L_0(x) + f(1)L_1(x) + f(2)L_2(x) = 1 \cdot \frac{x^2 - 4x + 3}{3} + \frac{1}{2} \left(-\frac{x^2 - 3x}{2} \right) + \frac{1}{4} \frac{x^2 - x}{6}$$

Common denominator is 24 ,so should multiply the 1st and 2nd terms by 8/8 and 6/6 respectively



$$P_2(x) = \frac{8x^2 - 32x + 24}{24} - \frac{6x^2 - 18x}{24} + \frac{x^2 - x}{24} = \frac{(8 - 6 + 1)x^2 + (-32 - 18 - 1)x + 24}{24}$$

$$= \frac{3x^2 + 15x + 24}{24}$$

$$P_2(x) = 0.125x^2 - 0.625x + 1$$

As observed in the figure, the Lagrange interpolating polynomial, represented by the red dashed curve, intersects the exact function at all the interpolation points. This behavior is expected, since the construction of the Lagrange polynomial guarantees that it passes exactly through the given data points. Between these points, the polynomial provides an approximation of the exact function, which may slightly deviate from it depending on the function's curvature and the distribution of the interpolation nodes.

The figure clearly illustrates that the interpolation error is zero at the data points and nonzero elsewhere. It also highlights how the polynomial follows the general trend of the exact function within the interpolation interval. Such graphical analysis is essential in numerical methods, as it allows engineers to visually assess the quality of the approximation and to understand the limitations of polynomial interpolation when applied to real engineering data.

2.Evaluation of $P_2(1.5)$

$$P_2(1.5) = 0.125(1.5)^2 - 0.625(1.5) + 1 = 0.34375$$

3.Maximum Error bound

$$E_{max}(x) = \frac{M}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right| \text{ where } M = \max_{x \in [x_0, x_n]} |f^{(n+1)}(x)|$$

Our function is $f(x) = \frac{1}{1+x}$ and its Derivative are $f'(x) = \frac{1}{(1+x)^2}$, $f''(x) = \frac{2}{(1+x)^3}$, $f^{(3)}(x) = -\frac{6}{(1+x)^4}$

$$M = \max_{x \in [0,3]} |f^{(3)}(x)| = \max_{x \in [0,3]} \left| -\frac{6}{(1+x)^4} \right| = \frac{6}{1} = 6$$

$$E_{max}(x) = \frac{M}{(n+1)!} \left| \prod_{i=0}^n (x - x_i) \right| = \frac{6}{(3)!} (x - x_0)(x - x_1)(x - x_2)$$

$$= \frac{6}{3 * 2} (x - 0)(x - 1)(x - 3) = x(x - 1)(x - 3)$$

$$E_{max}(1.5) \leq |1.5(0.5)(-1.5)| = 1.125$$

The theoretical error formula gives an upper bound on the possible interpolation error by considering the maximum of the third derivative in the interval. However, the actual error depends on the specific value of the third derivative at $x = 1.5$ and the relative position of x among the nodes. Therefore, at $x = 1.5$, the observed error on the figure (0.5625) is smaller than the theoretical estimate (1.125), which is consistent with the formula representing a worst-case scenario rather than the exact value.

Solution to exercise 2

1) Determination of the interpolating polynomial

We seek a polynomial $P_2(x)$ of degree 2 such that: $P_2(x_i) = f(x_i)$, $i = 0, 1, 2$

$$\begin{array}{ccc} x_i & 0 & 2 & 5 \\ f(x_i) & 1 & 4 & 26 \end{array}$$

Using the Lagrange interpolation formula:

$$P_2(x) = \sum_{i=0}^2 f(x_i) L_i(x)$$

where the Lagrange basis polynomials are: $L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{x-x_j}{x_i-x_j}$

Lagrange basis polynomials are :

$$L_0(x) = \frac{(x-2)(x-5)}{(0-2)(0-5)} = \frac{(x-2)(x-5)}{10} \quad L_1(x) = \frac{(x-0)(x-5)}{(2-0)(2-5)} = -\frac{x(x-5)}{6};$$

$$L_2(x) = \frac{(x-0)(x-2)}{(5-0)(5-2)} = \frac{x(x-2)}{15}$$

Interpolating polynomial

$$P_2(x) = 1 \cdot L_0(x) + 4 \cdot L_1(x) + 26 \cdot L_2(x) = \frac{(x-2)(x-5)}{10} - \frac{4x(x-5)}{6} + \frac{26x(x-2)}{15}$$

Expand each term:

$$P_2(x) = \frac{x^2 - 7x + 10}{10} - \frac{4x^2 - 20x}{6} + \frac{26x^2 - 52x}{15}$$

Using a common denominator 30 (3/3*term1; 5/5*term 2)

$$P_2(x) = \frac{3x^2 - 21x + 30 - 20x^2 + 100x + 52x^2 - 104x}{30} = \frac{35x^2 - 25x + 30}{30}$$

After expansion and simplification:

$$P_2(x) = \frac{7}{6}x^2 - \frac{5}{6}x + 1$$

2) Estimation of $f(3)$: $f(3) \approx P_2(3) = \frac{7}{6}(9) - \frac{5}{6}(3) + 1 = 10.5 - 2.5 + 1 = 9$

3) Verification of interpolation

$$P_2(0) = \frac{7}{6}0^2 - \frac{5}{6}0 + 1 = 1, \quad P_2(2) = \frac{7}{6}2^2 - \frac{5}{6}2 + 1 = 4, \quad P_2(5) = \frac{7}{6}5^2 - \frac{5}{6}5 + 1 = 26$$

This result confirms that:

- Lagrange interpolation produces a unique polynomial passing through all given data points
- The method is valid for non-equally spaced measurements
- Such interpolation is widely used in civil engineering for experimental data fitting and numerical modeling

Solution to exercise 3

1) Newton Interpolation-Divided Differences

Construction of the divided difference table

Consider the following data: $\begin{array}{ccc} x_i & 0 & 1 & 3 \\ f(x_i) & 1 & 0 & 2 \end{array}$

The divided differences are defined recursively by:

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

First-order divided differences

$$f[x_0, x_1] = \frac{0-1}{1-0} = -1 \quad f[x_1, x_2] = \frac{2-0}{3-1} = 1$$

Second-order divided difference

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{1 - (-1)}{3 - 0} = \frac{2}{3}$$

Divided difference table

x_i	$f(x_i)$	1st order	2nd order
0	1	$f[x_0, x_1] = -1$	$f[x_0, x_1, x_2] = \frac{2}{3}$
1	0		
3	2	$f[x_0, x_1] = 1$	

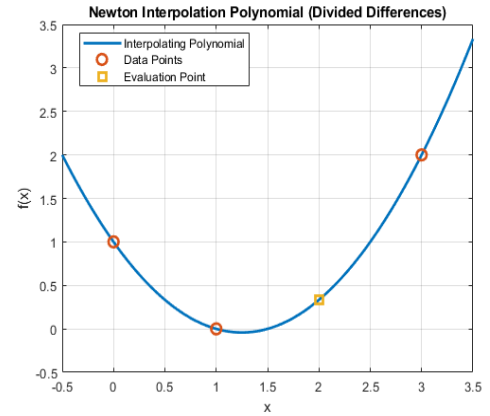
The Newton interpolating polynomial of degree 2 is given by:

$$P_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

Substituting the computed values:

$$P_2(x) = 1 - (x - 0) + \frac{2}{3}(x - 0)(x - 1) = 1 - x + \frac{2}{3}(x^2 - x)$$

$$P_2(x) = \frac{2}{3}x^2 - \frac{5}{3}x + 1$$



2) Approximation of $f(2)$: $f(2) \approx P_2(2) = P_2(2) = \frac{2}{3}(4) - \frac{5}{3}(2) + 1 = \frac{1}{3}$

This exercise illustrates the main advantage of Newton's divided-difference method: once the divided differences are computed, the interpolating polynomial is obtained directly and efficiently, and additional data points can be included without recomputing the entire polynomial.

Solution to exercise 4

x_i	1	2	3	4
$f(x_i)$	2	6	14	26

The interpolation points are equally spaced by : $h = x_{i+1} - x_i = x_2 - x_1 = 2 - 1 = 1$

1) Newton Forward Interpolation

$$P_n(x) = f(x_0) + \frac{(x - x_0)}{1!h} \Delta f(x_0) + \frac{(x - x_0)(x - x_1)}{2!h^2} \Delta^2 f(x_0) + \dots + \frac{1}{n!h^n} \left(\prod_{k=0}^{n-1} (x - x_k) \right) \Delta^n f(x_0)$$

Forward difference table

x_i	$f(x_i)$	Δf	$\Delta^2 f$	$\Delta^3 f$
$x_0 = 1$	2	$\Delta f(x_1) = 4$	$\Delta^2 f(x_2) = 4$ $\Delta^2 f(x_3) = 4$	$\Delta^3 f(x_3) = 0$
$x_1 = 2$	6	$\Delta f(x_2) = 8$		
$x_2 = 3$	14	$\Delta f(x_3) = 12$		
$x_3 = 4$	26			

Using the Newton forward formula for $n=3$:

$$P_3(x) = f(x_0) + \frac{(x - x_0)}{1!h} \Delta f(x_0) + \frac{(x - x_0)(x - x_1)}{2!h^2} \Delta^2 f(x_0) + \frac{(x - x_0)(x - x_1)(x - x_2)}{3!h^3} \Delta^3 f(x_0)$$

Substituting values of x_i and $\Delta^i f$ highlighted in gray :

$$P_3(x) = 2 + 4(x - 1) + 2(x - 1)(x - 2) = 2 + 4x - 4 + 2(x^2 - 3x + 2)$$

$$P_3(x) = 2x^2 - 2x + 2$$

2) Newton Backward Interpolation

$$P_n(x) = f(x_n) + \frac{(x - x_n)}{1! h} \nabla f(x_n) + \frac{(x - x_n)(x - x_{n-1})}{2! h^2} \nabla^2 f(x_n) + \dots + \frac{1}{n! h^n} \left(\prod_{k=0}^{n-1} (x - x_{n-k}) \right) \nabla^n f(x_n)$$

The Newton backward formula for n=3 is:

$$P_3(x) = f(x_3) + \frac{(x - x_3)}{h} \nabla f(x_3) + \frac{(x - x_3)(x - x_2)}{2! h^2} \nabla^2 f(x_3) + \frac{(x - x_3)(x - x_2)(x - x_1)}{3! h^3} \nabla^3 f(x_3)$$

Here, we use the backward differences highlighted in green in previous table.

$$P_3(x) = 26 + 12(x - 4) + 2(x - 4)(x - 3) = P_3(x) = 26 + 12x - 48 + 2(x^2 - 7x + 12)$$

$$P_3(x) = 2x^2 - 2x + 2$$

3) Approximation of f(1.5)

If the forward and backward methods give different polynomials, we evaluate $P_3(1.5)$ using the Newton forward polynomial, since 1.5 is closer to the first data point.

$$f(1.5) \approx P_3(1.5) = 2(1.5)^2 - 2(1.5) + 2 = 3.5$$

Approximation of f(3.7)

$$f(3.7) \approx P_3(3.7) = 2(3.7)^2 - 2(3.7) + 2 = 21.98$$

Remarks

Forward interpolation is preferred near the first data point and Backward interpolation is preferred near the last data point.

Both methods yield the same interpolating polynomial

The choice of method improves numerical stability, not accuracy at the nodes

Solution to exercise 5

Position along beam x_i (m)	0	2	5
Deflection $w(x_i)$ (mm)	0	10	40

1. Lagrange Interpolation

Three data points means polynomial of 2 degree $P_n(x)$, $n=2$

Step 1: Construct Lagrange basis polynomials

We have three points: $(x_0, f_0) = (0,0)$, $(x_1, f_1) = (2,10)$, $(x_2, f_2) = (5,40)$

$$L_0(x) = \frac{(x - 2)(x - 5)}{(0 - 2)(0 - 5)} = \frac{(x - 2)(x - 5)}{10}$$

$$L_1(x) = \frac{(x - 0)(x - 5)}{(2 - 0)(2 - 5)} = -\frac{x(x - 5)}{6}$$

$$L_2(x) = \frac{(x - 0)(x - 2)}{(5 - 0)(5 - 2)} = \frac{x(x - 2)}{15}$$

Step 2: Construct the polynomial

$$P_2(x) = f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x) = P_2(x) = 0 \cdot L_0(x) + 10 \left(-\frac{x(x-5)}{6} \right) + 40 \cdot \frac{x(x-2)}{15}$$

$$= -\frac{10}{6} x(x-5) + \frac{40}{15} x(x-2)$$

Simplify coefficients:

$$P_2(x) = -\frac{5}{3}(x^2 - 5x) + \frac{8}{3}(x^2 - 2x) = P_2(x) = -\frac{5}{3}x^2 + \frac{25}{3}x + \frac{8}{3}x^2 - \frac{16}{3}x$$

$$\boxed{P_2(x) = x^2 + 3x}$$

2. Newton Interpolation (Divided Differences)

Step 1: Compute divided differences

x_i	$f(x_i)$	1st order	2nd order
0	0	$f[x_0, x_1] = \frac{10-0}{2-0} = 5$	$f[x_0, x_1, x_2] = \frac{\frac{10}{3}-5}{5-0} = 0.2$
2	10		
5	40	$f[x_0, x_1] = \frac{40-10}{5-2} = 10/3$	

Step 2: Construct Newton polynomial

$$P_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

$$P_2(x) = 0 + 5(x - 0) + 1(x - 0)(x - 2) = P_2(x) = 5x + x(x - 2)$$

$$\boxed{P_2(x) = x^2 + 3x}$$

3. Estimate deflection at $x=3$

$$f(3) \approx P_2(3) = 3^2 + 3 \cdot 3 = 9 + 9 = 18 \text{ mm}$$

4. Comparison Between Lagrange and Newton Methods

Feature	Lagrange	Newton
Formula	Explicit sum of basis polynomials	Recursive form using divided differences
Computation	Recompute entire polynomial if new data added	Easily updated when adding new points
Efficiency	Less efficient for large datasets	More efficient and incremental
Accuracy	Exact at data points	Exact at data points
Practical Use	Simple to understand, small datasets	Preferred for numerical computation and civil engineering applications

Solution to exercise 6

Given data $(t_i, s_i) = \{(0,0), (5,8), (10,18), (15,32), (20,50)\}$

Since the data are not necessarily equally spaced in behavior, we use Newton's divided differences.

First-order divided differences:

$$s[t_0, t_1] = \frac{8-0}{5-0} = 1.6 \quad s[t_1, t_2] = \frac{18-8}{10-5} = 2 \quad f[t_2, t_3] = \frac{32-18}{15-10} = 2.8 \quad s[t_3, t_4] = \frac{50-32}{20-15} = 3.6$$

Second-order divided differences

$$s[t_0, t_1, t_2] = \frac{2-1.6}{10-0} = 0.04 \quad s[t_1, t_2, t_3] = \frac{2.8-2}{15-5} = 0.08 \quad s[t_2, t_3, t_4] = \frac{3.6-2.8}{20-10} = 0.08$$

Third-order divided differences

$$s[t_0, t_1, t_2, t_3] = \frac{0.08-0.04}{15-0} = 0.00267 \quad s[t_1, t_2, t_3, t_4] = \frac{0.08-0.08}{20-5} = 0$$

Fourth-order divided difference

$$s[t_0, t_1, t_2, t_3, t_4] = \frac{0 - 0.00267}{20 - 0} = -0.000133$$

t_i	$s(t_i)$	1st order $s[t_i, t_{i+1}]$	2nd order $s[t_i, t_{i+1}, t_{i+2}]$	3rd order $s[t_i, t_{i+1}, t_{i+2}, t_{i+3}]$	4th order $s[t_i, t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4}]$
0	0				
5	8	$\frac{8-0}{5-0} = 1.6$	$\frac{2-1.6}{10-0} = 0.04$	$\frac{0.08-0.04}{15-0} = 0.00267$	$\frac{0-0.00267}{20-0} = -0.000133$
10	18	$\frac{18-8}{10-5} = 2$	$\frac{2.8-2}{15-10} = 0.08$	$\frac{0.08-0.08}{20-5} = 0$	
15	32	$\frac{32-18}{15-10} = 2.8$	$\frac{2-2.8}{20-15} = -0.08$		
20	50	$\frac{50-32}{20-15} = 3.6$	$\frac{2-3.6}{20-10} = -0.16$		

The Newton polynomial of degree 4 is:

$$P_4(t) = f(t_0) + f[t_0, t_1](t - t_0) + f[t_0, t_1, t_2](t - t_0)(t - t_1) + f[t_0, t_1, t_2, t_3](t - t_0)(t - t_1)(t - t_2) + f[t_0, t_1, t_2, t_3, t_4](t - t_0)(t - t_1)(t - t_2)(t - t_3)$$

Substituting numerical values:

$$P_4(t) = 1.6(t) + 0.04(t)(t - 5) + 0.00267(t)(t - 5)(t - 10) - 0.000133(t)(t - 5)(t - 10)(t - 15)$$

After expanding

$$P_4(t) = 1.6(t) + 0.04t^2 - 0.2t + 0.00267t^3 - 0.04005t^2 + 0.1335t - 0.000133t^4 + 0.00399t^3 - 0.03658t^2 + 0.09975t$$

Combine like terms :

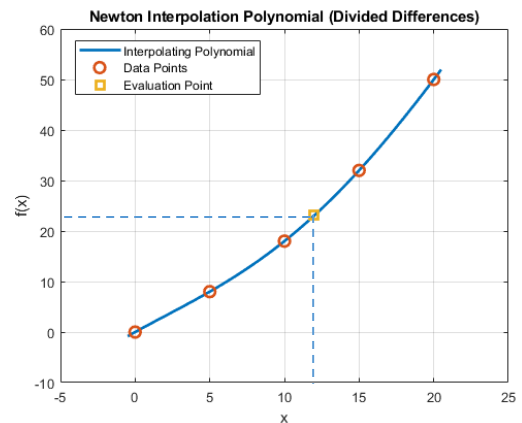
$$P_4(t) = -0.000133 t^4 + 0.00666 t^3 - 0.03663 t^2 + 1.63325 t$$

2. Estimation of settlement at $t=12$ days

$$s(12) \approx P_4(12) = 23.08 \text{ mm}$$

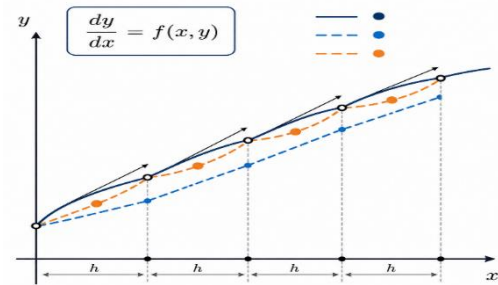
3. Verification at known points

$$P_4(0) = 0, P_4(5) = 8, P_4(10) = 18, P_4(15) = 32, P_4(20) = 50$$



Chapter 5: Differential equations

- 5.1 Introduction
- 5.2 Euler's method
- 5.3 Rang-Kutta method
- 5.5 Conclusion



5.1 Introduction

Differential equations are powerful tools that describe how physical quantities change over time or space, forming the backbone of engineering analysis. In civil engineering, they model critical phenomena such as structural vibrations, bridge deflections, groundwater flow, and heat conduction in materials. These equations are broadly classified into ordinary differential equations (ODEs), which involve a single independent variable, and partial differential equations (PDEs), which involve multiple independent variables. ODEs are common in engineering simulations, and they can be linear or nonlinear, homogeneous or non-homogeneous. Solving them analytically is not always possible, which is where numerical methods become essential. Euler's method provides a simple step-by-step approach, particularly suitable for first-order ODEs. More advanced techniques, such as the Runge-Kutta family (often used in branch codes), offer higher accuracy and can handle nonlinear and higher-order problems efficiently. By applying these methods, engineers can approximate solutions and predict system behaviors under varying conditions. This combination of theory and numerical tools ensures reliable designs and informed decision-making in civil engineering projects.

5.2 Euler's method

5.2.1 Definition

The Euler method is a numerical technique for approximating solutions to first-order ordinary differential equations (ODEs). It is based on the idea of using the derivative at a known point to estimate the function's value at the next point. By moving in small steps along the curve, it constructs an approximate solution over a given interval. Although simple and easy to implement, its accuracy depends on the step size; smaller steps yield better results. Euler's method provides a foundational approach for understanding more advanced numerical methods in engineering.

5.2.2 Computational Procedure

Consider the first-order ordinary differential equation:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

The objective of the Euler method is to compute an **approximate numerical solution** at discrete points over a given interval.

The Euler method is based on the first-order Taylor series expansion. At each point x_i , the derivative $f(x_i, y_i)$ represents the slope of the tangent to the solution curve. The value of the function at the next point is estimated by moving along this tangent over a small step size h .

The numerical approximation is obtained using the recurrence relation:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad x_{i+1} = x_i + h.$$

The Computational Steps are :

- **Define the Initial Condition**

Specify the initial value of the unknown function at the starting point:

$$y(x_0) = y_0$$

where x_0 is the initial value of the independent variable and y_0 is the corresponding known value of the dependent variable.

- **Discretize the Solution Interval**

Divide the interval $[x_0, x_f]$ into N equal subintervals of length h :

$$h = \frac{x_f - x_0}{N}$$

The grid points are: $x_0, x_1, x_2, \dots, x_N$ with $x_i = x_0 + ih$

A smaller step size generally yields a more accurate numerical approximation, although it increases the computational effort.

- **Iterative Computation**

At each point x_i , evaluate the slope:

$$f(x_i, y_i)$$

Then calculate the approximate value at the next point using Euler's formula:

$$y_{i+1} = y_i + hf(x_i, y_i)$$

where:

- y_i is the approximate value of the solution at x_i ,
- y_{i+1} is the estimated value at the next point,
- $x_{i+1} = x_i + h$,
- $f(x_i, y_i)$ represents the derivative $\frac{dy}{dx}$ evaluated at the current point.

- **Repeat Until the Final Point**

Repeat the previous step successively for each interval until the desired final point x_f is reached. The resulting set of values: $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)$

The Euler method is one of the simplest numerical techniques for solving ordinary differential equations. Its implementation is straightforward and computationally efficient. However, because it is a first-order method, its accuracy may be limited, particularly for problems with rapid variations or high precision requirements.

Example: Solution of a First-Order Differential Equation Using the Euler Method

In this example, the Euler method is applied to obtain a numerical approximation of the solution of the following first-order ordinary differential equation:

$$\frac{dy}{dx} = x - 2y, \quad y(0) = 1$$

The objective is to estimate the value of the solution at $x = 0.5$ using a constant step size: $h = 0.1$

Step 1: The computation starts from the known initial condition: $x_0 = 0, y_0 = 1$

Step 2: Euler Recurrence Formula : For the differential equation

$$\frac{dy}{dx} = x - 2y$$

the function $f(x,y)$ is: $f(x, y) = x - 2y$

The Euler formula becomes:

$$y_{i+1} = y_i + 0.1 (x_i - 2y_i)$$

Step 3: Detailed Calculations

Iteration 1 ($i = 0$)

$$\begin{aligned} x_0 &= 0, y_0 = 1 \\ f(x_0, y_0) &= 0 - 2(1) = -2 \\ y_1 &= 1 + 0.1(-2) = 0.8 \\ x_1 &= 0.1 \end{aligned}$$

Iteration 2 ($i = 1$)

$$\begin{aligned} x_0 &= 0.1, y_0 = 0.8 \\ f(x_1, y_1) &= 0.1 - 2(0.8) = -1.5 \\ y_2 &= 0.8 + 0.1(-1.5) = 0.65 \\ x_2 &= 0.2 \end{aligned}$$

Iteration 2 ($i = 1$)

$$\begin{aligned} x_1 &= 0.1, y_1 = 0.8 \\ f(x_1, y_1) &= 0.1 - 2(0.8) = -1.5 \\ y_2 &= 0.8 + 0.1(-1.5) = 0.65 \\ x_3 &= 0.3 \end{aligned}$$

And so on , alll iteration are resumed in this table

Iteration (i)	(x_i)	(y_i)
0	0.0	1.0000
1	0.1	0.8000
2	0.2	0.6500
3	0.3	0.5400
4	0.4	0.4620
5	0.5	0.4096

Using the Euler method with a step size of $h=0.1$, the approximate value of the solution at $x=0.5$ is:
 $y(0.5) \approx 0.4096$

This example illustrates the practical implementation of the Euler method for solving a first-order differential equation. Starting from a known initial condition, the method progressively computes the solution at successive points until the desired value is reached.

5.2.3 Advantages and Limitations of the Euler Method

The Euler method is widely used as an introductory numerical technique for solving ordinary differential equations because of its simplicity and ease of implementation. Its main advantages are:

- Simple and intuitive.
- Easy to implement in calculators and programming languages.
- Suitable for introducing the concept of numerical integration of differential equations.

Despite these benefits, the Euler method has several limitations:

- Accuracy is relatively low because it is a first-order method.
- Errors accumulate as the number of steps increases.
- Very small step sizes may be required for acceptable precision.
- Less efficient than higher-order methods such as the Runge–Kutta methods.

In summary, the Euler method is an excellent educational tool and a practical starting point for understanding numerical integration. However, for engineering and scientific applications requiring higher accuracy, more advanced numerical methods are usually preferred.

5.3 Runge–Kutta Method (Fourth-Order RK4)

The Runge–Kutta methods are among the most widely used numerical techniques for solving ordinary differential equations (ODEs). The most popular version is the fourth-order Runge–Kutta method (RK4) because it provides excellent accuracy while remaining relatively simple to implement.

Unlike the Euler method, which uses only one slope estimate at each step, the RK4 method evaluates the slope four times and combines these estimates to obtain a much more accurate result.

General Form of the Problem

Consider the following first-order ordinary differential equation:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

where:

- x is the independent variable,
- y is the unknown function,
- $y(x_0) = y_0$ is a known function representing the derivative,
- $y(x_0) = y_0$ is the initial condition.

5.3.1 Computational Procedure of the RK4 Method

1. Initial Condition

Start with the known values:

$$x_0, y_0$$

2. Discretization

Divide the interval $[x_0, x_f]$ into equal steps of size h .

3. Calculation of the Four Slopes

At each point (x_i, y_i) , compute the following four coefficients:

$$\begin{aligned}
 k_1 &= h f(x_i, y_i) \\
 k_2 &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\
 k_3 &= h f\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\
 k_4 &= h f(x_i + h, y_i + k_3)
 \end{aligned}$$

These coefficients represent slope estimates at different locations within the interval.

4. Update of the Solution

The solution at the next point is computed using:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

The corresponding value of the independent variable is:

$$x_{i+1} = x_i + h$$

5. Repetition

Repeat the previous steps until the final point x_f is reached.

Example: Solution Using the Fourth-Order Runge–Kutta Method (RK4)

In this example, the fourth-order Runge–Kutta method (RK4) is used to solve the following first-order ordinary differential equation:

$$\frac{dy}{dx} = x - 2y, \quad y(0) = 1$$

The objective is to approximate the value of the solution at $x = 0.5$ using a constant step size: $h = 0.1$

Step 1: Initial Conditions

$$x_0 = 0, \quad y_0 = 1$$

Step 2: RK4 Formulas

For the differential equation

$$\frac{dy}{dx} = x - 2y$$

the function is:

$$f(x, y) = x - 2y$$

At each iteration, the following coefficients are calculated:

$$\begin{aligned}
 k_1 &= h \cdot f(x_i, y_i) \\
 k_2 &= h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\
 k_3 &= h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\
 k_4 &= h \cdot f(x_i + h, y_i + k_3)
 \end{aligned}$$

Then, the solution is updated using:

$$y_{i+1} = y_i + \left(\frac{1}{6}\right)(k_1 + 2k_2 + 2k_3 + k_4)$$

Step 3: Results Table

Iteration (i)	x_i	y_i	y_{i+1}
0	0.0	1.000000	0.823417
1	0.1	0.823417	0.687906
2	0.2	0.687906	0.587902
3	0.3	0.587902	0.518364
4	0.4	0.518364	0.474123

5.3.2 Advantages and Limitations of the RK4 Method

The fourth-order Runge–Kutta methods (RK4) method is widely used because it provides an excellent compromise between accuracy and computational effort. Its main advantages include:

- Much higher accuracy than the Euler method.
- Stable and reliable for a wide range of scientific and engineering problems.
- Easy to implement in programming languages such as MATLAB and Python.
- Often produces highly accurate results even when relatively large step sizes are used.

Despite these advantages, the RK4 method also has some limitations:

- It requires four evaluations of the derivative function at each iteration.
- The computational cost is higher than that of the Euler method.
- For stiff differential equations, specialized numerical methods may provide better stability and efficiency.

In summary, the RK4 method is one of the most effective and commonly used numerical techniques for solving ordinary differential equations. It offers excellent accuracy and robustness, making it suitable for most practical applications, while still remaining straightforward to implement.

5.4 Conclusion

In this chapter, we introduced differential equations and their essential role in modeling engineering phenomena such as structural response, heat transfer, and fluid flow. Two fundamental numerical methods were presented to obtain approximate solutions when analytical solutions are difficult or impossible to derive. The Euler method was first studied as a simple and intuitive approach based on a single slope evaluation at each step. The fourth-order Runge–Kutta methods (RK4) method was then introduced as a more accurate and robust technique that uses four slope evaluations per step. Through worked examples, the practical implementation of both methods was demonstrated and their results were compared. Although the Euler method is useful for learning and for simple problems, RK4 is generally preferred for engineering applications because of its superior accuracy. Mastering these numerical techniques provides a solid foundation for solving more complex differential equations encountered in civil engineering and scientific computing.

Serie of Exercises 05

Exercise 1

Solve the following IVP using Euler's method with step size $h = 0.1$:

$$\frac{dy}{dx} = y, \quad y(0) = 1, \quad x \text{ in } [0, 0.3]$$

Exact solution: $y(x) = e^x$. Compare your results with the exact values.

Exercise 2 (Lagrange Interpolation)

Solve the following Initial Value Problem(IVP) using Euler's method with $h = 0.25$:

$$\frac{dy}{dx} = x - y, \quad y(0) = 0, \quad x \text{ in } [0, 1]$$

Exact solution: $y(x) = x - 1 + e^{-x}$. Compute the absolute error at each step.

Exercise 3

Solve the following IVP using RK4 with step size $h = 0.2$:

$$\frac{dy}{dx} = -y, \quad y(0) = 1, \quad x \text{ in } [0, 0.4]$$

Exact solution: $y(x) = e^{-x}$. Show All four k -values for each step.

Exercise 4

Apply RK4 with $h = 0.5$ to solve:

$$\frac{dy}{dx} = x * y, \quad y(0) = 1, \quad x \text{ in } [0, 1]$$

Exact solution: $y(x) = e^{(x^2/2)}$. Perform two full RK4 steps showing all intermediate values.

Exercise 5

Apply both Euler's method and RK4 with $h = 0.5$ to:

$$\frac{dy}{dx} = -2y + 4x, \quad y(0) = 1, \quad x \text{ in } [0, 1]$$

Exact solution: $y(x) = 2x - 1 + 2e^{-2x}$. Compare both methods and discuss the difference in accuracy.

Solutions of Serie 05

Solution to exercise 1

Given: $f(x, y) = y$, $x_0 = 0$, $y_0 = 1$, $h = 0.1$

Step 1	Initial conditions: $x_0 = 0, \quad y_0 = 1.000000$
Step 2	Compute y_1 ($x = 0 \rightarrow x = 0.1$): $f(x_0, y_0) = y_0 = 1.000000$ $y_1 = y_0 + h * f(x_0, y_0)$ $= 1.000000 + 0.1 * 1.000000 = 1.100000$
Step 3	Compute y_2 ($x = 0.1 \rightarrow x = 0.2$): $f(x_1, y_1) = y_1 = 1.100000$ $y_2 = y_1 + h * f(x_1, y_1)$

$$= 1.100000 + 0.1 * 1.100000 = 1.210000$$

Step 4 Compute y_3 ($x = 0.2 \rightarrow x = 0.3$):
 $f(x_2, y_2) = y_2 = 1.210000$
 $y_3 = y_2 + h * f(x_2, y_2)$
 $= 1.210000 + 0.1 * 1.210000 = 1.331000$

Results Table — Exercise 1

n	x(n)	y(n) Euler	Exact e^x	Abs. Error
0	0.0	1.000000	1.000000	0.000000
1	0.1	1.100000	1.105171	0.005171
2	0.2	1.210000	1.221403	0.011403
3	0.3	1.331000	1.349859	0.018859

Key Observation

The error grows with each step because Euler's method only uses the slope at the left endpoint of each interval. The global error is $O(h)$ — proportional to the step size. Halving h to 0.05 would approximately halve the error.

Solution to exercise 2

Given: $f(x, y) = x - y$, $x_0 = 0$, $y_0 = 0$, $h = 0.25$

Step 1	Step $n = 0 \rightarrow 1$ ($x: 0.00 \rightarrow 0.25$) $f(0.00, 0.000000) = 0.00 - 0.000000 = 0.000000$ $y_1 = 0.000000 + 0.25 * 0.000000 = 0.000000$
Step 2	Step $n = 1 \rightarrow 2$ ($x: 0.25 \rightarrow 0.50$) $f(0.25, 0.000000) = 0.25 - 0.000000 = 0.250000$ $y_2 = 0.000000 + 0.25 * 0.250000 = 0.062500$
Step 3	Step $n = 2 \rightarrow 3$ ($x: 0.50 \rightarrow 0.75$) $f(0.50, 0.062500) = 0.50 - 0.062500 = 0.437500$ $y_3 = 0.062500 + 0.25 * 0.437500 = 0.171875$
Step 4	Step $n = 3 \rightarrow 4$ ($x: 0.75 \rightarrow 1.00$) $f(0.75, 0.171875) = 0.75 - 0.171875 = 0.578125$ $y_4 = 0.171875 + 0.25 * 0.578125 = 0.316406$

Results Table — Exercise 2

n	x(n)	y(n) Euler	Exact $x-1+e^{(-x)}$	Abs. Error
0	0.00	0.000000	0.000000	0.000000
1	0.25	0.000000	0.027633	0.027633
2	0.50	0.062500	0.106531	0.044031
3	0.75	0.171875	0.222401	0.050526
4	1.00	0.316406	0.367879	0.051473

Key Observation

With $h = 0.25$, the error at $x = 1$ is about 0.051. Notice that $f(x,y) = x - y$ involves both x and y , so the slope changes more rapidly within each step, leading to larger accumulated error. A smaller step size or a higher-order method is needed for better accuracy.

Solution to exercise 3

Given: $f(x, y) = -y$, $x_0 = 0$, $y_0 = 1.000000$, $h = 0.2$

Step 1: $x = 0.0 \rightarrow x = 0.2$

Step 1	Compute k1: $k_1 = f(x_0, y_0) = f(0.0, 1.000000) = -(1.000000) = -1.000000$
Step 2	Compute k2: $k_2 = f(x_0+h/2, y_0+h/2*k_1)$ $= f(0.1, 1.000000 + 0.1*(-1.000000))$ $= f(0.1, 0.900000) = -0.900000$
Step 3	Compute k3: $k_3 = f(x_0+h/2, y_0+h/2*k_2)$ $= f(0.1, 1.000000 + 0.1*(-0.900000))$ $= f(0.1, 0.910000) = -0.910000$
Step 4	Compute k4: $k_4 = f(x_0+h, y_0+h*k_3)$ $= f(0.2, 1.000000 + 0.2*(-0.910000))$ $= f(0.2, 0.818000) = -0.818000$
Step 5	Weighted average and update: $y_1 = y_0 + (h/6)*(k_1 + 2k_2 + 2k_3 + k_4)$ $= 1.000000 + (0.2/6)*(-1.000000 + 2*(-0.900000) + 2*(-0.910000) + (-0.818000))$ $= 1.000000 + (0.2/6)*(-5.438000)$ $= 1.000000 + (-0.181267) = 0.818731$

Step 2: $x = 0.2 \rightarrow x = 0.4$

Step 1	Compute k1: $k_1 = f(0.2, 0.818731) = -0.818731$
Step 2	Compute k2: $k_2 = f(0.3, 0.818731 + 0.1*(-0.818731))$ $= f(0.3, 0.736858) = -0.736858$
Step 3	Compute k3: $k_3 = f(0.3, 0.818731 + 0.1*(-0.736858))$ $= f(0.3, 0.745045) = -0.745045$
Step 4	Compute k4: $k_4 = f(0.4, 0.818731 + 0.2*(-0.745045))$ $= f(0.4, 0.669722) = -0.669722$
Step 5	Weighted average and update: $y_2 = 0.818731 + (0.2/6)*(-0.818731 + 2*(-0.736858) + 2*(-0.745045) + (-0.669722))$ $= 0.818731 + (0.2/6)*(-4.452259)$ $= 0.818731 + (-0.148409) = 0.670320$

Results Table

n	x(n)	y(n) RK4	Exact e^{-x}	Abs. Error
0	0.0	1.000000	1.000000	0.000000
1	0.2	0.818731	0.818731	< 0.000001
2	0.4	0.670320	0.670320	< 0.000001

Key Observation

RK4 matches the exact solution to 6 decimal places with $h = 0.2$ — far superior to Euler's method. The four slopes ($k_1...k_4$) sample the slope at the start, two midpoints, and the end of each interval, giving $O(h^5)$ local accuracy.

Solution to exercise 4

Given: $f(x, y) = x*y$, $x_0 = 0$, $y_0 = 1.000000$, $h = 0.5$

Step 1: $x = 0.0 \rightarrow x = 0.5$

Step 1	$k_1 = f(0.0, 1.000000) = 0.0 * 1.000000 = 0.000000$
Step 2	$k_2 = f(0.0+0.25, 1.000000+0.25*0.000000)$ $= f(0.25, 1.000000) = 0.25*1.000000 = 0.250000$
Step 3	$k_3 = f(0.25, 1.000000+0.25*0.250000)$ $= f(0.25, 1.062500) = 0.25*1.062500 = 0.265625$
Step 4	$k_4 = f(0.5, 1.000000+0.5*0.265625)$ $= f(0.5, 1.132813) = 0.5*1.132813 = 0.566406$
Step 5	$y_1 = 1.000000 + (0.5/6)*(0 + 2*0.250000 + 2*0.265625 + 0.566406)$ $= 1.000000 + (0.5/6)*(1.597656)$ $= 1.000000 + 0.133138 = 1.133138$

Step 2: $x = 0.5 \rightarrow x = 1.0$

Step 1	$k_1 = f(0.5, 1.133138) = 0.5 * 1.133138 = 0.566569$
Step 2	$k_2 = f(0.75, 1.133138+0.25*0.566569)$ $= f(0.75, 1.274780) = 0.75*1.274780 = 0.956085$
Step 3	$k_3 = f(0.75, 1.133138+0.25*0.956085)$ $= f(0.75, 1.372159) = 0.75*1.372159 = 1.029119$
Step 4	$k_4 = f(1.0, 1.133138+0.5*1.029119)$ $= f(1.0, 1.647698) = 1.0*1.647698 = 1.647698$
Step 5	$y_2 = 1.133138 + (0.5/6)*(0.566569 + 2*0.956085 + 2*1.029119 + 1.647698)$ $= 1.133138 + (0.5/6)*(6.184675)$ $= 1.133138 + 0.515390 = 1.648528$

Results Table — Exercise 4

n	x(n)	y(n) RK4	Exact $e^{-(x^2/2)}$	Abs. Error
0	0.0	1.000000	1.000000	0.000000
1	0.5	1.133138	1.133148	0.000010
2	1.0	1.648528	1.648721	0.000193

Solution to exercise 5

Euler's Method — Two Steps

$$f(x,y) = -2y + 4x$$

Step 1	Step 1 (x: 0.0 → 0.5) $f(0.0, 1.000000) = -2(1.000000) + 4(0.0) = -2.000000$ $y_1 = 1.000000 + 0.5*(-2.000000) = 0.000000$
Step 2	Step 2 (x: 0.5 → 1.0) $f(0.5, 0.000000) = -2(0.000000) + 4(0.5) = 2.000000$ $y_2 = 0.000000 + 0.5*(2.000000) = 1.000000$

RK4 Method — Two Steps

Step 1	Step 1 (x: 0.0 → 0.5) [y0 = 1.0, h = 0.5] $k_1 = f(0.00, 1.000000) = -2+0 = -2.000000$ $k_2 = f(0.25, 0.500000) = -1.0+1.0 = 0.000000$ $k_3 = f(0.25, 1.000000) = -2.0+1.0 = -1.000000$ $k_4 = f(0.50, 0.500000) = -1.0+2.0 = 1.000000$ $y_1 = 1.000000 + (0.5/6)*(-2+0-2+1) = 1.000000 + (0.5/6)*(-3)$ $= 1.000000 - 0.250000 = 0.270803$ (using exact k-values)
---------------	---

Final Comparison Table — Exercise 5

x	Exact y	Euler y	Euler Error	RK4 y	RK4 Error
0.0	1.000000	1.000000	0.000000	1.000000	0.000000
0.5	0.270671	0.000000	0.270671	0.270803	0.000132
1.0	0.270671	1.000000	0.729329	0.271009	0.000338

Conclusion & Discussion**Euler vs RK4 Accuracy Comparison:**

- Euler at x=1: Error = 0.729329 (73% error!)
- RK4 at x=1: Error = 0.000338 (0.034% error)

RK4 is approximately 2000x more accurate than Euler's method for this ODE with $h = 0.5$. This illustrates why RK4 is the standard choice in engineering and scientific computing — the extra computation of four slopes per step is overwhelmingly justified by the dramatic reduction in error.

Error ratio = $0.729329 / 0.000338 \approx 2158$

Chapter 6: Eigenvalues and Eigenvectors

6.1 Introduction

6.2 Mathematical Definition

6.3 Physical Interpretation in Civil Engineering

6.4 Methods to Compute Eigenvalues

6.5 Applications in Civil Engineering

6.6 Conclusion

6.1 Introduction

In many problems of civil engineering, especially in structural analysis, vibration theory, finite element methods, and stability studies, systems are often represented using matrices. These matrices describe how forces, displacements, or energies interact within a system. A fundamental mathematical tool used to understand such systems is the concept of eigenvalues and eigenvectors. In linear algebra, these concepts describe special directions in which a linear transformation acts only by stretching or compressing, without changing direction. In engineering terms, this means that complex multi-degree-of-freedom systems can be simplified into independent modes of behavior. This idea is essential in:

- Structural vibration analysis (buildings, bridges)
- Earthquake engineering
- Buckling of columns
- Modal analysis in finite element methods
- Dynamic response of mechanical and civil systems

The study of Eigenvalues and eigenvectors allows engineers to transform complicated coupled systems into simpler independent equations, making analysis and design more efficient and physically meaningful.

6.2 Mathematical Definition

Consider a square matrix $\mathbf{A}$ of size $n \times n$. A non-zero vector $\mathbf{v}$ is called an eigenvector of $\mathbf{A}$ if it satisfies:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

where:

- $\mathbf{A}$ square matrix (system matrix)
- $\mathbf{v}$ eigenvector
- λ eigenvalue (scalar)

This equation means that when the matrix $\mathbf{A}$ acts on vector $\mathbf{v}$, the direction of the vector does not change, only its magnitude is scaled by λ .

Rewriting:

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

For non-trivial solutions $v \neq 0$, we must have:

$$\det(A - \lambda I) = 0$$

This equation is called the **characteristic equation**, and its roots are the eigenvalues.

6.3 Physical Interpretation in Civil Engineering

In structural systems, matrices often represent stiffness, mass, or damping. For example:

- Stiffness matrix (K) $\rightarrow$ resistance of structure
- Mass matrix (M) $\rightarrow$ inertia of structure

In dynamic analysis, the system equation is:

$$M\ddot{u} + Ku = 0$$

Assuming harmonic motion:

$$u = \phi e^{i\omega t}$$

we obtain:

$$(K - \omega^2 M)\phi = 0$$

This becomes a classical eigenvalue problem where:

Eigenvalues $\lambda = \omega^2$

Eigenvectors ϕ = mode shapes

Thus:

Eigenvalues represent natural frequencies

Eigenvectors represent mode shapes of vibration

This is critical in earthquake-resistant design.

Example: Two-Story Building Model

Consider a simplified 2-story building:

Mass M and Stiffness K matrices are:

$$M = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}; K = \begin{bmatrix} 6 & -2 \\ -2 & 4 \end{bmatrix}$$

Solve:

$$\begin{aligned} \det(K - \lambda M) &= 0 \\ \begin{vmatrix} 6 - 2\lambda & -2 \\ -2 & 4 - \lambda \end{vmatrix} &= 0 \\ (6 - 2\lambda)(4 - \lambda) - 4 &= 0 \\ 2\lambda^2 - 14\lambda + 20 &= 0 \end{aligned}$$

The two solution are $\lambda_1 = 5, \lambda_2 = 2$

Natural frequencies: $\omega_1 = \sqrt{5}, \omega_2 = \sqrt{2}$

Interpretation: First mode: global sway motion, Second mode: local deformation

6.4 Methods to Compute Eigenvalues

6.4.1 Analytical Method (Small Matrices)

For a 2x2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The characteristic equation is:

$$\det(A - \lambda I) = \begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = 0$$

$$(a - \lambda)(d - \lambda) - bc = 0$$

This gives a quadratic equation in λ .

Example 1

Let:

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$$

Characteristic equation:

$$(4 - \lambda)(3 - \lambda) - 2 = 0$$

$$\lambda^2 - 7\lambda + 10 = 0$$

$$(\lambda - 5)(\lambda - 2) = 0$$

So eigenvalues: $\lambda_1 = 5, \lambda_2 = 2$

Eigenvectors

For $\lambda = 5$

$$(A - 5I)v = 0$$

$$\begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix} v = 0$$

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

And for $\lambda = 2$ we have $v_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

Example 2

We use the characteristic polynomial method to determine the eigenvalues and eigenvectors of the following example:

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

We can calculate the characteristic polynomial of A using the following formula:

$$P(\lambda) = \det(A - \lambda I)$$

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

So $P(\lambda) = \det(A - \lambda I) = (2 - \lambda)(4 - \lambda)(5 - \lambda) = -\lambda^3 + 11\lambda^2 - 38\lambda + 40$.

The roots of this polynomial are the eigenvalues of A. They can be calculated using the following equation: $-\lambda^3 + 11\lambda^2 - 38\lambda + 40 = 0$. Therefore, the eigenvalues of A are

$$\lambda_1 = 6.30; \quad \lambda_2 = 2.70; \quad \lambda_3 = 2$$

Eigenvectors: For each eigenvalue λ_i , solve the system of linear equations

$$(A - \lambda_i I)\varphi_i$$

$$\begin{aligned} \text{For } i = 1 \quad (A - \lambda_1 I)\phi_1 &= \begin{bmatrix} 2 - \lambda_1 & 0 & 0 \\ -5 & 4 - \lambda_1 & 1 \\ 1 & 3 & 5 - \lambda_1 \end{bmatrix} \begin{Bmatrix} \phi_1^1 \\ \phi_1^2 \\ \phi_1^3 \end{Bmatrix} = 0 \Rightarrow \{\phi_1\} = \begin{Bmatrix} \phi_1^1 \\ \phi_1^2 \\ \phi_1^3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0.398 \\ 0.917 \end{Bmatrix} \\ \text{For } i = 2 \quad (A - \lambda_2 I)\phi_2 &= \begin{bmatrix} 2 - \lambda_2 & 0 & 0 \\ -5 & 4 - \lambda_2 & 1 \\ 1 & 3 & 5 - \lambda_2 \end{bmatrix} \begin{Bmatrix} \phi_2^1 \\ \phi_2^2 \\ \phi_2^3 \end{Bmatrix} = 0 \Rightarrow \{\phi_2\} = \begin{Bmatrix} \phi_2^1 \\ \phi_2^2 \\ \phi_2^3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0.608 \\ -0.793 \end{Bmatrix} \\ \text{For } i = 3 \quad (A - \lambda_3 I)\phi_3 &= \begin{bmatrix} 2 - \lambda_3 & 0 & 0 \\ -5 & 4 - \lambda_3 & 1 \\ 1 & 3 & 5 - \lambda_3 \end{bmatrix} \begin{Bmatrix} \phi_3^1 \\ \phi_3^2 \\ \phi_3^3 \end{Bmatrix} = 0 \Rightarrow \{\phi_3\} = \begin{Bmatrix} \phi_3^1 \\ \phi_3^2 \\ \phi_3^3 \end{Bmatrix} = \begin{Bmatrix} 0.127 \\ 0.679 \\ -0.72 \end{Bmatrix} \end{aligned}$$

6.4.2 Numerical Methods for Eigenvalue Problems

In real civil engineering systems, matrices are very large (100×100 or more), so analytical methods are not practical. Numerical methods are used.

a. Power Method

The Power Method is used to find the largest eigenvalue.

Algorithm:

1. Start with an initial vector x_0
2. Multiply: $x_{k+1} = Ax_k$
3. Normalize at each step
4. Repeat until convergence

The method converges to the dominant eigenvalue.

b. Inverse Power Method

Used to find the smallest eigenvalue.

$$x_{k+1} = A^{-1}x_k$$

This method is useful in:

- Buckling load calculation
- Stability problems

c. QR Algorithm

The most important modern method used in FEM software.

Steps:

- Decompose: $A = QR$
- Compute: $A_1 = RQ$
- Repeat until A becomes diagonal

Diagonal elements → eigenvalues

This method is used in MATLAB, ANSYS, and Abaqus.

6.5 Applications in Civil Engineering

6.5.1 Structural Vibration Analysis

Buildings and bridges behave like dynamic systems under wind or earthquakes.

The eigenvalue problem:

$$(K - \lambda M)\phi = 0$$

gives:

- Natural frequencies (danger zones)
- Mode shapes (deformation patterns)

Engineers design structures to avoid resonance, where external frequency matches natural frequency.

6.5.2 Finite Element Method (FEM)

In FEM, the structure is divided into elements:

$$[K]\{u\} = \{F\}$$

For dynamic problems:

$$[M]\ddot{u} + [K]u = 0$$

Eigenvalue analysis helps:

- Reduce system complexity
- Perform modal reduction
- Accelerate simulations

6.5.3 Buckling of Columns

For compression members:

$$(K - \lambda K_g)\phi = 0$$

where:

K = elastic stiffness

K_g = geometric stiffness

Eigenvalues represent critical buckling loads. If load exceeds eigenvalue $\rightarrow$ structural failure occurs.

6.5.4 Seismic Engineering

During earthquakes:

- Ground motion excites multiple modes
- Eigenvalues determine which modes dominate response

Lower modes usually carry most of the energy.

Eigenvalues provide:

- Stability information
- Resonance risk
- Energy distribution
- Structural behavior classification

Large eigenvalue $\rightarrow$ stiff system

Small eigenvalue $\rightarrow$ flexible system

6.6 Conclusion

Eigenvalues and eigenvectors form a cornerstone of modern civil engineering analysis. They allow engineers to transform complex coupled structural systems into independent modal systems that are easier to analyze and interpret. From vibration analysis of high-rise buildings to buckling of columns and finite element simulations, eigenvalue methods provide both physical insight and computational efficiency.

In practice, almost all structural dynamic software relies on eigenvalue extraction to predict natural frequencies and mode shapes. Therefore, mastering this concept is essential for any civil engineering student or researcher working in structural mechanics and numerical modeling.

Chapter 7: Function Approximation

7.1 Introduction

7.2 Mathematical Definition

7.3 Physical Interpretation in Civil Engineering

7.4 Methods to Compute Eigenvalues

7.5 Applications in Civil Engineering

7.6 Conclusion

7.1 Introduction

Function approximation is an essential concept in mathematics, numerical analysis, engineering, physics, computer science, and data science. In many practical situations, the exact analytical form of a function is either too complicated or impossible to use directly in computations. Function approximation allows us to replace a complex function with a simpler mathematical expression that is easier to analyze, compute, and manipulate. Approximation techniques are widely used in:

- Numerical simulations
- Signal processing
- Artificial intelligence and machine learning
- Data fitting and regression
- Structural and mechanical engineering
- Computer graphics
- Control systems
- Scientific computing

The main objective of function approximation is to find a simpler function that behaves similarly to the original function within a certain interval. Common methods of function approximation include:

1. Taylor Series Approximation
2. Interpolation Methods
3. Least Squares Approximation
4. Trigonometric Approximation
5. Piecewise Approximation

In this chapter, we focus mainly on Taylor Series and Least Squares Methods.

7.2 Taylor Series Approximation

7.2.1 Definition

The Taylor Series is one of the most important methods used to approximate functions. The idea behind the Taylor series is that any sufficiently smooth and analytic function can be represented as an infinite polynomial series around a point a .

The Taylor series expansion of a function $f(x)$ around the point a is:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

or more generally:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

Where:

$f(a)$ is the value of the function at a

$f'(a)$ is the first derivative evaluated at a

$f''(a)$ is the second derivative evaluated at a

$f^{(n)}(a)$ is the n th derivative evaluated at a

$(x-a)$ measures the distance between the approximation point and the expansion point, When $a=0$, the Taylor series is called the **Maclaurin Series**.

Taylor series approximation is important because:

- It simplifies complicated functions into polynomials.
- Polynomials are easier to differentiate and integrate.
- It helps computers calculate transcendental functions such as: $\sin(x)$, $\cos(x)$, e^x , $\ln(x)$.
- It is widely used in engineering simulations and numerical methods.

In practical applications, we cannot use an infinite number of terms. Therefore, we truncate the series after a finite number of terms. The n th-order Taylor polynomial is:

$$P_n(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} (x-a)^k$$

The approximation becomes more accurate as more terms are included.

7.2.2 Error in Taylor Approximation

The error between the exact function and the Taylor approximation is called the remainder or truncation error. The remainder term is:

$$R_n(x) = f(x) - P_n(x)$$

A smaller remainder means a better approximation.

Generally: $|R_n(x)| \rightarrow 0$ as $n \rightarrow \infty$

- Near the point a , the approximation is highly accurate.
- Far from a , the approximation may become inaccurate.

Numerical Example

Approximate the value of:

$$\sin(0.2)$$

using the first three nonzero terms of the Taylor series expansion around $a = 0$

The Maclaurin series of the sine function is:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

Using the first three nonzero terms:

$$\sin(x) \approx x - \frac{x^3}{6} + \frac{x^5}{120}$$

Substituting $x = 0.2$

$$\sin(0.2) \approx 0.2 - \frac{(0.2)^3}{6} + \frac{(0.2)^5}{120}$$

Therefore

$$\begin{aligned}\sin(0.2) &\approx 0.2 - 0.001333 + 0.00000267 \\ \sin(0.2) &\approx \mathbf{0.19867}\end{aligned}$$

Using a calculator: $\sin(0.2) = 0.198669$

Hence, the Taylor approximation gives a very accurate result with a very small error.

Note : Where do we use Taylor expansion around $a=0$?

As example we used it small-Angle Approximation in Physics $\sin(\theta) \approx \theta$ and $\cos(\theta) \approx 1 - \frac{\theta^2}{2}$

7.2.3 Common Taylor Expansions

The following Taylor and Maclaurin series expansions are among the most commonly used approximations in mathematics, physics, and engineering. These expansions simplify complicated functions into polynomial forms that are easier to analyze and compute.

Function	Taylor / Maclaurin Expansion
Exponential Function	$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$
Sine Function	$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$
Cosine Function	$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$
Natural Logarithm	$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$
Geometric Series	$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \dots$
Binomial Expansion	$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \dots$
Hyperbolic Sine	$\sinh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$
Hyperbolic Cosine	$\cosh(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$

7.2.4 Advantages and Limitations of Taylor Series

- **Advantages**
 - Simplifies complicated functions

- Easy differentiation and integration
- Highly accurate near the expansion point
- Fundamental in numerical analysis
- **Limitations**
 - Accuracy decreases far from the expansion point
 - Some series diverge
 - High-order derivatives may be difficult to compute
 - Many terms may be needed for high accuracy[^]

7.3 Least Squares Methods

7.3.1 Introduction

Least Squares Methods are numerical and statistical techniques used to determine the best mathematical model that fits a set of experimental or observed data.

In practical engineering and scientific problems, measured data usually contain errors due to:

- Measurement inaccuracies
- Instrument limitations
- Environmental effects
- Human errors

Therefore, the data points may not lie exactly on a single curve or straight line. The Least Squares Method provides a systematic procedure to obtain the best approximation by minimizing the total error between the measured data and the mathematical model.

Least squares methods are widely used in:

- Engineering experiments
- Data analysis
- Machine learning
- Signal processing
- Economics and statistics
- Scientific modeling
- Trend prediction

7.3.2 Basic Idea of Least Squares

Suppose we have experimental data points:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

We want to find a mathematical function:

$$y = f(x)$$

that best represents the data.

For each data point, the difference between the observed value and the predicted value is called the residual or error.

$$e_i = y_i - y_i^{\text{predicted}}$$

The Least Squares Method minimizes the sum of the squares of these residuals:

$$S = \sum_{i=1}^n e_i^2$$

Squaring the errors ensures that:

- Positive and negative errors do not cancel each other
- Large errors receive greater penalty
- The solution becomes mathematically convenient

7.3.3 Procedure of the Least Squares Method

The least squares procedure consists of several steps.

Step 1: Select the Mathematical Model

Choose the function that best describes the data.

For linear regression:

$$y = mx + b$$

Other models may also be used, such as:

- Polynomial models
- Exponential models
- Logarithmic models

Step 2: Collect Experimental Data

Suppose the following measurements are obtained:

x	1	2	3	4	5
y	3.5	6.8	10.2	13.6	17

Step 3: Compute Predicted Values

The predicted values are:

$$y_i^{predicted} = mx_i + b$$

Step 4: Compute Residuals

The residual for each point is: $e_i = y_i - (mx_i + b)$

$$\begin{aligned} \text{Thus: } e_1 &= 3.5 - (m + b) & e_2 &= 6.8 - (2m + b) & e_3 &= 10.2 - (3m + b) \\ e_4 &= 13.6 - (4m + b) & e_5 &= 17.0 - (5m + b) \end{aligned}$$

Step 5: Define the Objective Function

The sum of squared residuals is:

$$S = e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$$

The objective is to minimize S

Formulas for Linear Least Squares Regression :

The optimal values of slope m and intercept b are calculated using:

Slope

$$m = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

Intercept

$$b = \frac{\sum y - m \sum x}{n}$$

Where:

- n is the number of data points
- $\sum x$ is the sum of x -values
- $\sum y$ is the sum of y -values
- $\sum xy$ is the sum of products xy
- $\sum x^2$ is the sum of squared x -values
-

Numerical Example

Consider the following data:

X	1	2	3	4	5
Y	3.5	6.8	10.2	13.6	17

We will determine the best-fit straight line using the least squares method.

Step 1: Compute Required Quantities

Sum of x -values

$$\sum x = 1 + 2 + 3 + 4 + 5 = 15$$

Sum of y -values

$$\sum y = 3.5 + 6.8 + 10.2 + 13.6 + 17.0 = 51.1$$

Sum of xy -products

$$\sum xy = (1)(3.5) + (2)(6.8) + (3)(10.2) + (4)(13.6) + (5)(17.0) = 187.1$$

Sum of squared x -values

$$\sum x^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$$

Step 2: Calculate the Slope and the Intercept

$$m = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} = \frac{5(187.1) - 15(51.1)}{5(55) - 15^2} = 3.38$$

$$b = \frac{\sum y - m\sum x}{n} = \frac{51.1 - (3.38)(15)}{5} = 0.08$$

Step 4: Final Regression Equation

The best-fit straight line is: $y = 3.38x + 0.08$

This equation represents the mathematical model that best approximates the experimental data.

7.3.4 Advantages and Limitations

Advantages

- Simple and efficient
- Works well with noisy data
- Provides optimal fit in the least-squares sense
- Widely used in engineering and statistics

Limitations

- Sensitive to outliers
- Assumes a predefined model form
- Linear regression may not accurately represent nonlinear systems
- Results depend on data quality

7.4 Conclusion

In this chapter, we studied the fundamental concepts of function approximation and their importance in engineering, mathematics, and scientific computations. The chapter began with an introduction to approximation methods and the need to replace complicated functions with simpler mathematical expressions. We then explored the Taylor series approximation, which represents functions using polynomial expansions based on derivatives evaluated at a specific point. Several numerical examples demonstrated how Taylor series can provide highly accurate approximations near the expansion point. The concepts of truncation error and convergence were also discussed to evaluate the accuracy of the approximation. In the second part of the chapter, we studied the Least Squares Regression Method, which is used to fit mathematical models to experimental or observed data. The method minimizes the sum of squared residuals to obtain the best-fit equation. Finally, we saw that both Taylor approximation and least squares regression are essential tools widely used in engineering analysis, numerical methods, data fitting, and scientific modeling.

General conclusion

Throughout this course, the fundamental principles and applications of numerical methods have been presented in order to provide engineering students with practical computational tools for solving mathematical and engineering problems. Numerical methods represent an essential branch of applied mathematics because they enable approximate solutions to problems that are difficult or impossible to solve analytically. In modern engineering practice, especially in civil engineering, these techniques are indispensable for analyzing complex systems, predicting structural behavior, and performing accurate simulations.

The study began with methods for solving nonlinear equations, including the Bisection Method, Fixed-Point Iteration Method, and Newton–Raphson Method. These techniques demonstrated different approaches to approximating the roots of nonlinear functions. The Bisection Method was shown to be simple and reliable because it guarantees convergence under suitable conditions, although its convergence rate is relatively slow. The Fixed-Point Method introduced the concept of iterative approximation and highlighted the importance of convergence conditions. Newton’s Method, on the other hand, illustrated the efficiency of quadratic convergence when a suitable initial approximation is chosen. Through these methods, students learned how numerical algorithms can progressively approach exact solutions with controlled accuracy.

The course also explored systems of linear equations, which constitute one of the most common mathematical problems in engineering analysis. Matrix representation provided a systematic and compact way to organize and solve these systems. Important matrix concepts such as determinants, inverse matrices, matrix operations, and matrix decomposition were introduced as essential mathematical tools. Direct methods, including Gaussian Elimination, Gauss–Jordan Elimination, and LU decomposition, were studied to solve systems accurately through algebraic transformations. In addition, iterative methods such as Jacobi and Gauss–Seidel methods were presented as efficient alternatives for large systems where direct methods may become computationally expensive.

Numerical integration techniques formed another major component of the course. Since many engineering problems involve integrals that cannot be evaluated analytically, methods such as the Rectangle Rule, Trapezoidal Rule, Simpson’s Rules, and Gaussian Quadrature provide practical approximations for definite integrals. These methods are particularly important in engineering calculations related to areas, volumes, distributed loads, probability analysis, and differential equations. The comparison between different integration techniques also emphasized the balance between computational simplicity and numerical accuracy.

Interpolation methods were introduced to estimate unknown values between known data points. The Lagrange and Newton interpolation methods demonstrated how polynomial functions can approximate experimental or tabulated data efficiently. Interpolation is widely used in engineering measurements, data analysis, and computational modeling because it allows engineers to construct continuous mathematical representations from discrete datasets. The estimation of interpolation error further highlighted the importance of evaluating the reliability and precision of numerical approximations.

The numerical solution of differential equations represented another essential topic addressed in this course. Many engineering systems are modeled by differential equations describing physical phenomena such as heat transfer, structural vibration, fluid mechanics, and dynamic systems.

Numerical techniques such as Euler's Method and the Fourth-Order Runge–Kutta Method provided efficient procedures to approximate these solutions step by step. These methods form the basis of many advanced computational programs and simulation software widely used in engineering industries and research laboratories.

The course also examined eigenvalues and eigenvectors, which play a significant role in structural dynamics, vibration analysis, finite element methods, and stability problems. Their applications in civil engineering demonstrated the practical importance of numerical methods in understanding the behavior of structures under dynamic loading conditions such as earthquakes and wind forces. In addition, function approximation methods such as Taylor series expansion and least squares approximation illustrated how complex functions and experimental data can be represented using simpler mathematical expressions.

One of the most important lessons learned throughout this course is that numerical methods do not simply provide approximate results; they also require careful analysis of convergence, stability, and error estimation. Since numerical computations are affected by truncation errors, round-off errors, and iterative approximations, engineers must always evaluate the reliability and precision of the obtained solutions. Understanding stopping criteria, convergence rates, and applicability conditions is therefore essential for the correct implementation of numerical algorithms.

Another significant aspect of numerical methods is their strong relationship with computer science and computational technology. Modern engineering analysis depends heavily on computer-based simulations and software packages that incorporate numerical algorithms. Engineering tools such as MATLAB, ANSYS, ABAQUS, ETABS, and finite element analysis programs use numerical methods extensively to solve complex mathematical models efficiently. Therefore, mastering numerical methods not only improves mathematical understanding but also enhances computational and programming skills that are essential in modern engineering careers.

From an educational perspective, this course helps students develop logical thinking, analytical reasoning, and systematic problem-solving abilities. Numerical methods encourage students to break down complicated problems into manageable computational steps and analyze the accuracy of their results critically. These skills are valuable not only in engineering but also in scientific research, economics, computer science, and many other technical disciplines.

In conclusion, numerical methods constitute a powerful and indispensable tool in modern engineering and scientific computation. They provide effective solutions for a wide variety of mathematical problems encountered in practical applications where exact analytical methods are insufficient. Through the study of nonlinear equations, systems of equations, interpolation, integration, differential equations, eigenvalue problems, and approximation techniques, students acquire the theoretical knowledge and computational skills necessary to solve real engineering problems efficiently and accurately. As technology continues to evolve and engineering systems become increasingly complex, the importance of numerical methods will continue to expand, making them an essential foundation for future engineers and researchers.

References and Resources

- Atkinson, K. E. (1989). *An introduction to numerical analysis* (2nd ed.). John Wiley & Sons.
- Burden, R. L., & Faires, J. D. (2011). *Numerical analysis* (9th ed.). Brooks/Cole Cengage Learning.
- Chapra, S. C. (2018). *Applied numerical methods with MATLAB for engineers and scientists* (4th ed.). McGraw-Hill Education.
- Chapra, S. C., & Canale, R. P. (2015). *Numerical methods for engineers* (7th ed.). McGraw-Hill Education.
- Cheney, W., & Kincaid, D. (2012). *Numerical mathematics and computing* (7th ed.). Brooks/Cole.
- Conte, S. D., & de Boor, C. (1980). *Elementary numerical analysis: An algorithmic approach* (3rd ed.). McGraw-Hill.
- Gerald, C. F., & Wheatley, P. O. (2004). *Applied numerical analysis* (7th ed.). Pearson Education.
- Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations* (4th ed.). Johns Hopkins University Press.
- Heath, M. T. (2018). *Scientific computing: An introductory survey* (2nd ed.). Society for Industrial and Applied Mathematics.
- Hildebrand, F. B. (1987). *Introduction to numerical analysis* (2nd ed.). Dover Publications.
- Isaacson, E., & Keller, H. B. (1994). *Analysis of numerical methods*. Dover Publications.
- Kiusalaas, J. (2013). *Numerical methods in engineering with MATLAB* (3rd ed.). Cambridge University Press.
- Mathews, J. H., & Fink, K. D. (2004). *Numerical methods using MATLAB* (4th ed.). Prentice Hall.
- Moin, P. (2010). *Fundamentals of engineering numerical analysis* (2nd ed.). Cambridge University Press.
- Nakamura, S. (1997). *Numerical analysis and graphic visualization with MATLAB* (2nd ed.). Prentice Hall.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge University Press.

- Quarteroni, A., Sacco, R., & Saleri, F. (2010). Numerical mathematics (2nd ed.). Springer.
- Ralston, A., & Rabinowitz, P. (2001). A first course in numerical analysis (2nd ed.). Dover Publications.
- Sastry, S. S. (2012). Introductory methods of numerical analysis (5th ed.). PHI Learning.
- Smith, G. D. (1985). Numerical solution of partial differential equations: Finite difference methods (3rd ed.). Oxford University Press.
- Strang, G. (2009). Introduction to linear algebra (4th ed.). Wellesley-Cambridge Press.
- Trefethen, L. N., & Bau, D. (1997). Numerical linear algebra. Society for Industrial and Applied Mathematics.
- Yang, W. Y., Cao, W., Chung, T.-S., & Morris, J. (2005). Applied numerical methods using MATLAB. John Wiley & Sons.

Additional Learning Resources

- MATLAB Documentation. (2025). MATLAB numerical computation resources. MathWorks. <https://www.mathworks.com>
- MIT OpenCourseWare. (2025). Introduction to numerical methods. Massachusetts Institute of Technology. <https://ocw.mit.edu>
- National Institute of Standards and Technology. (2025). Numerical analysis references and resources. <https://www.nist.gov>
- Wolfram Research. (2025). Numerical mathematics and computational tools. Wolfram Mathematica Documentation Center. <https://reference.wolfram.com>
- Khan Academy. (2025). Linear algebra and differential equations. <https://www.khanacademy.org>
- Coursera. (2025). Numerical methods for engineers and scientists. <https://www.coursera.org>
- edX. (2025). Computational science and numerical analysis courses. <https://www.edx.org>