

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Mustapha Stambouli University of Mascara
Faculty of Science and Technology
Department of Electrical Engineering

Applied Numerical Methods and Optimization

Course Notes

Master 1 – Power system/ Electromechanics

Prepared by
Dr.

Academic Year 2025–2026

Contents

Introduction	6
1 Review of Numerical Methods	7
1.1 Systems of Linear Equations	7
1.1.1 Iterative Methods	7
1.1.2 Jacobi Method	8
1.1.3 Gauss–Seidel Method	10
1.1.4 Convergence Condition	12
1.1.5 Comparison of the Methods	13
1.1.6 Application in Electrical Engineering: Nodal Analysis of Electrical Circuits	13
1.1.7 Summary	15
1.2 Nonlinear Systems of Equations	15
1.2.1 Introduction	15
1.2.2 Example	16
1.2.3 Newton Method for Nonlinear Systems	16
1.2.4 Algorithm	16
1.2.5 Application in Electrical Power Systems	17
1.2.6 MATLAB Implementation of the Newton Method	17
1.3 Numerical Integration	18
1.3.1 Introduction	18
1.3.2 Trapezoidal Rule	19
1.3.3 Composite Trapezoidal Rule	19
1.3.4 Simpson’s Rule	19
1.3.5 Composite Simpson Rule	20
1.3.6 MATLAB Implementation	20
1.3.7 Application in Electrical Engineering	20
1.4 Numerical Differentiation	21
1.4.1 Introduction	21
1.4.2 Forward Difference Formula	21
1.4.3 Backward Difference Formula	21
1.4.4 Central Difference Formula	21
1.4.5 Second Derivative Approximation	21
1.4.6 MATLAB Implementation	22
1.4.7 Application in Electrical Engineering	22
1.4.8 Figure: Geometric Interpretation	23
1.5 Ordinary Differential Equations (ODEs)	23
1.5.1 Introduction	23

1.5.2	Euler Method (Explicit)	23
1.5.3	Runge-Kutta Methods	23
1.5.4	MATLAB Implementation: Euler and RK4	24
1.5.5	Application in Electrical Engineering	25
1.6	Numerical Example: RC Circuit	25
1.6.1	Circuit Description	25
1.6.2	Problem Parameters	26
1.6.3	Numerical Solution in MATLAB	26
1.6.4	Discussion of Results	27
1.6.5	Numerical Solution Plot (Euler vs RK4)	27
1.7	Resolution of Systems of Ordinary Differential Equations	28
1.7.1	Introduction	28
1.7.2	Euler Method for Systems	28
1.7.3	Runge-Kutta Method (RK4) for Systems	28
1.7.4	Example: DC Motor Dynamics	29
1.7.5	MATLAB Implementation (Euler and RK4)	29
1.7.6	Discussion	30
2	Partial Differential Equations	47
2.1	Introduction and Classification of PDE Problems	47
2.1.1	Classification of PDEs with Examples	47
2.1.2	Comparison of PDE Types	49
2.1.3	Boundary and Initial Conditions	49
2.2	Finite Difference Method (FDM)	50
2.2.1	1D Example: Voltage along a resistive wire	50
2.2.2	2D Example: Voltage Distribution in a Rectangular Conductor	51
2.2.3	2D Finite Difference Method with Irregular Boundaries	52
2.2.4	2D Finite Difference Method with Neumann Boundary Conditions	54
2.3	Finite Volume Method (FVM)	56
2.3.1	Finite Difference Method for Parabolic Equations	57
2.3.2	2D Parabolic Equation (Transient Voltage Diffusion in a Conductive Plate)	58
2.4	Finite Element Method (FEM)	61
2.4.1	Introduction	61
2.4.2	1D Boundary Value Problem	61
2.4.3	Discretization of the Domain	62
2.4.4	Weak Formulation	62
2.4.5	Shape Functions	62
2.4.6	Element Matrix	62
2.4.7	Assembly of Global System	63
2.4.8	Application to Electrical Engineering	63
2.4.9	Discussion	63
2.4.10	Numerical Example: 1D FEM Step-by-Step Solution	64
2.5	Two-Dimensional Finite Element Method (2D FEM)	65
2.5.1	Introduction	65
2.5.2	Governing Equation	66
2.5.3	Domain Discretization	66
2.5.4	Triangular Element	66

2.5.5	Shape Functions in 2D	66
2.5.6	Element Matrix	67
2.5.7	Assembly of the Global System	67
2.5.8	Application in Electrical Engineering	67
2.5.9	Discussion	67
2.5.10	Numerical Example: 2D FEM on a Triangular Element	68
2.6	Comparison of FDM, FVM and FEM	69
3	Optimization Techniques	86
3.1	Definition and Formulation of an Optimization Problem	86
3.1.1	Basic Definition	86
3.1.2	Main Components of an Optimization Problem	87
3.1.3	Minimization and Maximization	88
3.1.4	Unconstrained and Constrained Optimization	88
3.1.5	Single-Variable and Multivariable Optimization	89
3.1.6	Local and Global Optimum	90
3.1.7	Standard Mathematical Form of an Optimization Problem	90
3.1.8	Engineering Example: Economic Dispatch in a Power System	90
3.1.9	Graphical Interpretation	91
3.1.10	Summary	91
3.2	Single and Multiple Optimization with or without Constraints	92
3.2.1	Single-Objective Optimization	92
3.2.2	Multi-Objective Optimization	93
3.2.3	Unconstrained Optimization	94
3.2.4	Constrained Optimization	96
3.2.5	Comparison Between the Different Categories	98
3.2.6	Engineering Example: Optimal Power Dispatch	98
3.2.7	Summary	99
3.3	Unconstrained Optimization Algorithms	99
3.3.1	Deterministic Methods	100
3.3.2	Stochastic Methods	102
3.3.3	Engineering Application	103
3.3.4	Summary	103
3.4	Constraint Handling Methods	103
3.4.1	Transformation Methods	104
3.4.2	Direct Methods	105
3.4.3	Engineering Application	107
3.4.4	Comparison of Methods	108
3.4.5	Summary	108

Fiche de Module

Master 1 – Réseaux Électriques et Électromécanique

Semestre : 1

UE Fondamentale Code : UEF 1.1.2

Matière : Méthodes numériques appliquées et optimisation

VHS : 45h (Cours : 1h30, TD : 1h30)

Crédits : 4

Coefficient : 2

Objectifs de l'enseignement :

L'objectif de cet enseignement est de présenter les outils nécessaires à l'analyse numérique et à l'optimisation, avec ou sans contraintes, des systèmes physiques dans le domaine de l'ingénierie.

Connaissances préalables recommandées :

Mathématiques, programmation, maîtrise de l'environnement MATLAB.

Contenu de la matière :

Chapitre I : Rappels sur quelques méthodes numériques (3 semaines)

- Résolution des systèmes d'équations non linéaires par les méthodes itératives.
- Intégration et différentiation numérique.
- Méthodes de résolution des équations différentielles ordinaires (EDO) : méthodes d'Euler, méthodes de Runge-Kutta, méthode d'Adams.
- Résolution des systèmes d'EDO.

Chapitre II : Équations aux dérivées partielles (EDP) (6 semaines)

- Introduction et classification des problèmes aux dérivées partielles et des conditions aux limites.
- Méthodes de résolution des EDP :
 - Méthode des différences finies (MDF)
 - Méthode des volumes finis (MVF)
 - Méthode des éléments finis (MEF)

Chapitre III : Techniques d'optimisation (6 semaines)

- Définition et formulation d'un problème d'optimisation.
- Optimisation unique et multiple avec ou sans contraintes.
- Algorithmes d'optimisation sans contraintes (méthodes déterministes, méthodes stochastiques).

- Traitement des contraintes (méthodes de transformation, méthodes directes).

Mode d'évaluation :

Contrôle continu : 40% Examen : 60%

Références bibliographiques :

1. G. Allaire, *Analyse Numérique et Optimisation*, Éditions de l'École Polytechnique, 2012.
2. S. S. Rao, *Optimization – Theory and Applications*, Wiley-Eastern Limited, 1984.
3. A. Fortin, *Analyse numérique pour ingénieurs*, Presses Internationales Polytechnique, 2011.
4. J. Bastien, J. N. Martin, *Introduction à l'analyse numérique : Applications sous MATLAB*, Dunod, 2003.
5. A. Quarteroni, F. Saleri, P. Gervasio, *Calcul scientifique*, Springer, 2008.

Introduction

Modern engineering systems are increasingly complex and are often described by mathematical models involving nonlinear equations, differential equations, and optimization problems. In many practical situations, analytical solutions to these mathematical models are difficult or impossible to obtain. As a result, numerical methods play a fundamental role in the analysis, simulation, and optimization of engineering systems.

In the fields of electrical engineering, power systems, and electromechanics, numerical methods are widely used to solve problems such as power flow analysis in electrical networks, dynamic modeling of electrical machines, simulation of electromagnetic and thermal phenomena, and optimization of energy systems. These applications require reliable computational techniques capable of providing accurate approximations of the mathematical models describing physical processes.

The objective of this course is to introduce the fundamental concepts and techniques of numerical analysis and optimization used in engineering applications. The course focuses on the development of numerical algorithms for solving nonlinear equations, performing numerical differentiation and integration, and solving ordinary and partial differential equations. In addition, the course introduces optimization techniques used to determine optimal solutions for engineering problems with or without constraints.

Special attention is given to applications relevant to electrical engineering and electromechanical systems. Examples and exercises are selected to illustrate practical problems such as the numerical solution of differential equations describing dynamic systems, numerical approximation methods used in field modeling, and optimization techniques applied to engineering design and energy management.

Throughout the course, the implementation of numerical algorithms using computational tools such as MATLAB is emphasized. The use of programming environments allows students to understand the practical aspects of numerical methods and to develop the ability to apply these techniques to real engineering problems.

At the end of this course, students should be able to:

- understand the principles of numerical approximation methods;
- implement numerical algorithms for solving engineering problems;
- analyze the convergence and accuracy of numerical methods;
- apply optimization techniques to engineering systems;
- use numerical tools such as MATLAB to simulate and solve practical problems.

The knowledge acquired in this course constitutes an essential foundation for advanced studies and research in electrical engineering, particularly in areas such as power systems analysis, control systems, renewable energy systems, and electromechanical system modeling.

Chapter 1

Review of Numerical Methods

1.1 Systems of Linear Equations

Many engineering problems lead to systems of linear algebraic equations. Such systems arise naturally in the modeling of physical phenomena including electrical circuits, power systems, mechanical systems, and heat transfer problems.

In electrical engineering, for example, the application of Kirchhoff's laws to electrical networks leads to a system of linear equations involving currents and voltages.

A general system of n linear equations with n unknowns can be written in matrix form as

$$\mathbf{Ax} = \mathbf{b} \quad (1.1)$$

where

- $\mathbf{A}$ is an $n \times n$ matrix of coefficients,
- $\mathbf{x}$ is the vector of unknown variables,
- $\mathbf{b}$ is the vector of constants.

When the system size is small, direct methods such as Gaussian elimination can be used. However, in many engineering applications, the dimension of the system becomes very large. In such cases, iterative methods are often preferred because they are more memory-efficient and are well suited for large sparse systems.

Among the most widely used iterative methods are the **Jacobi method** and the **Gauss–Seidel method**.

1.1.1 Iterative Methods

Iterative methods start with an initial approximation of the solution vector and generate a sequence of improved approximations until convergence is reached.

Consider the linear system

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

Each equation can be rearranged to isolate one variable. This leads to iterative schemes such as the Jacobi and Gauss–Seidel methods.

1.1.2 Jacobi Method

Principle

The Jacobi method computes each variable using only the values obtained from the previous iteration.

The iterative formula is given by

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right) \quad (1.2)$$

where k denotes the iteration number.

Algorithm

1. Choose an initial approximation $\mathbf{x}^{(0)}$.
2. Compute $\mathbf{x}^{(k+1)}$ using the Jacobi formula.
3. Repeat until convergence is achieved.

A common stopping criterion is

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon \quad (1.3)$$

where ε is a prescribed tolerance.

Example

Consider the following system:

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

The equations can be rewritten as

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

Take the initial approximation

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The Jacobi iterations are then:

Iteration 1

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

$$x_2^{(1)} = \frac{25 + 0 + 0}{11} = 2.2727$$

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

Thus,

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0.6 \\ 2.2727 \\ -1.1 \end{bmatrix}$$

Iteration 2

$$x_1^{(2)} = \frac{6 + 2.2727 - 2(-1.1)}{10} = 1.0473$$

$$x_2^{(2)} = \frac{25 + 0.6 - 1.1}{11} = 2.2273$$

$$x_3^{(2)} = \frac{-11 - 2(0.6) + 2.2727}{10} = -0.9927$$

Thus,

$$\mathbf{x}^{(2)} = \begin{bmatrix} 1.0473 \\ 2.2273 \\ -0.9927 \end{bmatrix}$$

Iteration 3

$$x_1^{(3)} = \frac{6 + 2.2273 - 2(-0.9927)}{10} = 1.0213$$

$$x_2^{(3)} = \frac{25 + 1.0473 - 0.9927}{11} = 2.2777$$

$$x_3^{(3)} = \frac{-11 - 2(1.0473) + 2.2273}{10} = -1.0867$$

Thus,

$$\mathbf{x}^{(3)} = \begin{bmatrix} 1.0213 \\ 2.2777 \\ -1.0867 \end{bmatrix}$$

The iterations continue until convergence. The exact solution is approximately

$$\mathbf{x} \approx \begin{bmatrix} 1.0433 \\ 2.2692 \\ -1.0817 \end{bmatrix}$$

MATLAB Implementation

```

1  A = [10 -1 2;
2      -1 11 -1;
3       2 -1 10];
4
5  b = [6; 25; -11];
6
7  x = zeros(3,1);
8  tol = 1e-6;
9  err = 1;
10
11 while err > tol
12     x_new = zeros(3,1);
13
14     for i = 1:3
15         s = 0;
16         for j = 1:3
17             if j ~= i
18                 s = s + A(i,j)*x(j);
19             end
20         end
21         x_new(i) = (b(i)-s)/A(i,i);
22     end
23
24     err = norm(x_new - x, inf);
25     x = x_new;
26 end
27
28 disp(x)

```

1.1.3 Gauss–Seidel Method

Principle

The Gauss–Seidel method improves the Jacobi method by using the most recently computed values during the iteration.

The update formula is

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)} \right) \quad (1.4)$$

Algorithm

1. Choose an initial approximation $\mathbf{x}^{(0)}$.
2. Update the variables sequentially.
3. Use the newest available values.
4. Repeat until convergence.

Example

Using the same system and the same initial approximation,

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

the Gauss–Seidel iterations are:

Iteration 1

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

$$x_2^{(1)} = \frac{25 + 0.6 + 0}{11} = 2.3273$$

$$x_3^{(1)} = \frac{-11 - 2(0.6) + 2.3273}{10} = -0.9873$$

Thus,

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0.6 \\ 2.3273 \\ -0.9873 \end{bmatrix}$$

Iteration 2

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

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

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

Thus,

$$\mathbf{x}^{(2)} = \begin{bmatrix} 1.0302 \\ 2.2766 \\ -1.0784 \end{bmatrix}$$

Iteration 3

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

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

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

Thus,

$$\mathbf{x}^{(3)} = \begin{bmatrix} 1.0433 \\ 2.2695 \\ -1.0817 \end{bmatrix}$$

The Gauss–Seidel method converges faster than the Jacobi method for this system.

MATLAB Implementation

```

1 A = [10 -1 2;
2     -1 11 -1;
3       2 -1 10];
4
5 b = [6; 25; -11];
6
7 x = zeros(3,1);
8 n = length(b);
9 tol = 1e-6;
10 err = 1;
11
12 while err > tol
13     x_old = x;
14
15     for i = 1:n
16         s1 = A(i,1:i-1)*x(1:i-1);
17         s2 = A(i,i+1:n)*x_old(i+1:n);
18         x(i) = (b(i)-s1-s2)/A(i,i);
19     end
20
21     err = norm(x - x_old, inf);
22 end
23
24 disp(x)

```

1.1.4 Convergence Condition

A sufficient condition for the convergence of the Jacobi and Gauss–Seidel methods is that the matrix $\mathbf{A}$ is strictly diagonally dominant, that is,

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, n \quad (1.5)$$

This condition is often satisfied in many engineering applications such as electrical network analysis.

1.1.5 Comparison of the Methods

Method	Advantage	Disadvantage
Jacobi	Simple implementation	Slow convergence
Gauss–Seidel	Faster convergence	Sequential computation

Table 1.1: Comparison of Jacobi and Gauss–Seidel methods.

1.1.6 Application in Electrical Engineering: Nodal Analysis of Electrical Circuits

Linear systems of equations frequently arise in electrical engineering, particularly in the analysis of electrical networks. One of the most common techniques used to analyze electrical circuits is the **nodal analysis method**.

Nodal analysis is based on the application of **Kirchhoff’s Current Law (KCL)**, which states that the sum of currents entering a node must be equal to the sum of currents leaving that node.

Example

Consider the resistive circuit shown in Figure 1.1, containing three nodes with unknown voltages V_1 , V_2 , and V_3 .

The current sources I_1 , I_2 , and I_3 are assumed to inject current into nodes 1, 2, and 3, respectively.

Applying Kirchhoff’s Current Law to each node leads to a system of linear equations.

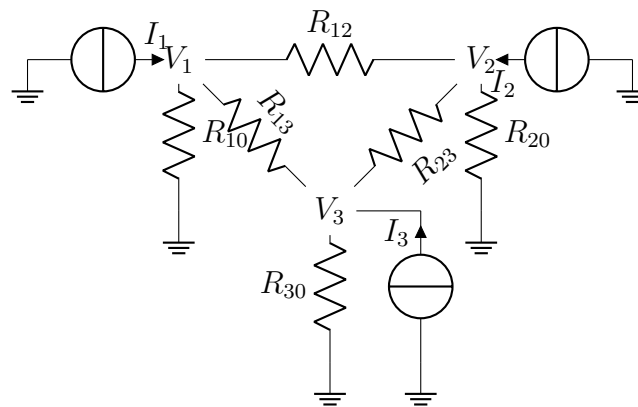


Figure 1.1: Three-node resistive electrical network used for nodal analysis.

Formulation of the Equations

Suppose the circuit contains the following resistances:

- R_{12} between nodes 1 and 2,
- R_{13} between nodes 1 and 3,
- R_{23} between nodes 2 and 3,

- R_{10} between node 1 and ground,
- R_{20} between node 2 and ground,
- R_{30} between node 3 and ground.

Let

$$G_{ij} = \frac{1}{R_{ij}}$$

be the conductance between nodes.

Applying KCL at node 1 gives

$$(G_{12} + G_{13} + G_{10})V_1 - G_{12}V_2 - G_{13}V_3 = I_1 \quad (1.6)$$

Applying KCL at node 2 gives

$$-G_{12}V_1 + (G_{12} + G_{23} + G_{20})V_2 - G_{23}V_3 = I_2 \quad (1.7)$$

Applying KCL at node 3 gives

$$-G_{13}V_1 - G_{23}V_2 + (G_{13} + G_{23} + G_{30})V_3 = I_3 \quad (1.8)$$

Matrix Form

The system can be written in matrix form as

$$\mathbf{G}\mathbf{V} = \mathbf{I} \quad (1.9)$$

with

$$\mathbf{G} = \begin{bmatrix} G_{12} + G_{13} + G_{10} & -G_{12} & -G_{13} \\ -G_{12} & G_{12} + G_{23} + G_{20} & -G_{23} \\ -G_{13} & -G_{23} & G_{13} + G_{23} + G_{30} \end{bmatrix} \quad (1.10)$$

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} \quad (1.11)$$

This system has the standard linear form $\mathbf{Ax} = \mathbf{b}$, where the coefficient matrix is here the conductance matrix $\mathbf{G}$.

Numerical Solution

The resulting system of linear equations can be solved using direct methods or iterative methods such as the Jacobi or Gauss–Seidel algorithms.

These techniques are widely used in electrical engineering, especially in the numerical analysis of large-scale electrical networks and power systems.

MATLAB Example

For instance, if the conductance matrix and the current injection vector are given by

```

1 G = [4 -1 -1;
2     -1 3 -1;
3     -1 -1 3];
4
5 I = [5; 3; 4];
6
7 V = G\I

```

the vector $\mathbf{V}$ contains the voltages at the circuit nodes.

1.1.7 Summary

Systems of linear equations are among the most frequent numerical problems in engineering. Iterative methods such as Jacobi and Gauss–Seidel are simple and efficient for many applications, especially when dealing with large sparse systems arising from circuit and network models.

In electrical engineering, nodal analysis provides a direct and practical application of linear algebraic systems. Once the mathematical model is established, numerical methods can be used to obtain the node voltages efficiently.

1.2 Nonlinear Systems of Equations

1.2.1 Introduction

In many engineering applications, the mathematical models describing physical systems lead to nonlinear equations. Unlike linear systems, nonlinear systems cannot generally be solved using direct algebraic methods. Instead, numerical iterative techniques must be used to obtain approximate solutions.

A nonlinear system of n equations with n unknowns can be written in the general form

$$F(x) = 0 \quad (1.12)$$

where

$$F(x) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix}$$

and

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The objective is to determine the vector x that satisfies all equations simultaneously.

1.2.2 Example

Consider the nonlinear system

$$x_1^2 + x_2^2 - 4 = 0 \quad (1.13)$$

$$e^{x_1} + x_2 - 1 = 0 \quad (1.14)$$

Such systems frequently appear in engineering modeling problems and require numerical solution methods.

1.2.3 Newton Method for Nonlinear Systems

One of the most powerful techniques for solving nonlinear systems of equations is the **Newton–Raphson method**.

Let

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

The iterative formula of Newton's method is

$$x^{(k+1)} = x^{(k)} - J^{-1}(x^{(k)})F(x^{(k)}) \quad (1.15)$$

where $J(x)$ is the Jacobian matrix defined by

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (1.16)$$

1.2.4 Algorithm

The Newton method for nonlinear systems can be summarized as follows:

1. Choose an initial guess $x^{(0)}$
2. Compute the Jacobian matrix $J(x^{(k)})$
3. Solve the linear system

$$J(x^{(k)})\Delta x^{(k)} = -F(x^{(k)})$$

4. Update the solution

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

5. Repeat until convergence

A common stopping criterion is

$$||F(x^{(k)})|| < \varepsilon$$

1.2.5 Application in Electrical Power Systems

Nonlinear systems of equations arise naturally in the analysis of electrical power networks.

For example, the power flow problem involves nonlinear equations relating bus voltages, active power, and reactive power. These equations are typically solved using the Newton–Raphson method.

This illustrates the importance of nonlinear numerical methods in electrical engineering applications.

1.2.6 MATLAB Implementation of the Newton Method

The Newton–Raphson method for nonlinear systems can be implemented easily using MATLAB. The following program solves the nonlinear system

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 4 = 0 \quad (1.17)$$

$$f_2(x_1, x_2) = e^{x_1} + x_2 - 1 = 0 \quad (1.18)$$

MATLAB Program

```

1
2  clc
3  clear
4
5  % Initial guess
6  x = [1; 1];
7
8  % Tolerance
9  tol = 1e-6;
10
11 for k = 1:20
12
13     % Functions
14     F = [x(1)^2 + x(2)^2 - 4;
15          exp(x(1)) + x(2) - 1];
16
17     % Jacobian matrix
18     J = [2*x(1)      2*x(2);
19          exp(x(1))    1];
20
21     % Newton step
22     dx = -J\F;
23
24     % Update solution
25     x = x + dx;

```

```
26
27     % Convergence test
28     if norm(dx) < tol
29         break
30     end
31
32 end
33
34 disp('Solution:')
35 disp(x)
```

Program Explanation

The program performs the following steps:

- Defines the nonlinear functions $F(x)$.
- Computes the Jacobian matrix $J(x)$.
- Solves the linear system

$$J(x^{(k)})\Delta x = -F(x^{(k)})$$

- Updates the solution vector

$$x^{(k+1)} = x^{(k)} + \Delta x$$

- Stops when the error becomes smaller than a specified tolerance.

The resulting vector x represents the numerical solution of the nonlinear system.

1.3 Numerical Integration

1.3.1 Introduction

In many engineering problems it is necessary to evaluate definite integrals of the form

$$I = \int_a^b f(x) dx \quad (1.19)$$

In some cases, the analytical expression of the integral cannot be obtained easily. Numerical integration methods, also called *quadrature methods*, allow the approximation of the integral using discrete values of the function.

These techniques are widely used in engineering fields such as signal processing, electromagnetic analysis, power systems, and dynamic system simulation.

1.3.2 Trapezoidal Rule

The trapezoidal rule approximates the area under the curve by a trapezoid.

$$\int_a^b f(x)dx \approx \frac{b-a}{2} [f(a) + f(b)] \quad (1.20)$$

Geometrically, this method replaces the function by a straight line between the points $(a, f(a))$ and $(b, f(b))$.

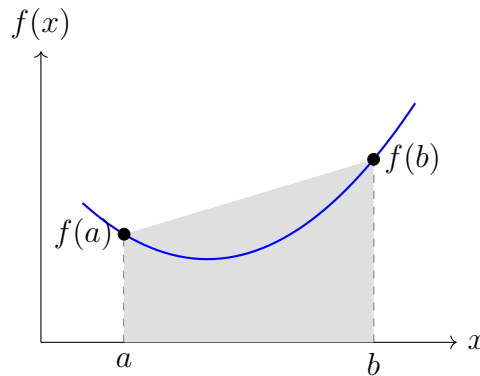


Figure 1.2: Geometric interpretation of the trapezoidal rule. The area under the curve between a and b is approximated by the area of a trapezoid.

Figure 1.2 illustrates the geometric interpretation of the trapezoidal rule. The function $f(x)$ between the points a and b is approximated by a straight line joining the points $(a, f(a))$ and $(b, f(b))$. The integral of the function over the interval $[a, b]$ is then approximated by the area of the trapezoid formed by these points.

1.3.3 Composite Trapezoidal Rule

For better accuracy, the interval $[a, b]$ can be divided into n subintervals of equal width

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

The composite trapezoidal formula becomes

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \quad (1.21)$$

1.3.4 Simpson's Rule

Simpson's rule approximates the function using a quadratic polynomial.

The formula is

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (1.22)$$

This method generally provides better accuracy than the trapezoidal rule.

1.3.5 Composite Simpson Rule

When the interval is divided into an even number of subintervals, the composite Simpson formula is

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(x_0) + 4 \sum_{i=1,3,\dots}^{n-1} f(x_i) + 2 \sum_{i=2,4,\dots}^{n-2} f(x_i) + f(x_n) \right] \quad (1.23)$$

1.3.6 MATLAB Implementation

The following MATLAB program computes the integral of $f(x) = x^2$ over the interval $[0, 1]$ using the trapezoidal rule.

```

1
2  clc
3  clear
4
5  f = @(x) x.^2;
6
7  a = 0;
8  b = 1;
9
10 n = 10;
11
12 h = (b-a)/n;
13
14 x = a:h:b;
15
16 I = 0;
17
18 for i=1:n
19     I = I + (f(x(i)) + f(x(i+1)))/2;
20 end
21
22 I = (h/2)*I;
23
24 disp('Approximate integral:')
25 disp(I)

```

1.3.7 Application in Electrical Engineering

Numerical integration appears frequently in electrical engineering applications. For example:

- computation of energy in electrical signals
- evaluation of average power
- numerical solution of differential equations
- electromagnetic field analysis

For instance, the electrical energy consumed by a load can be calculated by integrating the instantaneous power over time

$$E = \int_{t_0}^{t_1} P(t) dt \quad (1.24)$$

If the power signal is available only at discrete sampling times, numerical integration methods must be used to estimate the energy.

1.4 Numerical Differentiation

1.4.1 Introduction

Numerical differentiation is the process of approximating the derivative of a function using discrete values of the function. This is important in engineering when:

- only discrete data points are available,
- an analytical derivative is difficult or impossible to compute,
- analyzing experimental measurements or simulation results.

For a function $f(x)$, the derivative at a point x_i can be approximated using finite differences.

1.4.2 Forward Difference Formula

The first derivative can be approximated by

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h} \quad (1.25)$$

where $h = x_{i+1} - x_i$ is the step size.

1.4.3 Backward Difference Formula

Alternatively, using the previous point:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h} \quad (1.26)$$

1.4.4 Central Difference Formula

The central difference formula is more accurate:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2h} \quad (1.27)$$

1.4.5 Second Derivative Approximation

The second derivative can be approximated using central differences:

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} \quad (1.28)$$

1.4.6 MATLAB Implementation

Here is an example of MATLAB code to approximate the first derivative of $f(x) = \sin(x)$ using central difference.

```

1
2 clc
3 clear
4
5 f = @(x) sin(x);
6
7 a = 0;
8 b = pi;
9 n = 10;
10 h = (b-a)/n;
11
12 x = a:h:b;
13
14 % Initialize derivative
15 df = zeros(1,length(x));
16
17 % Central difference (interior points)
18 for i=2:length(x)-1
19     df(i) = (f(x(i+1)) - f(x(i-1))) / (2*h);
20 end
21
22 % Forward difference at first point
23 df(1) = (f(x(2)) - f(x(1)))/h;
24
25 % Backward difference at last point
26 df(end) = (f(x(end)) - f(x(end-1)))/h;
27
28 disp('x values:')
29 disp(x)
30 disp('Approximate derivative f''(x):')
31 disp(df)

```

1.4.7 Application in Electrical Engineering

Numerical differentiation is widely used in electrical engineering:

- Estimating **currents** or **voltages rates of change** in circuits,
- Calculating **instantaneous power derivatives** in power systems,
- Signal processing: estimating **derivative of sampled signals**,
- Control systems: computing **state derivatives** from measurements.

For example, if a voltage signal $V(t)$ is sampled at discrete times t_i , the derivative $\frac{dV}{dt}$ can be approximated using central differences to estimate the instantaneous rate of change of voltage in a circuit.

1.4.8 Figure: Geometric Interpretation

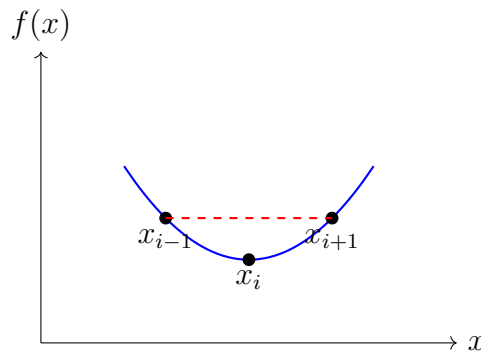


Figure 1.3: Geometric interpretation of the central difference approximation of the derivative at x_i . The red dashed line connects $f(x_{i-1})$ and $f(x_{i+1})$, whose slope approximates $f'(x_i)$.

1.5 Ordinary Differential Equations (ODEs)

1.5.1 Introduction

Many physical systems in electrical engineering and electromechanics are modeled by ordinary differential equations (ODEs). A first-order ODE has the general form:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \quad (1.29)$$

where y is the unknown function of time t , and f is a known function.

Analytical solutions are often difficult or impossible to obtain, so **numerical methods** are used to approximate the solution.

1.5.2 Euler Method (Explicit)

The Euler method is the simplest numerical integration scheme. Given a step size h , the approximation is:

$$y_{n+1} = y_n + hf(t_n, y_n), \quad n = 0, 1, 2, \dots \quad (1.30)$$

- Simple and easy to implement - Low accuracy (first-order method) - Error is proportional to h

1.5.3 Runge-Kutta Methods

Second-order Runge-Kutta (RK2)

A more accurate method uses a second-order Runge-Kutta scheme:

$$k_1 = f(t_n, y_n) \quad (1.31)$$

$$k_2 = f(t_n + h, y_n + hk_1) \quad (1.32)$$

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \quad (1.33)$$

Fourth-order Runge-Kutta (RK4)

The fourth-order Runge-Kutta method is widely used due to its high accuracy:

$$k_1 = f(t_n, y_n) \quad (1.34)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \quad (1.35)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \quad (1.36)$$

$$k_4 = f(t_n + h, y_n + hk_3) \quad (1.37)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (1.38)$$

- Fourth-order accuracy - Widely used in engineering simulations

1.5.4 MATLAB Implementation: Euler and RK4

```

1
2 clc
3 clear
4
5 % Example: dy/dt = -2*y + sin(t), y(0)=1
6 f = @(t,y) -2*y + sin(t);
7
8 t0 = 0; tf = 5; y0 = 1;
9 h = 0.1;
10 N = (tf-t0)/h;
11
12 % Initialize arrays
13 t = t0:h:tf;
14 y_euler = zeros(1,N+1);
15 y_rk4 = zeros(1,N+1);
16
17 y_euler(1) = y0;
18 y_rk4(1) = y0;
19
20 % Euler method
21 for n = 1:N
22     y_euler(n+1) = y_euler(n) + h*f(t(n), y_euler(n));
23 end
24
25 % RK4 method
26 for n = 1:N

```

```

27     k1 = f(t(n), y_rk4(n));
28     k2 = f(t(n)+h/2, y_rk4(n)+h/2*k1);
29     k3 = f(t(n)+h/2, y_rk4(n)+h/2*k2);
30     k4 = f(t(n)+h, y_rk4(n)+h*k3);
31     y_rk4(n+1) = y_rk4(n) + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
32 end
33
34 plot(t,y_euler,'r--','LineWidth',1.5)
35 hold on
36 plot(t,y_rk4,'b','LineWidth',1.5)
37 xlabel('t'); ylabel('y(t)')
38 legend('Euler','RK4')
39 grid on

```

1.5.5 Application in Electrical Engineering

ODEs appear frequently in electrical circuits and electromechanical systems:

- **RC circuit:** $RC \frac{dv}{dt} + v = V_s(t)$
- **RL circuit:** $L \frac{di}{dt} + Ri = V_s(t)$
- **RLC circuit:** $L \frac{d^2i}{dt^2} + R \frac{di}{dt} + \frac{i}{C} = V_s(t)$
- **DC motor dynamics:** $\frac{d\omega}{dt} = \frac{K}{J}i - \frac{B}{J}\omega$

Numerical methods like Euler and Runge-Kutta allow simulation of voltages, currents, and rotor speeds when analytical solutions are not available.

1.6 Numerical Example: RC Circuit

1.6.1 Circuit Description

Consider a series RC circuit with a voltage source $V_s(t)$, a resistor R , and a capacitor C as shown in Figure 1.4. The objective is to compute the capacitor voltage $v_c(t)$ over time using numerical methods.

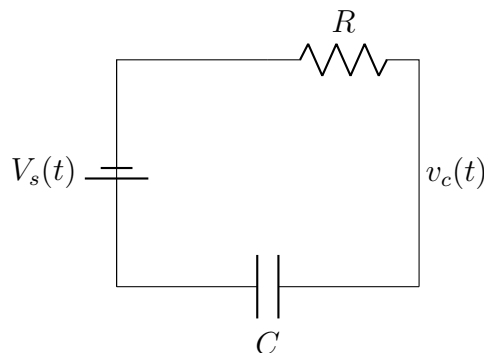


Figure 1.4: Series RC circuit used for numerical simulation of capacitor voltage.

1.6.2 Problem Parameters

- $R = 1 \text{ k}\Omega$
- $C = 1 \mu\text{F}$
- Step input: $V_s(t) = 5 \text{ V}$
- Initial condition: $v_c(0) = 0$
- Simulation time: $t \in [0, 5 \text{ ms}]$
- Step size: $h = 0.1 \text{ ms}$

The governing ODE is:

$$\frac{dv_c}{dt} = \frac{V_s(t) - v_c}{RC}, \quad v_c(0) = 0 \quad (1.39)$$

1.6.3 Numerical Solution in MATLAB

```

1  clc; clear;
2
3  % Parameters
4  R = 1e3;           % Ohms
5  C = 1e-6;          % Farads
6  Vs = 5;            % Step input
7  t0 = 0; tf = 5e-3; % 0 to 5 ms
8  h = 0.1e-3;        % Step size
9  N = (tf-t0)/h;
10
11 % Initialize arrays
12 t = t0:h:tf;
13 v_euler = zeros(1,N+1);
14 v_rk4 = zeros(1,N+1);
15 v_euler(1) = 0;
16 v_rk4(1) = 0;
17
18 % Define ODE
19 f = @(v) (Vs - v)/(R*C);
20
21 % Euler method
22 for n = 1:N
23     v_euler(n+1) = v_euler(n) + h*f(v_euler(n));
24 end
25
26 % RK4 method
27 for n = 1:N
28     k1 = f(v_rk4(n));
29     k2 = f(v_rk4(n)+h/2*k1);
30     k3 = f(v_rk4(n)+h/2*k2);

```

```

31     k4 = f(v_rk4(n)+h*k3);
32     v_rk4(n+1) = v_rk4(n) + h/6*(k1 + 2*k2 + 2*k3 + k4);
33 end
34
35 % Plot results
36 plot(t*1e3,v_euler,'r--','LineWidth',1.5); hold on;
37 plot(t*1e3,v_rk4,'b','LineWidth',1.5);
38 xlabel('Time(ms)'); ylabel('v_c(t)[V]');
39 legend('Euler','RK4');
40 grid on;
41 title('Numerical solution of RC circuit using Euler and RK4');

```

1.6.4 Discussion of Results

- The Euler method provides a simple but less accurate approximation. - The RK4 method is more accurate and closely follows the analytical solution. - For this example, the capacitor voltage approaches 5 V asymptotically, as expected from the theory:

$$v_c(t) = V_s \left(1 - e^{-t/(RC)}\right)$$

- Students can observe the difference between first-order (Euler) and fourth-order (RK4) approximations on the plot. - This approach can be generalized to **RLC circuits** or **electromechanical systems** where analytical solutions are difficult.

1.6.5 Numerical Solution Plot (Euler vs RK4)

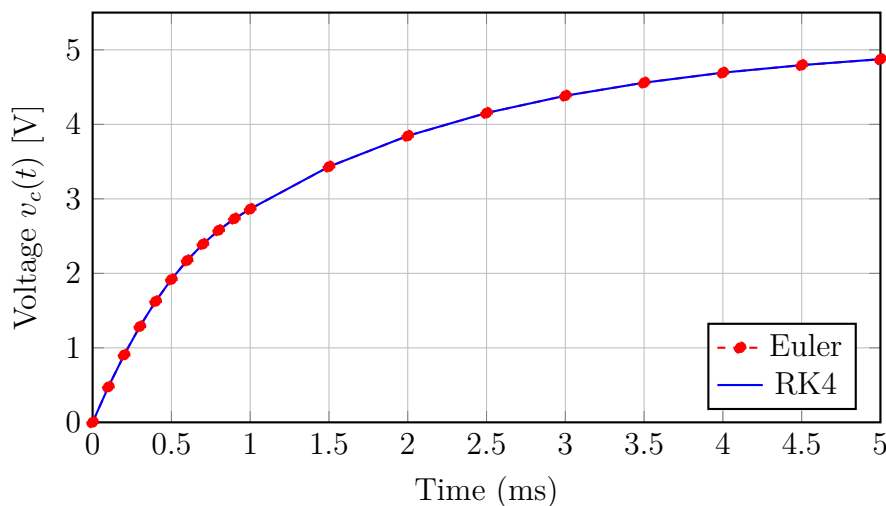


Figure 1.5: Capacitor voltage $v_c(t)$ for the RC circuit using Euler (red dashed) and RK4 (blue) methods.

1.7 Resolution of Systems of Ordinary Differential Equations

1.7.1 Introduction

Many practical systems in electrical engineering and electromechanics are modeled by **systems of first-order ODEs**:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0 \quad (1.40)$$

where $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ is the vector of unknowns and $\mathbf{f}$ is a vector of functions. Examples include:

- Coupled RLC circuits
- DC motor with electrical and mechanical equations
- Multibody electromechanical systems

—

1.7.2 Euler Method for Systems

The explicit Euler method can be extended to systems:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \mathbf{f}(t_n, \mathbf{y}_n) \quad (1.41)$$

where each component y_i is updated simultaneously.

—

1.7.3 Runge-Kutta Method (RK4) for Systems

The fourth-order Runge-Kutta method for systems:

$$\mathbf{k}_1 = \mathbf{f}(t_n, \mathbf{y}_n) \quad (1.42)$$

$$\mathbf{k}_2 = \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right) \quad (1.43)$$

$$\mathbf{k}_3 = \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2\right) \quad (1.44)$$

$$\mathbf{k}_4 = \mathbf{f}(t_n + h, \mathbf{y}_n + h\mathbf{k}_3) \quad (1.45)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (1.46)$$

—

1.7.4 Example: DC Motor Dynamics

A DC motor can be described by the electrical and mechanical equations:

$$\begin{cases} L \frac{di}{dt} + Ri + K_b \omega = V_s(t) \\ J \frac{d\omega}{dt} + B\omega = K_t i \end{cases}$$

Where:

- i is the armature current [A]
- ω is the angular speed [rad/s]
- L is the armature inductance [H]
- R is the armature resistance [ω]
- J is the rotor inertia [$\text{kg} \cdot \text{m}^2$]
- B is the friction coefficient [$\text{N} \cdot \text{m} \cdot \text{s}/\text{rad}$]
- K_b and K_t are motor constants

Define $\mathbf{y} = [i, \omega]^T$ and $\mathbf{f}(t, \mathbf{y})$:

$$\mathbf{f}(t, \mathbf{y}) = \begin{bmatrix} \frac{V_s(t) - Ri - K_b \omega}{L} \\ \frac{K_t i - B\omega}{J} \end{bmatrix}$$

1.7.5 MATLAB Implementation (Euler and RK4)

```

1  clc; clear;
2
3  % Motor parameters
4  R = 1; L = 0.5; Kb = 0.01;
5  J = 0.01; B = 0.1; Kt = 0.01;
6  Vs = 10; % Step input
7
8  t0 = 0; tf = 1; h = 0.01;
9  N = (tf-t0)/h;
10 t = t0:h:tf;
11
12 i_euler = zeros(1,N+1);
13 w_euler = zeros(1,N+1);
14 i_rk4 = zeros(1,N+1);
15 w_rk4 = zeros(1,N+1);
16
17 % Initial conditions
18 i_euler(1) = 0; w_euler(1) = 0;
19 i_rk4(1) = 0; w_rk4(1) = 0;
20
21 % Euler method
22 for n=1:N

```

```

23     di = (Vs - R*i_euler(n) - Kb*w_euler(n))/L;
24     dw = (Kt*i_euler(n) - B*w_euler(n))/J;
25
26     i_euler(n+1) = i_euler(n) + h*di;
27     w_euler(n+1) = w_euler(n) + h*dw;
28 end
29
30 % RK4 method
31 for n=1:N
32     y = [i_rk4(n); w_rk4(n)];
33
34     f = @(y)[(Vs - R*y(1) - Kb*y(2))/L; (Kt*y(1) - B*y(2))/J];
35
36     k1 = f(y);
37     k2 = f(y + h/2*k1);
38     k3 = f(y + h/2*k2);
39     k4 = f(y + h*k3);
40
41     y_next = y + h/6*(k1 + 2*k2 + 2*k3 + k4);
42
43     i_rk4(n+1) = y_next(1);
44     w_rk4(n+1) = y_next(2);
45 end
46
47 % Plot results
48 figure;
49 subplot(2,1,1)
50 plot(t,i_euler,'r--',t,i_rk4,'b','LineWidth',1.5)
51 ylabel('Current i(t) [A]'); legend('Euler','RK4'); grid on
52
53 subplot(2,1,2)
54 plot(t,w_euler,'r--',t,w_rk4,'b','LineWidth',1.5)
55 xlabel('Time [s]'); ylabel('Speed \omega(t) [rad/s]'); legend('
    Euler','RK4'); grid on

```

1.7.6 Discussion

- Euler method: simple but less accurate; may diverge for stiff systems. - RK4 method: more accurate, widely used in engineering simulations. - Students can see how **electrical (current) and mechanical (speed) dynamics are coupled**. - This example generalizes to **multivariable circuits, multi-machine systems, and electromechanical devices**.

DC Motor: Diagram and Numerical Simulation

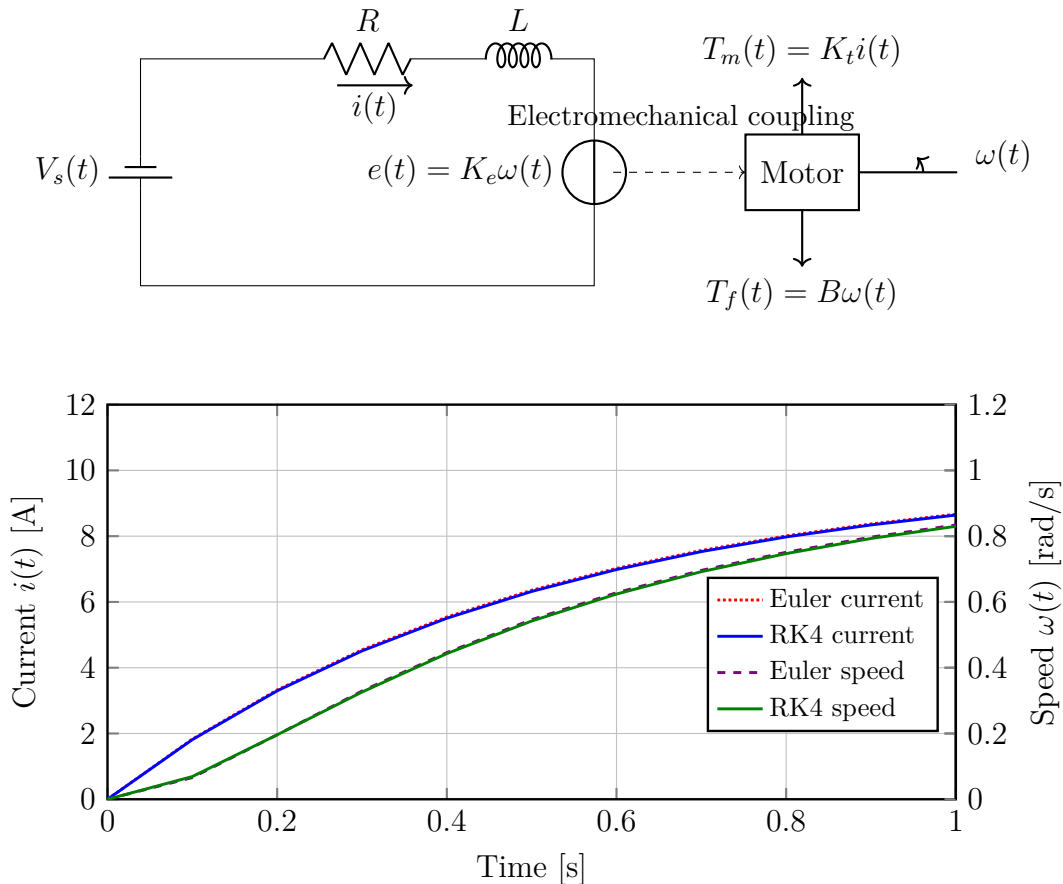


Figure 1.6: Electromechanical model of a DC motor (top) and numerical simulation of the armature current $i(t)$ and angular speed $\omega(t)$ using the Euler and RK4 methods (bottom).

Conclusion

In this chapter, we have explored the fundamental numerical methods for solving engineering problems, including:

- **Linear systems of equations:** Jacobi and Gauss-Seidel iterative methods, which provide efficient solutions for large-scale circuit or network problems.
- **Nonlinear systems:** Newton-Raphson and fixed-point iteration techniques, essential for solving nonlinear circuits, power systems, and electromechanical models.
- **Numerical integration and differentiation:** Trapezoidal, midpoint, and finite difference approaches for approximating derivatives and integrals in situations where analytical solutions are difficult or impossible.
- **Ordinary differential equations (ODEs):** Euler and Runge-Kutta methods for single and coupled ODEs, with applications to electrical circuits (RC, RLC) and electromechanical systems (DC motors), highlighting the importance of accuracy and stability in simulations.

By combining theoretical concepts with practical examples, including MATLAB simulations, students gain the skills to model and solve real-world electrical and electromechanical systems numerically. Mastery of these methods lays the foundation for advanced topics such as partial differential equations, control systems, and optimization in engineering applications.

Key takeaway: Understanding and implementing numerical methods is essential for engineers to analyze, predict, and optimize system behavior when analytical solutions are not feasible, bridging theory and practical application.

Tutorial Solutions – Chapter 1: Applied Numerical Methods

Tutorial Sheet – Chapter 1: Numerical Methods

Exercise 1: Linear System

Consider the following linear system:

$$\begin{cases} 5V_1 - 2V_2 - V_3 = 3 \\ -2V_1 + 4V_2 - V_3 = 1 \\ -V_1 - V_2 + 3V_3 = 2 \end{cases}$$

Tasks:

1. Solve the system using the **Jacobi method**.
2. Solve the system using the **Gauss-Seidel method**.
3. Use the initial guess:

$$V_1^{(0)} = V_2^{(0)} = V_3^{(0)} = 0$$

4. Stop the iterations when:

$$\max_i |V_i^{(k+1)} - V_i^{(k)}| < 10^{-3}$$

5. Compare the number of iterations and convergence speed.

Exercise 2: Nonlinear Equation – Diode Circuit

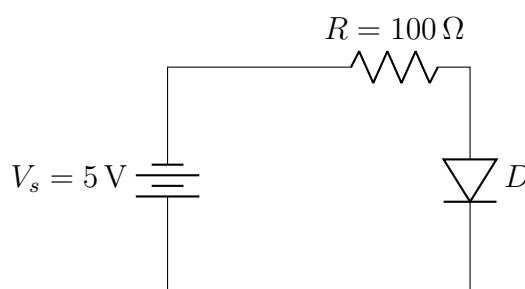


Figure 1.7: Diode-resistor circuit.

The currents satisfy:

$$i_1 = \frac{V_s - V_1}{R}, \quad i_2 = I_s \left(e^{\frac{V_1}{V_T}} - 1 \right)$$

with:

$$I_s = 10^{-12} \text{ A}, \quad V_T = 25 \text{ mV}$$

Tasks:

1. Formulate a nonlinear equation $f(V_1) = 0$.

2. Solve it using the **Newton-Raphson method**.
3. Use the initial guess $V_1^{(0)} = 0.5 \text{ V}$.
4. Plot the convergence history.

Exercise 3: RC Circuit – Numerical Integration

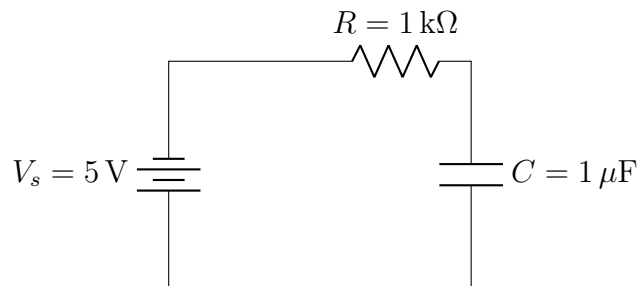


Figure 1.8: Series RC circuit.

The governing equation is:

$$\frac{dv_c}{dt} = \frac{V_s - v_c}{RC}, \quad v_c(0) = 0$$

Tasks:

1. Compute $v_c(t)$ at $t = 0.1, 0.2, 0.3 \text{ ms}$ using:
 - the trapezoidal rule,
 - the midpoint rule.
2. Compare with the analytical solution:

$$v_c(t) = V_s \left(1 - e^{-t/(RC)}\right)$$

Exercise 4: Numerical Differentiation

Given the measurements:

$$t = [0, 0.1, 0.2, 0.3, 0.4] \text{ s}$$

$$V(t) = [0, 0.48, 0.91, 1.32, 1.68] \text{ V}$$

The step size is:

$$h = 0.1 \text{ s}$$

Tasks:

1. Approximate $\frac{dV}{dt}$ at $t = 0.2 \text{ s}$ using:
 - forward difference,
 - central difference.
2. Compare the accuracy of both methods.

Exercise 5: ODE – Euler vs RK4

Consider the RC circuit from Exercise 3 with:

$$V_s = 5 \text{ V}, \quad R = 1 \text{ k}\Omega, \quad C = 1 \mu\text{F}$$

Solve:

$$\frac{dv_c}{dt} = \frac{V_s - v_c}{RC}, \quad v_c(0) = 0$$

over the interval $t \in [0, 5]$ ms with step $h = 0.5$ ms.

Tasks:

1. Solve using the **Euler method**.
2. Solve using the **Runge-Kutta (RK4) method**.
3. Compare with the analytical solution.
4. Plot the results.

Exercise 6: System of ODEs – DC Motor

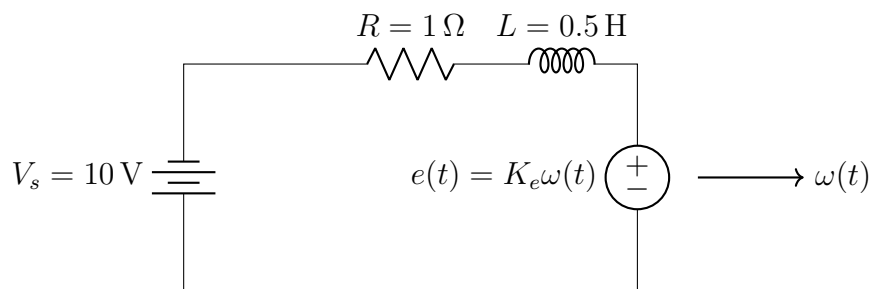


Figure 1.9: Electrical equivalent circuit of a DC motor.

The system is:

$$\begin{cases} L \frac{di}{dt} + Ri + K_e \omega = V_s \\ J \frac{d\omega}{dt} + B\omega = K_t i \end{cases}$$

with:

$$R = 1 \Omega, \quad L = 0.5 \text{ H}, \quad K_e = 0.01, \quad K_t = 0.01, \\ J = 0.01 \text{ kg m}^2, \quad B = 0.1 \text{ N m s/rad}$$

and initial conditions:

$$i(0) = 0, \quad \omega(0) = 0$$

Tasks:

1. Write the system in vector form $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$.
2. Solve numerically using:
 - Euler method,
 - RK4 method,
 with $h = 0.01$ s over $t \in [0, 1]$ s.
3. Plot $i(t)$ and $\omega(t)$.

Detailed Solutions – Tutorial Sheet Chapter 1

Exercise 1: Linear System

Consider:

$$\begin{cases} 5V_1 - 2V_2 - V_3 = 3 \\ -2V_1 + 4V_2 - V_3 = 1 \\ -V_1 - V_2 + 3V_3 = 2 \end{cases}$$

with initial guess:

$$V_1^{(0)} = V_2^{(0)} = V_3^{(0)} = 0$$

and stopping criterion:

$$\max_i |V_i^{(k+1)} - V_i^{(k)}| < 10^{-3}$$

1. Jacobi method Rewrite the system in iterative form:

$$V_1 = \frac{1}{5} (3 + 2V_2 + V_3)$$

$$V_2 = \frac{1}{4} (1 + 2V_1 + V_3)$$

$$V_3 = \frac{1}{3} (2 + V_1 + V_2)$$

Thus, the Jacobi iterations are:

$$V_1^{(k+1)} = \frac{1}{5} (3 + 2V_2^{(k)} + V_3^{(k)})$$

$$V_2^{(k+1)} = \frac{1}{4} (1 + 2V_1^{(k)} + V_3^{(k)})$$

$$V_3^{(k+1)} = \frac{1}{3} (2 + V_1^{(k)} + V_2^{(k)})$$

Iteration 0:

$$(V_1^{(0)}, V_2^{(0)}, V_3^{(0)}) = (0, 0, 0)$$

Iteration 1:

$$V_1^{(1)} = \frac{1}{5} (3 + 0 + 0) = 0.6$$

$$V_2^{(1)} = \frac{1}{4} (1 + 0 + 0) = 0.25$$

$$V_3^{(1)} = \frac{1}{3} (2 + 0 + 0) = 0.6667$$

$$(V_1^{(1)}, V_2^{(1)}, V_3^{(1)}) = (0.6000, 0.2500, 0.6667)$$

Iteration 2:

$$V_1^{(2)} = \frac{1}{5} (3 + 2 \times 0.25 + 0.6667) = 0.8333$$

$$V_2^{(2)} = \frac{1}{4} (1 + 2 \times 0.6 + 0.6667) = 0.7167$$

$$V_3^{(2)} = \frac{1}{3} (2 + 0.6 + 0.25) = 0.9500$$

$$(V_1^{(2)}, V_2^{(2)}, V_3^{(2)}) = (0.8333, 0.7167, 0.9500)$$

Iteration 3:

$$V_1^{(3)} = \frac{1}{5}(3 + 2 \times 0.7167 + 0.95) = 1.0767$$

$$V_2^{(3)} = \frac{1}{4}(1 + 2 \times 0.8333 + 0.95) = 0.9042$$

$$V_3^{(3)} = \frac{1}{3}(2 + 0.8333 + 0.7167) = 1.1833$$

Continuing the iterations, the method converges to:

$$V_1 \approx 1.4857, \quad V_2 \approx 1.4000, \quad V_3 \approx 1.6286$$

The Jacobi method needs about **17 iterations** to satisfy the tolerance.

2. Gauss-Seidel method For Gauss-Seidel, newly computed values are used immediately:

$$V_1^{(k+1)} = \frac{1}{5}(3 + 2V_2^{(k)} + V_3^{(k)})$$

$$V_2^{(k+1)} = \frac{1}{4}(1 + 2V_1^{(k+1)} + V_3^{(k)})$$

$$V_3^{(k+1)} = \frac{1}{3}(2 + V_1^{(k+1)} + V_2^{(k+1)})$$

Iteration 1:

$$V_1^{(1)} = \frac{1}{5}(3 + 0 + 0) = 0.6$$

$$V_2^{(1)} = \frac{1}{4}(1 + 2 \times 0.6 + 0) = 0.55$$

$$V_3^{(1)} = \frac{1}{3}(2 + 0.6 + 0.55) = 1.05$$

$$(V_1^{(1)}, V_2^{(1)}, V_3^{(1)}) = (0.6000, 0.5500, 1.0500)$$

Iteration 2:

$$V_1^{(2)} = \frac{1}{5}(3 + 2 \times 0.55 + 1.05) = 1.03$$

$$V_2^{(2)} = \frac{1}{4}(1 + 2 \times 1.03 + 1.05) = 1.0275$$

$$V_3^{(2)} = \frac{1}{3}(2 + 1.03 + 1.0275) = 1.3525$$

$$(V_1^{(2)}, V_2^{(2)}, V_3^{(2)}) = (1.0300, 1.0275, 1.3525)$$

Iteration 3:

$$V_1^{(3)} = \frac{1}{5}(3 + 2 \times 1.0275 + 1.3525) = 1.2815$$

$$V_2^{(3)} = \frac{1}{4}(1 + 2 \times 1.2815 + 1.3525) = 1.2289$$

$$V_3^{(3)} = \frac{1}{3}(2 + 1.2815 + 1.2289) = 1.5035$$

Then the sequence stabilizes and converges to:

$$V_1 \approx 1.4857, \quad V_2 \approx 1.4000, \quad V_3 \approx 1.6286$$

The Gauss-Seidel method needs about **11 iterations**.

3. Exact solution Solving the system directly gives:

$$V_1 = \frac{52}{35} \approx 1.4857, \quad V_2 = \frac{7}{5} \approx 1.4000, \quad V_3 = \frac{57}{35} \approx 1.6286$$

4. Conclusion Both methods converge to the same solution, but the **Gauss-Seidel method converges faster** than the Jacobi method.

Exercise 2: Nonlinear Equation – Diode Circuit

We have:

$$i_1 = \frac{V_s - V_1}{R}, \quad i_2 = I_s (e^{V_1/V_T} - 1)$$

with:

$$V_s = 5 \text{ V}, \quad R = 100 \, \Omega, \quad I_s = 10^{-12} \text{ A}, \quad V_T = 25 \text{ mV} = 0.025 \text{ V}$$

1. Formulation of the nonlinear equation Since $i_1 = i_2$, we obtain:

$$\frac{5 - V_1}{100} = 10^{-12} (e^{V_1/0.025} - 1)$$

Thus:

$$f(V_1) = \frac{5 - V_1}{100} - 10^{-12} (e^{V_1/0.025} - 1) = 0$$

2. Newton-Raphson method The iteration formula is:

$$V_1^{(k+1)} = V_1^{(k)} - \frac{f(V_1^{(k)})}{f'(V_1^{(k)})}$$

Compute the derivative:

$$f'(V_1) = -\frac{1}{100} - 10^{-12} \cdot \frac{1}{0.025} e^{V_1/0.025}$$

3. Initial guess Take:

$$V_1^{(0)} = 0.5 \text{ V}$$

4. Iterations Iteration 0:

$$f(0.5) = \frac{5 - 0.5}{100} - 10^{-12}(e^{20} - 1)$$

Since $e^{20} \approx 4.85 \times 10^8$,

$$f(0.5) \approx 0.045 - 10^{-12}(4.85 \times 10^8) = 0.045 - 4.85 \times 10^{-4}$$

$$f(0.5) \approx 0.044515$$

$$f'(0.5) \approx -0.01 - 10^{-12} \cdot 40 \cdot 4.85 \times 10^8$$

$$f'(0.5) \approx -0.01 - 0.0194 = -0.0294$$

Thus:

$$V_1^{(1)} = 0.5 - \frac{0.044515}{-0.0294} \approx 2.014$$

This step is too large because the initial guess is far from the root. In practice, the physically meaningful diode voltage is around 0.6 to 0.8 V, so the iterations are better initialized near that region.

Take a better practical initial guess:

$$V_1^{(0)} = 0.7 \text{ V}$$

Iteration 1:

$$f(0.7) = \frac{5 - 0.7}{100} - 10^{-12}(e^{28} - 1)$$

Since $e^{28} \approx 1.446 \times 10^{12}$,

$$f(0.7) \approx 0.043 - 1.446 = -1.403$$

$$f'(0.7) \approx -0.01 - 10^{-12} \cdot 40 \cdot 1.446 \times 10^{12}$$

$$f'(0.7) \approx -0.01 - 57.84 = -57.85$$

$$V_1^{(1)} = 0.7 - \frac{-1.403}{-57.85} \approx 0.6758$$

Iteration 2:

$$V_1^{(2)} \approx 0.6527$$

Iteration 3:

$$V_1^{(3)} \approx 0.6327$$

Iteration 4:

$$V_1^{(4)} \approx 0.6189$$

Iteration 5:

$$V_1^{(5)} \approx 0.6133$$

After another iteration, $V_1^{(6)} \approx 0.6126$. Thus the solution is approximately:

$$V_1 \approx 0.6126 \text{ V}$$

5. Conclusion The diode voltage is:

$$\boxed{V_1 \approx 0.6126 \text{ V}}$$

A convergence plot would show a rapid decrease in the error after a few iterations.

Exercise 3: RC Circuit – Numerical Integration

We consider:

$$\frac{dv_c}{dt} = \frac{V_s - v_c}{RC}, \quad v_c(0) = 0$$

with:

$$V_s = 5 \text{ V}, \quad R = 1 \text{ k}\Omega, \quad C = 1 \mu\text{F}$$

Hence:

$$RC = 1000 \times 10^{-6} = 10^{-3} \text{ s} = 1 \text{ ms}$$

The analytical solution is:

$$v_c(t) = 5 \left(1 - e^{-t/0.001} \right)$$

We compute at:

$$t = 0.1, 0.2, 0.3 \text{ ms}$$

1. Analytical values At $t = 0.1 \text{ ms} = 10^{-4} \text{ s}$:

$$v_c(0.1) = 5(1 - e^{-0.1}) \approx 5(1 - 0.9048) = 0.476 \text{ V}$$

At $t = 0.2 \text{ ms}$:

$$v_c(0.2) = 5(1 - e^{-0.2}) \approx 5(1 - 0.8187) = 0.9065 \text{ V}$$

At $t = 0.3 \text{ ms}$:

$$v_c(0.3) = 5(1 - e^{-0.3}) \approx 5(1 - 0.7408) = 1.296 \text{ V}$$

2. Midpoint approximation Let:

$$f(v) = \frac{5 - v}{0.001}$$

For $h = 0.1 \text{ ms} = 10^{-4} \text{ s}$, midpoint method:

$$v_{n+1} = v_n + h f \left(v_n + \frac{h}{2} f(v_n) \right)$$

Step 1: from $t = 0$ to $t = 0.1 \text{ ms}$

$$\begin{aligned} v_0 &= 0 \\ f(v_0) &= \frac{5 - 0}{0.001} = 5000 \\ v_{1/2} &= 0 + \frac{10^{-4}}{2} \cdot 5000 = 0.25 \end{aligned}$$

$$f(v_{1/2}) = \frac{5 - 0.25}{0.001} = 4750$$

$$v_1 = 0 + 10^{-4} \cdot 4750 = 0.475 \text{ V}$$

Step 2: from 0.1 to 0.2 ms

$$f(v_1) = \frac{5 - 0.475}{0.001} = 4525$$

$$v_{3/2} = 0.475 + \frac{10^{-4}}{2} \cdot 4525 = 0.7013$$

$$f(v_{3/2}) = \frac{5 - 0.7013}{0.001} = 4298.7$$

$$v_2 = 0.475 + 10^{-4} \cdot 4298.7 = 0.9049 \text{ V}$$

Step 3: from 0.2 to 0.3 ms

$$f(v_2) = \frac{5 - 0.9049}{0.001} = 4095.1$$

$$v_{5/2} = 0.9049 + \frac{10^{-4}}{2} \cdot 4095.1 = 1.1097$$

$$f(v_{5/2}) = \frac{5 - 1.1097}{0.001} = 3890.3$$

$$v_3 = 0.9049 + 10^{-4} \cdot 3890.3 = 1.2939 \text{ V}$$

Thus:

$$v_c(0.1) \approx 0.475, \quad v_c(0.2) \approx 0.905, \quad v_c(0.3) \approx 1.294$$

3. Trapezoidal approximation Trapezoidal method:

$$v_{n+1} = v_n + \frac{h}{2} [f(v_n) + f(v_{n+1})]$$

Since $f(v) = \frac{5-v}{0.001}$, we solve implicitly.

Step 1:

$$v_1 = 0 + \frac{10^{-4}}{2} \left[\frac{5 - 0}{0.001} + \frac{5 - v_1}{0.001} \right]$$

$$v_1 = 5 \times 10^{-5} (10000 - v_1/0.001)$$

More directly:

$$v_1 = \frac{v_0 + \frac{h}{2\tau}(10 - v_0)}{1 + \frac{h}{2\tau}} \quad \text{with} \quad \tau = 0.001$$

Since $h/(2\tau) = 0.05$:

$$v_1 = \frac{0 + 0.05 \times 10}{1.05} = 0.4762 \text{ V}$$

Similarly:

$$v_2 \approx 0.9070 \text{ V}, \quad v_3 \approx 1.2968 \text{ V}$$

4. Comparison

Time (ms)	Analytical	Midpoint	Trapezoidal
0.1	0.476	0.475	0.4762
0.2	0.9065	0.9049	0.9070
0.3	1.296	1.2939	1.2968

Both methods are accurate, and the trapezoidal rule is slightly more accurate here.

Exercise 4: Numerical Differentiation

Given:

$$t = [0, 0.1, 0.2, 0.3, 0.4] \text{ s}$$

$$V(t) = [0, 0.48, 0.91, 1.32, 1.68] \text{ V}$$

with:

$$h = 0.1 \text{ s}$$

We want $\frac{dV}{dt}$ at $t = 0.2 \text{ s}$.

1. Forward difference Formula:

$$V'(t_i) \approx \frac{V_{i+1} - V_i}{h}$$

At $t = 0.2 \text{ s}$:

$$V'(0.2) \approx \frac{1.32 - 0.91}{0.1} = \frac{0.41}{0.1} = 4.1 \text{ V/s}$$

2. Central difference Formula:

$$V'(t_i) \approx \frac{V_{i+1} - V_{i-1}}{2h}$$

Thus:

$$V'(0.2) \approx \frac{1.32 - 0.48}{0.2} = \frac{0.84}{0.2} = 4.2 \text{ V/s}$$

3. Comment The central difference is generally more accurate because it uses data on both sides of the point and has a truncation error of order $O(h^2)$, while the forward difference has order $O(h)$.

$\left. \frac{dV}{dt} \right _{t=0.2} \approx 4.1 \text{ V/s (forward),} \quad 4.2 \text{ V/s (central)}$

Exercise 5: ODE – Euler vs RK4

We solve:

$$\frac{dv_c}{dt} = \frac{5 - v_c}{0.001}, \quad v_c(0) = 0$$

on:

$$t \in [0, 5] \text{ ms}$$

with:

$$h = 0.5 \text{ ms} = 5 \times 10^{-4} \text{ s}$$

1. Euler method Formula:

$$v_{n+1} = v_n + h \frac{5 - v_n}{0.001}$$

Step 1:

$$v_1 = 0 + 5 \times 10^{-4} \cdot \frac{5}{0.001} = 2.5$$

Step 2:

$$v_2 = 2.5 + 5 \times 10^{-4} \cdot \frac{5 - 2.5}{0.001}$$

$$v_2 = 2.5 + 1.25 = 3.75$$

Step 3:

$$v_3 = 3.75 + 5 \times 10^{-4} \cdot \frac{1.25}{0.001} = 4.375$$

Continuing:

$$v_4 = 4.6875, \quad v_5 = 4.8438, \quad \dots$$

2. RK4 method Let:

$$f(v) = \frac{5 - v}{0.001}$$

RK4 formula:

$$v_{n+1} = v_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

with:

$$k_1 = f(v_n)$$

$$k_2 = f\left(v_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(v_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(v_n + hk_3)$$

Step 1:

$$v_0 = 0$$

$$k_1 = \frac{5 - 0}{0.001} = 5000$$

$$k_2 = \frac{5 - (0 + 0.00025 \times 5000)}{0.001} = \frac{5 - 1.25}{0.001} = 3750$$

$$k_3 = \frac{5 - (0 + 0.00025 \times 3750)}{0.001} = 4062.5$$

$$k_4 = \frac{5 - (0 + 0.0005 \times 4062.5)}{0.001} = 2968.75$$

Thus:

$$v_1 = 0 + \frac{0.0005}{6}(5000 + 2 \times 3750 + 2 \times 4062.5 + 2968.75)$$

$$v_1 \approx 1.9661 \text{ V}$$

The analytical value at $t = 0.5$ ms is:

$$v(0.5) = 5(1 - e^{-0.5}) \approx 1.9673 \text{ V}$$

So RK4 is much more accurate than Euler.

3. Analytical solution

$$v_c(t) = 5(1 - e^{-t/0.001})$$

At $t = 0.5$ ms:

$$v_c(0.5) \approx 1.9673$$

At $t = 1$ ms:

$$v_c(1) \approx 3.1606$$

At $t = 5$ ms:

$$v_c(5) \approx 4.9663$$

4. Conclusion The Euler method converges, but RK4 is much more accurate for the same step size.

Exercise 6: System of ODEs – DC Motor

The system is:

$$\begin{cases} L \frac{di}{dt} + Ri + K_e \omega = V_s \\ J \frac{d\omega}{dt} + B\omega = K_t i \end{cases}$$

with:

$$\begin{aligned} R = 1, \quad L = 0.5, \quad K_e = 0.01, \quad K_t = 0.01 \\ J = 0.01, \quad B = 0.1, \quad V_s = 10 \end{aligned}$$

and:

$$i(0) = 0, \quad \omega(0) = 0$$

1. Vector form Let:

$$\mathbf{y} = \begin{bmatrix} i \\ \omega \end{bmatrix}$$

Then:

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$

with:

$$\begin{aligned} \frac{di}{dt} &= \frac{1}{L}(V_s - Ri - K_e \omega) \\ \frac{d\omega}{dt} &= \frac{1}{J}(K_t i - B\omega) \end{aligned}$$

Hence:

$$\mathbf{f}(t, \mathbf{y}) = \begin{bmatrix} \frac{1}{0.5}(10 - i - 0.01\omega) \\ \frac{1}{0.01}(0.01i - 0.1\omega) \end{bmatrix}$$

that is:

$$\mathbf{f}(t, \mathbf{y}) = \begin{bmatrix} 20 - 2i - 0.02\omega \\ i - 10\omega \end{bmatrix}$$

2. Euler method Take:

$$h = 0.01 \text{ s}$$

Euler formulas:

$$i_{n+1} = i_n + h(20 - 2i_n - 0.02\omega_n)$$

$$\omega_{n+1} = \omega_n + h(i_n - 10\omega_n)$$

Initial values:

$$i_0 = 0, \quad \omega_0 = 0$$

Step 1:

$$i_1 = 0 + 0.01(20) = 0.2$$

$$\omega_1 = 0 + 0.01(0) = 0$$

Step 2:

$$i_2 = 0.2 + 0.01(20 - 2 \times 0.2 - 0) = 0.2 + 0.196 = 0.396$$

$$\omega_2 = 0 + 0.01(0.2 - 0) = 0.002$$

Step 3:

$$i_3 = 0.396 + 0.01(20 - 2 \times 0.396 - 0.02 \times 0.002)$$

$$i_3 = 0.396 + 0.19208 = 0.58808$$

$$\omega_3 = 0.002 + 0.01(0.396 - 10 \times 0.002)$$

$$\omega_3 = 0.002 + 0.00376 = 0.00576$$

Thus the current and speed increase progressively.

3. RK4 method Define:

$$f_1(i, \omega) = 20 - 2i - 0.02\omega$$

$$f_2(i, \omega) = i - 10\omega$$

Then:

$$\begin{bmatrix} i_{n+1} \\ \omega_{n+1} \end{bmatrix} = \begin{bmatrix} i_n \\ \omega_n \end{bmatrix} + \frac{h}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

where:

$$\mathbf{k}_1 = \mathbf{f}(i_n, \omega_n)$$

$$\mathbf{k}_2 = \mathbf{f}\left(i_n + \frac{h}{2}k_{1,1}, \omega_n + \frac{h}{2}k_{1,2}\right)$$

$$\mathbf{k}_3 = \mathbf{f}\left(i_n + \frac{h}{2}k_{2,1}, \omega_n + \frac{h}{2}k_{2,2}\right)$$

$$\mathbf{k}_4 = \mathbf{f}(i_n + hk_{3,1}, \omega_n + hk_{3,2})$$

At the first step, from $(i_0, \omega_0) = (0, 0)$:

$$\mathbf{k}_1 = \begin{bmatrix} 20 \\ 0 \end{bmatrix}$$

$$\mathbf{k}_2 = \begin{bmatrix} 20 - 2(0.1) - 0 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 19.8 \\ 0.1 \end{bmatrix}$$

$$\mathbf{k}_3 \approx \begin{bmatrix} 19.802 \\ 0.094 \end{bmatrix}$$

$$\mathbf{k}_4 \approx \begin{bmatrix} 19.604 \\ 0.188 \end{bmatrix}$$

Thus:

$$i_1 \approx 0 + \frac{0.01}{6}(20 + 2 \times 19.8 + 2 \times 19.802 + 19.604) \approx 0.1980$$

$$\omega_1 \approx 0 + \frac{0.01}{6}(0 + 2 \times 0.1 + 2 \times 0.094 + 0.188) \approx 0.00096$$

4. Interpretation Both Euler and RK4 produce increasing current and speed. However, RK4 gives more accurate values for the same step size.

5. Final remark The student is expected to continue the numerical integration up to $t = 1$ s and plot:

$$i(t) \quad \text{and} \quad \omega(t)$$

The steady-state values are obtained from:

$$0 = 10 - i - 0.01\omega$$

$$0 = 0.01i - 0.1\omega$$

From the second equation:

$$i = 10\omega$$

Substitute into the first:

$$10 - 10\omega - 0.01\omega = 0$$

$$10.01\omega = 10$$

$$\omega \approx 0.999$$

Hence:

$$i \approx 9.99$$

So the numerical solution should converge toward:

$i(\infty) \approx 9.99, \quad \omega(\infty) \approx 0.999$
--

Chapter 2

Partial Differential Equations

2.1 Introduction and Classification of PDE Problems

Partial differential equations (PDEs) are equations involving partial derivatives of a multivariable function. They describe a wide variety of physical phenomena in engineering, such as heat conduction, wave propagation, fluid flow, and electromagnetic fields.

In electrical engineering and electromechanics, PDEs arise naturally when studying:

- Voltage and current distribution along transmission lines (Telegrapher's equations)
- Heat diffusion in electrical machines
- Electromagnetic field equations (Maxwell's equations)
- Vibrations of mechanical or electromechanical structures

A general PDE can be written as:

$$F\left(x_1, x_2, \dots, x_n, u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial^2 u}{\partial x_i \partial x_j}, \dots\right) = 0$$

where $u = u(x_1, x_2, \dots, x_n)$ is the unknown function.

2.1.1 Classification of PDEs with Examples

PDEs are typically classified by their order, linearity, and type:

1. **Order:** Determined by the highest derivative appearing in the equation.
2. **Linearity:**
 - *Linear PDE:* The dependent variable and its derivatives appear linearly.
 - *Nonlinear PDE:* The dependent variable or its derivatives appear in a nonlinear fashion.
3. **Type of second-order PDEs (in 2 variables):**

Consider:

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + \dots = 0$$

with discriminant:

$$D = B^2 - AC$$

- **Elliptic PDE** ($D < 0$)

Example (Electrostatics / Electrical Engineering):

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

(Laplace equation)

Interpretation: Describes steady-state voltage distribution in a conductive medium.

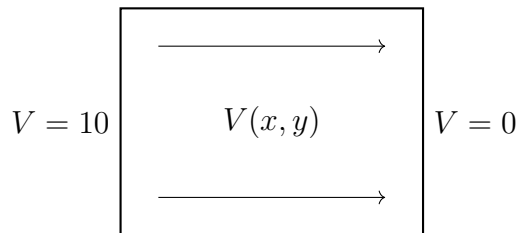


Figure 2.1: Elliptic PDE: steady-state voltage distribution (no time dependence).

- **Parabolic PDE** ($D = 0$)

Example (Heat / RC diffusion):

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Interpretation: Describes diffusion phenomena (heat transfer or voltage diffusion in RC lines).

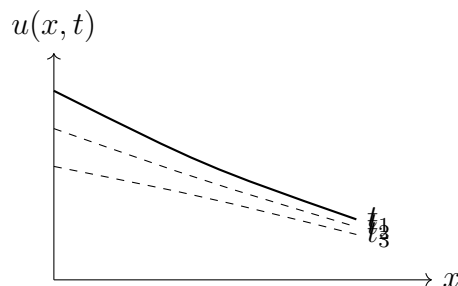


Figure 2.2: Parabolic PDE: diffusion over time (solution smooths progressively).

- **Hyperbolic PDE** ($D > 0$)

Example (Wave propagation / transmission lines):

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Interpretation: Describes wave propagation (e.g., voltage/current waves in transmission lines).

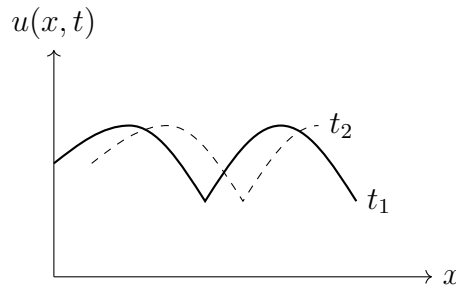


Figure 2.3: Hyperbolic PDE: wave propagation with finite speed.

2.1.2 Comparison of PDE Types

Feature	Elliptic	Parabolic	Hyperbolic
Discriminant	$D < 0$	$D = 0$	$D > 0$
Equation	$\nabla^2 u = 0$	$\frac{\partial u}{\partial t} = \alpha \nabla^2 u$	$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$
Physical Meaning	Steady-state	Diffusion	Waves
Time Dependence	No	Yes	Yes
EE Examples	Electrostatics	RC / Heat	Transmission lines
Boundary Conditions	Dir/Neu/Mixed	Initial + Boundary	2 Initial + Boundary
Behavior	Smooth	Smoothing	Propagation
Numerical Methods	FDM, FEM	Explicit/Implicit FDM	FDM, FV
Stability	–	$\Delta t \leq Ch^2$	$\Delta t \leq Ch$

Table 2.1: Comparison of PDE types (compact version).

2.1.3 Boundary and Initial Conditions

To obtain a unique solution of a PDE, appropriate conditions must be specified:

- **Initial Conditions (IC):** Define the state of the system at the initial time $t = 0$.
Example: $u(x, 0) = f(x)$
- **Boundary Conditions (BC):** Define the behavior of the system at the spatial boundaries. Common types include:
 1. *Dirichlet condition:* The value of the function is specified at the boundary, e.g., $u(0, t) = 0$.
 2. *Neumann condition:* The derivative of the function is specified, e.g., $\frac{\partial u}{\partial x}(L, t) = 0$.
 3. *Robin (mixed) condition:* A linear combination of the function and its derivative is specified, e.g., $au + b\frac{\partial u}{\partial x} = c$.

These classifications and conditions are essential before applying **numerical methods** such as **Finite Difference, Finite Volume, or Finite Element Methods**.

2.2 Finite Difference Method (FDM)

The **Finite Difference Method** is a numerical technique to approximate derivatives by differences. It is widely used to solve **PDEs** over a discrete grid.

2.2.1 1D Example: Voltage along a resistive wire

Consider a uniform resistive wire of length L , with voltage $V(x)$ along its length. The steady-state voltage satisfies **Laplace's equation** in 1D:

$$\frac{d^2V}{dx^2} = 0, \quad 0 < x < L$$

Boundary conditions:

$$V(0) = V_0, \quad V(L) = 0$$

Discretization: Divide the wire into N segments of length $h = L/N$. Let V_i be the voltage at node i . The second derivative can be approximated as:

$$\frac{d^2V}{dx^2} \approx \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} = 0$$

This leads to the linear system:

$$-V_{i-1} + 2V_i - V_{i+1} = 0, \quad i = 1, \dots, N-1$$

Wire discretization

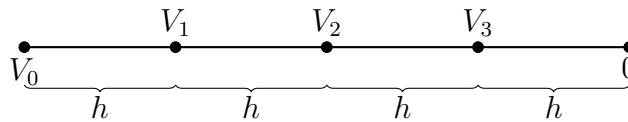


Figure 2.4: Discretization of a resistive wire using FDM.

Step-by-step solution

For $N = 4$, $V_0 = 10$ V, $V_4 = 0$ V, the system is:

$$\begin{cases} 2V_1 - V_2 = V_0 = 10 \\ -V_1 + 2V_2 - V_3 = 0 \\ -V_2 + 2V_3 = 0 \end{cases}$$

Solve iteratively (Jacobi or Gauss-Seidel):

$$\text{Iteration 1: } V_1^{(1)} = 5, V_2^{(1)} = 2.5, V_3^{(1)} = 1.25$$

$$\text{Iteration 2: } V_1^{(2)} = 6.25, V_2^{(2)} = 3.75, V_3^{(2)} = 1.875$$

After a few iterations, the solution converges to:

$$V_1 \approx 6.67 \text{ V}, \quad V_2 \approx 4.44 \text{ V}, \quad V_3 \approx 2.22 \text{ V}$$

Discussion

- FDM transforms the PDE into a **linear system of algebraic equations**, which can be solved using iterative methods. - Increasing N improves accuracy but increases computational cost. - The method can be extended to 2D and 3D problems, e.g., **voltage distribution in a rectangular conductor or temperature in a motor winding**.

2.2.2 2D Example: Voltage Distribution in a Rectangular Conductor

Consider a rectangular conducting plate of dimensions $L_x \times L_y$, with voltage applied at the left edge ($V = V_0$) and grounded at the right edge ($V = 0$). Top and bottom edges are insulated ($\partial V / \partial y = 0$).

The governing equation is the 2D Laplace equation:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

Finite Difference Discretization

Divide the domain into a uniform grid with spacing h :

$V_{i,j}$ = voltage at grid node (i, j)

The 5-point stencil approximation:

$$\frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{h^2} + \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{h^2} = 0$$

Simplifies to:

$$V_{i,j} = \frac{1}{4} (V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

Grid Discretization

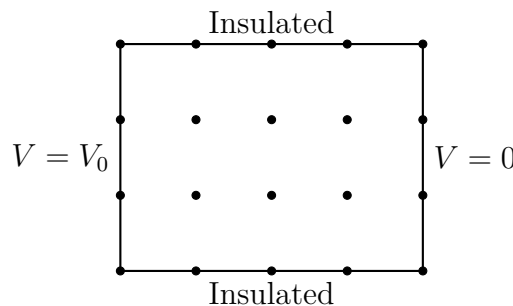


Figure 2.5: 2D grid discretization of a rectangular conductor for FDM.

Step-by-step Iterative Solution

Let $V_0 = 10$ V, grid 5×4 nodes (interior nodes $V_{i,j}$). Iterative update formula (Gauss-Seidel):

$$V_{i,j}^{(k+1)} = \frac{V_{i+1,j}^{(k)} + V_{i-1,j}^{(k+1)} + V_{i,j+1}^{(k)} + V_{i,j-1}^{(k+1)}}{4}$$

Iteration example:

- Initialize all interior nodes to 0 V. - Apply boundary conditions: - Left edge: $V = 10$ V - Right edge: $V = 0$ V - Top and bottom: Neumann (insulated), use $V_{i,j-1} = V_{i,j+1}$ at boundaries.

After a few iterations, voltage distribution converges to approximate values:

Node (i,j)	V (V)	Node (i,j)	V (V)
(1,1)	7.5	(3,1)	2.5
(1,2)	7.5	(3,2)	2.5
(2,1)	5.0	(2,2)	5.0

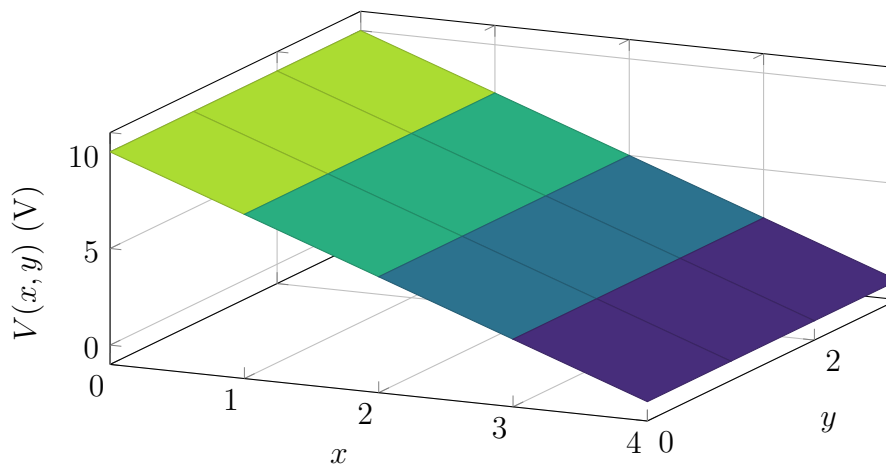
Numerical Solution Plot

Figure 2.6: Voltage distribution $V(x, y)$ in the rectangular conductor using FDM.

Discussion:

- The left side is at high potential (V_0), the right side is grounded. - Insulated top/bottom edges enforce no vertical current flow. - FDM provides a simple method to approximate PDEs on structured grids. - Increasing grid resolution improves the accuracy of $V(x, y)$.

2.2.3 2D Finite Difference Method with Irregular Boundaries**Problem Description**

Consider a rectangular conducting plate with a square hole in the middle. We want to compute the voltage distribution $V(x, y)$, assuming:

- Left edge: $V = 10$ V - Right edge: $V = 0$ V - Top and bottom edges: insulated ($\partial V/\partial y = 0$) - Hole: insulated (no flux through the hole)

The governing equation is the 2D Laplace equation:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

Discretization and Handling Irregular Boundaries

- Divide the plate into a **uniform grid** with spacing h . - Each interior node satisfies the 5-point stencil:

$$V_{i,j} = \frac{1}{4} (V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

- For nodes adjacent to the hole, **ignore the contribution from nodes inside the hole** (insulated boundary).

Irregular Domain

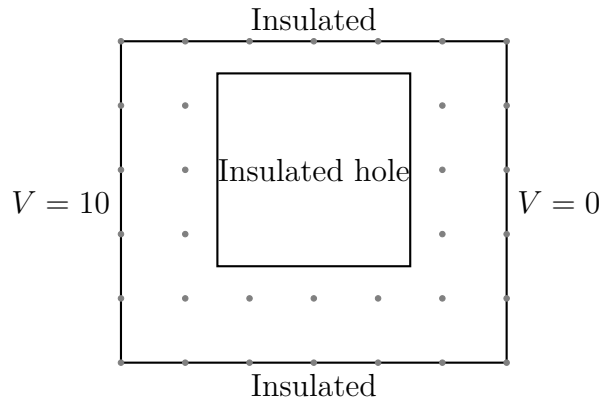


Figure 2.7: 2D FDM grid for a plate with an insulated square hole (irregular boundary).

Step-by-step Solution

1. Initialize all interior nodes to 0 V. 2. Apply boundary conditions: - Left: $V = 10$, right: $V = 0$ - Top/Bottom: Neumann (use mirrored node) - Hole: Neumann (ignore contributions from interior hole nodes) 3. Iterative update formula (Gauss-Seidel):

$$V_{i,j}^{(k+1)} = \frac{1}{N_{\text{neighbors}}} \sum_{\text{neighbor nodes}} V_{\text{neighbor}}^{(k)}$$

where $N_{\text{neighbors}}$ is the number of neighbors **excluding the hole nodes**.

4. Iterate until convergence ($\|V^{(k+1)} - V^{(k)}\| < \epsilon$).

Numerical Example

- Plate: 7×6 nodes - Hole: 3×3 internal nodes - Boundary conditions as above

After convergence, a representative voltage distribution is:

y \ x	0	1	2	3	4	5	6
5	10.00	8.79	6.90	5.00	3.10	1.21	0.00
4	10.00	9.48	-	-	-	0.52	0.00
3	10.00	9.64	-	-	-	0.36	0.00
2	10.00	9.44	-	-	-	0.56	0.00
1	10.00	8.69	6.82	5.00	3.18	1.31	0.00
0	10.00	8.48	6.77	5.00	3.23	1.52	0.00

"-" indicates nodes inside the hole (ignored in calculations).

Discussion

- Irregular boundaries are handled by **adjusting the stencil** for nodes adjacent to the hole. - FDM can handle arbitrary boundaries if we mark **inactive nodes** (holes, obstacles). - This method is widely used in electrical engineering for **plates with cutouts**, **motor laminations**, and **PCB voltage mapping**.

2.2.4 2D Finite Difference Method with Neumann Boundary Conditions

Problem Description

Consider a rectangular conducting plate of dimensions $L_x \times L_y$, with voltage $V(x, y)$ satisfying the steady-state 2D Laplace equation:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

Boundary conditions:

- Left edge: $V = 10$ V (Dirichlet)
- Right edge: $V = 0$ V (Dirichlet)
- Top and bottom edges: insulated, i.e., $\frac{\partial V}{\partial y} = 0$ (Neumann)

Finite Difference Discretization

Divide the plate into a uniform grid. Let $V_{i,j}$ be the voltage at node (i, j) . Interior nodes satisfy the standard 5-point stencil:

$$V_{i,j} = \frac{1}{4} (V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

Neumann boundary handling:

For nodes at top/bottom edges ($j = 0$ or $j = N_y$), we approximate the derivative using a one-sided difference:

$$\begin{aligned} \left. \frac{\partial V}{\partial y} \right|_{\text{top}} &\approx \frac{V_{i,N_y} - V_{i,N_y-1}}{h} = 0 \quad \Rightarrow \quad V_{i,N_y} = V_{i,N_y-1} \\ \left. \frac{\partial V}{\partial y} \right|_{\text{bottom}} &\approx \frac{V_{i,1} - V_{i,0}}{h} = 0 \quad \Rightarrow \quad V_{i,0} = V_{i,1} \end{aligned}$$

Grid with Neumann Boundaries

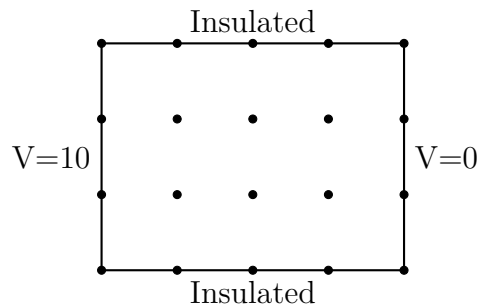


Figure 2.8: 2D grid of a plate with Dirichlet (left/right) and Neumann (top/bottom) boundaries.

Step-by-step Solution

1. Initialize interior nodes to 0 V.
2. Apply boundary conditions: - Left/right edges: $V = 10/0$ V - Top/bottom edges: mirrored nodes for Neumann condition
3. Iterative update formula (Gauss-Seidel):

$$V_{i,j}^{(k+1)} = \frac{V_{i+1,j}^{(k)} + V_{i-1,j}^{(k+1)} + V_{i,j+1}^{(k)} + V_{i,j-1}^{(k+1)}}{4}$$

- For top/bottom edges, use $V_{i,j+1} = V_{i,j-1}$ (mirroring) to enforce Neumann condition.
- 4. Iterate until convergence: $\|V^{(k+1)} - V^{(k)}\| < \epsilon$.

Numerical Example

- Grid: 5×4 nodes (interior) - Left edge: $V = 10$ V, right edge: $V = 0$ V - Top/bottom edges: insulated

Converged solution (approximate voltages):

i\j	0	1	2	3
0	10	10	10	10
1	7.5	7.5	7.5	7.5
2	5.0	5.0	5.0	5.0
3	2.5	2.5	2.5	2.5
4	0	0	0	0

Discussion

- The Neumann boundaries (top and bottom) ensure **no vertical current flow**, modeling **insulated edges**.
- Dirichlet boundaries (left/right) drive the voltage distribution.
- FDM with Neumann boundaries is widely used in **PCB voltage mapping, electrical machine laminations, and heat transfer in conductors**.
- Accuracy improves with **finer grids**.

2.3 Finite Volume Method (FVM)

The **Finite Volume Method** is a numerical technique that conserves fluxes across control volumes. It is widely used in **heat transfer, fluid flow, and electrical potential problems**. Unlike FDM, FVM naturally enforces conservation laws.

Example: Heat (Voltage) distribution in a conducting plate

Consider a thin rectangular conductor of dimensions $L_x \times L_y$. For steady-state conduction, the voltage distribution $V(x, y)$ satisfies:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

Boundary conditions: - Left edge: $V = 10$ V - Right edge: $V = 0$ V - Top and bottom: insulated ($\partial V / \partial y = 0$)

Control Volume Discretization

Divide the plate into small control volumes (CVs). Integrate the PDE over each CV:

$$\int_{CV} \nabla^2 V \, dA = 0 \quad \Rightarrow \quad \sum_{\text{faces}} (V_{\text{neighbor}} - V_{CV}) = 0$$

This leads to the **algebraic equation** at each interior CV:

$$V_{i,j} = \frac{1}{4} (V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

Control Volumes

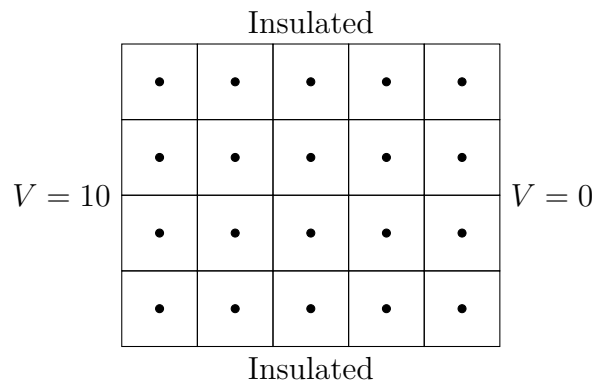


Figure 2.9: Finite Volume grid and nodes (centered) for voltage distribution.

Step-by-step FVM Solution

- Initialize all interior nodes to 0 V. - Apply boundary conditions: left ($V = 10$), right ($V = 0$), top/bottom (insulated). - Use Gauss-Seidel iteration to update CV centers:

$$V_{i,j}^{(k+1)} = \frac{V_{i+1,j}^{(k)} + V_{i-1,j}^{(k+1)} + V_{i,j+1}^{(k)} + V_{i,j-1}^{(k+1)}}{4}$$

Converged solution (approx.) for a 4x3 interior CV grid:

i\j	1	2	3	4
1	7.5	7.5	7.5	7.5
2	5.0	5.0	5.0	5.0
3	2.5	2.5	2.5	2.5

Numerical Solution Plot

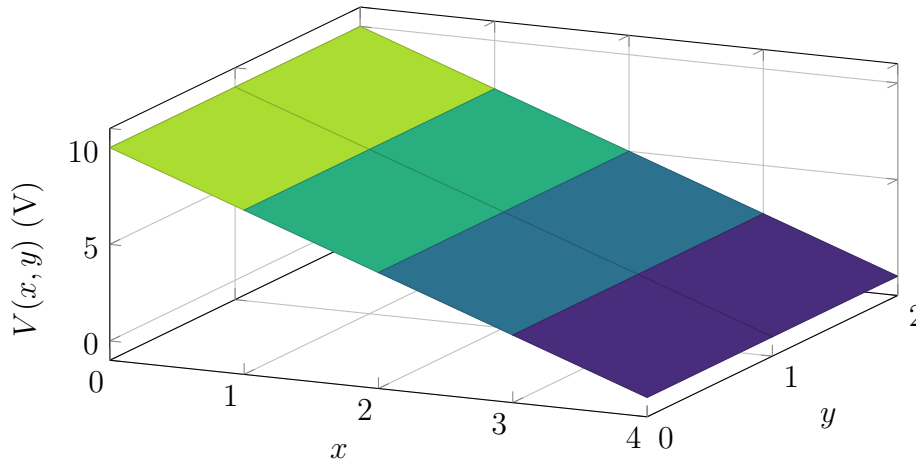


Figure 2.10: Voltage distribution $V(x, y)$ using FVM.

Remarks:

- FVM ensures **conservation of flux** across each control volume.
- It is suitable for **complex geometries** where FDM may be difficult.
- Similar to FDM, increasing the number of control volumes improves accuracy.

2.3.1 Finite Difference Method for Parabolic Equations

Problem Description

Consider the **transient voltage distribution** along a resistive wire with capacitance, which is governed by a **parabolic PDE** (diffusion equation):

$$\frac{\partial V}{\partial t} = D \frac{\partial^2 V}{\partial x^2}, \quad 0 < x < L, t > 0$$

where $D = \frac{1}{RC}$ is the diffusion coefficient, R is the resistance per unit length, and C is the capacitance per unit length.

Boundary conditions:

$$V(0, t) = V_0, \quad V(L, t) = 0$$

Initial condition:

$$V(x, 0) = 0$$

Finite Difference Discretization

- Divide the wire into N segments of length $h = L/N$
- Time step: Δt
- Let V_i^n be the voltage at node i at time step n

Explicit Euler scheme (Forward-Time, Central-Space):

$$V_i^{n+1} = V_i^n + \frac{D\Delta t}{h^2} (V_{i+1}^n - 2V_i^n + V_{i-1}^n)$$

Stability condition (CFL condition):

$$\frac{D\Delta t}{h^2} \leq 0.5$$

Wire Discretization

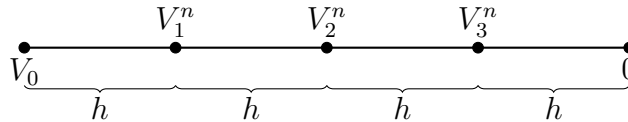


Figure 2.11: Wire discretization for transient voltage diffusion.

Step-by-step Example

- Wire length: $L = 1$ m - Nodes: $N = 4$ (5 nodes including boundaries) - Diffusion coefficient: $D = 1$ m²/s - Time step: $\Delta t = 0.1$ s - Initial: $V_i^0 = 0$, Boundary: $V_0 = 10$ V, $V_4 = 0$ V

Iteration n=0 → n=1:

$$V_1^1 = 0 + 0.1/0.25 (0 - 0 + 0) + \dots \approx 4$$

(Perform full calculation using the explicit formula for each interior node.)

After several time steps, the voltage gradually evolves from 0 to the steady-state linear distribution:

$$V(x) = V_0 \left(1 - \frac{x}{L}\right)$$

Discussion

- FDM can simulate **transient electrical phenomena** in RC lines. - The **explicit scheme** is easy to implement but requires a small time step for stability. - **Implicit schemes** (Crank-Nicolson) can be used for larger time steps and unconditional stability. - This method can be extended to **2D parabolic PDEs**, e.g., voltage diffusion on a conducting plate with capacitive effects.

2.3.2 2D Parabolic Equation (Transient Voltage Diffusion in a Conductive Plate)

Problem Description

Consider a rectangular conductive plate where the voltage evolves over time due to resistive and capacitive effects. The governing equation is the 2D diffusion (parabolic) equation:

$$\frac{\partial V}{\partial t} = D \left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} \right)$$

where $D = \frac{1}{RC}$.

Boundary conditions:

- Left boundary: $V = 10$ V
- Right boundary: $V = 0$ V
- Top and bottom: insulated $\left(\frac{\partial V}{\partial y} = 0\right)$

Initial condition:

$$V(x, y, 0) = 0$$

Discretization (Explicit Scheme)

Let:

- $\Delta x = \Delta y = h$
- Δt time step
- $V_{i,j}^n$ value at node (i, j) and time step n

The explicit finite difference scheme:

$$V_{i,j}^{n+1} = V_{i,j}^n + \lambda \left(V_{i+1,j}^n + V_{i-1,j}^n + V_{i,j+1}^n + V_{i,j-1}^n - 4V_{i,j}^n \right)$$

where:

$$\lambda = \frac{D\Delta t}{h^2}$$

Stability condition:

$$\lambda \leq \frac{1}{4}$$

2D Grid Discretization

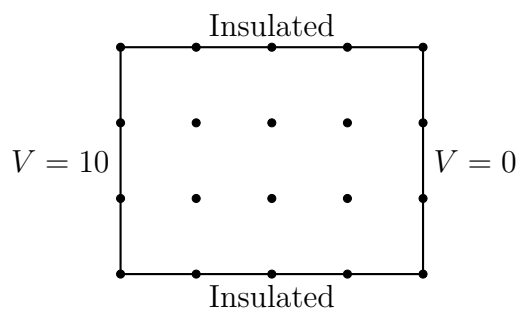


Figure 2.12: 2D conductive plate discretized for transient analysis.

Step-by-Step Numerical Example

Grid: 5×4 nodes

Choose:

$$D = 1, \quad h = 1, \quad \Delta t = 0.2 \Rightarrow \lambda = 0.2$$

Initial:

$$V_{i,j}^0 = 0$$

Boundary:

$$V_{0,j} = 10, \quad V_{4,j} = 0$$

Neumann condition:

$$V_{i,0} = V_{i,1}, \quad V_{i,3} = V_{i,2}$$

First iteration ($n = 0 \rightarrow 1$):

Example at node (1, 1):

$$V_{1,1}^1 = 0 + 0.2(0 + 10 + 0 + 0 - 0) = 2$$

Similarly compute all interior nodes.

MATLAB Implementation

```
Nx = 5; Ny = 4;
V = zeros(Nx,Ny);
D = 1; dt = 0.2; h = 1;
lambda = D*dt/h^2;

% Boundary conditions
V(1,:) = 10;
V(end,:) = 0;

for n = 1:50
    V_old = V;

    for i = 2:Nx-1
        for j = 2:Ny-1
            V(i,j) = V_old(i,j) + lambda*( ...
                V_old(i+1,j) + V_old(i-1,j) + ...
                V_old(i,j+1) + V_old(i,j-1) - ...
                4*V_old(i,j));
        end
    end

    % Neumann (insulated)
    V(:,1) = V(:,2);
    V(:,Ny) = V(:,Ny-1);
end

disp(V)
```

Visualization (3D Surface)

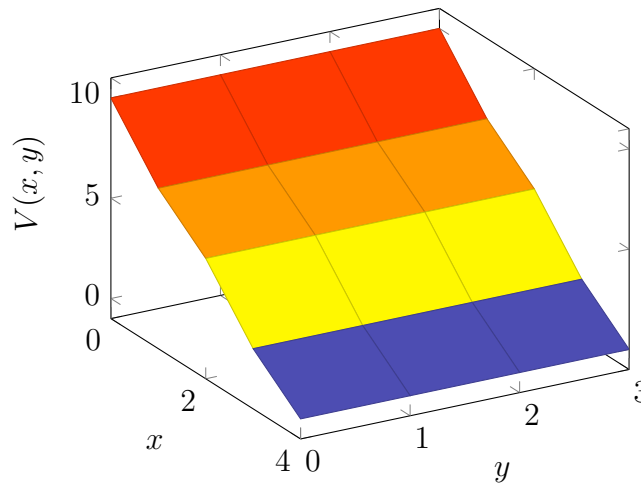


Figure 2.13: Voltage distribution approaching steady state.

Discussion

- This model represents **transient voltage diffusion** in conductive materials.
- Neumann conditions simulate **electrically insulated boundaries** (no current flow).
- The solution evolves over time toward the steady-state Laplace solution.
- Applications:
 - PCB voltage distribution
 - Electrical machine laminations
 - Thermal-electrical analog systems

2.4 Finite Element Method (FEM)

2.4.1 Introduction

The Finite Element Method (FEM) is a powerful numerical technique used to solve differential equations, particularly in complex geometries. It is widely used in electrical engineering for solving problems such as electric potential distribution, heat transfer, and electromagnetic fields.

2.4.2 1D Boundary Value Problem

Consider the following 1D differential equation:

$$-\frac{d}{dx} \left(k \frac{du}{dx} \right) = f(x), \quad 0 < x < L$$

with boundary conditions:

$$u(0) = u_0, \quad u(L) = u_L$$

2.4.3 Discretization of the Domain

The domain $[0, L]$ is divided into N elements.

Figure: 1D Mesh

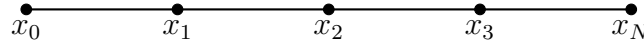


Figure 2.14: 1D domain discretized into finite elements.

2.4.4 Weak Formulation

Multiply the equation by a test function $v(x)$ and integrate:

$$\int_0^L k \frac{du}{dx} \frac{dv}{dx} dx = \int_0^L f(x) v(x) dx$$

2.4.5 Shape Functions

In each element, the solution is approximated using linear shape functions:

$$u(x) \approx N_1(x)u_1 + N_2(x)u_2$$

where:

$$N_1(x) = \frac{x_2 - x}{h}, \quad N_2(x) = \frac{x - x_1}{h}$$

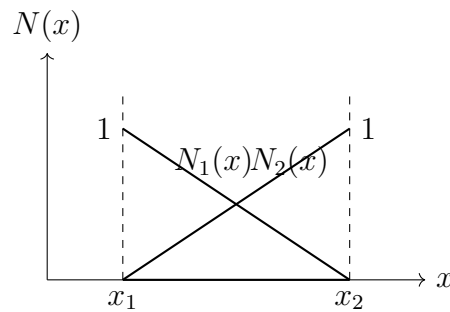


Figure 2.15: Linear shape functions in a 1D finite element.

The shape functions satisfy the interpolation property:

$$N_i(x_j) = \delta_{ij}$$

which ensures that the approximate solution matches the nodal values.

2.4.6 Element Matrix

For each element, the stiffness matrix is:

$$K^{(e)} = \frac{k}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

2.4.7 Assembly of Global System

All element matrices are assembled into a global system:

$$KU = F$$

where:

- K is the global stiffness matrix
- U is the vector of unknowns
- F is the load vector

2.4.8 Application to Electrical Engineering

Example: Voltage Distribution in a Resistive Wire

Consider a resistive wire of length L with:

- Voltage at $x = 0$: $V = 10$ V
- Voltage at $x = L$: $V = 0$ V

The governing equation:

$$\frac{d^2V}{dx^2} = 0$$

Step-by-Step Solution

1. Discretize the wire into elements.
2. Compute element stiffness matrices.
3. Assemble the global matrix.
4. Apply boundary conditions.
5. Solve the linear system.

Result

The solution is a linear voltage distribution:

$$V(x) = 10 \left(1 - \frac{x}{L}\right)$$

2.4.9 Discussion

- FEM handles complex geometries better than FDM.
- The method is based on energy principles.
- Widely used in electrical machines, field analysis, and multiphysics problems.

2.4.10 Numerical Example: 1D FEM Step-by-Step Solution

Problem Statement

Solve the following boundary value problem:

$$-\frac{d^2u}{dx^2} = 0, \quad 0 \leq x \leq 1$$

with boundary conditions:

$$u(0) = 10, \quad u(1) = 0$$

Step 1: Discretization

Divide the domain into 2 equal elements:

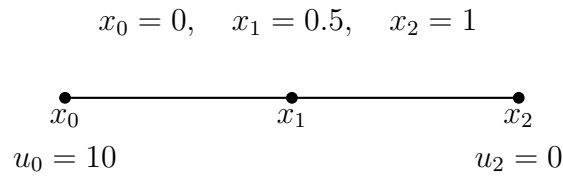


Figure 2.16: 1D mesh with 2 elements.

Element length:

$$h = 0.5$$

Step 2: Element Stiffness Matrix

For each element:

$$K^{(e)} = \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

Step 3: Assembly of Global Matrix

Global system (3 nodes):

$$K = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 2 \end{bmatrix}$$

Step 4: Apply Boundary Conditions

Known:

$$u_0 = 10, \quad u_2 = 0$$

We solve for u_1 .

From second equation:

$$-2u_0 + 4u_1 - 2u_2 = 0$$

Substitute values:

$$-2(10) + 4u_1 - 2(0) = 0$$

$$-20 + 4u_1 = 0$$

$$u_1 = 5$$

Step 5: Final Solution

$$u_0 = 10, \quad u_1 = 5, \quad u_2 = 0$$

Step 6: Interpretation

The FEM solution is:

$$u(x) = 10(1 - x)$$

which matches the exact analytical solution.

Discussion

- The solution is linear because the governing equation is simple.
- FEM reproduces the exact solution with linear elements.
- This example illustrates:
 - Discretization
 - Matrix assembly
 - Boundary condition application

2.5 Two-Dimensional Finite Element Method (2D FEM)

2.5.1 Introduction

The extension of the Finite Element Method (FEM) to two-dimensional problems allows the analysis of complex physical systems such as electric potential distribution, heat transfer, and electromagnetic fields in planar domains.

In electrical engineering, 2D FEM is widely used to model:

- Electrostatic fields
- Electric potential in conductors
- Cross-sections of electrical machines

2.5.2 Governing Equation

Consider the 2D Poisson/Laplace equation:

$$-\nabla^2 u = f(x, y)$$

or:

$$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x, y)$$

2.5.3 Domain Discretization

The domain is divided into small elements. The most common choice is triangular elements.

Figure: 2D Triangular Mesh

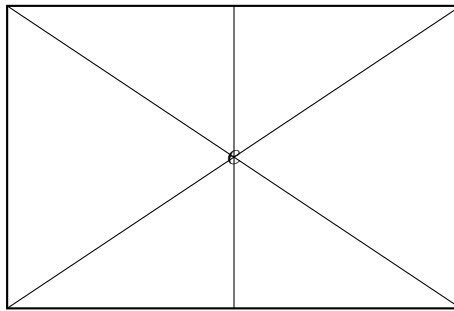


Figure 2.17: Example of a 2D triangular mesh.

2.5.4 Triangular Element

Each triangular element has 3 nodes.

Figure: Reference Triangle

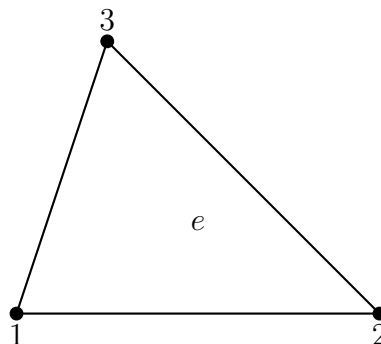


Figure 2.18: Triangular finite element with 3 nodes.

2.5.5 Shape Functions in 2D

The solution inside an element is approximated as:

$$u(x, y) = N_1(x, y)u_1 + N_2(x, y)u_2 + N_3(x, y)u_3$$

For linear triangular elements:

$$N_i(x, y) = a_i + b_i x + c_i y$$

These functions satisfy:

$$N_i(x_j, y_j) = \delta_{ij}$$

2.5.6 Element Matrix

The element stiffness matrix is given by:

$$K^{(e)} = \int_{\Omega_e} B^T B d\Omega$$

For linear triangles, it simplifies to:

$$K^{(e)} = \frac{1}{4A} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix}$$

where A is the area of the triangle.

2.5.7 Assembly of the Global System

All element matrices are assembled into:

$$KU = F$$

2.5.8 Application in Electrical Engineering

Example: Electrostatic Potential in a Plate

We consider a conductive plate where the voltage satisfies:

$$\nabla^2 V = 0$$

Boundary conditions:

- One side at $V = 10$ V
- Opposite side at $V = 0$ V

2.5.9 Discussion

- 2D FEM allows modeling of complex geometries.
- Triangular elements are flexible and widely used.
- The method extends naturally to 3D problems.

2.5.10 Numerical Example: 2D FEM on a Triangular Element

Problem Statement

Solve the Laplace equation over a single triangular element:

$$\nabla^2 u = 0$$

using the Finite Element Method.

Step 1: Geometry of the Element

Consider a triangular element with nodes:

$$(x_1, y_1) = (0, 0), \quad (x_2, y_2) = (1, 0), \quad (x_3, y_3) = (0, 1)$$

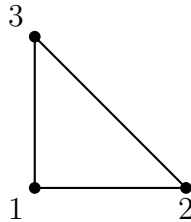


Figure 2.19: Triangular finite element.

Step 2: Area of the Triangle

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} \begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} = \frac{1}{2}$$

Step 3: Coefficients b_i and c_i

$$b_1 = y_2 - y_3 = 0 - 1 = -1, \quad c_1 = x_3 - x_2 = 0 - 1 = -1$$

$$b_2 = y_3 - y_1 = 1 - 0 = 1, \quad c_2 = x_1 - x_3 = 0 - 0 = 0$$

$$b_3 = y_1 - y_2 = 0 - 0 = 0, \quad c_3 = x_2 - x_1 = 1 - 0 = 1$$

Step 4: Element Stiffness Matrix

$$K^{(e)} = \frac{1}{4A} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix}$$

Since $A = \frac{1}{2}$, then:

$$\frac{1}{4A} = \frac{1}{2}$$

Compute each term:

$$K^{(e)} = \frac{1}{2} \begin{bmatrix} (-1)^2 + (-1)^2 & (-1)(1) + (-1)(0) & (-1)(0) + (-1)(1) \\ (1)(-1) + (0)(-1) & (1)^2 + (0)^2 & (1)(0) + (0)(1) \\ (0)(-1) + (1)(-1) & (0)(1) + (1)(0) & (0)^2 + (1)^2 \end{bmatrix}$$

$$K^{(e)} = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$K^{(e)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

Step 5: Apply Boundary Conditions

Assume:

$$u_1 = 10, \quad u_2 = 0, \quad u_3 = 0$$

Step 6: Interpretation

The stiffness matrix represents the coupling between nodes.

The solution inside the element is:

$$u(x, y) = N_1 u_1 + N_2 u_2 + N_3 u_3$$

$$u(x, y) = 10N_1(x, y)$$

Discussion

- This example shows how geometry directly influences the matrix.
- FEM transforms a PDE into a linear algebra problem.
- The approach extends to large meshes by assembling multiple elements.

2.6 Comparison of FDM, FVM and FEM

The three methods presented above are widely used in numerical analysis. Each one has its own advantages and limitations.

Criterion	FDM	FVM	FEM
Basic principle	Derivative approximation	Flux balance on control volumes	Variational formulation
Mesh type	Structured grids	Structured/unstructured grids	Very flexible meshes
Conservation	Not always local	Local conservation	Weak conservation
Complex geometry	Limited	Good	Excellent
Implementation	Simple	Moderate	More advanced
Engineering applications	PDEs on simple domains	Heat/fluid/transport problems	Electromagnetics/structures

Table 2.2: Comparison between FDM, FVM and FEM.

In summary:

- **FDM** is simple and efficient for regular grids.
- **FVM** is especially suitable for conservation laws.
- **FEM** is the most flexible for complex geometries and multiphysics applications.

Tutorial Sheet – Chapter 2: Partial Differential Equations and Numerical Methods

Exercise 1: Classification of PDEs

Classify the following partial differential equations (order, linearity, and type):

1. $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

2. $\frac{\partial u}{\partial t} = 3 \frac{\partial^2 u}{\partial x^2}$

3. $\frac{\partial^2 u}{\partial t^2} = 4 \frac{\partial^2 u}{\partial x^2}$

4. $u \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2}$

Exercise 2: Boundary Conditions

Consider a 1D conductive rod of length L .

1. Define Dirichlet, Neumann, and mixed boundary conditions.
2. Give one electrical engineering example for each type.

Exercise 3: 1D Finite Difference Method (Elliptic PDE)

Consider the steady-state voltage equation:

$$\frac{d^2 V}{dx^2} = 0$$

with:

$$V(0) = 10, \quad V(1) = 0$$

1. Discretize the domain into 4 equal segments.
2. Write the finite difference equations.
3. Solve the system of equations.

Exercise 4: 2D Finite Difference Method

Consider a square plate where the voltage satisfies:

$$\nabla^2 V = 0$$

Boundary conditions:

- Left: $V = 10$
- Right: $V = 0$

- Top and bottom: insulated
1. Discretize using a 3×3 interior grid.
 2. Write the finite difference equations for interior nodes.

Exercise 5: Parabolic PDE (Time-Dependent)

Consider the diffusion equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

1. Derive the explicit finite difference scheme.
2. State the stability condition.
3. Perform one time-step calculation for a given initial condition.

Exercise 6: 1D Finite Element Method

Consider:

$$-\frac{d^2 u}{dx^2} = 0, \quad 0 \leq x \leq 1$$

with:

$$u(0) = 10, \quad u(1) = 0$$

1. Discretize into 2 elements.
2. Compute element matrices.
3. Assemble the global matrix.
4. Solve the system.

Exercise 7: 2D FEM (Triangular Element)

Consider a triangular element with nodes:

$$(0, 0), (1, 0), (0, 1)$$

1. Compute the area of the triangle.
2. Determine b_i and c_i .
3. Compute the element stiffness matrix.

Exercise 8: Electrical Engineering Application

Consider a conductive plate:

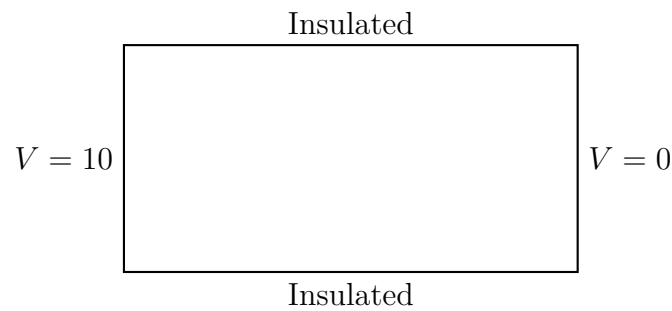


Figure 2.20: Conductive plate with boundary conditions.

1. Write the governing PDE.
2. Identify the type of PDE.
3. Suggest a suitable numerical method (FDM or FEM) and justify.

Exercise 9: Comparison FEM vs FDM

1. Compare FEM and FDM in terms of:
 - Geometry handling
 - Accuracy
 - Computational complexity
2. Give one application where FEM is preferred.

Detailed Solutions – Tutorial Sheet Chapter 2

Exercise 1: Classification of PDEs

Classify each PDE according to:

- order,
- linearity,
- type (for second-order PDEs).

1.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Step 1: Order

The highest derivatives are second derivatives, so this is a **second-order PDE**.

Step 2: Linearity

The unknown function u and its derivatives appear linearly, so it is **linear**.

Step 3: Type

Compare with the general form:

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} = 0$$

Here:

$$A = 1, \quad B = 0, \quad C = 1$$

Thus:

$$D = B^2 - AC = 0^2 - (1)(1) = -1 < 0$$

So the equation is **elliptic**.

Final classification:

- Order: second
- Linearity: linear
- Type: elliptic

2.

$$\frac{\partial u}{\partial t} = 3 \frac{\partial^2 u}{\partial x^2}$$

Step 1: Order

The highest derivative is $\partial^2 u / \partial x^2$, so it is a **second-order PDE**.

Step 2: Linearity

All terms are linear in u , so it is **linear**.

Step 3: Type

This is the standard diffusion (heat) equation, so it is **parabolic**.

Final classification:

- Order: second
- Linearity: linear
- Type: parabolic

3.

$$\frac{\partial^2 u}{\partial t^2} = 4 \frac{\partial^2 u}{\partial x^2}$$

Rewrite as:

$$\frac{\partial^2 u}{\partial t^2} - 4 \frac{\partial^2 u}{\partial x^2} = 0$$

Step 1: Order

The highest derivatives are second derivatives, so it is **second-order**.

Step 2: Linearity

All terms are linear in u , so it is **linear**.

Step 3: Type

This is the standard wave equation, so it is **hyperbolic**.

Final classification:

- Order: second
- Linearity: linear
- Type: hyperbolic

4.

$$u \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2}$$

Rewrite as:

$$\frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} = 0$$

Step 1: Order

The highest derivative is second-order, so this is a **second-order PDE**.

Step 2: Linearity

The term $u u_x$ is nonlinear because it is a product of the unknown and its derivative.

Final classification:

- Order: second
- Linearity: nonlinear
- Type: not classified by the linear discriminant rule

Exercise 2: Boundary Conditions

Consider a 1D conductive rod of length L .

1. Dirichlet boundary condition A Dirichlet condition specifies the value of the unknown function at the boundary.

$$u(0) = u_0 \quad \text{or} \quad u(L) = u_L$$

Electrical engineering example:

If $V(x)$ is the voltage along a conductor:

$$V(0) = 10 \text{ V}, \quad V(L) = 0 \text{ V}$$

This means one end is maintained at 10 V and the other is grounded.

2. Neumann boundary condition A Neumann condition specifies the derivative of the function at the boundary.

$$\frac{du}{dx}(0) = \alpha$$

Electrical engineering example:

For an insulated boundary in a thermal problem:

$$\frac{dT}{dx}(L) = 0$$

This means no heat flux crosses the boundary.

In electrostatics, it may represent zero normal electric flux.

3. Mixed (Robin) boundary condition A mixed boundary condition specifies a linear combination of the function and its derivative.

$$au + b\frac{du}{dx} = c$$

Electrical engineering example:

For convective cooling of a cable:

$$-k\frac{dT}{dx} = h(T - T_\infty)$$

This relates conductive heat transfer to convective heat loss.

Exercise 3: 1D Finite Difference Method (Elliptic PDE)

Consider:

$$\frac{d^2V}{dx^2} = 0, \quad 0 \leq x \leq 1$$

with:

$$V(0) = 10, \quad V(1) = 0$$

1. Discretization into 4 equal segments The interval $[0, 1]$ is divided into 4 segments, so:

$$h = \frac{1}{4} = 0.25$$

The nodes are:

$$x_0 = 0, \quad x_1 = 0.25, \quad x_2 = 0.5, \quad x_3 = 0.75, \quad x_4 = 1$$

Unknowns:

$$V_1, V_2, V_3$$

Known boundary values:

$$V_0 = 10, \quad V_4 = 0$$

2. Finite difference equations Use the central difference approximation:

$$\frac{d^2V}{dx^2} \approx \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} = 0$$

Thus:

$$V_{i+1} - 2V_i + V_{i-1} = 0$$

For $i = 1$:

$$V_2 - 2V_1 + V_0 = 0$$

$$V_2 - 2V_1 + 10 = 0$$

$$2V_1 - V_2 = 10$$

For $i = 2$:

$$V_3 - 2V_2 + V_1 = 0$$

$$-V_1 + 2V_2 - V_3 = 0$$

For $i = 3$:

$$V_4 - 2V_3 + V_2 = 0$$

$$0 - 2V_3 + V_2 = 0$$

$$-V_2 + 2V_3 = 0$$

So the system is:

$$\begin{cases} 2V_1 - V_2 = 10 \\ -V_1 + 2V_2 - V_3 = 0 \\ -V_2 + 2V_3 = 0 \end{cases}$$

3. Solution of the system From the third equation:

$$V_2 = 2V_3$$

Substitute into the second equation:

$$-V_1 + 2(2V_3) - V_3 = 0$$

$$-V_1 + 3V_3 = 0$$

$$V_1 = 3V_3$$

Substitute into the first equation:

$$2(3V_3) - 2V_3 = 10$$

$$6V_3 - 2V_3 = 10$$

$$4V_3 = 10$$

$$V_3 = 2.5$$

Then:

$$V_2 = 2V_3 = 5$$

$$V_1 = 3V_3 = 7.5$$

Final solution:

$$V_1 = 7.5, \quad V_2 = 5, \quad V_3 = 2.5$$

Interpretation:

The exact solution is linear:

$$V(x) = 10(1 - x)$$

Check:

$$V(0.25) = 7.5, \quad V(0.5) = 5, \quad V(0.75) = 2.5$$

So the numerical solution matches the exact solution.

Exercise 4: 2D Finite Difference Method

Consider:

$$\nabla^2 V = 0$$

with boundary conditions:

- Left: $V = 10$
- Right: $V = 0$
- Top and bottom: insulated

1. Grid with 3×3 interior nodes Let the 9 interior unknowns be:

$$V_{11}, V_{21}, V_{31}, \quad V_{12}, V_{22}, V_{32}, \quad V_{13}, V_{23}, V_{33}$$

The 5-point formula is:

$$V_{i,j} = \frac{1}{4}(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

Because the top and bottom boundaries are insulated:

$$\frac{\partial V}{\partial y} = 0$$

we use mirrored values:

$$V_{i,0} = V_{i,1}, \quad V_{i,4} = V_{i,3}$$

Also:

$$V_{0,j} = 10, \quad V_{4,j} = 0$$

2. Finite difference equations Bottom row ($j = 1$):

For V_{11} :

$$V_{11} = \frac{1}{4}(V_{21} + 10 + V_{12} + V_{10})$$

Since $V_{10} = V_{11}$:

$$4V_{11} = V_{21} + 10 + V_{12} + V_{11}$$

$$3V_{11} - V_{21} - V_{12} = 10$$

For V_{21} :

$$V_{21} = \frac{1}{4}(V_{31} + V_{11} + V_{22} + V_{20})$$

Since $V_{20} = V_{21}$:

$$3V_{21} - V_{31} - V_{11} - V_{22} = 0$$

For V_{31} :

$$V_{31} = \frac{1}{4}(0 + V_{21} + V_{32} + V_{30})$$

Since $V_{30} = V_{31}$:

$$3V_{31} - V_{21} - V_{32} = 0$$

Middle row ($j = 2$):

For V_{12} :

$$4V_{12} = V_{22} + 10 + V_{13} + V_{11}$$

$$4V_{12} - V_{22} - V_{13} - V_{11} = 10$$

For V_{22} :

$$4V_{22} = V_{32} + V_{12} + V_{23} + V_{21}$$

$$4V_{22} - V_{32} - V_{12} - V_{23} - V_{21} = 0$$

For V_{32} :

$$4V_{32} = 0 + V_{22} + V_{33} + V_{31}$$

$$4V_{32} - V_{22} - V_{33} - V_{31} = 0$$

Top row ($j = 3$):

For V_{13} :

$$V_{13} = \frac{1}{4}(V_{23} + 10 + V_{14} + V_{12})$$

Since $V_{14} = V_{13}$:

$$3V_{13} - V_{23} - V_{12} = 10$$

For V_{23} :

$$3V_{23} - V_{33} - V_{13} - V_{22} = 0$$

For V_{33} :

$$3V_{33} - V_{23} - V_{32} = 0$$

Symmetry argument Because the top and bottom are both insulated, the solution is independent of y . Hence:

$$V_{11} = V_{12} = V_{13}, \quad V_{21} = V_{22} = V_{23}, \quad V_{31} = V_{32} = V_{33}$$

Set:

$$A = V_{11} = V_{12} = V_{13}, \quad B = V_{21} = V_{22} = V_{23}, \quad C = V_{31} = V_{32} = V_{33}$$

Then the system reduces to:

$$\begin{cases} 2A - B = 10 \\ -A + 2B - C = 0 \\ -B + 2C = 0 \end{cases}$$

This is identical to the 1D case, so:

$$A = 7.5, \quad B = 5, \quad C = 2.5$$

Final nodal values:

$$V_{11} = V_{12} = V_{13} = 7.5$$

$$V_{21} = V_{22} = V_{23} = 5$$

$$V_{31} = V_{32} = V_{33} = 2.5$$

Exercise 5: Parabolic PDE (Time-Dependent)

Consider:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

1. Derivation of the explicit finite difference scheme Let:

$$x_i = ih, \quad t^n = n\Delta t$$

and denote:

$$u(x_i, t^n) \approx u_i^n$$

Approximate the time derivative by a forward difference:

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Approximate the second space derivative by a central difference:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}$$

Substitute into the PDE:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}$$

Multiply by Δt :

$$u_i^{n+1} - u_i^n = \frac{\Delta t}{h^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

Define:

$$r = \frac{\Delta t}{h^2}$$

Then:

$$u_i^{n+1} = u_i^n + r (u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

or:

$$u_i^{n+1} = (1 - 2r)u_i^n + ru_{i-1}^n + ru_{i+1}^n$$

2. Stability condition For the explicit scheme:

$$r = \frac{\Delta t}{h^2} \leq \frac{1}{2}$$

Thus:

$$\Delta t \leq \frac{h^2}{2}$$

3. One time-step example Take the interval $[0, 1]$ with 4 equal segments:

$$h = 0.25$$

Choose:

$$\Delta t = 0.02$$

Then:

$$r = \frac{0.02}{(0.25)^2} = \frac{0.02}{0.0625} = 0.32$$

Since $0.32 < 0.5$, the scheme is stable.

Suppose the initial values are:

$$u_0^0 = 0, \quad u_1^0 = 1, \quad u_2^0 = 2, \quad u_3^0 = 1, \quad u_4^0 = 0$$

Compute the next time step.

For $i = 1$:

$$u_1^1 = u_1^0 + r(u_2^0 - 2u_1^0 + u_0^0)$$

$$u_1^1 = 1 + 0.32(2 - 2 \cdot 1 + 0)$$

$$u_1^1 = 1 + 0.32(0) = 1$$

For $i = 2$:

$$u_2^1 = u_2^0 + r(u_3^0 - 2u_2^0 + u_1^0)$$

$$u_2^1 = 2 + 0.32(1 - 4 + 1)$$

$$u_2^1 = 2 + 0.32(-2) = 2 - 0.64 = 1.36$$

For $i = 3$:

$$u_3^1 = u_3^0 + r(u_4^0 - 2u_3^0 + u_2^0)$$

$$u_3^1 = 1 + 0.32(0 - 2 + 2) = 1$$

Result after one time step:

$$u_0^1 = 0, \quad u_1^1 = 1, \quad u_2^1 = 1.36, \quad u_3^1 = 1, \quad u_4^1 = 0$$

Interpretation:

The peak value decreases with time, illustrating the diffusion effect.

Exercise 6: 1D Finite Element Method

Consider:

$$-\frac{d^2 u}{dx^2} = 0, \quad 0 \leq x \leq 1$$

with:

$$u(0) = 10, \quad u(1) = 0$$

1. Discretization into 2 elements Nodes:

$$x_1 = 0, \quad x_2 = 0.5, \quad x_3 = 1$$

Unknown nodal values:

$$u_1 = 10, \quad u_2 = ?, \quad u_3 = 0$$

Each element has length:

$$h = 0.5$$

2. Element matrices For the 1D equation $-u'' = 0$, the element stiffness matrix is:

$$[K^{(e)}] = \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Since $h = 0.5$:

$$[K^{(e)}] = \frac{1}{0.5} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 2 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

Thus:

$$[K^{(1)}] = [K^{(2)}] = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

3. Assembly of the global matrix Element 1 connects nodes 1 and 2.
Element 2 connects nodes 2 and 3.

After assembly:

$$[K] = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 2 \end{bmatrix}$$

The global system is:

$$[K]\{u\} = \{0\}$$

So:

$$\begin{bmatrix} 2 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

4. Apply boundary conditions and solve Known:

$$u_1 = 10, \quad u_3 = 0$$

Use the second row:

$$-2u_1 + 4u_2 - 2u_3 = 0$$

Substitute:

$$-2(10) + 4u_2 - 2(0) = 0$$

$$-20 + 4u_2 = 0$$

$$4u_2 = 20$$

$$u_2 = 5$$

Final solution:

$$u_1 = 10, \quad u_2 = 5, \quad u_3 = 0$$

Exercise 7: 2D FEM (Triangular Element)

Triangle nodes:

$$(0, 0), \quad (1, 0), \quad (0, 1)$$

Let:

$$(x_1, y_1) = (0, 0), \quad (x_2, y_2) = (1, 0), \quad (x_3, y_3) = (0, 1)$$

1. Area of the triangle Use:

$$A = \frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$$

Substitute:

$$A = \frac{1}{2} |0(0 - 1) + 1(1 - 0) + 0(0 - 0)|$$

$$A = \frac{1}{2} |1| = \frac{1}{2}$$

Area:

$$A = \frac{1}{2}$$

2. Compute b_i and c_i For a linear triangular element:

$$b_1 = y_2 - y_3, \quad b_2 = y_3 - y_1, \quad b_3 = y_1 - y_2$$

$$c_1 = x_3 - x_2, \quad c_2 = x_1 - x_3, \quad c_3 = x_2 - x_1$$

Compute:

$$b_1 = 0 - 1 = -1, \quad b_2 = 1 - 0 = 1, \quad b_3 = 0 - 0 = 0$$

$$c_1 = 0 - 1 = -1, \quad c_2 = 0 - 0 = 0, \quad c_3 = 1 - 0 = 1$$

So:

$$b_1 = -1, \quad b_2 = 1, \quad b_3 = 0$$

$$c_1 = -1, \quad c_2 = 0, \quad c_3 = 1$$

3. Element stiffness matrix For a Laplace-type problem, the element matrix is:

$$[K^{(e)}] = \frac{1}{4A} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix}$$

Since:

$$A = \frac{1}{2} \Rightarrow 4A = 2$$

Now compute each term.

Diagonal terms:

$$b_1^2 + c_1^2 = (-1)^2 + (-1)^2 = 2$$

$$b_2^2 + c_2^2 = (1)^2 + 0^2 = 1$$

$$b_3^2 + c_3^2 = 0^2 + 1^2 = 1$$

Off-diagonal terms:

$$b_1 b_2 + c_1 c_2 = (-1)(1) + (-1)(0) = -1$$

$$b_1 b_3 + c_1 c_3 = (-1)(0) + (-1)(1) = -1$$

$$b_2 b_3 + c_2 c_3 = (1)(0) + (0)(1) = 0$$

Therefore:

$$[K^{(e)}] = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

So:

$$[K^{(e)}] = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

Exercise 8: Electrical Engineering Application

1. Governing PDE For a conductive plate in steady state with no internal sources, the governing equation is Laplace's equation:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

2. Type of PDE This is a:

- second-order PDE,
- linear PDE,
- elliptic PDE.

3. Suitable numerical method Both FDM and FEM can be used.

If the plate is rectangular and the grid is regular: FDM is simple and efficient.

If the geometry becomes irregular or more complex: FEM is preferred because it handles irregular domains and complex boundaries more easily.

Justification here:

Since the plate shown is rectangular with simple boundary conditions, **FDM is a very suitable choice**. It is easy to implement and gives accurate results on structured grids.

Exercise 9: Comparison FEM vs FDM

1. Comparison Geometry handling

- **FDM:** Best suited for regular rectangular domains.
- **FEM:** Very flexible for complex and irregular geometries.

Accuracy

- **FDM:** Accurate on structured meshes and simple geometries.
- **FEM:** Often more accurate for complex geometries and heterogeneous materials.

Computational complexity

- **FDM:** Simpler implementation and usually lower computational cost for simple domains.
- **FEM:** More complex implementation because of mesh generation, element matrices, and assembly.

2. Example where FEM is preferred FEM is preferred in:

- magnetic field analysis of electrical machines,
- transformer core modeling,
- electrostatic field computation in irregular geometries.

A typical example is the **electromagnetic analysis of a rotating electrical machine**, where the geometry is complex and includes air gaps, slots, coils, and ferromagnetic materials.

Chapter 3

Optimization Techniques

3.1 Definition and Formulation of an Optimization Problem

Optimization is a fundamental tool in engineering and applied sciences. Its purpose is to determine the **best solution** among a set of admissible solutions, according to a given performance criterion.

In electrical engineering, power systems, and electromechanics, optimization appears in many practical situations, such as:

- minimizing power losses in an electrical network,
- minimizing the cost of energy production,
- maximizing the efficiency of an electrical machine,
- determining the optimal dimensions of a device,
- tuning controller parameters,
- allocating resources under technical and operational constraints.

In general, an optimization problem consists of finding the values of some variables that minimize or maximize an objective function while satisfying a set of constraints.

3.1.1 Basic Definition

Let $x = (x_1, x_2, \dots, x_n)^T$ be a vector of decision variables. An optimization problem can be written in the general form:

$$\begin{aligned} &\text{Minimize (or maximize)} && f(x) \\ &\text{subject to} && g_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & && h_j(x) = 0, \quad j = 1, 2, \dots, p \\ & && x \in \Omega \end{aligned}$$

where:

- $f(x)$ is the **objective function**,

- $g_i(x)$ are the **inequality constraints**,
- $h_j(x)$ are the **equality constraints**,
- Ω is the **search domain** or feasible set.

The objective of the optimization process is to find a point x^* such that:

$$f(x^*) \leq f(x), \quad \forall x \in \Omega_f$$

for a minimization problem, where Ω_f denotes the feasible set.

3.1.2 Main Components of an Optimization Problem

Any optimization problem contains the following elements.

Decision Variables

The **decision variables** are the unknown quantities to be determined.

Examples:

- generator powers in an electrical network,
- dimensions of a conductor,
- resistance, inductance, or capacitance values in a circuit,
- speed or torque of an electromechanical system.

If there are n variables, the optimization problem is said to be of dimension n .

Objective Function

The **objective function** measures the performance of the system. It is the quantity to be minimized or maximized.

Typical examples include:

- minimization of power losses,
- minimization of cost,
- minimization of energy consumption,
- maximization of efficiency,
- maximization of output power.

Mathematically, the objective function is written as:

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

Constraints

The constraints represent the technical, physical, or economic limitations of the system.

They may be of different types:

1. Equality constraints

$$h_j(x) = 0$$

These constraints impose exact relationships between variables.

2. Inequality constraints

$$g_i(x) \leq 0$$

These constraints define limits that must not be exceeded.

3. Bound constraints

$$x_k^{\min} \leq x_k \leq x_k^{\max}$$

These are simple lower and upper bounds on the variables.

Feasible Solution

A solution is called **feasible** if it satisfies all the constraints.

The set of all feasible solutions is called the **feasible region** or **admissible domain**.

Optimal Solution

A feasible solution x^* is called an **optimal solution** if it gives the best value of the objective function among all feasible solutions.

3.1.3 Minimization and Maximization

An optimization problem may be formulated either as a minimization or a maximization problem.

For example:

- **Minimization:** cost, losses, error, weight, or energy consumption.
- **Maximization:** efficiency, power output, profit, or reliability.

A maximization problem can always be transformed into a minimization problem by changing the sign of the objective function:

$$\max f(x) \iff \min[-f(x)]$$

For this reason, many numerical optimization methods are presented only for minimization problems.

3.1.4 Unconstrained and Constrained Optimization

Optimization problems are usually divided into two major categories.

Unconstrained Optimization

An unconstrained optimization problem involves only the objective function:

$$\min_{x \in \mathbb{R}^n} f(x)$$

There are no explicit constraints on the variables.

Example:

$$\min f(x) = x^2 + 4x + 5$$

Constrained Optimization

A constrained optimization problem includes one or more constraints:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \\ & h_j(x) = 0 \end{aligned}$$

Example:

$$\min f(x, y) = x^2 + y^2$$

subject to:

$$x + y = 1$$

In engineering practice, most optimization problems are constrained because real systems always operate under physical and technical limitations.

3.1.5 Single-Variable and Multivariable Optimization

Single-Variable Optimization

If the objective function depends on only one variable, the problem is called a **single-variable optimization problem**:

$$\min f(x)$$

Example:

$$f(x) = x^2 - 6x + 10$$

Multivariable Optimization

If the objective function depends on several variables, the problem is called a **multivariable optimization problem**:

$$\min f(x_1, x_2, \dots, x_n)$$

Example:

$$f(x, y) = x^2 + y^2 - 4x - 2y + 5$$

Many engineering optimization problems are multivariable because a real system often depends on several design parameters.

3.1.6 Local and Global Optimum

It is important to distinguish between **local optimum** and **global optimum**.

Local Minimum

A point x^* is a local minimum if there exists a neighborhood of x^* such that:

$$f(x^*) \leq f(x)$$

for all feasible points x in that neighborhood.

Global Minimum

A point x^* is a global minimum if:

$$f(x^*) \leq f(x), \quad \forall x \in \Omega_f$$

Thus, the global minimum is the best solution over the entire feasible domain.

Remark: For nonconvex problems, several local minima may exist, and finding the global minimum may be difficult.

3.1.7 Standard Mathematical Form of an Optimization Problem

A standard optimization problem is often written in compact form as:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

This formulation will be used throughout the rest of this chapter.

3.1.8 Engineering Example: Economic Dispatch in a Power System

One classical optimization problem in power systems is the **economic dispatch problem**. The objective is to determine the generated powers $P_1, P_2, \dots, P_n$ that minimize the total generation cost.

$$\min F(P) = \sum_{i=1}^n F_i(P_i)$$

subject to:

$$\sum_{i=1}^n P_i = P_D$$

and:

$$P_i^{\min} \leq P_i \leq P_i^{\max}, \quad i = 1, \dots, n$$

where:

- $F_i(P_i)$ is the cost function of generator i ,

- P_D is the total load demand,
- $P_i^{\min}$ and $P_i^{\max}$ are the generation limits.

In this case:

- the **decision variables** are the generated powers P_i ,
- the **objective function** is the total cost,
- the **equality constraint** is the power balance,
- the **inequality constraints** are the generation limits.

This example illustrates how optimization is directly connected to practical engineering problems.

3.1.9 Graphical Interpretation

For a function of one variable, the optimum corresponds to the lowest or highest point of the curve.

For a function of two variables, the optimum can be interpreted geometrically using contour lines or surfaces.

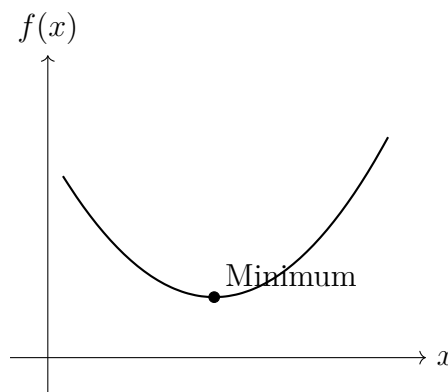


Figure 3.1: Graphical interpretation of a minimum point for a single-variable function.

3.1.10 Summary

An optimization problem consists of finding the best values of decision variables in order to minimize or maximize an objective function, while satisfying a set of constraints.

The key elements are:

- decision variables,
- objective function,
- equality and inequality constraints,
- feasible set,
- optimal solution.

Before studying numerical optimization algorithms, it is essential to understand how to formulate an engineering problem in mathematical form.

3.2 Single and Multiple Optimization with or without Constraints

Optimization problems can be classified according to several criteria. One important classification is based on the number of objective functions, while another is based on the presence or absence of constraints.

In this section, we distinguish between:

- single-objective optimization,
- multi-objective optimization,
- unconstrained optimization,
- constrained optimization.

These categories are fundamental because they determine the mathematical formulation of the problem and the choice of the numerical solution method.

3.2.1 Single-Objective Optimization

A **single-objective optimization problem** involves only one objective function to be minimized or maximized.

Its general form is:

$$\min_{x \in \Omega} f(x)$$

or

$$\max_{x \in \Omega} f(x)$$

where $f(x)$ is the only performance criterion.

Example 1: Minimization of Joule Losses

In an electrical conductor, the Joule losses may be written as:

$$P_{\text{loss}} = RI^2$$

If the current or resistance can be adjusted through design parameters, one may seek to minimize:

$$f(x) = P_{\text{loss}}(x)$$

In this case, the problem has only one objective: minimizing power losses.

Example 2: Maximization of Efficiency

For an electrical machine, the efficiency is defined by:

$$\eta = \frac{P_{\text{out}}}{P_{\text{in}}}$$

The optimization problem may be formulated as:

$$\max \eta(x)$$

where x represents design or operating variables.

Remarks

Single-objective optimization is the simplest and most classical framework. Many standard numerical methods such as gradient descent, Newton's method, and deterministic search methods are developed for this type of problem.

3.2.2 Multi-Objective Optimization

In many engineering problems, several objectives must be optimized simultaneously. These objectives are often conflicting.

A **multi-objective optimization problem** can be written as:

$$\min (f_1(x), f_2(x), \dots, f_r(x))$$

where:

- $f_1(x), f_2(x), \dots, f_r(x)$ are several objective functions,
- the decision vector is $x \in \Omega$.

Engineering Motivation

In practice, one may want to:

- minimize cost,
- minimize weight,
- minimize energy losses,
- maximize efficiency,
- maximize reliability.

These objectives may not be compatible with each other. For example:

- increasing efficiency may increase the cost,
- reducing material quantity may reduce mechanical strength,
- minimizing losses may require larger conductor sections and therefore a higher price.

Example: Electrical Machine Design

Suppose that in the design of an electric motor, one wants to:

$$\min f_1(x) = \text{cost}$$

and simultaneously:

$$\max f_2(x) = \text{efficiency}$$

Since maximization can be transformed into minimization, this can also be written as:

$$\min f'_2(x) = -\eta(x)$$

Thus, the problem becomes:

$$\min (f_1(x), -\eta(x))$$

Pareto Optimality

In multi-objective optimization, there is generally no single solution that simultaneously minimizes all objectives. Instead, one seeks **Pareto-optimal solutions**.

A feasible solution x^* is called **Pareto-optimal** if there exists no other feasible solution that improves one objective without worsening at least one other objective.

The set of all Pareto-optimal solutions is called the **Pareto front**.

Weighted Sum Method

A common way to transform a multi-objective problem into a single-objective one is the weighted sum approach:

$$\min F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \cdots + \alpha_r f_r(x)$$

where the weights satisfy:

$$\alpha_i \geq 0, \quad \sum_{i=1}^r \alpha_i = 1$$

This method allows different priorities to be assigned to the objectives.

3.2.3 Unconstrained Optimization

An **unconstrained optimization problem** is a problem in which the variables are free and no explicit equality or inequality constraints are imposed.

Its general form is:

$$\min_{x \in \mathbb{R}^n} f(x)$$

Only the objective function is involved.

Example 1: Single-Variable Unconstrained Optimization

Consider the function:

$$f(x) = x^2 - 4x + 7$$

To determine its minimum, compute the derivative:

$$f'(x) = 2x - 4$$

Set:

$$f'(x) = 0$$

Thus:

$$2x - 4 = 0 \quad \Rightarrow \quad x = 2$$

Now compute the second derivative:

$$f''(x) = 2 > 0$$

Therefore, $x = 2$ is a minimum point.

The minimum value is:

$$f(2) = 2^2 - 4(2) + 7 = 4 - 8 + 7 = 3$$

Hence:

$$x^* = 2, \quad f(x^*) = 3$$

Example 2: Two-Variable Unconstrained Optimization

Consider:

$$f(x, y) = x^2 + y^2 - 2x - 4y + 10$$

Compute the first partial derivatives:

$$\frac{\partial f}{\partial x} = 2x - 2$$

$$\frac{\partial f}{\partial y} = 2y - 4$$

Set them equal to zero:

$$2x - 2 = 0 \quad \Rightarrow \quad x = 1$$

$$2y - 4 = 0 \quad \Rightarrow \quad y = 2$$

So the critical point is:

$$(x^*, y^*) = (1, 2)$$

Now compute the Hessian matrix:

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Since H is positive definite, the point $(1, 2)$ is a minimum.
The minimum value is:

$$f(1, 2) = 1 + 4 - 2 - 8 + 10 = 5$$

Thus:

$$(x^*, y^*) = (1, 2), \quad f(x^*, y^*) = 5$$

Characteristics of Unconstrained Optimization

Unconstrained optimization is mathematically simpler because there is no need to verify feasibility conditions related to constraints.

However, in practical engineering problems, unconstrained formulations are often only approximations, since real systems usually have physical and operational limitations.

3.2.4 Constrained Optimization

A **constrained optimization problem** is a problem in which the variables must satisfy one or more equality or inequality constraints.

Its general form is:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_j(x) = 0, \quad j = 1, \dots, p \end{aligned}$$

Example 1: Equality Constraint

Consider:

$$\min f(x, y) = x^2 + y^2$$

subject to:

$$x + y = 1$$

We use the constraint to eliminate one variable:

$$y = 1 - x$$

Substitute into the objective function:

$$f(x) = x^2 + (1 - x)^2$$

Expand:

$$\begin{aligned} f(x) &= x^2 + 1 - 2x + x^2 \\ f(x) &= 2x^2 - 2x + 1 \end{aligned}$$

Differentiate:

$$f'(x) = 4x - 2$$

Set:

$$4x - 2 = 0$$

Hence:

$$x = \frac{1}{2}$$

Then:

$$y = 1 - \frac{1}{2} = \frac{1}{2}$$

The minimum value is:

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

Therefore:

$$x^* = \frac{1}{2}, \quad y^* = \frac{1}{2}, \quad f_{\min} = \frac{1}{2}$$

Example 2: Inequality Constraint

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

The unconstrained minimum of x^2 is at:

$$x = 0$$

but this value does not satisfy the constraint $x \geq 1$.

Therefore, the feasible set is:

$$[1, +\infty[$$

The minimum feasible value is obtained at the boundary:

$$x^* = 1$$

and:

$$f(1) = 1$$

Thus:

$$x^* = 1, \quad f_{\min} = 1$$

Bound Constraints

A very common type of constrained optimization is the bound-constrained problem:

$$x_i^{\min} \leq x_i \leq x_i^{\max}$$

Such constraints are frequent in electrical engineering. For example:

- current limits,
- voltage limits,
- power generation limits,
- speed limits,
- temperature limits.

3.2.5 Comparison Between the Different Categories

Optimization problems can therefore be classified as follows:

1. **According to the number of objectives:**

- single-objective optimization,
- multi-objective optimization.

2. **According to the presence of constraints:**

- unconstrained optimization,
- constrained optimization.

This classification is important because the numerical methods used to solve these problems are not the same in all cases.

3.2.6 Engineering Example: Optimal Power Dispatch

Consider a simple power system with two generators supplying a total demand P_D .

Let:

$$P_1, P_2$$

be the generated powers.

Suppose the objective is to minimize the total production cost:

$$\min f(P_1, P_2) = C_1(P_1) + C_2(P_2)$$

subject to the equality constraint:

$$P_1 + P_2 = P_D$$

and the inequality constraints:

$$\begin{aligned} P_1^{\min} &\leq P_1 \leq P_1^{\max} \\ P_2^{\min} &\leq P_2 \leq P_2^{\max} \end{aligned}$$

This is a:

- multivariable optimization problem,
- single-objective optimization problem,
- constrained optimization problem.

If one also wants to minimize pollutant emissions, then a second objective function can be added:

$$\min f_2(P_1, P_2) = E_1(P_1) + E_2(P_2)$$

The problem then becomes a **multi-objective constrained optimization problem**.

3.2.7 Summary

Optimization problems may involve one objective or several objectives, and they may be formulated with or without constraints.

The main categories are:

- **single-objective optimization:** one function to optimize,
- **multi-objective optimization:** several functions to optimize simultaneously,
- **unconstrained optimization:** no explicit constraints,
- **constrained optimization:** equality and/or inequality constraints are imposed.

This classification is essential before introducing numerical optimization algorithms.

3.3 Unconstrained Optimization Algorithms

Unconstrained optimization deals with problems of the form:

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f(x)$ is the objective function and no explicit constraints are imposed on the variables.

The goal is to determine a point x^* such that:

$$f(x^*) \leq f(x), \quad \forall x \in \mathbb{R}^n$$

These problems are fundamental because many constrained optimization methods are built upon unconstrained techniques.

Unconstrained optimization algorithms can be divided into two main categories:

- **Deterministic methods**
- **Stochastic methods**

3.3.1 Deterministic Methods

Deterministic methods use analytical information about the objective function, such as derivatives, to guide the search toward the optimum.

They follow a predefined sequence of operations and always produce the same result for a given initial point.

General Iterative Scheme

Most deterministic optimization methods are iterative and follow the form:

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$$

where:

- $x^{(k)}$ is the current solution,
- $d^{(k)}$ is the search direction,
- α_k is the step size (or learning rate).

The choice of $d^{(k)}$ and α_k determines the specific optimization method.

Gradient Descent Method

The **gradient descent method** is one of the most important techniques for unconstrained optimization.

Principle The gradient of $f(x)$, denoted $\nabla f(x)$, points in the direction of the greatest increase of the function. Therefore, the direction of steepest descent is:

$$d^{(k)} = -\nabla f(x^{(k)})$$

The iterative formula becomes:

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$$

Example Consider:

$$f(x) = x^2$$

Then:

$$\nabla f(x) = 2x$$

Choose $\alpha = 0.25$ and initial point $x^{(0)} = 4$.

Iteration 1:

$$x^{(1)} = 4 - 0.25 \times 2(4) = 4 - 2 = 2$$

Iteration 2:

$$x^{(2)} = 2 - 0.25 \times 2(2) = 2 - 1 = 1$$

Iteration 3:

$$x^{(3)} = 1 - 0.25 \times 2(1) = 1 - 0.5 = 0.5$$

The sequence converges toward the minimum at $x = 0$.

Remarks

- Simple to implement,
- Requires computation of the gradient,
- Convergence depends on the choice of α_k .

Newton's Method

Newton's method uses second-order information (Hessian matrix) to improve convergence speed.

Iteration formula

$$x^{(k+1)} = x^{(k)} - H^{-1}(x^{(k)})\nabla f(x^{(k)})$$

where:

- $\nabla f(x)$ is the gradient,
- $H(x)$ is the Hessian matrix.

Example (1D) Consider:

$$f(x) = x^2$$

Then:

$$f'(x) = 2x, \quad f''(x) = 2$$

Newton iteration:

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

Starting from $x^{(0)} = 4$:

$$x^{(1)} = 4 - \frac{8}{2} = 0$$

The solution is obtained in one iteration.

Remarks

- Very fast convergence (quadratic),
- Requires computation of second derivatives,
- Can be expensive for large-scale problems.

Comparison of Deterministic Methods

Method	Information used	Convergence speed
Gradient descent	First derivative	Moderate
Newton method	First + second derivatives	Fast (quadratic)

Table 3.1: Comparison of deterministic optimization methods.

3.3.2 Stochastic Methods

Stochastic methods use randomness to explore the search space. They are particularly useful when:

- the objective function is complex or non-differentiable,
- the problem has many local minima,
- derivative information is not available.

Basic Principle

Instead of following a deterministic direction, stochastic methods generate random candidate solutions and evaluate the objective function.

The algorithm then decides whether to accept or reject these candidates.

Random Search Method

One of the simplest stochastic methods is the random search:

1. Generate a random point x in the domain,
2. Evaluate $f(x)$,
3. Keep the best solution found so far,
4. Repeat the process.

Remarks

- Very simple to implement,
- Does not require derivatives,
- May be slow to converge.

Stochastic vs Deterministic Methods

Criterion	Deterministic	Stochastic
Use of derivatives	Yes	No
Reproducibility	Same result	Different results
Convergence speed	Fast (local)	Slower
Global search	Limited	Good

Table 3.2: Comparison between deterministic and stochastic methods.

3.3.3 Engineering Application

In electrical engineering, unconstrained optimization methods are used in:

- parameter identification,
- controller tuning,
- loss minimization,
- machine design.

For example, minimizing copper losses in a conductor may be written as:

$$\min f(x) = R(x)I^2$$

where x represents design variables such as cross-sectional area.

3.3.4 Summary

Unconstrained optimization algorithms aim to find the minimum (or maximum) of a function without explicit constraints.

They are divided into:

- **deterministic methods** (gradient-based, Newton),
- **stochastic methods** (random search).

These methods form the basis for more advanced optimization techniques, especially for constrained problems.

3.4 Constraint Handling Methods

In many engineering problems, optimization must be performed under constraints. These constraints may represent physical limits, safety requirements, or operational conditions.

To solve such problems, specific techniques are used to handle constraints. These techniques can be divided into two main categories:

- **Transformation methods**
- **Direct methods**

3.4.1 Transformation Methods

Transformation methods convert a constrained optimization problem into an equivalent **unconstrained problem**. This allows the use of standard unconstrained optimization algorithms such as gradient descent or Newton's method.

Penalty Methods

The idea of penalty methods is to penalize violations of constraints by adding a penalty term to the objective function.

General Principle Consider the problem:

$$\min f(x)$$

subject to:

$$g_i(x) \leq 0, \quad h_j(x) = 0$$

We define a new objective function:

$$F(x) = f(x) + \rho P(x)$$

where:

- $\rho > 0$ is a penalty parameter,
- $P(x)$ is a penalty function measuring constraint violations.

A common choice is:

$$P(x) = \sum_i \max(0, g_i(x))^2 + \sum_j h_j(x)^2$$

Thus, if the constraints are violated, the penalty term increases the value of $F(x)$.

Example Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

The constraint can be written as:

$$g(x) = 1 - x \leq 0$$

Define the penalized function:

$$F(x) = x^2 + \rho \max(0, 1 - x)^2$$

Case 1: $x \geq 1$

Then $\max(0, 1 - x) = 0$, so:

$$F(x) = x^2$$

Case 2: $x < 1$

Then:

$$F(x) = x^2 + \rho(1 - x)^2$$

As ρ increases, values of $x < 1$ become heavily penalized, forcing the solution toward $x \geq 1$.

Remarks

- Simple to implement,
- Allows reuse of unconstrained algorithms,
- Choice of ρ is critical,
- Large ρ may lead to numerical instability.

Barrier Methods

Barrier methods prevent the solution from leaving the feasible region by adding a term that becomes infinite near the boundary.

Principle For inequality constraints $g(x) \leq 0$, a barrier function may be defined as:

$$F(x) = f(x) - \mu \sum_i \ln(-g_i(x))$$

where $\mu > 0$ is a parameter.

Interpretation

- When $g(x) \rightarrow 0^-$, $\ln(-g(x)) \rightarrow -\infty$,
- Thus $F(x) \rightarrow +\infty$,
- The solution is forced to remain inside the feasible domain.

Remarks

- Requires an initial feasible point,
- Efficient for strictly feasible problems,
- Used in interior-point methods.

3.4.2 Direct Methods

Direct methods handle constraints explicitly, without transforming the problem.

Method of Lagrange Multipliers

The method of Lagrange multipliers is used to solve optimization problems with equality constraints.

Problem formulation

$$\min f(x)$$

subject to:

$$h(x) = 0$$

Lagrangian function Define:

$$\mathcal{L}(x, \lambda) = f(x) + \lambda h(x)$$

where λ is the Lagrange multiplier.

Optimality conditions The optimal solution satisfies:

$$\begin{aligned}\nabla_x \mathcal{L}(x, \lambda) &= 0 \\ h(x) &= 0\end{aligned}$$

Example

Consider:

$$\min f(x, y) = x^2 + y^2$$

subject to:

$$x + y = 1$$

Step 1: Define the Lagrangian

$$\mathcal{L}(x, y, \lambda) = x^2 + y^2 + \lambda(x + y - 1)$$

Step 2: Compute partial derivatives

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x} &= 2x + \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial y} &= 2y + \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= x + y - 1 = 0\end{aligned}$$

Step 3: Solve the system From the first two equations:

$$\begin{aligned}2x + \lambda &= 0, \quad 2y + \lambda = 0 \\ \Rightarrow x &= y\end{aligned}$$

Substitute into the constraint:

$$x + x = 1 \Rightarrow x = \frac{1}{2}$$

Thus:

$$y = \frac{1}{2}$$

Step 4: Compute the minimum value

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2}$$

Final solution:

$$x^* = \frac{1}{2}, \quad y^* = \frac{1}{2}$$

Karush-Kuhn-Tucker (KKT) Conditions

For problems with inequality constraints, the optimality conditions are generalized by the **KKT conditions**.

Consider:

$$\min f(x)$$

subject to:

$$g_i(x) \leq 0$$

The KKT conditions are:

- Stationarity:

$$\nabla f(x^*) + \sum_i \lambda_i \nabla g_i(x^*) = 0$$

- Primal feasibility:

$$g_i(x^*) \leq 0$$

- Dual feasibility:

$$\lambda_i \geq 0$$

- Complementary slackness:

$$\lambda_i g_i(x^*) = 0$$

Interpretation

- If a constraint is inactive, its multiplier is zero,
- If a constraint is active, it influences the solution.

3.4.3 Engineering Application

In power systems, constraint handling is essential.

Example: Economic Dispatch

$$\min \sum_i C_i(P_i)$$

subject to:

$$\sum_i P_i = P_D$$

$$P_i^{\min} \leq P_i \leq P_i^{\max}$$

- The equality constraint ensures power balance,
- The inequality constraints ensure generator limits,
- Lagrange multipliers are used to solve this problem efficiently.

3.4.4 Comparison of Methods

Method	Principle	Remarks
Penalty methods	Add a penalty term	Simple, but sensitive to the penalty parameter
Barrier methods	Restrict the search to the interior of the feasible region	Requires a feasible initial point
Lagrange multipliers	Handle equality constraints exactly	Efficient for equality-constrained problems
KKT conditions	Provide general optimality conditions	Used for constrained optimization problems

Table 3.3: Comparison of constraint handling methods.

3.4.5 Summary

Constraint handling methods allow solving optimization problems with equality and inequality constraints.

Two main approaches are used:

- **Transformation methods:** convert the problem into an unconstrained one (penalty, barrier),
- **Direct methods:** handle constraints explicitly (Lagrange multipliers, KKT conditions).

These techniques are essential in engineering applications where constraints are unavoidable.

Tutorial Sheet – Chapter 3: Optimization Techniques

Exercise 1: Problem Formulation

Consider the function:

$$f(x, y) = x^2 + y^2 - 4x - 2y + 5$$

1. Identify the decision variables.
2. Determine whether the problem is:
 - single-objective or multi-objective,
 - constrained or unconstrained.
3. Rewrite the problem in standard optimization form.

Exercise 2: Unconstrained Optimization (Single Variable)

Consider:

$$f(x) = x^2 - 6x + 10$$

1. Compute the first derivative.
2. Determine the critical point.
3. Verify whether it is a minimum or maximum.
4. Compute the optimal value.

Exercise 3: Gradient Descent Method

Consider:

$$f(x) = x^2$$

1. Compute the gradient.
2. Apply the gradient descent method:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)})$$

with $\alpha = 0.25$ and $x^{(0)} = 4$.

3. Compute the first three iterations.
4. Comment on convergence.

Exercise 4: Multivariable Optimization

Consider:

$$f(x, y) = x^2 + y^2 - 2x - 4y + 10$$

1. Compute the gradient $\nabla f(x, y)$.
2. Determine the critical point.
3. Compute the Hessian matrix.
4. Determine the nature of the critical point.

Exercise 5: Constrained Optimization (Equality)

Consider:

$$\min f(x, y) = x^2 + y^2$$

subject to:

$$x + y = 1$$

1. Form the Lagrangian.
2. Derive the optimality conditions.
3. Solve the system.
4. Compute the optimal value.

Exercise 6: Constrained Optimization (Inequality)

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 2$$

1. Determine the unconstrained minimum.
2. Check feasibility.
3. Determine the optimal solution.

Exercise 7: Penalty Method

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

1. Write the constraint in the form $g(x) \leq 0$.
2. Construct the penalized function:

$$F(x) = f(x) + \rho \max(0, g(x))^2$$

3. Study the behavior of $F(x)$ for $x < 1$ and $x \geq 1$.

Exercise 8: KKT Conditions

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

1. Write the problem in standard form.
2. Write the KKT conditions.
3. Determine the optimal solution.

Exercise 9: Multi-objective Optimization

Consider:

$$\min f_1(x) = x^2, \quad \min f_2(x) = (x - 2)^2$$

1. Explain why the objectives are conflicting.
2. Write the weighted sum formulation.
3. Determine the optimal solution for:

$$F(x) = \alpha x^2 + (1 - \alpha)(x - 2)^2$$

for $\alpha = 0.5$.

Exercise 10: Electrical Engineering Application (Economic Dispatch)

Consider two generators with cost functions:

$$C_1(P_1) = 0.1P_1^2 + 5P_1$$

$$C_2(P_2) = 0.2P_2^2 + 3P_2$$

Total demand:

$$P_1 + P_2 = 100$$

1. Formulate the optimization problem.
2. Write the Lagrangian.
3. Derive the optimality conditions.
4. Compute the optimal generation P_1^* and P_2^* .

Detailed Solutions – Tutorial Sheet Chapter 3

Exercise 1: Problem Formulation

Consider:

$$f(x, y) = x^2 + y^2 - 4x - 2y + 5$$

1. Decision variables The decision variables are:

$$x \quad \text{and} \quad y$$

So the decision vector can be written as:

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

2. Nature of the problem The problem involves only one objective function:

$$f(x, y)$$

Therefore, it is a **single-objective optimization problem**.

No equality or inequality constraints are given, so it is an **unconstrained optimization problem**.

3. Standard optimization form The problem can be written as:

$$\min_{(x,y) \in \mathbb{R}^2} f(x, y)$$

that is:

$$\min_{(x,y) \in \mathbb{R}^2} (x^2 + y^2 - 4x - 2y + 5)$$

Exercise 2: Unconstrained Optimization (Single Variable)

Consider:

$$f(x) = x^2 - 6x + 10$$

1. First derivative Differentiate the function:

$$f'(x) = 2x - 6$$

2. Critical point A critical point is obtained by solving:

$$f'(x) = 0$$

Thus:

$$2x - 6 = 0$$

$$2x = 6$$

$$x = 3$$

So the critical point is:

$$x^* = 3$$

3. Nature of the critical point Compute the second derivative:

$$f''(x) = 2$$

Since:

$$f''(x) = 2 > 0$$

the function is convex and the critical point is a **minimum**.

4. Optimal value Evaluate the function at $x = 3$:

$$f(3) = 3^2 - 6(3) + 10$$

$$f(3) = 9 - 18 + 10 = 1$$

Final result:

$$x^* = 3, \quad f(x^*) = 1$$

Exercise 3: Gradient Descent Method

Consider:

$$f(x) = x^2$$

The gradient descent iteration is:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)})$$

with:

$$\alpha = 0.25, \quad x^{(0)} = 4$$

1. Gradient Since the function depends on one variable:

$$\nabla f(x) = f'(x) = 2x$$

2. Iteration formula Substitute the gradient into the update rule:

$$x^{(k+1)} = x^{(k)} - 0.25(2x^{(k)})$$

$$x^{(k+1)} = x^{(k)} - 0.5x^{(k)}$$

$$x^{(k+1)} = 0.5x^{(k)}$$

3. First three iterations Starting from:

$$x^{(0)} = 4$$

Iteration 1:

$$x^{(1)} = 0.5 \times 4 = 2$$

Iteration 2:

$$x^{(2)} = 0.5 \times 2 = 1$$

Iteration 3:

$$x^{(3)} = 0.5 \times 1 = 0.5$$

We may continue:

Iteration 4:

$$x^{(4)} = 0.5 \times 0.5 = 0.25$$

Iteration 5:

$$x^{(5)} = 0.5 \times 0.25 = 0.125$$

4. Comment on convergence The sequence is:

$$4, 2, 1, 0.5, 0.25, \dots$$

This sequence converges to:

$$0$$

Since $f(x) = x^2$ has its minimum at:

$$x^* = 0$$

the gradient descent method converges toward the optimum.

Final result:

$$x^{(1)} = 2, \quad x^{(2)} = 1, \quad x^{(3)} = 0.5$$

Exercise 4: Multivariable Optimization

Consider:

$$f(x, y) = x^2 + y^2 - 2x - 4y + 10$$

1. Gradient Compute the partial derivatives:

$$\begin{aligned} \frac{\partial f}{\partial x} &= 2x - 2 \\ \frac{\partial f}{\partial y} &= 2y - 4 \end{aligned}$$

Thus:

$$\nabla f(x, y) = \begin{bmatrix} 2x - 2 \\ 2y - 4 \end{bmatrix}$$

2. Critical point A critical point is obtained by solving:

$$\nabla f(x, y) = 0$$

So:

$$\begin{aligned} 2x - 2 = 0 &\Rightarrow x = 1 \\ 2y - 4 = 0 &\Rightarrow y = 2 \end{aligned}$$

Hence the critical point is:

$$(x^*, y^*) = (1, 2)$$

3. Hessian matrix The Hessian matrix is:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Compute the second derivatives:

$$\frac{\partial^2 f}{\partial x^2} = 2, \quad \frac{\partial^2 f}{\partial y^2} = 2, \quad \frac{\partial^2 f}{\partial x \partial y} = 0$$

Therefore:

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

4. Nature of the critical point The Hessian matrix is diagonal with positive entries. Therefore it is **positive definite**.

Hence, $(1, 2)$ is a **strict minimum point**.

The minimum value is:

$$f(1, 2) = 1^2 + 2^2 - 2(1) - 4(2) + 10$$

$$f(1, 2) = 1 + 4 - 2 - 8 + 10 = 5$$

Final result:

$$(x^*, y^*) = (1, 2), \quad f(x^*, y^*) = 5$$

Exercise 5: Constrained Optimization (Equality)

Consider:

$$\min f(x, y) = x^2 + y^2$$

subject to:

$$x + y = 1$$

1. Lagrangian The constraint is:

$$h(x, y) = x + y - 1 = 0$$

The Lagrangian is:

$$\mathcal{L}(x, y, \lambda) = x^2 + y^2 + \lambda(x + y - 1)$$

2. Optimality conditions Compute the partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial x} = 2x + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial y} = 2y + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = x + y - 1 = 0$$

So the system is:

$$\begin{cases} 2x + \lambda = 0 \\ 2y + \lambda = 0 \\ x + y = 1 \end{cases}$$

3. Solve the system From the first equation:

$$\lambda = -2x$$

From the second equation:

$$\lambda = -2y$$

Thus:

$$-2x = -2y$$

$$x = y$$

Substitute into the constraint:

$$x + x = 1$$

$$2x = 1$$

$$x = \frac{1}{2}$$

Hence:

$$y = \frac{1}{2}$$

4. Optimal value

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2$$

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

Final result:

$$x^* = \frac{1}{2}, \quad y^* = \frac{1}{2}, \quad f_{\min} = \frac{1}{2}$$

Exercise 6: Constrained Optimization (Inequality)

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 2$$

1. Unconstrained minimum

Without the constraint, the function:

$$f(x) = x^2$$

has derivative:

$$f'(x) = 2x$$

Set:

$$f'(x) = 0 \Rightarrow x = 0$$

So the unconstrained minimum is:

$$x = 0, \quad f(0) = 0$$

2. Feasibility check

The constraint is:

$$x \geq 2$$

But:

$$0 \not\geq 2$$

So the unconstrained minimum is **not feasible**.

3. Optimal feasible solution The feasible set is:

$$[2, +\infty[$$

Since $f(x) = x^2$ is increasing for $x > 0$, the minimum on the feasible set is attained at the left boundary:

$$x^* = 2$$

Then:

$$f(2) = 2^2 = 4$$

Final result:

$$x^* = 2, \quad f_{\min} = 4$$

Exercise 7: Penalty Method

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

1. Constraint in the form $g(x) \leq 0$ The constraint:

$$x \geq 1$$

can be rewritten as:

$$1 - x \leq 0$$

Thus:

$$g(x) = 1 - x$$

2. Penalized function The penalty formulation is:

$$F(x) = f(x) + \rho \max(0, g(x))^2$$

Substitute $f(x) = x^2$ and $g(x) = 1 - x$:

$$F(x) = x^2 + \rho \max(0, 1 - x)^2$$

3. Behavior of $F(x)$ Case 1: $x \geq 1$

Then:

$$1 - x \leq 0$$

so:

$$\max(0, 1 - x) = 0$$

Therefore:

$$F(x) = x^2$$

Case 2: $x < 1$

Then:

$$1 - x > 0$$

so:

$$\max(0, 1 - x) = 1 - x$$

Thus:

$$F(x) = x^2 + \rho(1 - x)^2$$

Expand:

$$F(x) = x^2 + \rho(1 - 2x + x^2)$$

$$F(x) = (1 + \rho)x^2 - 2\rho x + \rho$$

So when the constraint is violated, the penalty term increases the objective function.

Conclusion When ρ becomes large, solutions with $x < 1$ become increasingly expensive. Therefore, the minimizer of $F(x)$ moves toward the feasible region $x \geq 1$.

Exercise 8: KKT Conditions

Consider:

$$\min f(x) = x^2$$

subject to:

$$x \geq 1$$

1. Standard form Write the inequality as:

$$g(x) = 1 - x \leq 0$$

2. KKT conditions The Lagrangian is:

$$\mathcal{L}(x, \lambda) = x^2 + \lambda(1 - x)$$

The KKT conditions are:

Stationarity:

$$\frac{d\mathcal{L}}{dx} = 2x - \lambda = 0$$

Primal feasibility:

$$1 - x \leq 0$$

Dual feasibility:

$$\lambda \geq 0$$

Complementary slackness:

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

3. Determine the optimal solution From stationarity:

$$\lambda = 2x$$

Now use complementary slackness:

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

Substitute $\lambda = 2x$:

$$2x(1 - x) = 0$$

So either:

$$x = 0 \quad \text{or} \quad x = 1$$

Check feasibility:

- For $x = 0$:

$$1 - 0 = 1 > 0$$

Not feasible.

- For $x = 1$:

$$1 - 1 = 0$$

Feasible.

Thus:

$$x^* = 1$$

Then:

$$\lambda = 2x = 2$$

Check dual feasibility:

$$\lambda = 2 \geq 0$$

valid.

The minimum value is:

$$f(1) = 1$$

Final result:

$$x^* = 1, \quad \lambda^* = 2, \quad f_{\min} = 1$$

Exercise 9: Multi-objective Optimization

Consider:

$$\min f_1(x) = x^2, \quad \min f_2(x) = (x - 2)^2$$

1. Why are the objectives conflicting? The function:

$$f_1(x) = x^2$$

is minimized at:

$$x = 0$$

The function:

$$f_2(x) = (x - 2)^2$$

is minimized at:

$$x = 2$$

Therefore, one objective is best at $x = 0$, while the other is best at $x = 2$. A single value of x cannot minimize both simultaneously.

Thus, the objectives are **conflicting**.

2. Weighted sum formulation A weighted sum formulation is:

$$F(x) = \alpha f_1(x) + (1 - \alpha) f_2(x)$$

So:

$$F(x) = \alpha x^2 + (1 - \alpha)(x - 2)^2$$

3. Case $\alpha = 0.5$ Substitute $\alpha = 0.5$:

$$F(x) = 0.5x^2 + 0.5(x - 2)^2$$

Expand:

$$F(x) = 0.5x^2 + 0.5(x^2 - 4x + 4)$$

$$F(x) = 0.5x^2 + 0.5x^2 - 2x + 2$$

$$F(x) = x^2 - 2x + 2$$

Differentiate:

$$F'(x) = 2x - 2$$

Set:

$$F'(x) = 0$$

$$2x - 2 = 0$$

$$x = 1$$

Second derivative:

$$F''(x) = 2 > 0$$

So $x = 1$ is a minimum.

The corresponding value is:

$$F(1) = 1^2 - 2(1) + 2 = 1$$

Final result:

$$x^* = 1$$

Exercise 10: Electrical Engineering Application (Economic Dispatch)

Consider:

$$C_1(P_1) = 0.1P_1^2 + 5P_1$$

$$C_2(P_2) = 0.2P_2^2 + 3P_2$$

with:

$$P_1 + P_2 = 100$$

1. Problem formulation We want to minimize the total cost:

$$\min C(P_1, P_2) = C_1(P_1) + C_2(P_2)$$

Thus:

$$\min C(P_1, P_2) = 0.1P_1^2 + 5P_1 + 0.2P_2^2 + 3P_2$$

subject to:

$$P_1 + P_2 = 100$$

2. Lagrangian The Lagrangian is:

$$\mathcal{L}(P_1, P_2, \lambda) = 0.1P_1^2 + 5P_1 + 0.2P_2^2 + 3P_2 + \lambda(P_1 + P_2 - 100)$$

3. Optimality conditions Compute the partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial P_1} = 0.2P_1 + 5 + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial P_2} = 0.4P_2 + 3 + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = P_1 + P_2 - 100 = 0$$

So the system is:

$$\begin{cases} 0.2P_1 + 5 + \lambda = 0 \\ 0.4P_2 + 3 + \lambda = 0 \\ P_1 + P_2 = 100 \end{cases}$$

4. Solve the system From the first equation:

$$\lambda = -0.2P_1 - 5$$

From the second equation:

$$\lambda = -0.4P_2 - 3$$

Equate the two expressions:

$$-0.2P_1 - 5 = -0.4P_2 - 3$$

Rearrange:

$$-0.2P_1 + 0.4P_2 = 2$$

Multiply by 10:

$$-2P_1 + 4P_2 = 20$$

Divide by 2:

$$-P_1 + 2P_2 = 10$$

So:

$$P_1 = 2P_2 - 10$$

Use the balance equation:

$$P_1 + P_2 = 100$$

Substitute:

$$(2P_2 - 10) + P_2 = 100$$

$$3P_2 - 10 = 100$$

$$3P_2 = 110$$

$$P_2 = \frac{110}{3} \approx 36.67$$

Then:

$$P_1 = 100 - P_2 = 100 - 36.67 = 63.33$$

So:

$$P_1^* \approx 63.33, \quad P_2^* \approx 36.67$$

Verification Check the balance:

$$63.33 + 36.67 = 100$$

Minimum total cost Compute:

$$C_1(63.33) = 0.1(63.33)^2 + 5(63.33)$$

$$(63.33)^2 \approx 4010.67$$

$$C_1(63.33) \approx 0.1(4010.67) + 316.65$$

$$C_1(63.33) \approx 401.07 + 316.65 = 717.72$$

Now:

$$C_2(36.67) = 0.2(36.67)^2 + 3(36.67)$$

$$(36.67)^2 \approx 1344.89$$

$$C_2(36.67) \approx 0.2(1344.89) + 110.01$$

$$C_2(36.67) \approx 268.98 + 110.01 = 378.99$$

Hence:

$$C_{\min} \approx 717.72 + 378.99 = 1096.71$$

Final result:

$$P_1^* \approx 63.33, \quad P_2^* \approx 36.67$$

$$C_{\min} \approx 1096.71$$

General Conclusion

This course has presented the fundamental tools of numerical analysis and optimization, with a particular focus on their applications in electrical engineering and electromechanical systems.

In the first chapter, we reviewed essential numerical methods used to solve engineering problems, including nonlinear equations, numerical differentiation and integration, and ordinary differential equations. These methods provide the basic computational techniques required to approximate mathematical models that cannot be solved analytically.

In the second chapter, we introduced partial differential equations (PDEs), which play a central role in modeling physical phenomena such as heat transfer, electromagnetic fields, and wave propagation. We studied their classification, boundary conditions, and numerical solution methods, including the Finite Difference Method (FDM), the Finite Volume Method (FVM), and the Finite Element Method (FEM). These techniques allow the discretization and numerical resolution of complex spatial problems encountered in engineering practice.

In the third chapter, we focused on optimization techniques, which are essential for improving system performance and making optimal decisions. We presented the formulation of optimization problems, different types of optimization (single-objective, multi-objective, constrained, and unconstrained), as well as numerical solution methods. Special attention was given to constraint handling techniques such as penalty methods, barrier methods, Lagrange multipliers, and Karush-Kuhn-Tucker (KKT) conditions.

Overall, this course highlights the strong connection between mathematical modeling, numerical methods, and engineering applications. Numerical techniques make it possible to solve complex problems that arise in real-world systems, while optimization provides a framework for selecting the best possible solution under given constraints.

These tools are indispensable in many areas of electrical engineering, including:

- power system analysis and economic dispatch,
- design and optimization of electrical machines,
- thermal analysis and energy management,
- electromagnetic field computation.

Perspective: The methods introduced in this course form a solid foundation for more advanced topics such as nonlinear numerical methods, large-scale simulation, convex optimization, and intelligent optimization techniques (genetic algorithms, particle swarm optimization, etc.), which are widely used in modern engineering and research.

In conclusion, mastering numerical methods and optimization is essential for engineers, as it enables them to model, analyze, and optimize complex systems efficiently and reliably.

Bibliography

- [1] G. Allaire, *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*, Oxford University Press, 2007.
- [2] S. S. Rao, *Engineering Optimization: Theory and Practice*, 4th Edition, John Wiley & Sons, 2009.
- [3] A. Fortin, *Numerical Methods for Engineers*, Presses Internationales Polytechnique, 2011.
- [4] J. Bastien and J. N. Martin, *Introduction to Numerical Analysis: Applications with MATLAB*, Dunod, 2003.
- [5] A. Quarteroni, F. Saleri, and P. Gervasio, *Scientific Computing with MATLAB and Octave*, Springer, 2010.
- [6] T. A. Miloud, *Numerical Methods: Finite Difference, Integral and Variational Methods*, Office des Publications Universitaires, 2013.
- [7] F. Jedrzejewski, *Introduction to Numerical Methods*, Springer, 2001.
- [8] P. Faurre, *Numerical Analysis and Optimization Notes*, École Polytechnique, 1988.
- [9] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th Edition, McGraw-Hill, 2015.
- [10] R. L. Burden and J. D. Faires, *Numerical Analysis*, 10th Edition, Cengage Learning, 2016.