

MTH209

INTRODUCTION TO NUMERICAL ANALYSIS



Download on the
App Store



GET IT ON
Google Play

Course video is available on our app

Series 1: Numerical Solutions of Algebraic and Transcendental Equations

Series Outline

Part 1.1 Introduction to Algebraic and Transcendental Equations

Definition and classification of equations

Importance of numerical solutions

Part 1.2 Iterative Methods for Solving Non-linear Equations

Bisection method

Newton–Raphson method

Secant method

Part 1.3 Numerical Methods for Transcendental Equations

Fixed-point iteration

Applications in science and engineering

Part 1.4 Systems of Linear Equations

Direct methods (Gaussian elimination, LU decomposition)

Iterative methods (Jacobi, Gauss–Seidel)

Introduction

Equations are central to mathematics, science, and engineering, yet many of them cannot be solved exactly using analytical methods. In practice, we often encounter algebraic equations of higher degree and transcendental equations involving functions such as sine, cosine, or exponential, where closed-form solutions are either unavailable or impractical. Numerical analysis provides systematic techniques to approximate these solutions with sufficient accuracy, making it an indispensable tool in applied contexts.

The aim of this Series is to introduce you to the numerical methods used for solving algebraic and transcendental equations, and to equip you with the ability to apply these techniques to real-world problems where exact solutions are not feasible.

We shall commence this Series by examining the nature and classification of algebraic and transcendental equations, and why numerical solutions are necessary.



Next, we will explore iterative methods for solving non-linear equations, including the bisection, Newton–Raphson, and secant methods. Subsequently, we will consider numerical approaches to transcendental equations, focusing on fixed-point iteration and applications in science and engineering. Finally, we will wrap up by studying systems of linear equations, where we will discuss both direct methods such as Gaussian elimination and iterative methods such as Jacobi and Gauss–Seidel.

Series Learning Outcomes

Upon completing this Series, you should be able to:

SLO 1.1 define algebraic and transcendental equations and explain their significance in numerical analysis,

SLO 1.2 apply iterative methods such as the bisection method, Newton–Raphson method, and secant method to solve non-linear equations,

SLO 1.3 demonstrate numerical techniques for solving transcendental equations, including fixed-point iteration,

SLO 1.4 analyse and solve systems of linear equations using direct methods such as Gaussian elimination and iterative methods such as Jacobi and Gauss–Seidel.

1.1 Introduction to Algebraic and Transcendental Equations

Algebraic and transcendental equations are central to numerical analysis because they frequently arise in mathematics, science, and engineering. An **algebraic equation** is one that involves only algebraic operations such as addition, subtraction, multiplication, division, and raising to powers. For example, $x^3 - 2x + 1 = 0$ is algebraic. A **transcendental equation** involves transcendental functions such as exponential, logarithmic, or trigonometric functions. For instance, $\sin(x) = x$ is transcendental because it contains the sine function.

Fill in the blank: An equation that contains trigonometric or exponential functions is called a _____ equation.

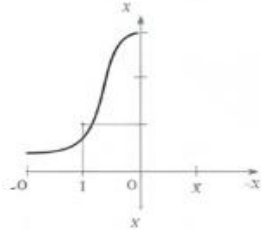


Algebraic vs. Transcendental Equations

Distinguishing Polynomials from Non-Polynomials

1. Algebraic Equations (Polynomials)

$$2x^3 - 4x^2 + x - 7 = 0$$



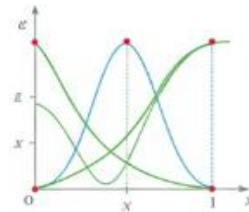
Example 1: $x^2 + 5x - 6 = 0$

$\sqrt{3x^4} = 9$

- Finite number of terms
- Involve only addition, subtraction, multiplication, non-negative integer exponents
- Can be solved exactly or numerically
- Can be solved exactly or numerically
- Graphs are smooth curves

2. Transcendental Equations (Non-Polynomials)

$$ex + \sin(x) = 0$$



Example 1: $\cos(x) = x$

Example 2: $2^x = \log(x)$

- Infinite series expansions
- Involve trigonometric, exponential, logarithmic functions
- Generally solved numerically
- Graphs can have complex behavior

Key Concept: Algebraic equations are built from polynomial operations; transcendental equations function that "transcend" that "algebra", requiring series or numerical methods for their solutions.

Open Educational Resource

Examples of algebraic and transcendental equations

1.1.1 Importance of numerical solutions

In many cases, equations cannot be solved exactly using analytical methods. Numerical methods provide approximate solutions that are sufficiently accurate for practical purposes. For example, while quadratic equations can be solved exactly, higher-order polynomials or transcendental equations often require iterative numerical approaches.

Fill in the blank: Numerical methods provide _____ solutions when exact analytical solutions are not possible.

Why numerical solutions are important Short description: A box highlighting reasons for using numerical methods.



Key Reasons for Using Numerical Methods

Reason	Description
Non-Solvable Equations	Most transcendental equations (e.g., $x + e^x = 0$) and high-degree polynomials cannot be solved for x using basic algebraic operations.
Complexity of Real Data	Real-world problems often involve data sets or measurements rather than neat, continuous functions, requiring algorithms to estimate values.
Differential Equations	Many physical systems are modeled by nonlinear differential equations that have no known closed-form solution.
Efficiency and Speed	For large-scale systems (like weather forecasting or structural analysis), computers can approximate solutions much faster than a human can perform symbolic manipulation.

1.1.2 Classification of equations

Equations can be classified based on their form:

Linear equations: Equations of the first degree, e.g., $2x+3=0$.

Non-linear equations: Equations involving higher powers or transcendental functions, e.g., $x^3-x=0$ or $\cos(x)=x$.

Systems of equations: Sets of equations involving multiple variables, which may be linear or non-linear.

Fill in the blank: Equations of the first degree, such as $2x+3=0$, are classified as _____ equations.

Classification of equations with examples Short description: A table showing linear, non-linear, and transcendental equations with examples.



Classification of Equations

Category	Defining Property		Typical Solution Methods
Linear Equations	$2x + 3 = 7$ $y = mx - b$ $3 + 4y - 2z = 0$		Direct algebraic manipulation, wdet nos, comisert of Eliminations Matric methods
Non-Linear Equations	$x^2 - 5 = 60$ $y = x^3 - 4 \cdot 6$ $y = x^3 = 110$	$x^2 + 5 = 60$ $y = x^3 - 40$ $xy + 2y = 10$	
Examples			
	$\sin(x) = x$ $ex^2 = 2 - x$	$\sin(x) =$ $\ln + 2 = 5$	
Transcendental Equations	$\sin(x) \cdot x = 5$	$\ln(x) \cdot x = 5$	Iterative methods functions analysis functions (o Numerical methods Newton's methods)

Important Notes:

The Peer Tip: The classification of Equations type is based on the method of solving them. In the above table, the equations are classified based on the method of solving them. The equations are classified into Linear, Non-Linear, and Transcendental. The equations are classified based on the method of solving them. The equations are classified into Linear, Non-Linear, and Transcendental.

Pilot Questions 1.1

An equation that involves only algebraic operations such as addition and powers is called: A. Transcendental equation B. Algebraic equation C. Linear system D. Numerical equation (Linked to SLO 1.1)

An equation that involves trigonometric or exponential functions is called: A. Algebraic equation B. Transcendental equation C. Linear equation D. Polynomial equation (Linked to SLO 1.1)



Numerical methods are used when: A. Exact solutions are always possible B. Approximate solutions are sufficient C. Equations are linear only D. Equations are quadratic only (Linked to SLO 1.1)

Equations of the first degree, such as $2x+3=0$, are classified as: A. Non-linear equations B. Linear equations C. Transcendental equations D. Systems of equations (Linked to SLO 1.1)

Which of the following is a transcendental equation? A. $x^2-4=0$ B. $\sin(x)=x$ C. $3x+2=0$ D. $x^3-x=0$ (Linked to SLO 1.1)

1.2 Iterative Methods for Solving Non-linear Equations

Non-linear equations are those that involve variables raised to powers greater than one, products of variables, or transcendental functions. Unlike linear equations, they often cannot be solved exactly, so we rely on **iterative methods**. An iterative method is a step-by-step numerical procedure that generates successive approximations to the solution until the desired accuracy is achieved.

Fill in the blank: A step-by-step numerical procedure that generates successive approximations to a solution is called an _____ method.

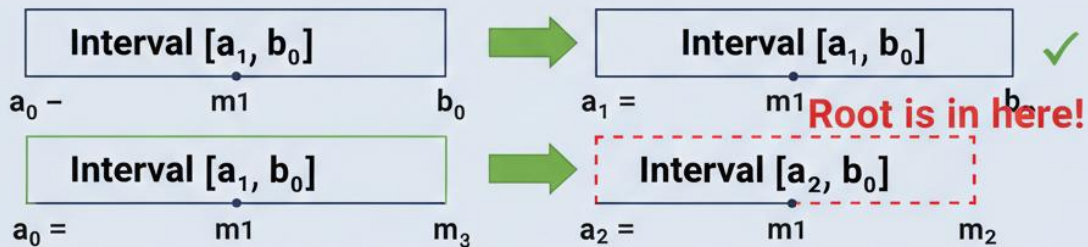
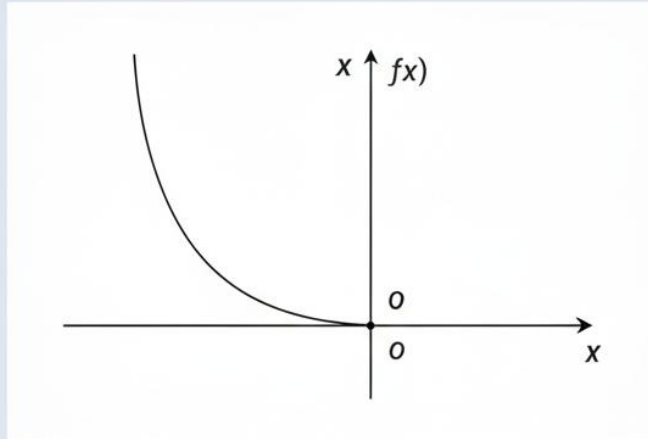
1.2.1 The bisection method

The **bisection method** is a simple iterative technique used to find the root of a continuous function. It works by repeatedly halving an interval where the function changes sign, ensuring that the root lies within the interval. At each step, the midpoint is calculated, and the interval is updated depending on the sign of the function at the midpoint.



Bisection Method for Root Finding

Visually Approximating a Root by Halving Intervals



Key Concept

The Bisection Method repeatedly halves the interval, ensuring root remains inside. The length of the interval is reduced by $\frac{1}{2}$ at each step, guarantee convergence to the root.

Open Educational Resource

Illustration of the bisection method

Fill in the blank: The bisection method requires that the function changes _____ within the chosen interval.

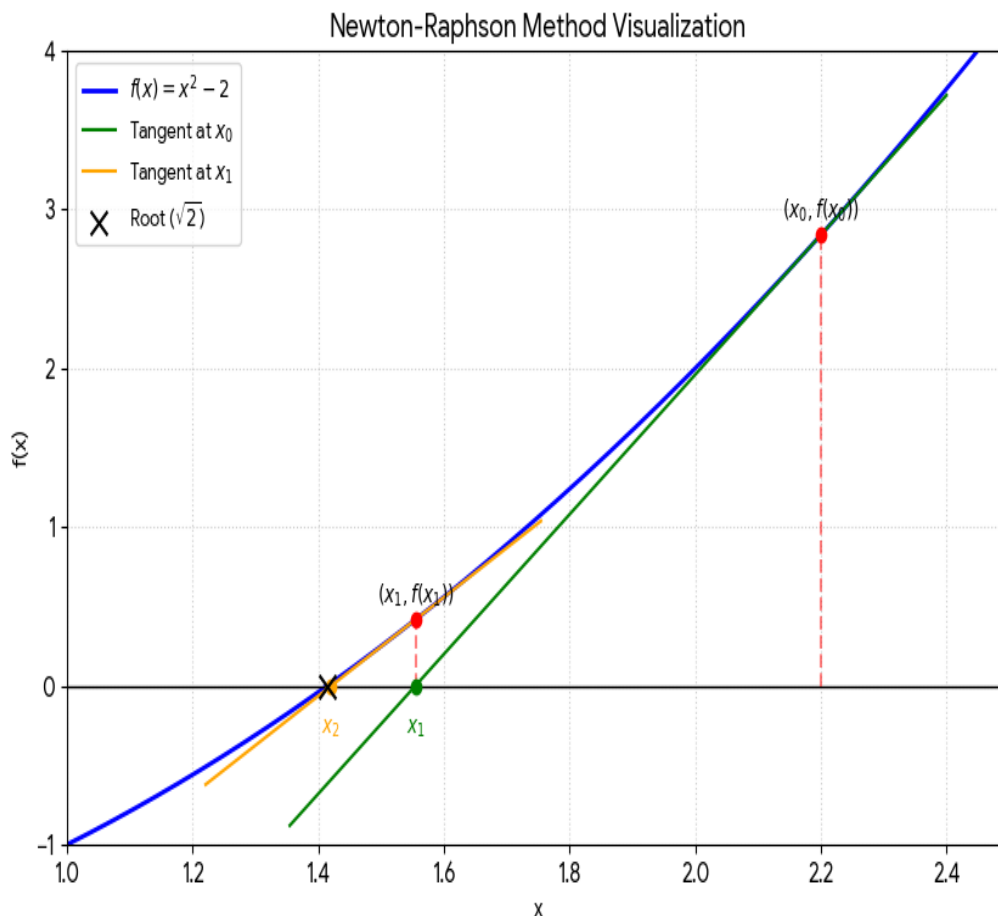
1.2.2 The Newton–Raphson method



The **Newton–Raphson method** is a powerful iterative technique that uses tangents to approximate the root of a function. Starting from an initial guess x_0 , the next approximation is given by:

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

This method converges rapidly if the initial guess is close to the actual root, but it requires the derivative of the function.



Raphson method using tangents to approximate roots

Fill in the blank: The Newton–Raphson method requires the function's _____ to calculate successive approximations.

1.2.3 The secant method



The **secant method** is similar to Newton–Raphson but does not require the derivative of the function. Instead, it approximates the derivative using two previous points. The formula is:

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

This makes the secant method useful when the derivative is difficult to compute. It generally converges faster than the bisection method but may be less stable than Newton–Raphson.

Fill in the blank: The secant method avoids the need for calculating the _____ of the function.

Comparison of iterative methods Short description: A box comparing bisection, Newton–Raphson, and secant methods in terms of speed, stability, and requirements.

Method Comparison Matrix

Feature	Bisection Method	Newton–Raphson	Secant Method
Convergence Speed	Linear (Slowest). Error halves each step.	Quadratic (Fastest). Correct digits double each step.	Superlinear (Medium). Between linear and quadratic.
Stability	Guaranteed to converge if a root is bracketed.	Unstable . Can diverge if x_0 is poor or $f'(x) \approx 0$.	Moderately Stable . Safer than Newton, but can still fail.
Requirements	Requires an interval $[a, b]$ where $f(a) \cdot f(b) < 0$.	Requires the derivative $f'(x)$.	Requires two initial guesses ; no derivative needed.
Best Used For	Getting a rough estimate safely.	Fine-tuning a root when the derivative is easy to find.	Problems where the derivative is too complex to calculate.

Pilot Questions 1.2

The method that repeatedly halves an interval to approximate a root is called:
A. Newton–Raphson method B. Secant method C. Bisection method D. Iteration method (Linked to SLO 1.2)

The Newton–Raphson method requires: A. The derivative of the function B. The midpoint of an interval C. Two previous approximations D. No initial guess (Linked to SLO 1.2)



The secant method avoids the need for: A. Function values B. Derivatives C. Initial guesses D. Intervals (Linked to SLO 1.2)

Which method is generally the slowest but most reliable? A. Newton–Raphson B. Secant C. Bisection D. Iterative substitution (Linked to SLO 1.2)

Which method uses tangents to approximate the root of a function? A. Bisection B. Newton–Raphson C. Secant D. Iterative substitution (Linked to SLO 1.2)

1.3 Numerical Methods for Transcendental Equations

Transcendental equations are those that involve transcendental functions such as exponential, logarithmic, or trigonometric functions. Unlike algebraic equations, they rarely have closed-form solutions, so numerical methods are essential to approximate their roots.

Fill in the blank: Equations that involve exponential or trigonometric functions are called _____ equations.

1.3.1 Fixed-point iteration

The **fixed-point iteration method** is a simple technique for solving transcendental equations. It involves rewriting the equation $f(x)=0$ in the form $x=g(x)$, then using the iterative formula:

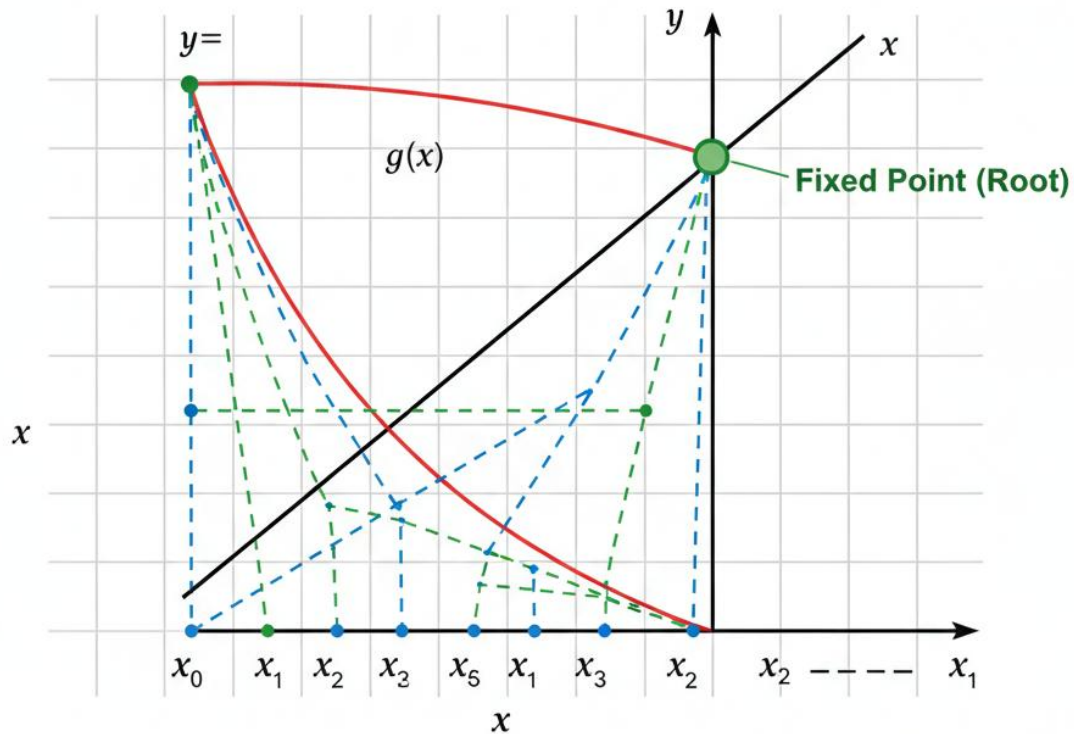
$$x_{n+1}=g(x_n)$$

Starting from an initial guess, successive approximations are generated until convergence is achieved. The method is easy to apply but requires careful choice of $g(x)$ to ensure convergence.



Fixed-Point Iteration Method

Visualizing Convergence to a Root



Key Concept

The Concept: The Fixed-Point Iteration method finds a root by transforming $f(x)=0$ into $x = g(x)$. Iterations $x_{n+1} = g(x_n)$ converge to a fixed point to a fixed point if $|g'(x)| < 1$ in the interval, creating a "cobweb" pattern.

Open Educational Resource

Fixed-point iteration approaching a solution

Fill in the blank: The fixed-point iteration method rewrites the equation $f(x)=0$ in the form _____.

1.3.2 Applications in science and engineering

Transcendental equations appear frequently in applied contexts. For example:

In physics, equations such as $\sin(x) = x$ arise in wave mechanics.



In engineering, exponential equations are used in modelling growth and decay processes.

In electrical engineering, transcendental equations describe resonance and oscillation phenomena.

Numerical methods allow us to approximate solutions to these equations, making them practical tools in real-world problem solving.

Examples of transcendental equations in applications Short description: A box listing examples of transcendental equations in physics and engineering.

Common Applications

Field	Phenomenon	Equation Type / Example
Quantum Mechanics	Particle in a Finite Potential Well	Solving for energy levels involves equations like $z \tan(z) = \sqrt{R^2 - z^2}$.
Heat Transfer	Cooling of Solid Objects	Determining temperature over time in fins or cylinders often requires roots of $Bi = \zeta \tan(\zeta)$ (where Bi is the Biot number).
Astrophysics	Orbital Mechanics	Kepler's Equation: $M = E - e \sin(E)$, used to find the position of a planet in an elliptical orbit.
Electromagnetics	Waveguides	Determining the "cut-off" frequencies for electromagnetic waves in circular or rectangular pipes.
Structural Eng.	Buckling of Columns	Finding the critical load for a beam with one end fixed and one end pinned: $\tan(kL) = kL$.

Fill in the blank: In engineering, transcendental equations are often used to describe _____ and oscillation phenomena.

Pilot Questions 1.3

The method that rewrites an equation in the form $x=g(x)$ is called: A. Newton–Raphson method B. Fixed-point iteration C. Secant method D. Bisection method (Linked to SLO 1.3)



Which of the following is a transcendental equation? A. $x^2 - 4 = 0$ B. $\cos(x) = x$ C. $3x + 2 = 0$ D. $x^3 - x = 0$ (Linked to SLO 1.3)

Fixed-point iteration requires careful choice of: A. Initial guess B. Function derivative C. Interval midpoint D. Function rewrite $g(x)$ (Linked to SLO 1.3)

Transcendental equations are common in: A. Literature B. Physics and engineering C. History D. Geography (Linked to SLO 1.3)

Which of the following applications involves transcendental equations? A. Modelling exponential decay B. Solving linear systems C. Quadratic factorisation D. Polynomial expansion (Linked to SLO 1.3)

1.4 Systems of Linear Equations

Systems of linear equations consist of two or more linear equations involving multiple variables. Solving such systems is a fundamental task in numerical analysis, as they appear in diverse applications such as engineering design, physics simulations, and economics. A **system of linear equations** can be expressed in matrix form as $Ax = b$, where A is a matrix of coefficients, x is a vector of unknowns, and b is a vector of constants.

Fill in the blank: A system of linear equations can be expressed in matrix form as _____.

1.4.1 Direct methods

Direct methods provide exact solutions in a finite number of steps, assuming no rounding errors. The most common direct method is **Gaussian elimination**, which systematically reduces the system to an equivalent one that is easier to solve. Another widely used technique is **LU decomposition**, which factors the coefficient matrix A into a lower triangular matrix L and an upper triangular matrix U .



Gauseian Eleminlation

Solving a 3x3 System of Linear Equations

1. Start with the System

$$\begin{aligned} x + 2y - 3z &= 9 \\ 2x - y + z &= 8 \\ 3x - y + z &= 8 \\ 3x + 0y - 3 &= 3 \end{aligned}$$

$$\left[\begin{array}{cccc|c} 1 & 2 & 3 & 3 & 9 \\ 2 & -1 & 1 & 1 & 8 \\ 3 & 0 & 0 & -1 & -12 \end{array} \right]$$

2. Forward Elmination

$$R_2 - 2R_1 \rightarrow R_2$$

$$\left[\begin{array}{cccc|c} 1 & 2 & 3 & 3 & 9 \\ 0 & -1 & 5 & 1 & -10 \\ 0 & -5 & 5 & -10 & -10 \end{array} \right]$$

$$R_2 / (-5) \rightarrow R_2$$

$$\left[\begin{array}{cccc|c} 1 & 2 & 3 & 3 & 9 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & -6 & -10 & -10 & -24 \end{array} \right]$$

$$R_3 + 6R_2 \rightarrow R_3$$

$$\left[\begin{array}{cccc|c} 1 & 2 & 3 & 3 & 9 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & -4 & -12 \end{array} \right]$$

3. Back Substtution

$$\left[\begin{array}{cccc|c} 1 & 2 & 3 & 3 & 9 \\ 0 & 1 & 3 & 1 & 2 \\ 0 & 0 & 0 & -4 & -12 \end{array} \right]$$

$$\begin{aligned} -4z &= -12 \Rightarrow z = 3 \\ \text{Plug } z=3 \text{ into } R_2: \\ y + (3) &= 2 \Rightarrow y = -1 \\ \text{Plug } y=-1, z=3 \text{ into } R_1: \\ x - 2 + 9 &= 9 \Rightarrow x = 2 \end{aligned}$$

Solution: $x=2, y=-1, z=3$

Key Concept: Gauseian Elimination transforms a system into an upper triangular matrix, which is then easily solved by working backwards.

Open Educational Resource

Gaussian elimination applied to a system of equations

Fill in the blank: LU decomposition factors a matrix into _____ and _____ triangular matrices.

1.4.2 Iterative methods

Iterative methods are particularly useful for large systems where direct methods become computationally expensive. Two common iterative methods are:



Jacobi method: Approximates each variable in turn using the most recent values.

Gauss–Seidel method: Similar to Jacobi but uses updated values immediately, often converging faster.

These methods generate successive approximations until the solution converges within a desired tolerance.

Comparison of Jacobi and Gauss–Seidel methods Short description: A box comparing Jacobi and Gauss–Seidel methods in terms of convergence speed and computational efficiency.

Performance Comparison Matrix

Feature	Jacobi Method	Gauss–Seidel Method
Basic Logic	Uses values from the previous iteration (k) to calculate all values for the current iteration ($k + 1$).	Uses the most recent values available (mixing current and previous iteration data).
Convergence Speed	Generally slower . Often requires twice as many iterations as Gauss–Seidel.	Generally faster . Usually converges in about half the iterations of Jacobi.
Memory Usage	Higher . Must store the entire previous vector ($x^{(k)}$) until the new one is finished.	Lower . Can overwrite values as they are calculated, requiring only one vector in memory.
Parallelization	High . Since steps are independent, it is easily computed on multiple CPU/GPU cores simultaneously.	Low . Inherently sequential because each calculation depends on the step immediately before it.

Fill in the blank: The iterative method that uses updated values immediately for faster convergence is called the _____ method.

Pilot Questions 1.4

A system of linear equations can be expressed in matrix form as: A. $Ax=b$ B. $Ax=xC$. $Ax=AD$. $Ax=0$ (Linked to SLO 1.4)

The method that reduces a system step by step to an equivalent one is: A. Jacobi method B. Gaussian elimination C. Secant method D. Newton–Raphson method (Linked to SLO 1.4)



LU decomposition factors a matrix into: A. Two diagonal matrices B. Lower and upper triangular matrices C. Two identity matrices D. Two polynomial matrices (Linked to SLO 1.4)

Which iterative method uses updated values immediately for faster convergence? A. Jacobi method B. Gauss–Seidel method C. Bisection method D. Newton–Raphson method (Linked to SLO 1.4)

Iterative methods are particularly useful for: A. Small systems B. Large systems C. Quadratic equations D. Transcendental equations (Linked to SLO 1.4)

Series Summary

In this Series, we have explored numerical solutions of algebraic and transcendental equations, which form the foundation of numerical analysis. We began by introducing algebraic and transcendental equations, highlighting their definitions, classifications, and the importance of numerical methods when exact solutions are not possible. We then examined iterative methods for solving non-linear equations, including the bisection, Newton–Raphson, and secant methods, each with its strengths and limitations. Following that, we studied numerical approaches to transcendental equations, focusing on fixed-point iteration and their applications in science and engineering. Finally, we rounded off the Series with systems of linear equations, discussing both direct methods such as Gaussian elimination and LU decomposition, and iterative methods such as Jacobi and Gauss–Seidel.

By completing this Series, you should now be able to define algebraic and transcendental equations, apply iterative methods to solve non-linear equations, demonstrate numerical techniques for transcendental equations, and analyse and solve systems of linear equations using both direct and iterative approaches. These abilities directly reflect the Series Learning Outcomes (SLOs) and provide you with essential skills for tackling practical problems in numerical analysis.

Glossary of Terms

Algebraic equation: An equation involving only algebraic operations such as addition, subtraction, multiplication, division, and powers.

Bisection method: An iterative technique that repeatedly halves an interval where a function changes sign to approximate a root.

Fixed-point iteration: A numerical method that rewrites an equation in the form $x=g(x)$ and generates successive approximations until convergence.



Gauss–Seidel method: An iterative technique for solving systems of linear equations that uses updated values immediately, often converging faster than the Jacobi method.

Gaussian elimination: A direct method for solving systems of linear equations by systematically reducing them to simpler forms.

Iterative method: A step-by-step numerical procedure that generates successive approximations to a solution.

Jacobi method: An iterative technique for solving systems of linear equations by approximating each variable in turn using the most recent values.

LU decomposition: A direct method that factors a matrix into lower and upper triangular matrices for easier solution of systems.

Newton–Raphson method: An iterative technique that uses tangents and derivatives to approximate the root of a function.

Secant method: An iterative technique similar to Newton–Raphson but avoids the need for derivatives by using two previous approximations.

System of linear equations: A set of two or more linear equations involving multiple variables, often expressed in matrix form as $Ax=b$.

Transcendental equation: An equation involving transcendental functions such as exponential, logarithmic, or trigonometric functions.

Concept Check Questions 1

Objective questions

Which of the following is an algebraic equation? A. $\sin(x)=x$ B. $x^3-2x+1=0$ C. $\cos(x)=x$ D. $e^x=2$ (Linked to SLO 1.1)

The method that uses tangents and derivatives to approximate a root is: A. Secant method B. Newton–Raphson method C. Bisection method D. Jacobi method (Linked to SLO 1.2)

LU decomposition factors a matrix into: A. Two diagonal matrices B. Lower and upper triangular matrices C. Two identity matrices D. Polynomial matrices (Linked to SLO 1.4)

Short-answer questions

Define a transcendental equation and give one example. (Linked to SLO 1.1)



Explain why iterative methods are important in solving non-linear equations.
(Linked to SLO 1.2)

Long-answer questions

Compare the bisection, Newton–Raphson, and secant methods in terms of speed, stability, and requirements. (Linked to SLO 1.2)

Basic response: Describe each method and its main requirement.

Value-added response: Analyse their relative strengths and weaknesses, including convergence speed and reliability.

Discuss the role of transcendental equations in applied science and engineering.
(Linked to SLO 1.3)

Basic response: Provide examples of transcendental equations in physics and engineering.

Value-added response: Evaluate how numerical methods make these equations solvable in practice, with reference to real-world applications.

Answers to Concept Check Questions 1

B. $x^3 - 2x + 1 = 0$

B. Newton–Raphson method

B. Lower and upper triangular matrices

A transcendental equation involves transcendental functions such as exponential, logarithmic, or trigonometric functions. Example: $\cos(x) = x$.

Iterative methods are important because many non-linear equations cannot be solved exactly, and iterative approaches provide successive approximations until convergence.

Basic response: The bisection method halves intervals, Newton–Raphson uses derivatives, and the secant method uses two previous approximations.

Value-added response: Bisection is slow but reliable, Newton–Raphson is fast but requires derivatives and a good initial guess, while the



secant method is faster than bisection and avoids derivatives but may be less stable.

Basic response: Transcendental equations appear in wave mechanics, exponential growth and decay, and resonance problems.

Value-added response: Numerical methods make it possible to approximate solutions in engineering design, physics simulations, and electrical circuit analysis where analytical solutions are impractical.

Answers to Pilot Questions 1

Algebraic equation

Transcendental equation

Approximate solutions are sufficient

Linear equations

$\sin(x)=x$

Bisection method

The derivative of the function

Derivatives

Bisection

Newton–Raphson

Fixed-point iteration

$\cos(x)=x$

Function rewrite $g(x)$

Physics and engineering

Modelling exponential decay

$Ax=b$

Gaussian elimination

Lower and upper triangular matrices



Gauss–Seidel method

Large systems

Series 2: Curve Fitting and Error Analysis

Series Outline

Part 2.1 Introduction to Curve Fitting and Error Concepts

Definition of curve fitting

Types of errors in numerical analysis

Part 2.2 Least Squares Method

Principle of least squares

Linear curve fitting

Polynomial curve fitting

Part 2.3 Interpolation Techniques

Newton's forward and backward interpolation

Lagrange interpolation

Part 2.4 Error Analysis in Numerical Methods

Sources of error (truncation, rounding)

Propagation of error

Error bounds and stability

Introduction

In many areas of applied mathematics, science, and engineering, we often need to approximate data or functions rather than work with exact values. This is where curve fitting and error analysis become essential. Curve fitting allows us to find mathematical models that best represent observed data, while error analysis helps us understand the accuracy and reliability of numerical results. Together, they provide the tools to make informed decisions when working with approximations.



The aim of this Series is to introduce you to the principles and techniques of curve fitting and error analysis, and to equip you with the ability to apply these methods to practical problems where data and numerical computations are involved.

We shall commence this Series by examining the basic concepts of curve fitting and the different types of errors encountered in numerical analysis. Next, we will explore the least squares method, focusing on linear and polynomial curve fitting. Subsequently, we will consider interpolation techniques, including Newton's forward and backward interpolation as well as Lagrange interpolation. Finally, we will wrap up by studying error analysis in numerical methods, where we will discuss sources of error, error propagation, and stability.

Series Learning Outcomes

Upon completing this Series, you should be able to:

SLO 2.1 define the concept of curve fitting and identify different types of errors in numerical analysis,

SLO 2.2 apply the least squares method to perform linear and polynomial curve fitting,

SLO 2.3 demonstrate interpolation techniques such as Newton's forward and backward interpolation and Lagrange interpolation,

SLO 2.4 analyse sources of error in numerical methods, including truncation and rounding, and evaluate their impact on accuracy and stability.

2.1 Introduction to Curve Fitting and Error Concepts

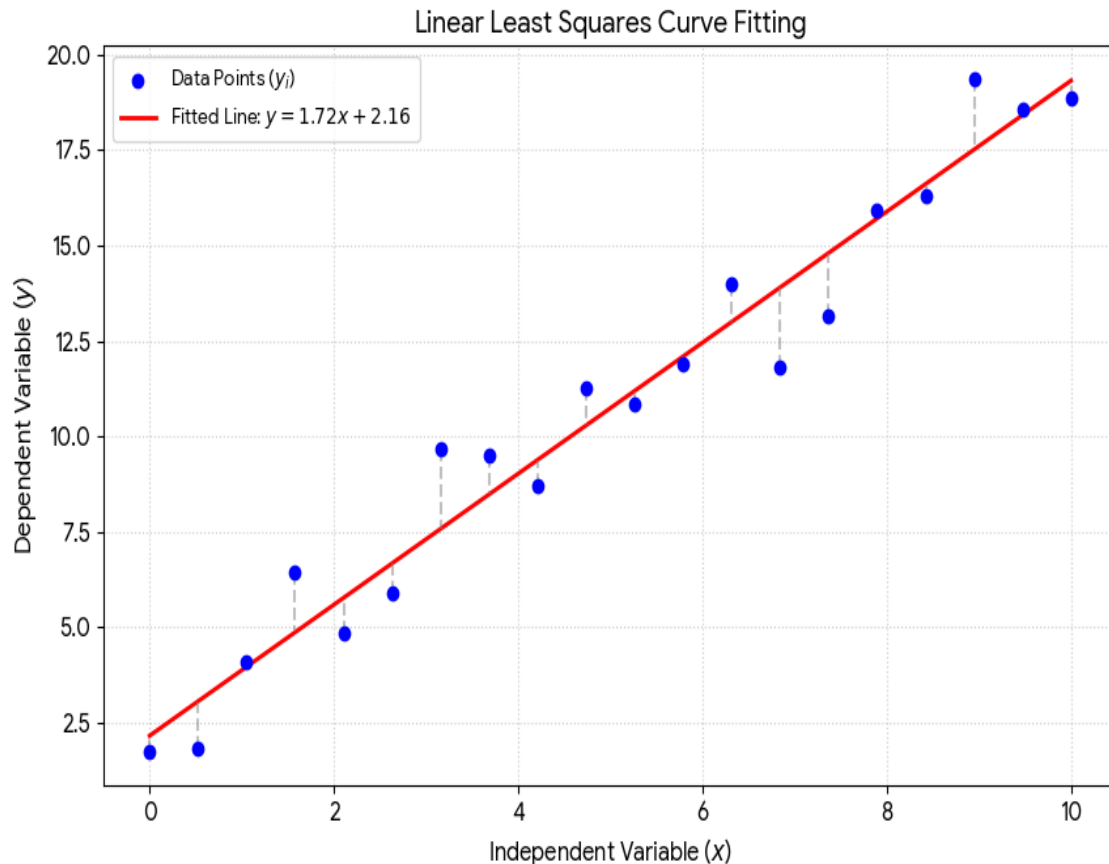
Curve fitting and error analysis are essential tools in numerical analysis because they allow us to approximate data and evaluate the accuracy of numerical computations. **Curve fitting** is the process of finding a mathematical function that best represents a set of observed data points. This is particularly useful when dealing with experimental data or when the exact relationship between variables is unknown. **Error analysis** refers to the study of inaccuracies that arise in numerical methods, helping us to measure and control the reliability of results.

Fill in the blank: The process of finding a mathematical function that best represents observed data points is called _____.

2.1.1 Definition of curve fitting



Curve fitting involves selecting a function that approximates the relationship between variables in a dataset. The chosen function may be linear, polynomial, or more complex depending on the nature of the data. The goal is to minimise the difference between the observed data points and the values predicted by the function.



Example of curve fitting with a linear function AI prompt: “

Fill in the blank: Curve fitting aims to minimise the difference between observed data points and _____ values.

2.1.2 Types of errors in numerical analysis

Errors in numerical analysis can be classified into several types:

Truncation error: The error that arises when an infinite process is approximated by a finite one, such as stopping a series expansion after a few terms.



Rounding error: The error caused by representing numbers with limited precision, common in computer calculations.

Propagation of error: The way errors accumulate and spread through successive steps in a numerical method.

Types of errors in numerical analysis Short description: A box listing truncation, rounding, and propagation errors with brief explanations

Core Error Definitions

Error Type	Definition	Common Cause
Truncation Error	The error caused by approximating a mathematical process that involves an infinite number of steps with a finite number of steps.	Using a Taylor series with only 3 terms instead of an infinite number; stopping an iterative method early.
Rounding Error	The error caused by the inability of a computer to represent real numbers with perfect precision due to finite memory (floating-point representation).	Representing $\frac{1}{3}$ as 0.333333 or π as 3.14159265.
Propagation Error	The error that occurs when initial inaccuracies (from rounding or truncation) are magnified through subsequent mathematical operations.	Subtracting two nearly equal numbers (catastrophic cancellation) or dividing by a very small number.

Fill in the blank: The error that arises when an infinite process is approximated by a finite one is called _____ error.

Pilot Questions 2.1

The process of finding a mathematical function that best represents observed data points is called: A. Interpolation B. Curve fitting C. Error analysis D. Approximation (Linked to SLO 2.1)

Which of the following is an example of curve fitting? A. Solving a quadratic equation B. Drawing a line through experimental data points C. Calculating a derivative D. Performing Gaussian elimination (Linked to SLO 2.1)

The error caused by representing numbers with limited precision is called: A. Truncation error B. Rounding error C. Propagation error D. Approximation error (Linked to SLO 2.1)



Which type of error arises when an infinite process is approximated by a finite one? A. Rounding error B. Truncation error C. Propagation error D. Systematic error (Linked to SLO 2.1)

Errors that accumulate and spread through successive steps in a numerical method are called: A. Rounding errors B. Propagation errors C. Truncation errors D. Approximation errors (Linked to SLO 2.1)

2.2 Least Squares Method

The **least squares method** is one of the most widely used techniques in curve fitting. It provides a systematic way to find the curve that best represents a set of data points by minimising the sum of the squares of the differences between the observed values and the values predicted by the curve. This ensures that the overall error is as small as possible, making the fitted curve a reliable representation of the data.

Fill in the blank: The least squares method minimises the sum of the _____ of the differences between observed and predicted values.

2.2.1 Principle of least squares

The principle of least squares is based on the idea that the best-fitting curve is the one that minimises the total squared error. If we have data points y_i , the error for each point is the difference between the observed value y_i and the predicted value $\hat{y}_i$. The least squares method seeks to minimise:

$$S = \sum (y_i - \hat{y}_i)^2$$

This principle applies to both linear and polynomial curve fitting.

Principle of least squares minimising squared errors

Fill in the blank: The least squares method seeks to minimise the total _____ error.

2.2.2 Linear curve fitting

Linear curve fitting involves fitting a straight line of the form $y = a + bx$ to a set of data points. The coefficients a and b are chosen to minimise the sum of squared errors. This is particularly useful when the relationship between variables is approximately linear.



Linear curve fitting using least squares

Fill in the blank: Linear curve fitting involves fitting a straight line of the form _____ to data points.

2.2.3 Polynomial curve fitting

Polynomial curve fitting extends the least squares method to higher-order polynomials, such as quadratic or cubic functions. This is useful when the data shows curvature that cannot be captured by a straight line. The polynomial coefficients are determined to minimise the squared error, just as in the linear case.

Polynomial curve fitting using least squares

Fill in the blank: Polynomial curve fitting is useful when the data shows _____ that cannot be captured by a straight line.

Pilot Questions 2.2

The least squares method minimises: A. The sum of absolute errors B. The sum of squared errors C. The maximum error D. The average error (Linked to SLO 2.2)

The principle of least squares seeks to minimise: A. Total squared error B. Total rounding error C. Total truncation error D. Total propagation error (Linked to SLO 2.2)

Linear curve fitting involves fitting a line of the form: A. $y = ax^2 + bx + c$ B. $y = a + bx$ C. $y = ax^3 + bx^2 + cx + d$ D. $y = \sin(x)$ (Linked to SLO 2.2)

Polynomial curve fitting is useful when: A. Data is perfectly linear B. Data shows curvature C. Data has no variation D. Data is constant (Linked to SLO 2.2)

Which of the following is an example of polynomial curve fitting? A. Fitting a straight line B. Fitting a quadratic curve C. Fitting a constant function D. Fitting a logarithmic curve (Linked to SLO 2.2)



2.3 Interpolation Techniques

Interpolation is a numerical method used to estimate values of a function between known data points. Unlike curve fitting, which seeks a general model for the data, **interpolation** ensures that the approximating function passes exactly through the given data points. This makes it particularly useful when precise values are required at intermediate points.

Fill in the blank: The numerical method that estimates values of a function between known data points is called _____.

2.3.1 Newton's forward and backward interpolation

Newton's forward interpolation is used when data points are near the beginning of a table, while **Newton's backward interpolation** is applied when data points are near the end. Both methods rely on finite differences to construct an interpolating polynomial.

The general form of Newton's forward interpolation is:

$$f(x) = f(x_0) + u\Delta f(x_0) + \frac{u(u-1)}{2!}\Delta^2 f(x_0) + \dots$$

where $u = \frac{x - x_0}{h}$, h is the interval size, and Δ represents finite differences.

Newton's forward interpolation polynomial

Fill in the blank: Newton's backward interpolation is most useful when data points are near the _____ of a table.

2.3.2 Lagrange interpolation

The **Lagrange interpolation formula** provides a polynomial that passes through all given data points. For $n+1$ data points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, the interpolating polynomial is:

$$P(x) = \sum_{j=0}^n y_j \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i}$$

This method is flexible and does not require equally spaced data points, making it widely applicable.



Lagrange interpolation polynomial

Fill in the blank: Lagrange interpolation does not require data points to be _____ spaced.

Pilot Questions 2.3

Interpolation ensures that the approximating function: A. Passes near the data points B. Passes exactly through the data points C. Minimises squared errors D. Avoids using data points (Linked to SLO 2.3)

Newton's forward interpolation is most useful when: A. Data points are near the end of a table B. Data points are near the beginning of a table C. Data points are evenly spaced D. Data points are random (Linked to SLO 2.3)

Newton's backward interpolation relies on: A. Derivatives B. Finite differences C. Least squares D. Error bounds (Linked to SLO 2.3)

Lagrange interpolation is advantageous because: A. It requires equally spaced data points B. It does not require equally spaced data points C. It minimises squared errors D. It avoids polynomial functions (Linked to SLO 2.3)

Which interpolation technique constructs a polynomial that passes through all given data points? A. Newton's forward interpolation B. Newton's backward interpolation C. Lagrange interpolation D. Least squares method (Linked to SLO 2.3)

2.4 Error Analysis in Numerical Methods

Error analysis is a crucial aspect of numerical methods because it helps us evaluate the accuracy and reliability of computed results. Every numerical approximation introduces some level of error, and understanding these errors allows us to control their impact and ensure that solutions remain meaningful in practice.

Fill in the blank: The study of inaccuracies that arise in numerical methods is called _____ analysis.

2.4.1 Sources of error

Errors in numerical methods arise from different sources:



Truncation error: Occurs when an infinite process, such as a series expansion, is cut short after a finite number of terms.

Rounding error: Results from representing numbers with limited precision, especially in computer arithmetic.

Modelling error: Arises when the mathematical model itself does not perfectly represent the real-world situation.

Sources of error in numerical methods Short description: A box listing truncation, rounding, and modelling errors with brief explanations.

Error Type	Definition	Example Scenario
Modelling Error	The difference between a real physical system and the mathematical equations used to represent it.	Ignoring air resistance in a projectile motion calculation or assuming a fluid is inviscid.
Truncation Error	The error caused by replacing an infinite mathematical process (like a limit or an infinite sum) with a finite one.	Using only the first few terms of a Taylor Series to approximate e^x instead of the infinite sequence.
Rounding Error	The discrepancy arising from the fact that computers represent real numbers with a finite number of digits (precision).	Representing the number $1/7$ as a decimal string like 0.14285714, losing the infinite tail of digits.

Fill in the blank: The error that arises when an infinite process is cut short is called _____ error.

2.4.2 Propagation of error

Errors do not remain isolated; they often propagate through successive steps of a numerical computation. For example, a small rounding error in one step may grow



larger as calculations continue. Understanding error propagation is essential for designing stable numerical algorithms.

Propagation of error in numerical computations

Fill in the blank: Small rounding errors may grow larger as calculations continue due to _____ of error.

2.4.3 Error bounds and stability

Numerical methods are evaluated not only by their accuracy but also by their **stability**. A stable method ensures that errors do not grow uncontrollably during computation. **Error bounds** provide estimates of the maximum possible error in a numerical solution, helping us judge whether the result is sufficiently reliable.

Comparison of stable vs unstable numerical methods Short description: A table comparing characteristics of stable and unstable methods with examples.

Feature	Stable Numerical Method	Unstable Numerical Method
Error Propagation	Damping: Errors in input or rounding are suppressed or stay bounded over time.	Amplification: Small initial errors or rounding discrepancies grow exponentially.
Sensitivity	Robust to small perturbations in data or machine precision.	Highly sensitive; small changes in input lead to wildly different outputs.
Convergence	Usually guaranteed to converge to the true solution if the method is consistent.	May diverge or oscillate away from the true solution even with small step sizes.
Error Bounds	Error is typically bounded by a small multiple of the machine epsilon: $ E = O(\epsilon)$.	Error bounds can grow without limit as the number of iterations increases.



Feature	Stable Numerical Method	Unstable Numerical Method
Conditioning	Often used to solve well-conditioned problems reliably.	Can turn even a well-conditioned problem into an inaccurate solution.

Fill in the blank: A numerical method is considered _____ if errors do not grow uncontrollably during computation.

Pilot Questions 2.4

The error that arises when an infinite process is cut short is called: A. Rounding error B. Truncation error C. Propagation error D. Modelling error (Linked to SLO 2.4)

Which type of error results from representing numbers with limited precision? A. Truncation error B. Rounding error C. Modelling error D. Propagation error (Linked to SLO 2.4)

Errors that spread through successive steps in a numerical computation are called: A. Rounding errors B. Propagation errors C. Truncation errors D. Approximation errors (Linked to SLO 2.4)

A numerical method is considered stable if: A. Errors grow uncontrollably B. Errors remain bounded C. Errors are ignored D. Errors are eliminated completely (Linked to SLO 2.4)

Error bounds provide: A. Exact solutions B. Estimates of maximum possible error C. Ways to eliminate all errors D. Random approximations (Linked to SLO 2.4)

Series Summary

In this Series, we examined the principles and applications of curve fitting and error analysis, which are vital tools in numerical analysis. We began with an introduction to curve fitting and error concepts, defining curve fitting as the process of finding a mathematical function that best represents observed data, and identifying key types of errors such as truncation, rounding, and propagation. We then explored the least squares method, focusing on its principle of minimising squared errors, and applied it to both linear and polynomial curve fitting.



Following that, we studied interpolation techniques, including Newton's forward and backward interpolation, which rely on finite differences, and Lagrange interpolation, which constructs a polynomial passing through all given data points. Finally, we rounded off the Series with error analysis in numerical methods, discussing sources of error, error propagation, error bounds, and the importance of stability in numerical algorithms.

By completing this Series, you should now be able to define curve fitting and error concepts, apply the least squares method to linear and polynomial data, demonstrate interpolation techniques, and analyse sources of error and their impact on numerical stability. These abilities directly reflect the Series Learning Outcomes (SLOs) and provide you with practical skills for handling data approximation and evaluating numerical accuracy.

Glossary of Terms

Curve fitting: The process of finding a mathematical function that best represents observed data points.

Error analysis: The study of inaccuracies that arise in numerical methods, used to evaluate accuracy and reliability.

Interpolation: A numerical method that estimates values of a function between known data points, ensuring the function passes exactly through them.

Least squares method: A technique that minimises the sum of squared differences between observed and predicted values to find the best-fitting curve.

Linear curve fitting: Curve fitting using a straight line of the form $y=a+bx$.

Polynomial curve fitting: Curve fitting using higher-order polynomials such as quadratic or cubic functions.

Newton's forward interpolation: An interpolation method using finite differences for data points near the beginning of a table.

Newton's backward interpolation: An interpolation method using finite differences for data points near the end of a table.

Lagrange interpolation: A polynomial interpolation method that does not require equally spaced data points.

Truncation error: Error arising when an infinite process is approximated by a finite one.



Rounding error: Error caused by representing numbers with limited precision.

Propagation of error: The way errors accumulate and spread through successive steps in a numerical method.

Error bounds: Estimates of the maximum possible error in a numerical solution.

Stability: A property of numerical methods where errors do not grow uncontrollably during computation.

Concept Check Questions 2

Objective questions

The process of finding a mathematical function that best represents observed data points is called: A. Interpolation B. Curve fitting C. Error analysis D. Approximation (Linked to SLO 2.1)

The least squares method minimises: A. The sum of absolute errors B. The sum of squared errors C. The maximum error D. The average error (Linked to SLO 2.2)

Newton's backward interpolation is most useful when: A. Data points are near the beginning of a table B. Data points are near the end of a table C. Data points are evenly spaced D. Data points are random (Linked to SLO 2.3)

Which type of error results from representing numbers with limited precision? A. Truncation error B. Rounding error C. Modelling error D. Propagation error (Linked to SLO 2.4)

A numerical method is considered stable if: A. Errors grow uncontrollably B. Errors remain bounded C. Errors are ignored D. Errors are eliminated completely (Linked to SLO 2.4)

Short-answer questions

Define interpolation and explain how it differs from curve fitting. (Linked to SLO 2.3)

State the principle of the least squares method in your own words. (Linked to SLO 2.2)

Identify two sources of error in numerical methods and briefly explain each. (Linked to SLO 2.4)



Long-answer questions

Compare linear and polynomial curve fitting in terms of applicability and accuracy. (Linked to SLO 2.2)

Basic response: Describe the difference between fitting a straight line and fitting a polynomial curve.

Value-added response: Analyse when each method is most appropriate, including examples of data patterns.

Discuss the importance of error analysis in numerical methods. (Linked to SLO 2.4)

Basic response: Explain why errors occur and why they must be considered.

Value-added response: Evaluate how error bounds and stability influence the reliability of numerical algorithms in real-world applications.

Answers to Concept Check Questions 2

B. Curve fitting

B. The sum of squared errors

B. Data points are near the end of a table

B. Rounding error

B. Errors remain bounded

Interpolation is a numerical method that estimates values between known data points, ensuring the function passes exactly through them. Curve fitting, by contrast, finds a general model that best represents data but does not necessarily pass through all points.

The least squares method seeks to minimise the total squared differences between observed and predicted values, ensuring the best fit.

Truncation error occurs when an infinite process is cut short, while rounding error arises from representing numbers with limited precision.

Basic response: Linear curve fitting uses a straight line, while polynomial curve fitting uses higher-order polynomials.

Value-added response: Linear fitting is suitable for approximately linear data, while polynomial fitting captures curvature. Polynomial fitting



may provide better accuracy but risks overfitting if the degree is too high.

Basic response: Error analysis is important because numerical methods always introduce inaccuracies, and these must be measured and controlled.

Value-added response: Error bounds provide estimates of maximum error, while stability ensures errors do not grow uncontrollably. Together, they determine whether numerical results are reliable in practical applications such as engineering simulations or scientific modelling.

Series 3: Numerical Differentiation and Integration

Series Outline

Part 3.1 Introduction to Numerical Differentiation and Integration

Motivation and importance

Basic concepts and definitions

Part 3.2 Numerical Differentiation

Finite difference approximations (forward, backward, central differences)

Higher-order derivatives

Error analysis in differentiation

Part 3.3 Numerical Integration

Trapezoidal rule

Simpson's rule

Midpoint rule

Error bounds in integration

Part 3.4 Applications of Numerical Differentiation and Integration

Engineering and physics examples

Computational efficiency considerations

Stability and accuracy in practice



Introduction

In many practical problems, we need to approximate derivatives and integrals when exact analytical solutions are either too complex or impossible to obtain. This is where **numerical differentiation and integration** become essential. Numerical differentiation provides approximate values of derivatives using discrete data points, while numerical integration estimates the area under a curve or the total accumulated value of a function.

These techniques are widely used in engineering, physics, and computational sciences. For example, engineers use numerical differentiation to analyse rates of change in systems, while numerical integration is applied to calculate distances, areas, and energy values from experimental data.

In this Series, we will begin with an introduction to the motivation and importance of numerical differentiation and integration. We will then study numerical differentiation methods, including finite difference approximations and error analysis. Next, we will explore numerical integration techniques such as the trapezoidal rule, Simpson's rule, and the midpoint rule, along with their error bounds. Finally, we will examine applications of these methods in science and engineering, focusing on computational efficiency and stability.

Series Learning Outcomes

Upon completing this Series, you should be able to:

SLO 3.1 explain the motivation and importance of numerical differentiation and integration in applied mathematics and computational sciences,

SLO 3.2 apply finite difference approximations (forward, backward, and central differences) to estimate derivatives and analyse associated errors,

SLO 3.3 use numerical integration techniques such as the trapezoidal rule, Simpson's rule, and the midpoint rule to approximate definite integrals,

SLO 3.4 evaluate the accuracy, stability, and efficiency of numerical differentiation and integration methods in practical applications.

3.1 Introduction to Numerical Differentiation and Integration

Numerical differentiation and integration are techniques used to approximate derivatives and integrals when exact analytical solutions are difficult or impossible to obtain. **Numerical differentiation** provides approximate values of derivatives



using discrete data points, while **numerical integration** estimates the area under a curve or the accumulated value of a function. These methods are essential in applied mathematics, engineering, and computational sciences, where data is often discrete and exact formulas are unavailable.

Fill in the blank: Approximating derivatives using discrete data points is called numerical _____.

3.1.1 Motivation and importance

The motivation for numerical differentiation and integration arises from practical problems. For example:

In physics, derivatives represent rates of change such as velocity and acceleration, while integrals represent quantities like displacement and energy.

In engineering, numerical integration is used to calculate stresses, loads, and material properties from experimental data.

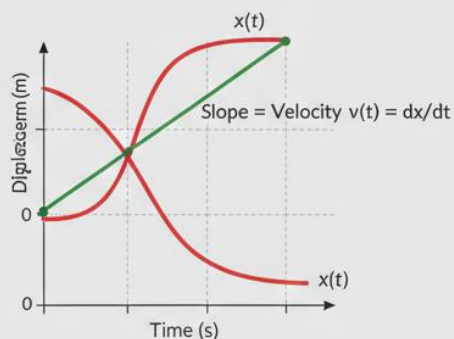
In computational sciences, both differentiation and integration are applied to solve complex models where analytical solutions are not feasible.



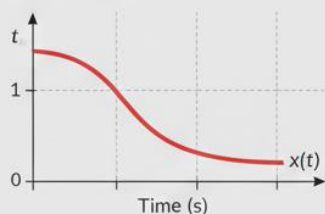
Calculus in Motion: Differentiation & Integration

Visualizing the Relationship Between Kinematic Variables

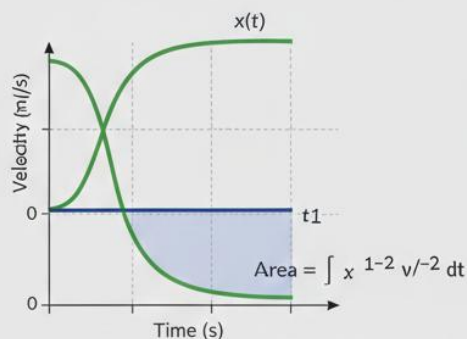
1. Differentiation: Finding Velocity from Displacement



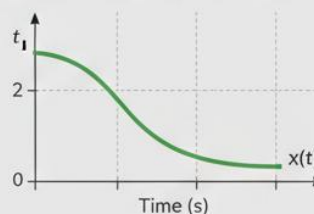
Velocity is the instantaneous rate of change of displacement.



2. Integration: Finding Displacement from Velocity



Displacement is the cumulative area under the velocity-time curve.



Key Concept: Differentiation finds the rate of change, linking displacement to velocity to acceleration. Integration sums these linking velocity back to displacement.

Open Educational Resource

Motivation for numerical differentiation and integration

Fill in the blank: In physics, derivatives represent rates of change such as _____ and acceleration.

3.1.2 Basic concepts and definitions

Finite differences: Approximations of derivatives using differences between successive data points.



Trapezoidal rule: A numerical integration method that approximates the area under a curve by dividing it into trapezoids.

Simpson's rule: A more accurate integration method that uses parabolic arcs to approximate the curve.

Error bounds: Estimates of the maximum possible error in numerical differentiation or integration.

Key concepts in numerical differentiation and integration Short description: A box listing finite differences, trapezoidal rule, Simpson's rule, and error bounds with brief definitions.

Numerical Differentiation (Finite Differences)

Concept	Definition	Formula Sketch
Forward Difference	Approximates the slope using the current point and the next point.	$f'(x) \approx \frac{f(x+h)-f(x)}{h}$
Backward Difference	Approximates the slope using the current point and the previous point.	$f'(x) \approx \frac{f(x)-f(x-h)}{h}$
Central Difference	Uses points on both sides of x . It is generally more accurate than forward or backward methods.	$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$

Numerical Integration (Quadrature)

Method	Definition	Characteristics
Trapezoidal Rule	Approximates the area under a curve by dividing it into trapezoids .	Simple to implement; error is proportional to h^2 .
Simpson's 1/3 Rule	Approximates the area using parabolic arcs (quadratics) instead of straight lines.	Much more accurate; requires an even number of intervals.
Simpson's 3/8 Rule	Uses cubic polynomials to approximate the area.	Best for intervals that are multiples of three.

the area under a curve by dividing it into _____.

Pilot Questions 3.1



Approximating derivatives using discrete data points is called: A. Numerical integration B. Numerical differentiation C. Curve fitting D. Error analysis (Linked to SLO 3.1)

In physics, derivatives represent: A. Energy and displacement B. Velocity and acceleration C. Stress and strain D. Area and volume (Linked to SLO 3.1)

The trapezoidal rule approximates the area under a curve by dividing it into: A. Rectangles B. Trapezoids C. Circles D. Parabolas (Linked to SLO 3.1)

Simpson's rule approximates the curve using: A. Straight lines B. Parabolic arcs C. Cubic splines D. Finite differences (Linked to SLO 3.1)

Error bounds provide: A. Exact solutions B. Estimates of maximum possible error C. Ways to eliminate all errors D. Random approximations (Linked to SLO 3.1)

3.2 Numerical Differentiation

Numerical differentiation is the process of approximating derivatives using discrete data points. Since exact derivatives require continuous functions, numerical methods are essential when dealing with tabulated data or functions that are too complex for analytical differentiation.

Fill in the blank: Approximating derivatives using discrete data points is called numerical _____.

3.2.1 Finite difference approximations

The most common approach to numerical differentiation is through **finite differences**, which estimate derivatives by calculating differences between successive data points.

Forward difference:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Backward difference:

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Central difference:

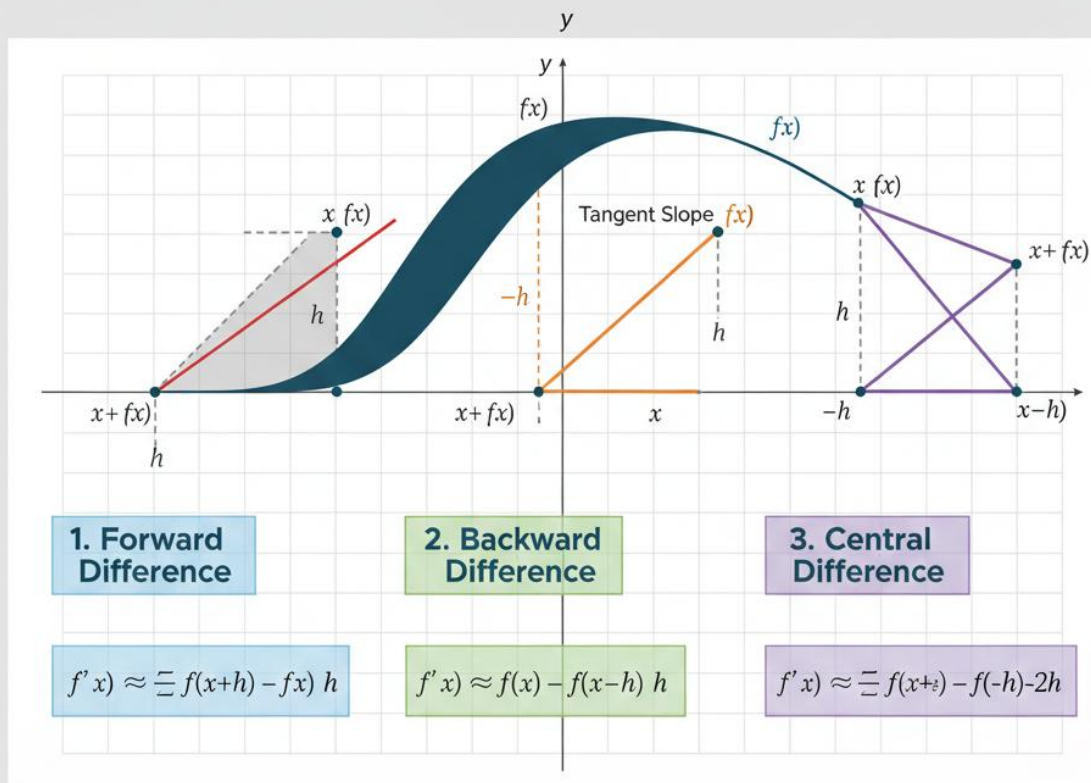
$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

The central difference is generally more accurate because it uses information from both sides of the point.



Finite Difference Approximations

Visualizing Forward, Backward, and Central Differences



Key Concept: Finite differences approximate instantaneous slope of the function using the slopes of secant. The Central Difference is generally the most accurate.



Open Educational Resource

Finite difference approximations for derivatives

Fill in the blank: The difference formula that uses points on both sides of the target value is called the _____ difference.

3.2.2 Higher-order derivatives



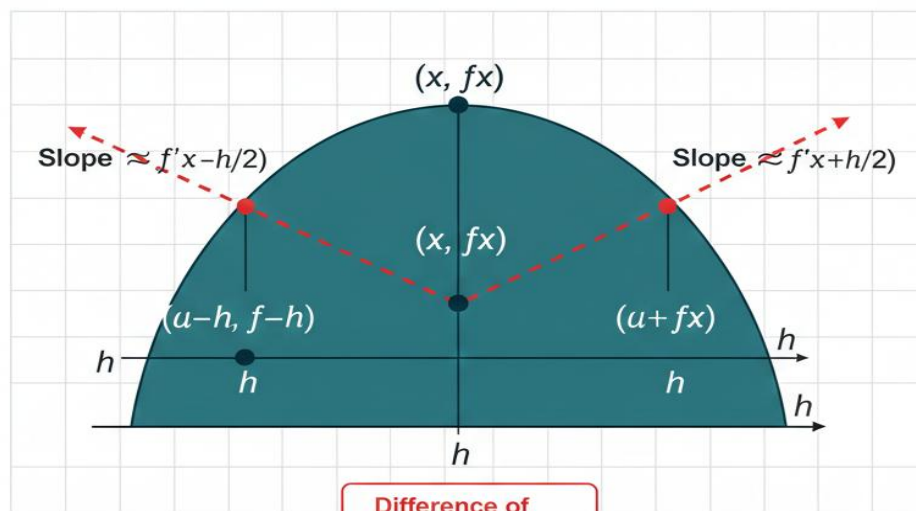
Finite difference methods can also be extended to approximate higher-order derivatives. For example, the second derivative can be approximated using:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

This formula is derived from combining forward and backward differences, and it is widely used in numerical simulations such as solving differential equations.

Second Derivative Approximation

Using the Central Finite Difference Formula



1. Forward Difference for $f'(x)$ at x :

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

2. Backward Difference for $f'(x)$ at x :

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Central Second Derivative:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



Key Concept: The second derivative measures the rate of change of the slope. The central difference formula is derived from the difference between two first-order differences to approximate the curvature at a point.

Open Educational Resource

Approximation of second derivative using finite differences

3.2.3 Error analysis in differentiation

Numerical differentiation introduces errors due to truncation and rounding.



Truncation error occurs because finite difference formulas are derived from Taylor series expansions that are cut short.

Rounding error arises from limited precision in computer arithmetic.

Balancing these errors is crucial: using a very small step size reduces truncation error but may increase rounding error.

Sources of error in numerical differentiation Short description: A box listing truncation and rounding errors with explanations.

The Two Primary Errors

Error Type	Definition	Behavior with Step Size (h)
Truncation Error	The error resulting from neglecting higher-order terms in the Taylor series expansion to create a finite formula.	Decreases as $h \rightarrow 0$. Smaller steps make the linear approximation closer to the actual curve.
Rounding Error	The error caused by the computer's finite precision (floating-point arithmetic) when performing subtractions and divisions.	Increases as $h \rightarrow 0$. When $f(x+h)$ and $f(x)$ are nearly identical, subtracting them causes "loss of significance."

Fill in the blank: Using a very small step size reduces truncation error but may increase _____ error.

Pilot Questions 3.2

The forward difference formula for the first derivative is: A. $f(x+h)-f(x)h$ B. $f(x)-f(x-h)h$ C. $f(x+h)-f(x-h)2h$ D. $f(x+h)-2f(x)+f(x-h)h^2$ (Linked to SLO 3.2)

The difference formula that uses points on both sides of the target value is: A. Forward difference B. Backward difference C. Central difference D. Second difference (Linked to SLO 3.2)

The second derivative can be approximated using: A. $f(x+h)-f(x)h$ B. $f(x)-f(x-h)h$ C. $f(x+h)-f(x-h)2h$ D. $f(x+h)-2f(x)+f(x-h)h^2$ (Linked to SLO 3.2)

Which type of error arises because finite difference formulas are derived from truncated Taylor series expansions? A. Rounding error B. Truncation error C. Propagation error D. Modelling error (Linked to SLO 3.2)

Using a very small step size reduces truncation error but may increase: A. Propagation error B. Rounding error C. Modelling error D. Approximation error (Linked to SLO 3.2)



3.3 Numerical Integration

Numerical integration is the process of approximating definite integrals when exact analytical solutions are difficult or impossible to obtain. It is especially useful when dealing with tabulated data or functions that are too complex to integrate symbolically.

Fill in the blank: Approximating definite integrals using numerical methods is called numerical _____.

3.3.1 Trapezoidal rule

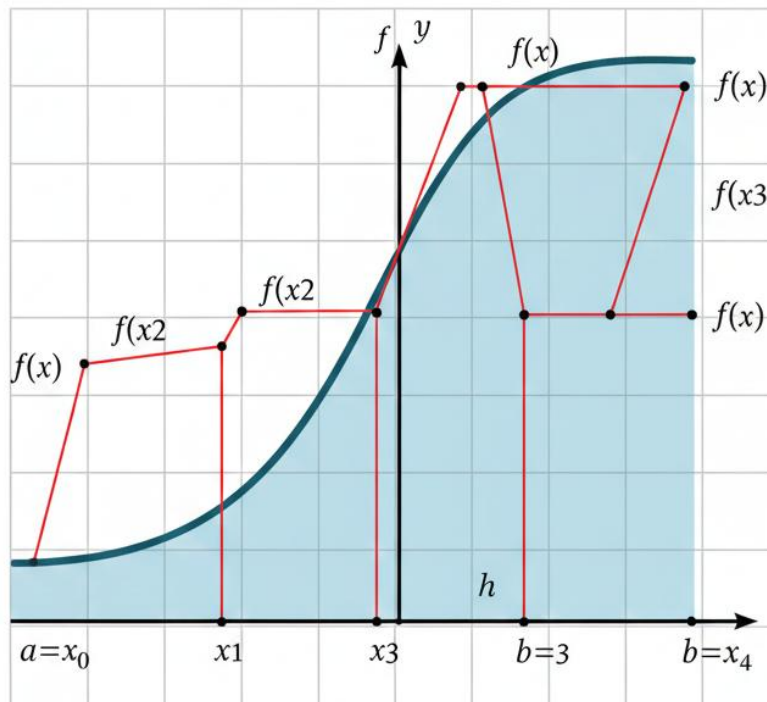
The **trapezoidal rule** approximates the area under a curve by dividing it into trapezoids. For an interval b , the formula is:

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

where $h = \frac{b-a}{n}$ and x_i are the partition points.



Trapezoidal Rule for Numerical Integration Approximating Area Under a Curve



$$\text{Area} \approx h \frac{2}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + f(x_4)]$$



Key Concept: The Trapezoidal Rule approximates the integral of function by summing summing trapezoids under the curve. It is first-order method, with error decreases proportionally to first step size ($O(h^2)$).

Open Educational Resource

Trapezoidal rule approximation

Fill in the blank: The trapezoidal rule approximates the area under a curve by dividing it into _____.

3.3.2 Simpson's rule

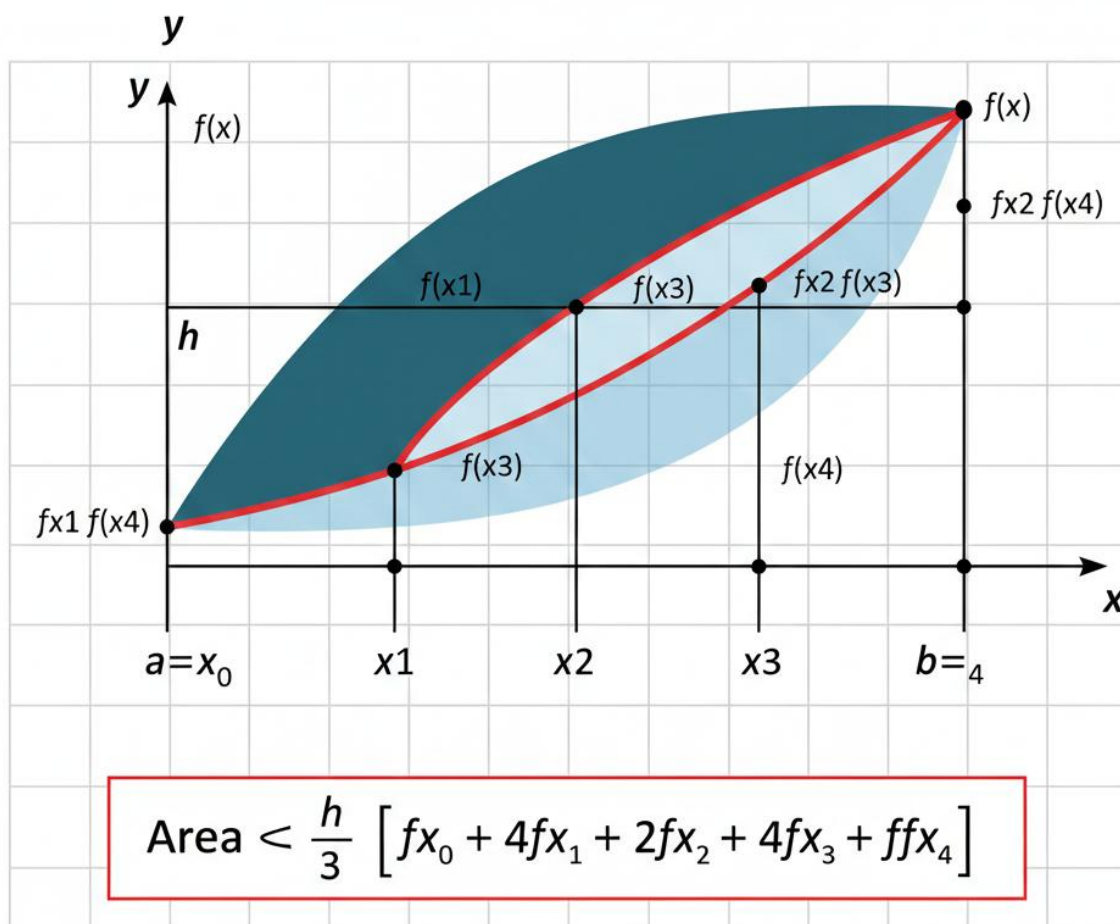


The **Simpson's rule** provides a more accurate approximation by fitting parabolic arcs to the curve. For an interval divided into an even number of subintervals:

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

This method is generally more accurate than the trapezoidal rule for smooth functions.

Simpson 1/3 Rule



Key Concept: Simpson's 1/3 Rule approximates the integral by fitting parabolic arcs to sets of three consecutive points. It is a more accurate, second-order method for approximating the area.

Open Educational Resource

Diagram showing parabolic arcs fitting the curve. Caption: Figure 3.5: Simpson's rule approximation



Fill in the blank: Simpson's rule approximates the curve using _____ arcs.

3.3.3 Midpoint rule

The **midpoint rule** estimates the integral by evaluating the function at the midpoint of each subinterval and multiplying by the width.

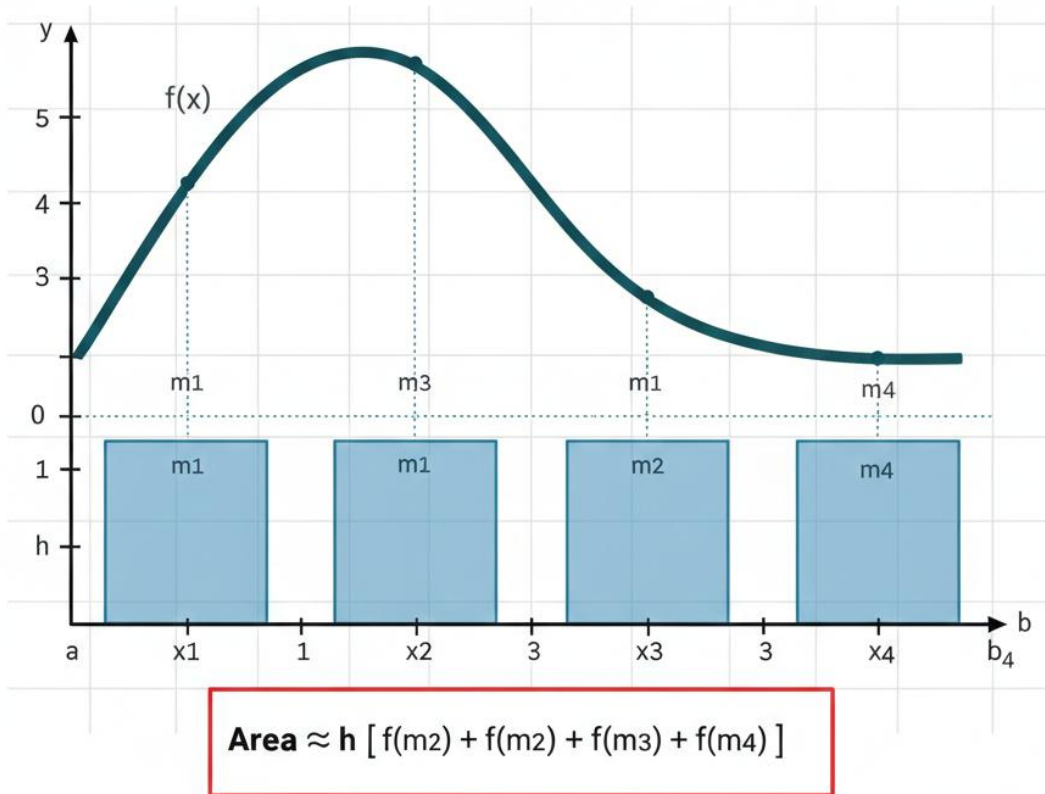
$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} h_i f(x_i + \frac{1}{2}h_i)$$

This method is simple and often provides better accuracy than the trapezoidal rule for certain functions.



Midpoint Rule for Numerical Integration

Approximating Area Under a Curve Using Centred Rectangles



Key Concept:

The Midpoint Rule approximates the integral of a function by summing the areas of rectangles centered at the midpoint of each subinterval. It is more accurate than the Trapezoidal Rule for the same number of intervals and a first-order method.

Open Educational Resource

Midpoint rule approximation

Fill in the blank: The midpoint rule estimates the integral by evaluating the function at the _____ of each subinterval.

3.3.4 Error bounds in integration

Each numerical integration method has an associated error bound:



Trapezoidal rule error depends on the second derivative of the function.

Simpson's rule error depends on the fourth derivative of the function.

Midpoint rule error also depends on the second derivative.

Error bounds help us estimate the maximum possible error and choose the most appropriate method for a given problem.

[Insert Box here] Caption: Box 3.3: Error bounds in numerical integration Short description: A box listing error bounds for trapezoidal, Simpson's, and midpoint rules. AI prompt: "Generate a text box listing error bounds for trapezoidal, Simpson's, and midpoint rules with brief explanations." Search string: "error bounds trapezoidal Simpson midpoint rule OER"

Error Bounds in Numerical Integration

In numerical integration, the **error bound** provides the maximum possible difference between the exact integral and our numerical approximation. Understanding these bounds helps determine how small the step size h needs to be to achieve a specific level of accuracy.

Comparison of Error Formulas

Method	Error Bound Formula ($ E $)	Explanation
--------	-------------------------------	-------------

Midpoint Rule	$\frac{(b-a)h^2}{24} f''(\xi) $	Uses the value at the center of the interval. Generally twice as accurate as the Trapezoidal rule for the same h .
Trapezoidal Rule	$\frac{(b-a)h^2}{12} f''(\xi) $	Uses a linear fit between endpoints. The error depends on the concavity (second derivative) of the function.
Simpson's 1/3 Rule	$\frac{(b-a)h^4}{180} f^{(4)}(\xi) $	Uses parabolic arcs. Because the error is proportional to h^4 , it converges much faster than the other two.



Key Performance Insights

- **The Power of h :** For Simpson's Rule, halving the step size ($h/2$) reduces the error by a factor of $2^4 = 16$. For the Trapezoidal and Midpoint rules, it only reduces the error by a factor of $2^2 = 4$.
- **The ξ Variable:** In these formulas, ξ is some unknown point in the interval $[a, b]$. In practice, we use the **maximum absolute value** of the required derivative over the interval to calculate the "worst-case" error.
- **Exact Results:**
 - Trapezoidal and Midpoint rules are **exact** for linear functions (where $f'' = 0$).
 - Simpson's Rule is surprisingly **exact** for all polynomials up to degree 3 (cubic), even though it only uses quadratic fits.

Fill in the blank: Simpson's rule error depends on the _____ derivative of the function.

Pilot Questions 3.3

The trapezoidal rule approximates the area under a curve by dividing it into: A. Rectangles B. Trapezoids C. Circles D. Parabolas (Linked to SLO 3.3)

Simpson's rule approximates the curve using: A. Straight lines B. Parabolic arcs C. Cubic splines D. Finite differences (Linked to SLO 3.3)

The midpoint rule estimates the integral by evaluating the function at: A. Endpoints of subintervals B. Midpoints of subintervals C. Random points D. Maximum points (Linked to SLO 3.3)

The trapezoidal rule error depends on the: A. First derivative B. Second derivative C. Third derivative D. Fourth derivative (Linked to SLO 3.3)

Simpson's rule error depends on the: A. First derivative B. Second derivative C. Third derivative D. Fourth derivative (Linked to SLO 3.3)

3.4 Applications of Numerical Differentiation and Integration

Numerical differentiation and integration are not just theoretical tools—they have wide-ranging applications in science, engineering, and computational fields. Their importance lies in the ability to approximate values when exact analytical solutions are unavailable or impractical.



Fill in the blank: Numerical differentiation and integration are widely applied in _____ and engineering.

3.4.1 Applications in engineering and physics

Engineering: Used to calculate stresses, loads, and material properties from experimental data. For example, numerical integration helps determine the total load on a beam from distributed forces.

Physics: Applied to compute displacement from velocity data (integration) or acceleration from displacement data (differentiation).

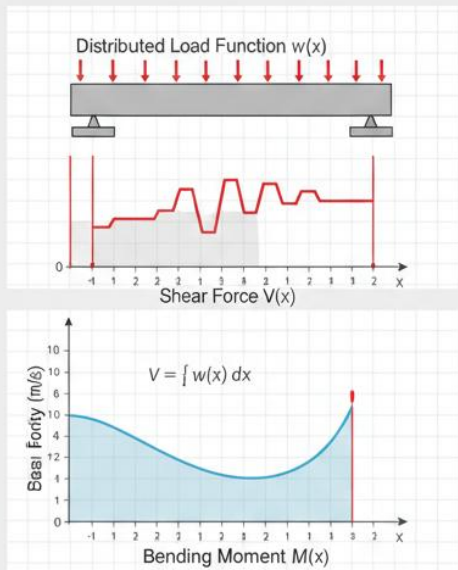
Environmental science: Used to estimate pollutant concentrations over time or energy balances in ecosystems.



Applications Numerical Integration in Engineering & Physics

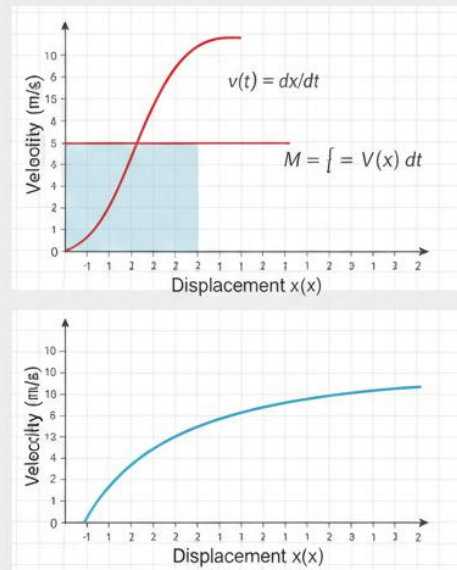
Solving Real-World Problems with Calculus

1. Beam Deflection from Load Data



Key Concept: Numerical integration finds cumulative effect of the load to determine internal forces.

2. Displacement from Velocity Data



Key Concept: Numerical integration finds a total distance traveled from velocity data time.

Key Concept: Numerical integration (e.g. Trapezoidal, Simpson's Rule) is crucial for summing up quantities over continuous, allowing to calculate force, work, and displacement when the data is discrete.

Open Educational Resource

Applications of numerical methods in engineering and physics

Fill in the blank: In physics, numerical integration can be used to compute _____ from velocity data.

3.4.2 Computational efficiency considerations

Numerical methods must balance accuracy with computational efficiency.

Efficiency: Choosing methods that provide reliable results without excessive computation.



Trade-offs: Simpson's rule is more accurate but requires more function evaluations than the trapezoidal rule.

Practicality: In large-scale simulations, efficiency often determines which method is chosen.

Efficiency trade-offs in numerical integration Short description: A box comparing trapezoidal, Simpson's, and midpoint rules in terms of accuracy and computational cost.

Performance Matrix

Method	Order of Accuracy	Computational Effort	Best Use Case
Midpoint Rule	$O(h^2)$	Low: 1 function evaluation per interval.	Best for fast approximations of smooth, symmetric functions.
Trapezoidal Rule	$O(h^2)$	Low: Uses endpoint values; very efficient for periodic functions.	Discrete data sets where only endpoints are known.
Simpson's 1/3 Rule	$O(h^4)$	Moderate: Requires 3 points per parabolic arc (midpoint + 2 endpoints).	High-precision requirements for smooth, continuous functions.

Fill in the blank: Simpson's rule is generally more accurate but requires more _____ evaluations.

3.4.3 Stability and accuracy in practice

In real-world applications, stability and accuracy are critical.

Stability: Ensures that errors do not grow uncontrollably during computations.

Accuracy: Determines how close the numerical result is to the true value.

Examples:

Weather forecasting models rely on stable numerical differentiation and integration.

Structural engineering simulations require accurate integration to predict stresses and strains.

Examples of stability and accuracy in practice Short description:



A table listing applications (weather forecasting, structural engineering, fluid dynamics) with notes on stability and accuracy requirements.

Field	Task	Method Type	Stability & Accuracy Notes
Kinematics	Finding acceleration from high-speed camera velocity data.	Differentiation	High Sensitivity: Noise in displacement data is amplified during differentiation. Requires smoothing or stable central differences.
Structural Engineering	Calculating total shear force and bending moments on a bridge beam.	Integration	Stable: Integration is a smoothing process. Trapezoidal rule is often sufficient, but accuracy is vital for safety factors.
Fluid Dynamics	Determining flow rates through a pipe using velocity profile sensors.	Integration	Accuracy Dependent: Simpson's rule is preferred to capture the parabolic nature of laminar flow profiles accurately.
Signal Processing	Edge detection in digital imaging (finding where intensity changes rapidly).	Differentiation	Unstable with Noise: High-frequency noise can create "ghost edges." Low-pass filters are usually



Field	Task	Method Type	Stability & Accuracy Notes
			applied before taking derivatives.
Thermodynamics	Calculating work done ($W = \int P \, dV$) during a non-ideal gas expansion.	Integration	Generally Robust: Usually involves discrete experimental P - V data points; requires stable handling of experimental error.

Fill in the blank: Weather forecasting models rely on stable numerical _____ and integration.

Pilot Questions 3.4

In physics, numerical integration can be used to compute: A. Velocity from displacement data B. Displacement from velocity data C. Acceleration from velocity data D. Stress from strain data (Linked to SLO 3.4)

Simpson's rule is generally more accurate but requires more: A. Function evaluations B. Derivatives C. Tables D. Approximations (Linked to SLO 3.4)

Stability in numerical methods ensures that: A. Errors grow uncontrollably B. Errors remain bounded C. Errors are ignored D. Errors are eliminated completely (Linked to SLO 3.4)

Which of the following applications relies heavily on stable numerical methods? A. Weather forecasting B. Solving quadratic equations C. Drawing graphs D. Basic arithmetic (Linked to SLO 3.4)

Accuracy in numerical methods determines: A. How close the numerical result is to the true value B. How fast the computation runs C. How many steps are required D. How errors propagate (Linked to SLO 3.4)



Series Summary

In this Series, we explored the principles and techniques of **numerical differentiation and integration**, which are fundamental tools in numerical analysis. We began with an introduction to the motivation and importance of these methods, highlighting their role in physics, engineering, and computational sciences. We then studied numerical differentiation, focusing on finite difference approximations (forward, backward, and central differences), higher-order derivatives, and error analysis. Next, we examined numerical integration techniques, including the trapezoidal rule, Simpson's rule, and the midpoint rule, along with their associated error bounds. Finally, we discussed practical applications of these methods in engineering and physics, computational efficiency considerations, and the importance of stability and accuracy in real-world contexts.

By completing this Series, you should now be able to explain why numerical differentiation and integration are necessary, apply finite difference formulas to approximate derivatives, use trapezoidal, Simpson's, and midpoint rules to approximate integrals, and evaluate the accuracy, stability, and efficiency of these methods in practice. These abilities directly reflect the Series Learning Outcomes (SLOs) and equip you with practical skills for handling problems where analytical solutions are not feasible.

Glossary of Terms

Numerical differentiation: Approximating derivatives using discrete data points.

Numerical integration: Approximating definite integrals using numerical methods.

Finite differences: Approximations of derivatives using differences between successive data points.

Forward difference: A finite difference formula using the current and next data point.

Backward difference: A finite difference formula using the current and previous data point.

Central difference: A finite difference formula using points on both sides of the target value.

Trapezoidal rule: A numerical integration method that approximates the area under a curve by dividing it into trapezoids.



Simpson's rule: A numerical integration method that approximates the curve using parabolic arcs.

Midpoint rule: A numerical integration method that estimates the integral by evaluating the function at the midpoint of each subinterval.

Error bounds: Estimates of the maximum possible error in numerical differentiation or integration.

Stability: A property of numerical methods where errors do not grow uncontrollably during computation.

Accuracy: The closeness of a numerical result to the true value.

Concept Check Questions 3

Objective questions

Approximating derivatives using discrete data points is called: A. Numerical integration B. Numerical differentiation C. Curve fitting D. Error analysis (Linked to SLO 3.1)

The central difference formula for the first derivative is: A. $f(x+h)-f(x)h$ B. $f(x)-f(x-h)h$ C. $f(x+h)-f(x-h)2h$ D. $f(x+h)-2f(x)+f(x-h)h^2$ (Linked to SLO 3.2)

The trapezoidal rule approximates the area under a curve by dividing it into: A. Rectangles B. Trapezoids C. Circles D. Parabolas (Linked to SLO 3.3)

Simpson's rule error depends on the: A. First derivative B. Second derivative C. Third derivative D. Fourth derivative (Linked to SLO 3.3)

Stability in numerical methods ensures that: A. Errors grow uncontrollably B. Errors remain bounded C. Errors are ignored D. Errors are eliminated completely (Linked to SLO 3.4)

Short-answer questions

Define numerical integration and explain why it is important. (Linked to SLO 3.3)

State the forward, backward, and central difference formulas for the first derivative. (Linked to SLO 3.2)

Identify two real-world applications of numerical differentiation and integration. (Linked to SLO 3.4)

Long-answer questions



Compare the trapezoidal rule, Simpson's rule, and midpoint rule in terms of accuracy and computational efficiency. (Linked to SLO 3.3)

Basic response: Describe how each rule approximates the integral.

Value-added response: Analyse their relative accuracy, error bounds, and efficiency trade-offs.

Discuss the importance of stability and accuracy in numerical methods with examples. (Linked to SLO 3.4)

Basic response: Explain what stability and accuracy mean in numerical computations.

Value-added response: Evaluate their role in applications such as weather forecasting and structural engineering.

Answers to Concept Check Questions 3

B. Numerical differentiation

C. $f(x+h)-f(x-h)2h$

B. Trapezoids

D. Fourth derivative

B. Errors remain bounded

Numerical integration is the process of approximating definite integrals using numerical methods. It is important because many functions cannot be integrated analytically, especially when dealing with experimental or tabulated data.

Forward difference: $f(x+h)-f(x)h$; Backward difference: $f(x)-f(x-h)h$; Central difference: $f(x+h)-f(x-h)2h$.

Applications include calculating displacement from velocity data in physics and determining stresses in engineering structures.

Basic response: The trapezoidal rule uses trapezoids, Simpson's rule uses parabolic arcs, and the midpoint rule uses midpoints of subintervals.

Value-added response: Simpson's rule is generally the most accurate but requires more function evaluations. The trapezoidal rule is simpler but less accurate, while the midpoint rule can outperform trapezoidal in certain cases. Error bounds depend on derivatives of the function.



Basic response: Stability ensures errors do not grow uncontrollably, and accuracy measures closeness to the true value.

Value-added response: In weather forecasting, stable methods prevent small errors from escalating into unreliable predictions. In structural engineering, accurate integration ensures correct stress and strain calculations, preventing unsafe designs.

Series 4: Numerical Solutions of Ordinary Differential Equations (ODEs)

Series Outline

Part 4.1 Introduction to Numerical Solutions of ODEs

Motivation and importance

Basic concepts and definitions

Part 4.2 Euler's Method

Derivation and formula

Step-by-step application

Error analysis

Part 4.3 Improved Methods (Heun's Method and Runge-Kutta Methods)

Heun's method (modified Euler)

Runge-Kutta methods (focus on RK4)

Accuracy and efficiency comparisons

Part 4.4 Applications of Numerical ODE Solutions

Physics (motion, population dynamics)

Engineering (heat transfer, circuits)

Computational considerations (stability, step size control)

Introduction

Ordinary Differential Equations (ODEs) are fundamental in modeling real-world phenomena across physics, engineering, biology, and economics. They describe



how quantities change over time, such as population growth, motion of objects, or heat transfer. However, many ODEs cannot be solved exactly using analytical methods, especially when they are nonlinear or complex. This is where **numerical methods** become essential.

Numerical solutions of ODEs provide approximate values of the unknown function at discrete points, allowing us to simulate and predict system behavior. These methods are widely used in computational sciences, where simulations of dynamic systems are critical.

In this Series, we will begin with an introduction to the motivation and importance of numerical solutions of ODEs. We will then study **Euler's method**, the simplest numerical approach, along with its error analysis. Next, we will explore improved methods such as **Heun's method** and the **Runge-Kutta methods**, focusing on the widely used fourth-order Runge-Kutta (RK4). Finally, we will examine applications of numerical ODE solutions in physics, engineering, and computational modeling, highlighting issues of stability, accuracy, and step size control.

Series Learning Outcomes

Upon completing this Series, you should be able to:

SLO 4.1 explain the motivation and importance of numerical solutions of ordinary differential equations (ODEs) in applied mathematics and computational sciences,

SLO 4.2 apply Euler's method to approximate solutions of first-order ODEs and analyse its error behaviour,

SLO 4.3 use improved numerical methods such as Heun's method and Runge-Kutta methods (with emphasis on RK4) to obtain more accurate solutions of ODEs,

SLO 4.4 evaluate the accuracy, stability, and efficiency of numerical ODE methods in practical applications across physics, engineering, and computational modelling.

4.1 Introduction to Numerical Solutions of ODEs



Ordinary Differential Equations (ODEs) describe how quantities change over time and are central to modeling dynamic systems in science and engineering. While some ODEs have exact analytical solutions, many real-world problems involve nonlinear or complex equations that cannot be solved symbolically. In such cases, **numerical methods** provide approximate solutions at discrete points, enabling us to simulate and predict system behavior.

Fill in the blank: Ordinary Differential Equations (ODEs) describe how _____ change over time.

4.1.1 Motivation and importance

Numerical solutions of ODEs are motivated by practical needs:

Physics: Modeling motion, heat transfer, and wave propagation.

Biology: Simulating population dynamics or disease spread.

Engineering: Analyzing circuits, fluid flow, and structural vibrations.

Economics: Predicting growth models and financial systems.

These applications highlight why numerical ODE methods are indispensable in computational sciences.

Motivation for numerical solutions of ODEs

Fill in the blank: In biology, numerical ODE solutions are used to simulate _____ dynamics.

4.1.2 Basic concepts and definitions

Initial value problem (IVP): An ODE with specified initial conditions, e.g., $y'(x)=f(x,y)$, $y(x_0)=y_0$.

Step size (h): The interval between successive approximations in numerical methods.

Local error: The error introduced in a single step of the method.

Global error: The cumulative error over multiple steps.



Stability: The property of a numerical method that prevents errors from growing uncontrollably.

A box listing IVP, step size, local error, global error, and stability with definitions.

Concept	Definition
Initial Value Problem (IVP)	An ODE accompanied by a specified value (initial condition) of the unknown function at a given point in the domain. It tells the solver where to "start."
Step Size (h)	The discrete increment of the independent variable (usually time) between successive calculation points.
Local Truncation Error (LTE)	The error introduced during a single step of the numerical method, assuming the previous value was perfectly accurate.
Global Truncation Error (GTE)	The cumulative error at the end of many steps. It accounts for how local errors propagate and grow throughout the entire simulation.
Stability	A method's ability to keep errors from growing exponentially. A stable method ensures that small perturbations in the data or rounding errors don't "blow up" the solution.
Order of Accuracy	Defined as $O(h^n)$. If a method is second-order ($n=2$), halving the step size reduces the global error by approximately a factor of 4.

Fill in the blank: An ODE with specified initial conditions is called an _____ value problem.

Pilot Questions 4.1



Ordinary Differential Equations (ODEs) describe how: A. Numbers change randomly B. Quantities change over time C. Errors accumulate D. Functions are approximated (Linked to SLO 4.1)

In physics, numerical ODE solutions can be used to model: A. Population growth B. Heat transfer C. Financial systems D. All of the above (Linked to SLO 4.1)

An ODE with specified initial conditions is called: A. Boundary value problem B. Initial value problem C. Stability problem D. Approximation problem (Linked to SLO 4.1)

The interval between successive approximations in numerical methods is called: A. Step size B. Local error C. Global error D. Stability (Linked to SLO 4.1)

4.2 Euler's Method

Euler's method is the simplest numerical technique for solving ordinary differential equations (ODEs). It provides an approximate solution by stepping forward from an initial condition using the slope of the function. Although not highly accurate, it is easy to understand and forms the foundation for more advanced methods.

Fill in the blank: Euler's method approximates the solution of an ODE by stepping forward using the _____ of the function.

4.2.1 Derivation and formula

Consider the initial value problem:

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

Euler's method uses the formula:

$$y_{n+1}=y_n+hf(x_n,y_n)$$

where h is the step size. This formula is derived from the tangent line approximation of the solution curve.

Euler's method using tangent line approximation

Fill in the blank: Euler's method formula is $y_{n+1} = y_n + h \cdot$ _____.



4.2.2 Step-by-step application

Steps for applying Euler's method:

Start with the initial condition y_0 .

Choose a step size h .

Compute the slope $f(x_n, y_n)$.

Update the solution using $y_{n+1} = y_n + hf(x_n, y_n)$.

Repeat for successive steps until the desired interval is covered.

Step-by-step application of Euler's method

step in Euler's method is to start with the _____ condition.

4.2.3 Error analysis

Euler's method introduces errors because it approximates the solution curve by straight lines.

Local truncation error: Proportional to h^2 .

Global error: Proportional to h .

Accuracy improves with smaller step sizes, but computational cost increases.

Error characteristics of Euler's method Short description: A box listing local truncation error, global error, and effect of step size.

Error Type	Definition	Order of Magnitude
Local Truncation Error (LTE)	The error incurred in a single step . It represents the difference between the true solution and the numerical approximation after one increment h .	$O(h^2)$



Error Type	Definition	Order of Magnitude
Global Truncation Error (GTE)	The cumulative error over the entire interval of interest. It is the sum of all local errors plus the effect of their propagation.	$O(h)$
Rounding Error	Discrepancies caused by the limited precision of floating-point arithmetic. If h is made extremely small to reduce truncation, rounding error begins to dominate.	N/A (Machine dependent)

Fill in the blank: The global error in Euler's method is proportional to _____.

Pilot Questions 4.2

Euler's method formula is: A. $y_{n+1} = y_n + hf(x_n, y_n)$ B. $y_{n+1} = y_n - hf(x_n, y_n)$ C. $y_{n+1} = y_n + h^2f(x_n, y_n)$ D. $y_{n+1} = y_n + f(x_n, y_n)$ (Linked to SLO 4.2)

The first step in Euler's method is to: A. Choose a step size B. Compute the slope C. Start with the initial condition D. Repeat the formula (Linked to SLO 4.2)

Euler's method approximates the solution curve using: A. Parabolic arcs B. Straight lines C. Cubic splines D. Exact formulas (Linked to SLO 4.2)

The local truncation error in Euler's method is proportional to: A. h B. h^2 C. h^3 D. h^4 (Linked to SLO 4.2)

4.3 Improved Methods (Heun's Method and Runge-Kutta Methods)

Euler's method is simple but not very accurate. To improve precision without drastically reducing step size, mathematicians developed **Heun's method** (also called the modified Euler method) and the family of **Runge-Kutta methods**,



especially the widely used **fourth-order Runge-Kutta (RK4)**. These methods balance accuracy and efficiency, making them practical for real-world applications.

Fill in the blank: Heun's method is also known as the _____ Euler method.

4.3.1 Heun's Method (Modified Euler)

Heun's method improves Euler's approach by averaging slopes:

Compute the slope at the beginning: $k_1 = f(x_n, y_n)$.

Predict the next value using Euler's method: $y^* = y_n + h k_1$.

Compute the slope at the predicted point: $k_2 = f(x_n + 1, y^*)$.

Average the slopes:

$$y_{n+1} = y_n + h(k_1 + k_2)$$

This reduces error significantly compared to Euler's method.

Heun's method slope averaging ”

Fill in the blank: Heun's method improves accuracy by averaging the _____ at the beginning and end of the interval.

4.3.2 Runge-Kutta Methods

Runge-Kutta methods generalize the idea of slope averaging. The most popular is the **fourth-order Runge-Kutta (RK4)**, which achieves high accuracy with moderate computational effort.

For RK4:

$$k_1 = f(x_n, y_n) \quad k_2 = f(x_n + h/2, y_n + h/2 k_1) \quad k_3 = f(x_n + h/2, y_n + h k_2) \quad k_4 = f(x_n + h, y_n + h k_3) \\ y_{n+1} = y_n + h(k_1 + 2k_2 + 2k_3 + k_4)$$

RK4 is widely used because it provides excellent accuracy without requiring extremely small step sizes.

Runge-Kutta 4th order method

Fill in the blank: The RK4 method uses _____ slope evaluations per step.



4.3.3 Accuracy and efficiency comparisons

Euler's method: Simple, but low accuracy.

Heun's method: More accurate than Euler, requires two slope evaluations per step.

RK4 method: Highly accurate, requires four slope evaluations per step, widely used in practice.

Trade-off: More slope evaluations increase accuracy but also computational cost.

Comparison of Euler, Heun, and RK4 methods Short description: A box comparing accuracy and computational cost of Euler, Heun, and RK4 methods.

Method Performance Matrix

Method	Order (Global Error)	Slopes per Step	Computational Cost	Best Use Case
Euler	$O(h)$	1 (k_1)	Very Low	Quick-and-dirty estimates or teaching the basics.
Heun	$O(h^2)$	2 (k_1, k_2)	Moderate	Simple real-time systems where $O(h)$ is too inaccurate.
RK4	$O(h^4)$	4 ($k_1 \dots k_4$)	High (per step)	Industry standard for non-stiff engineering simulations.

Fill in the blank: RK4 is widely used because it provides excellent _____ without requiring very small step sizes.

Pilot Questions 4.3

Heun's method is also known as the: A. Central difference method B. Modified Euler method C. Runge-Kutta method D. Trapezoidal method (Linked to SLO 4.3)

Heun's method improves accuracy by: A. Using parabolic arcs B. Averaging slopes at the beginning and end of the interval C. Reducing step size D. Ignoring errors (Linked to SLO 4.3)



The RK4 method uses how many slope evaluations per step? A. 1 B. 2 C. 3 D. 4
(Linked to SLO 4.3)

Which method is most widely used in practice due to its balance of accuracy and efficiency? A. Euler's method B. Heun's method C. RK4 method D. Midpoint method (Linked to SLO 4.3)

Increasing the number of slope evaluations in Runge-Kutta methods improves:
A. Efficiency only B. Accuracy only C. Both accuracy and computational cost D. Neither accuracy nor efficiency (Linked to SLO 4.3)

4.4 Applications of Numerical ODE Solutions

Numerical methods for solving ODEs are indispensable in real-world contexts where exact solutions are unavailable. They allow us to simulate dynamic systems, predict future behavior, and design engineering solutions with confidence.

Fill in the blank: Numerical ODE methods are widely applied in _____, biology, and engineering.

4.4.1 Applications in physics and biology

Physics:

Modeling motion under varying forces (e.g., projectile motion with air resistance).

Simulating heat transfer in materials.

Analyzing oscillations in mechanical systems.

Biology:

Modeling population dynamics (e.g., predator-prey systems).

Simulating disease spread using epidemiological models.

Studying biochemical reactions in cells.

Applications of numerical ODE solutions in physics and biology

Fill in the blank: In epidemiology, numerical ODE solutions are used to simulate the spread of _____.

4.4.2 Applications in engineering



Electrical engineering: Circuit analysis using differential equations for current and voltage.

Mechanical engineering: Modeling vibrations and stresses in structures.

Chemical engineering: Simulating reaction kinetics and diffusion processes.

Civil engineering: Predicting structural responses under dynamic loads.

Applications of numerical ODE solutions in engineering

Fill in the blank: In electrical engineering, ODEs are used to analyze _____ and voltage in circuits.

4.4.3 Computational considerations

Numerical ODE methods must balance accuracy, stability, and efficiency:

Accuracy: RK4 provides high accuracy but requires more computations.

Stability: Step size must be chosen carefully to prevent error growth.

Efficiency: In large-scale simulations, computational cost is a major factor.

Examples:

Weather forecasting models rely on stable ODE solvers.

Engineering simulations require efficient methods to handle large systems of equations.

A box listing accuracy, stability, and efficiency with examples.

Consideration	Goal	Risk of Failure
Accuracy	Match the "True" curve.	Results are smooth but mathematically wrong.
Stability	Keep the math from "exploding."	Results oscillate wildly or reach ∞ instantly.
Efficiency	Get results quickly.	Simulation takes too long to be useful.



Fill in the blank: Weather forecasting models rely on stable numerical _____ solvers.

Pilot Questions 4.4

In epidemiology, numerical ODE solutions are used to simulate: A. Population growth B. Disease spread C. Circuit analysis D. Heat transfer (Linked to SLO 4.4)

In electrical engineering, ODEs are used to analyze: A. Stress and strain B. Current and voltage in circuits C. Population dynamics D. Biochemical reactions (Linked to SLO 4.4)

Which of the following applications relies heavily on stable ODE solvers? A. Weather forecasting B. Solving quadratic equations C. Drawing graphs D. Basic arithmetic (Linked to SLO 4.4)

RK4 provides high accuracy but requires more: A. Step size reductions B. Function evaluations C. Initial conditions D. Error bounds (Linked to SLO 4.4)

Series Summary

In this Series, we studied **numerical solutions of ordinary differential equations (ODEs)**, a cornerstone of numerical analysis for modeling dynamic systems. We began with an introduction to the motivation and importance of numerical ODE methods, highlighting their role in physics, biology, engineering, and economics. We then explored **Euler's method**, the simplest numerical approach, including its derivation, step-by-step application, and error analysis. Next, we examined improved methods such as **Heun's method** (modified Euler) and the **Runge-Kutta methods**, focusing on the widely used **fourth-order Runge-Kutta (RK4)**, which balances accuracy and efficiency. Finally, we discussed applications of numerical ODE solutions in physics, biology, and engineering, along with computational considerations such as accuracy, stability, and efficiency.

By completing this Series, you should now be able to explain why numerical ODE solutions are necessary, apply Euler's method to approximate solutions of first-order ODEs, use Heun's method and RK4 for more accurate results, and evaluate the accuracy, stability, and efficiency of these methods in real-world applications. These abilities directly reflect the Series Learning



Outcomes (SLOs) and equip you with practical skills for solving dynamic problems numerically.

Glossary of Terms

Ordinary Differential Equation (ODE): An equation involving derivatives of a function with respect to one variable.

Initial Value Problem (IVP): An ODE with specified initial conditions.

Step size (h): The interval between successive approximations in numerical methods.

Euler's method: A simple numerical method that approximates solutions using tangent line steps.

Heun's method (Modified Euler): An improved method that averages slopes at the beginning and end of the interval.

Runge-Kutta methods: A family of numerical methods that improve accuracy by evaluating slopes multiple times per step.

RK4 method: The fourth-order Runge-Kutta method, widely used for its balance of accuracy and efficiency.

Local error: Error introduced in a single step of the method.

Global error: Cumulative error over multiple steps.

Stability: The property of a numerical method that prevents errors from growing uncontrollably.

Accuracy: The closeness of a numerical result to the true solution.

Concept Check Questions 4

Objective questions

Ordinary Differential Equations (ODEs) describe how: A. Numbers change randomly B. Quantities change over time C. Errors accumulate D. Functions are approximated (Linked to SLO 4.1)

Euler's method formula is: A. $y_{n+1} = y_n + hf(x_n, y_n)$ B. $y_{n+1} = y_n - hf(x_n, y_n)$ C. $y_{n+1} = y_n + h^2f(x_n, y_n)$ D. $y_{n+1} = y_n + f(x_n, y_n)$ (Linked to SLO 4.2)



Heun's method is also known as the: A. Central difference method B. Modified Euler method C. Runge-Kutta method D. Trapezoidal method (Linked to SLO 4.3)

The RK4 method uses how many slope evaluations per step? A. 1 B. 2 C. 3 D. 4 (Linked to SLO 4.3)

In electrical engineering, ODEs are used to analyze: A. Stress and strain B. Current and voltage in circuits C. Population dynamics D. Biochemical reactions (Linked to SLO 4.4)

Short-answer questions

Define an initial value problem (IVP) and give an example. (Linked to SLO 4.1)

State the step-by-step procedure for Euler's method. (Linked to SLO 4.2)

Explain how Heun's method improves upon Euler's method. (Linked to SLO 4.3)

Long-answer questions

Compare Euler's method, Heun's method, and RK4 in terms of accuracy and computational efficiency. (Linked to SLO 4.3)

Basic response: Describe how each method approximates the solution.

Value-added response: Analyse their relative accuracy, error behaviour, and computational trade-offs.

Discuss the importance of stability and accuracy in numerical ODE solutions with real-world examples. (Linked to SLO 4.4)

Basic response: Explain what stability and accuracy mean in numerical computations.

Value-added response: Evaluate their role in applications such as weather forecasting, epidemiology, and engineering simulations.

Answers to Concept Check Questions 4

B. Quantities change over time

A. $y_{n+1} = y_n + hf(x_n, y_n)$

B. Modified Euler method

D. 4



B. Current and voltage in circuits

An initial value problem (IVP) is an ODE with specified initial conditions.

Example: $y'(x)=x+y$, $y(0)=1$.

Steps: (i) Start with initial condition y_0 ; (ii) Choose step size h ; (iii) Compute slope $f(x_n, y_n)$; (iv) Update solution using $y_{n+1}=y_n+hf(x_n, y_n)$; (v) Repeat until interval is covered.

Heun's method improves accuracy by averaging slopes at the beginning and end of the interval, reducing error compared to Euler's straight-line approximation.

Basic response: Euler's method uses tangent line steps, Heun's method averages two slopes, RK4 averages four slopes.

Value-added response: Euler's method is simple but inaccurate; Heun's method improves accuracy with two evaluations; RK4 achieves high accuracy with four evaluations, balancing computational cost and precision.

Basic response: Stability ensures errors remain bounded, accuracy ensures closeness to the true solution.

Value-added response: In weather forecasting, stable methods prevent error escalation; in epidemiology, accurate solutions predict disease spread reliably; in engineering, accurate and stable ODE solvers ensure safe structural designs

Answers to Pilot Questions – Series 4

Here's the compiled set of answers for all pilot questions from **Series 4: Numerical Solutions of Ordinary Differential Equations (ODEs)**.

Part 4.1 Introduction to Numerical Solutions of ODEs

B. Quantities change over time

D. All of the above

B. Initial value problem

A. Step size

B. Errors remain bounded



Part 4.2 Euler's Method

- A. $y_{n+1} = y_n + hf(x_n, y_n)$
- C. Start with the initial condition
- B. Straight lines
- B. h^2
- A. h

Part 4.3 Improved Methods (Heun's Method and Runge-Kutta Methods)

- B. Modified Euler method
- B. Averaging slopes at the beginning and end of the interval
- D. 4
- C. RK4 method
- C. Both accuracy and computational cost

Part 4.4 Applications of Numerical ODE Solutions

- B. Disease spread
- B. Current and voltage in circuits
- A. Weather forecasting
- B. Function evaluations
- B. It allows large-scale simulations to be feasible

Series 5: Numerical Solutions of Systems of Linear Equations

Series Outline

Part 5.1 Introduction to Numerical Solutions of Linear Systems

Motivation and importance

Basic concepts and definitions

Part 5.2 Direct Methods



Gaussian elimination

LU decomposition

Error considerations

Part 5.3 Iterative Methods

Jacobi method

Gauss-Seidel method

Convergence and stability

Part 5.4 Applications of Numerical Linear Systems Solutions

Engineering (structural analysis, circuit networks)

Computer science (matrix computations, graphics)

Physics (finite element methods, simulations)

Introduction

Systems of linear equations are among the most common mathematical problems encountered in applied sciences and engineering. They arise in diverse contexts such as structural analysis, electrical circuits, computer graphics, and economic modeling. While small systems can often be solved exactly using algebraic techniques, large or complex systems require **numerical methods** to obtain practical solutions efficiently.

Numerical solutions of linear systems are essential because:

They allow us to handle very large systems that are computationally infeasible to solve symbolically.

They provide stable and efficient approximations when exact arithmetic is impossible due to rounding errors.

They form the backbone of many computational algorithms, including finite element methods and optimization techniques.

In this Series, we will begin with an introduction to the motivation and importance of numerical solutions of linear systems. We will then study **direct methods** such as Gaussian elimination and LU decomposition, followed by **iterative methods** like the Jacobi and Gauss-Seidel methods. Finally, we will explore applications in



engineering, computer science, and physics, while considering issues of convergence, stability, and efficiency.

Series Learning Outcomes

Upon completing this Series, you should be able to:

SLO 5.1 explain the motivation and importance of numerical solutions of systems of linear equations in applied mathematics and computational sciences,

SLO 5.2 apply direct methods such as Gaussian elimination and LU decomposition to solve systems of linear equations, and analyse their error behaviour,

SLO 5.3 use iterative methods such as the Jacobi and Gauss-Seidel methods to obtain solutions of linear systems, and evaluate their convergence and stability,

SLO 5.4 assess the accuracy, efficiency, and practical applications of numerical methods for solving linear systems in engineering, computer science, and physics.

5.1 Introduction to Numerical Solutions of Linear Systems

Systems of linear equations are fundamental in mathematics and applied sciences. They consist of multiple linear equations involving several unknowns, and the goal is to find values of the variables that satisfy all equations simultaneously. While small systems can be solved exactly using algebraic methods, large systems—common in engineering and computational applications—require efficient **numerical methods**.

Fill in the blank: Systems of linear equations involve multiple linear equations with several _____.

5.1.1 Motivation and importance

Numerical solutions of linear systems are crucial because:

Engineering: Used in structural analysis, circuit networks, and fluid mechanics.

Physics: Essential in simulations such as finite element methods and quantum mechanics.



Computer science: Applied in graphics, machine learning, and optimization problems.

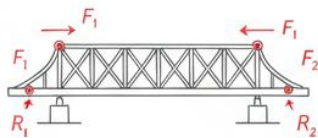
Economics: Used in input-output models and resource allocation.

These applications demonstrate why numerical methods are indispensable for solving large-scale systems efficiently.

Applications of Linear Systems: Engineering, Physics & Computer Science



1. Engineering: Structural Analysis

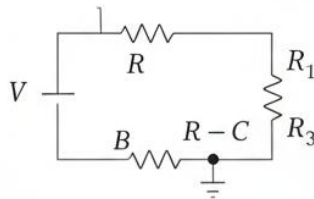


System of equations for joint forces:

System of equations: $Ax = b$, where A is the connectivity matrix, x has the tensions, and b has external forces.

Models forces in static structures.

2. Physics: Circuit Analysis (KCL/KVL)

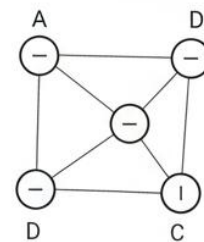


Applying Kirchhoff's Laws:

$$\begin{aligned} R_1 I_1 - R_{22} &= V \\ R_2 I_2 + R_3 &= 0 \\ I_1 - I_2 - I_3 &= 0 \end{aligned}$$

Solves for unknown currents/voltages

3. Computer Science: Google PageRank



Eigenvalue problem:

$$A^T p = p = \lambda_A C = p$$

where A is the link matrix, and p is the PageRank vector.

Ranks importance of web pages.



Key Concept: Linear systems $Ax = b$ and eigenvalue problems $Ax = \lambda x$ are fundamental to modeling interconnected systems and data dependencies in countless real-world applications.



Motivation for numerical solutions of linear systems



Fill in the blank: In computer science, numerical solutions of linear systems are applied in _____ learning and optimization problems.

5.1.2 Basic concepts and definitions

System of linear equations: A set of equations of the form $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$.

Coefficient matrix: The matrix containing the coefficients of the variables.

Solution vector: The vector of unknowns to be determined.

Consistency: A system is consistent if it has at least one solution.

Condition number: A measure of sensitivity of the solution to changes in input data.

A box listing system of equations, coefficient matrix, solution vector, consistency, and condition number with definitions.

Core Definitions

Concept	Definition
System of Linear Equations	A collection of two or more linear equations involving the same set of variables. In matrix form: $Ax = b$.
Coefficient Matrix (A)	An $m \times n$ matrix containing the coefficients of the variables. Its properties (sparsity, symmetry) determine the best solver.
Solution Vector (x)	The vector of unknown values that satisfies all equations in the system simultaneously.
Condition Number	A measure of how sensitive the solution x is to small changes or errors in the input data (A or b).
Residual Vector (r)	The difference $r = b - Ax$, where $\hat{x}$ is the calculated approximation. It measures how well the approximation fits the system.
Pivot Element	The element on the main diagonal used during Gaussian elimination to eliminate values below it. Selecting the largest available value (Pivoting) improves stability.



Fill in the blank: The matrix containing the coefficients of the variables in a linear system is called the _____ matrix.

Pilot Questions 5.1

Systems of linear equations involve: A. Nonlinear functions B. Multiple linear equations with several unknowns C. Random variables D. Differential equations (Linked to SLO 5.1)

Numerical solutions of linear systems are applied in: A. Engineering B. Physics C. Computer science D. All of the above (Linked to SLO 5.1)

The matrix containing the coefficients of the variables is called the: A. Solution vector B. Coefficient matrix C. Condition number D. Consistency matrix (Linked to SLO 5.1)

A system is consistent if: A. It has no solution B. It has at least one solution C. It has infinite solutions only D. It is sensitive to input changes (Linked to SLO 5.1)

The condition number measures: A. The number of equations in the system B. Sensitivity of the solution to changes in input data C. The number of unknowns D. The stability of iterative methods (Linked to SLO 5.1)

5.2 Direct Methods (Gaussian Elimination and LU Decomposition)

Direct methods provide exact solutions (up to rounding errors) for systems of linear equations in a finite number of steps. They are widely used for small to medium-sized systems and serve as the foundation for more advanced numerical techniques.

Fill in the blank: Direct methods provide _____ solutions for systems of linear equations in a finite number of steps.

5.2.1 Gaussian Elimination

Gaussian elimination systematically reduces a system of equations to an equivalent upper triangular form, making it easier to solve by back substitution.

Steps:

Write the system in augmented matrix form.

Use row operations to eliminate variables step by step.

Reduce to upper triangular form.



Solve using back substitution.

Advantages: Straightforward and general. Limitations: Computationally expensive for very large systems; sensitive to rounding errors.

Gaussian Elimination: From $Ax=b$ to Upper Triangular Form

The Journey to a Solvable System



1. Start with $Ax=b$ Analysis

$$A \left[\begin{array}{ccc|c} 2 & 1 & 1 & 12 \\ 4 & -2 & 3 & 16 \\ 4 & -2 & 3 & 16 \\ 6 & 1 & 4 & 22 \end{array} \right]$$

Goal: Eliminate values below the first pivot (2)

2. Eliminate Below Row 1

$$A \left[\begin{array}{ccc|c} 2 & 1 & 1 & 12 \\ 0 & -4 & 1 & -8 \\ 0 & -4 & 1 & -8 \\ 0 & -2 & 1 & -14 \end{array} \right]$$

$$R_2' = R_2 - 3 \cdot R_1$$

Goal: Eliminate values below the second pivot (-4)

3. Upper Triangular Form

$$A \left[\begin{array}{ccc|c} 2 & 1 & 1 & 12 \\ 0 & -4 & 1 & -8 \\ 0 & -3 & 1 & -10 \\ 0 & 0 & 0.5 & -10 \end{array} \right]$$

Now solve using Back Substitution!



Key Concept: Linear systems transform a system as into equivalent form that that is easy is solve by working traw sacwarks from the seut arld equation. It's many direnear solvers.



Open Educational Resource

Gaussian elimination process

5.2.2 LU Decomposition

LU decomposition factors a matrix A into the product of a lower triangular matrix L and an upper triangular matrix U :

$$A=LU$$



This allows solving systems more efficiently, especially when multiple right-hand sides are involved.

Steps:

Factor the coefficient matrix into L and U .

Solve $Ly=b$ using forward substitution.

Solve $Ux=y$ using back substitution.

Advantages: Efficient for repeated solutions with the same coefficient matrix.

Limitations: Requires pivoting for numerical stability.



LU Decomposition: Factoring a Matrix for Easier Solving

The Power of Triangular Matrices.

1. The Concept: $Ax=LUx=b$

$$A = LU$$

$A = \begin{bmatrix} 4 & 3 \\ 8 & -6 & 2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 4 & -6 & 2 \\ 1 & 1 \end{bmatrix}$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 8 & -6 & 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 9 & 0 & 3 \\ 2 \end{bmatrix}$$

Decomposes A into a Lower (L) and an Upper (U) triangular matrix.

2. Step 1: Forward Substitution ($Ly=b$)

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} y = \begin{bmatrix} 12 \\ -4 \\ 10 \end{bmatrix}$$

Solve for y. This is easy because L is triangular.

- $1y_1 = 12 \Rightarrow y_1 = 12$
- $2y_1 + 1y_2 = 4 \Rightarrow 2(12) + y_2 = -4$
- $0.5y_1 - 1y_2 + 1y_3 = 10 \Rightarrow y_2 = 28$
- $0.5(2) - 11 - 1(-28) + 10 = 10$
 $y_3 = -24$

3. Step 2: Backward Substitution ($Ux=y$)

$$U = \begin{bmatrix} 4 & 3 & 0 \\ 4 & -6 & 2 \\ 1 & 1 & 0 \end{bmatrix} x = \begin{bmatrix} 12 \\ -4 \\ 10 \end{bmatrix}$$

Solve for x. This is easy because U is triangular.

- $2x_3 = 24 \Rightarrow x_3 = 12$
- $-12x_2 = 0 = 28$
- $4x_1 + 3x_3 = 28 \Rightarrow x_2 = 7/3$
- $4x_1 + 3(7) + 1(2) = 12$ $x_3 = 7/4$
 $y_1 = 17/4$

$$\text{Solution: } x = [17/4, 7/3, -12]^T$$



Key Concept: LU Decomposition factors a dense matrix into two easy-to-solve triangular matrices. It is powerful direct method, especially you need to solve $Ax=b$ for multiple different b vectors.

LU decomposition process

Fill in the blank: LU decomposition factors a matrix into a lower triangular matrix and an _____ triangular matrix.

5.2.3 Error considerations

Rounding errors: Accumulate during elimination steps.



Pivoting: Used to reduce numerical instability.

Condition number: Determines sensitivity of the solution to input changes.

A box listing rounding errors, pivoting, and condition number.

Error Type	Description	Mitigation Strategy
Round-off Error	Occurs because computers represent numbers with finite precision (e.g., 64-bit floats). Small errors in each arithmetic operation accumulate.	Use Double Precision or higher-order data types.
Swamping	When a very large number is added to a very small one, the small number's contribution is "lost" due to limited significant digits.	Implement Partial Pivoting to avoid small divisors.
Division by Zero	Occurs if a pivot element is exactly zero during elimination, making the algorithm fail.	Pivoting swaps rows to move a non-zero value into the pivot position.
Ill-Conditioning	A property of the matrix itself where the rows are nearly parallel. Tiny changes in input lead to massive changes in output.	Calculate the Condition Number ; if too high, the results are unreliable regardless of the method.

Fill in the blank: Pivoting is used in Gaussian elimination and LU decomposition to reduce numerical _____.

Pilot Questions 5.2

Gaussian elimination reduces a system to: A. Lower triangular form B. Upper triangular form C. Diagonal form D. Identity matrix (Linked to SLO 5.2)



LU decomposition factors a matrix into: A. Two diagonal matrices B. Lower and upper triangular matrices C. Identity and diagonal matrices D. Symmetric matrices (Linked to SLO 5.2)

Which method is more efficient when solving multiple systems with the same coefficient matrix? A. Gaussian elimination B. LU decomposition C. Jacobi method D. Gauss-Seidel method (Linked to SLO 5.2)

Pivoting in direct methods is used to reduce: A. Computational cost B. Numerical instability C. Step size D. Condition number (Linked to SLO 5.2)

The condition number measures: A. Sensitivity of the solution to input changes B. Number of equations in the system C. Efficiency of iterative methods D. Accuracy of Gaussian elimination (Linked to SLO 5.2)

5.3 Iterative Methods (Jacobi and Gauss-Seidel)

Unlike direct methods, **iterative methods** generate successive approximations to the solution of a system of linear equations. They are especially useful for large, sparse systems where direct methods become computationally expensive.

Fill in the blank: Iterative methods generate successive _____ to the solution of a system of linear equations.

5.3.1 Jacobi Method

The Jacobi method updates each variable independently using values from the previous iteration.

For a system $Ax=b$, the update rule is:

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

Steps:

Start with an initial guess for x .

Update each variable using the formula above.

Repeat until convergence (changes become sufficiently small).

Advantages: Simple and parallelizable. Limitations: Convergence is not guaranteed unless the matrix is diagonally dominant.



Jacobi Method: Updating Variables Independently for Iterative Solutions

1. The Concept

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{array} \right\}$$

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3) / a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3) / a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2) / a_{33}$$

Using values from previous iteration (k)

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

2. The Iteration

Step 1: Initial Guess (k=0)

$$x^{(0)} = [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}]^T = [0, 0, 0]^T$$

Step 2: First Update (k=1)

$$x^{(1)} = \left(\frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}}{a_{11}}, \frac{b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)}}{a_{22}}, \frac{b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)}}{a_{33}} \right)^T$$

Step 3: Second Update (k=2)

$$x^{(2)} = \left(\frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)}}{a_{11}}, \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(1)}}{a_{22}}, \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}}{a_{33}} \right)^T$$

$$x^{(3)} = \left(\frac{b_1 - a_{12}x_2^{(2)} - a_{13}x_3^{(2)}}{a_{11}}, \frac{b_2 - a_{21}x_1^{(2)} - a_{23}x_3^{(2)}}{a_{22}}, \frac{b_3 - a_{31}x_1^{(2)} - a_{32}x_2^{(2)}}{a_{33}} \right)^T$$

Converges to True Solution x^*



Key Concept: The Jacobi Method is iterative technique that updates each variable's value independently, using only values from the previous time step. This makes it simple and parallelizable, but convergence is often slow.



Open
Educational
Resource

Jacobi method iteration process

Fill in the blank: The Jacobi method requires the coefficient matrix to be _____ dominant for guaranteed convergence.

5.3.2 Gauss-Seidel Method

The Gauss-Seidel method improves upon Jacobi by using updated values immediately within the same iteration.

Formula:

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

Steps:

Start with an initial guess.

Update each variable sequentially, using the most recent values.



Repeat until convergence.

Advantages: Faster convergence than Jacobi in many cases. Limitations: Still requires diagonal dominance or positive definiteness for guaranteed convergence.

Gauss-Seidel Method: Using Updated Values for Iterative Convergence

The Power of In-Place Updates

1. The Concept

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{array} \right.$$

$$l_1 = a_{21}x_1 + a_{22}x_2 - b_2 = b_3$$

$$x_1 = (2) - a_{22}^{-1}(-b_2 + a_{23}x_3)$$

$$x_1 = (3) - a_{31}^{-1}(-b_3 + a_{32}x_2 + a_{33}x_3)$$

Goal: Eliminate values below the first pivot (1)

Uses newly computed x_1 values immediately. Faster than Jacobi.

$$x^{(0)} = [0, 0, 0]^T$$

$$\begin{array}{c} \text{L} \\ \begin{array}{ccc} 1 & & \\ & 1 & \\ & & 0 \end{array} \end{array} = \begin{array}{c} \text{U} \\ \begin{array}{ccc} 1 & & \\ & 1 & \\ & & 12 \end{array} \end{array} \begin{array}{c} x \\ y \\ z \end{array} \begin{array}{c} \\ \\ [2-28, -25]^T \end{array}$$

1. $1y_1 = 12 \Rightarrow y_1 = 12$
2. $2y_1 + 1y_2 = 4 \Rightarrow 2(12) + y_2 = -4$
3. $0.5y_1 - 1y_2 + 1z = 10 \Rightarrow y_2 = 28$
3. $0.5(2) - 11 - 1(=28) + 10 = 10$

2. Step 1: Intimward Guesss (k=1)

$$x^{(0)} = b_1 = [x_1, 0, 0, 30]^T$$

$$x^{(1)} = (0) = [x_1, 0, 0, 33]^T$$

Step 3: Update (k=1)

$$x^{(1)} = b_1 - a_{21}x_1 - a_{32}x_2 - b_3$$

Now solves to to True Substution!

Faster Convergence x^*



Key Concept: Gauss-Seidel is intertive technique that uses updated valus they computed. This a computed. This sequential updating makes it hple faste but harder allelizte than Jacoci. differeslize than direyeyinar solvers.



Gauss-Seidel method iteration process

Fill in the blank: The Gauss-Seidel method uses the most _____ values within the same iteration.



5.3.3 Convergence and stability

Convergence criteria: Iterative methods converge if the coefficient matrix is diagonally dominant or positive definite.

Rate of convergence: Gauss-Seidel generally converges faster than Jacobi.

Stability: Poorly conditioned systems may cause divergence or slow convergence.

A box listing convergence criteria, rate of convergence, and stability issues.

Requirement	Description	Why it Matters
Strict Diagonal Dominance (SDD)	A matrix is SDD if the absolute value of the diagonal element in each row is greater than the sum of the absolute values of the other elements in that row.	If a matrix is SDD, both Jacobi and Gauss-Seidel are guaranteed to converge regardless of the initial guess.
Spectral Radius (ρ)	The largest absolute eigenvalue of the iteration matrix.	Convergence occurs if and only if $\rho < 1$. The smaller the ρ , the faster the method reaches the solution.
Symmetric Positive Definite (SPD)	A matrix that is symmetric ($A = A^T$) and all eigenvalues are positive.	Gauss-Seidel is guaranteed to converge for any SPD matrix, even if it is not diagonally dominant.

Fill in the blank: Iterative methods converge if the coefficient matrix is diagonally _____ or positive definite.

Pilot Questions 5.3



Iterative methods generate: A. Exact solutions in finite steps B. Successive approximations to the solution C. Random guesses D. Direct substitutions (Linked to SLO 5.3)

The Jacobi method updates each variable using: A. Current iteration values B. Previous iteration values C. Random values D. Exact solutions (Linked to SLO 5.3)

The Gauss-Seidel method improves upon Jacobi by: A. Using updated values immediately within the same iteration B. Reducing step size C. Ignoring convergence criteria D. Using diagonal matrices only (Linked to SLO 5.3)

Iterative methods converge if the coefficient matrix is: A. Symmetric B. Diagonally dominant or positive definite C. Sparse D. Upper triangular (Linked to SLO 5.3)

Which method generally converges faster? A. Jacobi method B. Gauss-Seidel method C. Gaussian elimination D. LU decomposition (Linked to SLO 5.3)

5.4 Applications of Numerical Linear Systems Solutions

Numerical methods for solving systems of linear equations are at the heart of modern computational science. They enable us to tackle problems that are too large or complex for exact algebraic solutions, making them indispensable in engineering, computer science, and physics.

Fill in the blank: Numerical solutions of linear systems are essential in engineering, _____ science, and physics.

5.4.1 Applications in engineering

Structural analysis: Determining forces and displacements in trusses and frames.

Circuit networks: Solving for currents and voltages in electrical circuits.

Fluid mechanics: Modeling flow in pipelines and networks.

Mechanical engineering: Stress-strain analysis in materials.

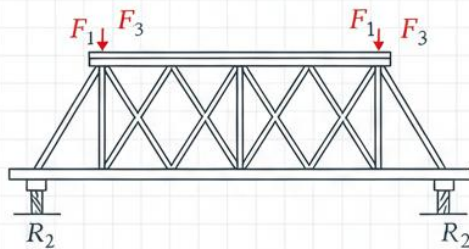


Applications of Linear Systems in Engineering

Solving Complex Networks

Open Educational Resource

1. Engineering: Truss Analysis

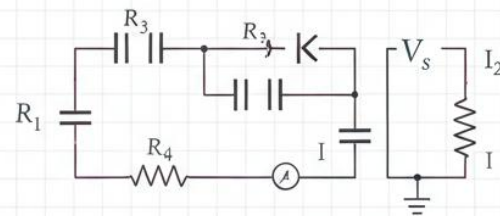


System of equations for joint forces:

$$Ax = b$$

Models forces in static structures, where A is the connectivity matrix, x has the tensions, and b has external forces.

2. Electrical Engineering: Circuit Network Analysis



Applying Kirchhoff's Laws (KCL/KVL):

$$V_s - I_1 R_1 - (I_1 - I_2) R_2 = 0$$

$$I_2 - I_2 R_3 - (I_1 - I_2) R_2 = 0$$

Solves for unknown branch currents/voltages in complex networks.



Key Concept: Linear systems $Ax = b$ are fundamental to modeling interconnected components and dependencies in countless real-world engineering applications.

Applications of numerical linear systems in engineering

Fill in the blank: In structural engineering, linear systems are used to determine forces and _____ in trusses and frames.

5.4.2 Applications in computer science

Graphics: Transformations and rendering rely on solving linear systems.

Machine learning: Training models often involves solving large systems of equations.

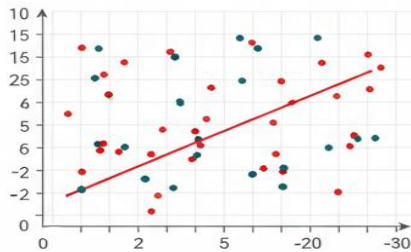
Optimization: Linear programming and resource allocation problems.



Data science: Regression analysis and statistical modeling.

Applications of Linear Systems in Computer Science: Solving Computer Science: Powering Modern Algorithms

1. Machine Learning: Data Representation & Regression



Dataset: X (features), y , labels)

Model: $y = Xw$ (weights)

Solution: $w = (X^T X)^{-1} X^T y$ (Equation)

Models relationships in data for predictions.

2. Computer Graphics: 2D Transformations



Scale: Rotate: Translate: (weights)

$S_x = I_3 + 0 \quad R_x \quad (cos - 0 \quad R_x - 0)$
(Normal Equation)

$$x' = M x$$

Manipulates objects



Key Concept: Linear systems $Ax = b$ are mathematical backbone of modeling many computer science algorithms, from analysis at aka in countless real-world graphics efficiently.



Applications of numerical linear systems in computer science

Fill in the blank: In machine learning, solving linear systems is often required during model _____.

5.4.3 Applications in physics

Finite element methods (FEM): Used in simulations of heat transfer, elasticity, and fluid flow.

Quantum mechanics: Solving Schrödinger's equation numerically.

Astrophysics: Modeling gravitational interactions in large systems.

Electromagnetism: Solving Maxwell's equations in discretized form.

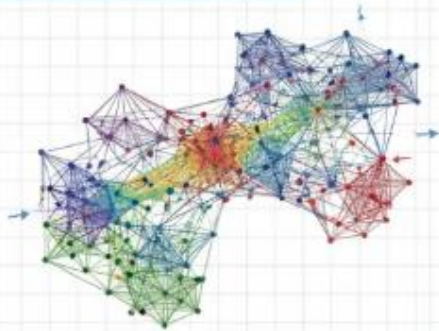


Applications of Linear Systems: Physics & Advanced Engineering

Solving Complex Systems



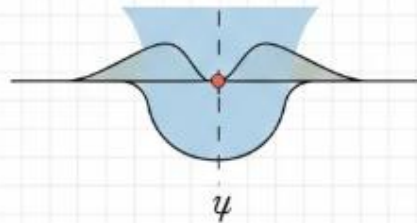
1. Advanced Engineering: Finite Element Method (FEM)



Equation: $Ku = f$ where K_u is large, sparse stiffness matrix, u , and f are displacements and applied forces.

Models complex geometries by breaking of small elements. Solves for stress, strain or heat distribution.

2. Quantum Mechanics: Schrödinger Equation



$$H\psi = E\psi \quad \psi(x)$$

This becomes a linear system $Ax = \lambda x$, when discretized on a grid, where Ax form H_{ij} , x is wavefunction values, λ is energy eigenvalues.

Models the energy levels (eigenvalues) and states (vectors) of a quantum system.



Key Concept: Linear systems are essential for modeling continuous physical phenomena by discretized into solvable, coupled equations. This allows numerical simulation of highly complex systems in both classical and quantum physics.

Applications of numerical linear systems in physics

Fill in the blank: Finite element methods in physics rely on solving large systems of _____ equations.

5.4.4 Computational considerations

Accuracy: Direct methods provide exact solutions (up to rounding errors), while iterative methods approximate solutions.



Efficiency: Iterative methods are preferred for very large, sparse systems.

Stability: Pivoting and preconditioning improve numerical stability.

Scalability: Parallel computing enhances performance for massive systems.

A box listing accuracy, efficiency, stability, and scalability with examples.

The Four Pillars of Numerical Linear Algebra

Consideration	Definition	High-Stakes Example
Accuracy	The closeness of the computed solution $\hat{x}$ to the true solution x . Affected by round-off error and matrix conditioning.	Structural Engineering: A small error in a bridge's stress matrix could lead to structural failure.
Stability	The algorithm's ability to resist the growth of rounding errors during calculation.	Long-term Simulations: Unstable solvers can cause errors to "snowball," resulting in non-physical outputs (like negative mass).
Efficiency	The computational cost (FLOPs) and time required to reach a solution.	Real-Time Graphics: A physics engine must solve thousands of collisions per second; speed is often prioritized over perfect accuracy.
Scalability	How the solver's performance changes as the number of variables (n) increases significantly.	Big Data/AI: Training a neural network with billions of parameters requires solvers that scale across thousands of GPUs.

Fill in the blank: Iterative methods are preferred for very large, _____ systems.

Pilot Questions 5.4

In structural engineering, linear systems are used to determine: A. Currents and voltages B. Forces and displacements in trusses and frames C. Regression coefficients D. Quantum states (Linked to SLO 5.4)



In computer graphics, solving linear systems is essential for: A. Rendering and transformations B. Disease modeling C. Circuit analysis D. Weather forecasting (Linked to SLO 5.4)

Finite element methods in physics rely on solving: A. Small systems of equations B. Large systems of linear equations C. Randomized equations D. Exact symbolic equations only (Linked to SLO 5.4)

Iterative methods are preferred for: A. Small dense systems B. Very large sparse systems C. Exact symbolic solutions D. Systems with no solution (Linked to SLO 5.4)

Pivoting in direct methods improves: A. Computational speed only B. Numerical stability C. Step size control D. Regression accuracy (Linked to SLO 5.4)

Series Summary

In this Series, we explored **numerical solutions of systems of linear equations**, a cornerstone of computational mathematics. We began with an introduction to the motivation and importance of solving linear systems numerically, emphasizing their role in engineering, physics, computer science, and economics. Next, we studied **direct methods**, including Gaussian elimination and LU decomposition, which provide exact solutions (up to rounding errors) in a finite number of steps. We then examined **iterative methods**, such as the Jacobi and Gauss-Seidel methods, which generate successive approximations and are especially useful for large, sparse systems. Finally, we discussed **applications** of numerical linear systems solutions in engineering (structural analysis, circuits), computer science (graphics, machine learning), and physics (finite element methods, quantum mechanics), along with computational considerations like accuracy, efficiency, stability, and scalability.

By completing this Series, you should now be able to explain why numerical solutions of linear systems are necessary, apply direct methods to solve systems, use iterative methods to approximate solutions, and evaluate their accuracy, efficiency, and stability in real-world applications. These abilities directly reflect the Series Learning Outcomes (SLOs) and equip you with practical skills for solving large-scale linear problems numerically.

Glossary of Terms

System of linear equations: A set of equations involving linear relationships among unknowns.



Coefficient matrix: The matrix containing coefficients of the variables in a linear system.

Solution vector: The vector of unknowns to be determined.

Consistency: A system is consistent if it has at least one solution.

Condition number: A measure of sensitivity of the solution to changes in input data.

Gaussian elimination: A direct method that reduces a system to upper triangular form for back substitution.

LU decomposition: A factorization method that expresses a matrix as the product of lower and upper triangular matrices.

Jacobi method: An iterative method that updates variables using values from the previous iteration.

Gauss-Seidel method: An iterative method that uses updated values immediately within the same iteration.

Convergence: The property of iterative methods to approach the true solution as iterations progress.

Stability: The ability of a numerical method to prevent errors from growing uncontrollably.

Sparse system: A system where most coefficients in the matrix are zero, common in large-scale applications.

Concept Check Questions 5

Objective questions

Systems of linear equations involve: A. Nonlinear functions B. Multiple linear equations with several unknowns C. Random variables D. Differential equations (Linked to SLO 5.1)

Gaussian elimination reduces a system to: A. Lower triangular form B. Upper triangular form C. Diagonal form D. Identity matrix (Linked to SLO 5.2)

LU decomposition factors a matrix into: A. Two diagonal matrices B. Lower and upper triangular matrices C. Identity and diagonal matrices D. Symmetric matrices (Linked to SLO 5.2)



The Jacobi method updates each variable using: A. Current iteration values B. Previous iteration values C. Random values D. Exact solutions (Linked to SLO 5.3)

Iterative methods converge if the coefficient matrix is: A. Symmetric B. Diagonally dominant or positive definite C. Sparse D. Upper triangular (Linked to SLO 5.3)

Short-answer questions

Define a coefficient matrix and explain its role in solving linear systems. (Linked to SLO 5.1)

State the step-by-step procedure for Gaussian elimination. (Linked to SLO 5.2)

Explain how the Gauss-Seidel method improves upon the Jacobi method. (Linked to SLO 5.3)

Long-answer questions

Compare direct methods (Gaussian elimination, LU decomposition) and iterative methods (Jacobi, Gauss-Seidel) in terms of accuracy, efficiency, and scalability. (Linked to SLO 5.3)

Basic response: Describe how each method works.

Value-added response: Analyse their relative strengths, weaknesses, and suitability for different problem sizes.

Discuss applications of numerical solutions of linear systems in engineering, computer science, and physics. (Linked to SLO 5.4)

Basic response: Provide examples of applications in each field.

Value-added response: Evaluate how accuracy, efficiency, and stability affect these applications.

Answers to Concept Check Questions 5

B. Multiple linear equations with several unknowns

B. Upper triangular form

B. Lower and upper triangular matrices

B. Previous iteration values

B. Diagonally dominant or positive definite



A coefficient matrix is the matrix containing the coefficients of the variables in a linear system. It plays a central role in representing the system in matrix form, enabling numerical methods to be applied efficiently.

Steps: (i) Write system in augmented matrix form; (ii) Use row operations to eliminate variables; (iii) Reduce to upper triangular form; (iv) Solve using back substitution.

The Gauss-Seidel method improves upon Jacobi by using updated values immediately within the same iteration, which generally accelerates convergence.

Basic response: Direct methods solve systems exactly in finite steps, while iterative methods approximate solutions through successive iterations.

Value-added response: Direct methods are accurate but computationally expensive for large systems; iterative methods are efficient for large, sparse systems but require convergence conditions. Scalability favors iterative methods in high-performance computing.

Basic response: Engineering (structural analysis, circuits), computer science (graphics, machine learning), physics (finite element methods, quantum mechanics).

Value-added response: Accuracy ensures reliable structural designs, efficiency enables large-scale machine learning, and stability is critical in physics simulations like FEM and quantum mechanics.



lanetonline.com

