An Introduction to Numerical Analysis with MATLAB

Lecture Notes

Mohammad Sabawi

Department of Mathematics
College of Education for Women
Tikrit University

Email: mohammad.sabawi@tu.edu.iq

01 October 2018

List of Tables

Contents

Preface

The aim of these class notes is to cover the necessary materials in a standard numerical analysis course and it is not intended to add to the plethora of Numerical Analysis texts. We tried our best to write these notes in concise, clear and accessible way, to make them more attractive to the readers. These lecture notes cover the basic and fundamental concepts and principles in numerical analysis and it is not a comprehensive introduction to numerical analysis. We emphasise in these notes on the mathematical principles via explaining them by the aid of numerical software **MATLAB**. The prerequisite material for this course are a course in **Calculus**, **Linear Algebra** and **Differential Equations**. A basic knowledge in MATLAB is helpful but it is not necessary. There is a glut of numerical software nowadays, among these we chose to use MATLAB because of its wide capabilities in scientific computing.

The notes contain sufficient material for a full year of study and can be covered in two courses for undergraduate mathematics and engineering students.These notes consist of eight chapters cover the basic and fundamental topics in numerical analysis. Each chapter contains some relevant examples to illustrate the concepts and ideas introduced in the chapter and ends with a set of exercises address the topics covered in each chapter.

Chapter 1

Introduction

1.1 Numerical Analysis: An Introduction

Numerical analysis is a branch of mathematics studies the methods and algorithms which used for solving a variety of problems in different areas of todays life such as *mathematics*, *physics*, *engineering*, *medicine* and *social* and *life sciences*. The main objective of numerical analysis is investigation finding new mathematical approaches for approximating the underlying problems, and also development of the current algorithms and numerical schemes to make them more efficient and reliable. The advent of computers revolutionise numerical analysis and nowadays with parallel and super computers the numerical computations became more easier compared with the past where solving simple problems take a long time, much effort and require hard work. In principle, numerical analysis mainly focuses on the ideas of *stability*, *convergence*, *accuracy*, *consistency* and *error analysis*. In the literature numerical analysis also known as **scientific computing**, **scientific computation**, **numerics**, **computational mathematics** and **numerical mathematics**. Numerical analysis can be divided into the following fields:

1. **Numerical Solutions of Linear Algebraic Equations**.
2. **Numerical Solutions of Nonlinear Algebraic Equations**.
3. **Interpolation and Extrapolation**.
4. **Approximation Theory and Curve Fitting**.
5. **Numerical Differentiation**.
6. **Numerical Integration**.

7. **Numerical Optimisation**.
8. **Numerical Solutions of Eigenvalue Problems**.
9. **Numerical Solutions of Ordinary Differential Equations**.
10. **Numerical Solutions of Partial Differential Equations**.
11. **Numerical Solutions of Integral Equations**.
12. **Numerical Modelling**.
13. **Numerical Simulation**.

Numerical analysis is dated back to the Babylonians works in approximating the square root of 2. During this long journey of evolution many many scientists contributed to its development and progress among these we just name a few such as **Lagrange**, **Gauss**, **Newton**, **Euler**, **Legendre** and **Simpson**.

1.2 Numbers Representation in Computer

Human beings do arithmetic in their daily life using the *decimal (base* 10*) number system.* Nowadays, most computers use *binary (base* 2*) number system.* We enter the information to computers using the decimal system but computers translate them to the binary system by using the *machine language.*

Definition 1 (**Scientific Notation**)**.** *Let k be a real number, then k can be written in the following form*

$$k = m \times 10^n,$$

*where m is any real number and the exponent n is an integer. This notation is called the **scientific notation** or **scientific form** and sometimes referred to as **standard form**.*

Example 1. *Write the following numbers in scientific notation:*

1. 0.00000834.

2. 25.45879.

3. 3400000.

4. 33.

5. $2,300,000,000$.

6. 2.718282.

Solution:

1. $0.00000834 = 8.34 \times 10^{-6}$.
2. $25.45879 = 2.545879 \times 10^{1}$.
3. $3400000 = 3.4 \times 10^{6}$.
4. $33 = 3.3 \times 10^{1}$.
5. 2.3×10^{9}.
6. $2.718282 = 2.718282 \times 10^{0}$.

1.2.1 Floating-Point Numbers

In the decimal system any real number $a \neq 0$ can be written in the **normalised decimal floating-point form** in the following way

$$a = \pm 0.d_1 d_2 d_3 \cdots d_k d_{k+1} d_{k+2} \cdots \times 10^n,\ 1 \leq d_1 \leq 9,\ 0 \leq d_i \leq 9, \tag{1.1}$$

for each $i = 2, \cdots$, and n is an integer called the **exponent** (n can be positive, negative or zero). In computers we use a finite number of digits in representing the numbers and we obtain the following form

$$b = \pm 0.d_1 d_2 d_3 \cdots d_k \times 10^n,\ 1 \leq d_1 \leq 9,\ 0 \leq d_i \leq 9, \tag{1.2}$$

for each $i = 2, \cdots, k$. These numbers are called **k-digit decimal machine numbers**.

Also, the normalised floating-point decimal representation of the number $a \neq 0$ can be written in other way as

$$a = \pm r \times 10^n,\ (\frac{1}{10} \leq r < 1), \tag{1.3}$$

the number r is called the **normalised mantissa**.

The floating-point representation in binary number system can be defined by the same way as in the decimal number system. If $a \neq 0$, it can be represented as

$$a = \pm p \times 2^m,\ (\frac{1}{2} \leq p < 1), \tag{1.4}$$

where $p = (0.b_1 b_2 b_3 \cdots)_2$, $b_1 = 1$.

1.3 Errors

Occurrence of error is unavoidable in the field of scientific computing. Instead, numerical analysts try to investigate the possible and best ways to minimise the error. The study of the error and how to estimate and minimise it are the fundamental issues in *error analysis.*

1.3.1 Error Analysis

In numerical analysis we approximate the exact solution of the problem by using numerical method and consequently an error is committed. The numerical error is the difference between the exact solution and the approximate solution.

Definition 2 (**Numerical Error**). *Let x be the exact solution of the underlying problem or a true value and x^* its approximate solution or approximate value, then the error (denoted by e) in solving this problem or in approximating the value of x is*

$$e = x - x^*. \tag{1.5}$$

1.3.2 Sources of Error in Numerical Computations

- **Blunders (Gross Errors)** These errors also called **humans errors**, and are caused by humans mistakes and oversight and can be minimised by taking care during scientific investigations. These errors will add to the total error of the underlying problem and can significantly affect the accuracy of solution.

- **Modelling Errors** These errors arise during the modelling process when scientists ignore effecting factors in the model to simplify the problem. Also, these errors known as **formulation errors**.

- **Data Uncertainty** These errors are due to the uncertainty of the physical problem data and also known as **data errors** or **noise**.

- **Discretisation Errors** Computers represent a function of continuous variable by a number of discrete values. Also, scientists approximate and replace complex continuous problems by discrete ones and this results in **discretisation errors**.

- **Loss of Significance** This phenomenon occurs when subtracting two nearly equal numbers and can be avoided by using some mathematical tricks such as algebraic manipulation. It is also known as **subtractive cancellation**, **catastrophic cancellation** or **loss of significant digits**.

- **Rounding Errors** Computers represent numbers in finite number of digits and hence some quantities cannot be represented exactly. The error caused by replacing a number a by its closest machine number is called the **roundoff error** or **round-off error** and the process is called **correct rounding**. This type of error happens when a true value of a real number x sometimes not stored or saved exactly due to the limited fixed precision of computer's representation.

- **Chopping Errors** These errors occur when chopping a number with infinite digits or a number with $k+1$ digits and replaced it by a $k-$digits number.

- **Truncation Errors** These errors arise when replacing complicated mathematical expressions by simple and elementary mathematical formulas. As an example of truncation error approximating a complicated function with truncated Taylor series. We will discuss truncation error in detailed way later.

1.3.3 Floating Point Representation

1.3.4 Absolute and Relative Errors

Definition 3 (Absolute Error). *The absolute error $\hat{e}$ of the error e is defined as the absolute value of the error e*

$$\hat{e} = |x - x^*|. \tag{1.6}$$

Definition 4 (Relative Error). *The relative error $\tilde{e}$ of the error e is defined as the ratio between the absolute error $\hat{e}$ and the absolute value of the true value x*

$$\tilde{e} = \frac{\hat{e}}{|x|} = \frac{|x - x^*|}{|x|},\ x \neq 0. \tag{1.7}$$

Example 2. *Let $x = 3.141592653589793$ is the value of the constant ratio π correct to 15 decimal places and $x^* = 3.14159265$ be an approximation of x. Compute the following quantities:*

a. The error.

b. The absolute error.

c. The relative error.

Solution:

a. The error

$$\begin{aligned} e = x - x^* &= 3.141592653589793 - 3.14159265 = 3.589792907376932e - 09 \\ &= 3.589792907376932 \times 10^{-9} = 0.000000003589792907376932. \end{aligned}$$

b. The absolute error

$$\hat{e} = |x - x^*| = |3.141592653589793 - 3.14159265| = 3.589792907376932e - 09.$$

c. The relative error

$$\begin{aligned} \tilde{e} &= \frac{\hat{e}}{|x|} = \frac{|x - x^*|}{|x|} = \frac{3.141592653589793 - 3.14159265}{3.141592653589793} \\ &= \frac{3.589792907376932e - 09}{3.141592653589793} = 1.142666571770530e - 09. \end{aligned}$$

Example 3. *Approximate the following decimal numbers to three significant digits by using rounding and chopping rules:*

1. $x_1 = 1.34579.$
2. $x_2 = 1.34679.$
3. $x_3 = 1.34479.$
4. $x_4 = 3.34379.$
5. $x_5 = 2.34579.$
6. $x_6 = 0.54387.$

Solution:

(i) ***Rounding:***

(a) $x_1 = 1.35.$

(b) $x_2 = 1.35.$

(c) $x_3 = 1.34.$

(d) $x_4 = 3.34.$

(e) $x_5 = 2.35.$

(f) $x_6 = 0.544.$

(ii) ***Chopping:***

(a) $x_1 = 1.34.$

(b) $x_2 = 1.34.$

(c) $x_3 = 1.34.$

(d) $x_4 = 3.34.$

(e) $x_5 = 2.34.$

(f) $x_6 = 0.543.$

1.4 Stable and Unstable Computations: Conditioning

Stability is one of the most important characteristics in any efficient and robust numerical scheme.

Definition 5 (**Numerical Stability**)**.** *The numerical algorithm or process is called **stable** if the final result is relatively not affected by the perturbations during computation process. In other words, the numerical method or technique is stable if small changes in the initial conditions or initial data will produce small changes in outputs or final results. Otherwise it is called **unstable**.*

The **stability** notion is analogous and closely related to the notion of **conditioning**.

Definition 6 (**Conditioning**)**.** ***Conditioning** is a measure of how **sensitive** the output to small changes in the input data. In literature **conditioning** is also called **sensitivity**.*

- *The problem is called **well-conditioned** or **insensitive** if small changes in the input data lead to small changes in the output data.*
- *The problem is called **ill-conditioned** or **sensitive** if small changes in the input data lead to big changes in the output data.*

Definition 7 (**Condition Number of a Function**)**.** *If f is a differentiable function at x in its domain then the **condition number** of f at x is*

$$cond(f(x)) = \frac{|xf'(x)|}{|f(x)|},\ f(x) \neq 0. \tag{1.8}$$

Note: Condition number of a function f at x in its domain sometimes denoted by $C_f(x)$.

Definition 8 (**Condition Number of a Matrix**)**.** *If A is a non-singular $n \times m$ matrix, the **condition number** of A is defined by*

$$cond(A) = \|A\|\|A^{-1}\|, \tag{1.9}$$

where

$$\|A\| = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|}, \tag{1.10}$$

and x is a $m \times 1$ column vector.

Definition 9 (**Well-Posed Problem**). *The problem is* ***well-posed*** *if satisfies the following three conditions:*

a. The solution exists.

b. The solution is unique.

c. The solution depends continuously on problem data.

Otherwise, the problem is called ***ill-posed****.*

Remark 1. *Note that:*

1. *The problem is* ***ill-posed*** *or* ***sensitive*** *if cond* $\gg 1$.
2. *The problem is* ***well-posed*** *or* ***insensitive*** *if cond* < 1.

Example 4. *Find the condition number of the function* $f(x) = \sqrt{x}$.

Solution:

$$f(x) = \sqrt{x} \implies f'(x) = \frac{1}{2\sqrt{x}},\ x \neq 0,$$

implies that

$$cond(f(x)) = \frac{|xf'(x)|}{|f(x)|} = \frac{|\frac{x}{2\sqrt{x}}|}{|\sqrt{x}|} = \frac{1}{2}.$$

This indicates that the small changes in the input data lead to changes in the output data of half size the changes in the input data.

Example 5. *Let*

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0.5 & 3 \\ 0.1 & 1 & 0.3 \end{bmatrix},$$

the inverse of A *can be computed by using MATLAB command* ***inv(A)*** *to obtain*

$$A^{-1} = \begin{bmatrix} 4.7500 & -2.1667 & 5.8333 \\ 0.5000 & -0.3333 & 1.6667 \\ -3.2500 & 1.8333 & -4.1667 \end{bmatrix}.$$

Also, the condition number of A *and its inverse can be computed using MATLAB commands* ***cond(A)*** *and* ***cond(inv(A))*** *to have cond*$(A) = 37.8704$ *and cond*$(A^{-1}) = 37.8704$. *We notice that the matrix* A *and its inverse have the same condition numbers.*

Definition 10 (**Accuracy**). *It is a measure of closeness of the approximate solution to the exact solution.*

Definition 11 (**Precision**). *It is a measure of closeness of the two or more measurements to each other.*

Remark 2. *Note that the accuracy and precision are different and they are not related. The problem maybe very accurate but imprecise and vice versa.*

1.5 Convergence and Order of Approximation

Convergence of the numerical solution to the analytical solution is one of the important characteristic in any good and reliable numerical scheme.

Definition 12 (**Convergence of a Sequence**). *Let $\{a_n\}_{n=1}^{\infty}$ be an infinite sequence of real numbers. This sequence is said to be **convergent** to a real number a (has a **limit** at a) if, for any $\epsilon > 0$ there exists a positive integer $N(\epsilon)$ such that*

$$|a_n - a| < \epsilon, \text{ whenever } n > N(\epsilon). \tag{1.11}$$

*Otherwise it is called a **divergent** sequence, a is called the **limit of the sequence** a_n. Other commonly used notations for convergence are:*

$$\lim_{n\to\infty} a_n = a \quad \text{or} \quad a_n \to a \ \text{ as } n, \quad \text{or} \quad \lim_{n\to\infty} (a_n - a) = 0, \tag{1.12}$$

*this means that the sequence $\{a_n\}_{n=1}^{\infty}$ **converges** to a otherwise it **diverges**.*

Definition 13 (**Order of Convergence**). *Let the sequence $\{a_n\}_{n=1}^{\infty}$ converges to a and set $e_n = a_n - a$ for any $n > 0$. If two positive constants M and q exist, such that*

$$\lim_{n\to\infty} \frac{|a_{n+1} - a|}{|a_n - a|^q} = \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^q} = M, \tag{1.13}$$

*then the sequence $\{a_n\}_{n=1}^{\infty}$ is to be convergent to a with the **order of convergence** q, the number M is called the **asymptotic error constant**.*

*If $q = 1$, the convergence is called **linear**.*
*If $q = 2$, the convergence is called **quadratic**.*

*If $q = 3$, the convergence is called **cubic**.*

Note that the convergence gets more rapid as q gets larger and larger.

Example 6. *Consider the sequence $\{\frac{1}{n}\}_{n=1}^{\infty}$, where n is a positive integer. Observe that $\frac{1}{n} \to 0$ as $n \to \infty$, it follows that*

$$\lim_{n\to\infty} \frac{1}{n} = 0.$$

Definition 14 (**Order of Approximation** $O(h^n)$)**.** *The function $f(h)$ is said to be **big Oh** of the function $g(h)$, if two real constants c, and C exist such that*

$$|f(h)| \leq C|g(h)| \quad \textit{whenever}\, h < c, \tag{1.14}$$

and denoted by $f(h) = O(g(h))$. The order of approximation is used to determine the rate at which a function grows.

Example 7. *Consider the functions $f(x) = x + 1$ and $g(x) = x^2$, where $x \geq 1$. Observe that $x \leq x^2$ and $1 \leq x^2$ for $x \geq 1$, hence $f(x) = x + 1 \leq 2x^2 = 2g(x)$ for $x \geq 1$. Consequently, $f(x) = O(g(x))$.*

Exercises

Exercise 1. *Write the following numbers in scientific form:*

1. 23.123.
2. $30,000,000$.
3. 0.000001573.
4. 39776444.
5. -345.386443.
6. -23000000.

Exercise 2. *Evaluate error, absolute error and relative error of the following values and their approximations:*

1. $x = 1,000,000,\ x^* = 999,999$.
2. $y = 0.00012887765,\ y^* = 0.00012897766$.
3. $z = 9776.96544,\ z^* = 9775.66544$.

Exercise 3. *Approximate the following numbers to four digits using rounding and chopping:*

1. 1.98876.
2. 33.87654.
3. 8.98879.
4. 2.88778.

Exercise 4. *Compute the condition number of the following functions:*

1. $f(x) = \cos(x)$.
2. $f(x) = \cos^{-1}(x)$.

.

Chapter 2

Numerical Solutions of Nonlinear Equations

2.1 Introduction

Nonlinear algebraic equations are wide spread in science and engineering and therefore their solutions are important scientific applications. There are a glut of numerical methods for solving these equations, and in these lecture notes, we study the most commonly used ones such as bisection, secant and Newton methods. Locating positions of roots of nonlinear equation is a topic of great importance in numerical mathematical analysis. The problem under consideration maybe has a root or has no root at all. The numerical methods which are used to find the roots of nonlinear equations are called **root-finding algorithms** or **numerical methods for locating a root**. **Iteration** is an important and basic concept in both *mathematics* and *computer science*, and has applications in *physics* and *engineering*. In these notes we consider a class of methods called **iterative methods** or **iteration methods** or **recursive methods** and as the name indicates a process is repeated until an acceptable solution is obtained.

Definition 15 (Zero of a Function)**.** *Let f be a real or complex valued function of a real or complex variable x. A real or complex number r satisfies $f(r) = 0$ is called* ***zero of*** *f or also called a* ***root of equation*** *$f(r) = 0$.*

Definition 16 (Order of a Zero)**.** *Let f and its derivatives $f', f'', \cdots, f^{(M)}$ are continuous and defined on an interval about the zero $x = r$. The function f or the equation $f(x) = 0$ is said to be has a zero or a root of order $M \geq 1$ at $x = r$ if and only if*

$$f(r) = 0,\ f'(r) = 0,\ f''(r) = 0, \cdots, f^{(M-1)}(r) = 0,\ f^{(M)} \neq 0. \tag{2.1}$$

If $M = 1$ then r is called a ***simple zero*** *or a* ***simple root****, and if $M > 1$ it is called a* ***multiple zero*** *or a* ***multiple root****. A zero (root) of order $M = 2$ is called a* ***double zero (root)****, and so on. Also, the zero (root) of order M is called a* ***zero (root) of multiplicity*** *M.*

Lemma 1. *If the function f has a zero r of multiplicity M, then there exists a continuous function h such that f can be factorised as*

$$f(x) = (x-r)^M h(x), \quad \lim_{x \to r} h(x) \neq 0. \tag{2.2}$$

Theorem 5 (Simple Zero Theorem)**.** *Assume that $f \in C^1[a,b]$. Then, f has a simple zero at $r \in (a,b)$ if and only if $f(r) = 0$ and $f'(r) \neq 0$.*

Example 8. *The function $f(x) = x^2 - 5x + 6 = (x-2)(x-3)$ has two real zeros $r_1 = 2$ and $r_2 = 3$, whereas the corresponding equation $x^2 - 5x + 6 = (x-2)(x-3) = 0$ has two real roots $r_1 = 2$ and $r_2 = 3$. According to the Lemma 1, the function f is factorised to*

$$f(x) = (x-2)^1(x-3) \text{ or } f(x) = (x-3)^1(x-2)$$

Example 9. *Show that the function $f(x) = e^{2x} - x^2 - 2x - 1$ has a zero of multiplicity 2 (double zero) at $x = 0$.*

Solution:

$$f(x) = e^{2x} - x^2 - 2x - 1, \; f'(x) = 2e^{2x} - 2x - 2, \text{ and } f''(x) = 4e^{2x} - 2.$$

Hence,

$$f(0) = e^0 - 0 - 0 - 1 = 0, \; f'(0) = 2e^0 - 0 - 2 = 0, \text{ and } f''(0) = 4e^0 - 2 = 2 \neq 0,$$

so, this implies that f ha s a double zero at $x = 0$.

2.2 Closed Methods

The basic idea of these methods is to find a closed interval $[a,b]$ no mattar how large such that it contains the root of the equation $f(x) = 0$ by stipulating that $f(a)$ and $f(b)$ have opposite signs, and for this reasons they are called **closed methods**. Also, these methods are known as **bracketing methods**. Once the interval is determined an iterative process is started until we reach a sufficiently small interval around the root for this reason these methods are termed **globally convergent methods**.

2.2.1 Bisection Method

It is a bracketing method used to find a zero of a continuous function f on the initial interval $[a, b]$ where a and b are real numbers, i.e. to find x such that $f(x) = 0$, this method requires that $f(a)$ and $f(b)$ have different signs. This method is based on the Intermediate Value Theorem, since f is continuous and has opposite signs on $[a, b]$ then there is a number r in $[a, b]$ such that $f(r) = 0$. This method is also known as **Bolzano method** or **bisection method of Bolzano** or **binary search method** or **interval halving method**. The first step in the solution process is to compute the midpoint $c = (a + b)/2$ of the interval $[a, b]$ and to proceed we consider the three cases:

1. If $f(a)f(c) < 0$ then r lies in $[a, c]$.
2. If $f(c)f(b) < 0$ then r lies in $[c, b]$.
3. If $f(c) = 0$ then the root is $c = r$.

To begin, set $a_1 = a$ and $b_1 = b$ and let $c_1 = \frac{a_1+b_1}{2}$ be the midpoint of the interval $[a_1, b_1] = [a, b]$.

- If $f(c_1) = 0$ then the root is $r = c_1$.
- If $f(c_1) \neq 0$ then either $f(a_1)f(c_1) < 0$ or $f(c_1)f(b_1) < 0$.
  - (i). If $f(a_1)f(c_1) < 0$ then r lies in $[a_1, c_1]$, and we squeeze the interval form the right and set $a_2 = a_1$ and $b_2 = c_1$, i.e. $[a_2, b_2] = [a_1, c_1]$.
  - (ii). If $f(c_1)f(b_1) < 0$ then r lies in $[c_1, b_1]$, and and we squeeze the interval form the left and set $a_2 = c_1$ and $b_2 = b_1$, i.e. $[a_2, b_2] = [c_1, b_1]$.
  - (iii). Compute $c_2 = \frac{a_2+b_2}{2}$ the midpoint of the interval $[a_2, b_2]$.
  - (iv). Then, we proceed in this way until we reach the nth interval $[a_n, b_n]$ and then compute its midpoint $c_n = \frac{a_n+b_n}{2}$.
- Finally, construct the interval $[a_{n+1}, b_{n+1}]$ which brackets the root and its midpoint $c_{n+1} = \frac{a_{n+}+b_{n+1}}{2}$ will be an approximation to the root r.

In the bisection method the initial interval $[a, b]$ is bisected and the interval width is decreased by half each time until we reach an arbitrarily small interval that brackets the root and we take the midpoint of this final interval as a reasonable approximation of the root r.

Remark 3. *(a). The interval $[a_{n+1}, b_{n+1}]$ is wide as half as the interval $[a_n, b_n]$ i.e. the width of each interval is as half as the width of the previous interval. Let $\{\ell_n\}$ be a sequence of widths of intervals $[a_n, b_n]$, i.e. $\ell_n = \frac{b_n - a_n}{2}$, $n = 1, 2, \cdots$. Hence $\lim_{n\to\infty} \ell_n = 0$, where ϵ is the preassigned value of the error (**tolerance**) i.e.*

$$|r_{n+1} - r_n| \leq \epsilon, \; n = 0, 1, \cdots . \tag{2.3}$$

(b). The sequence of left endpoints a_n, $n = 1, 2, \cdots$, is increasing and the sequence b_n, $n = 1, 2, \cdots$ of right endpoints is decreasing i.e.

$$a_0 \leq a_1 \leq \cdots \leq a_n \leq \cdots \leq r \leq \cdots \leq b_n \leq \cdots \leq b_1 \leq b_0. \tag{2.4}$$

Theorem 6 (Bisection Method Theorem). *Let $f \in C[a, b]$ such that $f(a)f(b) < 0$ and that there exists a number $r \in [a, b]$ such that $f(r) = 0$, and $\{c_n\}$ a sequence of the midpoints of intervals $[a_n, b_n]$ constructed by the bisection method, then the error in approximating the root r in the nth step is:*

$$|e_n| = |r - c_n| \leq \frac{b - a}{2^{n+1}}, \quad n = 0, 1, \cdots . \tag{2.5}$$

Hence, the sequence $\{c_n\}$ is convergent and its limit is the root r, i.e.

$$\lim_{n\to\infty} c_n = r. \tag{2.6}$$

The error bound in (2.5) can be used to evaluate the required predetermined accuracy of the method.

Proof. For Proof see the References or have a look in any standard numerical analysis text. □

Remark 4. • *The number N of repeated bisections required to compute the nth approximation (midpoint) c_n of the root r is:*

$$N = int\Big(\frac{ln(b - a) - ln(\epsilon)}{ln(2)}\Big). \tag{2.7}$$

The formula (2.7) is obtained from the error bound formula (2.5).

- *The width of the nth interval $[a_n, b_n]$ is:*

$$|b_n - a_n| = \frac{|b_0 - a_0|}{2^n}. \tag{2.8}$$

Example 10. *(a) Use bisection method to show that $f(x) = x\sin(x) - 1 = 0$ has a real root in $[0.5, 1.5]$. Compute eleven approximations (i.e. use $n = 10$) to the root.*

(b) Evaluate the number of computations N required to ensure that the error is less than the preassigned value (error bound) $\epsilon = 0.001$.

Solution:
(a) We start with initial interval $[a0, b0] = [0.5, 1.5]$ and compute $f(0.5) = -0.76028723$ and $f(1.5) = 0.49624248$. We notice that $f(a_0)$ and $f(b_0)$ have opposite signs and hence, there is a root in the interval $[0.5, 1.5]$. Compute the midpoint $c_0 = \frac{a_0+b_0}{2} = \frac{0.5+1.5}{2} = 1$ and $f(1) = -0.15852902$. The function changes sign on $[c_0, b_0] = [1, 1.5]$, so, we set $[a_1, b_1] = [c_0, b_0] = [1, 1.5]$, and compute the midpoint $c_1 = \frac{a_1+b_1}{2} = \frac{1+1.5}{2} = 1.25$ and $f(1.25) = 0.18623077$. Hence, the root lies in the interval $[a_1, c_1] = [1, 1.25]$. Set $[a_2, b_2] = [a_1, c_1] = [1, 1.25]$ and continue until we compute $c_{10} = 1.11376953125$. The details are explained in Table 2.1.

n	Left Endpoint a_n	Midpoint c_n	Right Endpoint b_n	Function Value $f(c_n)$
0	0.5	1	1.5	-0.15852902
1	1	1.25	1.5	0.18623077
2	1	1.125	1.25	0.01505104
3	1	1.0625	1.125	-0.07182663
4	1.0625	1.09375	1.125	-0.02836172
5	1.09375	1.109375	1.125	-0.00664277
6	1.109375	1.1171875	1.125	0.00420803
7	1.109375	1.11328125	1.1171875	-0.00121649
8	1.11328125	1.115234375	1.1171875	0.00149600
9	1.11328125	1.1142578125	1.115234375	0.00013981
10	1.11328125	1.11376953125	1.1142578125	-0.00053832

Table 2.1: Bisection Method Solution of Example 10

(b)

$$N = int\Big(\frac{ln(1.5-0.5) - ln(0.001)}{ln(2)}\Big) = int\Big(\frac{ln(1) - ln(0.001)}{ln(2)}\Big) =$$

$$int\Big(\frac{0-(-6.90775528)}{0.69314718}\Big) = int\Big(\frac{6.90775528}{0.69314718}\Big) = int(9.96578429) = 10.$$

2.2.2 False-Position Method

It also known as **regula falsi method**, it is similar to the bisection method in requiring that $f(a)$ and $f(b)$ have opposite signs. This method uses the abscissa of the point $(c, 0)$ at which the secant line called it SL joining the points $(a, f(a))$ and $(b, f(b))$ crosses the x-axis instead of using the midpoint of the interval as approximation of the zero of the function f as in the bisection method. To evaluate c, we need to compute the slope of line SL between the two points $(a, f(a))$ and $(b, f(b))$:

$$m = \frac{f(b) - f(a)}{b - a}.$$

Now, compute the slope of line SL between the two points $(c, f(c)) = (c, 0)$ and $(b, f(b))$:

$$m = \frac{f(b) - f(c)}{b - c} = \frac{f(b) - 0}{b - c} = \frac{f(b)}{b - c}.$$

By equating the two slopes, we obtain

$$\frac{f(b) - f(a)}{b - a} = \frac{f(b)}{b - c} \quad \Longrightarrow \quad c = b - \frac{f(b)(b - a)}{f(b) - f(a)}.$$

Now, we have the same possibilities as in the bisection method:

- If $f(c_0) = 0$ then the root is $r = c_0$.
- If $f(c_0) \neq 0$ then either $f(a_0)f(c_0) < 0$ or $f(c_0)f(b_0) < 0$.

  (i). If $f(a_0)f(c_0) < 0$ then r lies in $[a_0, c_0]$, and set $a_1 = a_0$ and $b_1 = c_0$, i.e. $[a_1, b_1] = [a_0, c_0]$.

  (ii). If $f(c_0)f(b_0) < 0$ then r lies in $[c_0, b_0]$, and set $a_1 = c_0$ and $b_1 = b_0$, i.e. $[a_1, b_1] = [c_0, b_0]$.

  (iii). Compute $c_1 = b_1 - \frac{f(b_1)(b_1 - a_1)}{f(b_1) - f(a_1)}$.

  (iv). Then, we proceed in this way until we reach the nth interval $[a_n, b_n]$ and then compute $c_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)}$.

The general formula of the false-position method is

$$c_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)}, \ n = 1, 2, \cdots, \tag{2.9}$$

start with an initial interval $[a_1, b_1] = [a, b]$ such that f has opposite signs on it, then the sequence $\{c_n\}_{n=1}^{\infty}$ of successive approximations of the root converges to the root r of the equation $f(x) = 0$. In general, the false-position method is faster than the bisection method. Note that the interval width $b_n - a_n$ is getting smaller as n gets larger but it is not necessarily to approaches zero. For example, if the curve of the function $y = f(x)$ is concave near the point $(r, 0)$ where the graph of the function crosses the x-axis, then one of the endpoints of the interval is fixed and the other endpoint moves to the root. The fixed endpoint is called the **stagnant endpoint**.

Example 11. *Show that* $f(x) = 2x^3 - x^2 + x - 1 = 0$ *has at least on root in* $[0, 1]$.

Solution:
Since $f(0) = -1$ and $f(1) = 1$, then Intermediate Value Theorem implies that this continuous function has a root in $[0, 1]$. Set $[a_0, b_0] = [0, 1]$ and compute $c_0 = b_0 - \frac{f(b_0)(b_0-a_0)}{f(b_0)-f(a_0)} = f(1) - \frac{f(1)(1-0)}{f(1)-f(0)} = 1 - \frac{1(1-0)}{1-(-1)} = 0.5$, also compute $f(c_0) = f(0.5) = -0.5$. Hence, the root lies in $[c_0, b_0]$ we squeeze from the left and set $a_1 = c_0 = 0.5$ and $b_1 = b_0 = 1$, to have $[a_1, b_1] = [0.5, 1]$. Now, compute the new approximation to the root $c_1 = b_1 - \frac{f(b_1)(b_1-a_1)}{f(b_1)-f(a_1)} = f(1) - \frac{f(1)(1-0.5)}{f(1)-f(0.5)} = 1 - \frac{1(1-0.5)}{1-(-0.5)} = 2/3 \approx 0.66666667$, $f(c_1) = -0.18518519$. The function has opposite signs on the interval $[c_1, b_1]$, set $a_2 = c_1 = 0.66666667$ and $b_2 = b_1 = 1$. so we have $[a_2, b_2] = [0.66666667, 1]$. Continue this we and stop at $c_7 = 0.73895443$.

n	a_n	c_n	b_n	$f(c_n)$
0	0	0.5	1	-0.5
1	0.5	0.66666667	1	-0.18518519
2	0.66666667	0.71875000	1	-0.05523681
3	0.71875000	0.73347215	1	-0.01532051
4	0.73347215	0.73749388	1	-0.00416160
5	0.73749388	0.73858180	1	-0.00112399
6	0.73858180	0.73887530	1	-0.00030311
7	0.73887530	0.73895443	1	-0.00008171

Table 2.2: False Position Method Solution of Example 11

2.3 Open Methods

In these methods we do not need to have an interval around the root to start solving the nonlinear equation $f(x) = 0$. We only need a sufficiently closed approximation to the root hence, the name **open methods** or **locally convergent methods**. The starting approximation is called the **starting**

value, **initial approximation**, **initial guess** or a **seed**. The bracketing methods are slow compared with open methods which are faster and have better convergence properties. In this section we study the fixed-point method, Newton's method and the secant method as examples of open methods. In the literature, the open methods are also known as **slope methods** since these methods use the slope of the tangent line of the graph of the function near the point $(r, 0)$ to derive a mathematical formula for computing the next iterations.

2.3.1 Fixed Point Method

It is an important and widely used method for finding the roots of nonlinear problems. This method relies on the iteration principle. Iteration is a fundamental concept in computer sciences and numerical analysis and is used for solving a wide variety of problems. Iteration and fixed point methods have many applications in *fractals (fractal geometry)*, *chaos theory* and *dynamical systems.*There is a strong connection between *root-finding problems* and *fixed point problems*, and in this section, we use fixed point problems to solve root-finding problems.

Definition 17 (Fixed Point)**.** *The number r is called a **fixed point** of the function g if $r = g(r)$.*

We start by transforming the root-finding problem $f(x) = 0$ to a fixed point problem $x = g(x)$ by algebraic manipulations. There are more than one way of rearranging $f(x) = 0$ into an equivalent form $x = g(x)$. Note that if r is a zero of the function f (i.e. r is a root of the equation $f(r) = 0$) then $r = g(r)$ i.e. r is a fixed point of the function g. Conversely, if g has a fixed point at r then the function $f(x) = x - g(x)$ has a zero at r. Geometrically, the fixed points of a function $y = g(x)$ are the points of intersection of its curve with the straight line $y = x$.

Example 12. *Find the fixed points of the function $g(x) = 2 - x^2$ and verify that they are the solutions to the equation $f(x) = x - g(x) = 0$.*

Solution : The fixed points of g are the points satisfying the fixed point equation $x = g(x)$, so intersect the graph of $y = g(x)$ with the graph of the straight line $y = x$

$$x = g(x) = 2 - x^2,$$

which implies that

$$-x^2 - x + 2 = -(x^2 + x - 2) = -(x - 1)(x + 2) = 0.$$

So, either $(x-1)=0$ implies $x=1$ or $(x+2)=0$ implies $x=-2$. Hence, the fixed points are $x=1$ and $x=-2$. We notice that these fixed points are the same the zeros of $f(x)=x-g(x)=-(x^2+x-2)=-(x-1)(x+2)=0$.

Definition 18 (**Fixed Point Iteration**). *The iteration $r_{n+1}=g(r_n)$, $n=0,1,\cdots$, obtained by using fixed point formula $x=g(x)$ is called a* ***fixed point iteration*** *or* ***functional iteration****. The numbers $r_n, n=0,1,\cdots$, are called* ***iterates*** *or* ***iterations***

In short, in the fixed point method we start with starting value r_0 and by using the repeated substitutions in the rule or function $g(x)$ we compute the successive or consecutive terms. For this reason the fixed-point method sometimes is referred to as **repeated substitution method**.

Theorem 7 (**Convergence of the Fixed Point Iteration**). *Let g is a continuous function and $\{r_n\}_{n=0}^{\infty}$ is a sequence of iterates generated by the fixed-point iteration rule $r_{n+1}=g(r_n), n=0,1,\cdots$. If the sequence $\{r_n\}_{n=0}^{\infty}$ is convergent and $\lim_{n\to\infty} r_n = r$, then r is a fixed point of the function $g(x)$.*

Theorem 8 (**Existence and Uniqueness of the Fixed Point**). *Assume that $g \in C[a,b]$.*

1. *If $g(x) \in [a,b]$ for all $x \in [a,b]$, then g has a at least one fixed point r in $[a,b]$.*

2. *If also, $g'(x)$ existed and defined on (a,b) and there exists a positive constant $K<1$ such that $|g'(x)| \le K < 1$, for all $x \in (a,b)$, then g has a unique fixed point r in $[a,b]$.*

3. *If g satisfies the conditions (1) and (2) , then for any number r_0 in $[a,b]$ the sequence $\{r_n\}_{n=0}^{\infty}$ of iterations generated by fixed point iteration $r_{n+1}=g(r_n)$, $n=0,1,\cdots$, converges to the unique fixed point r in $[a,b]$.*

Theorem 9 (**Fixed Point Theorem**). *Assume that*

(i) *$g, g' \in C[a,b]$.*

(ii) *K is a positive constant.*

(iii) *$r_0 \in (a,b)$ is an initial approximation.*

(iv) *$g(x) \in [a,b]$ for all $x \in [a,b]$.*

1. *If $|g'(x)| \leq K < 1$ for all $x \in (a, b)$, then the sequence of iterates $\{r_n\}_{n=0}^{\infty}$ converges to the unique fixed point $r \in [a, b]$ and r is called an* ***attractive fixed point****.*

2. *If $|g'(x)| > 1$ for all $x \in (a, b)$, then the sequence of iterates $\{r_n\}_{n=0}^{\infty}$ diverges and will not converge to the fixed point $r \in [a, b]$ and r is called a* ***repelling fixed point*** *and the iteration exhibits local divergence.*

Corollary 1 (**Fixed Point Iteration Error Bounds**). *If g satisfies the hypotheses of Fixed Point Theorem, then the error bounds for approximating r using r_n are given by*

$$|r - r_n| \leq K^n \max |r - r_0|, \tag{2.10}$$

and

$$|r - r_n| \leq \frac{K^n}{1 - K} \max |r_1 - r_0|, \; \textit{for all}\, n \geq 1. \tag{2.11}$$

Example 13. *Use the fixed point method to find the zero of the function $f(x) = x^3 - 3x^2 + 2$ in $[0, 2]$, start with $r_0 = 1.5$.*

Solution: There are many possibilities to write $f(x) = 0$ as a fixed point form $x = g(x)$ using mathematical manipulations.

(1) $x = g_1(x) = x - x^3 + 3x^2 - 2.$

(2) $x = g_2(x) = \left(\frac{x^3+2}{3}\right)^{1/2}.$

(3) $x = g_3(x) = -\left(\frac{x^3+2}{3}\right)^{1/2}.$

(4) $x = g_4(x) = \left(\frac{2}{3-x}\right)^{1/2}.$

(5) $x = g_5(x) = \frac{-2}{x\left(x-3\right)}.$

(6) $x = g_6(x) = \left(3x^2 - 2\right)^{1/3}.$

(7) $x = g_7(x) = \left(3x - \frac{2}{x}\right)^{1/2}.$

For example, to obtain $g_1(x)$ just add x to both sides of the equation $-f(x) = 0$ and this is the simplest way to write the problem as a fixed point form

$$-f(x) = 0, \quad -x^3 + 3x^2 - 2 = 0, \quad \text{so} \quad x = x - x^3 + 3x^2 - 2 = g_1(x).$$

Also, $g_2(x)$ and $g_3(x)$ can be obtained as follows:

$$x^3 - 3x^2 + 2 = 0, \quad \text{so} \quad 3x^2 = x^3 + 2, \quad \text{and} \quad x^2 = \frac{x^3 + 2}{3},$$

implies that

$$x = \pm\Big(\frac{x^3+2}{3}\Big)^{1/2}, \quad \text{so} \quad g_2(x) = \Big(\frac{x^3+2}{3}\Big)^{1/2}, \text{ and } \quad g_3(x) = -\Big(\frac{x^3+2}{3}\Big)^{1/2}.$$

Note that it is important to check that the fixed point of each derived function g is a solution to the problem $f(x) = 0$. For example, because the solution is positive and lies between 0 and 2, so we choose the positive function $g_2(x)$, since the negative function $g_3(x)$ is not a choice here. The results are outlined in Tables 2.3 and 2.4 below.

n	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	$g_5(x)$
0	1.5	1.5	1.5	1.5	1.5
1	2.875	1.33853153	-1.33853153	1.15470054	0.88888889
2	1.90820313	1.21081272	$0 - 0.36i$	1.04107393	1.06578947
3	3.88369668	1.12177435		1.01042940	0.97018561
4	-11.44518863	1.06639827		1.00261759	1.01559099
5	$1.8788e+03$	1.03484519		1.00065504	0.99238449
6	$-6.6210e+09$	1.01787695		1.00016380	1.00385153
7	$2.9025e+29$	1.00905819		1.00004095	0.99808533
8		1.00455985		1.00001024	1.00096009
9		1.00228772		1.00000256	0.99952065
10		1.00114582		1.00000064	1.00023985
11		1.00057340		1.00000016	0.99988012
12				1.00000004	1.00005995
13				1.00000001	0.99997003
14				1.00000000	1.00001499
15					0.99999251
16					1.00000375
17					0.99999813
18					1.00000094
19					0.99999953
20					1.00000024
21					0.99999988

Table 2.3: Fixed Point Method Solution of Example 13

n	$g_6(x)$	$g_7(x)$
0	1.5	1.5
1	1.68098770	1.77951304
2	1.86406700	2.05295790
3	2.03474597	2.27698696
4	2.18422416	2.43979654
5	2.30913228	2.54944095
6	2.40992853	2.61989258
7	2.48919424	2.66388582
8	2.55035309	2.69088731
9	2.59688496	2.70728880
10	2.63192563	2.71718971
11	2.65811433	2.72314423
12	2.67757909	2.72671736
13	2.69198765	2.72885862
14	2.70262171	2.73014079
15	2.71045296	2.73090817
16	2.71621092	2.73136731
17	2.72043954	2.73164198
18	2.72354236	2.73180628
19	2.72581768	2.73190456
20	2.72748542	2.73196334
21	2.72870741	2.73199849

Table 2.4: Fixed Point Method Solution of Example 13

Remark 5.

- *The sequence of iterations $\{r_n\}_{n=0}^{\infty}$ generated by the fixed-point iteration rule $r_{n+1} = g(r_n), n = 0, 1, \cdots$ is either convergent or divergent.*
- *If the sequence of iterations is divergent, then we may have different types of divergence behaviour such as **monotone** or **oscillating** or **cyclic(repeated)**.*
- *If the sequence of iterations is convergent, it may converge to another fixed point not the one we are interested in (may be it is not in the problem domain or domain of interest of the function g).*
- *If the sequence of iterations is convergent, the convergence may be **monotone** or **oscillating**.*
- *Note that the Fixed Point Theorem does not explain what is the case if $|g'(x)| = 1$. In this case, the sequence of iterations also is either*

convergent or divergent and this depends on the closeness of the starting value r_0 to the fixed point r.

2.3.2 Newton's Method

Newton's method is also known as **Newton-Raphson method** is one of the most powerful and efficient numerical methods for *root-finding problems.* It is well-known and popular method and there are several variants and extensions of this method. There are more than one approach for deriving this method such as the *graphical technique* and *Taylor series technique*, and here we use both of them, we start with Taylor series approach. Let $f, f^{'}, f^{''}$ are continuous functions on the interval $[a.b]$ (i.e. $f \in C^2[a,b]$). Let $r_0 \in [a,b]$ be an approximation to the zero r of the function f such that $f^{'}(r_0) \neq 0$ and r_0 is "sufficiently close to r i.e. $|r - r_0|$ is relatively small". Let's start with first Taylor polynomial of $f(x)$ expanded about the initial approximation r_0 and compute it at $x = r$:

$$f(r) = f(r_0) + (r - r_0)f^{'}(r_0) + \frac{(r - r_0)^2}{2} f^{''}(\xi(r)),$$

where $\xi(r)$ lies between r_0 and r. Using the fact that $f(r) = 0$, this leads to

$$0 = f(r_0) + (r - r_0)f^{'}(r_0) + \frac{(r - r_0)^2}{2} f^{''}(\xi(r)).$$

Since $|r - r_0|$ is small then $(r - r_0)^2$ is much smaller, so we can neglect the third term in Taylor's expansion which contains this term (quadratic power term) to have

$$0 = f(r_0) + (r - r_0)f^{'}(r_0).$$

Solving for r yields

$$r = r_0 - \frac{f(r_0)}{f^{'}(r_0)}.$$

To, proceed, set $r = r_1$ in the Newton's formula to compute r_1 by using the known value r_0

$$r_1 = r_0 - \frac{f(r_0)}{f^{'}(r_0)},$$

and then we compute r_2 using the known value r_1

$$r_2 = r_1 - \frac{f(r_1)}{f^{'}(r_1)},$$

and by following the same fashion, we compute r_3, r_4 and so on. The general or *nth* form of Newton's method is:

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}, \quad n = 1, 2, \cdots . \tag{2.12}$$

This is called **Newton's formula** or **Newton-Raphson formula**.

The graphical approach: Let r_0 be an initial guess of the solution r of the equation $f(x) = 0$, then the curve of the function f crosses the x-axis at the point $(r, 0)$. The point $(r_0, f(r_0)$ lies on the curve near the point $(r, 0)$. The tangent line to the curve of f at the point $(r_0, f(r_0)$ intersects the x-axis at the point $(r_1, 0)$, then r_1 is the new approximation of the root and is closer to the root than r_0. The slope of the tangent line L joining the points $(r_0, f(r_0)$ and $(r_1, 0)$ is:

$$m = \frac{0 - f(r_0)}{r_1 - r_0}, \tag{2.13}$$

and also, we have

$$m = f^{'}(r_0). \tag{2.14}$$

Equating the two slopes in (2.13) and (2.14), we get

$$f^{'}(r_0) = \frac{-f(r_0)}{r_1 - r_0}, \tag{2.15}$$

solving for the new approximation r_1 we have

$$r_1 = r_0 - \frac{f(r_0)}{f'(r_0)}. \tag{2.16}$$

By iterating (2.16), we obtain the Newton's formula in (2.12).

Theorem 10 (Convergence of the Newton's Method)**.** *Assume that the sequence of approximations* $\{r_n\}_{n=1}^{\infty}$ *of the root of the nonlinear equation* $(x) = 0$ *produced by the Newton's iterative formula* (2.12) *converges to the root* r*. Then, if* r *is a simple root, the convergence is quadratic and the error bound is:*

$$|E_n| = |r_n - r_{n-1}| \approx \frac{f^{''}(r)}{2f^{'}(r)}|E_{n-1}|^2, \textit{ for sufficiently large } n. \tag{2.17}$$

If r is a multiple root of order $M > 1$, then the convergence is linear and the error bound is:

$$|E_n| = |r_n - r_{n-1}| \approx \frac{M-1}{M}|E_{n-1}|, \text{ for sufficiently large } n. \tag{2.18}$$

Note that the asymptotic error constants in the case of quadratic and linear convergence are $A = \frac{f''(r)}{2f'(r)}$ and $A = \frac{M-1}{M}$, respectively.

Example 14. *Use Newton's method to find the positive root accurate to within 10^{-5} for $f(x) = 3x - e^x = 0$. Start with the initial guess $r_0 = 1.5$.*

Solution: Start by finding the derivative of $f(x)$:

$f(x) = 3x - e^x,\ f'(x) = 3 - e^x,\ r_0 = 1.5, f(r_0) = 0.01831093, f'(r_0) = -1.48168907,$

so, the Newton-Raphson iteration formula for this problem is:

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})} = r_{n-1} - \frac{3r_{n-1} - e^{r_{n-1}}}{3 - e^{r_{n-1}}},\ n = 1, 2, \cdots, .$$

Computing r_1 by using the known value r_0,

$$r_1 = r_0 - \frac{f(r_0)}{f'(r_0)} = r_0 - \frac{3r_0 - e^{r_0}}{3 - e^{r_0}} = 1.5 - \frac{3(1.5) - e^{1.5}}{3 - e^{1.5}} = 1.51235815.$$

Now, compute $f(r_1)$ and $f'(r_1)$,

$$f(r_1) = 3r_1 - e^{r_1} = -0.00034364, \quad f'(r_1) = 3 - e^{r_1} = -1.53741808.$$

Next, we compute r_2,

$$r_2 = r_1 - \frac{f(r_1)}{f'(r_1)} = r_1 - \frac{3r_1 - e^{r_1}}{3 - e^{r_1}} = 1.51213463,\ f(r_2) = -1.1e{-}07,\ f'(r_2) = -1.53640399.$$

A summary of the computations is given in Table 2.5.

n	r_n	$f(r_n)$	$f'(r_n)$
0	1.50000000	0.01831093	−1.48168907
1	1.51235815	−0.00034364	−1.53741808
2	1.51213463	−0.00000011	−1.53640399
3	1.51213455	−0.00000000	−1.53640365
4	1.51213455	0	−1.53640365

Table 2.5: Newton's Method Solution of Example 14

Remark 6. *One of the main drawbacks of the Newton's method is the possibility of division by zero when $f^{'}(r_{n-1}) = 0$ in (2.12). In this case as a remedy we compute $f(r_{n-1})$ and if it is sufficiently close to zero, then we consider r_{n-1} is a reasonable approximation to the root r. Also, we have another problem when $f^{'}(r_{n-1}) \approx 0$, i.e. when the tangent line to the curve of f at the point $(r_{n-1}, f(r_{n-1}))$ is nearly horizontal, then dividing by a very small number results in meaningless computations.*

Newton's Method for Finding the *nth* Roots

We start with square roots. Let $B > 0$ a real number and r_0 be an initial approximation to $\sqrt{B}$. Our goal is to find a square root of a number B. Let $x = \sqrt{B}$, so $x^2 = B$, which implies that $x^2 - B = 0$, define $f(x) = x^2 - B = 0$. Note that this equation has two roots $x = \pm\sqrt{B}$. Now, find the derivative of f, $f'(x) = 2x$ and use the Newton's fixed point formula

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

Now, using Newton's iteration formula

$$r_{n+1} = \frac{r_n + \frac{B}{r_n}}{2}, \quad n = 0, 1, \cdots .$$

The sequence of iterations $\{r_n\}_{n=0}^{\infty}$ converges to $\sqrt{B}$. Note that in computing the square root of B, we do not need to evaluate f and f' and this makes the calculations easier and faster since we just need the values of the iterates r_n, $n = 0, 1, \cdots$.

Example 15. *Use Newton's square-root algorithm to find $\sqrt{3}$, use $r_0 = 1$.*

Solution: Starting with $r_0 = 1$ when $n = 0$, we have

$$r_1 = \frac{r_0 + \frac{3}{r_0}}{2} = \frac{1 + 3}{2} = 2.$$

For $n = 1$,

$$r_2 = \frac{r_1 + \frac{3}{r_1}}{2} = \frac{2 + \frac{3}{2}}{2} = 1.75.$$

$$n = 2,\ r_3 = \frac{r_2 + \frac{3}{r_2}}{2} = \frac{1.75 + \frac{3}{1.75}}{2} = 1.732142857142857.$$

$$n = 3,\ r_4 = \frac{r_3 + \frac{3}{r_3}}{2} = 1.732050810014727.$$

A summary of results is given in Table 2.6

n	r_n
0	1
1	2
2	1.75
3	1.732142857142857
4	1.732050810014727
5	1.732050807568877
6	1.732050807568877

Table 2.6: Newton's Method Solution of Example 15

2.3.3 Secant Method

Newton's method is a very powerful and efficient technique for solving root-finding problems but one of the drawbacks of the method is the need of derivative evaluations of f at the approximations $r_n, n \geq 0$, and this is not a trivial task. To avoid this we introduce **secant method** which is a variation of Newton's method. Secant method is similar to the false position method but it differs in the way of choosing the succeeding terms.We start with two initial points $(r_0, f(r_0))$ and $(r_1, f(r_1))$ near the point $(r, 0)$, where r is the root of equation $f(x) = 0$. Define the point $(r_2, 0)$ to be the point of intersection of the secant line joining the points $(r_0, f(r_0))$ and $(r_1, f(r_1))$ with the x-axis. Geometrically, the abscissa of the point of intersection r_2 is closer to the root r than to either r_0 and r_1.

The slope of the secant line relating these three points $(r_0, f(r_0))$, $(r_1, f(r_1))$ and $(r_2, f(r_2))$ is:

$$m = \frac{f(r_1) - f(r_0)}{r_1 - r_0} \quad \text{and} \quad m = \frac{f(r_2) - f(r_1)}{r_2 - r_1} = \frac{0 - f(r_1)}{r_2 - r_1} = \frac{-f(r_1)}{r_2 - r_1}.$$

Equating the two values of the slope, we have

$$\frac{f(r_1) - f(r_0)}{r_1 - r_0} = \frac{-f(r_1)}{r_2 - r_1}.$$

Solving slope's equation for r_2, we obtain

$$r_2 = r_1 - \frac{f(r_1)(r_1 - r_0)}{f(r_1) - f(r_0)}.$$

So, the general form of the secant method is:

$$r_{n+2} = r_{n+1} - \frac{f(r_{n+1})(r_{n+1} - r_n)}{f(r_{n+1}) - f(r_n)}, \; n = 0, 1, \cdots, .$$

Example 16. *Find the root of equation $x - \cos(x) = 0$ using the secant method and the two initial guesses $r_0 = 0.5$ and $r_1 = 0.6$.*

Solution: To compute the first approximation r_2, we need to compute $f(r_0)$ and $f(r_1)$

$$f(r_0) = f(0.5) = 0.5 - \cos(0.5) = -0.377582560000000,$$

$$f(r_1) = f(0.6) = 0.6 - \cos(0.6) = -0.225335610000000.$$

So,

$$r_2 = r_1 - \frac{f(r_1)(r_1 - r_0)}{f(r_1) - f(r_0)} = 0.6 - \frac{f(0.6)(0.6 - 0.5)}{f(0.6) - f(0.5)} = 0.748006655882730,$$

$$f(r_2) = r_2 - \cos(r_2) = 0.014960500949714.$$

Now, we compute the next approximation r_3,

$$r_3 = r_2 - \frac{f(r_2)(r_2 - r_1)}{f(r_2) - f(r_1)} = 0.738791967963291, \ f(r_3) = -0.000490613128583.$$

Continuing until satisfying the required accuracy. A summary of the calculations is given in Table 2.7.

n	r_n	$f(r_n)$
0	0.500000000000000	−0.377582560000000
1	0.600000000000000	−0.225335610000000
2	0.748006655882730	0.014960500949714
3	0.738791967963291	−0.000490613128583
4	0.739084558312839	−0.000000962163319
5	0.739085133252381	0.000000000062293
6	0.739085133215161	0
7	0.739085133215161	0

Table 2.7: Secant Method Solution of Example 16

2.4 Acceleration of Iterative Methods

The linear convergence a sequence $\{r_n\}$ to the limit r such as the sequences of the fixed point iterations can be accelerated by using some techniques such as Aitken's Δ^2 method (Aitken's acceleration) and Steffensen's method. For more details see references [4, 16] and the references therein.

2.4.1 Modified Newton's Methods

Newton's method is a fixed point method since it can be written as

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

and in iterative way

$$r_n = g(r_{n-1}) = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}, \quad n = 1, 2, \cdots,$$

and this is called **Newton-Raphson iteration formula** or simply **Newton's iteration**. The convergence of Newton's method can be modified to accelerate its rate of convergence at the root $x = r$ of order $M > 1$

$$r_n = r_{n-1} - \frac{f(r_{n-1})f'(r_{n-1})}{(f'(r_{n-1}))^2 - f(r_{n-1})f''(r_{n-1})}, \quad n = 1, 2, \cdots .$$

This formula is called a **modified Newton's method**.

Also, Newton's method can be accelerated in an another way.

Theorem 11 (Acceleration of Newton's Iteration)**.** *Assume that Newton's method produces a linearly convergent sequence to the root $x = r$ of order $M > 1$. Then Newton's iteration formula*

$$r_n = r_{n-1} - \frac{Mf(r_{n-1})}{f'(r_{n-1})}, \quad n = 1, 2, \cdots,$$

produces a quadratically convergent sequence $\{r_n\}_{n=0}^{\infty}$ to the root $x = r$.

Example 17. *Show that $r = 1$ is a double zero (double root) of $f(x) = -x^3 + 3x - 2 = 0$. Start with $r_0 = 1.25$ as an initial guess of r and compare the performance of Newton's method and accelerated Newton's method for solving $f(x) = 0$.*

Solution : Since $r = 1$ is a double root then $M = 2$, so the accelerated Newton's method becomes

$$r_n = r_{n-1} - \frac{2f(r_{n-1})}{f'(r_{n-1})} = r_{n-1} - \frac{2(-r_{n-1}^3 + 3r_{n-1} - 2)}{-3r_{n-1}^2 + 3}, \quad n = 1, 2, \cdots,$$

or

$$r_n = r_{n-1} - \frac{-2r_{n-1}^3 + 6r_{n-1} - 4}{-3r_{n-1}^2 + 3}, \quad n = 1, 2, \cdots .$$

Start by computing r_1

$$r_1 = r_0 - \frac{-2r_0^3 + 6r_0 - 4}{-3r_0^2 + 3} = 1.25 - \frac{-2(1.25)^3 + 6(1.25) - 4}{-3(1.25)^2 + 3} = 1.00925926.$$

Table 2.8 compares the performance of both methods.

n	Newton's Method	Accelerated Newton's Method
0	1.25	1.25
1	1.12962963	1.00925926
2	1.06612990	1.00001422
3	1.03341772	1.00001422
4	1.01680039	1.00000000
5	1.00842352	
6	1.00421765	
7	1.00211030	
8	1.00105552	
9	1.00052785	
10	1.00026395	
11	1.00013198	
12	1.00006599	
13	1.00003300	
14	1.00001650	
15	1.00000825	
16	1.00000413	
17	1.00000207	
18	1.00000104	
19	1.00000052	
20	1.00000026	
21	1.00000013	
22	1.00000007	
23	1.00000004	
24	1.00000002	
24	1.00000001	
25	1.00000001	

Table 2.8: Newton's and Accelerated Newton's Methods Solutions of Example 17

2.5 Computing Roots of Polynomials

Computing roots of polynomials has important applications in different areas of mathematics and other sciences.

Definition 19 (*nth* Degree Polynomial)**.** *A polynomial of degree n has the general form*

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0,$$

where the coefficients a_i, $i = 0, 1, \cdots, n$, are real numbers (constants) and $a_n \neq 0$. The nth degree polynomial $P(x)$ is sometimes referred to as $P_n(x)$, and also named algebraic polynomial.

Note that the zero function $P(x) = 0$ is a polynomial but has no degree.There are several techniques for finding zeros of polynomials in the literature such as **Müller method**, **Laguerre's method**, **Bairstow method**, **Brent's method** and **Jenkins-Traub method**, and these methods are beyond the scope of this lecture notes and interested readers can see the references.

2.6 Numerical Solutions of Systems of Nonlinear Equations

Some phenomena in nature are modelled by systems of N nonlinear equations in N unknowns. These systems can be handled firstly by linearising them and then solving them in repeated way. Newton's method for a single nonlinear equation follows the same approach and can be easily extended for solving a system of nonlinear equations.

The general form of a system of N nonlinear equations in N unknowns x_i is:

$$\begin{aligned}
f_1(x_1, x_2, \cdots, x_N) &= 0 \\
f_2(x_1, x_2, \cdots, x_N) &= 0 \\
&\vdots \\
f_N(x_1, x_2, \cdots, x_N) &= 0.
\end{aligned}$$

Using vector notation, this system can be written in this concise form

$$F(\mathbf{X}) = 0,$$

where

$$\begin{aligned} F &= [f_1, f_2, \cdots, f_N]^T \\ X &= [x_1, x_2, \cdots, x_N]^T. \end{aligned}$$

Newton's formula for a single nonlinear equation can be extended to a system of nonlinear equations in the following form

$$X^{(k+1)} = X^{(k)} - [F^{'}(X^{(k)})]^{-1} F(X^{(k)}),$$

where $F^{'}(X^{(k)})$ is the **Jacobian matrix** which will be defined below. It contains the partial derivatives of F evaluated at $X^{(k)} = [x_1^{(k)}, x_2^{(k)}, \cdots, x_N^{(k)}]^T$. The above-mentioned formula is similar to the Newton's formula for a single nonlinear equation except that the derivative appeared in the numerator as an inverse of the Jacobian matrix. In practice, the inverse will not be computed since this is impractical because its computational cost and instead we will solve a related linear system.

The method will explained by solving a system of three nonlinear equations

$$\begin{aligned} f_1(x_1, x_2, x_3) &= 0 \\ f_2(x_1, x_2, x_3) &= 0 \\ f_3(x_1, x_2, x_3) &= 0. \end{aligned}$$

The Taylor series expansion in three variables x_1, x_2, x_3:

$$f_i(x_1 + h_1, x_2 + h_2, x_3 + h_3) = f_i(x_1, x_2, x_3) + h_1 \frac{\partial f_i}{\partial x_1} + h_2 \frac{\partial f_i}{\partial x_2} + h_3 \frac{\partial f_i}{\partial x_3} + \cdots,$$

where the partial derivatives are evaluated at the point (x_1, x_2, x_3). We consider just the linear terms in step sizes h_i for $i = 1, 2, 3$. Assume that we have in vector notation

$$0 \approx F(X^{(0)} + H^{(0)}) \approx F(X^{(0)}) + F^{'}(X^{(0)}) H^{(0)},$$

where $F^{'}(X^{(0)})$ is the **Jacobian matrix** at the initial guess $X^{(0)} = (x_1^0, x_2^0, x_3^0)$,

$$F^{'}(X^{(0)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix},$$

where the partial derivatives are evaluated as follows:

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_i(X^{(0)})}{\partial x_j}, i, j = 1, 2, 3.$$

If the Jacobian matrix $F^{'}(X^{(0)})$ is non singular i.e its inverse is existed, then solving for H, we find

$$H^{(0)} = -[F^{'}(X^{(0)})]^{-1}F(X^{(0)}).$$

The next iteration after correction $X^{(1)} = X^{(0)} + H^{(0)}$ is closer to the root than $X^{(0)}$. Hence, Newton's formula for the first iteration is

$$X^{(1)} = X^{(0)} - [F^{'}(X^{(0)})]^{-1}F(X^{(0)}).$$

Consequently, the general form of Newton's method for solving the nonlinear system is:

$$X^{(k+1)} = X^{(k)} - [F^{'}(X^{(k)})]^{-1}F(X^{(k)}), k = 0, 1, \dots .$$

To avoid computing the inverse of the Jacobian matrix at each iteration, we instead resort to solving the **Jacobian linear systems**

$$[F^{'}(X^{(k)})]H^{(k)} = -F(X^{(k)}), k = 0, 1, \dots ,$$

Hence, the next Newton's iteration is computed using the formula

$$X^{(k+1)} = X^{(k)} + H^{(k)}, k = 0, 1, \dots .$$

Example 18. *Solve the following nonlinear system using Newton's method. Start with the initial guess $X^{(0)} = (x_1^0 = 1, x_2^0 = 0, x_3^0 = 0)$. The exact solution to this system is $X = (x_1 = 0, x_2 = 1, x_3 = 1)$.*

$$\begin{aligned} x_1 + x_2 + x_3 &= 2 \\ x_1^2 + x_2^2 + x_3^2 &= 2 \\ e^{x_1} + x_1x_2 - x_1x_3 &= 1. \end{aligned}$$

Solution: We compute the Jacobian matrix

$$F^{'}(X) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2x_1 & 2x_2 & 2x_3 \\ e^{x_1} + x_2 - x_3 & x_1 & -x_1 \end{bmatrix}.$$

The Jacobian matrix at the initial guess $X^{(0)}$

$$F^{'}(X^{(0)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2x_1^0 & 2x_2^0 & 2x_3^0 \\ e^{x_1^{(0)}} + x_2^{(0)} - x_3^{(0)} & x_1^{(0)} & -x_1^{(0)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 0 \\ e^1 & 1 & -1 \end{bmatrix}.$$

Now, solving the Jacobian linear system for $H^{(0)}$

$$[F^{'}(X^{(0)})]H^{(0)} = -F(X^{(0)}),$$

we get

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 0 \\ e^1 & 1 & -1 \end{bmatrix} \begin{bmatrix} h_1^0 \\ h_2^0 \\ h_3^0 \end{bmatrix} = - \begin{bmatrix} -1 \\ -1 \\ e^1 - 1 \end{bmatrix},$$

implies that

$$H^{(0)} = \begin{bmatrix} 0.5000 \\ -1.2887 \\ 1.7887 \end{bmatrix}.$$

Hence,

$$X^{(1)} = X^{(0)} + H^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.5000 \\ -1.2887 \\ 1.7887 \end{bmatrix} = \begin{bmatrix} 1.5000 \\ -1.2887 \\ 1.7887 \end{bmatrix}.$$

Now, compute the Jacobian matrix for $X^{(1)}$.

$$F^{'}(X^{(1)}) = \begin{bmatrix} 1 & 1 & 1 \\ 2x_1^1 & 2x_2^1 & 2x_3^1 \\ e^{x_1^{(1)}} + x_2^{(1)} - x_3^{(1)} & x_1^{(1)} & -x_1^{(1)} \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 \\ 3.0000 & -2.5774 & 3.5774 \\ 1.4043 & 1.5000 & -1.5000 \end{bmatrix}.$$

$$F(X^{(1)}) = \begin{bmatrix} f_1(x_1,x_2,x_3) \\ f_2(x_1,x_2,x_3) \\ f_3(x_1,x_2,x_3) \end{bmatrix} = \begin{bmatrix} 0 \\ 5.1102 \\ -1.1344 \end{bmatrix}.$$

Solving for $H^{(1)}$, we have

$$\begin{bmatrix} 1.0000 & 1.0000 & 1.0000 \\ 3.0000 & -2.5774 & 3.5774 \\ 1.4043 & 1.5000 & -1.5000 \end{bmatrix} \begin{bmatrix} h_1^1 \\ h_2^1 \\ h_3^1 \end{bmatrix} = - \begin{bmatrix} 0 \\ 5.1102 \\ -1.1344 \end{bmatrix},$$

implies

$$H^{(1)} = \begin{bmatrix} -0.5172 \\ 0.8788 \\ -0.3616 \end{bmatrix}.$$

The next approximation is

$$X^{(2)} = X^{(1)} + H^{(1)} = \begin{bmatrix} 1.5000 \\ -1.2887 \\ 1.7887 \end{bmatrix} + \begin{bmatrix} -0.5172 \\ 0.8788 \\ -0.3616 \end{bmatrix} = \begin{bmatrix} 0.9828 \\ -0.4099 \\ 1.4271 \end{bmatrix}.$$

Continuing as before, until we reach the required results.

Remark 7 (Hybrid Methods). *The global methods are guaranteed to converge to the root of the problem if given an initial interval such that the function changes sign on this interval, but these methods are slow and have linear convergence rates. The local methods are faster but it is not guaranteed to converge to the root unless we start sufficiently close to the root, and these methods have higher order convergence rates. Hence, to make balance between the good features in both methods, there are some methods start few steps with closed methods to guarantee the convergence and then move to open methods to speed up the convergence, these methods are called the* ***hybrid methods****.*

Exercises

Exercise 12. *Solve Example 11 using the bisection method and compare the solution with false position method's solution of the same problem.*

Exercise 13. *Repeat solving Example 10 using the false position method and compare the results with the solution of the bisection method for the same problem.*

Exercise 14. *Find the solution to the equation* $e^x - x - 1 = 0$ *accurate to six decimal places (i.e.* $\epsilon = 0.0000001$*) using Newton's and modified Newton's methods. Start with* $r_0 = 0.6$*. Compare the results of both methods.*

Exercise 15. *Use the secant method to find the solution accurate to within* 10^{-5} *to the following problem* $x\sin(x) - 1 = 0, \quad 0 \leq x \leq 2$.

Exercise 16. *Use the fixed point method to locate the root of* $f(x) = x - e^{-x} = 0$*, start with an initial guess of* $x = 0.1$.

Exercise 17. *Let* $f(x) = x^2 - 5$ *and* $r_0 = 1.5$*. Use bisection, false position, secant, fixed point, Newton's and modified Newton's methods to find* r_7 *the approximation to the positive root* $r = \sqrt{5}$.

Exercise 18. *Use modified (accelerated) Newton's method to solve the equation* $x^2 - 3x - 1 = 0$ *in the interval* $[-1, 1]$.

Chapter 3

Solving Systems of Linear Equations

3.1 Introduction

Many phenomena and relationships in nature and real life applications are linear, meaning that results and their causes are proportional to each other. Solving linear algebraic equations is a topic of great importance in numerical analysis and other scientific disciplines such as engineering and physics. Solutions to Many problems reduced to solve a system of linear equations. For example, in finite element analysis a solution of a partial differential equation is reduced to solve a system of linear equations.

3.2 Norms of Matrix and Vectors

In error and convergence analyses we need a measure to determine the distance (difference) between the exact solution and approximate solution or to determine the differences between consecutive approximations.

Definition 20 (Vector Norm)**.** *A **vector norm** is a real-valued function* $\|.\| : \mathbb{R}^n \to \mathbb{R}$ *satisfies the following conditions:*

(i) $\|\mathbf{x}\| \geq 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$*.*

(ii) $\|\mathbf{x}\| = 0$ *if and only if* $\mathbf{x} = \mathbf{0}$ *for all* $\mathbf{x} \in \mathbb{R}^n$*.*

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ *for all* $\alpha \in \mathbb{R}$ *and* $\mathbf{x} \in \mathbb{R}^n$*.*

(iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ *(Triangle Inequality).*

Definition 21 (l_1 Vector Norm)**.** *Let* $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. *Then the* l_1 ***norm*** *for the vector* $\mathbf{x}$ *is defined by*

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

Definition 22 (Euclidean Vector Norm)**.** *Let* $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. *Then the* ***Euclidean norm*** *(*l_2 ***norm****) for the vector* $\mathbf{x}$ *is defined by*

$$\|\mathbf{x}\|_2 = \Big(\sum_{i=1}^{n} x_i^2\Big)^{1/2}.$$

Definition 23 (Maximum Vector Norm)**.** *Let* $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. *Then the* ***maximum norm*** *(*l_∞ ***norm****) for the vector* $\mathbf{x}$ *is defined by*

$$\|\mathbf{x}\|_\infty = \max_{1\le i\le n} |x_i|.$$

Remark 8. *Note that when* $n = 1$ *both norms reduce to the absolute value function of real numbers.*

Example 19. *Determine the* l_1 *norm,* l_2 *norm and* l_∞ *norm of the vector* $\mathbf{x} = (1, 0, -1, 2, 3)'$.

Solution: The required norms of vector $\mathbf{x} = (1, 0, -1, 2, 3)'$ in $\mathbb{R}^5$ are:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{5} |x_i| = |x_1|+|x_2|+|x_3|+|x_4|+|x_5| = |1|+|0|+|-1|+|2|+|3| = 7,$$

$$\begin{aligned}\|\mathbf{x}\|_2 &= \Big(\sum_{i=1}^{5} x_i^2\Big)^{1/2} = \Big(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2\Big)^{1/2} \\ &= \Big((1)^2 + (0)^2 + (-1)^2 + (2)^2 + (3)^2\Big)^{1/2} = \Big(15\Big)^{1/2},\end{aligned}$$

and

$$\begin{aligned}\|\mathbf{x}\|_\infty &= \max_{1\le i\le 5} |x_i| = \max\{|x_1|, |x_2|, |x_3|, |x_4|, |x_5|\} \\ &= \max\{|1|, |0|, |-1|, |2|, |3|\} = 3.\end{aligned}$$

Definition 24 (Matrix Norm). *A **matrix norm** is a real-valued function $\|.\| : \mathbb{R}^{n\times m} \to \mathbb{R}$ satisfies the following conditions:*

(i) $\|A\| \geq 0$ *for all* $A \in \mathbb{R}^{n\times m}$.

(ii) $\|A\| = 0$ *if and only if* $A = \mathbf{0}$ *for all* $A \in \mathbb{R}^{n\times m}$.

(iii) $\|\alpha A\| = |\alpha|\|A\|$ *for all* $\alpha \in \mathbb{R}$ *and* $A \in \mathbb{R}^{n\times m}$.

(iv) $\|A + B\| \leq \|A\| + \|B\|$ *for all* $A, B \in \mathbb{R}^{n\times m}$ *(Triangle Inequality).*

If matrix norm is related to a vector norm, then we have two additional properties:

(v) $\|AB\| \leq \|A\|\|B\|$ *for all* $A, B \in \mathbb{R}^{n\times m}$.

(vi) $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$ *for all* $A \in \mathbb{R}^{n\times m}$ *and* $\mathbf{x} \in \mathbb{R}^n$.

We give here some equivalent definitions of the matrix norm particularly when matrix norm is related to the vector norm.

Definition 25 (Subordinate Matrix Norm). *Let A is a $n \times n$ matrix and $\mathbf{x} \in \mathbb{R}^n$, then the **subordinate matrix norm** is defined by*

$$\|A\| = \sup\{\|A\mathbf{x}\| : \mathbf{x} \in \mathbb{R}^n and\ \|\mathbf{x}\| = 1\}.$$

or, alternatively

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|.$$

Definition 26 (Natural Matrix Norm). *Let A is a $n \times n$ matrix and for any $\mathbf{z} \neq \mathbf{0}$, and $\mathbf{x} = \frac{\mathbf{z}}{\|\mathbf{z}\|}$ is the unit vector. Then the **natural / reduced matrix norm** is defined by*

$$\max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| = \max_{\mathbf{z}\neq 0} \left\|A\Big(\frac{\mathbf{z}}{\|\mathbf{z}\|}\Big)\right\| = \max_{\mathbf{z}\neq 0} \frac{\|A\mathbf{z}\|}{\|\mathbf{z}\|},$$

or, alternatively

$$\|A\| = \max_{\mathbf{z}\neq 0} \frac{\|A\mathbf{z}\|}{\|\mathbf{z}\|}.$$

Definition 27 (l_1 Matrix Norm). *Let A is a $n\times n$ matrix and $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. Then the l_1 **matrix norm** is defined by*

$$\|A\|_1 = \max_{\|\mathbf{x}\|_1=1} \|A\mathbf{x}\|_1 = \max_{1\leq i\leq n} \sum_{i=1}^{n} |a_{ij}|.$$

Definition 28 (Spectral Matrix Norm)**.** *Let A is a $n \times n$ matrix and $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. Then the* ***spectral / l_2-matrix norm*** *is defined by*

$$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2 = \max_{1\le i\le n} \sqrt{|\sigma_{\max}|},$$

where σ_i are the eigenvalues of $A^T A$, which are called the ***singular values*** *of A and the largest eigenvalue in absolute value ($|\sigma_{\max}|$) is called the* ***spectral radius*** *of A.*

Definition 29 (l_∞ Matrix Norm)**.** *Let A is a $n\times n$ matrix and $\mathbf{x} = (x_1, x_2, \cdots, x_n)'$. Then the l_∞* ***(maximum)matrix norm*** *is defined by*

$$\|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|.$$

Remark 9. *Note that $\|I\| = 1$.*

Example 20. *Determine $\|A\|_\infty$ for the matrix*

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

Solution: For $i = 1$, we have

$$\sum_{j=1}^{3} |a_{1j}| = |a_{11}| + |a_{12}| + |a_{13}| = |1| + |-1| + |2| = 4,$$

and for $i = 2$, we obtain

$$\sum_{j=1}^{3} |a_{2j}| = |a_{21}| + |a_{22}| + |a_{23}| = |0| + |5| + |3| = 8,$$

for $i = 3$, we get

$$\sum_{j=1}^{3} |a_{3j}| = |a_{31}| + |a_{32}| + |a_{33}| = |-1| + |6| + |-4| = 11.$$

Consequently,

$$\|A\|_\infty = \max_{1\le i\le 3} \sum_{j=1}^{3} |a_{ij}| = \max\{4, 8, 11\} = 11.$$

3.3 Direct Methods

Direct methods are techniques used for solving and obtaining the exact solutions (in theory) of linear algebraic equations in a finite number of steps. The main widely used direct methods are **Gaussian elimination method** and **Gauss-Jordan method**.

Consider the following linear system of dimension $n \times (n+1)$

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

This system can be written in concise form by using matrix notation as $AX = B$ as follows:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{12} & \cdots & a_{1n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where $A_{n\times n}$ is square matrix and is called a **coefficient matrix**, $B_{n\times 1}$ is a column vector known as the **right hand side vector** and $X_{n\times 1}$ is a column vector known as **unknowns vector**. Also, this system can be written as

$$[A|B] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{12} & \cdots & a_{1n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{12} & \cdots & a_{1n} & b_n \end{array}\right],$$

where $[A|B]$ is called the **augmented matrix**.

3.3.1 Backward Substitution Method

Backward substitution also called **back substitution** is an algorithm or technique used for solving **upper-triangular systems** which are systems such that their coefficient matrices are upper-triangular matrices. Assume that we have the following upper-triangular system

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n-1}x_{n-1} + a_{1n}x_n &= b_1\\
a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n-1}x_{n-1} + a_{2n}x_n &= b_2\\
a_{33}x_3 + \cdots + a_{3n-1}x_{n-1} + a_{3n}x_n &= b_3\\
&\vdots\\
a_{n-1n-1}x_{n-1} + a_{n-1n}x_n &= b_{n-1}\\
a_{nn}x_n &= b_n.
\end{aligned}$$

To find a solution to this system we follow the following steps provided that $x_{rr} \neq 0,\ r = 1, 2, \cdots, n$:

(1) Solve the last (nth) equation for x_n:

$$x_n = \frac{b_n}{a_{nn}}.$$

(2) Substitute x_n in the next-to-last $((n-1)th)$ equation and solve it for x_{n-1}:

$$x_{n-1} = \frac{b_{n-1} - a_{n-1n}x_n}{a_{n-1n-1}}.$$

(3) Now, x_n and x_{n-1} are known and can be used to find x_{n-2}:

$$x_{n-2} = \frac{b_{n-2} - a_{n-1n-1}x_{n-1} - a_{n-1n}x_n}{a_{n-2n-2}}.$$

(4) Continuing in this way until we arrive at the general step:

$$x_r = \frac{b_r - \sum_{j=r+1}^{n} a_{rj}x_j}{a_{rr}}, \quad r = n-1, n-2, \cdots 1.$$

Example 21. *Solve the following linear system using back substitution method*

$$\begin{aligned}
3x_1 + 2x_2 - x_3 + x_4 &= 10\\
x_2 - x_3 + 2x_4 &= 9\\
3x_3 - x_4 &= 1\\
3x_4 &= 6
\end{aligned}$$

Solution: Solve the last equation for x_4 to obtain

$$x_4 = \frac{6}{3} = 2.$$

Substitute $x_4 = 2$ in the third equation, we have

$$x_3 = \frac{1 + x_4}{3} = \frac{1+2}{3} = \frac{3}{3} = 1.$$

Now, use values $x_3 = 1$ and $x_4 = 2$ in the second equation to find x_2

$$x_2 = 9 + x_3 - 2x_4 = 9 + 1 - 4 = 6.$$

Finally, solve the first equation for x_1 yields

$$x_1 = \frac{10 - 2x_2 + x_3 - x_4}{3} = \frac{10 - 12 + 1 - 2}{3} = \frac{-3}{3} = -1.$$

Example 22. *Show that the following linear system has no solution*

$$\begin{aligned} 3x_1 + 2x_2 - x_3 + x_4 &= 10 \\ 0x_2 - x_3 + 2x_4 &= 9 \\ 3x_3 - x_4 &= 1 \\ 3x_4 &= 6 \end{aligned}$$

Solution: Solve the last equation for x_4 to obtain

$$x_4 = \frac{6}{3} = 2.$$

Substitute $x_4 = 2$ in the third equation, we have

$$x_3 = \frac{1 + x_4}{3} = \frac{1+2}{3} = \frac{3}{3} = 1.$$

Also, from the second equation we have

$$x_3 = 9 - 2x_4 = 9 - 4 = 6.$$

This contradiction implies that the linear system in above has no solution.

Example 23. *Show that the following linear system has infinitely many solutions*

$$\begin{aligned} 3x_1 + 3x_2 - x_3 + x_4 &= 10 \\ 0x_2 + x_3 + 0x_4 &= 1 \\ 3x_3 - x_4 &= 1 \\ 3x_4 &= 6 \end{aligned}$$

Solution: Solve the last equation for x_4 to obtain

$$x_4 = \frac{6}{3} = 2.$$

Substitute $x_4 = 2$ in the third equation, we have

$$x_3 = \frac{1+x_4}{3} = \frac{1+2}{3} = \frac{3}{3} = 1.$$

Also, from the second equation we have

$$x_3 = 1.$$

Solve the first equation for x_2 yields

$$x_2 = \frac{10-3x_1+x_3-x_4}{3} = \frac{10-3x_1+1-2}{3} = \frac{9-3x_1}{3} = 3-x_1.$$

Note that the equation for x_2 has infinitely many solutions since it depends upon x_1 which takes infinitely many values. Now, let $x_1 = 1$, we have $x_2 = 2$. Hence the solution set of the system is:

$$x_1 = 1,\ x_2 = 2,\ x_3 = 1,\ x_4 = 2.$$

3.3.2 Forward Substitution Method

Forward substitution is an algorithm or technique used for solving **lower-triangular systems** which are systems such that their coefficient matrices are lower-triangular matrices.

$$\begin{aligned}
&a_{11}x_1 &&= b_1\\
&a_{21}x_1 + a_{22}x_2 &&= b_2\\
&a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &&= b_3\\
& &&\vdots\\
&a_{n-11}x_1 + a_{n-12}x_2 + a_{n-13}x_3 + \cdots + a_{n-1n-1}x_{n-1} &&= b_{n-1}\\
&a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn-1}x_{n-1} + a_{nn}x_n &&= b_n.
\end{aligned}$$

To find a solution to this system we follow the following steps provided that $x_{rr} \neq 0,\ r = 1, 2, \cdots, n$:

(1) Solve the first ($1st$) equation for x_1:

$$x_1 = \frac{b_1}{a_{11}}.$$

(2) Substitute x_1 in the second equation ($2nd$) equation and solve it for x_2:

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}.$$

(3) Now, x_1 and x_2 are known and can be used to find x_3:

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}.$$

(4) Continuing in this way until we arrive at the general step:

$$x_r = \frac{b_r - \sum_{j=1}^{r-1} a_{rj}x_j}{a_{rr}}, \quad r = 2, 3, \cdots n.$$

Example 24. *Use the forward substitution method for solving the following linear system*

$$\begin{aligned} &4x_1 &&= 8 \\ &2x_1 + x_2 &&= -1 \\ &x_1 - x_2 + 5x_3 &&= 0.5 \\ &0.1x_1 + 2x_2 - x_3 + 2x_4 &&= 2 \end{aligned}$$

,

Solution: Solving the first equation for x_1 yields

$$x_1 = \frac{8}{4} = 2.$$

Using the value of x_1 to find x_2

$$x_2 = \frac{-1 - 2x_1}{2} = \frac{-1 - 2(2)}{2} = -2.5.$$

Use x_1 and x_2 to find x_3

$$x_3 = \frac{0.5 - x_1 + x_2}{5} = \frac{0.5 - 2 - 2.5}{5} = \frac{-4}{5} = -0.8.$$

Finally, solve for x_4 to have

$$x_4 = \frac{2 - 0.1x_1 - 2x_2 + x_3}{2} = \frac{2 - 0.1(2) - 2(-2.5) - 0.8}{2} = \frac{6}{2} = 3.$$

Example 25. *Show that there is no solution to the linear system*

$$\begin{aligned}
&4x_1 &&= 8\\
&2x_1 + x_2 &&= -1\\
&x_1 - x_2 + 0x_3 &&= 0.5\\
&0.1x_1 + 2x_2 - x_3 + 2x_4 &&= 2
\end{aligned}$$

,

Solution: Solving the first equation for x_1 yields

$$x_1 = \frac{8}{4} = 2.$$

Using the value of x_1 in the second equation to find x_2

$$x_2 = \frac{-1 - 2x_1}{2} = \frac{-1 - 2(2)}{2} = -2.5.$$

From the third equation we have

$$x_2 = x_1 - 0.5 = 2 - 0. = 1.5.$$

This contradiction indicates that there is no solution to the system in above.

Example 26. *Show that there are infinitely many solution to the following linear system*

$$\begin{aligned}
&4x_1 &&= 8\\
&2x_1 + x_2 &&= -1\\
&0x_1 - x_2 + 0x_3 &&= 2.5\\
&0.1x_1 + 2x_2 - x_3 + 2x_4 &&= 2
\end{aligned}$$

,

Solution: Solving the first equation for x_1 yields

$$x_1 = \frac{8}{4} = 2.$$

Using the value of x_1 in the second equation to find x_2

$$x_2 = \frac{-1 - 2x_1}{2} = \frac{-1 - 2(2)}{2} = -2.5.$$

From the third equation we have

$$x_2 = -2.5.$$

Solving the last equation for x_3 we obtain

$$x_3 = -2 + 0.1x_1 + 2x_2 + 2x_4 = -2 + 0.1(2) + 2(-2.5) + 2x_4 = 2x_4 - 6.8,$$

which has infinitely many solutions. Hence, the above linear system has infinitely many solutions. If we choose $x_4 = 8$, then we get $x_3 = -0.8$. So, the solution set is:

$$x_1 = 2,\ x_2 = -2.5,\ x_3 = 9.8,\ x_4 = 8.$$

3.3.3 Gaussian Elimination Method

Gaussian elimination method is also known as **Gauss elimination method** or simply **elimination method**. It is a direct method used for solving a system of linear algebraic equations. In this method we transform the linear system to an equivalent upper or lower triangular system and then solve it by backward or forward substitution. The process of transforming the linear system to an equivalent upper or lower triangular system is called **trianguarisation**.

Definition 30 (Equivalent Systems)**.** *Two linear algebraic systems of dimension $n \times n$ is said to be they are* ***equivalent*** *if they have the same solution sets.*

Definition 31 (Elementary Transformations)**.** *The following operations performed on a linear system transform it to an equivalent system:*

- ***Interchanges:*** *Changing the order of any two equations in the system.*
- ***Replacement:*** *Any equation of the system can be replaced by itself and a nonzero multiple of any other equation in the system.*
- ***Scaling:*** *Multiplying any equation in the system by a nonzero real constant.*

Definition 32 (Elementary Row Operations)**.** *The following operations performed on a linear system transform it to an equivalent system:*

- ***Interchanges:*** *Changing the order of any two rows in the matrix.*
- ***Replacement:*** *Any row in the matrix can be replaced by its sum and a nonzero multiple of any other row in the matrix.*
- ***Scaling:*** *Multiplying any row in the matrix by a nonzero real constant.*

Pivoting

Pivoting is an important process used ins solving linear systems in conjunction with Gaussian elimination and there different types of pivoting strategies as outlined below:

1. **Trivial Pivoting:** The process of using the element (entry) a_{kk} in the coefficient matrix A to eliminate the entries $a_{rk},\ r = k+1, k+2, \cdots n$ is called **pivoting process**. The element a_{kk} is called **pivotal element** and the kth row is called **pivotal row**. If the entry $a_{kk} = 0$, then the row k cannot be used to eliminate the entries a_{rk}, $r = k+1, k+2, \cdots, n$ and we need to find a row r such that $a_{rk} \neq 0$, $r > k$, and then interchange the row k and the row r such that the pivotal element is nonzero. This process is called the **trivial pivoting**, also, if no interchange or switching between the rows is performed then the process is called only **pivoting** or **trivial pivoting**.

2. **Partial Pivoting:** To reduce the round-off errors or propagation of errors it is advisable to search for the the greatest element in the magnitude in column r that lies on or below the main diagonal, and then move it to the main diagonal in the pivotal row r to be the pivotal element and use it to eliminate the entries in the column r below the main diagonal, this process is called the **partial pivoting**. Determine row k below the main diagonal in which there is the largest element in the absolute value as follows:

$$a_{kr} = \max\{|a_{rr}|, |a_{r+1r}|, \cdots, |a_{n-1r}|, |a_{nr}|\}, \tag{3.1}$$

   and then interchange the row k and row r for $k > r$. Now, since the entry in the main diagonal has the larges absolute value then the values of all the multipliers are:

$$|m_{kr}| \leq 1, k = r+1, r+2, \cdots, n,$$

   and this will be helpful to keep the magnitudes of elements in the current matrix are relatively the same magnitudes of the elements in the original coefficient matrix.

3. **Scaled Pivoting:** In this approach, the pivoting element is chosen to be the largest in magnitude relative to the elements which lie in the same row. This type of pivoting is used when the entries in the same row vary largely in magnitude.

4. **Complete Pivoting:** In this technique, we use both partial and scaled pivoting and is sometimes referred to as **scaled partial pivoting** or **equilibrating**. In this process, we search all the entries in the column r that lie on or below the main diagonal for the largest entry in the magnitude relative to the entries in its row. Hence, We interchange both the columns and rows to find the largest entry in absolute value, i.e. we searching for largest entry un the matrix and for this reason this type of pivoting is also known as **maximal pivoting**. we start the process by searching all the rows r to n for the largest entry in absolute value in each row, we denote this element by p_k:

$$p_k = \max\{|a_{kr}|, |a_{kr+1}|, \cdots, |a_{kn-1}|, |a_{kn}|\},\ k = r, r+1, \cdots, n. \tag{3.2}$$

Then, to locate the pivoting row, we need to compute

$$\frac{a_{kr}}{p_k} = \max\{|\frac{a_{rr}}{p_r}|, |\frac{a_{r+1r}}{p_{r+1}}|, \cdots, |\frac{a_{n-1r}}{p_{n-1}}|, |\frac{a_{nr}}{p_n}|\}. \tag{3.3}$$

Then, interchange the row r and k, except the case when $r = k$.

Example 27. *Write the following linear system in the augmented form and then solve it by using Gauss elimination method with trivial pivoting.*

$$\begin{aligned} x_1 + 2x_2 - x_3 + 4x_4 &= 12 \\ 2x_1 + x_2 + x_3 + x_4 &= 10 \\ -3x_1 - x_2 + 4x_3 + x_4 &= 2 \\ x_1 + x_2 - x_3 + 3x_4 &= 6 \end{aligned}$$

.

Solution: The augmented matrix is

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 2 & 1 & 1 & 1 & 10 \\ -3 & -1 & 4 & 1 & 2 \\ 1 & 1 & -1 & 3 & 6 \end{array}\right]$$

The first row is the pivotal row, so the pivotal element is $a_{11} = 1$ and is used to eliminate the first column below the diagonal. We will denote by m_{r1} to the multiples of the row 1 subtracted from row r for $r = 2, 3, 4$. Multiplying the first row by $m_{21} = -2$ and add it to the second row to have

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ -3 & -1 & 4 & 1 & 2 \\ 1 & 1 & -1 & 3 & 6 \end{array}\right].$$

Now, multiply the first row by $m_{31} = 3$ and add it to the third row to obtain

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ 0 & 5 & 1 & 13 & 38 \\ 1 & 1 & -1 & 3 & 6 \end{array}\right].$$

Multiplying the first row by $m_{41} = -1$ and adding it to the fourth row yields

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ 0 & 5 & 1 & 13 & 38 \\ 0 & -1 & 0 & -1 & -6 \end{array}\right].$$

Now, the pivotal row is the second row and the pivotal element is $a_{22} = -3$. Multiply the second row by $m_{32} = \frac{5}{3}$ to have

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ 0 & 0 & 6 & 4/3 & 44/3 \\ 0 & -1 & 0 & -1 & -6 \end{array}\right].$$

Multiply the the second row by $m_{42} = \frac{-1}{3}$ and add it to the fourth row to obtain

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ 0 & 0 & 6 & 4/3 & 44/3 \\ 0 & 0 & -1 & 4/3 & -4/3 \end{array}\right].$$

Now, the pivotal row is the third row and the third element is $a_{33} = 6$. Finally, multiply the third row by $m_{43} = \frac{1}{6}$ to the fourth row to have

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 4 & 12 \\ 0 & -3 & 3 & -7 & -14 \\ 0 & 0 & 6 & 4/3 & 44/3 \\ 0 & 0 & 0 & 14/9 & 10/9 \end{array}\right].$$

Now, note that the coefficient matrix is transformed into an upper triangular matrix and can be solved by backward substitution method. Firstly, we from the last row we compute

$$x_4 = \frac{10/9}{14/9} = \frac{5}{7}.$$

Use the third row to solve for x_3

$$x_3 = \frac{44/3 - 4/3(5/7)}{6} = \frac{288/21}{6} = \frac{16}{7}.$$

Now, solve the second equation for x_2

$$x_2 = \frac{-14 - 3x_3 + 7x_4}{-3} = \frac{-14 - 3(16/7) + 7(5/7)}{-3} = \frac{111}{21} = \frac{37}{7}.$$

Finally, solve the first equation for x_1

$$x_1 = 12 - 2x_2 + x_3 - 4x_4 = 12 - 2(37/7) + 16/7 - 4(5/7) = \frac{6}{7}.$$

Example 28. *Solve the following linear system using Gauss elimination method by using forward substitution technique*

$$\begin{aligned} x_1 + 2x_2 + x_3 + 4x_4 &= 13 \\ 2x_1 + 0x_2 + 4x_3 + 3x_4 &= 28 \\ 4x_1 + 2x_2 + 2x_3 + x_4 &= 20 \\ -3x_1 + x_2 + 3x_3 + 2x_4 &= 6 \end{aligned}$$

.

Solution: We start our solution strategy by transforming this square system to equivalent lower-triangular system and then solve it by using forward substitution method. Write the system in augmented matrix form

$$\left[\begin{array}{cccc|c} 1 & 2 & 1 & 4 & 13 \\ 2 & 0 & 4 & 3 & 28 \\ 4 & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

$$\begin{array}{cccc|cc} a & b & c & d & e & \\ 1 & 1 & 1 & 1 & 1 & R_1 + 2R_2 \\ 0 & 1 & 0 & 0 & 1 & g \\ 0 & 0 & 1 & 0 & 1 & h \\ 0 & 0 & 0 & 1 & 1 & i \\ 0 & 0 & 0 & 0 & 1 & j \end{array}$$

Note that now the pivotal row is the fourth row and the pivotal element is $a_{44} = 2$. Multiply the fourth row by the multiple $m_{14} = -2$ and it to the first row to have

$$\left[\begin{array}{cccc|c} 7 & 0 & -5 & 0 & 1 \\ 2 & 0 & 4 & 3 & 28 \\ 4 & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

Multiply the fourth row by $m_{24} = \frac{-3}{2}$ and add it to the second row to obtain

$$\left[\begin{array}{cccc|c} 7 & 0 & -5 & 0 & 1 \\ 13/2 & -3/2 & -1/2 & 0 & 19 \\ 4 & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

Now multiply the fourth equation by $m_{34} = \frac{-1}{2}$ and add it to the third row to have

$$\left[\begin{array}{cccc|c} 7 & 0 & -5 & 0 & 1 \\ 13/2 & -3/2 & -1/2 & 0 & 19 \\ 11/2 & 3/2 & 1/2 & 0 & 17 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

The pivotal row now is the third row and the pivotal element is $a_{33} = 1/2$. Add the third row to the second row (i.e. multiply it by $m_{23} = 1$) to get

$$\left[\begin{array}{cccc|c} 7 & 0 & -5 & 0 & 1 \\ 12 & 0 & 0 & 0 & 36 \\ 11/2 & 3/2 & 1/2 & 0 & 17 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

Now

$$\left[\begin{array}{cccc|c} 12 & 0 & 0 & 0 & 36 \\ 11/2 & 3/2 & 1/2 & 0 & 17 \\ 7 & 0 & -5 & 0 & 1 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

The pivotal row (third row) is used to eliminate elements in the second row and the pivotal element is $a_{33} = -5$. Multiply the third row by $m_{23} = \frac{1}{10}$ to have

$$\left[\begin{array}{cccc|c} 12 & 0 & 0 & 0 & 36 \\ 31/5 & 3/2 & 0 & 0 & 171/10 \\ 7 & 0 & -5 & 0 & 1 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

Now, use forward substitution to solve the lower-triangular matrix. solve the first equation for x_1

$$x_1 = \frac{36}{12} = 3.$$

Use the equation to find x_2

$$x_2 = \frac{171/10 - (31/5)3}{3/2} = -1.$$

Now, solve the third equation for x_3

$$x_3 = \frac{1 - 7(3)}{-5} = 4.$$

Finally, solve the fourth equation for x_4

$$x_4 = \frac{6 - (-3)(3) - 1(-1) - 3(4)}{2} = 2.$$

3.3.4 Gauss-Jordan Elimination Method

In this method instead of transforming the coefficient matrix into upper or lower triangular system, we transform the coefficient matrix into diagonal (in particular identity) matrix using elementary row operations.

Example 29. *Solve the following linear system using Gauss-Jordan elimination method*

$$\begin{aligned} 3x_1 + 4x_2 + 3x_3 &= 10 \\ x_1 + 5x_2 - x_3 &= 7 \\ 6x_1 + 3x_2 + 7x_3 &= 15 \end{aligned}$$

.

Solution: Express the system in augmented matrix form

$$\left[\begin{array}{ccc|c} 3 & 4 & 3 & 10 \\ 1 & 5 & -1 & 7 \\ 6 & 3 & 7 & 15 \end{array}\right].$$

The pivot row is the first row and the pivot element is $a_{11} = 3$. Multiply it by $m_{11} = 1/3$ to get

$$\left[\begin{array}{ccc|c} 1 & 4/3 & 1 & 10/3 \\ 1 & 5 & -1 & 7 \\ 6 & 3 & 7 & 15 \end{array}\right].$$

Subtract the second equation from the first (i.e. multiply it by $m_{21} = -1$) and multiply the fist equation by $m_{31} = -6$ and add it to the third equation to have

$$\left[\begin{array}{ccc|c} 1 & 4/3 & 1 & 10/3 \\ 0 & -11/3 & 2 & -11/3 \\ 0 & -5 & 1 & -5 \end{array}\right].$$

Now, the pivot row is the second row and the pivot element $a_{22} = -11/3$. Multiply it by $m_{22} = -3/11$ to have

$$\left[\begin{array}{ccc|c} 1 & 4/3 & 1 & 10/3 \\ 0 & 1 & -6/11 & 1 \\ 0 & -5 & 1 & -5 \end{array}\right].$$

Multiply the first and third rows by $m_{12} = -4/3$ and $m_{32} = 5$ to obtain

$$\left[\begin{array}{ccc|c} 1 & 0 & 19/11 & 2 \\ 0 & 1 & -6/11 & 1 \\ 0 & 0 & -19/11 & 0 \end{array}\right].$$

The pivot element now is third row and the pivot element is $a_{33} = -19/11$. Multiply it by $m_{33} = -11/19$ to get

$$\left[\begin{array}{ccc|c} 1 & 0 & 19/11 & 2 \\ 0 & 1 & -6/11 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right].$$

Finally, multiply the third row by $m_{13} = -19/11$ and $m_{23} = 6/11$ and add it to the first and second rows to have

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right].$$

Hence, we have $x_1 = 2$, $x_2 = 1$ and $x_3 = 0$.

Example 30. *Solve the following linear system using Gauss-Jordan elimination method*

$$\begin{aligned} -2x_1 + x_2 + 5x_3 &= 15 \\ 4x_1 - 8x_2 + x_3 &= -21 \\ 4x_1 - x_2 + x_3 &= 7 \end{aligned}$$

.

Solution: Write the system in augmented matrix form

$$\left[\begin{array}{ccc|c} -2 & 1 & 5 & 15 \\ 4 & -8 & 1 & -21 \\ 4 & -1 & 1 & 7 \end{array}\right].$$

Multiply the first row by $m_{21} = m_{31} = -2$ and it to the second and third rows respectively, to obtain

$$\left[\begin{array}{ccc|c} -2 & 1 & 5 & 15 \\ 0 & -6 & 11 & 9 \\ 0 & 1 & 11 & 37 \end{array}\right].$$

Now, multiply the second row by $m_{12} = m_{32} = \frac{1}{6}$ and it to the first and third rows respectively, to have

$$\left[\begin{array}{ccc|c} -2 & 0 & 41/6 & 33/2 \\ 0 & -6 & 11 & 9 \\ 0 & 0 & 77/6 & 77/2 \end{array}\right].$$

Finally, multiply the third row by $m_{13} = \frac{-41}{77}$ and $m_{32} = \frac{-6}{7}$ and it to the first and third rows respectively, to obtain

$$\left[\begin{array}{ccc|c} -2 & 0 & 0 & -4 \\ 0 & -6 & 0 & -24 \\ 0 & 0 & 77/6 & 77/2 \end{array}\right],$$

implies that

$$x_1 = \frac{-4}{-2} = 2, \quad x_2 = \frac{-24}{-6} = 4 \quad \text{and} \quad x_3 = \frac{77/2}{77/6} = 3.$$

3.4 *LU* and Cholesky Factorisations

In this section we will discuss the triangular factorisations of matrices.

Definition 33 (Positive Definite Matrix)**.** *Let $A_{n\times n}$ be symmetric real matrix and $\mathbf{x} \in \mathbb{R}^n$ a nonzero vector. Then, A is said to be **positive definite matrix** if $A = A^T$ and $\mathbf{x}^T A\mathbf{x} > 0$ for any $\mathbf{x}$.*

Remark 10. *Note that the matrix A is nonsingular by definition.*

Definition 34 (Triangular Factorisation)**.** *Assume that A is a nonsingular matrix. It said to be A has a **triangular factorisation** or **triangular decomposition** if it can be factorised as a product of unit lower-triangular matrix L and an upper triangular matrix U:*

$$A = LU.$$

or in matrix form

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

*Note that since A is nonsingular matrix this implies that $u_{rr} \neq 0$ for all r and this is called **Doolittle factorisation**.*

Also, A can be expressed as a product of lower-triangular matrix L and unit upper triangular matrix U:

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

*and this is called **Crout factorisation**.*

To solve the linear system $AX = B$ using LU factorisation, we do the following two steps:

1. Using forward substitution to solve the the lower-triangular linear system $LY = B$ for Y.

2. Using backward substitution to solve the upper-triangular linear system $UX = Y$ for X.

Direct LU Factorisation Using Gaussian Elimination Method

The matrix A can be factored directly using Gauss elimination method without any row interchanges. In this case the matrix A is expressed in terms of the identity matrix I follows $A = IA$. We perform the row operations on the matrix A on the right and the resulting matrix it will be the upper triangular matrix U. The multipliers are stored in their appropriate places in the identity matrix on the left which will be the lower triangular matrix L. All this information is summarised in the next theorem.

Theorem 19 (**Direct LU Factorisation Without Row Interchanges**). *Assume that the linear system $AX = B$ can be solved using Gaussian elimination with no row interchanges. Then, the coefficient matrix A can be factored as a product of a lower triangular matrix L and an upper triangular matrix U as follows:*

$$A = LU.$$

The matrix L has 1's on its main diagonal and the matrix has nonzero entries on its main diagonal. After constructing the matrices L and U then the linear system can be solved in the following two steps:

(1). Solve the lower triangular system $LY = B$ for Y using the forward substitution method.

(2). Solve the upper triangular system $UX = Y$ for X using the backward substitution method.

Proof. For proof, see any standard text on numerical analysis or numerical linear algebra. □

The following example explains this type of LU factorisation.

Example 31. *Find the LU factorisation of the following matrix using Gaussian elimination without row interchanges*

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

Solution. *Writing the matrix A in terms of the identity matrix as follows*

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

The first row is used to eliminate the elements under the main diagonal (subdiagonal elements) in the first column. The multipliers of the first row are $m_{21} = a_{21}/a_{11} = -2/2 = -1$ *and* $m_{31} = a_{31}/a_{11} = 1/2 = 0.5$*, respectively.*

$$\begin{bmatrix} 2 & 4 & -1 \\ -2 & 3 & 1 \\ 1 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -1 \\ 0 & 7 & 0 \\ 0 & 3 & 6.5 \end{bmatrix}$$

Now, the second row is used to eliminate the entries below the main diagonal in the second column and the multiple of the second row is $m_{32} = a_{32}/a_{22} = 3/7$*. Hence, we have the following LU factorisation of* A

$$A = \begin{bmatrix} 2 & 4 & -1 \\ -2 & 3 & 1 \\ 1 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1/2 & 3/7 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -1 \\ 0 & 7 & 0 \\ 0 & 0 & 6.5 \end{bmatrix} = LU.$$

The LU Factorisation Without Using Gaussian Elimination Method

Example 32. *Solve the following linear system using LU (Doolittle) decomposition*

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

Solution: Express the system in matrix form

$$\left[\begin{array}{ccc|c} 2 & -3 & 1 & 2 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 0 \end{array}\right].$$

Factor A as follows:

$$\begin{bmatrix} 2 & -3 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}.$$

Find the values of the entries of matrices L and U. From the first column we have

$$2 = 1u_{11} \implies u_{11} = 2,$$

and

$$1 = l_{21}u_{11} = l_{21}2 \implies l_{21} = 0.5,$$

finally

$$-1 = l_{31}u_{11} = l_{31}2 \implies l_{31} = -0.5.$$

In the second column, we have

$$-3 = 1u_{12} \implies u_{12} = -3,$$

and

$$1 = l_{21}u_{12} + 1u_{22} = -1.5 + u_{22} \implies u_{22} = 2.5,$$

so

$$1 = l_{31}u_{12} + l_{32}u_{22} = (-0.5)(-3) + l_{32}(2.5) \implies l_{32} = -0.2.$$

Finally, in the third column we have

$$1 = 1u_{13} \implies u_{13} = 1,$$

and

$$-1 = l_{21}u_{13} + 1u_{23} = 0.5 + u_{23} \implies u_{23} = -1.5,$$

finally,

$$-1 = l_{31}u_{13} + l_{32}u_{23} + 1u_{33} = -0.5(1) + (-0.2)(-1.5) + u_{33} \implies u_{33} = -0.8.$$

Now, we have the LU factorisation

$$A = \begin{bmatrix} 2 & -3 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.5 & -0.2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 2.5 & -1.5 \\ 0 & 0 & -0.8 \end{bmatrix} = LU.$$

Now, we have the following lower-triangular linear system $LY = B$ for Y

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.5 & -0.2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}.$$

Write the system in augmented matrix form

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0.5 & 1 & 0 & -1 \\ -0.5 & -0.2 & 1 & 0 \end{array}\right].$$

Solve this system by forward substitution to have

$$y_1 = 2, \quad y_2 = -1 - 0.5(y_1) = -1 - 0.5(2) = -2,$$

and

$$y_3 = 0 + 0.5(y_1) + 0.2(y_2) = 0.5(2) + 0.2(-2) = 0.6.$$

Now, we have the following upper-triangular linear system $UX = Y$

$$\begin{bmatrix} 2 & -3 & 1 \\ 0 & 2.5 & -1.5 \\ 0 & 0 & -0.8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 0.6 \end{bmatrix}.$$

Express the system in augmented matrix form

$$\left[\begin{array}{ccc|c} 2 & -3 & 1 & 2 \\ 0 & 2.5 & -1.5 & -2 \\ 0 & 0 & -0.8 & 0.6 \end{array}\right].$$

Finally, use the values of Y to solve the upper-triangular linear system $UX = Y$ by back substitution to have

$$x_3 = \frac{0.6}{-0.8} = \frac{-3}{4}, \quad x_2 = \frac{-2+1.5(x_3)}{2.5} = \frac{-2+1.5(-3/4)}{2.5} = -5/4,$$

and

$$x_1 = \frac{2+3(x_2)-1(x_3)}{2} = \frac{2+3(-5/4)-(-3/4)}{2} = -1/2.$$

Definition 35 (Cholesky Factorisation)**.** *Let A be a real, symmetric and positive definite matrix. Then, it can be* ***factored*** *or* ***decomposed*** *in a unique way $A = LL^T$, in which L is a lower-triangular matrix with a positive diagonal, and is termed* ***Cholesky factorisation****.*

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

Example 33. *(a) Determine the Cholesky decomposition of the matrix*

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

(b) Then, use the decomposition from part (a) to solve the linear system

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

Solution: Factor A as a product LL^T as follows:

$$\begin{bmatrix} 2 & -1 & 2 \\ 4 & 3 & 3 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}.$$

From the first column we obtain

$$2 = l_{11}^2 \implies l_{11} = \sqrt{2},$$

$$4 = l_{21}l_{11} + l_{22}(0) \implies 4 = l_{21}\sqrt{2} \implies l_{21} = \frac{1}{2\sqrt{2}},$$

$$1 = l_{31}l_{11} + l_{32}(0) + l_{33}(0) \implies 1 = l_{31}\sqrt{2} \implies l_{31} = \frac{1}{\sqrt{2}}.$$

Now, from the second column we have

$$3 = l_{21}^2 + l_{22}^2 \implies 3 = \frac{1}{8} + l_{22}^2 \implies l_{22} = \sqrt{\frac{23}{8}},$$

$$1 = l_{31}l_{21} + l_{32}l_{22} + l_{33}(0) \implies 1 = \frac{1}{4} + l_{32}\sqrt{\frac{23}{8}} \implies l_{32} = \frac{3\sqrt{2}}{2\sqrt{23}}.$$

Finally, from the third column we get

$$2 = l_{31}^2 + l_{32}^2 + l_{33}^2 \implies 2 = \frac{1}{2} + \frac{9}{46} + l_{33}^2 \implies l_{33} = \sqrt{\frac{30}{23}}.$$

3.5 Iterative Methods

Direct methods are more efficient in solving linear systems of small dimensions in less computational cost than iterative methods. For large linear systems in particular for sparse linear systems iterative methods are more efficient for solving linear systems in terms of computational cost and effort compared to direct methods. In this section we will study the most common and basic iterative methods for solving linear algebraic systems which are **Jacobi method** and **Gauss-Siedel method**.

3.5.1 Jacobi Method

The general form of **Jacobi iterative method** for solving the *ith* equation in the linear system $AX = B$ for unknown $x_i, i = 1, , \cdots, n$ is:

$$x_i^k = \sum_{j=1}^{n} \Big(-\frac{a_{ij}x_j^{k-1}}{a_{ii}} \Big) + \frac{b_i}{a_{ii}}, \ j \neq i, \ a_{ii} \neq 0, \text{ for } i = 1, \cdots, n, \ k = 1, \cdots, n.$$

It is also known as **Jacobi iterative process** or **Jacobi iterative technique**

Example 34. *Solve the following linear system using Jacobi iterative method*

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 0 \\ x_1 + 3x_2 + x_3 &= 0.5 \\ x_1 + x_2 + 2.5x_3 &= 0 \end{aligned}$$

Solution: These equations can be written in the form

$$\begin{aligned} x_1 &= \frac{-x_2 - x_3}{2}, \\ x_2 &= \frac{0.5 - x_1 - x_3}{3}, \\ x_3 &= \frac{-x_1 - x_2}{2.5}. \end{aligned}$$

Writing these equations in iterative form

$$\begin{aligned} x_1^{k+1} &= \frac{-x_2^k - x_3^k}{2}, \\ x_2^{k+1} &= \frac{0.5 - x_1^k - x_3^k}{3}, \\ x_3^{k+1} &= \frac{-x_1^k - x_2^k}{2.5}. \end{aligned}$$

Let us start with initial guess $P_0 = (x_1^0, x_2^0, x_3^0) = (0, 0.1, -0.1)$. Substituting these values in the right-hand side of each equation in above to find the new iterations

$$\begin{aligned} x_1^1 &= \frac{-x_2^0 - x_3^0}{2} = \frac{-0.1 - (-0.1)}{2} = \frac{-0.1 + 0.1}{2} = 0, \\ x_2^1 &= \frac{0.5 - x_1^0 - x_3^0}{3} = \frac{0.5 - 0 - (-0.1)}{3} = 0.2, \\ x_3^1 &= \frac{-x_1^0 - x_2^0}{2.5} = \frac{-0 - 0.1}{2.5} = -0.04. \end{aligned}$$

Now, the new point $P_1 = (x_1^1, x_2^1, x_3^1) = (0, 0.2, -0.04)$ is used in the Jacobi iterative form to find the next approximation P_2

$$\begin{aligned}
x_1^2 &= \frac{-x_2^1 - x_3^1}{2} = \frac{-0.2 + 0.04}{2} = \frac{-0.16}{2} = -0.08,\\
x_2^2 &= \frac{0.5 - x_1^1 - x_3^1}{3} = \frac{0.5 + 0.04}{3} = \frac{0.54}{3} = 0.18,\\
x_3^2 &= \frac{-x_1^1 - x_2^1}{2.5} = \frac{-0 - 0.2}{2.5} = \frac{-0.2}{2.5} = -0.08.
\end{aligned}$$

The new point $P_2 = (x_1^2, x_2^2, x_3^2) = (-0.08, 0.18, -0.08)$ is closer to the solution than P_0 and P_1 and is used to find P_3

$$\begin{aligned}
x_1^3 &= \frac{-x_2^2 - x_3^2}{2} = \frac{-0.18 + 0.08}{2} = \frac{-0.1}{2} = -0.05,\\
x_2^3 &= \frac{0.5 - x_1^2 - x_3^2}{3} = \frac{0.5 + 0.08 + 0.08}{3} = \frac{0.66}{3} = 0.22,\\
x_3^3 &= \frac{-x_1^2 - x_2^2}{2.5} = \frac{0.08 - 0.18}{2.5} = \frac{-0.1}{2.5} = -0.04.
\end{aligned}$$

This Jacobi iteration process generates a sequence of points $\{P_n\} = \{(x_1^n, x_2^n, x_3^n)\}$ that converges to the solution $(x_1, x_2, x_3) = (-3/38, 4/19, -1/19) = (-0.078947368421053, 0.210526315789474, -0.052631578947368)$. The outline of the results is given in the Table 3.1.

n	x_1^n	x_2^n	x_3^n
0	0.000000000000000	0.100000000000000	−0.100000000000000
1	0.000000000000000	0.200000000000000	−0.040000000000000
2	−0.080000000000000	0.180000000000000	−0.080000000000000
3	−0.050000000000000	0.220000000000000	−0.040000000000000
4	−0.090000000000000	0.196666666666667	−0.068000000000000
5	−0.064333333333333	0.219333333333333	−0.042666666666667
6	−0.088333333333333	0.202333333333333	−0.062000000000000
7	−0.070166666666667	0.216777777777778	−0.045600000000000
8	−0.085588888888889	0.205255555555556	−0.058644444444444
9	−0.073305555555556	0.214744444444444	−0.047866666666667
10	−0.083438888888889	0.207057407407407	−0.056575555555556
11	−0.075240925925926	0.213338148148148	−0.049447407407407
12	−0.081945370370370	0.208229444444444	−0.055238888888889
13	−0.076495277777778	0.212394753086420	−0.050513629629630
14	−0.080940561728395	0.209002969135802	−0.054359790123457
15	−0.077321589506173	0.211766783950617	−0.051224962962963
16	−0.080270910493827	0.209515517489712	−0.053778077777778
17	−0.077868719855967	0.211349662757202	−0.051697842798354
18	−0.079825909979424	0.209855520884774	−0.053392377160494
19	−0.078231571862140	0.211072762379973	−0.052011844362140
20	−0.079530459008916	0.210081138741427	−0.053136476207133

Table 3.1: Jacobi Iterative Solution of Example 34

3.5.2 Gauss-Siedel Method

An improvement of Jacobi method can be made by using the recent values x_i^k, $i, k = 1, \cdots, n$, in the calculations once their values are obtained. This improvement is called **Gauss-Sedel iterative method** and its general form for solving the ith equation in the linear system $AX = B$ for unknown $x_i, i = 1, , \cdots, n$ is:

$$x_i^k = \sum_{j=1}^{i-1} \Big(- \frac{a_{ij} x_j^k}{a_{ii}} \Big) + \sum_{j=i+1}^{n} \Big(- \frac{a_{ij} x_j^{k-1}}{a_{ii}} \Big) + \frac{b_i}{a_{ii}},\ j \neq i,\ a_{ii} \neq 0,$$

for $i = 1, \cdots, n$, and $k = 1, \cdots, n$.

It is also known as **Gauss-Sedel iterative process** or **Gauss-Sedel iterative technique**

Example 35. *Solve the following linear system using Gauss-Siedel iterative method*

$$\begin{aligned} 2x_1 - 4x_2 + x_3 &= -1 \\ x_1 + x_2 + 6x_3 &= 1 \\ 3x_1 + 3x_2 + 5x_3 &= 4 \end{aligned}$$

.

Solution: Rearrange the system in above such that the coefficient matrix is strictly diagonally dominant

$$\begin{aligned} 3x_1 + 3x_2 + 5x_3 &= 4 \\ 2x_1 - 4x_2 + x_3 &= -1 \\ x_1 + x_2 + 6x_3 &= 1 \end{aligned}$$

.

These equations can be written in the form

$$\begin{aligned} x_1 &= \frac{4 - 3x_2 - 5x_3}{3}, \\ x_2 = \frac{-1 - 2x_1 - x_3}{-4} &= \frac{1 + 2x_1 + x_3}{4}, \\ x_3 &= \frac{1 - x_1 - x_2}{6}. \end{aligned}$$

This suggests the following Gauss-Siedel iterative process

$$\begin{aligned} x_1^{n+1} &= \frac{4 - 3x_2^n - 5x_3^n}{3}, \\ x_2^{n+1} &= \frac{1 + 2x_1^{n+1} + x_3^n}{4}, \\ x_3^{n+1} &= \frac{1 - x_1^{n+1} - x_2^{n+1}}{6}. \end{aligned}$$

We start with initial guess $P_0 = (x_1^0, x_2^0, x_3^0) = (1, 0.1, -1)$. Substitute $x_2^0 = 0.1$ and $x_3^0 = -1$ in the first equation and have

$$x_1^1 = \frac{4 - 3x_2^0 - 5x_3^0}{3} = \frac{4 - 3(0.1) - 5(-1)}{3} = \frac{8.7}{3} = 2.9.$$

Then, substitute the new value $x_1^1 = 2.9$ and $x_3^0 = -1$ into the second equation to obtain

$$x_2^1 = \frac{1 + 2x_1^1 + x_3^0}{4} = \frac{1 + 2(2.9) + (-1)}{4} = 1.45.$$

Finally, substitute the new values $x_1^1 = 2.9$ and $x_2^1 = 1.45$ in the third equation and get

$$x_3^1 = \frac{1 - x_1^1 - x_2^1}{6} = \frac{1 - 2.9 - 1.45}{6} = \frac{-3.35}{6} = -0.558333333333333.$$

Now, we have the now point $P_1 = (x_1^1, x_2^1, x_3^1) = (2.9, 1.45, -0.558333333333333)$ is used to find the next approximation P_2.

Substitute $x_2^1 = 1.45$ and $x_3^1 = -0.558333333333333$ in the first equation and get

$$\begin{aligned} x_1^2 &= \frac{4 - 3x_2^1 - 5x_3^1}{3} = \frac{4 - 3(1.45) - 5(-0.558333333333333)}{3} \\ &= \frac{2.441666666666666}{3} = 0.813888888888889. \end{aligned}$$

Then, substitute the new value $x_2^1 = 0.813888888888889$ and $x_3^1 = -0.558333333333333$ into the second equation to obtain

$$\begin{aligned} x_2^2 &= \frac{1 + 2x_1^2 + x_3^1}{4} = \frac{1 + 2(0.813888888888889) + (-0.558333333333333)}{4} \\ &= \frac{2.069444444444445}{4} = 0.517361111111111. \end{aligned}$$

Finally, substitute the new values $x_2^1 = 0.813888888888889$ and $x_2^2 = 0.517361111111111$ in the third equation and get

$$\begin{aligned} x_3^2 &= \frac{1 - x_1^2 - x_2^2}{6} = \frac{1 - 0.813888888888889 - 0.517361111111111}{6} \\ &= \frac{-0.331250000000000}{6} = -0.055208333333333. \end{aligned}$$

This iteration process generates a sequence of points $\{P_n\} = \{(x_1^n, x_2^n, x_3^n)\}$ that converges to the solution $(x_1, x_2, x_3) = (32/39, 25/39, -1/13) = (0.820512820512820, 0.641025641025641, -0.076923076923077)$. The results are given in the Table 3.2.

n	x_1^n	x_2^n	x_3^n
0	1.000000000000000	0.100000000000000	−1.000000000000000
1	2.900000000000000	1.450000000000000	−0.558333333333333
2	0.813888888888889	0.517361111111111	−0.055208333333333
3	0.907986111111111	0.690190972222222	−0.099696180555556
4	0.809302662037037	0.629727285879630	−0.073171657986111
5	0.825558810763889	0.644486490885417	−0.078340883608218
6	0.819414981794946	0.640122269995419	−0.076589541965061
7	0.820860299946349	0.641282764481909	−0.077023844071376
8	0.820423642303718	0.640955860134015	−0.076896583739622
9	0.820538446098689	0.641045077114439	−0.076930587202188
10	0.820505901555874	0.641020303977390	−0.076921034255544
11	0.820514753115183	0.641027117993706	−0.076923645184815
12	0.820512290647653	0.641025234027623	−0.076922920779213
13	0.820512967271065	0.641025753440729	−0.076923120118632
14	0.820512780090325	0.641025610015504	−0.076923065017638
15	0.820512831680559	0.641025649585870	−0.076923080211072
16	0.820512817432583	0.641025638663523	−0.076923076016018
17	0.820512821363173	0.641025641677582	−0.076923077173459
18	0.820512820278183	0.641025640845727	−0.076923076853985
19	0.820512820577581	0.641025641075294	−0.076923076942146
20	0.820512820494949	0.641025641011938	−0.076923076917814

Table 3.2: Gauss-Siedel Iterative Solution of Example 35

Exercises

Exercise 20. *Solve Example 27 Using Gauss elimination with forward substitution method. Compare the solution with solution of the same example.*

Exercise 21. *Solve Example 28 Using Gauss elimination with backward substitution method. Compare the solution with solution of the same example.*

Exercise 22. *Repeat Example 34 with Gauss-Siedel iteration. Compute five iterations and compare them with Jacobi iterations in the same example.*

Exercise 23. *Redo Example 35 with Jacobi iteration. Compute five iterations and compare them with Gauss-Siedel iterations in the same example.*

Exercise 24. *Use Gauss elimination with backward substitution method and three-digit rounding arithmetic to solve the following linear system*

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

.

Exercise 25. *(a) Determine the LU factorisation for matrix A in the linear system $AX = B$, where*

$$A = \begin{bmatrix} -1 & 1 & -2 \\ 2 & -1 & 1 \\ -4 & 1 & -2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}.$$

(b) Then use the factorisation to solve the system

$$\begin{aligned} -x_1 + x_2 - 2x_3 &= 2 \\ 2x_1 - x_2 + x_3 &= 1 \\ -4x_1 + x_2 - 2x_3 &= 4 \end{aligned}$$

.

Exercise 26. *Solve the following linear system using Gauss-Jordan elimination method*

$$\begin{aligned} -4x_1 - x_2 - 2x_3 &= -9 \\ -x_1 - x_2 + 3x_3 &= 9 \\ -2x_1 - 4x_2 + x_3 &= 5 \end{aligned}$$

.

Chapter 4

Curve Fitting and Approximation Theory

4.1 Introduction

In many scientific and engineering applications, scientists and engineers need to find the best fitting curve for experimental data for which there is no known function. **Curve fitting** is a branch of numerical analysis studies the mathematical framework for finding the best fitting (closest) curves for given sets of empirical data.

4.1.1 Linear Least Squares

In scientific applications experiments sometimes produce a collection of data points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ where their abscissas are different. We need to find a formula (function) $y = f(x)$ that relates these variables. We assume that the relation between these variables is linear and hence we can write the formula as

$$y = f(x) = ax + b. \tag{4.1}$$

Suppose that the numerical values $\{x_k\}$ and $\{y_k\}, k = 1, \cdots, n$ are accurate to several significant digits, i.e. the satisfy the following error formula

$$f(x_k) = y_k + e_k,\ 1 \leq k \leq n, \tag{4.2}$$

the quantities e_k are the errors and also called the **deviations** or **residuals**

$$e_k = y_k - f(x_k),\ 1 \leq k \leq n. \tag{4.3}$$

Now, our goal is to measure how far the curve $y = f(x)$ lies away from the data points. In fact, there are many norms to measure that and the most commonly used are:

$$E_1(f) = \frac{1}{n}\sum_{k=1}^{n} |f(x_k) - y_k|, \text{ **Average Error Norm**}, \tag{4.4}$$

$$E_2(f) = \Big(\frac{1}{n}\sum_{k=1}^{n} |f(x_k) - y_k|^2\Big)^{1/2}, \text{ **Root Mean Square Error Norm**}, \tag{4.5}$$

$$E_\infty(f) = \max_{k=1,\cdots,n} \Big\{|f(x_k) - y_k|\Big\}, \text{ **Maximum Error Norm**}. \tag{4.6}$$

The maximum error E_∞ is obviously the largest since it gives the maximum value of the error. The average error E_1 is widely used since it is simple and easy to compute, it averages the absolute values of the error at the given data points. The E_2 is the most commonly used in statistics and it is a traditional choice, it is simply computes the square root of the mean of error squares. To find the best fitting line to a set of given data we need to minimise the quantities in the three equations above (4.4)-(4.6). We choose the error E_2 since it is easy to minimise computationally.

Let $\{(x_k, y_k)\}_{k=1}^{n}$ be a set of given data points such that their abscissas are distinct. The line in (4.1) is called the **least-squares line** because it minimises the E_2 error. The error E_2 is a minimum if and only if (iff) the expression $n(E_2(f))^2 = \sum_{k=1}^{n} |f(x_k) - y_k|^2$ is a minimum. To find the unknown coefficients of the least-squares line, we solve linear system of two equations

$$\begin{aligned} \left(\sum_{k=1}^{n} x_k^2\right) a \quad + \quad \left(\sum_{k=1}^{n} x_k\right) b &= \sum_{k=1}^{n} x_k y_k, \\ \left(\sum_{k=1}^{n} x_k\right) a \quad + \quad nb &= \sum_{k=1}^{n} y_k, \end{aligned} \tag{4.7}$$

these equations are called the **normal equations**. Upon solving this linear system for a and b, we have

$$a = \frac{\sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k - \sum_{k=1}^{n} x_k y_k \sum_{k=1}^{n} x_k}{n\left(\sum_{k=1}^{n} x_k^2\right) - \left(\sum_{k=1}^{n} x_k\right)^2}, \tag{4.8}$$

$$b = \frac{n\sum_{k=1}^{n} x_k y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} y_k}{n\left(\sum_{k=1}^{n} x_k^2\right) - \left(\sum_{k=1}^{n} x_k\right)^2}. \tag{4.9}$$

Chapter 5

Interpolation and Extrapolation

5.1 Introduction

In applied sciences and engineering, scientists and engineers often collect a number of data points of different scientific phenomena via experimentation and sampling. In many cases they need to estimate (interpolate) a function at a point its functional value is not in the range of the collected data. Interpolation is a branch of numerical analysis studies the methods and techniques of estimating an unknown value of a function at an intermediate value of its independent variable. Also, interpolation is used to replace a complicated function by a simpler one.

5.2 Lagrange Interpolation

Suppose that we would like to interpolate an arbitrary function f at a set of limited points $x_0, x_1, \cdots, x_n$. These $n+1$ points are known as **interpolation nodes** in interpolation theory. Firstly, we need to introduce a system of $n+1$ special polynomials of degree n known as **interpolating polynomials** or **cardinal polynomials**. These polynomials are denoted by $\ell_0, \ell_1, \cdots, \ell_n$ and defined using Kronecker delta notation as follows

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

.

Then, we can interpolate the function f by a polynomial P_n of degree n defined by

$$P_n(x) = \sum_{i=0}^{n} \ell_i(x) f(x_i),$$

this polynomial is called **Lagrange polynomial** or **Lagrange form of the interpolation polynomial**, and it is a linear combination of the cardinal polynomials $\ell_i, i = 0, 1, \cdots, n$. Moreover, it coincides with the function f at the the nodes $x_j, j = 0, 1, \cdots, n$, namely

$$P_n(x_j) = \sum_{i=0}^{n} \ell_i(x_j) f(x_j) = \ell_j(x_j) f(x_j) = f(x_j).$$

The interpolating polynomials can be expressed as a product of n linear factors

$$\ell_i(x) = \prod_{j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)},\ i = 0, 1, \cdots, .$$

i.e.

$$\ell_i(x) = \frac{(x - x_0)}{(x_i - x_0)} \frac{(x - x_1)}{(x_i - x_1)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)}.$$

Example 36. *Determine the linear Lagrange polynomial that passes through the points* $(1, 5)$ *and* $(4, 2)$ *and use it to interpolate the linear function at* $x = 3$.

Solution: Writing out the cardinal polynomials

$$\ell_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} = \frac{(x - 4)}{(1 - 4)} = \frac{-1}{3}(x - 4),$$

and

$$\ell_1(x) = \frac{(x - x_0)}{(x_1 - x_0)} = \frac{(x - 1)}{(4 - 1)} = \frac{1}{3}(x - 1).$$

Hence, the Lagrange polynomial is

$$P_1(x) = \sum_{i=0}^{1} \ell_i(x) f(x_i) = \ell_0(x) f(x_0) + \ell_1(x) f(x_1) =$$

$$\frac{-1}{3}(x - 4)(5) + \frac{1}{3}(x - 1)(2) = -x + 6.$$

So,

$$P_1(3) = -(3) + 6 = 3.$$

Note that

$$P_1(1) = -(1) + 6 = 5 = f(1), \text{ and } P_1(4) = -(4) + 6 = 2 = f(4).$$

Example 37. *Find the Lagrange polynomial that interpolates the following data*

x	1	2	2.5	3	4	5
$f(x)$	0	5	6.5	7	3	1

Solution: The cardinal polynomials are:

$$\begin{aligned}\ell_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)(x_0-x_5)}\\ &= \frac{(x-2)(x-2.5)(x-3)(x-4)(x-5)}{(1-2)(1-2.5)(1-3)(1-4)(1-5)}\\ &= -\frac{1}{36}x^5 + \frac{11}{24}x^4 - \frac{53}{18}x^3 + \frac{221}{24}x^2 - \frac{505}{36}x + \frac{25}{3},\end{aligned}$$

$$\begin{aligned}\ell_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)(x_1-x_5)}\\ &= \frac{(x-1)(x-2.5)(x-3)(x-4)(x-5)}{(2-1)(2-2.5)(2-3)(2-4)(2-5)}\\ &= \frac{1}{3}x^5 - \frac{31}{6}x^4 + \frac{61}{2}x^3 - \frac{509}{6}x^2 + \frac{655}{6}x - 50,\end{aligned}$$

$$\begin{aligned}\ell_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)(x-x_5)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)(x_2-x_5)}\\ &= \frac{(x-1)(x-2)(x-3)(x-4)(x-5)}{(2.5-1)(2.5-2)(2.5-3)(2.5-4)(2.5-5)}\\ &= -\frac{5000}{7031}x^5 + \frac{75000}{7031}x^4 - \frac{425000}{7031}x^3 + \frac{1125000}{7031}x^2 - \frac{1370000}{7031}x + \frac{600000}{7031},\end{aligned}$$

$$\begin{aligned}
\ell_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_4)(x-x_5)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)(x_3-x_4)(x_3-x_5)} \\
&= \frac{(x-1)(x-2)(x-2.5)(x-4)(x-5)}{(3-1)(3-2)(3-2.5)(3-4)(3-5)} \\
&= \frac{1}{2}x^5 - \frac{29}{4}x^4 + \frac{79}{2}x^3 - \frac{401}{4}x^2 + \frac{235}{2}x - 50,
\end{aligned}$$

$$\begin{aligned}
\ell_4(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_5)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)(x_4-x_5)} \\
&= \frac{(x-1)(x-2)(x-2.5)(x-3)(x-5)}{(4-1)(4-2)(4-2.5)(4-3)(4-5)} \\
&= -\frac{1}{9}x^5 + \frac{3}{2}x^4 - \frac{137}{18}x^3 + \frac{109}{6}x^2 - \frac{365}{18}x + \frac{25}{3},
\end{aligned}$$

$$\begin{aligned}
\ell_5(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_5-x_0)(x_5-x_1)(x_5-x_2)(x_5-x_3)(x_5-x_4)} \\
&= \frac{(x-1)(x-2)(x-2.5)(x-3)(x-4)}{(5-1)(5-2)(5-2.5)(5-3)(5-4)} \\
&= \frac{1}{60}x^5 - \frac{5}{24}x^4 + x^3 - \frac{55}{24}x^2 + \frac{149}{60}x - 1.
\end{aligned}$$

Hence, the Lagrange polynomial is

$$
\begin{aligned}
P_5(x) = \sum_{i=0}^{6} \ell_i(x) f(x_i) = \ell_i(x)f(x_i) + \ell_i(x)f(x_i) + \\
\ell_i(x)f(x_i) + \ell_i(x)f(x_i) + \ell_i(x)f(x_i) + \ell_i(x)f(x_i) = \\
-\frac{1}{36}x^5 + \frac{11}{24}x^4 - \frac{53}{18}x^3 + \frac{221}{24}x^2 - \frac{505}{36}x + \frac{25}{3}\,(0) + \\
\frac{1}{3}x^5 - \frac{31}{6}x^4 + \frac{61}{2}x^3 - \frac{509}{6}x^2 + \frac{655}{6}x - 50\,(5) + \\
-\frac{5000}{7031}x^5 + \frac{75000}{7031}x^4 - \frac{425000}{7031}x^3 + \frac{1125000}{7031}x^2 - \frac{1370000}{7031}x + \frac{600000}{7031}\,(6.5) + \\
\frac{1}{2}x^5 - \frac{29}{4}x^4 + \frac{79}{2}x^3 - \frac{401}{4}x^2 + \frac{235}{2}x - 50\,(7) + \\
-\frac{1}{9}x^5 + \frac{3}{2}x^4 - \frac{137}{18}x^3 + \frac{109}{6}x^2 - \frac{365}{18}x + \frac{25}{3}\,(3) + \\
\frac{1}{60}x^5 - \frac{5}{24}x^4 + x^3 - \frac{55}{24}x^2 + \frac{149}{60}x - 1\,(1).
\end{aligned}
$$

After some mathematical manipulation, we have

$$
\begin{aligned}
P_5(x) = -\frac{8}{316395}x^5 + \frac{8}{21093}x^4 - \frac{2722272566677}{1266637395197952}x^3 + \frac{200167100491}{35184372088832}x^2 \\
-\frac{10969157106929}{1583296743997440}x - \frac{187476506320011}{8796093022208}.
\end{aligned}
$$

Note that the Lagrange interpolant is used to interpolate a function at a set of non-equally spaced points.

5.3 Newton's Difference Interpolation Formula

Newton's interpolation formula is used to interpolate a function at a set of given equally spaced points $x_0, x_1, \cdots, x_n$. Before we start we need to define the finite divided differences.

5.3.1 Finite Divided Differences

The first finite divided difference of the function f is in general given by

$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}.$$

The second finite divided difference is the difference between the two divided difference, is represented by

$$f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}.$$

Likewise, the *nth* finite divided difference is expressed by

$$f[x_0, x_1, \cdots, x_{n-1}, x_n] = \frac{f[x_1, \cdots, x_{n-1}, x_n] - f[x_0, x_1, \cdots, x_{n-1}]}{x_n - x_0}.$$

Note that the zero-order difference is defined as

$$f[x_i] = f(x_i) = f_i.$$

Also, observe that

$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i} = \frac{f(x_i) - f(x_j)}{x_i - x_j} = f[x_j, x_i].$$

The divided differences is summarised in the divided difference table given below:

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	f_0	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
x_1	f_1	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$
x_2	f_2	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$	
x_3	f_3	$f[x_3, x_4]$		
x_4	f_4			

Example 38. *Compute the divided differences of the following data*

x	0.5000	1.000	1.500	2.000	2.5000
$f(x)$	1.1250	3.000	7.3750	15.0000	26.6250

Solution: Using the standard notation the first finite divided differences are:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{3 - 1.1250}{1 - 0.5} = \frac{1.8750}{0.5} = 3.7500.$$

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{7.3750 - 3}{1.5 - 1} = \frac{4.3750}{0.5} = 8.7500.$$

$$f[x_2, x_3] = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{15 - 7.3750}{2 - 1.5} = \frac{7.6250}{0.5} = 15.2500.$$

$$f[x_3, x_4] = \frac{f(x_4) - f(x_3)}{x_4 - x_3} = \frac{26.6250 - 15}{2.5 - 2} = \frac{11.6250}{0.5} = 23.2500.$$

Now, using the computed first divided differences, we compute the second divided differences

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{8.7500 - 3.7500}{1.5 - 0.5} = \frac{5}{1} = 5.000.$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{8.7500 - 3.7500}{1.5 - 0.5} = \frac{5}{1} = 5.000.$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{15.2500 - 8.7500}{2 - 1} = \frac{6.5000}{1} = 6.5000.$$

$$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2} = \frac{23.2500 - 15.2500}{2.5 - 1.5} = \frac{8.0000}{1} = 8.0000.$$

Finally, we compute the third divided differences using the computed second divided differences

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{6.5000 - 5.0000}{2 - 0.5} = \frac{1.5000}{1.5000} = 1.0000.$$

$$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1} = \frac{8.0000 - 6.5000}{2.5 - 1} = \frac{1.5000}{1.5000} = 1.0000.$$

The results are outlined in the following table

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0.5000	1.1250	3.7500	5.000	1.0000
1.000	3.000	8.7500	6.5000	1.0000
1.5000	7.3750	15.2500	8.0000	
2.000	15.0000	23.2500		
2.5000	26.6250			

5.3.2 Newton's Interpolation Divided Difference Formula

The general form of Newton's interpolation polynomial of order n for $n+1$ data points is

$$\begin{aligned}P_n(x) = d_0 + d_1(x-x_0) + d_2(x-x_0)(x-x_1) + d_3(x-x_0)(x-x_1)(x-x_2) + \\ \cdots + d_n(x-x_0)(x-x_1) + d_3(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1}),\end{aligned}$$

where

$$d_0 = f[x_0],$$

$$d_1 = f[x_0, x_1],$$

$$d_2 = f[x_0, x_1, x_2],$$

$$d_3 = f[x_0, x_1, x_2, x_3],$$

$$\vdots$$

$$d_n = f[x_0, x_1, \cdots, x_n],$$

Example 39. *Use the data from Example 38 to construct Newton's interpolation divided difference formula, and use it to evaluate $f(0)$, $f(3)$, and $f(3.25)$.*

Solution: The Newton's polynomial of third order for the data in the table above is

$$P_3(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + d_3(x - x_0)(x - x_1)(x - x_2) = \\ 1.1250 + 3.75(x - 0.5) + 5(x - 0.5)(x - 1) + (x - 0.5)(x - 1)(x - 1.5).$$

After some mathematical manipulation, we have

$$P_3(x) = x^3 + 2x^2 - x + 1.$$

Hence,

$$f(0) = P_3(0) = 1, \ f(3) = P_3(3) = 43, \ f(3.25) = P_3(3.25) = 53.2031.$$

5.4 Extrapolation

In numerical analysis, **extrapolation** is used to estimate a value of a function at a point beyond the range of its known values $x_0, x_1, \cdots, x_n$. Extrapolation compared to interpolation is more likely to produce meaningless results. There are many methods in extrapolation, in these notes, we consider the linear and polynomial extrapolation.

5.4.1 Linear Extrapolation

Linear extrapolation is used to estimate an approximately linear function by extending its graph not far away from its known values. Assume that we have a set of values of some unknown function f at some points in its domain. Let the function f has values y_{n-1} and y_n at the points x_{n-1} and x_n respectively. We can estimate the function value y at the point x near the points x_{n-1} and x_n by constructing a tangent line to the data points (x_{n-1}, y_{n-1}) and (x_n, y_n) to obtain

$$y(x) = y_{n-1} + \frac{x - x_{n-1}}{x - x_n}(y_n - y_{n-1}).$$

Note that when $x_{n-1} < x < x_n$ then the extrapolation is turned to interpolation process.

Example 40. *Use the following two points* $(0.1, 1.1)$ *and* $(0.35, 1.35)$ *to extrapolate the value of the unknown function* f *at* $x = 0$.

Solution: The general formula of the linear extrapolation is

$$y(x) = y_1 + \frac{x - x_1}{x - x_2}(y_2 - y_1).$$

Here $x_1 = 0.1, y_1 = 1.1, x_2 = 0.35$, and $y_2 = 1.35$.

Substituting these values in the linear extrapolation formula, we have

$$y(0) = 1.1 + \frac{0 - 0.1}{0 - 0.35}(1.35 - 1.1) = 1.1 + \frac{0.1}{0.35}(0.25) = 1.1714.$$

5.4.2 Polynomial Extrapolation

To approximate a function by a high order polynomial near the end of a given set of data or at a point beyond the original observed values, Lagrange interpolation or Newton interpolation can be used to extrapolate the resulting polynomial at the required data. Care has to be taken since the extrapolation error will grow due to the **Runge's Phenomenon**.

Example 41. *Use the data in the table below to extrapolate* $f(600)$

x	300	400	500
$f(x)$	0.616	0.525	0.457

Solution: Constructing the divided difference table of the data

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$
300	0.616	-0.00091	0.00000115
400	0.525	-0.00068	
500	0.457		

From the table in above, we can write out the Newton's difference polynomial as follows:

$$P_2(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) =$$
$$0.616 - 0.00091(x - 300) + 0.00000115(x - 300)(x - 400).$$

After some simplifications, we have

$$P_2(x) = 0.00000115\,x^2 - 0.00175\,x + 1.0270.$$

So

$$f(300) \approx P_2(300) = 0.00000115(300)^2 - 0.00175(300) + 1.0270 = 0.391.$$

5.5 Some Important MATLAB Functions in Numerical Analysis

Using *fzero* function to evaluate roots of an equation

The MATLAB function fzero is used to find the real roots of a single equation. The general syntax of the function is

```
>> fzero(f, r_0)
```

where f is the function which we want to find its roots and r_0 is the initial guess. Let us consider the function in Example 14 for using Newton-Raphson Method

```
>> fzero(@(x) 3*x-exp(x), 1.5)

ans =

    1.512134551657843
```

There is another version of this function is

```
>> fzero(f, [a  b])
```

where a and b are such that $a \leq r \leq b$. For example, consider the problem in Example 10

```
>> fzero(@(x) x*sin(x)-1, [0.5 1.5])

ans =

    1.114157140871930
```

The MATLAB builtin function *roots* is a powerful tool for finding the roots of the polynomials and its syntax is

```
>> x = roots(c)
```

where c is the vector of coefficients of the underlying polynomial. For example, applying this function for the polynomial in Example 11, we have

```
>> c = [2 -1 1 -1];
x = roots(c)

x =

  -0.119491810752253 + 0.813834558901752i
  -0.119491810752253 - 0.813834558901752i
   0.738983621504507 + 0.000000000000000i
```

Also, we can use the MATLAB function *polyval(c, x)* to evaluate the value of the polynomial whose coefficient vector is c at x.

```
>> x = 0;
>> c = [1 1 1];
>> y = polyval(c,x)

y =

     1
```

To solve a linear system $AX = B$ in MATLAB we can use the backslash or "Left Division" operator as follows

```
>> X = A\ B
```

or, using the inverse function as in

```
>> X = inv(A)*B
```

For example, consider the solution of the linear system in Example 27

```
>> A = [1 2 -1 4; 2 1 1 1; -3 -1 4 1; 1 1 -1 3];
>> B = [12 10 2 6]';
>> X=A\B

X =

    0.857142857142858
    5.285714285714286
    2.285714285714286
    0.714285714285714

>> rat(X)

ans =

   4 18 char array

      '1 + 1/(-7)         '
      '5 + 1/(4 + 1/(-2))'
      '2 + 1/(3 + 1/(2))  '
      '1 + 1/(-4 + 1/(2))'
```

or

```
>> X = inv(A)*B

X =

    0.857142857142857
    5.285714285714288
    2.285714285714286
    0.714285714285714
```

We note that the solutions obtained using these two MATLAB functions are equal.

MATLAB has a builtin function *lu* which can be used to find the *LU* (Doolittle) factorisation of a given matrix *A* as following

```
>> [L U] = lu(A)
```

as in the following example (Example)

```
>> A = [2 -3 1; 1 1 -1; -1 1 -1];
>> [L U] = lu(A)

L =

    1.0000         0         0
    0.5000    1.0000         0
   -0.5000   -0.2000    1.0000


U =

    2.0000   -3.0000    1.0000
         0    2.5000   -1.5000
         0         0   -0.8000
```

We can check that the results are correct by computing the original matrix *A* as

```
>> L*U

ans =

     2    -3     1
     1     1    -1
    -1     1    -1
```

Now, we can solve the problem by implementing the following steps:

```
>> B = [2; -1; 0];
>> Y = L\B
```

```
Y =

     2.0000
    -2.0000
     0.6000

>> X = U\Y

X =

    -0.5000
    -1.2500
    -0.7500
```

Also, Cholesky decomposition of a given matrix A can be computed by using the MATLAB builtin function *chol* which has the following syntax

```
>> U = chol(A)
```

where A is the given matrix and U is the upper triangular matrix such that $A = U'U$. As an example we consider the following problem.

```
>> A = [2 1 0; 1 2 1; 0 1 2];
>> U = chol(A)

U =

     1.4142    0.7071         0
          0    1.2247    0.8165
          0         0    1.1547
```

We can test now the correctness of the factorisation of the matrix A by computing it as

```
>> U'*U

ans =

     2.0000    1.0000         0
```

```
    1.0000    2.0000    1.0000
         0    1.0000    2.0000
```

To generate the solution, we solve the following lower and upper triangular linear systems respectively

```
>> B = [-1 -4 2]';
>> Y = U'\B

Y =

   -0.7071
   -2.8577
    3.7528

>> X = U\Y

X =

    1.7500
   -4.5000
    3.2500
```

MATLAB has another powerful tool for solving linear and as well nonlinear systems which is *solve* function and the general syntax of this function is

```
[S1,S2,...,SN] = solve(eq1,eq2,...,eqN)
```

where $eq1, eq2, \cdots, eqN$ are symbolic equations (also can be symbolic inequalities and expressions), and $S1, S2, \cdots, SN$ is the required solution to this system of equations. For example, let us consider the linear system in Example 30:

```
>> syms x1 x2 x3
>> [x1,x2,x3] = solve(-2*x1 + x2 + 5*x3 == 15, 4*x1 - 8*x2 +
x3 == -21, 4*x1 - x2 + x3 == 7)

x1 =

2
```

```
x2 =

4


x3 =

3
```

Now, we solve the nonlinear problem in Example 17 using this function to have

```
>> syms x
>> solve(x^3+3*x-2)

ans =

root(z^3 + 3*z - 2, z, 1)
```

The MATLAB builtin function *polyfit* can be used to find the coefficients of the polynomial $P(x)$ of degree n that fits the data $y_0 = f(x_0)$, $y_1 = f(x_1), \cdots, y_n = f(x_n)$, its syntax is

```
polyfit(x,y,n)
```

We apply this function for solving the problem in Example 38

```
>> x = [0.5 1 1.5 2 2.5];
>> y = [1.1250 3.000 7.3750 15.0000 26.6250];
>> p = polyfit(x,y,4)

p =

-0.000000000000003    1.000000000000026    1.999999999999929
-0.999999999999921    0.999999999999972
```

Now, we can use the function *polyval* to compute

```
>> x_data = [0 3 3.25];
>> y_data = polyval(p,x_data)

y_data =

   0.999999999999972   42.999999999999993   53.203124999999986
```

Chapter 6

Numerical Differentiation

6.1 Introduction

Computing the derivative of an unknown function or sometimes complicated function is not uneasy task and it is of essential importance in solving the ordinary and partial differential equations.

6.2 Differentiation Formulas

We know from calculus that the derivative of a function f at x_0 is

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0+h) - f(x_0)}{h}.$$

To derive the finite difference approximations of the derivatives of a function f at x_0 we need the forward and backward Taylor series expansions of f as follows:

$$f(x+h) = f(x) + hf^{'}(x) + \frac{h^2}{2!}f^{''} + \frac{h^3}{3!}f^{'''} + \frac{h^4}{4!}f^{(4)} + \cdots \quad (6.1)$$

$$f(x-h) = f(x) - hf^{'}(x) + \frac{h^2}{2!}f^{''} - \frac{h^3}{3!}f^{'''} + \frac{h^4}{4!}f^{(4)} - \cdots \quad (6.2)$$

$$f(x+2h) = f(x) + 2hf^{'}(x) + \frac{(2h)^2}{2!}f^{''} + \frac{(2h)^3}{3!}f^{'''} + \frac{(2h)^4}{4!}f^{(4)} + \cdots \quad (6.3)$$

$$f(x-2h) = f(x) - 2hf^{'}(x) + \frac{(2h)^2}{2!}f^{''} - \frac{(2h)^3}{3!}f^{'''} + \frac{(2h)^4}{4!}f^{(4)} - \cdots \quad (6.4)$$

The sum of (6.1) and (6.2) give us

$$f(x+h)+f(x-h)=2f(x)+h^2f^{'}(x)+h^2f^{''}+\frac{h^4}{12}f^{(4)}+\cdots \tag{6.5}$$

The difference of (6.1) and (6.2) is

$$f(x+h)-f(x-h)=2hf^{'}(x)+\frac{h^3}{3}f^{('''})+\cdots . \tag{6.6}$$

Adding (6.3) and (6.4) yields

$$f(x+2h)+f(x-2h)=2f(x)+4h^2f^{''}+\frac{4h^4}{3}f^{(4)}+\cdots . \tag{6.7}$$

Subtracting (6.4) from (6.3) results in

$$f(x+2h)-f(x-2h)=4hf^{'}+\frac{8h^3}{3}f^{('''})+\cdots . \tag{6.8}$$

We notice that the sums formulas contain even derivatives while the difference formulas contain odd derivatives. Solving Eq. (6.1) for $f^{'}(x)$ yields the **forward difference formula**

$$f'(x)=\frac{f(x+h)-f(x)}{h}. \tag{6.9}$$

From Eq. (6.1) we obtain the **backward difference formula**

$$f'(x)=\frac{f(x)-f(x-h)}{h}. \tag{6.10}$$

By solving Eq. (6.6) for $f^{'}(x)$ we obtain the **central difference formula**

$$f'(x)=\frac{f(x+h)-f(x-h)}{2h}. \tag{6.11}$$

Chapter 7

Numerical Integration

7.1 Introduction

In solving daily life problems, we sometimes encountered with integrations problems whose integrals cannot be computed analytically. In these scenarios we resort to numerical integration to integrate these problems numerically by using approximating methods. In these lecture notes, we focus on Newton-Cotes formulas of integration.

7.2 Newton-Cotes Formulas of Integration

Newton-Cotes quadrature formulas are numerical formulas used to approximate the definite integral $\int_a^b f(x)\,dx$. In these formulas the function of integration or the **integrand** is replaced by an interpolating polynomial. These integration rules are said to be **closed** if they include the function values at the end of the integration interval. Otherwise, they are called **open**.

7.2.1 Closed Newton-Cotes Integration Rules

The general structure of these formulas is as follows: Let $a = x_0, b = x_n$ and the step size $h = \frac{b-a}{n}$, then the internal integration points can be defined by $x_i = x_0 + ih, i = 1, \cdots, n$, where $f_i = f(x_i), i = 0, \cdots, n$. Here, we consider some of these closed rules such as:

(a) **Trapezoid Rule**:

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{1}{2}h[f_0 + f_1].$$

(b) **Simpson's $\frac{1}{3}$ Rule**:

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{1}{3}h[f_0 + 4f_1 + f_2].$$

(c) **Simpson's $\frac{3}{8}$ Rule**:

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3}{8}h[f_0 + 3f_1 + 3f_2 + f_3].$$

(d) **Boole's Rule**:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2}{45}h[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4].$$

(e) **Six-Point Newton-Cotes Closed Rule**:

$$\int_{x_0}^{x_5} f(x)\,dx = \frac{5}{288}h[19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5].$$

Example 42. *Consider the function $f(x) = e^{x^2}$, use the above-mentioned quadrature rules to approximate the integral $\int_0^{0.6} e^{x^2}\,dx$.*

Solution: Let us divide the interval of integration $I = [a, b] = [0, 0.6]$ to six equal subintervals (i.e. $n = 6$). Hence, $h = (b - a)/n = (0.6 - 0)/6 = 0.1$. The function values at the end points are:

$$f_0 = f(x_0) = e^{x_0^2} = e^{(0)^2} = 1,$$

and

$$f_6 = f(x_6) = e^{x_6^2} = e^{(0.6)^2} = 1.43332941.$$

Now, we compute the points of integration $x_i = x_0 + ih, i = 1, \cdots, 6$.

So, when $i = 1$

$$x_1 = x_0 + h = 0 + 0.1 = 0.1,\ f_1 = f(x_1) = e^{x_1^2} = e^{(0.1)^2} = 1.01005017,$$

when $i = 2$

$$x_2 = x_0 + 2h = 0 + 2(0.1) = 0.2,\ f_2 = f(x_2) = e^{x_2^2} = e^{(0.2)^2} = 1.04081078,$$

when $i = 3$

$$x_3 = x_0 + 3h = 0 + 3(0.1) = 0.3,\ f_3 = f(x_3) = e^{x_3^2} = e^{(0.3)^2} = 1.09417428,$$

when $i = 4$

$$x_4 = x_0 + 4h = 0 + 4(0.1) = 0.4,\ f_4 = f(x_4) = e^{x_4^2} = e^{(0.4)^2} = 1.17351087,$$

when $i = 4$

$$x_5 = x_0 + 5h = 0 + 5(0.1) = 0.5,\ f_5 = f(x_5) = e^{x_5^2} = e^{(0.5)^2} = 1.28402542,$$

The computations are summarised in the table below:

x	$f(x)$
0	1
0.1	1.01005017
0.2	1.04081078
0.3	1.09417428
0.4	1.17351087
0.5	1.28402542
0.6	1.43332941

(a) **Trapezoid Rule**: The interval of integration is $I = [a, b] = [0, 0.6]$, so, $n = 1$. Hence, $h = (b - a)/n = (0.6 - 0)/1 = 0.6$. The function values at the end points $x_0 = 0$ and $x_1 = 0.6$ are:

$$f_0 = f(x_0) = e^{x_0^2} = e^{(0)^2} = 1,$$

and

$$f_1 = f(x_1) = e^{x_6^2} = e^{(0.6)^2} = 1.43332941.$$

Consequently, we get

$$\int_{x_0}^{x_1} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{1}{2}h[f_0+f_1] = \frac{1}{2}(0.6)[1+1.43332941] = 0.72999882.$$

(b) **Simpson's $\frac{1}{3}$ Rule**: We divide the integration interval $I = [a, b] = [0, 0.6]$ to two equal subintervals (i.e. $n = 2$). Hence, $h = (b - a)/n = (0.6 - 0)/2 = 0.3$. The integration nodes are $x_0 = 0$, $x_1 = x_0 + h = 0+(1)(0.3) = 0.3$, and $x_2 = 0.6$. The functional values at the integration nodes are:

$$f_0 = 1,\ f_1 = f(x_1) = e^{x_1^2} = e^{(0.3)^2} = 1.09417428,\ f_2 = 1.43332941.$$

$$\int_{x_0}^{x_2} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{1}{3}h[f_0 + 4f_1 + f_2] = \frac{1}{3}(0.3)[1 + 4(1.09417428)$$
$$+1.43332941] = 0.68100265.$$

(c) **Simpson's $\frac{3}{8}$ Rule**: In this rule, we divide the integration interval $I = [0, 0.6]$ to three equal subintervals (i.e. $n = 3$). Hence, $h = (b - a)/n = (0.6 - 0)/3 = 0.2$. The integration nodes are $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.2) = 0.2$, $x_2 = x_0 + 2h = 0 + 2(0.2) = 0.4$ and $x_3 = 0.6$. The functional values at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.04081078,\ f_2 = 1.17351087,\ f_3 = 1.43332941.$$

$$\int_{x_0}^{x_3} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{3}{8}h[f_0 + 3f_1 + 3f_2 + f_3] = \frac{3}{8}(0.2)[1 + 3(1.04081078)$$
$$+3(1.17351087) + 1.43332941] = 0.68072208.$$

(d) **Boole's Rule**: In Boole's rule, we divide the integration interval $I = [0, 0.6]$ to four equal subintervals, $n = 4$. So, $h = (b-a)/n = (0.6-0)/4 = 0.15$. The integration nodes are $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.15) = 0.15$, $x_2 = x_0 + 2h = 0 + 2(0.15) = 0.3$, $x_3 = x_0 + 3h = 0 + 3(0.15) = 0.45$ and $x_4 = 0.6$. The functional values at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.02275503,\ f_2 = 1.09417428,\ f_3 = 1.22446006,\ f_4 = 1.43332941.$$

$$\int_{x_0}^{x_4} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{2}{45}h[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4]$$
$$= \frac{2}{45}(0.15)[7(1) + 32(1.02275503) + 12(1.09417428)$$
$$+32(1.22446006) + 7(1.43332941)] = 0.68049520.$$

(e) **Six-Point Newton-Cotes Closed Rule**: We divide the integration interval $I = [0, 0.6]$ to five equal subintervals, $n = 5$. So, $h = (b - a)/n = (0.6 - 0)/5 = 0.12$. The integration nodes are $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.12) = 0.12$, $x_2 = x_0 + 2h = 0 + 2(0.12) = 0.24$, $x_3 = x_0 + 3h = 0 + 3(0.12) = 0.36$, $x_4 = x_0 + 4h = 0 + 4(0.12) = 0.48$ and $x_5 = 0.6$. The functional values at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.01450418,\ f_2 = 1.05929119,\ f_3 = 1.13837294,\ f_4 = 1.25910355,$$
$$f_5 = 1.43332941.$$

$$\int_{x_0}^{x_5} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{5}{288}h[19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5]$$
$$= \frac{5}{288}(0.12)[19(1) + 75(1.01450418) + 50(1.05929119)$$
$$+50(1.13837294) + 75(1.25910355) + 19(1.43332941)] = 0.68049384.$$

7.2.2 Open Newton-Cotes Integration Rules

The general structure of these formulas is the same as the general structure of the closed rules except that the two end points $x_0 = a, x_n = b$ and their functional values $f(x_0) = f(a), f(x_n) = f(b)$ are not included in the integrations formulas. We consider the following open rules:

(a) **Midpoint Rule**:

$$\int_{x_0}^{x_2} f(x)\,dx = 2hf_1.$$

(b) **Two-Point Newton-Cotes Open Rule**:

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3}{2}h[f_1 + f_2].$$

(c) **Three-Point Newton-Cotes Open Rule**:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{4}{3}h[2f_1 - f_2 + 2f_3].$$

(d) **Four-Point Newton-Cotes Open Rule**:

$$\int_{x_0}^{x_5} f(x)\,dx = \frac{5}{24}h[11f_1 + f_2 + f_3 + 11f_4].$$

(e) **Five-Point Newton-Cotes Open Rule**:

$$\int_{x_0}^{x_6} f(x)\,dx = \frac{6}{20}h[11f_1 - 14f_2 + 26f_3 - 14f_4 + 11f_5].$$

Example 43. *Redo Example 42 use the above-mentioned quadrature open rules to approximate the integral* $\int_0^{0.6} e^{x^2}\,dx$.

(a) **Midpoint Rule**: We divide the integration interval $I = [0, 0.6]$ to two equal subintervals, $n = 2$. So, $h = (b - a)/n = (0.6 - 0)/2 = 0.3$, so, $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.3) = 0.3$ and $x_2 = 0.6$. The function values at the integration points are:

$$f_0 = 1,\ f_1 = 1.09417428,\ f_2 = 1.43332941.$$

$$\int_{x_0}^{x_2} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = 2hf_1 = 2(0.3)(1.09417428) = 0.65650457.$$

(b) **Two-Point Newton-Cotes Open Rule**: Now, we divide the interval of integration $I = [0, 0.6]$ to three equal subintervals, i.e. $n = 3$. So, $h = (b - a)/n = (0.6 - 0)/3 = 0.2$, so, $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.2) = 0.2$, $x_2 = x_0 + (2)h = 0 + (2)(0.2) = 0.4$ and $x_3 = 0.6$. The functional values at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.04081078,\ f_2 = 1.17351087,\ f_3 = 1.43332941.$$

$$\int_{x_0}^{x_3} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{3}{2}h[f_1 + f_2] = \frac{3}{2}(0.2)[1.04081078 + 1.17351087]$$
$$= 0.66429650.$$

(c) **Three-Point Newton-Cotes Open Rule**: In this rule, the interval of integration $I = [0, 0.6]$ is divided to four equal subintervals, i.e. $n = 4$. So, $h = (b - a)/n = (0.6 - 0)/4 = 0.15$, so, $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.15) = 0.15$, $x_2 = x_0 + (2)h = 0 + (2)(0.15) = 0.3$, $x_3 = x_0 + (3)h = 0 + (3)(0.15) = 0.45$ and $x_4 = 0.6$. The values of the function at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.02275503,\ f_2 = 1.09417428,\ f_3 = 1.22446006,\ f_4 = 1.43332941.$$

$$\int_{x_0}^{x_4} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{4}{3}h[2f_1 - f_2 + 2f_3] = \frac{4}{3}(0.15)[2(1.02275503)$$
$$-1.09417428 + 2(1.22446006)] = 0.68005118.$$

(d) **Four-Point Newton-Cotes Open Rule**: We divide the interval of integration $I = [0, 0.6]$ to five equal subintervals, i.e. $n = 5$. So, $h = (b - a)/n = (0.6 - 0)/5 = 0.12$, hence, $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.12) = 0.12$, $x_2 = x_0 + (2)h = 0 + (2)(0.12) = 0.24$, $x_3 = x_0 + (3)h = 0 + (3)(0.12) = 0.36$, $x_4 = x_0 + (4)h = 0 + (4)(0.12) = 0.48$ and $x_5 = 0.6$. The function values at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.01450418,\ f_2 = 1.05929119,\ f_3 = 1.13837294,\ f_4 = 1.25910355,$$
$$f_5 = 1.43332941.$$

$$\int_{x_0}^{x_5} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{5}{24}h[11f_1 + f_2 + f_3 + 11f_4] = \frac{5}{24}(0.12)[11(1.01450418)$$
$$+1.05929119 + 1.13837294 + 11(1.25910355)] = 0.68018373.$$

(e) **Five-Point Newton-Cotes Open Rule**: In this rule, the interval of integration $I = [0, 0.6]$ is divided to six equal subintervals, i.e. $n = 6$. So, $h = (b-a)/n = (0.6-0)/6 = 0.1$, so, $x_0 = 0$, $x_1 = x_0 + (1)h = 0 + (1)(0.1) = 0.1$, $x_2 = x_0 + (2)h = 0 + (2)(0.1) = 0.2$, $x_3 = x_0 + (3)h = 0 + (3)(0.1) = 0.3$, $x_4 = x_0 + (4)h = 0 + (4)(0.1) = 0.4$, $x_5 = x_0 + (5)h = 0 + (5)(0.5) = 0.4$ and $x_6 = 0.6$. The values of the function at the integration nodes are:

$$f_0 = 1,\ f_1 = 1.01005017,\ f_2 = 1.04081078,\ f_3 = 1.09417428, f_4 = 1.17351087,$$
$$f_5 = 1.28402542,\ f_6 = 1.43332941.$$

$$\int_{x_0}^{x_6} f(x)\,dx = \int_0^{0.6} e^{x^2}\,dx = \frac{6}{20}h[11f_1 - 14f_2 + 26f_3 - 14f_4 + 11f_5]$$
$$= \frac{6}{20}(0.1)[11(1.01005017) - 14(1.04081078) + 26(1.09417428) -$$
$$14(1.17351087) + 11(1.28402542)] = 0.68048579.$$

Chapter 8

Numerical Solutions for Ordinary Differential Equations

8.1 Introduction

In the real world around us there are a plenty of phenomena and problems, their models are ordinary differential equations (ODEs). As an classic example is Newton's law of motion and Loktta-Volterra equation. The ODE (or ode) is called **an initial value problem** if all the conditions on the problem are given at some starting value of the independent variable, and this condition is called the **the initial condition**. On the other hand, the ODE is termed **boundary value problem** when the conditions of the problem are specified at the boundary of the problem and as special case the problem is said to be two-point boundary value problem if these conditions are given at two points on the boundary of some region of interest, these conditions are called **boundary conditions**.

8.2 Taylor Series Method

Taylor series is used to write most functions as power series. Let us consider the Taylor expansion of the function y:

$$\begin{aligned} y(t+h) = y(t) + hy^{'}(t) + \frac{1}{2!}h^2y^{''}(t) + \frac{1}{3!}h^3y^{'''}(t) \\ +\frac{1}{4!}h^4y^4(t) + \cdots + \frac{1}{m!}h^my^m(t) + \cdots . \end{aligned} \tag{8.1}$$

Remember that this is infinite series, and for practical applications the series is truncated after $m+1$ terms. If h is small and the derivatives

$y^{'}(t), y^{''}(t), y^{'''}(t), y^4(t), \cdots y^m(t)$ are known, then we can compute $y(t+h)$ in somehow accurate way. If we truncate the series after the term $\frac{1}{m!}h^m y^m(t)$ then the method is called the **Taylor series method of order m**. The simplest method when $m = 1$ is called **Euler method**.

8.3 Euler Method

Euler method is the simplest method for solving the ordinary differential equations, It serves as a simple model for theoretical studies and investigations and rarely used in practice. It is used as the basis for studying more complicated and advanced methods and techniques. Euler method is used for solving the well-posed initial value problem

$$\frac{dy}{dt} = f(t, y), \ a \leq t \leq b, \ y(a) = y_0. \tag{8.2}$$

To derive Euler approximation of this model, we use only the linear terms in Taylor series:

$$y(t) = y(t_0) + (t - t_0)y^{'}(t_0) + \frac{(t - t_0)^2}{2}y^{''}(\xi),$$

where $\xi \in [a, b]$, and define $h = t - t_0$ as the **increment** in t, and also called the **step size**, we get

$$y(t) = y(t_0) + hy^{'}(t_0) + O(h^2),$$

if h is small enough, then the error $= O(h^2) = \frac{h^2}{2}y^{''}(\xi)$ is smaller and can neglected to obtain

$$y(t) = y(t_0) + hy^{'}(t_0).$$

Once we computed y at $t_0 + h$ then, we can compute its value at the next point i.e. $y(t_0 + 2h$ and so on. Hence, the general formula is:

$$y_{n+1} = y_n + hy^{'}_n. \tag{8.3}$$

This is the **Euler method**, or also called **Euler-Cauchy method** or **point-slope method**.

We approximate the continuous solution of this problem by a discrete approximation at some points in the domain interval $[a, b]$, these points are termed **mesh points**, **grid points** or **collocation points**. Once the discrete solution is computed at these mesh points, the approximate solution at the other points can be obtained by interpolation or extrapolation.

8.4 Runge-Kutta Methods

The simple Euler method is derived by using one term of Taylor series expansion about $t = t_0$. The modified Euler method is derived by using two terms of Taylor series. Two German mathematicians Runge and Kutta derived new methods by using more terms than the first two terms in the Taylor series. These methods are called **Runge-Kutta methods** after them.

Second Order Runge Kutta Method

We start with simplest Runge-Kutta methods which is the **second order Runge-Kutta method**.

The general form of the second order Runge-Kutta method is:

$$y(t+h) = y(t) + w_1 hf(t,y) + w_2 hf(t+\alpha h, y+\beta hf(t,y)), \tag{8.4}$$

or, equivalently,

$$y(t+h) = y(t) + w_1 K_1 + w_2 K_2, \tag{8.5}$$

where

$$\begin{cases} K_1 = hf(t,y), \\ K_2 = hf(t+\alpha h, y+\beta K_1). \end{cases}$$

The objective is to find the values of the unknowns w_1, w_2, α and β such that the Eq. (8.4) is as accurate as possible. Using Taylor series expansion of terms up to the third order, we have

$$w_1 = \frac{1}{2},\ w_2 = \frac{1}{2},\ \alpha = 1,\ \beta = 1.$$

This results in

$$y(t+h) = y(t) + \frac{h}{2}f(t,y) + \frac{h}{2}f(t+h, y+hf(t,y)), \tag{8.6}$$

or,

$$y(t+h) = y(t) + \frac{1}{2}(K_1 + K_2), \tag{8.7}$$

where

$$\begin{cases} K_1 = hf(t,y), \\ K_2 = hf(t+h, y+K_1). \end{cases}$$

Third Order Runge Kutta Method

The formula of the Runge-Kutta method of order three is:

$$y(t+h) = y(t) + \frac{1}{6}(K_1 + 4K_2 + K_3), \tag{8.8}$$

where

$$\begin{cases} K_1 = hf(t, y), \\ K_2 = hf(t + \frac{1}{2}h, y + \frac{1}{2}K_1), \\ K_3 = hf(t + h, y - hK_1 + 2hK_2), \end{cases}$$

Fourth Order Runge Kutta Method

The classical formula of the Runge-Kutta method of order four is:

$$y(t+h) = y(t) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4), \tag{8.9}$$

where

$$\begin{cases} K_1 = hf(t, y), \\ K_2 = hf(t + \frac{1}{2}h, y + \frac{1}{2}K_1), \\ K_3 = hf(t + \frac{1}{2}h, y + \frac{1}{2}K_2), \\ K_4 = hf(t + h, y + K_3), \end{cases}$$

Fifth Order Runge Kutta Method

This method was derived by Butcher in 1964. The formula of the Runge-Kutta method of order five is:

$$y(t+h) = y(t) + \frac{h}{90}(7K_1 + 32K_2 + 2K_3 + 12K_4 + 32K_5 + 7K_6), \tag{8.10}$$

where

$$\begin{cases} K_1 = hf(t, y), \\ K_2 = hf(t + \frac{1}{4}h, y + \frac{1}{4}hK_1), \\ K_3 = hf(t + \frac{1}{4}h, y + \frac{1}{8}hK_1 + \frac{1}{8}hK_2), \\ K_4 = hf(t + \frac{1}{2}h, y - \frac{1}{2}hK_2 + hK_3), \\ K_5 = hf(t + \frac{3}{4}h, y + \frac{3}{16}hK_1 + \frac{9}{16}hK_4), \\ K_6 = hf(t + h, y - \frac{3}{7}hK_1 + \frac{2}{7}hK_2 + \frac{12}{7}hK_3 - \frac{12}{7}hK_4 + \frac{8}{7}hK_5), \end{cases}$$

8.5 Midpoint Method

The midpoint method is a special case of Runge-Kutta method of order 2, if we assume that $w_2 = 1$ implies $w_1 = 0$ and $\alpha = \beta = \frac{1}{2}$, then Eq. (8.4) becomes

$$y(t+h) = y(t) + hf(t + \frac{1}{2}h, y + \frac{1}{2}hf(t,y)), \tag{8.11}$$

or, equivalently,

$$y(t+h) = y(t) + hK_2, \tag{8.12}$$

where

$$\begin{cases} K_1 = hf(t,y), \\ K_2 = hf(t + \frac{1}{2}h, y + \frac{1}{2}hK_1). \end{cases}$$

Bibliography

[1] A. Householder, *Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.

[2] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis*, 2nd Edition, McGraw-Hill, New York, 1978.

[3] C. Gerald and P. Wheatley, *Applied Numerical Analysis*, 2nd Edition, Addison-Wesley, Reading, Mass, MA, 1987.

[4] C. Gerald and P. Wheatley, *Applied Numerical Analysis*, 7th International Edition, Pearson/Addison-Wesley, Boston, 2004.

[5] D. Kincaid and W. Cheney, *Numerical Mathematics and Computing*, 4th Edition, Brooks/Cole Publishing Company, 1999.

[6] D. Kincaid and W. Cheney, *Numerical Mathematics and Computing*, 5th Edition, Brooks/Cole Publishing Company, 2004.

[7] D. Kincaid and W. Cheney, *Numerical Mathematics and Computing*, 7th International Edition, Brooks/Cole Publishing Company, 2013.

[8] F. Hildebrand, *Introduction to Numerical Analysis*, 2nd Edition, McGraw-Hill, New York, 1974.

[9] G. Phillips and P. Taylor, *Theory and Applications of Numerical Analysis*, Academic Press, New York, 1973.

[10] J. Mathews and K. Fink, *Numerical Methods Using MATLAB*, Pearson Education, Inc., 4^{th} Edition, 2004.

[11] K. Atkinson, *Elementary Numerical Analysis*, John Wiley & Sons, 1985.

[12] L. Hageman and D. Young, *Applied Iterative Methods*, Academic Press, New York, 1981.

[13] M. Heath, *Scientific Computing: An Introductory Survey*, 2^{nd} Edition, McGraw-Hill, New York, 2002.

[14] P. Linear, *Theoretical Numerical Analysis*, John Wiley & Sons, New York, 1979.

[15] R. Burden and J. Faires, *Numerical Analysis*, 3^{rd} Edition, Mass Prindle, Weber & Schmidt, 1985.

[16] R. Burden and J. Faires, *Numerical Analysis*, 9^{th} Edition, Brooks/Cole, 2011.

[17] S. Conte and C. deBoor, *Elementary Numerical Analysis*, 3^{rd} Edition, McGraw-Hill, New York, 1980.

[18] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3^{rd} Edition, Cambridge University Press, 2007.