Chapter 1: Introduction

1 Numerical Analysis

Definition

Numerical analysis is **a field of study** focused on developing and analyzing methods for solving mathematical problems that are difficult or impossible to solve exactly using traditional algebraic techniques. It involves studying the **accuracy, stability, and efficiency** of numerical methods.

2 Numerical Algorithm (Numerical Method)

Numerical Algorithm

A Numerical Algorithms are practical techniques used to perform **numerical computations**. providing an approximate solution and is used to solve problems involving large amounts of data. The algorithm is based on a well-defined iterative sequence, starting with an initial solution that progressively converges toward the desired result with each iteration.

$$\begin{cases} (U_n) & n \geq 0 \\ S_0 & \text{Initial Solution (Starting Point)} \end{cases}$$

3 Convergence Speed (Order of Convergence)

Convergence Speed

The number of iterations required to find the solution we are looking for:

- Linear Order: 1 (slow)
- Quadratic Order: 2 (faster)
- >> 2 (very fast)

4 Interpolation

Interpolation

Estimates the value between two known points of a function allowing for a smoother representation of the function's behavior.

5 Approximation

Approximation

Approximates the formula of a function from a set of values, with the objective of finding a simpler function that represents the general trend of the data, even if it doesn't pass through every point exactly.

6 Error

Error

An error represents the difference between the actual solution and the computed result. It indicates how far we are from the true solution. There are two cases:

- **Evaluation**: We know the exact solution, so we can directly calculate the error:

$$\boxed{E_r = |\overline{x} - x_{\text{app}}|}$$

- **Estimation**: We don't know the exact solution, so we only have an estimate of the error, based on the output of the algorithm:

$$\boxed{E_r = |\overline{x} - x_{\text{app}}| \leq \text{Algo}}$$

Where:

- E_r : The error value.
- $\overline{x}$: The exact solution.
- x_{app} : The approximate solution.
- Algo : The error value found by the algorithm.

7 Optimization

Optimization

Optimization in numerical algorithms refers to two things:

- **Error**: We aim to minimize the error in order to achieve the most accurate approximate solution.
- **Convergence Speed**: The higher the order of convergence, the less time the algorithm will take to converge to the solution we are looking for.

Chapter 2: Approached Resolution of Non-Linear Equation $f(x) = 0$

1 $f(x) = 0$ Equation

$f(x) = 0$

Let $\alpha \in \mathbb{R}$ such that $f(\alpha) = 0$.
Graphically, $f(x) = 0$ means that the function f intersects the x-axis.
In numerical methods, we approximate α using a sequence (x_i) such that:

$$\alpha \approx x_i, \quad \text{for some } i$$

Note

We use numerical methods only if the equation's degree is greater than 2 or if it is non-trivial, such as $3x - e^{-x} = 0$. For linear or quadratic equations, exact methods are sufficient, and numerical methods are not needed.

2 Sequence

Sequence

Each numerical algorithm (method) has their specific sequence that must convegre to the deseried solution

3 Initial Steps Of Any Numerical Algorithm

Initial Steps

Every numerical algorithm must first go through two crucial steps:

1. Determine the number of solutions.
2. Define the range of each solution within a closed, bounded, and continuous interval $[a, b]$.

Note

- **Terminal Phase:** In numerical analysis, we aim for real numerical values rather than exact expressions in fractional or functional form.
- **Differences Between Algorithms:** The main difference between numerical algorithms lies in how they approximate and compute solution values for each interval identified in the initial step.
- **Exercises That Do Not Mention the Initial Steps:** Even if an exercise does not explicitly state the need for the initial steps, we must always perform them, as they are crucial regardless of the algorithm used.

3.1 Finding the Number of Solutions

Number of Solutions

To determine the number of solutions, we first analyze the **monotonicity** of the function and construct its **variation table**. Then, we apply the **Intermediate Value Theorem (IVT)** corollary to determine the number of roots:

If $f : I \to \mathbb{R}$ is **continuous** on the **closed and bounded interval** $[a, b]$:

- $f(a) \cdot f(b) < 0$:
  - **No monotonicity** $\to$ At least one root exists.
  - **Strictly monotonic** $\to$ Exactly one root exists.
- $f(a) \cdot f(b) > 0$:
  - 0 or even number of solution

3.2 Range for Each Solution

Range

Sometimes, key points can be found by differentiating. If not, we can reduce the range by testing values, ideally until the interval length reaches 1 for better performance.

Example :

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

Finding Intervalle Of Definition

- e^{-x} is defined in $\mathbb{R}$
- $ln(x)$ is defined in $]0, +\infty[$

$$\boxed{D_f = \mathbb{R} \cap \ ]0, +\infty[\ = \ ]0, +\infty[}$$

Differentiate

Since f is the sum of two functions that are continuous and differentiable on $D_f \Rightarrow f$ is also continious and differentiable on D_f.

$$f'(x) = -e^{-x} - \frac{1}{x} = -(e^{-x} + \frac{1}{x})$$

Since $e^{-x} > 0$ and $\frac{1}{x} > 0$ for all $x \in D_f =]0, +\infty[\Rightarrow e^{-x} + \frac{1}{x} > 0$.
Thus, $f'(x) < 0$ for all $x \in D_f$, meaning f' is strictly negative and f is strictly decreasing on D_f (Montonic).

Variation Table

x	0	1	2	$+\infty$
$f'(x)$		$-$		
$f(x)$	$+\infty$	$\searrow$		$-\infty$

$f(1) \approx 0.36 > 0$, $f(2) \approx -0.55 < 0$

IVT

f is continuous and strictly montonic in $I = [1, 2]$, and $f(1) \cdot f(2) < 0 \Rightarrow$ Exactly one root

4 Algorithm (Building The Sequence)

4.1 Dichotomy (Bisection)

Condition

This method is based on the **Intermediate Value Theorem (IVT)** and therefore requires the function f to be continuous. Additionally, we need a closed and bounded interval $[a, b]$ that contains a **single** root $\alpha \in [a, b]$.

4.1.1 Order Of Convergence

Order

The bisection method has a **linear order of convergence** : 1 , meaning the error decreases at a constant rate in each iteration.

4.1.2 Guaranteed Convergence

Guarantee

Even though the bisection method is slow, it always guarantees to converge to the desired solution.

4.1.3 Sequence

Sequence

For each iteration, we divide the interval $[a_n, b_n]$ into two equal sub-intervals, where $[a_0, b_0] = [a, b]$ and the midpoint is given by:

$$\boxed{x_n = \frac{a_n + b_n}{2} \ , \ \forall n \geq 0}$$

For each iteration, we determine the correct sub-interval for the next step:

- If $f(a_n) \cdot f(x_n) < 0$, then $\alpha \in [a_n, x_n]$, so we set:

$$[a_{n+1}, b_{n+1}] = [a_n, x_n]$$

- If $f(b_n) \cdot f(x_n) < 0$, then $\alpha \in [x_n, b_n]$, so we set:

$$[a_{n+1}, b_{n+1}] = [x_n, b_n]$$

4.1.4 Error Estimation

Error Estimation

$$\boxed{E_n = |\alpha - x_n| \leq \frac{b-a}{2^{n+1}} \;\;, \;\; \forall n \geq 0}$$

4.1.5 Tolerance

Tolerance

Tolerance is a fixed value set by the user to ensure that the error does not exceed a predefined bound, denoted by ϵ.

$$\boxed{E_n = |\alpha - x_n| \leq \frac{b-a}{2^{n+1}} \leq \epsilon \;\;, \;\; \forall n \geq 0}$$

4.1.6 Number Of Iterations

Number Of Iterations

$$\epsilon \geq \frac{b-a}{2^{n+1}}$$

$$\frac{2^{n+1}}{b-a} \geq \frac{1}{\epsilon}$$

$$2^{n+1} \geq \frac{b-a}{\epsilon}$$

$$\ln(2^{n+1}) \geq \ln\left(\frac{b-a}{\epsilon}\right)$$

$$(n+1)\ln(2) \geq \ln\left(\frac{b-a}{\epsilon}\right)$$

$$n \geq \frac{\ln\left(\frac{b-a}{\epsilon}\right)}{\ln(2)} - 1$$

$$\boxed{n = \left\lceil \frac{\ln\left(\frac{b-a}{\epsilon}\right)}{\ln(2)} - 1 \right\rceil}$$

4.1.7 Solution Intervalle

Solution Intervalle

$$|\alpha - x_n| \leq \epsilon$$

$$-\epsilon \leq \alpha - x_n \leq \epsilon$$

$$\boxed{x_n - \epsilon \leq \alpha \leq \epsilon + x_n}$$

Note

- The number of iterations is affected by the length of the interval $[a, b]$. By convention, it is better to choose an interval of length 1 for better performance.
- Length of an interval $[a, b]$: $b - a$.
- Midpoint (bisection) of an interval $[a, b]$: $\dfrac{a + b}{2}$.
- Some exercises may not provide the problem directly, requiring us to model it.

Example

approximate the value of $\sqrt[3]{80}$ with the tolerance $\epsilon = 10^{-1}$

Modelisation Of The Problem

$$x = \sqrt[3]{80} \Rightarrow x^3 = 80 \Rightarrow \boxed{f(x) = x^3 - 80}$$

Finding Intervalle Of Definition

$$\text{Since } f \text{ is a polynomial function} \Rightarrow \boxed{D_f = \mathbb{R}}$$

Differentiate

Since f is a polynomial function, it's continuous and differentiable on D_f.

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

Since $f'(x) \geq 0$ on $D_f \Rightarrow f$ is increasing on D_f (monotonic)

Variation Table

x	$-\infty$		0		4		5		$+\infty$
$f'(x)$		$+$	0			$+$			
$f(x)$	$-\infty$			$\nearrow$					$+\infty$

$f(0) = -80$, which means that $\alpha \in [0, +\infty[$. We are going to shorten the interval by testing the points 4 and 5: $f(4) = -16$ and $f(5) = 45$.

Intermediate Value Theorem (IVT)

Since f is continuous and strictly monotonic in $I = [4, 5]$, and $f(4) \cdot f(5) < 0$, $\Rightarrow$ There exists exactly one root in the interval $[4, 5]$.

Number Of Iteration

$$n = \left\lceil \frac{\ln\left(\frac{b-a}{\epsilon}\right)}{\ln(2)} - 1 \right\rceil = \left\lceil \frac{\ln\left(\frac{5-4}{10^{-1}}\right)}{\ln(2)} - 1 \right\rceil = \boxed{\lceil 2.3 \rceil = 3}$$

Bisection

Iterration 1

$$x_0 = \frac{b_0 - a_0}{2} = \frac{1}{2}$$

$$f(x_0) = -79.875$$

$$f(x_0).f(b_0) < 0 \Rightarrow [a_1, b_1] = [x_0, b_0]$$

Iterration 2

$$x_1 = \frac{b_0 - x_0}{2} = \frac{9}{4}$$

$$f(x_1) \approx -68.6$$

$$f(x_1).f(b_0) < 0 \Rightarrow [a_2, b_2] = [x_1, b_0]$$

Iterration 3

$$x_2 = \frac{b_0 - x_1}{2} = \frac{11}{8}$$

Solution Interval

$$\boxed{x_3 - \epsilon \leq \alpha \leq \epsilon + x_3}$$