|  |  |  |
| --- | --- | --- |
|  | **BAHRIA UNIVERSITY, (Karachi Campus)**  *Department of Software Engineering*  **Assignment 1 - Fall 2024** |  |



COURSE TITLE: **NUMERICAL ANALYSIS** COURSE CODE: **GSC-321**

Class: **BSE-VII (A,B)** Time Allowed:  **1 Week.**

Course Instructor: **Engr. Rahemeen** Max. Marks: **5 marks**

Submission Date: **18-10-2024**

**Name: Muhammad Shoaib Akhter Qadri Enrollment No: 02-131212-009**

**Question No. 1 [CLO1: 5 Marks]**

1. **Define numerical stability in the context of numerical algorithms. How can you determine whether a numerical algorithm is stable?**

**Numerical Stability in the Context of Numerical Algorithms:**

**Numerical stability** refers to the behavior of numerical algorithms when they are subject to small perturbations, such as rounding errors, truncation errors, or slight variations in input data. A numerically stable algorithm produces results that are not disproportionately affected by these small errors. In other words, the errors introduced during computations do not grow uncontrollably as the algorithm proceeds, ensuring that the final output remains close to the true value despite minor inaccuracies during calculations.

**Determining Whether an Algorithm is Numerically Stable:**

1. **Forward Error Analysis**: This involves analyzing the difference between the computed solution and the true solution as a function of the input errors. If the output error is comparable in size to the input error (scaled by a reasonable factor), the algorithm is considered stable.
2. **Backward Error Analysis**: Instead of comparing the computed solution with the exact one, backward error analysis examines how much the input data must be perturbed for the computed solution to be the exact solution of the perturbed problem. If the perturbations are small, the algorithm is considered backward stable.
3. **Condition Number**: For problems involving matrices or functions, the condition number measures how sensitive the output is to small changes in the input. If an algorithm maintains the error within a bound relative to the condition number, it is considered stable.
4. **Growth of Error During Iterations**: In iterative algorithms, monitoring the error growth with each iteration can indicate stability. If errors grow exponentially, the algorithm may be unstable. However, if errors either remain bounded or decrease, the algorithm is stable.

Stable algorithms prevent error amplification and ensure reliable results, even when computations are carried out with limited precision

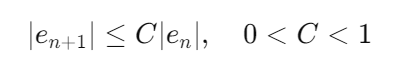
1. **Discuss the difference between linear convergence and quadratic convergence in the context of iterative numerical methods.**

**Difference Between Linear Convergence and Quadratic Convergence in Iterative Numerical Methods:**

In iterative numerical methods, **convergence** refers to the rate at which a sequence of approximations approaches the true solution. The convergence rate impacts the efficiency and speed of the algorithm. The two most common types of convergence are **linear convergence** and **quadratic convergence**.

1. **Linear Convergence:**

* **Definition**: Linear convergence occurs when the error in successive iterations decreases by a constant factor at each step. Mathematically, if ene\_nen​ is the error at the nnn-th iteration, the error is said to converge linearly if:

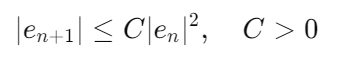
****

Here, CCC is a constant that defines how quickly the error decreases with each iteration.

* **Characteristics**:
  + Convergence is relatively slow.
  + Each iteration reduces the error by a fixed proportion.
  + Example: The **Bisection Method** for root finding converges linearly, halving the interval containing the root at each step.
* **Implication**: Linear convergence is often sufficient when high precision is not immediately necessary or when other factors like computational complexity dominate.

1. **Quadratic Convergence:**

* **Definition**: Quadratic convergence occurs when the error decreases at a much faster rate, specifically, the square of the error at the nnn-th iteration gives an upper bound for the error at the (n+1)(n+1)(n+1)-th iteration. Mathematically, quadratic convergence is expressed as:



* **Characteristics**:
  + Convergence is much faster compared to linear convergence.
  + As the error gets smaller, the improvement between successive iterations accelerates significantly.
  + Example: **Newton’s Method** for finding roots of a function exhibits quadratic convergence when close to the actual root.
* **Implication**: Quadratic convergence is highly desirable in numerical methods because it dramatically reduces the number of iterations required to achieve a given accuracy. However, it usually requires better initial approximations to guarantee this fast convergence.

**Key Differences:**

1. **Rate of Error Reduction**:
   * In linear convergence, the error is reduced by a constant factor with each iteration.
   * In quadratic convergence, the error is reduced by the square of the previous error, leading to a much faster reduction in error.
2. **Speed**:
   * Linear convergence is slower and often requires more iterations to achieve the desired accuracy.
   * Quadratic convergence is much faster and can achieve high precision in just a few iterations, especially when the initial guess is close to the true solution.
3. **Examples**:
   * **Linear Convergence**: The Bisection Method, Fixed Point Iteration.
   * **Quadratic Convergence**: Newton’s Method, Secant Method (in some cases).