Ilmu Komputer
Universitas Pendidikan Ganesha

Numerical Methods

Error Analysis

Dr. Komang Setemen, S.Si., M.T.

Outline

- Error Analysis
- Sources of Error in Numerical Computations
- Absolute and Relative Errors
- Roundoff and Truncation Errors
- Numbers Representation in Computer
- Floating-Point Numbers

Error Analysis

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

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.

Error Analysis

Definition (Numerical Error)

Let $\boldsymbol{x}$ *be the exact solution of the underlying problem and* $\boldsymbol{x}^*$ *its approximate solution, then the error (denoted by* $\boldsymbol{e}$*) in solving this problem is*

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}^*$$

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 minimized 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**

Sources of Error in Numerical Computations

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

Absolute and Relative Errors

- **Absolute Error**. The absolute error $\hat{\boldsymbol{e}}$ of the error **e** is defined as the absolute value of the error **e**

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

- **Relative Error**. The relative error $\tilde{\boldsymbol{e}}$ of the error e is defined as the ratio between the absolute error $\hat{\boldsymbol{e}}$ and the absolute value of the exact solution **x**

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

Absolute and Relative Errors

- **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.

a. e = x − x* = 3.141592653589793 − 3.14159265 = $3.589792907376932 \times 10^{-9}$

= $3.589792907376932 \times 10^{-9}$ = 0.000000003589792907376932

b. Absolute error:

$\hat{e} = |x - x^*|$ = |3.141592653589793−3.14159265| = $3.589792907376932 \times 10^{-9}$

Absolute and Relative Errors

c. Relative Error:

$$\tilde{e} = \frac{\hat{e}}{|x|} = \frac{|x - x^*|}{|x|} = \frac{3.141592653589793 - 3.14159265}{3.141592653589793}$$

$$= \frac{3.589792907376932\text{x}10^{-9}}{3.141592653589793} = 1.142666571770530\text{x}10^{-9}$$

Roundoff and Truncation 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 and the process is called correct rounding.
- Truncation errors also sometimes called chopping errors are occurred when chopping an infinite number and replaced it by a finite number or by truncated a series after finite number of terms.

Roundoff and Truncation Errors

- Example 3. Approximate the following decimal numbers to three digits by using rounding and chopping (truncation) rules:

1. x1 = 1.34579. Rounding=1.35, Chopping=1.34
2. x2 = 1.34679. Rounding=1.35, Chopping=1.34
3. x3 = 1.34479. Rounding=1.34, Chopping=1.34
4. x4 = 3.34379. Rounding=1.34, Chopping=1.34
5. x5 = 2.34579. Rounding=1.35, Chopping=1.34

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 transform them to the binary system by using the machine language.

Numbers Representation in Computer

- **Scientific Notation**. Let k be a real number, then k can be written in the following form

  $$\mathbf{k = m \times 10^n}$$

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**.

Numbers Representation in Computer

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

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}

Floating-Point Numbers

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

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

for each i = 2, · · · , 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

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

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

Floating-Point Numbers

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

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

the number r is called the **normalized mantissa**

Thank you