MODULE -1

TOPIC: ERRORS AND FLOATING POINT ARITHMETIC

Introduction to Numerical Methods

The major reasons for evolvement of numerical methods are as follows:

1. **Limitations of analytical methods**: If we consider a non linear transcendental equation like $x\log x = 1.2$ or any other algebraic equation of higher degree like $x^5 - 5x^4 + 3x^2 + 2x - 1 = 0,$ it is not very much easy to solve these equations by any of the methods discussed so far like graphical methods, factorization methods, etc. (the methods known to us so far for solving nonlinear equations, differential equations, etc., are generally referred to as ***analytical methods (also called closed form)),***analytical methods are limited in nature, there are many problems in science and engineering whose analytical solution are not possible or very difficult to obtain.
2. **Solutions to tabulated data:** If we are given the distance travelled by the car at regular intervals of time in the form of table as follows

t:	0	10	20	30	40	50
S:	0	20	35	55	78	100

   If the distance travelled in 45 minutes or velocity of the car at t = 40 minutes is to be obtained. Since given values are at discrete set of points, the function s = f(t) is not known, so then we have to turn towards numerical methods for such type of problems.

Therefore, the aim of numerical methods is to provide constructive methods for obtaining answers to such problems in numerical form. Even if there is possibility of obtaining analytical solutions to the problem even then numerical methods are used since numerical methods solves the problems with the help of

basic arithmetic operations, that's why numerical methods are easier and has become indispensable tools for today's scientists and engineers. So, in short, the following definition can be stated

Numerical Methods are the tools to solve complicated mathematical, scientific, & engineering problems by means of basic arithmetic operations.

NOTE: The results obtained through numerical methods are purely numerical in nature, whereas analytical methods gives a solution in terms of mathematical function that can be evaluated for specific instances. The behaviour & properties of the function are clearly visible by analytical solution, but this is not the case with numerical solution, so in this aspect analytical soln. has an advantage over numerical soln.

Significant Digits: Any digit is a significant digit except when it is used to fix the decimal point or to fill the places of unknown or discarded digits.

Rules for finding Significant digits:

i. All non zero digits are significant.
ii. All zeroes occurring between non-zero digits are significant.
iii. Trailing zeroes following a decimal point are significant. e.g. 3.50,65.0 and .230 have three significant digits each.
iv. Zeroes b/w the decimal point and preceding a non-zero digit are not significant
    e.g. 0001234 has 4 significant digits
v. When the decimal point is not written, trailing zeroes may or may not be considered to be significant, that is, 4500 may have 2 significant, 3 significant, or 4 significant digits. Since, the trailing zeros may be used to fill the places of unknowns. But, if the source of the trailing zeros is not confirmed, then the trailing zeros not considered to be significant, that is,

4500 has 2 significant digits, however, 4500.0 has 5 significant digits (that is, trailing zeros written after decimal point are significant).

vi. When any number is written in the form of scientific notation as M X 10^e,then the number of significant digits is the number of digits explicitly in M. 4.5 X 10^3 has 2 significant digits, 4.50 X 10^3 has 3 significant digits, 4.500 X 10^3 has 4 significant digits.

vii. If all the rules are not applicable for any number, then it would be better to decide through the original definition of significant digits.

Accuracy: Accuracy refers to the no. of significant digits in any number e.g. 57.897 is accurate to 5 significant digits

Precision: Precision refers to the no. of decimal positions i.e. the order of magnitude of the last digit in a no, e.g. 57.897 has a precision of .001 or 10^{-3}

FLOATING POINT ARITHMETIC

The first step in computation with digital computers is to convert the decimal numbers to another number system (say), β (called radix or base, $\beta = 10$ for decimal and $\beta = 2$ for binary systems), understandable to that particular computer and then to store these converted numbers in computer memory. The memory of the digital computer is divided into separate cells called **words**. Each word can hold the same no. of digits called bits, with respect to its base plus a sign. Negative numbers are stored as absolute value plus a sign or in complement form. The no. of digits which can be stored in a computer word is called its **word length**. The word length varies from one computer to another. The numbers in the computer word can be stored in two forms: Fixed point form and Floating point form. For convenience, to store these forms, the following assumptions can be made

a) Assume a hypothetical computer in which each location(word) can store 6 digits (bits) and having provision to store one or more signs.

b) Here, the whole process is for numbers in decimal system, assuming that the same will be true for corresponding binary no's inside the computer's memory.

I. Fixed Point Form: In this form, a $t-$digit is assumed to have its fixed decimal point at the left hand end of the word. It means that all the numbers are assumed to be less than 1 in magnitude. The fixed point number with base β and $t-$digits word length may be written as

$$\pm(\alpha_1\beta^{-1}+\alpha_2\beta^{-2}+.....+\alpha_t\beta^{-t})=\pm\sum_{k=1}^{t}\alpha_k\beta^{-k}$$

where $0\le\alpha_k\le\beta$. The disadvantage of fixed point form to keep every number less than one in magnitude during computation.

II. Floating Point Form: Most computers use floating point representation, which is recognized by four parameters, the base β, the number of digits t and the exponent range (m,M). A floating point number is a number represented in the form:

$$.d_1d_2.....d_t\times\beta^e$$

where $d_1,d_2,....,d_2$ are integers and satisfy $0\le d_i\le\beta$ and the exponent is such that $m\le e\le M$. The fractional part $.d_1d_2.....d_t$ is called the Mantissa and it lies between +1 and -1. So, it has two parts: Mantissa and Exponent (similar to Scientific form). The zero floating number is $.00.....0\times\beta^e$.

Normalized Floating point form: If in a floating point form of any no. the decimal is on the left of the first non zero digit(most significant digit), then such type of floating point no. are called normalized floating point no.'s and process of shifting of decimal point to the left of first non zero digit is called normalization e.g. the number 0.006831 in normalized floating point form in 4 digit mantissa is stored as 0.6831E-02 or 0.6831 x 10^{-2}

Note: For normalized floating point number, the mantissa should satisfy the following conditions (in decimal system)

Positive no's. $0.1 \leq \text{mantissa} < 1.0$

Negative no's.: $-1.0 < \text{mantissa} \leq -0.1$

In general, $0.1 \leq |\text{mantissa}| < 1.0$. For any base β, the value of mantissa lies in the interval $\left[\frac{1}{\beta},1\right)$ for positive numbers and $\left(-1,-\frac{1}{\beta}\right]$ for negative numbers.

$t-$digit Mantissa Standard Form: A non zero floating point number is **in** $t-$ digit mantissa standard form if it is normalized and its mantissa consists of exactly $t-$digits. If a number x has the representation in the form

$$x = .d_1 d_2 .....d_t d_{t+1} .....\times \beta^e \qquad (1.1)$$

then the floating point number $x^* = d_1 d_2 .....d_t \times \beta^e$ can be obtained in the following two ways:

a) ***Chopping***: Here we neglect $d_{t+1}, d_{t+2}\ldots\ldots$ in equation (1.1) above and obtain

$$x^* = .d_1 d_2 .....d_t \times \beta^e$$

b) ***Rounding***: Here the mantissa part in equation (1.1) above is written as

$$.d_1 d_2 .....d_t \{d_{t+1} + \frac{1}{2}\beta\}$$

and first t digits are taken to write the floating point no.

Example: 0.1686E03 = .168E03 for chopping

= .168{6+(1/2)10}E03=.169E03 for rounding

Here, symmetric rounding can also be used, that is,

i. If d_{t+1} is greater than 5, then increase d_t by one.

ii. If d_{t+1} is lesser than 5, then leave d_t unchanged.

iii. If d_{t+1} is equal to 5, then leave d_t unchanged, if it is even and increase by one if it is odd.

ERROR

Error is the difference between the actual(true) value and the approximate value obtained from experimental observation or from numerical computation.

$$\text{Error} = \text{True value} - \text{Approximate value} = X_t - X_a$$

where X_t is the true value and X_a is the approximate value. There are three major techniques to estimate errors

1. **Absolute error**: Since error analysis has to do with magnitude and not with the sign so we have this absolute error

$$\text{Absolute error}(E_a) = |x_t - x_a|$$

2. **Relative error:** In many cases, absolute error may not take reflect its influence correctly as it does not take into account the order of magnitude of the value under study, e.g. an error of 1 gram is much more significant in the weight of 10 gm. gold chain than in the bag of 1000 kg. rice. To overcome this, the relative error is used which is nothing but " Normalised Absolute error"

$$\text{Relative error } (E_r) = \frac{|x_t - x_a|}{|x_t|} = \frac{\text{Absolute Error}}{|\text{True Value}|}$$

3. **Percentage error:** $\text{Percentage error}(E_p) = E_r \times 100$

NOTE:

i. If x is a real number in base 10 and x^* is its machine representation (fixed point or floating form up to $t-$ digits, then

$$\text{Absolute error} = |x - x^*| \leq \begin{cases} 10^{-t} & \text{(chopping)} \\ \frac{1}{2} \times 10^{-t} & \text{(rounding)} \end{cases}$$

$$\text{Relative error} = \frac{|x - x^*|}{|x|} \leq \begin{cases} 10^{1-t} & \text{(chopping)} \\ \frac{1}{2} \times 10^{1-t} & \text{(rounding)} \end{cases}$$

For any base or radix β,

$$\text{Absolute error} = |x - x^*| \leq \begin{cases} \beta^{-t} & \text{(chopping)} \\ \frac{1}{2} \times \beta^{-t} & \text{(rounding)} \end{cases}$$

$$\text{Relative error} = \frac{|x - x^*|}{|x|} \leq \begin{cases} \beta^{1-t} & \text{(chopping)} \\ \frac{1}{2} \times \beta^{1-t} & \text{(rounding)} \end{cases}$$

ii. The above result can be extended to the case of numbers also, that is, if a number is correct to $t-$decimal places, then

$$\text{Absolute error} = |x_t - x_a| \leq \begin{cases} 10^{-t} & \text{(chopping)} \\ \frac{1}{2} \times 10^{-t} & \text{(rounding)} \end{cases}$$

$$\text{Relative error} = \frac{|x_t - x_a|}{|x_t|} \leq \begin{cases} 10^{1-t} & \text{(chopping)} \\ \frac{1}{2} \times 10^{1-t} & \text{(rounding)} \end{cases}$$

SOURCES OF ERROR

The errors may be classified into the following major categories:

I. **Inherent Error:** Errors which are already present in the statement of the problem before its solution are called Inherent(or Input errors).The inherent error arises either:

   i. Due to simplified assumptions in the mathematical formulation of the problem

ii. Due to the errors in the physical measurements of the parameters of the problem. The error in physical measurements occurs since data for a problem are obtained by some experimental means and therefore of limited accuracy and precision, since every computing can measure the quantity upto certain value. Inherent errors can be minimized by taking better data or by using high precision computing aid and refined the mathematical model by making less assumptions.

II. **Round off error:** These errors arises as computers(or any computing aid) can retain only a fixed no. of significant figures during the calculation e.g. no's. such as π, e etc. can't be expressed only by a fixed no. of significant figures, hence they cannot be represented exactly by the computer(or any computing aid). In addition, because computers uses a base 2 representation and they can't precisely represent certain exact base 10 no's. e.g.$(0.7)_{10}=(.101100110\ldots\ldots\ldots)_2$**.** In both the cases there is loss of significant figures. Hence, the errors caused due to the omission of significant figures is called ROUND OFF ERROR. In other words it can be said that the round off error is the quantity which must be added to the finite representation of a computed no. in order to make it the true representation of that number**.** Round off errors are reduced by:

i. on preferring to rounding then to chop, since the error in rounding is about one half to the error in chopping.

ii. minimise the no. of arithmetic operations whenever possible.

iii. changing the calculation procedure so as to avoid subtraction of two nearly equal no's.

iv. by using double precision in floating point arithmetic ,but this usually increases the execution time.

III. **Truncation error:** Truncation errors are those that result from using an approximate formula in place of an exact mathematical formula ie. on replacing an infinite series by a finite one ,so the terms omitted introduces

error called truncation error e.g. an infinite exponential series: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$is replaced by a finite series, say, $e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$, for computation. In other words, the truncation error is the quantity which must be added to the true representation of the quantity in order that the result be equal to the quantity we are seeking to generate. Truncation errors are reduced by taking more terms in a series which usually increases the no. of arithmetic operations. To study truncation error, Taylor series with remainder term is used.

PROPOGATION OF ERROR(GENERAL ERROR FORMULA)

This formula is used to find the error in dependent variable when the amount of errors in each of the independent variable is given. In particular, if $y = f(x_1, x_2)$ be a function of two variables x_1 and x_2. If δx_1 and δx_2 be the errors in x_1 and x_2 respectively, then the error δy in y is given by:

$$y + \delta y = f(x_1 + \delta x_1, x_2 + \delta x_2) \qquad (1.2)$$

Expanding by Taylor series for two variables

$$y + \delta y = f(x_1, x_2) + \left(\frac{\partial f}{\partial x_1} \delta x_1 + \frac{\partial f}{\partial x_2} \delta x_2 \right) + \text{terms involving higher powers of } \delta x_1 \text{ and } \delta x_2$$

If the errors δx_1 and δx_2 are so small, then the squares and higher powers can be neglected, then equation (1.2) gives

$$y + \delta y = f(x_1, x_2) + \left(\frac{\partial f}{\partial x_1} \delta x_1 + \frac{\partial f}{\partial x_2} \delta x_2 \right) \qquad (1.3)$$

Now, the error δy in y is

$$\delta y = \frac{\partial f}{\partial x_1} \delta x_1 + \frac{\partial f}{\partial x_2} \delta x_2$$

Relative error in y is given by

$$E_r = \frac{\delta y}{y} = \frac{\partial f}{\partial x_1}\frac{\delta x_1}{y} + \frac{\partial f}{\partial x_2}\frac{\delta x_2}{y}$$

In general, the error δy if y is the function of n variables i.e. $y = f(x_1, x_2, ..., x_n)$ corresponding to the errors δx_i in x_i $(i = 1,2,..n)$ is given by:

$$\delta y = \left(\frac{\partial f}{\partial x_1}\delta x_1 + \frac{\partial f}{\partial x_2}\delta x_2 + \frac{\partial f}{\partial x_2}\delta x_2 + ...\frac{\partial f}{\partial x_n}\delta x_n \right)$$

$$E_r = \frac{\delta y}{y} = \left(\frac{\partial f}{\partial x_1}\frac{\delta x_1}{y} + \frac{\partial f}{\partial x_2}\frac{\delta x_2}{y} + \frac{\partial f}{\partial x_2}\frac{\delta x_3}{y} + ...\frac{\partial f}{\partial x_n}\frac{\delta x_n}{y} \right)$$

Maximum relative error is given by

$$(E_r)_{\max} = \left|\frac{\delta y}{y}\right| = \left(\left|\frac{\partial f}{\partial x_1}\frac{\delta x_1}{y}\right| + \left|\frac{\partial f}{\partial x_2}\frac{\delta x_2}{y}\right| + \left|\frac{\partial f}{\partial x_2}\frac{\delta x_3}{y}\right| + ...+ \left|.\frac{\partial f}{\partial x_n}\frac{\delta x_n}{y}\right| \right)$$

REFERENCES

1. Jain M.K, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age Publications, 2004.
2. S.C. Chapra and R. P. Canale, Numerical Methods for Engineers, McGraw Hill, 1985.
3. C.F. Gerald and P.O. Wheatley, Applied Numerical Analysis, Pearson Education, Seventh Edition, 2003.
4. S.S. Sastry, Introductory Methods of Numerical Analysis, PHI.