UNIT 1 COMPUTER ARITHMETIC

Structure **Page Nos.**

1.0 INTRODUCTION

When a calculator or digital computer is used to perform numerical calculations, an unavoidable error, called round-off error must be considered. This error arises because the arithmetic operations performed in a machine involve numbers with only a finite number of digits, with the result that many calculations are performed with approximate representation of actual numbers. The computer has the in–built capability to perform only the basic arithmetic operations of addition, subtraction, multiplication and division. While formulating algorithm all other mathematical operators are reduced to these basic operations even when solving problems involving the operations of calculus. We will discuss the way arithmetic operations are carried out in a computer and some of the peculiarities of computer arithmetics. Finally we dwell on the propagation of errors.

1.1 OBJECTIVES

After going through this unit, you should be able to:

- learn about floating-point representation of numbers;
- learn about non-associativity of arithmetic in computer;
- learn about sources of errors;
- understand the propagation of errors in subsequent calculations;
- understand the effect of loss of significant digits in computation; and
- know when an algorithm is unstable.

1.2 FLOATING POINT ARITHMETIC AND ERRORS

First of all we discuss representation of numbers in floating point format.

1.2.1 Floating Point Representation of Numbers

There are two types of numbers, which are used in calculations:

1. Integers: 1, … −3, −2, −1, 0, 1, 2, 3, …

2. Other Real Numbers, such as numbers with decimal point.

In computers, all the numbers are represented by a (fixed) finite number of digits. Thus, not all integers can be represented in a computer. Only finite number of integers, depending upon the computer system, can be represented. On the other hand, the problem for non-integer real numbers is still more serious, particularly for non-terminating fractions.

Definition 1 (Floating Point Numbers): Scientific calculations are usually carried out in floating point arithmetic in computers.

An n-digit floating-point number in base β (a given natural number), has the form

$$x = \pm\, (.d_1d_2\ldots d_n)_\beta \beta^e,\ 0 \le d_i < \beta,\ m \le e \le M;\ I = 1, 2, \ldots n,\ d_1 \ne 0;$$

where $(.d_1d_2\ldots d_n)_\beta$ is a β– fraction called mantissa and its value is given by

$(.d_1d_2\ldots d_n)_\beta = d_1 \times \frac{1}{\beta} + d_2 \times \frac{1}{\beta^2} + \ + d_n \times \frac{1}{\beta^n}$; e is an integer called the exponent.

The exponent e is also limited to range $m < e < M$, where m and M are integers varying from computer to computer. Usually, $m = -M$.

In IBM 1130, m = –128 (in binary), –39 (decimal) and M = 127 (in binary), 38 (in decimal).

For most of the computers β = 2 (binary), on some computers β = 16 (hexadecimal) and in pocket calculators β = 10 (decimal).

The precision or length n of floating-point numbers on any computer is usually determined by the word length of the computer.

Representation of real numbers in the computers:

There are two commonly used ways of approximating a given real number x into an n– digits floating point number, i.e. through rounding and chopping. If a number x has the representation in the form $x = (d_1d_2\ldots d_{n+1}\ldots)\,\beta^e$, then the floating point number fl(x) in n-digit – mantissa can be obtained in the floating two ways:

Definition 2 (Rounding): fl(x) is chosen as the n–digit floating-point number nearest to x. If the fractional part of $x = d_1d_2\ldots d_{n+1}$ requires more than n digits, then if $d_{n+1} < \frac{1}{2}\beta$, then x is represented as $(.d_1\ d_2\ldots d_n)\ \beta^e$ else, it is written as $(.d_1\ d_2\ldots d_{n-1}\ (d_n+1)\beta^e$

Example 1: $\mathrm{fl}\left(\frac{2}{3}\right) = .666667 \times 10^0$ in 6 decimal digit floating point representation.

Definition 3 (Chopping): fl(x) is chosen as the floating point number obtained by deleting all the digits except the left-most n digits. Here $d_{n+1}\ldots$ etc. are neglected and $\mathrm{fl}(x) = d_1d_2\ldots d_n\beta^e$.

Example 2: If number of digits n = 2, $\mathrm{fl}\left(\frac{2}{3}\right) = (.67) \times 10^0$ rounded

$(.66) \times 10^0$ chopped

$\mathrm{fl}(-83.7) = -(0.84) \times 10^3$ rounded

$-(0.83) \times 10^3$ chopped.

On some computers, this definition of fl(x) is modified in case $|x| \geq \beta^M$ (*overflow*) or $0 < |x| \leq \beta^m$ (*under flow*), where m and M are the bounds on the exponents. Either fl(x) is not defined in this case causing a stop or else fl(x) is represented by a special number which is not subject to the usual rules of arithmetic, when combined with ordinary floating point number.

Definition 4: Let fl(x) be floating point representation of real number x. Then $e_x = |x - fl(x)|$ is called round-off (absolute) error,

$r_x = \dfrac{x - fl(x)}{x}$ is called the relative error.

Theorem: If fl(x) is the n – digit floating point representation in base β of a real number x, then r_x the relative error in x satisfies the following:

(i) $|r_x| < \frac{1}{2}\beta^{1-n}$ if rounding is used.

(ii) $0 \leq |r_x| \leq \beta^{1-n}$ if chopping is used.

For proving (i), you may use the following:

Case 1.

$$d_{n+1} < \frac{1}{2}\beta, \text{ then } fl(x) = \pm(.d_1d_2...d_n)\beta^e$$

$$|x\text{-}fl(x)| = d_{n+1}, d_{n+2} ...\beta^{e-n-1}$$

$$\leq \frac{1}{2}\beta.\beta^{e-n-1} = \frac{1}{2}\beta^{e-n}$$

Case 2.

$$d_{n+1} \geq \frac{1}{2}\beta,$$

$$fl(x) = \pm\{(.d_1d_2...d_n)\beta^e + \beta^{e-n}\}$$

$$|x\text{-}fl(x)| = .\left|-d_{n+1}, d_{n+2}\ .\beta^{e\text{-}n\text{-}1} + \beta^{e-n}\right|$$

$$= \beta^{e-n-1}\left|d_{n+1}.\ d_{n+2} - \beta\right|$$

$$\leq \beta^{e-n-1} \times \frac{1}{2}\beta = \frac{1}{2}\beta^{e-n}$$

1.2.2 Sources of Errors

We list below the types of errors that are encountered while carrying out numerical calculation to solve a problem.

1. Round off errors arise due to floating point representation of initial data in the machine. Subsequent errors in the solution due to this is called propagated errors.

2. Due to finite digit arithmetic operations, the computer generates, in the solution of a problem errors known as generated errors or rounding errors.

3. Sensitivity of the algorithm of the numerical process used for computing f(x): if small changes in the initial data x lead to large errors in the value of f(x) then the algorithm is called *unstable*.

4. Error due to finite representation of an inherently infinite process. For example, consider the use of a finite number of terms in the infinite series expansions of

Sin x, Cos x or f(x) by Maclaurin's or Taylor Series expression. Such errors are called truncation errors.

Generated Error

Error arising due to inexact arithmetic operation is called generated error. Inexact arithmetic operation results due to finite digit arithmetic operations in the machine. If arithmetic operation is done with the (ideal) infinite digit representation then this error would not appear. During an arithmetic operation on two floating point numbers of same length n, we obtain a floating point number of different length m (usually m > n). Computer can not store the resulting number exactly since it can represent numbers a length n. So only n digits are stored. This gives rise to error.

Example 3: Let $a = .75632 \times 10^2$ and $b = .235472 \times 10^{-1}$
$a + b = 75.632 + 0.023$
$= 75.655472$ in accumulator
$a + b = .756555 \times 10$ if 6 decimal digit arithmetic is used.

We denote the corresponding machine operation by superscript * i.e.

$$a + {}^* b = .756555 \times 10^2 \ (.756555E2)$$

Example 4: Let $a = .23 \times 10^1$ and $b = .30 \times 10^2$

$$\frac{a}{b} = \frac{23}{300} = (0.075666E2)$$

If two decimal digit arithmetic is used then $\frac{a}{b}{}^* = .76 \times 10^{-1}$ (0.76E – 1)

In general, let w^* be computer operation corresponding to arithmetic operation w on x and y.

Generated error is given by $xwy - xw^*y$. However, computers are designed in such a way that

$xw^*y = fl(xwy)$. So the relative generated error

$$r.g.e. = r_{xwy} = \frac{xwy - xw^* y}{xwy}$$

we observe that in n – digit arithmetic

$$|r.g.e.| < \frac{1}{2}\,\beta^{1-n}, \text{ if rounding is used.}$$

$$0 \le |r.g.e.| < \beta^{1-n}, \text{ if chopping is used.}$$

Due to generated error, the associative and the distributive laws of arithmetic are not satisfied in some cases as shown below:

In a computer $3 \times \frac{1}{3}$ would be represented as 0.999999 (in case of six significant digit) but by hand computation it is one. This simple illustration suggested that everything does not go well on computers. More precisely 0.333333 + 0.333333 +0.333333 = 0.999999.

1.2.3 Non-Associativity of Arithmetic

Example 5: Let a = 0.345×10^0, b = 0.245×10^{-3} and c = 0.432×10^{-3}. Using

3-digit decimal arithmetic with rounding, we have

$$\begin{aligned}
b + c &= 0.000245 + 0.000432 \\
&= 0.000677 \text{ (in accumulator)} \\
&= 0.677 \times 10^{-3} \\
a + (b + c) &= 0.345 + 0.000677 \text{ (in accumulator)} \\
&= 0.346 \times 10^0 \text{ (in memory) with rounding} \\
a + b &= 0.345 \times 10^0 + 0.245 \times 10^{-3} \\
&= 0.345 \times 10^0 \text{ (in memory)} \\
(a + b) + c &= 0.345432 \text{ (in accumulator)} \\
&= 0.345 \times 10^0 \text{ (in memory)}
\end{aligned}$$

Hence we see that

$$(a + b) + c \neq a + (b + c)$$

Example 6: Let a = 0.41, b = 0.36 and c = 0.70.

Using two decimal digit arithmetic with rounding we have,

$$\frac{(a-b)}{c} = .71 \times 10^{-1}$$

and $\frac{a}{c} - \frac{b}{c} = .59 - .51 = .80 \times 10^{-1}$

while true value of $\frac{(a-b)}{c}$ = 0.071428 ….

i.e. $\frac{(a-b)}{c} \neq \frac{a}{c} - \frac{b}{c}$

These above examples show that error is due to finite digit arithmetic.

Definition 5: If x* is an approximation to x, then we say that x* approximates x to n significant β digits provided absolute error satisfies

$$\left| x - x^* \right| \le \frac{1}{2} \beta^{s-n+1},$$

with s the largest integer such that $\beta^s \le |x|$.

From the above definition, we derive the following:

x* is said to approximate x correct to n – significant β digits, if

$$\frac{\left| x - x^* \right|}{x} \le \frac{1}{2} \beta^{1-n}$$

In numerical problems we will use the following modified definition.

Definition 6: x^* is said to approximate x correct to n decimal places (to n places after the decimal)

If $\left| x - x^* \right| \le \frac{1}{2} 10^{-n}$

In n β –digit number, x^* is said to approximate x correct to n places after the dot if $\frac{\left| x - x^* \right|}{x} \le \beta^{-n}$.

Example7: Let x^* = .568 approximate to x = .5675

$x - x^* = -.0005$

$$\left|x - x^*\right| = 0.0005 = \frac{1}{2}(.001) = \frac{1}{2} \times 10^{-3}$$

So x^* approximates x correct to 3 decimal place.

Example 8: Let x = 4.5 approximate to x = 4.49998.

$$x - x^* = -.00002$$

$$\frac{\left|x - x^*\right|}{x} = 0.0000044 \leq .000005$$

$$\leq \frac{1}{2}(.00001) = \frac{1}{2}10^{-5} = \frac{1}{2} \times 10^{1-6}$$

Hence, x^* approximates x correct to 6 significant decimal digits.

1.2.4 Propagated Error

In a numerical problem, the true value of numbers may not be used exactly i.e. in place of true values of the numbers, some approximate values like floating point numbers are used initially. The error arising in the problem due to these inexact/approximate values is called propagated error.

Let x^* and y^* be approximations to x and y respectively and w denote arithmetic operation.

The propagated error = $xwy - x^* wy^*$

r.p.e. = relative propagated error

$$= \frac{xwy - x^* wy^*}{xwy}$$

Total Error: Let x^* and y^* be approximations to x and y respectively and let w^* be the machine operation corresponding to the arithmetic operation w. Total relative error

$$r_{xwy} = \frac{xwy - x^* w^* y^*}{xwy}$$

$$= \frac{xwy - x^* wy^*}{xwy} + \frac{x^* wy^* - x^* w^* y^*}{xwy}$$

$$= \frac{xwy - x^* wy^*}{xwy} + \frac{x^* wy^* - x^* w^* y^*}{x^* wy^*}$$

for the first approximation. So total relative error = relative propagated error + relative generated error.

Therefore, $\left|r_{xwy}\right| < 10^{1-n}$ if rounded.
$\left|r_{xwy}\right| < 2.10^{1-n}$ if chopped.
Where $\beta = 10$.

Propagation of error in functional evaluation of a single variable.

Let f(x) be evaluated and x^* be an approximation to x. Then the (absolute) error in evaluation of f(x) is $f(x) - f(x^*)$ and relative error is

$$r_{f(x)} = \frac{f(x) - f(x^*)}{f(x)} \qquad (1)$$

suppose x = $x^* + e_x$, by Taylor's Series, we get $f(x) = f(x^*) + e_x f(x^*) + \ldots$ neglecting higher order term in e_x in the series, we get

$$r_{f(x)} = \frac{e_x f(x^*)}{f(x)} - \frac{e_x}{x} \cong \frac{xf'(x^*)}{f(x)} = r_x . \frac{xf(x^*)}{f(x)}$$

$$|r_{f(x)}| = |r_x| \left| \frac{xf(x^*)}{f(x)} \right|$$

Note: For evaluation of f(x) in denominator of r.h.s. after simplification, f(x) must be replaced by $f(x^*)$ in some cases. So

$$|r_{f(x)}| = |r_x| \left| \frac{xf'(x^*)}{f(x)} \right|$$

The expression $\left| \frac{xf'(x^*)}{f(x)} \right|$ is called condition number of f(x) at x. The larger the condition number, the more ill-conditioned the function is said to be.

Example 9:

1. Let $f(x) = x^{\frac{1}{10}}$ and x approximates x^* correct to n significant decimal digits. Prove that $f(x^*)$ approximates f(x) correct to (n+1) significant decimal digits.

$$r_{f(x)} = r_x . \frac{xf'(x^*)}{f(x)}$$

$$= r_x . \frac{x \cdot \frac{1}{10} x^{*-\frac{9}{10}}}{x^{\frac{1}{10}}}$$

$$= \left(\frac{1}{10}\right) r_x$$

$$|r_{f(x)}| = \left(\frac{1}{10}\right) |r_x| \le \frac{1}{10} . \frac{1}{2} . 10^{1-n} = \frac{1}{2} 10^{1-(n+1)}$$

Therefore, $f(x^*)$ approximates f(x) correct to (n + 1) significant digits.

Example 10: The function $f(x^*) = e^x$ is to be evaluated for any x, $0 \le x \le 50$, correct to at least 6 significant digits. What digit arithmetic should be used to get the required accuracy?

$$|r_{f(x)}| = |r_x| \left| \frac{xf'(x^x)}{f(x)} \right|$$

$$= |r_x| \left| \frac{x.e^{x^*}}{e^x} \right|$$

$$= |r_x| \, |x|$$

Let n digit arithmetic be used, then

$$|r_x| < \frac{1}{2} 10^{1-n}$$

This is possible, if $|x| \, |r_x| \le \frac{1}{2} 10^{1-6}$

or $50. \frac{1}{2} 10^{1-n} \le \frac{1}{2} 10^{1-6}$

$$.\frac{1}{2}\,10^{1-n} \le \left(\frac{1}{100}\right)10^{1-6}$$

$$10^{1-n} \le 2.10^{1-8}$$

$$\text{or } 10^{-n} \le 10^{-8}.2$$

$$-n \le -8 + log_{10}^{2}$$

$$8 - log_{10}^{2} \le n \text{ or } 8 - .3 \le n$$

That is n ≥ 8.

Hence, n ≥ 8 digit arithmetic must be used.

Propagated Error in a function of two variables.

Let x^* and y^* be approximations to x and y respectively.

For evaluating f(x, y), we actually calculate $f(x^*, y^*)$

$e_{f(x,\,y)} = f(x, y) - f(x^*, y^*)$

but $f(x, y) = f(x^* + e_x, y^* + e_y)$

$= f(x^*, y^*) + (e_x f_x + e_f f_f)_{(x^*,\,y^*)}$ – higher order term. Therefore, $e_{f(x,\,y)} = (e_x f_x + e_f f_f)_{(x^*,\,y^*)}$. For relative error divide this by f(x, y).

Now we can find the results for propagated error in an addition, multiplication, subtraction and division by using the above results.

(a) **Addition**: $f(x,y) = x + y$

$$e_{x+y} = e_x + e_y$$

$$r_{x+y} = \frac{xe_x}{x(x+y)} + \frac{ye_y}{y(x+y)}$$

$$= r_x \frac{x}{x+y} + r_y \frac{y}{x+y}$$

(b) **Multiplication**: $f(x,y) = xy$

$$e_{x+y} = e_x y + e_y x$$

$$r_{xy} = \frac{e_x}{x} + \frac{e_y}{y}$$

$$= r_x + r_y$$

(c) **Subtraction**: $f(x,y) = x - y$

$$e_{x-y} = e_x y - e_y x$$

$$r_{x-y} = \frac{xe_x}{x(x-y)} - \frac{ye_y}{y(x-y)}$$

$$= r_x \frac{x}{x-y} + r_y \frac{y}{x-y}$$

(d) **Division**: $f(x,y) = \frac{x}{y}$

$$e_{\frac{x}{y}} = e_x.\frac{1}{y} - e_y.\frac{x}{y^2}$$

$$r_{\frac{x}{y}} = \frac{e_x}{x} - \frac{e_y}{y}$$

$$= r_x - r_y$$

1.3 SOME PITFALLS IN COMPUTATIONS

As mentioned earlier, the computer arithmetic is not completely exact. Computer arithmetic sometimes leads to undesirable consequences, which we discuss below:

1.3.1 Loss of Significant Digits

One of the most common (and often avoidable) ways of increasing the importance of an error is known as loss of significant digits.

Loss of significant digits in subtraction of two nearly equal numbers:

The above result of subtraction shows that x and y are nearly equal then the relative error

$$r_{x-y} = r_x \frac{x}{x-y} - r_y \frac{y}{x-y}$$

will become very large and further becomes large if r_x and r_y are of opposite signs.

Suppose we want to calculate the number z = x – y and x^* and y^* are approximations for x and y respectively, good to r digits and assume that x and y do not agree in the most left significant digit, then $z^* = x^* - y^*$ is as good approximation to x – y as x^* and y^* to x and y..

But if x^* and y^* agree at left most digits (one or more) then the left most digits will cancel and there will be loss of significant digits.

The more the digit on left agrees the more loss of significant digits would take place. A similar loss in significant digits occurs when a number is divided by a small number (or multiplied by a very large number).

Remark 1

To avoid this loss of significant digits, in algebraic expressions, we must rationalize and in case of trigonometric functions, Taylor's series must be used.

If no alternative formulation to avoid the loss of significant digits is possible, then carry more significant digits in calculation using floating-using numbers in double precision.

Example 11: Let $x^* = .3454$ and $y^* = .3443$ be approximations to x and y respectively correct to 3 significant digits. Further let $z^* = x^* - y^*$ be the approximation to x – y, then show that the relative error in z^* as an approximation to x – y can be as large as 100 times the relative error in x or y.

Solution:

Given, $|r_x|, \; |r_y|, \leq \frac{1}{2} 10^{1-3}$

$z^* = x^* - y^* = .3454 - .3443$

$= .0011$

$= .11 \times 10^{-2}$

This is correct to one significant digit since last digits 4 in x^* and 3 in y^* are not reliable and second significant digit of i^* is derived from the fourth digits of x^* and y^*.

Max. $|r_z| = \frac{1}{2} 10^{1-1} = \frac{1}{2} = 100. \frac{1}{2}.10^{-2}$

$\geq 100 \; |r_x|, 100 \; |r_y|$

Example 12: Let x = .657562 × 10^3 and y = .657557 × 10^3. If we round these numbers then

$x^* = .65756 \times 10^3$ and $y^* = .65756 \times 10^3$. (n = 5)
$x - y = .000005 \times 10^3 = .005$

while $x^* - y^* = 0$, this is due to loss of significant digits.

Now

$$\frac{u}{x-y} = \frac{.253\times 10^{-2}}{.005} = \frac{253}{500} \neq \frac{1}{2}$$

whereas $\frac{u^*}{x^* - y} = \infty$

Example 13: Solve the quadratic equation $x^2 + 9.9\,x - 1 = 0$ using two decimal digit floating arithmetic with rounding.

Solution:

Solving the quadratic equation, we have

$$x = \frac{-b+\sqrt{b^2-4ac}}{2a} = \frac{-9.9+\sqrt{(9.9)^2-4.1.(-1)}}{2}$$

$$= \frac{-9.9+\sqrt{102}}{2} = \frac{-9.9+10}{2} = \frac{.1}{2} = .05$$

while the true solutions are – 10 and 0.1. Now, if we rationalize the expression.

$$x = \frac{-b+\sqrt{b^2-4ac}}{2a} = \frac{-4ac}{2a(b+\sqrt{b^2-4ac})}$$

$$= \frac{-2c}{b+\sqrt{b^2-4ac})} = \frac{2}{9.9+\sqrt{102}}$$

$$= \frac{2}{9.9+10} = \frac{2}{19.9} = \frac{2}{20} \cong .1\ .(0.1000024)$$

which is one of the true solutions.

1.3.2 Instability of Algorithms

An algorithm is a procedure that describes, an unambiguous manner, a finite sequence of steps to be performed in a specified order. The object of the algorithm generally is to implement a numerical procedure to solve a problem or to find an approximate solution of the problem.

In numerical algorithm errors grow in each step of calculation. Let ε be an initial error and $R_n(\varepsilon)$ represents the growth of an error at the nth step after n subsequence operation due to ε.

If $R_n(\varepsilon) \approx C\, n\, \varepsilon$, where C is a constant independent of n, then the growth of error is called linear. Such linear growth of error is unavoidable and is not serious and the results are generally accepted when C and ε are small. An algorithm that exhibits linear growth of error is stable.

If $|R_n(\varepsilon)| \approx Ck^n\varepsilon$, $k > 1$, $C > 0$, k and C are independent of n, then growth of error is called exponential. Since the term k^n becomes large for even relatively small values of n. The final result will be completely erroneous in case of exponential growth of error. Such algorithm is called unstable.

Example 14:

Let $y_n = n! \left\{ e - \left(1 + \frac{1}{1!} + \frac{1}{2!} +, + \frac{1}{n!} \right) \right\}$ (1)

$$y_n = \frac{1}{n+1} + \frac{1}{(n+1)(n+2)} + \ldots.. \quad (2)$$

$$y_n < \frac{1}{n} + \frac{1}{n^2} + \frac{1}{n^3} + \ldots.$$

$$0 \le y_n < \frac{\frac{1}{n}}{1 - \frac{1}{n}} = \frac{1}{n-1}$$

$$y_n \to 0 \text{ as } n \to \infty$$

i.e. $\{y_n\}$ is monotonically decreasing sequence which converges to zero. The value of y_9 using (2) is $y_9 = .10991$ correct to 5 significant figures.

Now if we use (1) by writing

$$y_{n+1} = (n+1)! \left\{ e - \left(1 + \frac{1}{1!} + \frac{1}{2!} +, + \frac{1}{(n+1)!} \right) \right\}$$

$$i.e., y_{n+1} = (n+1)\, y_n - 1$$

Using (3) and starting with

$y_0 = e - 1 = 1.7183$, we get
$y_1 = .7183$
$y_2 = .4366$
$y_3 = .3098$
$y_4 = .2392$
$y_5 = .1960$
$y_6 = .1760$
$y_7 = .2320$
$y_8 = .8560$
$y_9 = 6.7040$

This value is not correct even to a single significant digit, because algorithm is unstable. This is shown computationally. Now we show it theoretically.

Let y^*_n be computed value by (3), then we have

$$y_{n+1} = (n+1)\, y_n - 1$$
$$y^*_{n+1} = (n+1)\, y^*_n - 1$$

$$y_{n+1} - y^*_{n+1} = (n+1)\,(y_n - y^*_n)$$

i.e. $e_{n+1} = (n+1)\, e_n$

$e_{n+1} = (n+1)!\, e_o$

$|e_{n+1}| > 2^n\, |e_o|$ for $n > 1$

$|e_n| > \frac{1}{2}.2^n\, |e_o|$

Here k = 2, hence growth of error is exponential and the algorithm is unstable.

Example 15: The integral $E_n = \int_0^1 x^n e^{x-1} dx$ is positive for all $n \geq 0$. But if we integrate by parts, we get $E_n = 1 - nE_n$ $(= x^n e^{x-1} \int_0^1 - \int_0^1 n\, x^{n-1} e^{x-1} dx)$.

Starting from $E_1 = .36787968$ as an approximation to $\frac{1}{e}$ (accurate value of E_1) correct to 7 significant digits, we observe that E_n becomes negative after a finite number of iteration (in 8 digit arithmetic). Explain.

Solution

Let E^*_n be computed value of E_n.

$E_n - E^*_n = -n(E_{n-1} - E_{n-1})$

$e_n = (-1)^n\, n! e_n$

$|e_n| \geq ½ .2^n\, |e_o|$ hence process is unstable.

Using 4 digit floating point arithmetic and $E_1 = 0.3678 \times 10^0$ we have $E_2 = 0.2650$, $E_3 = 0.2050$, $E_4 = 0.1800$, $E_5 = 0.1000$, $E_6 = 0.4000$. By inspection of the arithmetic, the error in the result is due to rounding error committed in approximating E_2.

Correct values are $E_1 = 0.367879$, $E_2 = 0.264242$. Such an algorithm is known as an unstable algorithm. This algorithm can be made into a stable one by rewriting $E_{n-1} = \frac{1 - E_n}{n}$, n = … 4, 3, 2. This algorithm works backward from large n towards small number. To obtain a starting value one can use the following:

$$E_n = \leq \int_0^1 x^n d^x = \frac{1}{n+1}.$$

1.4 SUMMARY

In this unit we have covered the following:

After discussing floating-point representation of numbers we have discussed the arithmetic operations with normalized floating-point numbers. This leads to a discussion on rounding errors. Also we have discussed other sources of errors… like propagated errors loss of significant digits etc. Very brief idea about stability or instability of a numerical algorithm is presented also.

1.5 EXERCISES

E1) Give the floating point representation of the following numbers in 2 decimal digit and 4 decimal digit floating point number using (i) rounding and (ii)

chopping.
(a) 37.21829
(b) 0.022718
(c) 3000527.11059

E2) Show that a(b – c) ≠ ab – ac
where
$a = .5555 \times 10^1$
$b = .4545 \times 10^1$
$c = .4535 \times 10^1$

E3) How many bits of significance will be lost in the following subtraction?
37.593621 – 37.584216

E4) What is the relative error in the computation of x – y, where x = 0.3721448693 and y = 0.3720214371 with five decimal digit of accuracy?

E5) If x^* approximates x correct to 4 significant decimal figures/digits, then calculate to how many significant decimal figures/digits $e^{x^*/100}$ approximates $e^{x/100}$.

E6) Find a way to calculate
(i) $f(x) = \sqrt{x^2+1} - 1$
(ii) $f(x) = x - Sin\, x$
(iii) $f(x) = x - \sqrt{x^2 - \alpha}$

correctly to the number of digits used when it is near zero for (i) and (ii),very much larger than α for (iii)

E7) Evaluate $f(x) = \dfrac{x^3}{x - Sinx}$ when $x = .12 \times 10^{-10}$ using two digit arithmetic.

E8) Let u = $\dfrac{a-b}{c}$ and v = $\dfrac{a}{c} - \dfrac{b}{c}$ when a = .41, b = .36 and c = .70. Using two digit arithmetic show that $|e_v|$ is nearly two times $|e_u|$.

E9) Find the condition number of
(i) $f(x) = \sqrt{x}$
(ii) $f(x) = \dfrac{10}{1-x^2}$
and comment on its evaluation.

E10) Consider the solution of quadratic equation

$$x^2 + 111.11x + 1.2121 = 0$$

using five-decimal digit floating point chopped arithmetic.

1.6 SOLUTIONS/ANSWERS

E1) (a)

rounding	chopping
$.37 \times 10^2$	$.37 \times 10^2$
$.3722 \times 10^2$	$.3721 \times 10^2$

(b) $.23 \times 10^{-1}$ $.22 \times 10^{-1}$
$.2272 \times 10^{-1}$ $.2271 \times 10^{-1}$

(c) $.31 \times 10^{2}$ $.30 \times 10^{2}$
$.3056 \times 10^{2}$ $.3055 \times 10^{2}$

Note: Let x be approximated by
$a_p \ \ldots a_1 a_0 . a_{-1} a_{-2} \ .. \ a_{-q}$.
In case $a_{-q-1} > 5$, x is rounded to
$a_p \ \ldots a_1 a_0 . a_{-1} a_{-2} \ \ldots \ (a_{-q} + 1)$

In case $a_{-q-1} = 5$ which is followed by at least one non-zero digit, x is rounded to
$a_p \ \ldots a_1 a_0 . a_{-1} a_{-2} \ .. \ a_{-q+1} .(a_{-q} + 1)$

In case $a_{-q-1} = 5$, being the last non-zero digit, x is rounded to
$a_p \ \ldots a_1 a_0 . a_{-1} a_{-2} \ .. \ a_{-q}$
if a_{-q} is even or to
$a_p \ \ldots a_1 a_0 . a_{-1} a_{-2} \ .. \ a_{-q+1} .(a_{-q} + 1)$
If a_{-q} if odd.

E2) Let
$a = .5555 \times 10^1$
$b = .4545 \times 10^1$
$c = .4535 \times 10^1$

$b - c = .0010 \times 10^1 = .1000 \times 10^{-1}$

$a(b - c) = (.5555 \times 10^1) \times (.1000 \times 10^{-1})$
$= .05555 \times 10^0$
$= .5550 \times 10^{-1}$

$ab = (.5555 \times 10^1)\ (.4545 \times 10^1)$
$= (.2524 \times 10^2)$

$ac = (.5555 \times 10^1)\ (.4535 \times 10^1)$
$= (.2519 \times 10^2)$

and $ab - ac = .2524 \times 10^2 - .2519 \times 10^2$
$= .0005 \times 10^2$
$= .5000 \times 10^{-1}$

Hence $a(b - c) \neq ab - ac$

E3) $37.593621 - 37.584216$
i.e. $(0.37593621)10^2 - (0.37584216)10^2$

Here $x^* = (0.37593621)10^2$, $y^* = (0.37584216)10^2$
and assume each to be an approximation to x and y, respectively, correct to seven significant digits.

Then, in eight-digit floating-point arithmetic,
$= (0.00009405)10^2$
$z^* = x^* - y^* = (0.94050000)10^{-2}$
is the exact difference between x^* and y^*. But as an approximation to $z = x - y$, z^* is good only to three digits, since the fourth significant digit of z^* is derived from the eighth digits of x^* and y^*, and both possibly in error. Here while the error in z^* as an approximation to $z = x - y$ is at most the sum of the errors in x^*

and y^*, the relative error in z^* is possibly 10,000 times the relative error in x^* or y^*. Loss of significant digits is, therefore, dangerous only if we wish to keep the relative error small.

Given $|r_x|, |r_y| < \frac{1}{2} 10^{1-7}$

$z^* = (0.9405) 10^{-2}$

is correct to three significant digits.

Max $|r_z| = \frac{1}{2} 10^{1-3} = 10,000. \frac{1}{2} 10^{-6} \geq 10,000 |r_z|, 10,000 |r_y|$

E4) With five decimal digit accuracy

$x^* = 0.37214 \times 10^0 \qquad y^* = 0.37202 \times 10^0$

$x^* - y^* = 0.00012$ while $x - y = 0.0001234322$

$$\frac{|(x-y)-(x^*-y^*)|}{|x-y|} = \frac{0.0000034322}{0.0001234322} \approx 3 \times 10^{-2}$$

The magnitude of this relative error is quite large when compared with the relative errors of x^* and y^* (which cannot exceed 5×10^{-5} and in this case it is approximately 1.3×10^{-5})

E5) Here $f(x) = e^{x/100}$

$$r_{f(x)} \approx r_x . \frac{xf'(x^*)}{f(x)} \approx r_x \frac{xf'(x^*)}{f(x)} = r_x . e^{x/100} . \frac{1}{100} . \frac{1}{e^{x/100}}$$

i.e.

$$r_{f(x)} \approx \frac{1}{100} |r_x| \leq \frac{1}{100} . \frac{1}{2} 10^{1-4} = \frac{1}{2} 10^{1-6}.$$

Therefore, $e^{x^*/100}$ approximates $e^{x/100}$ correct for 6 significant decimal digits.

E6) (i) Consider the function:

$f(x) = \sqrt{x^2+1} - 1$ whose value may be required for x near 0. Since $\sqrt{x^2+1} \approx 1$ when $x \approx 0$, we see that there is a potential loss of significant digits in the subtraction. If we use five-decimal digit arithmetic and if $x = 10^{-3}$, then $f(x)$ will be computed as 0.

Whereas if we rationalise and write

$$f(x) = \frac{\left(\sqrt{x^2+1}-1\right)\left(\sqrt{x^2+1}+1\right)}{\left(\sqrt{x^2+1}+1\right)} = \frac{x^2}{\sqrt{x^2+1}+1}$$

we get the value as $\frac{1}{2} \times 10^{-6}$

(ii) Consider the function:

$f(x) = x - \text{Sin}\, x$ whose value is required near $x = 0$. The loss of significant digits can be recognised since Sin $x \approx x$ when $x \cong 0$.

To avoid the loss of significance we use the Taylor (Maclaurin) series for $\operatorname{Sin} x$

$$\operatorname{Sin} x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

Then $f(x) = x \ - Sin\, x = \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots$

The series starting with $\frac{x^3}{6}$ is very effective for calculation $f(x)$ when x is small.

(iii) Consider the function:

$$f(x) = x - \sqrt{x^2 - \alpha}$$

$$\text{as } f(x) = \frac{\left(x - \sqrt{x^2 - \alpha}\right)}{x + \sqrt{x^2 - \alpha}}\left(x + \sqrt{x^2 - \alpha}\right) = \frac{\alpha}{x + \sqrt{x^2 - \alpha}}$$

Since when x is very large compared to α, there will be loss of significant digits in subtraction.

E7)

$$Sinx = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots$$

$$Sinx = \left(.12\times10^{-10}\right) = .12\times10^{-10} - .17\times10^{-32} + \ldots \approx .12\times10^{-10}$$

So $f(x) = \frac{x^3}{x - Sinx} = \infty$

But $f(x) = \frac{x^3}{x - Sinx}$ can be simplified to

$$= \frac{x^3}{\frac{x^3}{3!} - \frac{x^5}{5!} + \ldots} = \frac{1}{\frac{1}{3!} - \frac{x^2}{5!} + \ldots}$$

The value of $\frac{x^3}{x - Sinx}$ for $= .12\times10^{-10}$

is $\frac{1}{\frac{1}{3!}} = 6.$

E8) Using two digit arithmetic

u = $\frac{a-b}{c}$ = .71 $\times 10^{-1}$

v = $\frac{a}{c} - \frac{b}{c}$ = .59 – .51 = .80 $\times 10^{-1}$

True value = .071428

u – fl(u) = $|e_u|$ = .000428

v – fl(v) = $|e_v|$ = .0008572

Thus, $|e_v|$ is nearly two times of $|e_u|$ indicating that u is more accurate than v.

E9) The word condition is used to describe the sensitivity of the function value f(x) to changes in the argument x. The informal formula for Condition of f at x

$$= max\left\{ \frac{f(x)-f(x^*)}{f(x)} \Big/ \left|\frac{x-x^*}{x}\right| : |x-x^*|'' \ small'' \right\}$$

$$\approx \left|\frac{f'(x)x}{f(x)}\right|$$

The larger the condition, the more ill-conditioned the function is said to be.

If $f(x)=\sqrt{x}$, the condition of f is approximately

$$\left|\frac{f'(x)x}{f(x)}\right| = \frac{\left[\frac{1}{2\sqrt{x}}\right]x}{\sqrt{x}} = \frac{1}{2}$$

This indicates that taking square root is a well conditioned process.

But if $f(x)=\dfrac{10}{1-x^2}$

$$\left|\frac{f^1(x)x}{f(x)}\right| = \frac{20x/(1-x^2)x}{10x/(1-x^2)} = \frac{2x^2}{|1-x^2|}$$

This number can be very large when x is near 1 or –1 signalling that the function is quite ill-conditioned.

E10) Let us calculate

$$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

$$x_1 = \frac{-111.11+111.09}{2}$$

$= -$ 0.01000

while in fact $x_1 = -0.010910,$ correct to the number of digits shown.

However, if we calculate x_1 as

$$x_1 = \frac{2c}{b+\sqrt{b^2-4ac}}$$

in five-decimal digit arithmetic $x_1 = -0.010910$ which is accurate to five digits.

$$x_1 = \frac{-2\times 1.2121}{111.11+111.09} \quad = \frac{-2.4242}{222.20}$$

$$= -\frac{24242}{2222000} = -0.0109099 \quad = -.0109099$$