APPENDIX : MORE DETAILED DISCUSSION OF NUMBERS & DISASTERS

The unit is an optional, though highly, desirable, reading. The unit is included for better understanding of number systems, which play significant role in better understanding of the discipline of Numerical Techniques

A.0 Introduction

We have earlier mentioned that the discipline of Numerical Techniques is about

- numbers, rather special type of numbers called computer numbers, and
- application of (some restricted version of) the four arithmetic operations, viz., + (plus), − (minus), * (multiplication) and ÷ (division) on these special numbers.

Therefore, let us, first, recall some important sets of numbers, which have been introduced to us earlier, from school days. Then we will discuss algebraic systems (**to be called simply *systems*)** of numbers, and finally notations for numbers.

A.1 Sets of Numbers

God made the natural numbers, rest made the man
KrÖnecker

Set of **Natural numbers** denoted by **N**, where N = {0, 1, 2, 3, 4, ... } or N = {1, 2, 3, 4, ….}
Set of **Integers** denoted by **I, or Z,** where I = { …., −4, −3, −2, −1, 0, 1, 2, 3, 4, …..}

Justification of the KrÖnecker's statement can be seen through the following explanation: An *integer* can be considered as an ordered pair (m, n) of *natural* numbers. For example, −3 may be considered as the pair (2, 5), or (4, 7) of natural numbers and integer 3 as (5, 2), or (7, 4). Further, operations on integers, in this representation, can be realized as:

$(m_1, n_1) + (m_2, n_2) = (m_1 + m_2, n_1 + n_2)$, $(m_1, n_1) - (m_2, n_2) = (n_1 + n_2, m_1 + m_2)$
and $(m_1, n_1) * (m_2, n_2) = (m_1 m_2 + n_1 n_2 , n_1 m_2 + m_1 n_2)$ etc.

Similarly, members of sets like Q etc., discussed below can be structured directly or indirectly from N

Set of **Rational Numbers** denoted by **Q,** where
Q = {a/b, where *a* and *b* are integers and *b* is not 0 }

Set of **Real Numbers** denoted by **R**. ….. There are different ways of looking at or thinking of Real Numbers. ***One of the intuitive ways of thinking of real numbers*** is as the numbers that correspond to the points on a straight line extended infinitely in both the

directions, such that one of the points on the line is marked as 0 and another point (different from, and to the right of, the earlier point) is marked as 1. Then to each of the points on this line, a unique real number is associated.

A more formal way is to consider the set of real numbers as extension of the rational numbers, where a real number is the limit of a convergent sequence of rational numbers. … There is a large subset of real numbers, no member of which is a rational number. A real number which is not a rational number is called **irrational number. For example,** $\sqrt{2}$ is an irrational number.

Set of **Complex Numbers** denoted by **C**, where C = {$a + bi$ or $a + ib$ where a and b are real numbers and i is the square root of −1}

By minor notational modifications (e.g., by writing an integer, say, 4 as a rational number 4/1; and by writing a real number, say $\sqrt{2}$ as a complex number $\sqrt{2} + 0\,i$), we can easily see that $N \subset I \subset Q \subset R \subset C$.

When we do not have any specific set under consideration, the set may be referred to as a ***set of numbers***, and a member of the set as just ***number.***

Apart from these well-known sets of numbers, there are sets of numbers that may be useful in our later discussion. Next, we discuss two such sets.

Set of algebraic Numbers *(no standard notation for the set)*, where an *Algebraic number* is a number that is a root of a non-zero polynomial equation[1] with rational numbers as coefficients. For example,

- Every rational numbers is algebraic (e.g., the rational number a/b, with $b \neq 0$, is a root of the polynomial equation: $bx - a = 0$). Thus, a real number, which is not algebraic, must be irrational number.
- Even, some irrational numbers are algebraic, e.g., $\sqrt{2}$ is an algebraic number, because, it satisfies the polynomial equation: $x^2 - 2 = 0$. In general, n^{th} root of a rational number a/b, with $b \neq 0$, is algebraic, because, it is a root of the polynomial equation: $b*x^n - a = 0$
- Even, a complex number may be an algebraic number, as each of the complex numbers $\sqrt{2}\,i$ (= $0 + \sqrt{2}\,i$) and $-\sqrt{2}\,i$ is algebraic, because, each satisfies the polynomial equation: $x^2 + 2 = 0$.

Set of Transcendental Numbers *(again, no standard notation for the set),* where, a *transcendental number* is a complex number (and, hence, also, a real number, as, $R \subset C$), which is not algebraic. From the above examples, it is clear that a rational number can not be transcendental, and some, but not all, irrational numbers, may be transcendental. The most prominent examples of transcendental numbers are $\boldsymbol{\pi}$ and $\boldsymbol{e}$. [2]

---

[1] We may recall that a polynomial P(x) is an expression of the form: $a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots\ldots + a_{n-1}x + a_n$, where a_i is a number and x is a variable. Then, P(x) = 0 represents a polynomial equation.

[2] It should be noted that it is quite complex task to show a number as transcendental. In order to show a number, say, n, to be transcendental, theoretically, it is required to ensure that for *each* polynomial equation P(x) = 0, n is not a root of the equation. And, there are infinitely many polynomial equations. This direct method for showing a number as transcendental, can not be used. There are other methods for the purpose.

A.2 Algebraic Systems of Numbers (To be called, simply, ***Systems of Numbers***)

In order to discuss, system of numbers, to begin with, we need to understand the concept of ***operation on a set*** For this purpose, recall that N, the set of Natural numbers, **is closed under** '+' (plus). By ***'N is closed under +'*** ,we mean: if we take (any) two natural numbers, say m and n, then m + n is *also* a natural number.

But, N, the set of Natural numbers, is not closed under '–' (minus). In other words, *for some* natural numbers m and n, m – n may not be a natural number, for example, for 3 and 5, 3 – 5 is not a natural number.(Of course, 3 – 5 = – 2 is an integer)
These facts are also stated by saying: '+' is a binary operation on N, but, '–' is not a binary operation on N. Here, the word ***binary*** means that in order to apply '+', we need (exactly) two members of N.

In the light of above illustration of binary operation, we may recall many such statements including the following:

(i) * (multiplication) is a binary operation on N (or, equivalently, we can say that N is closed under the binary operation *)

(ii) – (minus) is a binary operation on I (or, equivalently, we can say that I is closed under the binary operation –) etc.[3]

However, there are operations on numbers, which may require only one number ***(the number is called argument of the operation)*** of the set. For example, The ***square*** operation on a set of numbers takes only one number and returns its square, for example, square(3) = 9.

Thus, some operations (e.g., square) on a set of numbers may take only one argument. Such operations are called ***unary*** *operations.* Other operations may take two arguments (e.g., +, – , *, ÷) from a set of numbers. Such operations are called ***binary*** *operations.*

There are operations which may take three arguments and are called ***ternary operations***. There can be operations requiring no argument (e.g., multiplicative identity of the set of numbers) and there can be operations requiring 4 or more arguments. ***But one of the defining characteristics of an operation on a set is that the result of the operation must be a unique member of the same set.***

Definition: Algebraic System of Numbers: A set of numbers, say, S, along with a (finite) set of operations on S, is called an algebraic system of numbers. In stead of 'algebraic system', we may use the word **'system'**.

Notation for a System: If O_1 , O_2 ,….. O_n are some n operations on a set S, then, we denote the corresponding system ***as*** $<S, O_1, O_2, \ldots O_n>$, ***or as*** $(S, O_1, O_2, \ldots O_n)$.

Examples & Non-examples of Systems of Numbers:

---

[3] (iii) * (multiplication) is a binary operation on I(or, equivalently, we can say that I is closed under the binary operation *)
(iv) ÷ (division) is neither a binary operation on N nor on I.
(v) ÷ (division) is a binary operation on Q ~{0} , and (vi) on R~ {0}, and also (vii) on C~ {0}.

1. **Examples of number systems**

Each of following is a system of numbers: $< N, + >$, $< N, * >$, and $< N, +, * >$ etc.[4]

2. **Non-examples of number systems**

Each of following is NOT a system of numbers: $< N, - >$, $< N, \div >$, $< N, -, \div >$ $< I, \div >$ etc. [5]

Some more operations on Sets of numbers:

zero-ary operations:

(i) The numeral 1 is multiplicative identity for every number (i.e., 1* n = n, for every number n). Thus '1 as multiplicative identity' may be treated as an operation. It is a zero-ary operation on each of N, I, Q, R and C. *(because, we are not required to supply any number to know its multiplicative identity. On the other hand, to know the result on application of +, we must supply two numbers)*

(ii) The numeral 0 is additive identity for every number (i.e., 0+ n = n, for every number n). Thus '0 as additive identity' may be treated as an operation. It is a zero-ary operation on each of N, I, Q, R and C.

Unary operations:

(i) We know *square* (let us denote it by *Sq*) of a natural number is also a natural Number. Thus, Sq is an operation on N. *(As it requires only one number to return the answer, it is a unary operation.)* Similarly, Sq is a unary operation on each of I, Q, R and C.

(ii) **The square-rooting ($\sqrt{}$)** *is **not** an operation on N (because, $\sqrt{2}$ is not in N, whereas 2 is in N). Similarly,* square-rooting *is not an operation on I, Q, R(because, $\sqrt{(-2)}$ is not in I, also not in Q, and also not in R, therefore, it is not an operation on I, not an operation on Q, and it is not an operation on R.*

But, **square-rooting** *is an operation on C. Also, as it requires only one (complex) number to return the answer, it is a **unary** operation on C*

(iii) For any natural number $n \geq 2$, taking n^{th} root of a number **($\sqrt[n]{}$)** *is **not** an operation on N (because, $\sqrt[n]{2}$ is not in N, for 2 in N). Similarly, it is not an operation on I, Q, R(because, $\sqrt[n]{(-2)}$ is not in I, also not in*

---

[4] **Each of following is also a system of numbers:** $< I, + >$, $< I, - >$, $< I, * >$, $< I, +, * >$, $< I, +, - >$, $< I, +, -, * >$ and $< Q, + >$, $< Q, - >$, $< Q, * >$, $< Q, +, * >$, $< Q, +, - >$, $< Q, +, -, * >$, $< Q \sim \{0\}, \div >$, $< Q \sim \{0\}, +, \div >$, $< Q \sim \{0\}, +, -, *, \div >$ and $< R, + >$, $< Q, - >$, $< R, * >$, $< Q, +, * >$, $< R, +, - >$, $< R, +, -, * >$, $< R \sim \{0\}, \div >$, $< R \sim \{0\}, +, \div >$, and $< R \sim \{0\}, +, -, *, \div >$ and $< C, + >$, $< C, - >$, $< C, * >$, $< C, +, * >$, $< C, +, - >$, $< C, +, -, * >$, $< C \sim \{0\}, \div >$, $< C \sim \{0\}, +, \div >$, and $< C \sim \{0\}, +, -, *, \div >$.

[5] Each of following is also NOT a system of numbers: **$< Q, +, -, *, \div >$ and $< R, \div >$, $< R, +, \div >$, $< R, +, -, *, \div >$ and $< C, \div >$, $< C, +, \div >$, $< C, +, -, *, \div >$. because division by 0 is not defined**

Q, and also not in R, therefore, it is not an operation on I, not an operation on Q, and it is not an operation on R.

But, taking n^{th} root of a number ($\sqrt[n]{}$) *is an operation on C. Also, as it requires only one (complex) number to return the answer, it is a* ***unary*** *operation on C*

3. Some more examples of number systems

By adding any of one, two or more of the operations, viz., Sq, 1, 0, $\sqrt{}$, and $\sqrt[n]{}$ to some of the number systems mentioned above, we may get a new number system.

For example, <N, +, *, 1, Sq > is a number system….

Similarly, < C $\sim\{0\}$, +, −, *, $\div$, $\sqrt[n]{}$ > is a number system.

4. Some more Non-examples of number systems

However, each of following is **NOT** a system of numbers:

< N, + , $\sqrt[n]{}$ >, < N, *, $\sqrt[n]{}$ >, and < N, +, *, $\sqrt[n]{}$ > etc.[6]

Apart from *operations,* various number systems have *relations*[7], which also may be unary, binary etc. For, example, '<' is a binary relation on < N, +, * >. Actually, '<' is a binary relation on each of the number system discussed above, except systems on C, the set of complex numbers. Then, we can define '*the minimum element*' or simply, *'the minimum'* and *'the maximum element'* or simply, *'the maximum'* of a number system, in the usual sense of these terms.

A.3 Some Properties of Sets and Systems of Numbers:

---

1. Each of the set N, I, Q, R and C is an *infinite* set.
2. None of the number systems discussed above, is bounded above, i.e., has the *maximum element*.
3. N has *the minimum element* (0,if, N is taken as {0, 1, 2,3, …..} and 1, if N is taken as {1, 2,3, …..})….. But, *none* of the other number systems mentioned above has the minimum element.
4. The set of real numbers does not have the *least positive real number*. Because, between 0 and any positive real number, say r, lies the positive real number r/2…..The same is true of rational numbers.
5. For each of the relevant number systems, mentioned above, on the sets N, I, Q , R , and C , each of the following holds

---

[6] Each of following is also **NOT** a system of numbers:
< I, +, $\sqrt[n]{}$ >, < I, −, $\sqrt[n]{}$ >, < I, *, $\sqrt[n]{}$ >, < I, +, *, $\sqrt[n]{}$ >, < I, +, − , *, $\sqrt[n]{}$ > and
< Q, +, $\sqrt[n]{}$ >, < Q, −, $\sqrt[n]{}$ >, < Q, *, $\sqrt[n]{}$ >, < Q, −, *, $\sqrt[n]{}$ > , < Q, −, −, $\sqrt[n]{}$ > .

[7] An *operation,* say +, on a set of numbers, again say, N, takes two numbers and *returns (i.e., gives an answer as) a number, in this case, an element of N.* Similarly, a *relation*, which also may be unary, binary, ternary, etc, takes appropriate number of numbers (in the case of binary relation, it takes two) **but** *returns (i.e., gives an answer as) 'True' or 'False'.* For example, the relation of '<' takes two integers, say 3 and 5 and returns 'True', because, 3< 5 is True. However, if 7 and 5 are given as arguments, then it returns 'False', because, 7< 5 is false.

(i) '+' is Commutative in numbers, i.e., $x + y = y + x$, for any numbers x and y

(ii) '*' is Commutative in numbers, i.e., $x * y = y * x$, for any numbers x and y

(iii) '+' is Associative in numbers, i.e., $(x + y) + z = x + (y + z)$, for any numbers x, y and z

(iv) '*' is Associative in numbers, i.e., $(x * y) * z = x * (y * z)$, for any numbers x, y and z,

However,

(v) '−' is NOT Associative in numbers, i.e., $(x - y) - z = x - (y - z)$, for any numbers x, y and z

For example, $10 - (4 - 6) = 8 \neq 0 = (10 - 4) - 6$

and

(vi) '÷' is NOT Associative in numbers, i.e., $(x \div y) \div z = x \div (y \div z)$, for any numbers x, y and z

for example, $128 \div (8 \div 4) = 64 \neq 4 = (128 \div 8) \div 4$

(vii) '*' is Left & Right Distributive over '+', i.e.,

(a) $x * (y + z) = (x * y) + (x * z)$, for numbers x, y and z *(left)*

(b) $(y + z) * x = (y * x) + (z * x)$, for numbers x, y and z *(right)*

(as, '' is both left and right distributive over '+', **we just say** '*' is distributive over '+')*

(viii) '*' is Distributive over **'−', i.e.,**

(a) $x * (y - z) = (x * y) - (x * z)$, for numbers x, y and z *(left)*

(b) $(y - z) * x = (y * x) - (z * x)$, for numbers x, y and z *(right)*

(ix) '$\div$' is (only) right distributive over the operation '+', i.e.,

$(y + z) \div x = (y \div x) + (z \div x)$, for numbers x, y and z

(x) '$\div$' is (only) right distributive over the operation '−', i.e.,

$(y - z) \div x = (y \div x) - (z \div x)$, for numbers x, y and z **.**

However,

(xi) the operation '$\div$' is NOT Left distributive over the operation '+', i.e.,

$x \div (y + z) \neq (x \div y) + (x \div z)$, for numbers x, y and z in Q, R or C

For example, $120 \div (4 + 6) = 12 \neq 50 = (120 \div 4) + (120 \div 6)$. *A*nd also,

and

(xii) the operation '$\div$' is NOT Left distributive over the operation '−' i.e.,

$x \div (y - z) \neq (x \div y) - (x \div z)$, for numbers x, y and z in Q, R or C

For example, $120 \div (4 - 6) = 60 \neq 10 = (120 \div 4) - (120 \div 6)$

Remark 1: *The above discussion in this subsection, of the properties of numbers is significant in the light of the fact that many of the above mentioned properties of numbers may not hold in the set of numbers that can be stored in a computer system.*

Remark 2: *In the above discussion, the use of the word **number** is inaccurate. Actually, a number is a concept (a mental entity), which may be represented in some (physical) forms, so that we can experience the concept through our senses. The number, the name of which is, say,* **ten** *in English and* **नौ** *in Hindi, and* zehn *in German language may be represented as 10 as decimal numeral , X as Roman numeral, 1010 as binary numeral. As, you may have already noticed, the physical representation of a number is called its **numeral**. Thus, number and numeral are two different*

entities, incorrectly taken to be the same. Also, a particular number is unique, but, it can have many (physical) representations, each being called a numeral, corresponding to the number.

The difference between ***number*** *and* ***numeral*** *may be further clarified from the following explanation: We have the* ***concept*** *of the animal* ***that is called*** COW *in English,* **xk;** *in Hindi and* KUH *in German language. The animal, represented as* cow *in English, has four legs; however, its representation in English:* ***cow,*** *is a word in English and has three letters, but does not have four legs.*

However, due to usage, though inaccurate, over centuries, in stead of the word ***numeral,*** *almost, the word* ***numbe****r is used. Except for the discussion of Subsection 1.0.4, we will also not differentiate between Number and Numeral.*

A good notation

A. 4 Numerals: Notations for Numbers

First, we recall some well-known sets used to denote numbers. These sets are called *sets of numerals* and then discuss various *numeral systems,* developed on these numeral sets, for representing numbers.

A.4.0 Sets of Numerals: We are already familiar with some of the sets of numerals. The most familiar, and frequently used, set is **Decimal** *Numeral Set*. It is called *Decimal,* because, it uses ten figures, or digits*, viz., digits from the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} of ten digits.*

Another numeral set, familiar to computer science students, is **binary** numeral set. It is called binary*,* because, it uses two figures, or digits*, viz., digits from the set {0, 1} of two digits.* In this case, 0 and 1 are called ***bits***.

Also, ***Roman Numeral Set,*** is well-known. This set uses figures/ digits/ letters from the set { I, V, X, L, C, D, M, …}, where, I represents 1 (of decimal numeral system), V represents 5, X represents 10, L represents 50, C represents 100, D represents 500 and M represents 1000. etc.[8]

Apart from these sets of numerals, in context of computer systems, we also come across

i. **Hexadecimal numeral set,** which uses figures/ digits, viz., from the set: *{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E} of sixteen digits.*

ii. **Octal numeral set,** which uses figures/ digits, viz., from the set: *{0, 1, 2, 3, 4, 5, 6, 7} of eight digits.*

iii. **Ternary numeral set,** which uses figures/ digits, viz., from the set: *{0, 1, 2} of three digits*.

iv. **Base/ radix r numeral set**, where r is a natural number, which uses figures/ digits, viz., from the set: *{0, 1, 2, ..., r-1 } of r digits*.

Except for set of Roman numerals, all other sets of numerals may be considered as set of radix r numerals, with r = 2 for binary, r = 8 for octal, r = 10 for decimal and r = 16 for hexadecimal

[8]Appropriate choice of numeral system has significant role in solving problems, particularly solving problems *efficiently*. For example, it is a child's play to get the answer for 46* 37 (in decimal numeral system) as a single number. However, using Roman numerals, i.e., writing VLI * XXXVII, in stead of , 46 * 37, it is really very difficult to get the same number, using only Roman numerals, as answer.

A.4.1 Number representation using a set of numerals: a number is represented by a *string of digits* from the set of numerals under consideration, e.g., 3426 in decimal, IX in Roman 10010110 in binary and 37A08 in hexadecimal.

A.4.2 Value of the number denoted by a string[9]: Using either of numeral sets *introduced* in 1.0.4.0, there are **different schemes, i.e., sets of rules for interpreting a string as a number**.

For example, the string '4723', according to *usual decimal system*, represents the number: $4*10^3 + 7*10^2 + 2*10^1 + 3*10^0$. Also, the string 'CLVII', according to the *Roman system,* represents the (decimal) number: 100 + 50 + 5 + 1 +1 = 157 (decimal), where C denotes 100, L denotes 50, V denotes 5 and I denotes 1. Similarly, the *binary string* 10010110 may be interpreted as the number (with value in decimal): $1* 2^7 + 0* 2^6 + 0* 2^5 + 1* 2^4 + 0* 2^3 + 1* 2^2 + 1* 2^1 + 0* 2^0$

A.4.3 Notation for Base/ Radix in Number Representation: From a string of digits, by itself, it may not be clear whether it is string of binary, octal, decimal, or hexadecimal digits. For example, the string 10010110 may be equally considered as a string of binary, octal, decimal, or hexadecimal digits. Similarly, the string 4607542 may be equally considered as a string of octal, decimal, or hexadecimal digits.

Thus the same string **10010110 may determine the decimal number**

- $1* 2^7 + 0* 2^6 + 0* 2^5 + 1* 2^4 + 0* 2^3 + 1* 2^2 + 1* 2^1 + 0* 2^0$,
  if the string is assumed to be **binary,**
- $1* 10^7 + 0* 10^6 + 0* 10^5 + 1* 10^4 + 0* 10^3 + 1* 10^2 + 1* 10^1 + 0* 2^0$,
  if the string is assumed to be **decimal** etc.[10].

Therefore, in order to avoid confusion about which numeral set a string belongs to, a suffix in the following manner is used

- $(string)_2$ for binary, e.g., string 10010110, if treated as binary will be denoted as $(10010110)_2$
- $(string)_{10}$ for decimal, e.g., string 10010110, if treated as decimal will be denoted as $(10010110)_{10}$ etc[11]

---

[9] For finding the value of a string of digits, at the top level, the numeral systems may be divided into two classes:

- ***positional numeral systems and***
- ***non-positional numeral systems.***

Most of the numeral systems are positional. **In the positional system,** the position of a digit determines the value contributed to the number by the digit. For example, in the decimal numeral representation 4723, the digit 4 , because of its position, contributes 4000, the digit 7 contributes 700 and so on.

The Roman numeral system is a well-known non-positional numeral system. In a non-positional numeral system, the contribution of a digit to the value of the number does not depend on its position in the string of the digits. For example, in the Roman numeral, I always contributes 1, V contributes 5, X contributes 10, L contributes 50 etc. to the value of the number represented. Thus, LXI, represents 61, with L contributing 50, X contributing 10 and I contributing 1. **However, even the set of Roman numerals is not pure non-positional system:** If x and y are two Roman digits such that the digit x represents a number less than a number represented by the digit y, then the numbers represented by the strings xy and the strings yx are different. We know IX represents (decimal) 9 and XI represents (decimal) 11.

[10] The same string **10010110 may determine the decimal number**

$1* 8^7 + 0* 8^6 + 0* 8^5 + 1* 8^4 + 0* 8^3 + 1* 8^2 + 1* 8^1 + 0* 8^0$,
if the string is assumed to be **octal**, and as
$1* 16^7 + 0* 16^6 + 0* 16^5 + 1* 16^4 + 0* 16^3 + 1* 16^2 + 1* 16^1 + 0* 16^0$,
if the string is assumed to be **hexadecimal**.

However, if there is no possibility of confusion, the suffix is not used.

A.4.4 A set of numerals along with a scheme for interpreting a sequence of digits as number, is called a ***numeral system (or slightly incorrectly, a number system).*** You may notice that the decimal system and Roman system use different ways/ schemes for getting a number from a string of digits.

It may be noted that even for the same set of numerals, there may be different schemes/ rules of interpretation, with different schemes giving different values for a given string. For example, for interpreting a binary string as a number, there are two well-known schemes:

- Fixed-point and
- Floating-point.

These schemes will be discussed later in detail.

The number systems, that we will discuss, are positional number systems based **on only decimal and binary. However,** with minor modifications, the discussion can be generalized to r-radix set of numerals. These numeral systems will be called, though slightly incorrectly, as number systems.

A.5 ESSENTIAL FEATURES OF COMPUTER-REPRESENTED NUMBERS

As mentioned earlier, not all real numbers can be represented in a computer system. The numbers that can be represented in a computer system will be called ***computer numbers,*** computer-represented numbers or sometimes, as ***computer represent-able numbers*** also.

Here, we mention some of the essential features of these numbers, specially, with respect to and in comparison with the properties of the number systems discussed in the subsection above.

1. A computer number is necessarily a binary number, i.e., it is a (finite) string of only 0's and 1's. (in this case, 0 and 1 are called ***bits***). However, a particular string of bits may represent different numbers according to different schemes, i.e., sets of rules for interpreting a string as a number. *(to be discussed in more detail later)*

2. There is no unique set of computer represent-able numbers. The numbers that can be represented in a computer depend on the computer system under consideration….The numbers that can be represented in a computer system depend on the word size of the computer system and the scheme of representation used.

[11] To indicate numeral system used for a given string, a suffix in the following manner is used: $(string)_8$ for octal, e.g., string 10010110, if treated as octal will be denoted as $(10010110)_8$ …… and as ….$(string)_{16}$ for hexadecimal, e.g., string 10010110, if treated as hexadecimal will be denoted as $(10010110)_{16}$

3. No real number, which is irrational number, can be represented in any computer system…. Only finitely many real numbers, each of which must also be rational, can be computer represent-able[12].

4. Each of the other real numbers, when required to be stored in a computer, is approximated appropriately, to a computer number (of the computer system)

5. Whatever may be the computer system under consideration, the number of computer represent-able numbers, though substantially very large, is *finite* only. Therefore, *not* even all natural numbers (and, hence, all integers, all rational numbers and all real numbers) can be represented in a computer system.

6. Computer represent-able numbers have minimum positive computer number…. However, each computer system has its own unique minimum positive computer represent-able number. This number is generally called ***machine-epsilon*** of the computer system.

7. **Computer number zero is not the same as real number Zero**: If x is any real number such that |x|, if after rounding, is less than machine epsilon, say, ε, then x is represented by zero. Thus, computer zero represents not a single real number 0, but all the infinitely many real numbers of an interval contained in $]-\varepsilon, \varepsilon[$, and as ε, and hence, the computer number zero also varies, from computer to computer.

8. Each computer system has its own unique maximum *positive* computer represent-able number. The number depends on the word size and the scheme of representation used.

9. We elaborate further, the statement under point 1 above: *A computer number is necessarily a binary number, i.e., it is a string of only 0's and 1's. However, a particular string of bits may represent different numbers according to different schemes, i.e., sets of rules for interpreting a string as a number.*

The schemes for interpreting a string as a number, at the top may be categorized into two major classes: (i) Fixed point representation and (ii) Floating point representation schemes.

(The Floating point representation scheme has already been discussed in detail in Unit 1)

Further, ***the Fixed point representation class has a number of schemes including***: (a) binary (b) BCD (Binary Coded Decimal) (c) Excess-3 (d) Gray code (e) signed magnitude, (f) signed 1's complement and (g) signed 2's complement representation schemes, and some combinations of these.

Similarly, ***the Floating point representation scheme may associate different numbers to a particular string of bits, according to*** (i) how the string is considered as composed of two parts, viz., mantissa and exponent, and (ii) the choice of the base and the choice of the bias or characteristic.

[12] But, as mentioned above, even not all rational numbers are computer represent-able. For example, 1/3 is a rational number, which can not be represented as a finite binary string, and hence, is not a computer number. Further, it may be noted that some rational numbers which can be represented as a finite string of *decimal* digits, may not be written as a finite string of bits. For example, 1/5 can be written as: 0.2, a finite decimal string, but can be written only as an infinite *binary* string: 0.00110011……

A.6 Disasters due to Numerical Errors[13]

1. Patriot Missile Failure

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Saudi Arabia, failed to intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks and killed 28 soldiers. A report of the General Accounting office, GAO/IMTEC-92-26, entitled *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia* reported on the cause of the failure. It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors. Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds. This calculation was performed using a 24 bit fixed point register. In particular, the value 1/10, which has a non-terminating binary expansion, was chopped at 24 bits after the radix point. The small chopping error, when multiplied by the large number giving the time in tenths of a second, lead to a significant error

In other words, the binary expansion of 1/10 is .0001100110011001100110011001100.... Now the 24 bit register in the Patriot stored instead 0.00011001100110011001100 introducing an error of 0.00000000000000000000000011001100... binary, or about 0.000000095 decimal. Multiplying by the number of tenths of a second in 100 hours gives 0.000000095×100×60×60×10=0.34.) A Scud travels at about 1,676 meters per second, and so travels more than half a kilometer in this time. This was far enough that the incoming Scud was outside the "range gate" that the Patriot tracked. Ironically, the fact that the bad time calculation had been improved in some parts of the code, but not all, contributed to the problem, since it meant that the inaccuracies did not cancel.

2. Explosion of the Ariane 5

On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after lift-off. The rocket was on its first voyage, after a decade of development costing $7 billion. The destroyed rocket and its cargo were valued at $500 million. A board of inquiry investigated the causes of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system. Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer. The number was larger than 32,768, the largest integer storeable in a 16 bit signed integer, and thus the conversion failed.

3. Rounding error changes Parliament makeup

We experienced a shattering computer error during a German election this past Sunday (5 April). The elections to the parliament for the state of Schleswig-Holstein were affected. German elections are quite complicated to calculate. First, there is the 5% clause: no party with less than 5% of the vote may be seated in parliament. All the votes for this party are lost. Seats are distributed by direct vote and by list. All persons winning a precinct vote (i.e. having more votes than any other candidate in the precinct) are seated. Then a complicated system (often D'Hondt, now they have newer systems) is invoked that seats persons from the

[13] The instances have been taken on Aug, 22, 2013 from the site:
http://ta.twi.tude.nl/users/vuik/wi211/disasters.html

party lists according to the proportion of the votes for each party. Often quite a number of extra seats (and office space and salaries) are necessary so that the seat distribution reflects the vote percentages each party got.

On Sunday the votes were being counted, and it looked like the Green party was hanging on by their teeth to a vote percentage of exactly 5%. This meant that the Social Democrats (SPD) could not have anyone from their list seated, which was most unfortunate, as the candidate for minister president was number one on the list, and the SPD won all precincts: no extra seats needed.

After midnight (and after the election results were published) someone discovered that the Greens actually only had 4.97% of the vote. The program that prints out the percentages only uses one place after the decimal, and had rounded the count up to 5%! This software had been used for years , and no one had thought to turn off the rounding at this very critical (and IMHO very undemocratic) region!

So 4.97% of the votes were thrown away, the seats were recalculated, the SPD got to seat one person from the list, and now have a one seat majority in the parliament. And the newspapers are clucking about the "computers" making such a mistake. ***Reported by Debora Weber-Wulff, 7 Apr 1992***

4. The sinking of the Sleipner A offshore platform

The Sleipner A platform produces oil and gas in the North Sea and is supported on the seabed at a water depth of 82 m. It is a Condeep type platform with a concrete gravity base structure consisting of 24 cells and with a total base area of 16 000 m^2. The concrete base structure for Sleipner A sprang a leak and sank under a controlled ballasting operation during preparation for deck mating in Gandsfjorden outside Stavanger, Norway on 23 August 1991.

A committee for investigation into the accident was constituted. The conclusion of the investigation was that the loss was caused by a failure in a cell wall, resulting in a serious crack and a leakage that the pumps were not able to cope with. Etc.

. When the first model sank in August 1991, the crash caused a seismic event registering 3.0 on the Richter scale, and left nothing but a pile of debris at 220m of depth. The failure involved a total economic loss of about $700 million.

The 24 cells and 4 shafts referred to above are shown to the left while at the sea surface. The cells are 12m in diameter. The cell wall failure was traced to a tricell, a triangular concrete frame placed where the cells meet. At right one is pictured undergoing failure testing.

The post accident investigation traced the error to inaccurate finite element approximation of the linear elastic model of the tricell (using the popular finite element program NASTRAN). The shear stresses were underestimated by 47%, leading to insufficient design. In particular, certain concrete walls were not thick enough. More careful finite element analysis, made after the accident, predicted that failure would occur with this design at a depth of 62m, which matches well with the actual occurrence at 65m. This description is adapted from *The sinking of the Sleipner A offshore platform by Douglas N. Arnold.*

UNIT 0: OVERVIEW OF MACRO-ISSUES IN 'NUMERICAL ANALYSIS & TECHNIQUES'

The unit is optional, though highly desirable, reading. It is included for a quick run-through for the first time, and for later references, from time to time, for better understanding of the subject-matter

0.0: Introduction

Whenever a new academic discipline is intended to be pursued, some questions, about the discipline, including the following ones arise naturally: Why, at all, we should study the discipline?, What is its domain of study or subject-matter of the discipline? , What are its distinct features, special tools and techniques?. Also, we will like to know the opinions of the experts in the field in respect of major issues, including such questions. In this Unit, we just briefly discuss, rather only enumerate, the questions along with opinions of some experts in the field.

0.1: WHY (COMPUTER-ORIENTED) NUMERICAL TECHNIQUES?

A mathematician knows how to solve a problem— but he can't do it
W.E. Milne [1]

The reason is that a mathematician, being a mathematician, uses all sorts of mathematical assets including mathematical concepts, notations, techniques, and at the top of all these, mathematical thinking, intuition, reasoning and habit in solving problems. Doing so, though, may be quite useful in

[1] **Page 1 of Introduction to Numerical Analysis (Second Edition) by Carl-Erik Fr□berg (Addison Wesley, 1981)**

solving many problems, yet may be quite problematic unless done carefully, while using computer as a tool for solving problems.

Because of our habitual (*mathematical*) thinking we use, without second thoughts, many mathematical identities, including the following ones

$$a+ (b + c) = (a + b) + c$$
$$a/ (b * c) = (a/b)/c, \text{ and}$$
$$(1 + a) / 2 = 1/ 2 + a/ 2.$$

However, neither of these may be correct in many cases of *numerical* computation, and use of these identities blindly, may lead to completely erroneous results.

Also, many a time, we know a ***mathematical*** solution to the problem, but the solution can not be useful because of various practical reasons, including the fact that the mathematical solution may involve infinitely many computational steps. For example, there is a mathematical solution to the problem of finding value of e^x, for some given value of x, by using the formula

$$1+ x + x^2/ 2! + x^3/ 3! + x^4/ 4! + \cdots\cdots,$$

which converges absolutely to the function e^x for any value of x. However, apart from its being an infinite process, which is not realizable on a computer system; otherwise also, it may not help us in obtaining a *numerical* solution. The reason being that the use of the series to evaluate e^{-100} would be completely impractical, because of the fact that we would need to compute about 100 terms of the series before the size of the terms would begin to decrease. (The example is from Page 1 of *A Survey of Numerical Mathematics Vol. 1 by Young & Gregory (Addision & Wesley,1972))*

These two examples amply illustrate that a different framework of mind and, at least, some different set of techniques (to be called ***numerical techniques****) are required to successfully solve problems numerically.*

On closer examination, these problems arise because of the fact that Mathematics (especially as it is currently taught) and numerical analysis differ from each other more than is usually realized. The most obvious differences are that mathematics regularly uses the infinite both for representation and for processes, whereas computing is necessarily done on a finite machine in a finite time[2]. *This fact of difference is repeatedly emphasized in these lecture notes, and also, in every numerical methods' book.*

In this context, it is not irrelevant to mention the too obvious fact that computer has become an indispensible tool to solve mathematical problems, specially, because of its fast speed, iterative capability and the capability to represent very large to very small quantities, much more precisely than human beings can do using pen and paper etc.

However, as mentioned earlier, computer is a finite machine— a machine having pre-assigned *machine-specific* finite space (1, 2 or 4 etc., number of words in memory) for representing quantities and finite time to accomplish a task (what will be the utility of a solution, if it is delivered after infinite time, i.e., after eternity). And, as clarified earlier through the two examples, it has to be used with utmost care, particularly while using it for solving numerical problems based on mathematical results or solutions. While adapting a mathematical solution for execution on a computer, we have to be *perennially aware*

[2] **Page 2 Numerical Methods for Scientists and Engineers (First Edition) by R.W. Hamming (McGraw-Hill, 1962)**

that each (specific) computer requires the computer-specific adapting of a mathematical solution. Forgetting these facts, even momentarily, has lead to a number of disasters, due to numerical errors, including the following ones: *Patriot Missile Failure, Explosion of the Ariane 5, EURO page: Conversion Arithmetics, The Vancouver Stock Exchange, Rounding error changes Parliament makeup, The sinking of the Sleipner An offshore platform, Tacoma bridge failure (wrong design), 200 million dollar typing error (typing error) and What's 77.1 x 850? Don't ask Excel 2007* [3].

Computer-oriented numerical techniques, among other matters, help us in adapting appropriately mathematical solutions for execution on computer—rather help us in specific adapting for each (specific) computer and, hence help us in avoiding many potential disasters.

0.2: WHAT ARE NUMERICAL TECHNIQUES?

The purpose of numerical analysis is 'insight, not numbers'
R.W. Hamming in [4]

(Though numerical techniques have been used for hundreds of years, yet the advent of computer has enhanced many-folds the frequency of use and utility of numerical techniques in solving mathematical problems. ***Hence, by a 'numerical technique', we will invariably mean 'Computer oriented numerical technique')***

The explanation in previous section, also gives an idea of the need to know the differences between, on one hand, a mathematical approach/ technique, in general and, on the other hand, a numerical approach/ technique for solving mathematical problems. In this respect, the following points may be noted:

1. **Numerical Techniques** are about designing algorithms or constructive methods for solving *mathematical* problems, which (i.e., algorithms)
   - use only *computer-represent-able numbers (to be defined later, and to be called only* ***computer numbers****)* for representing data/ information and
   - use only the ***numerical operations***, i.e., plus, minus, multiplication and division, on the computer numbers, for transforming data/ information….*Even a simple operation like square-rooting ($\sqrt{}$) is not a numeric operation*, and has to be realized through some algorithm, which can use only the above-mentioned (numerical) operations.

2. **Rounding:** There are only finitely many computer numbers, and the number of real numbers, is infinite. Therefore, not all real numbers (even, not all natural numbers) can be expressed as computer numbers. Those real numbers (and complex numbers, expressed as pair of real numbers), which are not computer numbers, have to be approximated and represented in the computer, by computer numbers. The process is called ***rounding***, and induces ***rounding error***.

3. Further, numerical operations when applied to *computer numbers* in the usual arithmetic sense, may result in a *real number* that may not be a *computer number*. Such a *real* number has, again, to be appropriately approximated to a computer number through *'rounding'*.

---

[3] In order to emphasize the point, discussion of some of the disasters is included in the Appendix of the block.

[4] **Page 3 Numerical Methods for Scientists and Engineers (Second Edition) by R.W. Hamming (McGraw-Hill, 1973)**

4. **Truncation:** An algorithm or a constructive method, by definition, is *finite*. Thus, any infinite process/ method, including the one mentioned above in respect of e^x, is not constructive or algorithmic……. An infinite mathematical process, if required to be used, has to be replaced or approximated by some appropriate finite process. The process of approximation, is called *truncation*, and induces *truncation error.*

5. An *analytic* function is a function which, directly or indirectly, involves the concept of limit.
The set of analytical functions include: all trigonometric functions like Sin x *(rather, only sin, though conventionally written informally as sin(x)*), log (x), e^x , d/ dx (i.e., derivative), and $\int f(x)\,dx$ (i.e., integration).

The evaluation of an analytical function, in general, is an infinite process. As mentioned above, for some value of x, e^x may be evaluated by using the (infinite) formula: $e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + \cdots$. Evaluation of an analytical function using such a formula is called *analytical solution*, which not being finite is *not* a numerical solution.

6. **Iteration** is an important (numerical) technique for reformulating and/ or solving many mathematical problems into numerical/ computational problems, specially,
   (i) when a mathematical problem does not have an algorithmic/ computational solution ,e.g., the problem of finding roots of a general polynomial equation of degree five or more or
   (ii) when a mathematical problem involves infinity
      (a) directly, in the form of an infinite series/ process, as in finding the value of e^x, for some value of x, using the formula mentioned above, or
      (b) indirectly, to approximate irrational numbers and other non-computer numbers, which may occur as either final answer or during the solution process. Thus, iteration may be used in finding better approximation of the root of the equation $x^2 = 2$, after starting with some reasonably appropriate initial guess.

Iteration/ iterative method (as opposed to direct method) is an important numerical technique. … specially useful when no direct method may be available. **(P.13/ Young & Gregory)**

7. **Examining & mathematically analyzing** a problem before, during and after attempting a solution, and, if required, **mathematically reformulating** the problem at any stage, in order to get a better, if not perfect, solution of the problem under consideration, *are mathematical techniques.*
   For example, we may *first* attempt to evaluate **f(x) = tan x – sin x** at, say x = 0.1250, *in the usual way*, by evaluating each of tan (0.125) and sin (0.1250), and then subtracting the latter from the former to get the result. However, on analysis, it is found that, if we first use the trigonometric expansion of tan x and sin x as follows:
   $\tan x = x + (1/3)x^3 + (2/15)x^5 + (17/315)x^7 + \cdots$ and
   $\sin x = x - (1/6)x^3 + (1/120)x^5 - (1/5040)x^7 + \cdots$,
   and reformulate the function as
   $f(x) = (1/3)x^3 + (1/8)x^5 + (13/240)x^7 + \cdots$,
   then with this reformulation, a better approximation of f(x) is obtained.

8. **Discretization**: is a specialized technique of mathematically analyzing and reformulating the problem, in which the reformulation is restricted to replacement of *continuous type mathematical*

concepts by *(numerically) computable objects*. The replacement may be made in the beginning itself and then **only** the reformulated problem is attempted to be solved. The most well-known examples of discretization are in respect of replacement of (mathematical continuous concepts of) integral and differential. Using Trapezoidal rule:

$$\int_a^b f(x) \approx \sum_{k=0}^{n-1} (1/2)\, h\, [\, f(x_k) + f(x_{k-1}) \,],$$

the mathematical continuous concept of integral on the left is replaced by the computable object of finite sum on the right . Similarly, the mathematical continuous concept of derivative $y'(x_k)$ is replaced by the computable object, viz., divided difference : $(y_{k+1} - y_k) / (y_{k+1} - y_k)$.

Discretization , being an approximation of *mathematical continuous concepts*, also introduces error, which we call *approximation/ discretization error* and this is another type of error, different from truncation and round-off .

The above-mentioned techniques, particularly, truncation, iteration and disretization , are not necessarily distinct., and, may be overlapping.

9. **Not every mathematical technique is necessarily a numeric technique.** One of the non-constructive (or non-algorithmic) mathematical techniques of proof is proof-by-contradiction. However, the mathematical technique **proof-by-contradiction, not being constructive, is not a numerical technique.**

Some Remarks:

Remark 1. At this stage, we may note the **difference between** a numerical **technique and** a numerical **solution** of a problem. **A numerical solution** is an algorithm that involves only computer numbers and four elementary arithmetic operations**.**

On the other hand, **a numerical technique** may, if required, use general mathematical knowledge, tools and techniques which **may help** in solving a problem numerically. A technique, may help in solving a problem by mathematically analyzing the problem and then reformulating the problem into other problems, which may be numerically solvable. Thus, ***the techniques: Truncation, Iteration and Discretization, are numerical techniques, which use mathematical tool and techniques, for appropriate numerical actions.***

In solving problems numerically, of course, the *mechanical power* of the computer is an indispensible tool in executing the algorithm. But the power of computer comes in to play only when an appropriate algorithm is already designed.
However, for designing an appropriate algorithm, the choice of appropriate numerical techniques is required. As per state of art in solving problems numerically, the choice of appropriate techniques is not a mechanical task, i.e., there is no systematic method for choosing appropriate techniques, and requires (human) intelligence and practice. With practice, the process of making appropriate choices gets refined leading to *insight*, which is what **R.W. Hamming has emphasized above**.

Remark 2. From the description of numerical techniques in *1.of What are numerical techniques?*, it may be concluded that discipline of (Computer-Oriented) Numerical Techniques is a specialized sub-discipline of Design & Analysis of Algorithm, in which

(i) data/ information is represented only in the form of computer numbers,

(ii) the operations for information transformation are only the four elementary Arithmetic operations, and

(iii) mathematical problems may be reformulated into some numerical problems, using some numerical approximation techniques.

0.3 Numbers: Sets, systems & Notations

(This subsection is a summary of parts of Appendix in this Block. For more detailed discussion on this topic, refer to the Appendix)

We have earlier mentioned that the discipline of Numerical Techniques is about

- numbers, rather special type of numbers called computer numbers, and
- application of (some restricted version of) the four arithmetic operations, viz., + (plus), (minus), * (multiplication) and ÷ (division) on these special numbers.

Therefore, let us, first, recall some important sets of numbers, which have been introduced to us earlier, some of these even from school days. Then we will discuss computer numbers vis-à-vis these numbers

0.3.1 Sets of Numbers

Set of **Natural numbers** denoted by **N**, where N = {0, 1, 2, 3, 4, ... } or N = { 1, 2, 3, 4, ... }
Set of **Integers** denoted by **I, or Z,** where I (or Z) = { …., −4, −3, −2, −1, 0, 1, 2, 3, 4, …..}
Set of **Rational Numbers** denoted by **Q,** where
Q = {a/b, where *a* and *b* are integers and *b* is not 0 }

Set of **Real Numbers** denoted by **R**. ….. There are different ways of looking at or thinking of Real Numbers. ***One of the intuitive ways of thinking of real numbers*** is as the numbers that correspond to the points on a straight line extended infinitely in both the directions, such that one of the points on the line is marked as 0 and another point (different from, and to the right of, the earlier point) is marked as 1. Then to each of the points on this line, a unique real number is associated.

Set of **Complex Numbers** denoted by **C**, where C = {$a + bi$ or $a + ib$ where *a* and *b* are real numbers and *i* is the square root of −1}

By minor notational modifications (e.g., by writing an integer, say, 4 as a rational number 4/1; and by writing a real number, say $\sqrt{2}$ as a complex number $\sqrt{2} + 0\,i$), we can easily see that **N ⊂ I ⊂ Q ⊂ R ⊂ C.**

When we do not have any specific set under consideration, the set may be referred to as a ***set of numbers***, and a member of the set as just ***number.***

Apart from these well-known sets of numbers, there are sets of numbers that may be useful in our later discussion. Next, we discuss two such sets.

Set of algebraic Numbers *(no standard notation for the set)*, where an *Algebraic number* is a number that is a root of a non-zero polynomial equation[5] with rational numbers as coefficients. For example,

- Every rational numbers is algebraic (e.g., the rational number a/b, with $b \neq 0$, is a root of the polynomial equation: $bx - a = 0$). Thus, a real number, which is not algebraic, must be irrational number.
- Even, some irrational numbers are algebraic, e.g., $\sqrt{2}$ is an algebraic number, because, it satisfies the polynomial equation: $x^2 - 2 = 0$. In general, n^{th} root of a rational number a/b, with $b \neq 0$, is algebraic, because, it is a root of the polynomial equation: $b*x^n - a = 0$
- Even, a complex number may be an algebraic number, as each of the complex numbers $\sqrt{2}$ i (= $0+\sqrt{2}$ i) and $-\sqrt{2}$ i is algebraic, because, each satisfies the polynomial equation: $x^2 + 2 = 0$.

Set of Transcendental Numbers *(again, no standard notation for the set),* where, a *transcendental number* is a complex number (and, hence, also, a real number, as, $R \subset C$), which is not algebraic. From the above examples, it is clear that a rational number can not be transcendental, and some, but not all, irrational numbers, may be transcendental. The most prominent examples of transcendental numbers are $\boldsymbol{\pi}$ and ***e***. [6]

0.3.2 Algebraic Systems of Numbers (To be called, simply, ***Systems of Numbers***)

In order to discuss, system of numbers, to begin with, we need to understand the concept of ***operation on a set*** For this purpose, recall that N, the set of Natural numbers, **is closed under** '+' (plus). By ***'N is closed under +'*** ,we mean: if we take (any) two natural numbers, say m and n, then m + n is *also* a natural number.

But, N, the set of Natural numbers, is not closed under '–' (minus). In other words, *for some* natural numbers m and n, m – n may not be a natural number, for example, for 3 and 5, 3 – 5 is not a natural number.(Of course, 3 – 5 = – 2 is an integer)
These facts are also stated by saying: '+' is a binary operation on N, but, '–' is not a binary operation on N. Here, the word ***binary*** means that in order to apply '+', we need (exactly) two members of N.

In the light of above illustration of binary operation, we may recall many such statements including the following:

(i) * (multiplication) is a binary operation on N (or, equivalently, we can say that N is closed under the binary operation *)

(ii) – (minus) is a binary operation on I (or, equivalently, we can say that I is closed under the binary operation –) etc**.**

However, there are operations on numbers, which may require only one number ***(the number is called argument of the operation)*** of the set. For example, The ***square*** operation on a set of numbers takes only one number and returns its square, for example, square(3) = 9.

---

[5] We may recall that a polynomial P(x) is an expression of the form: $a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots\ldots + a_{n-1}x + a_n$, where a_i is a number and x is a variable. Then, P(x) = 0 represents a polynomial equation.

[6] It should be noted that it is quite complex task to show a number as transcendental. In order to show a number, say, n, to be transcendental, theoretically, it is required to ensure that for *each* polynomial equation P(x) = 0, n is not a root of the equation. And, there are infinitely many polynomial equations. This direct method for showing a number as transcendental, can not be used. There are other methods for the purpose.

Thus, some operations (e.g., square) on a set of numbers may take only one argument. Such operations are called ***unary*** *operations.* Other operations may take two arguments (e.g., +, – , *, ÷) from a set of numbers. Such operations are called ***binary*** *operations.* There are operations which may take three arguments and are called ***ternary operations***. Even some operations may take zero number of arguments.

Definition: Algebraic System of Numbers: A set of numbers, say, S, along with a (finite) set of operations on S, is called an algebraic system of numbers. In stead of 'algebraic system', we may use the word **'system'**.

Notation for a System: If O_1 , O_2 ,….. O_n are some n operations on a set S, then, we denote the corresponding system ***as*** $\langle S, O_1, O_2, \ldots O_n \rangle$, ***or as*** $(S, O_1, O_2, \ldots O_n)$.

Examples & Non-examples of Systems of Numbers:

1. **Examples of number systems**
   Each of following is a system of numbers: < N, + >, < N, * >, and < N, +, * > etc**.**

2. Non-examples of number systems: Each of following is NOT a system of numbers: < N, − >, < N, ÷ >, < N, − , ÷ >, and < I, ÷ > etc.

Remark : In the above discussion, the use of the word **number** is inaccurate. Actually, a number is a concept (a mental entity), which may be represented in some (physical) forms, so that we can experience the concept through our senses. The number, the name of which is, say, **ten i**n English and **nl** in Hindi, and zehn in German language may be represented as 10 as decimal numeral , X as Roman numeral, 1010 as binary numeral. As, you may have already noticed, the physical representation of a number is called its **numeral**. Thus, number and numeral are two different entities, incorrectly taken to be the same. Also, a particular number is unique, but, it can have many (physical) representations, each being called a numeral, corresponding to the number.

The difference between **number** and **numeral** may be further clarified from the following explanation: We have the **concept** of the animal **that is called** COW in English, **xk;** in Hindi and KUH in German language. The animal, represented as cow in English, has four legs; however, its representation in English: **cow,** is a word in English and has three letters, but does not have four legs.

However, due to usage, though inaccurate, over centuries, in stead of the word **numeral,** almost, the word **numbe**r is used. Except for the discussion in the following subsection, we will also not differentiate between Number and Numeral.

0.3.3 Numerals: Notations for Numbers

First, we recall some well-known sets used to denote numbers. These sets are called *sets of numerals* and then discuss various *numeral systems,* developed on these numeral sets, for representing numbers.

0.3.3.1 Sets of Numerals: We are already familiar with some of the sets of numerals. The most familiar, and frequently used, set is **Decimal** *Numeral Set*. It is called *Decimal,* because, it uses ten figures, or digits*, viz., digits from the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} of ten digits.*

Another numeral set, familiar to computer science students, is **binary** numeral set. It is called binary, because, it uses two figures, or digits, *viz., digits from the set {0, 1} of two digits*. In this case, 0 and 1 are called ***bits***.

Also, ***Roman Numeral Set,*** is well-known. This set uses figures/ digits/ letters from the set { I, V, X, L, C, D, M, …}, where, I represents 1 (of decimal numeral system), V represents 5, X represents 10, L represents 50, C represents 100, D represents 500 and M represents 1000. etc.[7]

Apart from these sets of numerals, in context of computer systems, we also come across

i. **Hexadecimal numeral set,** which uses figures/ digits, viz., from the set: *{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E} of sixteen digits.*
ii. **Octal numeral set,** which uses figures/ digits, viz., from the set: *{0, 1, 2, 3, 4, 5, 6, 7} of eight digits.*

0.3.3..2 Number representation using a set of numerals: a number is represented by a *string of digits* from the set of numerals under consideration, e.g., 3426 in decimal, IX in Roman 10010110 in binary and 37A08 in hexadecimal.

0.3.3..3 Value of the number denoted by a string: Using either of numeral sets *introduced* in 0.3.3.1 above, there are **different schemes, i.e., sets of rules for interpreting a string as a number**.

For example, the string '4723', according to *usual decimal system*, represents the number: $4*10^3 + 7*10^2 + 2*10^1 + 3*10^0$. Also, the string 'CLVII', according to the *Roman system,* represents the (decimal) number: 100 + 50 + 5 + 1 +1 = 157 (decimal), where C denotes 100, L denotes 50, V denotes 5 and I denotes 1. Similarly, the *binary string* 10010110 may be interpreted as the number (with value in decimal): $1* 2^7 + 0* 2^6 + 0* 2^5 + 1* 2^4 + 0* 2^3 + 1* 2^2 + 1* 2^1 + 0* 2^0$

A computer number is necessarily a binary number, i.e., it is a string of only 0's and 1's. However, a particular string of bits may represent different numbers according to different schemes, i.e., sets of rules for interpreting a string as a number.

The schemes for interpreting a string as a number, at the top may be categorized into two major classes: (i) Fixed point representation and (ii) Floating point representation schemes.

(The Floating point representation scheme will be discussed in detail in the next unit)

Further, *the Fixed point representation class* has a number of schemes including: (a) binary (b) BCD (Binary Coded Decimal) (c) Excess-3 (d) Gray code (e) signed magnitude, (f) signed 1's complement and (g) signed 2's complement representation schemes, and some combinations of these.

---

[7]Appropriate choice of numeral system has significant role in solving problems, particularly solving problems *efficiently*. For example, it is a child's play to get the answer for 46* 37 (in decimal numeral system) as a single number. However, using Roman numerals, i.e., writing VLI * XXXVII, in stead of , 46 * 37, it is really very difficult to get the same number, using only Roman numerals, as answer.

Similarly, *the Floating point representation scheme* may associate different numbers to a particular string of bits, according to (i) how the string is considered as composed of two parts, viz., mantissa and exponent, and (ii) the choice of the base and the choice of the bias or characteristic etc.

PLEASE TAKE CARE SO THAT THE TABLE ON NEXT PAGE IS NOT DISTURBED

0.3.4: Table of Contrasting & Other Properties of Conventional Number Systems and Computer Number Systems

Properties of Conventional Number Systems:	**Properties of computer numbers**
1. Each of the set N, I, Q, R and C is an *infinite* set.	**1.** The number of computer represent-able numbers, is *finite* only, though substantially large. Therefore, *not* even all natural numbers (and, hence, all integers, all rational numbers and all real numbers) can be represented in a computer system.
2. Each of these sets/ systems of numbers is unique, independent of representation scheme	*2.* The set of computer numbers is not unique. Each computer system has its unique set of computer-represent-able numbers, which may be different from those of another computer system. The numbers that can be represented in a computer system depend on the word size of the computer system and the scheme of representation used.
3. None of the number systems mentioned above, is bounded above, i.e., has the *maximum element*.	*3.* For each computer system, the set of computer numbers has the *maximum element*, i.e., the set of computer numbers is bounded above. However, the maximum element is, again, computer dependent.
4. Except N, the set of natural numbers, none of the other number systems I, Q, R has the *minimum* element.	*4.* For each computer system, the set of computer numbers has the *minimum element*, i.e., the set is bounded below. However, the minimum element is, again, computer dependent. The number is close to the number $-(max)$, *where max* denotes maximum computer represent-able number, and the minimum depends on scheme of representation.
5. The set of real (or rational) numbers does not have the *least positive real number*. Because, between 0 and any positive real/ rational number, say r, lies the positive real/ rational number r/2.	5. Computer represent-able numbers *have minimum positive computer number*. However, each computer system has its own unique minimum positive computer represent-able number. This number is generally called ***machine-epsilon*** of the computer system.
6. The number 0 is a single number in conventional number systems	**6. computer number zero is not the same as real number Zero**: If x is any real number such that |x|, if after rounding, is less than machine epsilon, say, ε, then x is represented by zero. Thus, computer zero represents *not a single real number 0*, but all the infinitely many real numbers of an interval contained in $]-\varepsilon, \varepsilon[$, and ε ***varies from computer to computer***

Properties of Conventional Number Systems continued:

7. If r is a root of an an equation f(x) = 0, in mathematical sense, then it is necessary that f(r) = 0.

8.
   a. Each of the Number systems: N, I, Q, R, and C **is closed** under + and *, i.e., for two numbers a and b in any one of these sets, then the sum a + b and product a * b are also in the same set
   b. Each of the Number systems: I, Q, R, and C is closed under − , i.e., for two numbers a and b in any one of these sets, then the difference a − b is also in the same set
   c. Each of the Number systems: R~{0} and C ~{0}is closed under ÷ (division) , i.e., for two numbers a and b in any one of these sets, then the quotient a ÷ b is also in the same set

9. For each of the number systems: N, I, Q , R , and C, the following hold
   a. '+' is Commutative in numbers, i.e., x + y = y + x, for any numbers x and y
   b. '*' is Commutative in numbers, i.e., x * y = y * x, for any numbers x and y

10. For each of the number systems: N, I, Q , R , and C, the following hold
   a. '+' is Associative in numbers, i.e., (x + y) + z = x + (y + z), for any numbers x, y and z
   b. '*' is Associative in numbers, i.e., (x * y) * z = x * (y * z), for any numbers x, y and z,

11. The conventional number systems are mainly for human understanding and comprehension of number size. *Mainly decimal number system* have been used to represent numbers, because of decimal system being in use over a long period and its capability to represent numbers in a uniform, easy to understand manner.

Properties of computer numbers continued:

7. in view of the above statement at 6., for a *computed real root*, say r, of an equation f(x) = 0, it does not necessarily mean f(r) = 0. It may only mean that f(r), as a real number, lies in the interval] −ε, ε [

8.
   a. **The set of computer numbers is not closed under each of the four numerical operations**. For example, if M denotes the maximum represent-table number in a computer system, then M + M and M * M are NOT computer represent-able numbers

*Thus, set of computer numbers is NOT closed under + (sum) and * (product)*

Also, then M − (− 1) = M + 1 is NOT computer represent-able number.

Thus, set of computer numbers is NOT closed under − (difference)

Also, each of the numbers 1 and 3 is computer represent-able number, but, 1÷ 3 = 1/3 is not a computer number.

Thus, set of computer numbers is NOT closed under ÷ (division)

9. For computer numbers, the following hold
   a. '+' is Commutative in numbers, i.e., x + y = y + x, for any numbers x and y
   b. '*' is Commutative in numbers, i.e., x * y = y * x, for any numbers x and y

10. For computer numbers the following Hold: a. '+' is NOT Associative, i.e., (x + y) + z ≠ x + (y + z), for computer numbers x, y and z

   b. '*' is NOT Associative i.e., (x * y) * z ≠ x * (y * z), for computer numbers x, y and z,

11. A computer number is necessarily a binary number, i.e., it is a (finite) string of only 0's and 1's. (in this case, 0 and 1 are called ***bits***). However, as mentioned earlier, a particular string of bits may represent different numbers according to different schemes, i.e., sets of rules for interpreting a string as a number.

Properties of computer numbers continued: Further, in context of Property 1. of computer numbers, it may be stated that no real number, which is irrational number, can be represented in any computer system…. Only finitely many real numbers, each of which must also be rational, can be computer represent-able. But, as mentioned above, even not all rational numbers are computer represent-able. For example, 1/3 is a rational number, which can not be represented as a finite binary string, and hence, is not a computer number.

Further, it may be noted that some rational numbers which can be represented as a finite string of *decimal* digits, may not be written as a finite string of bits. For example, 1/5 can be written as: 0.2, a finite decimal string, but can be written only as an infinite *binary* string: 0.00110011……

Each of the real numbers, which can not be represented in a computer system, if required to be stored in a computer, is approximated appropriately, to a computer number (of the computer system)

0.4 Definitions & comments by Pioneers and Leading Writers about what Numerical Analysis is

---

1. K.E. Atkinson**:** Numerical analysis is the area of mathematics and computer science that creates, analyzes, and implements algorithms for solving numerically the problems of continuous mathematics.

   Such problems originate generally from real-world applications of algebra, geometry and calculus, and they involve variables which vary continuously; these problems occur throughout the natural sciences, social sciences, engineering, medicine, and business.
2. **Hammerlin & Hoffman:** Numerical Analysis is the mathematics of constructive methods, which can be realized numerically. Thus, one of the problems of numerical analysis is to design computer algorithms for either exactly or approximately solving problems in mathematics itself, or in its applications in natural sciences, technology, economics, and so forth.
3. **R.W. Hamming *(Page 1/ Numerical Methods for Scientists and Engineers):*** *Mathematics versus Numerical Analysis*…..Mathematics and numerical analysis differ from each other more than is usually realized. The most obvious differences are that *mathematics regularly uses the infinite* both for representation of numbers and for processes, whereas *computing is necessarily done on a finite machine in a finite time*. The finite representation of numbers in the machine leads to round-off errors, whereas the finite representation leads to truncation errors
4. **Young & Gregory *(P.1/ A survey of Numerical Mathematics, Vol. 1)*** Numerical analysis is concerned with the application of mathematics to the development of constructive, or algorithmic, methods which can be used to obtain numerical solutions to mathematical problems
5. **E.K. Blum *(Preface/ Numerical Analysis and Computation: Theory and Practice):*** Numerical analysis, in essence, is a branch of mathematics which deals with the numerical—and therefore constructive—solutions of problems formulated and studied in other branches of mathematics
6. **Wikipedia: Numerical analysis** is the study of algorithms that use numerical approximation (as opposed to general symbolic manipulations) for the problems of mathematical analysis (as distinguished from discrete mathematics).
7. **From Encyclopedia of Mathematics:** The branch of mathematics concerned with finding accurate approximations to the solutions of problems whose exact solution is either impossible

or infeasible to determine. In addition to the approximate solution, a realistic bound is needed for the error associated with the approximate solution.

Typically, a mathematical model for a particular problem, generally consisting of mathematical equations with constraint conditions, is constructed by specialists in the area concerned with the problem. Numerical analysis is concerned with devising methods for approximating the solution to the model, and analyzing the results for stability, speed of implementation, and appropriateness to the situation.

0.5 REFERENCES

---

1. Numerical Methods Using MATLAB (Fourth Edition) by J.H. Mathews & K.D. Fink (PHI, 2004)
2. Elements of Numerical Analysis by R.S. Gupta (Macmillan India Ltd., 2009)
3. Computer-Oriented Numerical Methods (Third Edition) by V. Rajaraman (P.H.I, 1999)
4. Numerical Analysis and Algorithms by Pradip Niyogi (Tata McGraw-Hill Pub.2003)
5. Theory and Problems on Numerical Analysis by F. Scheid (Schaum Series,1989)

For advanced Learners

6. Introduction to Numerical Computation (Second Edition) by James S. Vandergraft (Academic Press (1983))
7. Numerical Methods for Scientists and Engineers (Second Edition) by R. W. Hamming (McGraw-Hill, 1973)
8. Introduction to Numerical Analysis (Second Edition) by Carl-Eric Froberg (Addison-Wesley, 1981)
9. Basic Computational Mathematics by V.F. D'yachnko (Mir Publishers, 1979)
10. Elementary Numerical Analysis (3rd Edition) by S. D. Conte & C. DeBoor (McGraw-Hill, 1981)

11. **Free** NUMERICAL METHODS WITH APPLICATIONS
Authors: Autar K Kaw | **Co-Author**: Egwu E Kalu, Duc Nguyen
Contributors: Glen Besterfield, Sudeep Sarkar, Henry Welch, Ali Yalcin, Venkat Bhethanabotla
Website http://mathforcollege.com/textbook_index.html

UNIT 1: COMPUTER ARITHMETIC AND SOLUTION OF LINEAR AND NON-LINEAR EQUATIONS

Structure **Page Nos.**

1.0 Introduction

In view of the fact that numbers and arithmetic play an important role, not only in our academic matters, but in every day life also, even children are taught these almost from the very beginning in the school. Numbers play even bigger, rather indispensible, role in our understanding of computers systems, their functioning and their application; in view of which, the *Number systems* have been discussed in some form or other, in our earlier courses, including BCS-011, BCS-012, MCS-012, MCS-013 and MCS-021.

However, as has been emphasized in Unit 0, using computers to solve numerical problems, requires still deeper understanding of the number systems, specially, *computer* number systems, particularly, in view of the fact that slight lack of understanding of the numbers or lack of attention in their use may lead to disasters involving huge loss of life and property. In view of these facts, first, we discussed conventional numbers, their sets and systems, in some detail in Unit 0, and in more details in Appendix of the Block. In this unit, we discuss a particular, and most frequent, type of numbers, viz., floating point numbers and issues related to these numbers.

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.

Supplementary material

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

$$x = \pm (.d_1 d_2 \ldots d_n)_\beta \beta^e,\ 0 \le d_i < \beta,\ m \le e \le M;\ I = 1, 2, \ldots n,\ d_1 \neq 0;$$

where $(.d_1 d_2 \ldots d_n)_\beta$ is a β– fraction called mantissa and its value is given by

$$(.d_1 d_2 \ldots d_n)_\beta = d_1 \times \frac{1}{\beta} + d_2 \times \frac{1}{\beta^2} + \quad + d_n \times \frac{1}{\beta^n};$$ 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 $x = (d_1 d_2 \ldots d_{n+1} \ldots)\, \beta^e$, 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 $x = d_1 d_2 \ldots d_{n+1}$ 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 $\mathrm{fl}(x) = d_1 d_2 \ldots d_n \beta^e$.

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

$$\mathrm{fl}(-83.7) = \begin{cases} (.66) \times 10^0 \text{ chopped} \\ -(0.84) \times 10^3 \text{ rounded} \\ -(0.83) \times 10^3 \text{ chopped.} \end{cases}$$

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
$e_x = |x - \mathrm{fl}(x)|$ is called round-off (absolute) error,

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

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.

$$d_{n+1} < \frac{1}{2}\beta, \text{ then } fl(x) = \pm(.d_1 d_2 \ldots d_n)\beta^e$$
$$|x - fl(x)| = d_{n+1}, d_{n+2} \ldots \beta^{e-n-1}$$
$$\leq \frac{1}{2}\beta.\beta^{e-n-1} = \frac{1}{2}\beta^{e-n}$$

Case 2.

$$d_{n+1} \geq \frac{1}{2}\beta,$$
$$fl(x) = \pm\{(.d_1 d_2 \ldots d_n)\beta^e + \beta^{e-n}\}$$
$$|x - fl(x)| = .\left| -d_{n+1}, d_{n+2} .\beta^{e-n-1} + \beta^{e-n} \right|$$

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

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

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

$0 \le |r.g.e.| < \beta^{1-n}$, 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 $a = 0.345 \times 10^0$, $b = 0.245 \times 10^{-3}$ and $c = 0.432 \times 10^{-3}$. Using

3-digit decimal arithmetic with rounding, we have

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

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 $\frac{(a-b)}{c} = 0.071428 \ldots$.

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.

$x - x^* = -.00002$

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

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

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, $|r_{xwy}| < 10^{1-n}$ if rounded.
$|r_{xwy}| < 2.10^{1-n}$ 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

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

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

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.

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

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

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

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

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

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

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

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

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

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

That is $n \ge 8$.

Hence, $n \ge 8$ 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**: $f(x,y) = \frac{x}{y}$

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

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

$$= r_x - r_y$$

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 $x = .657562 \times 10^3$ and $y = .657557 \times 10^3$. 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)

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

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

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

$y_n \to 0$ 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.

$$E_n - E_n^* = -n(E_{n-1} - E_{n-1})$$
$$e_n = (-1)^n n! e_n$$
$$|e_n| \geq \tfrac{1}{2} .2^n |e_o| \text{ hence process is unstable.}$$

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

		rounding	**chopping**
(a)		$.37 \times 10^2$	$.37 \times 10^2$
		$.3722 \times 10^2$	$.3721 \times 10^2$
(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
$a_p \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-q}$.
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
$a_p \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-q+1} . (a_{-q} + 1)$

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

$f(x) = x - \sin x$ 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 Sin x

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

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:

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

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

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

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 $= .12 \times 10^{-10}$

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

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

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

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 $f(x)=\frac{10}{1-x^2}$

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

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.

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

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

UNIT 1 SOLUTION OF LINEAR ALGEBRAIC EQUATIONS

Structure **Page Nos.**

1.0 INTRODUCTION

In Block 1, we have discussed various numerical methods for finding the approximate roots of an equation $f(x) = 0$. Another important problem of applied mathematics is to find the (approximate) solution of systems of linear equations. Such systems of linear equations arise in a large number of areas, both directly in the modelling physical situations and indirectly in the numerical solution of other mathematical models. Linear algebraic systems also appear in the optimization theory, least square fitting of data, numerical solution of boundary value problems of ODE's and PDE's etc.

In this unit we will consider two techniques for solving systems of linear algebraic equations – Direct method and Iterative method.

These methods are specially suited for computers. Direct methods are those that, in the absence of round-off or other errors, yield the exact solution in a finite number of elementary arithmetic operations. In practice, because a computer works with a finite word length, direct methods do not yield exact solutions.

Indeed, errors arising from round-off, instability, and loss of significance may lead to extremely poor or even useless results. The fundamental method used for direct solution is Gauss elimination.

Iterative methods are those which start with an initial approximations and which, by applying a suitably chosen algorithm, lead to successively better approximations. By this method, even if the process converges, we can only hope to obtain an approximate solution. The important advantages of iterative methods are the simplicity and uniformity of the operations to be performed and well suited for computers and their relative insensivity to the growth of round-off errors.

So far, you know about the well-known Cramer's rule for solving such a system of equations. The Cramer's rule, although the simplest and the most direct method, remains a theoretical rule since it is a thoroughly inefficient numerical method where even for a system of ten equations, the total number of arithmetical operations required in the process is astronomically high and will take a huge chunk of computer time.

1.1 OBJECTIVES

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

- obtain the solution of system of linear algebraic equations by direct methods such as Cramer's rule, and Gauss elimination method;
- use the pivoting technique while transforming the coefficient matrix to upper triangular matrix;
- obtain the solution of system of linear equations, $A\mathbf{x} = \mathbf{b}$ when the matrix A is large or spare, by using one of the iterative methods – Jacobi or the Gauss-Seidel method;
- predict whether the iterative methods converge or not; and
- state the difference between the direct and iterative methods.

1.2 PRELIMINARIES

Let us consider a system of n linear algebraic equations in n unknowns

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots &\quad + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots &\quad + a_{2n}x_n = b_2 \\ a_{n1}x_1 + a_{n2}x_2 + \ldots &\quad + a_{nn}x_n = b_n \end{aligned} \qquad (1.2.1)$$

Where the coefficients a_{ij} and the constants b_i are real and known. This system of equations in matrix form may be written as

$$A\mathbf{x} = \mathbf{b} \quad \text{where } A = (a_{ij})_{n \times n} \qquad (1.2.2)$$
$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \text{ and } \mathbf{b} = (b_1, b_2, \ldots, b_n)^T.$$

A is called the coefficient matrix.

We are interested in finding the values x_i, $i = 1, 2 \ldots n$ if they exist, satisfying Equation (3.3.2).

We now give the following

Definition 1: A matrix in which all the off-diagonal elements are zero, i.e. $a_{ij} = 0$ for $i \neq j$ is called a diagonal matrix; e.g., $A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$ is a 3×3 diagonal matrix.

A square matrix is said to be upper – triangular if
$a_{ij} = 0$ for $i > j$, e.g.,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

Definition 2: A system of linear equations (3.3.2) is said to be consistent thus exists a solution. The system is said to be inconsistent if no solution exists. The system of equations (3.3.2) is said to be homogeneous if vector $\underline{b} = \underline{0}$, that is, all $b_i = 0$, otherwise the system is called non-homogeneous.

We state the following useful result on the solvability of linear systems.

Theorem 1: A non-homogeneous system of n linear equations in n unknown has a unique solution if and only if the coefficient matrix A is non singular (det A ≠ 0) and the solution can be expressed as $\mathbf{x} = A^{-1}\mathbf{b}$.

1.3 DIRECT METHODS

In schools, generally Cramer's rule/method is taught to solve system of simultaneous equations, based on the evaluation of determinants. This is a direct method. When n is small (say, 3 or 4), this rule is satisfactory. However, the number of multiplication operations needed increases very rapidly as the number of equations increases as shown below:

Number of equations	**Number of multiplication operations**
2	8
3	51
4	364
5	2885
.	
.	
.	
10	359251210

Hence a different approach is needed to solve such a system of equations on a computer. Thus, Cramer's rule, although the simplest and the most direct method, remains a theoretical rule and we have to look for other efficient direct methods. We are going to discuss one such direct method – Gauss' elimination method next after stating Cramer's Rule for the sake of completeness.

1.3.1 Cramer's Rule

In the system of equation (3.3.2), let $\Delta = \det(A)$ and $\mathbf{b} \neq 0$. Then the solutions of the system is obtained as $x_i = \Delta_i / \Delta$, i = 1, 2, … , n

where Δ_i is the determinant of the matrix obtained from A by replacing the i^{th} column of Δ by vector **b**.

1.3.2 Gauss Elimination Method

In Gauss's elimination method, one usually finds successively a finite number of linear systems equivalent to the given one such that the final system is so simple that its solution may be readily computed. In this method, the matrix A is reduced to the form U (upper triangular matrix) by using the elementary row operations like

(i) interchanging any two rows
(ii) multiplying (or dividing) any row by a non-zero constant
(iii) adding (or subtracting) a constant multiple of one row to another row.

If any matrix A is transformed to another matrix B by a series of row operations, we say that A and B are equivalent matrices. More specifically we have.

Definition 3: A matrix B is said to be row-equivalent to a matrix A, if B can be obtained from A by a using a finite number of row operations.
Two linear systems $A\mathbf{x} = \mathbf{b}$ and $A'\mathbf{x} = \mathbf{b}'$ are said to be equivalent if they have the same solution. Hence, if a sequence of elementary operations on $A\mathbf{x} = \mathbf{b}$ produces the new system $A'\mathbf{x} = \mathbf{b}'$, then the systems $A\mathbf{x} = \mathbf{b}$ and $A'\mathbf{x} = \mathbf{b}'$ are equivalent.

Let us illustrate (Naive) Gauss elimination method by considering a system of three equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \qquad (1.3.1)$$

Let $a_{11} \neq 0$.. We multiply first equation of the system by $-\frac{a_{22}}{a_{11}}$ and add to the second equation . Then we multiply the first equation by $-\frac{a_{31}}{a_{11}}$ and add to the third equation. The new equivalent system (first derived system) then becomes

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 &= b_2^{(1)} \\ a_{32}^{(1)}x_3 + a_{33}^{(1)}x_3 &= b_3^{(1)} \end{aligned} \qquad (1.3.2)$$

where

$$a_{22}^{(1)} = a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12}, \quad a_{23}^{(1)} = a_{23} - \frac{a_{21}}{a_{11}} \cdot a_{13},$$

$$b_2^{(1)} = b_2 - \frac{a_{21}}{a_{11}} \cdot b_1, \text{ etc.}$$

Next, we multiply the second equation of the derived system provided $a_{22}^{(1)} \neq 0$, by $-\frac{a_{32}^{(1)}}{a_{22}^{(1)}}$ and add to the third equation of (3.4.2). The system becomes

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 &= b_2^{(1)} \\ a_{33}^{(2)}x_3 &= b_3^{(2)} \end{aligned} \qquad (1.3.3)$$

where

$$a_{33}^{(2)} = a_{33}^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} \cdot a_{23}^{(1)}$$

and

$$b_3^{(2)} = b_3^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} b_2^{(1)}.$$

This system is an upper-triangular system and can be solved using back substitutions method provided $a_{33}^{(2)} \neq 0$. That is, the last equation gives $x_3 = \frac{b_3^{(2)}}{a_{33}^{(2)}}$; then substituting this value of x_3 in the last but one equation (second) we get the value of x_2 and then substituting the obtained values of x_3 and x_2 in the first equation we compute x_1. This process of solving an upper-triangular system of linear equations in often called **back substitution**. We illustrate this by the following example:

Example 1: Solve the following system of equations consisting of four equations.

(Equation 1)
$$\begin{aligned} E_1: &\quad x_1 + x_2 + 0.x_3 + 3x_4 = 4 \\ E_2: &\quad 2x_1 + x_2 - x_3 + x_4 = 1 \\ E_3: &\quad 3x_1 - x_2 - x_3 + 2x_4 = -3 \\ E_4: &\quad -x_1 + 2x_2 + 3x_3 - x_4 = 4. \end{aligned}$$

Solution: The first step is to use first equation to eliminate the unknown x_1 from second, third and fourth equation. This is accomplished by performing $E_2 - 2E_1$, $E_3 - 3E_1$ and $E_4 + E_1$. This gives the derived system as

$$\begin{aligned}
E'_1: &\quad x_1 + x_2 + 0.x_3 + 3x_4 = 4\\
E'_2: &\quad -x_2 - x_3 + 5x_4 = -7\\
E'_3: &\quad -4x_2 - x_3 - 7x_4 = -15\\
E'_4: &\quad 3x_2 + 3x_3 + 2x_4 = 8.
\end{aligned}$$

In this new system, E'_2 is used to eliminate x_2 from E'_3 and E'_4 by performing the operations $E'_3 - 4E'_2$ and $E'_4+3E'_2$. The resulting system is

$$\begin{aligned}
E''_1: &\quad x_1 + x_2 + 0.x_3 + 3x_4 = 4\\
E''_2: &\quad -x_2 - x_3 + 5x_4 = -7\\
E''_3: &\quad 3x_3 + 13x_4 = 13\\
E''_4: &\quad -13x_4 = -13.
\end{aligned}$$

This system of equation is now in triangular form and can be solved by back substitution. E''_4 gives $x_4 = 1$, E''_3 gives

$$x_3 = \frac{1}{3}(13 - 13x_4) = \frac{1}{3}(13 - 13 \times 1) = 0.$$

E''_2 gives $x_2 = -(-7 + 5x_4 + x_3) = -(-7 + 5 \times 1 + 0) = 2$ and E''_1 gives $x_1 = 4 - 3x_4 - x_2 = 4 - 3 \times 1 - 2 = -1$.

The above procedure can be carried out conveniently in matrix form as shown below:

We consider the Augmented matrix [A1**b**] and perform the elementary row operations on the augmented matrix.

$$[A \mid \mathbf{b}] \quad \left[\begin{array}{cccc|c} 1 & 2 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right] \quad R_2 - 2R_1,\ R_3 - 3R_1 \text{ and } R_4 + R_1 \text{ gives}$$

$$= \quad \left[\begin{array}{cccc|c} 1 & 2 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & -4 & -1 & -7 & -15 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right] \quad R_3 - 4R_2,\ R_4 + 3R_2 \text{ gives}$$

$$= \quad \left[\begin{array}{cccc|c} 1 & 2 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{array}\right]$$

This is the final equivalent system:

$$\begin{aligned}
x_1 + x_2 + 0x_3 + 3x_4 &= 4\\
-x_2 - x_3 - 5x_4 &= -7\\
3x_3 + 13x_4 &= 13\\
-13x_4 &= -13.
\end{aligned}$$

The method works with the assumption that none of the elements a_{11}, $a_{22}^{(1)}$, …, $a_{n-1,n-1}^{(n-2)}$, $a_{n,n}^{(n-1)}$ is zero. This does not necessarily mean that the linear system is not solvable, but the following technique may yield the solution:

Suppose $a_{kk}^{(k-1)} = 0$ for some $k = 2, \ldots, n-2$. The kth column of $(k-1)$th equivalent system from the kth row is searched for the first non zero entry. If $a_{pk}^{(k)} \neq 0$ for some p,

$k + 1 \le p \le n$, then interchange R_k by R_p to obtain an equivalent system and continue the procedure. If $a_{pk}^{(k)} = 0$ for p = k, k + 1, …, n, it can be shown that the linear system does not have a unique solution and hence the procedure is terminated.

You may now solve the following exercises:

E1) Solve the system of equations
$3x_1 + 2x_2 + x_3 = 3$
$2x_1 + x_2 + x_3 = 0$
$6x_1 + 2x_2 + 4x_3 = 6$
using Gauss elimination method. Does the solution exist?

E2) Solve the system of equations
$16x_1 + 22x_2 + 4x_3 = -2$
$4x_1 - 3x_2 + 2x_3 = 9$
$12x_1 + 25x_2 + 2x_3 = -11$
using Gauss elimination method and comment on the nature of the solution.

E3) Solve the system of equations by Gauss elimination.
$x_1 - x_2 + 2x_3 - x_4 = -8$
$2x_1 - 2x_2 + 3x_3 - 3x_4 = -20$
$x_1 + x_2 + x_3 + 0.x_4 = -2$
$x_1 - x_2 + 4x_3 + 3x_4 = 4$

E4) Solve the system of equations by Gauss elimination.
$x_1 + x_2 + x_3 + x_4 = 7$
$x_1 + x_2 + 0.x_3 + 2x_4 = 8$
$2x_1 + 2x_2 + 3x_3 + 0.x_4 = 10$
$-x_1 - x_2 - 2x_3 + 2x_4 = 0$

E5) Solve the system of equation by Gauss elimination.
$x_1 + x_2 + x_3 + x_4 = 7$
$x_1 + x_2 + 2x_4 = 5$
$2x_1 + 2x_2 + 3x_3 = 10$
$-x_1 - x_2 - 2x_3 + 2x_4 = 0$

It can be shown that in Gauss elimination procedure and back substitution $(2n^3 + 3n^2 - 5n)/6 + \frac{n^2 + n}{2}$ multiplications/divisions and $\frac{n^3 - n}{3} + \frac{n^2 - n}{2}$ additions/subtractions are performed respectively. The total arithmetic operation involved in this method of solving a $n \times n$ linear system is $\frac{n^3 + 3n^2 - n}{3}$ multiplication/divisions and $\frac{2n^3 + 3n^n - 5n}{6}$ additions/subtractions.

Definition 4: In Gauss elimination procedure, the diagonal elements $a_{11}, a_{22}^{(1)}, a_{33}^{(2)}$, which have been used as divisors are called pivots and the corresponding equations, are called pivotal equations.

1.3.3 Pivoting Strategies

If at any stage of the Gauss elimination, one of these pivots say $a_{ii}^{i-1} (a_{11}^{(0)} = a_{11})$, vanishes then we have indicated a modified procedure. But it may also happen that the pivot $a_{ii}^{(i-1)}$, though not zero, may be very small in magnitude compared to the

remaining elements (≥ i) in the ith column. Using a small number as divisor may lead to growth of the round-off error. The use of large multipliers like

$$\frac{-a_{i+1}^{(i-12)}, i}{a_{ii}^{(i-1)}}, \frac{a_{i+2,i}^{(i-1)}}{a_{ii}^{(i-1)}}$$

etc. will lend to magnification of errors both during the elimination phase and during the back substitution phase of the solution procedure. This can be avoided by rearranging the remaining rows (from ith row up to nth row) so as to obtain a non-vanishing pivot or to choose one that is largest in magnitude in that column. This is called pivoting strategy.

There are two types of pivoting strategies: partial pivoting (maximal column pivoting) and complete pivoting. We shall confine to simple partial pivoting and complete pivoting. That is, the method of scaled partial pivoting will not be discussed. Also there is a convenient way of carrying out the pivoting procedure where instead of interchanging the equations all the time, the n original equations and the various changes made in them can be recorded in a systematic way using the augmented matrix [A1**b**] and storing the multiplies and maintaining pivotal vector. We shall just illustrate this with the help of an example. However, leaving aside the complexities of notations, the procedure is useful in computation of the solution of a linear system of equations.

If exact arithmetic is used throughout the computation, pivoting is not necessary unless the pivot vanishes. But, if computation is carried up to a fixed number of digits (precision fixed), we get accurate results if pivoting is used.

The following example illustrates the effect of round-off error while performing Gauss elimination:

Example 2: Solve by the Gauss elimination the following system using four-digit arithmetic with rounding.

$$0.003000x_1 + 59.14\ x_2 = 59.17$$
$$5.291x_1 - 6.130x_2 = 46.78.$$

Solution: The first pivot element $a_{11}^0 = a_{11} = 0.0030$ and its associated multiplier is

$$\frac{5.291}{0.0030} = 1763.66 \approx 1763$$

Performing the operation of elimination of x_1 from the second equation with appropriate rounding we got

$$0.003000\ x_1 + 59.14x_2 = 59.17$$
$$-104300\ x_2 = -104400$$

By backward substitution we have

$$x_2 = 1.001 \text{ and } x_1 = \frac{59.17 - (59.14)(1.001)}{0.00300} = -10.0$$

The linear system has the exact solution $x_1 = 10.00$ and $x_2 = 1{,}000$.

However, if we use the second equation as the first pivotal equation and solve the system, the four digit arithmetic with rounding yields solution as $x_1 = 10.00$ and $x_2 = 1.000$. This brings out the importance of partial or maximal column pivoting.

Partial pivoting (Column Pivoting)

In the first stage of elimination, instead of using $a_{11} \neq 0$ as the pivot element, the first column of the matrix A ([A1**b**]) is searched for the largest element in magnitude and this largest element is then brought at the position of the first pivot by interchanging first row with the row having the largest element in magnitude in the first column.

Next, after elimination of x_1, the second column of the derived system is searched for the largest element in magnitude among the (n – 1) element leaving the first element. Then this largest element in magnitude is brought at the position of the second pivot by interchanging the second row with the row having the largest element in the second column of the derived system. The process of searching and interchanging is repeated in all the (n – 1) stages of elimination. For selecting the pivot we have the following algorithm:

For i = 1, 2, …, n find j such that

$$\left|a_{ji}^{(i-1)}\right| = \max_{i \le k \le n} \left|a_{ki}^{(i-1)}\right| \qquad \left(a_{ji}^{0} = a_{ji}\right)$$

Interchange ith and jth rows and eliminate x_i.

Complete Pivoting

In the first stage of elimination, we look for the largest element in magnitude in the entire matrix A first. If the element is a_{pq}, then we interchange first row with pth row and interchange first column with qth column, so that a_{pq} can be used as a first pivot. After eliminating x_q, the process is repeated in the derived system, more specifically in the square matrix of order n – 1, leaving the first row and first column. Obviously, complete pivoting is quite cumbersome.

Scaled partial pivoting (Sealed column pivoting)

First a scale factor d_i for each row i is defined by $d_i = \max_{i \le j \le n} \left|a_{ij}\right|$

If $d_i = 0$ for any i, there is no unique solution and procedure is terminated. In the first stage choose the first integer k such that

$$\left|a_{k1}\right| / d_k = \max_{i \le j \le n} \left|a_{j1} / d_j\right|$$

interchange first row and kth row and eliminate x_1. The process is repeated in the derived system leaving aside first row and first column.

We now illustrate these pivoting strategies in the following examples.

Example 3: Solve the following system of linear equations with partial pivoting

$$\begin{aligned} x_1 - x_2 + 3x_3 &= 3 \\ 2x_1 + x_2 + 4x_3 &= 7 \\ 3x_1 + 5x_2 - 2x_3 &= 6 \end{aligned}$$

$$[A1b] = \left(\begin{array}{ccc|c} 1 & -1 & 3 & 3 \\ 2 & 1 & 4 & 7 \\ 3 & 5 & -2 & 6 \end{array}\right) \quad R_1 - \frac{1}{3}R_3, R_2 - \frac{2}{3}R_3$$

$$= \left(\begin{array}{ccc|c} 0 & -\frac{8}{3} & \frac{11}{3} & 1 \\ 0 & -\frac{7}{3} & \frac{16}{3} & 3 \\ 3 & 5 & -2 & 6 \end{array}\right) \qquad R_2 - \frac{7}{3}\cdot\frac{3}{8}R_1$$

$$\left(\begin{array}{ccc|c} 0 & -\frac{8}{3} & \frac{11}{3} & 1 \\ 0 & 0 & \frac{51}{24} & \frac{17}{8} \\ 3 & 5 & -2 & 6 \end{array}\right)$$

Re-arranging the equations (3rd equation becomes the first equation and first equation becomes the second equation in the derived system), we have

$$3x_1 + 5x_2 - 2x_3 = 6$$
$$-\frac{8}{3}x_2 + \frac{11}{3}x_3 = 1$$
$$\frac{51}{24}x_3 = \frac{17}{18}$$

Using back substitution we have $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$.

You may now solve the following exercises:

E6) Solve the system of linear equation given in the Example 3 by complete pivoting.

E7) Solve the system of linear equation given in Example 3 by scaled partial pivoting.

E8) Solve the system of equations with partial (maximal column) pivoting.

$$x_1 + x_2 + x_3 = 6$$
$$3x_1 + 3x_2 + 4x_3 = 20$$
$$2x_1 + x_2 + 3x_3 = 13$$

1.4 ITERATIVE METHODS

Consider the system of equations

$$A\mathbf{x} = \mathbf{b} \qquad \ldots (1.4.1)$$

Where A is an n × n non-singular matrix. An iterative technique to solve the n x n linear system (1.4.1) starts with an initial approximation $\underline{x}^{(0)}$ to the solution **x**, and generates a sequence of vectors $\{\underline{x}^k$ that **converges** to **x**, the actual solution vector (When $\max_{1\le i\le n}\left|x_i^{(k)} - x_i\right| < \varepsilon$ for some k when ε is a given small positive numbers.).

Most of these iterative techniques entails a process that converts the system A**x** = **b** into an equivalent system of the form **x** = T**x** + **c** for some n × n matrix T and vector **c**. In general we can write the iteration method for solving the linear system (3.5.1) in the form

$$x^{(k+1)} = Tx^{(k)} + c \qquad k = 0, 1, 2, \ldots,$$

T is called the iteration matrix and depends on A, **c** is a column vector which depends on A and **b**. We illustrate this by the following example.

Iterative methods are generally used when the system is large (when n > 50) and the matrix is sparse (matrices with very few non-zero entries).

Example 4: Convert the following linear system of equations into equivalent form $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6 \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

Solution: We solve the ith equation for x_i (assuming that $a_{ii} \neq 0 \ \forall$ i. If not, we can interchange equations so that is possible)

$$x_1 = + \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5}$$

$$x_2 = \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}$$

$$x_3 = -\frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10}$$

$$x_4 = -\frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}$$

Here $T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix}$ and $\mathbf{c} = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}$

1.4.1 The Jacobi Iterative Method

This method consists of solving the ith equation of $A\mathbf{x} = \mathbf{b}$ for x_i, to obtain

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{-a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}} \quad \text{for } i = 1, 2, \ldots, n$$

provided $a_{ii} \neq 0$.

We generate $\mathbf{x}^{(k+1)}$ from $\mathbf{x}^{(k)}$ for $k \geq 0$ by

$$x_i^{(k+1)} = \sum_{\substack{j=i \\ j \neq i}}^{n} \left(-a_{ij}x_j^{(k)} + b_i \right) \Big/ a_{ii} \quad i = 1,2,\ldots,n \qquad (1.4.2)$$

We state below a sufficient condition for convergence of the Jacobi Method.

Theorem

If the matrix A is strictly diagonally dominant, that is, if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, 2, \ldots n.$$

then the Jacobi iteration method (3.5.2) converges for any initial approximation $\mathbf{x}^{(0)}$.

Generally $\mathbf{x}^{(0)} = \mathbf{0}$ is taken in the absence of any better initial approximation.

Example 5: Solve the linear system $A\mathbf{x} = \mathbf{b}$ given in previous example (Example 4) by Jacobi method rounded to four decimal places.

Solution: Letting $\mathbf{x}^{(0)} = (0,0,0,0)^T$, we get

$$\mathbf{x}^{(1)} = (0.6000, 2.2727 - 1.1000, 1.8750)^T$$

$$\mathbf{x}^{(2)} = (1.0473, 1.7159, -0.8052, 0.8852)^T \text{ and}$$

$$\mathbf{x}^{(3)} = (0.9326, 2.0533, -1.0493, 1.1309)^T$$

Proceeding similarly one can obtain

$$\mathbf{x}^{(5)} = (0.9890, 2.0114, -1.0103, 1.0214)^T \text{ and}$$

$$\mathbf{x}^{(10)} = (1.0001, 1.9998, -0.9998, 0.9998)^T.$$

The solution is $\mathbf{x} = (1, 2, -1, 1)^T$. You may note that $\mathbf{x}^{(10)}$ is a good approximation to the exact solution compared to $\mathbf{x}^{(5)}$.

You also observe that A is strictly diagonally dominant
(since $10 > 1 + 2$, $11 > 1 + 1 + 3$ $10 > 2 + 1 + 1$ and $8 > 3 + 1$).

Now we see how $A\mathbf{x} = \mathbf{b}$ is transformed to an equivalent system $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

The matrix can be written as

$A = D + L + U$ where

$$D = \begin{bmatrix} a_{11}, & 0 & \ldots & 0 \\ 0 & a_{22}, & \ldots & 0 \\ 0 & \ldots & 0, & a_{nn} \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & \ldots & \ldots & 0 \\ a_2 & 0 & \ldots & \ldots & 0 \\ a_3, & a_{32} & 0 & \ldots & 0 \\ a_n, & a_{n2} & \ldots & a_{n,\,n-1} & \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & a_{12} & & \ldots & a_{1n} \\ 0 & 0 & a_{23} & \ldots & a_{2n} \\ 0 & 0 & 0 & \ldots & a_{n-1,\,n} \end{bmatrix}$$

Since $(D + L + U)\,\mathbf{x} = \mathbf{b}$

$$D\mathbf{x} = -(L + U)\,\mathbf{x} + \mathbf{b}$$

$$\mathbf{x} = -D^{-1}(L + U)\,\mathbf{x} + D^{-1}\mathbf{b}$$

i.e. $T = D^{-1}(L + U)$ and $\mathbf{c} = D^{-1}\mathbf{b}$.

In Jacobi method, each of the equations is simultaneously changed by using the most recent set of **x**-values. Hence the Jacobi method is called method of simultaneous displacements.

You may now solve the following exercises:

E9) Perform five iterations of the Jacobi method for solving the system of equations.

$$\begin{bmatrix} 5 & -1 & -1 & -1 \\ -1 & 10 & -1 & -1 \\ -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -4 \\ 12 \\ 8 \\ 34 \end{bmatrix}$$

Starting with $\mathbf{x}^{(0)} = (0,0,0,0)$. The exact solution is $\mathbf{x} = (1,2,3,4)T^n$. How good $\mathbf{x}^{(5)}$ as an approximation to $\mathbf{x}$?

E10) Perform four iterations of the Jacobi method for solving the following system of equations.

$$\begin{bmatrix} 2 & -1 & -0 & -0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

With $\mathbf{x}^{(0)} = (0.5, 0.5, 0.5, 0.5)^T$. Here $\mathbf{x} = (1, 1, 1, 1)^T$. How good $\mathbf{x}^{(5)}$ as an approximation to $\mathbf{x}$?

1.4.2 The Gauss-Seidel Iteration Method

In this method, we can write the iterative scheme of the system of equations $A\mathbf{x} = \mathbf{b}$ as follows:

$$a_{11}x_1^{(k+1)} = -a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \ \ldots \ - a_n x_n^{(k)} + b_1$$

$$a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} = -a_{23}x_3^{(k)} - \ \ldots \ - a_{2n}x_n^{(k)} + b_2$$

.

.

$$a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} \ldots + a_{nn}x_n^{(k+1)} = + b_n$$

In matrix form, this system can be written as $(D + L)\ \mathbf{x}^{(k+1)} = -U\,\mathbf{x}^{(k)} + \mathbf{b}$ with the same notation as adopted in Jacobi method.

From the above, we get

$$x^{(k+1)} = -(D+L)^{-1}Ux^{(k)} + (D+L)^{-1}b$$

$$= T\mathbf{x}^{(k)} + \boldsymbol{c}_n$$

i.e. $T = -(D+L)^{-1n}\,U$ and $\mathbf{c} = (D + L)^{-1}\mathbf{b}$

This iteration method is also known as the method of successive displacement.

For computation point of view, we rewrite (A x) as

$$x_i^{(k+1)} = -\frac{1}{a_{ii}}\left[\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^{x} a_{ij}x_j^{(k)} - b_i\right]$$

$$i = 1, 2, \ldots, n$$

Also in this case, if A is diagonally dominant, then iteration method always converges. In general Gauss-Seidel method will converge if the Jacobi method converges and will converge at a faster rate. You can observe this in the following example. We have not considered the problem: How many iterations are needed to have a reasonably good approximation to **x**? This needs the concept of matrix norm.

Example 6: Solve the linear system A**x** = **b** given in Example 4 by Gauss-Seidel method rounded to four decimal places. The equations can be written as follows:

$$
\begin{aligned}
x_1^{(k+1)} &= \frac{1}{10}x_2^{(k)} - \frac{1}{3}x_3^{(k)} + \frac{3}{5} \\
x_2^{(k+1)} &= \frac{1}{11}x_1^{(k+1)} + \frac{1}{11}x_3^{k} - \frac{3}{11}x_4^{(k)} + \frac{25}{11} \\
x_3^{(k+1)} &= -\frac{1}{3}x_1^{(k+1)} + \frac{1}{10}x_2^{(k+1)} + \frac{1}{10}x_4^{(k)} - \frac{11}{10} \\
x_4^{(k+1)} &= -\frac{3}{8}x_2^{(k+1)} + \frac{1}{8}x_3^{(k+1)} + \frac{15}{8}.
\end{aligned}
$$

Letting $\mathbf{x}^{(0)} = (0, 0, 0, 0)^T$ we have from first equation

$$
\begin{aligned}
x_1^{(1)} &= 0.6000 \\
x_2^{(1)} &= \frac{0.6000}{3} + \frac{25}{11} = 2.3273 \\
x_3^{(1)} &= -\frac{0.6000}{3} + \frac{1}{10}(2.3273) - \frac{11}{10} = -0.1200 + 0.2327 - 1.1000 = -0.9873 \\
x_4^{(1)} &= -\frac{3}{8}(2.3273) + \frac{1}{8}(-0.9873) + \frac{15}{8}
\end{aligned}
$$

$= -0.8727 - 0.1234 + 1.8750$

$= 0.8789$

Using $\mathbf{x}^{(1)}$ we get

$$\mathbf{x}^{(2)} = (1.0300, 2.037, -1.014, 0.9844)^T$$

and we can check that

$$\mathbf{x}^{(5)} = (1.0001, 2.0000, -1.0000, 1.0000)^T$$

Note that $\mathbf{x}^{(5)}$ is a good approximation to the exact solution. Here are a few exercises for you to solve.

You may now solve the following exercises:

E11) Perform four iterations (rounded to four decimal places) using Jacobi Method and Gauss-Seidel method for the following system of equations.

$$
\begin{bmatrix} -8 & 1 & 1 \\ 1 & -5 & -1 \\ 1 & 1 & -4 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =
\begin{bmatrix} 1 \\ 16 \\ 7 \end{bmatrix}
$$

With $\mathbf{x}^{(0)} = (0, 0, 0)^T$. The exact solution is $(-1, -4, -3)^T$. Which method gives better approximation to the exact solution?

E12) For linear system given in E10), use the Gauss Seidel method for solving the system starting with $\mathbf{x}^{(0)} = (0.5, 0.5, 0.5, 0.5)^T$ obtain $\mathbf{x}^{(4)}$ by Gauss-Seidel method and compare this with $\mathbf{x}^{(4)}$ obtained by Jacobi method in E10).

1.4.3 Comparison of Direct and Iterative Methods

Both the methods have their strengths and weaknesses and a choice is based on the particular linear system to be solved. We mention a few of these below:

Direct Method

1. The direct methods are generally used when the matrix A is dense or filled, that is, there are few zero elements, and the order of the matrix is not very large, say n < 50.

2. The rounding errors may become quite large for ill conditioned equations (If at any stage during the application of pivoting strategy, it is found that all values of $\{ |a_{mk}\ 1|$ for m = k + 1, to n are less than a pre-assigned small quantity ε, then the equations are ill-conditioned and no useful solution is obtained). Ill-conditioned matrices are not discussed in this unit.

Iterative Method

1. These methods are generally used when the matrix A is sparse and the order of the matrix A is very large say n > 50. Sparse matrices have very few non-zero elements.

2. An important advantage of the iterative methods is the small rounding error. Thus, these methods are good choice for ill-conditioned systems.

3. However, convergence may be guaranteed only under special conditions. But when convergence is assured, this method is better than direct.

With this we conclude this unit. Let us now recollect the main points discussed in this unit.

1.5 SUMMARY

In this unit we have dealt with the following:

1. We have discussed the direct methods and the iterative techniques for solving linear system of equations A**x** = **b** where A is an n × n, non-singular matrix.

2. The direct methods produce the exact solution in a finite number of steps provided there are no round off errors. Direct method is used for linear system A**x** = **b** where the matrix A is dense and order of the matrix is less than 50.

3. In direct methods, we have discussed Gauss elimination, and Gauss elimination with partial (maximal column) pivoting and complete or total pivoting.

4. We have discussed two iterative methods, Jacobi method and Gauss-Seidel method and stated the convergence criterion for the iteration scheme. The iterative methods are suitable for solving linear systems when the matrix is sparse and the order of the matrix is greater than 50.

1.6 SOLUTION/ANSWERS

E1) $[A1\mathbf{b}]$ $\left[\begin{array}{ccc|c} 3 & 2 & 1 & 3 \\ 2 & 1 & -1 & 0 \\ 6 & 2 & 4 & 6 \end{array}\right]$

$a_{11} \neq 0 \longrightarrow$ $\left[\begin{array}{ccc|c} 3 & 2 & 1 & 3 \\ 0 & \frac{1}{3} & -\frac{1}{3} & -2 \\ 0 & -2 & 2 & 0 \end{array}\right]$

$a_{22}^{(1)} \neq 0 \longrightarrow$ $\left[\begin{array}{ccc|c} 3 & 2 & 1 & 3 \\ 0 & -\frac{1}{3} & -\frac{1}{3} & -2 \\ 0 & 0 & 0 & 12 \end{array}\right]$

This system has no solution since x_3 cannot be determined from the last equation. This system is said to be inconsistent. Also note that del (A) = 0.

E2) $[A1\mathbf{b}]$ $\left[\begin{array}{ccc|c} 16 & 22 & 4 & -2 \\ 4 & -3 & 2 & 9 \\ 12 & 25 & 2 & -11 \end{array}\right]$

$a_{11} \neq 0 \longrightarrow$ $\left[\begin{array}{ccc|c} 16 & 22 & 4 & 2 \\ 0 & -\frac{17}{2} & 1 & \frac{19}{2} \\ 0 & \frac{17}{2} & -1 & -\frac{19}{2} \end{array}\right]$

$a_{22}^{(1)} \neq 0 \longrightarrow$ $\left[\begin{array}{ccc|c} 16 & 22 & 4 & 2 \\ 0 & -\frac{17}{2} & 1 & \frac{19}{2} \\ 0 & 0 & 0 & 0 \end{array}\right]$ $\Rightarrow x_3 =$ arbitrary value, $x_2 =$

$\frac{-2}{17}\left(\frac{19}{2} - x_3\right)$ and $x_3 = \frac{1}{6}(-2 - 22x_3 - 22x_3)$

This system has infinitely many solutions. Also you may check that det (A) = 0.

E3) Final derived system:

$\left[\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 0 & 2 & 4 \end{array}\right]$ and the solution is $x_4 = 2$, $x_3 = 2$ $x_2 = 3$, $x_1 = -7$.

E4) Final derived system:

$$\left[\begin{array}{cccc|c} 1 & -1 & 1 & 1 & 7 \\ 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 1 & -2 & -4 \\ 0 & 0 & 0 & 1 & 3 \end{array}\right]$$ and the solution is

$x_4 = 3$, $x_3 = 2$, x_2 arbitrary and $x_1 = 2 - x_2$.

Thus this linear system has infinite number of solutions.

E5) Final derived system:

$$\left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 7 \\ 0 & 0 & -1 & 1 & -2 \\ 0 & 0 & 1 & -2 & -4 \\ 0 & 0 & 0 & 1 & 3 \end{array}\right]$$ and the solution does not

exist since we have $x_4 = 3$, $x_3 = 2$ and third equation ϕ $-x_3 + x_4 = -2$ implies $1 = -2$, leading to a contradiction.

E6) Since $|a_{32}|$ is maximum we rewrite the system as

$$\left[\begin{array}{ccc|c} 5 & 3 & -2 & 6 \\ 1 & 2 & 4 & 7 \\ -1 & 1 & 3 & 3 \end{array}\right]$$ by interchanging R_1 and R_3 and C_1 and C_2

$R_2 - \frac{1}{5}R_1, R_3 + \frac{1}{5}R_1$ gives

$$\left[\begin{array}{ccc|c} 5 & 3 & -2 & 6 \\ 0 & \frac{7}{5} & \frac{22}{5} & \frac{29}{5} \\ 0 & \frac{8}{5} & \frac{13}{5} & \frac{21}{5} \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 5 & -2 & 3 & 6 \\ 0 & \frac{22}{5} & \frac{7}{5} & \frac{29}{5} \\ 0 & \frac{13}{5} & \frac{8}{5} & \frac{21}{5} \end{array}\right]$$ by interchanging C_2 and C_3

Since $|a_{23}|$ is maximum –

By $R_3 - \frac{5}{12} x \frac{13}{15} R_2$ we have

$$\left[\begin{array}{ccc|c} 5 & -2 & 3 & 6 \\ 0 & \frac{22}{5} & \frac{7}{5} & \frac{29}{5} \\ 0 & 0 & \frac{17}{22} & \frac{43}{22} \end{array}\right]$$

$$5x_2 + 3x_1 - 2x_3 = 6$$
$$\frac{22}{5}x_3 + \frac{7}{5}x_2 = \frac{29}{5}$$
$$\frac{17}{22}x_2 = \frac{17}{22}$$

We have $x_2 = 1$, $\frac{22}{5}x_3 = \frac{29}{5} - \frac{7}{5} = \frac{22}{5} \Rightarrow x_3 = 1$

$3x_1 = 6 - 5 + 2 \Rightarrow x_1 = 1$

E7) For solving the linear system by sealed partial pivoting we note that $d_1 = 3$, $d_2 = 4$ and $d_3 = 5$ in

$$W = [A1\mathbf{b}] = \begin{bmatrix} 1 & -1 & 3 & 3 \\ 2 & 1 & 4 & 7 \\ 3 & 5 & -2 & 6 \end{bmatrix} \quad p = [a, 2, 3]^T$$

Since $\max\left\{\frac{1}{3}, \frac{2}{4}, \frac{3}{5}\right\} = \frac{3}{5}$, third equation is chosen as the first pivotal equation.

Eliminating x_1 we have

d

$$\begin{bmatrix} 3 \\ \\ 4 \\ \\ 5 \end{bmatrix} \quad W^1 = \begin{bmatrix} \frac{1}{3} & \frac{8}{3} & \frac{11}{3} & 1 \\ -\frac{2}{3} & -\frac{7}{3} & \frac{16}{3} & 3 \\ 3 & 5 & -2 & 6 \end{bmatrix}$$

where we have used a square to enclose the pivot element and

in place of zero entries, after elimination of x_1 from 1st and 2nd equation, we have stored the multipliers. Here $m_{11} = \frac{a_{11}}{a_{31}} = \frac{1}{3}$ *and* $m_{2,1} = \frac{a_{21}}{a_{31}} = \frac{2}{3}$

Instead of interchanging rows (here R_1 and R_3)we keep track of the pivotal equations being used by the vector $p=[3,2,1]^T$

In the next step we consider $\max\left\{\frac{7}{3} \cdot \frac{1}{4}\frac{8}{3} \cdot \frac{1}{3}\right\} = \frac{8}{9}$

So the second pivotal equation is the first equation.

i.e. $p = [3, 1, 3]^T$ and multiplier is $-\frac{7}{3} - \frac{8}{3} = \frac{7}{8} = m_{2,2}$

$$\frac{1}{3} \quad \text{and } W^{(2)} = \begin{bmatrix} -\frac{8}{3} & \frac{11}{3} & 1 \\ \frac{7}{8} & \frac{17}{8} & \frac{17}{8} \\ 5 & -2 & 6 \end{bmatrix} \quad p = [3, 1, 2]^T$$

$$3 \quad \begin{bmatrix} -\frac{8}{3} & \frac{11}{3} & 1 \\ \frac{7}{8} & \frac{17}{8} & \frac{17}{8} \\ 5 & -2 & 6 \end{bmatrix}$$

The triangular system is as follows:

$$\begin{aligned} 3x_1 + 5x_2 - 2x_3 &= 6 \\ -\frac{8}{3}x_2 + \frac{11}{3}x_3 &= 1 \\ \frac{17}{8}x_3 &= \frac{17}{8} \end{aligned}$$

By back substitution, this yields $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$.

Remark: The **p** vector and storing of multipliers help solving the system A**x** = **b**' where b is changed **b**'.

E8) $[A, \mathbf{b}] = \begin{bmatrix} 1 & 1 & 1 & 6 \\ 3 & 3 & 4 & 20 \\ 2 & 1 & 3 & 13 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 3 & 4 & 20 \\ 1 & 1 & 1 & 6 \\ 2 & 1 & 3 & 13 \end{bmatrix}$

$$\xrightarrow{R_2 - \frac{1}{3}R_1, R_3 - \frac{2}{3}R_1} \begin{bmatrix} 3 & 3 & 4 & 20 \\ 0 & 0 & -\frac{1}{3} & -\frac{2}{3} \\ 0 & -1 & \frac{1}{3} & -\frac{1}{3} \end{bmatrix} \quad \begin{bmatrix} 13 & 3 & 4 & 20 \\ 0 & -1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & -\frac{1}{3} & -\frac{2}{3} \end{bmatrix}$$

Since the resultant matrix is in triangular form, using back substitution we get $x_3 = 2$, $x_2 = 1$ and $x_1 = 3$.

E9) Using $\mathbf{x}^{(0)} = [0, 0, 0, 0]^T$ we have

$\mathbf{x}^{(1)} = [-0.8, 1.2, 1.6, 3.4]^T$

$\mathbf{x}^{(2)} = [0.44, 1.62, 2.36, 3.6]^T$

$\mathbf{x}^{(3)} = [0.716, 1.84, 2.732, 3.842]^T$

$\mathbf{x}^{(4)} = [0.8823, 1.9290, 2.8796, 3.9288]^T$

E10) Using $\mathbf{x}^{(0)} = [0.5, 0.5, 0.5, 0.5]^T$, we have

$\mathbf{x}^{(1)} = [0.75, 0.5, 0.5, 0.75]^T$

$\mathbf{x}^{(2)} = [0.75, 0.625, 0.625, 0.75]^T$

$\mathbf{x}^{(3)} = [0.8125, 0.6875, 0.6875, 0.8125]^T$

$\mathbf{x}^{(4)} = [0.8438, 0.75, 0.75, 0.8438]^T$

E 11) By Jacobi method we have

$\mathbf{x}^{(1)} = [-0.125, -3.2, -1.75]^T$

$\mathbf{x}^{(2)} = [-0.7438, -3.5750, -2.5813]^T$

$\mathbf{x}^{(3)} = [-0.8945, -3.8650, -2.8297]^T$

$\mathbf{x}^{(4)} = [-0.9618, -3.9448, -2.9399]$

Where as by Gauss-Seidel method, we have

$\mathbf{x}^{(1)} = [-0.125, -3.225, -2.5875]^T$

$\mathbf{x}^{(2)} = [-0.8516, -3.8878, -2.9349]^T$

$\mathbf{x}^{(3)} = [-0.9778, -3.9825, -2.9901]^T$

$\mathbf{x}^{(4)} = [-0.9966, -3.9973, -2.9985]^T$

E12) Starting with the initial approximation

$\mathbf{x}^{(0)} = [0.5, 0.5, 0.5, 0.5]^T$, we have the following iterates:

$\mathbf{x}^{(1)} = [0.75, 0.625, 0.5625, 0.7813]^T$

$\mathbf{x}^{(2)} = [0.8125, 0.6875, 0.7344, 0.8672]^T$

$$\mathbf{x}^{(3)} = [0.8438, 0.7891, 0.8282, 0.9141]^T$$

$$\mathbf{x}^{(4)} = [0.8946, 0.8614, 0.8878, 0.9439]^T$$

Since the exact solution is $\mathbf{x} = [1, 1, 1, 1]^T$, the Gauss, Seidel method gives better approximation than the Jacobi method at fourth iteration.

UNIT 3 SOLUTION OF NON-LINEAR EQUATI(

Structure **Page Nos.**

2.0 INTRODUCTION

In this unit we will discuss one of the most basic problems in numerical analysis. The problem is called a root-finding problem and consists of finding values of the variable x (real) that satisfy the equation $f(x) = 0$, for a given function f. Let f be a real-value function of a real variable. Any real number α for which $f(\alpha) = 0$ is called a root of that equation or a zero of f. We shall confine our discussion to locating only the real roots of f(x), that is, locating non-real complex roots of $f(x) = 0$ will not be discussed. This is one of the oldest numerical approximation problems. The procedures we will discuss range from the classical Newton-Raphson method developed primarily by Isaac Newton over 300 years ago to methods that were established in the recent past.

Myriads of methods are available for locating zeros of functions and in first section we discuss bisection methods and fixed point method. In the second section, Chord Method for finding roots will be discussed. More specifically, we will take up regula-falsi method (or method of false position), Newton-Raphson method, and secant method. In section 3, we will discuss error analysis for iterative methods or convergence analysis of iterative method.

We shall consider the problem of numerical computation of the real roots of a given equation

$$f(x) = 0$$

which may be algebraic or transcendental. It will be assumed that the function f(x) is continuously differentiable a sufficient number of times. Mostly, we shall confine to simple roots and indicate the iteration function for multiple roots in case of Newton Raphson method.

All the methods for numerical solution of equations discussed here will consist of two steps. First step is about the location of the roots, that is, rough approximate value of the roots are obtained as initial approximation to a root. Second step consists of methods, which improve the rough value of each root.

A method for improvement of the value of a root at a second step usually involves a process of successive approximation of iteration. In such a process of successive approximation a sequence $\{X_n\}$ $n = 0, 1, 2, \ldots$ is generated by the method used starting with the initial approximation x_o of the root α obtained in the first step such that the sequence $\{X_n\}$ converges to α as $n \to \infty$. This x_n is called the nth approximation of nth iterate and it gives a sufficiently accurate value of the root α.

For the first step we need the following theorem:

Theorem 1: If f(x) is continuous in the closed internal [a, b] and f(a) are of opposite signs, then there is at least one real root α of the equation f(x) = 0 such that $a < \alpha < b$.

If further f(x) is differentiable in the open interval (a, b) and either $f'(x) < 0$ or $f'(x) > 0$ in (a, b) then f(x) is strictly monotonic in [a, b] and the root α is unique.

We shall not discuss the case of complex roots, roots of simultaneous equations nor shall we take up cases when all roots are targeted at the same time, in this unit.

2.1 OBJECTIVES

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

- find an approximate real root of the equation f(x) = 0 by various methods;
- know the conditions under which the particular iterative process converges;
- define 'order of convergence' of an iterative method; and

know how fast an iterative method converges.

2.2: SOLUTION OF NONLINEAR EQUATIONS

UNIT 3 Let f(x) be a real-valued function of x defined over a finite interval. We assume it is continuous and differentiable. If f(x) vanishes for some value $x = \alpha$, say, i.e. $f(\alpha) = 0$, then we say $x = \alpha$ is a root of the equation f(x) = 0 or that function f(x) has a zero at $x = \alpha$. We shall discuss methods for finding the roots of an equation f(x) = 0 where f(x) may contain algebraic or transcendental expressions. We shall be interested in real roots only. It is also assumed that the roots are simple (non-repeated) and isolated and well-separated i.e. there is a finite neighbourhood about the root in which no other root exists. All the methods discussed will be iterative type, i.e. we start from an approximate value of the root and improve it by applying the method successively until two values agree within desired accuracy. It is important to note that approximate root is not chosen arbitrarily. Instead, we look for an interval in which only one root lies and choose the initial value suitably in that interval. Usually we have to compute the function values at several points but sometimes we have to get the approximate value graphically close to the exact root.

Method of Successive Substitution (Fixed Point Method)

Suppose we have to find the roots of the equation f(x) = 0. We express it in the form $x = \phi(x)$ and the iterative scheme is given as

$$x_{n+1} = \phi(x_n)$$

where x_n denotes the n^{th} iterated value which is known and x_{n+1} denotes (n + 1)th approximated value which is to be computed. However, f(x) = 0 can be expressed in the form $x = \phi(x)$ in many ways but the corresponding iterative may not converge in all cases to the true value, rather it may diverge start giving absurd values. It can be proved that necessary and sufficient condition for convergence of the scheme is that the modulus of the first derivative of $\phi(x)$ i.e. $\phi'(x)$ at the exact root should be less than 1 i.e. if α is the exact root then $|\phi'(\alpha)| < 1$. But since we do not know the exact root which is to be computed we test the condition for convergence at the initial approximation i.e. $|\phi'(x_0)| < 1$. Hence, it is necessary that the initial approximation should be taken quite close to the exact root and test the condition before starting the iteration. This method is also known as 'fixed point' method since the

mapping $x = \phi(x)$ maps the root α to itself since $\alpha = \phi(\alpha)$ i.e. α remains unchanged (fixed) under the mapping $x = \phi(x)$.

Example

Find the positive root of $x^3 - 2x - 8 = 0$ by method of successive substitution correct upto two places of decimal.

Solution

$f(x) = x^3 - 2x - 8$

To find the approximate location of the root (+ ive) we try to evaluate the function values at different x and tabulate as follows :

x	0	1	2	3	x > 3
f(x)	– 8	– 9	– 4	13	+ ive
Sign of f(x)	–	–	–	+	+

The root lies between 2 and 3. Let us choose the initial approximation as $x_0 = 2.5$.

Let us express f(x) = 0 as $x = \phi(x)$ in the following forms and check whether $|\phi'(\alpha)| < 1$ for x = 2.5.

(i) $x = x^3 - x - 8$

(ii) $x = \frac{1}{2}(x^3 - 8)$

(iii) $x = (2x + 8)^{\frac{1}{3}}$

We see that in cases (i) and (ii) $|\phi'(x)| > 1$, hence we should discard these representations. As the third case satisfies the condition, $|\phi'(x)| = \left| \frac{1}{3(2x + 8)^{\frac{2}{3}}} \right| < 1$ for x = 2.5 we have the iteration scheme as,

$$x_{n+1} = (2x_n + 8)^{\frac{1}{3}}$$

Starting from $x_0 = 2.5$, we get the successive iterates as shown in the table below :

n	0	1	2	3
$\mathbf{x_n}$	2.5	2.35	2.33	2.33

Bisection Method (Method of Halving)

In this method we find an interval in which the root lies and that there is no other root in that interval. Then we keep on narrowing down the interval to half at each successive iteration. We proceed as follows :

(1) Find interval $I = (x_1, x_2)$ in which the root of f(x) = 0 lies and that there is no other root in I.

(2) Bisect the interval at $x = \frac{x_1 + x_2}{2}$ and compute f(x). If | f(x) | is less than the desired accuracy then it is the root of f(x) = 0.

(3) Otherwise check sign of f(x). If sign {f(x)} = sign {f(x_2)} then root lies in the interval [x_1, x] and if they are of opposite signs then the root lies in the interval [x, x_2]. Change x to x_2 or x_1 accordingly. We may test sign of f(x) × f(x_2) for same sign or opposite signs.

(4) Check the length of interval | $x_1 - x_2$ |. If an accuracy of say, two decimal places is required then stop the process when the length of the interval is 0.005 or less. We may take the midvalue $x = \frac{x_1 + x_2}{2}$ as the root of f(x) = 0. The convergence of this method is very slow in the beginning.

Example

Find the positive root of the equation $x^3 + 4x^2 - 10 = 0$ by bisection method correct upto two places of decimal.

Solution

$f(x) \equiv x^3 + 4x^2 - 10 = 0$

Let us find location of the + ive roots.

x	0	1	2	> 2
f(x)	– 10	– 5	14	
Sign f(x)	–	–	+	+

There is only one + ive root and it lies between 1 and 2. Let $x_1 = 1$ and $x_2 = 2$; at x = 1, f(x) is – ive and at x = 2, f(x) is + ive. We examine the sign of f(x) at $x = \frac{x_1 + x_2}{2} = 1.5$ and check whether the root lies in the interval (1, 1.5) or (1.5, 2). Let us show the computations in the table below :

Iteration No. ————

Block 2 INTERPOLATION

Structure **Page Nos.**

2.2 operators

Let us suppose that values of a function $y = f(x)$ are known at $(n + 1)$ points, say $x = x_i$, $i = 0(1)\ n$ and that $x_0 < x_1 < x_2 \ldots < x_n$. The function $f(x)$ may or may not be known explicitly. That is, we are given $(n + 1)$ pair of values (x_0, y_0), (x_1, y_1), . . . (x_n, y_n) and the problem is to find the value of y at some intermediate point x in the interval $[x_o, x_n]$. The process of computing/determining this value is known as 'interpolation'. There are several methods of interpolation when the abscissas x_i, $i = 0(1)_n$, known as tabular points, are equidistant i.e. $x_i – x_{i-1} = h$, $i = 1(1)n$ is same throughout. Before discussing these methods we need to introduce some operators.

Unit 1 : Operators

i) Forward difference (FD) operator: Δ(Delta)

$\Delta f(x) = f(x+h) – f(x)$

ii) Backward Difference (BD) operator : ∇ (Inverted Delta)

$\nabla f(x) = f(x) – f(x – h)$

iii) Central Difference (CD) operator : δ (Small Delta)

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

iv) Averaging operator : μ (mew)

$$\mu f(x) = \frac{1}{2}\left[f\left(x + \frac{1}{2}h\right) + f\left(x - \frac{1}{2}h\right)\right]$$

v) Shift operator : E

$Ef(x) = f(x+h)$

vi) Differential Operator : D

$$Df(x) = \frac{d}{dx} f(x)$$

The operators can be applied repeatedly. For example, in case of FD operator we may have,

$$Df(x) = f(x+h) - f(x)$$

$$D^2 f(x) = DDf(x) = D\{f(x+h) - f(x)\}$$

$$Df(x+h) - Df(x) = f(x+2h) - 2f(x+h) + f(x), \text{ etc.}$$

Similarly we can express BD and CD respectively as follows:

$$\nabla^2 f(x) = \nabla\nabla f(x) = f(x) - 2f(x-h) + f(x-2h), \text{ etc.}$$

$$d^2 f(x) = d.df(x) = f(x-h) - 2f(x) + f(x+h), \text{ etc.}$$

In case of shift operator we can have,

$E^p f(x) = f(x+ph)$

where p may be +ve or –ve and integer or fractional.

Interrelation between Operators

The operators are interrelated as one operator can be expressed in terms of the other. For example, Δ, ∇ and δ can be expressed in terms of E as follows:

$\Delta f(x) = f(x+h) - f(x) = (E-1) f(x)$ or $\Delta \equiv E - 1$

$$\nabla f(x) = f(x) - f(x-h) = \left(1 - E^{-1}\right) f(x) \text{ or } \nabla \equiv 1 - E^{-1}$$

$$df(x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right) = \left(E^{\frac{1}{2}} - E^{-\frac{1}{2}}\right) f(x) \text{ or } \delta \equiv E^{\frac{1}{2}} - E^{-\frac{1}{2}}$$

We can derive other relationships among various operators as shown in Table 1.

Table 1: Interrelations between various operators

Application of Δ on polynomial functions

i) $f(x) = c$, a constant

$\Delta f(x) = \Delta . cx^0 = c\Delta x^0 = c\{(x+h)^0 - x^0\} = c(1-1) = 0$

ii) $f(x) = x$

$Df(x) = Dx = (x+h) - x = h$

$D^2 f(x) = Dh = 0$

iii) $f(x) = x^2$

$Dx^2 = (x+h)^2 - x^2 = 2hx + h^2$

$D^2 x^2 = 2h.Dx + Dh^2 = 2h^2$

$D^3 x^2 = 0.$

Proceeding in this manner it can be shown that if $f(x) = x^n$ then $D^n x^n = n!h^n$ and $D^{n+1}x^n = 0$. It can be further extended when f(x) is a polynomial of degree n, say

$f(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + ... + a_n$. Then $D^n f(x) = a_0 D^n x^n + a_1 . D^n x^{n-1} + a_2 D^n . x^{n-2} + ... + a_n D^n . 1$

$= a_0 . n! h^n$, as rest of differences will be zero.

Similar results hold for BD and CD operators.

Unit 2; Interpolation with equal interval

Difference Table

A difference table is central to all interpolation formulas (see Table 2). Suppose we are given the data (x_i, y_i), i = 0,1,2… where $x_{i+1} - x_i = h$ or $x_{i+1} = x_i + h$ and y_i represents value of y corresponding to $x = x_i = x_0 + ih$. We can express FD of y_i as follows,

$Dy_i = y_{i+1} - y_i$; $D^2 y_i = y_{i+2} - 2y_{i+1} + y_i$ etc.

It will be convenient if we express Δ in terms of E i.e.

$\Delta \equiv E - 1$. In that case,

$Dy_i = (E - 1)y_i = Ey_i - y_i = y_{i+1} - y_i$

$D^2 y_i = (E - 1)^2 y_i = (E^2 - 2E + 1)y_i = y_{i+2} - 2y_i + y_i$, etc

We can obtain similar expressions for BD and CD. In Table 2, values of x_i with corresponding value of y_i are tabulated vertically.

Table 2: Difference Table

i	x_i	y_i	1st diff.	2nd diff.	3rd diff.	4th diff.
0	–1	6.5625				
			7.5000			
1	0	14.0625		–15.0000		
			–7.5000		12.0000	
2	1	6.5625		–3.0000		24.0000
			–10.5000		36.0000	
3	2	–3.9375		33.0000		24.0000
			22.5000		60.0000	
4	3	18.5625		93.0000		
			115.5000			
5	4	134.0625				

Note: Above values are taken from $y = x^4 - 8.5x^2 + 14.0625$. Note that fourth differences are constant.

In the next column first differences, Δy_i, i = 0(1)4 are computed and placed as shown in the table. Then second differences $D^2 y_i = D(Dy_i)$, i = 0(1)3, third differences $D^3 y_i = D\left(D^2 y_i\right)$, i = 0(1)2 and finally fourth differences $D^4 y_i = D(D^3 y_i), i = 0,1$ are computed.

It is important to note that the difference table is always made in the above manner while the differences may be expressed by FD, BD or CD as required. That is, in the column of first differences we have ,

$7.5000 = y_1 - y_0 = Dy_0 = \tilde{N} y_1 = dy_{\frac{1}{2}}$

$-7.5000 = y_2 - y_1 = Dy_1 = \tilde{N} y_2 = dy_{\frac{3}{2}}$

$-15.0000 = Dy_1 - Dy_0 = D(Dy_0) = D^2 y_0$

$= \tilde{N} y_2 - \tilde{N} y_1 = \tilde{N}(\tilde{N} y_2) = \tilde{N}^2 y_2$

$= dy_{\frac{3}{2}} - dy_{\frac{1}{2}} = d(dY_I) = d^2 y_i$, etc.

Hence the differences in a difference table appear as shown in Table 3, Table 4 and Table 5.

Table 3: Differences express in forward difference notation

i	x_i	y_i	1st diff.	2nd diff.	3rd diff.	4th diff.
0	x_0	y_0				
			Dy_0			
1	x_1	y_1		$D^2 y_0$		
			Dy_1		$D^3 y_0$	
2	x_2	y_2		$D^2 y_1$		$D^4 y_0$

			Dy_2		D^3y_1	
3	x_3	y_3		D^2y_2		
			Dy_3			
4	x_4	y_4				

Table 4: Differences expressed in backward difference notation

i	$\mathbf{x_i}$	$\mathbf{y_i}$	**1st diff.**	**2nd diff.**	**3rd diff.**	**4th diff.**
0	x_0	y_0				
			$\tilde{N}y_1$			
1	x_1	y_1		$\tilde{N}^2y_2$		
			$\tilde{N}y_2$		$\tilde{N}^3y_3$	
2	x_2	y_2		$\tilde{N}^2y_3$		$\tilde{N}^4y_4$
			$\tilde{N}y_3$		$\tilde{N}^3y_4$	
3	x_3	y_3		$\tilde{N}^2y_4$		
			$\tilde{N}y_4$			
4	x_4	y_4				

Table 5: Differences expressed in central difference notation

i	$\mathbf{x_i}$	$\mathbf{y_i}$	**1st diff.**	**2nd diff.**	**3rd diff.**	**4th diff.**
0	x_0	y_0				
			$dy_{\frac{1}{2}}$			
1	x_1	y_1		d^2y_1		
			$dy_{\frac{3}{2}}$		$d^3y_{\frac{3}{2}}$	
2	x_2	y_2		d^2y_2		d^4y_2
			$dy_{\frac{5}{2}}$		$d^3y_{\frac{5}{2}}$	
3	x_3	y_3		d^2y_3		
			$dy_{\frac{7}{2}}$			
4	x_4	y_4				

It must be observed that in Table 3 y_0 and its differences appear sloping downwards, in Table 4, y_4 and its differences appear in an upward slope while in Table 5, y_i and its even differences appear in a straight line (see y_2).

Interpolation Formulas

We shall now discuss some interpolation formulas using FD, BD and CD. Let us suppose we have to compute the value of y corresponding to $x = x_p$, denoted by y_p. We have $x_p = x_0 + ph$ or $p = \frac{x_p - x_0}{h}$ and $y_p = E^p y_0$. Expressing E in terms of D or $\tilde{N}$ we get FD and BD interpolation formulas respectively. We choose x_0 so as to include more terms in the formula.

i)

Unit2 : Numerical Integration

We shall be interested in evaluating the definite integral $I = \int_a^b f(x)\,dx$ where the function f(x) is defined at each point in [a, b] and there is no discontinuity. We also assume that f(x) possess same sign in [a, b]. It should be remembered that $\int_a^b f(x)\,dx$ represents the area enclosed by the curve y = f(x), the x-axis y = 0 and the vertical lines x = a and x = b. To compute the area numerically is called 'quadrature' in engineering parlance.

The first step for evaluating the integral is to divide the interval [a, b] into suitable number of sub-intervals, say n, each of width h, such that $h = \frac{b-a}{n}$. Let us denote these points of division on x-axis as $a = x_0 < x_1 < x_2 \ldots < x_n = b$ so that n sub-intervals may be defined as $[x_i, x_{i+1}]$, $i = 0(1)\, n-1$. Let the function values $f(x_i)$, $i = 0\,(1)\,n$ be known.

Various integration formulas are devised by approximating the integral over one interval or two intervals or three intervals or in general k intervals at a time and then summing them up over the entire interval [a, b]. Obviously n has to be chosen as multiple of k, say mk = n, m is an integer. That is, if a formula involving k intervals is used then it will be invoked m times to cover the interval

[a, b], where $m = \frac{n}{kh}$. We shall now discuss formulas when k = 1 and 2.

(i) Rectangular Rule/Formula

We approximate the integral over an interval $[x_i, x_{i+1}]$ as,

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = (x_{i+1} - x_i)\, f(x_i) = h\, f(x_i), \quad i = 0, 1, 2, \ldots, n-1$$

Adding over all the intervals and denoting $f(x_i) = f_i = y_i$ the formula may be written as,

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_n} y(x)\,dx = h\,\{y_0 + y_1 + y_2 + \ldots + y_{n-1}\}$$

Geometrically, the integral in each interval is represented by the area of a rectangle (see Figure 1).

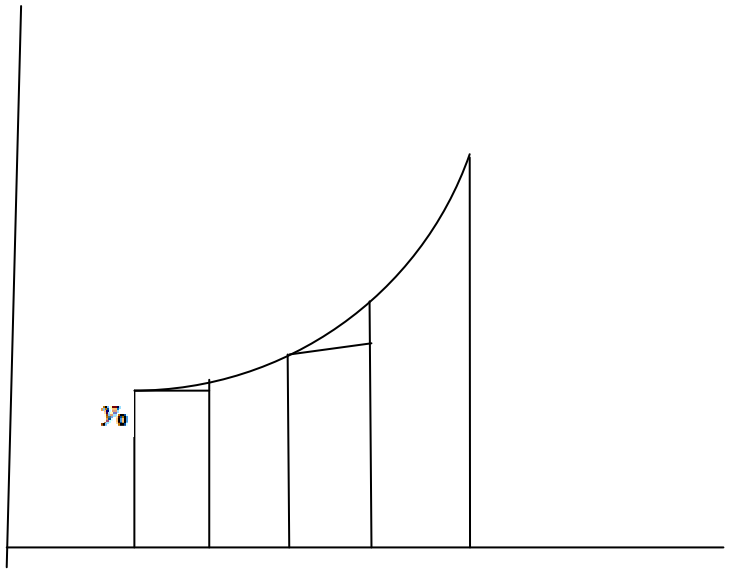

Figure 1 : Rectangular Rule

(ii) Trapezoidal Rule/Formula

We approximate the function f(x) in the interval $[x_i, x_{i+1}]$ by a straight line joining the points (x_i, y_i) and $[x_{i+1}, y_{i+1}]$. For convenience let us consider the integral in the first interval $[x_0, x_1]$. The line joining (x_0, y_0) and (x_1, y_1) may be written from the Lagrange's formula as,

$$y(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

Now, $$\int_{x_0}^{x_1} f(x)\, dx = \int_{x_0}^{x_i} \left(\frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 \right) dx$$

$$= \frac{1}{2h} \left[-(x - x_1)^2 y_0 + (x - x_0)^2 y_1 \right]_{x = x_0}^{x = x_1}$$

$$= \frac{h}{2} (y_0 + y_1)$$

Adding over all the intervals the Trapezoidal formula/rule may be written as,

$$\int_a^b f(x)\, dx = \int_{x_0}^{x_n} y(x)\, dx = \frac{h}{2} \{(y_0 + y_1) + (y_1 + y_2) + \ldots + (y_{n-2} + y_{n-1}) + (y_{n-1} + y_n)\}$$

$$= \frac{h}{2} \{y_0 + 2(y_1 + y_2 + \ldots + y_{n-1}) y_n\}$$

or, $$= h \left\{ \frac{y_0 + y_n}{2} + (y_1 + y_2 + \ldots + y_{n-1}) \right\}$$

Geometrically, the integral in an interval is approximated by the area of a trapezium (see Figure 2).

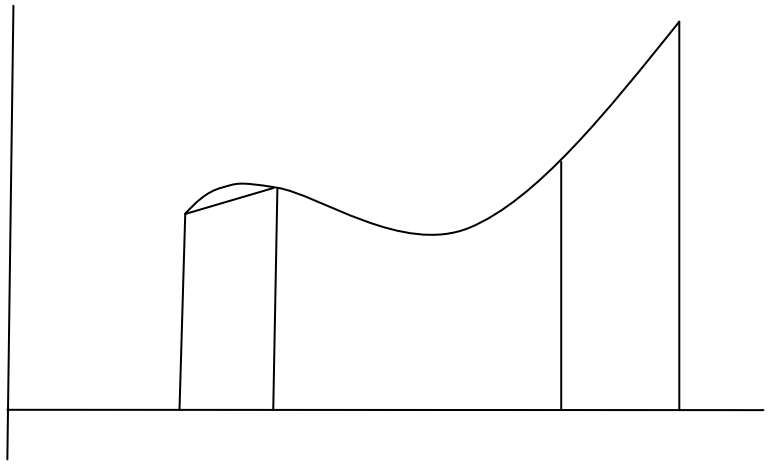

Figure 2 : Trapezoidal Rule

(iii) Simpson's 1/3rd Rule/Formula

In this case the integral is evaluated over two intervals at a time, say $[x_0, x_1]$ and $[x_1, x_2]$. The function f(x) is approximated by a quadratic passing through the points (x_0, y_0) and (x_1, y_1) and
(x_2, y_2). From Lagrange's formula we may write the quadratic as,

$$y(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2$$

Integrating term by term we get,

$$\int_{x_0}^{x_2} \frac{(x - x_1)(x - x_2)}{(-h)(-2h)} dx = \frac{1}{2h^2}\left[(x - x_1)\frac{(x - x_2)^2}{2} - \frac{(x - x_2)^3}{6}\right]_{x_0}^{x_2} = \frac{h}{3}$$

$$\int_{x_0}^{x_2} \frac{(x - x_0)(x - x_2)}{h(-h)} dx = \frac{1}{h^2}\left[(x - x_0)\frac{(x - x_2)^2}{2} - \frac{(x - x_2)^3}{6}\right]_{x_0}^{x_2} = \frac{4}{3}h$$

$$\int_{x_0}^{x_2} \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} dx = \frac{1}{2h^2}\left[(x - x_0)\frac{(x - x_1)^2}{2} - \frac{(x - x_1)^3}{6}\right]_{x_0}^{x_2} = \frac{h}{3}$$

Hence we get,

$$\int_{x_0}^{x_2} f(x)\,dx = \int_{x_0}^{x_2} y(x)\,dx = \frac{h}{3}y_0 + \frac{4h}{3}y_1 + \frac{h}{3}y_2$$

$$= \frac{h}{3}(y_0 + 4y_1 + y_2)$$

Applying this formula over next two intervals and then next two and so on for $\frac{n}{2}$ times and adding we get

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_n} y(x)\,dx = \int_{x_0}^{x_2} y(x)\,dx = \int_{x_2}^{x_4} y(x)\,dx + \ldots + \int_{x_{n-2}}^{x_n} y(x)\,dx$$

$$= \frac{h}{3}[(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \ldots + (y_{n-2} + 4y_{n-1} + y_n)]$$

$$= \frac{h}{3}[y_0 + y_n + 4(y_1 + y_3 + \ldots + y_{n-1}) + 2(y_2 + y_4 + \ldots + y_{n-2})]$$

Obviously n should be chosen as a multiple of 2 i.e. an even number for applying this formula.

Example

Evaluate the integral $I = \int_0^1 \frac{dx}{\sqrt{1 + x^2}}$ by trapezoidal rule dividing the interval [0, 1] into five equal parts. Compute upto five decimals.

Solution

$$n = 5;\ h = \frac{1-0}{5} = 0.2$$

i	0	1	2	3	4	5
x	0	0.2	0.4	0.6	0.8	1.0

$$\sqrt{\qquad}$$

Unit-1 Block3 : Differentiation, Integration and Differential Equations

Unit1 : Numerical Differentiation

Let us suppose that (n + 1) function values $y_i = f(x)$ are given for $x = x_i$, $i = 0\ (1)_n$ and that $x_1 < x_2 \ldots < x_n$ and $x_i - x_{i-1} = h$, $i = 1\ (1)_n$.

Assuming that f(x) is differentiable in the interval $1 = [x_0, x_n]$ we can compute the derivatives of f(x) at any point x in l by differentiating the interpolation formulas. Since the formulas are expressed in terms of p, let us note the following relation,

For $x = x_p,\ x_p = x_0 + ph$

gives $\dfrac{dp}{dx} = \dfrac{1}{h}$ at $x = x_p$

Hence at $x = x_p,\ \dfrac{dy}{dx} = \dfrac{dy}{dx} \cdot \dfrac{dp}{dx} = \dfrac{1}{h}\dfrac{dy}{dp}$

and $\dfrac{d^2y}{dx^2} = \dfrac{1}{h^2} \cdot \dfrac{d^2y}{dp^2}$, etc.

(i) Differentiating FD interpolation formula we get $(0 \le p < 1)$,

$$\frac{dy}{dx} = \frac{1}{h}\left\{\Delta y_0 + \frac{2p-1}{2}\Delta^2 y_0 + \frac{3p^2 - 6p + 2}{6}\Delta^3 y_0 + \ldots\right\}$$

$$\frac{d^2y}{dx^2} = \frac{1}{h^2}\{\Delta^2 y_0 + (p-1)\Delta^3 y_0 + \ldots\}$$

(ii) Differentiating BD formula we get $(-1 < p \le 0)$

$$\frac{dy}{dx} = \frac{1}{h}\left\{\nabla y_0 + \frac{2p+1}{1}\nabla^2 y_0 + \frac{3p^2 + 6p + 2}{6}\nabla^3 y_0 + \ldots\right\}$$

$$\frac{d^2y}{dx^2} = \frac{1}{h^2}\{\nabla^2 y_0 + (p+1)\nabla^3 y_0 + \ldots\}$$

(iii) Differentiation of Stirling's formula gives $(-0.5 \le p \le 0.5)$,

$$\frac{dy}{dx} = \frac{1}{h}\left\{\mu\delta y_0 + p\delta^2 y_0 + \frac{3p^2 - 1}{6}\mu\delta^3 y_0 + \frac{2p^3 - p}{12}\delta^4 y_0 + \ldots\right\}$$

$$\frac{d^2y}{dx^2} = \frac{1}{h^2}\left\{\delta^2 y_0 + p\,\mu\delta^3 y_0 + \frac{6p^2 - 1}{12}\delta^4 y_0 + \ldots\right\}$$

(iv) By differentiating Bessel's formula we get $(0.25 \le p \le 0.75)$

$$\frac{dy}{dx} = \frac{1}{h}\left\{\delta y_{\frac{1}{2}} + \frac{2p-1}{2}\delta^2 y_{\frac{1}{2}} + \frac{6p^2 - 6p + 1}{12}\delta^3 y_{\frac{1}{2}} + \frac{2p^3 - 3p^2 - p + 1}{12}\mu\delta^4 y_{\frac{1}{2}} + \ldots\right\}$$

$$\frac{d^2y}{dx^2} = \frac{1}{h^2}\left\{\mu\delta^2 y_{\frac{1}{2}} + \frac{2p-1}{2}\delta^3 y_{\frac{1}{2}} + \frac{6p^2 - 6p - 1}{12}\mu\delta^4 y_{\frac{1}{2}} + \ldots\right\}$$

It must be remembered, as already indicated, that appropriate formula should be used i.e. FD formula at the upper end of the table, BD formula near the lower end of the table and CD formula in the middle

of the table. Further, the point $x = x_0$ has to be chosen suitably according to the formula used. It may also be noted that to find a derivative at the tabular point $x = x_i$, $i = 0\ (1)_n$ the value of $p = 0$.

It may also be mentioned that in most of the cases we do not go beyond second or third differences in a formula for computing derivatives.

Example

The values of $y = \sqrt{x}$ are given below for $x = 1.5\ (0.5)\ 3.5$.

x	1.5	2.0	2.5	3.0	3.5
$\sqrt{\ }$					