**Interpolation:**

The process of estimating the value of y, for any intermediate value of x, is called interpolation.

**Extrapolation:**

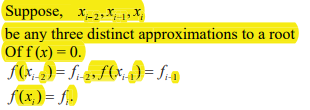
The method of computing the value of y, for a given value of x, lying outside the table of values of x is known as extrapolation.

**Muller’s Method:**

In Muller’s method, f (x) = 0 is approximated by a second degree polynomial; that is by a quadratic equation that fits through three points in the vicinity of a root. The roots of this quadratic equation are then approximated to the roots of the equation f (x) 0.This method is iterative in nature and does not require the evaluation of derivatives as in Newton.

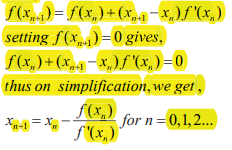
Raphson method. This method can also be used to determine both real and complex roots

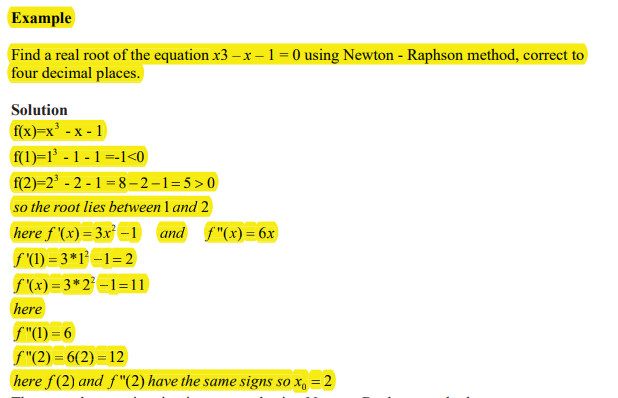
of f(x) = 0.



**Newton -Raphson Method**

This method is one of the most powerful method and well known methods, used for finding a root of f(x)=0 the formula many be derived in many ways the simplest way to derive this formula is by using the first two terms in Taylor’s series expansion of the form,

****

****

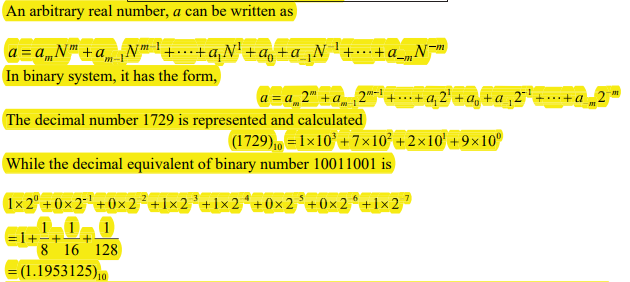
**Numerical Analysis:**

Introduction We begin this chapter with some of the basic concept of representation of numbers on computers and errors introduced during computation. Problem solving using computers and the steps involved are also discussed in brief.

***Number (s) System (s)***

In our daily life, we use numbers based on the decimal system. In this system, we use ten symbols 0, 1,…,9 and the number 10 is called the base of the system. Thus, when a base N is given, we need N different symbols 0, 1, 2, …,(N – 1) to represent an arbitrary number. The number systems commonly used in computers are

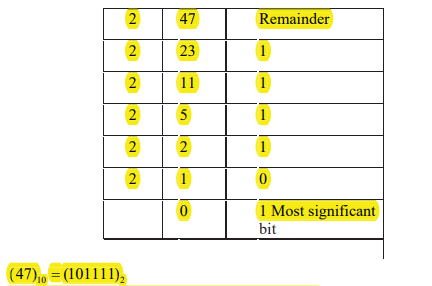
|  |  |
| --- | --- |
| Base, n | number |
| **2** | **Binary** |
| **8** | **Octal** |
| **10** | **Decimal** |
| **16** | **Hexadecimal** |



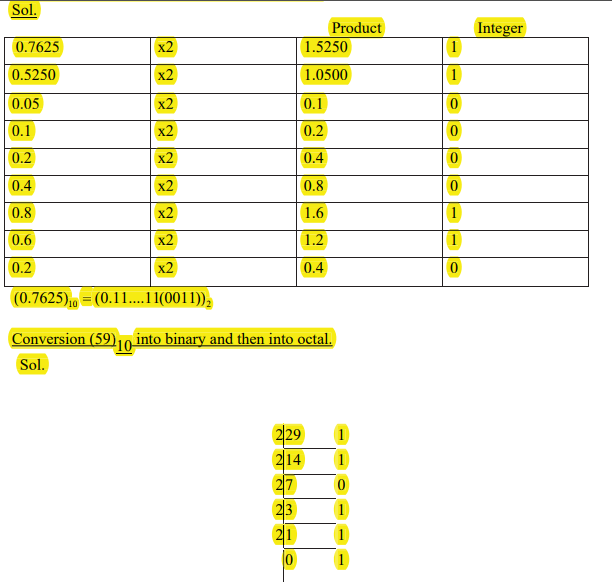
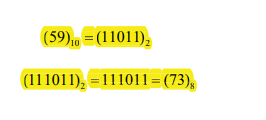
Electronic computers use binary sy p stem whose base is 2. The two symbols used in this system are 0 and 1, which are called binary digits or simply bits. The internal representation of any data within a computer is in binary form. However, we prefer , data input and output of numerical results in decimal system. Within the computer, the arithmetic is carried out in binary form.

**Q) Conversion of decimal number 47 into its binary equivalent.**

**Sol)**



**Q) Binary equivalent of the decimal fraction 0.7625.**



**Errors in Computations:**

Numerically, computed solutions are subject to certain errors. It may be fruitful to identify the error sources and their growth while classifying the errors in numerical computation. These are

***Inherent errors,***

***Local round-off errors***

***Local truncation errors***

***Inherent errors:*** It is that quantity of error which is present in the statement of the problem itself, before finding its solution. It arises due to the simplified assumptions made in the mathematical modeling of a problem. It can also arise when the data is obtained from certain physical measurements of the parameters of the problem.

***Local round-off errors:*** Every computer has a finite word length and therefore it is possible to store only a fixed number of digits of a given input number. Since computers store information in binary form, storing an exact decimal number in its binary form into the computer memory gives an error. This error is computer dependent. At the end of computation of a particular problem, the final results in the computer, which is obviously in binary form, should be converted into decimal form-a form understandable to the user-before their print out. Therefore, an additional error is committed at this stage too. This error is called local round-off error.

(0.7625)10 = (0.110000110011)2

If a particular computer system has a word length of 12 bits only, then the decimal number 0.7625 is stored in the computer memory in binary form as 0.110000110011. However, it is equivalent to 0.76245. Thus, in storing the number 0.7625, we have committed an error equal to 0.00005, which is the round-off error; inherent with the computer system considered. Thus, we define the error as Error = True value – Computed value Absolute error, denoted by |Error|,

While, the relative error is defined as

Relative error= |Error| \ |True value|

**Local truncation error:** It is generally easier to expand a function into a power series using Taylor series expansion and evaluate it by retaining the first few terms. For example, we may approximate the function f (x) = cos x by the series

