Institute of Information Technology

Subject: Numerical Techniques Laboratory (**ICT – 2106)**

Exp. No.-1

Name of the Exp.: Errors in Numerical Techniques: Calculation of Truncation Error in a Series Approximation.

Introduction:

In Numerical Analysis, we usually come across two types of errors.

i) Inherent Errors: Most numerical computations are inexact, either due to the given data being approximate, or due to the limitations of the computing aids: mathematical tables, desk calculators or the digital computers. All these errors comprise inherent errors. Again, we come across numbers with large number of digits after decimal and we need to cut them to a usable number. This process is called *Rounding off.*

Inherent Errors can be minimized by obtaining better data, correcting obvious errors in the data, using computer aid of higher precision and carrying the computations to more significant figures at each step of the computation in case of hand calculations. *Round off error* can be reduced by carrying the computation to more significant digits at each step of the computation. At each step of the computation retain at least one more significant figure than that given in the data and then round off the result at the end.

ii) Truncation Errors. These are errors caused by using approximate formulae in computations –such as the one that arises when a function f(x) is evaluated from an infinite series for *x* after truncating it at a certain stage

Truncation error can be minimized by increasing the number of terms in the series approximation.

It is desirable to make all these errors as small as possible**.**

Objective of the Experiment:

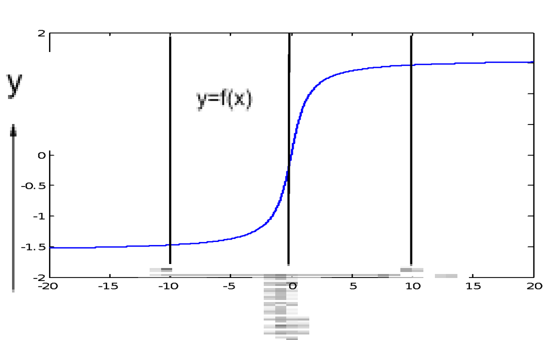
- To write a program to calculate the truncation error of a Maclaurin's series expansion of f(x) for a given length, n. and observe that the error can be made small by increasing the length of the series.
- To write a program in order to find out the maximum error of Maclaurin's series of n terms using Remainder term Rn(x).
- To write a program in order to find out the required number of term, n of a series expansion for a given accuracy.

Theory:

If f(x) is continuous and possesses continuous derivatives of order n in an interval a<x<b that includes $x=\xi$ **.** then in that interval

Taylor's series for f(x) at b=x is given by

$$y = f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \ldots\ldots\frac{(x-a)^{n-1}}{(n-1)!}f^{(n-1)}(a) + R_n(x)$$



where Rn(x) is the remainder term, can be expressed in the form

$$R_n(x) = \frac{(x-a)^n}{n!}f^n(\xi), a < \xi < x$$

and $f^n(\xi)$ is the n times differentiation of $f(x)$ at $x=\xi$ **.** Putting a=0, we get the Maclaurin's series Expansion of f(x) like following,

$$y = f(x) = f(0) + (x)f'(0) + \frac{(x)^2}{2!}f''(0) + \ldots\ldots\frac{(x)^{n-1}}{(n-1)!}f^{(n-1)}(a) + R_n(x)$$

And Rn(x) becomes $R_n(x) = \frac{(x)^n}{n!} f^n(\xi), 0 < \xi < x$

The truncation error of the Maclaurin's series expansion of a function can be evaluated in the following two ways:

1.

The truncation error committed in a Maclaurin's series approximation after n terms, can be evaluated by using the remainder term Rn(x).

i.e. Maximum Absolute truncation error= $R_n(x) = \frac{(x)^n}{n!} f^n(\xi), 0 < \xi < x$

2.

Again, If the approximated value of f(x), Va, is calculated by summing the first n terms of the series expansion and the actual value, V_t, can be determined by using program function. Then the absolute truncation error, Ea can be calculated as

Absolute truncation error $E_A = V_t - V_A$.

So, Relative error, $E_R = \frac{E_A}{V_t}$

Now if the accuracy required is specified in advance, then it would be possible to find n, the number of terms of the series expansion.

Problems:

1. Write a program to calculate the approximated value of f(x)=cos(x) upto first 5 terms at x=1 (radian). Find out the absolute and relative errors. What happens to the error when number of terms is increased from 5 to 12?

(Hints: for true value of cos(x) use the function cos(x) of the program, for approximate value of cos(x), use series expansion of cos(x). Absolute error, E_A=abs(True value-approximate value), Relative $E_R = E_A$ /true value

2. Write a program to find out the number of terms N of the series f(x)=cos(x) in series expansion, such that their sum gives the value of f(x) correct to 8 decimal point. When 0<x<1. What is the value of error at this point?

(Hints: For the value of error correct to m decimal point, Relative error, $E_R < \frac{1}{2}.10^{-m}$)

3. Find out the maximum error in the series expansion of cos(x)using Remainder term. for the value of n, obtained in Ques. No. 2, at x=1, which gives the value of f(x) correct to 8 decimal point. And compare it with the absolute error obtained in Ques. No. 2 Discuss why the value of error from Ques. No. 2 and value of maximum error from Ques No.3 is slightly different?

   (Hints: Use Remainder term to calculate maximum error. Here n=2*N+1 when N is the number of terms of Maclaurin's Series of cos (x.),

   To find out n times differentiation of cos(x) in order to find out Maximum error use

   a)the expression cos''(x)=cos(n*pi/2+x) or b) the Matlab 7 functions like following :

   syms x %define f(x)as an analytic or symbolic expression

   f=cos(x)

   D=diff(f,n) %returns analytical result of n times differentiation of analytic function f(x) in D.

   So, the result of D will give the analytic expression of n times differentiation of cos(x).

4. Write a program to find out the number of terms n of the series f(x)=cos(x), without series expansion such that their sum gives the value of f(x) correct to 8 decimal point, when x=1.(Hints: Use the Remainder term. to calculate maximum error).
5. Comment and discuss on the results of the program.

Reference Book:

Introductory Methods of Numerical Analysis - S. S. Sastry; Prentice Hall of India.

Some sample codes

```
x=1.2; %value at which error has to be determined
f=@(x) exp(x); %defining the function
true_derivative=exp(x);%exact derivative result
for i=1:20
h(i,1)=0+.01*i;
n_d(i,1)=(f(x+h(i,1))-f(x))/h(i,1);%numerical derivative
```

```
e(i,1)=n_d(i,1)-true_derivative; %error
err(i,1)=abs(e(i,1)); %absolute error
end
plot(h,err);
```