DEEN DAYAL UPADHYAYA COLLEGE

NUMERICAL METHODS

Practical File

Semester-VI

Submitted by:-

Dheeraj

B.Sc. Mathematical Sciences

Roll no. -21MTS5712

Submitted to:-

Mrs. Rashmi Gupta

INDEX

Serial No. (Practical no.)	Date	Practical name
1.	30/01/2024	Bisection Method.
2.	06/02/2024 09/02/2024	Secant Method Regula-Falsi Method.
3.	13/03/2024	Newton Raphson Method
4.	13/02/2024	Gauss Elimination Method.
5.	20/02/2024 23/02/2024	Gauss-Jacobi Method Gauss-Seidel Method.
6.	27/02/2024 05/03/2024	Lagrange Interpolation Newton Interpolation
7.	19/03/2024	Trapezoidal rule

% DHEERAJ%   %
%21 MTS5712 %
% Date - 30 / 01 / 2024 %

% Practical - 1 Bisection Method with conditions of convergenc %
% Aim - To perform the bisection method for the functions f1 (x) = x^3 + 2 * x^2 - 3 * x - 1, f2 (x) = x^3 + 2 * x^2 - 3 * x - 3 and f3 (x) = sin x on the intervals [1, 2], [1, 2] and [3, 4] respectively within an absolute convergence of 10^-7 %

In[37]:=
```
Bisection[a0_, b0_, m_] := Module[{a = N[a0], b = N[b0]}, c = (a + b) / 2;
   k = 0;
   While[k < m && ((b - a) / 2) > 10^(-7), If[Sign[f[b]] == Sign[f[c]], b = c, a = c];
    c = (a + b) / 2;
    k = k + 1;];
   Print["c=", NumberForm[c, 16]];
   Print["f[c]=", NumberForm[f[c], 16]];];
```

In[38]:=
```
f[x_] = x^3 + 2 * x^2 - 3 * x - 1;
Bisection[1, 2, 30]
```

c=1.198691189289093

f[c]=$-3.310740535056311\times10^{-7}$

In[40]:=
```
f[x_] = x^3 + x^2 - 3 * x - 3;
Bisection[1, 2, 30]
```

c=1.732050836086273

f[c]=$2.698915366750043\times10^{-7}$

In[42]:=
```
f[x_] = Sin[x];
Bisection[3, 4, 20]
```

c=3.141592502593994

f[c]=$1.50995799097837\times10^{-7}$

% DHEERAJ
%21 MTS5712 %
% Date - 06 / 02 / 2024 %

% Practical - 2 Secant Method %
% Aim - To perform the iterations of Secant Method for the functions f1 (x) = x^3 + 2 * x - 5, f2 (x) = cos x - x and f3 (x) = sin x on the intervals [1, 2], [0, 1] and [0, 1] respectively within an absolute convergence of 5 * 10^-7 %

In[1]:=
```
SecantMethod[x0_, x1_, max_] := Module[{}, k = 1; p0 = N[x0];
  p1 = N[x1];
  p2 = p1;
  p1 = p0;
  While[(k < max && Abs[f[p2]] > 0.0000005),
   p0 = p1;
   p1 = p2;
   p2 = p1 - (f[p1] (p1 - p0) / (f[p1] - f[p0]));
   k = k + 1;];
  Print["p", k, "=", NumberForm[p2, 11]];
  Print["f[p", k, "]=", NumberForm[f[p2], 11`]];]
f[x_] := x^3 - 2 * x - 5;
SecantMethod[3, 2, 50]
```

p6=2.0945514815

NumberForm: Formatting specification 11. should be a positive integer or a pair of positive integers.

f[p6]=1.18847×10^{-11}

In[4]:=
```
f[x_] := x^3 - 2 * x - 5;
SecantMethod[1, 2, 50]
```

p6=2.0945514814

NumberForm: Formatting specification 11. should be a positive integer or a pair of positive integers.

f[p6]=-2.00905×10^{-9}

In[6]:=
```
f[x_] := Cos[x] - x;
SecantMethod[0, 1, 50]
```

p5=0.73908511213

NumberForm: Formatting specification 11. should be a positive integer or a pair of positive integers.

f[p5]=3.52926×10^{-8}

In[8]:=
```
f[x_] := Sin[x];
SecantMethod[0, 1, 50]
```

p2=0.

NumberForm: Formatting specification 11. should be a positive integer or a pair of positive integers.

f[p2]=0.

In[10]:=
```
f[x_] := Sin[x];
SecantMethod[3, 4, 50]
```

p4=3.141592728

NumberForm: Formatting specification 11. should be a positive integer or a pair of positive integers.

f[p4]=-7.43951×10^{-8}

% DHEERAJ
%21 MTS5712 %
% Date - 09 / 02 / 2024 %
% Pratical 2 - Regula Falsi method %
Aim - To perform thre iterations of REgula Falsi method for the functions f1 (x) = x^3 + 2 * x^2 - 3 * x - 1,
f2 (x) = x^3 + 2 x - 1 and f (x) = e^-x on the intervals [1, 2], [0, 1] and [0, 1] respectively withinn an absolute convergence of 10^- (12)

In[1]:=
```
Regulafalsi[a0_, b0_, m_] := Module[{}, a = N[a0]; b = N[b0];
  If[f[a] * f[b] > 0, Print["interval is not correct"]; Break[],
   c = (a * f[b] - b * f[a]) / (f[b] - f[a]);
   k = 0;
   While[(k < m && Abs[f[c]] > 10^(-12)), If[Sign[f[b]] == Sign[f[c]], b = c, a = c;];
        c = (a * f[b] - b * f[a]) / (f[b] - f[a]);
        k = k + 1;];
   Print["the result after ", k, " iterations= ", NumberForm[c, 16]];
   Print["f[c]=", NumberForm[f[c], 16]];];]
```

In[2]:=
```
f[x_] := x^3 + 2 * x^2 - 3 * x - 1;
Regulafalsi[1, 2, 50]
```

the result after 35 iterations= 1.19869124351587

f[c]=$-7.780442956573097\times10^{-13}$

In[4]:=
```
f[x_] := x^3 + 2 * x - 1;
Regulafalsi[0, 1, 50]
```

the result after 22 iterations= 0.4533976515162839

f[c]=$-3.137490267590692\times10^{-13}$

In[6]:=
```
f[x_] := Exp[-x] - x;
Regulafalsi[0, 1, 30]
```

the result after 12 iterations= 0.5671432904099458

f[c]=$-2.537969834293108\times10^{-13}$

% DHEERAJ
%21 MTS5712 %
% Date - 13 / 02 / 2023 %
% Aim - To perform the iterations of the Newton raphson method for the functions f (x) = x^3 + 2 * x^2 - 3 * x - 1,
f2 (x) = Cos[x] - x and f3 (x) = e^[-x] - x on the intervals [1, 2], [0, 1] and [0, 1] respectively within an absolute convergence of 10^-8 %

```
In[3]:= NewtonRaphson[x0_, max_] := Module[{}, k = 0; p0 = N[x0];
    p1 = p0;
    While[(k < max && Abs[f[p1]] > 0.00000001),
     p0 = p1;
     If[f'[p0] == 0, Print["p0 is not correct"]; Exit[];,
      p1 = p0 - f[p0] / f'[p0];
      k = k + 1;];];
    Print["p after ", k, "iterations =", NumberForm[p1, 16]];
    Print["f[p]=", NumberForm[f[p1], 16]];]
```

```
In[4]:= f[x_] := x^3 + 2 * x^2 - 3 * x - 1;
NewtonRaphson[2, 13];
```

p after 5iterations =1.19869124352843

f[p]=$7.59046159259924 \times 10^{-11}$

```
In[6]:= f[x_] := Cos[x] - x;
NewtonRaphson[1, 13];
```

p after 3iterations =0.739085133385284

f[p]=$-2.847205804457076 \times 10^{-10}$

```
In[8]:= f[x_] := Exp[-x] - x;
NewtonRaphson[1, 20];
```

p after 3iterations =0.567143285989123

f[p]=$6.927808993140161 \times 10^{-9}$

% DHEERAJ
%21 MTS5712 %
% Date - 13 / 02 / 2023 %
% Practical 4 %
% Aim - To perform Gaussian Elimination method for the matrix : %

```
In[40]:= Gausselim[A0_] := Module[{a = N[A0]}, Print[MatrixForm[a]];
  size = Dimensions[a];
  n = size[[1]];
  m = size[[2]];
  For[i = 1, i ≤ n - 1, i = i + 1,
   For[k = i + 1, k ≤ n, k = k + 1,
     (factor = a[[k, i]] / a[[i, i]];);
        For[p = i, p ≤ m, p = p + 1,
         a[[k, p]] = a[[k, p]] - factor * a[[i, p]];];];];
  Print[MatrixForm[a]];
  ClearAll[x, i];
  x[n] = a[[n, m]] / a[[n, n]];
  Print[x[n]];
  For[i = n - 1, i ≥ 1, i = i - 1,
   s = 0;
   For[j = i + 1, j ≤ n, j = j + 1,
    s = s + a[[i, j]] * x[j];];
   x[i] = (a[[i, m]] - s) / (a[[i, i]]);
   Print[x[i]];];]

In[41]:= a = {{2, 1, 1, 10}, {3, 2, 3, 18}, {1, 4, 9, 16}};
Gausselim[a]
```

$$\begin{pmatrix} 2. & 1. & 1. & 10. \\ 3. & 2. & 3. & 18. \\ 1. & 4. & 9. & 16. \end{pmatrix}$$

$$\begin{pmatrix} 2. & 1. & 1. & 10. \\ 0. & 0.5 & 1.5 & 3. \\ 0. & 0. & -2. & -10. \end{pmatrix}$$

5.

-9.

7.

% DHEERAJ %
%21 MTS5712 %
%20 / 02 / 2024 %

```
(*Aim-To solve the following system of linear equations by
  using Gaussjacobi method within an absolute tolerance of 5*10(-6);
4*x1-x2=2
   -x1+4x2-x3=4
    -x2+4x3=10
*)
(*Programming:*)


Gaussjacobi[A0_, B0_, X0_, max_] :=
 Module[{A = N[A0], B = N[B0], i, j, k = 0, n = Length[X0], X = X0, Xold = X0},
  Print["X", 0, "=" X];
  While[k < max,
   For[i = 1, i ≤ n, i = i + 1,
    X[[i]] =
```

$$\left(B[[i]] - \sum_{j=1}^{i-1} A[[i,j]] * Xold[[j]] - \sum_{j=i+1}^{n} A[[i,j]] * Xold[[j]]\right) \Big/ A[[i,i]]];$$

```
   Print["X", k + 1, "=", NumberForm[X, 10]];
   If[Max[Abs[X - Xold]] < 5 * 10^(-6),
    Print["Solution with convergence tolerance of 5*10^(-6)=",
     NumberForm[X, 10]];
    Break[];,
    Xold = X;
    k = k + 1;];];]
```

$$A0 = \begin{pmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{pmatrix}; B0 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}; X0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix};$$

```
Gaussjacobi[A0, B0, X0, 50];
```

```
X0{{0}, {0}, {0}}
X1={{0.25}, {-0.25}, {0.25}}
X2={{0.4375}, {-0.4375}, {0.375}}
X3={{0.5625}, {-0.5625}, {0.46875}}
X4={{0.6484375}, {-0.6484375}, {0.53125}}
X5={{0.70703125}, {-0.70703125}, {0.57421875}}
X6={{0.7470703125}, {-0.7470703125}, {0.603515625}}
X7={{0.7744140625}, {-0.7744140625}, {0.6235351563}}
X8={{0.7930908203}, {-0.7930908203}, {0.6372070313}}
X9={{0.805847168}, {-0.805847168}, {0.6465454102}}
X10={{0.8145599365}, {-0.8145599365}, {0.652923584}}
X11={{0.8205108643}, {-0.8205108643}, {0.6572799683}}
X12={{0.8245754242}, {-0.8245754242}, {0.6602554321}}
X13={{0.8273515701}, {-0.8273515701}, {0.6622877121}}
X14={{0.8292477131}, {-0.8292477131}, {0.6636757851}}
X15={{0.8305428028}, {-0.8305428028}, {0.6646238565}}
X16={{0.8314273655}, {-0.8314273655}, {0.6652714014}}
X17={{0.8320315331}, {-0.8320315331}, {0.6657136828}}
X18={{0.8324441873}, {-0.8324441873}, {0.6660157666}}
X19={{0.8327260353}, {-0.8327260353}, {0.6662220936}}
X20={{0.832918541}, {-0.832918541}, {0.6663630176}}
X21={{0.8330500249}, {-0.8330500249}, {0.6664592705}}
X22={{0.8331398301}, {-0.8331398301}, {0.6665250125}}
X23={{0.8332011682}, {-0.8332011682}, {0.666569915}}
X24={{0.8332430628}, {-0.8332430628}, {0.6666005841}}
X25={{0.8332716774}, {-0.8332716774}, {0.6666215314}}
X26={{0.8332912216}, {-0.8332912216}, {0.6666358387}}
X27={{0.8333045705}, {-0.8333045705}, {0.6666456108}}
X28={{0.8333136879}, {-0.8333136879}, {0.6666522852}}
X29={{0.8333199153}, {-0.8333199153}, {0.666656844}}
X30={{0.8333241686}, {-0.8333241686}, {0.6666599576}}
Solution with convergence tolerance of 5*10^(-6)=
 {{0.8333241686}, {-0.8333241686}, {0.6666599576}}
```

```
(*%DHEERAJ       %
 %21MTS5712
 %23/02/2024%
(*Aim-To solve the following system of linear equations by
  using GaussSeidel method within an absolute tolerance of 5*10(-6);
4*x1-x2=2
   -x1+4x2-x3=4
    -x2+4x3=10 *)
*)
(*Programming:*)
GaussSeidal[A0_, B0_, X0_, max_] :=
 Module[{A = N[A0], B = N[B0], i, j, k = 0, n = Length[X0], X = X0, Xold = X0},
  Print["X", 0, "=" X];
  While[k < max,
   For[i = 1, i ≤ n, i = i + 1,
    X[[i]] = (B[[i]] - Sum[A[[i, j]] * X[[j]], {j, 1, i-1}] - Sum[A[[i, j]] * Xold[[j]], {j, i+1, n}]) / A[[i, i]]];
   Print["X", k + 1, "=", NumberForm[X, 10]];
   If[Max[Abs[X - Xold]] < 5 * 10^(-6),
    Print["Solution with convergence tolerance of 5*10^(-6)=",
     NumberForm[X, 10]];
    Break[];,
    Xold = X;
    k = k + 1;];];]
```

$$A0 = \begin{pmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{pmatrix};\ B0 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix};\ X0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix};$$

```
GaussSeidal[A0, B0, X0, 3];
```

X0{{0}, {0}, {0}}

X1={{0.25}, {-0.375}, {0.40625}}

X2={{0.5390625}, {-0.62109375}, {0.5400390625}}

X3={{0.6955566406}, {-0.7327880859}, {0.6070861816}}

```
(*DHEERAJ
 21MTS5712
 27/02/2024 *)
(* Aim- To estimate the values of e^-5,
e^-0.7 and e^0.3 by constructing the lagrange's form of intepolatin of
 polynomial for function passing through (-1,e^-1), (0,1) and (1,e^1) *)
(*  PROGRAMMING: *)
```

```
In[5]:= Lagrange1[x_, f_, y_] := Module[{},
   s = 0; m = Length[x]; p = 1;
   For[i = 1, i ≤ m, i = i + 1,
    For[j = 1, j ≤ m, j = j + 1,
     If[j ≠ i,
       p = p * (y - x[[j]]) / (x[[i]] - x[[j]]); Continue;];];
    s = s + p * f[[i]]; p = 1;];
   Print["Function value at y=", s];
   Print["Absolute error=", Abs[s - e^y]];]
```

```
In[6]:= x = {-1, 0, 1};
f = {Exp[-1], 1, Exp[1]};
Lagrange1[x, f, 0.5]
```

Function value at y=1.72337

Absolute error=Abs$\left[1.72337 - e^{0.5}\right]$

```
In[9]:= Lagrange1[x, f, -0.7]
```

Function value at y=0.443469

Absolute error=Abs$\left[0.443469 - \frac{1}{e^{0.7}}\right]$

```
In[10]:= Lagrange1[x, f, 0.3]
```

Function value at y=1.40144

Absolute error=Abs$\left[1.40144 - e^{0.3}\right]$

In[1]:=

```
% Dheeraj %
% roll no - 21 MTS5712 %
%5 / 3 / 24 %
```

Practical 6(ii)-Newton interpolation

In[5]:=

```
NthDividedDiff[x0_, f0_, start_, end_] :=
  Module[{x = x0, f = f0, i = start, j = end, ans}, If[i == j, Return[f[[i]]],
     ans = (NthDividedDiff[x, f, i + 1, j] - NthDividedDiff[x, f, i, j - 1]) /
       (x[[j]] - x[[i]]);
     Return[ans]];];
```

In[6]:=

```
NewtonDDPoly[x0_, f0_] := Module[{x1 = x0, f = f0, n, P, k, j},
   n = Length[x1];
   P[y_] = 0;
   For[i = 1, i ≤ n, i++, prod[y_] = 1;
    For[k = 1, k ≤ i - 1, k++, prod[y_] = prod[y] * (y - x1[[k]])];
    P[y_] = P[y] + NthDividedDiff[x1, f, 1, i] * prod[y]];
   Return[P[y]];];
nodes = {0, 1, 3};
values = {1, 3, 55};
NewtonPoly[y_] = NewtonDDPoly[nodes, values];
NewtonPoly[y]
NewtonPoly[y_] = Simplify[NewtonPoly[y]];
NewtonPoly[y]
NewtonPoly[2]
```

Out[10]= $1 + 2\,y + 8\,(-1 + y)\,y$

Out[12]= $1 - 6\,y + 8\,y^2$

Out[13]= 21

```
In[1]:= % NAME = Dheeraj
% ROLL NO. = 21 MTS5712
% DATE = 19 / 3 / 24
%Practical 7-tapezoidal rule

In[2]:=
trapezoidalRule[a0_, b0_, n_, f_] := Module[{a = a0, b = b0, h, ai}, h = (b - a) / n;
  ai = h / 2 (f[a] + f[b] + 2 Sum[f[a + h * k], {k, 1, n-1}]);
  Return ai ;
]

In[3]:=
f[x_] := 1 / (1 + x)

In[6]:=
N trapezoidalRule 0, 1, 2, f
Out[6]= 0.708333

In[7]:=
N[trapezoidalRule[0, 1, 4, f]]
Out[7]= 0.697024

In[8]:=
N[trapezoidalRule[0, 1, 8, f]]
Out[8]= 0.694122
```