ACHARYA NARENDRA DEV COLLEGE
UNIVERSITY OF DELHI

B.Sc. Physics Hons.
Sem IV
Numerical Methods
Practical File

-Ayush Mishra

Submitted To :-

Mr. Brijendra Sir

Dr. Rakesh Sir

Submitted By-

Ayush Mishra

AC-815

B.Sc. Physics Hons

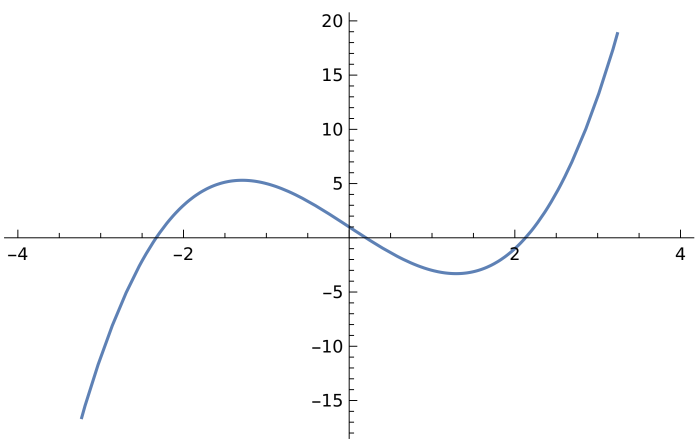
Index

Practical No.-1.
Bisection Method

Q-1- Find the roots of f (x) = by using bisection method.

In[5]:=
```
bisection[f_, ao_, bo_, n_] := Module[{}, a = N[ao]; b = N[bo];
If[f[a] * f[b] > 0, Print["Bisectionmethod can be applied"];
Return[]];
p = (a + b) / 2;
i = 1;
While[i ≤ n,
If[f[a] * f[p] < 0, b = p, a = p];
Print[i, " ", a, " ", b];
i++;
p = (a + b) / 2];
Print["Root = ", p]]
f[x_] := x ^ 3 - 5 * x + 1
Plot[f[x], {x, -4, 4}]
```

Out[7]=



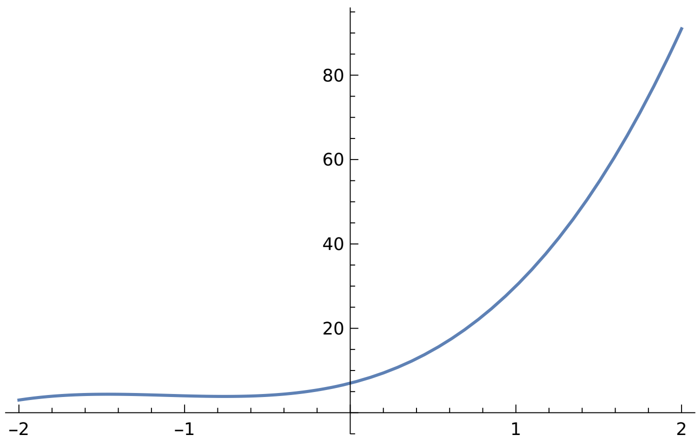
In[8]:=
```
bisection[f, 2, 3, 7]
```

```
1 2. 2.5
2 2. 2.25
3 2.125 2.25
4 2.125 2.1875
5 2.125 2.15625
6 2.125 2.14063
7 2.125 2.13281
Root = 2.12891
```

Q-2: Find the roots of f (x) = $3x^3 + 10x^2 + 10x + 7$ by using bisection method.

In[9]:=
```
bisection[f_, ao_, bo_, n_] := Module[{}, a = N[ao]; b = N[bo];
If[f[a] * f[b] > 0, Print["Bisectionmethod can be applied"];
Return[]];
p = (a + b) / 2;
i = 1;
While[i ≤ n,
If[f[a] * f[p] < 0, b = p, a = p];
Print[i, " ", a, " ", b];
i++;
p = (a + b) / 2];
Print["Root = ", p]]
f[x_] := 3 * x ^ 3 + 10 * x ^ 2 + 10 * x + 7
Plot[f[x], {x, -2, 2}]
```

Out[11]=



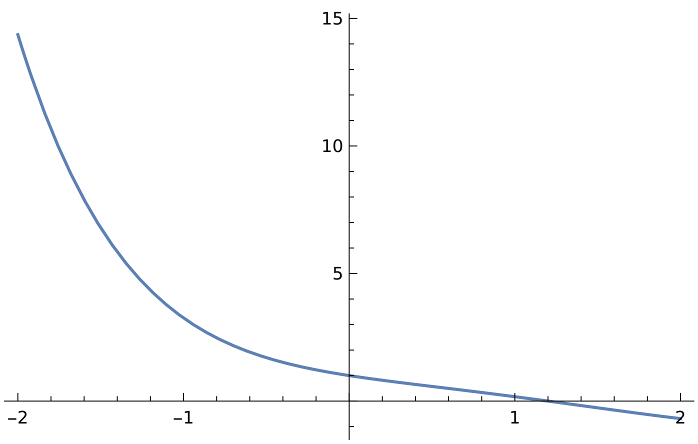
In[12]:=
```
bisection[f, -3, -2, 10]
```

```
1 -2.5 -2.
2 -2.5 -2.25
3 -2.375 -2.25
4 -2.375 -2.3125
5 -2.34375 -2.3125
6 -2.34375 -2.32813
7 -2.33594 -2.32813
8 -2.33594 -2.33203
9 -2.33398 -2.33203
10 -2.33398 -2.33301
Root = -2.3335
```

Q-3: Find the roots of f (x)= by using bisection method.

In[104]:=
```
bisection[f_, ao_, bo_, n_] := Module[{}, a = N[ao]; b = N[bo];
If[f[a] * f[b] > 0, Print["Bisectionmethod can be applied"];
Return[]];
p = (a + b) / 2;
i = 1;
While[i ≤ n,
If[f[a] * f[p] < 0, b = p, a = p];
Print[i, " ", a, " ", b];
i ++;
p = (a + b) / 2];
Print["Root = ", p]]
f[x_] := Cos[x] - x * E ^ (-x)
Plot[f[x], {x, -2, 2}]
```

Out[106]=



In[42]:=
```
bisection[f, 0, 2, 10]
```

```
1 1. 2.
2 1. 1.5
3 1. 1.25
4 1.125 1.25
5 1.1875 1.25
6 1.1875 1.21875
7 1.1875 1.20313
8 1.19531 1.20313
9 1.19922 1.20313
10 1.19922 1.20117
Root = 1.2002
```

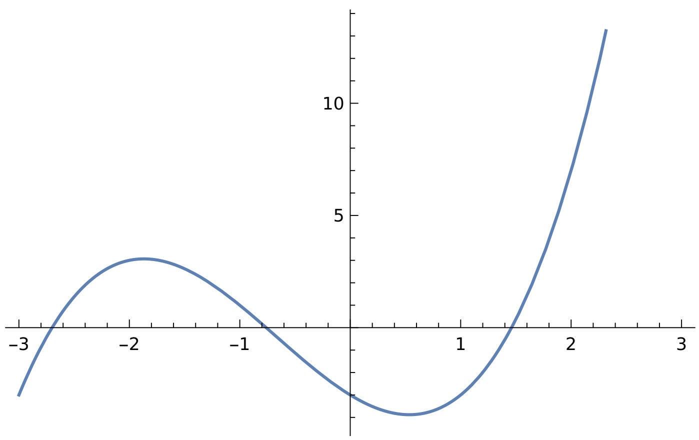
Practical No.-2.

Secant Method

Q-1: Find the root of the function f(x) = .

In[47]:=
```
secant[f_, ao_, bo_, n_] := Module[{}, p0 = N[ao];
p1 = N[bo];
If[f[p0] * f[p1] > 0, Print["secant method can be applied "];
Return[]];
i = 1;
While[i ≤ n, p2 = N[p1 - ((p1 - p0) * f[p1] / (f[p1] - f[p0]))];
Print[i, " ", p0, " ", p1];
i++;
p0 = p1;
p1 = p2];
Print["Root =", p2]]
f[x_] := x ^ 3 + 2 x ^ 2 - 3 * x - 3;
Plot[f[x], {x, - 3, 3}]
secant[f, 1, 3, 5]
```

Out[49]=

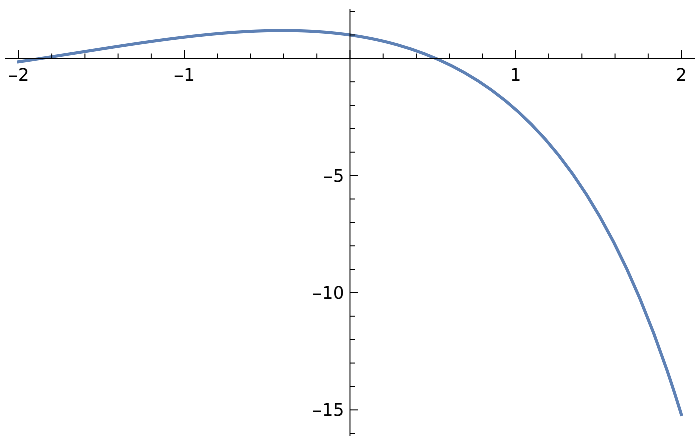



```
1 1. 3.
2 3. 1.16667
3 1.16667 1.28075
4 1.28075 1.50933
5 1.50933 1.454
Root =1.46029
```

Q-2- Find the root of the function f(x) = ☐.

In[51]:=
```
secant[f_, ao_, bo_, n_] := Module[{}, p0 = N[ao];
p1 = N[bo];
If[f[p0] * f[p1] > 0, Print["secant method can be applied "];
Return[]];
i = 1;
While[i ≤ n, p2 = N[p1 - ((p1 - p0) * f[p1] / (f[p1] - f[p0]))];
Print[i, " ", p0, " ", p1];
i++;
p0 = p1;
p1 = p2];
Print["Root =", p2]]
f[x_] := Cos[x] - x * E ^ x;
Plot[f[x], {x, - 2, 2}]
secant[f, 0, 2, 7]
```

Out[53]=

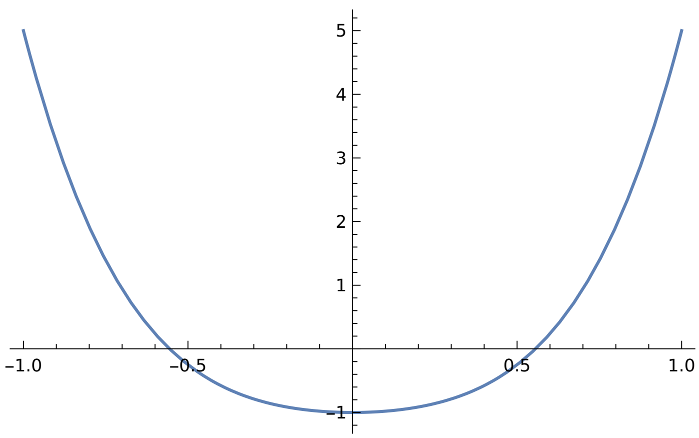



```
1 0. 2.
2 2. 0.123501
3 0.123501 0.223208
4 0.223208 0.666784
5 0.666784 0.478581
6 0.478581 0.513121
7 0.513121 0.517913
Root =0.517757
```

Q-3 Find the root of the function f(x) =

In[55]:=
```
secant[f_, ao_, bo_, n_] := Module[{}, p0 = N[ao];
p1 = N[bo];
If[f[p0] * f[p1] > 0, Print["secant method can be applied "];
Return[]];
i = 1;
While[i ≤ n, p2 = N[p1 - ((p1 - p0) * f[p1] / (f[p1] - f[p0]))];
Print[i, " ", p0, " ", p1];
i++;
p0 = p1;
p1 = p2];
Print["Root = ", p2]]
f[x_] := 4 * x ^ 4 + 2 * x ^ 2 - 1;
Plot[f[x], {x, - 1, 1}]
secant[f, 0, 1, 7]
```

Out[57]=




```
1 0. 1.
2 1. 0.166667
3 0.166667 0.298701
4 0.298701 0.986288
5 0.986288 0.397064
6 0.397064 0.461935
7 0.461935 0.592611
Root = 0.549029
```

Practical No.-3.

Regula Falsi Method

Q-1: Solve The Equation f(x) = Using Regula Falsi Method

In[62]:=
```
f[x_] = x ^ 3 + 3 x ^ 2 - 10;
RegulaFalsi [ao_, bo_, m_] := Module[{}, a = N[ao];
b = N[bo];
c = (a f[b] - b f[a]) / (f[b] - f[a]);
k = 0;
While[k < m, 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[" c = ", NumberForm [c, 16]];
Print[" f[c] = ", NumberForm [f[c], 16]];]
RegulaFalsi [1, 2, 30]
```

c = 1.492033301171817

f[c] = $-4.440892098500626\times10^{-16}$

Q-2 Solve The Equation f(x) = ____ Using Regula Falsi Method.

In[65]:=
```
f[x_] = Cos[x] - x * E ^ x;
RegulaFalsi [ao_, bo_, m_] := Module[{}, a = N[ao];
b = N[bo];
c = (a f[b] - b f[a]) / (f[b] - f[a]);
k = 0;
While[k < m, 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[" c = ", NumberForm [c, 16]];
Print[" f[c] = ", NumberForm [f[c], 16]];]
RegulaFalsi [0, 2, 30]
```

c = 0.5177446104539607

f[c] = 0.00003879648444338191

Q-3 Solve The Equation f(x) = ____ Using Regula Falsi Method.

In[68]:=
```
f[x_] = 4*x^4+2*x^2-1;
RegulaFalsi[ao_, bo_, m_] := Module[{}, a = N[ao];
b = N[bo];
c = (a f[b] - b f[a]) / (f[b] - f[a]);
k = 0;
While[k < m, 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[" c = ", NumberForm[c, 16]];
Print[" f[c] = ", NumberForm[f[c], 16]];]
RegulaFalsi[0, 1, 20]
```

```
c = 0.555887532165347
f[c] = -0.00002703820024263504
```

Practical No.-4.

Newton Raphson Method

Q-1: Find the roots of the equation of f (x) = $\boxed{}$ by using Newton Raphson method

In[74]:=
```
newtonraphson[f_, p0_, eps_] := Module[{}, pold = N[p0];
i = 1; pnew = 0;
df[x_] = D[f[x], x];
While[i ≤ 50 && Abs[N[f[pold]]] > eps, pnew = N[pold - N[f[pold]] / N[df[pold]]];
Print[i, " ", pnew]; i++; pold = pnew];
Print[" Root = ", pnew]]
f[x_] := x^3-x-4;
newtonraphson[f, 0.5, .0000002]
```

```
1 -17.
2 -11.3418
3 -7.57045
4 -5.05309
5 -3.36038
6 -2.18673
7 -1.26733
8 -0.0185779
9 -4.00413
10 -2.64117
11 -1.64841
12 -0.6933
13 7.54196
14 5.08119
15 3.48408
16 2.50125
17 1.98646
18 1.81557
19 1.79655
20 1.79632
21 1.79632
 Root = 1.79632
```

Q-2 Find the roots of the equation of f (x) = $\boxed{}$ by using Newton Raphson method.

```
In[86]:= newtonraphson[f_, p0_, eps_] := Module[{}, pold = N[p0];
i = 1; pnew = 0;
df[x_] = D[f[x], x];
While[i ≤ 50 && Abs[N[f[pold]]] > eps, pnew = N[pold - N[f[pold]] / N[df[pold]]];
Print[i, " ", pnew]; i++; pold = pnew];
Print[" Root = ", pnew]]
f[x_] := 3 * x ^ 3 + 10 * x ^ 2 + 10 * x + 7;
newtonraphson[f, -2, .00000002]
```

```
1 -2.5
2 -2.35385
3 -2.3337
4 -2.33333
5 -2.33333
 Root = -2.33333
```

Q-3 Find the roots of the equation of f (x) = by using Newton Raphson method.

In[89]:=
```
newtonraphson[f_, p0_, eps_] := Module[{}, pold = N[p0];
i = 1; pnew = 0;
df[x_] = D[f[x], x];
While[i ≤ 50 && Abs[N[f[pold]]] > eps, pnew = N[pold - N[f[pold]] / N[df[pold]]];
Print[i, " ", pnew]; i++; pold = pnew];
Print[" Root = ", pnew]]
f[x_] := x^4 - x - 10;
newtonraphson[f, 0.5, .0000002]
```

```
1 -20.375
2 -15.2811
3 -11.4607
4 -8.59577
5 -6.44823
6 -4.84098
7 -3.64474
8 -2.77088
9 -2.17017
10 -1.82753
11 -1.71019
12 -1.69761
13 -1.69747
14 -1.69747
 Root = -1.69747
```

Practical No.- 5.

Trapezoidal rule

Q-1: To find the approx value of integral $\int x^2 \, dx$ from x= 2 to x = 10, by using trapezoidal rule .

```
In[90]:= ClearAll[n, x, f]
a = 2
b = 10
n = 15
sum = 0
h = (b - a) / n
f[x_] = x ^ 2
For[i = 1, i ≤ n - 1, i ++, sum += N[f[x] /. x → (a + i * h)]]
sum = N[(2 * sum + f[x] /. x → b) * h / 2]
```

Out[91]= 2

Out[92]= 10

Out[93]= 15

Out[94]= 0

Out[95]= $\frac{8}{15}$

Out[96]= x^2

Out[98]= 329.979

Q-2: To find the approx value of integral $\int x^3 + 8 \, dx$ from x= 2 to x = 10, by using trapezoidal rule .

In[99]:=
```
ClearAll[n, x, f]
a = 2
b = 10
n = 15
sum = 0
h = (b - a) / n
f[x_] = x ^ 2 + 3 x
For[i = 1, i ≤ n - 1, i++, sum += N[f[x] /. x → (a + i * h)]]
sum = N[(2 * sum + f[x] /. x → b) * h / 2]
```

Out[100]=

2

Out[101]=

10

Out[102]=

15

Out[103]=

0

Out[104]=

$\frac{8}{15}$

Out[105]=

$3 x + x^2$

Out[107]=

472.379

Q-3: To find the approx value of integral $\int x^3 + 8 \, dx$ from x= 1 to x = 4, by using trapezoidal rule.

In[108]:=
```
ClearAll[n, x, f]
a = 1
b = 4
n = 15
sum = 0
h = (b - a) / n
f[x_] = x ^ 3 + 8
For[i = 1, i ≤ n - 1, i ++, sum += N[f[x] /. x → (a + i * h)]]
sum = N[(2 * sum + f[x] /. x → b) * h / 2]
```

Out[109]=

1

Out[110]=

4

Out[111]=

15

Out[112]=

0

Out[113]=

$\frac{1}{5}$

Out[114]=

$8 + x^3$

Out[116]=

87.

PRACTICAL-6

LAGRANGE 'S INTERPOLATION METHOD

Q1. The function y = f(x) is given at the points (-1,-2),(1,0),(4,63),(7,342). Find f(5.0) using Lagrange interpolation .

In[1]:=
```
No = 4; sum = 0;
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])];
Print[sum]
sum = 0;
points = {{-1, -2}, {1, 0}, {4, 63}, {7, 342}};
No = Length[points]
y = points⟦All, 1⟧
f = points⟦All, 2⟧
lagrange[No_, n_] :=
 Product[If[Equal[k, n], 1, (x - y⟦k⟧) / (y⟦n⟧ - y⟦k⟧)], {k, 1, No}]
For[i = 1, i ≤ No, i++, sum += (f⟦i⟧ * lagrange[No, i])]
Expand[sum]
sum /. x → 5.0
```

$$\frac{f[x[1]]\,(x-x[2])\,(x-x[3])\,(x-x[4])}{(x[1]-x[2])\,(x[1]-x[3])\,(x[1]-x[4])}+\frac{f[x[2]]\,(x-x[1])\,(x-x[3])\,(x-x[4])}{(-x[1]+x[2])\,(x[2]-x[3])\,(x[2]-x[4])}+\frac{f[x[3]]\,(x-x[1])\,(x-x[2])\,(x-x[4])}{(-x[1]+x[3])\,(-x[2]+x[3])\,(x[3]-x[4])}+\frac{f[x[4]]\,(x-x[1])\,(x-x[2])\,(x-x[3])}{(-x[1]+x[4])\,(-x[2]+x[4])\,(-x[3]+x[4])}$$

Out[7]= 4

Out[8]= {-1, 1, 4, 7}

Out[9]= {-2, 0, 63, 342}

Out[12]= $-1 + x^3$

Out[13]= 124.

Q2. The function y = f(x) is given at the points (2,15),(4,5),(5,6),(6,19). Find f(5.5) using Lagrange interpolation .

In[40]:=
```
No = 4; sum = 0;
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])];
Print[sum]
sum = 0;
points = {{2, 15}, {4, 5}, {5, 6}, {6, 19}};
No = Length[points]
y = points〚All, 1〛
f = points〚All, 2〛
lagrange[No_, n_] :=
 Product[If[Equal[k, n], 1, (x - y〚k〛) / (y〚n〛 - y〚k〛)], {k, 1, No}]
For[i = 1, i ≤ No, i++, sum += (f〚i〛 * lagrange[No, i])]
Expand[sum]
sum /. x → 5.5
```

$$\frac{(x-x[2])\,(x-x[3])\,(x-x[4])\,\{210,\ 5,\ 6,\ 19\}[x[1]]}{(x[1]-x[2])\,(x[1]-x[3])\,(x[1]-x[4])}+\frac{(x-x[1])\,(x-x[3])\,(x-x[4])\,\{210,\ 5,\ 6,\ 19\}[x[2]]}{(-x[1]+x[2])\,(x[2]-x[3])\,(x[2]-x[4])}+\frac{(x-x[1])\,(x-x[2])\,(x-x[4])\,\{210,\ 5,\ 6,\ 19\}[x[3]]}{(-x[1]+x[3])\,(-x[2]+x[3])\,(x[3]-x[4])}+\frac{(x-x[1])\,(x-x[2])\,(x-x[3])\,\{210,\ 5,\ 6,\ 19\}[x[4]]}{(-x[1]+x[4])\,(-x[2]+x[4])\,(-x[3]+x[4])}$$

Out[46]= 4

Out[47]= {2, 4, 5, 6}

Out[48]= {15, 5, 6, 19}

Out[51]= $1+21\,x-9\,x^2+x^3$

Out[52]= 10.625

Q3. The function $y = f(x)$ is given at the points (0,1),(1,14),(2,15),(4,5).
Find f(8.0) using Lagrange interpolation .

In[53]:=
```
No = 4; sum = 0;
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])];
Print[sum]
sum = 0;
points = {{0, 1}, {1, 14}, {2, 15}, {4, 5}};
No = Length[points]
y = points⟦All, 1⟧
f = points⟦All, 2⟧
lagrange[No_, n_] :=
 Product[If[Equal[k, n], 1, (x - y⟦k⟧) / (y⟦n⟧ - y⟦k⟧)], {k, 1, No}]
For[i = 1, i ≤ No, i++, sum += (f⟦i⟧ * lagrange[No, i])]
Expand[sum]
sum /. x → 8.0
```

$$\frac{(x-x[2])\,(x-x[3])\,(x-x[4])\,\{15,\ 5,\ 6,\ 19\}[x[1]]}{(x[1]-x[2])\,(x[1]-x[3])\,(x[1]-x[4])}+\frac{(x-x[1])\,(x-x[3])\,(x-x[4])\,\{15,\ 5,\ 6,\ 19\}[x[2]]}{(-x[1]+x[2])\,(x[2]-x[3])\,(x[2]-x[4])}+\frac{(x-x[1])\,(x-x[2])\,(x-x[4])\,\{15,\ 5,\ 6,\ 19\}[x[3]]}{(-x[1]+x[3])\,(-x[2]+x[3])\,(x[3]-x[4])}+\frac{(x-x[1])\,(x-x[2])\,(x-x[3])\,\{15,\ 5,\ 6,\ 19\}[x[4]]}{(-x[1]+x[4])\,(-x[2]+x[4])\,(-x[3]+x[4])}$$

Out[59]= 4

Out[60]= {0, 1, 2, 4}

Out[61]= {1, 14, 15, 5}

Out[64]= $1+21\,x-9\,x^2+x^3$

Out[65]= 105.

Practical-07

Gauss Jacobi Iteration Method

Q1 = Solve the following equation using jacobi method

10 x + y + z = 12

x + 10 y + z = 12

x + y + 10 z = 12

Taking initial approx . = [0, 0, 0]

```
In[ ]:= Jacobi[A0_, B0_, P0_, max_] := Module[{A = N[A0], B = N[B0],
      i, j, k = 0, n = Length[P0], P = P0, Pold = P0}, Print[" P "_0, " = ", P];
    While[k < max,
      For[i = 1, i ≤ n, i++,
        P[[i]] = 1/A[[i, i]] (B[[i]] + A[[i, i]] × Pold[[i]] - Sum[A[[i, j]] × Pold[[j]], {j, 1, n}])];
      Print["P "_(k+1), " = ", P];
      Pold = P;
      k = k + 1;];
    Return[P];];
A = (10  1  1
     1  10  1
     1   1  10);
B = {12, 12, 12};
vars = {"x1", "x2", "x3", "x4"};
Print["Solve the system"];
Print[MatrixForm[A], MatrixForm[vars], " = ", MatrixForm[B]]
P = {0, 0, 0};
X = Jacobi[A, B, P, 8];
```

$$P[[i]] = \frac{1}{A[[i,\ i]]}\left(B[[i]] + A[[i,\ i]] \times Pold[[i]] - \sum_{j=1}^{n} A[[i,\ j]] \times Pold[[j]]\right)$$

Solve the system

$$\begin{pmatrix} 10 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 10 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \\ 12 \end{pmatrix}$$

$P_0 = \{0, 0, 0\}$

$P_1 = \{1.2, 1.2, 1.2\}$

$P_2 = \{0.96, 0.96, 0.96\}$

$P_3 = \{1.008, 1.008, 1.008\}$

$P_4 = \{0.9984, 0.9984, 0.9984\}$

$P_5 = \{1.00032, 1.00032, 1.00032\}$

$P_6 = \{0.999936, 0.999936, 0.999936\}$

$P_7 = \{1.00001, 1.00001, 1.00001\}$

$P_8 = \{0.999997, 0.999997, 0.999997\}$

Q2.Solve the following equation using jacobi method

$4x + 3y + 2z = 16$

$2x + 3y + 4z = 20$

$x + 2y + z = 8$

Taking initial approx. = [0, 0, 0]

```
In[ ]:= Jacobi[A0_, B0_, P0_, max_] := Module[{A = N[A0], B = N[B0],
      i, j, k = 0, n = Length[P0], P = P0, Pold = P0}, Print[" P "_0, " = ", P];
    While[k < max,
      For[i = 1, i ≤ n, i++,
        P[[i]] = 1/A[[i, i]] (B[[i]] + A[[i, i]] × Pold[[i]] - Sum[A[[i, j]] × Pold[[j]], {j, 1, n}])];
      Print["P "_(k+1), " = ", P];
      Pold = P;
      k = k + 1;];
    Return[P];];
A = {{4, 3, 2}, {2, 3, 4}, {1, 2, 1}};
B = {16, 20, 8};
vars = {"x1", "x2", "x3"};
Print["Solve the system"];
Print[MatrixForm[A], MatrixForm[vars], " = ", MatrixForm[B]]
P = {0, 0, 0};
X = Jacobi[A, B, P, 10];
```

Solve the system

$$\begin{pmatrix} 4 & 3 & 2 \\ 2 & 3 & 4 \\ 1 & 2 & 1 \end{pmatrix}\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} 16 \\ 20 \\ 8 \end{pmatrix}$$

P_0 = {0, 0, 0}

P_1 = {4., 6.66667, 8.}

P_2 = {-5., -6.66667, -9.33333}

P_3 = {13.6667, 22.4444, 26.3333}

P_4 = {-26., -37.5556, -50.5556}

P_5 = {57.4444, 91.4074, 109.111}

P_6 = {-119.111, -177.111, -232.259}

P_7 = {252.963, 395.753, 481.333}

P_8 = {-533.481, -803.753, -1036.47}

P_9 = {1125.05, 1744.28, 2148.99}

P_{10} = {-2378.7, -3608.68, -4605.61}

Q3.Solve the following equation using jacobi method

28 x + 4 y − z = 32

2 x + 17 y + 4 z = 35

x + 3 y ++ 10 z = 24

Taking initial approx . = [0, 0, 0]

In[ ]:=
```
Jacobi[A0_, B0_, P0_, max_] := Module[{A = N[A0], B = N[B0],
    i, j, k = 0, n = Length[P0], P = P0, Pold = P0}, Print[" P "_0, " = ", P];
  While[k < max,
    For[i = 1, i ≤ n, i ++,
      P[[i]] = 1/A[[i, i]] (B[[i]] + A[[i, i]] × Pold[[i]] - Sum[A[[i, j]] × Pold[[j]], {j, 1, n}])];
    Print["P "_(k+1), " = ", P];
    Pold = P;
    k = k + 1;];
  Return[P];];
A = (28 4 1
     2 17 4
     1 3 10);
B = {32, 35, 24};
vars = {"x1", "x2", "x3"};
Print["Solve the system"];
Print[MatrixForm[A], MatrixForm[vars], " = ", MatrixForm[B]]
P = {0, 0, 0};
X = Jacobi[A, B, P, 10];
```

Solve the system

$$\begin{pmatrix} 28 & 4 & 1 \\ 2 & 17 & 4 \\ 1 & 3 & 10 \end{pmatrix}\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} 32 \\ 35 \\ 24 \end{pmatrix}$$

P_0 = {0, 0, 0}

P_1 = {1.14286 , 2.05882 , 2.4}

P_2 = {0.763025 , 1.35966 , 1.66807 }

P_3 = {0.889046 , 1.57657 , 1.9158 }

P_4 = {0.849212 , 1.50345 , 1.83812 }

P_5 = {0.862431 , 1.52642 , 1.86404 }

P_6 = {0.858225 , 1.51876 , 1.85583 }

P_7 = {0.859611 , 1.52119 , 1.85855 }

P_8 = {0.859168 , 1.52039 , 1.85768 }

P_9 = {0.859313 , 1.52064 , 1.85797 }

P_{10} = {0.859266 , 1.52056 , 1.85788 }

PRACTICAL NO.:-8

NEWTON'S INTERPOLATION METHOD

Q1. The function y = f(x) is given at the points (-1,-2),(1,0),(4,63),(7,342). Find f(5.0) using Newton interpolation .

In[232]:=

```
In[10] := sum = 0;
points = {{-1, -2}, {1, 0}, {4, 63}, {7, 342}};
n = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
dd[k_] :=
 Sum[(f[[i]] / Product [If[Equal[j, i], 1, (y[[i]] - y[[j]])], {j, 1, k}]), {i, 1, k}]
p[x_] = Sum[(dd[i] * Product [If[i ≤ j , 1, x - y[[j]]], {j, 1, i - 1}]),
  {i, 1, n}]
Simplify [p[x]]
Evaluate [p[5.0]]
```

Out[234]= 4

Out[235]= {-1, 1, 4, 7}

Out[236]= {-2, 0, 63, 342}

Out[238]= $-1 + x + 4(-1 + x)(1 + x) + (-4 + x)(-1 + x)(1 + x)$

Out[239]= $-1 + x^3$

Out[240]= 124.

Q2. The function y = f(x) is given at the points (2,15),(4,5),(5,6),(6,19). Find f(5.5) using Newton interpolation .

In[241]:=
```
In[10] := sum = 0;
points = {{2, 15}, {4, 5}, {5, 6}, {6, 19}};
n = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
dd[k_] :=
 Sum[(f[[i]] / Product [If[Equal[j, i], 1, (y[[i]] - y[[j]])], {j, 1, k}]), {i, 1, k}]
p[x_] = Sum[(dd[i] * Product [If[i ≤ j , 1, x - y[[j]]], {j, 1, i - 1}]),
  {i, 1, n}]
Simplify [p[x]]
Evaluate [p[5.5]]
```

Out[243]= 4

Out[244]= {2, 4, 5, 6}

Out[245]= {15, 5, 6, 19}

Out[247]= $15 - 5(-2 + x) + 2(-4 + x)(-2 + x) + (-5 + x)(-4 + x)(-2 + x)$

Out[248]= $1 + 21x - 9x^2 + x^3$

Out[249]= 10.625

Q3. The function $y = f(x)$ is given at the points (0,1),(1,14),(2,15),(4,5). Find f(8.0) using Newton interpolation .

In[250]:=
```
In[10] := sum = 0;
points = {{0, 1}, {1, 14}, {2, 15}, {4, 5}};
n = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
dd[k_] :=
 Sum[(f[[i]] / Product [If[Equal[j, i], 1, (y[[i]] - y[[j]])], {j, 1, k}]), {i, 1, k}]
p[x_] = Sum[(dd[i] * Product [If[i ≤ j , 1, x - y[[j]]], {j, 1, i - 1}]),
  {i, 1, n}]
Simplify [p[x]]
Evaluate [p[8.0]]
```

Out[252]= 4

Out[253]= {0, 1, 2, 4}

Out[254]= {1, 14, 15, 5}

Out[256]= $1 + 13x - 6(-1 + x)x + (-2 + x)(-1 + x)x$

Out[257]= $1 + 21x - 9x^2 + x^3$

Out[258]= 105.

Practical N0.-:9

Gauss Elimination Method

Q1.Solve the equation

x1+x2+x3=1

4*x1+3*x2-x3=6

3*x1-5*x2-3*x3=4

In[ ]:=
$$m = \begin{pmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & -5 & -3 \end{pmatrix};\ x = \begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix};\ b = \begin{pmatrix} 1 \\ 6 \\ 4 \end{pmatrix};\ m.x == b$$

```
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[ ]= {{x1 + x2 + x3}, {4 x1 + 3 x2 − x3}, {3 x1 − 5 x2 − 3 x3}} == {{1}, {6}, {4}}

Out[ ]//MatrixForm=
$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 4 & 3 & -1 & 6 \\ 3 & -5 & -3 & 4 \end{pmatrix}$$

Out[ ]//MatrixForm=
$$\begin{pmatrix} 1 & 0 & 0 & \frac{21}{17} \\ 0 & 1 & 0 & \frac{7}{34} \\ 0 & 0 & 1 & -\frac{15}{34} \end{pmatrix}$$

Out[ ]= $\left\{\left\{\frac{21}{17}\right\}, \left\{\frac{7}{34}\right\}, \left\{-\frac{15}{34}\right\}\right\}$

Q2.Solve the equation

2x+3y+z=1

2x+y+z=8

4x+4y+5z=6

In[ ]:=
$$m = \begin{pmatrix} 2 & 3 & 1 \\ 2 & 1 & 1 \\ 4 & 4 & 5 \end{pmatrix}; x = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; b = \begin{pmatrix} 1 \\ 8 \\ 6 \end{pmatrix};$$

```
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[ ]//MatrixForm=
$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 2 & 1 & 1 & 8 \\ 4 & 4 & 5 & 6 \end{pmatrix}$$

Out[ ]//MatrixForm=
$$\begin{pmatrix} 1 & 0 & 0 & \frac{25}{4} \\ 0 & 1 & 0 & -\frac{7}{2} \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Out[ ]= $\left\{\left\{\frac{25}{4}\right\}, \left\{-\frac{7}{2}\right\}, \{-1\}\right\}$

Q3.Solve the equation

8x+3y+z=5

2x+8y-z=13

x+7y-2z=12

In[ ]:=
$$m = \begin{pmatrix} 8 & 3 & 1 \\ 2 & 8 & -1 \\ 1 & 7 & -2 \end{pmatrix}; x = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; b = \begin{pmatrix} 5 \\ 13 \\ 12 \end{pmatrix};$$

```
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[ ]//MatrixForm=
$$\begin{pmatrix} 8 & 3 & 1 & 5 \\ 2 & 8 & -1 & 13 \\ 1 & 7 & -2 & 12 \end{pmatrix}$$

Out[ ]//MatrixForm=
$$\begin{pmatrix} 1 & 0 & 0 & \frac{8}{57} \\ 0 & 1 & 0 & \frac{86}{57} \\ 0 & 0 & 1 & -\frac{37}{57} \end{pmatrix}$$

Out[ ]= $\left\{\left\{\frac{8}{57}\right\}, \left\{\frac{86}{57}\right\}, \left\{-\frac{37}{57}\right\}\right\}$

Practical N0.-: 9
(Gauss Elimination Method)

Q1.Solve the equation

x1+x2+x3=1

x1+x2-x3=3

x1-x2-x3=-1

In[ ]:=
```
m = {{1, 1, 1}, {1, 1, -1}, {1, -1, -1}}; x = {{x1}, {x2}, {x3}}; b = {{1}, {3}, {-1}}; m.x == b
```

Out[ ]= {{x1 + x2 + x3}, {x1 + x2 - x3}, {x1 - x2 - x3}} == {{1}, {3}, {-1}}

In[ ]:=
```
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 3 \\ 1 & -1 & -1 & -1 \end{pmatrix}$$

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Out[ ]= {{0}, {2}, {-1}}

Q2.Solve the equation

3x+2y+4z=7

2x+y+z=7

x+3y+5z=2

In[ ]:=
```
    3 2 4      x      7
m = 2 1 1; x = y; b = 7;
    1 3 5      z      2
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 3 & 2 & 4 & 7 \\ 2 & 1 & 1 & 7 \\ 1 & 3 & 5 & 2 \end{pmatrix}$$

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -2 \end{pmatrix}$$

Out[ ]= {{3}, {3}, {-2}}

Q3.Solve the equation

10x+y+z=12

2x+10y+z=13

x+y+3z=5

In[1]:=
```
    10  1  1      x      12
m =  2 10  1; x = y; b = 13;
     1  1  3      z       5
ArrayFlatten[{{m, b}}] // MatrixForm
RowReduce[%] // MatrixForm
LinearSolve[m, b]
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 10 & 1 & 1 & 12 \\ 2 & 10 & 1 & 13 \\ 1 & 1 & 3 & 5 \end{pmatrix}$$

Out[3]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Out[4]= {{1}, {1}, {1}}