Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1.	By use of C-program given above, find the root of the equation $x^4-x-10=0$ by Bisection method.	1-4			
2.	Write a C-program for finding the root of the equation $x^3-2x-5=0$ by using Bisection method in the interval [2,3] correct upto three decimal place.	5-8			
3.	By using C-program, find a real root of the eqn $x^2+x-1=0$ which lies in [0,1] correct upto three place of decimal by method of false Position	9-12			
4.	By using above C-program find the real root between 1.5 and 1.6 four decimal of the eqn $x^6-x^4-x^3-3=0$ by false position method.	13-17			

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
5.	Write a c-program by using Newton-Raphson method. eqn $x^3-3x-5=0$ correct to four decimal places which line in [2,3],	18-21			
6.	By using above c-program eqn $x^3+x^2-1=0$ correct to four places of decimal by Newton-Raphson method.	22-25			
7.	By use of c-program given above slove the by Grauss-elimination method. $8x_1+4x_2+2x_3=24$ $4x_1+10x_2+4x_4+5x_3=32$ $2x_1+5x_2+4x_4+6.5x_3=26$ $4x_2+4x_3+9x_4=21$	26-31			
8.	By use of c-program given above eqn slove the by Grauss-eli. Method. $2x+8y-2z=14$ $x+6y-z=13$ $2x-y+2z=5$	32-37			

Expt. No. 01 Date

Page No. 01

Experiment No-01

Object :-

By use of C- program given above, find the root of the equation -

$$x^4 - x - 10 = 0$$

by Bisection Method

Soln We know that if the function $f(x)$ is, continous between two real values x_0, x and $f(x_0)$ and $f(x_1)$ are of opposite signs, i.e., $f(x)$

$f(x) < 0$ then there must be a root of $f(x) = 0$ between x_0 and x, let $f(x)$ be negative and $f(x)$ be positive, Now

Now divide x, is the real root root of $f(x)=0$
But if $f(x_1) \neq 0$ then the real root of $f(x) = 0$

will lie either in $[x_0\ x_1]$ or $[x_1, x_1]$.
If $f(x)$ is positive, then the root will lie in $[x_1, x_1]$.

Since $f(x_2) < 0$ and if $f(x_2)$ is negative the root lies in $[x_2, x_2]$.

Since $f(x_2) \cdot f(x_2)$.

Teacher's Signature

Date

Expt. No. Page No. 02

similarly,
the interval in which the origin is located is further divided into two equal parts until the value of the origin is known to the desired accuracy.

The c-program to find the roots of the equation $f(x) = 0$ by bisection method.

```
/* BISECTION METHOD */

# include < stdio.h >
# include < conio.h >
# include < math.h >

# define f(x) x* x* x - 2.0* x 5.0 /*
  write given equation */

void main ( ),

float x, x0, x1, x2, y1, y2, e.,

clrscr ( ),

Print (" Input two Initial starting roots \n" );
  Scanf (%.f %.f", x0, x1);
```

Teacher's Signature

```
Point ("Input the tolerance value \n");
  Scanf ("%.f", e);

      Yo = f(xo);
      Y1 = f(x1);

      if (yo* y1 > 0.0>

{
      Point ("\n Initial are unsuitable \n")

{
      else

{
      while (fabs (x1-xo)/x1 ) > e)

{

      x2 = (xo+x1) /2;
      y2 = f(x2);

      if (yo* y2 >0)

      xo = x2;

else
      x1 = x2;
```

Teacher's Signature _____

Expt. No. 02 Date

Page No. 05

Experiment No - 02

Object :-

Write a c- program for finding the root of the equation -

$$x^3 - 2x - 5 = 0$$

by using Bisection method in the interval [2, 3] correct up to three-decimal places.

Soln :-

We know that if the function f(x) is continuous between two real values x_0, x and $f(x_1)$ and $f(x_1)$. are of opposite signs, i.e., f(x), f(x) < 0, then there must be a root of f(x) = 0 between x_0 and x, Let f(x) be negative and f(x) be positive.

Now divide the interval [x] into two equal parts by a<0, x. point lying between them.

If f(x) = 0, then x, is the real root of f(x)=0. But if $f(x_1) \neq 0$ then x is the real of f(x) = 0.

Will lie either in $[x_0, x_1]$ or $[x_1, x_1]$. If $f(x_1)$ is positive, then the root will lie in $[x_1, x_1]$. since $f(x_2) \cdot f(x_2) < 0$ and if $f(x_2)$ is

Teacher's Signature

```
}
    Print ("Solution converges to a root\n");
    Print ("In Root of the given equation is Y.
                                    8.3f", x2);
}
    getch ( );
}
```

Input :-

Input two initial starting roots

1.2

Input the tolerance value

0.0001

Output :-

Solution converges to a roots

Roots of the given equation

is = 1.856

Teacher's Signature _____

negative the root lies in $[x_2, x_2]$;
since $f(x_2) \cdot f(x_2)$

Similarly,
The interval in which the origion the origin is located is further divided into two equal parts until the value of the origin is knows to the desired accuracy.

The c-program to find the roots of the equation $f(x) = 0$ by bisection method.

```
/* BISECTION METHOD */

# include < stdio.h >
# include < conio.h >
# include < math.h >

# define f(x) x* x* x* - 2.0* x 5.0 /*
        write given equation */

Void main ( )

{
    float x, x0, x1, x2, y0, y2, e;
    clrscr ( );
```

Teacher's Signature _____

```
Print ("Input two Initial Starting roots \n");
Scanf (%f %f", x0, x1);

Print ("Input the tolerance value \n");
Scanf ("%f, e);

    y0 = f(x0);
    y1 = f(x1);

    if (y0* y1 > 0.0)

{
    Print ("In Initial values are unsuitable\n");

{
    else

{
    while ((fabs (x1-x0) / x1) >e)

{
    x2 = (x0+x1)/2;
    y2 = f(x2);

    if (y0* y2 >0)

    x0 = x2;
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 08

```
else
        x1 = x2;
}
Print ("Solution converges to a root \n");
Print ("\n Root of the given equation
                        is %.3f", x2);
}

    getch ( );
}
```

Input :-

Input Two Initial Starting roots

2, 3

Input The Tolerance Value

0.0001

Output :-

Solution converge to a root

Root of the given equation

is = 2.095

Teacher's Signature _____

Expt. No. 03 Date

Page No. 09

Experiment No-03

object :-

By using c-program, find a real root of the equation -

$$x^2 + x - 1 = 0$$

Which lies in [0,1] correct upto three places of decimal by method of false position.

Soln :-

The iteration formula for the false position method.

(Regula falsi method) is as follows;"

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

where, n = 1, 2, 3...

c - program

```
/* Root of an Equation by Regula - falsi method*/

# include < stdio.h >
# include < conio.h >
# include < math.h >

# define f(x)  x*x + x - 1  write given equation*/

void main ( )
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 10

```
{
    float x0, x1, x2, f0, f1, f2, e;
    int max_iteration

clrscr ( ) ;

.   Print (" Input Two Intial Roots, maximum
        iteration and tolerance Value : ");

    Scanf (" %f %f %d %f", &x0, &x1,
        & max_iteration f0 = f(x0);

    f1 = f(x1)
    if (f0* f1 > 0.0)

{
    Print (" \ Initial roots are unsuitable \n");

}
    else
        {
            for (i=1; i< max_iteration; i++)

{
    x2 = (x0* f1 - x1* f0) / (f1 - f0);

    f2 = f(x2)

if (fabs (f2) < e)
```

Teacher's Signature _____

```
{
Print ("\Solution is converge \n");
Print ("\Root of the given equation is
                        =%.8.4f\n", x2);

max_iteration = i;

}
else
{
if (f2* f0<0)
{
x1 = x2;
f1 = f2;
}
else
{
x0 = x2;
f0 = f2;
}
}
```

Teacher's Signature _____

```
}
    getch ( );
}
```

Input :-

Input Two Initial Boads, maximum iterations and tolerance value;

0 1 1.0 0.0001

Output :-

Solution is converge

Roots of the given equation is = 0.618

Teacher's Signature _____

Expt. No. 04 Date

Page No. 13

Experiment No - 04

object :-

By using alave C-program find the real between 1.5 and 1.6 to four decimal of the equation —

$$x^6 - x^4 - x^3 - 3 = 0$$

by false Position method.

Solⁿ :-

The iteration formula for the false Position method (Regula) falsi method) is as follows;"

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Where $n = 1, 2, 3 \ldots$

C-program

```
/* Root of an equation by Regula - falsi
                                    Method */

# include < stdio . h >
# include < conio . h >
# include < math . h >

# define f(x) x* x + x - 1 /* write
    given equation */
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 14

```
Vaid main ( ) ;
{
    float x0, x1, x2, f0, f1, f2, e;
    int max_ iteration
    closer ( ) ;
    Print ("Input Two Initial Roots, Maximum
      iteration and tolerance Value: ");
    Scanf("%f %f %d %f", &x0, &x1
                  & max_ iteration, & e);
    f0 = f(x0);
    f1 = f(x1);
    if (f0* f1 > 0.0)
    {
    Print f ("\Initial roots are unsuitable\n");
    }
    else
    {
    for (i=1; i<= max_ iteration; i++);
```

Teacher's Signature _____

```
{
    x2 = (x0* f1 - x1* f(0) / (f1 - f0);
    f2 = f(x2)

    if (fabs (f2) < e)

{
    Print f ("\n Solution is conver \n")

    Print ("\n Root of the given equation
                is = %.8.4 f. \n", x2);

    max_ iteration = i ;

}
    if (f2* f0 < 0)

    {

        x1 = x2 ;

        f1 = f2 ;

    }
    else

    {

        x0 = x2 ;
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 17

```
            f0 = f2;
        }
    }
}
    getch ( );
}
```

Input :-

Input Two Initial Roots, maximum iterations and tolerance value.

1.5 1.6 10 0.00001

Output :-

Solution in converge

Root of the given equation is = 1.5018.

Teacher's Signature _____

Expt. No. 05 Date

Page No. 18

Experiment No - 05

object :-

Write a c- program by using Newton-Raphson method, find the root of the equation -

$$x^3 - 3x - 5 = 0$$

correct to four decimal places line in [2, 3]

Soln :-

The iterative formula for Neaton-Rephson method.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Where, $f'(x_n)$ is the first derivative of $f(x_n)$ with respect to x_n.

c- program.

```
/* Newton - Raphson method */
# include < stdio.h >
# include < conio.h >
# include < math.h >
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 19

```
#define f(x) x*x*x-3*x-5 /* Given equation */

#define df(x) 3*x*x-3 /* Differential of equation */

Void main ( )

{

long float x0, x1, f0, f1, y1, x, e;
long int i, maxiter;

clrscr ( );

Print ("\n Input initial root, max. iterations
and tolerance Value: \n");

Scanf ("%.1f, %.1d %.1f", &x0
& maxiter, &e);

for (i=1; i<=maxiter, i++)

{

f0 = f(x0);

f1 = df(x0);

x1 = x0 - (f0/f1);
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 20

```
        if ((fabs(x1-x0)/x1 < e)
        {
            Print f("\n solution converges to a
                                    roots\n");
        Print f("\n Number of Iteration
                                = %.1d\n", i);
        Print f("\n Root of the given equation
                                is = %.6.4f", x1);
        i = maxiter
        }
        else
        {
            x0 = x1;
        }
        getch( );
}
```

Teacher's Signature _____

Input :-

Input Initial root, max. iterations and tolerance value.

2 10 0.00001

output :-

Solution Converges to a root

Number of Iterations = 4

Root of the given equation is = 2.2790

Teacher's Signature _____

Expt. No. 06 Date

Page No. 22

Experiment No - 06

object :-

By using above C- program, find the root of the equation -

$$x^3 + x^2 - 1 = 0$$

correct to four places of decimal by Newton - Raphson method.

Soln :-

The iterative formula for Newton-Rephson method.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Where,

$f'(x_n)$ is the first derivative of $f(x_n)$ with respect to x_n.

C - program.

```
/* Newton - Raphson method */

# include < stdio.h >
# include < conio.h >
# include < math.h >
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 23

```
# define f(x) x* x* x-3* x-5 /* Given equation */

# define df(x) 3* x* x-3 /* Differantial of equation */

Vaid main ( )

{

    long float xo, x1, fo, f1, y1, x, e;
    long int i. maxiter;

    clrscr ( );

    Print {"| Input Initial root, max. iterations and taleranca value :|n");

    Scanf ("%. 1f %.1d %. 1f", &xo, &maxiter, &e);

    for (i=1; i <= maxiter, i++);

{

    fo = f(xo);

    f1 = df(xo);

    x1 = xo -(fo/ f1);
```

Teacher's Signature _____

```
if ((f abs (x1 - x0) / x1) < e)
{
Print f ("\n Solution Converges to a
                                roots \n");
Print f (" Number of Iteration
                        = %.1d \n", i);
Print f (" \n Root of the given equation
                    is = %. 8.4 \n f", x1);
i = maxiter;
}
else
{
{
x0 = x1
}
}
}
getch ( );
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 25

Input :-

Input Initial root, max. iterations and talerance value:

0.5 10 0.00001

Output :-

Solution converges to a root

Number of Iterations = 5

Root of the given equation is = 0.7548.

Teacher's Signature _____

Expt. No. 07 Date

Page No. 26

Experiment No - 07

object :-

By use of c- program given above the following System of equation by Galles-elimination method:

$$8x_1 + 4x_2 + 2x_3 = 24$$
$$4x_1 + 10x_2 + 5x_3 + 4x_4 = 32$$
$$2x_1 + 5x_2 + 6.5x_3 + 4x_4 = 26$$
$$4x_2 + 4x_3 + 9x_4 = 21$$

Soln :-

Gauss Elimination method :-

under this method the coefficient matrix with in the augmented matrix of the given System of linear algebraic equation is converted into an upper trianguler matrix using elementary row operation.

There after, the values of the unknown Varibles are determined using the back-Substitution method.

[computer Application with c- program]

```
/* c- program for Gauss - Elimination
# include < stdio.h >
# include < conio.h >
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 27

```
Vaid main ( )
{
    int i, j, k, n;
    float a [10] [10], x [10], sum, temp
    Print (" Enter Number of System of
                equations \n \n ");
    Scanf (" %. d", &n );
    Print f (" Number of equations
                are = %. d \n", n);
    closer ( );
    Print f (" Enter coefficients of liner
        coefficients of augmented matrix
                equations \
                \n \n");
    for (i=0, i<n: i++)
    {
        for (j=0; j<m ; j++)
        {
```

Teacher's Signature _____

```
for (j=0; j<n+1:j++)
scanf ("%.f", &n[i][i];
}
/* Calulate upper triangular matrix */
for (i=1 : i<n-1; i++)
{
temp = 0.0;
for (j=i+1;j<=n; j++)
{
temp = a[i][i]/a[i][i];
for (k=i; k<=n+1; k++)
a[i][k] = a[j][k] -( temp*
                     a[j][k]);
}
}
/* Print upper triangular matrix */
```

Teacher's Signature _____

```
Point ("\n upper triangular matrix is :\n\n");
for (i=1; i<=n; i++)
{
    for (j=1; j<=n+1; j++)
{
    for (j=1; j<=n+1; j++)
Print f ("%.8.2f", a[i][i];
Print f ("\n");
}
/* evaluate unknowns by back
                    substitution */
x[n] = a[n] [n+1] / a[n][n];
for (i=n-1; i>=1; i--)
{
    sum = 0.0;
    for (i+1; j<=n; j++)
{
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 30

```
        sum = sum + (a[i][j] * x[i]);
    }
    x[i] = a[i][n+1] - sum / a[i][i]
}
Printf("\n Solution of the equation is\n");
for (i = 1; i <= n; i++)
Printf("x[%d] = %8.3f\n", i, x[i]);
getch ( );
}
```

Input :-

Enter Number of system of equations.

Number of Equations are = 4

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 31.

Enter Coefficients of linear equations / cofficients of augmented matrix :

8	4	2	0	24
4	10	5	4	32
2	5	6.5	4	26
0	4	4	9	21

output :-

Upper Traingular matrix is :

8.00	4.00	2.00	0.00	24.00
0.00	8.00	4.00	4.00	20.00
0.00	0.00	4.00	2.00	10.00
0.00	0.00	0.00	6.00	6.00

Solution of the equations is :

x[1] = 2.00 x[2] = 1.00

x[3] = 2.00 x[4] = 1.00

Teacher's Signature _____

Expt. No. 08 Date

Page No. 32

Experiment No - 08

object :-

Write a c-Program for the following system of linear equations by using Gauss - Elimination method.

$$2x - 8y - 2z = 14$$
$$x + 6y - 2 = 13$$
$$2x - y + 2z = 5$$

Soln :-

Gauss - Elimination method :-

Under this method the cofficient matrix with in the augmented matrix of the given System of linear algebraic equations is converted into an upper traingular matrix using elementary row operations.

Thereafter, the values of the unknown variables are determined using the back-substitution method.

[computer Applications with c- program]

```
/* c- program for Gauss - Elimination method
# include < stdio . h >
# include < conio . h >
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. 33

```
Void main ( )
{
int i, j, k, n;
float a[10][10], x[10], sum, temp;
Print f (" Enter Number of System of
                    equations \n \n");
scanf (" %. d, &n);
Print f (" Number of Equations are = %. d \n \n", n);
clrscr ( );
Print f (" Enter Coefficients of lines
                    equations /
Coefficient of augmented matrix \n \n");
for (i=0, i<n; i++)
{
for (i=0, j<n+1; j++)
scanf (" %. f", &a[i][j]);
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 34

```
/* calculate upper triangular matrix */
for (i=1 ; i<n-1; i++)
{
    temp = 0.0 ;
    for (j=i+1 ; j<=n; j++)
{
    temp = a[i][i] / a[i][j] ;
for (k=i; k<=n+1 ; k++)
    a[i][k] = a[j][k] - temp * a[i]
}
}
/* Print upper triangular matrix */
Print f("\n upper Triangular matrix
                    is : \n \n");
for (i=1 ; i<=n; i++)
```

Teacher's Signature _____

```
{
    for (j = 1 ; j <= n+1 ; j++)
    Print f ("-%. 8.2 f", a[i] [j]);
    Print ("\n");
}
    /* evaluate unknowns by back substitution */
    x[n] = a[n] [n+1] / a[n] [n];
    for (i = n-1 ; i >= 1 ....)
    {
        Sum = 0.0 ;
        for (j = i+1 ; j <= n ; j++)
        {
            Sum = Sum + (a[i] [j] * x[j]);
        }
        x[i] = a[i] [n+1] = Sum / a[i] [i]
}
```

Teacher's Signature _____

```
Print f ("n|Solution of the equation is : |n|n");
for (i=1 <= n; i++)
Print ("x [%d] = %8.3f|n", i, x[i]);
getch ( );
}
```

Input :—

Enter Number of System of equations.

Number of equations are = 3

Enter coefficients of linear equations / cofficients of augmented matrix :

$$\begin{bmatrix} 2 & 8 & 2 & 14 \\ 2 & 6 & -1 & 13 \\ 2 & -1 & 2 & 5 \end{bmatrix}$$

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 37

output 1 :

Upper Triangular matrix is :

2.00	8.00	2.00	14.00
0.00	2.00	-2.00	6.00
0.00	0.00	-9.00	18.00

Solution of the equation is :

X[1] = 5.00

X[2] = 1.00

X[3] = -2.00

Teacher's Signature _____