

## LAB-1

### OBJECTIVE 1:

- i. Write a program in “C” Language to deduce absolute and relative errors.

#### Program:

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main()
{
    double abs_err, rel_err, p_rel_err, t_val, a_val;
    printf(“\n INPUT TRUE VALUE:”);
    scanf(“%lf”, &t_val);
    printf(“\n INPUT APPROXIMATE VALUE:”);
    scanf(“%lf”, &a_val);
    abs_err=fabs(t_val-a_val);
    rel_err=abs_err/t_val;
    p_rel_err=rel_err*100;
    printf(“\nABSOLUTE ERROR= %lf”, abs_err);
    printf(“\nRELATIVE ERROR= %lf”, rel_err);
    printf(“\nPERCENTAGE RELATIVE ERROR= %lf”, p_rel_err);
    getch();
}
```

### OBJECTIVE 2:

- ii. Write a program in “C” Language to evaluate a given polynomial equation and then to deduce errors involved in it.

#### Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a[100],sum=0,x;
    float approx;
    int n,i;
    clrscr();
    printf(“Enter the degree of the polynomial”);
    scanf(“%d”,&n);
    printf(“Enter the coefficients into the array”);
    for(i=n;i>=0;i--)
    {
```

```

        scanf("%f",&a[i]);
    }
    printf("Enter the value of x");
    scanf("%f",&x);
    for(i=n;i>0;i--)
    {
        sum=(sum+a[i])*x;
    }
    sum=sum+a[0];
    printf("\n value of the polynomial is=%f", sum);
    printf("Enter the approximate value: ");
    scanf("%f",&approx);
    printf("Absolute error=%f",(sum-approx));
    getch();
}

```

### OBJECTIVE 3:

- iii. Write a program in "C" Language to deduce errors upto 2 decimal digits by round-off and truncation.
- 

## LAB-2

### OBJECTIVE 1:

- i. Write a program in "C" Language that will find the root of the Algebraic equations using Bisection method.

#### Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#define e 0.001
#define f(x) x*x*x-4*x-9
void main()
{
    float x0,x1,x2;
    float f0,f1,f2;
    int i=0;
    clrscr();
    printf("Enter the values of x0 and x1");
    scanf("%f %f", &x0, &x1);
    do
    {
        f0=f(x0);
        f1=f(x1);

```

```

        x2=(x0+x1)/2;
        f2=f(x2);
        if(f0*f2<0)
        {
            x1=x2;
        }
        else
        {
            x0=x2;
        }
        i++;
        printf("no of iterations %d", i);
        printf("root=%f", x2);
        printf("value of function=%f\n",f2);
    }while(fabs(f2)>e);
    getch();
}

```

## OBJECTIVE 2:

- ii. Write a program in "C" Language that will find the root of the Transcendental equations using Bisection method.

-----

## LAB-3

### OBJECTIVE 1:

- i. Write a program in "C" Language that will find the root of the Algebraic equations using Regula-Falsi method.

### Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#define e 0.001;
#define f(x) x*x*x -2*x -5
void main()
{
    int i=0;
    float x0,x1,x2,f0,f1,f2;
    printf("enter the values of x0 and x1");
    scanf("%f%f",&x0,&x1);
    do
    {

```

```

        fo=f(x0);
        f1=f(x1);
        f2=f(x2);
        x2=((x0*f1)-(x1*f0))/(f1-f0);
        if( f0*f2<0)
        {
            x1=x2;
            f1=f2;
        }
        else
        {
            x0=x2;
            f0=f2;
        }
        i++;
        printf("No of iterations=%d\t",i);
        printf("root=%f\t",x2);
        printf("value of function=%f\n",f2);
    }while(fabs(f2)>e);
    getch();
}

```

## OBJECTIVE 2:

- ii. Write a program in “C” Language that will find the root of the Transcendental equations using Regula-Falsi method.
- 

## LAB-4

### OBJECTIVE 1:

- i. Write a program in “C” Language that will find the root of the Algebraic equations using Newton- Raphson method.

### Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#define e 0.0001
#define f(x) x*x*x -4*x -9
#define df(x) 3*x*x -4

int main()

```

```

{
float x0,x1,f0,f1,df0;
int i=0;
printf("\n input initial value of x0: \n");
scanf("%f",&x0);
do
{
    f0=f(x0);
    df0=df(x0);
    x1=x0-(f0/df0);
    f1=f(x1);
    x0=x1;
    i++;
printf("\n No of iterations=%d\t",i);
printf("root=%f\t",x1);
printf("Value of function=%f\n",f1);
}while(fabs(f1)>e);
return 0;
}

```

**OBJECTIVE 2:**

- i. Write a program in “C” Language that will find the root of the Transcendental equations upto 4 significant digits using Newton- Raphson method.
- .....

**LAB-5**

**OBJECTIVE 1:**

- i. Write a program in “C” Language that will find the root of the Algebraic equations using Iterative method (Gauss Seidel).
- .....

**LAB-6**

**OBJECTIVE 1:**

- i. Write a program in “C” Language to implement trapezoidal rule for known functions.

**Program:**

```

#include <stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<string.h>
float fun(float);
void main()
{
float result=1;
float a,b;
float h,sum;
int i,j;
int n;
clrscr();
printf("\n\nEnter the range-");
printf("\n\nLower Limit a - ");
scanf("%f" ,&a);
printf("\n\n Upper Limit b - ");
scanf("%f" ,&b);
printf("\n\n Enter number of subintervals - ");
scanf("%d" ,&n);
h=(b-a)/n;
sum=0;
sum=(fun(a)+fun(b))/2.0;
for(i=1;i<n;i++)
{
sum+=fun(a+i*h);
}
result=sum*h;
printf("\n\n\n Value of the integral is %6.4f\t",result);
printf("\n\n\n Press Enter to Exit");
getch();
}
float fun(float x)
{
float temp;
temp = exp(-0.5*x);
return temp;
}

```

**OBJECTIVE 2:**

- i. Write a program in “C” Language to implement trapezoidal rule for tabulated functions.

.....

## LAB-7

### OBJECTIVE 1:

- i. Write a program in “C” Language to implement Simpson 1/3 rule for known functions.

### Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<string.h>
float fun(float);
void main()
{
float result=1;
float a,b;
float sum,h, s2, s4,x;
int i,j,n;
clrscr();
printf("\n Enter the range - ");
printf("\n Lower Limit a - ");
scanf("%f",&a)
;printf("\n Upper limit b - ");
scanf("%f",&b);
printf("\n\n Enter number of subintervals - ");
scanf("%d",&n);
h=(b-a)/n;
sum=0;
sum=fun(a)+fun(b);
s2=s4=0.0;
for(i=1;i<n;i+=2)
{
x=a+i*h;
s4=s4+fun(x);
}
for(i=2;i<n;i+=2)
{
x=a+i*h;
s2=s2+fun(x);
}
result=(h/3.0)*(sum+2*s2+4*s4);
printf("\n\n Value of integral is %9.4f\t",result);
getch();}
float fun(float x)
{
float temp;
temp=exp(-0.5*x);
```

```
return temp;
}
```

## OBJECTIVE 2:

- i. Write a program in “C” Language to implement Simpson 1/3 rule for tabulated functions.
- .....

## LAB-8

### OBJECTIVE 1:

- i. Write a program in “C” Language to implement Simpson 3/8 rule for known functions.

### Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<string.h>
float fun(float);
void main()
{
float result=1;
float a, b;
float sum,h, s2, s3,x;
int i,j,n;
clrscr();
printf("\n Enter the range - ");
printf("\n Lower Limit a - ");
scanf("%f",&a);
printf("\n Upper limit b - ");
scanf("%f",&b);
printf("\n\n Enter number of subintervals - ");
scanf("%d",&n);
h=(b-a)/n;
sum=0;
sum=fun(a)+fun(b);
s2=s3=0.0;
for(i=1;i<n;i+=3)
{
x=a+i*h;
s3=s3+fun(x)+fun(x+h);
}
```



```

for(i=3;i<n;i+=3)
{
x=a+i*h;
s2=s2+fun(x);
}
result=(3.0*h/8.0)*(sum+2*s2+3*s3);
printf("\n\nValue of integral is %6.4f\t",result);
getch();
}
float fun(float x)
{
float temp;
temp=1.0+x*x*x;
return temp;
}

```

## OBJECTIVE 2:

- i. Write a program in “C” Language to implement Simpson 3/8 rule for tabulated functions.

.....

## LAB-9

### OBJECTIVE 1:

- i. Write a program in “C” Language to implement Newton’s Forward Interpolation Formula.

#### Program:

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <process.h>
#include <string.h>
void main()
{
int n;
int i,j;
float ax[10];
float ay[10];
float x;
float y = 0;
float h;
float p;
float diff[20][20];

```

```

float y1,y2,y3,y4;
clrscr();
printf("\n Enter the number of terms - ");
scanf("%d",&n);
printf("Enter the value in the form of x - ");
for (i=0;i<n;i++)
{
printf("Enter the value of x%d - ",i+1);
scanf("%f",&ax[i]);
}
printf("\n Enter the value in the form of y - ");
for (i=0;i<n;i++)
{
printf ("Enter the value of y%d - ", i+1);
scanf ("%f",&ay [i]);
}
printf("\nEnter the value of x for");
printf("\nwhich you want the value of y - ");
scanf("%f",&x);
h=ax[1]-ax[0];
for(i=0;i<n-1;i++)
{
diff[i][1]=ay[i+1]-ay[i];
}
for(j=2;j<=4;j++)
{
for(i=0;i<n-j;i++)
{
diff[i][j]=diff[i+1][j-1]-diff[i][j-1];
} }
i=0;
do
{
i++;
}
while(ax[i]<x);
i--;
p=(x-ax[i])/h;
y1=p*diff[i-1][1];
y2=p*(p+1)*diff[i-1][2]/2;
y3=(p+1)*p*(p-1)*diff[i-2][3]/6;
y4=(p+2)*(p+1)*p*(p-1)*diff[i-3][4]/24;
y=ay[i]+y1+y2+y3+y4;
printf("\nwhen x=%6.4f, y=%6.8f ",x,y);
getch();
}

```

.....

## LAB-10

### OBJECTIVE 1:

- i. Write a program in “C” Language to implement Newton’s Backward Interpolation Formula.

#### Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<string.h>
void main()
{
int n,i,j,k;
float mx[10],my[10],x,x0=0,y0,sum,h,fun,p,diff[20][20],y1,y2,y3,y4;
clrscr();
printf("\n enter the no. of terms - ");
scanf("%d",&n);
printf("\n enter the value in the form of x - ");
for(i=0;i<n;i++)
{
printf("\n enter the value of x%d- ",i+1);
scanf("%f",&mx[i]);
}
printf("\n enter the value in the form of y - ");
for(i=0;i<n;i++)
{
printf("\n\n enter the value of y%d- ",i+1);
scanf("%f",&my[i]);
}
printf("\n enter the value of x for");
printf("\nwhich you want the value of of y -");
scanf("%f",&x);h=mx[1]-mx[0];
for(i=0;i<n-1;i++)
{
diff[i][1]=my[i+1]-my[i];
}
for(j=2;j<=4;j++)
{
for(i=0;i<n-j;i++)
{
diff[i][j]=diff[i+1][j-1]-diff[i][j-1];
} }
i=0;
while(!mx[i]>x)
{
i++;
```

```

}
x0=mx[i];
sum=0;
y0=my[i];
fun=1;
p=(x-x0)/h;
sum=y0;
for(k=1;k<=4;k++)
{
fun=(fun*(p-(k-1))/k);
sum=sum+fun*diff[i][k];}
printf("\n when x=%6.4f,y=%6.8f",x,sum);
printf("\n press enter to exit");
getch();
}

```

---

## LAB-11

### OBJECTIVE 1:

Write a program in “C” Language to implement Guass Forward Interpolation Formula.

### Program:

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <process.h>
#include <string.h>
void main()
{
int n;
int i,j;
float ax[10];
float ay[10];
float x;
float nr,dr;
float y=0; float h;
float p;
float diff[20][20];
float y1,y2,y3,y4;
clrscr();
printf(" Enter the number of terms - ");
scanf("%d",&n);
printf("\n Enter the value in the form of x - ");
for (i=0;i<n;i++)
{
printf(" Enter the value of x%d - ",i+1);

```

```

scanf("%f",&ax[i]);
}
printf(" Enter the value in the form of y - ");
for(i=0;i<n;i++)
{
printf("Enter the value of y%d - ",i+1);
scanf("%f",&ay[i]);
}
printf("\nEnter the value of x for - ");
printf("\nwhich you want the value of y - ");
scanf ("%f",&x);
h=ax[1]-ax[0];
for(i=0;i<n-1;i++)
{
diff[i][1]=ay[i+1]-ay[i];
}
for(j=2;j<=4;j++)
{
for(i=0;i<n-j;i++)
{
diff[i][j]=diff[i+1][j-1]-diff[i][j-1];
} }
i=0;
do {
i++;
}
while(ax[i]<x);
i--;
p=(x-ax[i])/h;
y1=p*diff[i][1];
y2=p*(p-1)*diff[i-1][2]/2;
y3=(p+1)*p*(p-1)*diff[i-2][3]/6;
y4=(p+1)*p*(p-1)*(p-2)*diff[i-3][4]/24;
y=ay[i]+y1+y2+y3+y4;
printf("\nwhen x=%6.2f,y=%6.3f ",x,y);
getch();
}

```

---

## LAB-12

### OBJECTIVE 1:

Write a program in “C” Language to implement Gauss Backward Interpolation Formula.

#### Program:

```

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

```

```

#include <math.h>
#include <process.h>
#include <string.h>
void main()
{
int n;
int i,j; float ax[10];
float ay[10];
float x;
float y=0;
float h;
float p;
float diff[20][20];
float y1,y2,y3,y4;
clrscr();
printf("\n Enter the number of terms - ");
scanf("%d",&n);
printf("\n Enter the value in the form of x - ");
for (i=0;i<n;i++)
{
printf("\n\n Enter the value of x%d - ",i+1);
scanf("%f",&ax[i]);
}
printf("\n\n Enter the value in the form of y - ");
for(i=0;i<n;i++)
{
printf("\n Enter the value of y%d - ",i+1);
scanf("%f",&ay[i]);
}
printf("\n\n Enter the value of x for - ");
printf("\n\n which you want the value of y - ");
scanf("%f",&x);
h=ax[1]-ax[0];
for(i=0;i<n-1;i++)
{
diff[i][1]=ay[i+1]-ay[i];
}
for(j=2;j<=4;j++)
{
for(i=0;i<n-j;i++)
{
diff[i][j]=diff[i+1][j-1]-diff[i][j-1];
} }
i=0;
do {
i++;
}
while (ax[i]<x);
i--;
p=(x-ax[i])/h;

```

```

y1=p*diff[i-1][1];
y2=p*(p+1)*diff[i-1][2]/2;
y3=(p+1)*p*(p-1)*diff[i-2][3]/6;
y4=(p+2)*(p+1)*p*(p-1)*diff[i-3][4]/24;
y=ay[i]+y1+y2+y3+y4;
printf("\nwhen x=%6.1f,y=%6.4f ",x,y);
getch();
}

```

---

## LAB-13

### OBJECTIVE 1:

Write a program in “C” Language to implement Bessel’s Interpolation Formula.

#### Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<conio.h>
void main(){
int n; // no. of terms.
int i,j; // Loop variables
float ax[10]; // 'X' array limit 9
float ay[10]; // 'Y' array limit 9
float x; // User Query for what value of X
float y; // Calculated value for coressponding X.
float h; // Calc. Section
float p; // Calc. Section
float diff[20][20]; // to store Y
float y1,y2,y3,y4; // Formulae variables.
clrscr();
printf("\t\t !! BESSELS INTERPOLATION FORMULA!! ");
// Input section.
printf("\n\n Enter the no. of terms -> ");
scanf("%d",&n);
// Input Sequel for array X
printf("\n\n Enter the value in the form of x -> ");
// Input loop for X.
for(i=0;i<n;i++)
{
printf("\n Enter the value of x%d -> ",i+1);
scanf("%f",&ax[i]);
}
// Input sequel for array Y.
printf("\n\n Enter the value in the form of y -> ");
// Input loop for Y.for(i=0;i<n;i++)

```

```

{
printf("\n Enter the value of y%d -> ",i+1);
scanf("%f",&ay[i]);
}
// Inputting the required value quarry
printf("\n\n Enter the value of x for ");
printf("\n which u want the value of y -> ");
scanf("%f",&x);
// Calculation and processing section.
h=ax[1]-ax[0];
for(i=0;i<n-1;i++)
diff[i][1]=ay[i+1]-ay[i];
for(j=2;j<=4;j++)
for(i=0;i<n-j;i++)
diff[i][j]=diff[i+1][j-1]-diff[i][j-1];
i=0;
do
{
i++;
}
while(ax[i]<x);
i--;
p=(x-ax[i])/h;
y1=p*diff[i][1];
y2=p*(p-1)*(diff[i][2]+diff[i-1][2])/4;
y3=p*(p-1)*(p-0.5)*diff[i-1][3]/6;y4=(p+1)*p*(p-1)*(p-2)*(diff[i-2][4]+diff[i-1][4])/48;
// Taking sum
y=ay[i]+y1+y2+y3+y4;
// Output Section
printf("\n When x = %6.2f , y = %6.8f",x,y);
// Invoke user watch halt function
printf("\n\n\n\t\t !! PRESS ENTER TO EXIT !! ");
getch();
}

```

---

## LAB-14

### OBJECTIVE 1:

Write a program in “C” Language to implement Sterling’s Interpolation Formula.

---



## LAB-15

### OBJECTIVE 1:

Write a program in “C” Language to implement Newton’s Divided Difference Formula.

#### Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float x[10],y[10][10],sum,p,u,temp;
int i,n,j,k=0,f,m;
float fact(int);
clrscr();
printf("\nhow many record you will be enter: ");
scanf("%d",&n);
for(i=0; i<n; i++)
{
printf("\n\nenter the value of x%d: ",i);
scanf("%f",&x[i]);
printf("\n\nenter the value of f(x%d): ",i);
scanf("%f",&y[k][i]);
}
printf("\n\nEnter X for finding f(x): ");
scanf("%f",&p);
for(i=1;i<n;i++)
{
k=i;
for(j=0;j<n-i;j++)
{
y[i][j]=(y[i-1][j+1]-y[i-1][j])/(x[k]-x[j]);
k++;
} }
printf("\n_____ \n");
printf("\n x(i)\t y(i)\t y1(i) y2(i) y3(i) y4(i)");
printf("\n_____ \n");
for(i=0;i<n;i++)
{
printf("\n %.3f",x[i]);
for(j=0;j<n-i;j++)
{
printf(" ");
printf(" %.3f",y[j][i]);
}
printf("\n");
}
}
```

```

i=0;
do
{
if(x[i]<p && p<x[i+1])
k=1;else
i++;
}while(k != 1);
f=i;
sum=0;
for(i=0;i<n-1;i++)
{
k=f;
temp=1;
for(j=0;j<i;j++)
{
temp = temp * (p - x[k]);
k++;
}
sum = sum + temp*(y[i][f]);
}
printf("\n\n f(%.2f) = %f ",p,sum);
getch();
}

```

---

## LAB-16

### OBJECTIVE 1:

Write a program in “C” Language to implement Lagrange’s Interpolation Formula.

### Program:

```

#include<stdio.h>
#include<conio.h>
#define MAX 10
void main()
{
float x[MAX],y[MAX],k=0,z,nr,dr;
int i,j,m;
//clrscr();
printf("\n enter the range ");
scanf("%d",&m);
printf("\n enter the x value ");
for(i=0;i<m;i++)
scanf("%f",&x[i]);
printf("\n enter the y value ");
for(i=0;i<m;i++)
scanf("%f",&y[i]);

```

```

printf("\n enter value OF Z to be calculated ");
scanf("%f",&z);
for(i=0;i<m;i++)
{ nr=1;dr=1;
for(j=0;j<m;j++)
{
if (j!=i)
{
nr=nr*(z-x[j]);
dr=dr*(x[i]-x[j]);
} }
k=k+((nr/dr)*y[i]);}
printf("\n final result=%f\n",k);
getch();
}

```

---

### LAB-17

#### OBJECTIVE 1:

Write a program in “C” Language to implement Numerical Differentiations.

---

### LAB-18

#### OBJECTIVE 1:

Write a program in “C” Language to implement Least Square Method for Curve Fitting.

#### Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float x[10],y[10],a[10][10];
int i,j,k,n,itr;
printf("\n ENTER THE SIZE OF MATRIX n:");
scanf("%d",&n);
printf("\n ENTER MATRIX ELEMENTS AND RHS:\n");
for(i=1;i<=n;i++)
{
for(j=1;j<=n+1;j++)
scanf("%f",&a[i][j]);
}

```

```

for(i=1;i<=n;i++)
{
x[i]=0.0;
y[j]=0.0;
}
itr=0.0;
top:
itr=itr+1;
for(i=1;i<=n;i++)
{
x[i]=a[i][n+1];
for(j=1;j<=n;j++)
{
if(i==j)
continue;
else
x[i]=x[i]-a[i][j]*x[j];
}
x[i]=x[i]/a[i][j];
}
for(k=1;k<=n;k++)
if(fabs(x[k]-y[k])>0.0001)
{
printf("\n ITERATION=%d",itr);
for(i=1;i<=n;i++)
{
y[i]=x[i];
printf("\n x(%d)=%f",i,x[i]);
}
goto top;
}
else continue;
getch();
}

```