C-Programs

Lagrange Interpolation:

Code:

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

void main()
{
 float x[100], y[100], xp, yp=0, p;
 int i,j,n;
//      clrscr();
 /* Input Section */
 printf("Enter number of data: ");
 scanf("%d", &n);
 printf("Enter data:\n");
 for(i=1;i<=n;i++)
 {
  printf("x[%d] = ", i);
  scanf("%f", &x[i]);
  printf("y[%d] = ", i);
  scanf("%f", &y[i]);
 }
 printf("Enter interpolation point: ");
 scanf("%f", &xp);
 /* Implementing Lagrange Interpolation */
 for(i=1;i<=n;i++)
 {
  p=1;
  for(j=1;j<=n;j++)
  {
   if(i!=j)
   {
       p = p* (xp - x[j])/(x[i] - x[j]);
   }
  }
  yp = yp + p * y[i];
 }
 printf("Interpolated value at %.3f is %.3f.", xp, yp);
 getch();
}
```

Input/Output:

```
Enter number of data: 5
Enter data:
x[1] = 5
y[1] = 150
x[2] = 7
y[2] = 392
x[3] = 11
y[3] = 1452
x[4] = 13
y[4] = 2366
x[5] = 17
y[5] = 5202
Enter interpolation point: 9
Interpolated value at 9.000 is 810.000.
```

Gauss Jordan:

Code:

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

#define SIZE 10

int main()
{
 float a[SIZE][SIZE], x[SIZE], ratio;
 int i,j,k,n;
//              clrscr();
 /* Inputs */
 /* 1. Reading number of unknowns */
 printf("Enter number of unknowns: ");
 scanf("%d", &n);
 /* 2. Reading Augmented Matrix */
 printf("Enter coefficients of Augmented Matrix:\n");
 for(i=1;i<=n;i++)
 {
  for(j=1;j<=n+1;j++)
  {
   printf("a[%d][%d] = ",i,j);
   scanf("%f", &a[i][j]);
  }
 }
 /* Applying Gauss Jordan Elimination */
 for(i=1;i<=n;i++)
 {
  if(a[i][i] == 0.0)
  {
   printf("Mathematical Error!");
   exit(0);
  }
  for(j=1;j<=n;j++)
  {
   if(i!=j)
   {
    ratio = a[j][i]/a[i][i];
    for(k=1;k<=n+1;k++)
    {
       a[j][k] = a[j][k] - ratio*a[i][k];
    }
   }
  }
 }
 /* Obtaining Solution */
 for(i=1;i<=n;i++)
 {
       x[i] = a[i][n+1]/a[i][i];
 }
 /* Displaying Solution */
 printf("\nSolution:\n");
 for(i=1;i<=n;i++)
 {
       printf("x[%d] = %0.3f\n",i, x[i]);
 }
 getch();
 return(0);
}
```

Input/Output:

```
Enter number of unknowns: 4
Enter coefficients of Augmented Matrix:
a[1][1] = 1
a[1][2] = 2
a[1][3] = 3
a[1][4] = -1
a[1][5] = 10
a[2][1] = 2
a[2][2] = 3
a[2][3] = -3
a[2][4] = -1
a[2][5] = 1
a[3][1] = 2
a[3][2] = -1
a[3][3] = 2
a[3][4] = 3
a[3][5] = 7
a[4][1] = 3
a[4][2] = 2
a[4][3] = -4
a[4][4] = 3
a[4][5] = 2

Solution:
x[1] = 1.000
x[2] = 2.000
x[3] = 2.000
x[4] = 1.000
```

Gauss-Seidel Method

Code:

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

/* Arrange systems of linear
   equations to be solved in
   diagonally dominant form
   and form equation for each
   unknown and define here
*/
/* In this example we are solving
   3x + 20y - z = -18
   2x - 3y + 20z = 25
   20x + y - 2z = 17
*/
/* Arranging given system of linear
   equations in diagonally dominant
   form:
   20x + y - 2z = 17
   3x + 20y -z = -18
   2x - 3y + 20z = 25
*/
/* Equations:
   x = (17-y+2z)/20
   y = (-18-3x+z)/20
   z = (25-2x+3y)/20
```

```
*/
/* Defining function */
#define f1(x,y,z)  (17-y+2*z)/20
#define f2(x,y,z)  (-18-3*x+z)/20
#define f3(x,y,z)  (25-2*x+3*y)/20

/* Main function */
int main()
{
float x0=0, y0=0, z0=0, x1, y1, z1, e1, e2, e3, e;
int count=1;
//  clrscr();
printf("Enter tolerable error:\n");
scanf("%f", &e);

printf("\nCount\tx\ty\tz\n");
do
{
  /* Calculation */
  x1 = f1(x0,y0,z0);
  y1 = f2(x1,y0,z0);
  z1 = f3(x1,y1,z0);
  printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

  /* Error */
  e1 = fabs(x0-x1);
  e2 = fabs(y0-y1);
  e3 = fabs(z0-z1);

  count++;

  /* Set value for next iteration */
  x0 = x1;
  y0 = y1;
  z0 = z1;

}while(e1>e && e2>e && e3>e);

printf("\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n",x1,y1,z1);

getch();
return 0;
}
```

Input/Output:

```
Enter tolerable error:
0.0001

Count   x       y       z
1       0.8500  -1.0275 1.0109
2       1.0025  -0.9998 0.9998
3       1.0000  -1.0000 1.0000
4       1.0000  -1.0000 1.0000

Solution: x=1.000, y=-1.000 and z = 1.000
```

Shooting Method:

Code:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
float f1(float x,float y,float z)
{
    return(z);
}
float f2(float x,float y,float z)
{
    return(x+y);
}
float shoot(float x0,float y0,float z0,float xn,float h,int p)
{
    float x,y,z,k1,k2,k3,k4,l1,l2,l3,l4,k,l,x1,y1,z1;
    x=x0;
    y=y0;
    z=z0;
    do
    {
        k1=h*f1(x,y,z);
        l1=h*f2(x,y,z);
        k2=h*f1(x+h/2.0,y+k1/2.0,z+l1/2.0);
        l2=h*f2(x+h/2.0,y+k1/2.0,z+l1/2.0);
        k3=h*f1(x+h/2.0,y+k2/2.0,z+l2/2.0);
        l3=h*f2(x+h/2.0,y+k2/2.0,z+l2/2.0);
        k4=h*f1(x+h,y+k3,z+l3);
        l4=h*f2(x+h,y+k3,z+l3);
        l=1/6.0*(l1+2*l2+2*l3+l4);
        k=1/6.0*(k1+2*k2+2*k3+k4);
        y1=y+k;
        x1=x+h;
        z1=z+l;
        x=x1;
        y=y1;
        z=z1;
        if(p==1)
        {
            printf("\n%f\t%f",x,y);
        }
    }while(x<xn);
    return(y);
}
main()
{
    float x0,y0,h,xn,yn,z0,m1,m2,m3,b,b1,b2,b3,e;
    int p=0;
    printf("\n  Enter x0,y0,xn,yn,h:");
    scanf("%f%f%f%f%f",&x0,&y0,&xn,&yn,&h);
    printf("\n  Enter the trial M1:");
    scanf("%f",&m1);
    b=yn;
    z0=m1;
    b1=shoot(x0,y0,z0,xn,h,p=1);
    printf("\nB1 is %f",b1);
    if(fabs(b1-b)<0.00005)
    {
        printf("\n  The value of x and respective z are:\n");
        e=shoot(x0,y0,z0,xn,h,p=1);
        return(0);
    }
    else
```

```
    {
    printf("\nEnter the value of M2:");
    scanf("%f",&m2);
    z0=m2;
    b2=shoot(x0,y0,z0,xn,h,p=1);
    printf("\nB2 is %f",b2);
    }
    if(fabs(b2-b)<0.00005)
    {
         printf("\n  The value of x and respective z are\n");
         e= shoot(x0,y0,z0,xn,h,p=1);
         return(0);
    }
    else
    {
        printf("\nM2=%f\tM1=%f",m2,m1);
        m3=m2+(((m2-m1)*(b-b2))/(1.0*(b2-b1)));
        if(b1-b2==0)
        exit(0);

        printf("\nExact value of M =%f",m3);
        z0=m3;
        b3=shoot(x0,y0,z0,xn,h,p=0);
    }
    if(fabs(b3-b)<0.000005)
    {
        printf("\nThere is solution :\n");
        e=shoot(x0,y0,z0,xn,h,p=1);
        exit(0);
    }
       do
       {
           m1=m2;
           m2=m3;
           b1=b2;
           b2=b3;
           m3=m2+(((m2-m1)*(b-b2))/(1.0*(b2-b1)));
           z0=m3;
           b3=shoot(x0,y0,z0,xn,h,p=0);

       }while(fabs(b3-b)<0.0005);
       z0=m3;
       e=shoot(x0,y0,z0,xn,h,p=1);
     }
```

Input/Output:

```
  Enter x0,y0,xn,yn,h:0 1 3 4 0.5

  Enter the trial M1:0

0.500000        1.148438
1.000000        1.717346
1.500000        2.979375
2.000000        5.383970
2.500000        9.672014
3.000000        17.064804
B1 is 17.064804
Enter the value of M2:1

0.500000        1.669271
```

```
1.000000        2.891934
1.500000        5.107366
2.000000        9.008181
2.500000        15.716896
3.000000        27.072250
B2 is 27.072250
M2=1.000000     M1=0.000000
Exact value of M =-1.305508
There is solution :

0.500000        0.468485
1.000000        0.183912
1.500000        0.201267
2.000000        0.652534
2.500000        1.780369
3.000000        3.999996
```