Lagrange in C

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


float lintp(float *xx,float *yy, float x, int n)
{
   int i,j;
   float p,val;

   for (i=0;i<n;i++)
   {
      p=1;
       for(j=0;j<n;j++)
       {
          if(i!=j)
          {
            p=p*(x-xx[j])/(xx[i]-xx[j]);
          }
       }

      val = val + p*yy[i];
   }
  return val;
}

int locate(float *xx,int t,float x)
{
   unsigned long ju,jm,jl;
   int ascnd;
   int j;
   jl=0;
   ju=t;
```

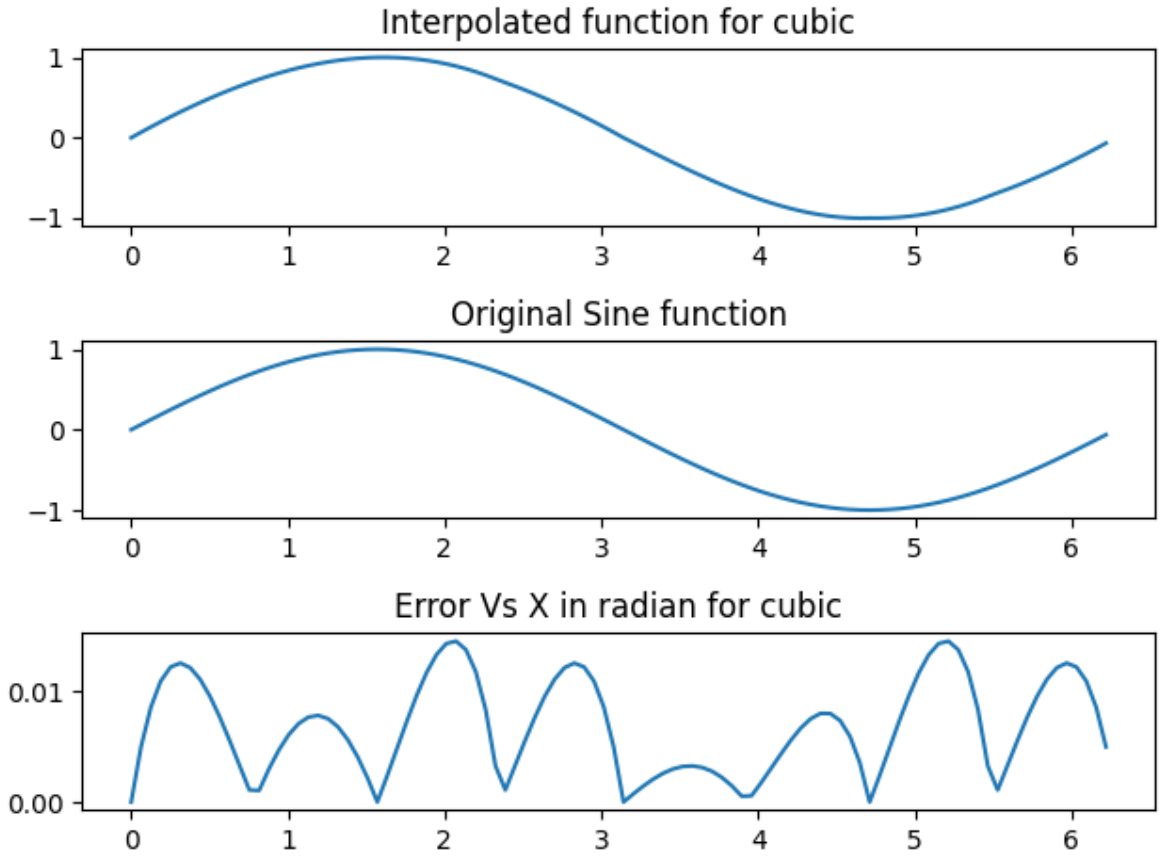
```
    ascnd=(xx[t-1]>=xx[0]);
    while (ju-jl > 1){
        jm= (ju+jl) >>1;
        if(x>=xx[jm] == ascnd)
            jl=jm;
        else
            ju=jm;
    }
    if (x==xx[0]) j=0;
    else if(x==xx[t-1]) j=t-1;
    else j=jl;
    return j ;
}


void main()
{
    int i,j=0;
    float xx[9],yy[9],valuex[4],valuey[4];
    xx[0]=0;
    yy[0]=0;
    for(i=1;i<9;i++)
    {
        xx[i]=xx[i-1]+ M_PI_4;
        yy[i]=sin(xx[i]);
    }
    //For 8th order interpolation
    float var=0;
    FILE *fptr;
    fptr = fopen("output.txt","w+");
    for(i=0;i<100;i++)
    {
        fprintf(fptr,"%f\t",var);
            fprintf(fptr,"%f\n",lintp(xx,yy,var,9));
        var=var+ (M_PI_2/25);

    }
    fclose(fptr);
```

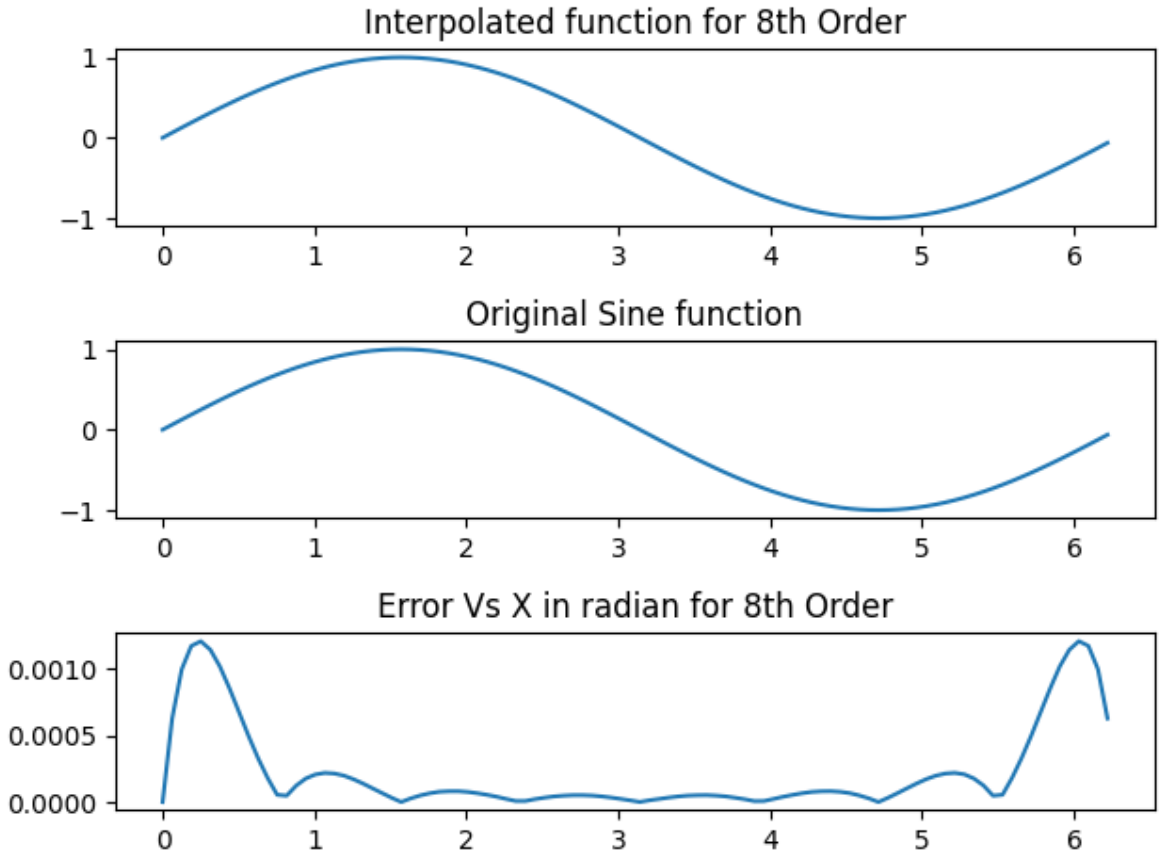
```
    int v;

    //For cubic interpolation
    var=0;
    FILE *fptr1;
    fptr1 = fopen("output1.txt","w+");
    for (int i=0;i<100;i++){
            fprintf(fptr1,"%f\t",var);
     v = locate(xx,9,var);
     if(v<=2){
            for (int c=0;c<4;c++){
                    valuex[c]=xx[c];
                    valuey[c]=yy[c];
       }
     }

     if(v>=3){
        if(v==7 || v==8){
           for(j=0;j<4;j++)
           {
              valuex[j]=xx[j+5];
              valuey[j]=yy[j+5];
           }
        }
        j=0;
            for (int d=v-2;d<=v+1;d++){
                    valuex[j]=xx[d];
                    valuey[j]=yy[d];
          j++;
        }
     }


            fprintf(fptr1,"%f\n",lintp(valuex,valuey,var,4));
       var=var+ (M_PI_2/25);
    }
    fclose(fptr1);
}
```

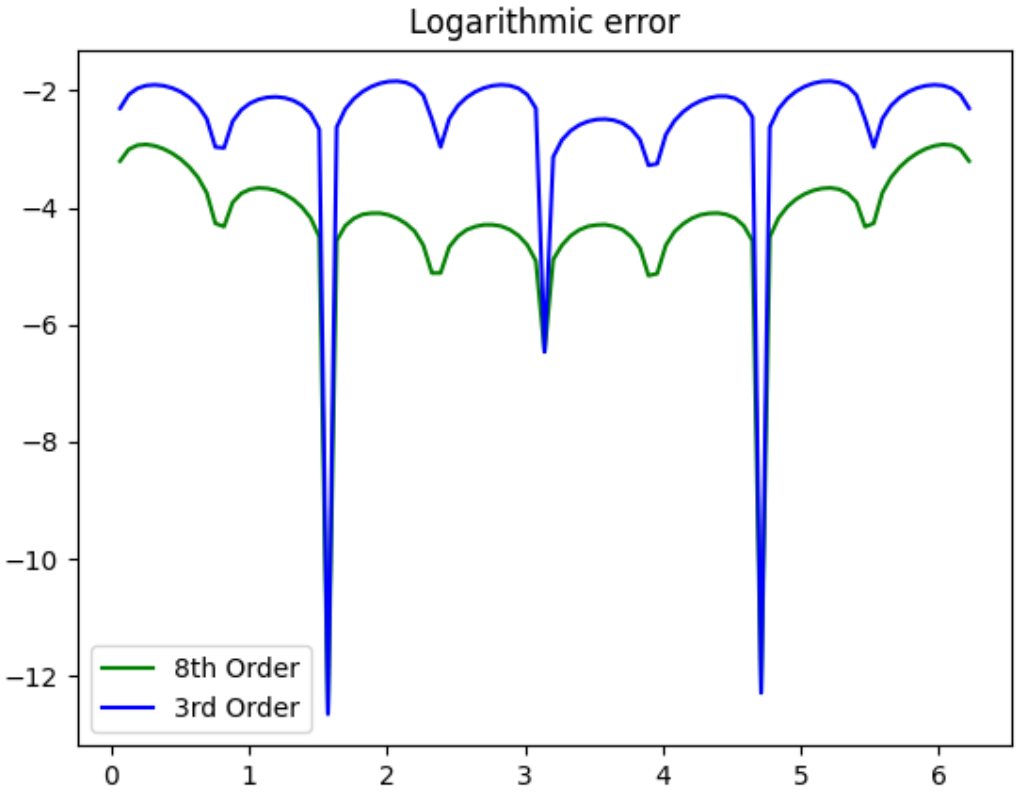
Lagrange: Plotting in Python

```
from matplotlib import colors
import numpy as np
import matplotlib.pyplot as plt
import math


#all information for 8th Order interpolation
x=np.loadtxt("output.txt",usecols=0)
y=np.loadtxt("output.txt",usecols=1)

f=np.sin(x)

error=np.absolute(f-y)
f3=np.delete(f,0,0)
y3=np.delete(y,0,0)
x3=np.delete(x,0,0)
val=np.log10(np.absolute(f3-y3))
plt.plot(x3,val,color="green",label='8th Order')

#cubic
x1=np.loadtxt("output1.txt",usecols=0)
y1=np.loadtxt("output1.txt",usecols=1)
f1=np.sin(x1)

error1=np.absolute(f1-y1)
f2=np.delete(f1,0,0)
y2=np.delete(y1,0,0)
x2=np.delete(x1,0,0)

val=np.log10(np.absolute(f2-y2))
plt.plot(x2,val,color="blue",label="3rd Order")
plt.legend()
plt.title("Logarithmic error")
plt.savefig("Plot_log_error.png")
```

```
#Plotting for 8th order
figure,axis=plt.subplots(3)

axis[0].plot(x,y)
axis[0].set_title("Interpolated function for 8th Order")


axis[1].plot(x,f)
axis[1].set_title("Original Sine function")


axis[2].plot(x,error)
axis[2].set_title("Error Vs X in radian for 8th Order")
figure.tight_layout()

plt.savefig("Plot_8.png")


#Plotting for cubic interpolation
figure,axis1=plt.subplots(3)

axis1[0].plot(x1,y1)
axis1[0].set_title("Interpolated function for cubic")

axis1[1].plot(x1,f1)
axis1[1].set_title("Original Sine function")

axis1[2].plot(x1,error1)
axis1[2].set_title("Error Vs X in radian for cubic")
figure.tight_layout()
plt.savefig("Plot_3.png")
```


Interpolated function for cubic
1
0
−1
0
1
2
3
4
5
6
Original Sine function
1
0
−1
0
1
2
3
4
5
6
Error Vs X in radian for cubic
0.01
0.00
0
1
2
3
4
5
6


Interpolated function for 8th Order
1
0
−1
0
1
2
3
4
5
6
Original Sine function
1
0
−1
0
1
2
3
4
5
6
Error Vs X in radian for 8th Order
0.0010
0.0005
0.0000
0
1
2
3
4
5
6


Logarithmic error
−2
−4
−6
−8
−10
−12
0
1
2
3
4
5
6
8th Order
3rd Order