**Practical No. 11**

|  |  |
| --- | --- |
| Write a C program for Bezier Curve. |  |

***Title****:-*

***Relevant Course Outcome(s):-***

Implement Various Clipping algorithms and given curve generating algorithms.

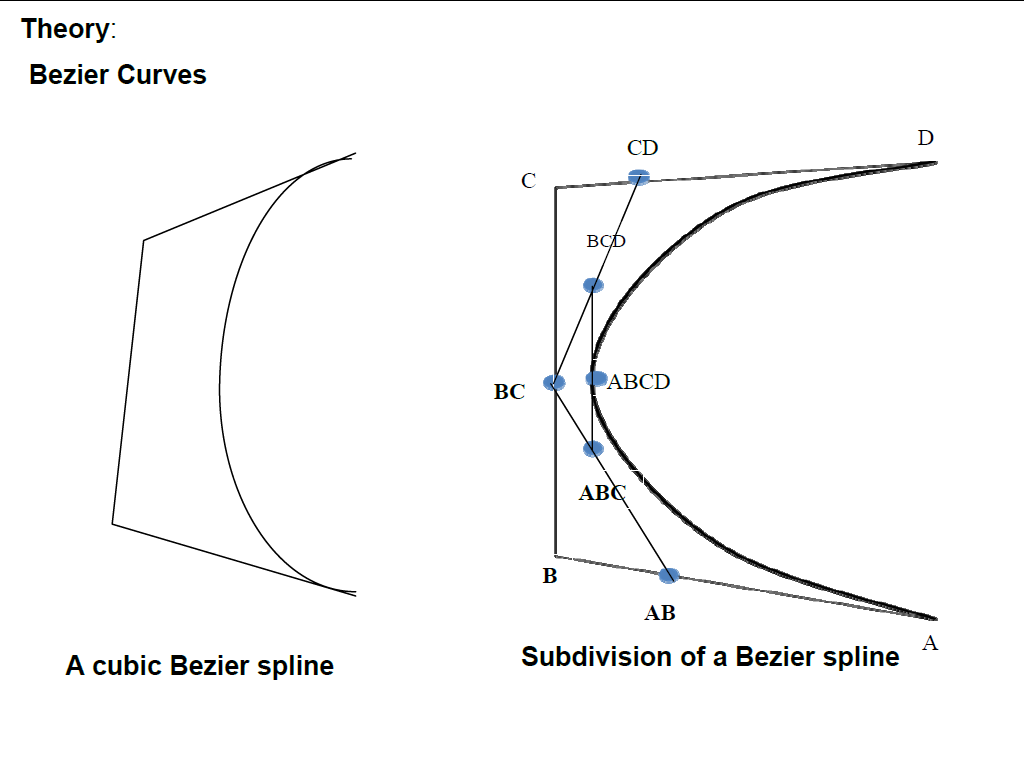
***Resources Required (Hardware & Softwares):-***

A Desktop PC/ Laptop

Ansi C/ Turbo C/ (Any distribution) installed

***Theory****:-*

***Bezier Curves***



Bezier curves is another approach for the construction of the curve. In cubic Bezier curve four control points are used to specify complete curve.

The equations for Bezier curve are as given below

x=x4u3+3x3u2(1-u)+3x2u(1-u)2+x1(1-u)3

y=y4u3+3y3u2(1-u)+3y2u(1-u)2+y1(1-u)3

z=z4u3+3z3u2(1-u)+3z2u(1-u)2+z1(1-u)3

Another approach to construct the Bezier curve is called midpoint approach. In this approach the Bezier curve can be constructed simply by taking midpoints. In midpoint approach midpoints of the lines connecting four control points(A,B,C,D) are determined (AB,BC,CD).These midpoints are connected by the line segments and their midpoints ABC and BCD are determined. Finally these two midpoints are connected by line segments and its midpoint ABCD is determined.

The point ABCD on the Bezier curve divides the original curve into two sections .This males the points A,AB,ABC and ABCD are the control points for the first section and the points ABCD,BCD,CD and D are the control points for the second section. By considering two sections separately we can get two more sections for each separate section i.e .the original Bezier curve gets divided into four different curves. This process can be repeated to split the curve into smaller sections until we have sections so short that they can be replaced by straight lines or even until the sections are not bigger than individual pixels.

**Properties of Bezier Curve:-**

1. Simple approach to design a curve as it needs only 4 control points

2. Bezier curve always passes through the first and last control pointe i.e curve has same end points as the guiding polygon

3. The direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments

4. The curve lies entirely within the convex half formed by four control points

5. The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points

6. This curve can be generated using midpoint approach instead of using equations.

***Algorithm:-***

1. Get four control points say A(xA,yA), B(xB,yB),C(xC,yC),D(xD,yD)

2. Divide the curve represented by points A,B,C and D in two sections

xAB=(xA+xB)/2 yAB=(yA+yB)/2

xBC=(xB+xC)/2

yBC=(yB+yC)/2

xCD=(xC+xD)/2

yCD=(yC+yD)/2

xABC=(xAB+xBC)/2

yABC=(yAB+yBC)/2

xBCD=(xBC+xCD)/2

yBCD=(yBC+yCD)/2

xABCD=(xABC+xBCD)/2

yABCD=(yABC+yBCD)/2

3. Repeat the step 2 for section A,AB,ABC and ABCD and section ABCD,BCD,CD and D

4. Repeat step 3 until we have sections so short that they can be replaced by straight lines

5. Replace small sections by straight lines.

6. Stop.

**Program for Drawing a Curve Using Bezier’s Algorithm**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int gd,gm,maxx,maxy;

float xxx[4][2];

void line1(float x2,float y2)

{

line(xxx[0][0],xxx[0][1],x2,y2);

xxx[0][0]=x2;

xxx[0][1]=y2;

}

bezier(float xb,float yb,float xc,float yc,float xd,float yd,int n)

{

float xab,yab,xbc,ybc,xcd,ycd;

float xabc,yabc,xbcd,ybcd;

float xabcd,yabcd;

if(n==0)

{

line1(xb,yb);

line1(xc,yc);

line1(xd,yd);

}

else

{

xab=(xxx[0][0]+xb)/2;

yab=(xxx[0][1]+yb)/2;

xbc=(xb+xc)/2;

ybc=(yb+yc)/2;

xcd=(xc+xd)/2;

ycd=(yc+yd)/2;

xabc=(xab+xbc)/2;

yabc=(yab+ybc)/2;

xbcd=(xbc+xcd)/2;

ybcd=(ybc+ycd)/2;

xabcd=(xabc+xbcd)/2;

yabcd=(yabc+ybcd)/2;

n=n-1;

bezier(xab,yab,xabc,yabc,xabcd,yabcd,n);

bezier(xbcd,ybcd,xcd,ycd,xd,yd,n);

}

}

void igraph()

{

detectgraph(&gd,&gm);

if(gd<0)

{

puts("can not detect the graphics card");

exit(1);

}

initgraph(&gd,&gm,"c:\\tc\\bgi");

}

void main()

{

int i;

float temp1,temp2;

igraph();

for(i=0;i<4;i++)

{

printf("enter (x,y) cordinates of point%d:",i+1);

scanf("%f%f",&temp1,&temp2);

xxx[i][0]=temp1;

xxx[i][1]=temp2;

}

bezier(xxx[1][0],xxx[1][1],xxx[2][0],xxx[2][1],xxx[3][0],xxx[3][1],8);

delay(100);

getch();

closegraph();

}

**Output:- ( Paste your own Output )**

***Conclusion:-***

Thus, we have written a C program for Bezier Curve