Chapter 5: Introduction to Curves

000552

Curve Generation

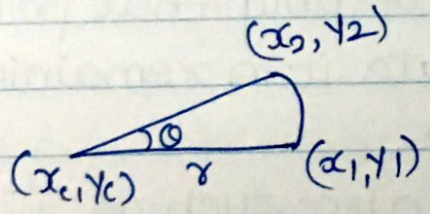
For generations of the curves, we are having mainly two options.

1. use of curve generation method
2. Approximate the curve

1. Curve generation method:-

By using DDA, bresenham or midpoint circle generation, we will get actual curve.

When we are using DDA algo to generate a circular arc, we need radius & some angle θ to draw a curve



If we want to draw a curve from point (x_1, y_1) to (x_2, y_2) ie of angle θ, then we have to find the co-ordinate of (x_1, y_1) as

$$x_1 = x_c + r$$

where, r is radius

& $y_1 = y_c$

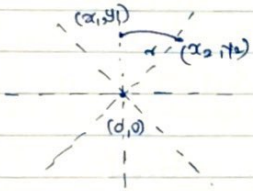
This (x_1, y_1) will be 1st point on the curve

Now, to plot a 2nd point on the curve we have to say.

$$x = x_c + r\cos\theta$$
$$y = y_c + r\sin\theta$$

where,

θ will be some stepping angle. Like this we have to find all the points from (x_1, y_1) to (x_2, y_2) on the path of curve, by increasing every time θ by some stepping value, till the θ will reach to final angle value



When we are using any circle generating algo, we should not use symmetry property of the circle, bcz here we have to plot only some part of circle. After finding any particular point on the path of curve. we have to plot only that point, we should not replicate it in remaining quadrants or octants.

Some drawback also, such as

1. We need more info that just its end points.
2. When we scale a line, it remains line only ie basic properties are not changing. But if we scale a curve it may behave differently ie when a circle is scaled in one direction, it acts as an ellipse. So basic properties are getting change.
- Clipping will be difficult for curves
- every curve which we want to draw is not simple to implement by this method.

So we have to use another method to draw curves, which is called as approximation method.

2. Approximation

Instead of using readmade algo, we are approximating the curve by small straight lines.

For this we have to use interpolation technique.

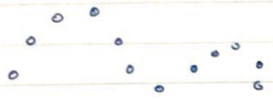
We can draw an approximation to curve if we have array of sample points. We know that curve is a set of points or pixels which lie on the required curve, then we can draw the required curve by filling portions of the curve with pieces of known curves which pass through nearby sample points. The gap between the sample points can be filled by finding the co-ordinates of the points along the known approximating curve & connecting these points with line segment.

The main task in this process is to find the suitable mathematical expression for the known curve.

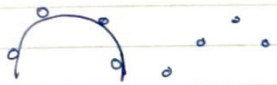
There are polynomial, trigonometric, exponential & other classes of functions that can be used to approximate the curve. Usually polynomial fn in parametric form are preferred. The polynomial functions in the parametric form

$x = f_x(u)$
$y = f_y(u)$
$z = f_z(u)$

unknown curve



known sample point



Fit a region with a known curve

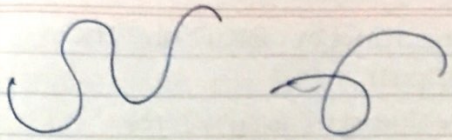
calculate more points from the known curve

Actually draw straight-line segments connecting points.

We can realize from eqn that the difference between 2 & 3 dimensions is just the addition of the third eqn for z.

Furthermore the parametric form treats all the three dimensions equally & allows multiple values (several values of y or z for given x value).

Due to these multiple values curves can double back or even cross themselves



- We have seen that, we have to draw the curve by determining the intermediate points between the known sample points.
- This can be achieved using interpolation technique

Let us see the interpolation process.

Suppose we want a polynomial curve that will pass through n sample points.

$(x_1, y_1, z_1), (x_2, y_2, z_2) \ldots, (x_n, y_n, z_n)$

We will construct the fn as sum of terms, one term for each sample point.

These fn can be given as,

$$f_x(u) = \sum_{i=1}^{n} x_i B_i(u)$$

$$f_y(u) = \sum_{i=1}^{n} y_i B_i(u)$$

$$f_z(u) = \sum_{i=1}^{n} z_i B_i(u)$$

The function $B_i(u)$ is called blending fn. for each value of parameter u, the blending fn determines how much the i^{th} sample point affects the position of the curve.

In other word we can say that each sample points tries to pull the curve in its

direction & the function $B_i(u)$ give the strength of the pull.

- If for some value of u, $B_i(u) = 1$ for unique value of i (i.e $B_i(u) = 0$ for other values of i) then i^{th} sample point has complete control of the curve & the curve will pass through i^{th} sample point.
- For different value of u, some other sample point may have complete control of the curve.
- In such case the curve will pass through that point as well.
- In general, the blending fn give control of the curve to each of the sample points in turns for different values of u.
- Let us assume that the first sample point (x_1, y_1, z_1) has complete control when $u=-1$, the second when $u=0$, then $u=1$, & so on. i.e

When $u=-1 \Rightarrow B_1(u) = 1$ & 0 for $u = 0,1,2 \ldots n-2$

$u=0 \Rightarrow B_2(u) = 1$ & 0 for $u = -1, 1, \ldots n-2$

$\vdots$

$u=(n-2) \Rightarrow B_n(u) = 1$ & 0 for $u = -1, 0, \ldots (n-1)$

To get $B_1(u) = 1$ at $u=-1$ & 0 for $u = 0,1,2, \ldots n-2$

expression for $B_1(u)$ can be given as

$$B_1(u) = \frac{u(u-1)(u-2)\ldots[u-(n-2)]}{(-1)(-2)\ldots(1-n)}$$

where denominator term is a constant used. In general form ith blending function which is 1 at $u=i-2$ & 0 for other integers can be given as:

$$B_i(u) = \frac{(u+1)(u)(u-1)\ldots[u-(i-3)][u-(i-1)][u-(i-2)]}{(i-1)(i-2)(i-3)\ldots(1)(-1)\ldots(i-n)}$$

The approximation of the curve using above expression is called Lagrange interpolation. From the above expression blending fn for four sample points can be given as

$$B_1(u) = \frac{u(u-1)(u-2)}{(-1)(-2)(-3)}$$

$$B_2(u) = \frac{(u+1)(u-1)(u-2)}{1(-1)(-2)}$$

$$B_3(u) = \frac{(u+1)(u)(u-1)}{(2)(1)(-1)}$$

$$B_4(u) = \frac{(u+1)u(u+1)}{(3)(2)(1)}$$

using above blending functions, the expression for the curve passing through sampling points can be realized as follows.

$$x = x_1 B_1(u) + x_2 B_2(u) + x_3 B_3(u) + x_4 B_4(u)$$

$$y = y_1 B_1(u) + y_2 B_2(u) + y_3 B_2(u) + y_4 B_4(u)$$

$$z = z_1 B_1(u) + z_2 B_2(u) + z_3 B_3(u) + z_4 B_4(u)$$

Interpolating algorithm

1. Get the sample points
2. Get intermediate values of u to determine intermediate point
3. Calculate blending fn values for middle section of the curve
4. Calculate blending fn values for first section of the curve
5. Calculate blending fn values for last section of the curves.
6. Multiply the sample points by blending fns to give points on approximation curve.
7. Connect the neighbouring points using straight line segments.
8. stop.

S.N.J.B.'s
SHRI H. H. J. B. POLYTECHNIC, CHANDWAD
CLASS TEST I / II (201 - 201)

Course & Semester :		Roll No.:	Date : / /201
Name of Subject :			Marks obtained :
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

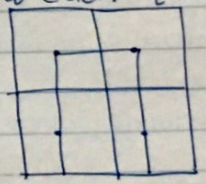
Supplied By : SNJB

Types of curves

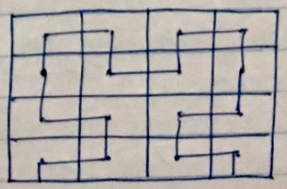
000553

① Hilberts Curve

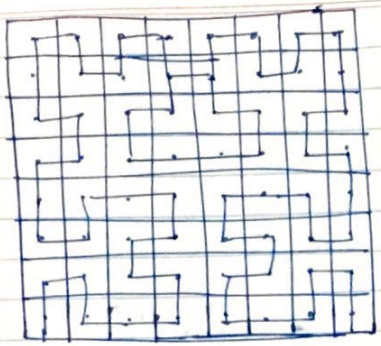
It can constructed by following successive approximations. If a square is divided into four quadrants we can draw the first approximation to the Hilberts curve by connecting centre point of each quadrant.



The second approximation to the Hilbert's curve can be drawn by further subdividing each of the quadrants & connecting their centers before moving to next major quadrant



The third approximation subdivides the quadart again. We can draw third approximations to Hilbert's curve by connecting the centres of finest level of quadrants before stepping to the next level of the quadrant



from above three figures we can easily note following points about Hilbert's curve

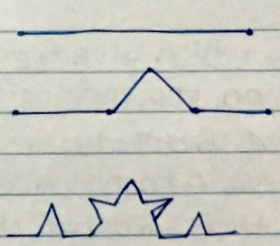
1. If we infinitely extend the approximations to the Hilbert's curve, the curve fills the smaller quadrants but nevers crosses-itself.
2. The curve is arbitrarily close to every point in square
3. The curve passes through a point on grid which betcomes twice as fine with each subdivision.
4. There is no limit to subdivisions & therefore length of curve is infinite
5. With each subdivision length of curve increases by factor of 4.
6. At each subdivision the scale changes by 2 but length changes by 4. therefore for Hilbert's curve topological dimension is one but the fractal dimension is 2.

(2) Koch Curve

The Koch curve can be drawn by dividing line into 4 equal segments with scaling factor 1/3 & middle two segments are so adjusted that they form adjacent sides of an equilateral triangle

This is the first approximation to the Koch curve.

To apply the second approximation to the Koch curve. we have to repeat the above process for each of the four segments.



original Length

1st Approximation

and Approximation.

The resultant curves has moore wiggles & its length is 16/9 times the original length.

From the above figures we can easily note following points about the Koch curve:

1. Each repetition increases the length of the curve by factor 4/3.
2. Length of curve is infinite
3. Unlike Hilbert's curve, it doesn't fill an area.
4. It doesn't device much from its original shape.

5. If we reduce the scale of the curve by 3, we find the curve that looks just like the original one; but we must assemble 4 such curves to make the original, so we have

$$4 = 3^D$$

solving for D we get

$$D = \log_3 4 = \log 4 / \log 3$$

$$\cong 1.2618$$

Therefore for koch curve topological dimension is 1 but fractal dimension is 1.2618.

Curves & surfaces which give fractal dimension greater than the topological dimension are called fractals.

The Hilbert's curve & koch curves are fractals, bcz their fractal dimensions are greater than their topological dimension which is 1.

3) Bezier Curves

It is determined by a defining polygon.

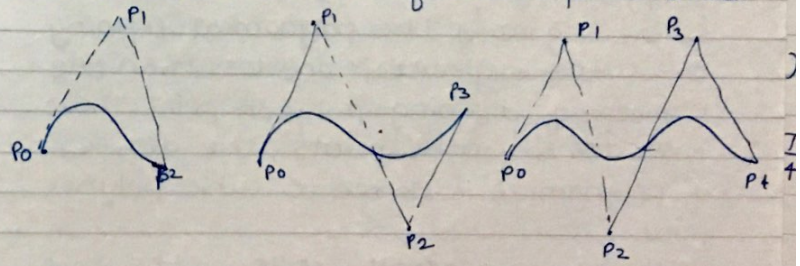
It have a number of properties that makes them highly useful & convenient for curve & surface design. They are also easy to implement.

Therefore, they are widely available in various CAD systems & in general graphics package.

Cubic Bezier Curve

They are provide reasonable design flexibility & also avoid the large number of calculations.

However, no. of control point increase, the degree of the Bezier polynomial also increase, Bcz in Bezier curve a degree of a polynomial is one less than the number of control point used.



The Bezier curves can be specified with boundary conditions, with characterizing matrix or with blending fn. out of these, blending fn specification is the most convenient way for general Bezier curves.

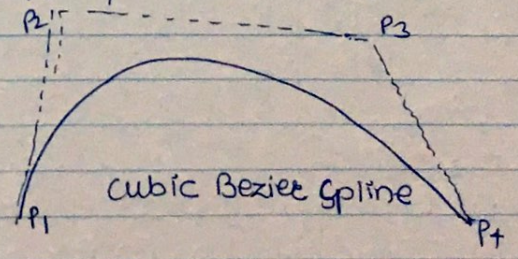
consider that the curve has n+1 control point := (x_k, y_k, z_k)... where k varies from 0 to n. The co-ordinates of these control points can be blended to produce position vector P(u), which give the path of approximating Bezier polynomial fn between P_0 & P_n.
The position vector can be given by,

$$P(u) = \sum_{k=0}^{n} P_k \, BEZ_{k,n}(u) \quad 0 \le u \le 1$$

Properties of Bezier Curve

1. The basis functions are real.
2. Bezier curve always passes through the first & last control points ie curve has same end points as the guiding polygon.
3. The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for 4 control points, the degree of polynomial is three ie cubic polynomial.
4. The curve generally follows the shape of the defining polygon.
5. The direction of the tangent vector at the end points is the same as that of the vector determined by first & last segments.
6. The curve lie entirely within the convex hull formed by four control points.
7. The convex hull property for bezier curve ensures that the polynomial smoothly follows the control points.
8. The curve exhibits the variation dimishing property. This means that the curve doesn't oscillate about any straight line more often than the defining polygon.
9. The curve is invariant under an affine transformation.

In cubic Bezier curve four control points are used to specify complete curve.
Unlike the B-spline curve, we do not add intermediate points & smoothly extend bezier curve, but we pick four more points & construct a second curve which can be attached to the first.
The second curve can be attached to the first curve smoothly by selecting appropriate control points.



Bezier curve begins at first control point & ends at the foweth control point. This means that if we want to connect two bezier curves, we have to make

first control point of the second curve match the last control point of the first curve.

We can also observe that at the start of the curve, the curve is tanget to the line connecting first & second control points. Similarly at the end of curve, the curve is tanget to the line connecting the third & foweth control point.

This means that, to join two Bezier curves smoothly, we have to place the third & the foweth control points of the first curve on the same line specified by the first & second control points of the second curve.

The Bezier matrix for periodic cubic polynomial is

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$\therefore P(u) = U.M_B.G_B.$

Where, $G_B = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$

$P(u) = U.M_B.G_B$

$= (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u)P_3 + u^3 P_4$

Therefore, the fowe blending fn for cubic Bezier curve are

$BEZ_{0,3}(u) = (1-u)^3$

$BEZ_{1,3}(u) = 3u(1-u)^3$

$BEZ_{2,3}(u) = 3u^2(1-u)$

$BEZ_{3,3}(u) = u^3$

Course & Semester :		Roll No.:	Date : / /201
Name of Subject :			Marks obtained :
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

Supplied By : SNJB

000554

Q. construct the Bezier curve of order 3 & with 4 polygon vertices A(1,1), B(2,3), C(4,3) & D(5,4)

→ The equation for Bezier curve is given as

$P(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u)P_3 + u^3 P_4$

where, P(u) is the point on the curve P_1, P_2, P_3 P_4

Let us take $u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$.

$P(0) = P_1 = (1,1)$

$\therefore P(\frac{1}{4}) = (1-\frac{1}{4})^3 P_1 + 3(\frac{1}{4})(1-\frac{1}{4})^2 P_2 + 3(\frac{1}{4})^2(1-\frac{1}{4}) P_3 + (\frac{1}{4})^3 P_4$

$= \frac{27}{64}(1,1) + \frac{27}{64}(2,3) + \frac{9}{64}(4,3) + \frac{1}{64}(6,4)$

$= \left[\frac{27}{64} \times 1 + \frac{27}{64} \times 2 + \frac{9}{64} \times 4 + \frac{1}{64} \times 6, \frac{27}{64} \times 1 + \frac{27}{64} \times 3 + \frac{9}{64} \times 3 + \frac{1}{64} \times 4\right]$

$= \left[\frac{123}{64}, \frac{139}{64}\right] = (1.9218, 2.1718)$

$\therefore P(\frac{1}{2}) = (1-\frac{1}{2})^3 P_1 + 3(\frac{1}{2})(1-\frac{1}{2})^2 P_2 + 3(\frac{1}{2})^2(1-\frac{1}{2}) P_3 + (\frac{1}{2})^2$

$= \left[\frac{25}{8}, \frac{23}{8}\right]$

$= (3.125, 2.875)$

$$P\left(\frac{3}{4}\right) = \left(1-\frac{3}{4}\right)^3 P_1 + 3\frac{3}{4}\left(1-\frac{3}{4}\right)^3 P_2 + 3\left(\frac{3}{4}\right)^2 \left(1-\frac{3}{4}\right) P_3 + \left(\frac{3}{4}\right)^3 P_4$$

$$= \left(\frac{289}{64}, \frac{217}{64}\right)$$

$$= (4.5156, 3.375)$$

$$P(1) = P_3 = (6, 4)$$

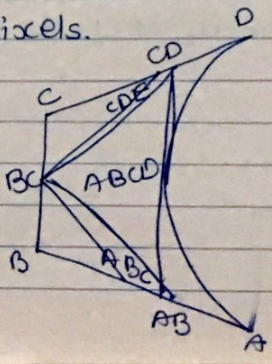
Midpoint approach

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 line segments & their midpoints ABC & BCD are determined. Finally these two midpoints are connected by line segments & its midpoint. ABCD is determined.

The point ABCD on the bezier curve divides the original curve into 2 sections. This makes the point A, AB, ABC, ABCD are the control points for the first section & the points ABCD, BCD, CD & D are the control points for the second section.

By consideeing two sections separately we can get two more sections for each separate section ie 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.



Algorithm

1. Get four control points say
   $A(x_A, y_A)$, $B(x_B, y_B)$, $C(x_C, y_C)$, $D(x_D, y_D)$
2. Divide the curve represented by points A, B, C & D intwo sections.

$$x_{AB} = (x_A + x_B)/2;$$
$$y_{AB} = (y_A + y_B)/2;$$
$$x_{BC} = (x_B + x_C)/2;$$
$$y_{BC} = (x_B + y_C)/2;$$
$$x_{CD} = (x_C + x_D)/2;$$
$$y_{CD} = (y_C + y_D)/2$$
$$x_{ABC} = (x_{AB} + x_{BC})/2;$$
$$y_{ABC} = (y_{AB} + y_{BC})/2;$$
$$x_{BCD} = (x_{BC} + x_{CD})/2;$$
$$y_{BCD} = (y_{BC} + y_{CD})/2;$$
$$x_{ABCD} = (x_{ABC} + x_{BCD})/2;$$
$$x_{ABCD} = (y_{ABC} + y_{BCD})/2;$$

3. Repeat the step 2 for section A, AB, ABC & ABCD & section ABCD, BCD, CD, D.
4. Repeat step 3 until we have section so short that they can be replaced by straight line
5. Replace small sections by straight lines
6. Stop.

* B-spline

which contains the Bernstein basis as a special case.

The B-spline basis is nonglobal. It is nonglobal.bcz each vertex B_i is associated with a unique basic function. Thus, each vertex affects the shape of the curve only over a range of parameter values where its associated basis fn is nonzero.

The Bspline basis also allows the order of basis fn & hence the degree of resulting curve is independent on the number of vertices It is possible to change the degree of the resulting curve without changing the no of vertices of the defining polygon.

If $P(u)$ be the position vectors along the curve as fn of parameter u, B-spline curve is given by,

$$P(u) = \sum_{i=1}^{n+1} B_i N_{i,k}(u) \quad u_{min} \le u \le u_{max}, \quad 2 \le K \le n \le +1$$

where,

B_i are the position vector of the n+1 defining polygon vertices & the $N_{i,k}$ are the normalized B spline basic fn. of order K, the basic fn $N_{i,k}(u)$ are defined as

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } x_i \le u \le x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

The value of x_i are the elements of knot vector satisfying the relation $x_i \leq x_{i+1}$.
the parameter u varies from u_{min} to u_{max} along the curve P(u).
The choice of knot vector has a significant influence on the Bspline basis fn $N_{i,k}(u)$ & hence on the resulting B-spline curve.
There are 3 types of knot vector uniform, open uniform, nonuniform

Properties of B-spline curve -
- The sum of Bspline basic fn for any parameter value u is 1.
ie $\sum_{i=1}^{n+1} N_{i,k}(u) = 1$
- each basic fn is positive or zero for all parameter values ie $N_{i,k} \geq 0$
- except for K=1 each basis fn has precisely one maximum value
- The maximum order of the curve is equal to the number of control point of defining polygon.
- The curve exhibits the variation diminishing property. Thus curve doesn't oscillate about any straight line more often than its defining polygon.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon
- The curve line with convex hull of its defining polygon.

Advantages of Bsplines over Bezier curve
- The degree of Bspline polynomial is independent on the no. of control points of defining polygon.
- Bspline allows local control over the curve surface bcz each control point affects the shape of curve only over a range of parameter values where its associated basis fn is nonzero.