Unit-5

3D Objects Representation

⊛ Representing Surfaces: Representation schemes for solid objects are often divided into two broad categories: Boundry representations (B-reps) which describes three-dimensional objects as a set of surfaces that seperate the object interior from the environment. This is one of the representation scheme. Polygon facets and spline patches are it's examples.

Another representation scheme is Space-partioning representation which are used to describe interior properties by partitioning the spatial region containing an object into a set of small, non-overlapping and contigious solids. A common space partitioning description for a 3D object is an octree representation.

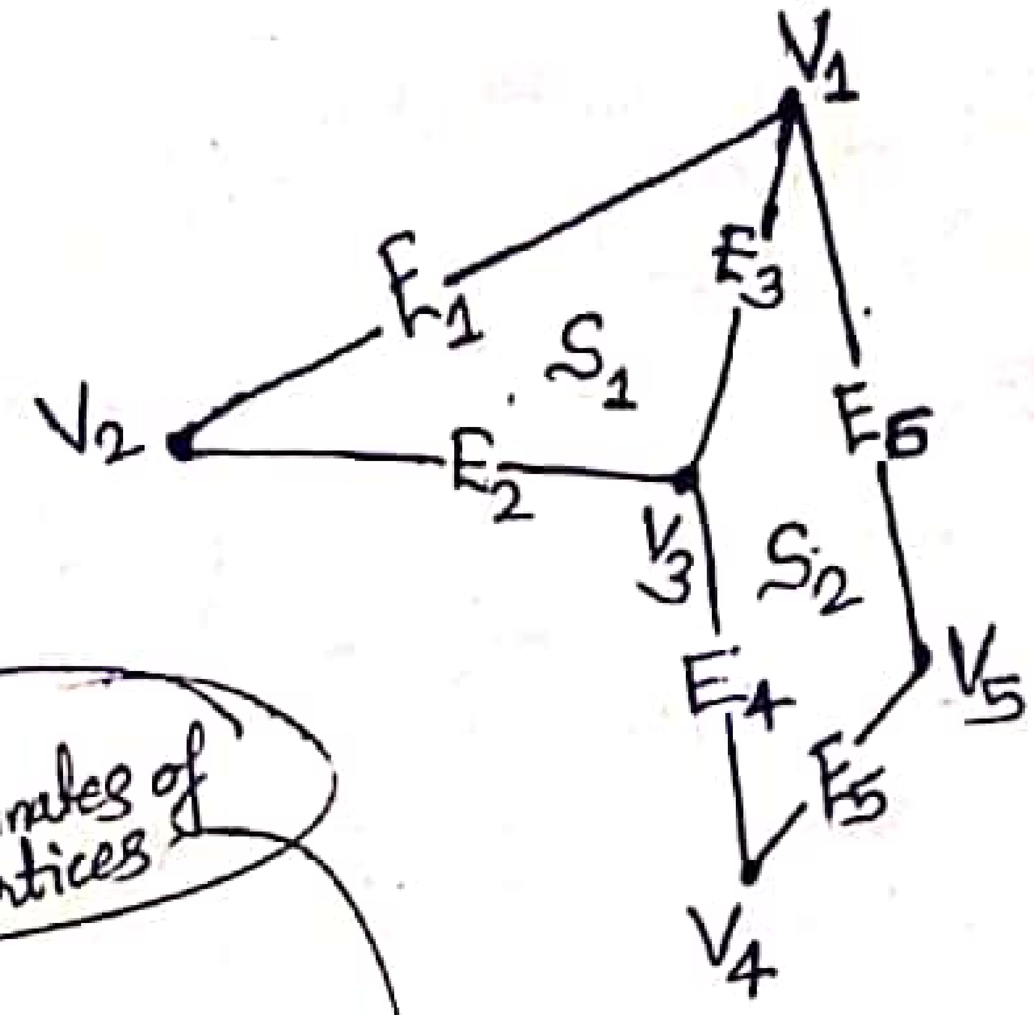
⊛. Polygon Surfaces:-

The most commonly used boundry representation for a 3D graphics object is a set of surface polygons that enclose the interior object. Many graphics systems store all object descriptions as set of surface polygons. This simplifies and speeds up the surface rendering and display of objects.

1) Polygon Tables:

Polygon table is the specification of polygon surfaces using vertex coordinates and other attributes. Polygon tables or Polygon data tables can be organized into two groups: geometric tables and attribute tables. For storing geometric data we create three lists; a vertex table, an edge, and a polygon table. Co-ordinate values for each vertex in the object are stored in vertex table. The edge table contains pointers back into the vertex table to identify the vertices for each polygon edge. And the polygon table contains pointers back into the edge table to identify the edges for each polygon. Attribute table contains qualatitative properties like degree of transparency, surface reflectivity etc.

Scanned with CamScanner



coordinates of vertices

Vertex Table
$V_1 : x_1, y_1, z_1$
$V_2 : x_2, y_2, z_2$
$V_3 : x_3, y_3, z_3$
$V_4 : x_4, y_4, z_4$
$V_5 : x_5, y_5, z_5$

Edge Table
$E_1 : V_1, V_2$
$E_2 : V_2, V_3$
$E_3 : V_3, V_1$
$E_4 : V_3, V_4$
$E_5 : V_4, V_5$
$E_6 : V_5, V_1$

Polygon-Surface table.
$S_1 : E_1, E_2, E_3$
$S_2 : E_3, E_4, E_5, E_6$.

fig. Geometric data table representation for two adjcent polygon surfaces, formed with six edges and five vertices.

$E_1 : V_1, V_2, S_1$
$E_2 : V_2, V_3, S_1$
$E_3 : V_3, V_1, S_1, S_2$
$E_4 : V_3, V_4, S_2$
$E_5 : V_4, V_5, S_2$
$E_6 : V_5, V_1, S_2$

fig. Edge table including pointers to polygon table.

Some consistency checks of the geometric data table are as follows:-

→ Every vertex is listed as an endpoint for at least 2 edges.

→ Every edge is a part of at least one polygon.

→ Every polygon is closed.

Scanned with CamScanner

ii) Polygon Mesh:

Using a set of connected polygonally bounded planor surfaces to represent an object, which may have curved surfaces or curved edges is called polygon mesh. The wireframe of such object can be displayed quickly to give general indication of the surface structure.

Realistic renderings can be produced by interpolating shading patterns ~~according~~ across the polygon surfaces to eliminate or reduce the presence of polygon edge boundries. Fast hardware-implemented polygon renders are capable of displaying upto 1,000,000 or more shaded triangles per second, including the application of surface texture and special lighting effects. Common types of polygon meshes are triangle strip and quadrilateral mesh.

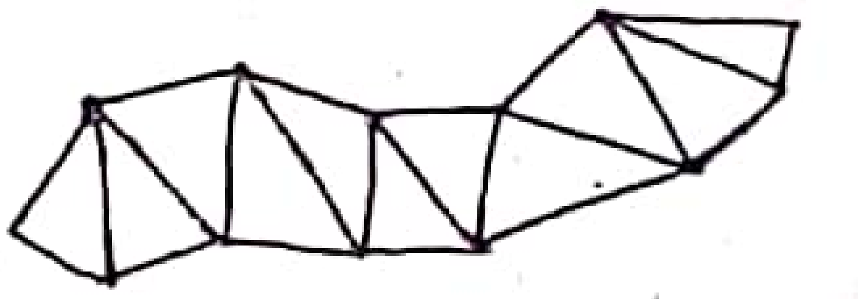


fig. A triangle strip formed with 11 triangles connecting 13 vertices.

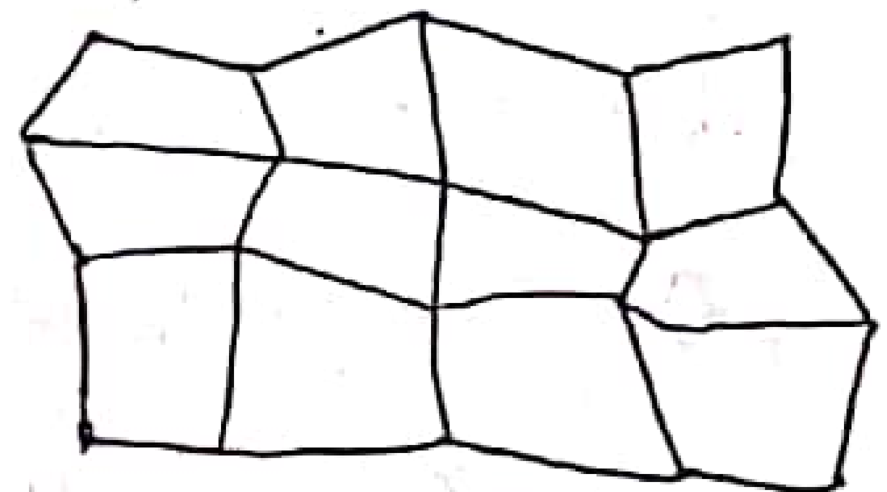


fig. A quadrilateral mesh connecting 12 quadrilaterals constructed from a 5 by 4 input vertex array.

iii) Plane Equations:

The equation for a plane surface is $Ax+By+Cz+D=0$. where (x,y,z) is any point on the plane, and the coefficients A, B, C and D are constants describing the spatial properties of the plane.

We can obtain values of A, B, C and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. For that we can select three successive polygon vertices (x_1,y_1,z_1), (x_2,y_2,z_2) and (x_3,y_3,z_3) and solve the following set of simultaneous linear plane equations for the ratios $\frac{A}{D}$, $\frac{B}{D}$ and $\frac{C}{D}$.

$$\frac{A}{D}x_k+\frac{B}{D}y_k+\frac{C}{D}z_k=-1 \text{ ; for } k=1,2,3.$$

Scanned with CamScanner

The solution for this set of equations can be obtained in determinant form, using Cramer's rule as;

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \quad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix} \quad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

$$\& \quad D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Expanding the determinants, we can write the calculations for the plane coefficients in the form;

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$D = -x_1(y_2 z_3 - y_3 z_2) - x_2(y_3 z_1 - y_1 z_3) - x_3(y_1 z_2 - y_2 z_1)$$

⇒ We can identify the point as either inside or outside the plane surface according to the sign (negative or positive) of

$$A_x + B_y + C_z + D$$

If $A_x + B_y + C_z + D < 0$, then the point (x, y, z) is inside the surface.

If $A_x + B_y + C_z + D > 0$, then the point (x, y, z) is outside the surface.

These inequality tests are valid in a right handed Cartesian system, provided the plane parameters A, B, C and D were calculated using vertices selected in a counter clockwise order when viewing the surface in an outside-to-inside direction.

<u>Note</u>: If $A_x + B_y + C_z + D \neq 0$, then it means the point is not on the plane.

Scanned with CamScanner

iv) Surface normal and Spatial orientation of surfaces:-

A normal is a term used in computer graphics to describe the orientation of a geometric object at a point on the surface. The normal to a surface at point P can be seen as the vector perpendicular to a plane tangent to the surface at P. At the same time, the direction of this vector determines the orientation of the surface. In the case of polygons, this direction is usually determined by the right hand rule. Normal plays an important role in shading where they are used to compute the brightness of the objects.

Spatial Orientation of a polygon face is the vertex coordinates values and the equations of the polygon surfaces. The general equation of a plane containing a polygon is

$$A_x + B_y + C_z + D = 0.$$

where (x, y, z) is any point on plane.

⊛ Wireframe Representation:-

If the object is defined only by a set of nodes, and a set of connecting the nodes, then the resulting object representation is called a wireframe model. In this method, a 3D object is represented as a list of straight lines, each of which is represented by its two end points, (x_1, y_1, z_1) and (x_2, y_2, z_2). This method only shows skeletal structure of objects.

A wireframe model consists of edges, vertices and polygons. The edges maybe curved or straight line segments.

Advantages and Disadvantages:-

Wireframe model are used in engineering applications. They are easy to construct. If they are compound of straight lines they are easy to clip and ~~mainupe~~ manipulate.

But for building realistic models, we must use a very large number of polygons to achieve the illusions of roundness and smoothness.

⊛ **Blobby Objects:-** The objects that do not maintain a fixed shape but change their surface characteristics in certain motions are known as blobby objects. For example: molecular structures, water droplets, melting objects etc. Several models have been developed for representing blobby objects as distribution functions over a region of space. One way is to use Gaussian density function. Other methods for generating blobby objects use quadratic density function.

#Representing Curves:

In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all the time and we need to draw curves many times to draw an object.

A curve is an infinitely large set of points. Curves are broadly classified into three categories- explicit, implicit and parametric curves.

i) Implicit curves → Implicit curve representations define the set of points on a curve employing a procedure that can test to see if a point is on the curve. Usually, an implicit curve is defined by an implicit function of the form:

$$f(x,y) = 0 \quad (\text{In } 2D)$$
$$f(x,y,z) = 0 \quad (\text{In } 3D)$$

A common example is the circle, whose implicit representation is;

$$x^2 + y^2 - R^2 = 0.$$

ii) Explicit curves → A mathematical function $y = f(x)$ can be plotted as a curve. Such a function is the explicit representation of curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function.

Parametric Curves → Curves having parametric form are called parametric curves. A two-dimensional parametric curve has the following form:

$$P(t) = f(t), g(t) \text{ or } P(t) = x(t), y(t).$$

The functions of f and g becomes the (x,y) coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval $[a,b]$ normally $[0,1]$.

@ Parametric cubic curves:

Once we decide parametric polynomial curves, we must choose the degree of the curve. If we choose a high degree, there is more danger that the curve will become rougher. On other hand, if we pick too low degree, we may not have enough parameters with which to work.

Algebraic representation of parametric curves

→ Parametric linear curve: $p(u) = au + b$

$$x = a_x u + b_x$$
$$y = a_y u + b_y$$
$$z = a_z u + b_z.$$

→ Parametric cubic curve: $p(u) = au^3 + bu^2 + cu + d$

$$x = a_x u^3 + b_x u^2 + c_x u + d_x$$
$$y = a_y u^3 + b_y u^2 + c_y u + d_y$$
$$z = a_z u^3 + b_z u^2 + c_z u + d_z.$$

Advantages

→ More degrees of freedom.
→ Directly transformable.
→ Dimension independent
→ No infinite slope problems.
→ Seperates dependent and independent variables.
→ Inherently bounded.
→ Easy to express in vector and matrix form.
→ Common form for many curves and surfaces.

Scanned with CamScanner

(b) Spline Representation:

Spline means a flexible strip used to produce a smooth curve through a designated set of points. Several small weights are distributed along the length of the strip to hold it in position on the drafting table as the curve is drawn.

We can mathematically describe such a curve with a piecewise cubic polynomial function ie, spline curves. Then a spline surface can be described with 2 sets of orthogonal spline curves. Splines are used in graphics applications to design curve and surface shapes.

(c) Cubic spline interpolation:-

This method gives an interpolating polynomial that is smoother and has smaller error than some other interpolating polynomials such as Lagrange polynomial and Newton polynomial. Cubic polynomials provide a reasonable compromise between flexibility and speed of computation. Cubic spline requires less calculations compared to higher order polynomials and consume less memory. They are also more flexible for modeling arbitrary curve shape.

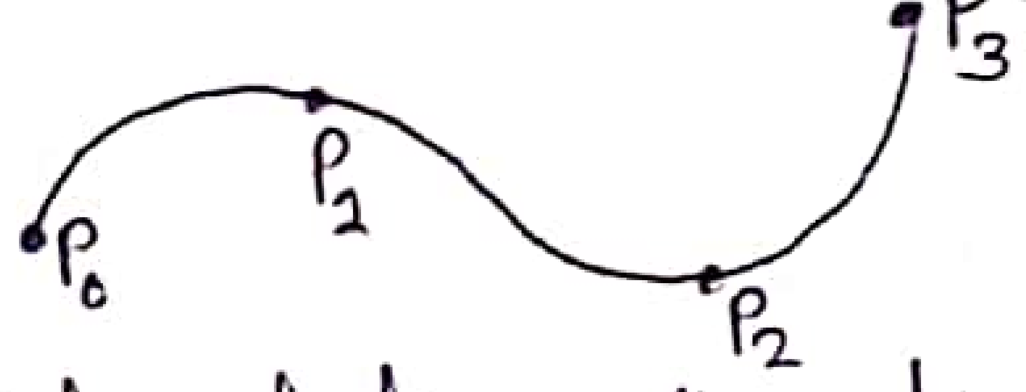


fig: Interpolation with cubic splines between 4 control points.

Suppose we have n+1 control points specified with coordinates

$$P_k = (x_k, y_k, z_k), \quad k = 0, 1, 2, \ldots n.$$

The parametric cubic polynomial that is to be filled between each pair of control points with the following set of equations.

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$
$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \qquad (0 \leq u \leq 1)$$
$$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$$

Values for coefficients a,b,c,d are determined by setting enough boundary conditions at control-point positions.

Scanned with CamScanner

(d) Hermite Curves:-

Hermite curves are very easy to calculate but also very powerful. They are used to smoothly interpolate between key-points. Hermite curves work on any number of dimensions. To calculate hermite curve we need the following vectors:

P1: the start point of the curve.

T1: The tangent to how the curve leaves the start point.

P2: The endpoint of the curve.

T2: The tangent to how the curve meets the endpoint.

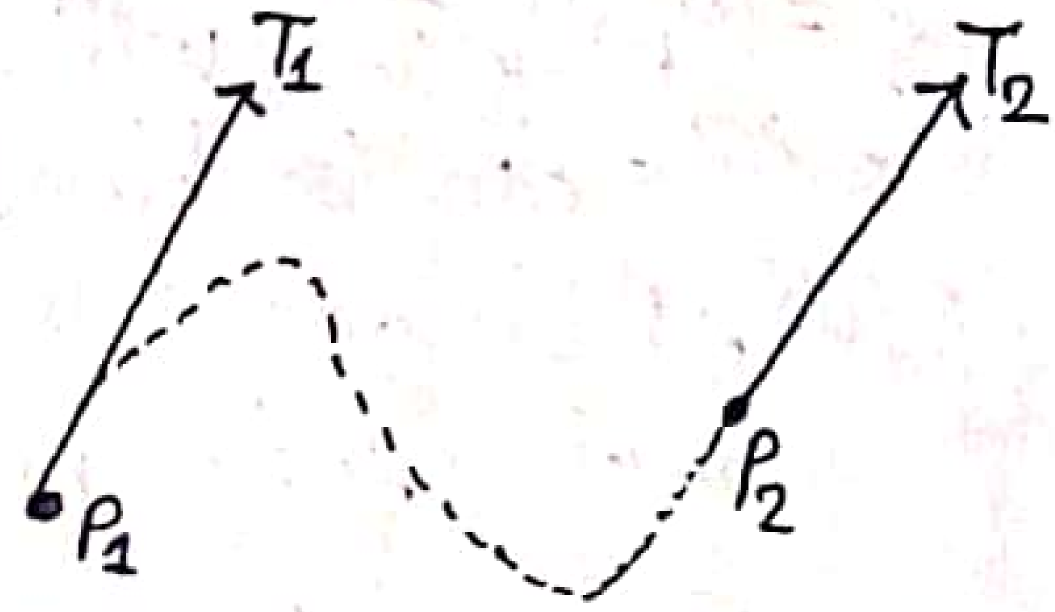


fig. Hermite curve

Matrix form of Hermite Curve:-

$$S = \begin{bmatrix} s^3 \\ s^2 \\ s \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix} \qquad h = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Vector S: The interpolation-point and its powers upto 3.

Vector C: The parameters of our hermite curve.

Matrix h: The matrix form of the 4 hermite polynomials.

⇒ To calculate a point on the curve we build the vector S, multiply it with the matrix h and then multiply with C.

i.e, $P = S \times h \times C$.

(e) Bezier Curves and Surfaces:

This spline approximation method was developed by French engineer Pierre Bezier. Beizer splines have a number of properties that make them highly useful and convinent for curve and surface design. They are also easy to implement. For these reasons Beizer splines are widely available in various CAD systems.

Scanned with CamScanner

Bezier Curves:

This method employs <u>control points</u> and produces an approximating curve. Bezier curve can be specified with boundary conditions, with a characterizing matrix, or with blending functions. For general Bezier curves, the blending function specification is the most convinent.

Suppose we are given n+1 control-point positions: $P_k = (x_k, y_k, z_k)$, with k varying from 0 to n. These coordinate points can be blended to produce the following position vector P(u), which describes the path of an approximating Bezier polynomial function between P_0 and P_n.

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

The Bezier blending functions $BEZ_{k,n}(u)$ are the Bernstein polynomial.

$$BEZ_{k,n}(u) = C(n,k)\, u^k (1-u)^{n-k}$$

where $C(n,k)$ are the binomial coefficients;

$$C(n,k) = \frac{n!}{k!\,(n-k)!}$$

The equation $p(u) = \sum_{k=0}^{n} P_k \, BEZ_{k,n}(u)$, where $0 \le u \le 1$ represents a set of three parametric equations for individual curve condition.

$$x(u) = \sum_{k=0}^{n} x_k \, BEZ_{k,n}(u)$$

$$y(u) = \sum_{k=0}^{n} y_k \, BEZ_{k,n}(u)$$

$$z(u) = \sum_{k=0}^{n} z_k \, BEZ_{k,n}(u).$$

Two points generate simple Bezier, three points generate a parabola, four points a cubic curve. i.e, (n-1) degree of polynomial equation for the n control points.

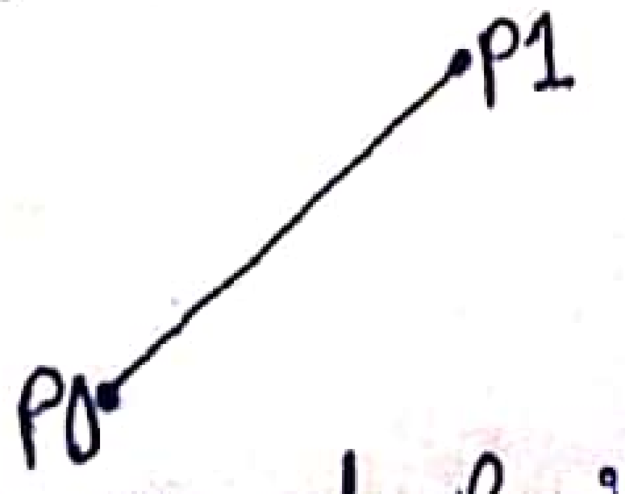


fig. Simple Bezier Curve

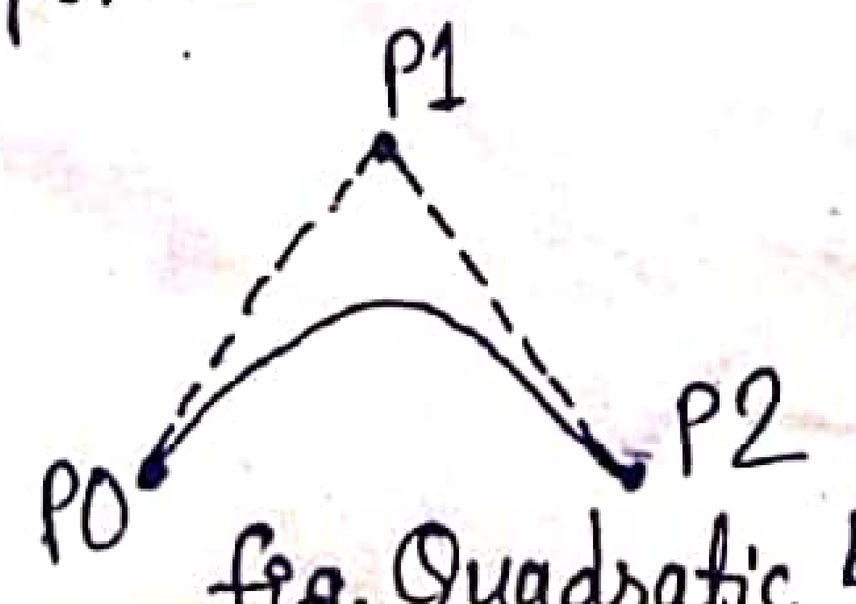


fig. Quadratic Bezier Curve

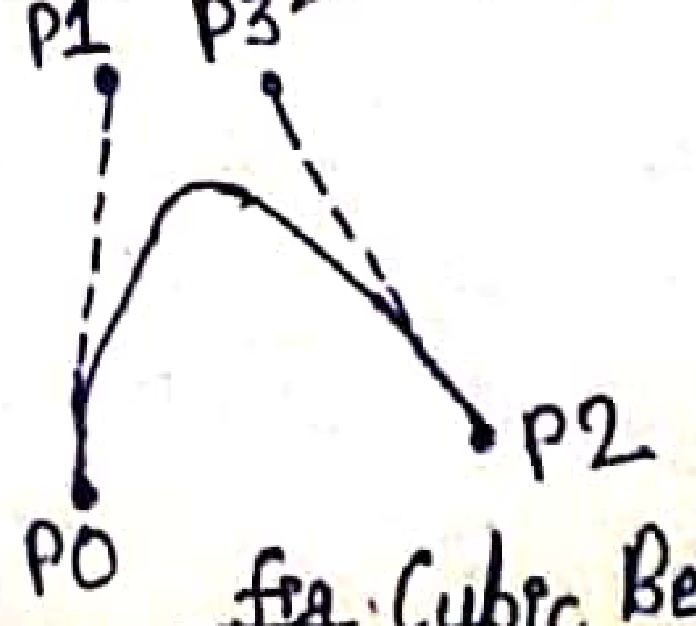


fig. Cubic Bezier Curve

Properties of Bezier Curve:

i) It always passes through the first and last control points. That is, the boundry conditions at the two ends of the curve are;

$$P(0) = P_0$$
$$P(1) = P_n.$$

ii) It lies within the convex hull (convex polygon boundry) of the control points.

iii) The degree of 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 3 i.e, cubic polynomial.

iv) A Bezier curve generally follows the shape of the defining polygon.

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

Example: Construct the Beizer curve of order 3 and with 4 polygon vertices A(1,1), B(2,3), C(4,3) and D (6,4).

Solution:

The equation for the Beizer curve is given as.

$$P(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u)P_3 + u^3 P_4, \quad \text{for } 0 \leq u \leq 1.$$

$0 \leq u \leq 1$ condition को आवश्यक लिन

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

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

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

i.e, coordinates 1st given in Q

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

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

$$= \left(\frac{27}{64}\times1 + \frac{27}{64}\times2 + \frac{9}{64}\times4 + \frac{1}{64}\times6, \ \frac{27}{64}\times1 + \frac{27}{64}\times3 + \frac{9}{64}\times3 + \frac{1}{64}\times4\right)$$

$$= \left(\frac{123}{64}, \frac{139}{64}\right)$$

$$= (1.9218, 2.1718).$$

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

$$=\frac{1}{8}(1,1)+\frac{3}{8}(2,3)+\frac{3}{8}(4,3)+\frac{1}{8}(6,4).$$

$$=\left(\frac{1}{8}\times1+\frac{3}{8}\times2+\frac{3}{8}\times4+\frac{1}{8}\times6,\ \frac{1}{8}\times1+\frac{3}{8}\times3+\frac{3}{8}\times3+\frac{1}{8}\times4\right)$$

$$=\left(\frac{25}{8},\frac{23}{8}\right)$$

$$=(3.125,\ 2.875).$$

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

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

$$=\left(\frac{1}{64}\times1+\frac{9}{64}\times2+\frac{27}{64}\times4+\frac{27}{64}\times6,\ \frac{1}{64}\times1+\frac{9}{64}\times3+\frac{27}{64}\times3+\frac{27}{64}\times4\right)$$

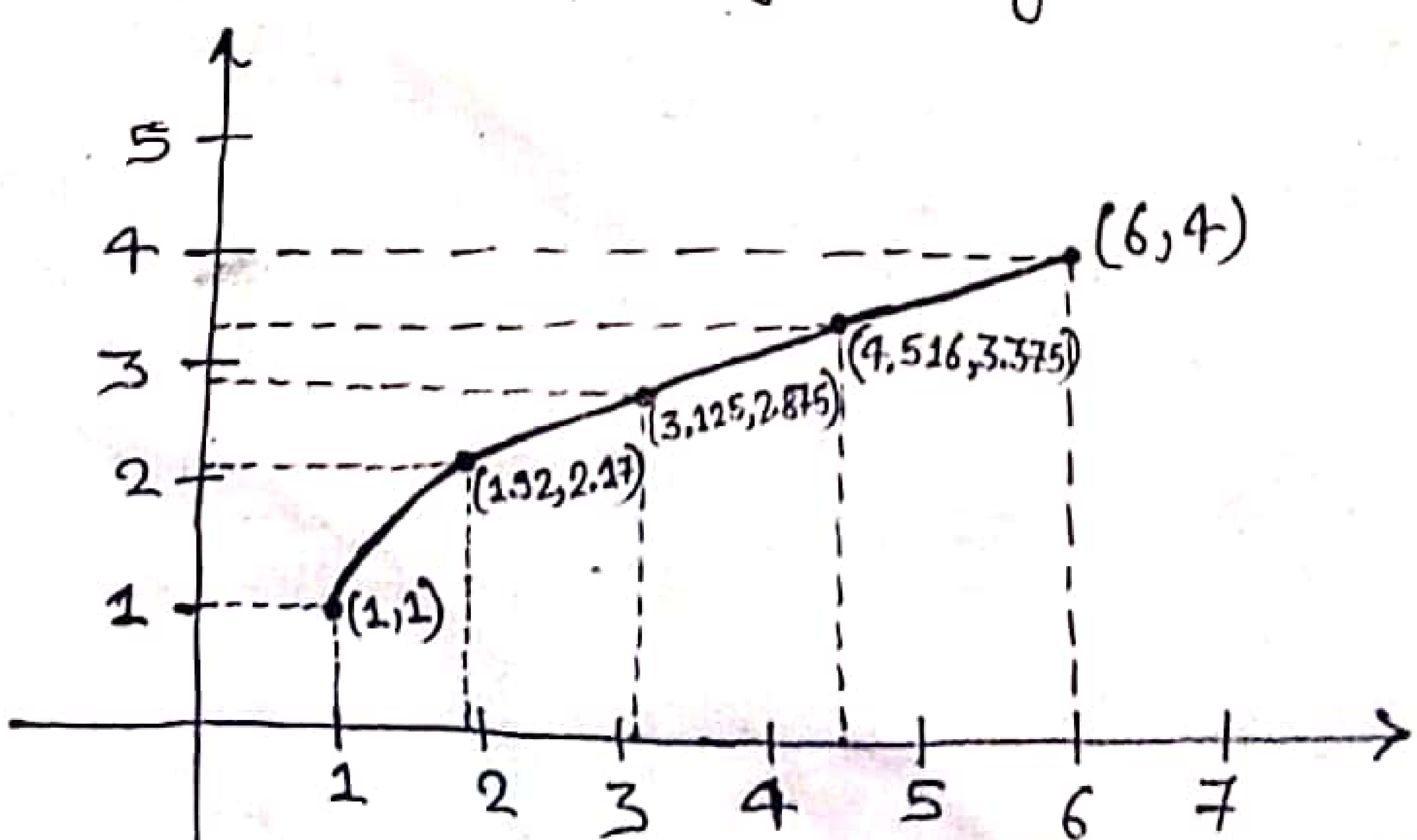
$$=\left(\frac{289}{64},\frac{217}{64}\right)$$

$$=(4.5156,\ 3.375).$$

i.e, last coordinate given in question

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

The graph below shows that calculated points of the Beizer curve and curve passing through it.



Bezier Surface:

Two sets of orthogonal Bezier curves can be used to design an object surface by specifying by an input mesh of control points. The parametric vector function for the Bezier surface is formed as the cartesian product of Bezier blending functions:

$$P(u,v) = \sum_{j=0}^{m} \sum_{k=0}^{n} P_{j,k} \, BEZ_{j,m}(v) \, BEZ_{k,n}(u).$$

with $P_{j,k}$ specifying the location of the $(m+1)$ by $(n+1)$ control points. A smooth transition is assured by establishing both zero-order and first-order continuity at the boundry line. Zero-order continuity is obtained by matching control points at the boundry. First-order continuity is obtained by choosing control points along a straight line across the boundry and by maintaining a constant ratio of collinear line segments for each set of specified control points across section boundaries

(f) B-Spline Curves and Surface:

These are the most widely used class of approximating splines. B-Spline have following two advantages over Bezier Splines:

i) The degree of a B-spline can be set independently of the number of control points.

ii) B-splines allow local control over the shape of a spline curve or surface.

→ The disadvantage is that B-splines are more complex than Bezier splines.

B-Spline Curves:- The designation "B" stands for Basis, so the full name of this approach is basis spline which contains the Bernstein basis as a special case.

There is most widely used class of approximating splines. B-spline has a general expression for the calculation of coordinate positions along a curve in a blending function as:

$$P(u) = \sum_{k=0}^{n} B_i N_{i,k}(u) \;, \quad u_{min} \le u \le u_{max} \;, \quad 2 \le k \le n+1;$$

Scanned with CamScanner

Where B_i are the position vectors of the $n+1$ defining polygon vertices and the $N_{i,k}$ are the normalized B-spline basis functions. For the ith normalized B-spline basis function of order k, the basis function $N_{i,k}(u)$ are defined as

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

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

The values of x_i are the elements of a knot vector satisfying the relation $x_i \le x_{i+1}$. 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 B-spline basis functions $N_{i,k}(u)$ and hence on the resulting B-spline curve. There are three types of knot vector: uniform, open uniform and on uniform.

→ In uniform knot vector, individual values are equally spaced. For example [0 1 2 3 4].

For a given order k, uniform knot vectors give periodic uniform basis functions;

$$N_{i,1}(u) = N_{i-1,k}(u-1) = N_{i+1,k}(u+1)$$

→ An open uniform knot vector has multiplicity of knot values at the ends equal to the order k of the B-spline basis function. Internal ~~values are~~ knot values are equally spaced. Examples are:-

$k=2$ [0 0 1 2 3 3]

$k=3$ [0 0 0 1 2 3 3 3]

$k=4$ [0 0 0 0 1 2 2 2 2]

Generally an open uniform knot vector is given by, $x_i = 0$ where $1 \le i \le k$.

$x_i = i-k$ where $k+1 \le i \le n+1$

$x_i = n-k+2$ where $n+2 \le i \le n+k+1$

The resulting B-spline curve is a Bezier curve.

Properties of B-spline curve:-

i) The degree of B-spline polynomial is independent on the number of vertices of defining polygon.

ii) The maximum order of the curve is equal to the number of vertices of defining polygon.

iii) Each basis function is positive or zero for all parameter values.

iv) Each basis function has precisely one maximum value, except for $k=1$.

v) The sum of B-spline basis functions for any parameter value is 1.

vi) B-spline allows local control over the curve surface because each vertex affects the shape of a curve.

Example:- Construct the B-spline curve of order 4 and with 4 polygon vertices $A(1,1)$, $B(2,3)$, $C(4,3)$ and $D(6,2)$.

Solution:

Here, $n=3$ and $k=4$, we have open uniform knot vector as $X = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]$ and we have basis functions for various parameters are as follows:-

$0 \le u < 1$

$N_{4,1}(u) = 1; \quad N_{i,1}(u) = 0, \ i \neq 4$

$N_{3,2}(u) = (1-u); \quad N_{4,2}(u) = u, \ N_{i,2}(u) = 0, \ i \neq 3,4.$

$N_{2,3}(u) = (1-u)^2; \quad N_{3,3}(u) = 2u(1-u);$

$N_{4,3}(u) = u^2; \quad N_{i,3}(u) = 0, \ i \neq 2,3,4.$

$N_{1,4}(u) = (1-t)^3; \quad N_{2,4}(u) = u(1-u)^2 + 2u(1-u)^2 = 3u(1-u)^2;$

$N_{3,4}(u) = 2u^2(1-u) + (1-u)u^2 = 3u^2(1-u); \quad N_{4,4}(u) = u^3.$

The parametric B-Spline is

$$P(u) = AN_{1,4}(u) + BN_{2,4}(u) + CN_{3,4}(u) + DN_{4,4}(u).$$

$$\boxed{\therefore P(u) = (1-u)^3 A + (1-u)^2 B + (1-u) C + u^3 D.}$$

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

$\therefore P(0) = A = (1,1).$

Scanned with CamScanner

$$P\left(\frac{1}{4}\right) = \left(1-\frac{1}{4}\right)^3 A + 3\times\frac{1}{4}\left(1-\frac{1}{4}\right)^2 B + 3\left(\frac{1}{4}\right)^2\left(1-\frac{1}{4}\right)C + \left(\frac{1}{4}\right)^3 D.$$

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

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

$$= \left(\frac{123}{64}, \frac{137}{64}\right)$$

$$= (1.9218, 2.14).$$

$$P\left(\frac{1}{2}\right) = \left(1-\frac{1}{2}\right)^3 A + 3\times\frac{1}{2}\left(1-\frac{1}{2}\right)^2 B + 3\left(\frac{1}{2}\right)^2\left(1-\frac{1}{2}\right)C + \left(\frac{1}{2}\right)^3 D$$

$$= \frac{1}{8}(1,1) + \frac{3}{8}(2,3) + \frac{3}{8}(4,3) + \frac{1}{8}(6,2)$$

$$= \left(\frac{1}{8}\times 1 + \frac{3}{8}\times 2 + \frac{3}{8}\times 4 + \frac{1}{8}\times 6,\ \frac{1}{8}\times 1 + \frac{3}{8}\times 3 + \frac{3}{8}\times 3 + \frac{1}{8}\times 2\right)$$

$$= \left(\frac{25}{8}, \frac{21}{8}\right)$$

$$= (3.125, 2.625).$$

$$P\left(\frac{3}{4}\right) = \left(1-\frac{3}{4}\right)^3 A + 3\times\frac{3}{4}\left(1-\frac{3}{4}\right)^2 B + 3\left(\frac{3}{4}\right)^2\left(1-\frac{3}{4}\right)C + \left(\frac{3}{4}\right)^3 D.$$

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

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

$$= \left(\frac{289}{64}, \frac{163}{64}\right)$$

$$= (4.5156, 2.5468)$$

$$\therefore P(1) = D = (6,2)$$

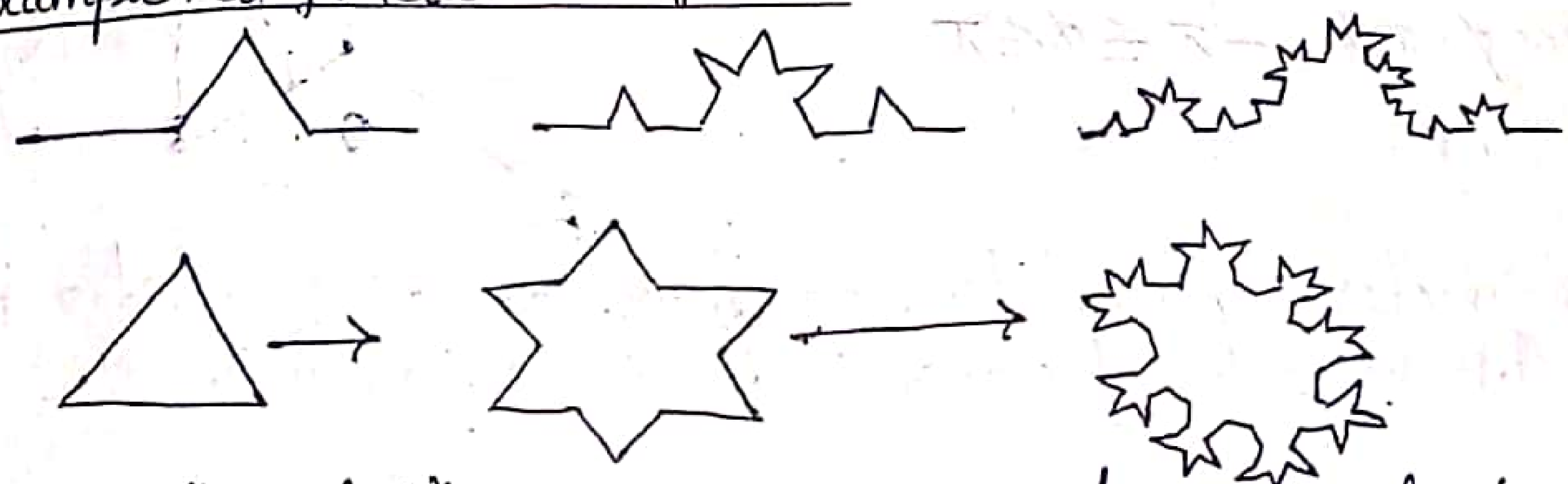
<u>Note</u>:- We can draw graph in similar way as we did finally in Bezier spline.

(8). Fractals and it's applications:-

Those objects which are self-similar at all resolutions are called fractal objects. Most of the natural objects such as trees, mountains, coastlines etc. are considered as fractal objects because no matter how far or how close one ~~clicks~~ looks at them, they always appear somewhat similar. Fractal objects can also be generated recursively by applying the same transformation fraction to an object.

e.g. Scale down + rotate + translate.

For example, a fractal Snowflake:



The name "Fractal" comes from it's property: fractional dimension.

Applications:

i) Fractals are used to predict or analyze various biological phenomena such as growth pattern of bacteria.

ii) Fractals are used to capture images of complex structures such as clouds.

iii) Speaking of imaging is one of the most important use of fractals with regards to image compressing.

iv) It is widely used in image synthesis and computer animation.

(*). Quadric Surface:-

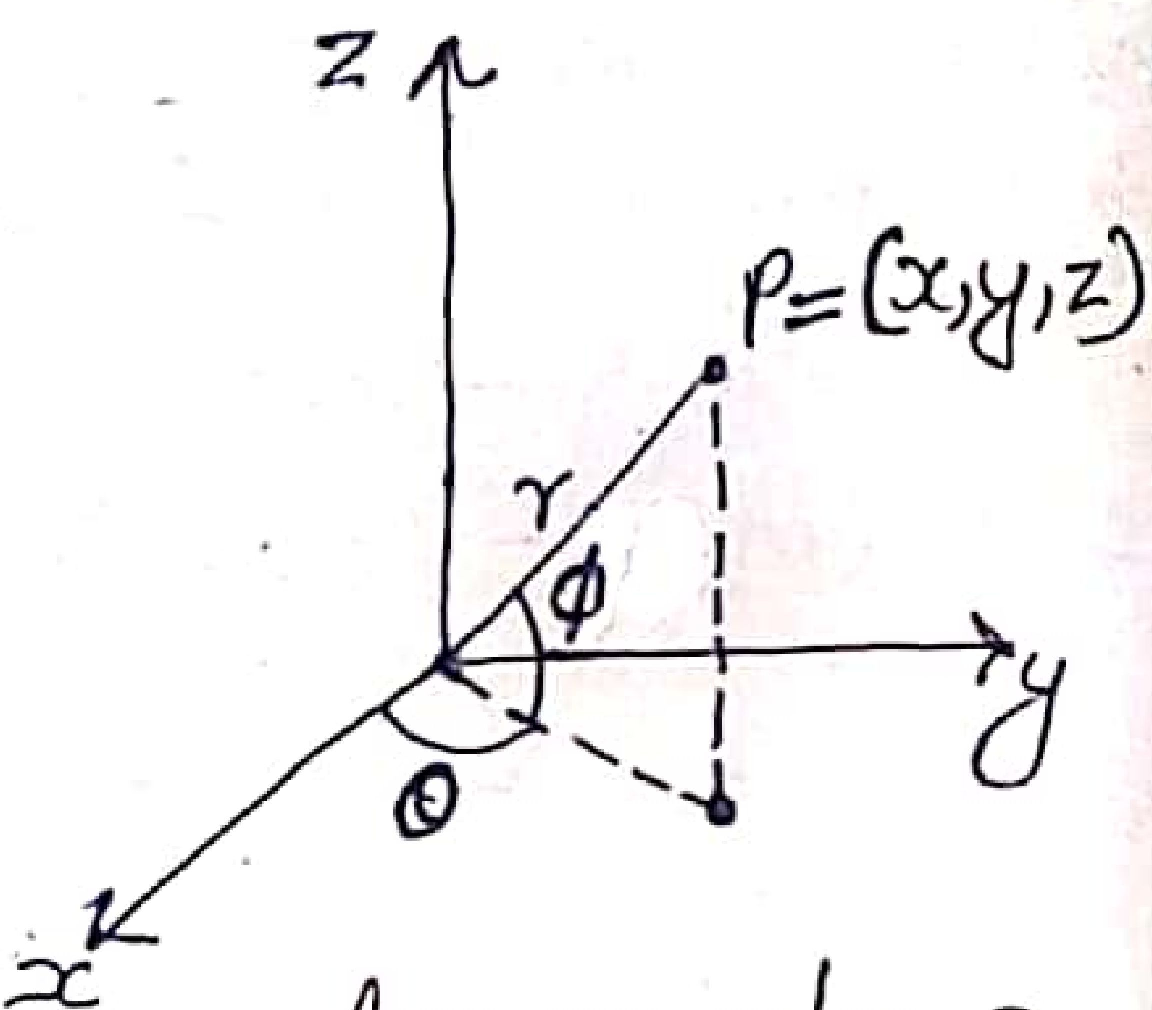
A frequently-used class of objects are the quadric surfaces, which are described with second-degree equations (quadratics). They include spheres, ellipsoids, tori, paraboloids and hyperboloids. Quadric surfaces; particularly spheres and ellipsoids are common elements of graphics scenes.

Scanned by TapScanner
Scanned with CamScanner

Sphere: A spherical surface with radius r centered on the coordinate's origin is defined as the set of points (x, y, z) that satisfy the equation;

$$x^2+y^2+z^2=r^2.$$

The spherical surface can be represented in parametric form by using latitude and longitudes angles as;



$x = r\cos\phi\cos\theta, \quad \frac{-\pi}{2} \leq \phi \leq \frac{\pi}{2}$

$y = r\cos\phi\sin\theta, \quad -\pi \leq \phi \leq \pi$

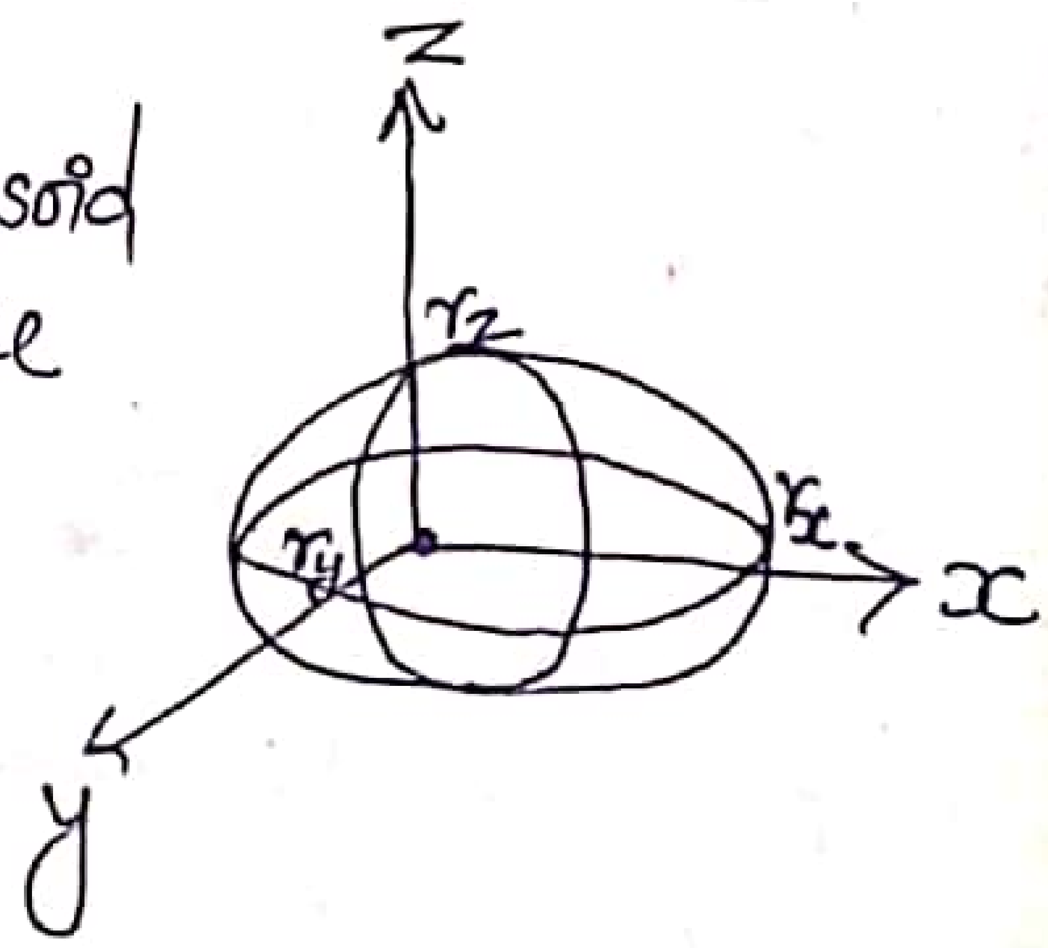
$z = r\sin\phi.$

The parameter representation in above equation provides a symmetric range for the angular parameter θ and ϕ.

Ellipsoid: Ellipsoid surface is an extension of a spherical surface where the radius in three mutually perpendicular directions can have different values. The cartesian representation for points over the surface of an ellipsoid centred on the origin is;

$$\left(\frac{x}{r_x}\right)^2+\left(\frac{y}{r_y}\right)^2+\left(\frac{z}{r_z}\right)^2=1.$$

The parametric representation for the ellipsoid in terms of the latitude angle ϕ and the longitude angle θ is;



$x = r_x\cos\phi.\cos\theta, \quad \frac{-\pi}{2} \leq \phi \leq \frac{\pi}{2}$

$y = r_y\cos\phi\sin\theta, \quad -\pi \leq \phi \leq \pi$

$z = r_z\sin\phi.$

Scanned with CamScanner