CSC209: Computer Graphics

Unit 5 – 3D Object Representation

SUSAN SUNUWAR

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 Curves
  - Implicit Curves
  - Parametric Curves

Explicit Curves

- A mathematical function $y = f(x)$ can be plotted as a curve, such a function is the explicit representation of curve. $[y = mx + c]$
- As it cannot represent vertical lines, the explicit representation is not general.
- For each value of x, only a single value of y is normally computed by the function

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 if the form:

$$f\ (x, y) = 0$$

$$f\ (x, y, z) = 0$$

$$x^2 + \ y^2 \ - \ r^2 = 0$$

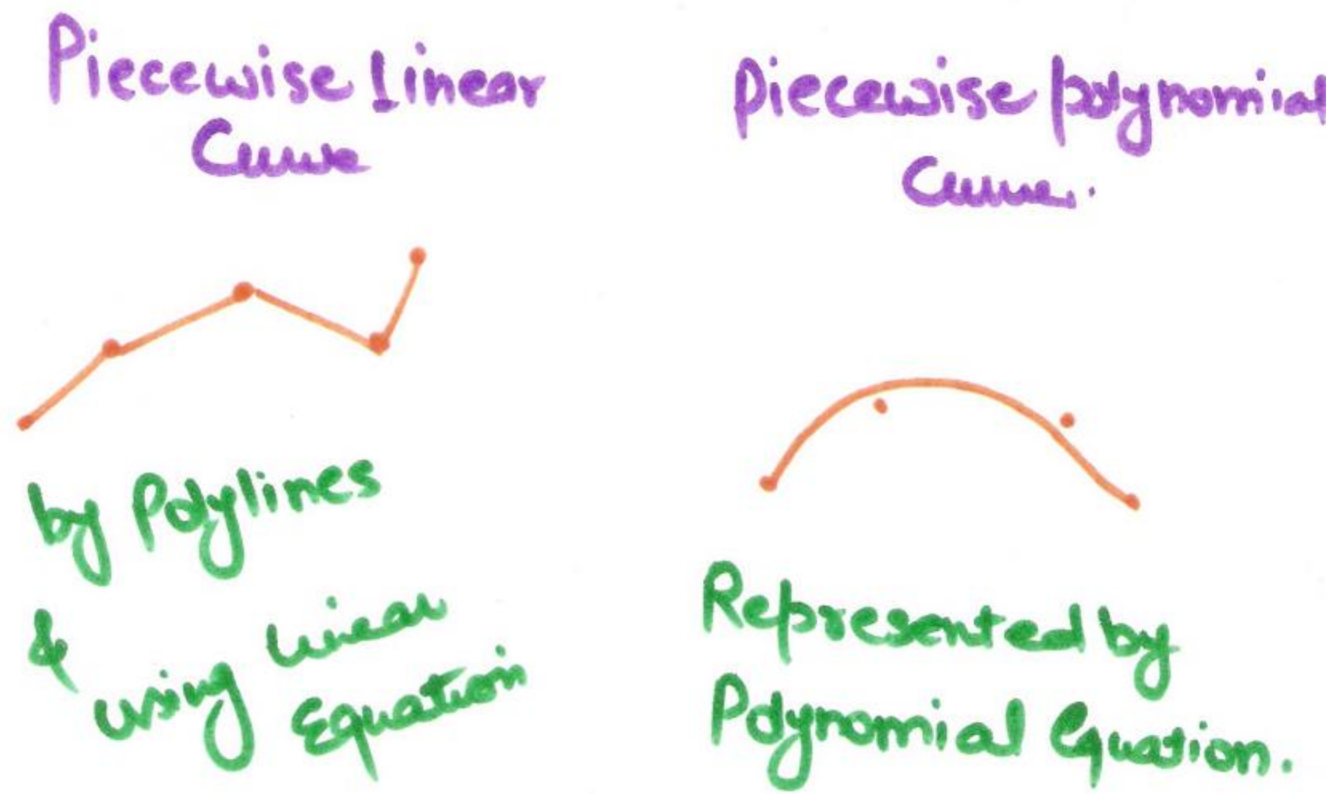
Parametric Curves

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

$$P(t) = f\ (t), g(t)\ or\ P(t) = x(t), y(t)$$

- 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 $[0,1]$

- A curve is approximated by a piecewise polynomial curve instead of piecewise linear curve.



Parametric Cubic Curves

- Low degree - no flexibility
- High degree – complex
- 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

$$p(u) = au^3 + bu^2 + cu + d$$
$$x = a_x u^3 + b_x u^2 + c_x u + d_x$$
$$y = a_y u^3 + b_y u^2 + c_y u + d_y$$
$$z = a_z u^3 + b_z u^2 + c_z u + d_z.$$

Parametric Cubic Curves

- A parametric cubic curve is defined as

$$P(t) = \sum_{i=0}^{3} a_i t^i \qquad 0 <= t <= 1 \text{ ------ (i)}$$

Expanding equation (i) yield

$$P(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \text{------------------------------ (ii)}$$

This equation is separated into three components of P (t)

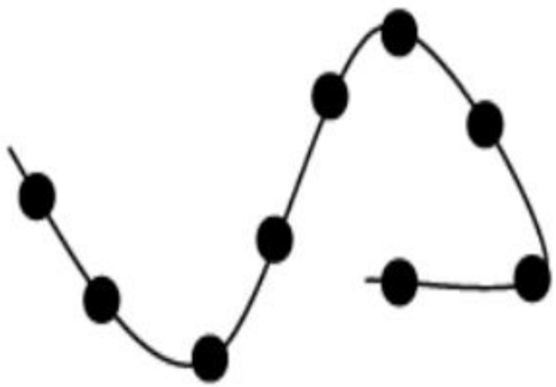
$$x(t) = a_{3x}t^3 + a_{2x}t^2 + a_{1x}t + a_{0x}$$
$$y(t) = a_{3y}t^3 + a_{2y}t^2 + a_{1y}t + a_{0y}$$
$$z(t) = a_{3z}t^3 + a_{2z}t^2 + a_{1z}t + a_{0z} \text{ --------------------(iii)}$$

Spline Representation

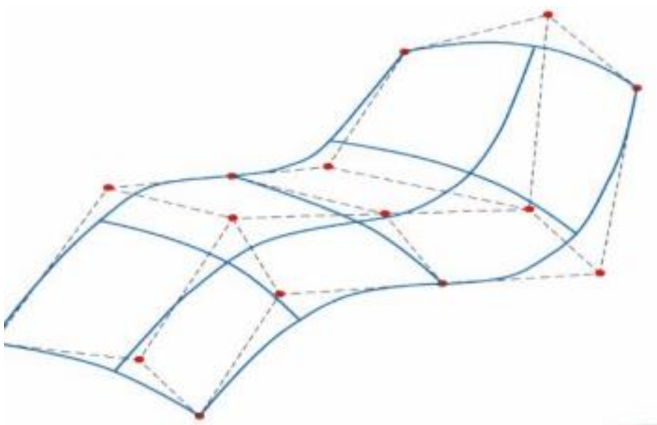
- Spline means a flexible strip used to produce a smooth curve through a designated set of points.
- We can mathematically describe such a curve with a piecewise cubic polynomial function ,i.e., spline curves.

Spline Representation

- We can mathematically describe a curve with a piecewise cubic polynomial function whose first and second derivatives are continuous across the various curve sections.
- Splines are used in graphics applications to design curve and surface shapes, to digitize drawings for computer storage, and to specify animation paths for the objects or image. Typical CAD applications for splines include the design of automobile bodies, aircraft and spacecraft surfaces, and ship hulls.



Fig(a): A spline curve through nine control points



Fig(b): A spline surface through 15 control points

Control points

We specify a spline curve by giving a set of coordinate positions, called ***control points***, which indicates the general shape of the curve These, control points are then fitted with piecewise continuous parametric polynomial functions in one of two ways.

- ***Interpolation curve***: The polynomial sections are fitted by passing the curve through each control points. Interpolation curves are commonly used to digitize drawings or to specify animation paths.
- ***Approximation curve***: The polynomials are fitted to the general control-point path without necessarily passing through any control point. Approximation curves are primarily used as design tools to structure object surfaces.

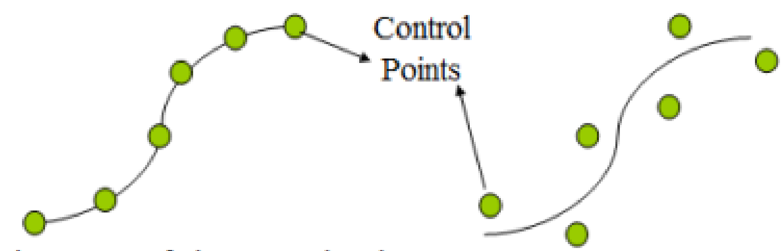


Fig. A set of six control points ***interpolated*** with piecewise continuous polynomial sections

Fig. A set of six control points ***approximated*** with piecewise continuous polynomial sections

A spline curve is defined, modified, and manipulated with operations on the control points. By interactively selecting spatial positions for the control points, a designer can set up an initial curve. After the polynomial fit is displayed for a given set of control points, the designer can then reposition some or all of the control points to restructure the shape of the curve. In addition, the curve can be translated, rotated, or scaled with transformations applied to the control points. CAD packages can also insert extra control points to aid a designer in adjusting the curve shapes.

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.

Spline Specifications

- There are three equivalent methods for specifying a particular spline representation.
  - **Boundary Conditions**
  - **Characterizing Matrix**
  - **Blending Functions or Basis Functions**

Boundary Conditions

- We can state the set of boundary conditions that are imposed on the spline.
- Suppose we have parametric cubic polynomial representation specified by the following set of equations:

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

Boundary Conditions

- Boundary conditions for this curve might be set, for example, on the endpoint coordinates $x(0)$ and $x(1)$ and on the parametric first derivatives at the endpoints $x'(0)$ and $x'(1)$. These four boundary conditions are sufficient to determine the values of the four coefficients a_x, b_x, c_x, and d_x.
- Similar approach can be used to determine values for y and z coordinate.

Characterizing Matrix

- We can state the matrix that characterizes the spline.
- From the boundary condition, we can obtain the characterizing matrix for spline. Then the matrix representation for the x-coordinate can be written as:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} = T.C$$

Where T is the row matrix of powers of parameter t, and C is the coefficient column matrix.

- Similar approach for y and z coordinate.

Blending Functions or Basis Functions

- We can state the set of blending functions (or basis functions) that determine how specified geometric constraints (boundary conditions) on the curve are combined to calculate positions along the curve path.
- Polynomial representation for coordinate x in terms of the geometric constraint parameters:

$$x(t) = \sum_{0}^{3} x_i . B_i(t)$$

Where x_i is the x-coordinate of the control point and $B_i(t)$ is i^{th} blending function which can be obtained by using Lagrange interpolation method.

$$y(t) = \sum_{0}^{3} y_i . B_i(t)$$

$$z(t) = \sum_{0}^{3} z_i . B_i(t)$$

Cubic spline interpolation

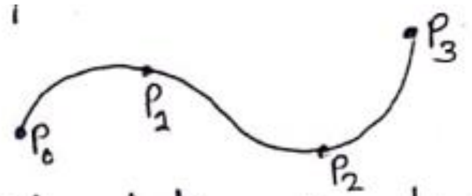


fig: Interpolation with cubic splines between 4 control points.

- Suppose we have $n + 1$ control points specified with coordinates

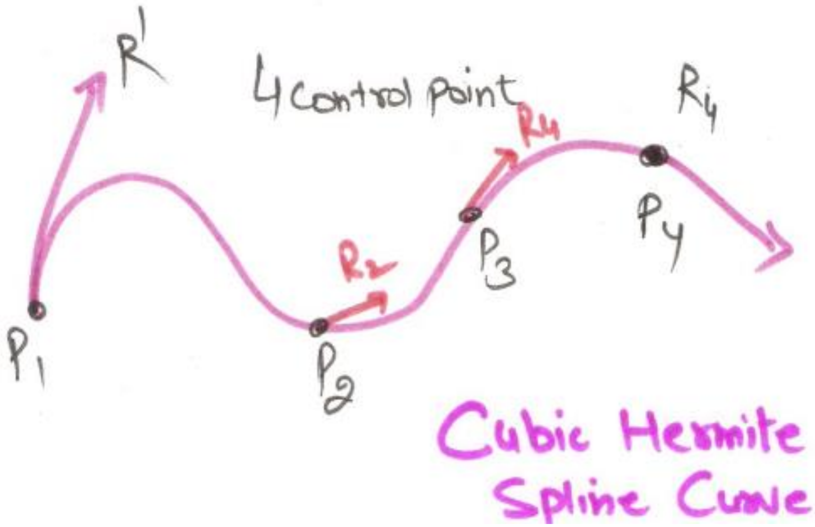
$$P_k = (x_k, y_k, z_k), k = 0,1,2, \dots n$$

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

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

Hermite Curves

- Hermite curves are very easy to calculate but also very powerful. They are used to smoothly interpolate between key-points.
- It is an interpolation spline curve.
- The Hermite form of the cubic polynomial curve segment is determined by constraints on the end points P_1 & P_2 and the tangent vectors at the end points R_1 & R_2.
  - P_1: the start point of the curve
  - R_1: the tangent to how the curve leaves the start point
  - P_2: the end point of the curve
  - R_2: the tangent to how the curve meets the end point



Let $Q(t)$ is the Curve where $t \in [0,1]$

$Q(t) = (x(t)\ \ y(t)\ \ z(t))$, $t \in [0,1)$ where all points Satisfy Cubic parametricity.

As we know the general Curve Equation

$P(t) = at^3 + bt^2 + ct + d \quad 0 \leq t \leq 1$ t is parameter

So $x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \cdot 1$

$y(t) = b_y t^3 + b_y t^2 + c_y t + d_y \cdot 1$

$z(t) = a_z t^3 + b_z t^2 \quad c_z t + d_z \cdot 1$

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \underbrace{\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}}_{T} \cdot \underbrace{\begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}}_{C}$$

$$Q(t) = T \cdot C$$

but As per Rule for Specifying a spline curve we need a basis f^n matrix So

$$Q(t) = T \cdot \underbrace{M}_{\text{basis Matrix}} \cdot \underbrace{G}_{\text{Geometry Vector}} \quad \text{where } [C = M \cdot G]$$

Let for Hermite we may write it as

$$Q(t) = T \cdot M_H \cdot G_H.$$

M_H = Hermite basis matrix
G_H = Hermite Geometry Matrix Vector

$$= [t^3\ t^2\ t\ 1] \cdot M_H \cdot G_H$$

$$Q_x(t)\Big|_{t=0} = P_1(\cancel{t}) = [0\ 0\ 0\ 1] \cdot M_H \cdot G_{H_x} \quad - \text{(A)}$$

$$Q_x(t)\Big|_{t=1} = P_4(t) = [1\ 1\ 1\ 1] \cdot M_H \cdot G_{H_x} \quad - \text{(B)}$$

$$Q'_{(x)}t\Big|_{t=0} = R_1(t)\,[0\ 0\ 1\ 0] \cdot M_H \cdot G_{H_x} \quad - \text{(C)}$$

$$Q'_{(x)}t\Big|_{t=1} = R_4(t)\,[3\ 2\ 1\ 0]\, M_H . G_{H_x} \quad - \text{(D)}$$

From A, B, C, D Equations

$$\begin{bmatrix} P_1(x) \\ P_4(x) \\ R_1(x) \\ R_4(x) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} M_H \cdot G_{Hx}}_{C}$$

$$M_H \cdot G_{Hx} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{bmatrix}$$

So

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ \& }$$

$$G_{Hx} = \begin{bmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{bmatrix}$$

$Q(t) = T \cdot M_H \cdot G_H$

$$\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}$$

$$Q(t) = (2t^3 - 3t^2 + 1) P_1 + (-2t^3 + 3t^2) P_4 + (t^3 - 2t^2 + t) R_1 + (t^3 - t^2) R_4$$

$$= P_1 H_0(t) + P_4 H_1(t) + R_1 H_2(t) + R_4 H_3(t)$$

$H_0(t)$, $H_1(t)$, $H_2(t)$, $H_3(t)$ Hermite blending fn.

Bezier Curves and Surfaces

- This is spline approximation method, developed by the Engineer Pierre Bezier for use in the design of automobile body.
- Beziers spline has several properties that make them highly useful and convenient for curve and surface design. They are easy to implement. For this reason, Bezier spline is widely available in various CAD systems.

Bezier Curves

- A Bezier curve is an approximated parametric curve frequently used in computer graphics and related fields.
- A Bezier curve can be fitted to any number of control points.
- The Bezier curve can be specified with boundary condition, with characterizing matrix or blending functions. But for general Bezier curves, blending function specification is most convenient.
- A set of characteristic polynomial approximating functions, called Bezier blending functions are used to blend the control points to produce a Bezier curve segment.
- The degree of a Bezier curve segment is determined by the number of control points to be fitted with that curve segment.

Mathematical Definition

- Given a set of $(n + 1)$ control points, $P_0(x_0, y_0, z_0), P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), \dots, P_n(x_n, y_n, z_n)$, a parametric Bezier curve (Bernstein-Bezier curve) segment can be defined by

$$C(u) = \sum_{i=0}^{n} P_i \,.BEZ_{i,n}\,(u) \qquad \dots(1)$$

Here, u is a parameter such that $0 \le u \le 1$.

$BEZ_{i,n}\,(u)$ is a Bezier Blending function or Bernstein Polynomial and is defined by:

$$BEZ_{i,n}(u) = C(n,i)u^i\,(1-u)^{n-i} \qquad \dots(2)$$

where $C(n,i)$ are the binomial coefficients given by:

$$C(n,i) = \frac{n!}{(n-i)!\,i!} \qquad \dots(3)$$

Mathematical Definition

$$Cubic\ Bezier, n = 3$$

$$C(u) = P_0.BEZ_{0,3(u)} + P_1.BEZ_{1,3}(u) + P_2.BEZ_{2,3}(u) + P_3.BEZ_{3,3}(u)$$

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

$$= \frac{3!}{3!\,0!}(1-u)^3 = \boldsymbol{(1-u)^3}$$

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

$$= \frac{3!}{2!\,1!}.u.(1-u)^2 = \boldsymbol{3u(1-u)^2}$$

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

$$= \frac{3!}{1!\,2!}u^2(1-u) = \boldsymbol{3u^2(1-u)}$$

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

$$= \frac{3!}{0!\,3!}.u^3 = \boldsymbol{u^3}$$

Mathematical Definition

$$C(u) = P_0.(1-u)^3 + P_1.3u.(1-u)^2 + P_2.3u^2.(1-u) + P_3.u^3$$

$$C(u_x) = P_{x0}.(1-u)^3 + P_{x1}.3u.(1-u)^2 + P_{x2}.3u^2.(1-u) + P_{x3}.u^3$$

$$C(u_y) = P_{y0}.(1-u)^3 + P_{y1}.3u.(1-u)^2 + P_{y2}.3u^2.(1-u) + P_{y3}.u^3$$

$$C(u_z) = P_{z0}.(1-u)^3 + P_{z1}.3u.(1-u)^2 + P_{z2}.3u^2.(1-u) + P_{z3}.u^3$$

Mathematical Definition

Equation (1) represents a set of three parametric equations, so for individual curve coordinates:

$$x(u) = \sum_{i=0}^{n} x_i \,.\, BEZ_{i,n}(u)$$

$$y(u) = \sum_{i=0}^{n} y_i \,.\, BEZ_{i,n}(u)$$

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

As a rule, a Bezier curve is a polynomial of degree one less than the number of control points used: Three points generate a parabola; four points a cubic curve, and so forth.

Bezier Curve

- Bezier curves are widely used in computer graphics to model smooth curves. As the curve is completely contained in the convex hull of its control points, the points can be graphically displayed and used to manipulate the curve.

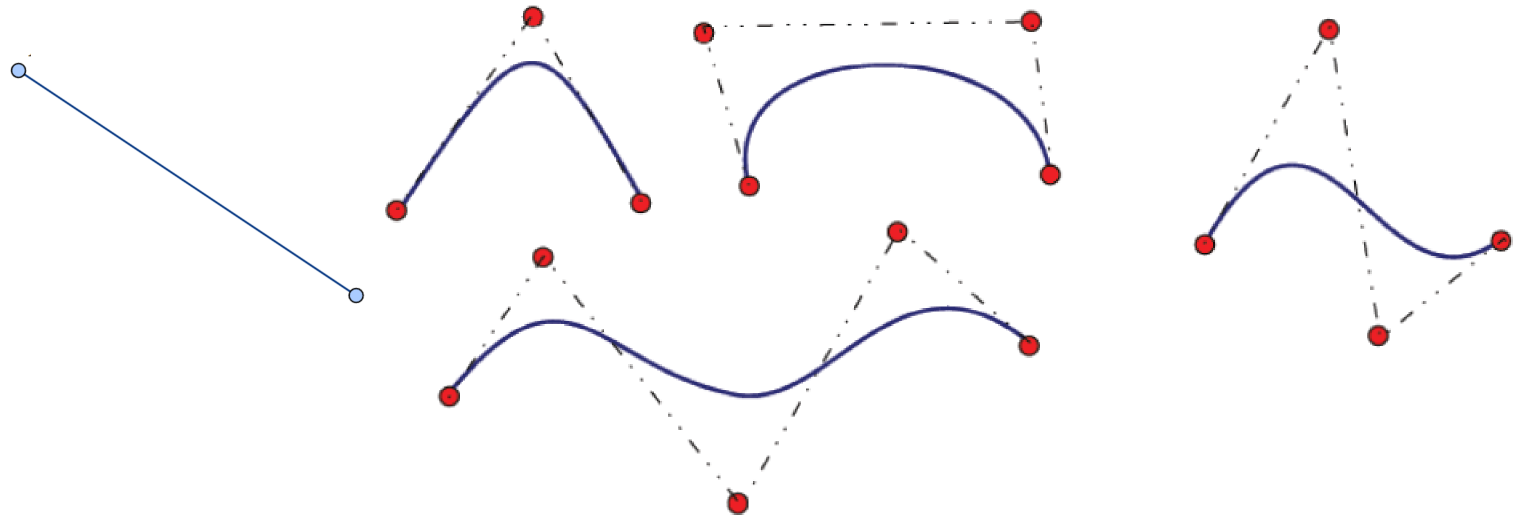
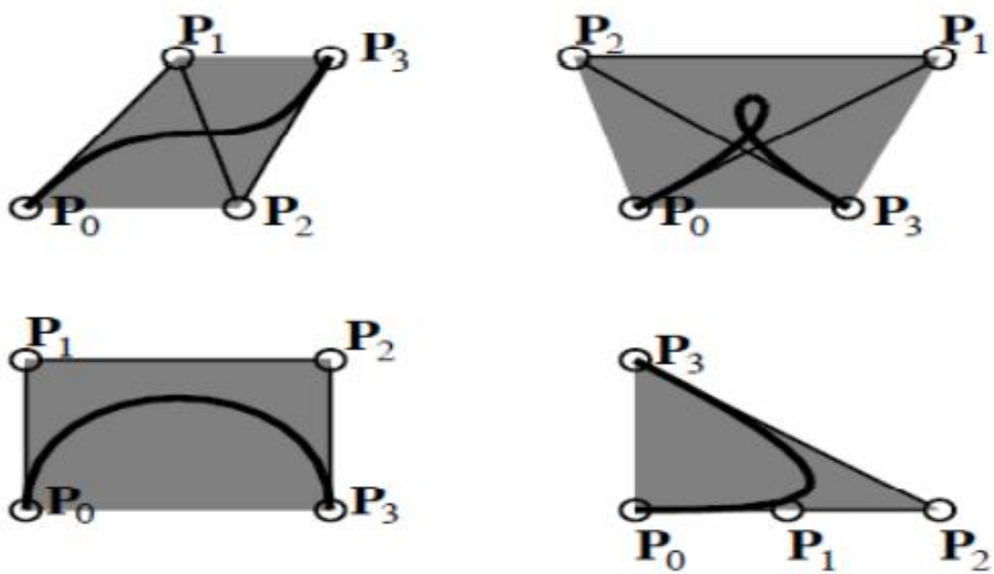


Fig: Examples of two-dimensional Bezier curves generated from two, three, four, and five control points. Dashed lines connect the control-point positions.

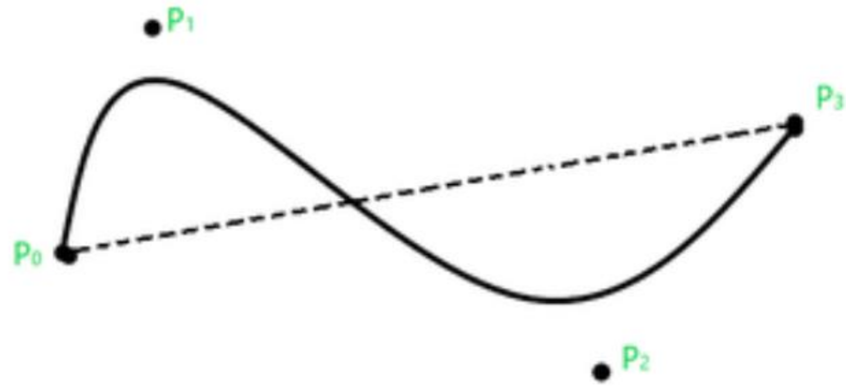
Properties of Bezier Curves

- The Bezier curve always passes through the first and last control points ,i.e., is $C(u = 0) = P_0$ and $C(u = 1) = P_n$, since $u = 0$ (for first point) and $u = 1$ (for last control point).
- The degree of polynomial defining the curve segment is one less than the number of defining polygon points. Therefore, for 4 control points the degree of polynomial is three, for three control points the degree of polynomial is 2 and so on.
- The curve generally follows the shape of the defining polygon.
- The curve is always contained within the convex hull (The convex polygon boundary that encloses a set of control points is called the convex hull) of the control points.

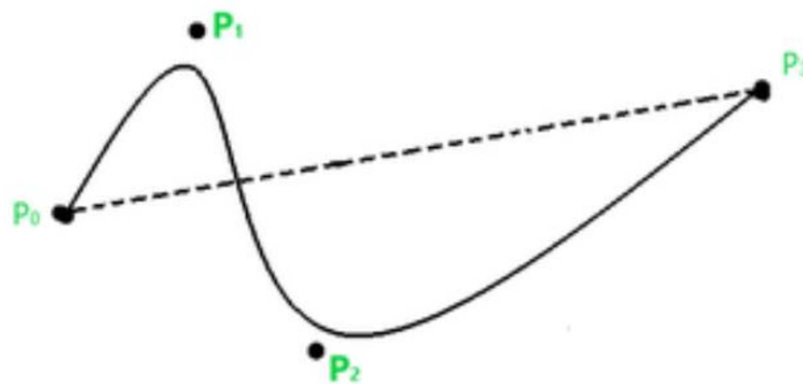
Properties of Bezier Curves



Bezier Curves



Control points { P0, P1, P2, P3}



Control points { P0, P1, P2, P3}

Bezier Surface

- A Bezier Surface is formed as the Cartesian product of the blending functions of two orthogonal Bezier curves. Two sets of orthogonal Bezier curves can be used to design an object surface by specifying an input mesh of control points. That is,

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

where, $P(u,v)$ is any point on the surface and $P_{i,j}$ specify the location of the $(m + 1)$ by $(n + 1)$ control points.

- The control points are connected by dashed lines, and the solid lines show curves of constant u and constant v. Each curve of constant u is plotted by varying v over the interval from 0 to 1, with u fixed at one of the values in this unit interval. Curves of constant v are plotted similarly.

Bezier Surface

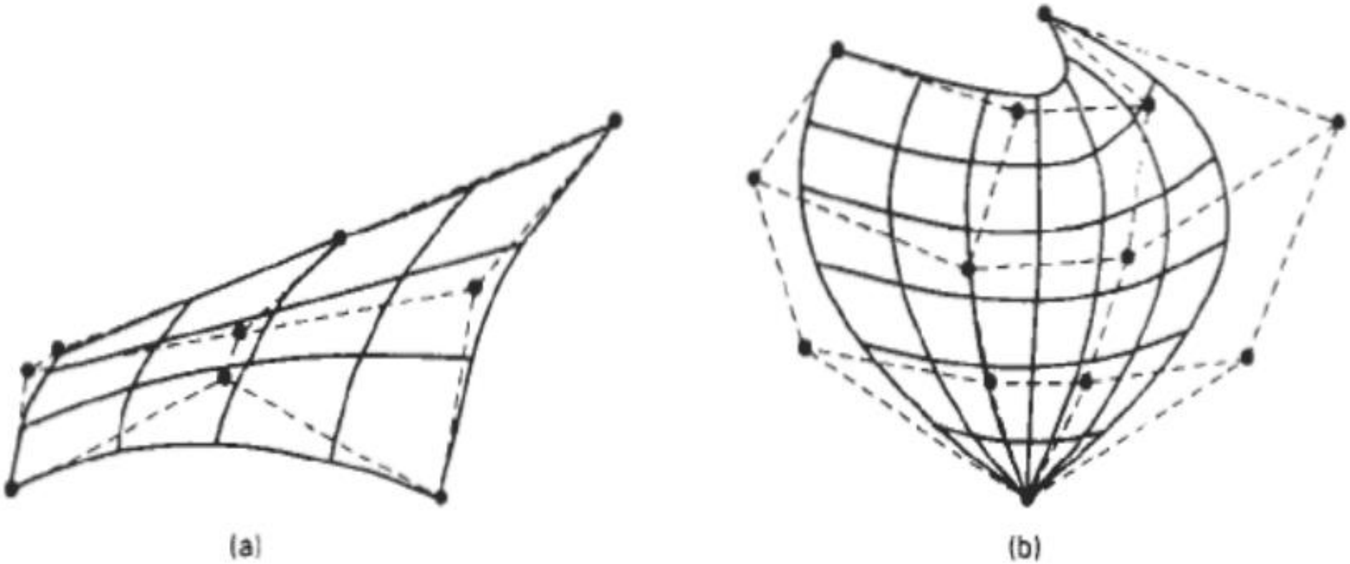
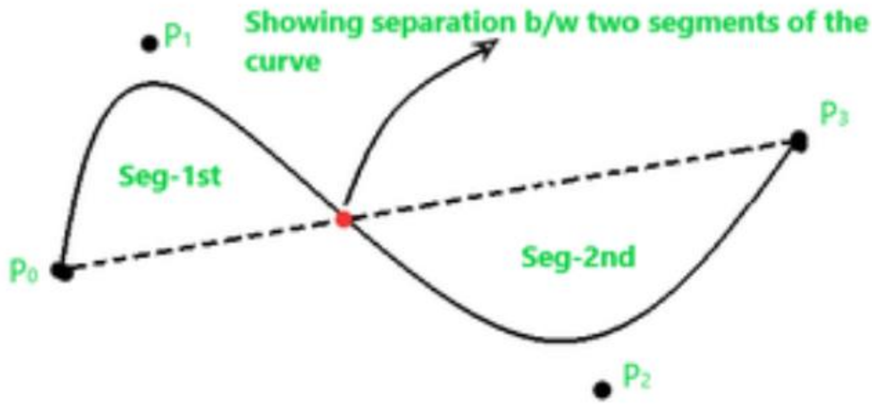


Fig: Bezier surfaces constructed tor (a) m = 3, n = 3, and (b) m = 4, n = 4. Dashed lines connect the control points.

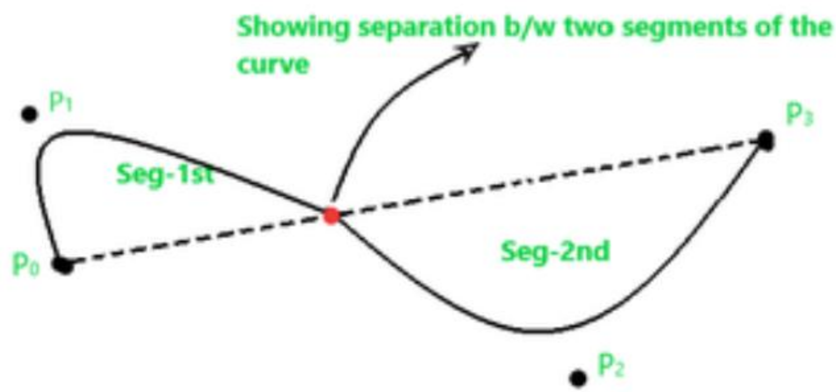
B-spline Curve and surface

- These are the most widely used class of approximating splines. B-spline have following two advantages over Bezier splines:
  - The degree of a B-spline can be set independently of the number of control points.
  - B-splines allow local control over the shape of a spline curve or surface[only a specific segment of the curve-shape gets changes or affected by the changing of the corresponding location of the control points].
- The disadvantage is that B-splines are more complex than Bezier splines.

B-spline Curve



Control points { P0, P1, P2, P3}



Control points { P0, P1, P2, P3}

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:

$$Q(u) = \sum_{i=0}^{n} P_i N_{i,k}(u),$$

$$0 \leq u \leq n - k + 2$$

$$2 \leq k \leq n + 1$$

where P_i are the control points of the $n + 1$ defining polygon vertices and the $N_{i,k}$ are the normalized B-spline basis functions.

B-spline Curves

- For the i^{th} normalized B-spline basis function of order k, the basis function $N_{i,k}\ (u)$ are defined as

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

$$t_i(0 \le i \le n + k) \ \rightarrow knot\ values$$

$$\boldsymbol{t_i = 0\ if\ i < k}$$

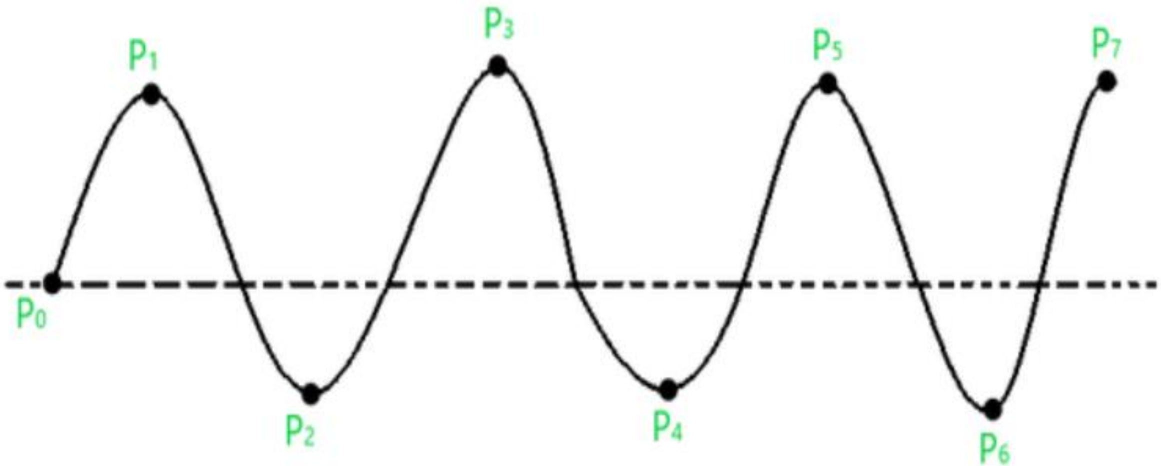
$$\boldsymbol{t_i = i - k + 1\ if\ k \le i \le n}$$

$$\boldsymbol{t_i = n - k + 2\ if\ i > n}$$

$$N_{i,k}(u) = \begin{cases} 1 & if\ t_i \le u \le t_{i+1} \\ 0 & otherwise \end{cases}$$

The values of t_i are the elements of a knot vector satisfying the relation $t_i \le t_{i+1}$. The parameter u varies from $0 \le u \le n - k + 2$ 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.

B-spline Curve



- We have “n+1” control points in the above, so, n+1=8, so n=7.
- Let’s assume that the order of this curve is ‘k’, so the curve that we get will be of a polynomial degree of “k-1”. Conventionally it’s said that the value of ‘k’ must be in the range: $2 \leq k \leq n+1$. So, let us assume $k = 4$, so the curve degree will be $k - 1 = 3$
- The total number of segments for this curve will be calculated through the following formula:
  Total no. of seg = $n - k + 2 = 7 - 4 + 2 = 5$

B-spline Curve

Segments	Control points	Parameter
S_0	P_0,P_1,P_2,P_3	$0 \leq u < 1$
S_1	P_1,P_2,P_3,P_4	$1 \leq u < 2$
S_2	P_2,P_3,P_4,P_5	$2 \leq u < 3$
S_3	P_3,P_4,P_5,P_6	$3 \leq u < 4$
S_4	P_4,P_5,P_6,P_7	$4 \leq u < 5$

On the basis of the knot points and interval length of segment there are two types of spline;

- **Periodic B-spline:** Knot points are equi-space to each other and splines are generated through the set of the equi-interval segments then such splines are called periodic B-splines.
- **Aperiodic B-spline:** If knot points are not equi-space to each other and splines are not generated through the set of the equi interval segments then such splines are called aperiodic B-splines.

Knot vector

There are three general classifications for knot vectors: uniform, open uniform and non-uniform.

➢ ***Uniform, periodic B-splines:***

When the spacing between knot values is constant, the resulting curve is a uniform B-spline. For e.g. {0, 1, 2, 3, 4, 5}

- Uniform B-splines have periodic blending function. That is, for given value of 'n' and 'd', all blending functions have the same shape
- Periodic splines are particularly useful for generating certain closed curves.

- ***Open uniform B-splines:***

For open B-splines, the knot spacing is uniform expect at the ends where knot values are repeated 'd' times. For e.g.

$\{0, 0, 1, 2, 3, 3\}$ for d=2, and n=3

- ***Non-uniform B-splines:***

For non-uniform B-splines, we can choose multiple internal knot values and unequal spacing between the knot values. For e.g.

$\{0, 1, 2, 3, 3, 4\}$

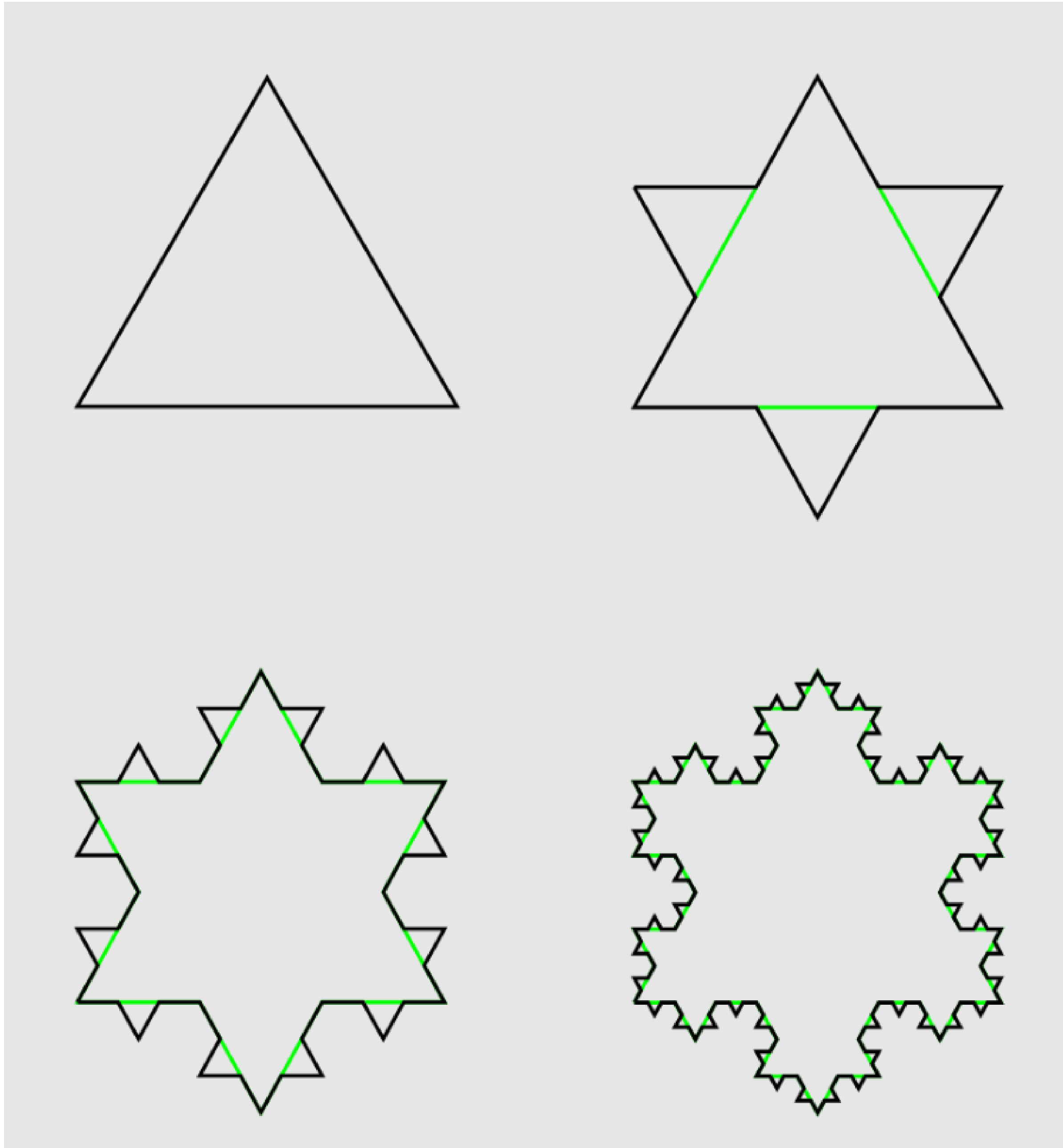
$\{0, 0.2, 0.6, 0.9, 1.0\}$

- Non-uniform B-splines provide increased flexibility in controlling a curve shape.

Fractals and its applications

- A curve or geometrical object each part of which has the same statistical character.
- Those objects which are 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 looks at them, they always appear somewhat similar.
- Fractal objects can also be generated by recursively by applying the same transformation function to an object.

  E.g.: scale down + rotate + translate



Applications

- Fractals are used to capture images of complex structures such as clouds, terrain, mountain
- To compress images
- 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.

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:

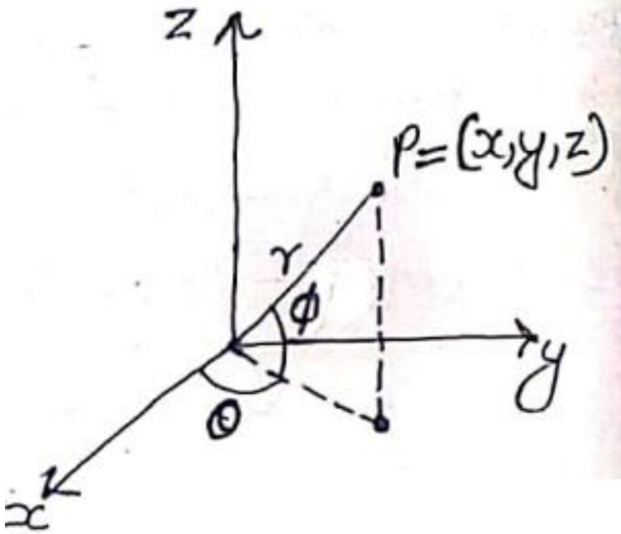
$$x^2 + y^2 + z^2 = r^2$$

- The spherical surface can be represented in parametric form by using latitude and longitude angles as:

$$x = rcos\phi cos\theta, -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$$
$$y = rcos\phi sin\theta, -\pi \leq \phi \leq \pi$$
$$z = rsin\phi$$

Sphere

- 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 centered on the origin is:

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

Ellipsoid

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

$$x = r_x cos\phi cos\theta, -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$$
$$y = r_y cos\phi sin\theta, -\pi \leq \phi \leq \pi$$
$$z = r_z sin\phi$$

Ellipsoid