RCS-603: COMPUTER GRAPHICS
UNIT-IV

Presented By :

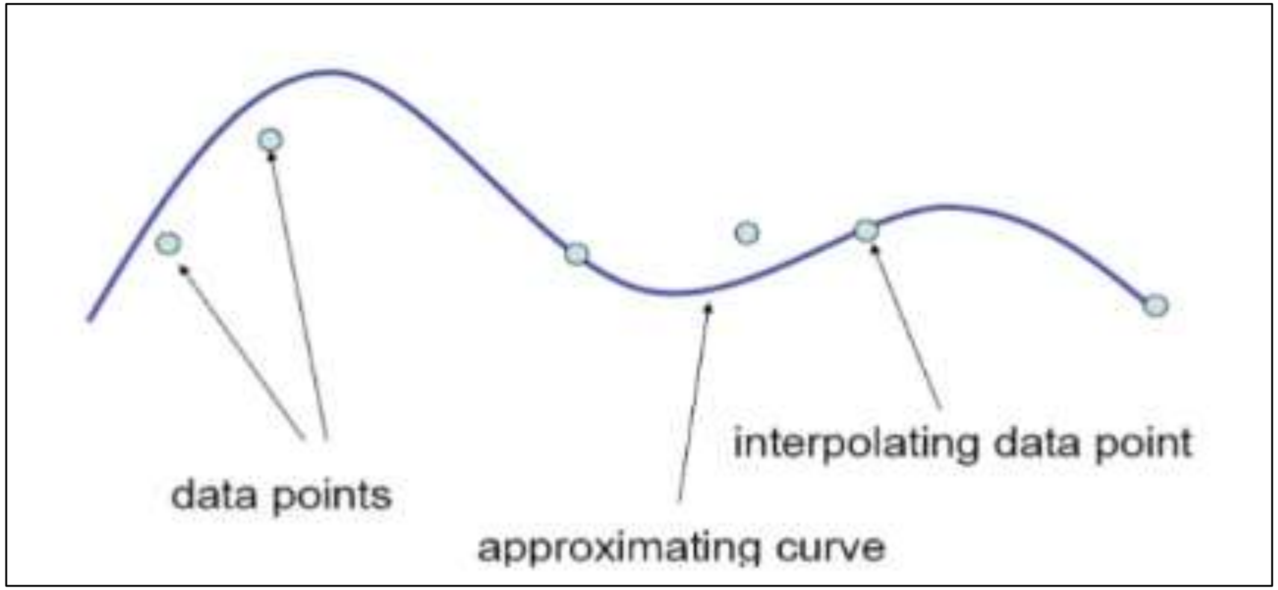
Dr. Vinod Jain (Associate Professor, GLBITM)

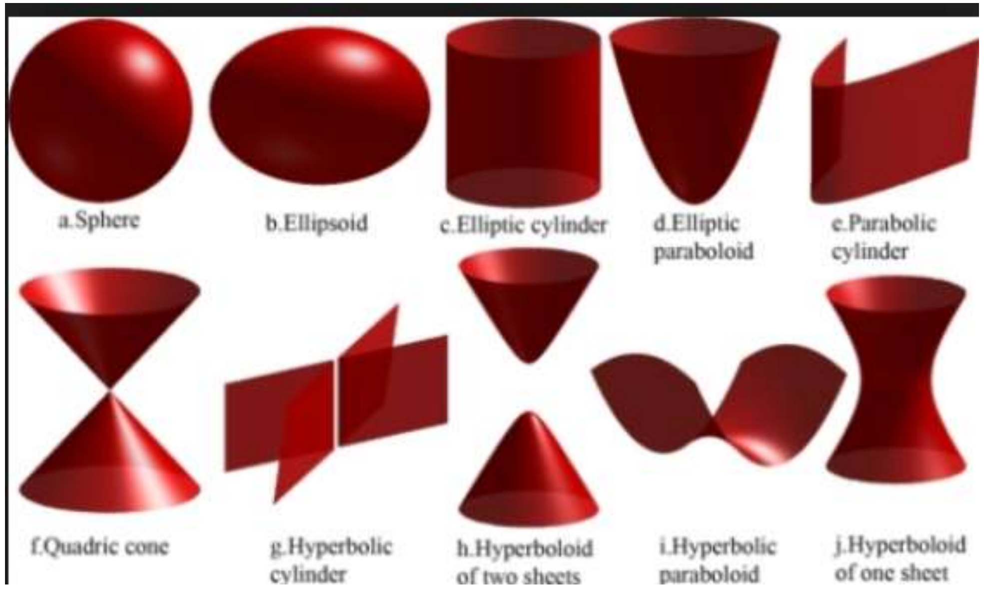
GL BAJAJ
Institute of Technology & Management
[Approved by AICTE, Govt. of India & Affiliated to Dr. APJ Abdul Kalam Technical University, Lucknow, U.P. India]
Department Of Computer Science & Engineering

Unit- IV - Curves and Surfaces:

1. Quadric surfaces
2. Spheres
3. Ellipsoid
4. Blobby objects
5. Introductory concepts of Spline
6. Bspline and Bezier curves and surfaces.

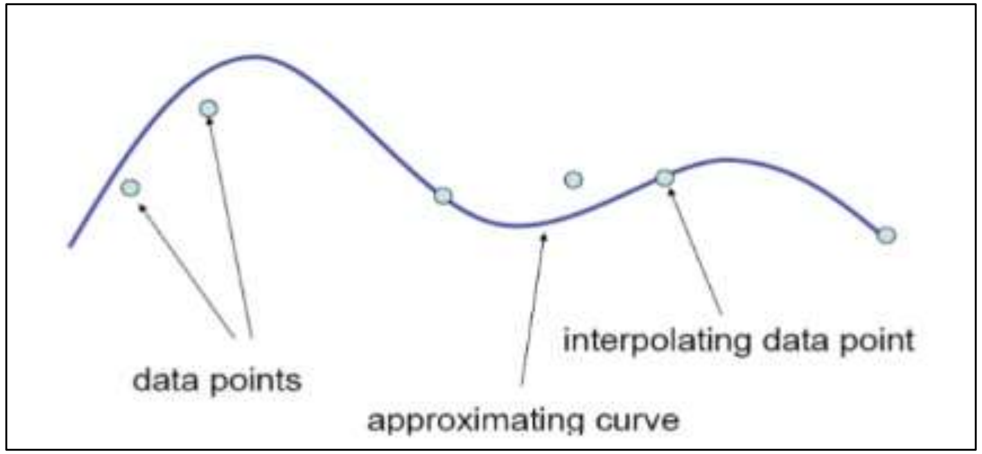
Curves and Surfaces





Curves and Surfaces

- Displays of three dimensional curved lines and surfaces can be **generated from an input set of mathematical functions** defining the objects or from a set of users specified data points.
- When functions are specified**, a package can project the defining equations for a curve to the display plane** and plot pixel positions along the path of the projected function.



Quadric surfaces

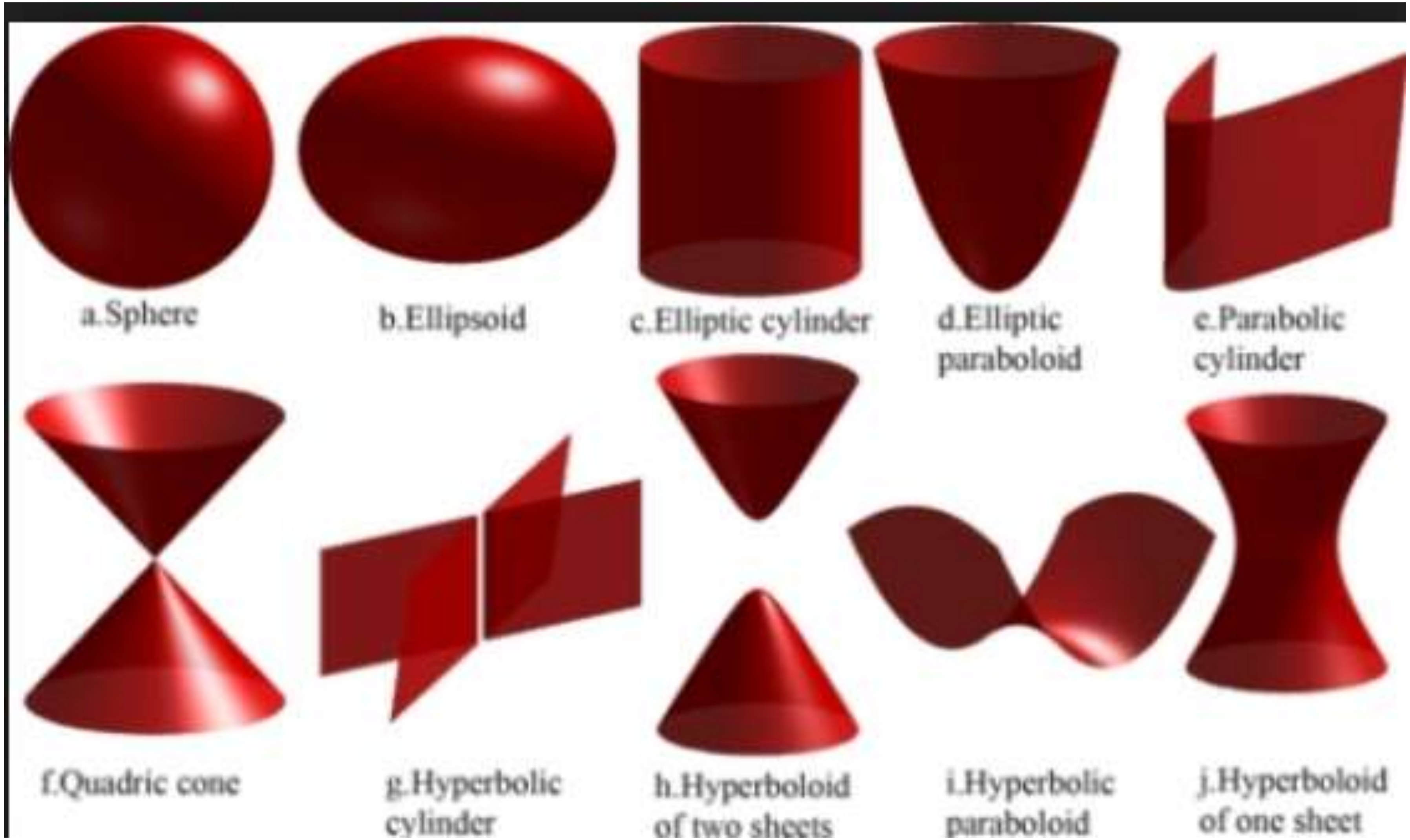
- A frequently used class of objects are the quadric surfaces, which are described with **second-degree equations (quadratics).**
- They include

1. Spheres,
2. Ellipsoids,
3. Paraboloids,
4. Hyperboloids etc.

Sphere

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

Quadric surfaces



Sphere

- In Cartesian coordinates, a spherical surface with radius r centered on the coordinate origin is defined as the set of points (x, y, **z)** that satisfy the equation

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

Sphere in parametric form

- We can also describe the spherical surface in parametric form, using **latitude and longitude angles.**

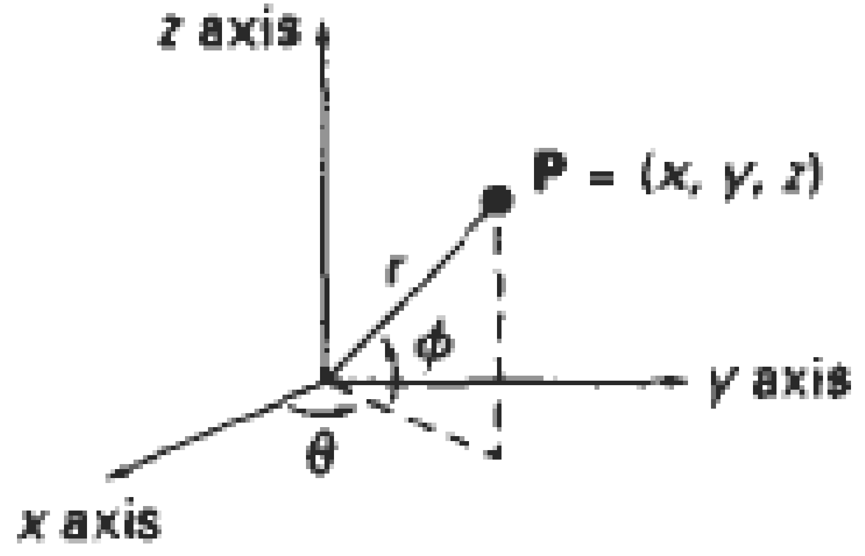
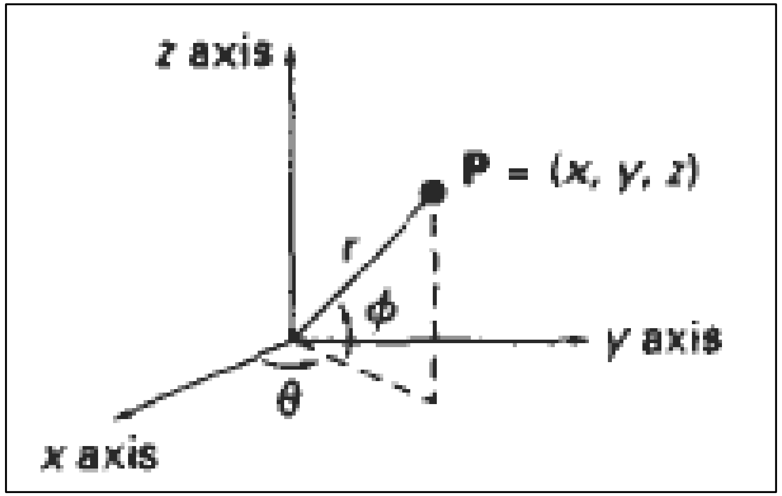


Figure 10-8
Parametric coordinate position (r, θ, ϕ) on the surface of a sphere with radius r.

Sphere in parametric form

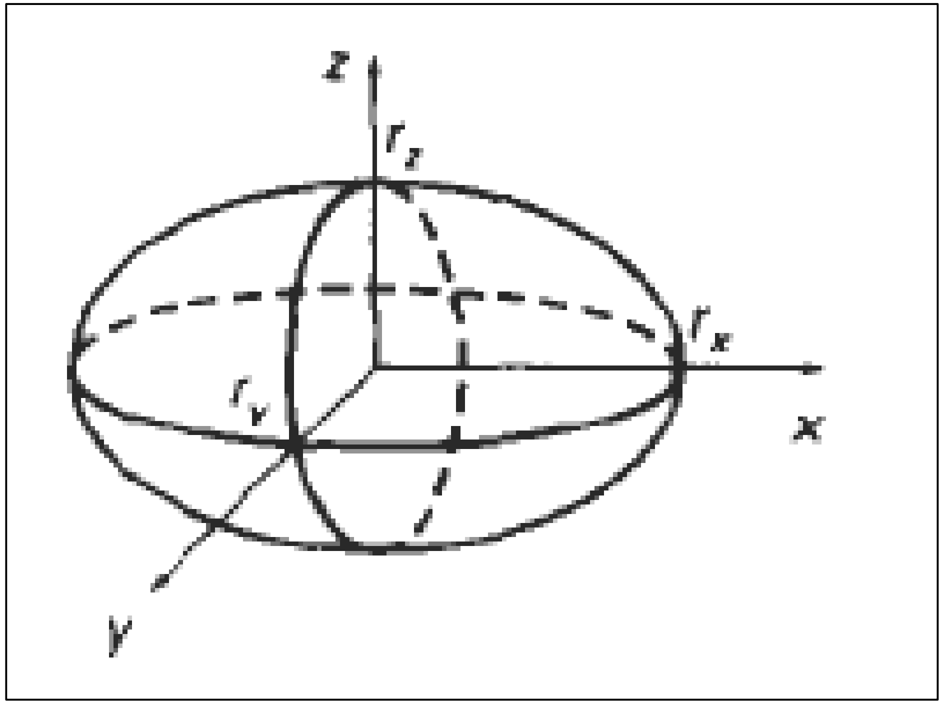
We can also describe the spherical surface in parametric form, using latitude and longitude angles (Fig. 10-8):

$$
\begin{aligned}
x &= r\cos\phi\cos\theta, & -\pi/2 \le \phi \le \pi/2 \\
y &= r\cos\phi\sin\theta, & -\pi \le \theta \le \pi \qquad (10\text{-}8) \\
z &= r\sin\phi
\end{aligned}
$$



Ellipsoid

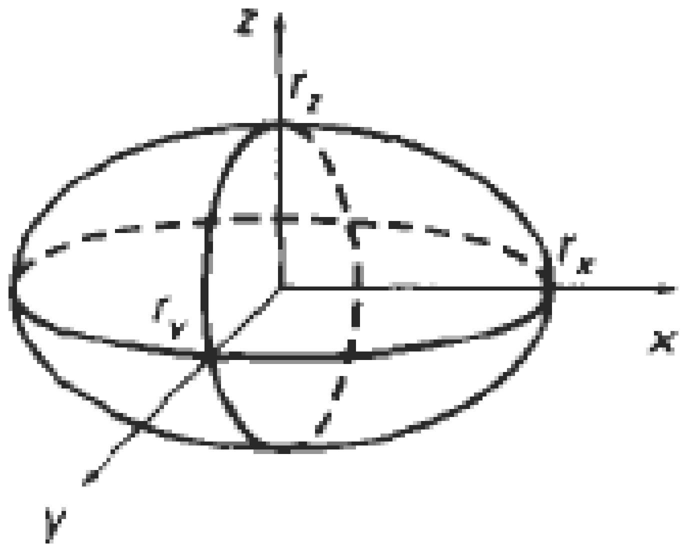
- An ellipsoidal surface can be described as an extension of a spherical surface, **where the radii in three mutually perpendicular directions can have different values.**



Ellipsoid

- The **Cartesian representation** for points over the surface of an ellipsoid centered on the origin is

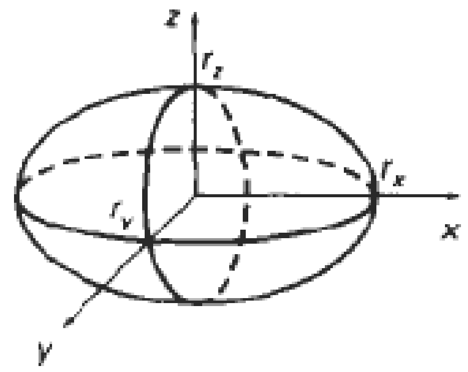
$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$



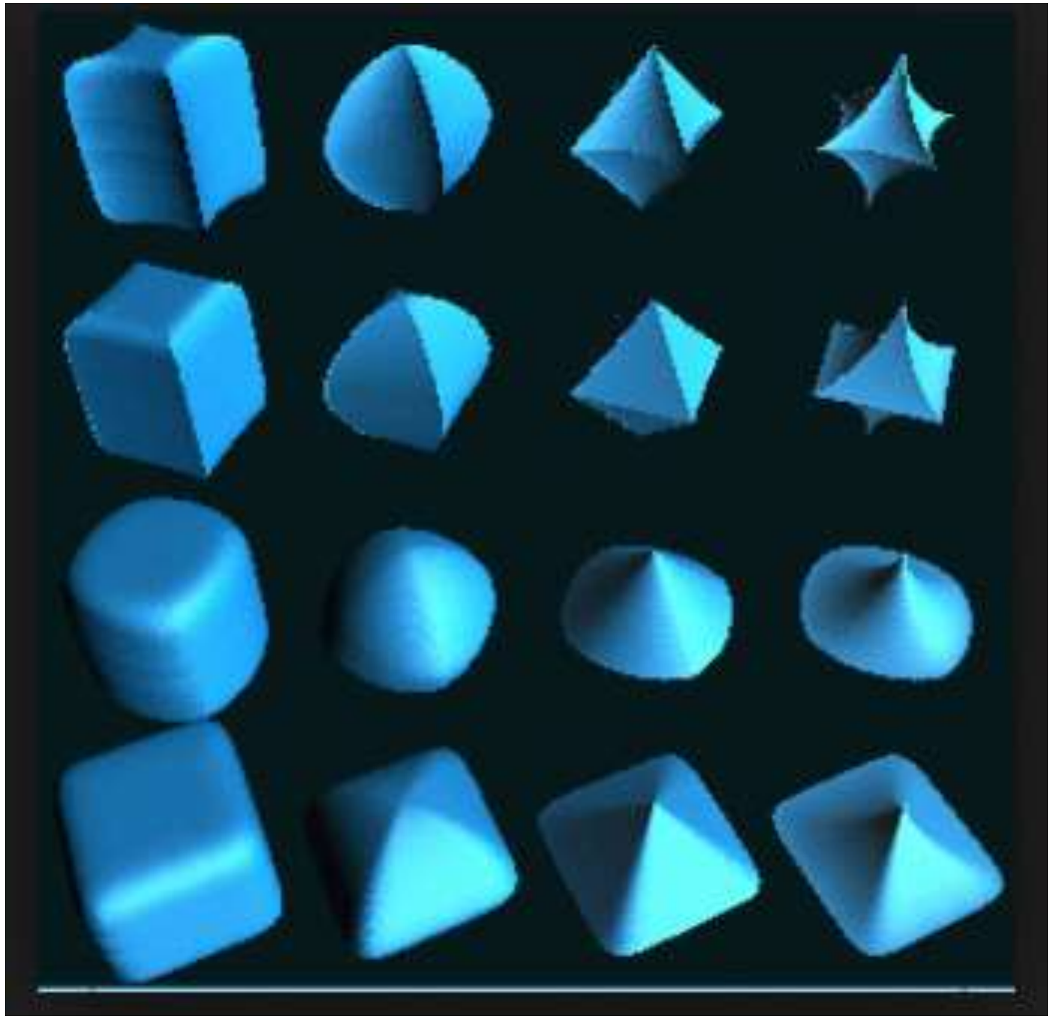
Ellipsoid - Parametric representation

And a parametric representation for the ellipsoid in terms of the latitude angle ϕ and the longitude angle θ in Fig. 10-8 is

$$\begin{aligned} x &= r_x \cos\phi \cos\theta, \qquad -\pi/2 \le \phi \le \pi/2 \\ y &= r_y \cos\phi \sin\theta, \qquad -\pi \le \theta \le \pi \\ z &= r_z \sin\phi \end{aligned} \tag{10-10}$$

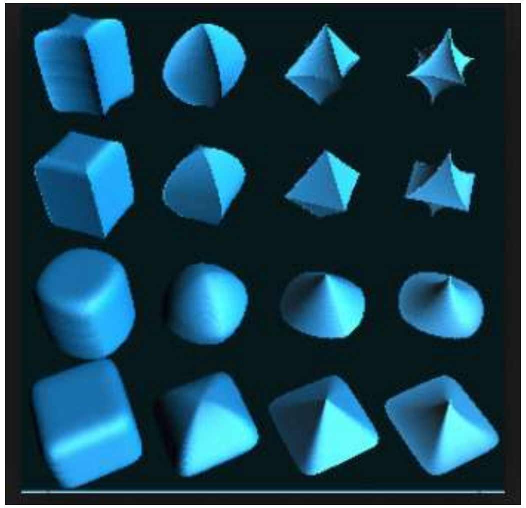


Superquadrics



Superquadrics

- Superquadrics are formed by incorporating **additional parameters** into the quadric equations to provide **increased flexibility** for adjusting object shapes.
- The number of additional parameters used is equal to the dimension of the object: **one parameter for curves and two parameters for surfaces.**



Superquadrics

1. Superellipse
2. Superellipsoid

Superellipse

Figure 10-12
Superellipses plotted with different values for parameter s and with $r_x = r_y$.

superellipse

- We obtain a Cartesian representation for a superellipse from the corresponding equation for an ellipse by **allowing the exponent on the *x* and y terms to be variable.**

$$\left(\frac{x}{r_x}\right)^{2/s} + \left(\frac{y}{r_y}\right)^{2/s} = 1 \qquad (10\text{-}13)$$

where parameter s can be assigned any real value. When $s = 1$, we get an ordinary ellipse.

Superellipse – In parametric form

$$\left(\frac{x}{r_x}\right)^{2/s} + \left(\frac{y}{r_y}\right)^{2/s} = 1 \qquad (10\text{-}13)$$

where parameter s can be assigned any real value. When $s = 1$, we get an ordinary ellipse.

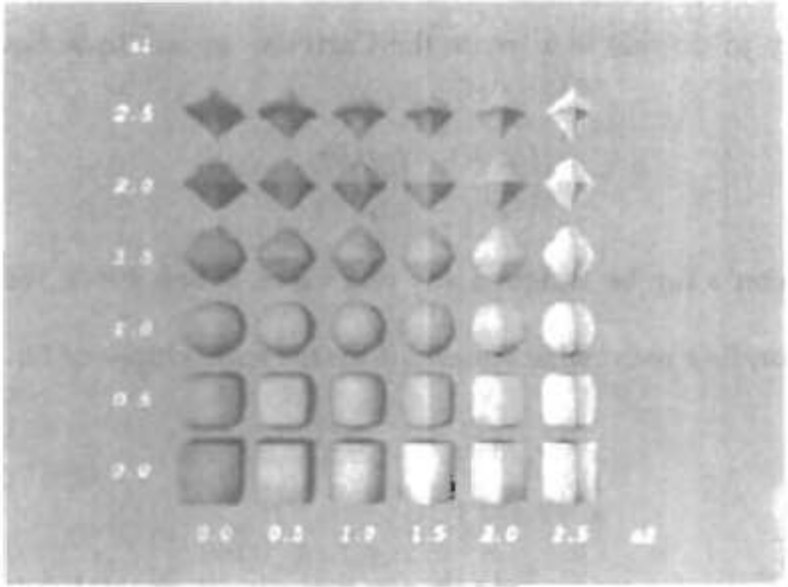
Corresponding parametric equations for the superellipse of Eq. 10-13 can be expressed as

$$\begin{aligned} x &= r_x \cos^s \theta, \qquad -\pi \le \theta \le \pi \\ y &= r_y \sin^s \theta \end{aligned} \qquad (10\text{-}14)$$

Superellipse

Figure 10-12
Superellipses plotted with different values for parameter s and with $r_x = r_y$.

Superellipsoids



Superellipsoids

A Cartesian representation for a superellipsoid is obtained from the equation for an ellipsoid by incorporating two exponent parameters:

$$\left[\left(\frac{x}{r_x}\right)^{2/s_2} + \left(\frac{y}{r_y}\right)^{2/s_2}\right]^{s_2/s_1} + \left(\frac{z}{r_z}\right)^{2/s_1} = 1 \qquad (10\text{-}15)$$

For $s_1 = s_2 = 1$, we have an ordinary ellipsoid.

We can then write the corresponding parametric representation for the superellipsoid of Eq. 10-15 as

$$\begin{aligned} x &= r_x \cos^{s_1}\phi \cos^{s_2}\theta, & -\pi/2 \le \phi \le \pi/2 \\ y &= r_y \cos^{s_1}\phi \sin^{s_2}\theta, & -\pi \le \theta \le \pi \qquad (10\text{-}16) \\ z &= r_z \sin^{s_1}\phi \end{aligned}$$

Superellipsoids

- Figure 10-13 illustrates supersphere shapes that can be generated using various values for parameters **s,** and **s2.**
- **These** and other superquadric shapes can be combined to create more complex **structures,** such as furniture, threaded bolts, and other hardware

Superellipsoids

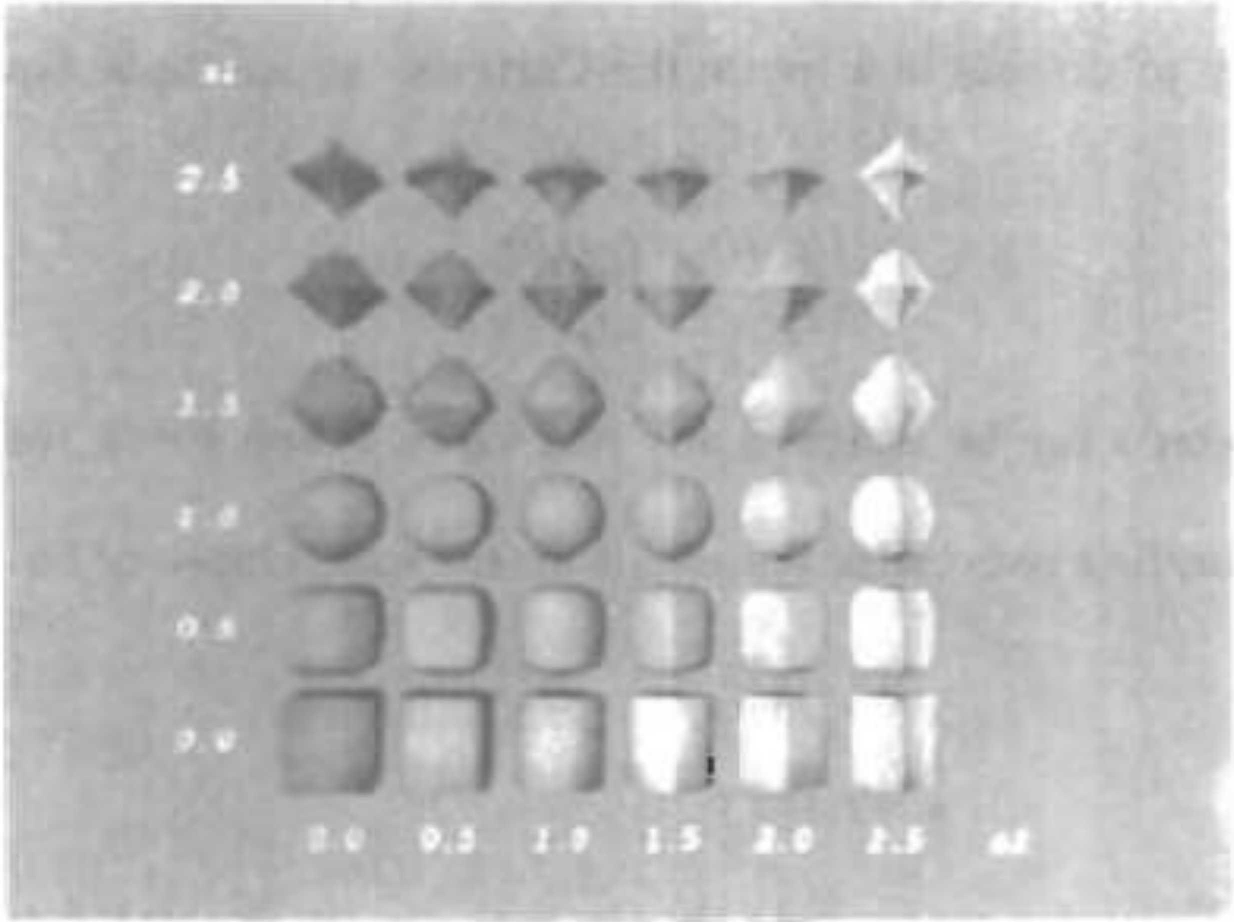


Figure 10-13
Superellipsoids plotted with different values for parameters s_1 and s_2, and with $r_x = r_y = r_z$.

Blobby objects

Figure 10-14
Molecular bonding. As two molecules move away from each other, the surface shapes stretch, snap, and finally contract into spheres.

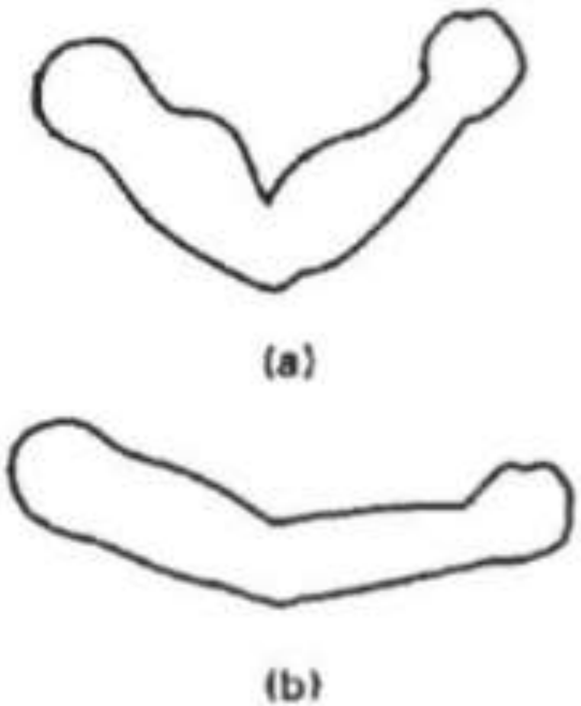
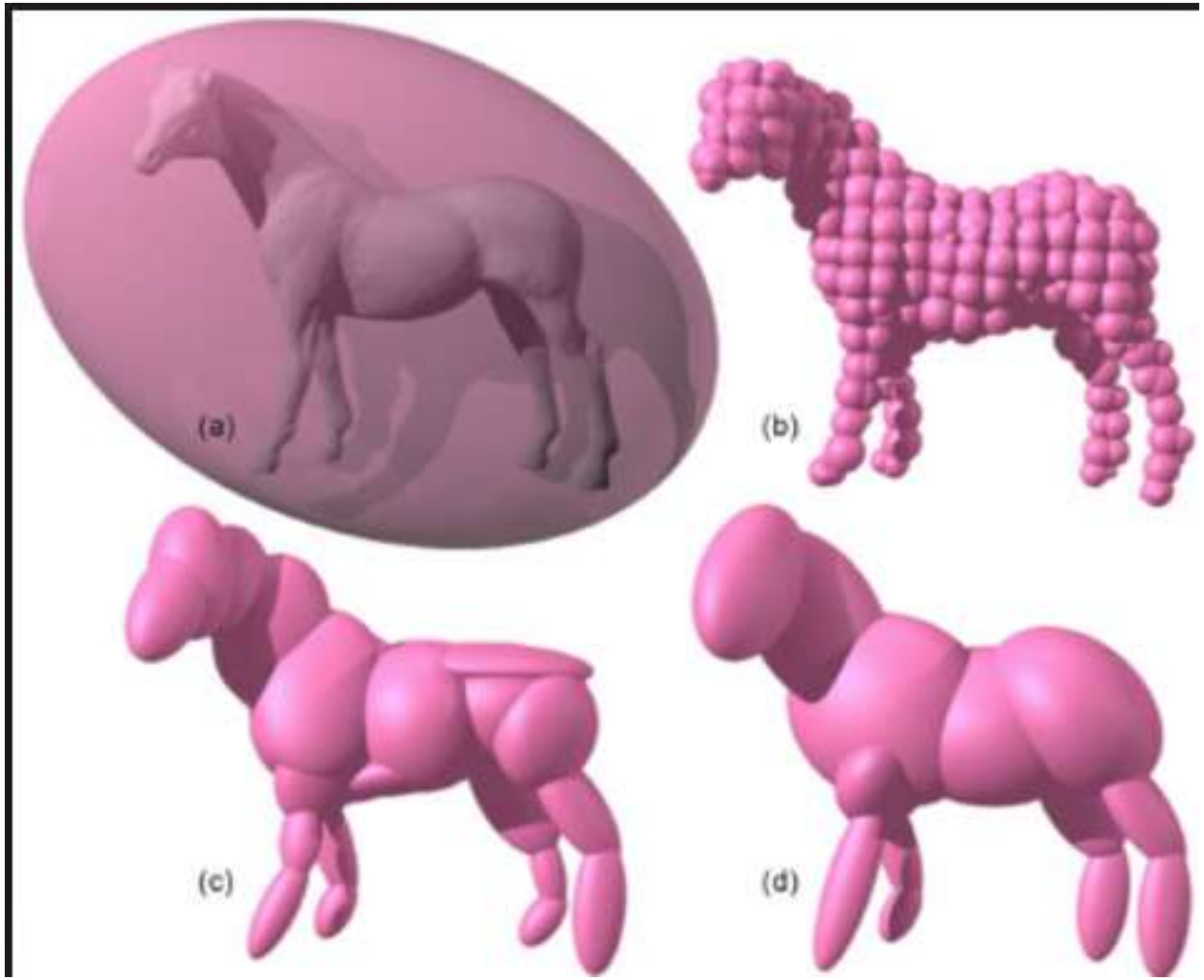


Figure 10-15
Blobby muscle shapes in a human arm.

Blobby objects



Blobby objects

- Some objects **do not maintain a fixed shape**
- They change their surface characteristics in certain motions
- These objects are referred to as blobby objects, **since their shapes show a certain degree of fluidity**
- Examples in this class of objects include

1. water droplets
2. melting objects
3. muscle shapes in the human body.

Blobby objects

- Several models have been developed for representing blobby objects as distribution functions over a region of space.
- Combinations of Gaussian density functions, or "bumps“ (Fig 10.16)

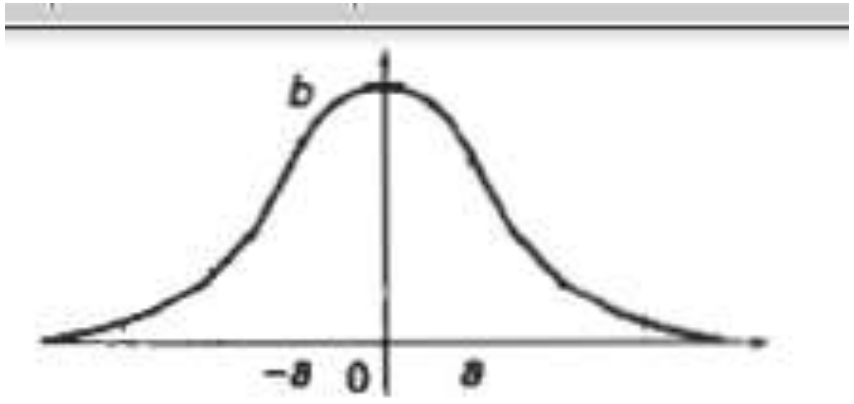


Figure 10-16
A three-dimensional Gaussian bump centered at position 0, with height b and standard deviation a.

Blobby objects

- Several models have been developed for representing blobby objects as distribution functions over a region of space.
- Combinations of Gaussian density functions, or "bumps“ (Fig 10.16)

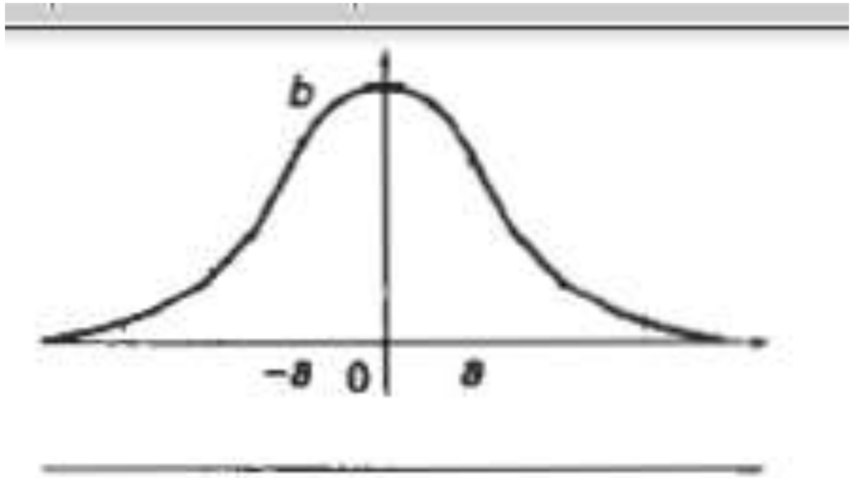


Figure 10-16
A three-dimensional Gaussian bump centered at position 0, with height *b* and standard deviation *a*.

Blobby objects

- A surface function is then defined as

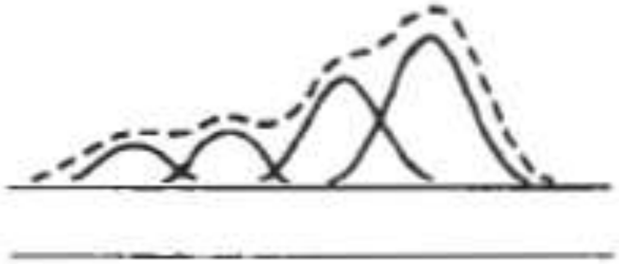
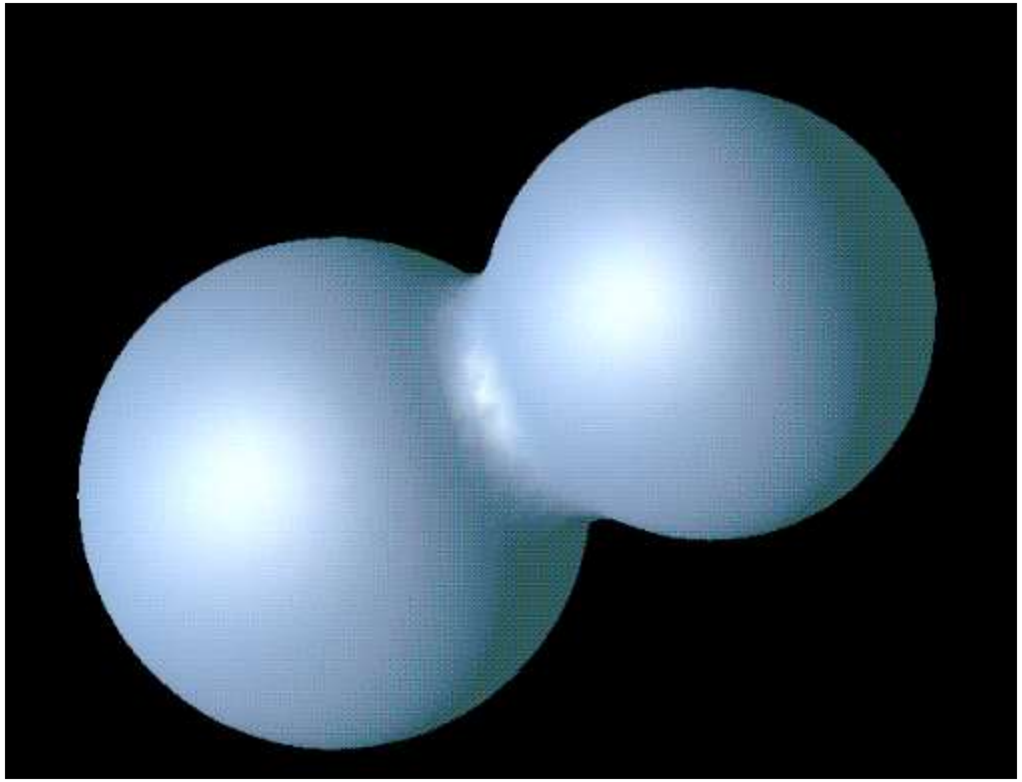


Figure 10-17
A composite blobby object formed with four Gaussian bumps.

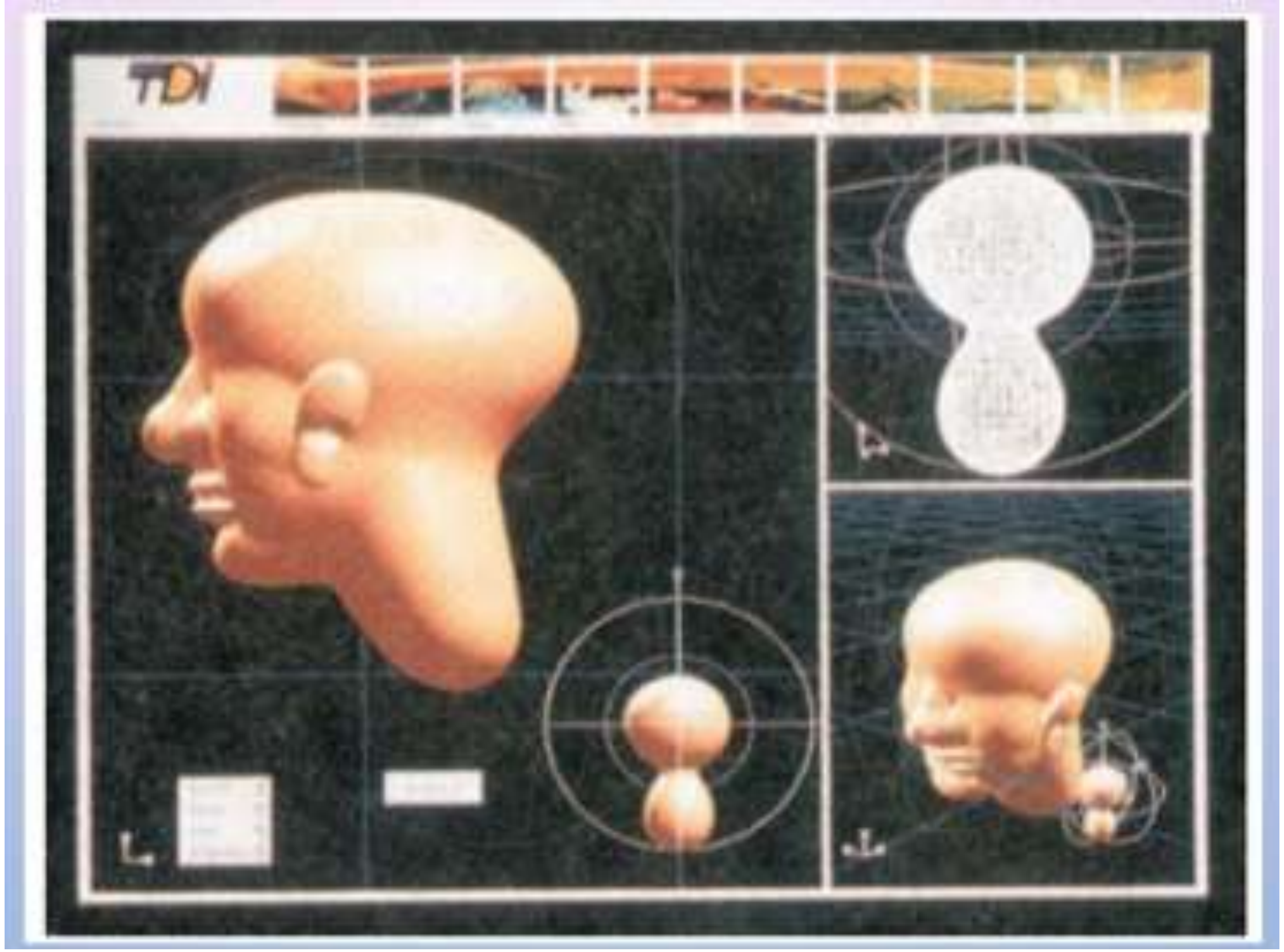
$$f(x, y, z) = \sum_k b_k e^{-a_k r_k^2} - T = 0$$

where $r_k^2 = \sqrt{x_k^2 + y_k^2 + z_k^2}$, parameter T is some specified threshold, and parameters a and b are used to adjust the amount of blobbiness of the individual objects. Negative values for parameter b can be used to produce dents instead of bumps.

Blobby objects – Metaballs



Blobby objects



Blobby objects - metaball

- The "metaball" model describes blobby objects as combinations of quadratic density functions of the form

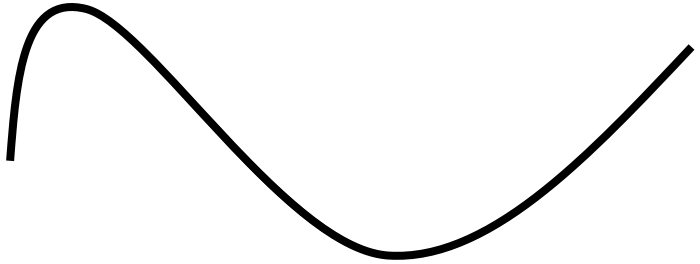
$$f(r) = \begin{cases} b(1 - 3r^2/d^2), & \text{if } 0 < r \le d/3 \\ \frac{3}{2}b(1 - r/d)^2, & \text{if } d/3 < r \le d \\ 0, & \text{if } r > d \end{cases}$$

And the "soft object" model uses the function

$$f(r) = \begin{cases} 1 - \frac{22r^2}{9d^2} + \frac{17r^4}{9d^4} - \frac{4r^6}{9d^6}, & \text{if } 0 < r \le d \\ 0, & \text{if } r > d \end{cases}$$

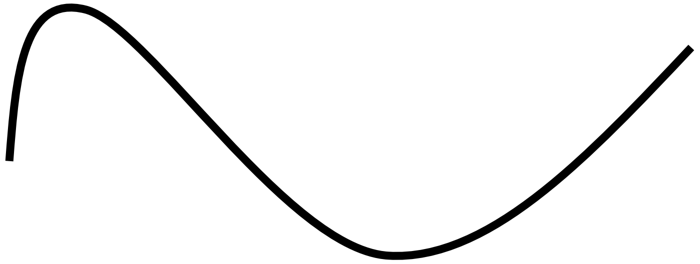
Spline

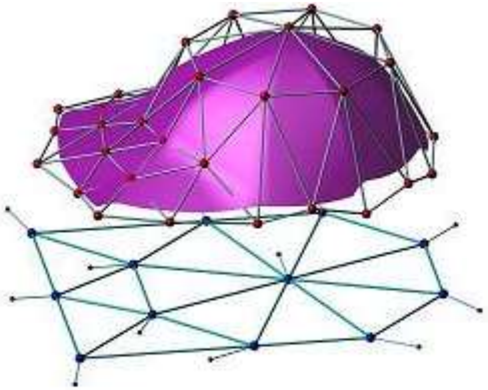
- Drafting terminology
  - Spline is a flexible strip that is easily flexed to pass through a series of design points (control points) to produce a smooth curve.
- Spline curve – a piecewise polynomial (cubic) curve whose first and second derivatives are continuous across the various curve sections.

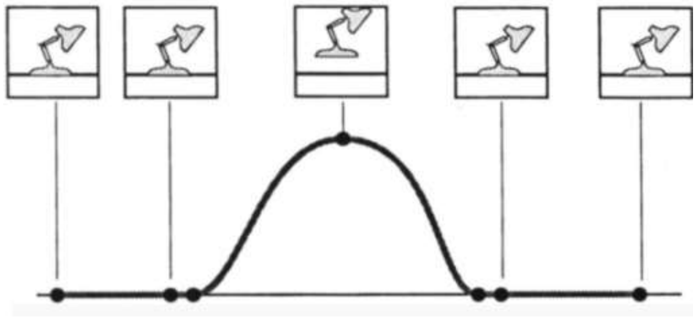


Spline Representations

- A spline is a smooth curve defined mathematically using a set of constraints
- Splines have many uses:
  - 2D illustration
  - Fonts
  - 3D Modelling
  - Animation

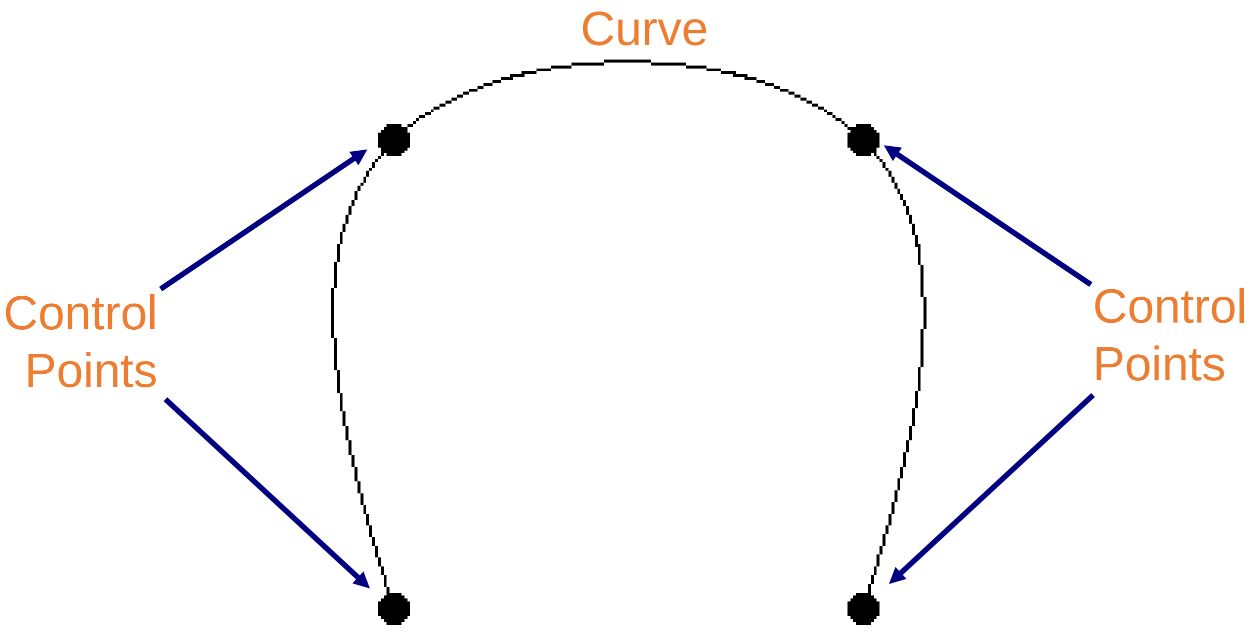






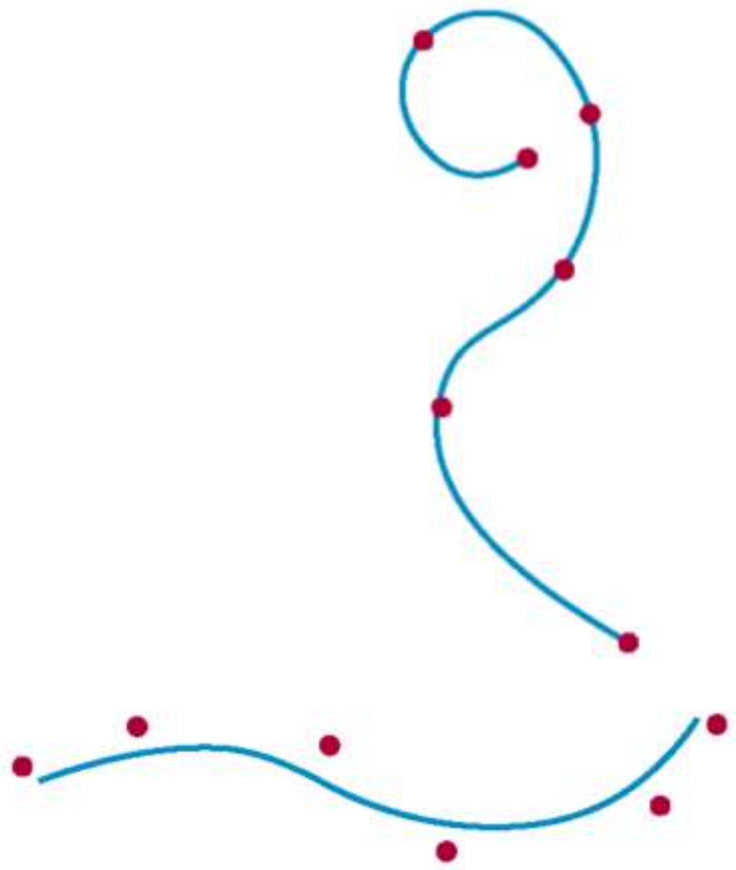
Big Idea

- User specifies control points
- Defines a smooth curve



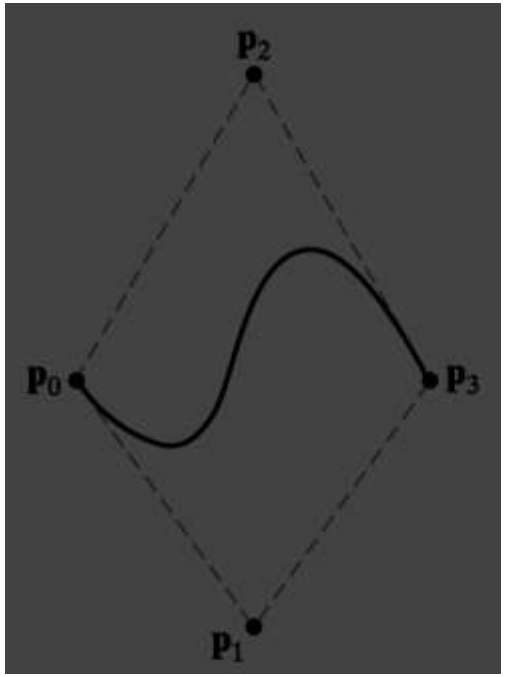
Interpolation Vs Approximation

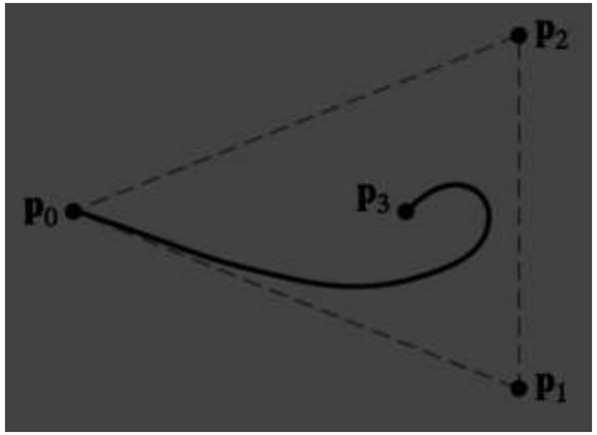
- A spline curve is specified using a set of **control points**
- There are two ways to fit a curve to these points:
  - **Interpolation -** the curve passes through all of the control points
  - **Approximation -** the curve does not pass through all of the control points
  - Approximation for structure or shape
  - Interpolation for animation



Convex Hulls

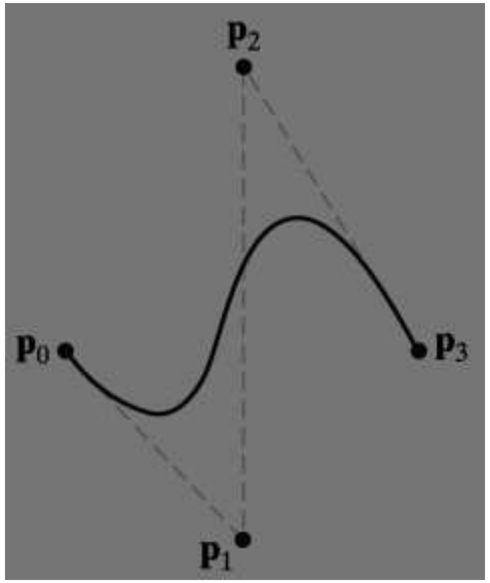
- The boundary formed by the set of control points for a spline is known as a **convex hull**
- Think of an elastic band stretched around the control points

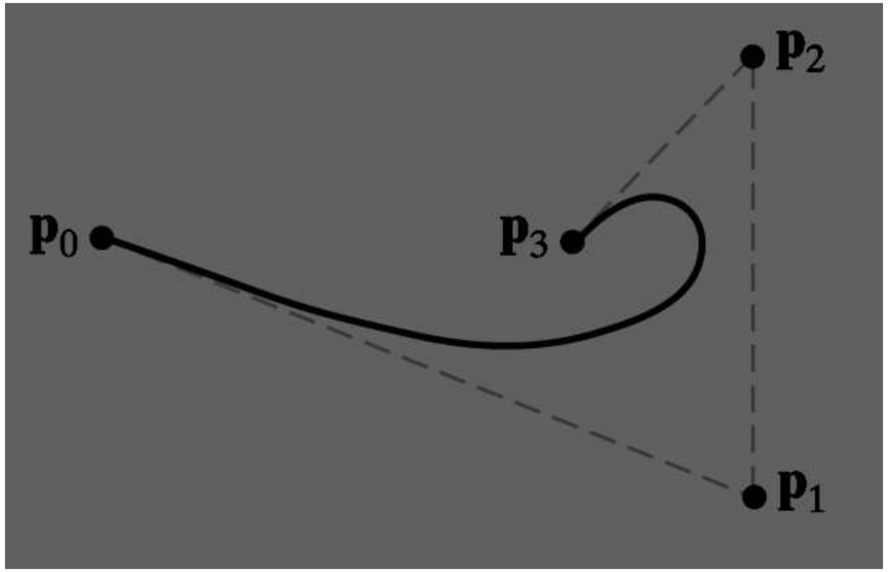




Control Graphs

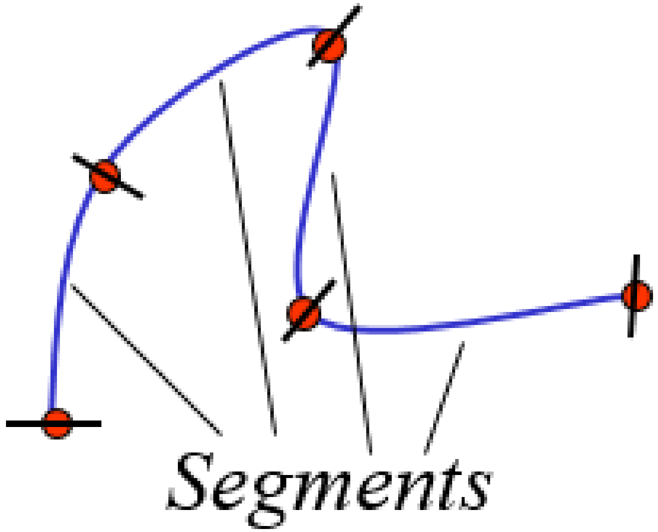
- A polyline connecting the control points in order is known as a **control graph**
- Usually displayed to help designers keep track of their splines





Piecewise cubic splines

Splines in computer graphics:
Piecewise cubic splines



Types of Curves

- A curve is an infinitely large set of points. Each point has two neighbors except endpoints. Curves can be broadly classified into three categories −
- **explicit, implicit,** and **parametric curves**.
- Implicit Curves

Implicit Curves

- Implicit curve representations define the set of points on a curve by employing a procedure that can test to see if a point in on the curve.
- Usually, an implicit curve is defined by an implicit function of the form −
- $f(x, y) = 0$
- Eg. A common example is the circle, whose implicit representation is
- $\mathbf{x^2 + y^2 - R^2 = 0}$

Explicit Curves

- A mathematical function $y = f(x)$ can be plotted as a curve.
- Such a function is the explicit representation of the curve.

Parametric curve

- The explicit and implicit curve representations can be used only when the function is known.
- Curves having parametric form are called parametric curves.
- In practice the parametric curves are used.
- Every point on the curve is having two neighbors (other than the end points).

Parametric curve

- A two-dimensional parametric curve has the following form –
- P(t) = f(t), g(t) or P(t) = x(t), y(t)
- The functions f and g become the (x, y) coordinates of any point on the curve, **and the points are obtained when the parameter t (or u)** is varied over a certain interval [a, b], normally [0, 1].

Parametric Continuity Conditions

- To ensure a smooth transition from one section of a piecewise parametric curve to the next, we can impose **various continuity conditions** at the connection points.
- If each section of a spline is described with a set of parametric coordinate functions of the form

$$x = x(u), \qquad y = y(u), \qquad z = z(u), \qquad u_1 \leq u \leq u_2$$

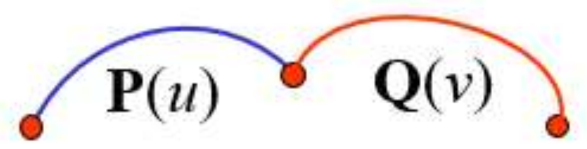
Parametric Continuity Conditions

- Three types of continuity

1. Zero Order Continuity
2. First Order Continuity
3. Second Order Continuity

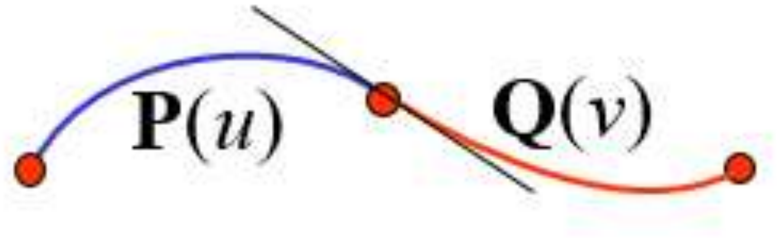
Zero Order Continuity

- Two piece of curve must meet at transition point
- Segments have to match 'nicely'.
- Given two segments **P**(u) and **Q**(v).
- We consider the transition of **P**(1) to **Q**(0).
- Zero-order parametric continuity
- C^0: **P**(1) = **Q**(0).
- Endpoint of **P**(u) coincides with start point **Q**(v).



First Order Continuity

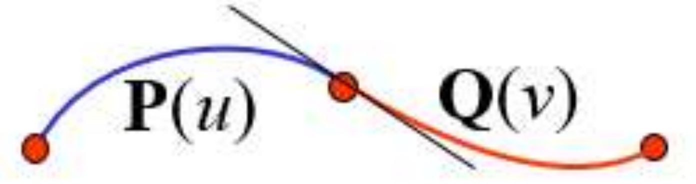
- First parametric derivatives (tangent lines) of the coordinate functions two successive curve sections are equal at their joining point.
- Segments have to match ‘nicely’.
- Given two segments **P**(u) and **Q**(v).
- We consider the transition of **P**(1) to **Q**(0).
- First order parametric continuity
- C^1: d**P**(1)/du = d**Q**(0)/dv.



- Direction of **P**(1) coincides with direction of **Q**(0).

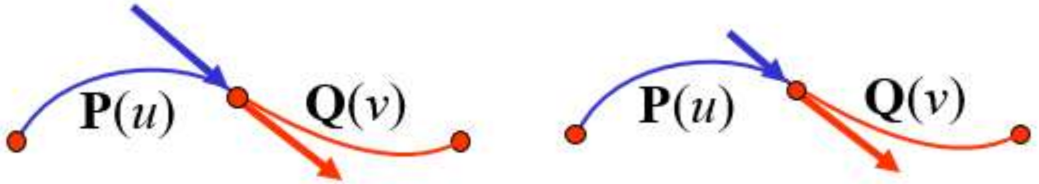
Second Order Continuity

- Second-order parametric continuity, or **C^2** continuity, means that both the **first and second parametric derivatives** of the two curve sections are the same at the intersection.
- Given two segments **P**(u) and **Q**(v).
- We consider the transition of **P**(1) to **Q**(0).
- Second order parametric continuity
- C^2: $d^2\mathbf{P}(1)/du^2 = d^2\mathbf{Q}(0)/dv^2$.
- **Curvatures in P(1) and Q(0) are equal.**



Geometric Continuity

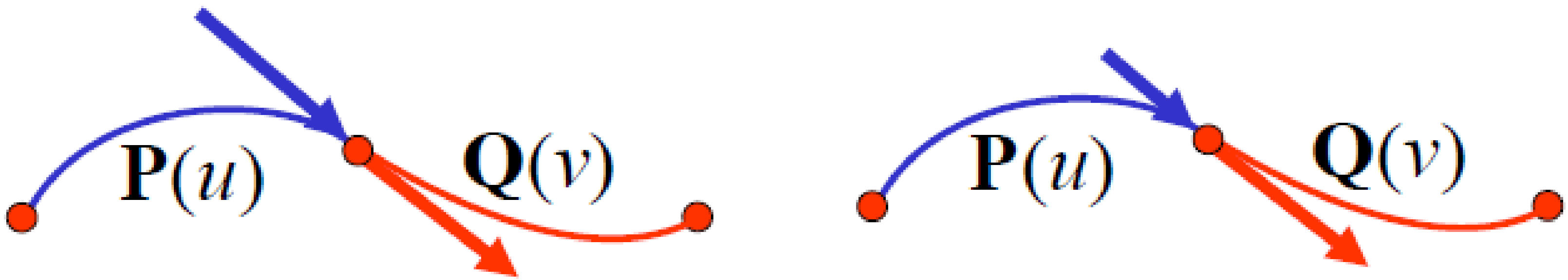
- It suffices to require that the directions are the same:
- geometric continuity.



Geometric Continuity

An alternate method for joining two successive curve sections

It suffices to require that the directions are the same: *geometric continuity.*



Geometric Continuity

An alternate method for joining two successive curve sections

1. Zero Order Geometric Continuity
2. First Order Geometric Continuity
3. Second Order Geometric Continuity

Geometric Continuity

1. Zero Order Geometric Continuity
   - the two curves sections must have the same coordinate position at the boundary point (**Same as zero order parametric continuity**)
2. First Order Geometric Continuity
   - the parametric first derivatives are **proportional** at the intersection of two successive sections (In parametric continuity these are equal)
3. Second Order Geometric Continuity
   - both the **first and second order derivatives of the two curve sections are proportional** at their boundary

Spline Representation

- There are three equivalent methods for specifying a particular spline representation:
- (1) We can state the set of boundary conditions that are imposed on the spline;
- (2) we can state the matrix that characterizes the spline;
- (3) we can state the set of blending functions

boundary conditions

- Boundary conditions for this curve might be set,
- for example, on the endpoint coordinates *x(0)* and x(l) and on the parametric first derivatives at the endpoints *x'(0)* and x ' (1) .
- **Boundary conditions are sufficient to determine the values of the four coefficients ax, *bx,* cx, and *dx.***

boundary conditions

$$\mathbf{P}(u) = \left(x(u), y(u), z(u)\right) = \begin{pmatrix} a_x u^3 + b_x u^2 + c_x u + d_x \\ a_y u^3 + b_y u^2 + c_y u + d_y \\ a_z u^3 + b_z u^2 + c_z u + d_z \end{pmatrix}^{\mathrm{T}}, \text{ with } 0 \le u \le 1$$

$$= \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix} = \mathbf{UC}.$$

U: *Powers of u* **C**: *Coefficient matrix* H&B 8-8:420-425

Matrix Form

- we can obtain the matrix that characterizes this spline curve by first rewriting Eq as the matrix product

Variant 2 :

$$\mathbf{P}(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} M_{00} & M_{01} & M_{02} & M_{03} \\ M_{10} & M_{11} & M_{12} & M_{13} \\ M_{20} & M_{21} & M_{22} & M_{23} \\ M_{30} & M_{31} & M_{32} & M_{33} \end{pmatrix} \boxed{\begin{pmatrix} P_{0x} & P_{0y} & P_{0z} \\ P_{1x} & P_{1y} & P_{1z} \\ P_{2x} & P_{2y} & P_{2z} \\ P_{3x} & P_{3y} & P_{3z} \end{pmatrix}}$$

$$= \mathbf{U}\mathbf{M}_{\text{spline}} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = \mathbf{U}\mathbf{M}_{\text{spline}}\mathbf{M}_{\text{geom}}$$

Control points or
Control vectors

H&B 8-8:420-425

Matrix $\mathbf{M}_{\text{spline}}$:'translates' geometric info to coefficients

blending functions

Variant 3:

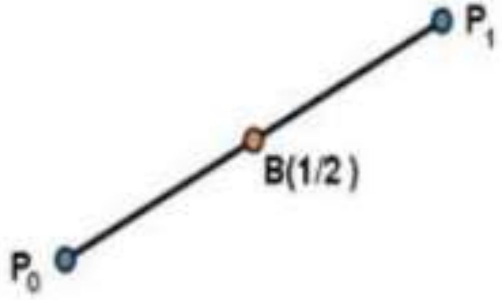
$$\mathbf{P}(u) = B_0(u)\mathbf{P}_0 + B_1(u)\mathbf{P}_1 + B_2(u)\mathbf{P}_2 + B_3(u)\mathbf{P}_3$$

$$= \sum_{k=0}^{3} B_k(u)\mathbf{P}_k$$

with $B_k(u) = b_{k3}u^3 + b_{k2}u^2 + b_{k1}u + b_{k0}$ blending functions

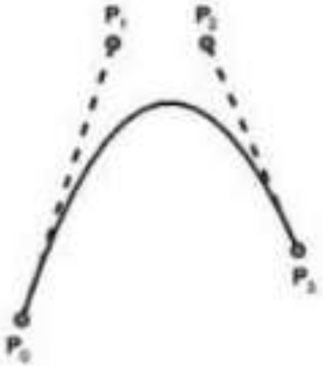
Bezier curves

- Bezier curve is discovered by the French engineer **Pierre Bézier**.
- These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve.



Simple Bezier Curve

Quadratic Bazier Curve



Cubic Bazier Curve

Bezier curves

can be represented mathematically as –

$$\sum_{k=0}^{n} P_i B_i^n(t)$$

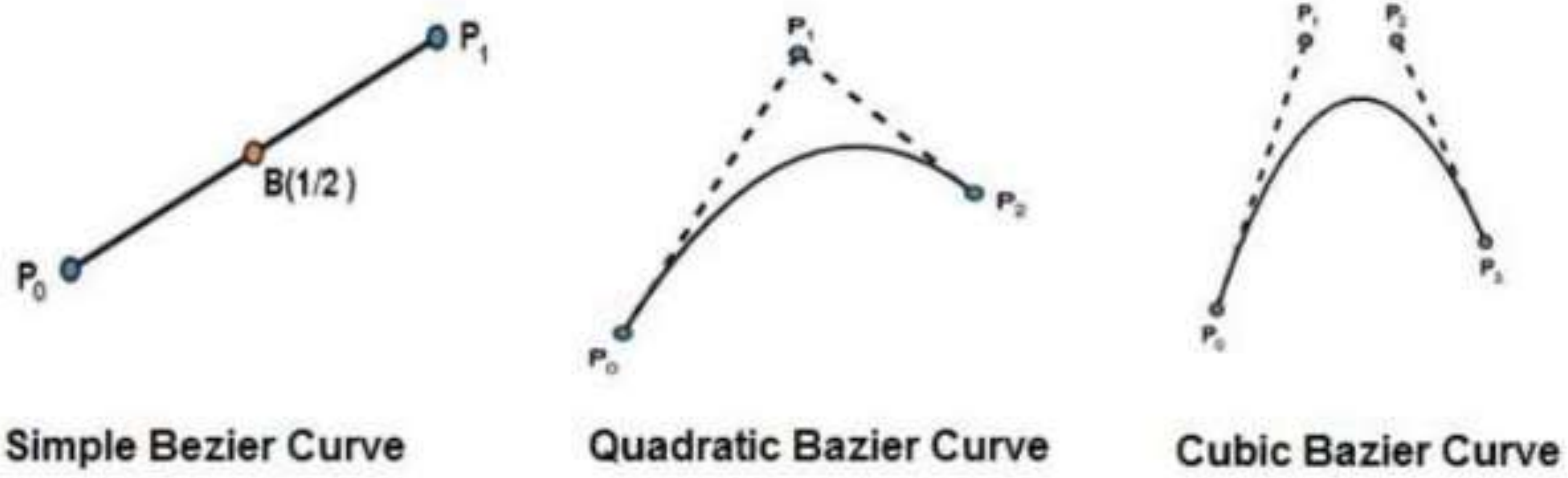
Where p_i is the set of points and $B_i^n(t)$ represents the Bernstein polynomials which are given by –

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

Where **n** is the polynomial degree, **i** is the index, and **t** is the variable.

Bezier curves

- The simplest Bézier curve is the straight line from the point P0 to P1.
- A quadratic Bezier curve is determined by three control points.
- A cubic Bezier curve is determined by four control points.



Properties of Bezier curves

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.

Properties of Bezier curves

- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

Bezier curves

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

$$\mathbf{P}(u) = \sum_{k=0}^{n} \mathbf{p}_k BEZ_{k,n}(u), \qquad 0 \le u \le 1$$

Bezier curves

$$\mathbf{P}(u) = \sum_{k=0}^{n} \mathbf{p}_k BEZ_{k,n}(u), \qquad 0 \le u \le 1 \tag{10-40}$$

The Bézier blending functions $BEZ_{k,n}(u)$ are the *Bernstein polynomials:*

$$BEZ_{k,n}(u) = C(n, k)u^k(1 - u)^{n-k} \tag{10-41}$$

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

$$C(n, k) = \frac{n!}{k!(n - k)!} \tag{10-42}$$

Bezier curves

three parametric equations for the individual curve coordinates:

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

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

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

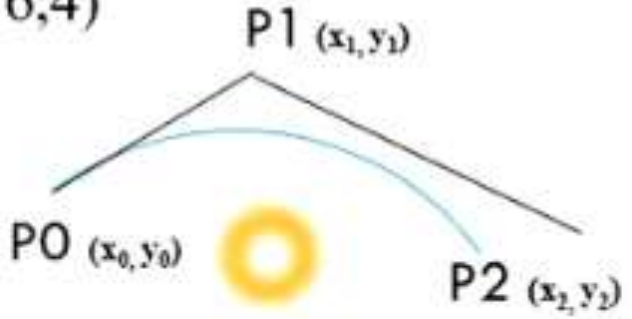
Bezier curves Numericals and derivation

Question: - Construct Bezier Curve for control points P0(4,2) P1(8,8) P2(16,4)

Solution: - 3 Control Points

so degree = 2

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(u), \quad 0 \le u \le 1$$



Bezier curves and surfaces

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(u), \quad 0 \le u \le 1$$

$$P(u) = P_0 B_{0,2}(u) + P_1 B_{1,2}(u) + P_2 B_{2,2}(u)$$

Now parametric equation

$$x = x_0 B_{0,2}(u) + x_1 B_{1,2}(u) + x_2 B_{2,2}(u)$$

$$y = y_0 B_{0,2}(u) + y_1 B_{1,2}(u) + y_2 B_{2,2}(u)$$

Bezier curves and surfaces

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

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

$$P(u) = P_0\, B_{0,2}(u) + P_1\, B_{1,2}(u) + P_2\, B_{2,2}(u)$$

Now parametric equation

$$x = x_0\, B_{0,2}(u) + x_1\, B_{1,2}(u) + x_2\, B_{2,2}(u)$$
$$y = y_0\, B_{0,2}(u) + y_1\, B_{1,2}(u) + y_2\, B_{2,2}(u)$$

Now $B_{0,2}(u) = C(n,i)\, u^i (1-u)^{n-1}$

$\Rightarrow \quad (n!)\,/\,i!(n-i)!\; u^i (1-u)^{n-1}$

$\Rightarrow (2!)\,/\,0!(2-0)!\; u^i (1-u)^{n-1} \quad \Rightarrow$ $\mathbf{(1-u)^2}$

$B_{1,2}(u) = C(n,i)\, u^i (1-u)^{n-1}$ after solving $\Rightarrow$ $\mathbf{2u(1-u)}$

$B_{1,2}(u) = C(n,i)\, u^i (1-u)^{n-1}$ after solving $\Rightarrow$ $\mathbf{u^2}$

Bezier curves and surfaces

Now after substitute values of blending function

$$x = x_0\,\mathbf{(1\text{-}u)^2} + x_1\,\mathbf{2u(1\text{-}u)} + x_2\,\mathbf{u^2}$$
$$y = y_0\,\mathbf{(1\text{-}u)^2} + y_1\,\mathbf{2u(1\text{-}u)} + y_2\,\mathbf{u^2}$$

Now put the value of $x_0 = 4$ $x_1 = 8$ $x_2 = 16$

$y_0 = 2$ $y_1 = 8$ $y_2 = 4$

$x = 4\,\mathbf{(1\text{-}u)^2} + 8 * \mathbf{2 * u(1\text{-}u)} + 16\,\mathbf{u^2}$ **after solving equation x =** $4u^2 + 8u + 4$

$y = 2\,\mathbf{(1\text{-}u)^2} + 8 * \mathbf{2 * u(1\text{-}u)} + 4\,\mathbf{u^2}$ **after solving equation y** $= -10u^2 + 12u + 2$

Bezier curves and surfaces

$x = 4\,(1\text{-}u)^2 + 8 * \mathbf{2*u(1\text{-}u)} + 16\,\mathbf{u^2}$ **after solving equation** $\mathbf{x} =\ 4\,u^2 + 8\,u + 4$

$y = 2\,(1\text{-}u)^2 + 8 * \mathbf{2*u(1\text{-}u)} + 4\,\mathbf{u^2}$ **after solving equation** $\mathbf{y} = -10\,u^2 + 12\,u + 2$

u	x(u)	y(u)
0	4	2
0.2	5.76	4.0
0.4	7.84	5.20
0.6	10.24	5.6
0.8	12.96	5.2
1	16	4

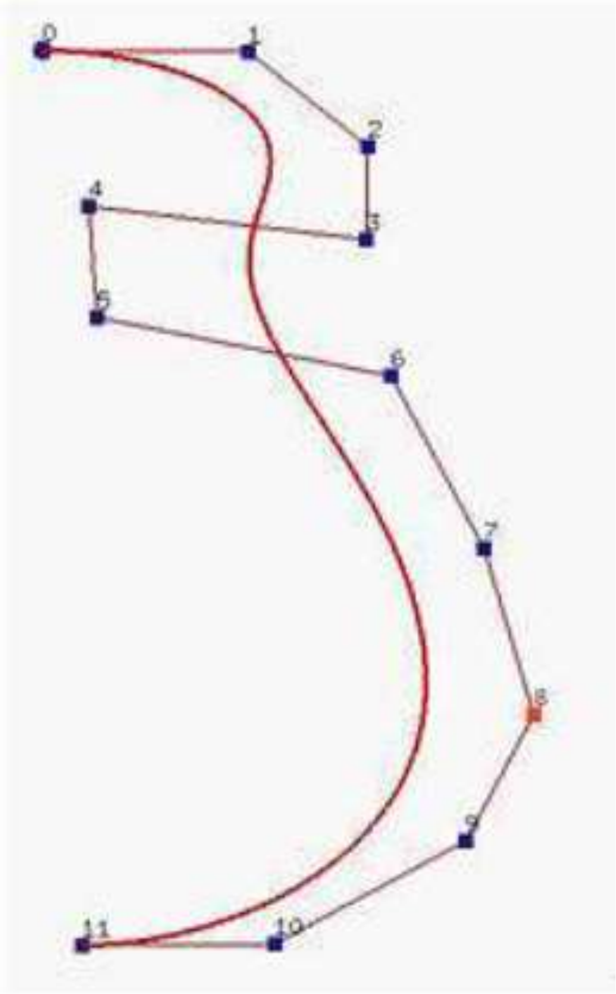
Bezier curves and surfaces

- 1)Given control points (10,100), (50, 100), (70,120) and (100, 150). Calculate coordinates of any four points lying on the corresponding Beizer curve.

- 2) Set up the equation of Beizer curve and roughly trace it for three control points (1,1), (2,2) and (3,1).
  (From CO-RCS603.4)

B-Spline

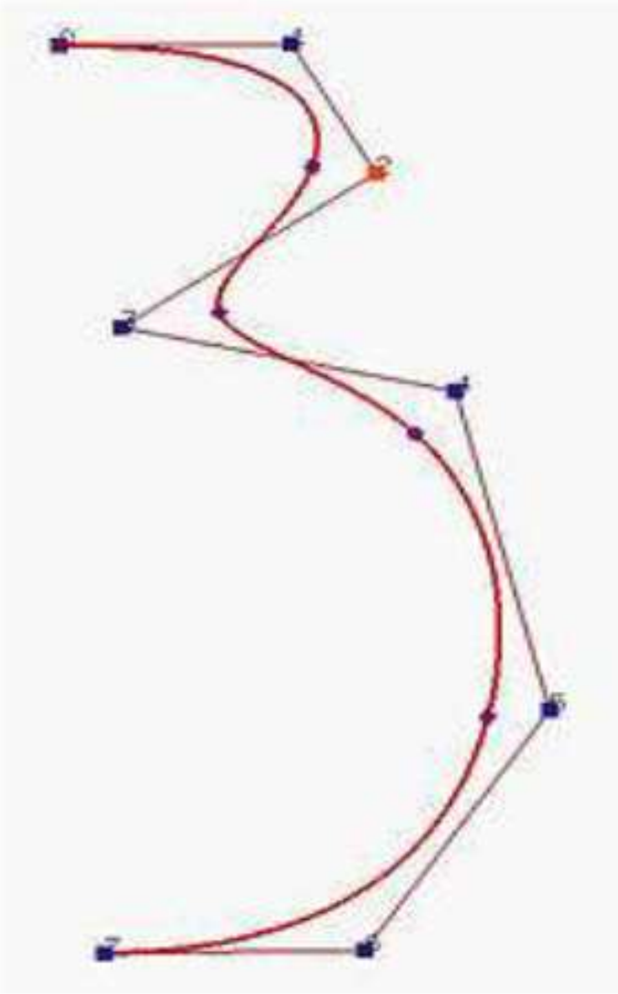
- Motivation (recall bezier curve)
  - The degree of a Bezier Curve is determined by the number of control points
  - E. g. (bezier curve degree 11) – difficult to bend the "neck" toward the line segment $\mathbf{P}_4\mathbf{P}_5$.
  - Of course, we can add more control points.
  - BUT this will increase the degree of the curve → increase computational burden



B-Spline

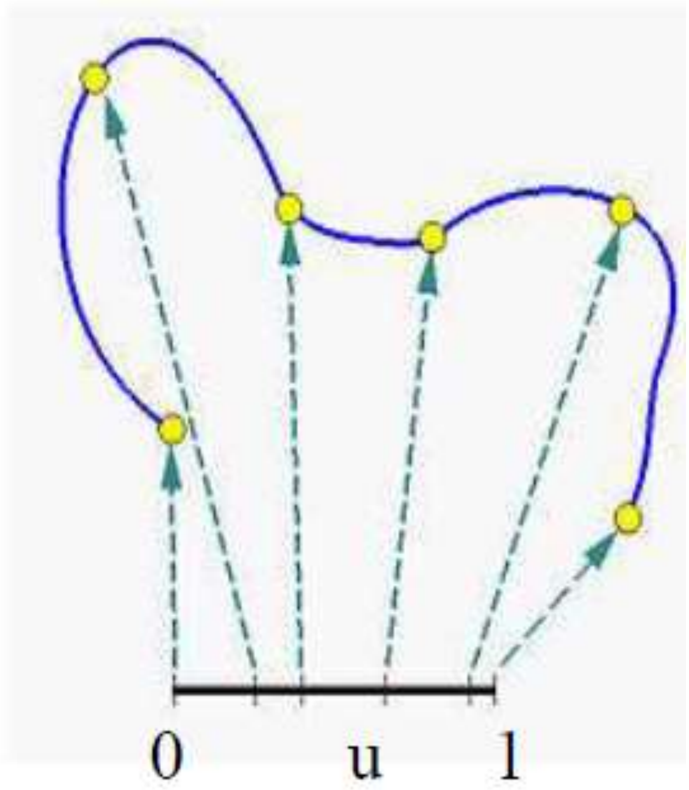
Motivation (recall bezier curve)

- moving a control point affects the shape of the entire curve- (*global modification property*) – undesirable.
- Thus, the solution is B-Spline – the degree of the curve is independent of the number of control points
- E.g - right figure – a B-spline curve of degree 3 defined by 8 control points



B-Spline

- In fact, there are five Bézier curve segments of degree 3 joining together to form the B-spline curve defined by the control points
- little dots subdivide the B-spline curve into Bézier curve segments.
- Subdividing the curve directly is difficult to do → so, subdivide the domain of the curve by points called *knots*



B-Spline

- In summary, to design a B-spline curve, we need a set of control points, a set of knots and a degree of curve.

B-Spline : definition

- $P(u) = \sum N_{i,k}(u)p_i$ $(u_0 < u < u_m)$
- u_i → knot
- $[u_i, u_{i+1})$ → knot span
- $(u_0, u_1, u_2, \ldots.\ u_m\ )$ → knot vector
- The point on the curve that corresponds to a knot u_i, → knot point ,$P(u_i)$
- If knots are equally space → uniform (e.g, 0, 0.2, 0.4, 0.6…)
- Otherwise → non uniform (e.g: 0, 0.1, 0.3, 0.4, 0.8 …)

Type of B-Spline uniform knot vector

Non-periodic knots (open knots)

-First and last knots are duplicated k times.
-E.g (0,0,0,1,2,2,2)
-Curve pass through the first and last control points

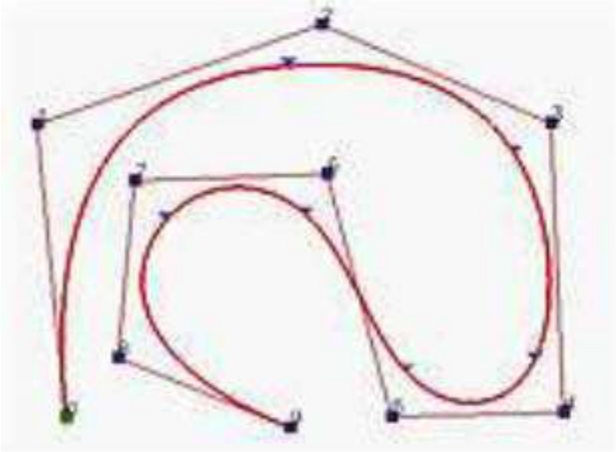
Periodic knots (non-open knots)

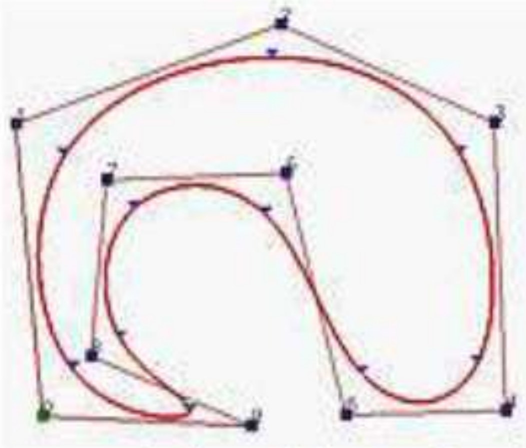
-First and last knots are not duplicated – same contribution.
-E.g (0, 1, 2, 3)
-Curve doesn't pass through end points.
- used to generate closed curves (when first = last control points)

Type of B-Spline knot vector

Non-periodic knots
(open knots)

Periodic knots
(non-open knots)





(Closed knots)

Non-periodic (open) uniform B-Spline

- The knot spacing is evenly spaced except at the ends where knot values are repeated *k* times.
- E.g $P(u) = \sum_{i=0}^{n} N_{i,k}(u)p_i \qquad (u_0 < u < u_m)$
- Degree = k-1, number of control points = n + 1
- Number of knots = m + 1 @ n+ k + 1

→ for degree = 1 and number of control points = 4 → (k = 2, n = 3)

→ Number of knots = n + k + 1 = 6

non periodic uniform knot vector (0,0,1,2,3, 3)

* Knot value between 0 and 3 are equally spaced → uniform

Non-periodic (open) uniform B-Spline

- For any value of parameters k and n, non periodic knots are determined from

$$u_i = \begin{cases} 0 & 0 \le i < k \\ i - k + 1 & k \le i \le n \\ n - k + 2 & n < i \le n+k \end{cases} \qquad (1.3)$$

e.g $\quad k=2,\ n = 3$

$$u_i = \begin{cases} 0 & 0 \le i < 2 \\ i - 2 + 1 & 2 \le i \le 3 \\ 3 - 2 + 2 & 3 < i \le 5 \end{cases}$$

$u = (0, 0, 1, 2, 3, 3)$

B-Spline basis function

$$N_{i,k}(u) = (u - u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1} - u_i} + (u_{i+k} - u)\frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \qquad (1.1)$$

$$N_{i,1} = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & \text{Otherwise} \end{cases} \qquad (1.2)$$

Non-periodic (open) uniform B-Spline

Example

- Find the knot values of a non periodic uniform B-Spline which has degree = 2 and 3 control points. Then, find the equation of B-Spline curve in polynomial form.

Non-periodic (open) uniform B-Spline

Answer

- Degree = k-1 = 2 → k=3
- Control points = n + 1 = 3 → n=2
- Number of knot = n + k + 1 = 6
- Knot values → u_0=0, u_1=0, u_2=0, u_3=1,u_4=1,u_5= 1

Non-periodic (open) uniform B-Spline

Answer(cont)

- To obtain the polynomial equation,
  $P(u) = \sum_{i=0}^{n} N_{i,k}(u)p_i$
- $= \sum_{i=0}^{2} N_{i,3}(u)p_i$
- $= N_{0,3}(u)p_0 + N_{1,3}(u)p_1 + N_{2,3}(u)p_2$

- firstly, find the $N_{i,k}(u)$ using the knot value that shown above, start from $k=1$ to $k=3$

The polynomial equation, $P(u) = \sum_{i=0}^{n} N_{i,k}(u)p_i$

- $P(u) = N_{0,3}(u)p_0 + N_{1,3}(u)p_1 + N_{2,3}(u)p_2$
- **$= (1- u)^2\, p_0 + 2u(1-u)\, p_1 + u^2 p_2 \quad (0 <= u <= 1)$**

Bspline curves and surfaces

- From pdf

References

- Bezier Curve
- https://www.tutorialspoint.com/computer_graphics/computer_graphics_curves.htm