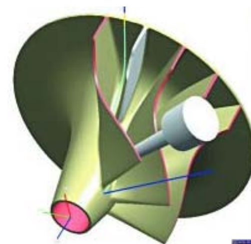
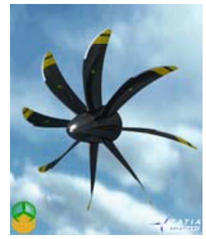
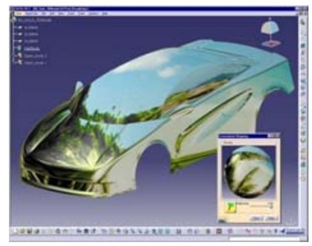
Curves

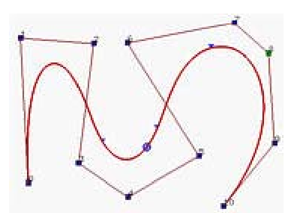
- Foundation of Free-form Surfaces

Why Study Curves?

- Curves are the basics for surfaces
- When asked to modify a particular entity on a CAD system, knowledge of the entities can increase your productivity
- Understand how the math presentation of various curve entities relates to a user interface
- Understand what is impossible and which way can be more efficient when creating or modifying an entity

Why Not Simply Use a Point Matrix to Represent a Curve?

- Storage issue and limited resolution
- Computation and transformation
- Difficulties in calculating the intersections or curves and physical properties of objects
- Difficulties in design (e.g. control shapes of an existing object)
- Poor surface finish of manufactured parts

Advantages of Analytical Representation for Geometric Entities

- A few parameters to store
- Designers know the effect of data points on curve behavior, <u>control</u>, <u>continuity</u>, and curvature
- Facilitate calculations of intersections, object properties, etc.

Analytic Curves vs. Synthetic Curves

- Analytic Curves are points, lines, arcs and circles, fillets and chamfers, and conics (ellipses, parabolas, and hyperbolas)

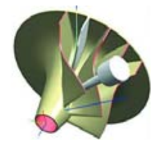
- Synthetic curves include various types of splines (cubic spline, B-spline, Beta-spline) and Bezier curves.

Curved Surfaces

- In CAD, We want to find a math form for representing curved surfaces, that :
  (a) look nice (smooth contours)
  (b) is easy to manipulate and manufacture
  (c) follows prescribed shape (airfoil design)

To study the curved surface, we need to start from curves.

Parametric Representation

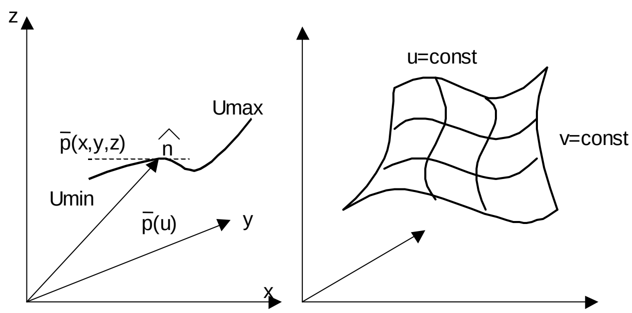


* a curve

$$\overline{P}(u) = [x(u), y(u), z(u)]^T$$

* a surface

$$\overline{P}(u,v) = [x(u,v), y(u,v), z(u,v)]^T$$



We can represent any functions of curve (curved surface) using parametric equation.

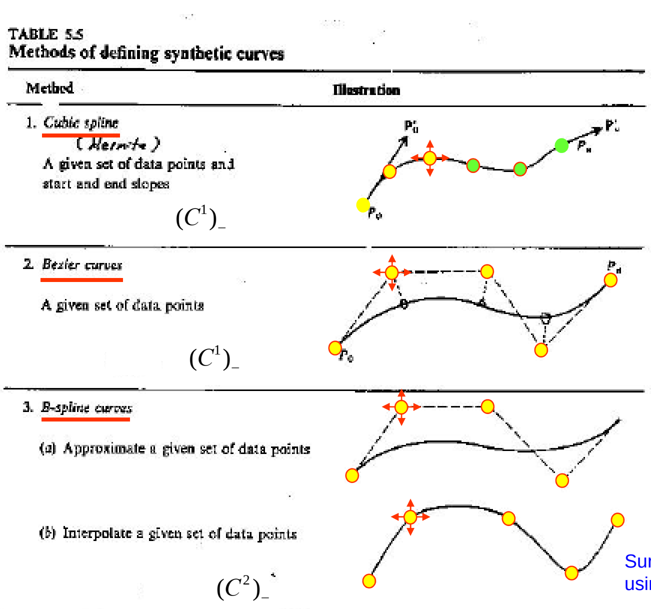


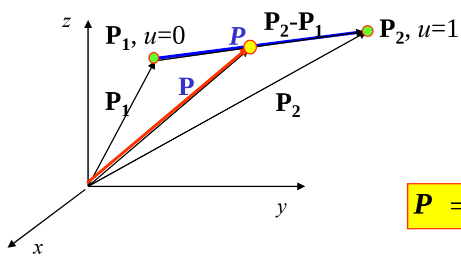
TABLE 5.5
Methods of defining synthetic curves

Method	Illustration
1. Cubic spline (Hermite) A given set of data points and start and end slopes (C^1)	
2. Bezier curves A given set of data points (C^1)	
3. B-spline curves (a) Approximate a given set of data points (b) Interpolate a given set of data points (C^2)	

(C^i) -- *ith* order continuity

Parametric Representation of Lines

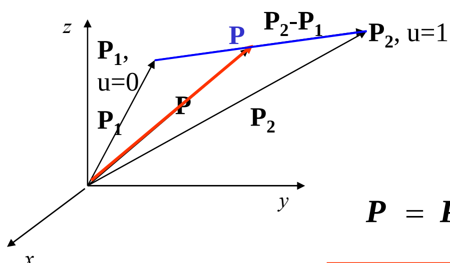
- How is a line equation converted by the CAD/CAM software into the line database?
- How are the mathematical equation correlated to user commands to generate a line?



$$P = P_1 + (P - P_1)$$
$$P - P_1 = u\,(P_2 - P_1)$$

$$\boxed{P = P_1 + u(P_2 - P_1),\ \ 0 \le u \le 1}$$

Lines



$$P = P_1 + u(P_2 - P_1),\ \ 0 \le u \le 1$$

$$\begin{cases} x = x_1 + u(x_2 - x_1) \\ y = y_1 + u(y_2 - y_1) & 0 \le u \le 1 \\ z = z_1 + u(z_2 - z_1) \end{cases}$$

Circle

Representation 1 (Non-parametric)

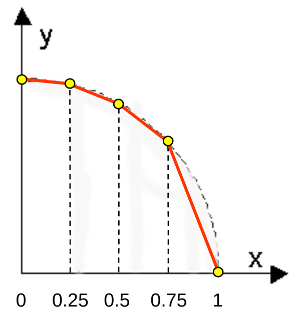
$$x^2 + y^2 = 1$$

(a)

$$x = u \quad (parameter)$$

$$y = \sqrt{1 - u^2}$$

- poor definition
- square root complicated to compute



Circle

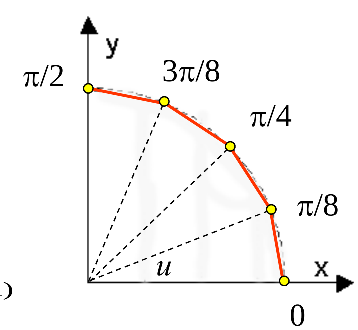
Representation 2

(b)

$$x = \cos u$$

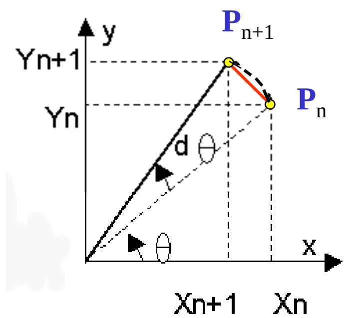
$$y = \sin u$$

- better definition than (a)
- but still slow



Circle

Representation 3



Recursive approach

$$\begin{cases} x_n = r\cos\theta \\ y_n = r\sin\theta \end{cases}$$ -- P_n

$$x_{n+1} = r\cos(\theta + d\theta) = r\cos\theta\cos d\theta - r\sin\theta\sin d\theta$$

$$\boxed{\begin{cases} x_{n+1} = x_n\cos d\theta - y_n\sin d\theta \\ y_{n+1} = y_n\cos d\theta + x_n\sin d\theta \end{cases}}$$ -- P_{n+1}

Observation: curves are represented by a series of line-segments

Similarly all conic sections can be represented.

Ellipse

$$\begin{cases} x = x_o + A\cos\theta \\ y = y_o + B\sin\theta \qquad 0 \le \theta \le 2\pi \\ z = z_o \end{cases}$$

The computer uses the same method as in the Representation 3 of circle to reduce the amount of calculation.

Parabola

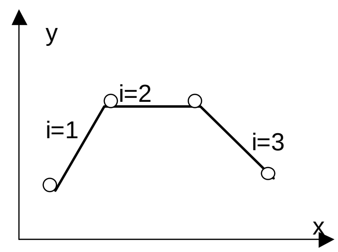
$$\begin{cases} x = x_o + Au^2 \\ y = y_o + 2Au \qquad 0 \le u \le \infty \\ z = z_o \end{cases}$$

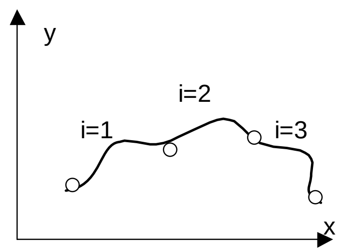
Parametric Representation of Synthetic Curves

- Analytic curves are usually not sufficient to meet geometric design requirements of mechanical parts.
- Many products need **free-form**, or **synthetic curved surfaces**.
- Examples: car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles, and bottles
- The need for synthetic curves in design arises on occasions:
  - when a curve is represented by a collection of measured data points and (**generation**)
  - when a curve must change to meet new design requirements. (**modification**)

The Order of Continuity

The order of continuity is a term usually used to measure the degree of continuous derivatives (C^0, C^1, C^2).





Simplest Case Linear Segment

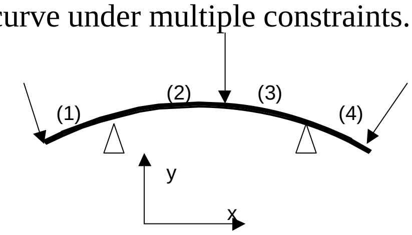
$y_i = a_{i0} + a_{i1}x$

High order polynomial may lead to "ripples"

$y_i = a_{i0} + a_{i1}x + ... + a_{in}x^n$

Splines – Ideal Order

Splines — a mechanical beam with bending deflections, or a smooth curve under multiple constraints.

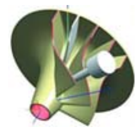


$$y''(x) = R(x) = \frac{M(x)}{EI} = \frac{a_i x + b_i}{EI}$$

$$y(x) = \frac{1}{EI}\left[\frac{a_i}{6}x^3 + \frac{b_i}{2}x^2 + c_i x + d_i\right]$$

Cubic Spline

Hermite Cubic Splines



$$\vec{P}(u) = [x(u), y(u), z(u)]^T$$

Cubic Spline

$$\begin{cases} x(u) = c_{3x}u^3 + c_{2x}u^2 + c_{1x}u + c_{0x} \\ y(u) = c_{3y}u^3 + c_{2y}u^2 + c_{1y}u + c_{0y} \\ z(u) = c_{3z}u^3 + c_{2z}u^2 + c_{1z}u + c_{0z} \end{cases}$$

3×4 = 12 coefficients to be determined

$$\vec{p}(u) = [x(u)\ \ y(u)\ \ z(u)]^T = \sum_{i=0}^{3} \vec{C}_i u^i \qquad (0 \le u \le 1)$$

$$= [u^3\ u^2\ u\ 1]\begin{bmatrix} \vec{C}_3 \\ \vec{C}_2 \\ \vec{C}_1 \\ \vec{C}_0 \end{bmatrix} = [U^T][\vec{C}]$$

Hermite Cubic Splines

$$\vec{P} = \sum_{i=0}^{3} \vec{C}_i u^i = \vec{C}_3 u^3 + \vec{C}_2 u^2 + \vec{C}_1 u^1 + \vec{C}_0$$

$$\vec{P}' = 3\vec{C}_3 u^2 + 2\vec{C}_2 u + \vec{C}_1$$

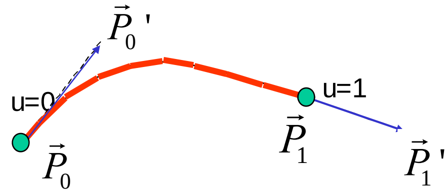
Two End Points

$$u = 0 \qquad \begin{cases} \vec{P}_0 = \vec{C}_0 \\ \vec{P}_0' = \vec{C}_1 \end{cases}$$

$$u = 1 \qquad \begin{cases} \vec{P}_1 = \vec{C}_3 + \vec{C}_2 + \vec{C}_1 + \vec{C}_0 \\ \vec{P}_1' = 3\vec{C}_3 + 2\vec{C}_2 + \vec{C}_1 \end{cases}$$

4×3 equations from two control points

Boundary Conditions: Location of the two end points and their slopes



Hermite Cubic Splines

$$\vec{C}_0 = \vec{P}_0$$
$$\vec{C}_1 = \vec{P}_0'$$
$$\vec{C}_2 = 3(\vec{P}_1 - \vec{P}_0) - 2\vec{P}_0' - \vec{P}_1'$$
$$\vec{C}_3 = 2(\vec{P}_0 - \vec{P}_1) + \vec{P}_0' + \vec{P}_1'$$

12 unknowns
and
12 equations

$$\boxed{\vec{P} = \sum_{i=0}^{3} \vec{C}_i u^i = \vec{C}_3 u^3 + \vec{C}_2 u^2 + \vec{C}_1 u^1 + \vec{C}_0}$$

$$\vec{P}(u) = (2u^3 - 3u^2 + 1)\vec{P}_0 + (-2u^3 + 3u^2)\vec{P}_1 + (u^3 - 2u^2 + u)\vec{P}_0' + (u^3 - u^2)\vec{P}_1'$$

All parameters can be determined

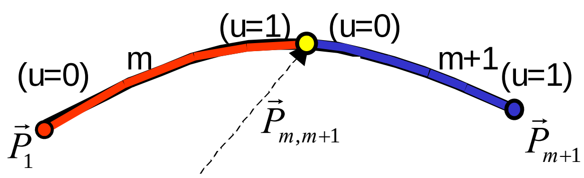
Hermite Cubic Splines

- $$\vec{P}(u) = (2u^3 - 3u^2 + 1)\vec{P}_0 + (-2u^3 + 3u^2)\vec{P}_1 + (u^3 - 2u^2 + u)\vec{P}_0' + (u^3 - u^2)\vec{P}_1'$$
$$= \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{P}_0 \\ \vec{P}_1 \\ \vec{P}_0' \\ \vec{P}_1' \end{bmatrix}$$
$$= \boxed{U^T \; [M_H] \; \vec{V}} \qquad 0 \le u \le 1$$

Based on: Location of the two end points and their slopes

- $$\vec{P}'(u) = (6u^2 - 6u)\vec{P}_0 + (-6u^2 + 6u)\vec{P}_1 + (3u^2 - 4u + 1)\vec{P}_0' + (3u^2 - 2u)\vec{P}_1' \qquad 0 \le u \le 1$$

Joining Cubic Spline Segments



$\vec{P}_m(0) = \vec{P}_1,\ \vec{P}'_m(0) = \vec{P}'_1;\quad \vec{P}_{m+1}(1) = \vec{P}_{m+1},\ \vec{P}'_{m+1}(1) = \vec{P}'_{m+1}$

$$\boxed{\begin{aligned} &\vec{P}_m(1) = \vec{P}_{m+1}(0) = \vec{P}_{m,m+1};\\ &\vec{P}'_m(1) = \vec{P}'_{m+1}(0);\\ &\vec{P}''_m(1) = \vec{P}''_{m+1}(0) \end{aligned}}$$

2 curve segments & 24 unknowns

6 eqs. from first point,
6 eqs. from last point,
4×3 eqs. from joining point

Go through center point, have same 1st and 2nd order derivatives

Joining Cubic Spline Segments

$$\vec{P}'(u) = (6u^2 - 6u)\vec{P}_0 + (-6u^2 + 6u)\vec{P}_1 + (3u^2 - 4u + 1)\vec{P}'_0 + (3u^2 - 2u)\vec{P}'_1 \qquad 0 \le u \le 1$$

$$\vec{P}'' = (12u - 6)\vec{P}_0 + (-12u + 6)\vec{P}_1 + (6u - 4)\vec{P}'_0 + (6u - 2)\vec{P}'_1$$

$$\vec{P}''(u_1 = 1) = \vec{P}''(u_2 = 0)$$

$$\vec{P}''(u_1 = 1) = 6\vec{P}_0 - 6\vec{P}_1 + 2\vec{P}'_0 + 4\vec{P}'_1$$

$$\vec{P}''(u_2 = 0) = -6\vec{P}_1 + 6\vec{P}_2 - 4\vec{P}'_1 - 2\vec{P}'_2$$

$\ldots$

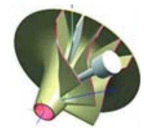
Questions on Cubic Splines

- What are the control parameters to change the shape of a cubic spline?
- What if I want to change a local curvature?
- Is there any way I can increase the order of continuity on a cubic spline?
- Can I improve the order of continuity by adding more points?

Disadvantages of Cubic Splines

- The order of the curve is always constant regardless of the number of data points. In order to increase the flexibility of the curve, more points must be provided, thus creating more spline segments which are still of cubic order.
- The control of the curve is through the change of the positions of data points or the end slope change. The global control characteristics is not intuitive.

Bezier Curve



- P. Bezier of the French automobile company of Renault first introduced the Bezier curve.
- A system for designing sculptured surfaces of automobile bodies (based on the Bezier curve)

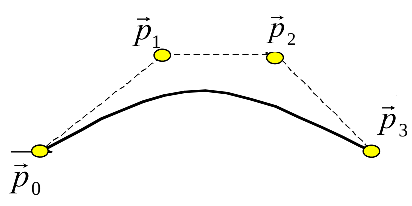
- passes $\overrightarrow{p_0}$ and $\overrightarrow{p_n}$, the two end points.

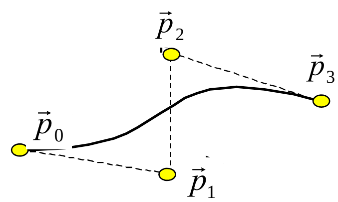
- has end point derivatives:

$$\boxed{\vec{p}_0\,' = n(\vec{p}_1 - \vec{p}_0) \qquad \vec{p}_n\,' = n(\vec{p}_n - \vec{p}_{n-1})}$$

- uses a vector of control points, representing the n+1 vertices of a "characteristic polygon".

Math Expression





$$\boxed{\overrightarrow{p}(u) = \sum_{i=0}^{n} \overrightarrow{p_i} B_{i,n}(u)}$$

n — segment(each polygon)

n+1 — vertices (each polygon) and number of control points

$u \in [0, \quad 1]$

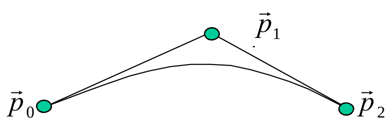
Bernstein Polynomial

$$B_{i,n}(u) = \frac{n!}{i!(n-i)\,!} u^i (1-u)^{n-i}$$

$B_{i,n}(u)$ is a function of the number of curve segments, n, and i.

i	0	1	2
$\frac{n!}{i!(n-i)!}$	$\frac{2!}{0!\ \ 2!} = 1$	$\frac{2!}{1!\ \ 1!} = 2$	$\frac{2!}{2!\ \ 0!} = 1$

An Example: If $n = 2$,
then $n+1 = 3$ vertices



i	0	1	2
$\frac{n!}{i!(n-i)!}$	$\frac{2!}{0!\ \ 2!} = 1$	$\frac{2!}{1!\ \ 1!} = 2$	$\frac{2!}{2!\ \ 0!} = 1$

$$\overrightarrow{p}(u) = \sum_{i=0}^{n} \overrightarrow{p_i} B_{i,n}(u)$$

$$\boxed{B_{i,n}(u) = \frac{n!}{i!(n-i)\,!} u^i (1-u)^{n-i}}$$

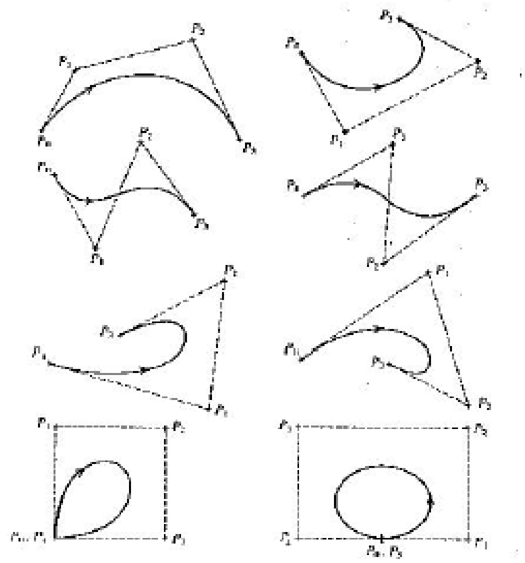
$$\boxed{\vec{p}(u) = 1 \times (1-u)^2 \vec{p}_0 + 2 \times u(1-u)\vec{p}_1 + 1 \times u^2 \vec{p}_2}$$

$$\vec{p}\,'(u) = -2(1-u)\vec{p}_0 + 2(1-2u)\vec{p}_1 + 2u\vec{p}_2$$

$$\vec{p}(0) = \vec{p}_0 \qquad \vec{p}\,'(0) = 2(\vec{p}_1 - \vec{p}_0)$$

$$\vec{p}(1) = \vec{p}_2 \qquad \vec{p}\,'(1) = 2(\vec{p}_2 - \vec{p}_1)$$

The order of Bezier curve is a function of the number of control points. Four control points (n=3) always produce a cubic Bezier curve.



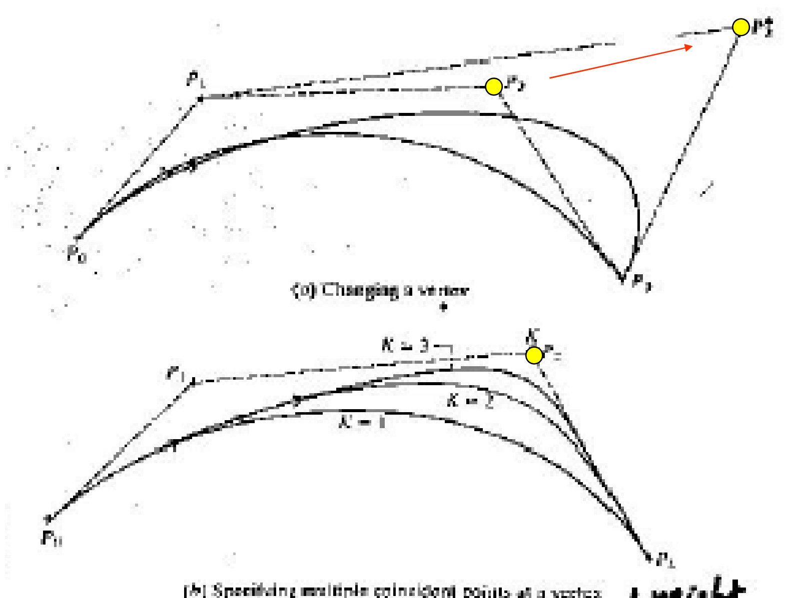
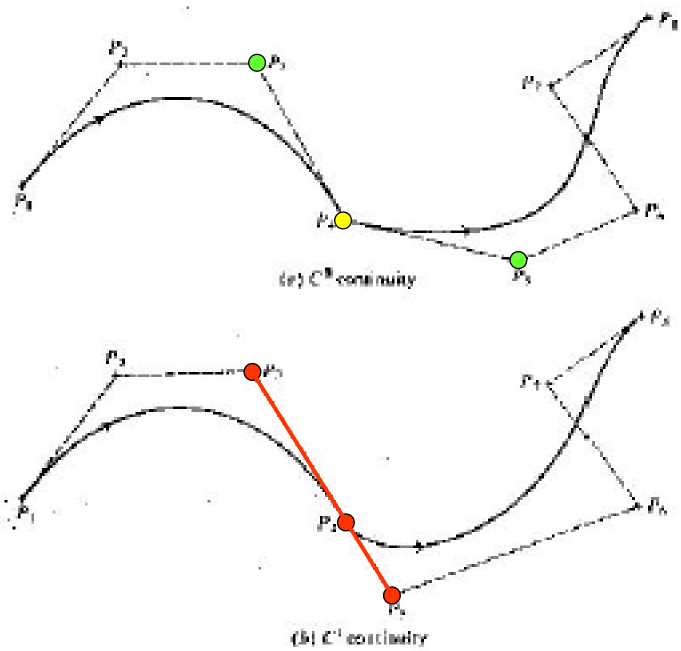


FIGURE 5-46
Modifications of cubic Bezier curve.



(a) C^0 continuity

(b) C^1 continuity

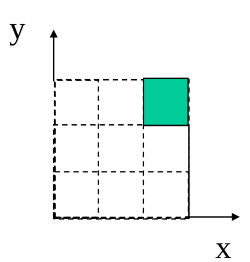
An Example

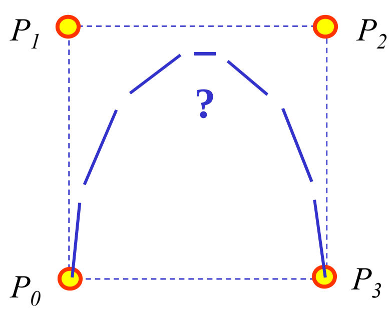
The coordinates of four control points relative to a current WCS are given by

$$P_0 = [2 \quad 2 \quad 0]^T, P_1 = [2 \quad 3 \quad 0]^T, P_2 = [3 \quad 3 \quad 0]^T, \& P_3 = [3 \quad 2 \quad 0]^T$$

Find the equation of the resulting Bezier curve. Also find points on curve for

$u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \& 1$





Solution

$$P(u) = P_0 B_{0,3} + P_1 B_{1,3} + P_2 B_{2,3} + P_3 B_{2,3} \qquad 0 \le u \le 1$$

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

e.g.

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

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

...

$$P(u) = P_0 (1-u)^3 + 3P_1 u (1-u)^2 + 3P_2 u^2 (1-u) + P_3 u^3 \qquad 0 \le u \le 1$$

Substituting the u values into his equation gives

u = 0, ¼, ½, ¾, 1

$$P(0) = P_0 = [2 \quad 2 \quad 0]^T$$

$$P\left(\frac{1}{4}\right) = \frac{27}{64} P_0 + \frac{27}{64} P_1 + \frac{9}{64} P_2 + \frac{1}{64} P_3 = [2.156 \quad 2.563 \quad 0]^T$$

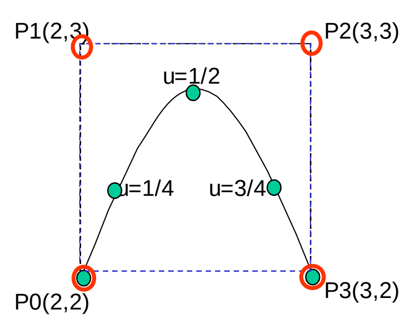
$$P\left(\frac{1}{2}\right) = \frac{1}{8} P_0 + \frac{3}{8} P_1 + \frac{3}{8} P_2 + \frac{1}{8} P_3 = [2.5 \quad 2.75 \quad 0]^T$$

$$P\left(\frac{3}{4}\right) = \frac{1}{64} P_0 + \frac{9}{64} P_1 + \frac{27}{64} P_2 + \frac{27}{64} P_3 = [2.844 \quad 2.563 \quad 0]^T$$

$$P(1) = P_3 = [3 \quad 2 \quad 0]^T$$

- control points, P_1, P_2, P_3, & P_4,

- points on curve, $P(u)$

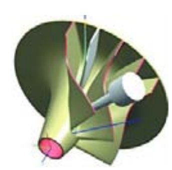


Improvements of Bezier Curve Over the Cubic Spline

- The shape of Bezier curve is controlled only by its defining points (control points). First derivatives are not used in the curve development as in the cubic spline.
- The order or the degree of the Bezier curve is variable and is related to the number of points defining it; *n+1* points define a *n*th degree curve.

  This is not the case for cubic splines where the degree is always cubic for a spline segment.
- The Bezier curve is smoother than the cubic splines because it has higher-order derivatives.

B-Spline



- A Generalization from Bezier Curve
- Better local control
- Degree of resulting curve is independent to the number of control points.

Math Representation

$$\overline{P}(u) = \sum_{i=0}^{n} \overline{P}_i \times N_{i,k}(u) \qquad 0 \le u \le u_{\max}$$

(k-1) degree of polynomial with (n+1) control points

- $\overline{P}_0, \overline{P}_1, \ldots, \overline{P}_n$ — n+1 control points.
- $N_{i,k}(u)$ — B-spline function (to be calculated in a recursive form)

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

Parametric Knots

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

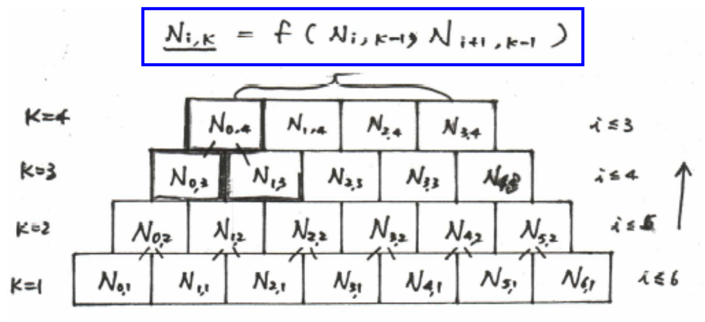
u_j : parametric knots (or knot values), for an open curve B-spline:

$$u_j = \begin{cases} 0 & j < k \\ j - k + 1 & k \le j \le n \\ n - k + 2 & j > n \end{cases}$$

where, $0 \le j \le n+k$, thus if a curve with (k-1) degree and (n+1) control points is to be developed, (n+k+1) knots then are required with $0 \le u \le u_{max} = n - k + 2$

Knot Value Calculation

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



$n = 3$; 4 control points
$k = 4$; 4-1=3 cubic polynomial
$0 \leq j \leq n+k = 7$

n increases – wider base
k increases – wider & taller

Calculation of $N_{i,k}(u)$ Function

$$\overline{P}(u) = \sum_{i=0}^{n} \overline{P}_i \times N_{i,k}(u) \qquad 0 \leq u \leq u_{\max}$$

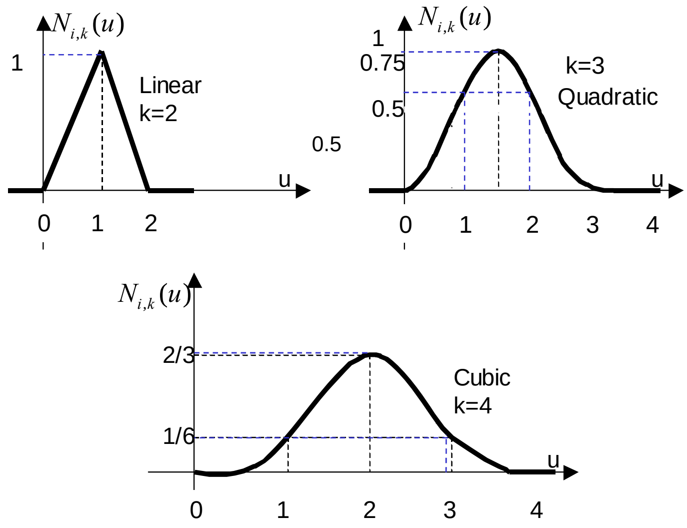
$$N_{i,k}(u) = (u - u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1} - u_i} + (u_{i+k} - u)\frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}}$$

$$N_{i,1} = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 1 & u = u_{\max} \ and \ u \leq u_{i+1} \ and \ u - u_i = 1 \\ 0 & otherwise \end{cases}$$

$$\boxed{\frac{0}{0} = 0}$$

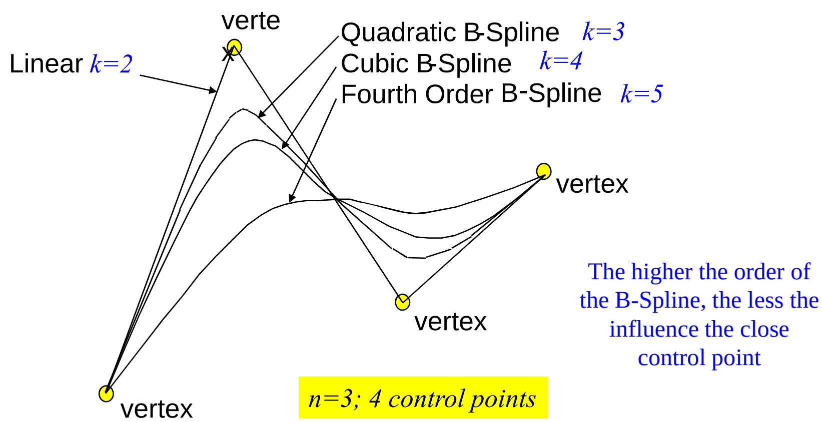
Role of B-Spline Function

(Calculating Weights $N_{i,k}(u)$ from Nearby Control Points)



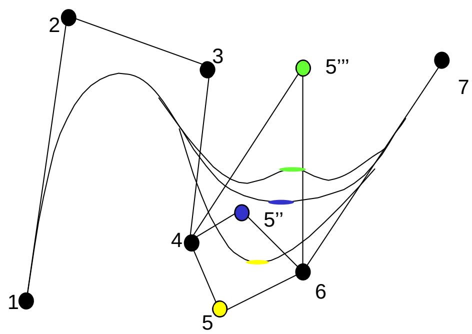
Properties of B-Spline

- Number of control points independent of degree of polynomial



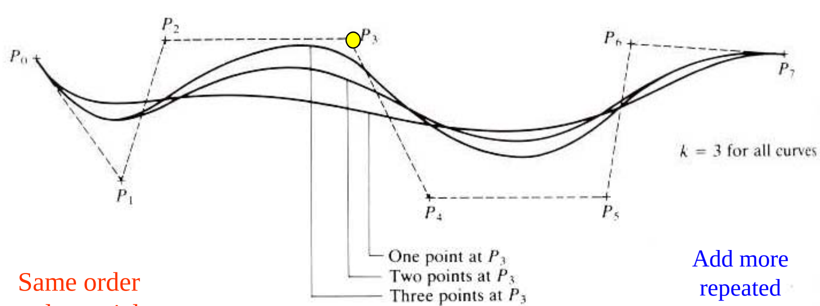
Properties of B-Spline

- B-spline allows better local control. Shape of the curve can be adjusted by moving the control points.
- Local control: a control point only influences *k* segments.



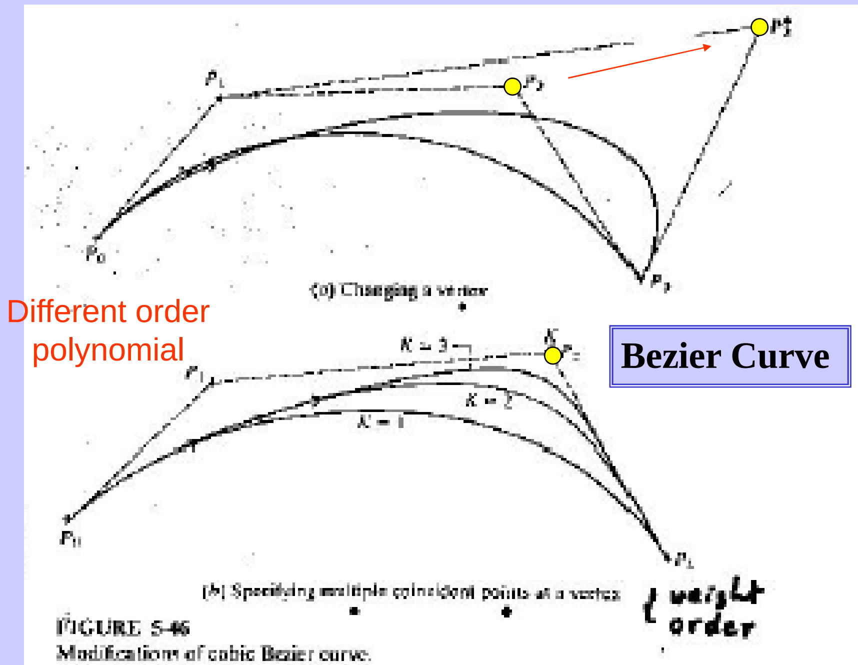
Properties of B-Spline

Repeated values of a control point can pull a B-spline curve forward to vertex. ("Interactive curve control")



Same order polynomial

Add more repeated control points to pull the curve



An Example

Find the equation of a cubic B-spline curve defined by the same control points as in the last example.

How does the curve compare with the Bezier curve?

Example 5.19. The coordinates of four control points relative to a current WCS are given by

$$\mathbf{P}_0 = [2 \quad 2 \quad 0]^T, \quad \mathbf{P}_1 = [2 \quad 3 \quad 0]^T, \quad \mathbf{P}_2 = [3 \quad 3 \quad 0]^T, \quad \text{and} \quad \mathbf{P}_3 = [3 \quad 2 \quad 0]^T$$

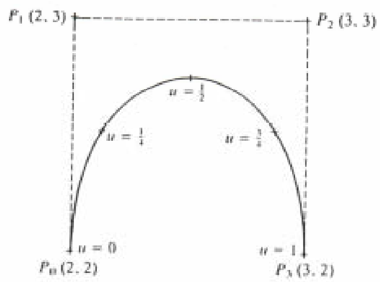
Find the equation of the resulting Bezier curve. Also find points on the curve for $u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$, and 1.

Solution. Equation (5.91) gives

$$\boxed{\mathbf{P}(u) = \mathbf{P}_0 B_{0,3} + \mathbf{P}_1 B_{1,3} + \mathbf{P}_2 B_{2,3} + \mathbf{P}_3 B_{3,3}, \qquad 0 \le u \le 1}$$

Using Eqs. (5.92) and (5.93), the above equation becomes

$$\boxed{\mathbf{P}(u) = \mathbf{P}_0(1-u)^3 + 3\mathbf{P}_1 u(1-u)^2 + 3\mathbf{P}_2 u^2(1-u) + \mathbf{P}_3 u^3, \qquad 0 \le u \le 1}$$



Example Problem for Finding the Bezier Curve

FIGURE 5-49
Bezier curve and generated points.

Example Problem for Finding the Bezier Curve

$$\mathbf{P}(u) = \mathbf{P}_0(1-u)^3 + 3\mathbf{P}_1 u(1-u)^2 + 3\mathbf{P}_2 u^2(1-u) + \mathbf{P}_3 u^3, \qquad 0 \le u \le 1$$

$$\mathbf{P}_0 = [2 \quad 2 \quad 0]^T, \quad \mathbf{P}_1 = [2 \quad 3 \quad 0]^T, \quad \mathbf{P}_2 = [3 \quad 3 \quad 0]^T, \quad \text{and} \quad \mathbf{P}_3 = [3 \quad 2 \quad 0]^T$$

Substituting the u values into this equation gives

$$\mathbf{P}(0) = \mathbf{P}_0 = [2 \quad 2 \quad 0]^T$$

$$\mathbf{P}\left(\frac{1}{4}\right) = \frac{27}{64}\mathbf{P}_0 + \frac{27}{64}\mathbf{P}_1 + \frac{9}{64}\mathbf{P}_2 + \frac{1}{64}\mathbf{P}_3 = [2.156 \quad 2.563 \quad 0]^T$$

$$\mathbf{P}\left(\frac{1}{2}\right) = \frac{1}{8}\mathbf{P}_0 + \frac{3}{8}\mathbf{P}_1 + \frac{3}{8}\mathbf{P}_2 + \frac{1}{8}\mathbf{P}_3 = [2.5 \quad 2.75 \quad 0]^T$$

$$\mathbf{P}\left(\frac{3}{4}\right) = \frac{1}{64}\mathbf{P}_0 + \frac{9}{64}\mathbf{P}_1 + \frac{27}{64}\mathbf{P}_2 + \frac{27}{64}\mathbf{P}_3 = [2.844 \quad 2.563 \quad 0]^T$$

$$\mathbf{P}(1) = \mathbf{P}_3 = [3 \quad 2 \quad 0]^T$$

Observe that $\sum_{i=0}^{3} B_{i,3}$ is always equal to unity for any u value. Figure 5-49 shows the curve and the points.

Finding the B-Spline Curve for the Same Example Problem

Example 5.21. Find the equation of a cubic B-spline curve defined by the same control points as in Example 5.19. How does the curve compare with the Bezier curve?

① cubic curve : $k=4$, four points : $n=3$ $(0,1,2,3)$

$$\boxed{u_j = \begin{cases} 0 \\ j-k+1 \\ n-k+2 \end{cases} = \begin{cases} 0 & j<k=4 \\ j-3 & 4\le j\le 3 \\ 1 & j>3 \end{cases}}$$

$u_{max} = n-k+2 = 1$

$0 \le u \le 1$

$0 \le j \le n+k = 7$

② $\boxed{\bar{P}(u) = \sum_{0}^{n} \bar{P}_i \cdot N_{ik}(u)}$

$$\boxed{\begin{aligned} N_{i,1} &= \begin{cases} 1 & u_i < u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,k} &= (u-u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1}-u_i} + (u_{i+k}-u)\frac{N_{i+1,k-1}(u)}{u_{i+k}-u_{i+1}} \end{aligned}}$$

Solution. This cubic spline has $k = 4$ and $n = 3$. Eight knots are needed to calculate the B-spline functions. Equation (5.106) gives the knot vector

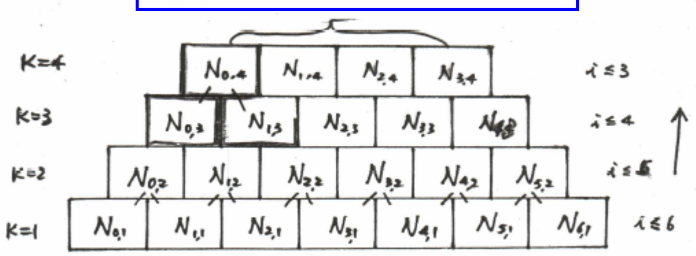
Values to be Calculated

① $\boxed{[u_0 \quad u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7]}$

The range of u [Eq. (5.108)] is $0 \le u \le 1$. Equation (5.103) gives

② $\mathrm{P}(u) = \bar{\mathrm{P}}_0 N_{0,4} + \bar{\mathrm{P}}_1 N_{1,4} + \bar{\mathrm{P}}_2 N_{2,4} + \bar{\mathrm{P}}_3 N_{3,4}, \qquad 0 \le u \le 1$

$$\boxed{N_{i,k} = f(N_{i,k-1}, N_{i+1,k-1})}$$



$n = 3$; 4 control points
$k = 4$; 4-1=3 cubic polynomial
u_j: $0 \le j \le n+k = 7$

n increases – wider base
k increases – wider & taller

Calculating the Knots, u_j

cubic curve: $k=4$, four points: $n=3$ $(0, 1, 2, 3)$

$$u_j = \begin{cases} 0 \\ j-k+1 \\ n-k+2 \end{cases} = \begin{cases} 0 & j<k=4 \\ j-3 & 4\le j\le 3 \\ 1 & j>3 \end{cases}$$

$u_{max} = n-k+2 = 1$

$0 \le u \le 1$

$0 \le j \le n+k = 7$

$[u_0 \ u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6 \ u_7]$ as $[0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]$

Calculating $N_{i,1}$

$$N_{i,1} = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 1 & u = u_{max} \ and \ u \le u_{i+1} \ and \ u - u_i = 1 \\ 0 & otherwise \end{cases}$$

$$\frac{0}{0} = 0$$

① $N_{i,k}$. $(k=1)$ knots $\begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

△ $i = 0, 1, 2$ $\quad u_i = 0 \le u \le u_{i+1} = 0$ (or $u = 0$)

$$N_{0,1} = N_{1,1} = N_{2,1} = \begin{cases} 1 & u=0 \\ 0 & elsewhere \end{cases}$$

△ $i = 3$ $\quad u_i = 0 \le u \le u_{i+1} = 1$ (or $0 \le u \le 1$)

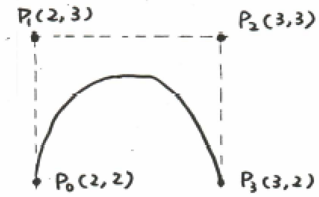
$$N_{3,1} = \begin{cases} 1, & 0 \le u \le 1 \\ 0, & otherwise \end{cases}$$

△ $i = 4, 5, 6$ $\quad u_i = 1 \le u \le u_{i+1} = 1$ (or $u = 1$)

$$N_{4,1} = N_{5,1} = N_{6,1} = \begin{cases} 1 & u=1 \\ 0 & elsewhere \end{cases}$$

Calculating $N_{i,k}$

To calculate the above B-spline functions, use Eqs. (5.104) and (5.105) together with the knot vector as follows:



k=1
i+k≤7
(6)

$$N_{0,1} = N_{1,1} = N_{2,1} = \begin{cases} 1, & u = 0 \\ 0, & \text{elsewhere} \end{cases}$$

$$N_{3,1} = \begin{cases} 1, & 0 \le u \le 1 \\ 0, & \text{elsewhere} \end{cases}$$

$$N_{4,1} = N_{5,1} = N_{6,1} = \begin{cases} 1, & u = 1 \\ 0, & \text{elsewhere} \end{cases}$$

k=2
i+k≤7
(5)

$$N_{0,2} = (u - u_0)\frac{N_{0,1}}{u_1 - u_0} + (u_2 - u)\frac{N_{1,1}}{u_2 - u_1} = \frac{uN_{0,1}}{0} + \frac{(-u)N_{1,1}}{0} = 0$$

$$N_{1,2} = (u - u_1)\frac{N_{1,1}}{u_2 - u_1} + (u_3 - u)\frac{N_{2,1}}{u_3 - u_2} = \frac{uN_{1,1}}{0} + \frac{(-u)N_{2,1}}{0} = 0$$

$$N_{2,2} = (u - u_2)\frac{N_{2,1}}{u_3 - u_2} + (u_4 - u)\frac{N_{3,1}}{u_4 - u_3} = \frac{uN_{2,1}}{0} + \frac{(1-u)N_{3,1}}{1} = (1-u)N_{3,1}$$

$$N_{3,2} = (u - u_3)\frac{N_{3,1}}{u_4 - u_3} + (u_5 - u)\frac{N_{4,1}}{u_5 - u_4} = uN_{3,1} + \frac{(1-u)N_{4,1}}{0} = uN_{3,1}$$

$$N_{4,2} = (u - u_4)\frac{N_{4,1}}{u_5 - u_4} + (u_6 - u)\frac{N_{5,1}}{u_6 - u_5} = (u-1)\frac{N_{4,1}}{0} + \frac{(1-u)N_{5,1}}{0} = 0$$

$$N_{5,2} = (u - u_5)\frac{N_{5,1}}{u_6 - u_5} + (u_7 - u)\frac{N_{6,1}}{u_7 - u_6} = \frac{(u-1)N_{5,1}}{0} + \frac{(1-u)N_{6,1}}{0} = 0$$

Calculating $N_{i,k}$

k=3
i+k≤7
(4)

$$N_{0,3} = (u - u_0)\frac{N_{0,2}}{u_2 - u_0} + (u_3 - u)\frac{N_{1,2}}{u_3 - u_1} = u\frac{0}{0} + (-u)\frac{0}{0} = 0$$

$$N_{1,3} = (u - u_1)\frac{N_{1,2}}{u_3 - u_1} + (u_4 - u)\frac{N_{2,2}}{u_4 - u_2} = u\frac{N_{1,2}}{0} + \frac{(1-u)N_{2,2}}{1} = (1-u)^2 N_{3,1}$$

$$N_{2,3} = (u - u_2)\frac{N_{2,2}}{u_4 - u_2} + (u_5 - u)\frac{N_{3,2}}{u_5 - u_3} = uN_{2,2} + (1-u)N_{3,2} = 2u(1-u)N_{3,1}$$

$$N_{3,3} = (u - u_3)\frac{N_{3,2}}{u_5 - u_3} + (u_6 - u)\frac{N_{4,2}}{u_6 - u_4} = u^2 N_{3,1} + (1-u)\frac{N_{4,2}}{0} = u^2 N_{3,1}$$

$$N_{4,3} = (u - u_4)\frac{N_{4,2}}{u_6 - u_4} + (u_7 - u)\frac{N_{5,2}}{u_7 - u_5} = (u-1)\frac{N_{4,2}}{0} + (1-u)\frac{N_{5,2}}{0} = 0$$

k=4
i+k≤7
(3)

$$N_{0,4} = (u - u_0)\frac{N_{0,3}}{u_3 - u_0} + (u_4 - u)\frac{N_{1,3}}{u_4 - u_1} = (1-u)^3 N_{3,1}$$

$$N_{1,4} = (u - u_1)\frac{N_{1,3}}{u_4 - u_1} + (u_5 - u)\frac{N_{2,3}}{u_5 - u_2} = 3u(1-u)^2 N_{3,1}$$

$$N_{2,4} = (u - u_2)\frac{N_{2,3}}{u_5 - u_2} + (u_6 - u)\frac{N_{3,3}}{u_6 - u_3} = 3u^2(1-u)N_{3,1}$$

$$N_{3,4} = (u - u_3)\frac{N_{3,3}}{u_6 - u_3} + (u_7 - u)\frac{N_{4,3}}{u_7 - u_4} = u^3 N_{3,1}$$

Result

Substituting $N_{i,4}$ into Eq. (5.115) gives

$$\boxed{\mathbf{P}(u) = [\mathbf{P}_0(1-u)^3 + 3\mathbf{P}_1 u(1-u)^2 + 3\mathbf{P}_2 u^2(1-u) + \mathbf{P}_3 u^3]N_{3,1}, \qquad 0 \le u \le 1}$$

Substituting $N_{3,1}$ into this equation gives the curve equation as

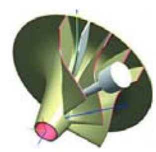
$$\boxed{\mathbf{P}(u) = \mathbf{P}_0(1-u)^3 + 3\mathbf{P}_1 u(1-u)^2 + 3\mathbf{P}_2 u^2(1-u) + \mathbf{P}_3 u^3, \qquad 0 \le u \le 1}$$

This equation is the same as the one for the Bezier curve in Example 5.19. Thus the cubic B-spline curve defined by four control points is identical to the cubic Bezier curve defined by the same points. This fact can be generalized for a $(k-1)$-degree curve as mentioned earlier.

n + 1 control points: 3+1=4
k – 1 degree curve: 4-1=3

4 control points – cubic polynomial

Non-uniform Rational B-Spline Curve (NURBS)



Rational B-Spline

$$\boxed{\overline{P}(u) = \sum_{i=0}^{n} \overline{P}_i \times R_{i,k}(u) \qquad 0 \le u \le u_{\max}}$$

$$R_{i,k} = \frac{h_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)} \qquad (h_i - scalar)$$

If $h_i = 1$, then $R_{i,k}(u) = N_{i,k}(u)$,it is the representation of a B-Spline curve.

Industry Standard Today!

What is NURBS ?

rational and B-spline.

$$\bar{P}(u) = \sum_{i=0}^{n} \bar{P}_i R_{i,k}(u) \qquad 0 \le u \le u_{max}$$

Basis fun.

$$R_{i,k}(u) = \frac{h_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)} \qquad (h_i \text{ - scalar,}$$

a homogeneous coordinate vector $H = [h_0, h_1, \cdots h_n]^T$ is introduced and the basis function is defined by the algebraic ratio of two polynomials.

The curves are defined in the homogeneous space using homogeneous coordinates $(x^*, y^*, z^*, h)^T$.

If $h_i = 1$, $R_{i,k}(u) = N_{i,k}(u)$

Development of NURBS

- Boeing: Tiger System in 1979
- SDRC: Geomod in 1993
- University of Utah: Alpha-1 in 1981
- Industry Standard: IGES, PHIGS, PDES, Pro/E, etc.

Advantages of NURBS

- Serve as a genuine generalizations of non-rational B-spline forms as well as rational and non-rational Bezier curves and surfaces
- Offer a common mathematical form for representing both standard analytic shapes (conics, quadratics, surface of revolution, etc) and free-from curves and surfaces precisely. B-splines can only approximate conic curves.
- Provide the flexibility to design a large variety of shapes by using control points and weights. increasing the weights has the effect of drawing a curve toward the control point.
- Have a powerful tool kit (knot insertion/refinement/removal, degree elevation, splitting, etc.
- Invariant under scaling, rotation, translation, and projections.
- Reasonably fast and computationally stable.
- Clear geometric interpretations

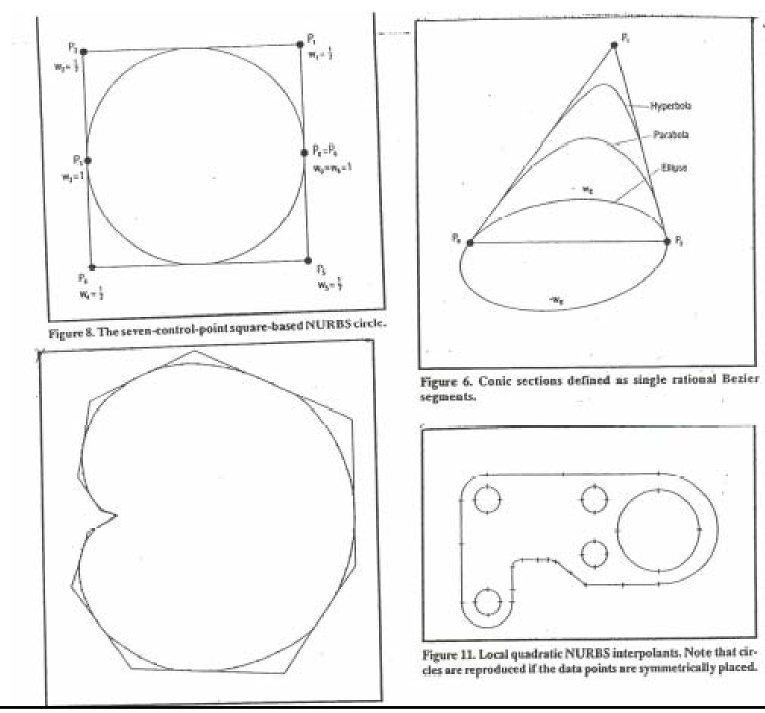
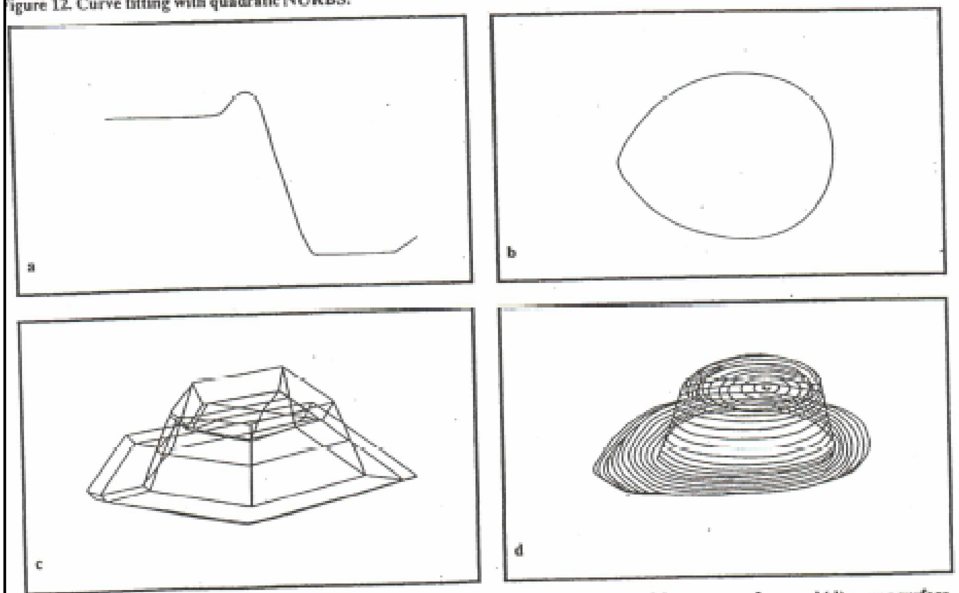


Figure 8. The seven-control-point square-based NURBS circle.

Figure 6. Conic sections defined as single rational Bezier segments.

Figure 11. Local quadratic NURBS interpolants. Note that circles are reproduced if the data points are symmetrically placed.

igure 12. Curve fitting with quadratic NURBS.



Quick Questions

n + 1 – number of control points; & *k* – degree of the curve (polynomial)

- For B-spline, *k* should always be less than or equal to *n+1*. T or F? why?
- What are the two major advantages of B-spline over Bezier curve in real design?
- If *k* increases in its reasonable range, will the corresponding B-spline curve move closer to the control polygon.?
- What are the required user inputs to construct a Hermite Cubic, Bezier, B-spline and NURBS curve segment?
- By adding more control points in a small region when defining a Bezier curve, one can manipulate the curvature of the curve without affecting other curve sections. T or F.?
- B-spline curves are identical to Bezier curves when *k=n+1*. T. or F?