Representation and Manipulation of Curves

How does the computer store
curve information?

Types of Curve Equations

- Implicit non-parametric
$$x^2 + y^2 - R^2 = 0, \quad z = 0$$
- Explicit non-parametric
$$y = \pm\sqrt{R^2 - x^2}, \quad z = 0$$
- Parametric
$$x = R\cos\theta, \quad y = R\sin\theta, \quad z = 0 \quad (0 \le \theta \le 2\pi)$$

To evaluate points on a curve at given intervals, the parametric representation is used.

For calculating intersections, a combination of parametric and non-parametric is used.

Straight Lines

- An explicit non-parametric form:
  - 2D: $y = mx + b$
  - 3D: $y = mx + b; \quad z = nx + c;$
- An implicit non-parametric form:
  - 2D: $ax + by + c = 0$
  - 3D: $a_1x + b_1y + c_1z + d_1 = 0; \quad a_2x + b_2y + c_2z + d_2 = 0$
- Parametric form:
  - 2D: $x = x_0 + ut; \quad y = y_0 + vt$
  - 3D: $x = x_0 + ut; \quad y = y_0 + vt; \quad z = z_0 + wt$

A line segment is established by specifying range for t (e.g., $0 \le t \le 1$).

Conic Sections

- Circle: $x = R\cos\theta$, $y = R\sin\theta \quad z = 0$
- Ellipse: $x = a\cos\theta$, $y = b\sin\theta \quad z = 0$
- Hyperbola: $x = a\cosh u$, $y = b\sinh u \quad z = 0$
- Parabola: $x = cu^2$, $y = u \quad z = 0$

A conic arc is established by specifying range for θ (e.g., $0 \le \theta \le \pi/2$).

Circle in 3D space

- What is the equation for the upright circle shown?

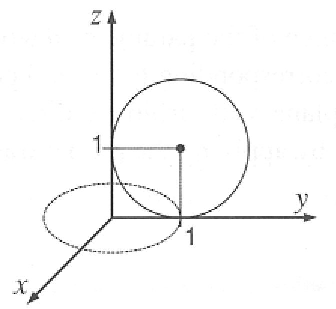


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

- To position a conic in 3D space we must use transformation matrices.

Circle in 3D space

$$[x' \quad y' \quad z' \quad 1]^T = Trans(0,1,1)\,RotY(-90°)[x \quad y \quad z \quad 1]^T$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-90°) & 0 & -\sin(-90°) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-90°) & 0 & \cos(-90°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

$$x' = 0$$

$$y' = y + 1 = R\sin\theta + 1$$

$$z' = x + 1 = R\sin\theta + 1 \qquad (0 \le \theta \le 2\pi)$$

Freeform Curves – Hermite Curves

- The most common freeform curves in CAD are polynomial of order 3:

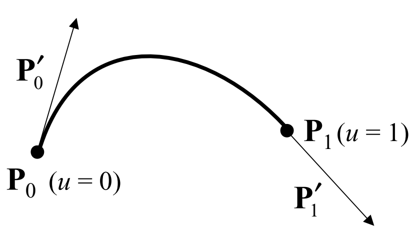
$$\mathbf{P}(u)=\begin{bmatrix}x(u)\\ y(u)\\ z(u)\end{bmatrix}=\begin{bmatrix}a_{0,0}+a_{1,0}u+a_{2,0}u^2+a_{3,0}u^3\\ a_{0,1}+a_{1,1}u+a_{2,1}u^2+a_{3,1}u^3\\ a_{0,2}+a_{1,2}u+a_{2,2}u^2+a_{3,2}u^3\end{bmatrix}$$

$$=\begin{bmatrix}a_{0,0}\\ a_{0,1}\\ a_{0,2}\end{bmatrix}+\begin{bmatrix}a_{1,0}\\ a_{1,1}\\ a_{1,2}\end{bmatrix}u+\begin{bmatrix}a_{2,0}\\ a_{2,1}\\ a_{2,2}\end{bmatrix}u^2+\begin{bmatrix}a_{3,0}\\ a_{3,1}\\ a_{3,2}\end{bmatrix}u^3$$

$$=\mathbf{a}_0+\mathbf{a}_1u+\mathbf{a}_2u^2+\mathbf{a}_3u^3 \qquad (0\le u\le 1)$$

Hermite Curves

- The coefficients $\boldsymbol{a}_0$, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ are hard for a designer to specify because the geometric affect is not intuitive. To solve this problem, consider the first point and last point and their derivatives with respect to *u*.



$$\mathbf{P}(u)=\mathbf{a}_0+\mathbf{a}_1u+\mathbf{a}_2u^2+\mathbf{a}_3u^3$$
$$\mathbf{P}(0)=\mathbf{a}_0=\mathbf{P}_0$$
$$\mathbf{P}(1)=\mathbf{a}_0+\mathbf{a}_1+\mathbf{a}_2+\mathbf{a}_3=\mathbf{P}_1$$

$$\mathbf{P}'(u)=\mathbf{a}_1+2\mathbf{a}_2u+3\mathbf{a}_3u^2$$
$$\mathbf{P}'(0)=\mathbf{a}_1=\mathbf{P}'_0$$
$$\mathbf{P}'(1)=\mathbf{a}_1+2\mathbf{a}_2+3\mathbf{a}_3=\mathbf{P}'_1$$

Hermite Curves

- Solving for ($\boldsymbol{a}_0$, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$) yields:

$$\begin{aligned}
\mathbf{a}_0 &= \mathbf{P}_0 \\
\mathbf{a}_1 &= \mathbf{P}_0' \\
\mathbf{a}_2 &= -3\mathbf{P}_0 + 3\mathbf{P}_1 - 2\mathbf{P}_0' - \mathbf{P}_1' \\
\mathbf{a}_3 &= 2\mathbf{P}_0 - 2\mathbf{P}_1 + \mathbf{P}_0' + \mathbf{P}_1'
\end{aligned}$$

- Substituting this into the first equation, yields:

$$\mathbf{P}(u) = [1 - 3u^2 + 2u^3 \quad 3u^2 - 2u^3 \quad u - 2u^2 + u^3 \quad -u^2 + u^3] \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_0' \\ \mathbf{P}_1' \end{bmatrix}$$

Hermite Curves

The relative importance of each vector $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_0$' and $\mathbf{P}_1$' is given by the "blending functions":

$$\begin{aligned}
f_1(u) &= 1 - 3u^2 + 2u^3 \\
f_2(u) &= 3u^2 - 2u^3 \\
f_3(u) &= u - 2u^2 - u^3 \\
f_4(u) &= -u^2 + u^3
\end{aligned}$$

The affect of the each vector is shown in the figure:

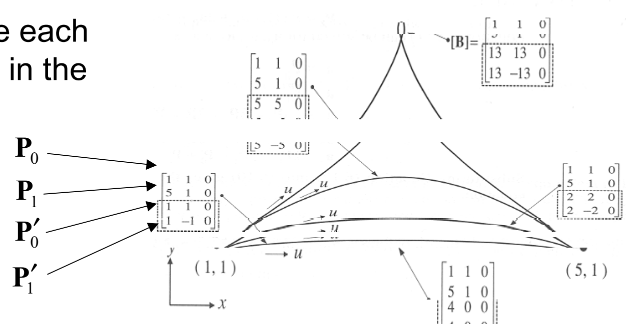
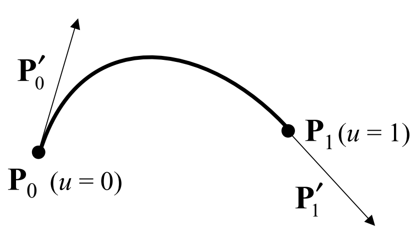


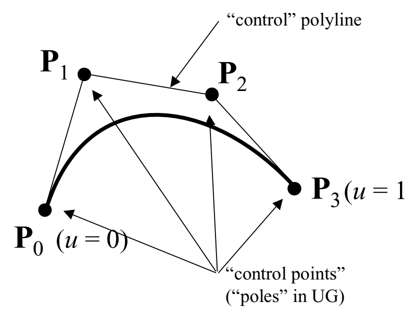
Figure is from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Bezier Curve

- This is the “Single” option for creating spline curves in UG.
- The Bezier curve uses “control points” instead of tangent vectors to control the shape of the curve.



Hermite Curve



Degree 3 Bezier Curve

Bezier Curve

The equation for a Bezier curve is:

$$\mathbf{P}(u) = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i} \mathbf{P}_i$$

Blending function

“Blending functions” $B_{i,n}(u)$ yield the influence of each control point $\boldsymbol{P}_i$ as a function of u:

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

where: $\binom{n}{i} = \frac{n!}{i!\,(n-i)!}$

MEEM4403 Computer-Aided Design Methods

Bezier Curve – 3rd Order

$$\mathbf{P}(u)=\sum_{i=0}^{n}\binom{n}{i}u^i(1-u)^{n-i}\mathbf{P}_i$$

$$\mathbf{P}(u)=\sum_{i=0}^{3}\binom{3}{i}u^i(1-u)^{3-i}\mathbf{P}_i$$

$$\begin{aligned}\mathbf{P}(u)=&1\cdot u^0(1-u)^3\mathbf{P}_0\\&+3\cdot u(1-u)^2\mathbf{P}_1\\&+3\cdot u^2(1-u)\mathbf{P}_2\\&+1\cdot u^3(1-u)^0\mathbf{P}_3\end{aligned}$$

$$\binom{3}{0}=\frac{3!}{0!\cdot 3!}=\frac{3\cdot 2\cdot 1}{1\cdot 3\cdot 2\cdot 1}=1$$

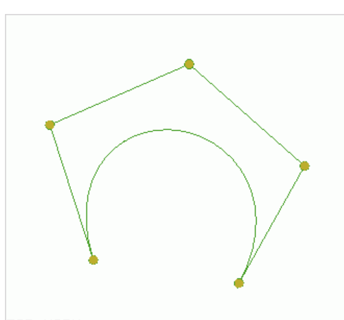
$$\binom{3}{1}=\frac{3!}{1!\cdot 2!}=\frac{3\cdot 2\cdot 1}{1\cdot 2\cdot 1}=3$$

$$\binom{3}{2}=\frac{3!}{2!\cdot 1!}=\frac{3\cdot 2\cdot 1}{2\cdot 1\cdot 1}=3$$

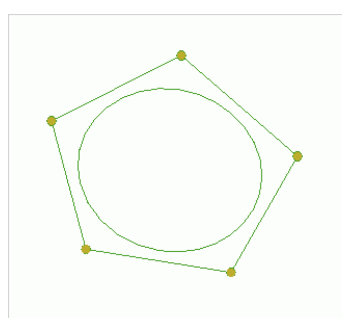
$$\binom{3}{3}=\frac{3!}{3!\cdot 0!}=\frac{3\cdot 2\cdot 1}{3\cdot 2\cdot 1\cdot 1}=1$$

Closed versus Open Curves

- A Bezier curve can be “open” or “closed”



“open”



“closed”

Bezier Curve Properties

- degree of curve is equal to number of control points - 1
- open curves always pass through first and last point.
- tangent at first point is given by the direction of the first segment of control polygon.
- the *n*th derivative of the curve at the first point is given by the first $n + 1$ control points.
- the same curve (and properties) exist when starting with the last control point.
- the curve is always inside the convex hull of the control polygon:

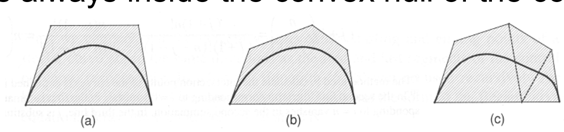
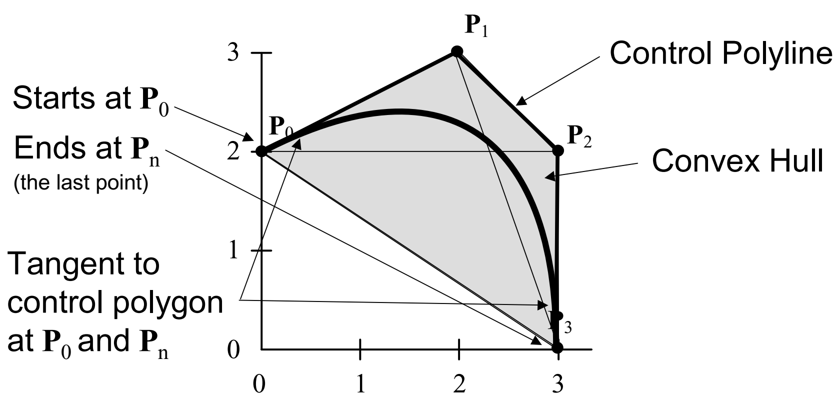


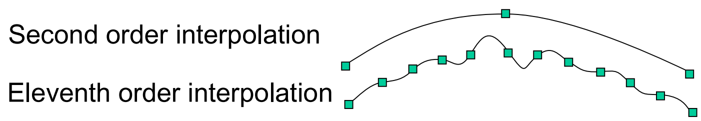
Figure is from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

1. For the cubic Bezier curve $\mathbf{P}(u)$ $[0 \leq u \leq 1]$ with the control points shown,
   - draw the convex hull,
   - draw the control polyline,
   - draw a rough sketch of the curve, and
   - calculate the values of the blending functions ($B_{i,n}$) for $u = 0.2$.
   - use the blending functions to calculate the position of the curve point $\mathbf{P}$ (0.2).



Limitations of Bezier curves

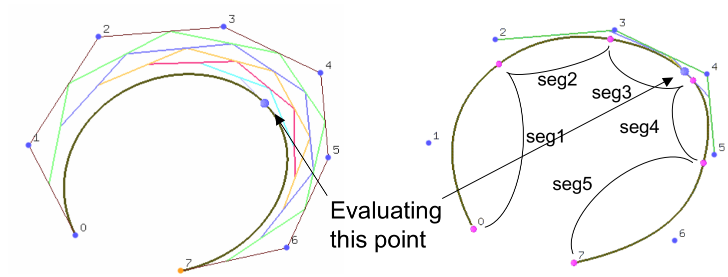
- More complicated shapes require higher order Bezier curves. However, higher order curves are inherently more wavy.



- Bezier curves can not be modified "locally." Movement of any control point will affect the whole curve.

Enter the B-Spline!

- Instead of considering EVERY control point when evaluating a point, only consider closest k.



Bezier (n = 7)   B-spline (n = 7, k = 4)

- The B-spline curve is composed of n - k + 2 segments, each of degree k-1.

B-spline Equation

- The B-spline equation is recursive. It is given by:

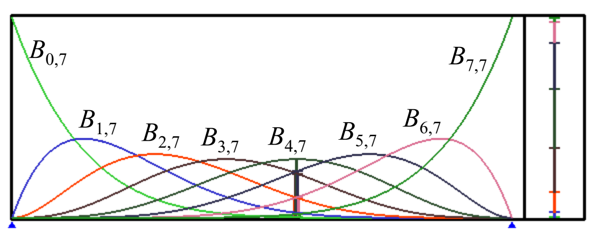
$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(u) \qquad (t_{k-1} \le u \le t_{n+1})$$

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

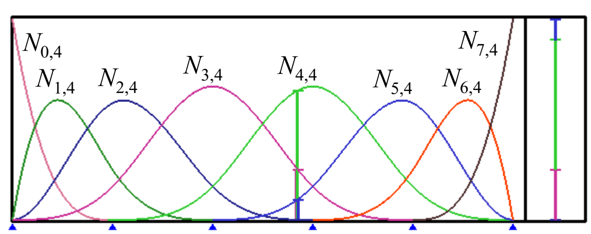
$$N_{i,1}(u) = \begin{cases} 1 & t_i \le u \le t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Compare Blending Functions

Bezier Blending Functions ($B_{i,n}$)

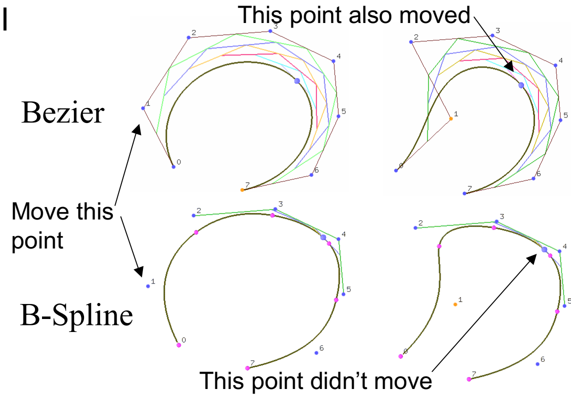


B-Spline Blending Functions ($N_{i,k}$)



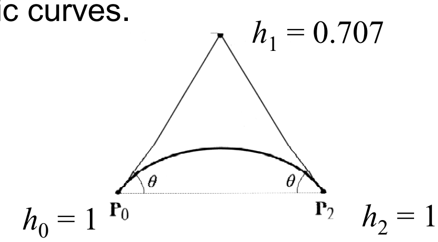
Properties of B-Splines

- composed of multiple connected polynomial curves
- each segment affected by *k* control points
- each control point affects *k* segments, max.
- Inside convex hull
- local modification
- symmetric



NURBS curves

- NURBS means "**Non-uniform Rational B-Spline**".
- Many CAD systems use this as the ONLY internal geometry representation (even for straight lines and circles).
- NURBS have a weighting factor h_i associated with each control point:
- NURBS curves are useful because they allow exact representation of conic curves.



NURBS curves

The NURBS equation is:

Regular B-Spline

$$\mathbf{P}(u) = \frac{\sum_{i=0}^{n} h_i \mathbf{P}_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)}$$

B-Spline equation applied to weighting.

NURBS

- NURBS curves are useful because they allow exact representation of conic curves.
- To create a circular arc (less than 180º) using a NURBS curve:
  - use k = 3 (degree = 2)
  - arrange control points in triangle with two equal angles as shown
  - use weightings: $h_0 = h_2 = 1$ and $h_1 = \cos\theta$