

Parametric Surfaces

CS 418

Intro to Computer Graphics

John C. Hart

Space Curves

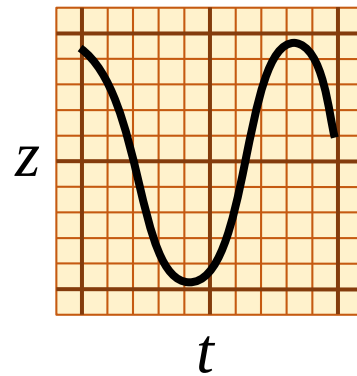
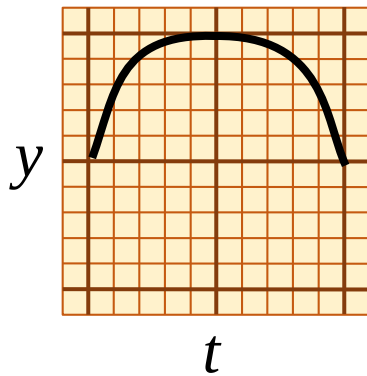
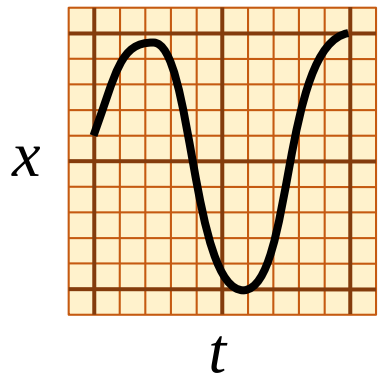
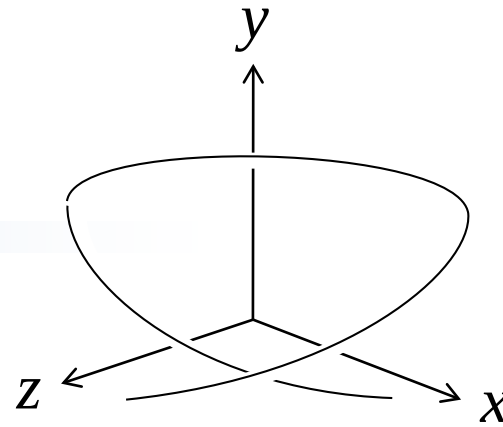
Separate into three coordinate functions

$$\mathbf{p}(t) = (x(t), y(t), z(t))$$

$$x(t) = (1-t)^3 x_0 + 3t(1-t)^2 x_1 + 3t^2(1-t) x_2 + t^3 x_3$$

$$y(t) = (1-t)^3 y_0 + 3t(1-t)^2 y_1 + 3t^2(1-t) y_2 + t^3 y_3$$

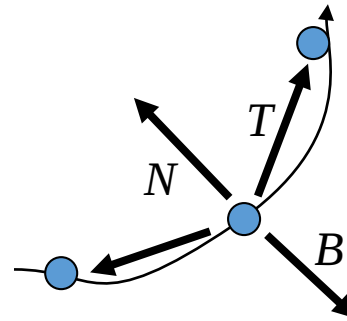
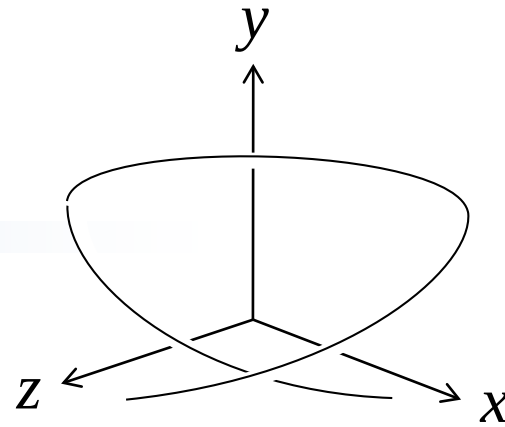
$$z(t) = (1-t)^3 z_0 + 3t(1-t)^2 z_1 + 3t^2(1-t) z_2 + t^3 z_3$$



Space Curves

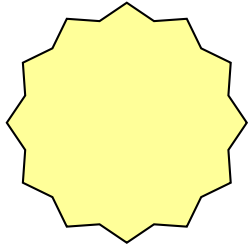
Make your own roller-coaster ride

- Camera position along space curve
- Look at point is next position along space curve (tangent)
- Binormal is cross product of vector to next position with vector to previous position
- Up direction (normal) is cross product of binormal with tangent

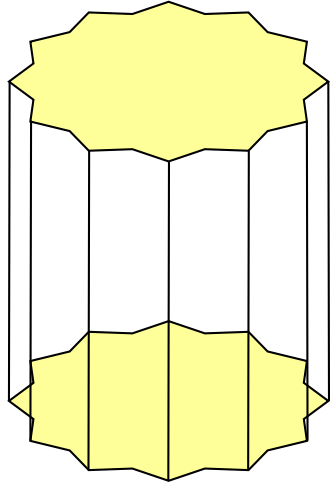


Extrusion

- Two 3-D copies of each 2-D curve



$$\mathbf{p}(t) = (x(t), y(t))$$

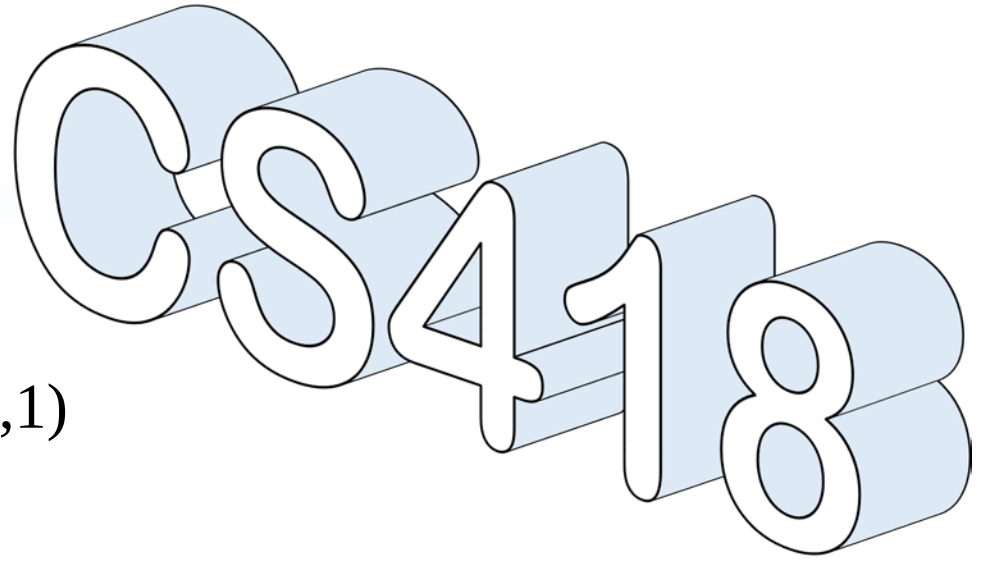


$$\mathbf{p}_1(t) = (x(t), y(t), 1)$$

$$\mathbf{p}_0(t) = (x(t), y(t), 0)$$

- Create a mesh of quads (or tri-strip)

$$\mathbf{p}_0(t), \mathbf{p}_1(t), \mathbf{p}_0(t+\Delta t), \mathbf{p}_1(t+\Delta t)$$



Generalized Cylinder

- Construct a 2-D profile curve

$$\mathbf{q}(s) = (a(s), b(s))$$

- Construct a space curve

$$\mathbf{p}(t) = (x(t), y(t), z(t))$$

- Construct a Frenet frame at each point along space curve

$$T(t) = \mathbf{p}(t+\Delta t) - \mathbf{p}(t)$$

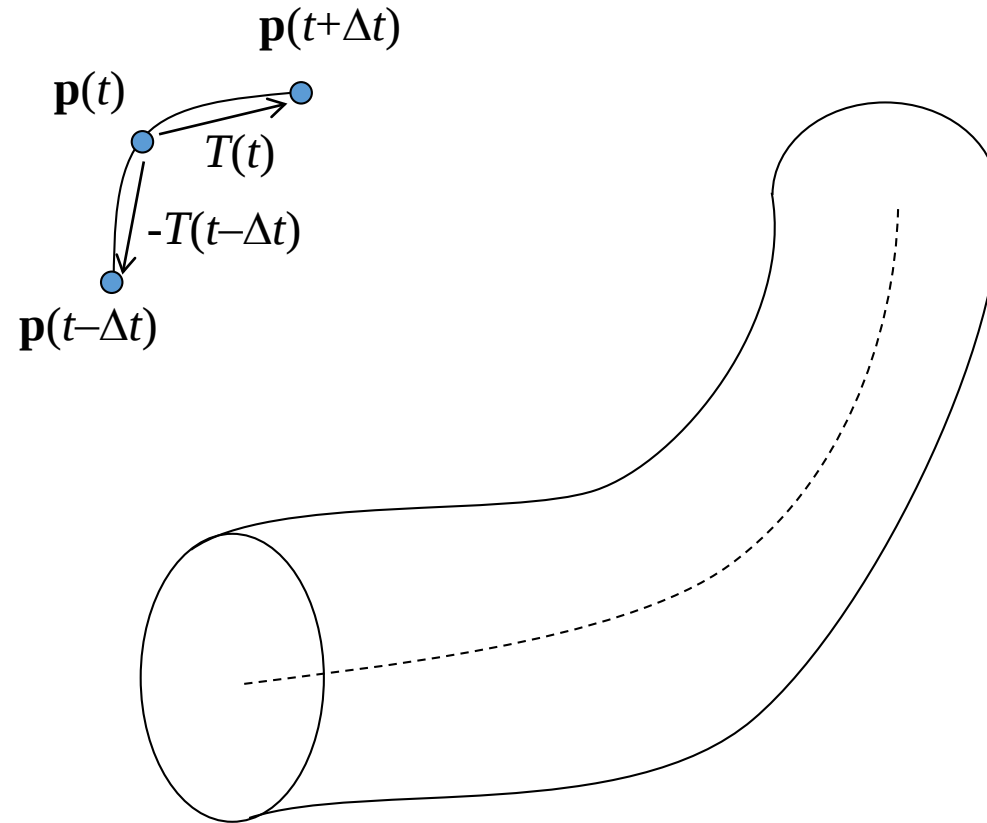
$$B(t) = T(t) \times -T(t - \Delta t)$$

$$N(t) = B(t) \times T(t)$$

(all normalized)

- Plot 2-D curve in (N, B) space

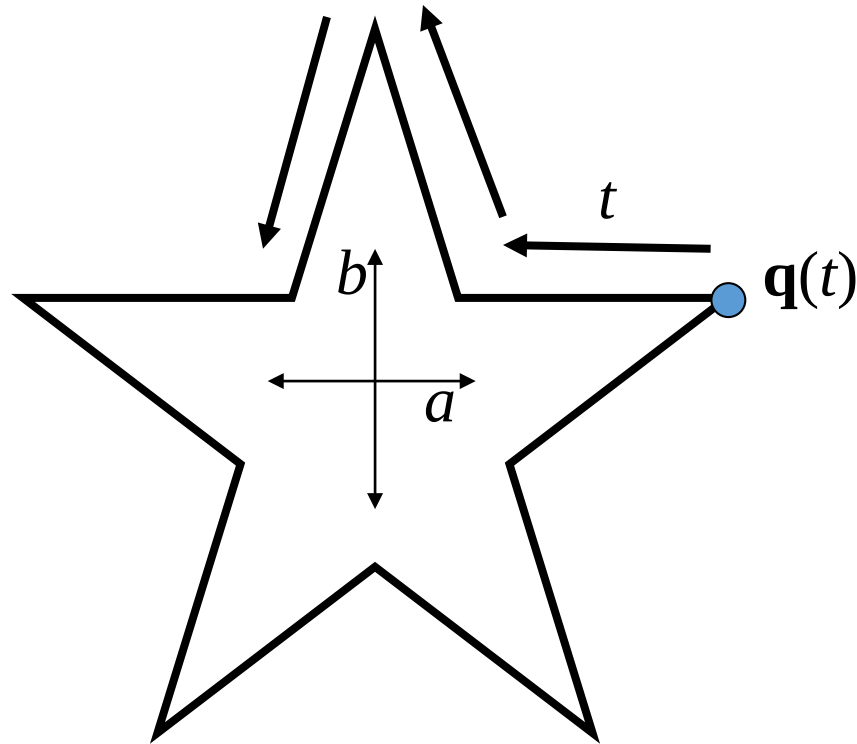
$$\mathbf{gc}(s, t) = \mathbf{p}(t) + a(s) N(t) + b(s) B(t)$$



Generalized Cylinder

- Construct a 2-D profile curve

$$\mathbf{q}(t) = (a(t), b(t))$$



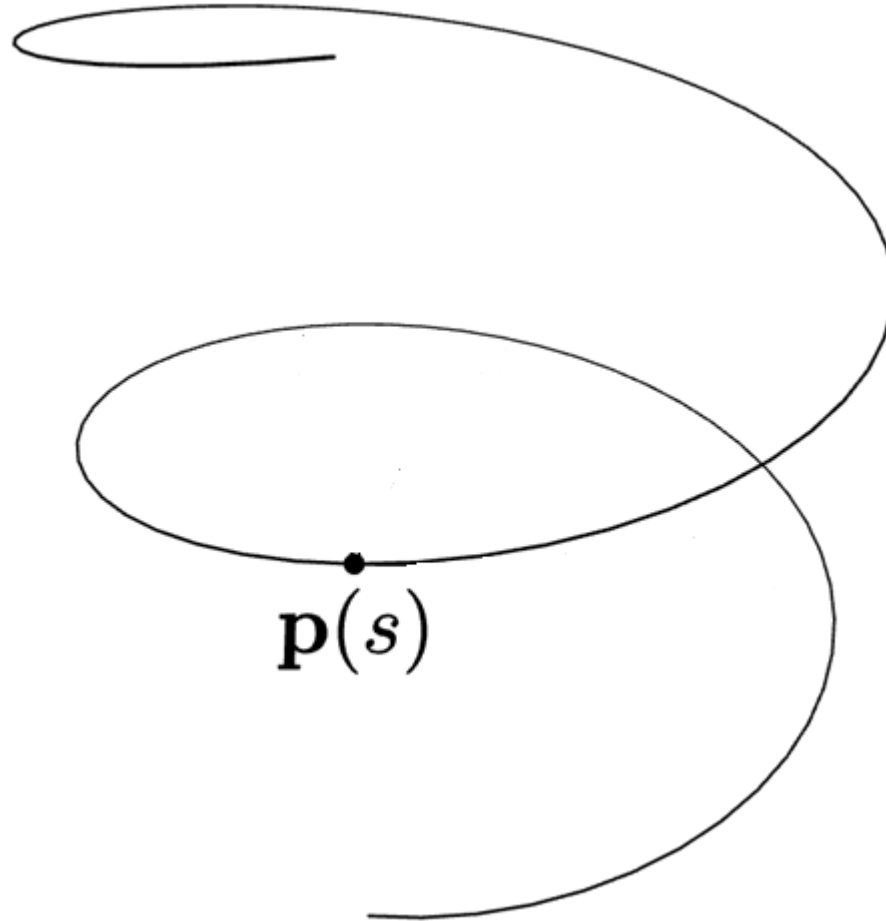
Generalized Cylinder

- Construct a 2-D profile curve

$$\mathbf{q}(t) = (a(t), b(t))$$

- Construct a space curve

$$\mathbf{p}(s) = (x(s), y(s), z(s))$$



Generalized Cylinder

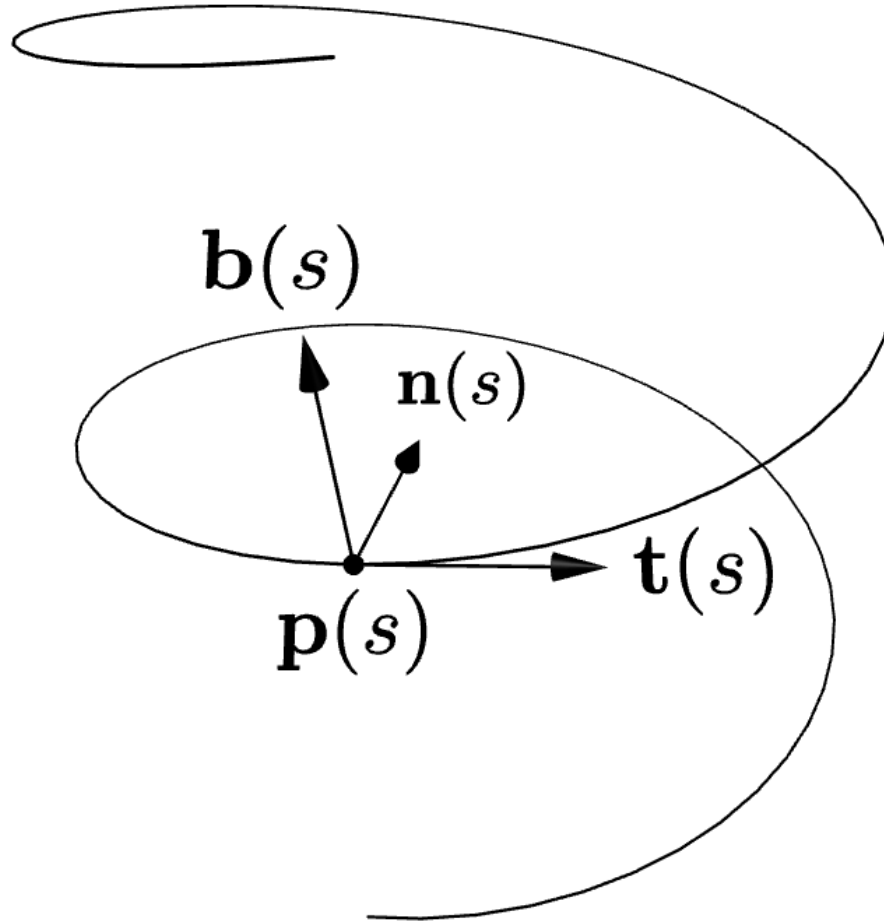
- Construct a Frenet frame at each point along space curve

$$\mathbf{t}(s) = \mathbf{p}(s + \Delta s) - \mathbf{p}(s)$$

$$\mathbf{b}(s) = \mathbf{t}(s) \times -\mathbf{t}(s - \Delta s)$$

$$\mathbf{n}(s) = \mathbf{b}(s) \times \mathbf{t}(s)$$

(all normalized)



Generalized Cylinder

- Construct a 2-D profile curve

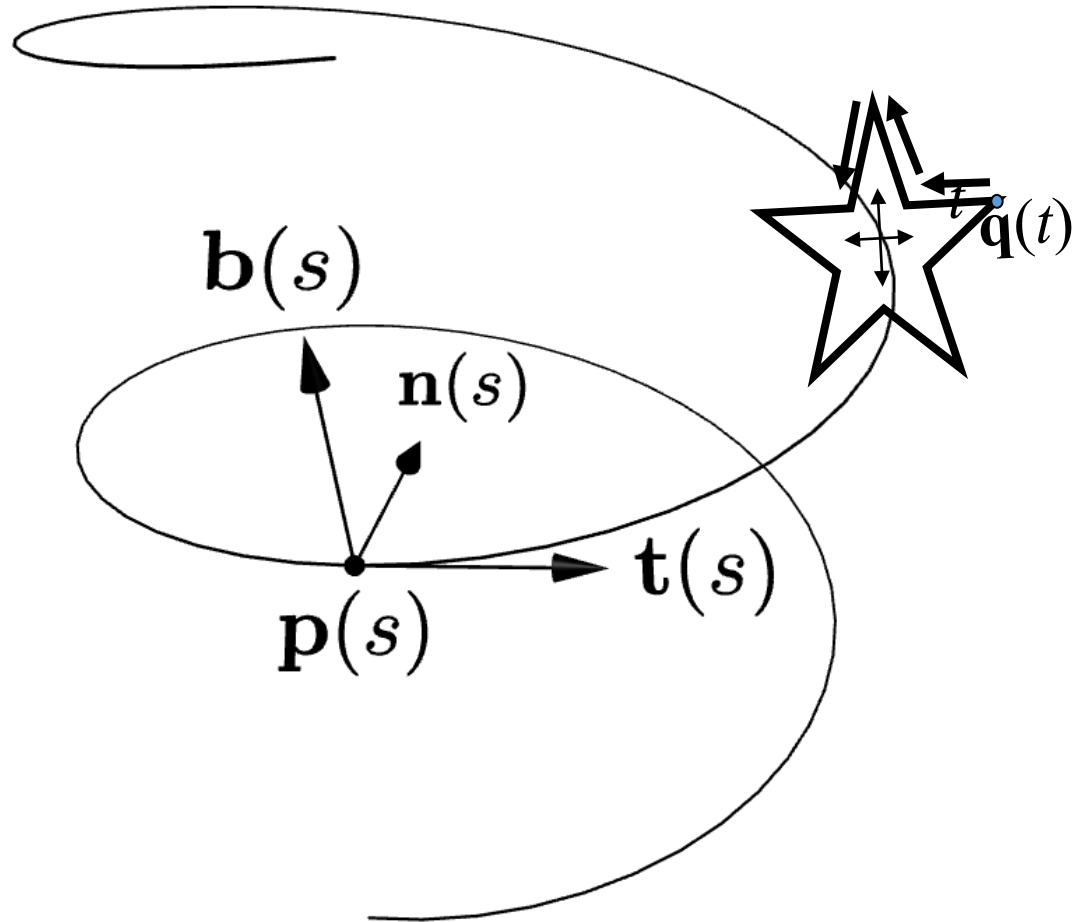
$$\mathbf{q}(t) = (a(t), b(t))$$

- Construct a space curve

$$\mathbf{p}(s) = (x(s), y(s), z(s))$$

- Plot 2-D curve in (N, B) space

$$\mathbf{gc}(s, t) = \mathbf{p}(s) + a(t) \mathbf{n}(s) + b(t) \mathbf{b}(s)$$



Generalized Cylinder

- Construct a 2-D profile curve

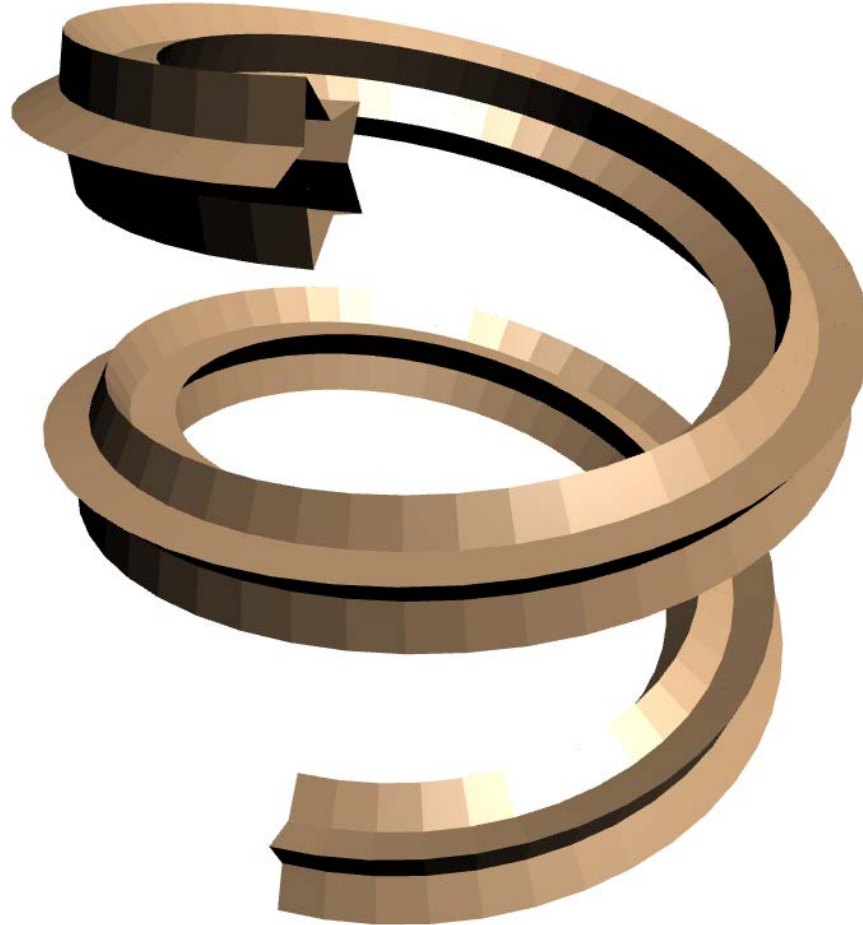
$$\mathbf{q}(t) = (a(t), b(t))$$

- Construct a space curve

$$\mathbf{p}(s) = (x(s), y(s), z(s))$$

- Plot 2-D curve in (N,B) space

$$\mathbf{gc}(s,t) = \mathbf{p}(s) + a(t) \mathbf{n}(s) + b(t) \mathbf{b}(s)$$



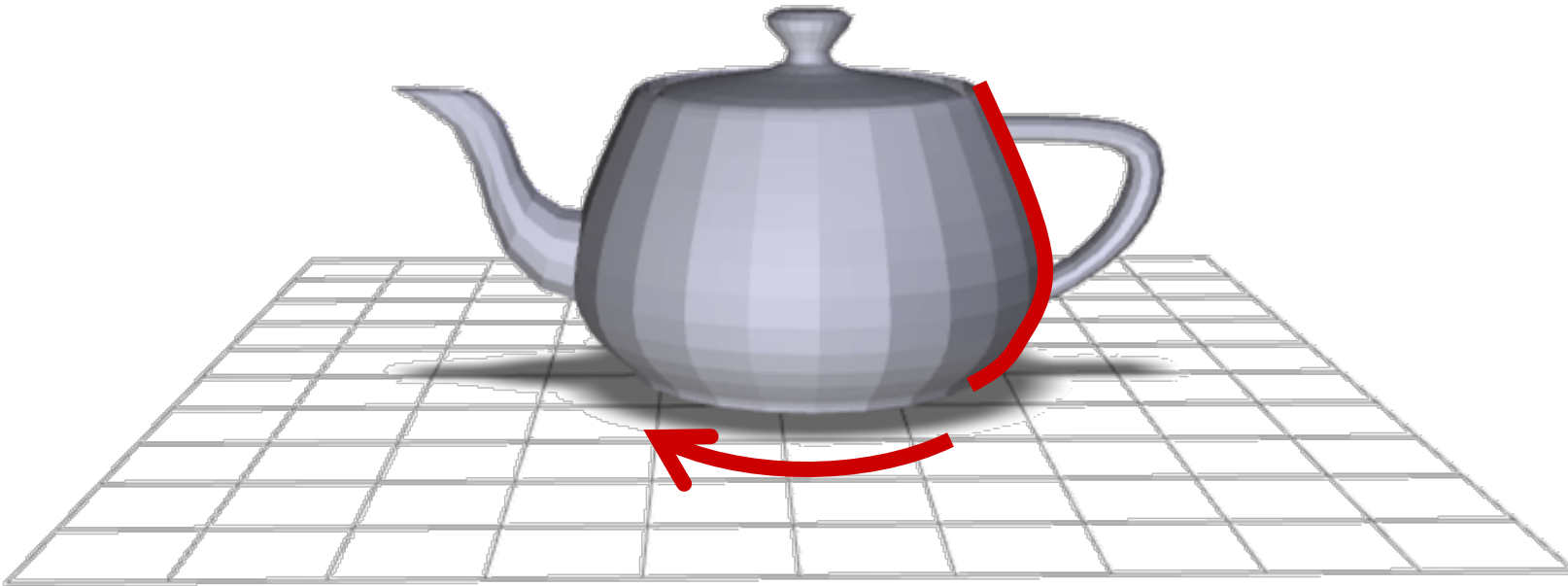
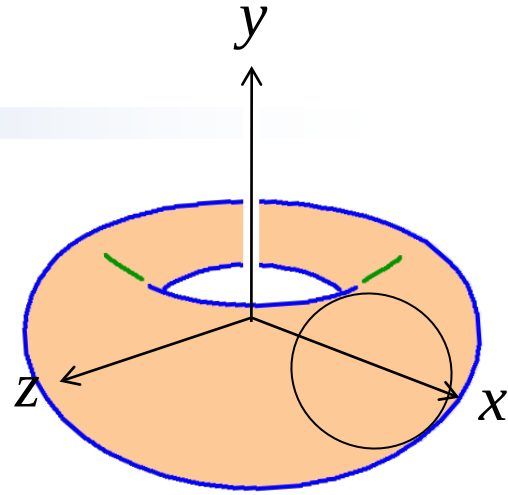
Revolution

- Construct a 2-D profile curve

$$\mathbf{q}(t) = (a(t), b(t))$$

- Rotate about y axis

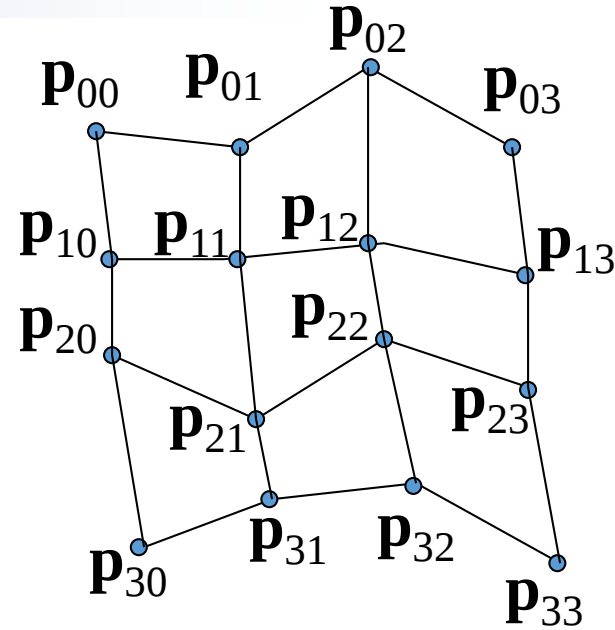
$$\mathbf{p}(s,t) = (a(t) \cos 2\pi s, b(t), a(t) \sin 2\pi s)$$



Bezier Patches

- Bezier patch
 - Tensor product of two Bezier curves

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n B_j^n(s) B_i^n(t) \mathbf{p}_{ij}$$



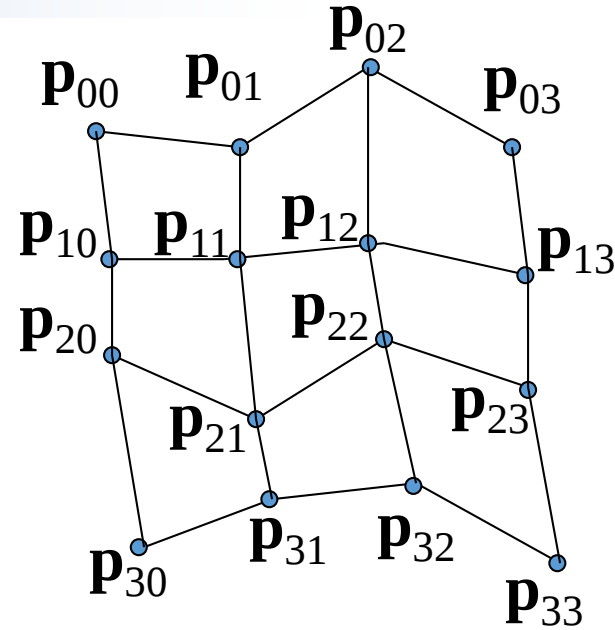
Bezier Patches

- Bezier patch
 - Tensor product of two Bezier curves

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n B_j^n(s) B_i^n(t) \mathbf{p}_{ij}$$

- Product of Bernstein polynomials

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n \left(B_j^n(s) B_i^n(t) \right) \mathbf{p}_{ij}$$



Bezier Patches

- Bezier patch
 - Tensor product of two Bezier curves

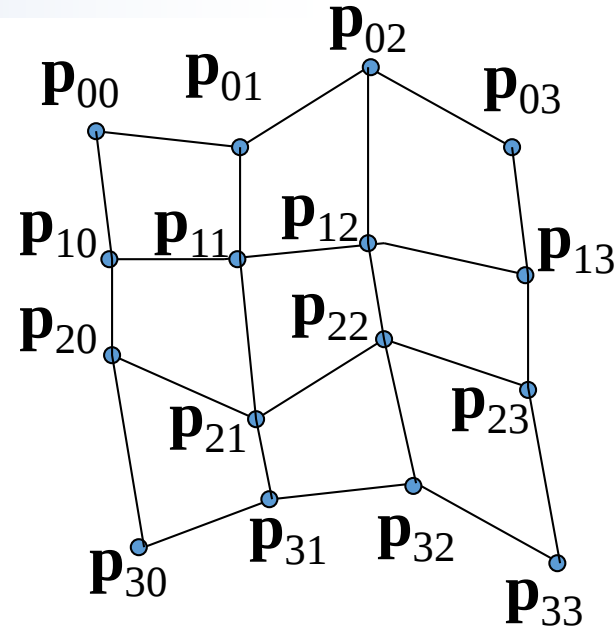
$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n B_j^n(s) B_i^n(t) \mathbf{p}_{ij}$$

- Product of Bernstein polynomials

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n \left(B_j^n(s) B_i^n(t) \right) \mathbf{p}_{ij}$$

- Bernstein interpolation of Bernstein polynomials

$$p(s, t) = \sum_{j=1}^n B_j^n(s) \left(\sum_{i=1}^n B_i^n(t) (\mathbf{p}_i) \right)_j$$



Bezier Patches

- Bezier patch
 - Tensor product of two Bezier curves

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n B_j^n(s) B_i^n(t) \mathbf{p}_{ij}$$

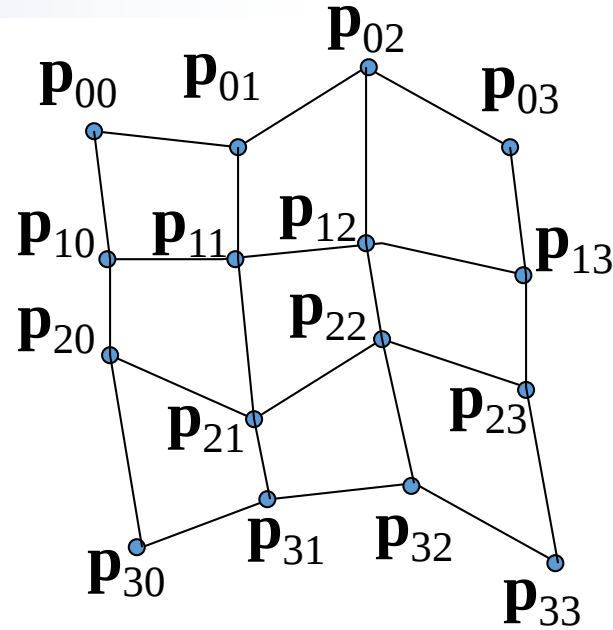
- Product of Bernstein polynomials

$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n \left(B_j^n(s) B_i^n(t) \right) \mathbf{p}_{ij}$$

- Bernstein interpolation of Bernstein polynomials

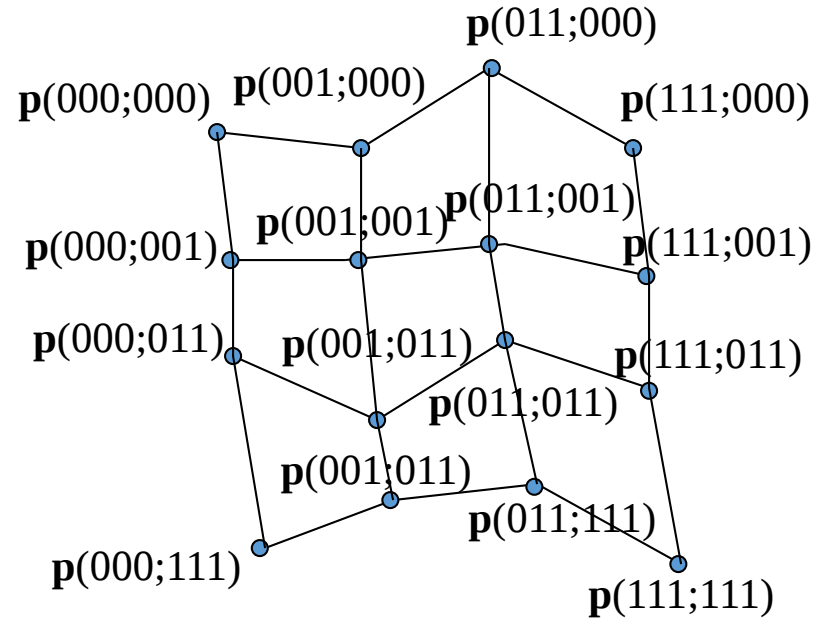
$$p(s, t) = \sum_{j=1}^n B_j^n(s) \left(\sum_{i=1}^n B_i^n(t) (\mathbf{p}_i) \right)_j$$

- Works same way for B-splines



Blossoming Patches

- Curves: $p(t) \rightarrow p(t,t,t)$
- Patches: $p(s,t) \rightarrow p(s,s,s;t,t,t)$
- Variables not allowed to cross the semicolon
- In patches, bilinear interpolation replaces linear interpolation in curves



Blossoming Patches

- Curves: $p(t) \rightarrow p(t,t,t)$
- Patches: $p(s,t) \rightarrow p(s,s,s;t,t,t)$
- Variables not allowed to cross the semicolon
- In patches, bilinear interpolation replaces linear interpolation in curves

