Parametric Representation

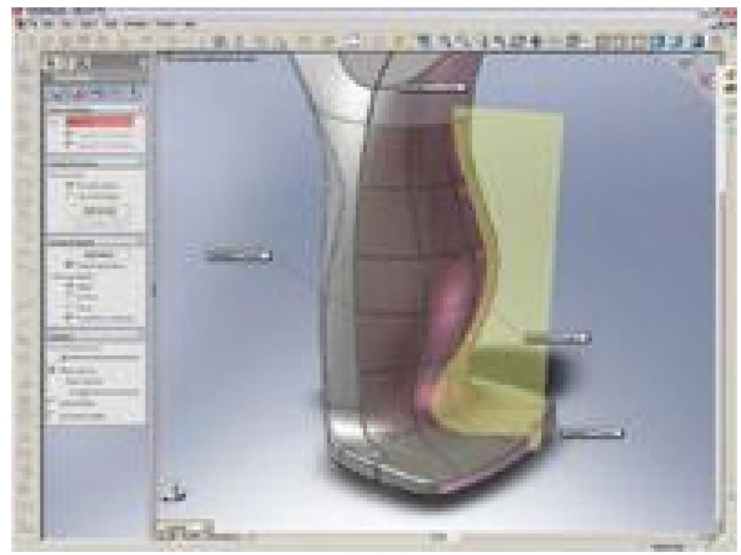
Qixing Huang

Nov. 7th 2018

Parametric Representation

- Widely used in Graphics/CAD/Industrial design

- What we will learn
  - Hermite
  - Bézier
  - Bspline
  - Many of their variants



Hermite curves

- A cubic polynomial
- Polynomial can be specified by the position of, and gradient at, each endpoint of curve
- Determine: x = X(t) in terms of x0, x0’, x1, x1’

The Hermite matrix: M_H

- The resultant polynomial can be expressed in matrix form:

$X(t) = t^T M_H q$ (q is the control vector)

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0' \\ x_1 \\ x_1' \end{bmatrix}$$

We can now define a parametric polynomial for each coordinate required independently, ie. X(t), Y(t) and Z(t)

Hermite Basis (Blending) Functions

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0' \\ x_1 \\ x_1' \end{bmatrix}$$

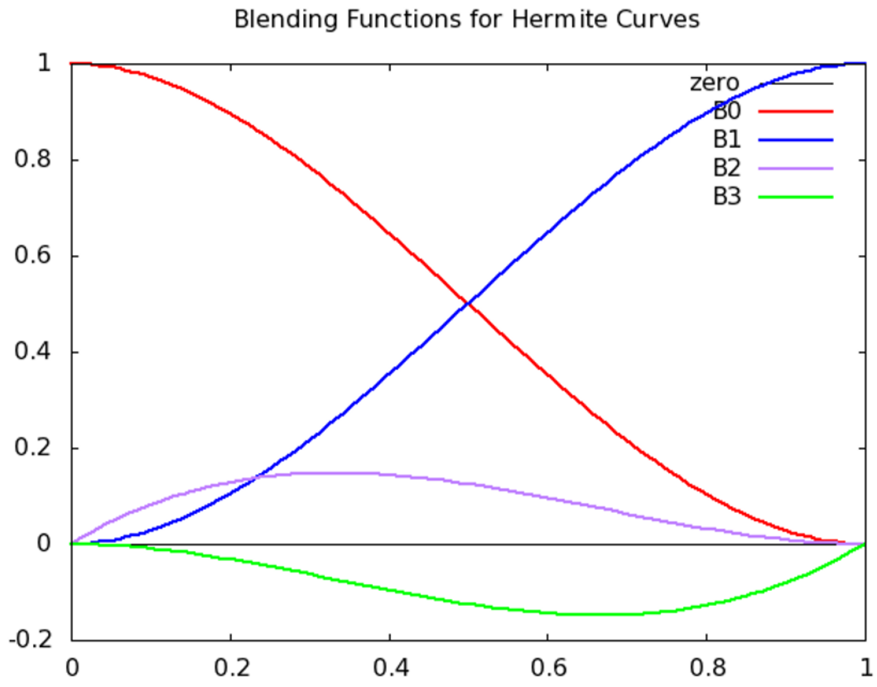
$$= \underline{\underline{(2t^3 - 3t^2 + 1)}}x_0 + \underline{\underline{(t^3 - 2t^2 + t)}}x_0' + \underline{\underline{(-2t^3 + 3t^2)}}x_1 + \underline{\underline{(t^3 - t^2)}}x_1'$$

Hermite Basis (Blending) Functions

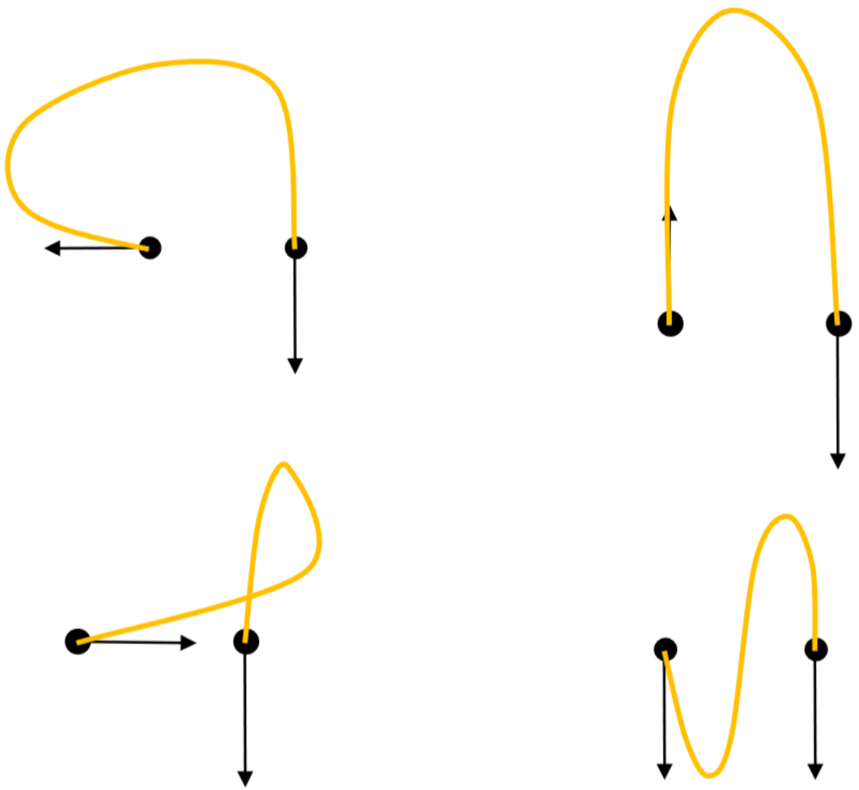
$$X(t) = \underline{\underline{(2t^3 - 3t^2 + 1)}}x_0 + \underline{\underline{(t^3 - 2t^2 + t)}}x_0' + \underline{\underline{(-2t^3 + 3t^2)}}x_1 + \underline{\underline{(t^3 - t^2)}}x_1'$$

The plot shows the shape of the so-called *blending functions.*

Note that at each end only position is non-zero, so the curve must touch the endpoints



Hermite curves can be hard to model



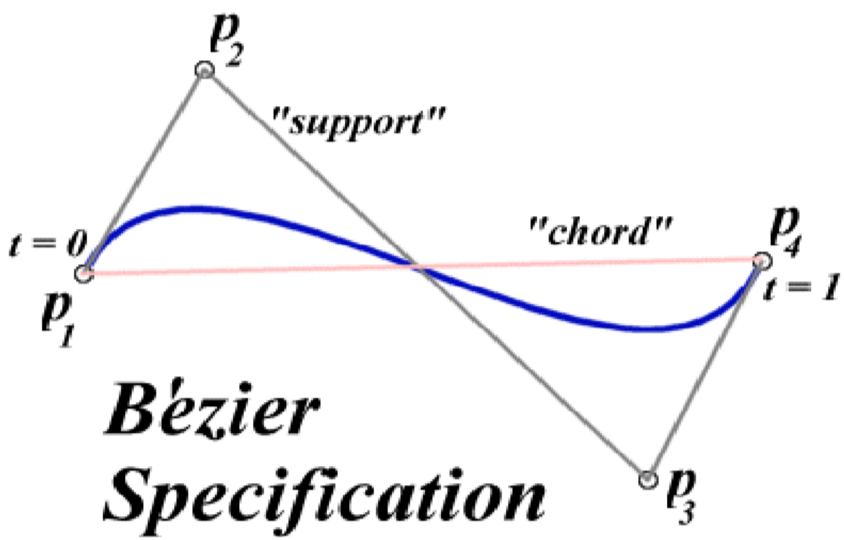
Note that the shape of the curve may not be intuitive from the boundary constraints

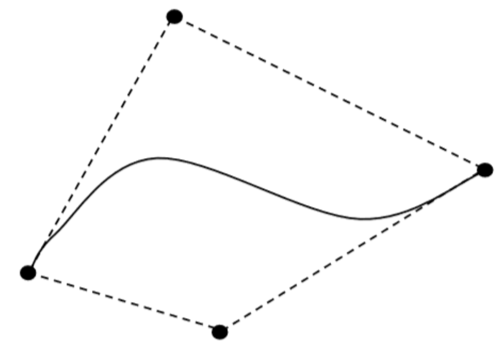
Bézier Curves

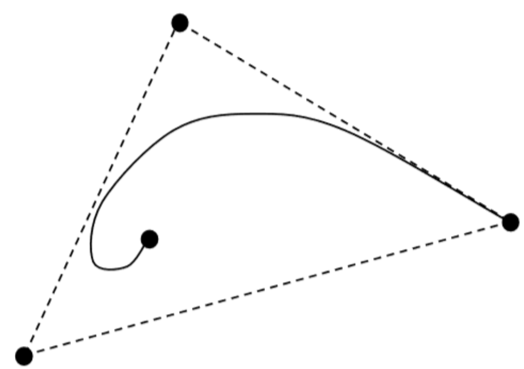
- Hermite cubic curves are mainly designed to be stitched into long curves
  - Yet the shapes are hard to control
- More intuitive to only specify points.
- Pierre Bézier (an engineer at Renault) specified 2 endpoints and 2 additional control points to specify the gradient at the endpoints.
- Strongly related to the Hermite curve

Bézier Curves

- Note the Convex Hull has been shown as a dashed line – used as a bounding extent for intersection purposes







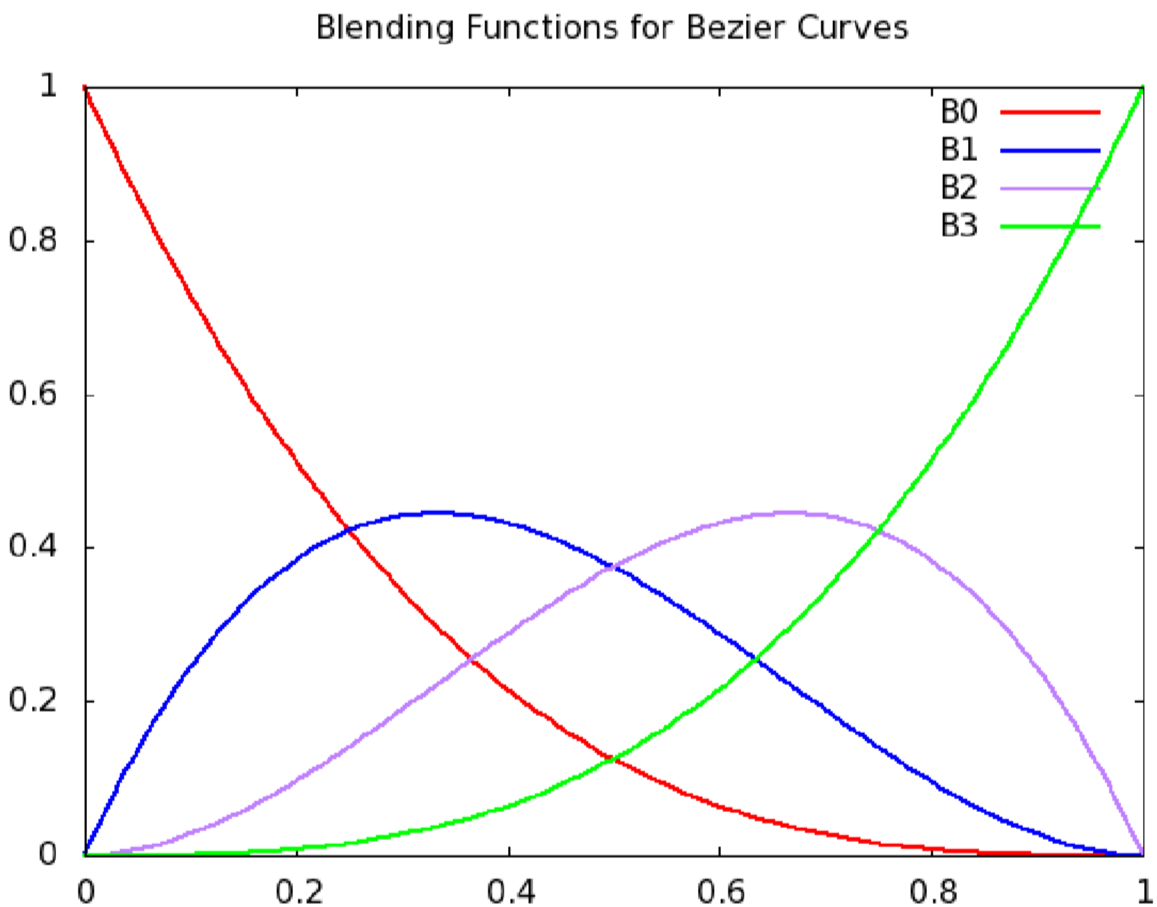
Bézier Matrix

- The cubic form is the most popular
  $X(t) = t^T M_B q$ (M_B is the Bézier matrix)
- With *n=4* and *r=0,1,2,3* *we get:*

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

- Similarly for Y(t) and Z(t)

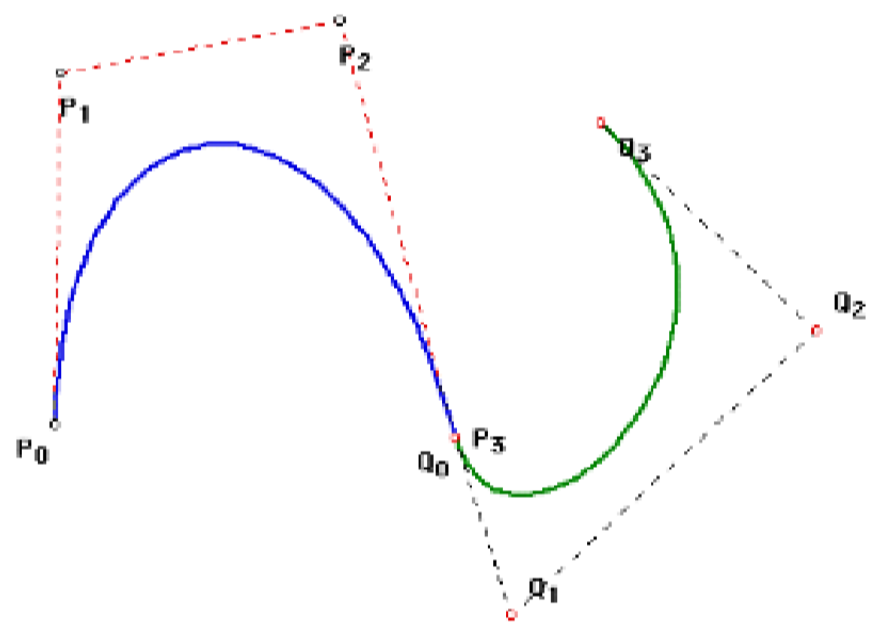
Bézier blending functions



Joining Bezier Curves

- G continuity is provided at the endpoint when $P_2 - P_3 = k\,(Q_1 - Q_0)$
- if $k=1$, C continuity is obtained

$$P_0 \cdot (1-t)^3 + P_1 \cdot 3 \cdot t \cdot (1-t)^2 + P_2 \cdot 3 \cdot t^2 \cdot (1-t) + P_3 \cdot t^3$$



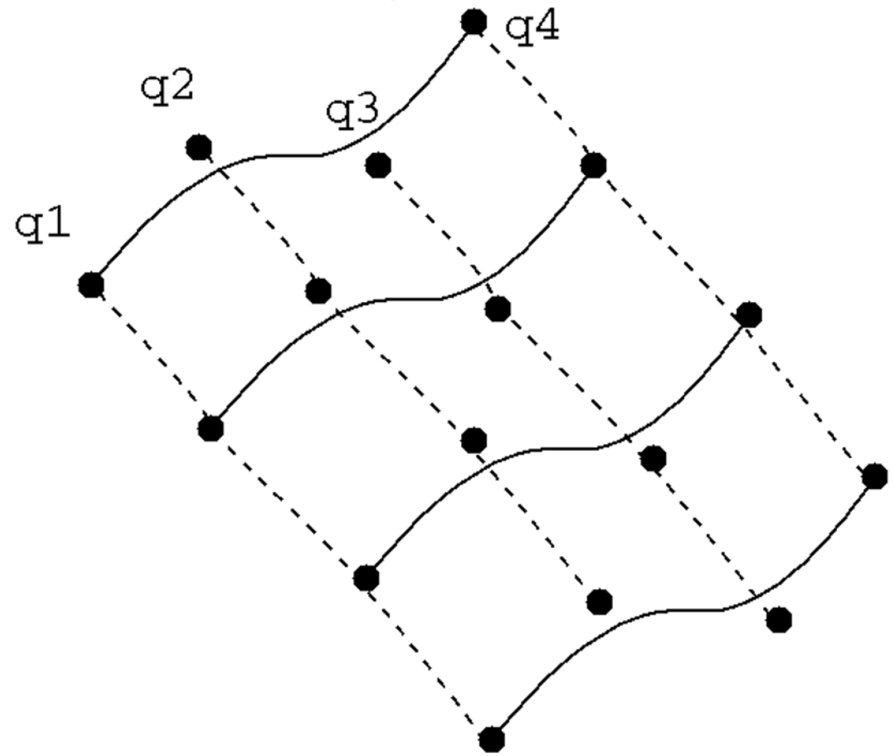
Bicubic patches

- The concept of parametric curves can be extended to surfaces

- The cubic parametric curve is in the form of $Q(t)=\mathbf{s}^T\mathbf{M}\ \mathbf{q}$ where $\mathbf{q}=(q_1,q_2,q_3,q_4)$: q_i control points, $\mathbf{M}$ is the basis matrix (Hermite or Bezier,…), $\mathbf{s}^T=(s^3, s^2, s, 1)$

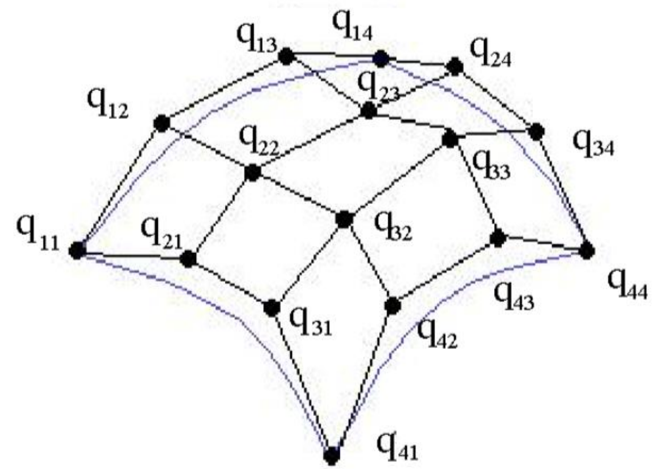
Bicubic patches

- Now we assume q_i to vary along a parameter s,
- $Q_i(s,t)=\mathbf{s}^T\mathbf{M}\,[q_1(t),q_2(t),q_3(t),q_4(t)]$
- $q_i(t)$ are themselves cubic curves, we can write them in the form …



Bézier example

- We compute (x,y,z) by



$$P(s,t) = \sum_{i=1}^{4}\sum_{j=1}^{4} P_{i,j} B_i(s) B_j(t)$$

$$x(s,t) = \begin{pmatrix} B_1(s) & B_2(s) & B_3(s) & B_4(s) \end{pmatrix} \begin{pmatrix} P_{11}^x & P_{12}^x & P_{13}^x & P_{14}^x \\ P_{21}^x & P_{22}^x & P_{23}^x & P_{24}^x \\ P_{31}^x & P_{32}^x & P_{33}^x & P_{34}^x \\ P_{41}^x & P_{42}^x & P_{43}^x & P_{44}^x \end{pmatrix} \begin{pmatrix} B_1(t) \\ B_2(t) \\ B_3(t) \\ B_4(t) \end{pmatrix}$$

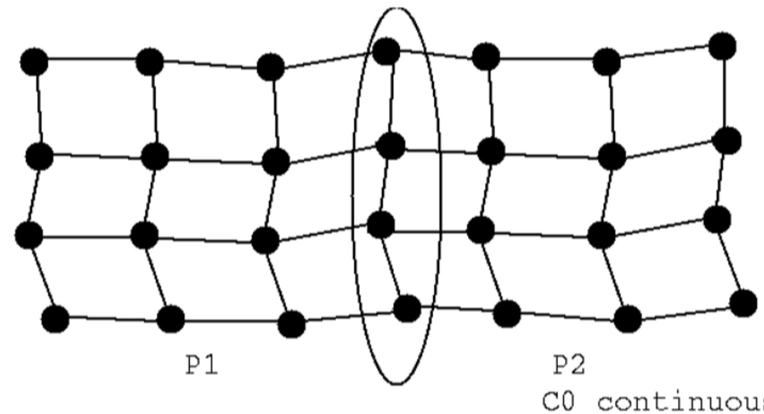
$\mathbf{s}^T M$

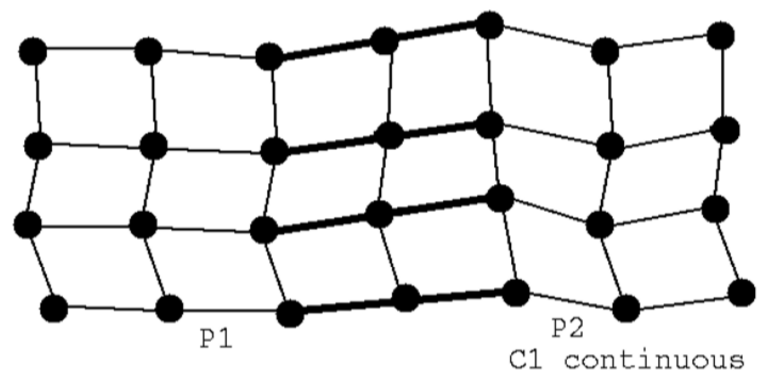
$M\mathbf{t}$

Replace *x* by *y* and *z*

Continuity of Bicubic Patches

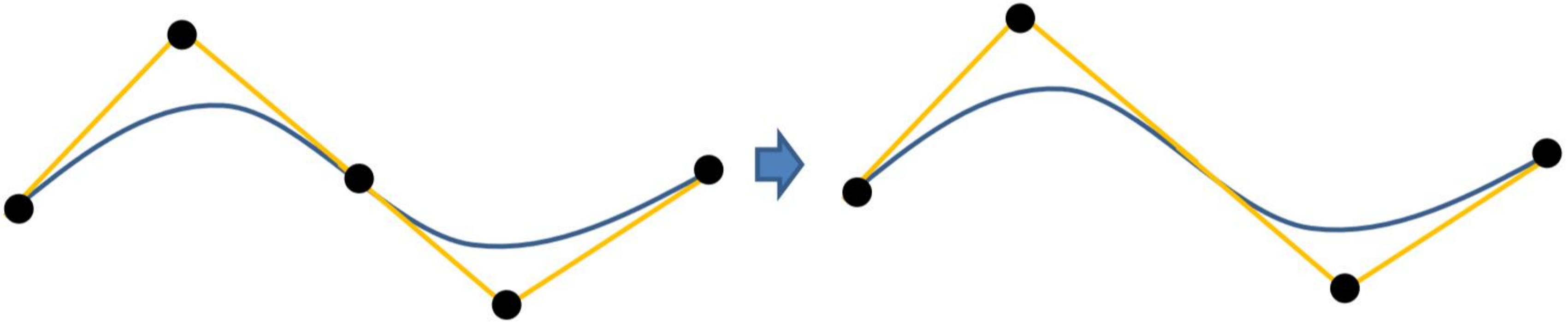
- Hermite and Bézier patches
  - C^0 continuity when sharing boundary control points
  - C^1 continuity when sharing boundary control points and boundary edge vectors





B-Spline Curves

Motivating Example



Uniform Bspline

$$c_1(s) = \mathbf{p}_0(1-s)^2 + \mathbf{p}_1 2(1-s)s + \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} s^2.$$

$$c_2(s) = \frac{\mathbf{p}_1 + \mathbf{p}_2}{2}(2-s)^2 + \mathbf{p}_2 2(2-s)(s-1) + \mathbf{p}_3(s-1)^2.$$

$$c(s) = \mathbf{p}_0 N_0(s) + \mathbf{p}_1 N_1(s) + \mathbf{p}_2 N_2(s) + \mathbf{p}_3 N_3(s).$$

Basis functions

$$N_0(s) = \begin{cases} (1-s)^2 & 0 \le s \le 1 \\ 0 & 1 \le s \le 2 \end{cases}$$

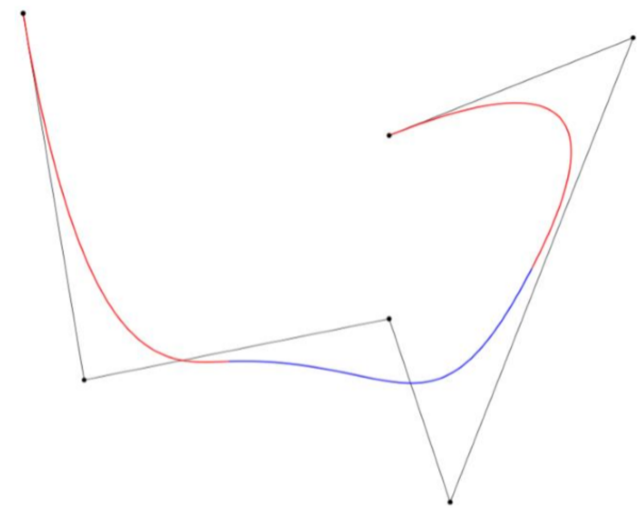
$$N_1(s) = \begin{cases} 2(1-s)s + \frac{s^2}{2} & 0 \le s \le 1 \\ \frac{(2-s)^2}{2} & 1 \le s \le 2 \end{cases}$$

$$N_2(s) = \begin{cases} \frac{s^2}{2} & 0 \le s \le 1 \\ \frac{(2-s)^2}{2} + 2(2-s)(s-1) & 1 \le s \le 2 \end{cases}$$

$$N_3(s) = \begin{cases} 0 & 0 \le s \le 1 \\ (s-1)^2 & 1 \le s \le 2 \end{cases}$$

Generalization to Bspline definition

- Control points
- Knot vector $\boldsymbol{t} = (t_0, t_1, ... t_n)$



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

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

B-Spline Curve

- Bsplines are summarized from curves that stitch Bezier segments together

- Start with a sequence of control points

- Select four from middle of sequence (p_{i-2}, p_{i-1}, p_i, p_{i+1})
  - Bezier and Hermite goes between p_{i-2} and p_{i+1}
  - B-Spline doesn't interpolate (touch) any of them but approximates the going through p_{i-1} and p_i

Uniform B-Splines

- Approximating Splines
- Approximates n+1 control points
  - $P_0, P_1, \ldots, P_n, n \geq 3$
- Curve consists of n –2 cubic polynomial segments
  - $Q_3, Q_4, \ldots Q_n$
- t varies along B-spline as Q_i: $t_i <= t < t_{i+1}$
- t_i (i = integer) are knot points that join segment Q_{i-1} to Q_i
- Curve is uniform because knots are spaced at equal intervals of parameter, t

Uniform B-Splines

- First curve segment, Q_3, is defined by first four control points

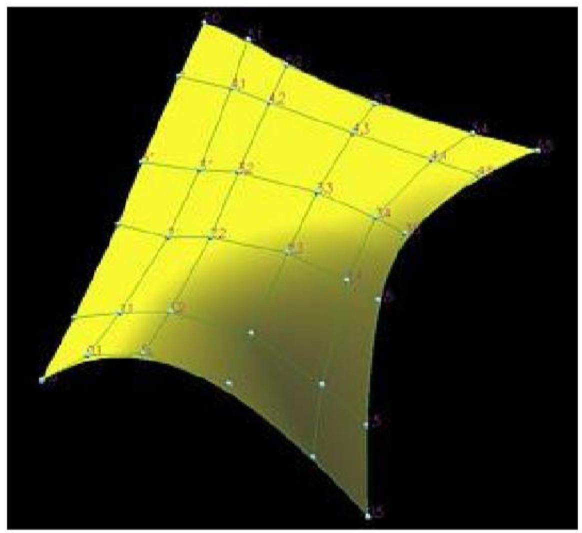
- Last curve segment, Q_m, is defined by last four control points, P_{m-3}, P_{m-2}, P_{m-1}, P_m

- Each control point affects four curve segments

B-Spline Surfaces

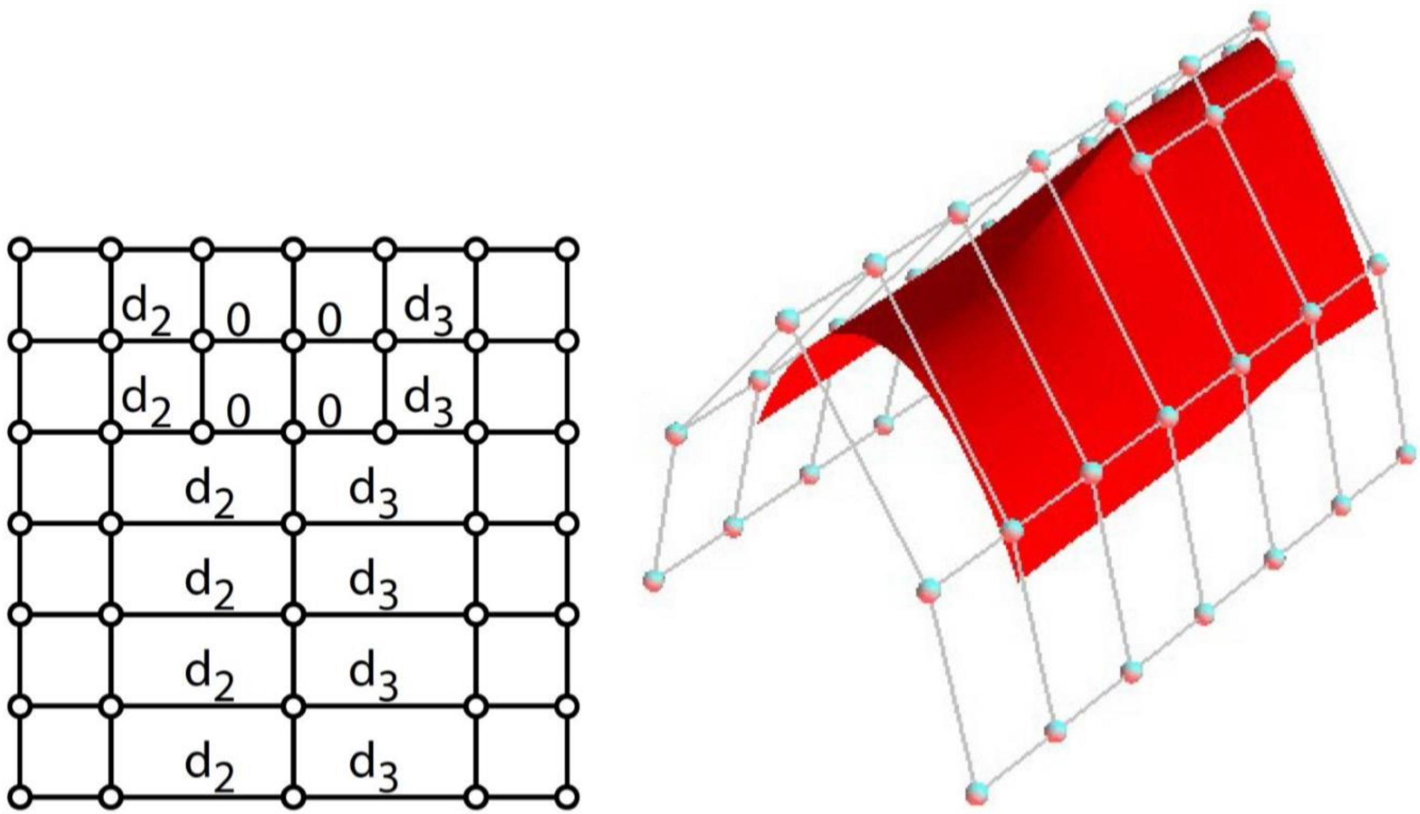
Bspline Surfaces

- The same way to we generalize Bezier curves to Bezier surfaces



$$\mathbf{p}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) \mathbf{p}_{i,j}$$

TSpline

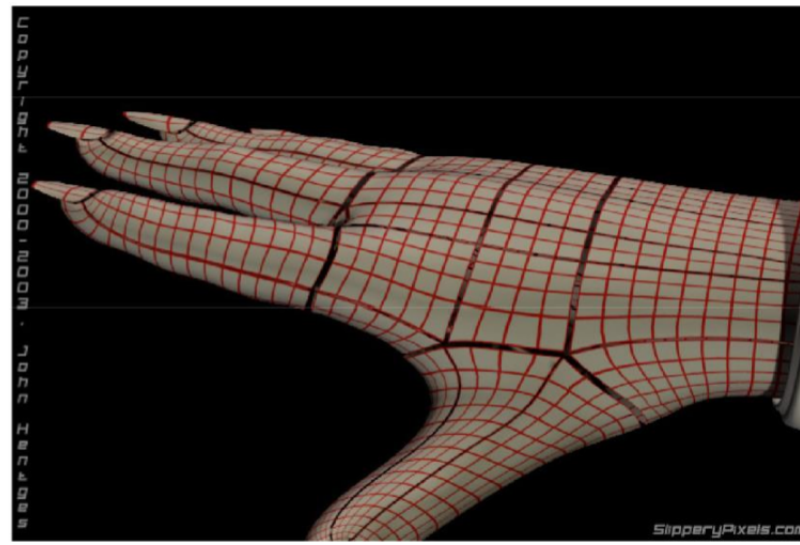


[Sederberg et al 03]

Subdivision Surfaces

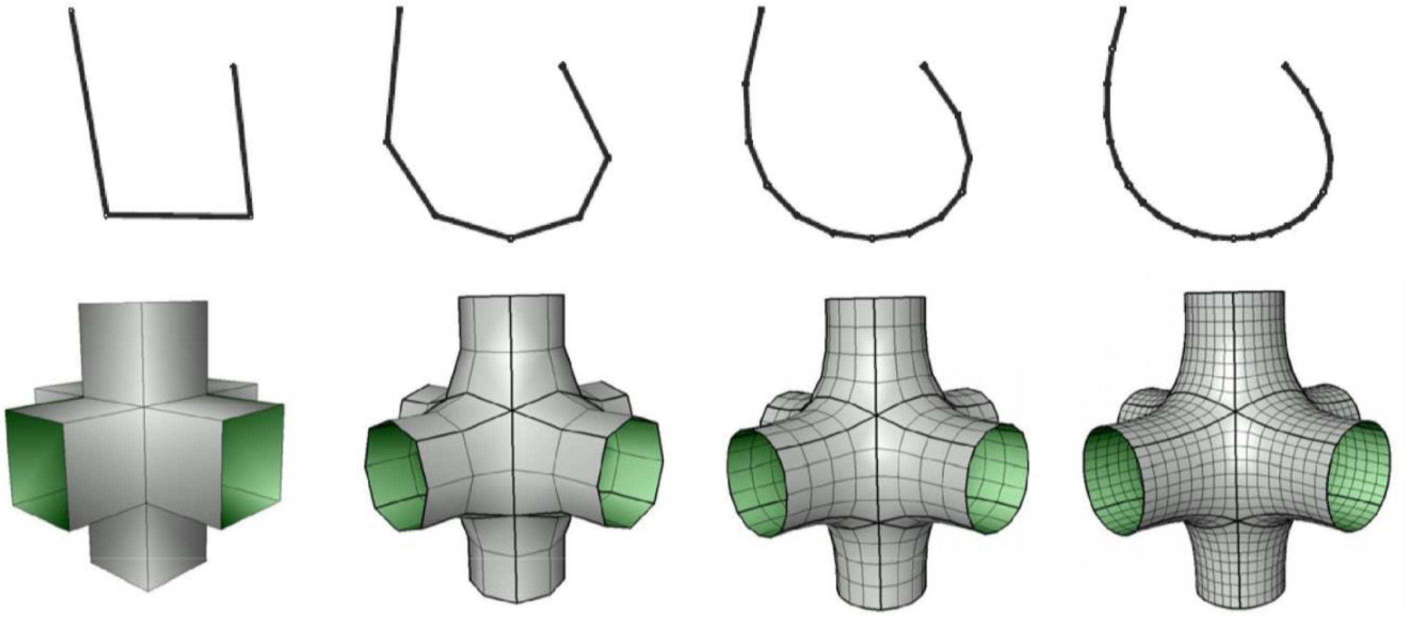
Problems with NURBS

- A single NURBS patch is either a disk a tube or a torus
- Must use many NURBS patches to model complex geometry
- When deforming a surface made of NURBS patches,cracks arise at the seams



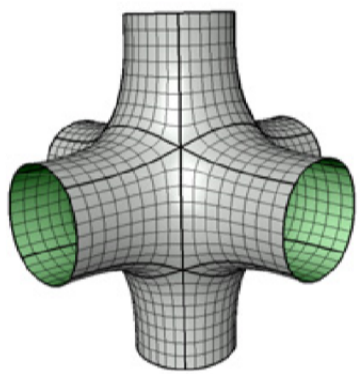
Subdivision

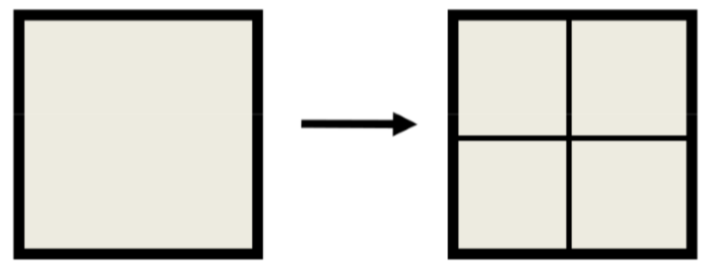
- “Subdivision defines a smooth curve or surfac eaes the limit of a sequence of successive refi nements”



Subdivision Rules

- How the connectivity changes

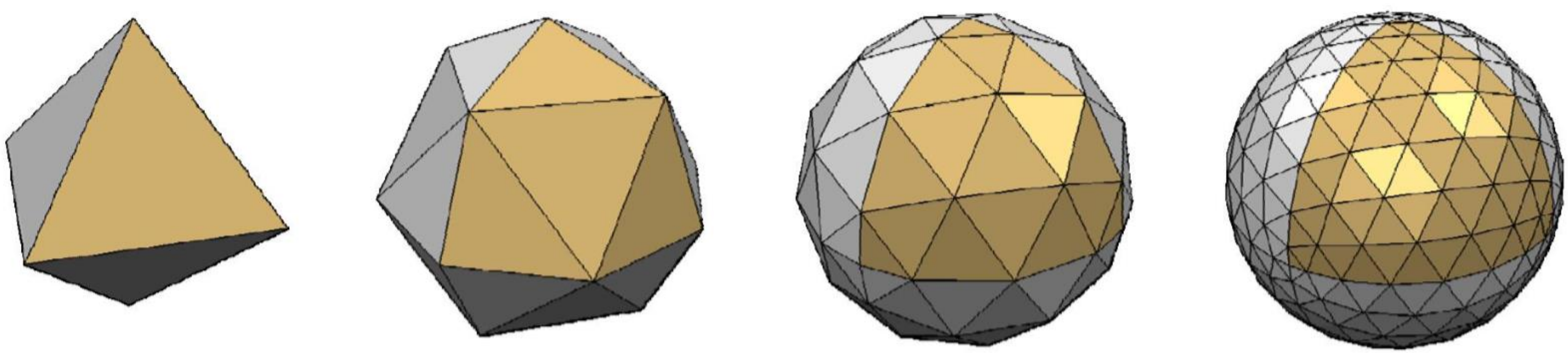




- How the geometry changes
  - Old points
  - New points

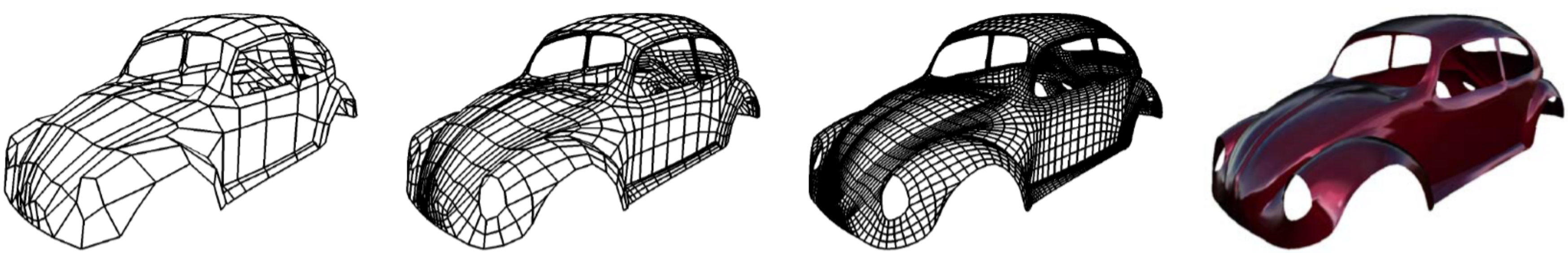
Subdivision Surfaces

- Generalization of spline curves / surfaces
  - Arbitrary control meshes
  - Successive refinement (subdivision)
  - Converges to smooth limit surface
  - Connection between splines and meshes



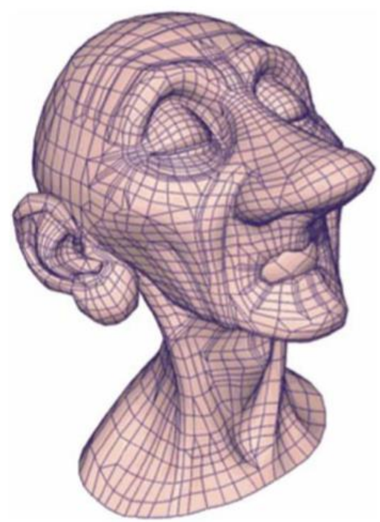
Subdivision Surfaces

- Generalization of spline curves / surfaces
  - Arbitrary control meshes
  - Successive refinement (subdivision)
  - Converges to smooth limit surface
  - Connection between splines and meshes



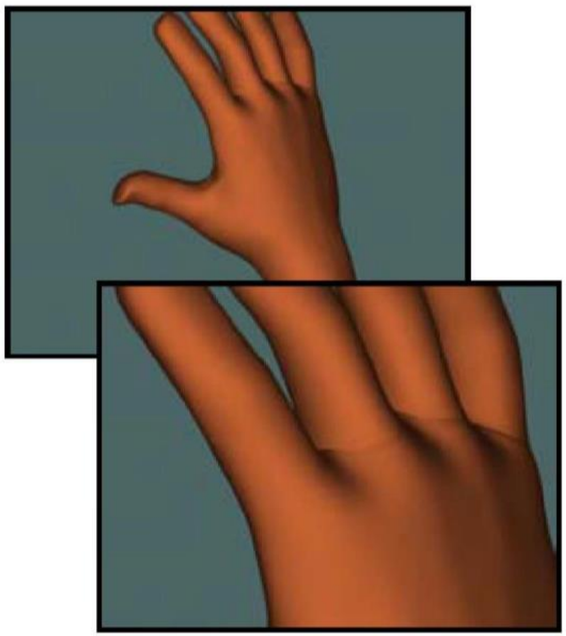
Example: Geri's Game (Pixar)

- Subdivision used for
  - Geri's hands and head
  - Clothing
  - Tie and shoes

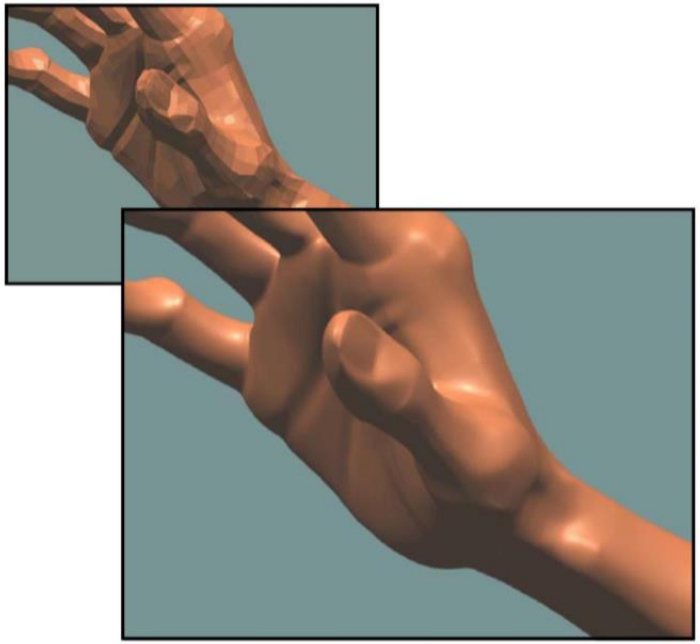


Example: Geri's Game (Pixar)

Woody's hand (NURBS)



Geri's hand (subdivision)



Example: Geri's Game (Pixar)

Sharp and semi-sharp features

Example: Games

- Supported in hardware in DirectX 11