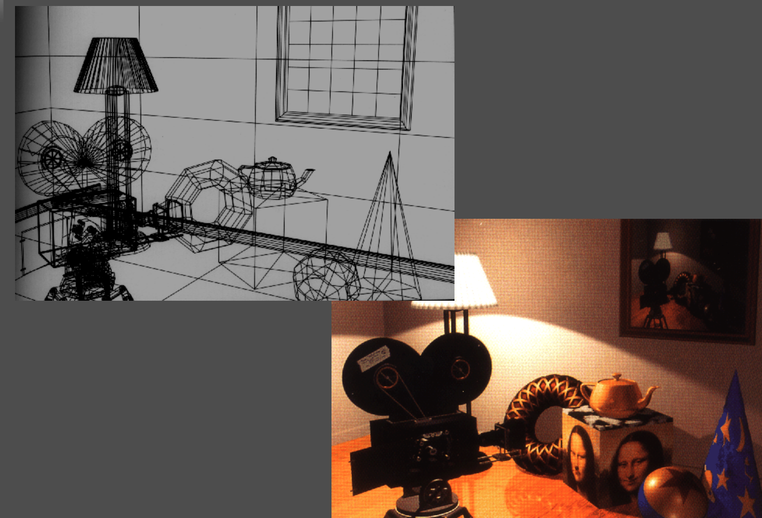
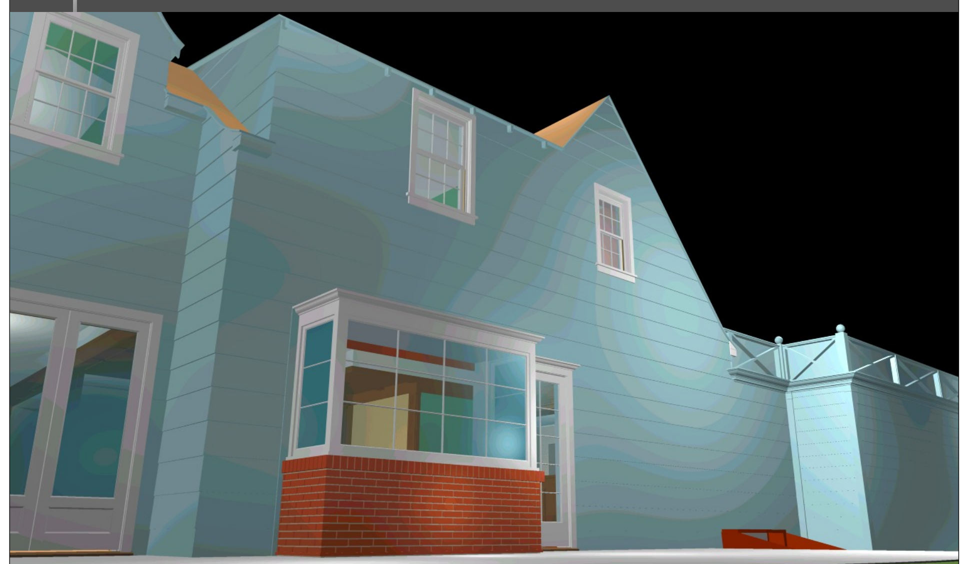
Geometric Modeling

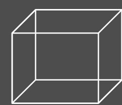
An Example



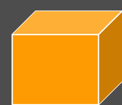


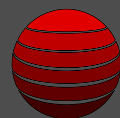
Outline

- Objective: Develop methods and algorithms to mathematically model shape of real world objects
- Categories:
  - Wire-frame representations

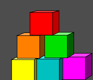


  - Boundary representations



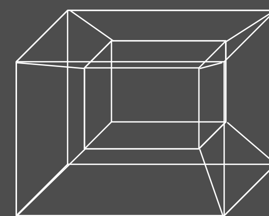


  - Volumetric representations



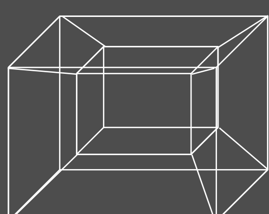
Wire-Frame Representation

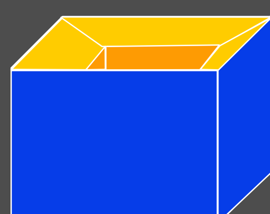
- Object is represented as as a set of points and edges (a graph) containing topological information.
- Used for fast display in interactive systems.
- Can be ambiguous:



Wire-Frame Representation

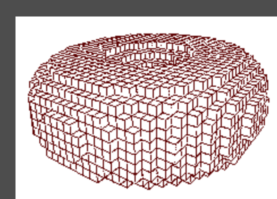
- Object is represented as as a set of points and edges (a graph) containing topological information.
- Used for fast display in interactive systems.
- Can be ambiguous:

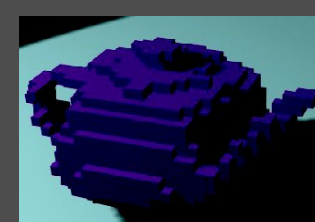




Volumetric Representation

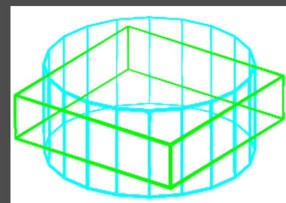
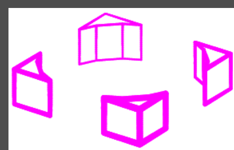
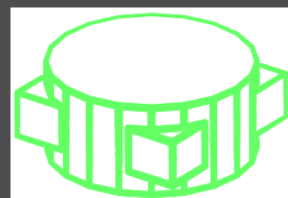
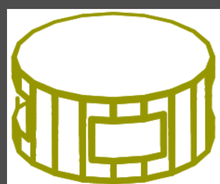
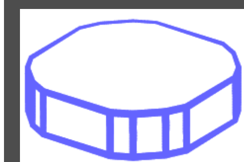
- Voxel based (voxel = 3D pixels).
- **Advantages**: simple and robust Boolean operations, in/out tests, can represent and model the *interior* of the object.
- **Disadvantages**: memory consuming, non-smooth, difficult to manipulate.





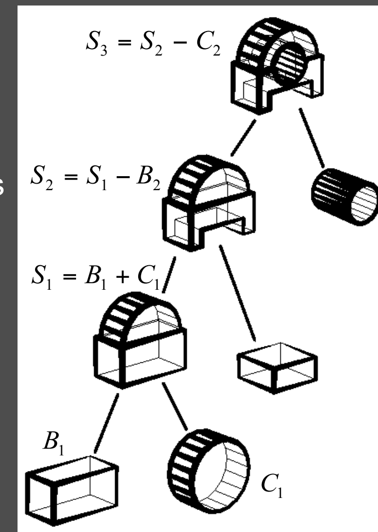
Constructive Solid Geometry

- Use set of volumetric primitives
  - Box, sphere, cylinder, cone, etc…
- For constructing complex objects use Boolean operations
  - Union
  - Intersection
  - Subtraction
  - Complement

CSG Trees

- Operations performed recursively
- Final object stored as sequence (tree) of operations on primitives
- Common in CAD packages –
  - mechanical parts fit well into primitive based framework
- Can be extended with free-form primitives



Freeform Representation

- Explicit form: $z = z(x, y)$
- Implicit form: $f(x, y, z) = 0$
- Parametric form: $S(u, v) = [x(u, v), y(u, v), z(u, v)]$
- Example – origin centered sphere of radius R:

Explicit :

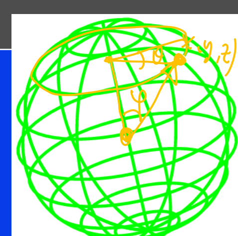
$$z = +\sqrt{R^2 - x^2 - y^2} \cup z = -\sqrt{R^2 - x^2 - y^2}$$

Implicit :

$$x^2 + y^2 + z^2 - R^2 = 0$$

Parametric :

$$(x, y, z) = (R\cos\theta\cos\psi, R\sin\theta\cos\psi, R\sin\psi), \theta \in [0, 2\pi], \psi \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}]$$



Freeform Representation

- Explicit form: $z = z(x, y)$

Explicit is a special case of implicit and parametric form

- Implicit form: $f(x, y, z) = 0$
- Parametric form: $S(u, v) = [x(u, v), y(u, v), z(u, v)]$
- Example – origin centered sphere of radius R:

Explicit :

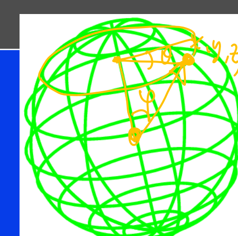
$$z = +\sqrt{R^2 - x^2 - y^2} \cup z = -\sqrt{R^2 - x^2 - y^2}$$

Implicit :

$$x^2 + y^2 + z^2 - R^2 = 0$$

Parametric :

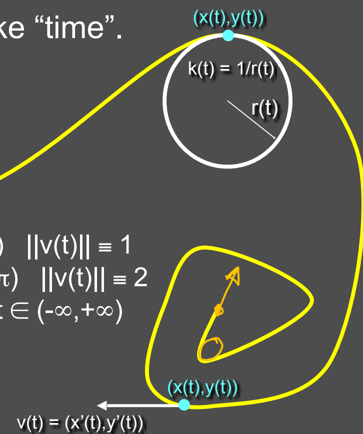
$$(x, y, z) = (R\cos\theta\cos\psi, R\sin\theta\cos\psi, R\sin\psi), \theta \in [0, 2\pi], \psi \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}]$$



Parametric Curves

- Analogous to trajectory of particle in space.
- Single parameter $t \in [T_1, T_2]$ – like "time".
- position = $p(t) = (x(t), y(t))$, velocity = $v(t) = (x'(t), y'(t))$

$t = T_1$

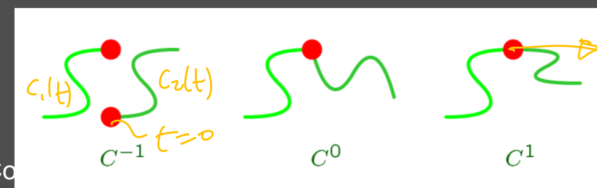


- Circle:
  - $x(t) = \cos(t)$, $y(t) = \sin(t)$ $t \in [0, 2\pi)$ $\|v(t)\| \equiv 1$
  - $x(t) = \cos(2t)$, $y(t) = \sin(2t)$ $t \in [0, \pi)$ $\|v(t)\| \equiv 2$
  - $x(t) = (1-t^2)/(1+t^2)$, $y(t) = 2t/(1+t^2)$ $t \in (-\infty, +\infty)$

Mathematical Continuity

- $C_1(t)$ & $C_2(t)$, $t \in [0,1]$ - parametric curves
- Level of continuity of the curves at $C_1(1)$ and $C_2(0)$ is:
  - C^{-1}: $C_1(1) \neq C_2(0)$ (discontinuous)
  - C^0: $C_1(1) = C_2(0)$ (positional continuity)
  - C^k, $k > 0$: continuous up to k^{th} derivative

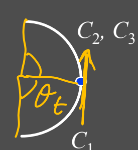
$$C_1^{(j)}(1) = C_2^{(j)}(0), \quad 0 \le j \le k$$



- Co[...] is similarly defined - for polynomial bases it is C^∞

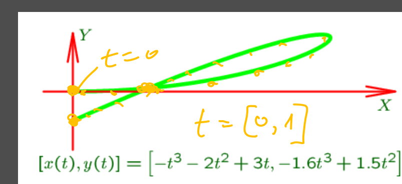
Geometric Continuity

- Mathematical continuity is sometimes too strong
- May be relaxed to geometric continuity
  - G^k, $k \le 0$: Same as C^k
  - G^k, $k = 1$: $C'_1(1) = \alpha C'_2(0)$
  - G^k, $k \ge 0$: There is a reparameterization of $C_1(t)$ & $C_2(t)$, where the two are C^k
- E.g.
  - $C_1(t) = [\cos(t), \sin(t)]$, $t \in [-\pi/2, 0]$ $\|V\| = 1$
    $C_2(t) = [\cos(t), \sin(t)]$, $t \in [0, \pi/2]$ $\|V\| = 1$
    $C_3(t) = [\cos(2t), \sin(2t)]$, $t \in [0, \pi/4]$ $\|V\| = 2$
  - $C_1(t)$ & $C_2(t)$ are C^1 (& G^1) continuous
  - $C_1(t)$ & $C_3(t)$ are G^1 continuous (not C^1)



Polynomial Bases

- *Monomial* basis $\{1, x, x^2, x^3, \ldots\}$
  - Coefficients are geometrically meaningless
  - Manipulation is not robust
- Number of coefficients = polynomial rank



- We seek coefficients with geometrically intuitive meanings
- Polynomials are easy to analyze, derivatives remain polynomial, etc.
- Other polynomial bases (with better geometric intuition):
  - Lagrange (Interpolation scheme)
  - Hermite (Interpolation scheme)
  - Bezier (Approximation scheme)
  - B-Spline (Approximation scheme)

Cubic Hermite Basis

- Set of polynomials of degree k is linear vector space of degree $k+1$
- The canonical, monomial basis for polynomials is

  $\{1, x, x^2, x^3, \ldots\}$
- Define geometrically-oriented basis for cubic polynomials

  $h_{i,j}(t)$: $i, j = 0,1,\ \ t \in [0,1]$
- Has to satisfy:

Curve	$h(0)$	$h(1)$	$h'(0)$	$h'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

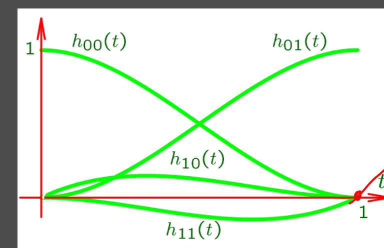
Hermite Cubic Basis

Curve	$h(0)$	$h(1)$	$h'(0)$	$h'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

- The four cubics which satisfy these conditions are

$$h_{00}(t) = t^2(2t-3)+1 \qquad h_{01}(t) = -t^2(2t-3)$$
$$h_{10}(t) = t(t-1)^2 \qquad h_{11}(t) = t^2(t-1)$$

- Obtained by solving four linear equations in four unknowns for each basis function
- **Prove**: Hermite cubic polynomials are linearly independent and form a basis for cubics



Hermite Cubic Basis (cont'd)

- Lets solve for $h_{00}(t)$ as an example.
- $h_{00}(t) = a\,t^3 + b\,t^2 + c\,t + d$

  must satisfy the following four constraints:

Curve	$h(0)$	$h(1)$	$h'(0)$	$h'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

$$h_{00}(0) = 1 = d,$$
$$h_{00}(1) = 0 = a+b+c+d,$$
$$h_{00}'(0) = 0 = c,$$
$$h_{00}'(1) = 0 = 3a+2b+c.$$

- Four linear equations in four unknowns.

Hermite Cubic Basis (cont'd)

$$C(t) = P_0 h_{00}(t) + P_1 h_{01}(t) + T_0 h_{10}(t) + T_1 h_{11}(t)$$

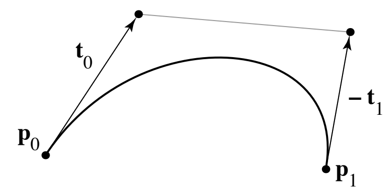
Hermite Cubic Basis (cont'd)

To generate a curve through P_0 & P_1 with slopes T_0 & T_1, use

$$\boxed{C(t) = P_0 h_{00}(t) + P_1 h_{01}(t) + T_0 h_{10}(t) + T_1 h_{11}(t)}$$

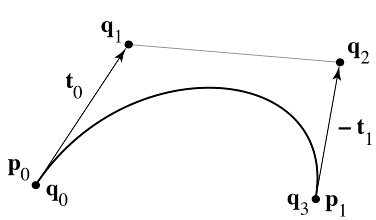
Hermite to Bézier

- Mixture of points and vectors is awkward
- Specify tangents as differences of points



Hermite to Bézier

- Mixture of points and vectors is awkward
- Specify tangents as differences of points

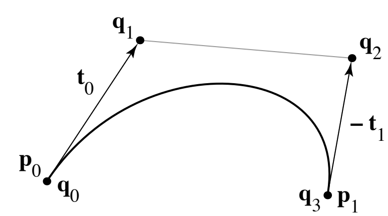


I'm calling these points **q** just for this slide and the next one.

– note derivative is defined as 3 times offset

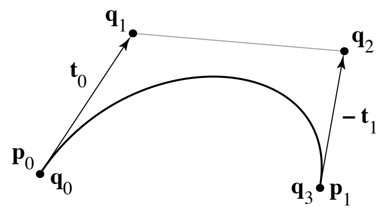
Hermite to Bézier

$$\mathbf{p}_0 = \mathbf{q}_0$$
$$\mathbf{p}_1 = \mathbf{q}_3$$
$$\mathbf{t}_0 = 3(\mathbf{q}_1 - \mathbf{q}_0)$$
$$\mathbf{t}_1 = 3(\mathbf{q}_3 - \mathbf{q}_2)$$



$$\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{v}_0 \\ \mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

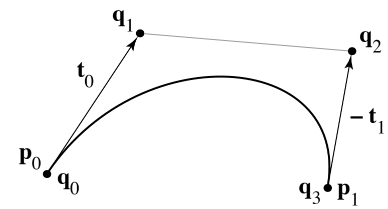
Hermite to Bézier



$$\mathbf{p}_0 = \mathbf{q}_0$$
$$\mathbf{p}_1 = \mathbf{q}_3$$
$$\mathbf{t}_0 = 3(\mathbf{q}_1 - \mathbf{q}_0)$$
$$\mathbf{t}_1 = 3(\mathbf{q}_3 - \mathbf{q}_2)$$

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

Hermite to Bézier



$$\mathbf{p}_0 = \mathbf{q}_0$$
$$\mathbf{p}_1 = \mathbf{q}_3$$
$$\mathbf{t}_0 = 3(\mathbf{q}_1 - \mathbf{q}_0)$$
$$\mathbf{t}_1 = 3(\mathbf{q}_3 - \mathbf{q}_2)$$

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

© 2017 Steve Marschner • 23

Bézier matrix

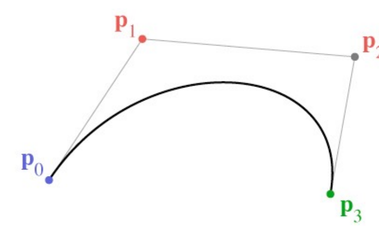
$$\mathbf{f}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

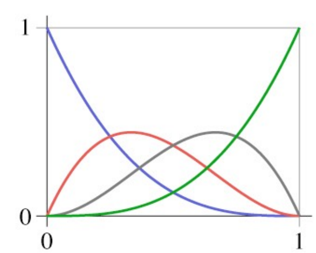
– note that these are the Bernstein polynomials

$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

and that defines Bézier curves for any degree

Bézier basis



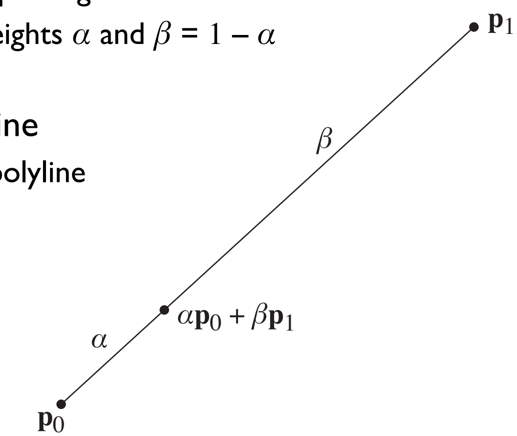


Another way to Bézier segments

- A really boring spline segment: f(t) = p0
  - it only has one control point
  - the curve stays at that point for the whole time
- Only good for building a *piecewise constant* spline
  - a.k.a. a set of points

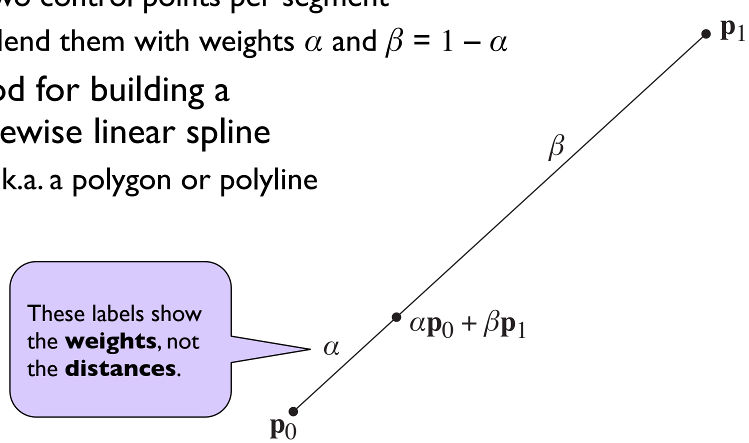
Another way to Bézier segments

- A piecewise linear spline segment
  - two control points per segment
  - blend them with weights α and $\beta = 1 - \alpha$
- Good for building a piecewise linear spline
  - a.k.a. a polygon or polyline



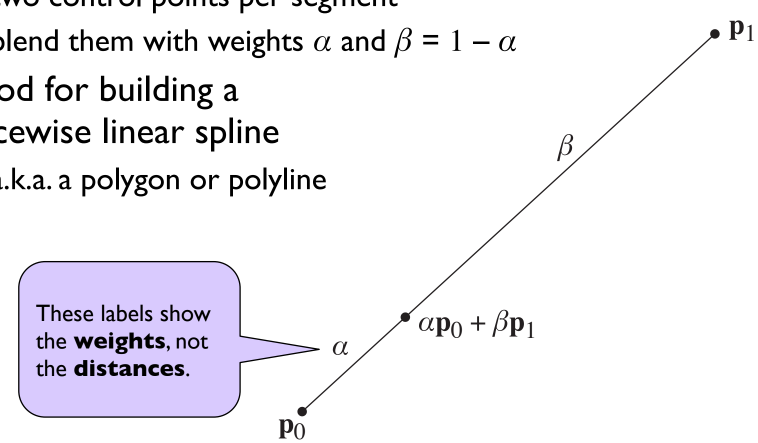
Another way to Bézier segments

- A piecewise linear spline segment
  - two control points per segment
  - blend them with weights α and $\beta = 1 - \alpha$
- Good for building a piecewise linear spline
  - a.k.a. a polygon or polyline



Another way to Bézier segments

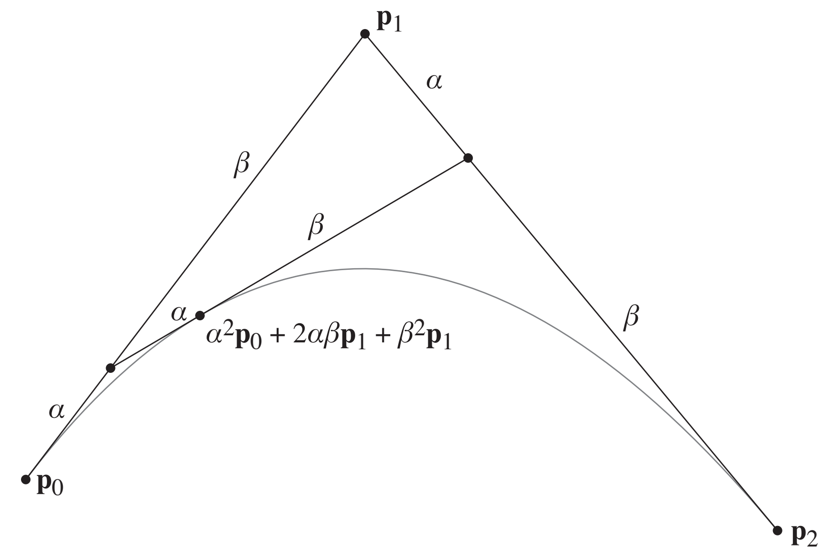
- A piecewise linear spline segment
  - two control points per segment
  - blend them with weights α and $\beta = 1 - \alpha$
- Good for building a piecewise linear spline
  - a.k.a. a polygon or polyline



Another way to Bézier segments

- A linear blend of two piecewise linear segments
  - three control points now
  - interpolate on both segments using α and β
  - blend the results with the same weights
- makes a quadratic spline segment
  - finally, a curve!

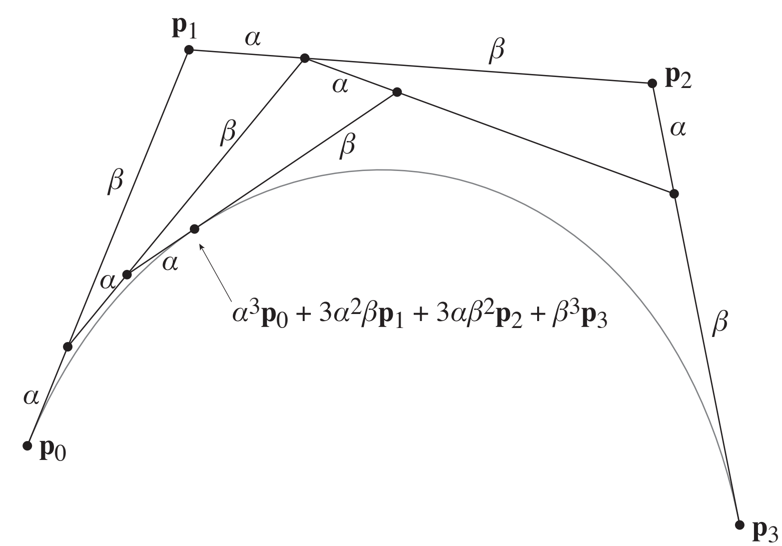
$$
\begin{aligned}
\mathbf{p}_{1,0} &= \alpha\mathbf{p}_0 + \beta\mathbf{p}_1 \\
\mathbf{p}_{1,1} &= \alpha\mathbf{p}_1 + \beta\mathbf{p}_2 \\
\mathbf{p}_{2,0} &= \alpha\mathbf{p}_{1,0} + \beta\mathbf{p}_{1,1} \\
&= \alpha\alpha\mathbf{p}_0 + \alpha\beta\mathbf{p}_1 + \beta\alpha\mathbf{p}_1 + \beta\beta\mathbf{p}_2 \\
&= \alpha^2\mathbf{p}_0 + 2\alpha\beta\mathbf{p}_1 + \beta^2\mathbf{p}_2
\end{aligned}
$$



Another way to Bézier segments

- Cubic segment: blend of two quadratic segments
  - four control points now (overlapping sets of 3)
  - interpolate on each quadratic using α and β
  - blend the results with the same weights
- makes a cubic spline segment
  - this is the familiar one for graphics—but you can keep going

$$
\begin{aligned}
\mathbf{p}_{3,0} =& \alpha\mathbf{p}_{2,0} + \beta\mathbf{p}_{2,1} \\
=& \alpha\alpha\alpha\mathbf{p}_0 + \alpha\alpha\beta\mathbf{p}_1 + \alpha\beta\alpha\mathbf{p}_1 + \alpha\beta\beta\mathbf{p}_2 \\
& \beta\alpha\alpha\mathbf{p}_1 + \beta\alpha\beta\mathbf{p}_2 + \beta\beta\alpha\mathbf{p}_2 + \beta\beta\beta\mathbf{p}_3 \\
=& \alpha^3\mathbf{p}_0 + 3\alpha^2\beta\mathbf{p}_1 + 3\alpha\beta^2\mathbf{p}_2 + \beta^3\mathbf{p}_3
\end{aligned}
$$

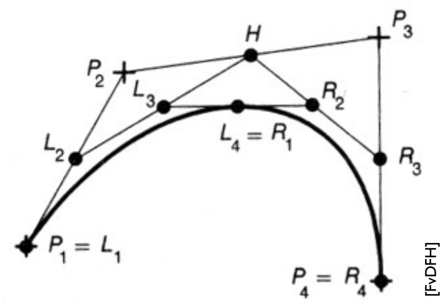


de Casteljau's algorithm

- A recurrence for computing points on Bézier spline segments:

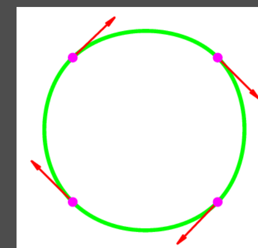
$$\mathbf{p}_{0,i} = \mathbf{p}_i$$
$$\mathbf{p}_{n,i} = \alpha \mathbf{p}_{n-1,i} + \beta \mathbf{p}_{n-1,i+1}$$

- Cool additional feature: also subdivides the segment into two shorter ones



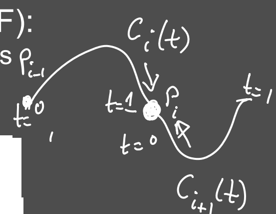
Parametric Splines

Fit spline independently for x(t) and y(t) to obtain C(t)



Cubic Splines

- Standard spline input – set of points $\{P_i\}_{i=0,\,n}$
  - No derivatives' specified as input
- Interpolate by n cubic segments ($4n$ DOF):
  - Derive $\{T_i\}_{i=0,\ldots,n}$ from C^2 continuity constraints
  - Solve $4n$ linear equations in $4n$ unknowns



$$C_0(0) = P_0; \quad C_n(1) = P_1$$
$$C_i(1) = P_i = C_{i+1}(0)$$
$$C_i'(1) = C_{i+1}'(0) \quad i = 1 \ldots n-1$$

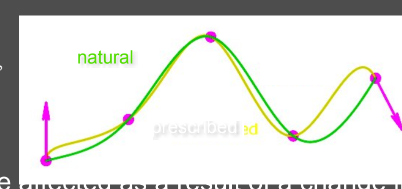
C^1 continuity constraints ($n-1$ equations):

$$C_i'(1) = C_{i+1}'(0) \;\; i = 1,..,n-1$$

C^2 continuity constraints ($n-1$ equations):

Cubic Splines

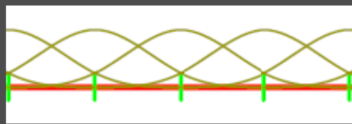
- Have two degrees of freedom left (to reach $4n$ DOF)
- Options
  - Natural end conditions: $C_1''(0) = 0,\ C_n''(1) = 0$
  - Complete end conditions: $C_1'(0) = 0,\ C_n'(1) = 0$
  - Prescribed end conditions (derivatives available at the ends): $C_1'(0) = T_0,\ C_n'(1) = T_n$
  - Periodic end conditions $C_1'(0) = C_n'(1),\ C_1''(0) = C_n''(1),$



- Question: What parts of $C(t)$ are affected as a result of a change in P_i ?

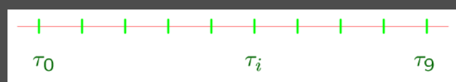
B-Spline Curves

- Idea: Generate basis where functions are continuous across the domains with *local support*



$$C(t)=\sum_{i=0}^{n-1} P_i N_i\,(t)$$

- For each parameter value only a finite set of basis functions is non-zero
- The parametric domain is subdivided into sections at parameter values called *knots*, $\{\tau_i\}$.
- The B-spline functions are then defined over the knots
- The knots are called *uniform knots* if $\tau_i - \tau_{i-1} = c$, constant. WLOG, assume $c = 1$.



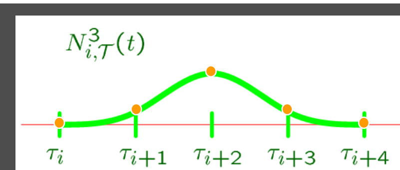
Uniform Cubic B-Spline Curves

- Definition (uniform knot sequence, $\tau_i - \tau_{i-1} = 1$):

$$\gamma(t)=\sum_{i=0}^{n-1} P_i N_i^3(t), \quad t\in[3,n)$$

$$N_i^3(t)=\begin{cases} r^3/6 & r=t-\tau_i & t\in[\tau_i,\tau_{i+1}) \\ (-3r^3+3r^2+3r+1)/6 & r=t-\tau_{i+1} & t\in[\tau_{i+1},\tau_{i+2}) \\ (3r^3-6r^2+4)/6 & r=t-\tau_{i+2} & t\in[\tau_{i+2},\tau_{i+3}) \\ (1-r)^3/6 & r=t-\tau_{i+3} & t\in[\tau_{i+3},\tau_{i+4}) \end{cases}$$

$r \in [0,1]$



0 1 2 3 4

$N_i^3(t)=0$ *elsewhere*

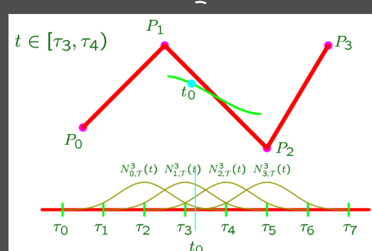
Uniform Cubic B-Spline Curves

- For any $t \in [3, n]$: $\sum_{i=0}^{n-1} N_i(t) = 1$ (prove it!)
- For any $t \in [3, n]$ at most four basis functions are non zero
- Any point on a cubic B-Spline is a convex combination of at most *four* control points

Let $t_0 \in [\tau_3, \tau_4)$. Then,

$$\gamma(t)\big|_{t=t_0} = \sum_{i=0}^{n-1} P_i N_i^3(t_0)$$

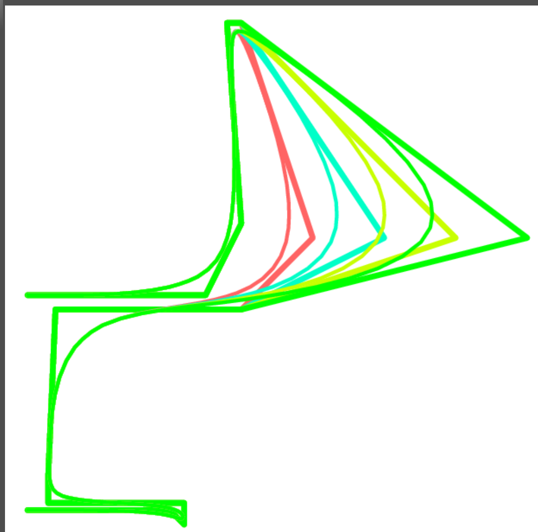
$$= \sum_{i=\tau_3-3}^{\tau_3} P_i N_i^3(t_0).$$



Boundary Conditions for Cubic B-Spline Curves

- B-Splines do not interpolate control points
  - in particular, the uniform cubic B-spline curves do not interpolate the end points of the curve.
  - Why is the end points' interpolation important?
- Two ways are common to force endpoint interpolation:
  - Let $P_0 = P_1 = P_2$ (same for other end)
  - Add a new control point (same for other end) $P_{-1} = 2P_0 - P_1$ and a new basis function $N_{-1}^3(t)$.

Local Control of B-spline Curves



Control point P_i affects $\gamma(t)$ only for $t \in (\tau_i, \tau_{i+4})$

Properties of B-Spline Curves

- $$\gamma(t) = \sum_{i=0}^{n-1} P_i N_i^3(t), \qquad t \in [3, n)$$
- For n control points, $\gamma(t)$ is a piecewise polynomial of degree 3, defined over $t \in [3, n)$
- $$\gamma(t) \in \bigcup_{i=0}^{n-4} CH(P_i, .., P_{i+3})$$
- $\gamma(t)$ is *affine invariant*
- $\gamma(t)$ follows the general shape of the control polygon and it is intuitive and ease to control its shape

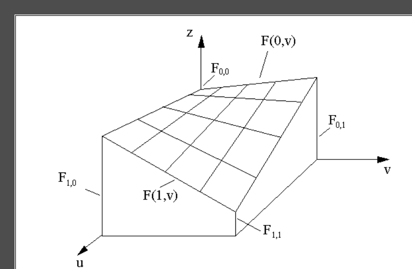
Surface Constructors

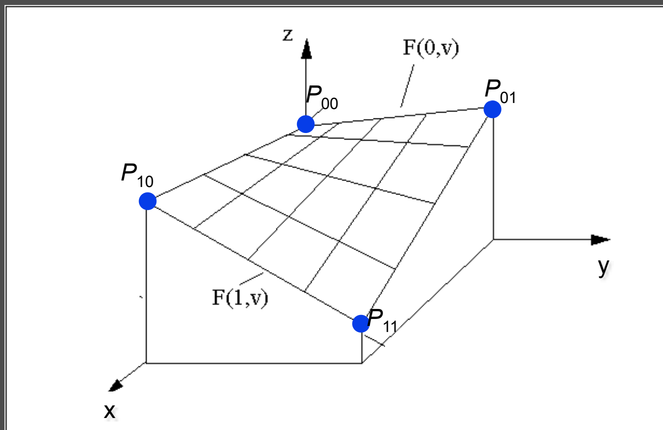
- Construction of the geometry is a first stage in any *image synthesis* process
- Use a set of high level, simple and intuitive, surface constructors:
  - Bilinear patch
  - Ruled surface
  - Boolean sum
  - Surface of Revolution
  - Extrusion surface
  - Surface from curves (skinning)
  - Swept surface

Bilinear Patches

- Bilinear interpolation of 4 3D points - 2D analog of 1D linear interpolation between 2 points in the plane
- Given P_{00}, P_{01}, P_{10}, P_{11} the bilinear surface for $u,v \in [0,1]$ is:

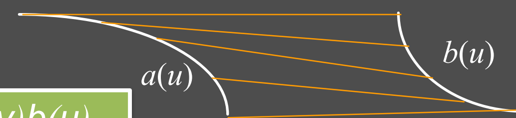
$$P(u,v) = (1-u)(1-v)P_{00} + (1-u)vP_{01} + u(1-v)P_{10} + uvP_{11}$$

- 



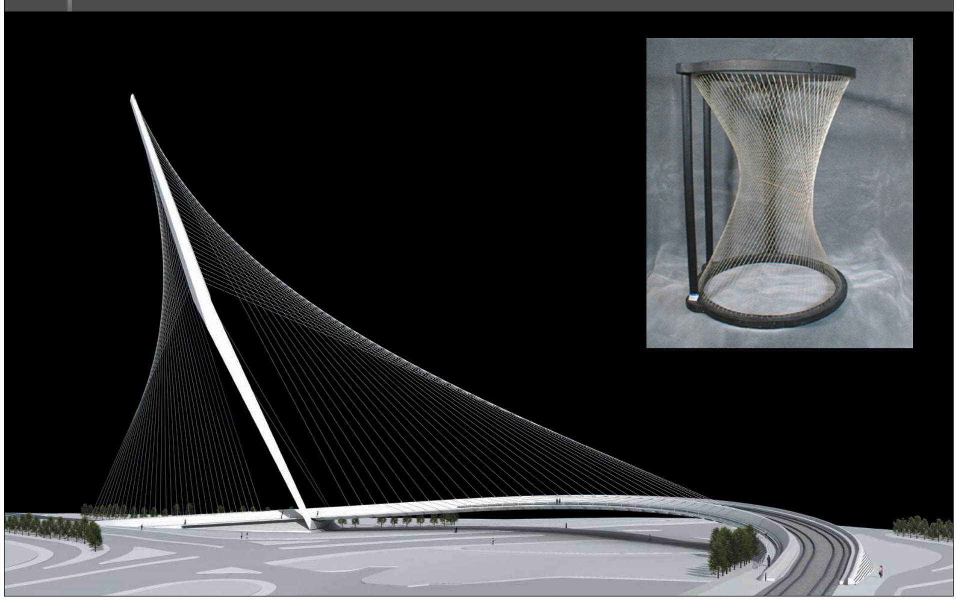
Ruled Surfaces

- Given two curves $a(t)$ and $b(t)$, the corresponding ruled surface between them is:



$$S(u,v) = v\,a(u) + (1-v)b(u)$$

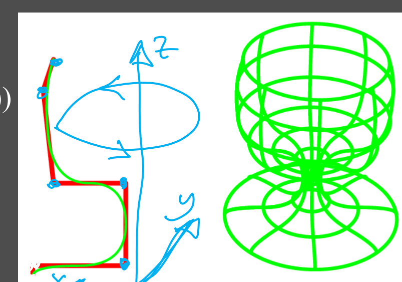
  - The corresponding points on $a(u)$ and $b(u)$ are connected by straight lines
- Questions:
  - When is a ruled surface a bilinear patch?
  - When is a bilinear patch a ruled surface?

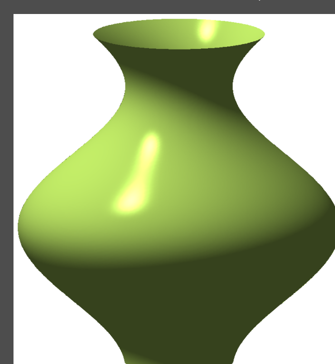


Surface of Revolution

- Rotate a, usually planar, curve around an axis

Consider curve $\beta(t) = (\beta_x(t), 0, \beta_z(t))$ and let Z be the axis of revolution. Then,

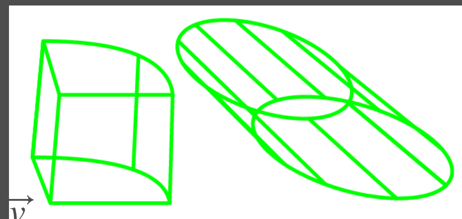




$$x(u,v) = \beta_x(u)\cos(v),$$
$$y(u,v) = \beta_x(u)\sin(v),$$
$$z(u,v) = \beta_z(u).$$

Extrusion

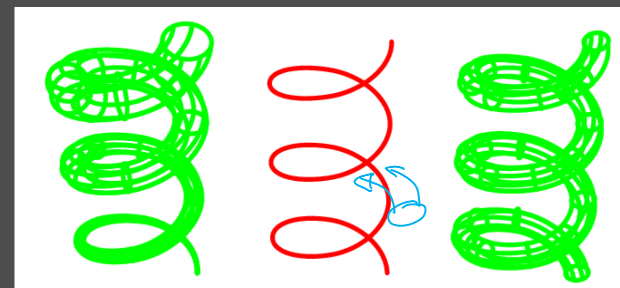
- Extrusion of a, usually planar, curve along a linear segment.



- Consider curve $\beta(t)$ and vector $\vec{v}$
- Then $$t' \cdot \vec{v} + \beta(t), \qquad 0 \leq t, t' \leq 1,$$

Sweep Surface

- Rigid motion of one (cross section) curve along another (axis) curve: S(u,v)
- In general, keeping one u fixed will generate a curve, which is a rigid motion (translation and ROTATION) of S(0,u)



- The cross section may change as it is swept