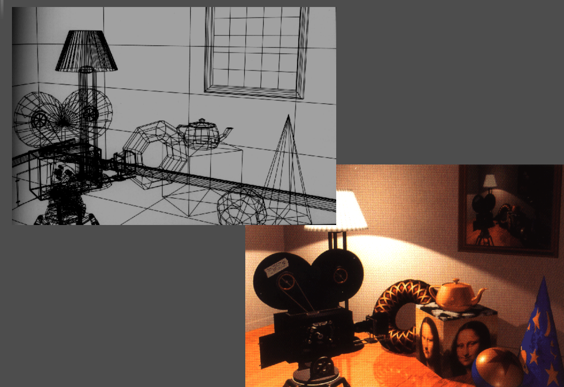
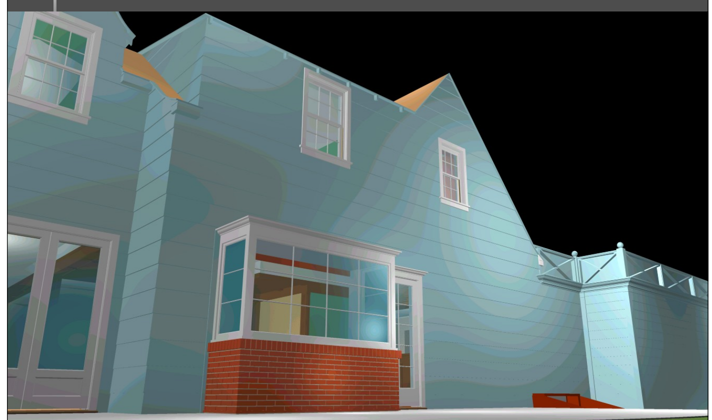
Geometric Modeling

An Example



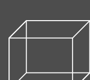
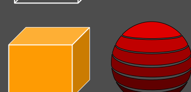
2

3



4

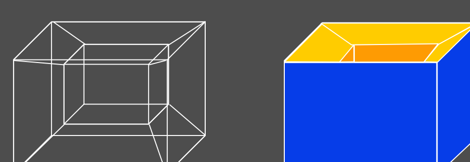
Outline

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

5

Wire-Frame Representation

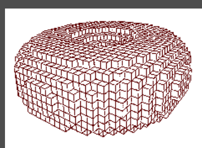
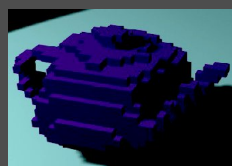
- Object is represented as as a set of points and edges (a graph) containing topological information.
- Used for fast display in interactive systems.
- Can be ambiguous:



6

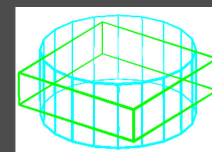
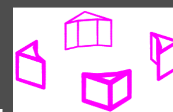
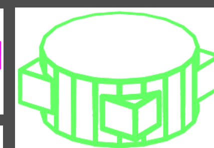
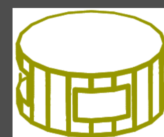
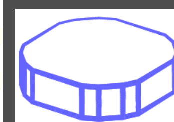
Volumetric Representation

- 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.

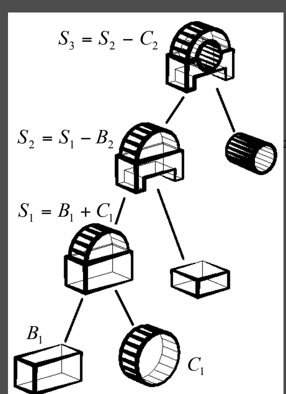
Constructive Solid Geometry

- 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: $S(u, v) = [x(u, v), y(u, v), z(u, v)]$
  - Useful to assign to texture - the (u,v) coordinates indicates a textile
- Example – origin centered sphere of radius R:

Explicit is a special case of implicit and parametric form

Explicit:

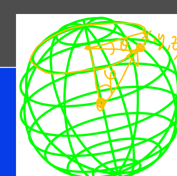
$z = +\sqrt{R^2 - x^2 - y^2} \cup z = -\sqrt{R^2 - x^2 - y^2}$

Implicit:

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

Parametric:

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



Motivation: Surface of Revolution

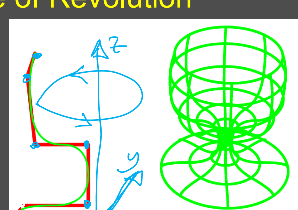
- Rotate a, usually planar, curve around the z-axis

First construct the curve in the

XZ-plane

https://www.geogebra.org/m/q8qf6jtd

$$\beta(t) = (\beta_x(t), \beta_z(t))$$



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

Big picture - what do we want

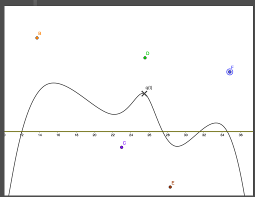
The designer could control a small number of points , and a curve connecting theses points is generated.

- Desired Properties:
- Easily controlled - small number of controlled points, and should be easy to predict the effect of each
- Effect should be local and stable (hopefully small change of control parameter ⇒ small change of the curve)
- Locality changes are **near** the control point
- Continuity. C^1 continuity. Geometric continuity (will discuss later)
- Easy to calculate, calculate intersection points etc. (nothing more complicated than cubic)

(unfortunately) before creating splines, lets create a segment of the spline. Then we could stitch them.

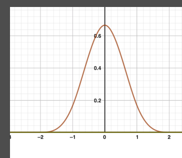
If theses segments starts/stops at control points, they must `glue' nicely.

B-Spline (Basic spline)

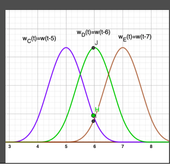


- The user enters a sequence of control points ($\{p_1 \ldots p_n\}$
- Each point on the curve is a convex combination of these points
- $C(t) = \sum w_i(t) \cdot p_i$
- As the weights changes in time, the curve is created
- $w_i(t) = w(t-i)$
- So $C(t) = \sum w_i(t-i) \cdot p_i$

$$w(t) = \begin{cases} 0 & : t \le -2 \\ \frac{(t+2)^3}{6} & : -2 \le t \le -1 \\ \frac{-3(t+1)^3 + 3(t+1)^2 + 3(t+1) + 1}{6} & : -1 \le t \le 0 \\ \frac{3t^3 - 6t^2 + 4}{6} & : 0 \le t \le 1 \\ \frac{(2-t)^3}{6} & : 1 < t \le 2 \\ 0 & : \text{otherwise} \end{cases}$$



https://www.geogebra.org/m/tpjr4sn4



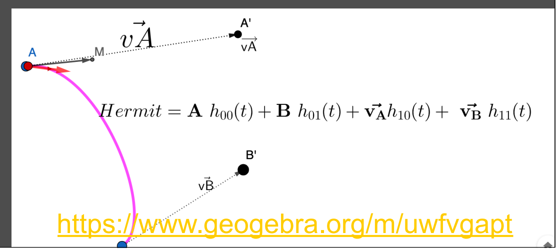
Problems: The curve is attracted to the control points,
but in general, not passing through the points.
Nor does is start nor ends at the first/last control point (boundary conditions)
…but this could be hacked.

Hermit Curve

Input - 2 points A,B
and two directions (vectors) $\overrightarrow{vA}, \overrightarrow{vB}$

Output- a curve that is defined for every $0 \le t \le 1$, and assigns a location and a direction of a 'bug' to each time t.

At every time, the curve is defined by a **convex combination** of these parameters (each with some weight). The weights changes in time

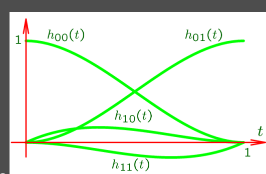


$$Hermit = \mathbf{A}\ h_{00}(t) + \mathbf{B}\ h_{01}(t) + \vec{\mathbf{v}}_{\mathbf{A}} h_{10}(t) + \vec{\mathbf{v}}_{\mathbf{B}}\ h_{11}(t)$$

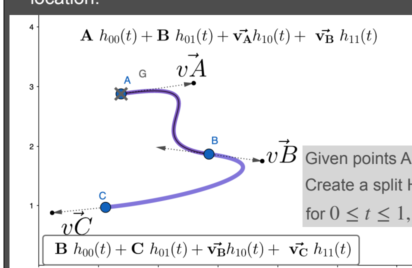
https://www.geogebra.org/m/uwfvgapt

Concatenating Hermit Curves into a Spline

- $h_{00}(t), h_{01}(t)$ that will handle location at beginning of end of the segment.
- $h_{10}(t), h_{10}(t)$ effects the start and end directions.
- Note $h_{00}(0) = 1$ Means that the curve $\mathbf{A}h_{00}(t) + \mathbf{B}h_{01}(t)$ starts at A and ends at B. No need to find a polynomial that will fit only specific location.



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



Given points A,B,C, each with directs direction,
Create a split H such that
for $0 \le t \le 1$, use A, B, and for $1 \le t \le 2$, use B, C

Catmull-Rom Spline do OK

Actually doing excellent for many applications

Not too much control, not too little - back to talk about them later

Big picture - what do we want

The designer could control a small number of points , and a curve connecting theses points is generated.

- Desired Properties:
- Easily controlled - small number of controlled points, and should be easy to predict the effect of each
- Effect should be local and stable (hopefully small change of control parameter ⇒ small change of the curve)
- Locality changes are **near** the control point
- Continuity. C^1 continuity. Geometric continuity (will discuss later)
- Easy to calculate, calculate intersection points etc. (nothing more complicated than cubic)

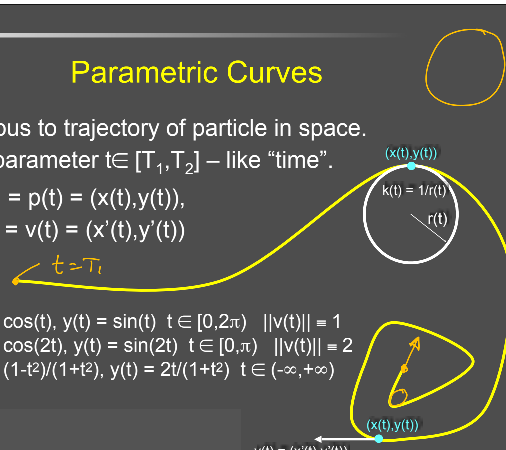
If we want more control, we could construct the curve from segments connecting the control points (e.g. every third control point) ant more control, we

If theses segments starts/stops at control points, they must `glue' nicely.

Lets see how this is done with **Hermite curves.**

Parametric Curves

- Analogous to trajectory of particle in space.
- Single parameter t∈ [T_1,T_2] – like "time".
- position = p(t) = (x(t),y(t)),
  velocity = v(t) = (x'(t),y'(t))
- Circle:
  - x(t) = cos(t), y(t) = sin(t) t∈[0,2π) ||v(t)|| = 1
  - x(t) = cos(2t), y(t) = sin(2t) t∈[0,π) ||v(t)|| = 2
  - x(t) = (1-t²)/(1+t²), y(t) = 2t/(1+t²) t∈(-∞,+∞)



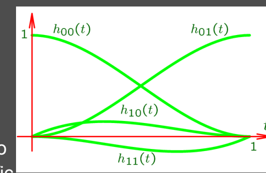
Why cubics (deg=3) ?

Simplicity, locality, finding intersections easily.
At least 4 control points (hence cubic)

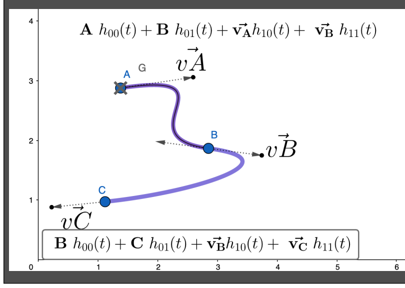
Reason: Want to define curves in 3D.
If we use only 3 control points, the curve will be contained in a plane

The four Cubic Hermite weights function

- $h_{00}(t), h_{01}(t)$ that will handle location at beginning of end of the segment.
- $h_{10}(t), h_{10}(t)$ effects the start and end directions.
- Note $h_{00}(0) = 1$ Means that the curve $\mathbf{A}h_{00}(t) + \mathbf{B}h_{01}(t)$ starts at A and ends at B. No need to find a polynomial that will fit only specific location.

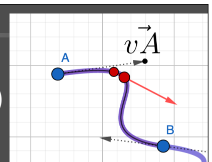


Curve	$h(0)$	$h(1)$	$h'(0)$	$h'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1



What about $h_{1,0}$ and $h_{1,1}$

- What is the direction of the curve $C(t)$ near B ? It is the direction of the vector $\frac{1}{\Delta t}\left(C(t-\Delta t) - C(t)\right)$
- This is just the derivative, which is

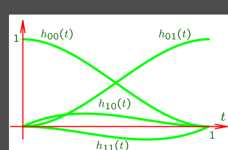


$\mathbf{A}\ h'_{00}(t) + \mathbf{B}\ h'_{01}(t) + \overrightarrow{\mathbf{v_A}} h'_{10}(t) + \overrightarrow{\mathbf{v_B}} h'_{11}(t)$

https://www.geogebra.org/m/z46dmqkp

- Note that $h'_{01}(0) = h'_{00}(0) = 0$, so they cannot change the slope at A

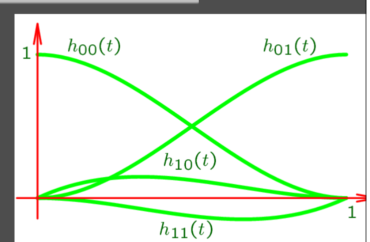
The only function of the four that has non-zero derivative at $t = 0$ is $h'_{10}(0)$. So we could add $h_{10}(t)\ \overrightarrow{vA}$ to the curve,
it will not mess the other properties, since $h_{10}(0) = h_{10}(1) = h'_{10}(0) = 0$



Hermite Cubic Basis

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

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 (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.

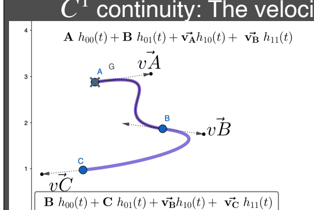
Hermite Cubic Basis (cont'd)

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

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

The segments glue nicely, as long as the velocity vectors are opposite

C^1 continuity: The velocity vector is continuous



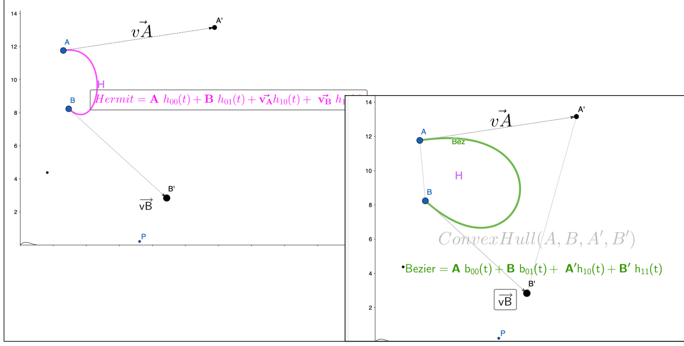
Besier curves

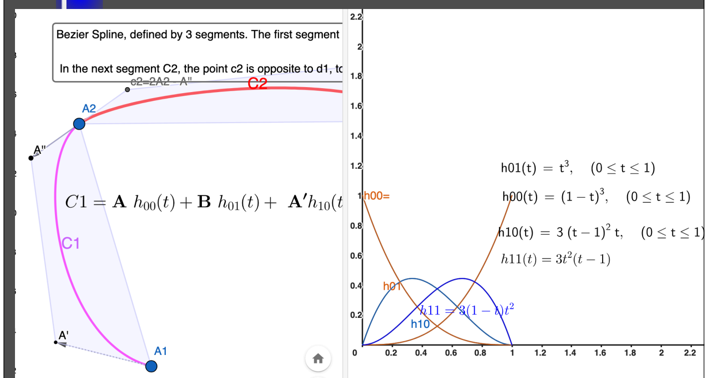
Similar to Hermite. The input is slightly different:

Input to Hermite: Two points and two tangents

Input to Beiser: 2 endspoints and two control points

Sum of weights =1. Always in convex Hull of points



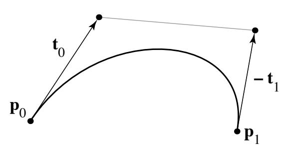


26 https://www.geogebra.org/m/szgz85dn

Hermite to Bézier

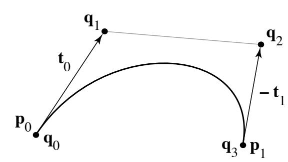
Bezier curve - need 4 control points, but passes through first and last.

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



Hermite to Bézier

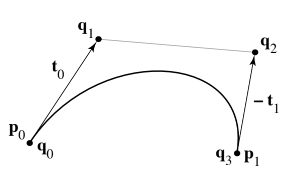
$$\mathbf{p}_0 = \mathbf{q}_0$$
$$\mathbf{p}_1 = \mathbf{q}_3$$
$$\mathbf{t}_0 = 3(\mathbf{q}_1 - \mathbf{q}_0)$$
$$\mathbf{t}_1 = 3(\mathbf{q}_3 - \mathbf{q}_2)$$



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

Hermite to Bézier

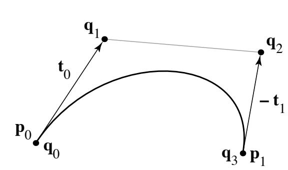
$$\mathbf{p}_0 = \mathbf{q}_0$$
$$\mathbf{p}_1 = \mathbf{q}_3$$
$$\mathbf{t}_0 = 3(\mathbf{q}_1 - \mathbf{q}_0)$$
$$\mathbf{t}_1 = 3(\mathbf{q}_3 - \mathbf{q}_2)$$



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

Hermite to Bézier

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



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

Bézier matrix

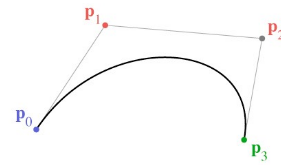
$$\mathbf{f}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

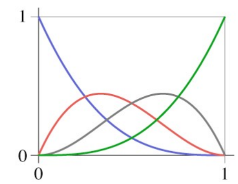
– note that these are the Bernstein polynomials

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

and that defines Bézier curves for any degree

Bézier basis





Another way to Bézier segments

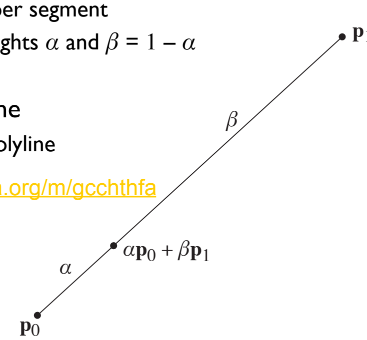
- A really boring spline segment: f(t) = p0
  - it only has one control point
  - the curve stays at that point for the whole time
- Only good for building a *piecewise constant* spline
  - a.k.a. a set of points

https://www.geogebra.org/m/gcchthfa

Another way to Bézier segments

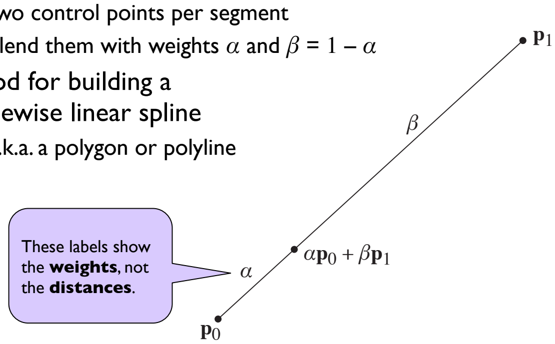
- A piecewise linear spline segment
  - two control points per segment
  - blend them with weights α and $\beta = 1 - \alpha$
- Good for building a piecewise linear spline
  - a.k.a. a polygon or polyline

https://www.geogebra.org/m/gcchthfa



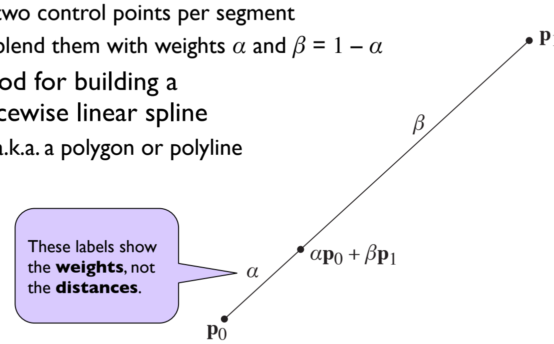
Another way to Bézier segments

- 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!

$$\beta = 1 - \alpha$$

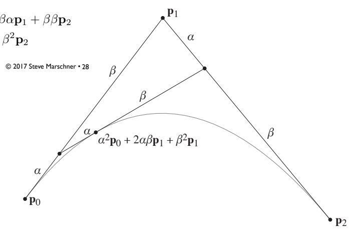
$$\begin{aligned}
\mathbf{p}_{1,0} &= \alpha\mathbf{p}_0 + \beta\mathbf{p}_1 \\
\mathbf{p}_{1,1} &= \alpha\mathbf{p}_1 + \beta\mathbf{p}_2 \\
\mathbf{p}_{2,0} &= \alpha\mathbf{p}_{1,0} + \beta\mathbf{p}_{1,1} \\
&= \alpha\alpha\mathbf{p}_0 + \alpha\beta\mathbf{p}_1 + \beta\alpha\mathbf{p}_1 + \beta\beta\mathbf{p}_2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

Another way to Bézier segments

- 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!

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

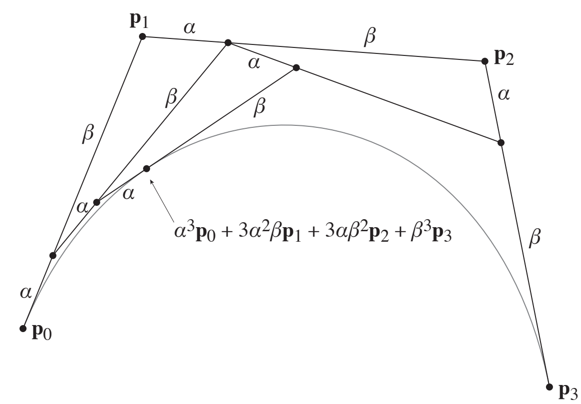
Cornell CS4620 Fall 2017 • Lecture 16 © 2017 Steve Marschner • 28



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

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

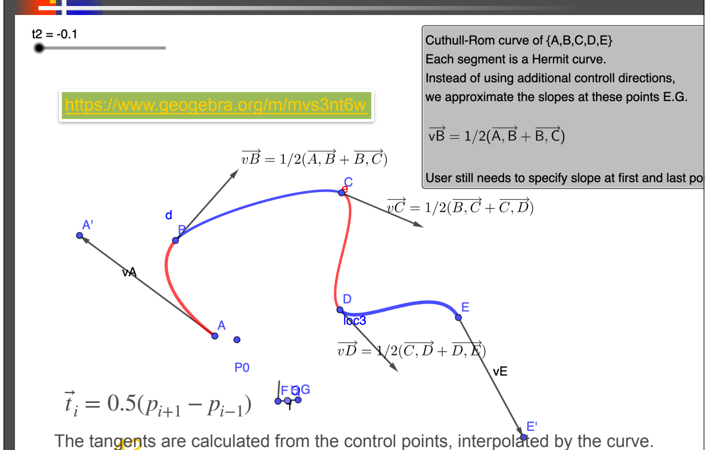


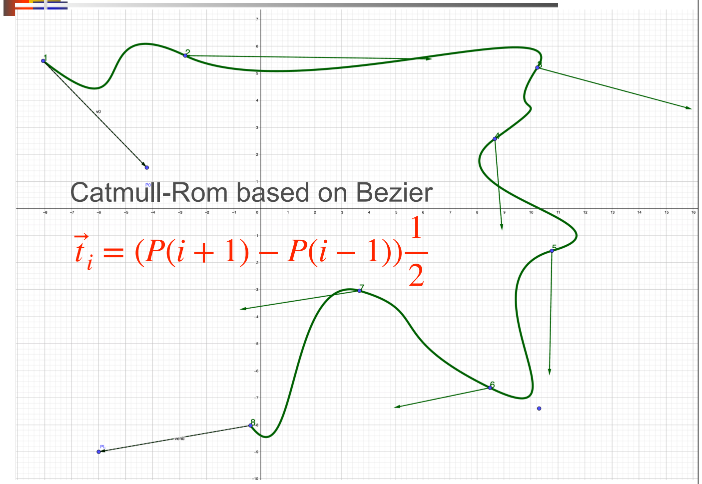
Cathull-Rom curve:

Problem - Bsplines is an approximating curve, but not interpolating one. Not passing through the control points:

Hermit Splines are interpolating, but requires the user to specify tangents at each control points , Bezier splines all requires the user to specify tangent for every control points. Possibly this is fine, possibly annoying.

Catmull-Rom uses Hermits, but tangents are calculated based on other control points.





Catmull-Rom based on Bezier

$$\vec{t}_i = (P(i+1) - P(i-1))\frac{1}{2}$$

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

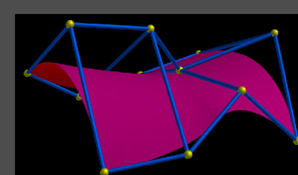
Surface represented implicitly *s(u,v)*

Sphere

$$(\cos u \cos v, \quad \cos u \sin v, \quad \sin u)$$

https://www.geogebra.org/m/gqsg9spt

From Curves to Surfaces (cont'd)



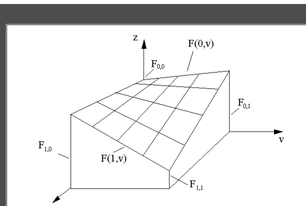
$$\boxed{S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} Q_{ij} B_j(v) B_i(u)}$$

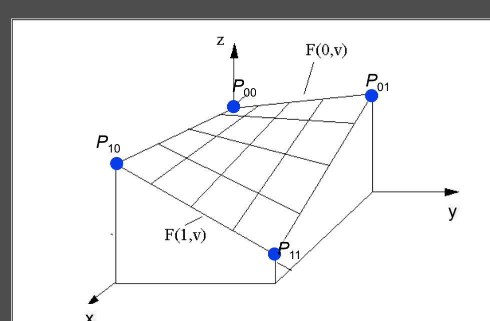
$(x(u,v), y(u,v), z(u,v))$

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:

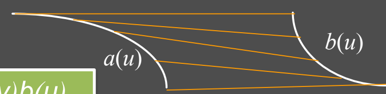
$$P(u,v) = (1-u)(1-v)P_{00} + (1-u)vP_{01} + u(1-v)P_{10} + uvP_{11}$$





Ruled Surfaces

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



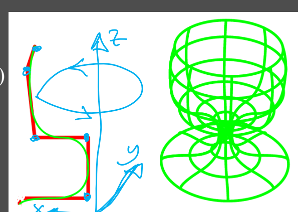
$S(u,v) = v\,a(u) + (1-v)b(u)$

  - 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

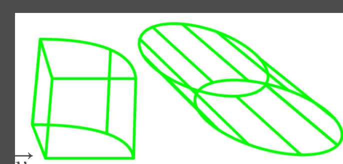
Consider curve $\beta(t) = (\beta_x(t), 0, \beta_z(t))$ and let Z be the axis of revolution. Then,



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

Extrusion

- Extrusion of a, usually planar, curve along a linear segment.
- Consider curve $\beta(t)$ and vector $\vec{v}$

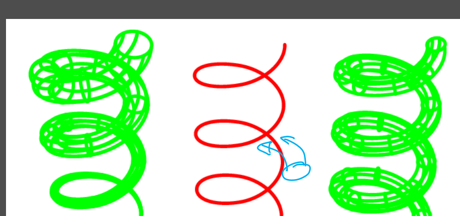


- Then

$$t' \cdot \vec{v} + \beta(t), \qquad 0 \le t, t' \le 1,$$

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