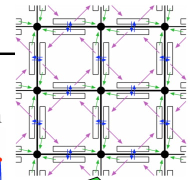
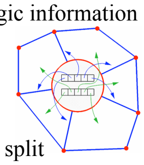
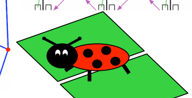
Spline Curves

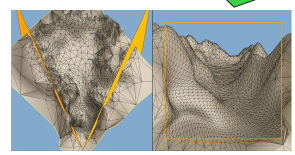
Last Time?







- Adjacency Data Structures
  - Geometric & topologic information
  - Dynamic allocation
  - Efficiency of access
- Mesh Simplification
  - edge collapse/vertex split
  - geomorphs
  - progressive transmission
  - view-dependent refinement

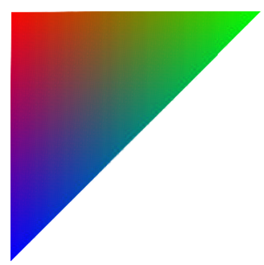


Today

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

Color Interpolation

- Interpolate colors of the 3 vertices
- Linear interpolation, barycentric coordinates

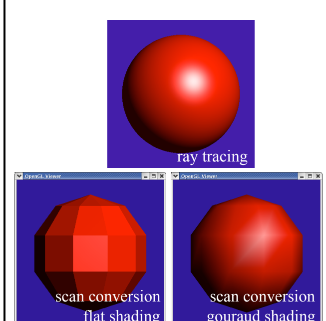


```
glBegin(GL_TRIANGLES);
glColor3f(1.0,0.0,0.0);
glVertex3f(…);
glColor3f(0.0,1.0,0.0);
glVertex3f(…);
glColor3f(0.0,0.0,1.0);
glVertex3f(…);
glEnd();
```

glShadeModel (GL_SMOOTH);

- From OpenGL Reference Manual:
  - Smooth shading, the default, causes the computed colors of vertices to be interpolated as the primitive is rasterized, typically assigning different colors to each resulting pixel fragment.
  - Flat shading selects the computed color of just one vertex and assigns it to all the pixel fragments generated by rasterizing a single primitive.
  - In either case, the computed color of a vertex is the result of lighting if lighting is enabled, or it is the current color at the time the vertex was specified if lighting is disabled.

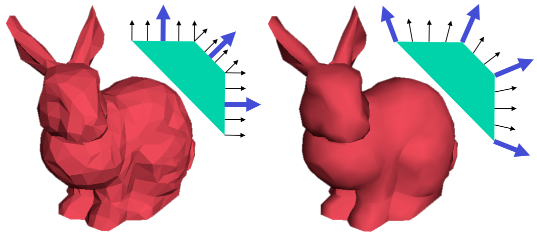
Normal Interpolation




```
glBegin(GL_TRIANGLES);
glNormal3f(…);
glVertex3f(…);
glNormal3f(…);
glVertex3f(…);
glNormal3f(…);
glVertex3f(…);
glEnd();
```

Gouraud Shading

- Instead of shading with the normal of the triangle, we'll shade the vertices with the *average normal* and *interpolate the shaded color* across each face



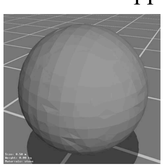
- How do we compute Average Normals? Is it expensive??

Today

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

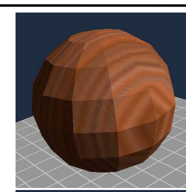
Limitations of Polygonal Meshes

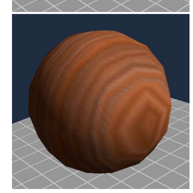
- Planar facets (& silhouettes)
- Fixed resolution
- Deformation is difficult
- No natural parameterization (for texture mapping)



Gouraud not always good enough

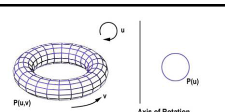
- Still low, fixed resolution (missing fine details)
- Still have polygonal silhouettes
- Intersection depth is planar (e.g. ray tracing visualization)
- Collisions problems for simulation
- Solid Texturing problems
- ...



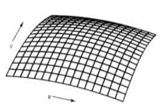


Some Non-Polygonal Modeling Tools

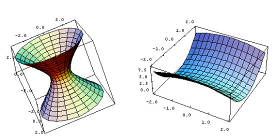
Extrusion



Surface of Revolution

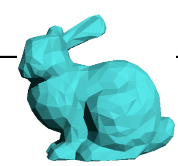


Spline Surfaces/Patches



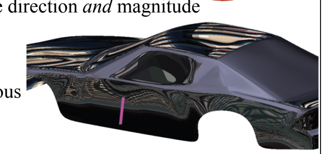
Quadrics and other implicit polynomials

Continuity definitions:



- C^0 continuous
  - curve/surface has no breaks/gaps/holes
- G^1 continuous
  - tangent at joint has same direction
- C^1 continuous
  - curve/surface derivative is continuous
  - tangent at joint has same direction *and* magnitude

- C^n continuous
  - curve/surface through n^{th} derivative is continuous
  - important for shading



"Shape Optimization Using Reflection Lines", Tosun et al., 2007

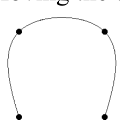
Questions?

Today

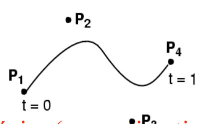
- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

Definition: What's a Spline?

- Smooth curve defined by some control points
- Moving the control points changes the curve

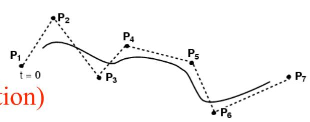


Interpolation

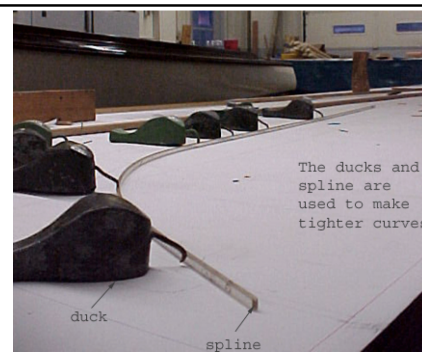


Bézier (approximation)

BSpline (approximation)

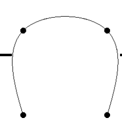


Interpolation Curves / Splines



www.abm.org

Interpolation Curves

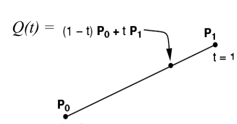


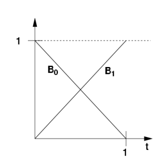
- Curve is constrained to pass through all control points
- Given points P_0, P_1, ... P_n, find lowest degree polynomial which passes through the points

$$x(t) = a_{n-1}t^{n-1} + ... + a_2t^2 + a_1t + a_0$$
$$y(t) = b_{n-1}t^{n-1} + ... + b_2t^2 + b_1t + b_0$$

Linear Interpolation

- Simplest "curve" between two points





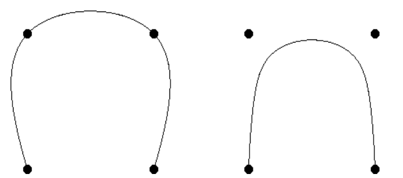
Spline Basis Functions

a.k.a. Blending Functions

$$Q(t) = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix} = \begin{pmatrix} (P_0) & (P_1) \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

$$Q(t) = \mathbf{GBT(t)} = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

Interpolation vs. Approximation Curves

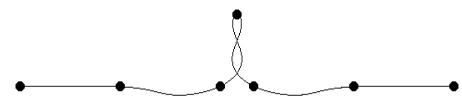


Interpolation
curve must pass through control points

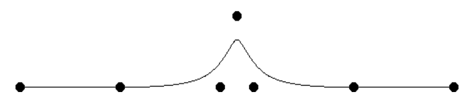
Approximation
curve is influenced by control points

Interpolation vs. Approximation Curves

- Interpolation Curve – over constrained → lots of (undesirable?) oscillations



- Approximation Curve – more reasonable?



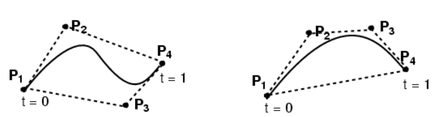
Questions?

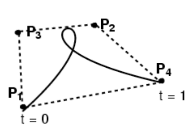
Today

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $\mathbf{P}_1$ to $(\mathbf{P}_2\text{-}\mathbf{P}_1)$ and at $\mathbf{P}_4$ to $(\mathbf{P}_4\text{-}\mathbf{P}_3)$

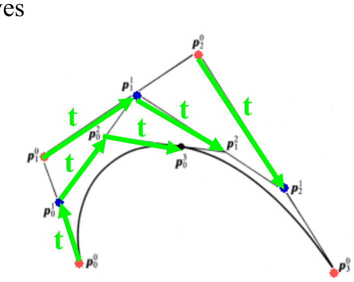




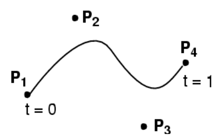
A Bézier curve is bounded by the convex hull of its control points.

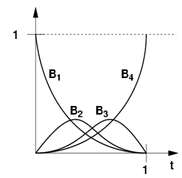
Cubic Bézier Curve

- de Casteljau's algorithm for constructing Bézier curves



Cubic Bézier Curve





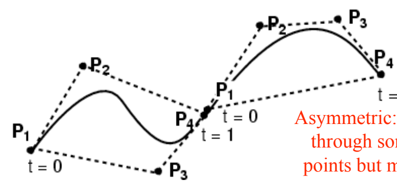
$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$$

$$Q(t) = \mathbf{GBT(t)} \qquad B_{Bezier} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Bernstein Polynomials

$$B_1(t) = (1-t)^3;\ B_2(t) = 3t(1-t)^2;\ B_3(t) = 3t^2(1-t);\ B_4(t) = t^3$$

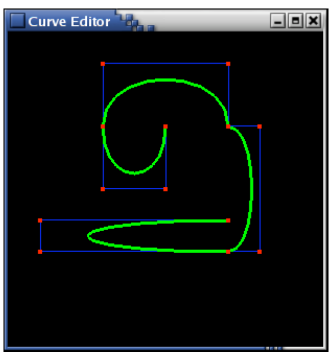
Connecting Cubic Bézier Curves



Asymmetric: Curve goes through some control points but misses others

- How can we guarantee C^0 continuity?
- How can we guarantee G^1 continuity?
- How can we guarantee C^1 continuity?
- Can't guarantee higher C^2 or higher continuity

Connecting Cubic Bézier Curves



- Where is this curve
  - C^0 continuous?
  - G^1 continuous?
  - C^1 continuous?
- What's the relationship between:
  - the # of control points, and
  - the # of cubic Bézier subcurves?

Higher-Order Bézier Curves

- > 4 control points
- Bernstein Polynomials as the basis functions

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \qquad 0 \le i \le n$$

- Every control point affects the entire curve
  - Not simply a local effect
  - More difficult to control for modeling

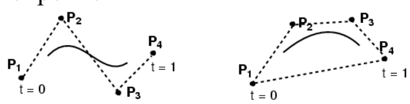
Questions?

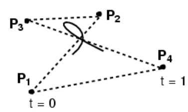
Today

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

Cubic BSplines

- ≥ 4 control points
- Locally cubic
- Curve is not constrained to pass through any control points

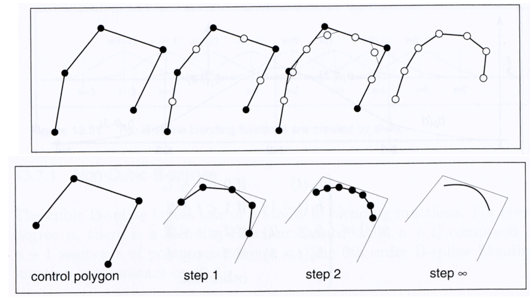




A BSpline curve is also bounded by the convex hull of its control points.

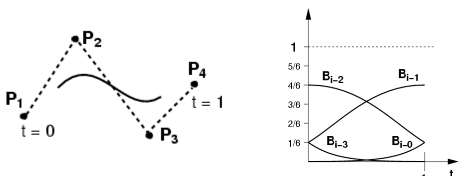
Cubic BSplines

- Iterative method for constructing BSplines



Shirley, Fundamentals of Computer Graphics

Cubic BSplines

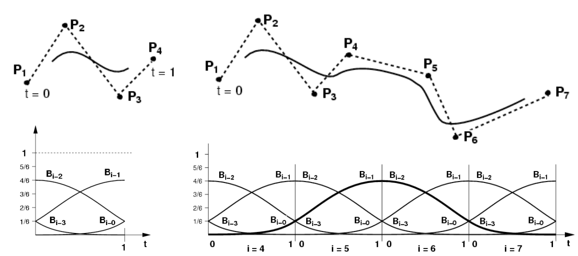


$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_i$$

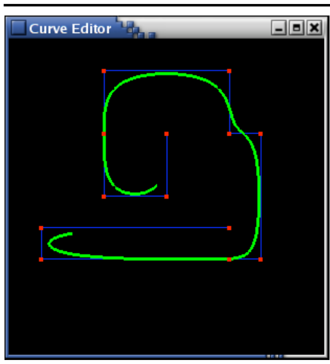
$$Q(t) = \mathbf{GBT(t)} \qquad B_{B-Spline} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Connecting Cubic BSpline Curves

- Can be chained together
- Better control locally (windowing)

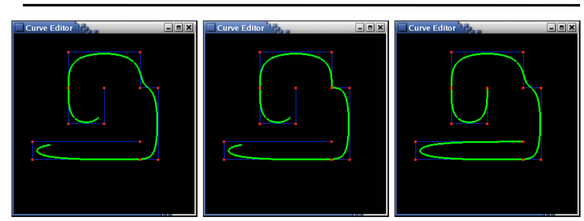


Connecting Cubic BSpline Curves



- What's the relationship between
  - the # of control points, and
  - the # of cubic BSpline subcurves?

BSpline Curve Control Points



Default BSpline

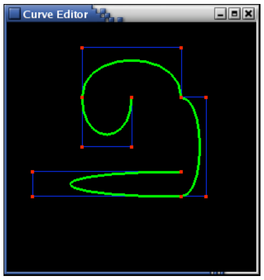
BSpline with Discontinuity

Repeat interior control point

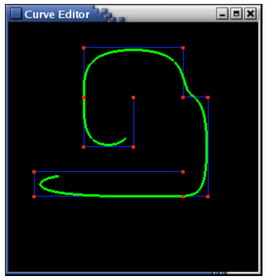
BSpline which passes through end points

Repeat end points

Bézier is not the same as BSpline



Bézier

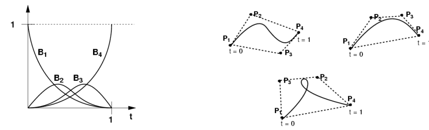


BSpline

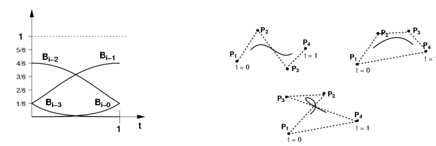
Bézier is not the same as BSpline

- Relationship to the control points is different

Bézier



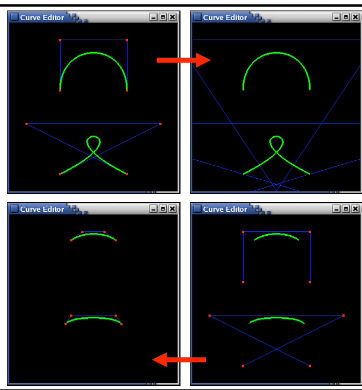
BSpline



Converting between Bézier & BSpline

original control points as Bézier

new BSpline control points to match Bézier



new Bézier control points to match BSpline

original control points as BSpline

Converting between Bézier & BSpline

- Using the basis functions:

$$B_{Bezier} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$B_{B-Spline} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$Q(t) = \mathbf{GBT(t)} = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

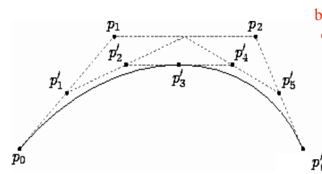
NURBS (generalized BSplines)

- BSpline: uniform cubic BSpline

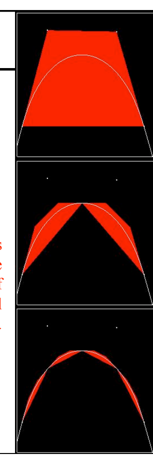
- NURBS: Non-Uniform Rational BSpline
  - non-uniform = different spacing between the blending functions, a.k.a. knots
  - rational = ratio of polynomials (instead of cubic)

Neat Bezier Spline Trick

- A Bezier curve with 4 control points:
  - $P_0 \quad P_1 \quad P_2 \quad P_3$
- Can be split into 2 new Bezier curves:
  - $P_0 \quad P'_1 \quad P'_2 \quad P'_3$
  - $P'_3 \quad P'_4 \quad P'_5 \quad P_3$



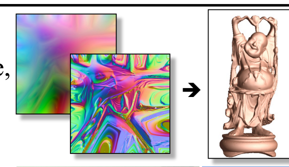
A Bézier curve is bounded by the convex hull of its control points.



Questions?

Readings for Today *(pick one)*

- "Geometry Images", Gu, Gortler, & Hoppe, SIGGRAPH 2002



- "Teddy: A Sketching Interface for 3D Freeform Design", Igarashi et al., SIGGRAPH 1999

- Post a comment or question on the LMS discussion by 10am on Tuesday

Reading for Friday (2/4)

- DeRose, Kass, & Truong, "Subdivision Surfaces in Character Animation", SIGGRAPH 1998

- Post a comment or question on the LMS discussion by 10am on Friday