Curve Surfaces

CS5500 Computer Graphics
May 11, 2006

Examples of Curve Surfaces

- Spheres
- The body of a car
- Almost everything in nature

Representations

- Simple (or “explicit”) functions.
- Implicit functions.
- Parametric functions.

Explicit Functions

- For example: $z = f(x, y)$
  - Independent variables: x and y
  - Dependent variable: z
- Easy to render:
  - For the above, loop over x and y.
- But too limited:
  - For example, how do you describe a sphere centered at the origin?
  - $z = (r^2-x^2-y^2)^{1/2}$ gives us the upper hemisphere only.

Implicit Functions

- $0 = f(x, y, z)$
  - All variables are independent variables.
- Sphere: $x^2+y^2+z^2-r^2 = 0$
- More powerful than explicit functions, but harder to render.

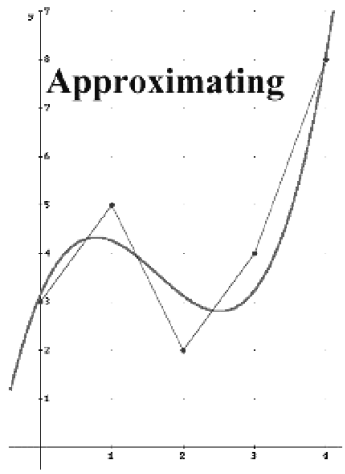
Parametric Functions

- $x = f_x(u, v)$
- $y = f_y(u, v)$
- $z = f_z(u, v)$
- Cubic curve: $p(u) = c_0+c_1u+c_2u^2+c_3u^3$
- Sphere:
  - $x = r \cos(u)\cos(v)$
  - $y = r \sin(u)\cos(v)$
  - $z = r \sin(v)$
- To render it, loop over u and v.

But, how do we design or specify a surface?

Control Points

- Like bending a piece of wood, we control its shape at some control points.
- Some control points lie on the curve and some don't. Those lie on the curves are called knots.

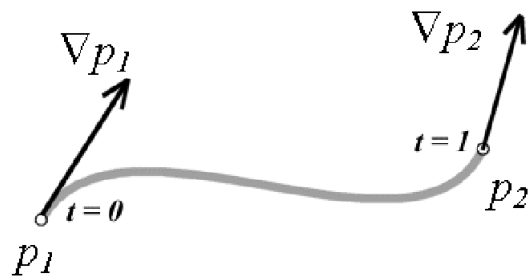


Interpolation

- Let $p(u) = c_0+c_1u+c_2u^2+c_3u^3$
- Given 4 control points p_0, p_1, p_2, p_3, we may make p(u) pass through all of them at u=0, 1/3, 2/3, 1.
- See Section 10.4 for the derivation of c = $[c_0, c_1, c_2, c_3]^T$

Hermite Specification

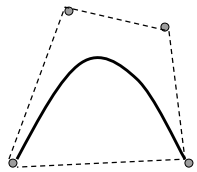
- Specify a curve by two knots and two tangent vectors at the endpoints.



Hermite Specification

Bezier Curve

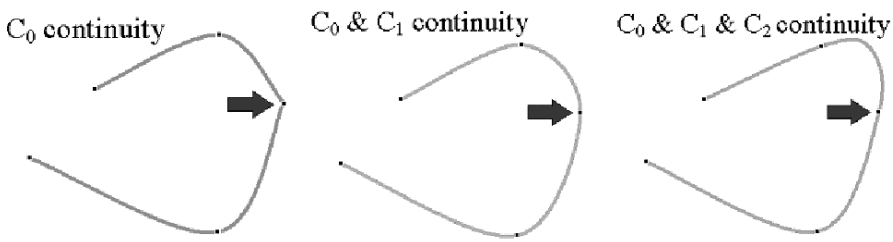
- Instead of interpolating all 4 control points (p_0, p_1, p_2, p_3), p_1 and p_2 controls the tangents at p_0 and p_3.
- The curve lies in the convex hull of the four control points.



$$p(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

Piecewise Curve Segments

- For curves with more than 4 control points, we may either:
  - Increase the degree of polynomials, or
  - Join piecewise segments.
- Do pieces meet smoothly at the join points?



C_n vs G_n Continuity

- C_n means continuity at n-th derivative.
- G_n doesn't require the exact match of n-th derivatives at the joint, just being proportional.
- The tangents point in the same direction, but they may have different magnitudes.

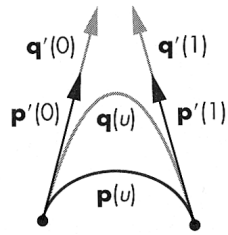


Figure 10.16 Change of magnitude in G^1 continuity.

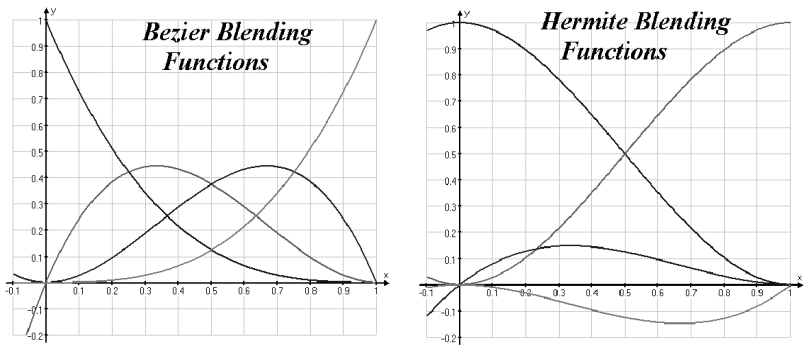
B-Spline

- If we don't require the curve to pass through any control point, we may have more control at the join points.
- To define the curve between p_i and p_{i+1}, use also p_{i-1} and p_{i+2}

NURBS

- Non-uniform Rational B-Spline.
- In NURBS, we may use the weights to change the importance of a control point.
- We won't discuss it in depth here. For details, see Sections 10.8.

Blending Polynomial



Q: What are the blending polynomials for interpolation? (A: See P.488, Fig 10.11)

For A More Formal Discussion

- The above discussion is aimed at stimulating your interest.
- For a more formal discussion, especially if you're interested in researches in these areas, see Angel's Sections 10.2 to 10.7
- [Bonus] Tensor-product surfaces are mentioned in 10.4.2

But, the graphics hardware knows triangles only…

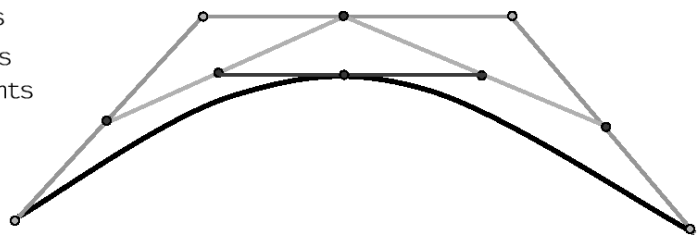
Tessellation

- Curve surfaces can be approximated by (a lot of) polygons for the purpose of rendering.
- The tessellation may be static (done before rendering) or dynamic (during rendering).

Subdivision

- For example, the “de Casteljau” algorithm for rendering Bezier Splines

1. Find midpoints of support
2. Connect with new segments
3. Find midpoints of new segments
4. Connect with new segment
5. Find its midpoint



Curves and Surfaces in OpenGL

- OpenGL supports curves and surfaces through evaluators.
- OpenGL Utility library, GLU, provides a set of NURBS functions.
- For more information:
  - See Angel's section 10.12
- The Utah teapot is available as an object in GLUT.