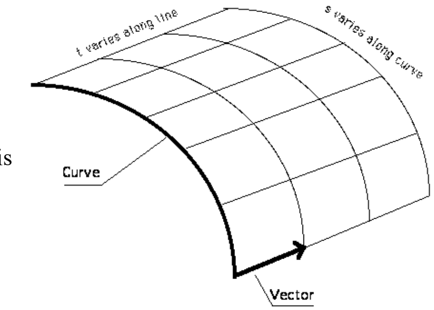
Generating Surfaces

- We can construct surfaces from curves in a variety of user intuitive ways
  - Extruded surfaces
  - Generalized cones
  - Surfaces of revolution
  - Sweeping (generalized cylinders)
- In many the examples that follow, we will assume that we know how to generate a 3D parameteric curve (studied later)
  - e.g. twisted cubic as (t, t*t, t*t*t)

Extruded surfaces

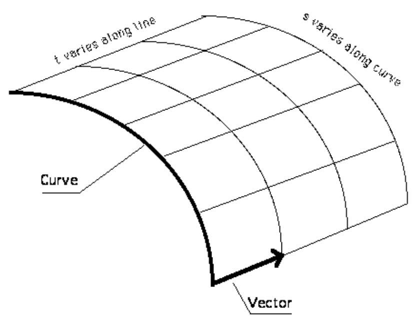
- Geometrical model - Pasta machine
- Take curve and "extrude" surface along vector
- Many human artifacts have this form - rolled steel, etc.



Parametric formula?

Extruded surfaces

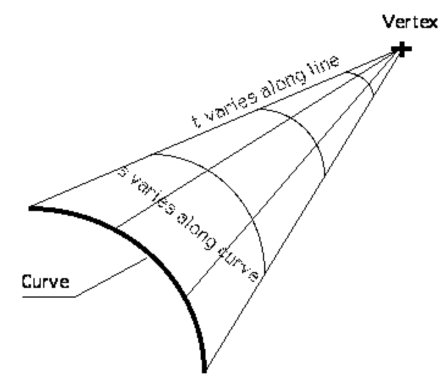
- Geometrical model - Pasta machine
- Take curve and "extrude" surface along vector
- Many human artifacts have this form - rolled steel, etc.



$$(x(s,t), y(s,t), z(s,t)) = (x_c(s), y_c(s), z_c(s)) + t(v_0, v_1, v_2)$$

Cones

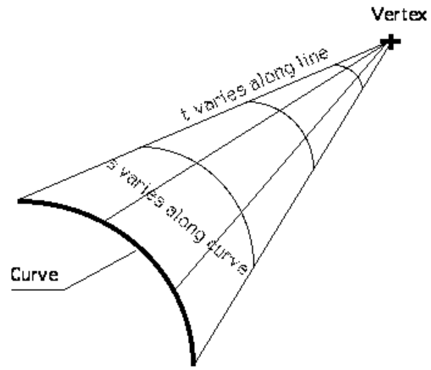
- From every point on a curve, construct a line segment through a single fixed point in space - the vertex
- Curve can be space or plane curve, but shouldn't pass through the vertex



Parametric formula?

Cones

- From every point on a curve, construct a line segment through a single fixed point in space - the vertex
- Curve can be space or plane curve, but shouldn't pass through the vertex

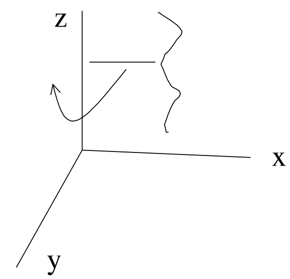


$$(x(s,t), y(s,t), z(s,t)) = (1-t)(x_c(s), y_c(s), z_c(s)) + t(v_0, v_1, v_2)$$

Surfaces of revolution

- Plane curve + axis
- "spin" plane curve around axis to get surface
- Choice of plane is arbitrary, choice of axis affects surface
- In the example to the right, curve is on x-z plane, axis is z axis.
- So curve is $(x_c(s), z_c(s))$

Parametric formula?

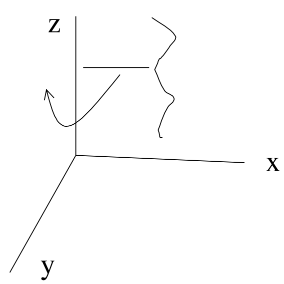


Surfaces of revolution

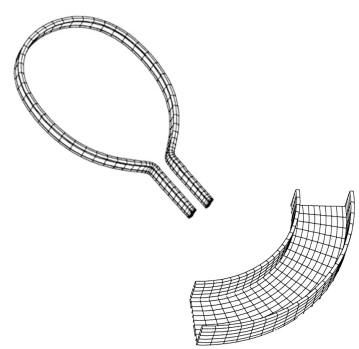
- Plane curve + axis
- "spin" plane curve around axis to get surface
- Choice of plane is arbitrary, choice of axis affects surface
- In the example to the right, curve is on x-z plane, axis is z axis. (Think of $x_c(s)$ as a radius)

$$(x(s,t), y(s,t), z(s,t)) =$$

$$(x_c(s)\cos(t), x_c(s)\sin(t), z_c(s))$$



Sweeps/Generalized Cylinders



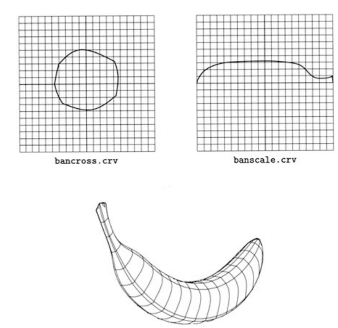
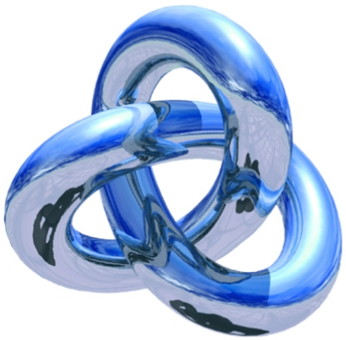


Figure 3.8: Banana example. A banana is represented by an affine transformation surface. The cross section is scaled, translated along z from -1 to 1, and rotated around the y axis.

[Synder 92, via CMU course page]

Sweeps/Generalized Cylinders

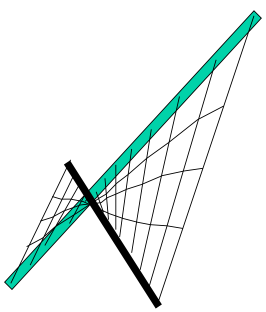


MetaCreations, via CMU course page

Ruled surfaces -1

- Popular, because it's easy to build a curved surface out of straight segments - e.g. pavilions, etc.
- Take two space curves, and join corresponding points—same *s* parameter value—with line segment.
- Even if space curves are lines, the surface is usually curved.

Ruled Surfaces - 2



Easy to explain,
hard to draw!

Ruled surfaces -3

Parameterized form

$$(x(s,t), y(s,t), z(s,t)) = (1-t)(x_1(s), y_1(s), z_1(s)) + t(x_2(s), y_2(s), z_2(s))$$

Normals

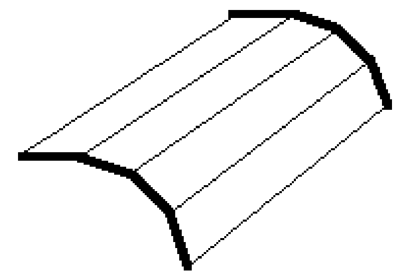
- Normal is cross product of tangent in t direction and s direction.

$$\left(\frac{\delta x}{\delta t},\frac{\delta y}{\delta t},\frac{\delta z}{\delta t}\right)\times\left(\frac{\delta x}{\delta s},\frac{\delta y}{\delta s},\frac{\delta z}{\delta s}\right)$$

- Examples
  - Cylinder: normal is cross-product of curve tangent and direction vector
  - Surface of revolution: take curve normal and spin round axis

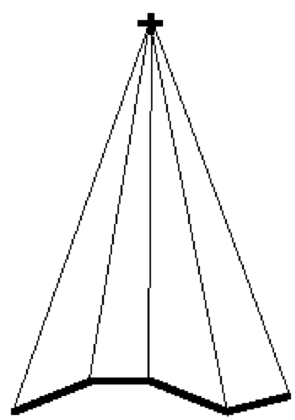
Rendering

- Cylinders: small steps along curve, straight segments along t generate polygons; exact normal is known.



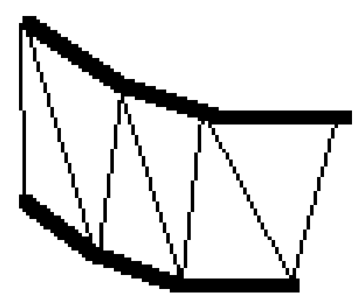
Rendering

- Cone: small steps in s generate straight edges, join with vertex to get triangles, normals known exactly except at vertex.



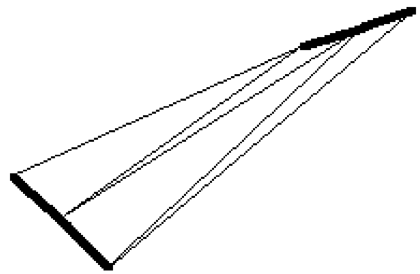
Rendering

- Surface of revolution: small steps in s generate strips, small steps in t along the strip generate edges; join up to form triangles. Normals known exactly.



Rendering

- Ruled surface: steps in s generate polygons, join opposite sides to make triangles - otherwise “non planar polygons” result. Normals known exactly.
- **Must** understand why rectangular sections do not work!



Specifying Curves from Points

- Want to modulate curves via “control” points.
- Strategy depends on application. Possibilities:
  - Force a polynomial of degree N-1 through N points (Lagrange interpolate)
  - Specify a combination of “anchor” points and derivatives (Hermite interpolate)
  - Other “blends” (Bezier, B-splines)--more useful than Lagrange/Hermite

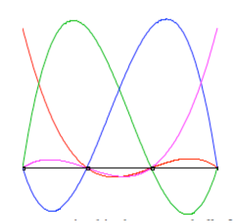
Specifying Curves from Points-II

- Issues:
  - Continuity of curve and derivatives (geometric, parametric)
  - Local versus global control
  - Polynomials verses other forms
  - Higher polynomial degree versus stitching lower order polynomials together
  - Polynomial degree (usually 3--fewer is not flexible enough, and higher gives hard to control wiggles).
  - It is relatively easy to fit a curve through points in explicit form, but we will use parametric form as it more useful in graphics.

Lagrange Interpolate (degree 3)

- Want a parametric curve that passes through (interpolates) four points.
- Use the points to combine four Lagrange polynomials (blending functions)
- As the parameter goes through each of 4 particular values, one blending function is 1, and the other 3 are zero.

$$\sum_{i \in \text{points}} p_i \phi_i^{(l)}(t)$$

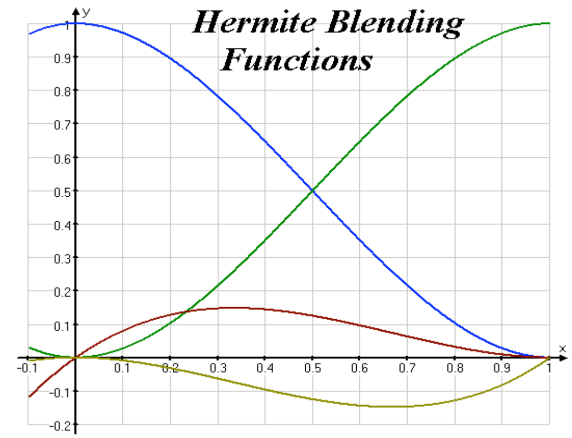


Hermite (H&B, page 426)

- Hermite interpolate
  - Curve passes through specified points **and** has specified derivatives at those points.

- Standard degree 3 case: 2 points, 2 derrivatives at those points
- 4 functions of degree 3, two each of two kinds
  - one at an endpoint, zero at the other, AND derrivative is zero at both
  - derivative is one at an endpoint and zero at others, AND value is zero at the endpoints.

$$\sum_{i\in\text{points}} p_i \phi_i^{(h)}(t) + \sum_{i\in\text{points}} v_i \phi_i^{(hd)}(t)$$



From www.cs.virginia.edu/~gfx/Courses/ 2002/Intro.fall.02