Surfaces and solids

CS 4620 Lecture 19

Modeling in 3D

- Representing subsets of 3D space
  - volumes (3D subsets)
  - surfaces (2D subsets)
  - curves (1D subsets)
  - points (0D subsets)

Representing geometry

- In order of dimension…
- Points: trivial case
- Curves
  - normally use parametric representation
  - line—just a point and a vector (like ray in ray tracer)
    - polylines (approximation scheme for drawing)
  - more general curves: usually use splines
    - **p**(t) is from R to R^3
    - **p** is defined by piecewise polynomial functions

Representing geometry

- Surfaces
  - this case starts to get interesting
  - implicit and parametric representations both useful
  - example: plane
    - implicit: vector from point perpendicular to normal
    - parametric: point plus scaled tangents
  - example: sphere
    - implicit: distance from center equals *r*
    - parametric: write out in spherical coordinates
      - messiness of parametric form not unusual

Specific surface representations

- Parametric spline surfaces
  - extrusions
  - surfaces of revolution
  - generalized cylinders
  - spline patches

- Pause for differential geometry review…
  - plane and space curves, tangent vectors
  - parametric surfaces, isolines, tangent vectors, normals

From curves to surfaces

- So far have discussed spline curves in 2D
  - it turns out that this already provides of the mathematical machinery for several ways of building curved surfaces
- Building surfaces from 2D curves
  - extrusions and surfaces of revolution
- Building surfaces from 2D and 3D curves
  - generalized swept surfaces
- Building surfaces from spline patches
  - generalizing spline curves to spline patches
- Also to think about: generating triangles

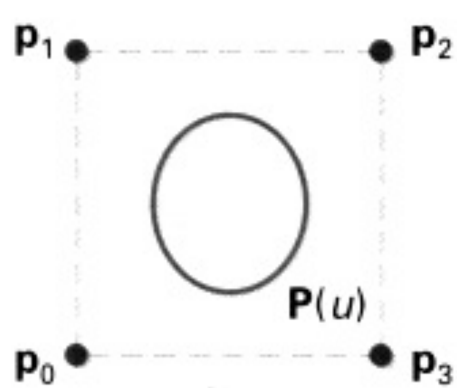
Extrusions

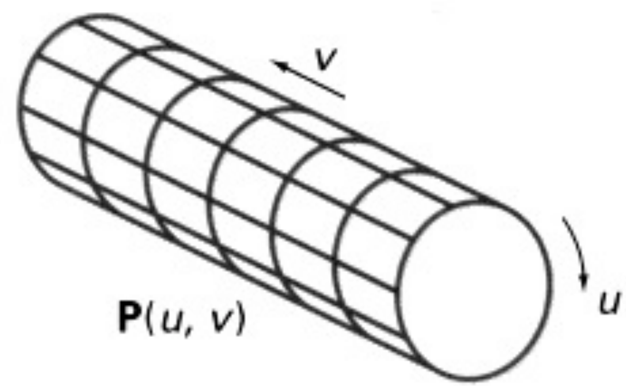
- Given a spline curve $C \in \mathbb{R}^2$, define $S \in \mathbb{R}^3$ by

$$S = C \times [a, b]$$

- This produces a “tube” with the given cross section
  - Circle: cylinder; “L”: shelf bracket; “I”: I beam
- It is parameterized by the spline *t* and the interval [*a*, *b*]

$$s(t, s) = [c_x(t), c_y(t), s]^T$$



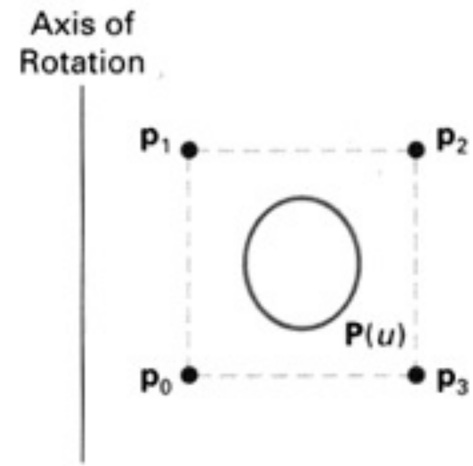


Surfaces of revolution

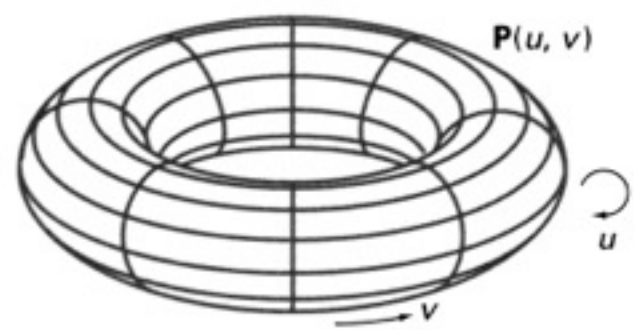
- Take a 2D curve and spin it around an axis
- Given curve **c**(t) in the plane, the surface is defined easily in cylindrical coordinates:

$$\mathbf{s}(t, s) = (r, \phi, z) = (c_x(t), s, c_y(t))$$

  – the torus is an example in which the curve **c** is a circle



(a)

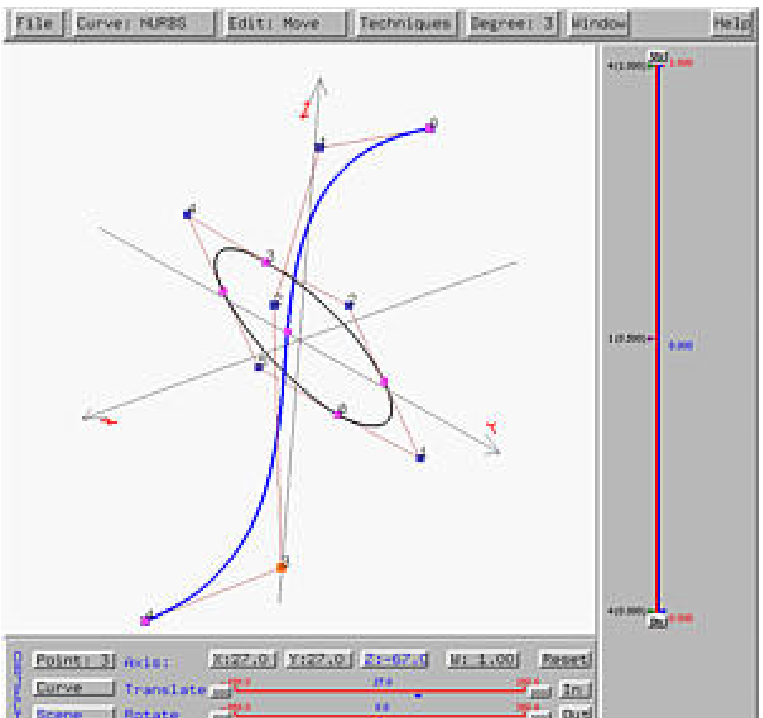


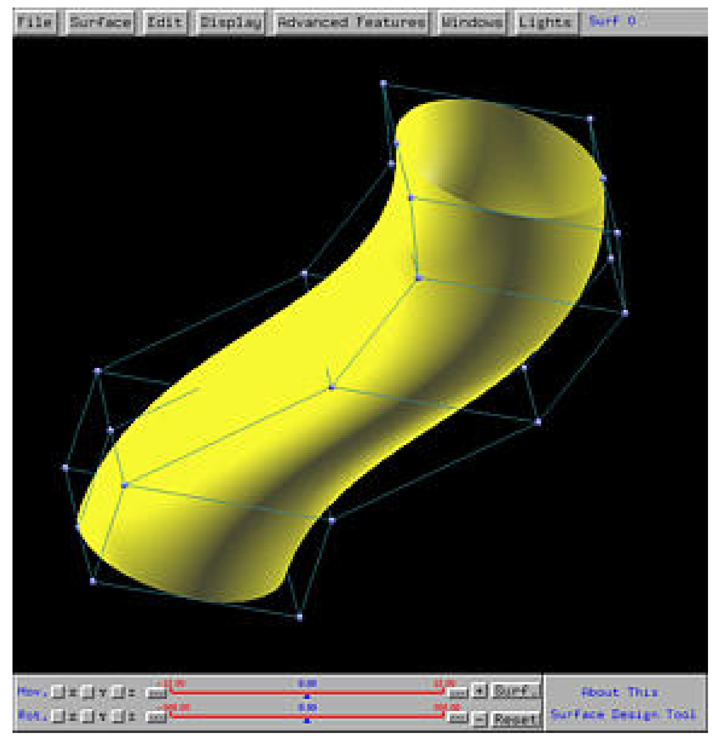
(b)

[Hearn & Baker]

Swept surfaces

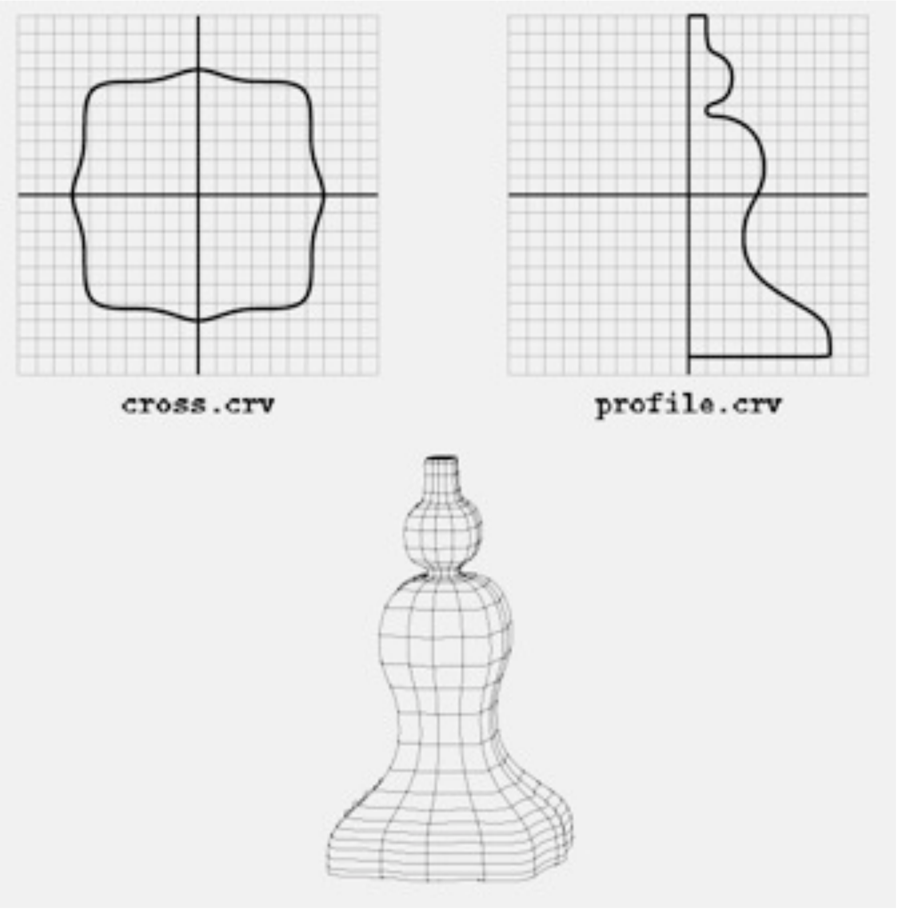
- Surface defined by a *cross section* moving along a *spine*
- Simple version: a single 3D curve for spine and a single 2D curve for the cross section





Generalized cylinders

- General swept surfaces
  - varying radius
  - varying cross-section
  - curved axis

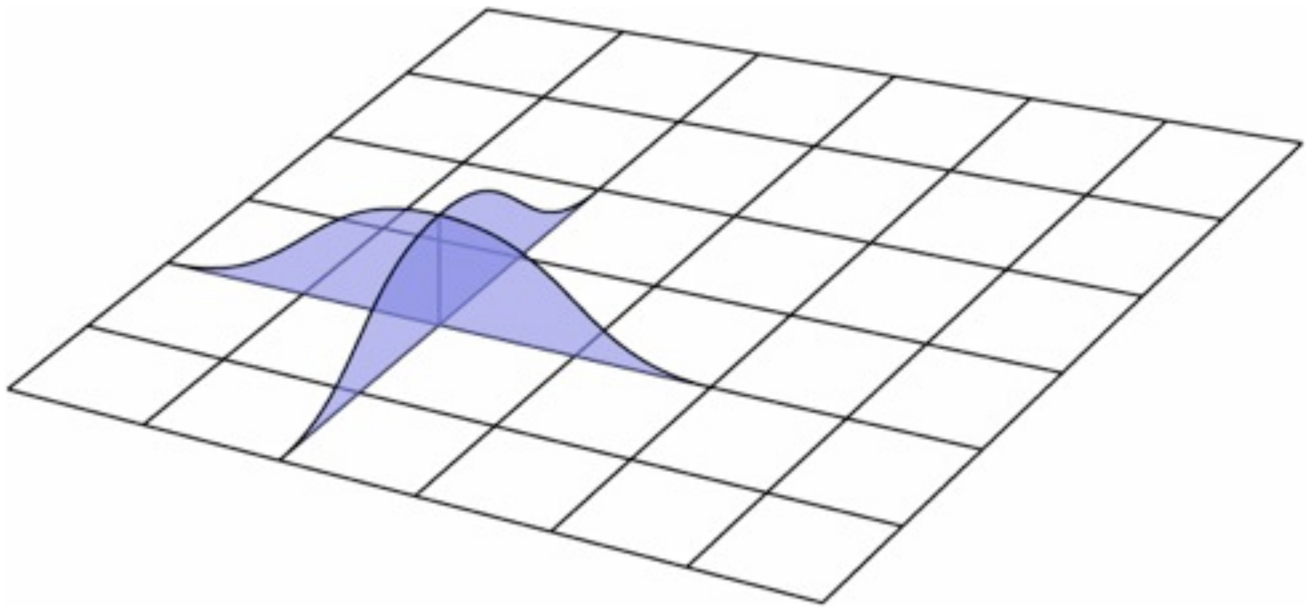


[Snyder 1992]

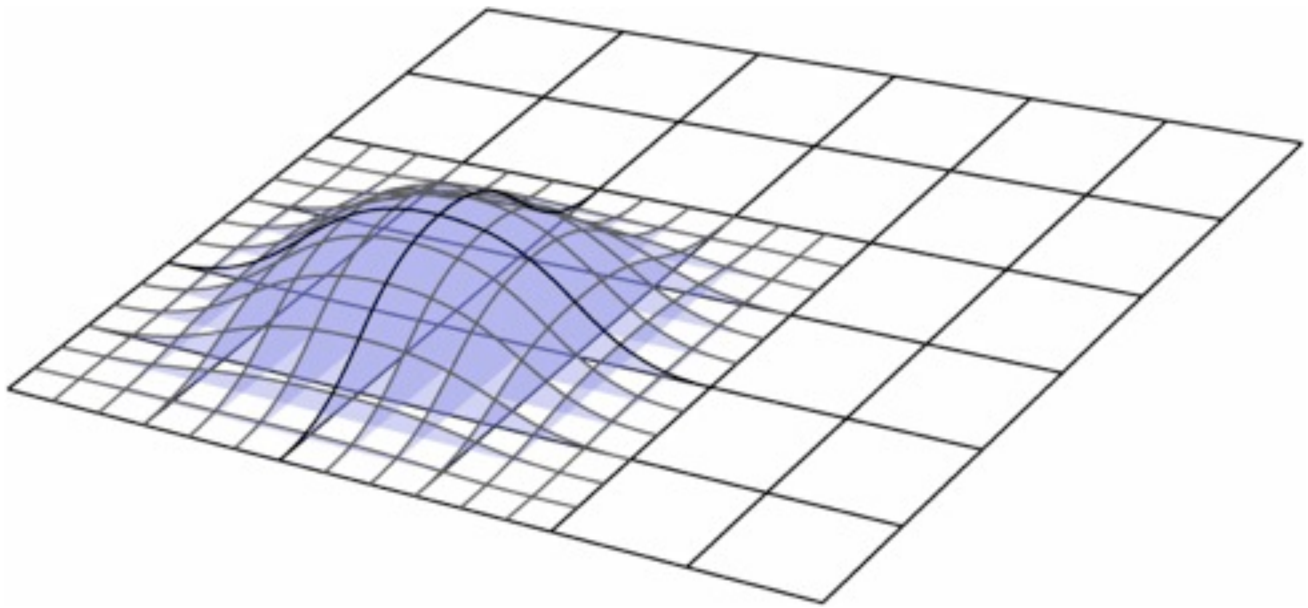
From curves to surface patches

- Curve was sum of weighted 1D basis functions
- Surface is sum of weighted 2D basis functions
  - construct them as separable products of 1D fns.
  - choice of different splines
    - spline type
    - order
    - closed/open (B-spline)

Separable product construction

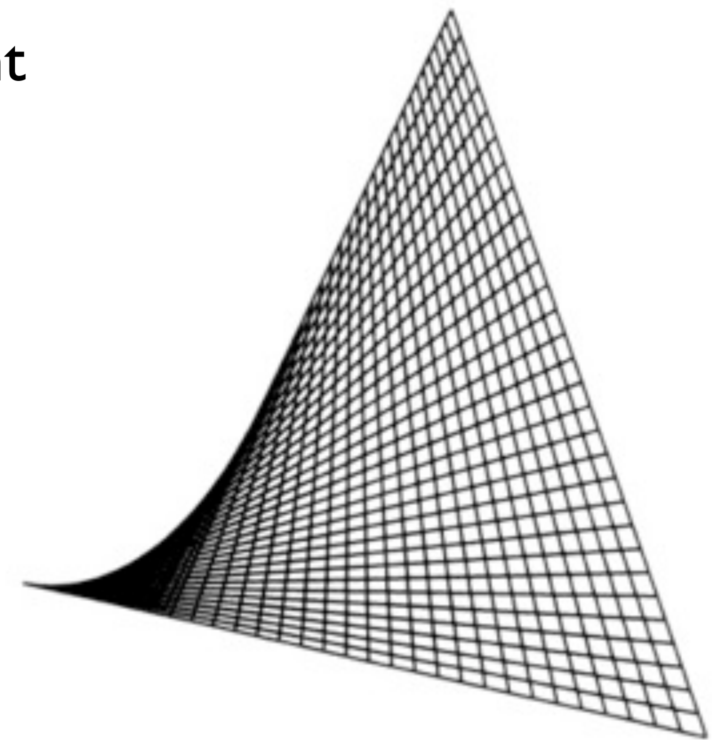


Separable product construction



Bilinear patch

- Simplest case: 4 points, cross product of two linear segments
  - basis function is a 3D tent



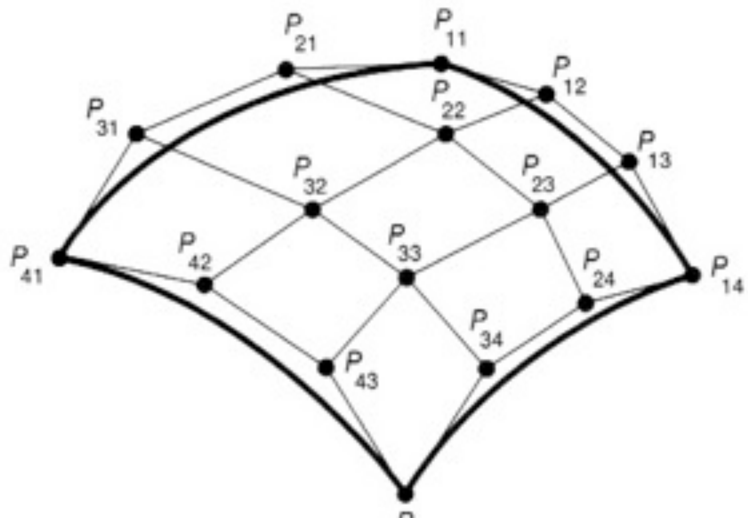
(c)

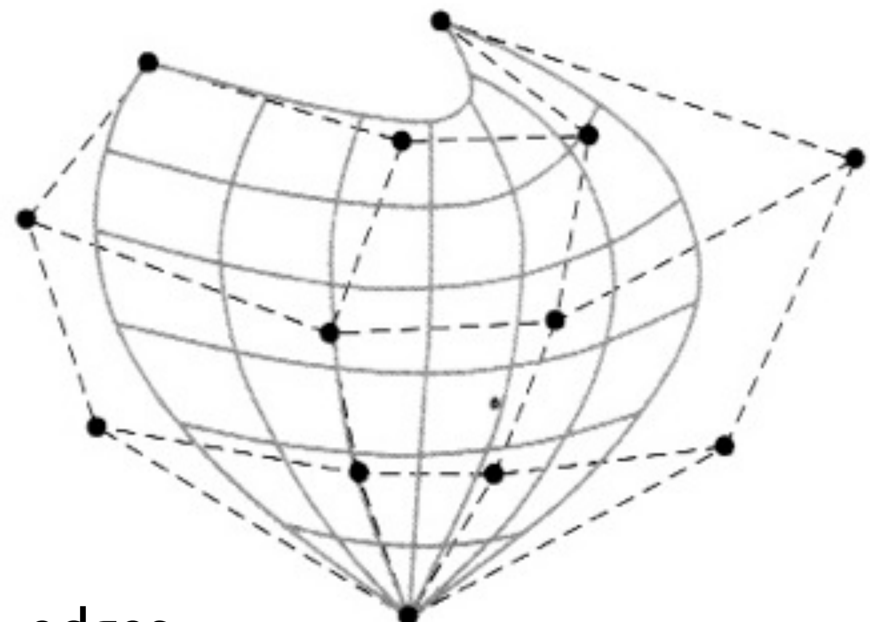
[Rogers]

Bicubic Bézier patch

- Cross product of two cubic Bézier segments

[Foley et al.]





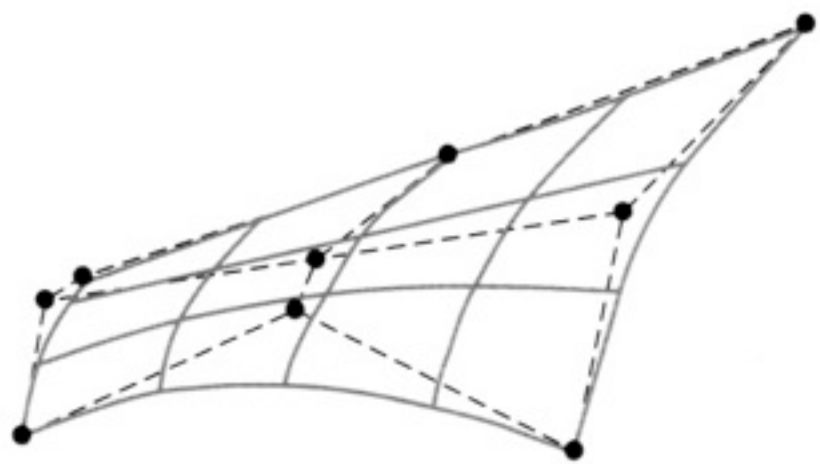
(b)

[Hearn & Baker]

  - properties that carry over
    - interpolation at corners, edges
    - tangency at corners, edges
    - convex hull

Biquadratic Bézier patch

- Cross product of quadratic Bézier curves

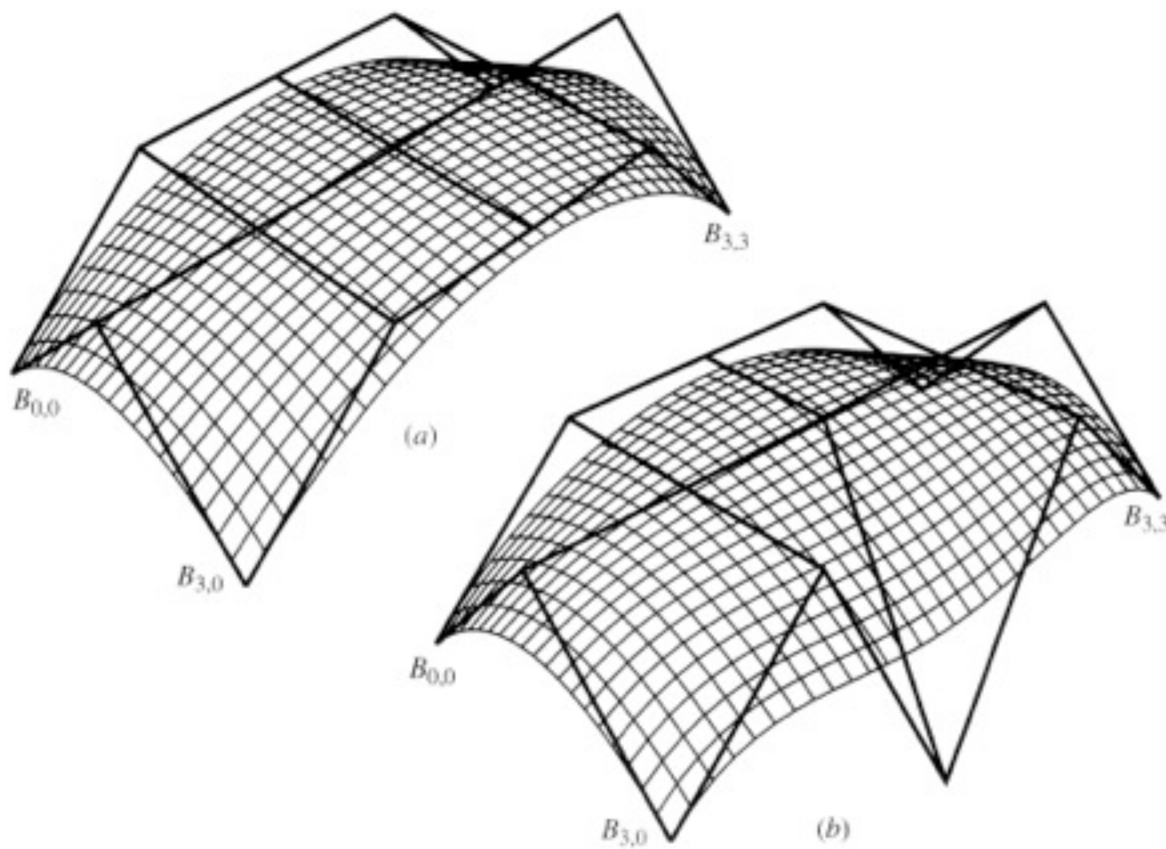


(a)

[Hearn & Baker]

3x5 Bézier patch

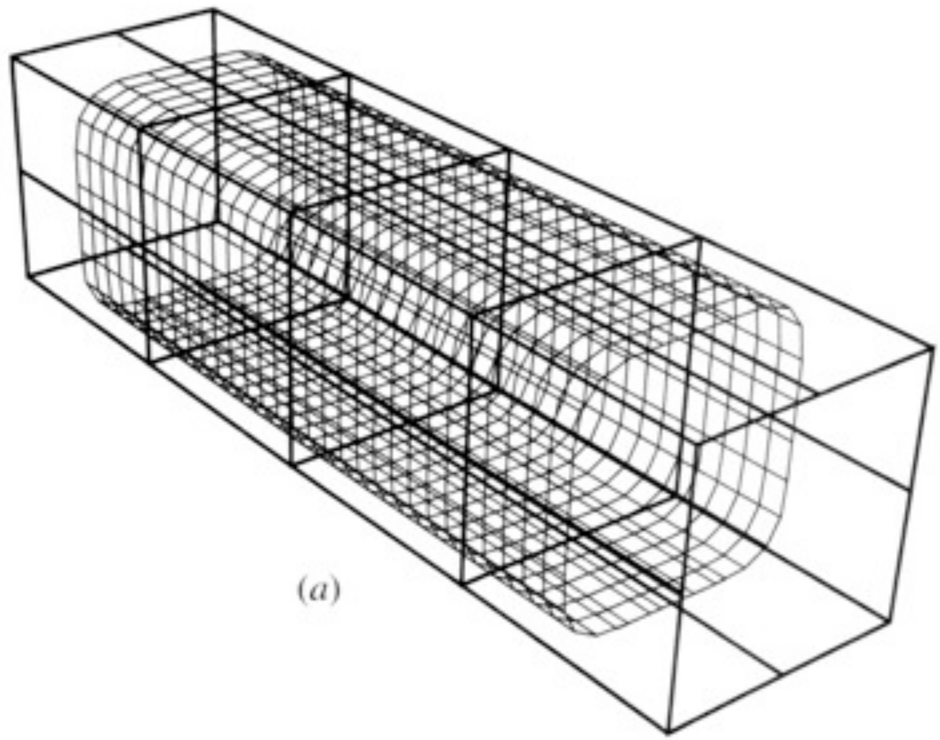
- Cross product of quadratic and quartic Béziers



[Rogers]

Cylindrical B-spline surfaces

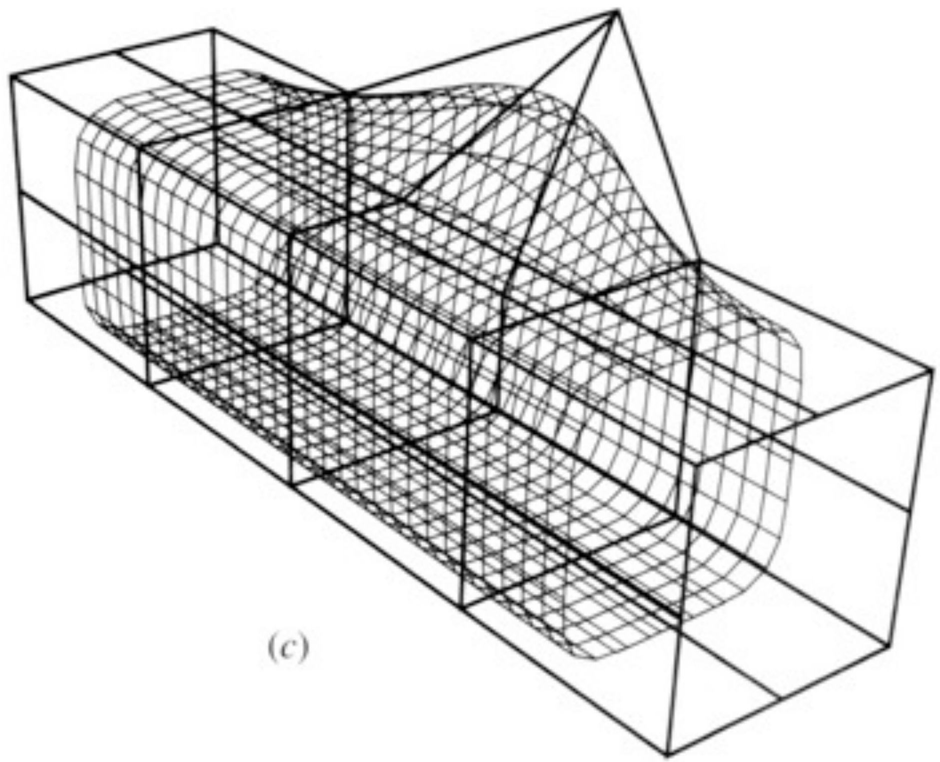
- Cross product of closed and open cubic B-splines



[Rogers]

Cylindrical B-spline surfaces

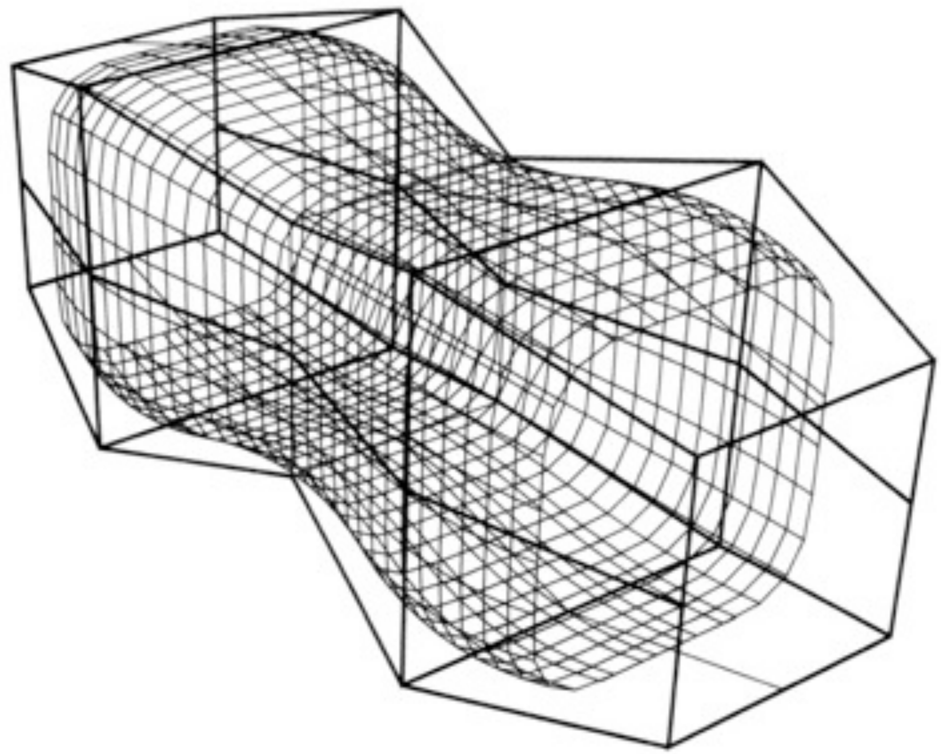
- Cross product of closed and open cubic B-splines


(c)

[Rogers]

Cylindrical B-spline surfaces

- Cross product of closed and open cubic B-splines



(b)

[Rogers]

Topological inflexibility of splines

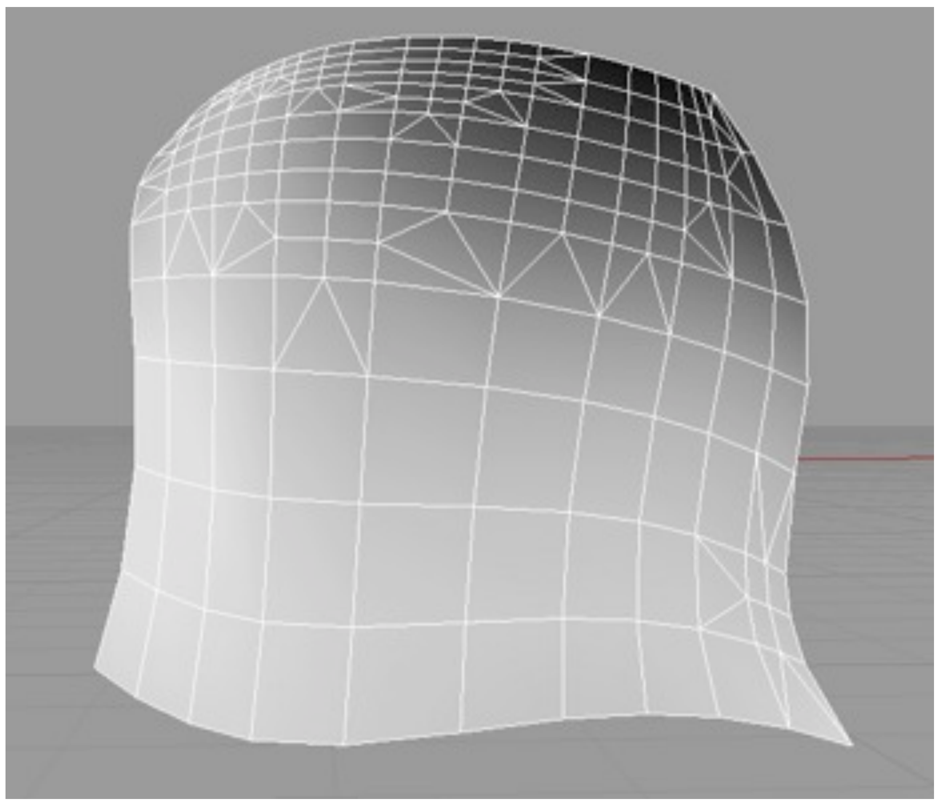
- Spline patches are generally rectangular
  - can wrap them around to make tube-like or torus-like topologies
- Continuity can be readily enforced at edges between patches and at corners where 4 patches join
  - doesn't escape from the topological limitation
- Tiling other kinds of surfaces *must* lead to places where the "wrong" number of patches come together
  - enforcing continuity in these cases is complicated, and is the source of many headaches

Approximating spline surfaces

- Similarly to curves, approximate with simple primitives
  - in surface case, triangles or quads
  - quads widely used because they fit in parameter space
    - generally eventually rendered as pairs of triangles
- adaptive subdivision
  - basic approach: recursively test flatness
    - if the patch is not flat enough, subdivide into four using curve subdivision twice, and recursively process each piece
  - as with curves, convex hull property is useful for termination testing (and is inherited from the curves)

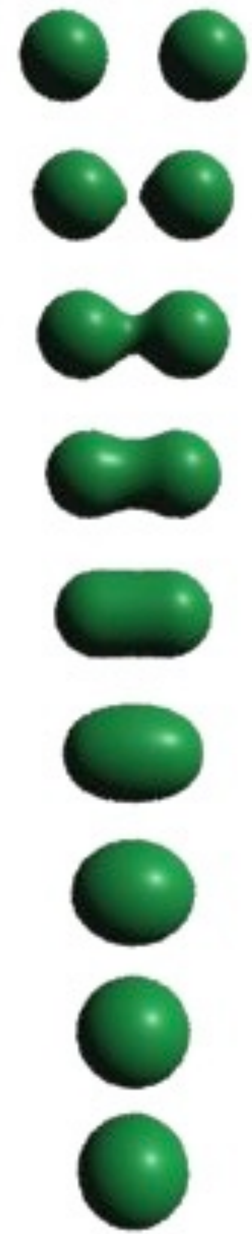
Approximating spline surfaces

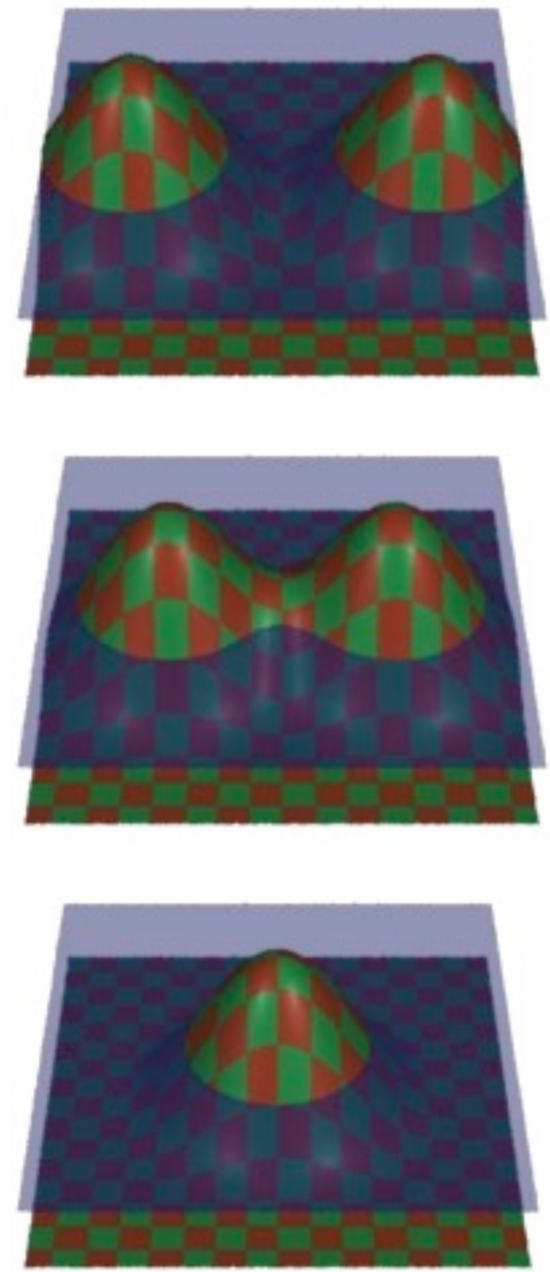
- With adaptive subdivision, must take care with cracks
  - (at the boundaries between degrees of subdivision)



Specific surface reps.

- Algebraic implicit surfaces
  - defined as zero sets of fairly arbitrary functions
  - good news: CSG is easy using min/max
  - bad news: rendering is tough
    - ray tracing: intersect arbitrary zero sets w/ray
    - pipeline: need to convert to triangles
  - e.g. "blobby" modeling

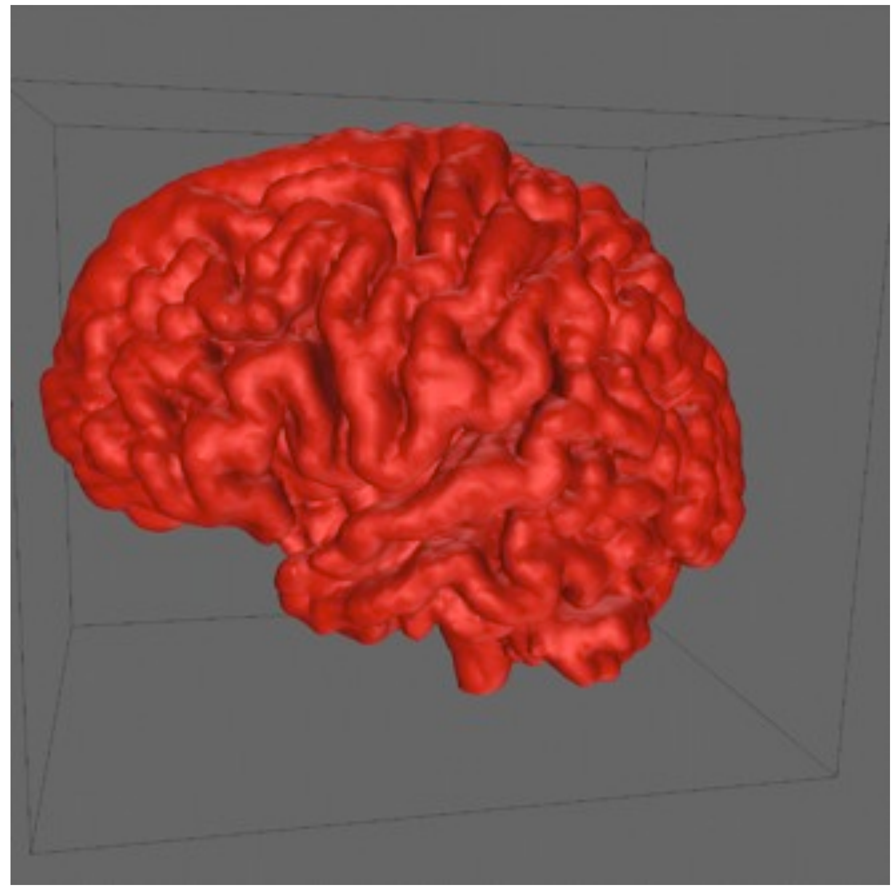




[Pixar | RenderMan Artist tools]

Specific surface representations

- Isosurface of volume data
  - implicit representation
  - function defined by regular samples on a 3D grid
    - (like an image but in 3D)
  - example uses:
    - medical imaging
    - numerical simulation



[source unknown]

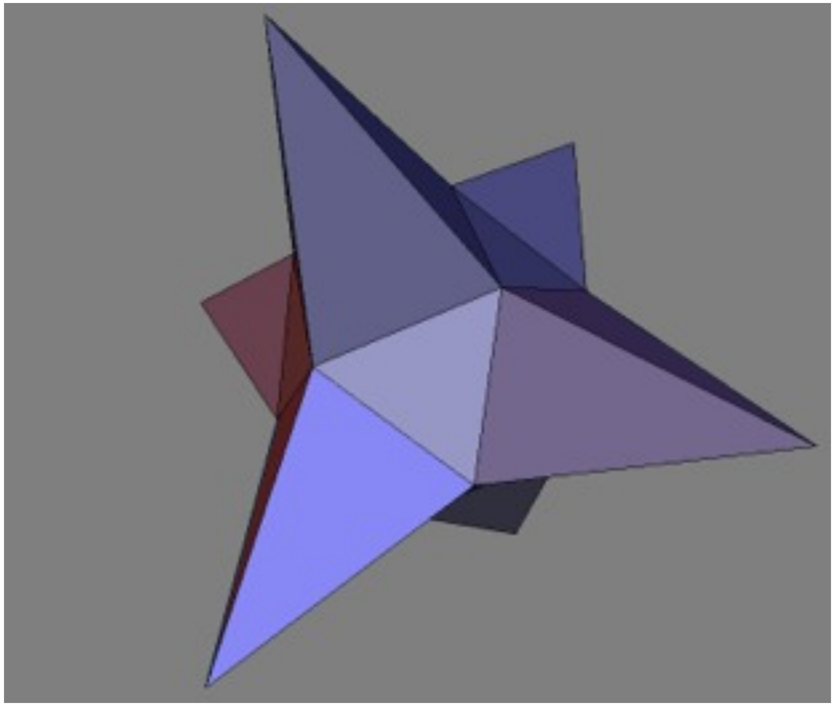
Specific surface representations

- Triangle or polygon meshes
  - parametric (per face)
  - very widely used
    - final representation for pipeline rendering
    - these days restricting to triangles is common
  - rather unstructured
    - need to be careful to enforce necessary constraints
    - to bound a volume need a watertight *manifold* mesh

[Foley et al.]

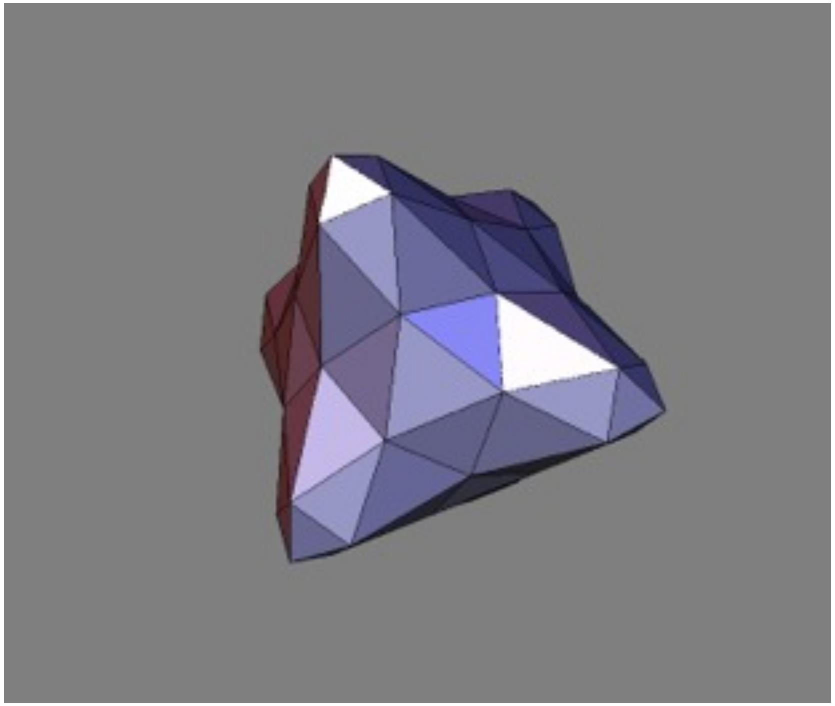
Specific surface representations

- Subdivision surfaces
  - based on polygon meshes (quads or triangles)
  - rules for subdividing surface by adding new vertices
  - converges to continuous limit surface
  - next lecture



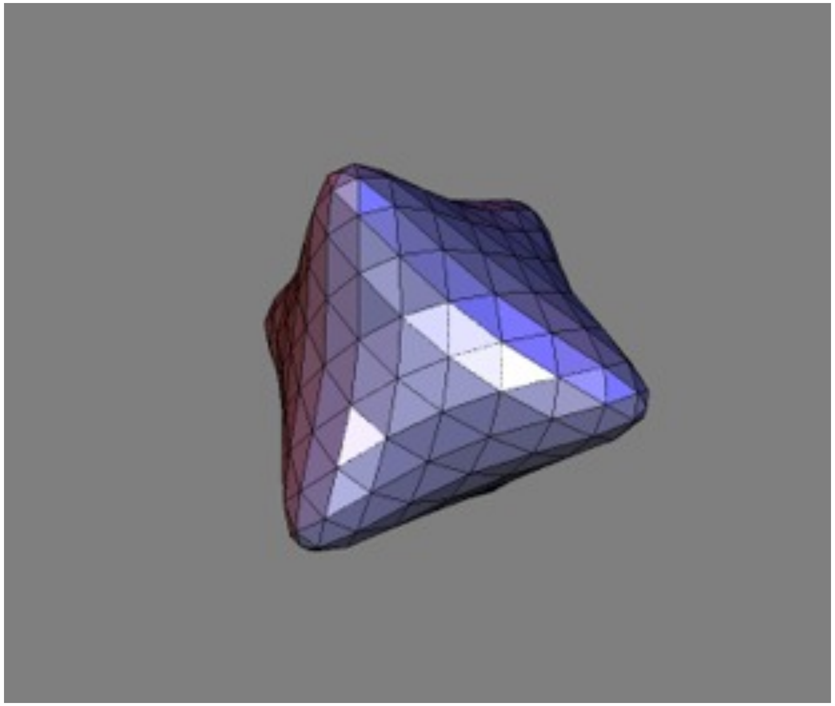
Specific surface representations

- Subdivision surfaces
  - based on polygon meshes (quads or triangles)
  - rules for subdividing surface by adding new vertices
  - converges to continuous limit surface
  - next lecture



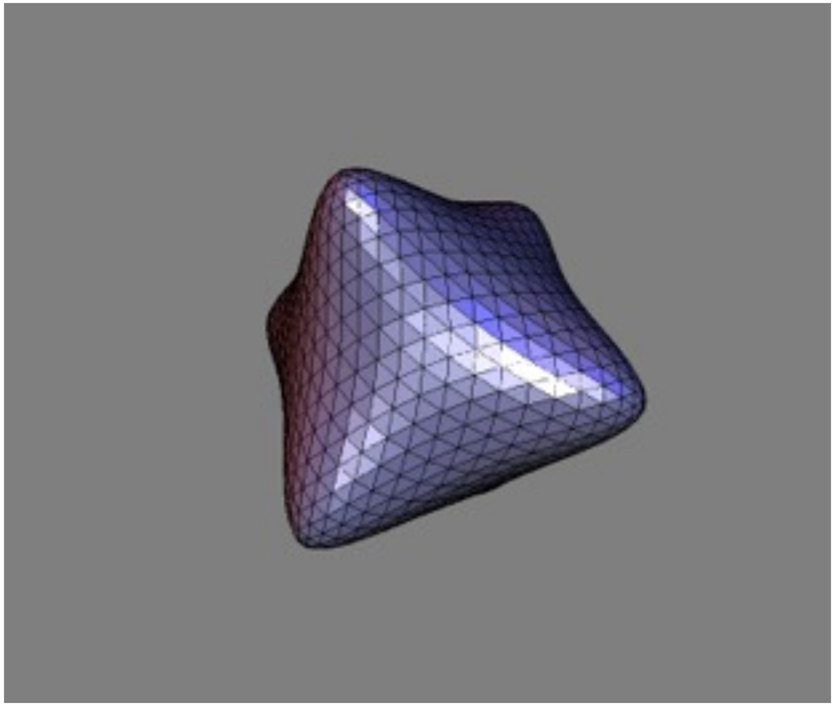
Specific surface representations

- Subdivision surfaces
  - based on polygon meshes (quads or triangles)
  - rules for subdividing surface by adding new vertices
  - converges to continuous limit surface
  - next lecture



Specific surface representations

- Subdivision surfaces
  - based on polygon meshes (quads or triangles)
  - rules for subdividing surface by adding new vertices
  - converges to continuous limit surface
  - next lecture



Representing geometry

- Volumes
  - boundary representations (B-reps)
    - just represent the boundary surface
    - convenient for many applications
    - must be closed (watertight) to be meaningful
      - an important constraint to maintain in many applications

Representing geometry

- Volumes
  - CSG (Constructive Solid Geometry)
    - apply boolean operations on solids
    - simple to define
    - simple to compute in some cases
      - [e.g. ray tracing, implicit surfaces]
    - difficult to compute stably with B-reps
      - [e.g. coincident surfaces]