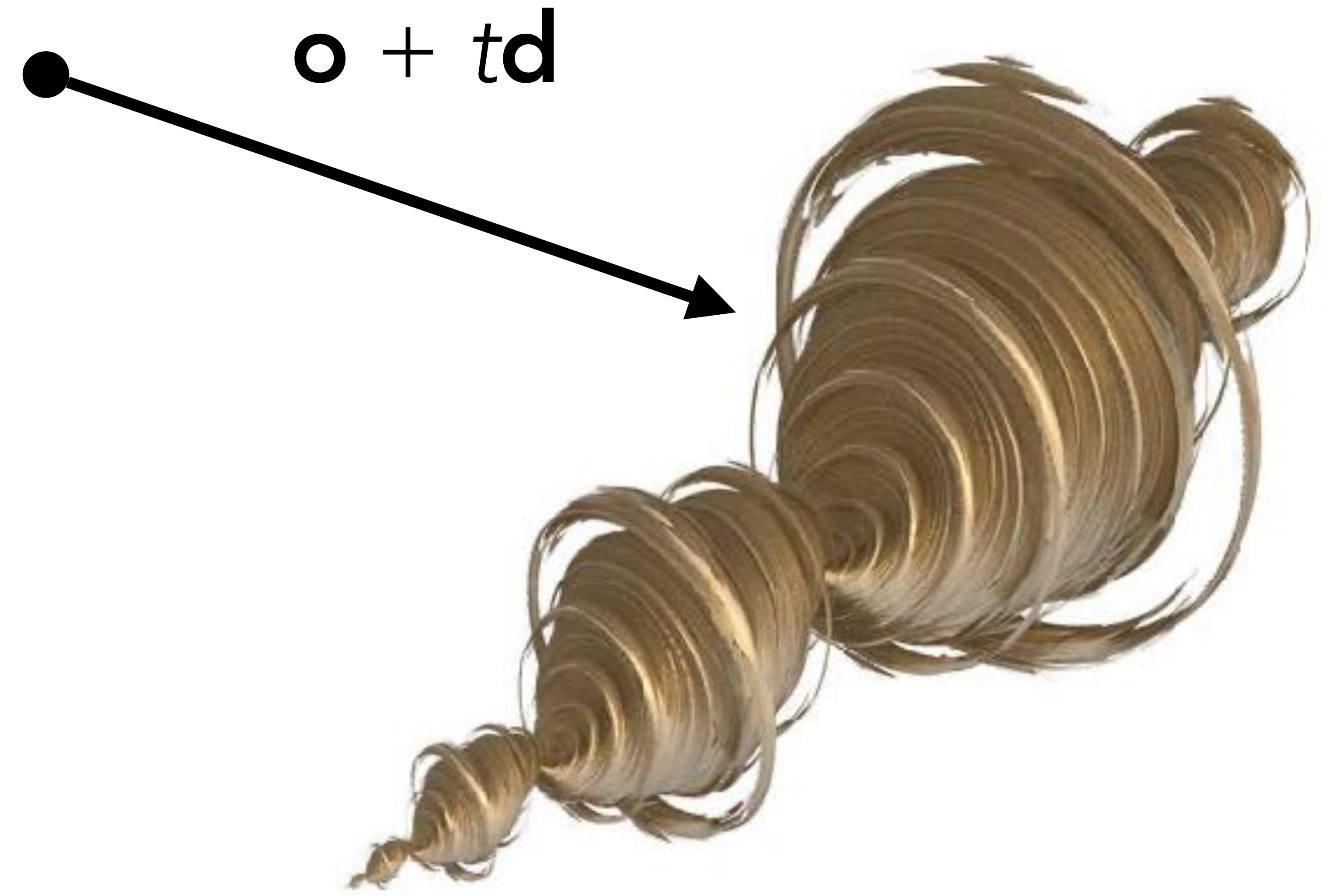
COL781: Computer Graphics

14. Introduction to Modeling

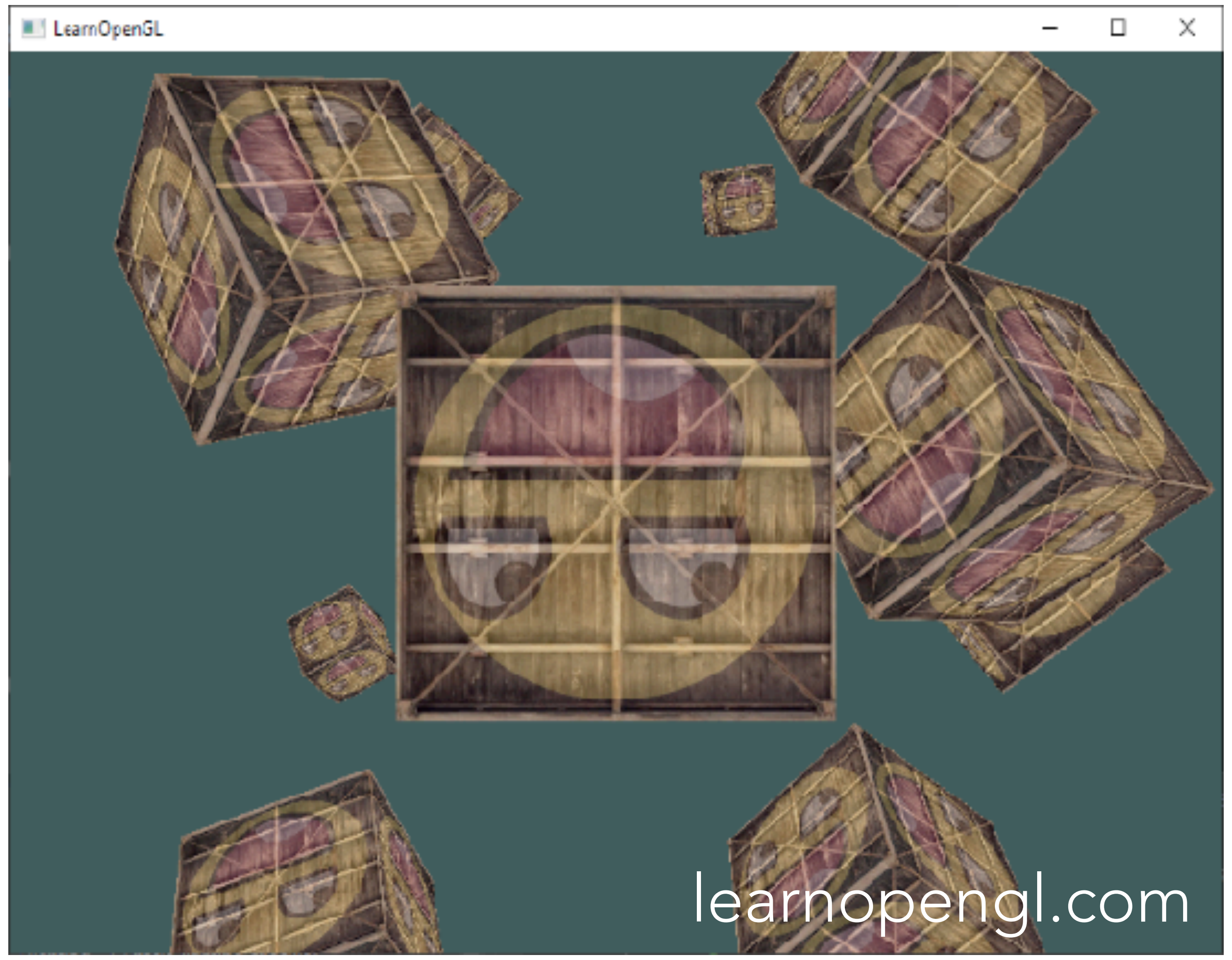
Last class's homework



Suppose you have a specialized, optimized code for intersecting any ray **o** + *t***d** with specific complicated shape *S*.

If the shape is transformed by some matrix **A**, is there an easy way to intersect the transformed shape with a ray without designing a new optimized code?

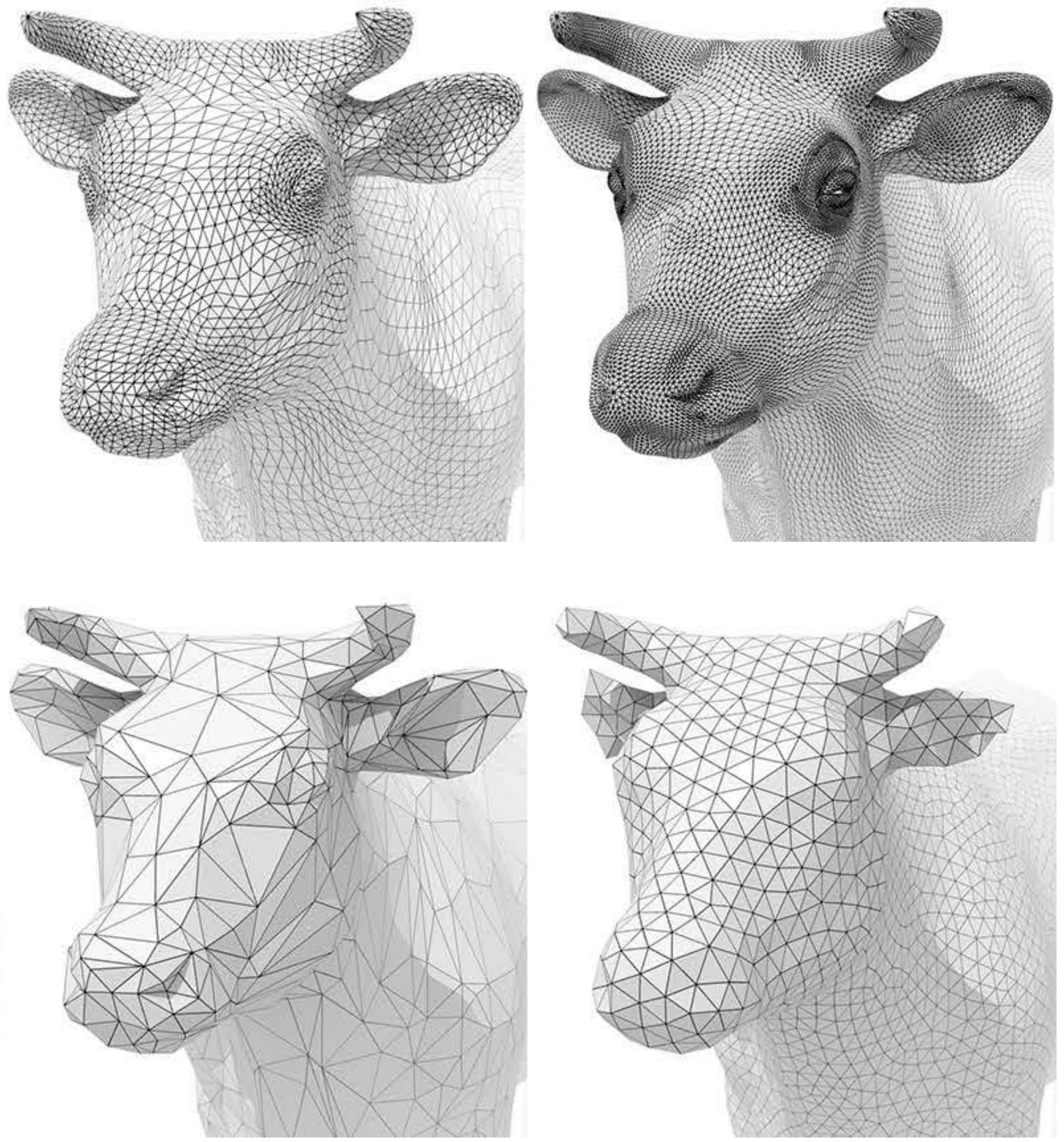
So far, we know how to make crude pictures of polygonal shapes.



Eventually, we will want to make photorealistic movies of complicated shapes!

RENDERING **ANIMATION** **MODELING**

Course content

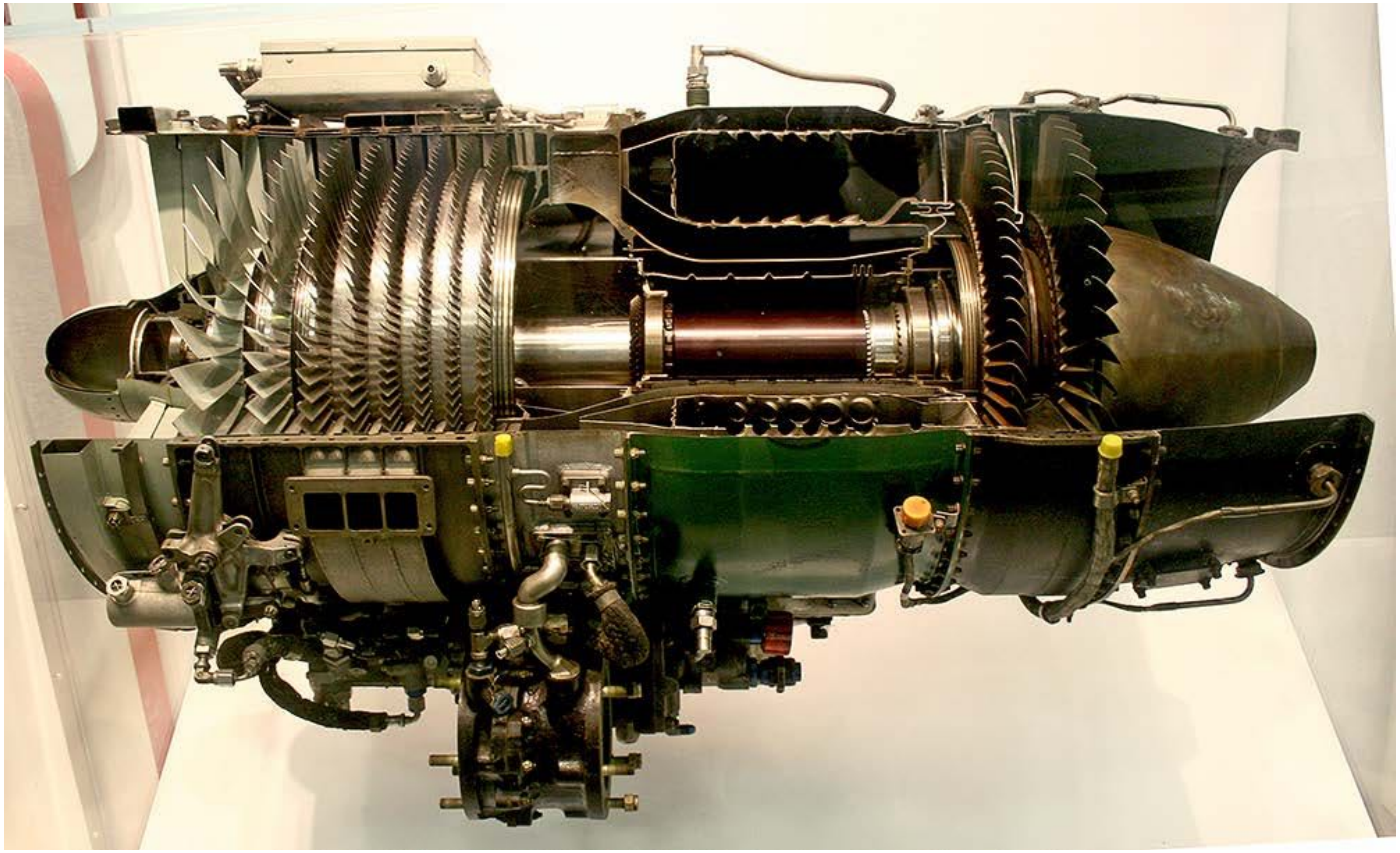


Modeling

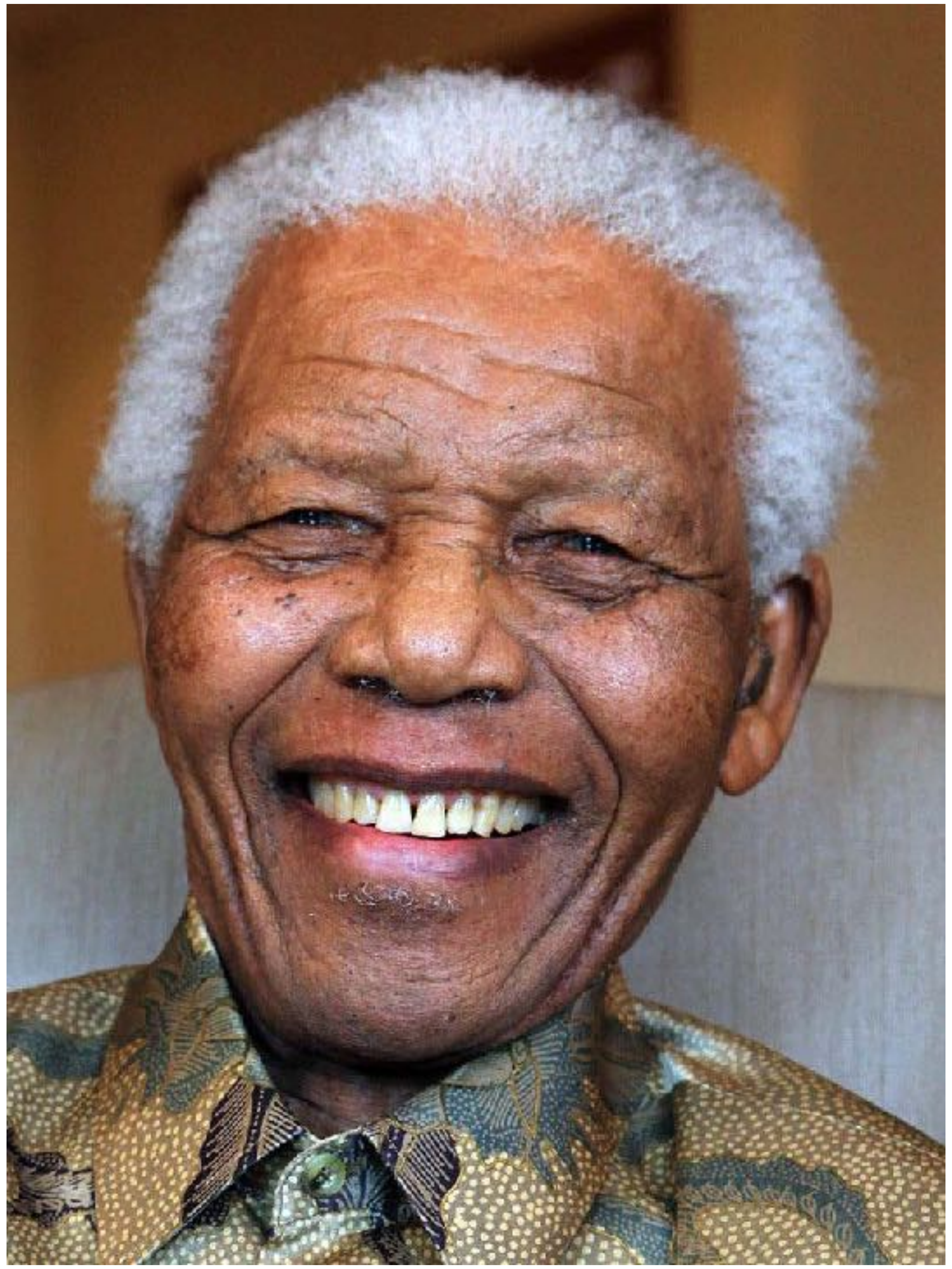
Rendering

Animation

Examples of geometry



Examples of geometry



Examples of geometry

Examples of geometry

Not just surfaces

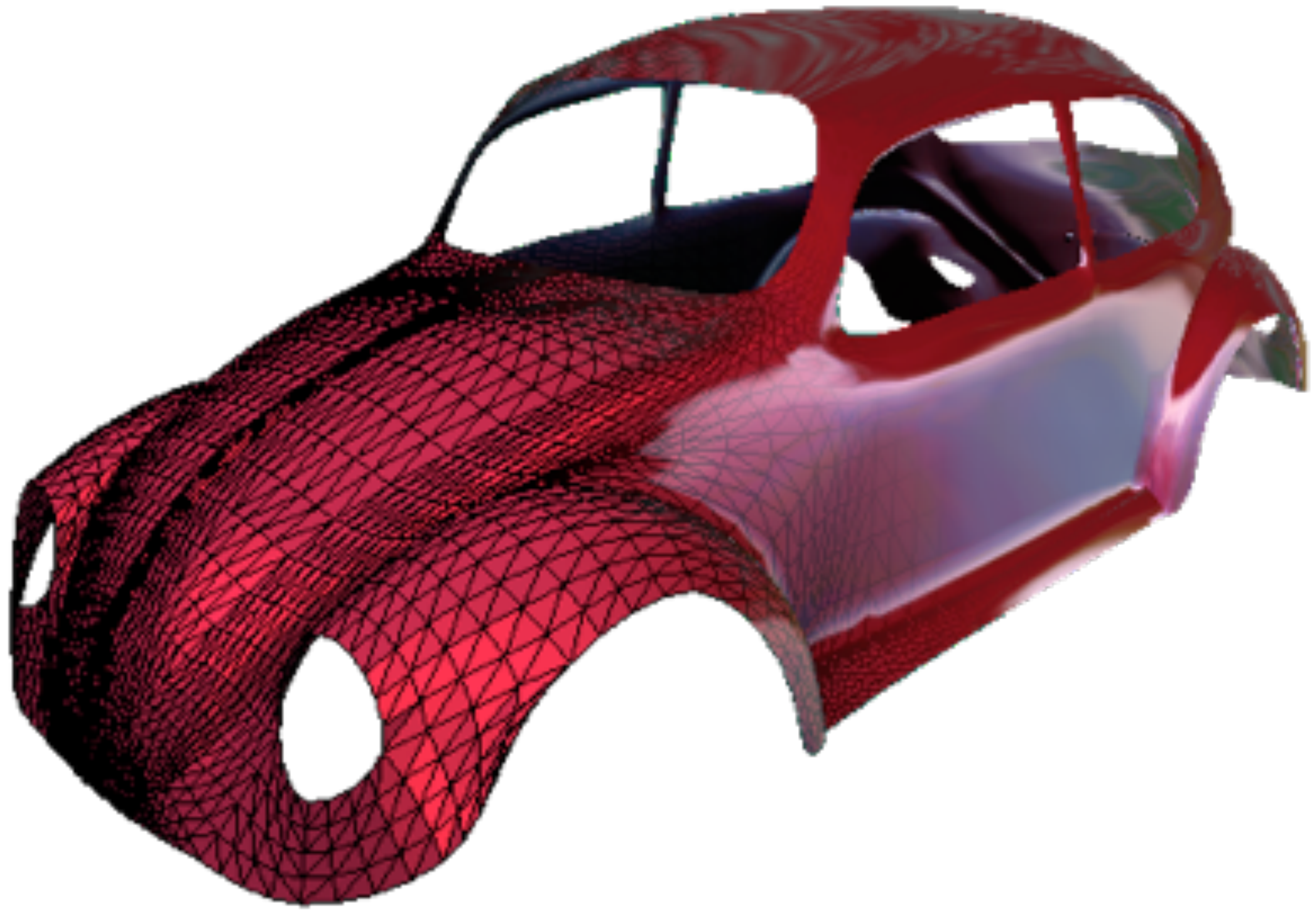

Kaufman et al. 2014

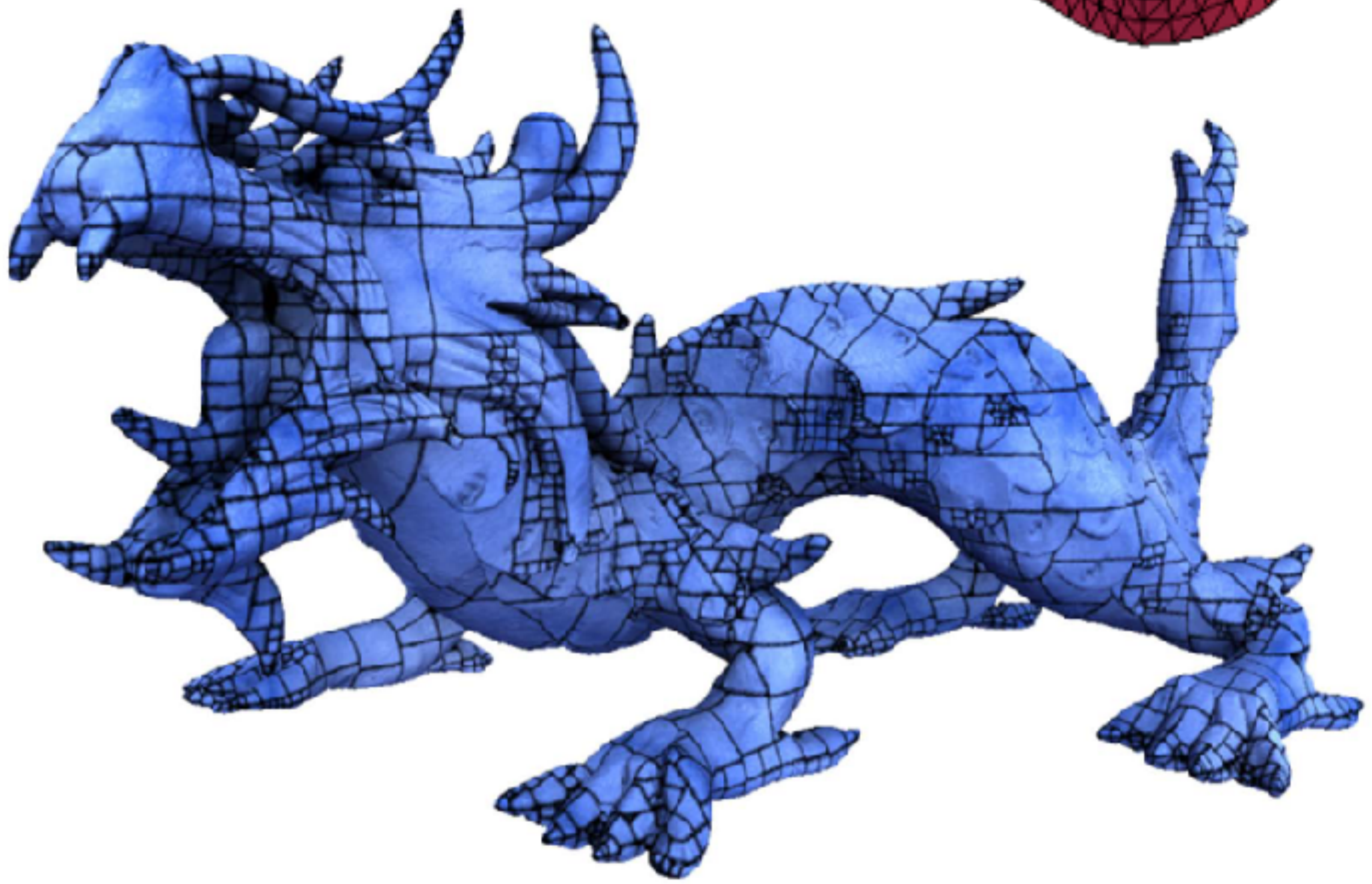

Bitterli et al. 2018

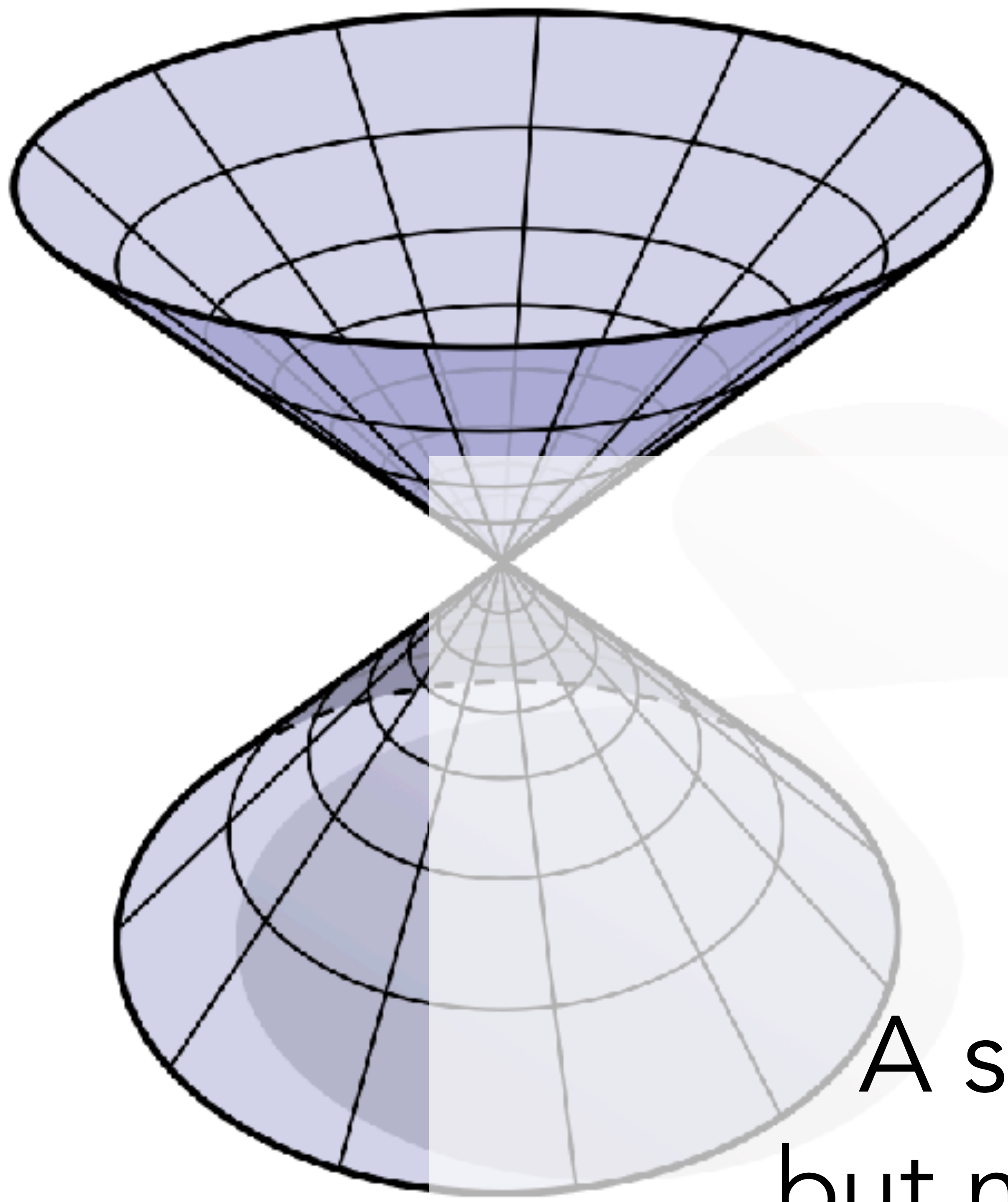
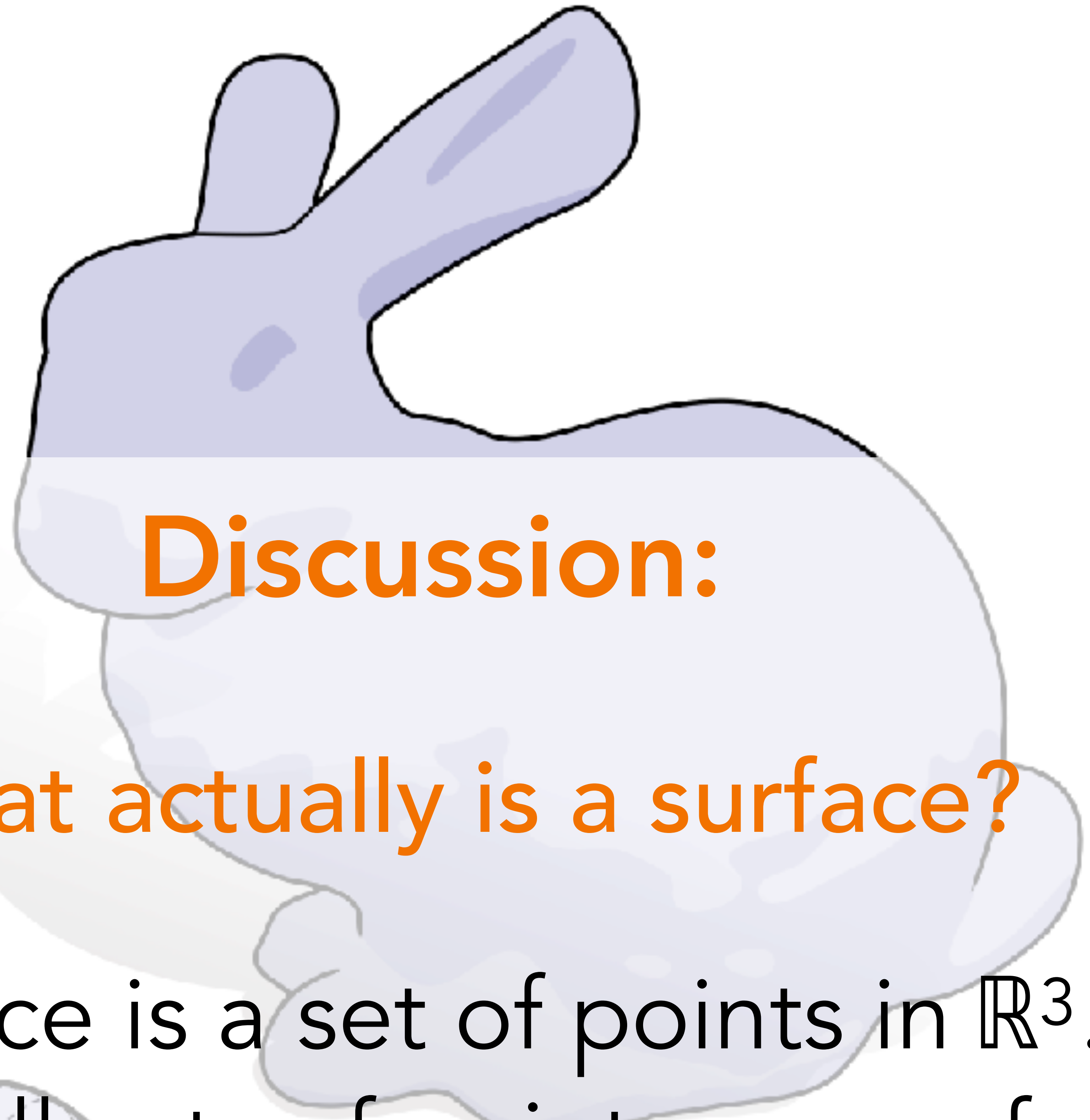
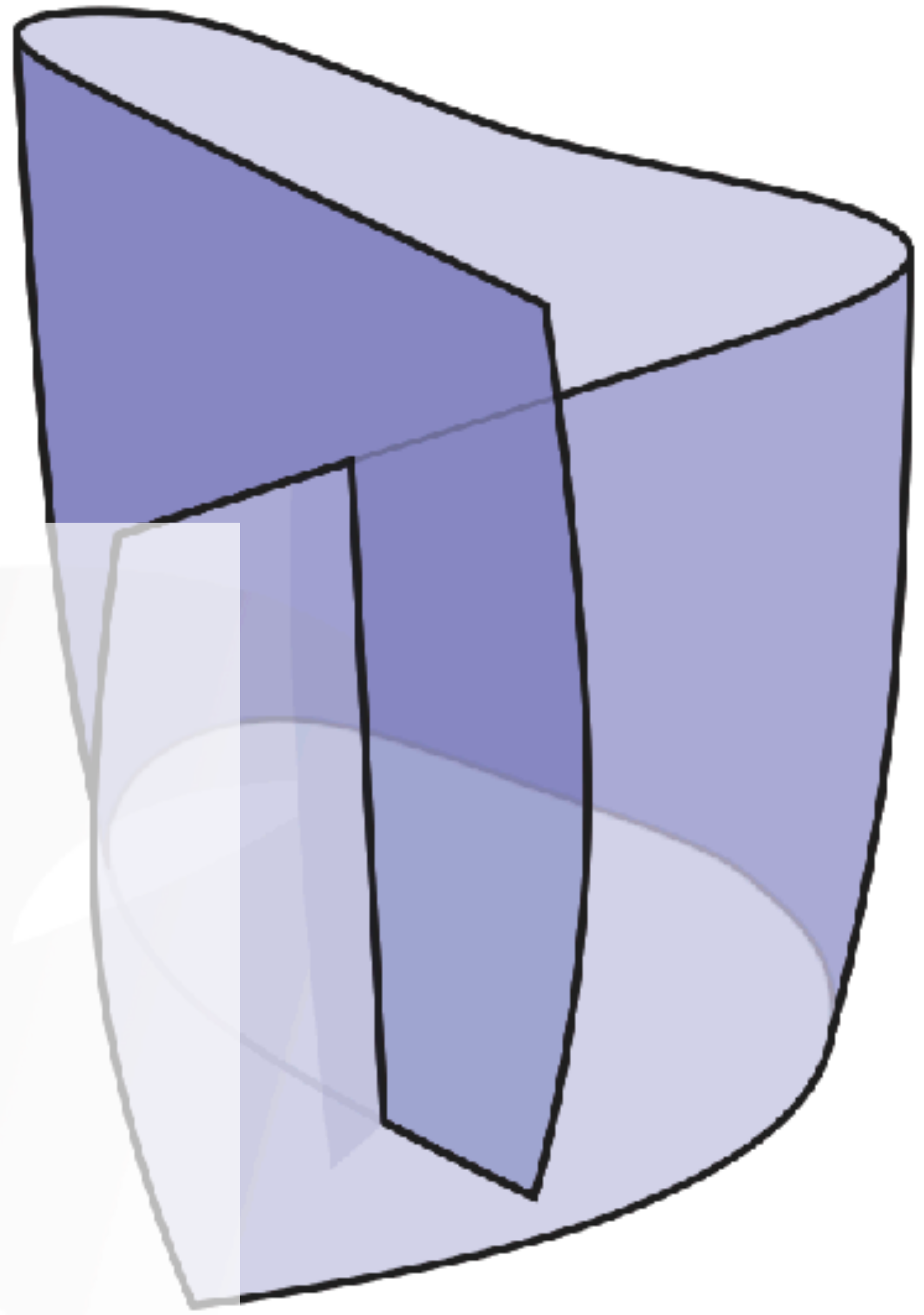
Roadmap

Next few weeks: How to do geometry
(mostly surfaces, a bit of curves, no volumes)

- Representations
- Manipulation and editing
- Geometric queries





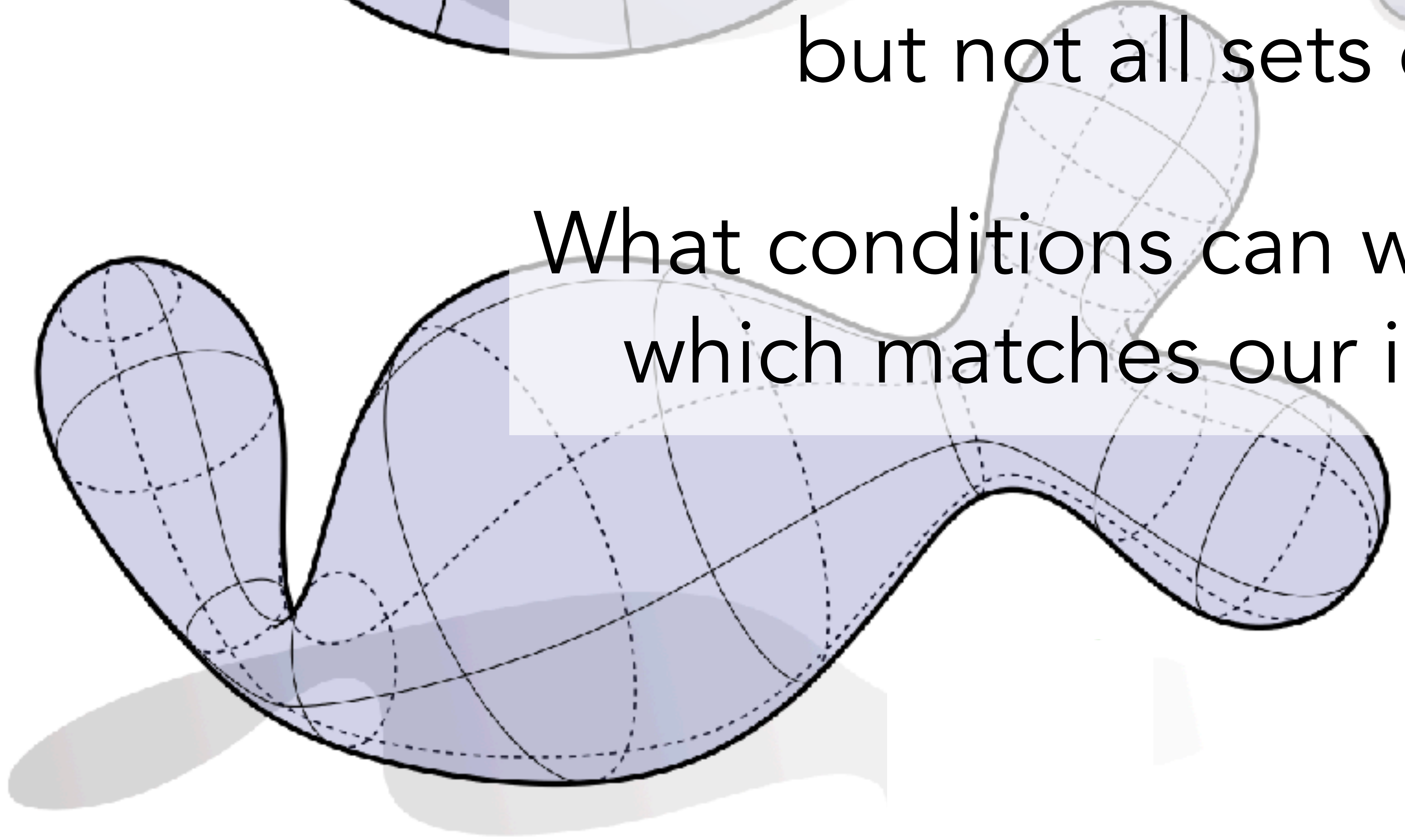
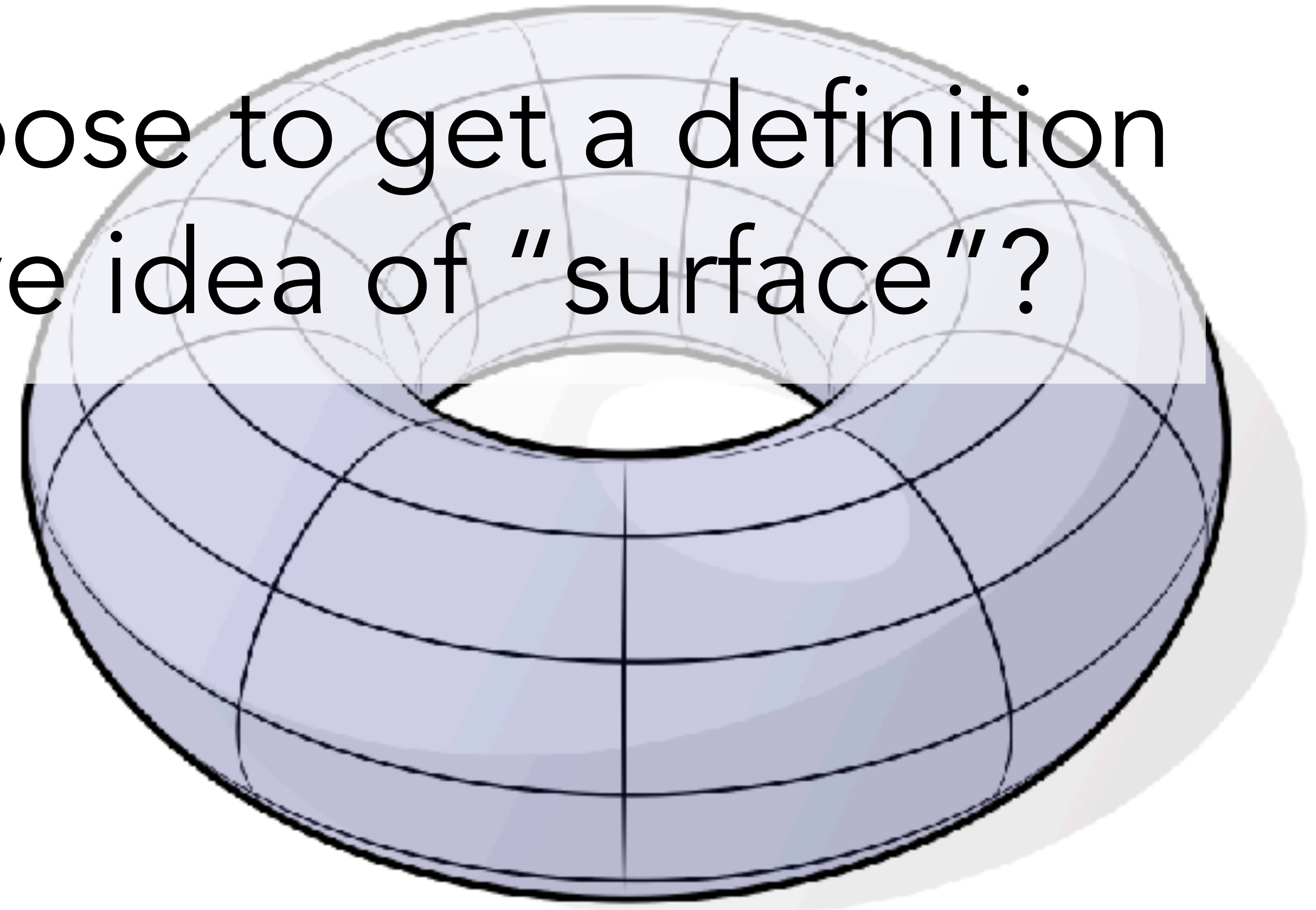




Discussion:

What actually is a surface?

A surface is a set of points in $\mathbb{R}^3$…
but not all sets of points are surfaces.

What conditions can we impose to get a definition
which matches our intuitive idea of "surface"?

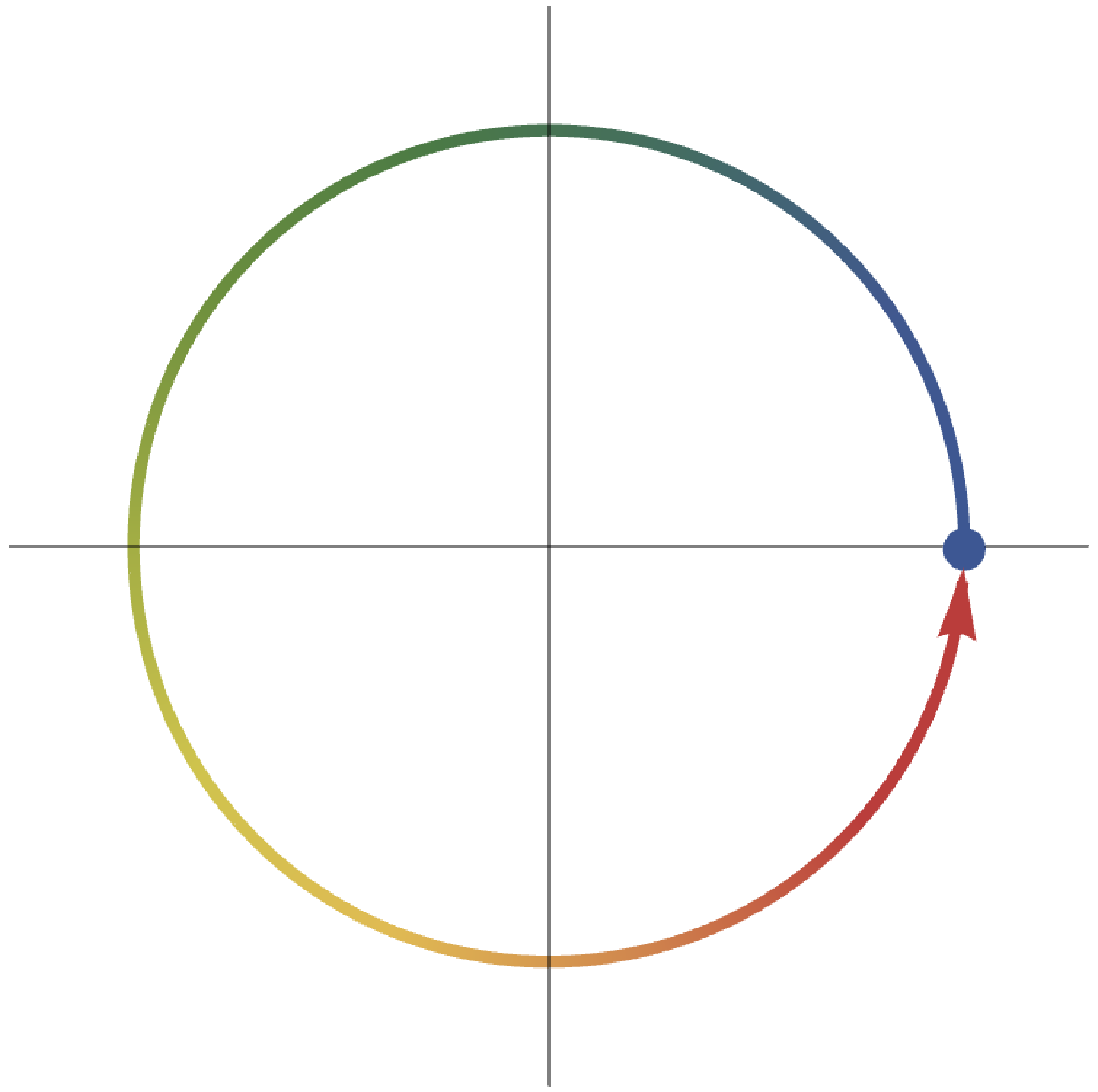



Keenan Crane

How to define a unit circle in 2D?

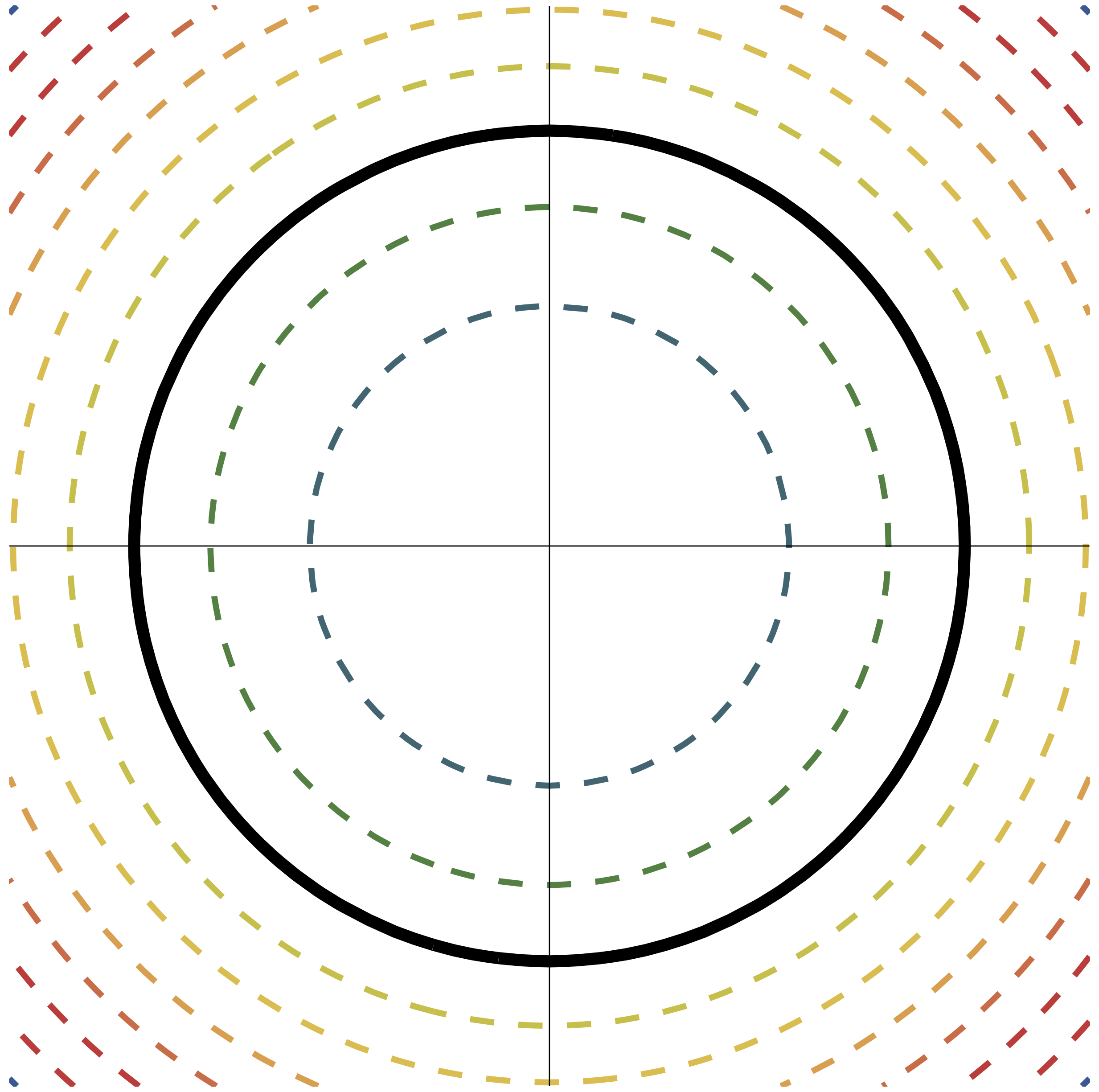
Explicit:

$$\{(\cos\theta, \sin\theta) : 0 \leq \theta < 2\pi\}$$



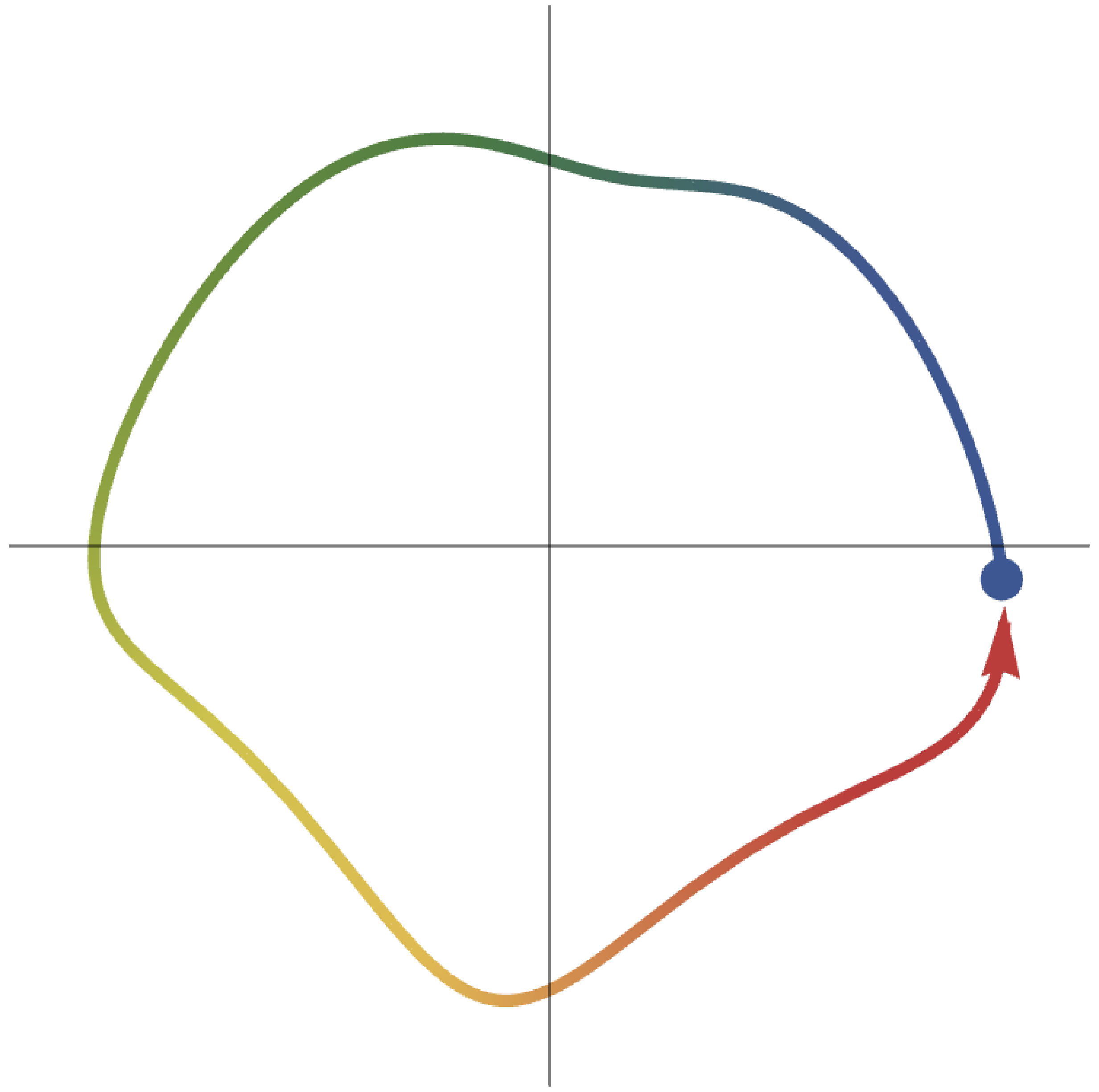
Implicit:

$$\{(x, y) : x^2 + y^2 - 1 = 0\}$$



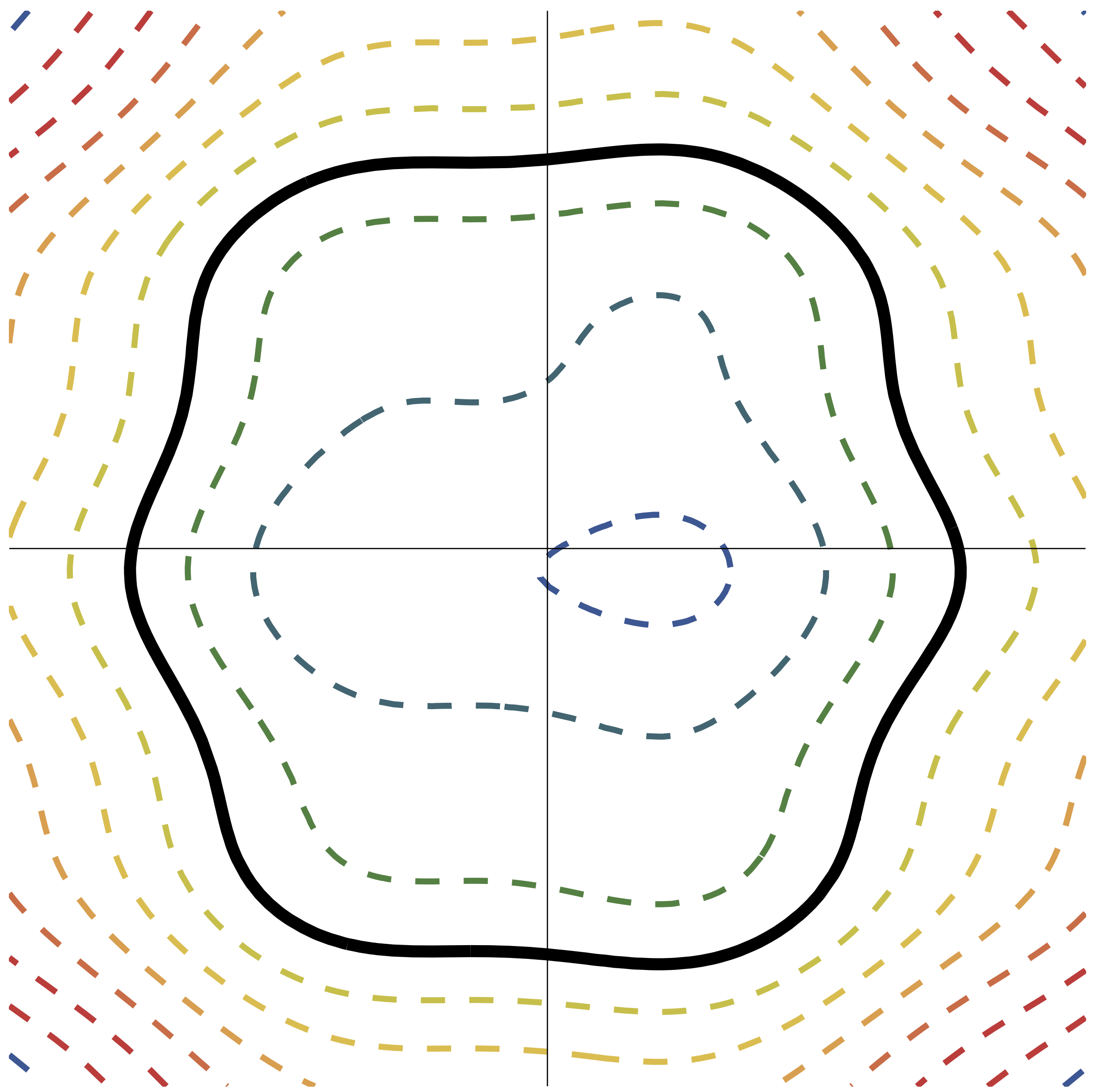
Explicit:

$$\{(x(t),\ y(t)) \colon t \in [a, b]\}$$



Implicit:

$$\{(x,\ y) \colon f(x, y) = 0\}$$

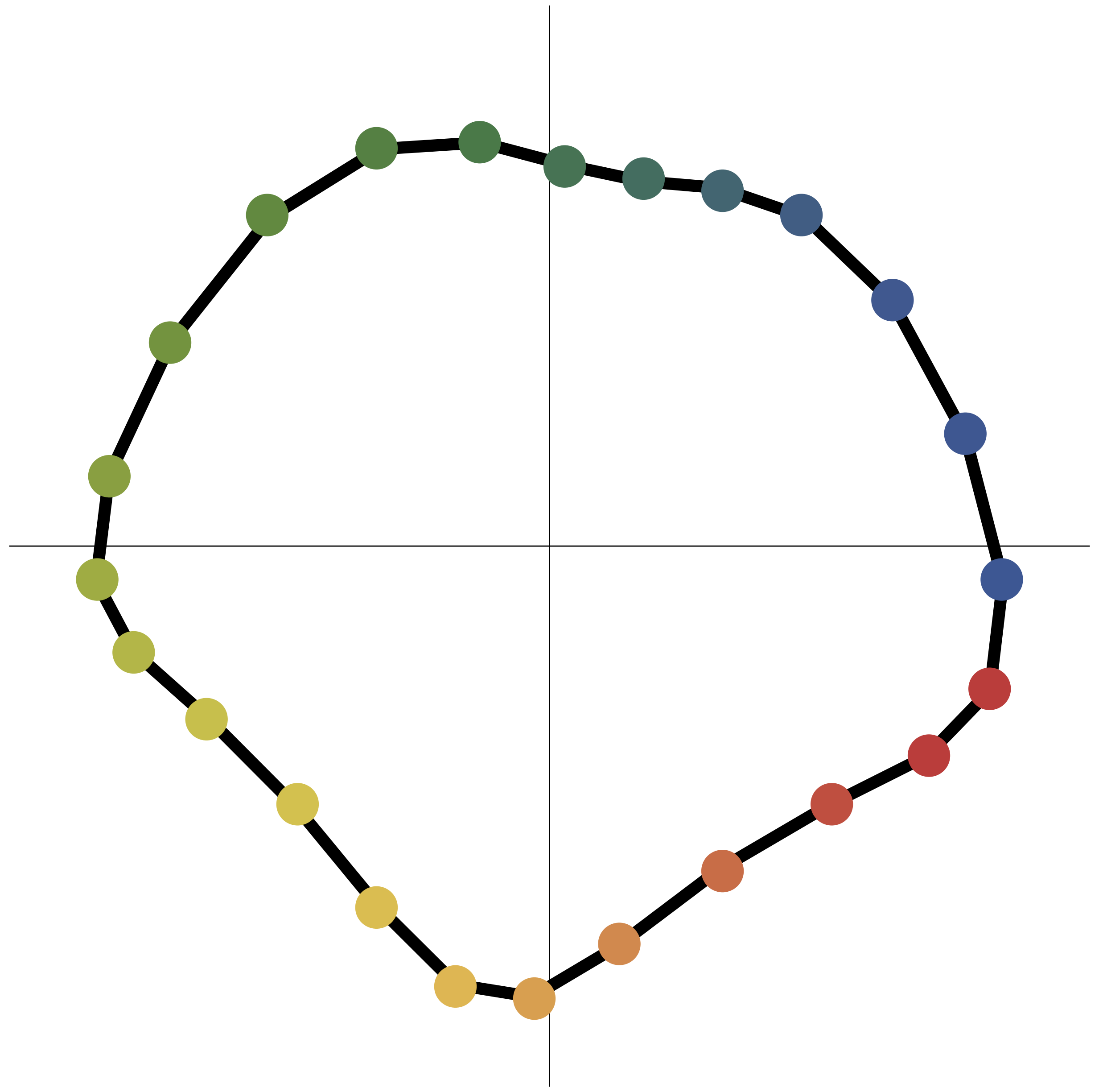


When is it easy to **generate** an arbitrary point on the curve?

When is it easy to **test** if a given point lies on the curve?

How to **draw** a curve given in one of these forms?

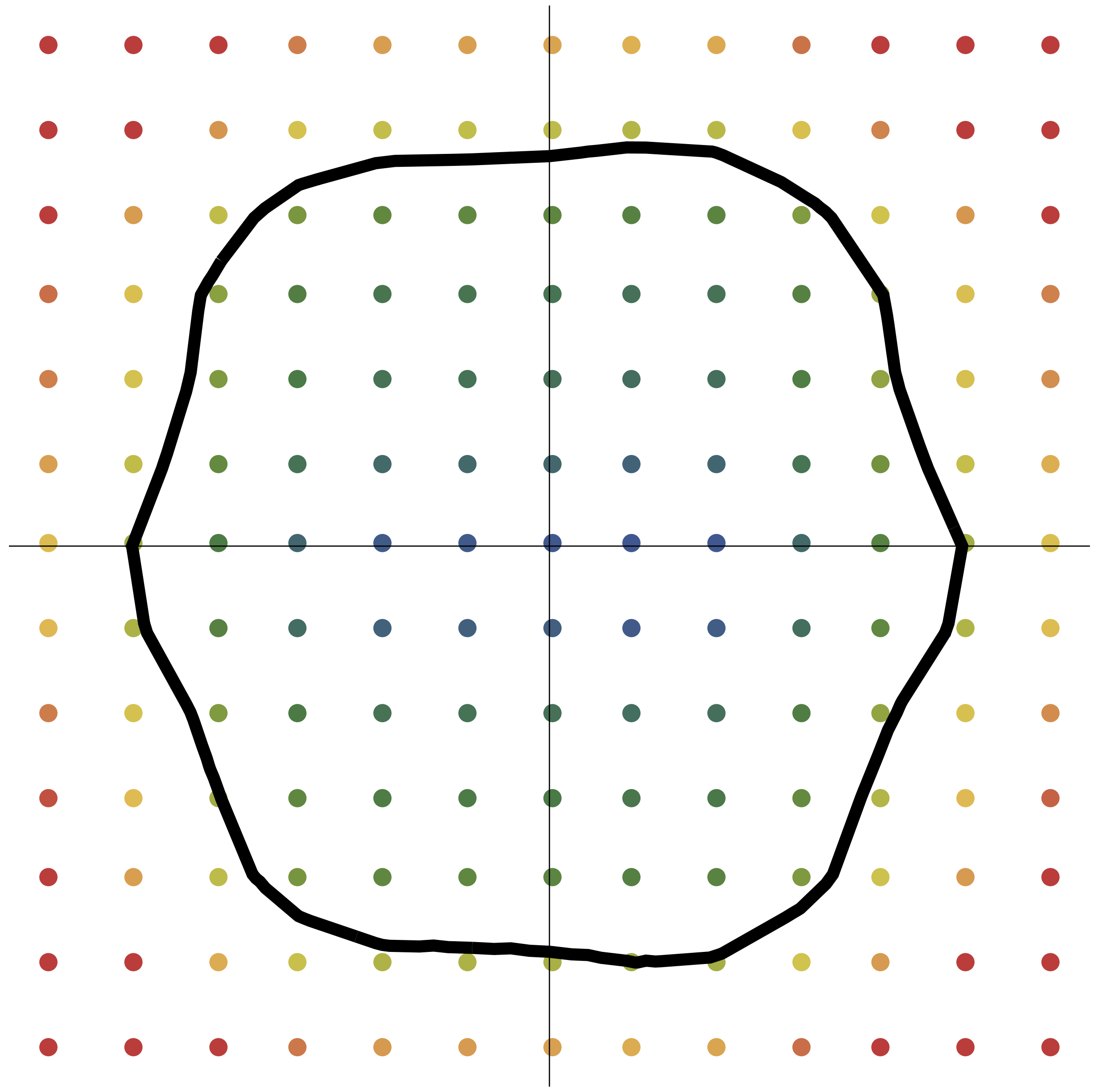
$\{(x(t), y(t)) : t \in [a, b]\}$



Sample points at various values of t

Connect by polyline

$\{(x, y) : f(x, y) = 0\}$



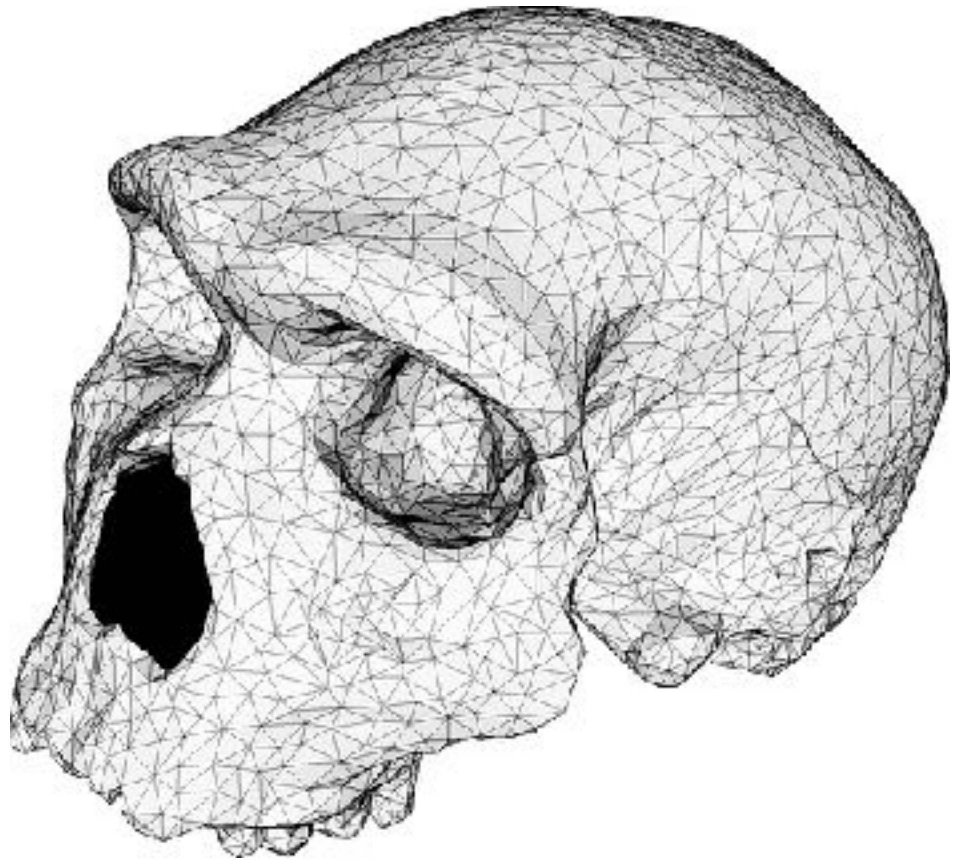
Sample f at various points (x, y)

Draw boundary between + and – points

Representing geometry in 3D

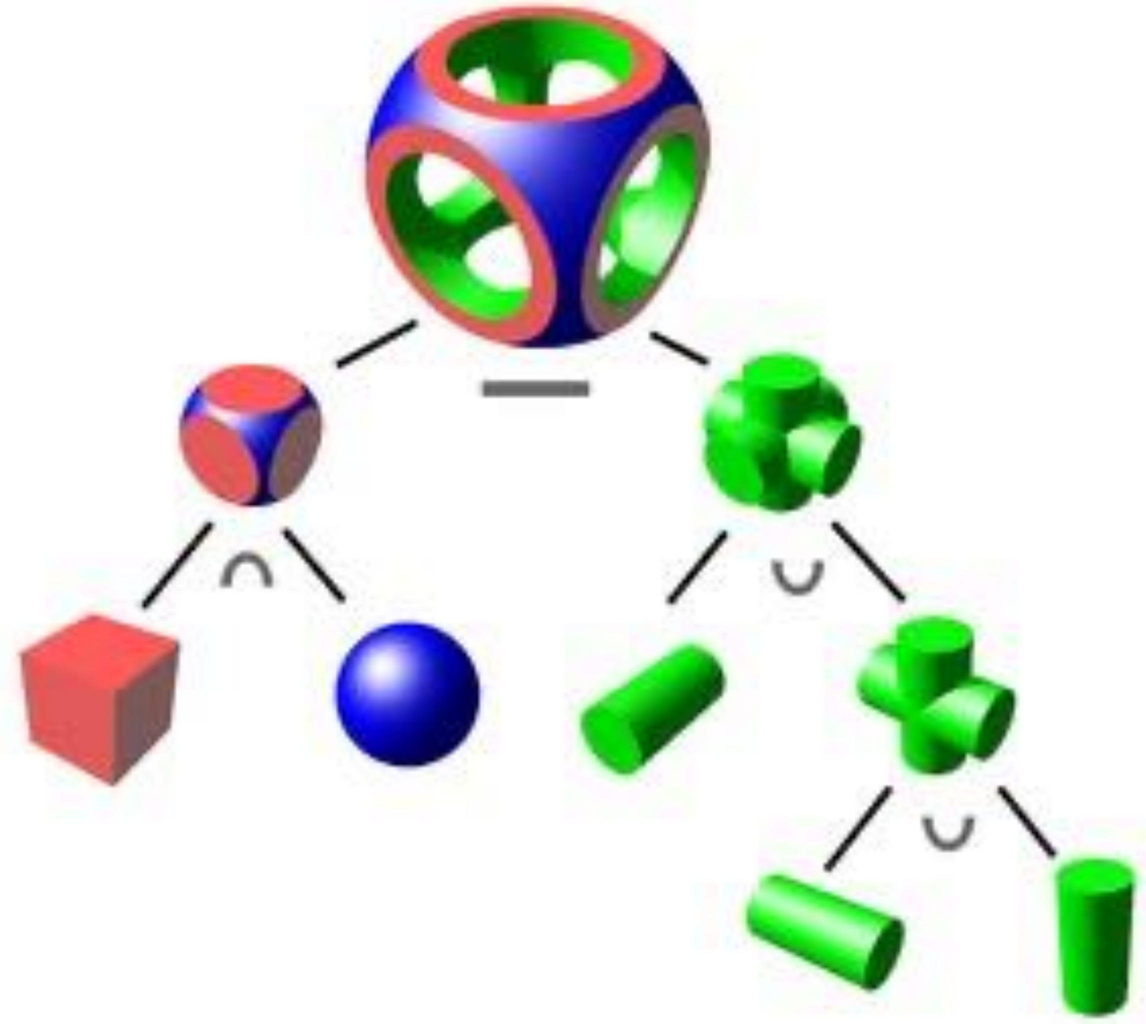
Explicit:

- Polygon meshes
- Parametric curves and surfaces
- Subdivision surfaces
- Point clouds



Implicit:

- Algebraic surfaces, distance fields
- Constructive solid geometry
- “Blobby” surfaces
- Level sets



Implicit representations

Implicit surfaces

Defined as the zero set of a given function

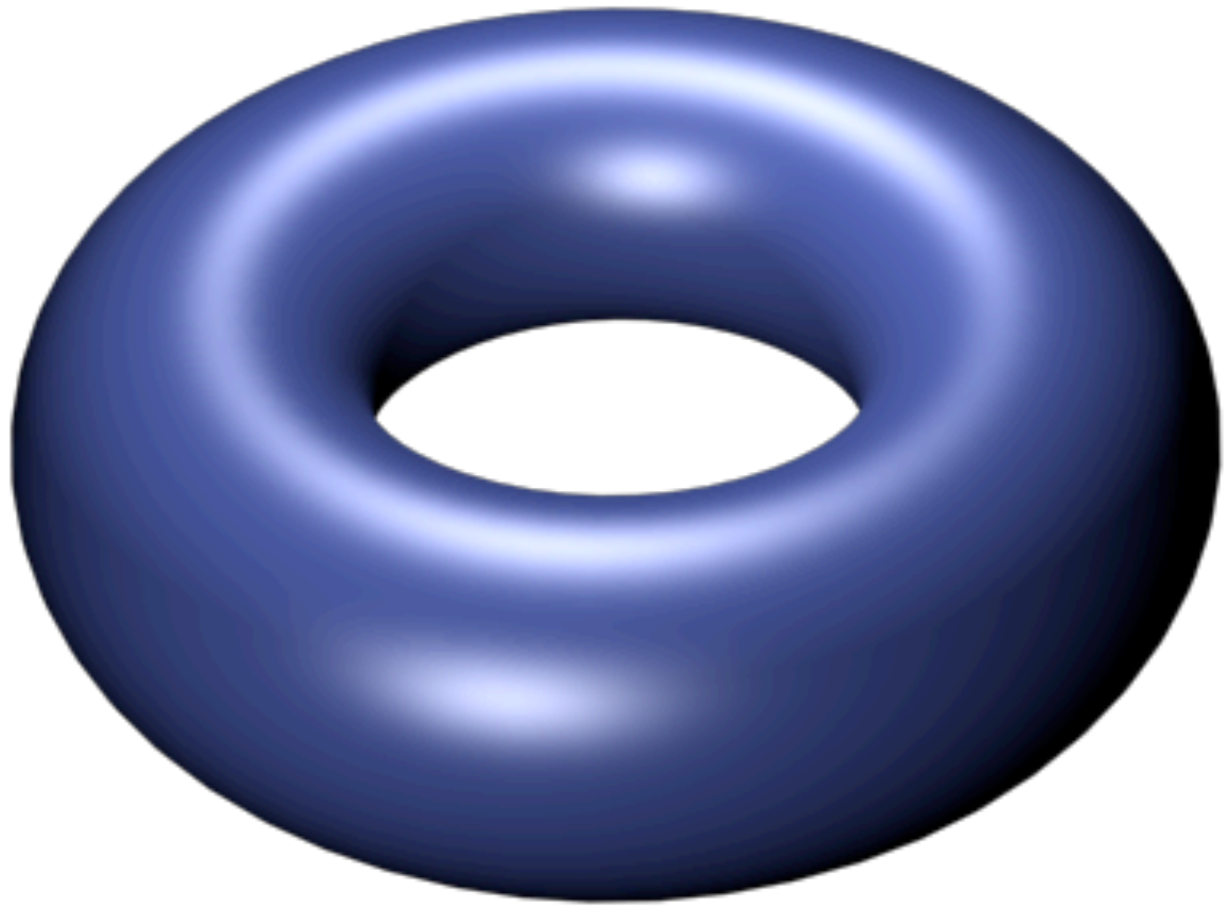
$$S = \{(x, y, z) : f(x, y, z) = 0\}$$

Algebraic surface: f is a polynomial

Signed distance field:

$$f(\mathbf{p}) = \begin{cases} \text{dist}(\mathbf{p}, S) & \text{if } \mathbf{p} \text{ is outside } S, \\ -\text{dist}(\mathbf{p}, S) & \text{if } \mathbf{p} \text{ is inside.} \end{cases}$$

Simple formulas only exist in very special cases…

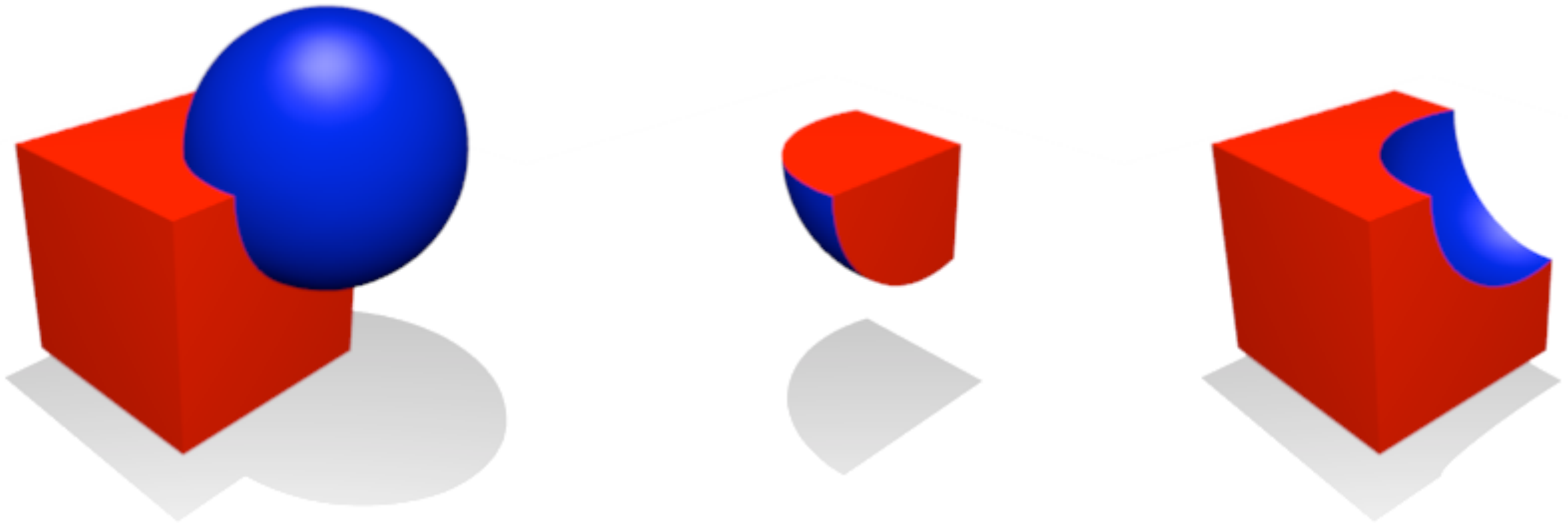


$$(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0$$

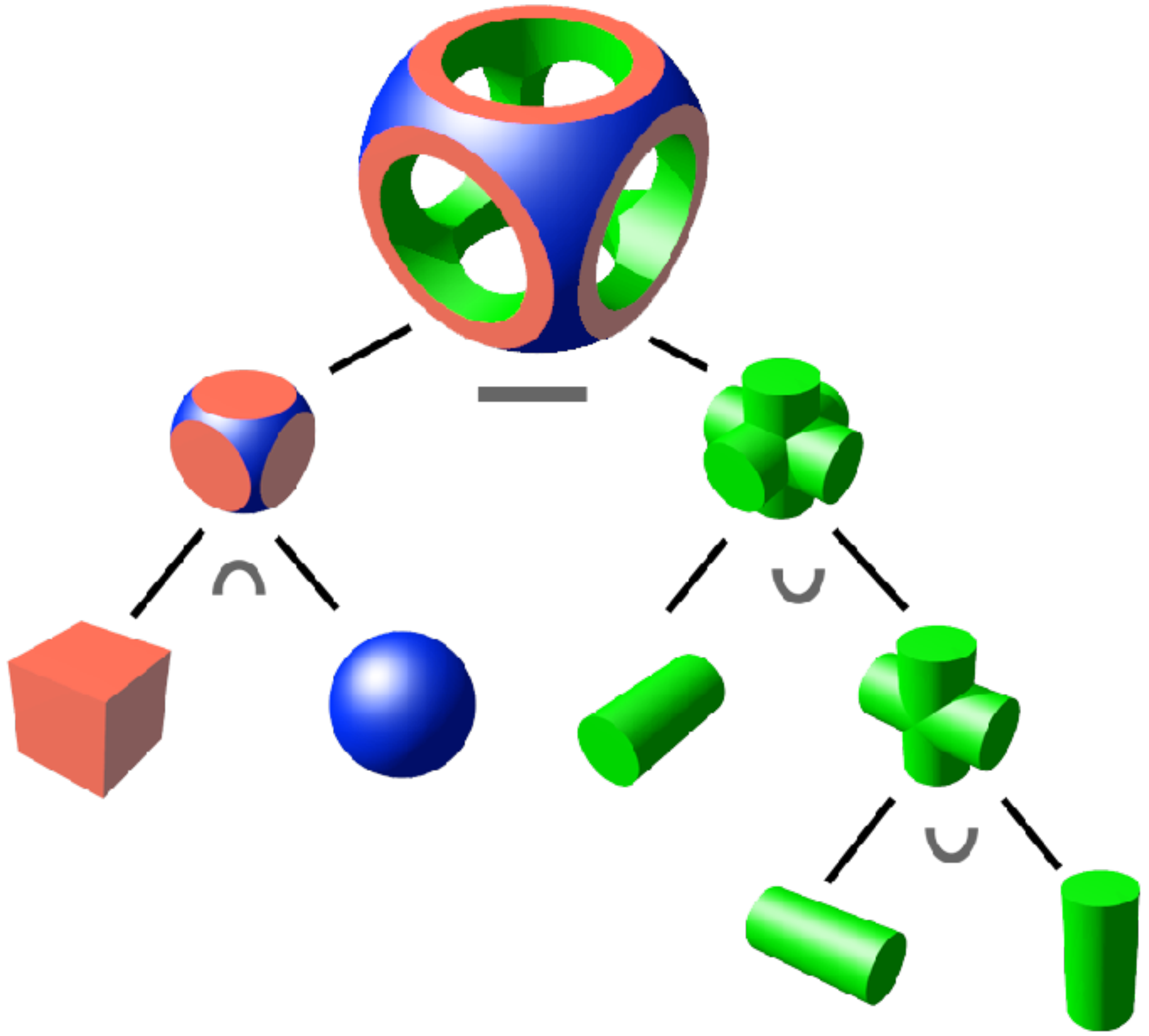
Constructive solid geometry

An implicit representation defines both a surface, $f(\mathbf{p}) = 0$, and its enclosed volume, $f(\mathbf{p}) \le 0$.

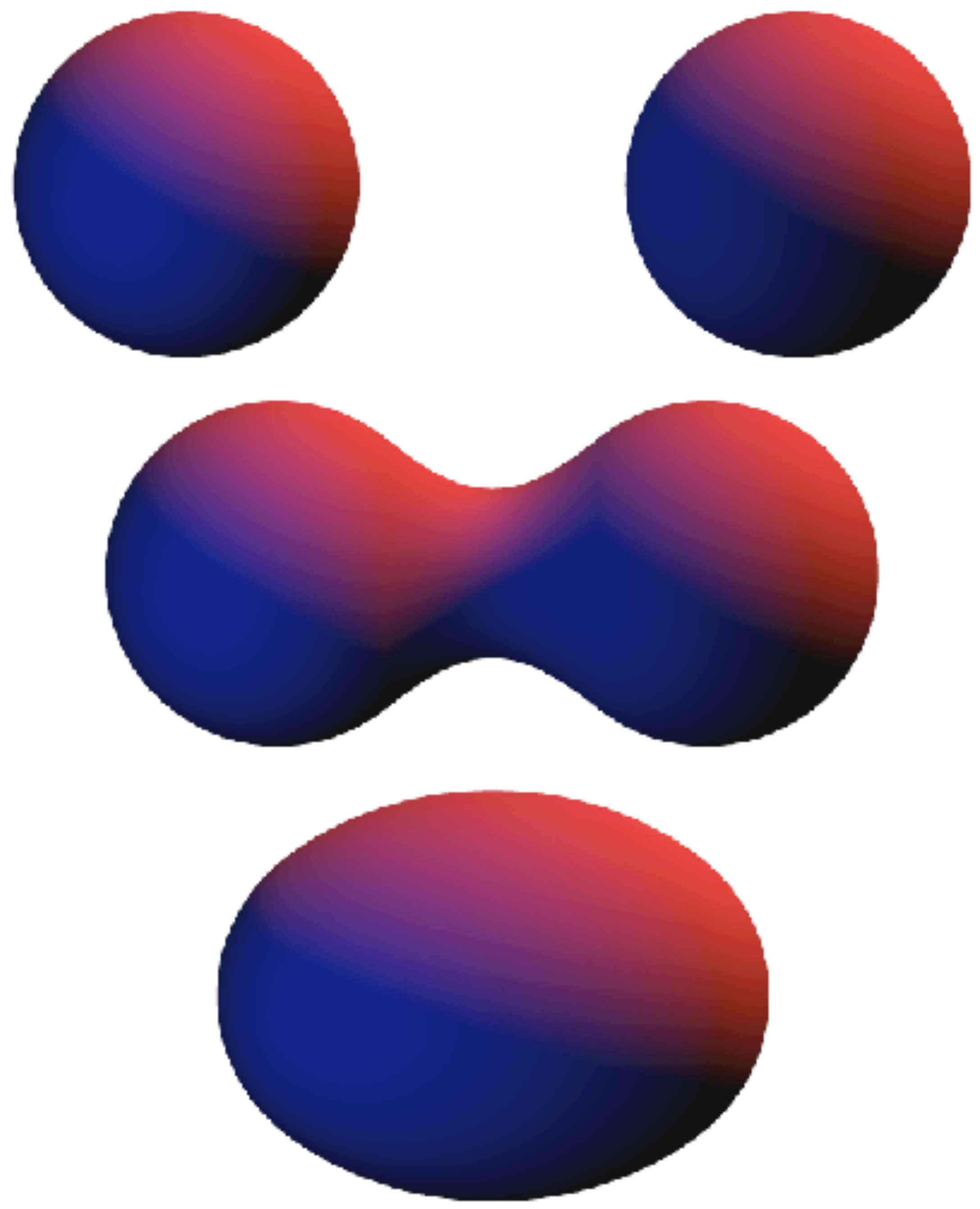
So we can do set operations on the volume:



Union Intersection Difference



Smooth implicit modeling



Instead of a Boolean operation, blend together the implicit functions of two surfaces.

e.g. $$f_i(\mathbf{p}) = \exp(-\|\mathbf{p}-\mathbf{c}_i\|^2/r_i^2)$$

$$S = \{\mathbf{p} : \sum f_i(\mathbf{p}) = 0.5\}$$

A.k.a. metaballs, blobbies, soft objects, …

Choice of blending operation can give useful effects:

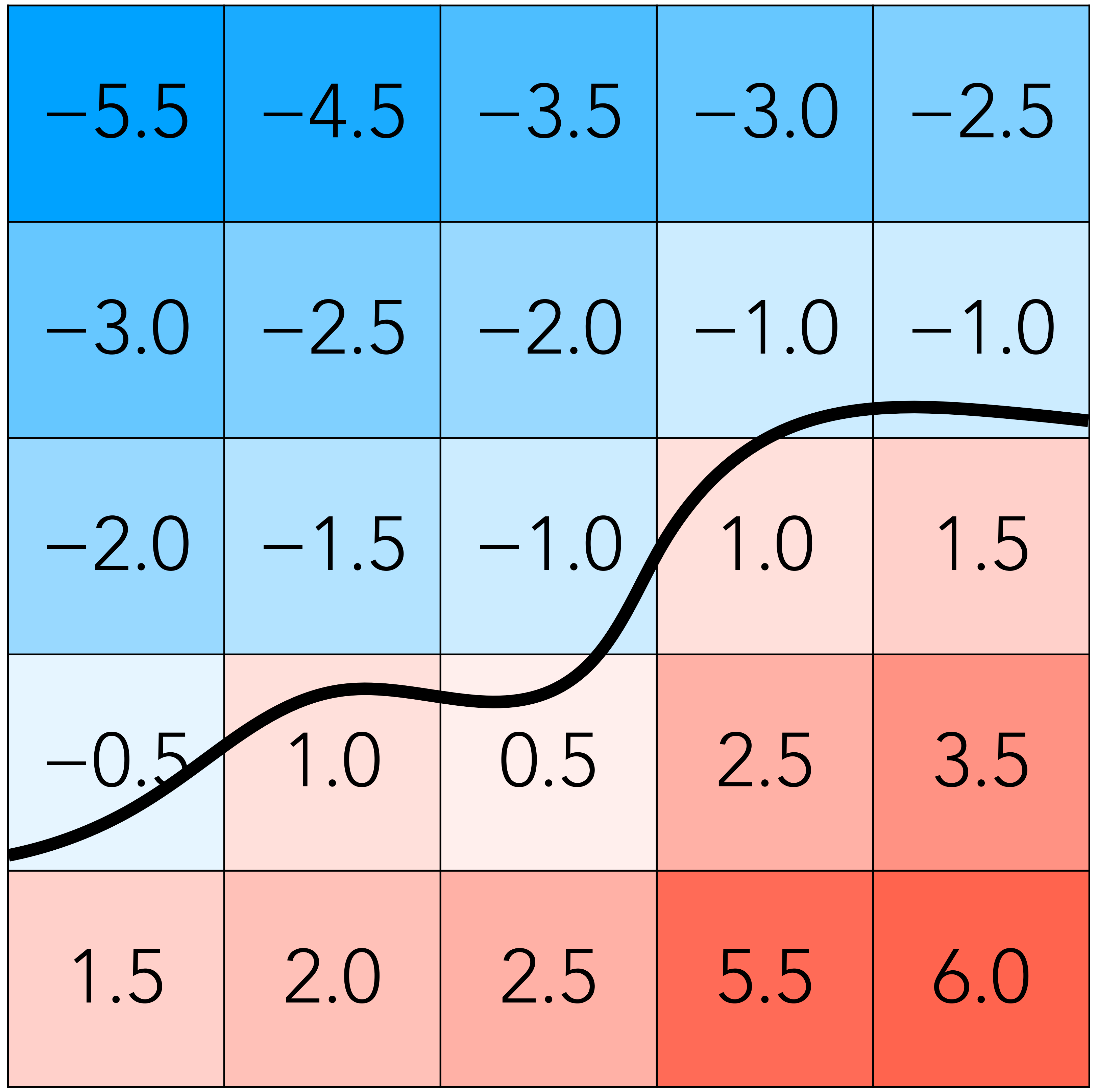
Angles et al. 2017

Level sets

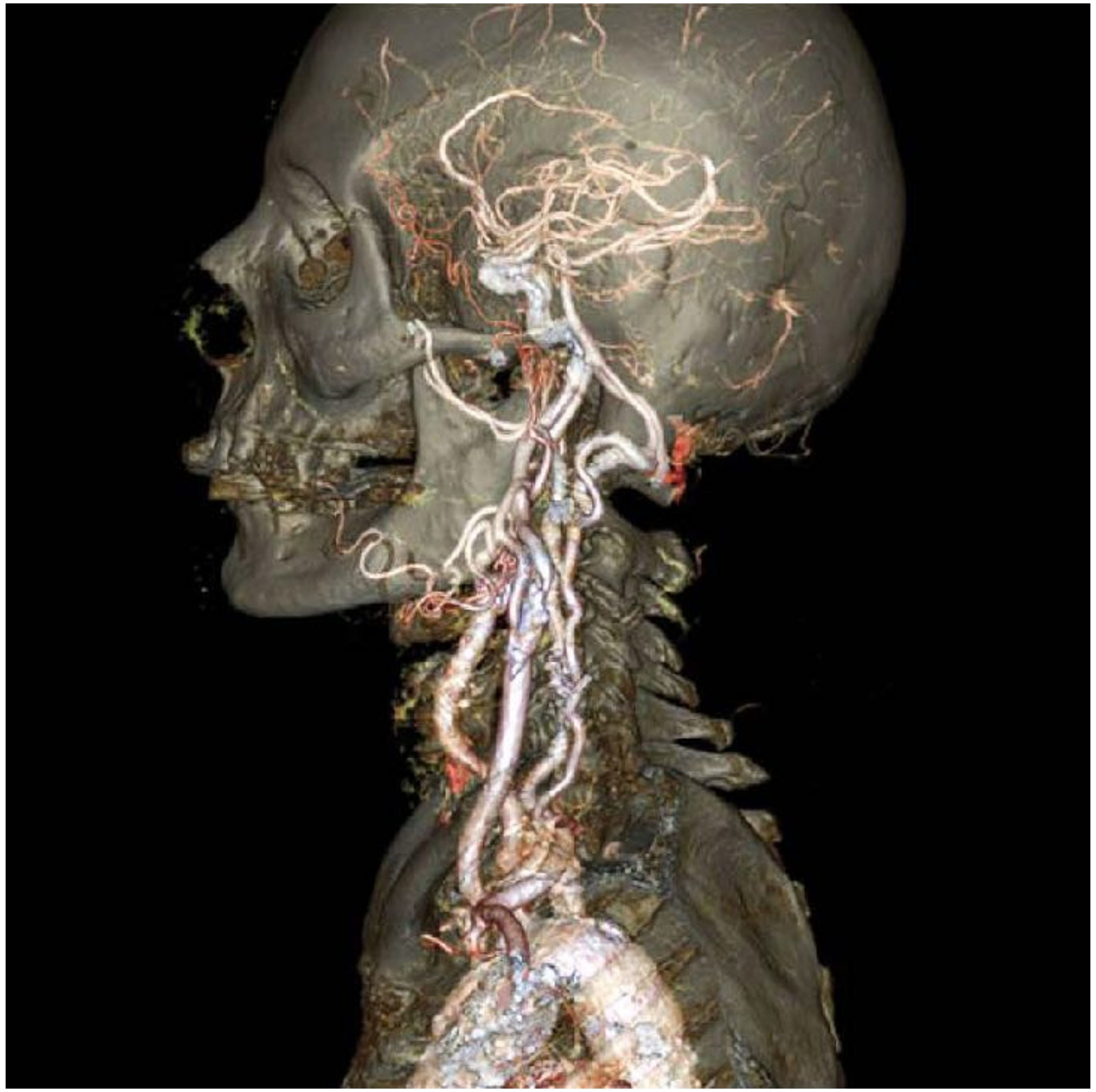
Implicit representations are useful for changing topology (merging / splitting), but usually no closed form for $f(x, y, z)$

Just store sampled values on a grid!

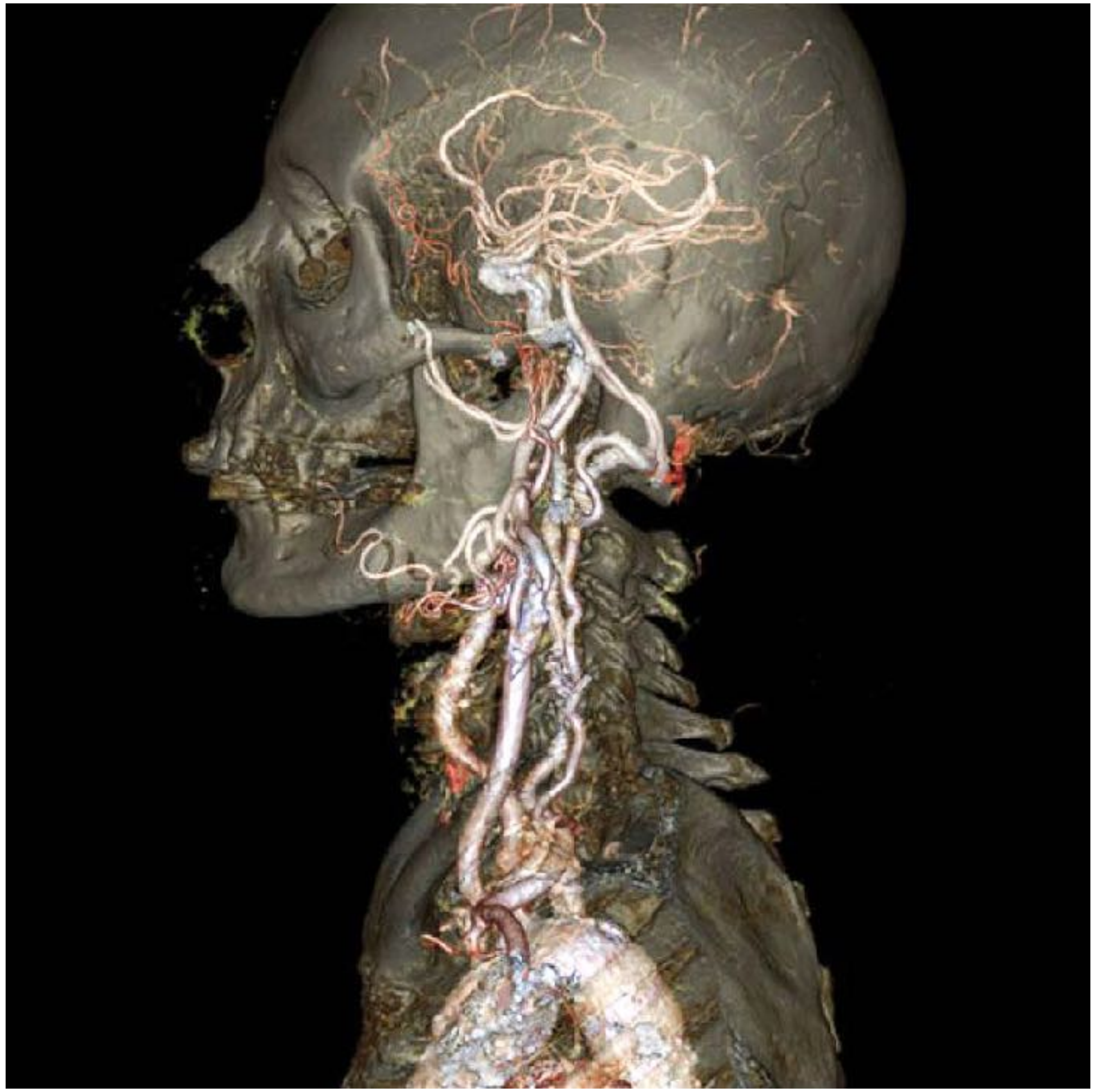
- Surface is wherever interpolated value is 0
- Modify surface by changing values on the grid

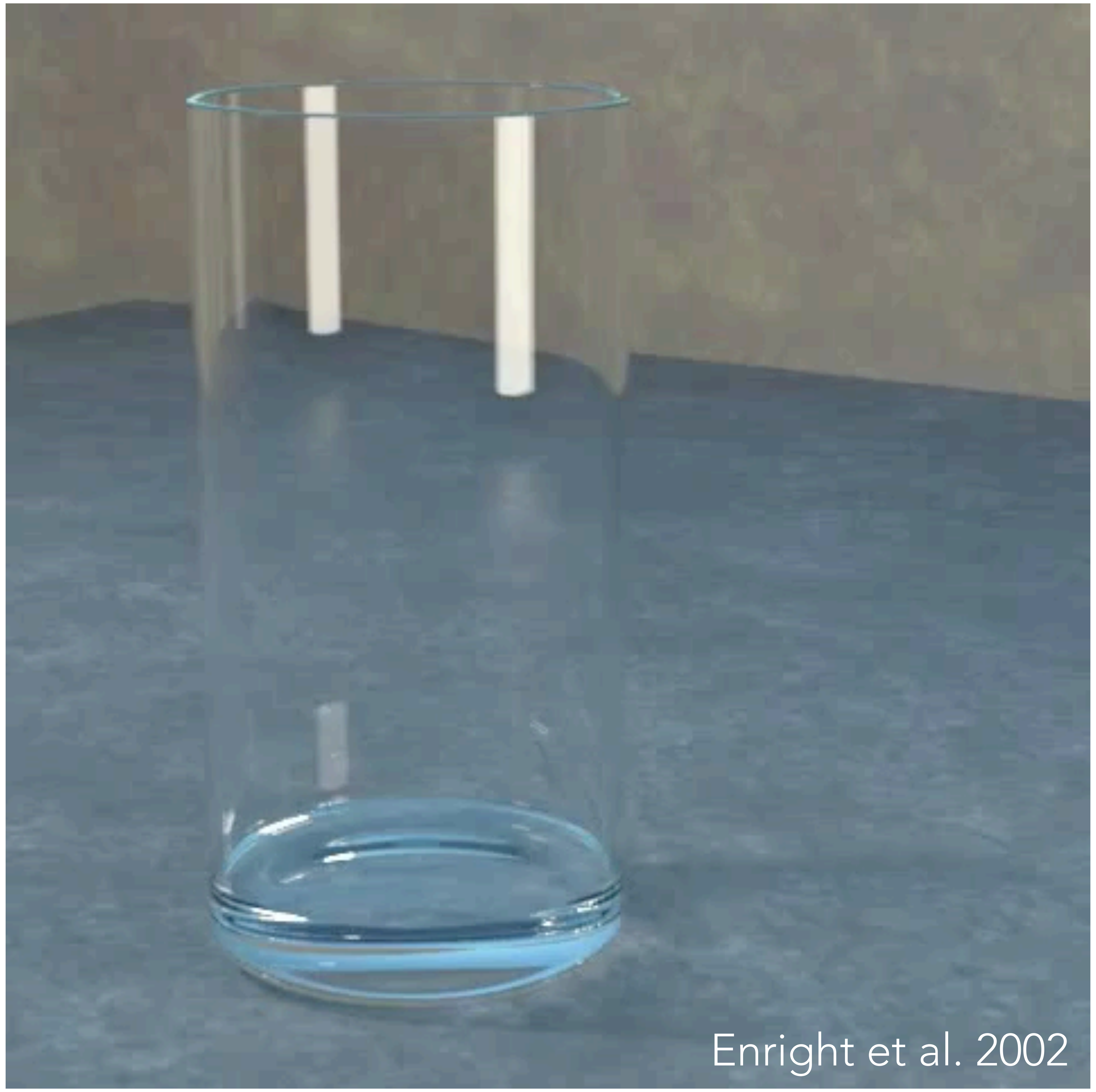


Level sets



Level sets

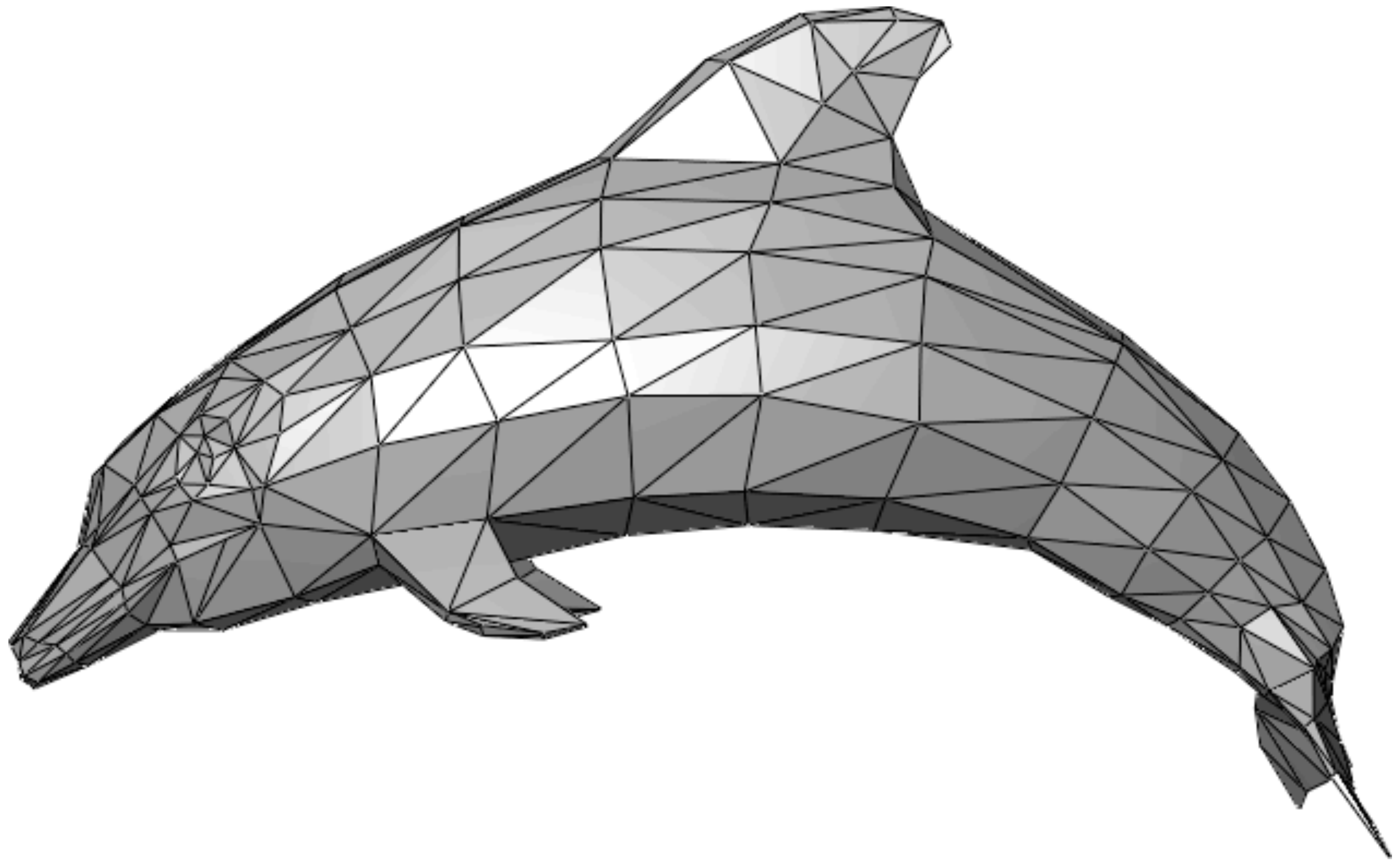




Enright et al. 2002

Explicit representations

Polygon meshes



We've already seen these.

- Vertices $(x, y, z) \in \mathbb{R}^3$
- Triangles stored via vertex indices $(i, j, k) \in \mathbb{N}^3$

How would you sample an arbitrary point on the surface (not just a vertex)?

Can also allow arbitrary polygons $(i_1, i_2, i_3, \ldots)$. But triangles and quads are most common.

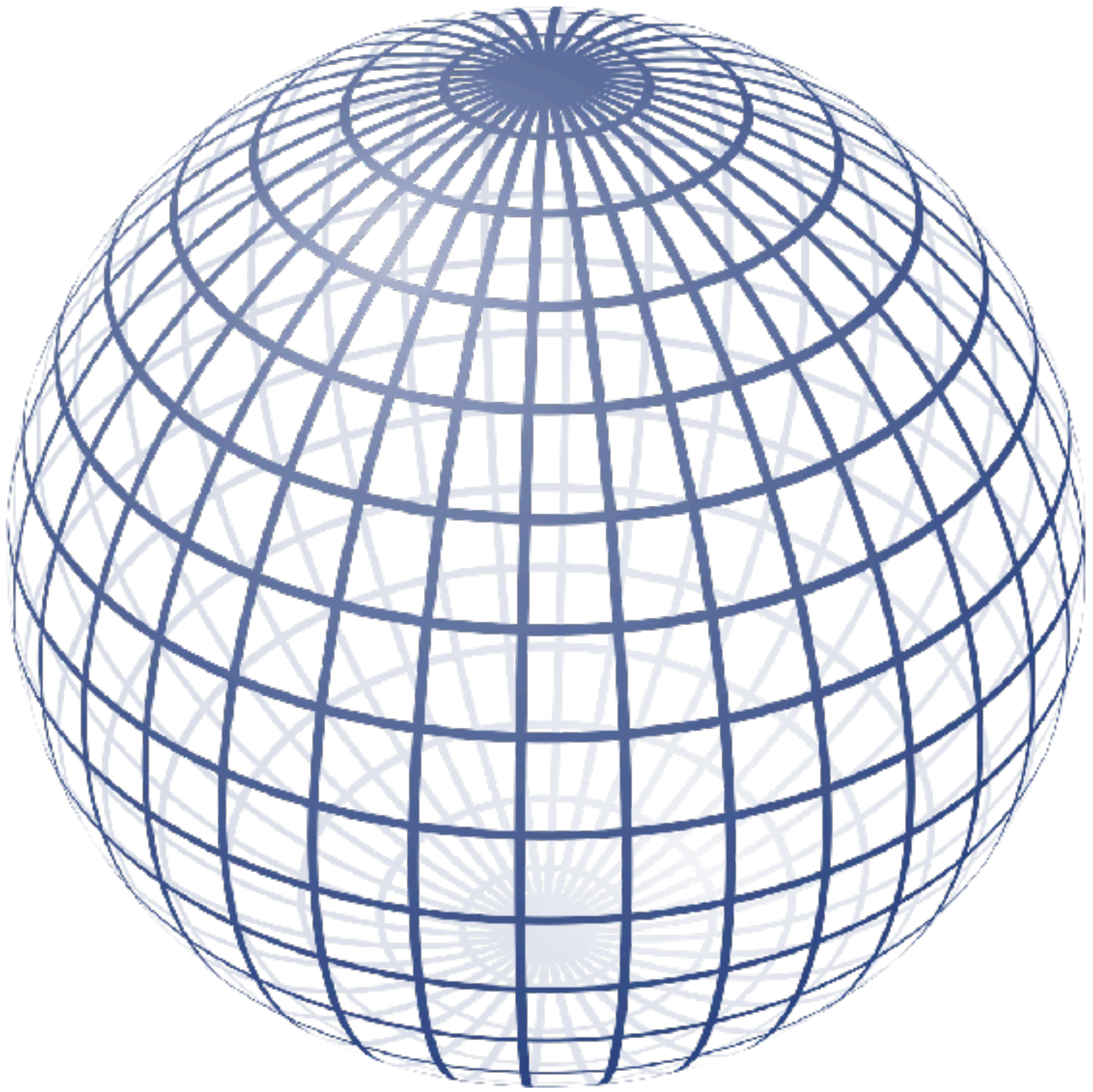
Parametric surfaces

Given by a map from (some subset of) $\mathbb{R}^2$ to $\mathbb{R}^3$.

$$x = f(u, v),$$
$$y = g(u, v),$$
$$z = h(u, v)$$

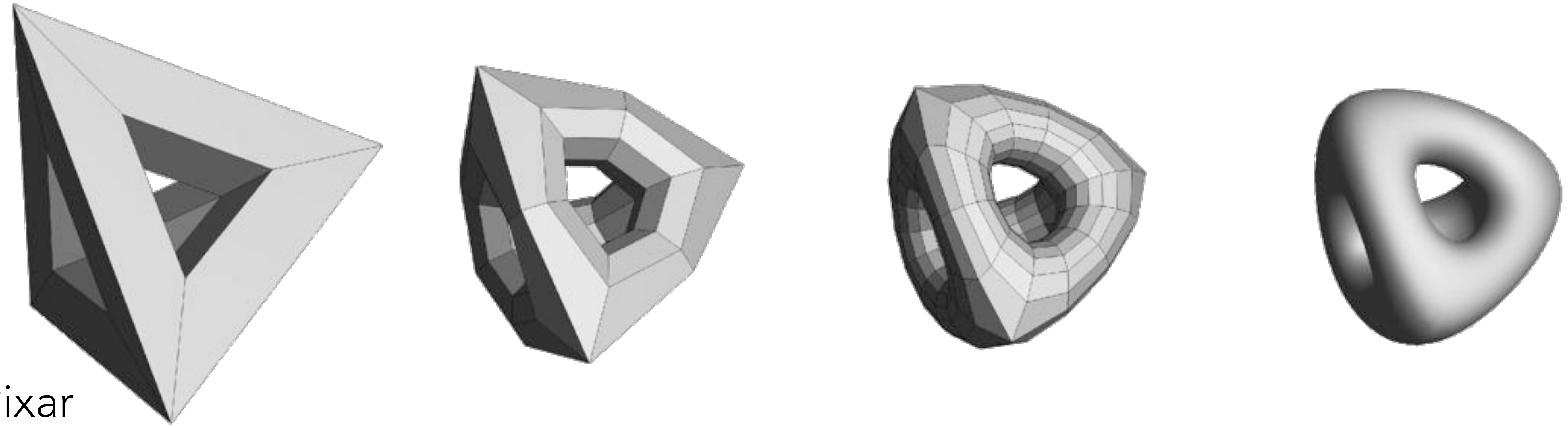
e.g. a sphere is (sin u cos v, sin u sin v, cos v)

In practice, f, g, h are usually piecewise polynomial functions a.k.a. splines



Subdivision surfaces

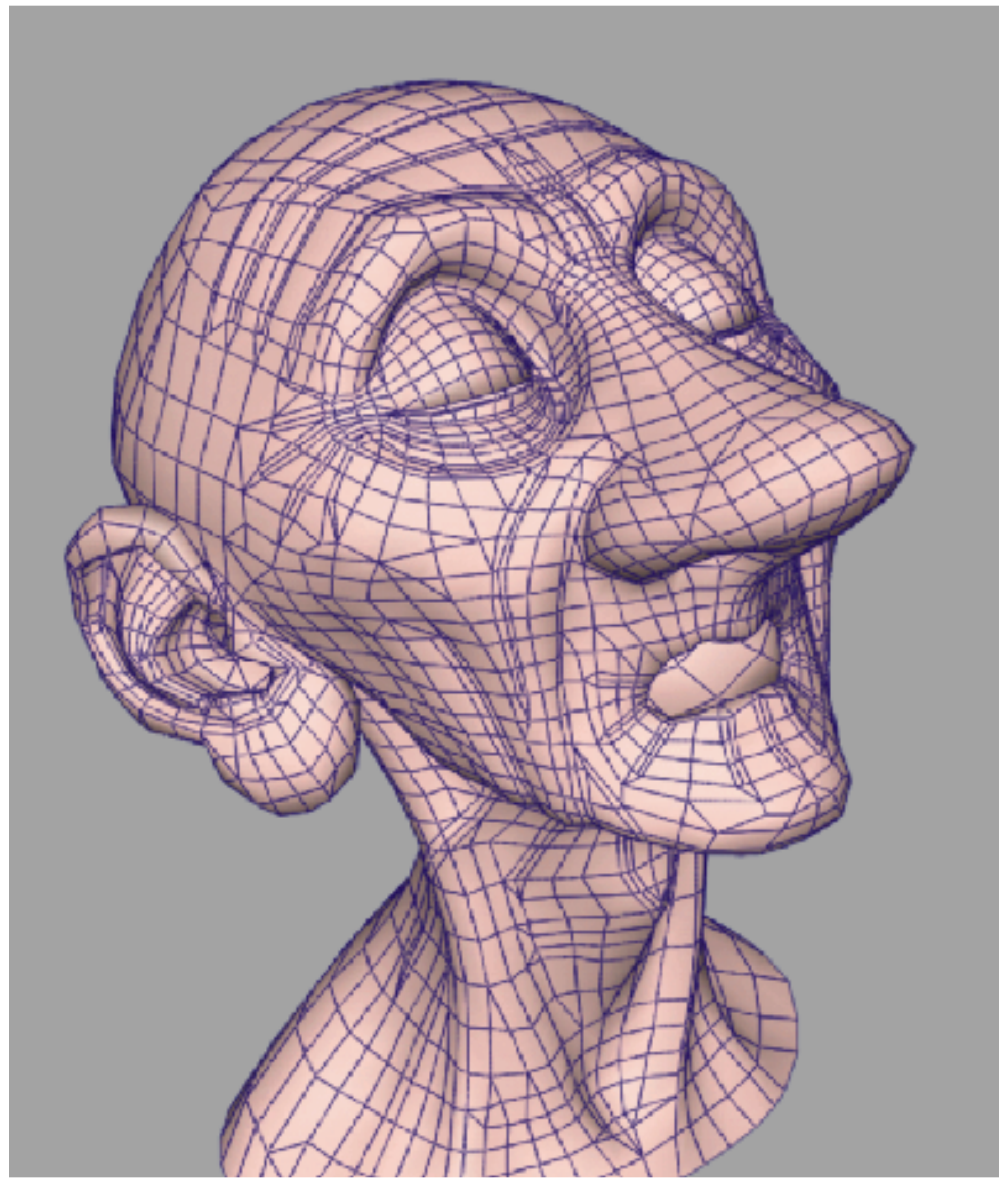
Another way to define a smooth surface: Take a coarse polygon mesh and repeatedly subdivide and smooth it.



Pixar

Various smoothing rules for triangle and polygon meshes

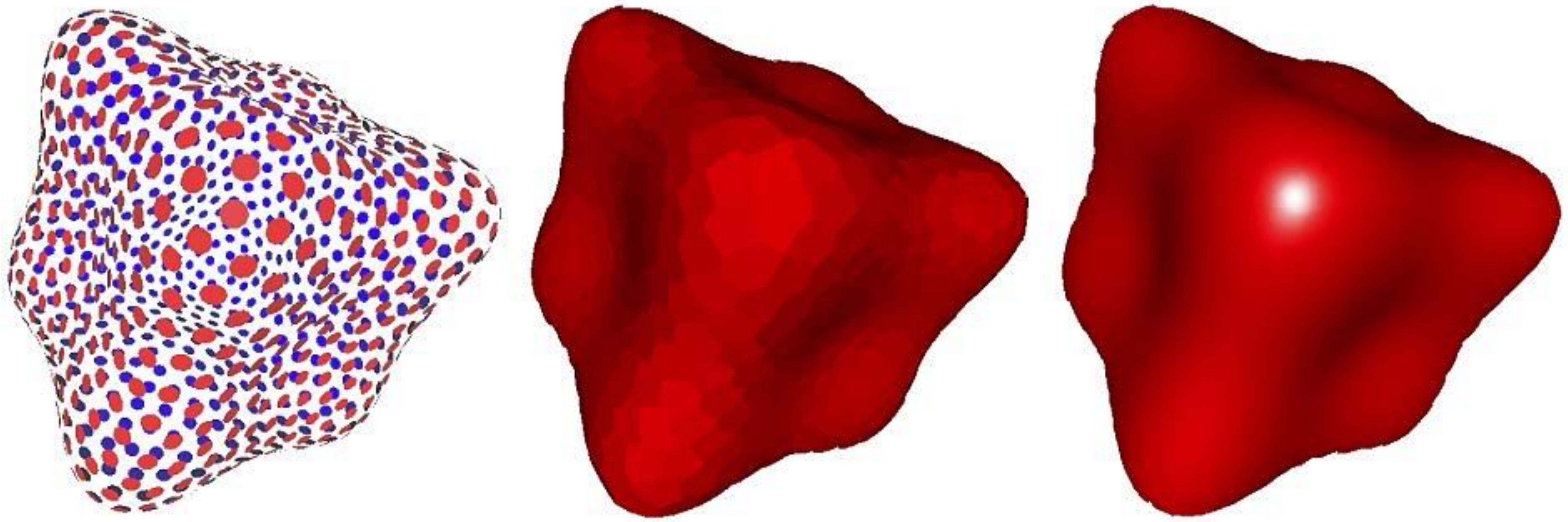
Widely used in practice for character animation



Point clouds

What if you just store a finite set of points (*x*, *y*, *z*) from the surface?

(Optionally including normals)



- Very flexible representation
- Various schemes to reconstruct surface between sampled points
- Harder to do processing, editing, simulation, …

Some things to think about

1. If you are given a closed 2D curve in parametric form, e.g. as a polygon, how would you evaluate the **signed** distance at any point (x, y)?

2. If you are given a 2D curve in implicit form, e.g. as $f(x, y) = 0$ with f given as a black box, how might you find a polygonal approximation of it?