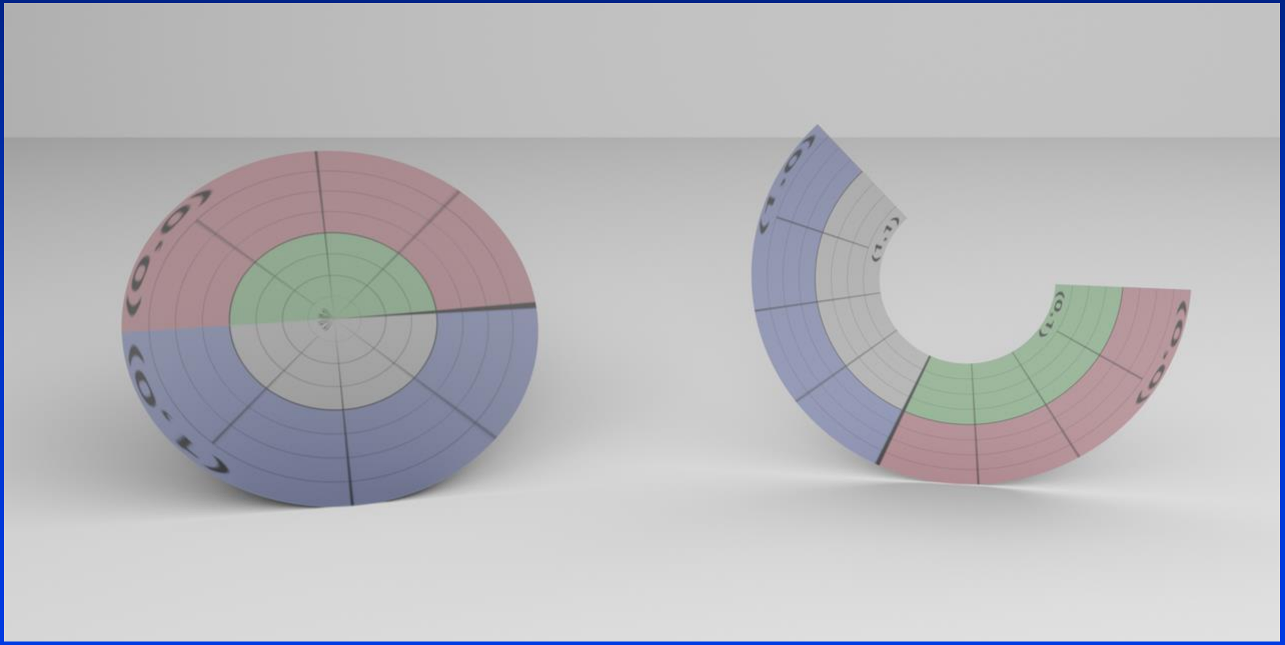
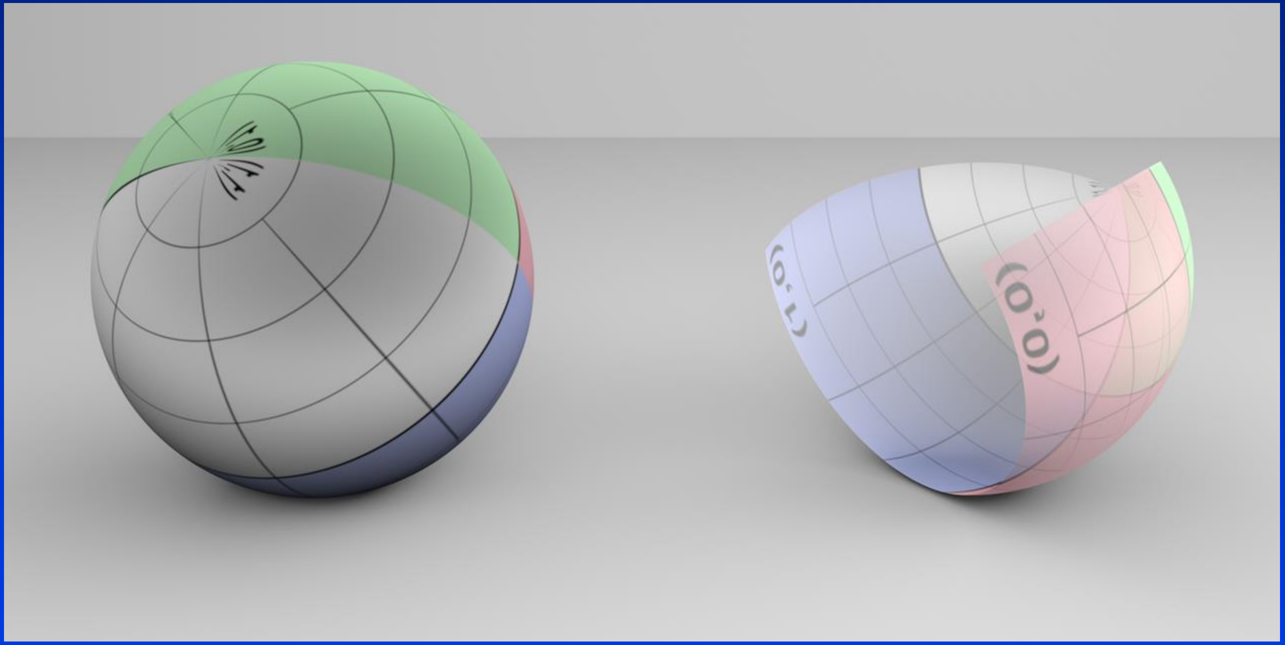
CSE328 Fundamentals of Computer Graphics: Concepts, Theory, Algorithms, and Applications

Hong Qin
Department of Computer Science
Stony Brook University (SUNY at Stony Brook)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu
http://www.cs.stonybrook.edu/~qin

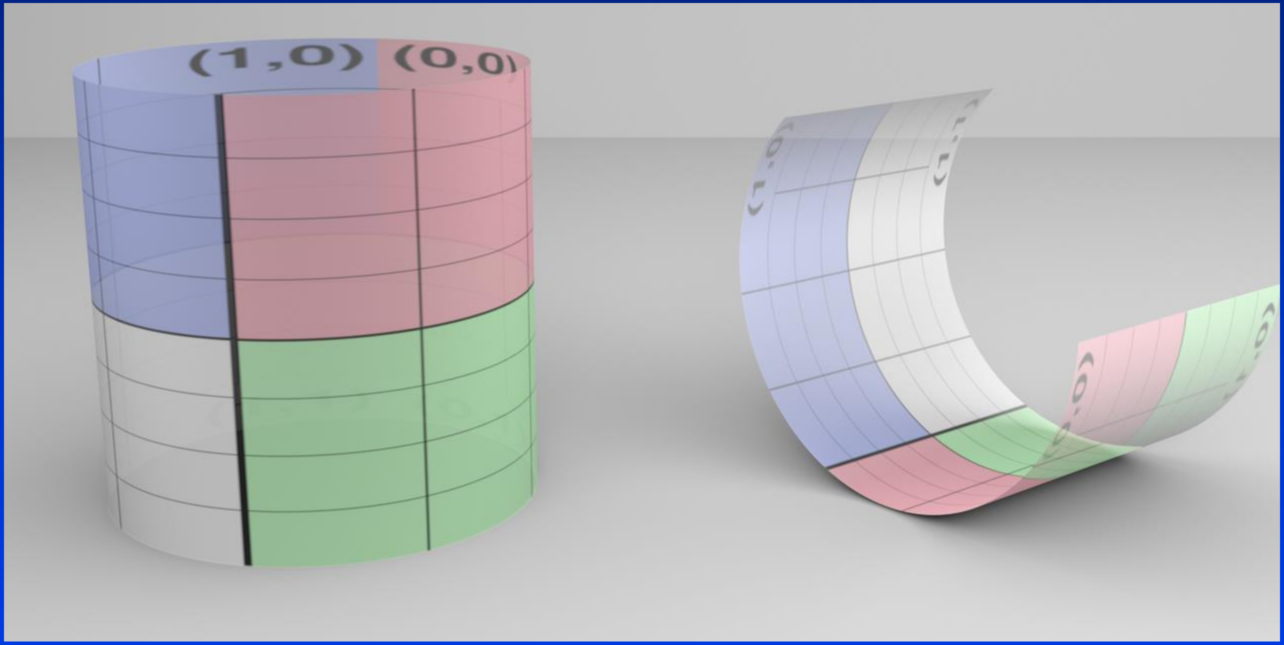
Disk



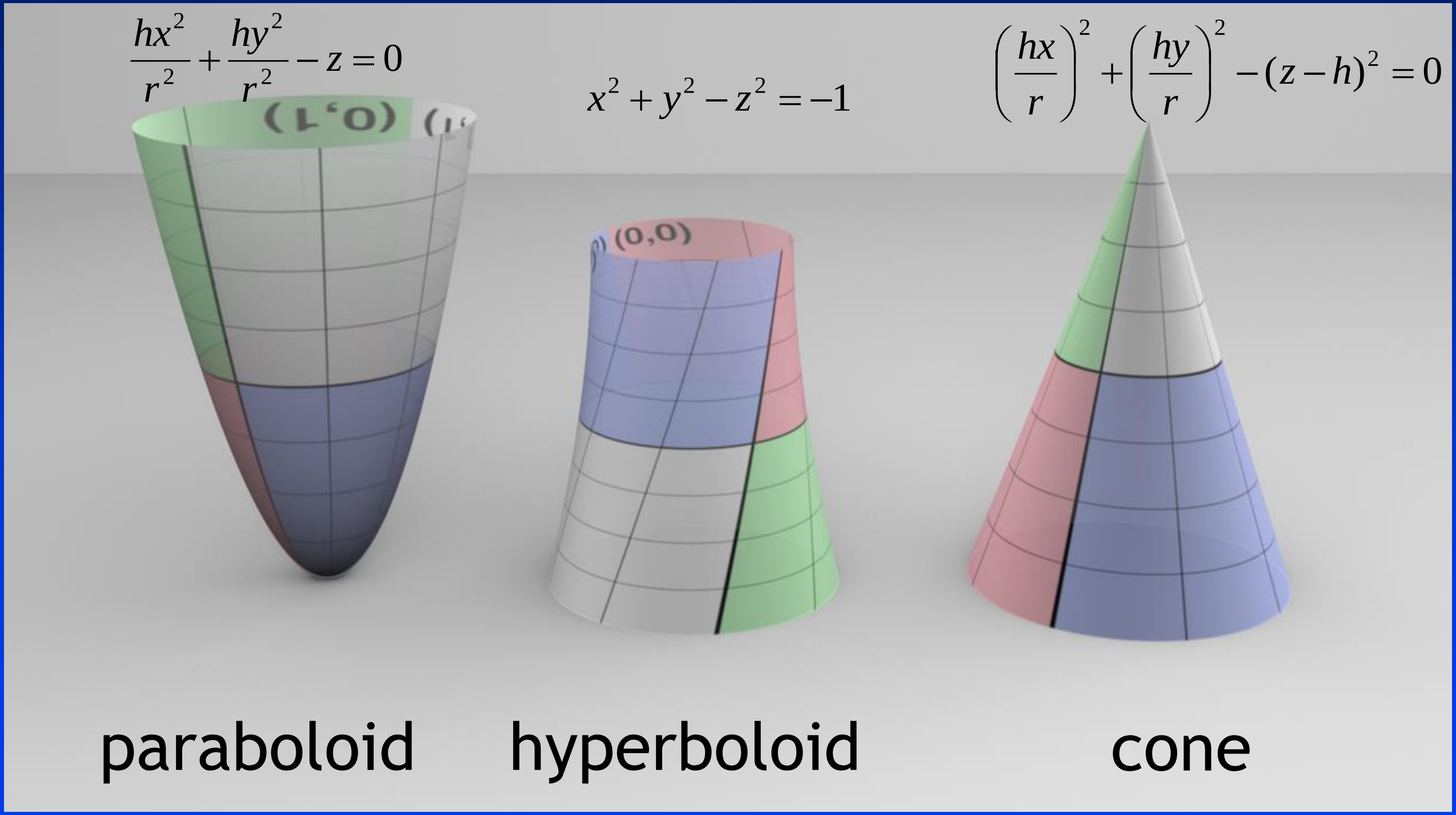
Sphere



Cylinder



Other Quadrics

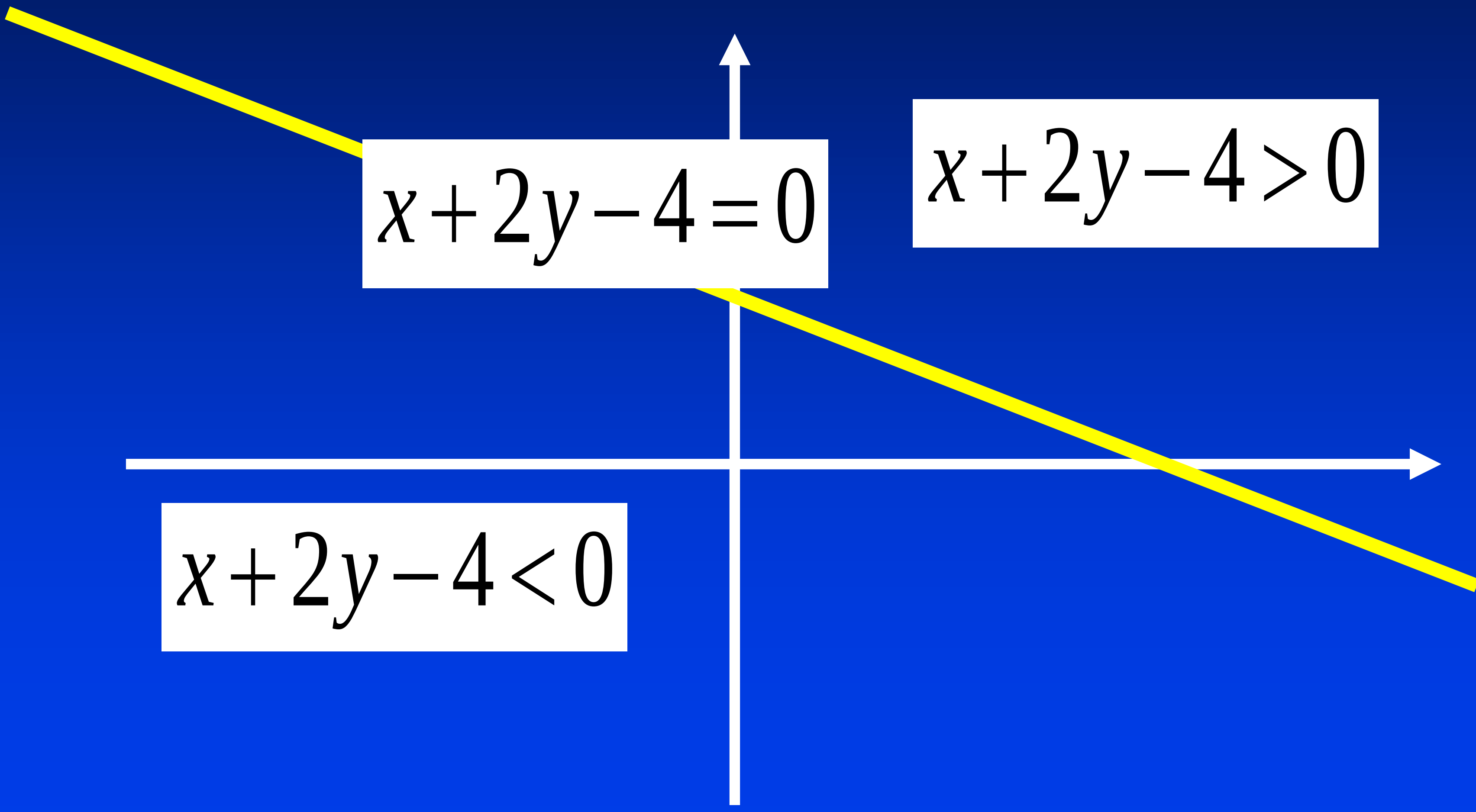


Popular Shapes

But they can also be represented by implicit functions $f(x,y,z)=0$

Implicit Surfaces

Straight Line (Implicit Representation)



Straight Line

- Mathematics (Implicit representation)

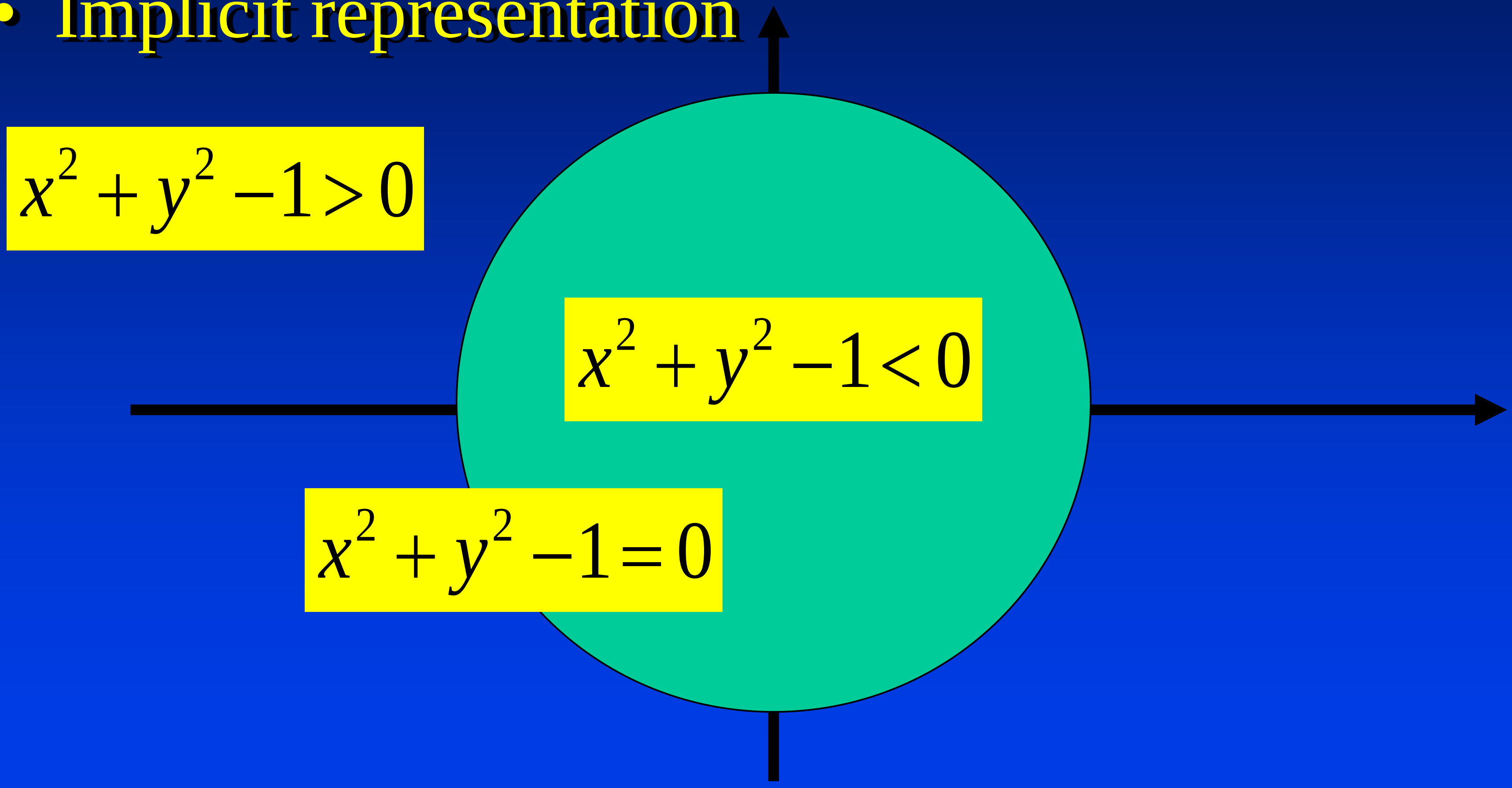
$$ax + by + c = 0$$
$$+\alpha(ax + by + c) = 0$$
$$-\alpha(ax + y + c) = 0$$

- Example

$$x + 2y - 4 = 0$$

Circle

- Implicit representation



Conic Sections

- Mathematics

$$ax^2 + 2bxy + cy^2 + dx + ey + f = 0$$

- Examples
  - Ellipse
  - Hyperbola
  - Parabola
  - Empty set
  - Point
  - Pair of lines
  - Parallel lines
  - Repeated lines

$$2x^2 + 3y^2 - 5 = 0$$
$$2x^2 - 3y^2 - 5 = 0$$
$$2x^2 + 3y = 0$$
$$2x^2 + 3y^2 + 1 = 0$$
$$2x^2 + 3y^2 = 0$$
$$2x^2 - 3y^2 = 0$$
$$2x^2 - 7 = 0$$
$$2x^2 = 0$$

Conics

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$$

$$\mathbf{PQP}^T = 0$$

$$\mathbf{Q} = \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix}$$

$$\mathbf{P} = [x \quad y \quad 1]$$

Table 2.1 Conic curve characteristics

k	$\lvert\mathbf{Q}\rvert$	Other conditions	Type
0	$\neq 0$		Parabola
0	0	$C \neq 0, E^2 - CF > 0$	Two parallel real lines
0	0	$C \neq 0, E^2 - CF = 0$	Two parallel coincident lines
0	0	$C \neq 0, E^2 - CF < 0$	Two parallel imaginary lines
0	0	$C = B = 0, D^2 - AF > 0$	Two parallel real lines
0	0	$C = B = 0, D^2 - AF = 0$	Two parallel coincident lines
0	0	$C = B = 0, D^2 - AF < 0$	Two parallel imaginary lines
<0	0		Point ellipse
<0	$\neq 0$	$-C\lvert\mathbf{Q}\rvert > 0$	Real ellipse
<0	$\neq 0$	$-C\lvert\mathbf{Q}\rvert < 0$	Imaginary ellipse
<0	$\neq 0$		Hyperbola
<0	0		Two intersecting lines

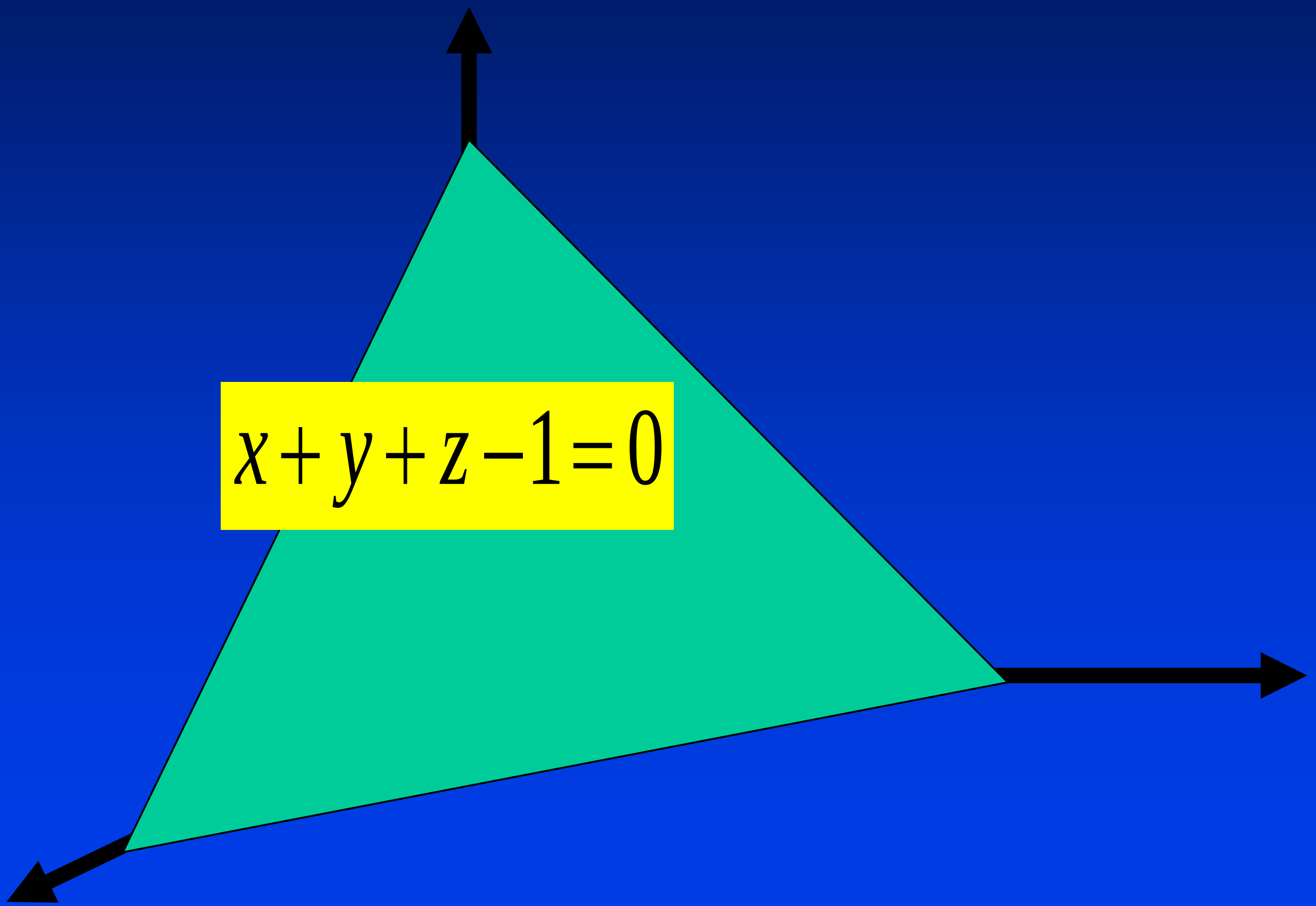
Conics

- Parametric equations of conics
- Generalization to higher-degree curves
- How about non-planar (spatial) curves

Plane Equation and its Normal

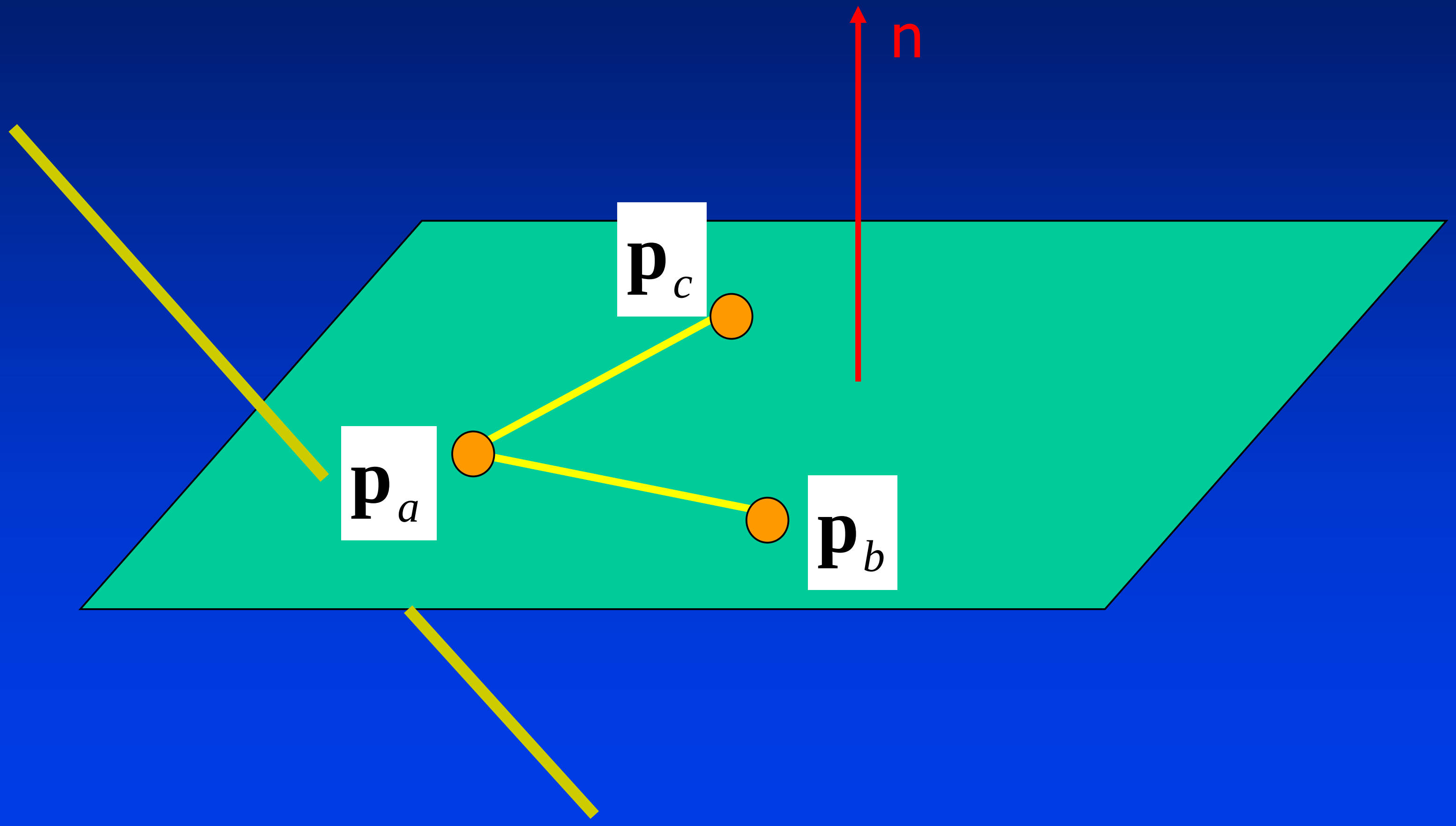
- Chapter 4.7!!!

Plane



$$x + y + z - 1 = 0$$

Plane and Intersection



Plane

- Example $x+y+z-1=0$
- General plane equation $ax+by+cz+y=0$
- Normal of the plane $$\mathbf{n}=\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$
- Arbitrary point on the plane

$$\mathbf{p}_a=\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

Plane

- Plane equation derivation

$$(x-a_x)a+(y-a_y)b+(z-a_z)c=0$$
$$ax+by+cz-(a_x a+a_y b+a_z c)=0$$

- Parametric representation (given three points on the plane and they are non-collinear!)

$$\mathbf{p}(u,v)=\mathbf{p}_a+(\mathbf{p}_b-\mathbf{p}_a)u+(\mathbf{p}_c-\mathbf{p}_a)v$$

Plane

- Explicit expression (if c is non-zero)

$$z = -\frac{1}{c}(ax + by + d)$$

- Line-plane intersection

$$\mathbf{l}(u) = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0)u$$

$$(\mathbf{n})(\mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0)u) + d = 0$$

$$u = -\frac{\mathbf{np}_0}{\mathbf{np}_1 - \mathbf{np}_0} = -\frac{plane(\mathbf{p}_0)}{plane(\mathbf{p}_1) - plane(\mathbf{p}_0)}$$

Circle

- Implicit equation $x^2 + y^2 - 1 = 0$
- Parametric function

$$\mathbf{c}(\theta) = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

$$0 <= \theta <= 2\pi$$

- Parametric representation using rational polynomials (the first quadrant)

$$x(u) = \frac{1-u^2}{1+u^2}$$

$$y(u) = \frac{2u}{1+u^2}$$

$$u \in [0,1]$$

- Parametric representation is not unique!

What are Implicit Surfaces?

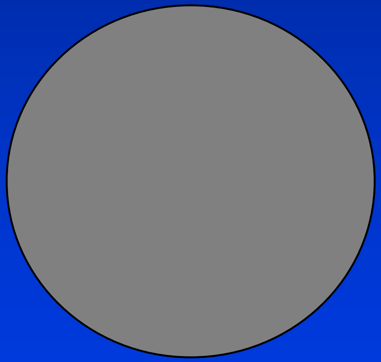
- 2D Geometric shapes that exist in 3D space, frequently defined by (algebraic) functions
- Surface representation through a function f(x, y, z) = 0
- Most methods of analysis assume f is continuous and not everywhere 0.
- Some objects are easy represent this way
  - Spheres, ellipses, and similar
  - More generally, quadratic surfaces:

$$ax^2 + bx + cy^2 + dy + ez^2 + fz + g = 0$$

  - Shapes depends on all the parameters *a,b,c,d,e,f,g*

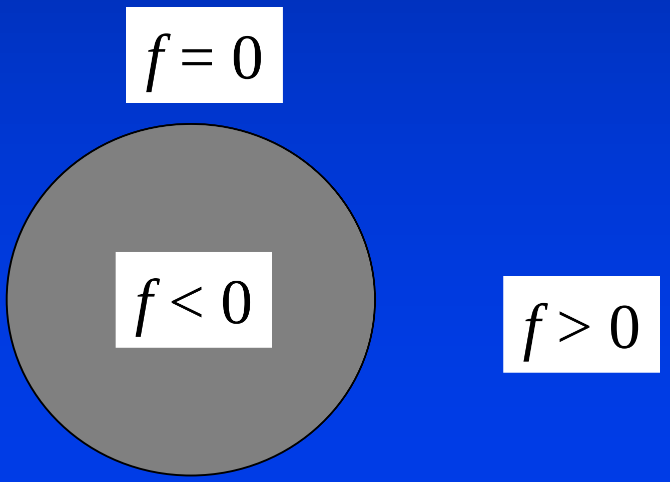
Example of an Implicit Surface

- 3D Sphere centered at the origin
  - $x^2 + y^2 + z^2 = r^2$
  - $x^2 + y^2 + z^2 - r^2 = 0$



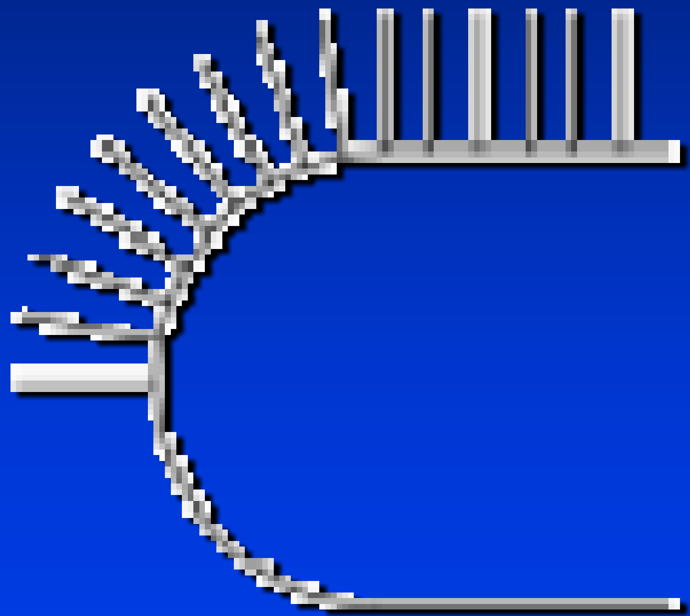
Point Classification

- Inside Region: f < 0
- Outside Region: f > 0
- Or vice versa depending on the function



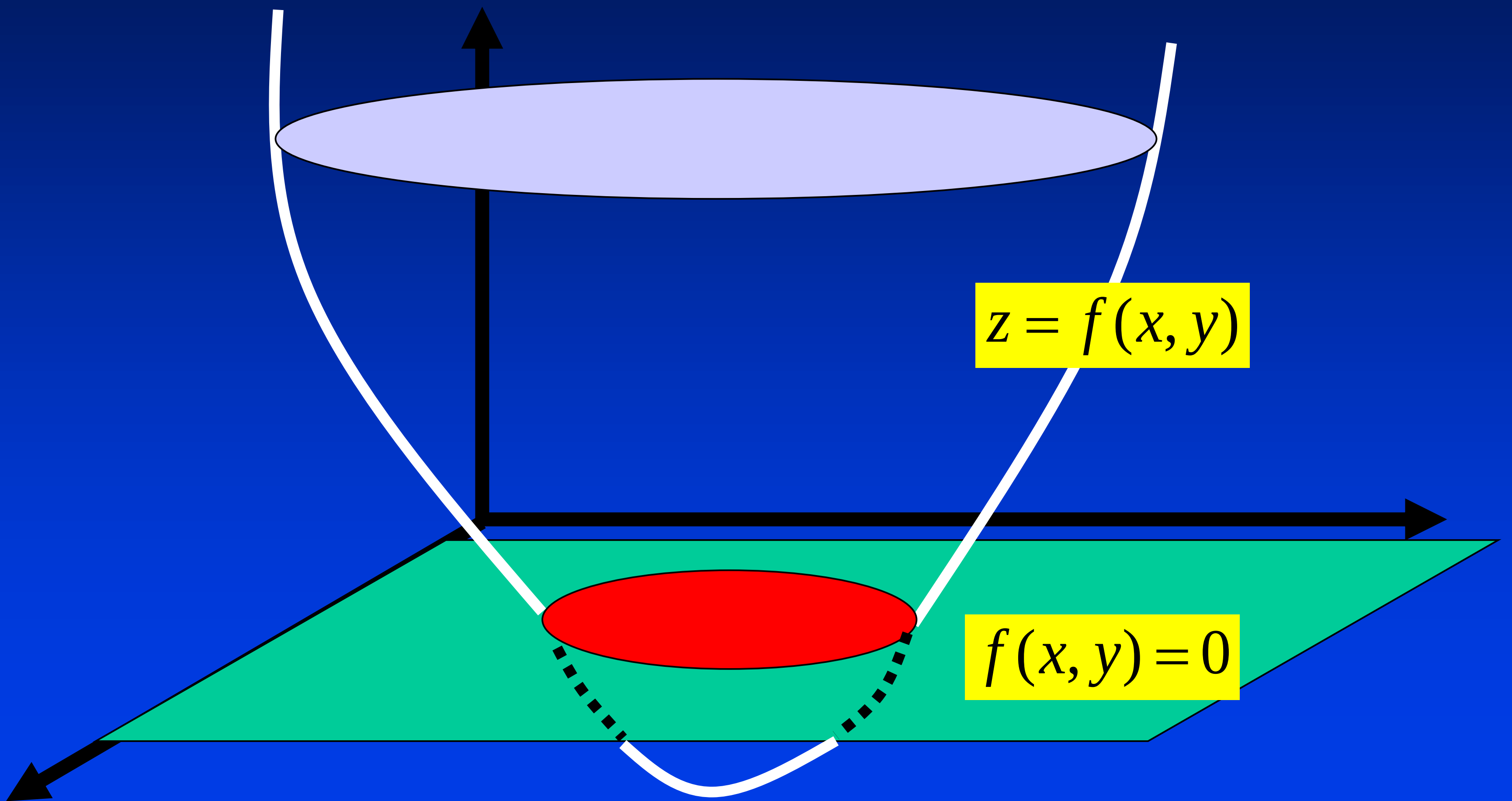
Surface Normals

- Usually gradient of the function
  – $\nabla f(x,y,z) = (\delta f/\delta x, \delta f/\delta y, \delta f/\delta z)$
- Points at increasing f



Properties of Implicits

- Easy to check if a point is inside the implicit surface or NOT
  - Simply evaluate f at that point
- Fairly easy to check ray intersection
  - Substitute ray equation into f for simple functions
  - Binary search


$z = f(x, y)$
$f(x, y) = 0$

Implicit Equations for Curves

- Describe an implicit relationship
- Planar curve (point set) $\{(x,y) \mid f(x,y)=0\}$
- The implicit function is not unique

$$\{(x,y) \mid +\alpha f(x,y)=0\}$$
$$\{(x,y) \mid -\alpha f(x,y)=0\}$$

- Comparison with parametric representation

$$\mathbf{p}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}$$

Implicit Equations for Curves

- Implicit function is a level-set

$$\begin{cases} z = & f(x, y) \\ z = & 0 \end{cases}$$

- Examples (straight line and conic sections)

$$ax + by + c = 0$$

$$ax^2 + 2bxy + cy^2 + dx + ey + f = 0$$

- Other examples
  – Parabola, two parallel lines, ellipse, hyperbola, two intersection lines

Implicit Functions for Curves

- Parametric equations of conics
- Generalization to higher-degree curves
- How about non-planar (spatial) curves

Types of Implicit Surfaces

- Mathematic
  - Polynomial or *Algebraic*
  - Non polynomial or *Transcendental*
    - Exponential, trigonometric, etc.
- Procedural
  - Black box function

Implicit Equations for Surfaces

- Surface mathematics $\{(x,y,z) \mid f(x,y,z)=0\}$
- Again, the implicit function for surfaces is not unique

$$\{(x,y,z) \mid +\alpha f(x,y,z)=0\}$$
$$\{(x,y,z) \mid -\alpha f(x,y,z)=0\}$$

- Comparison with parametric representation

$$\mathbf{p}(u,v)=\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix}$$

Implicit Equations for Surfaces

- Surface defined by implicit function is a level-set

$$\begin{cases} w = & f(x, y, z) \\ w = & 0 \end{cases}$$

- Examples
  - Plane, quadric surfaces, tori, superquadrics, blobby objects
- Parametric representation of quadric surfaces
- Generalization to higher-degree surfaces

Quadric Surfaces

- Implicit functions
- Examples

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + jz + k = 0$$

  - Sphere
  - Cylinder
  - Cone
  - Paraboloid
  - Ellipsoid
  - Hyperboloid

$$x^2 + y^2 + z^2 - 1 = 0$$
$$x^2 + y^2 - 1 = 0$$
$$x^2 + y^2 - z^2 = 0$$
$$x^2 + y^2 + z = 0$$
$$2x^2 + 3y^2 + 4z^2 - 5 = 0$$
$$x^2 + y^2 - z^2 + 4 = 0$$

- More
  - Two parallel planes, two intersecting planes, single plane, line, point

Quadric Surfaces

- Implicit surface equation

$$f(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0$$

- An alternative representation

$$P^{\mathrm{T}} \bullet Q \bullet P = 0$$

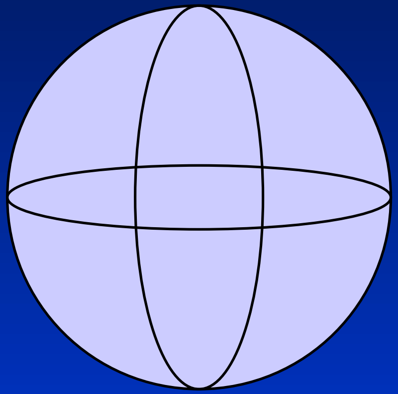
with

$$Q = \begin{bmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{bmatrix} \qquad P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Quadrics: Parametric Representation

- Sphere

$$x^2 + y^2 + z^2 - r^2 = 0$$
$$x = r\cos(\alpha)\cos(\beta)$$
$$y = r\cos(\alpha)\sin(\beta)$$
$$z = r\sin(\alpha)$$
$$\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]; \beta \in [-\pi, \pi]$$



- Ellipsoid

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$
$$x = a\cos(\alpha)\cos(\beta)$$
$$y = b\cos(\alpha)\sin(\beta)$$
$$z = c\sin(\alpha)$$
$$\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]; \beta \in [-\pi, \pi]$$

- Geometric meaning of these parameters

Quadric Surfaces

- Modeling advantages
  - computing the surface normal
  - testing whether a point is on the surface
  - computing *z* given *x* and *y*
  - calculating intersections of one surface with another

Generalization

- Higher-degree polynomials

$$\sum_i \sum_j \sum_k a_{ijk} x^i y^j z^k = 0$$

- Non polynomials

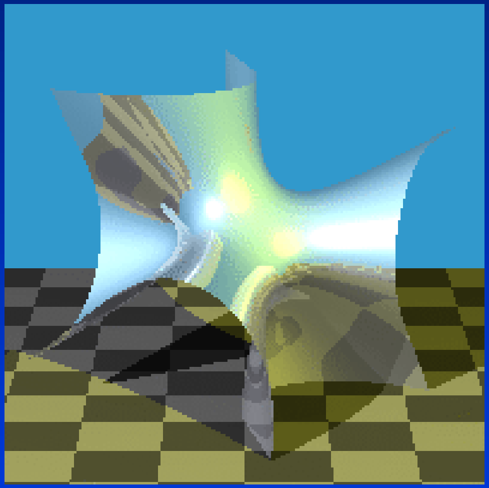
Algebraic Function

- Parametric representation is popular, but…
- Formulation

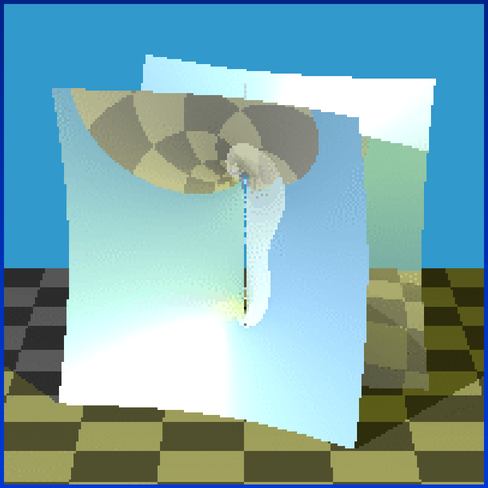
$$\sum_i \sum_j \sum_k a_{ijk} x^i y^j z^k = 0$$

- Properties…
  - Powerful, but lack of modeling tools

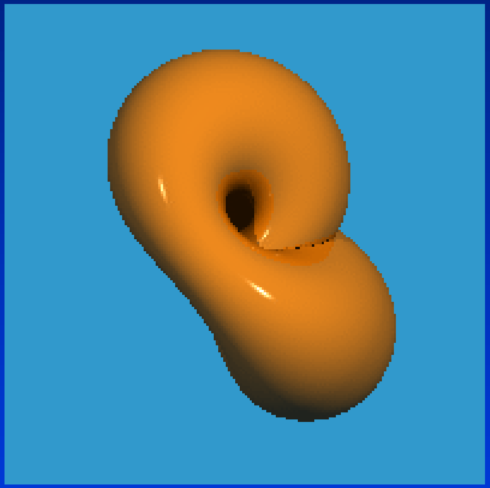
Algebraic Surfaces



Cubic

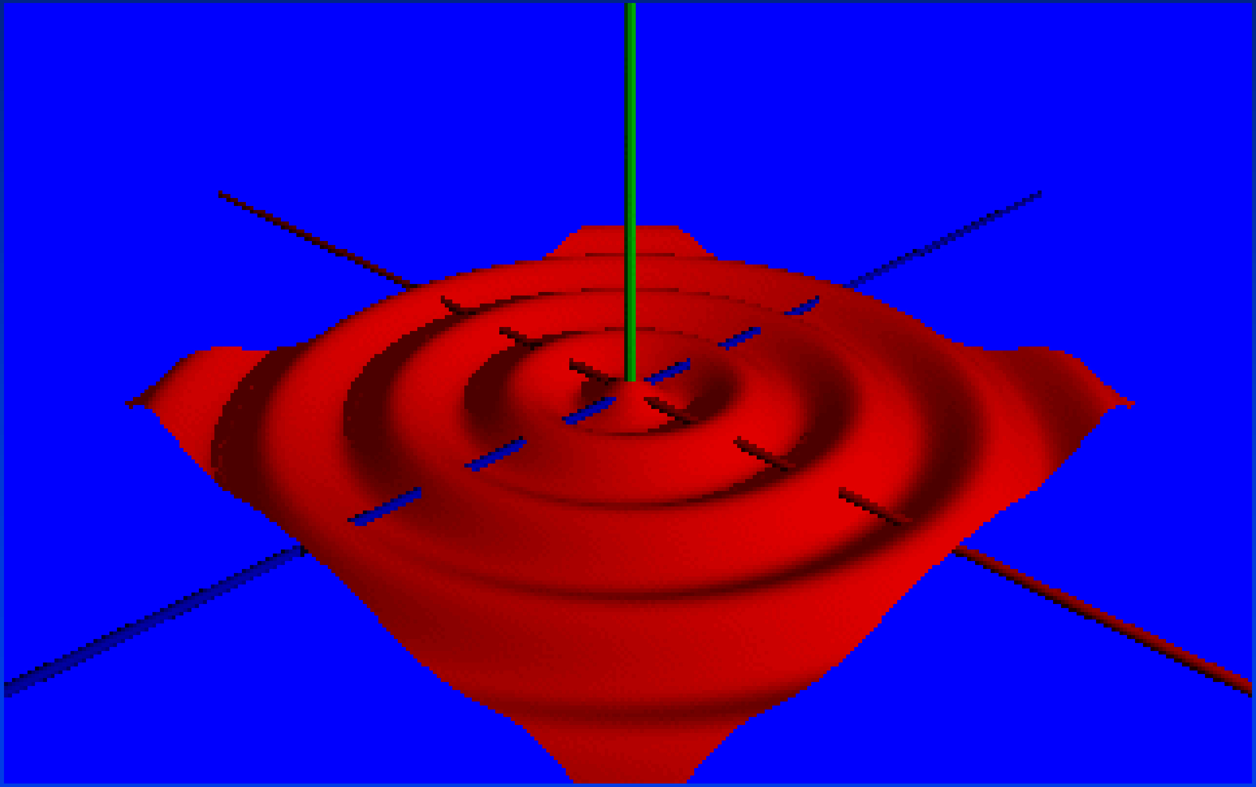


Degree 4



Degree 6

Non-Algebraic Surfaces



Spatial Curves

- Intersection of two surfaces

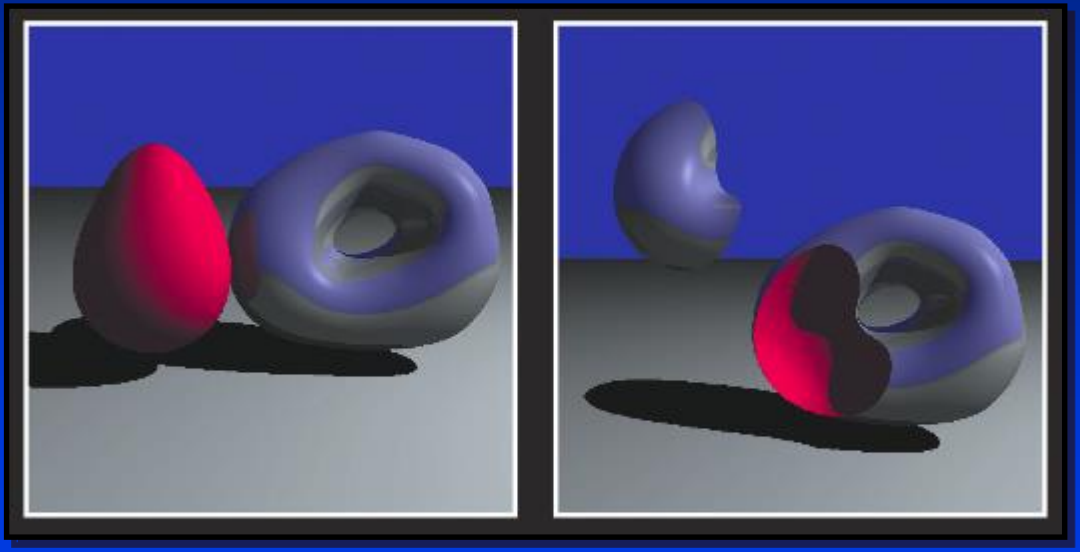
$$\begin{cases} f(x,y,z)=0 \\ g(x,y,z)=0 \end{cases}$$

Algebraic Solid

- Half space $\{(x,y,z) \mid f(x,y,z) <= 0\}; or$
  $\{(x,y,z) \mid f(x,y,z) >= 0\}$
- Useful for complex objects (refer to notes on solid modeling)

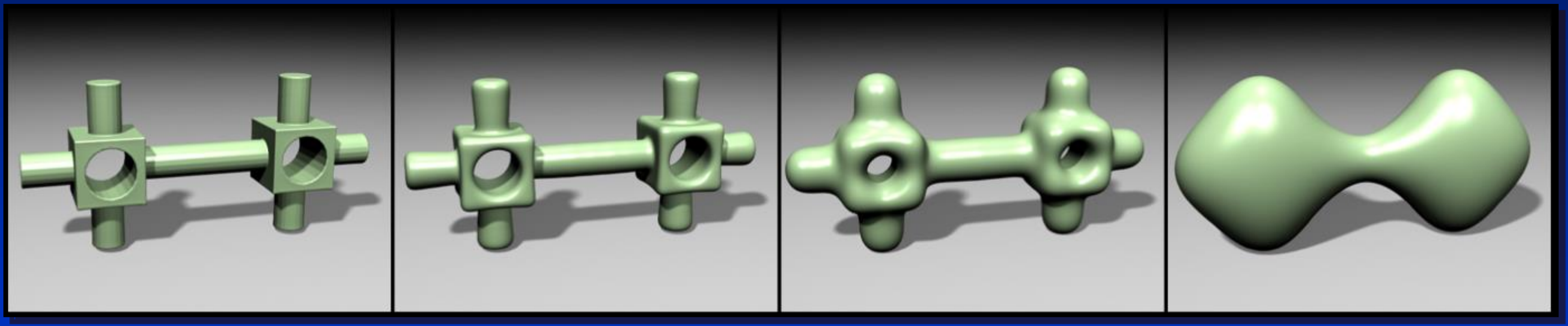
$$\mathbf{f}(x,y,z) = \begin{bmatrix} f_1(x,y,z) \\ f_2(x,y,z) \\ f_3(x,y,z) \\ \cdots \end{bmatrix} = \mathbf{0}$$

Implicit Surfaces



CSG on implicit surfaces

Implicit Surfaces



Object made by CSG
Converted to polygons
Converted to implicit surface

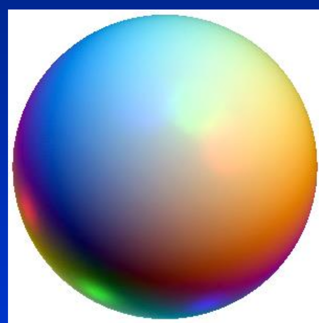
Implicit Surfaces: Applications

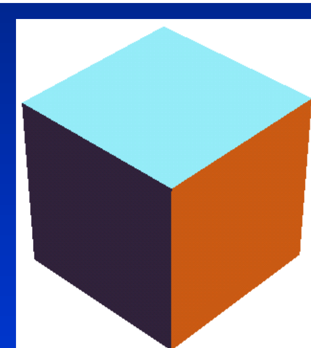
- Zero sets of implicit functions.

$$f(x, y, z) = 0$$

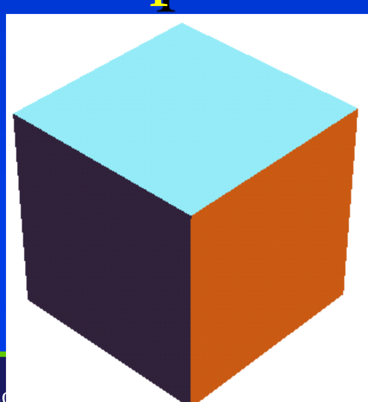
$$r^2 - x^2 - y^2 - z^2 > 0$$

$$(l - |x| > 0) \cap (l - |y| > 0) \cap (l - |z| > 0)$$





- CSG operations.



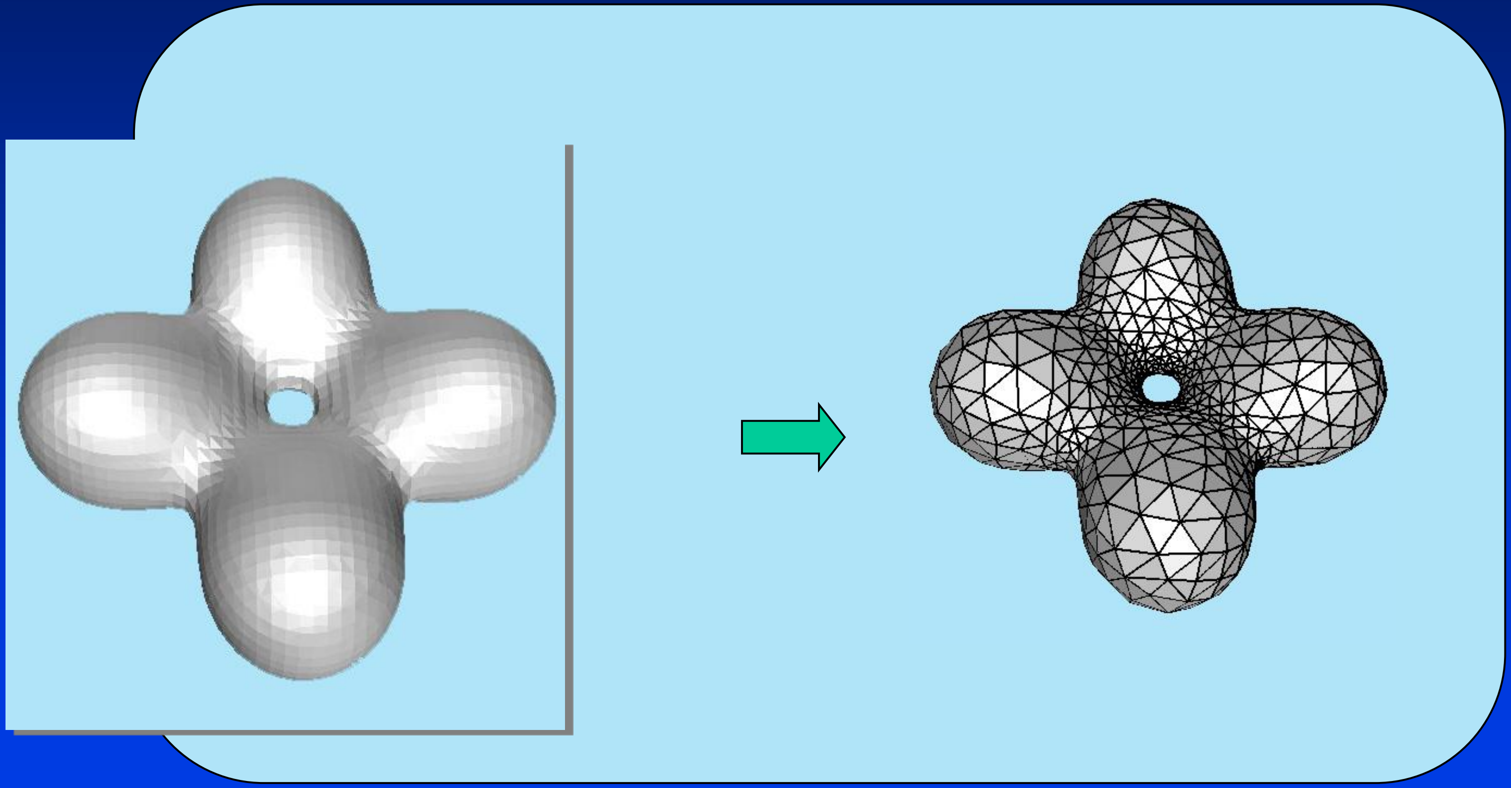
-

=

Polygonization

- Conversion of implicit surface to polygonal mesh
- Display implicit surface using polygons
- Real-time approximate visualization method
- Two steps
  - Partition space into cells
  - Fit a polygon to surface in each cell

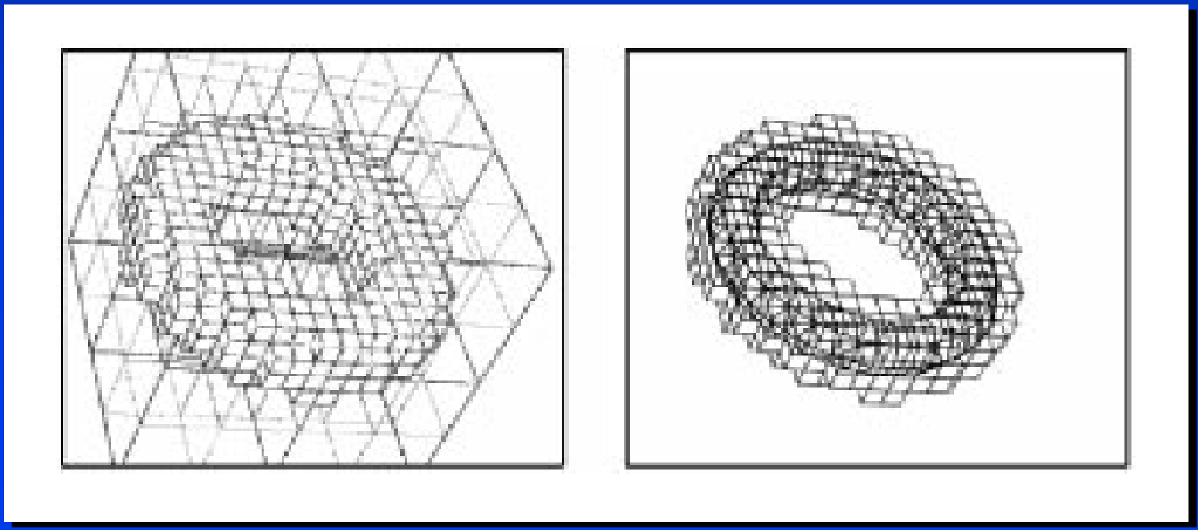
Implicit Surface (Polygonal Representation)



$$F: R^3 => R, \Sigma = F^{-1}(0)$$

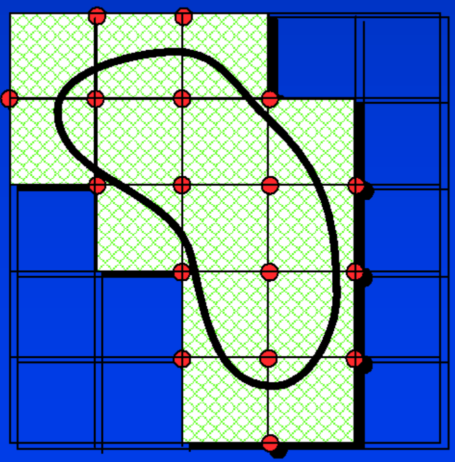
Spatial Partitioning

- Subdivision
  - Start with root cell and subdivide
  - Continue subdividing
  - traverse cells



Spatial Partitioning

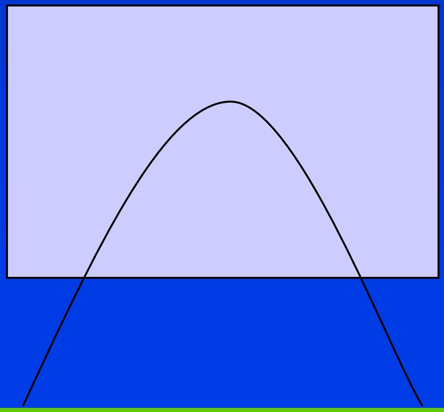
- Exhaustive enumeration
  - Divide space into regular lattice of cells
  - Traverse cells in order to arrive at polygonization



Space Partitioning Criteria

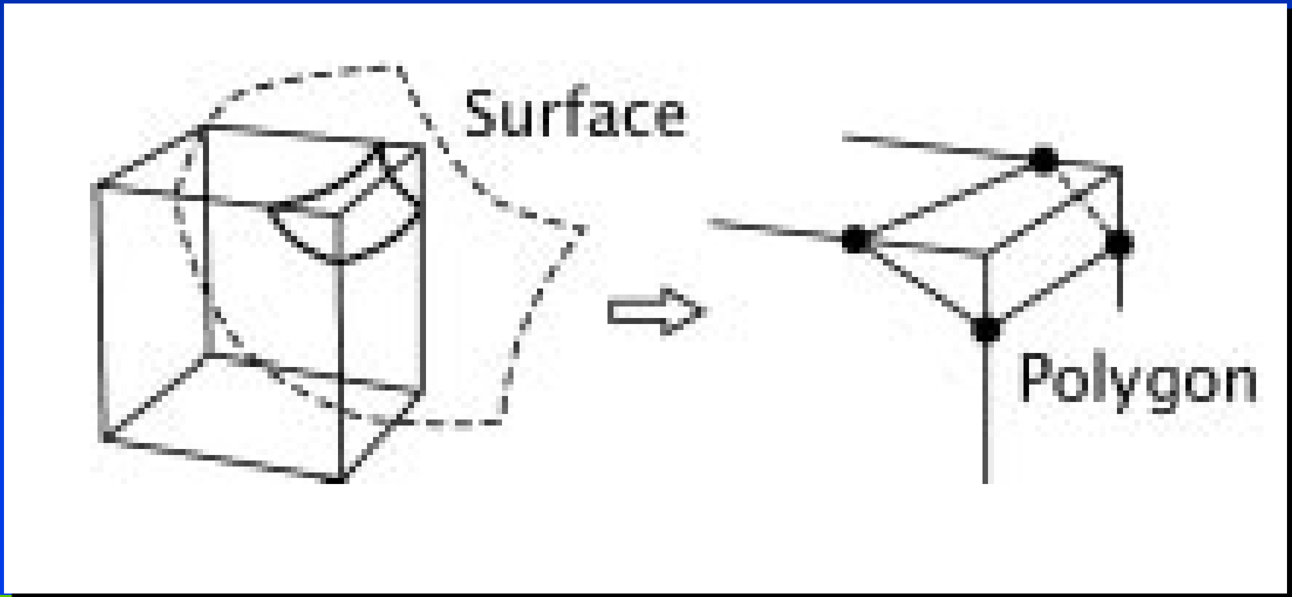
How do we know if a cell actually contains the surface?

- Straddling Cells
  - At least one vertex inside and outside surface
  - Non-straddling cells can still contain surface
- Guarantees
  - Interval analysis
  - Lipschitz condition



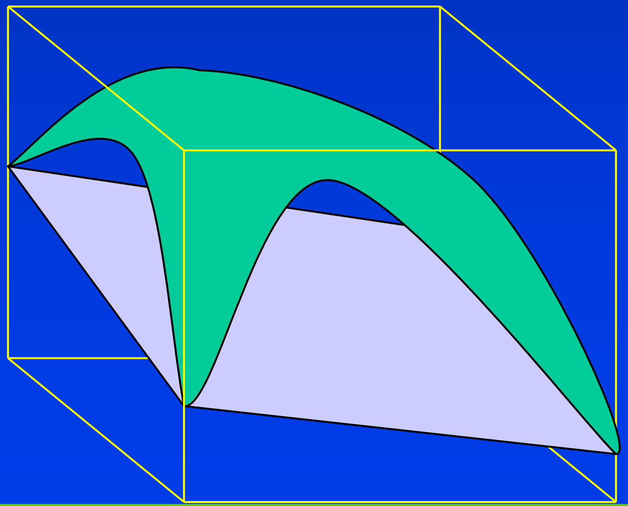
Polygonal Representation

- Partition space into convex cells
- Find cells that intersect the surface (*traverse cells*)
- Compute surface vertices



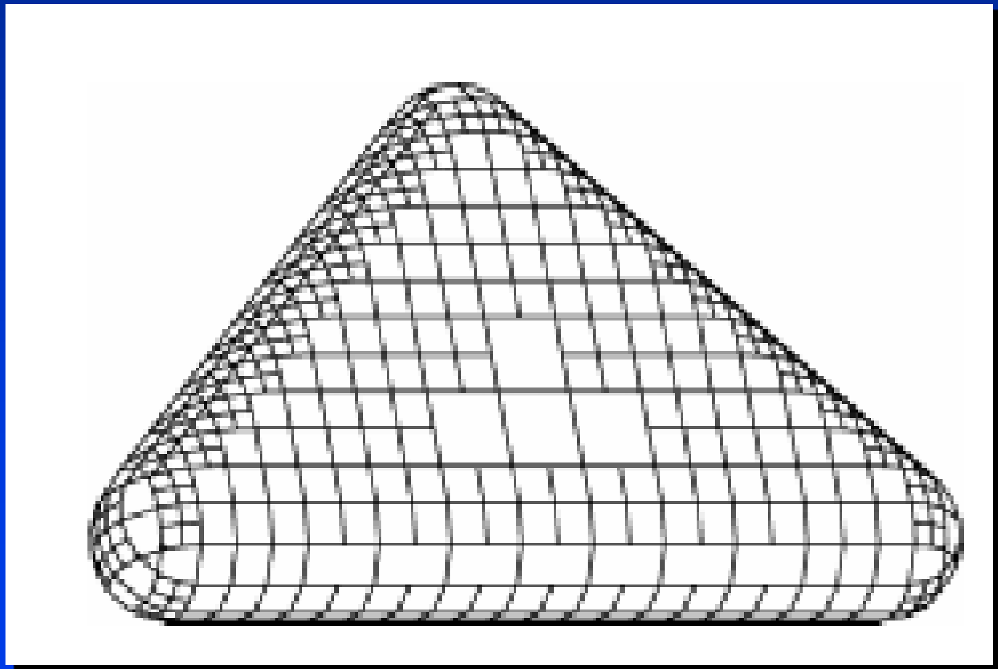
Cell Polygonization

- We will need to find those cells that actually contain parts of surface
- Need to approximate surface within cell
- Basic idea: use piecewise-linear approximation (polygon)



Spatial Partitioning

- Adaptive polygonization

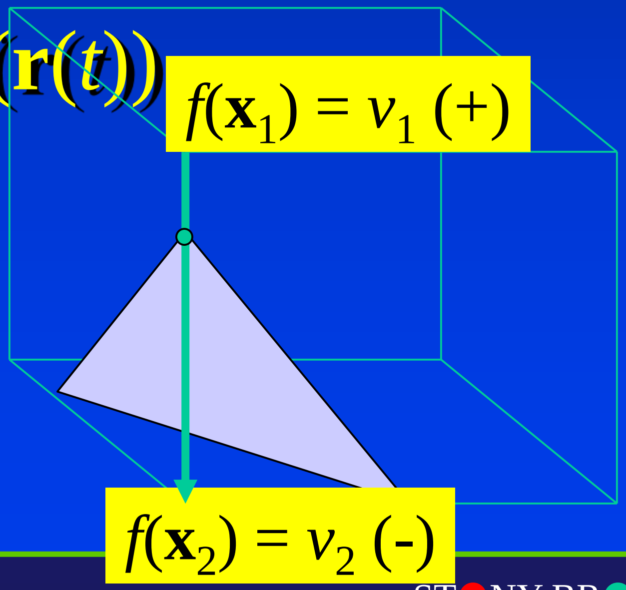


Surface Vertex Computations

- Determine where implicit surface intersects cell edges
- EITHER linear interpolate function values to approximate
- OR numerically find zero of $f(\mathbf{r}(t))$

$$\mathbf{r}(t) = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$$

$$0 \le t \le 1$$

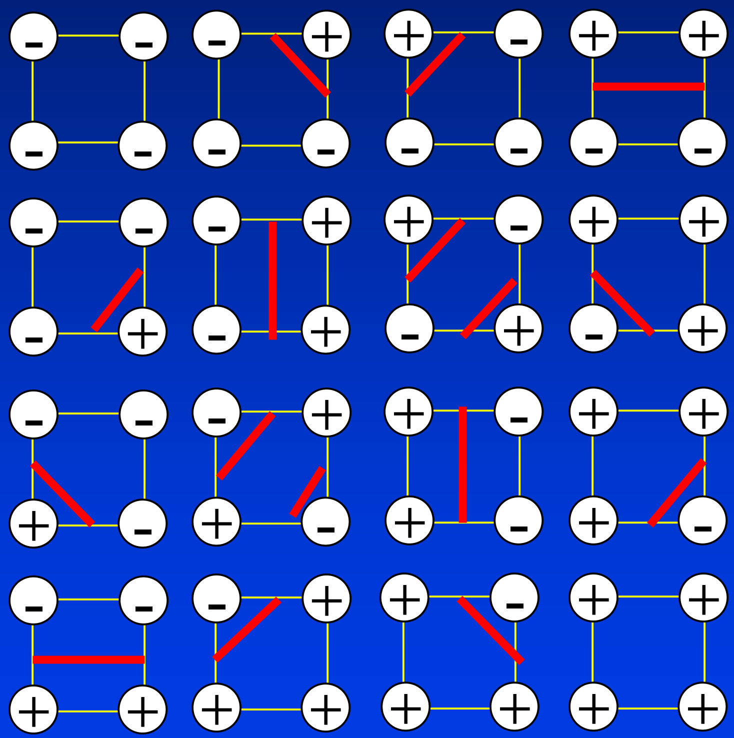


$$\mathbf{x} = \frac{v_1}{v_1 + v_2}\mathbf{x}_1 + \frac{v_2}{v_1 + v_2}\mathbf{x}_2$$

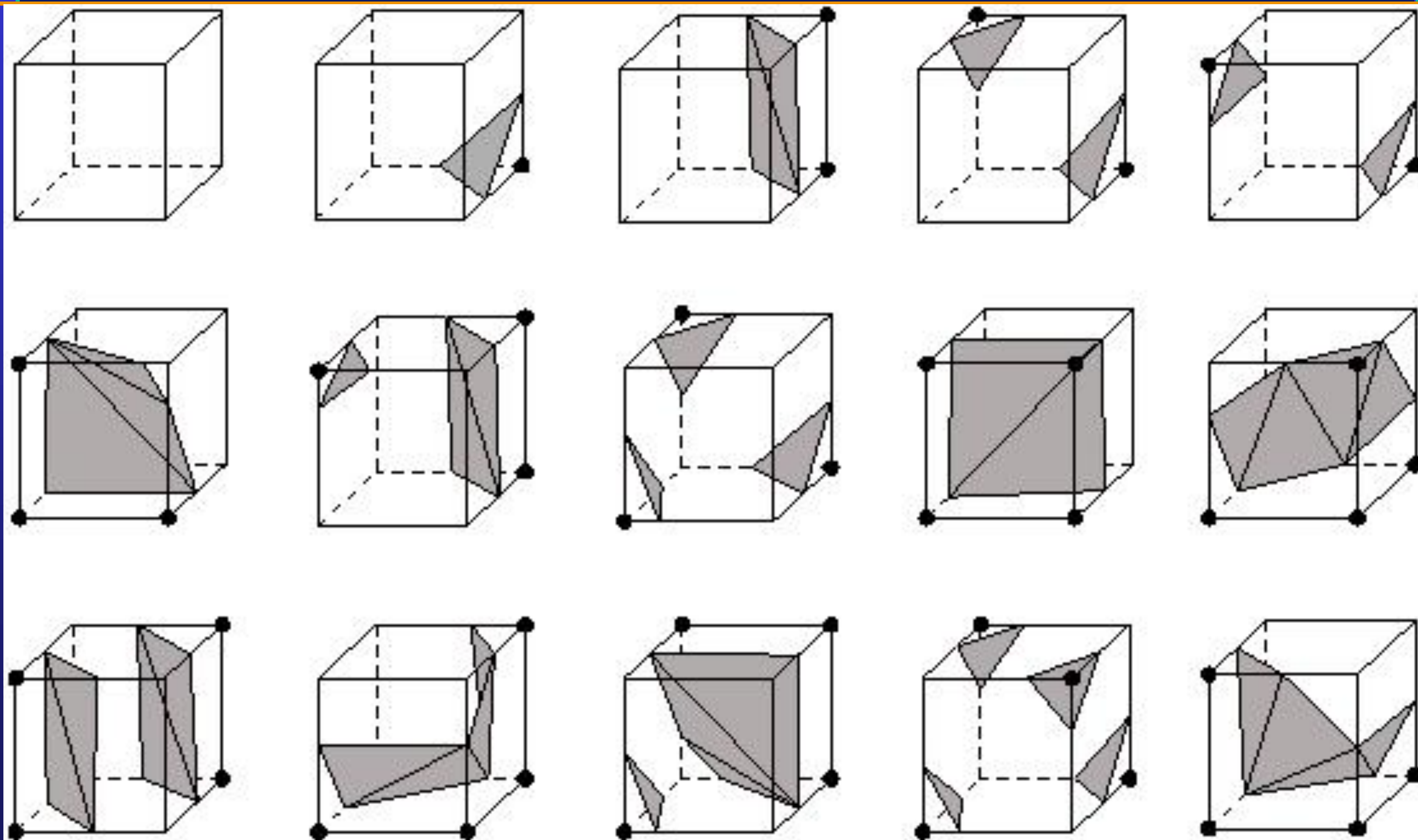
Department of Computer Science
Center for Visual Computing
STONY BROOK
STATE UNIVERSITY OF NEW YORK

Polygonal Shape

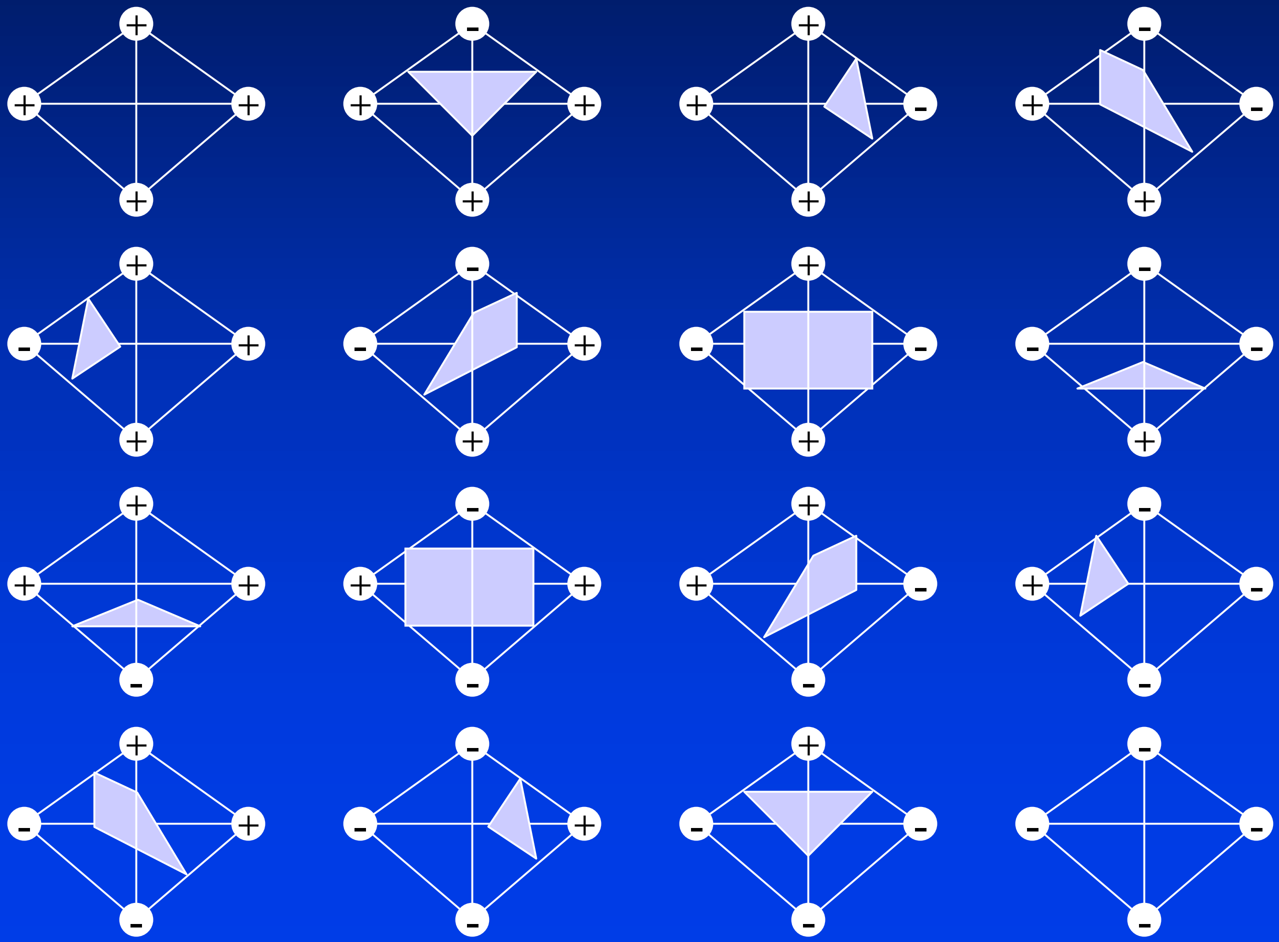
- Use table indexed by vertex signs and consider all possible combinations
- Let + be 1, - be 0
- Table size
  - Tetrahedral cells: 16 entries
  - Cubic cells: 256 entries
- E.g., 2-D - 16 square cells



Determining Intersections

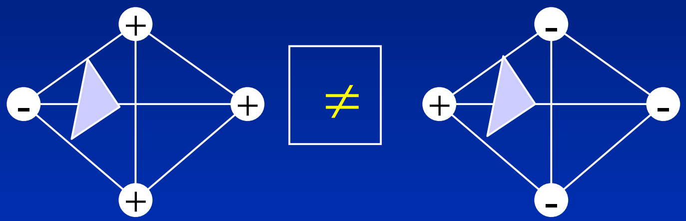


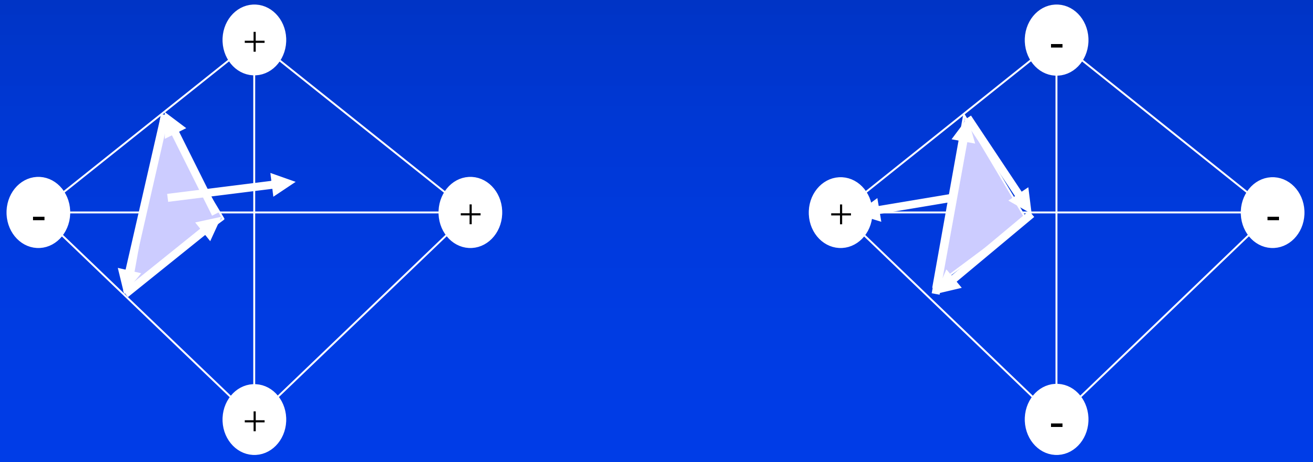
Tetrahedral Cell Polygons



Orientation

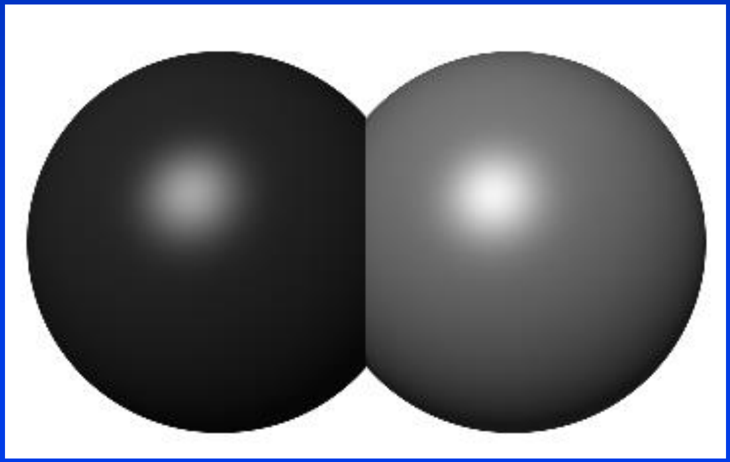
- Consistency allows polygons to be drawn with correct orientation
- Supports backface culling

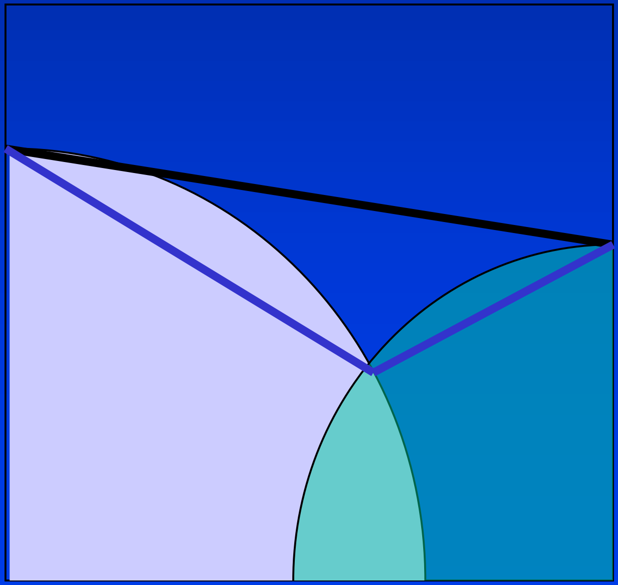




CSG Polygonization

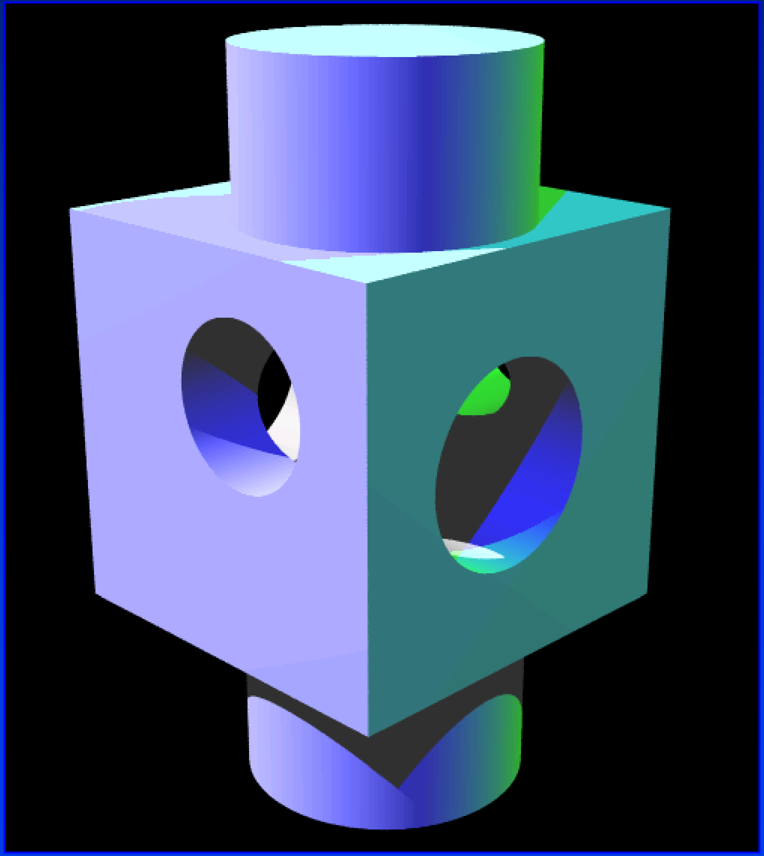
- Polygonization can smooth crease edges caused by CSG operations
- Polygonization needs to add polygon vertices along crease edges



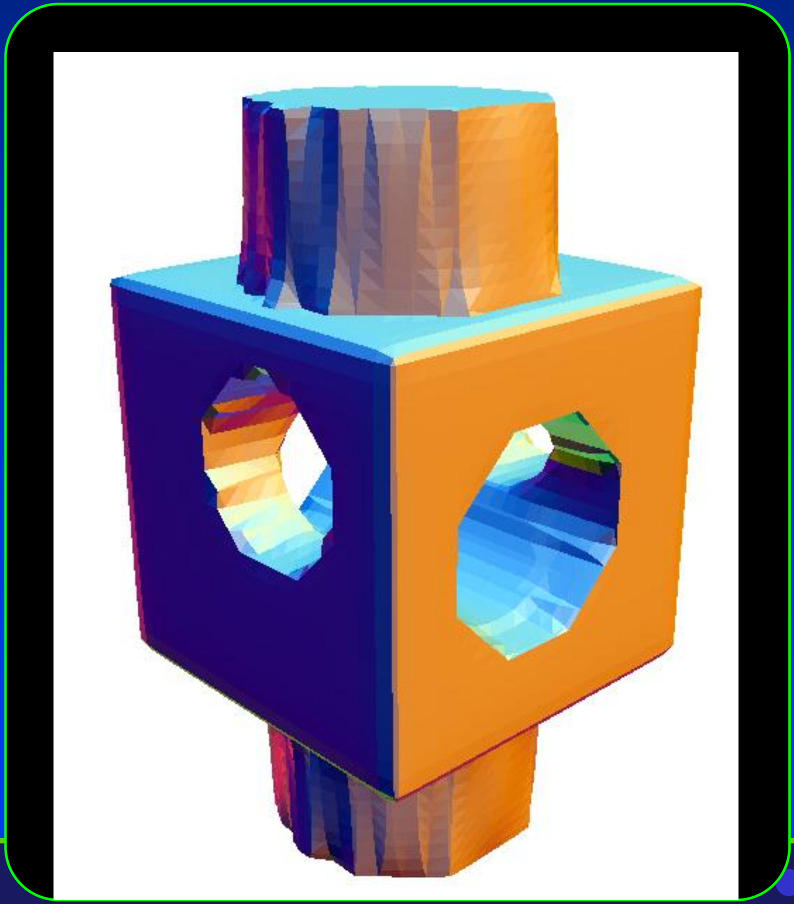


Visualization of Implicit Surfaces

Ray-tracing

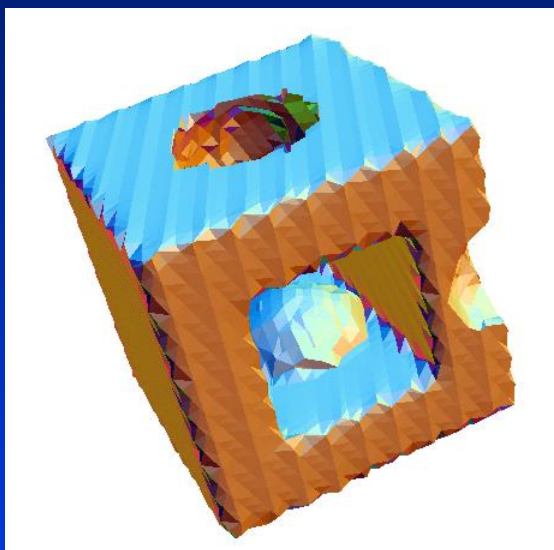


Polygonization
(e.g. Marching cubes method)

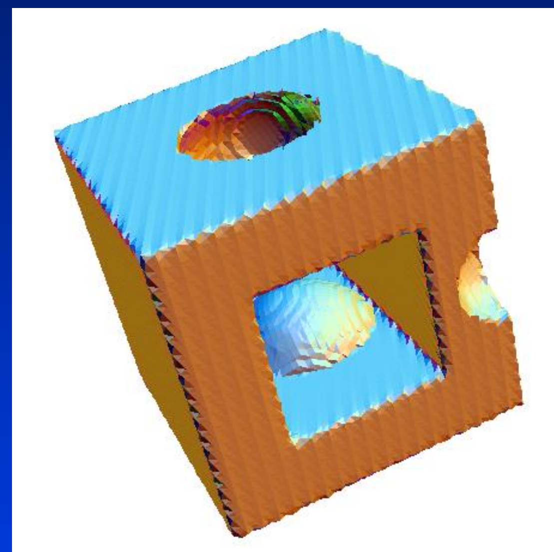


Problem of Polygonization

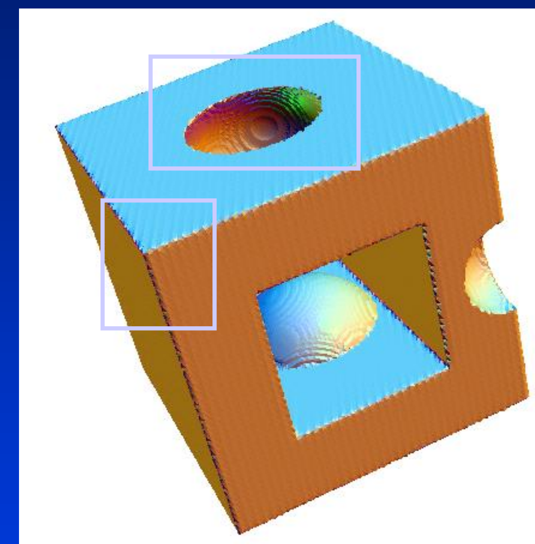
50^3 grid



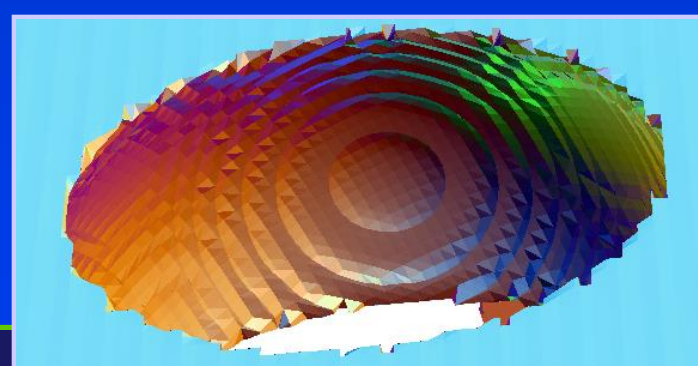
100^3 grid

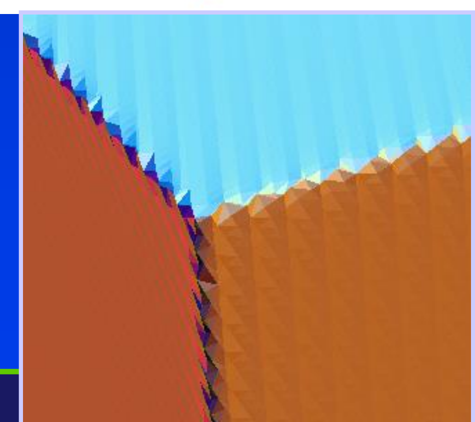


200^3 grid



- Sharp features are broken





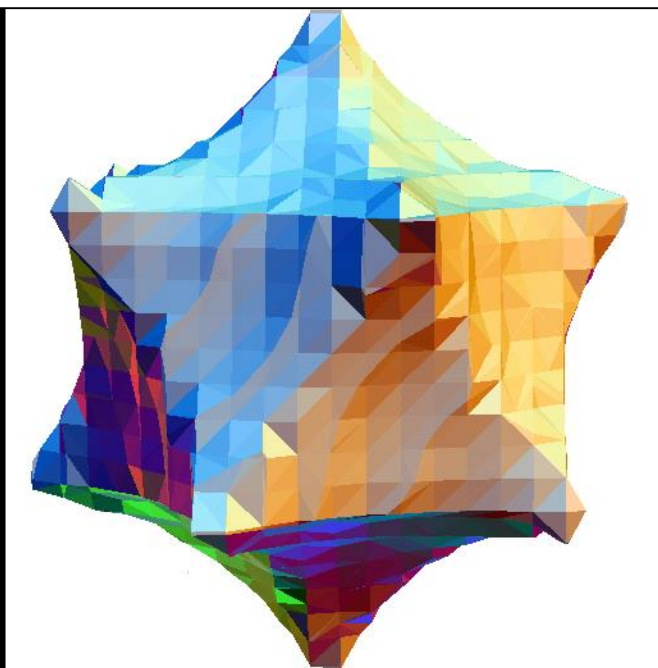
Reconstruction of Sharp Features

Input

Implicit function : $f(x, y, z)$

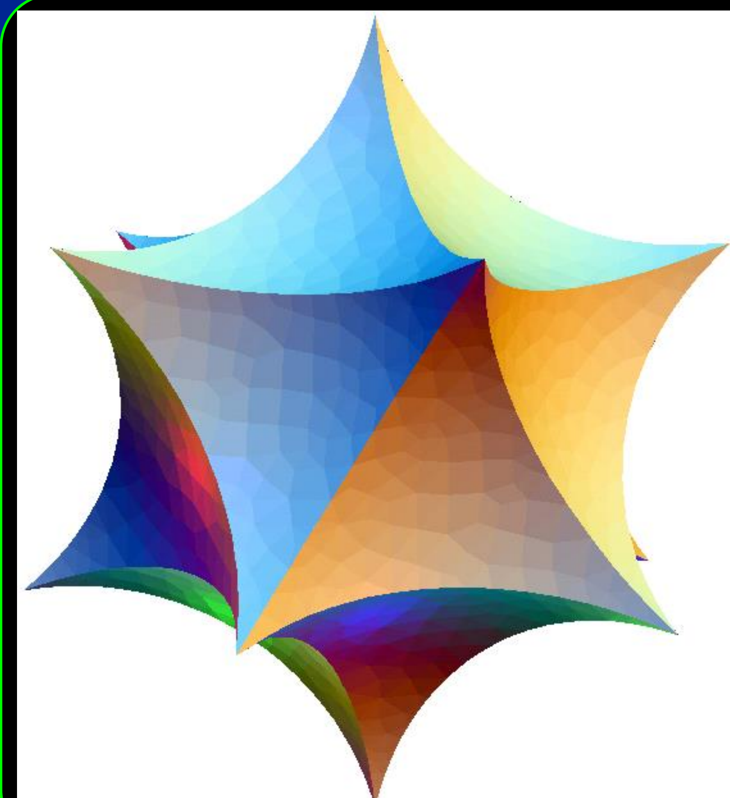
and

Rough Polygonization
(Correct topology)



Post-processing

Output

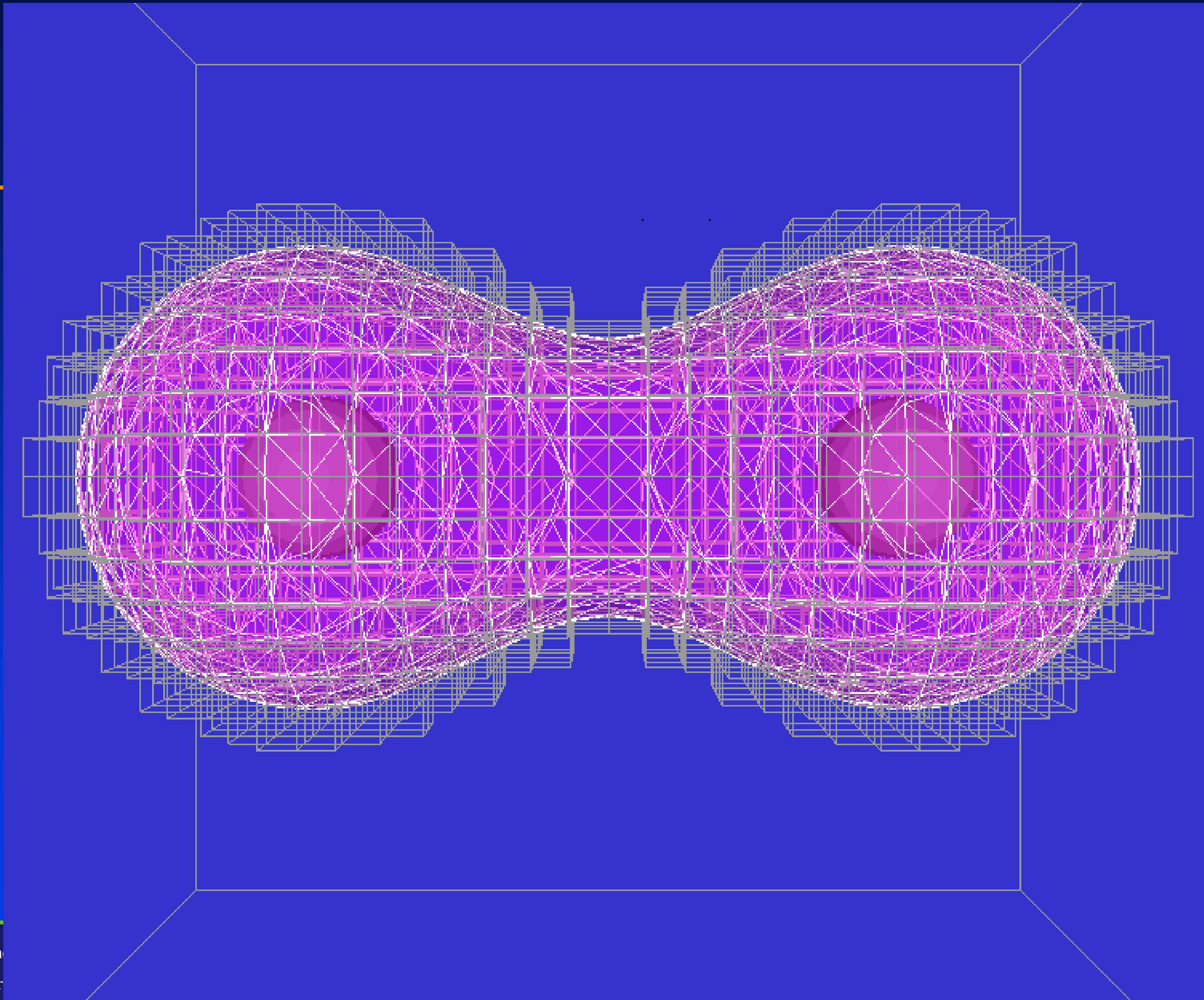


Blobs and Metaballs

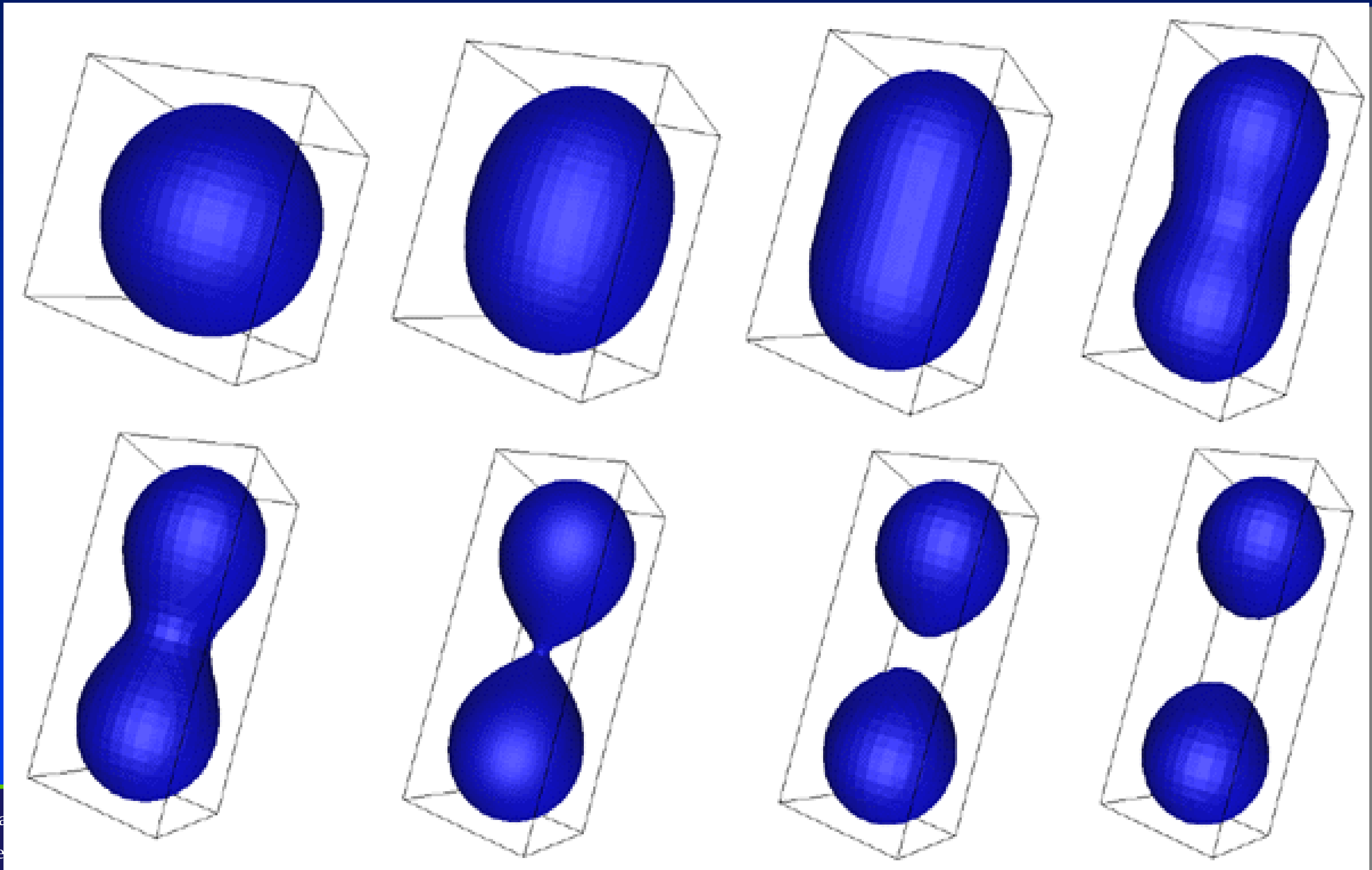
- Define the location of some points
- For each point, define a function on the distance to a given point, *(x,y,z)*
- Sum these functions up, and use them to define (surface) geometry via an implicit function
- Question: if I have two special points, in 2D, and my function is just the distance, what shape results?
- More generally, use Gaussian functions of distance, or other forms
  - Various results are called blobs or metaballs

What Is This?

- “Metaball, or ‘Blobby’, Modeling is a technique which uses implicit surfaces to produce models which seem more ‘organic’ or ‘blobby’ than conventional models built from flat planes and rigid angles”

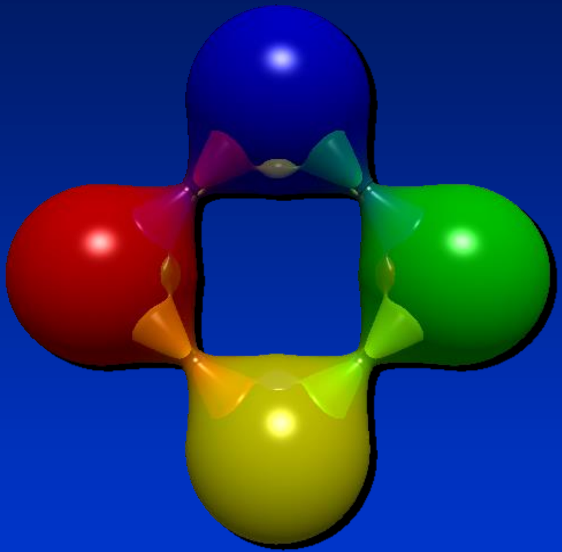


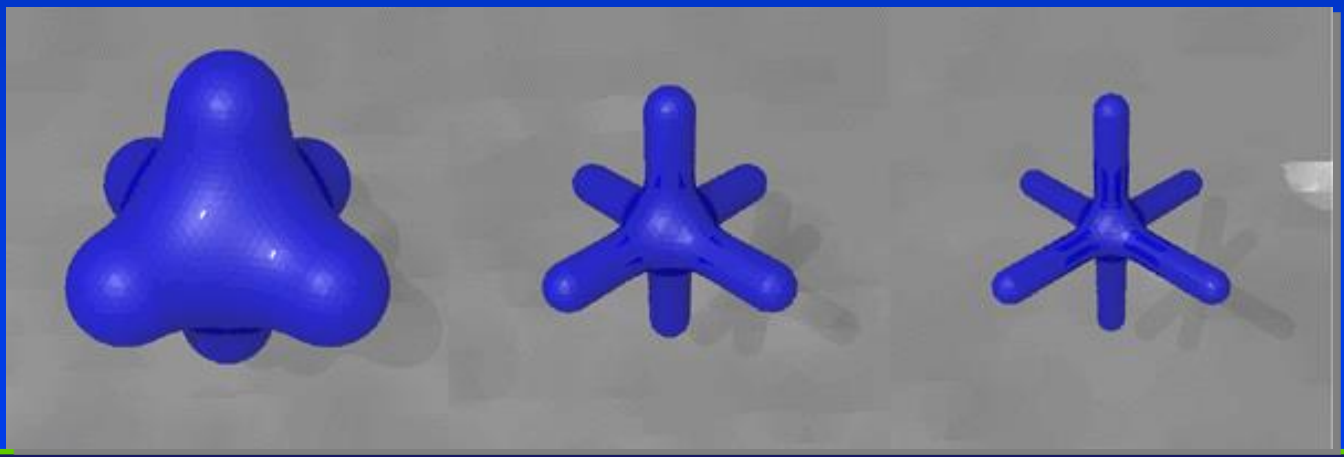
Distance Functions



Case Studies: Distance Functions

- D(**p**) = R
  - Sphere: distance to a point
  - Cylinder: distance to a line
  - More examples





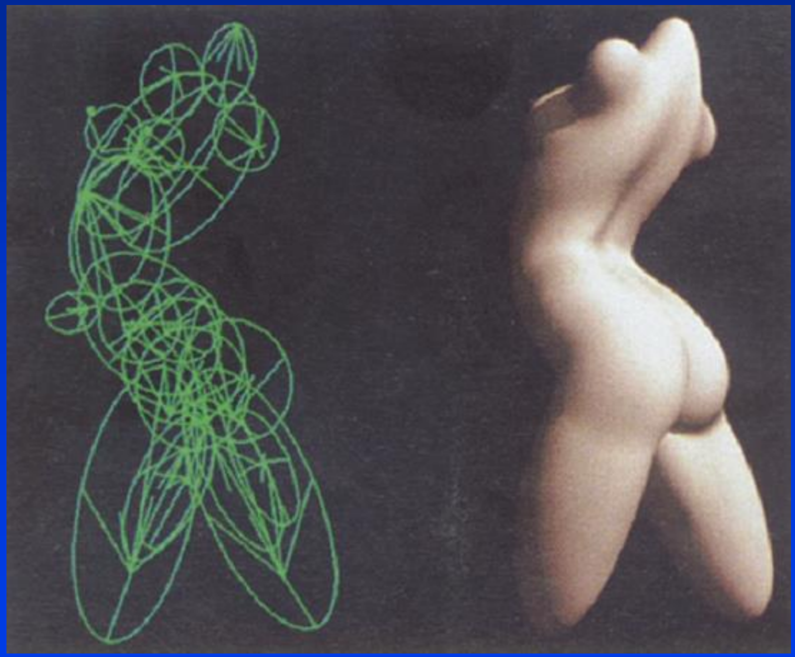
Blobby Models

- Blobby models [Blinn 82], also known as metaballs [Nishimura and Hirai 85] or soft objects [Wyvill and Wyvill 86, 88]
- A blobby model — a center surrounded by a density field, where the density attributed to the center decreases with distance from the center.
- By simply summing the influences of each blobby model on a given location, we can obtain very smooth blends of the spherical density fields.

$$G(x, y, z) = \sum_i g_i(x, y, z) - threshold = 0$$

Design Using Blobs

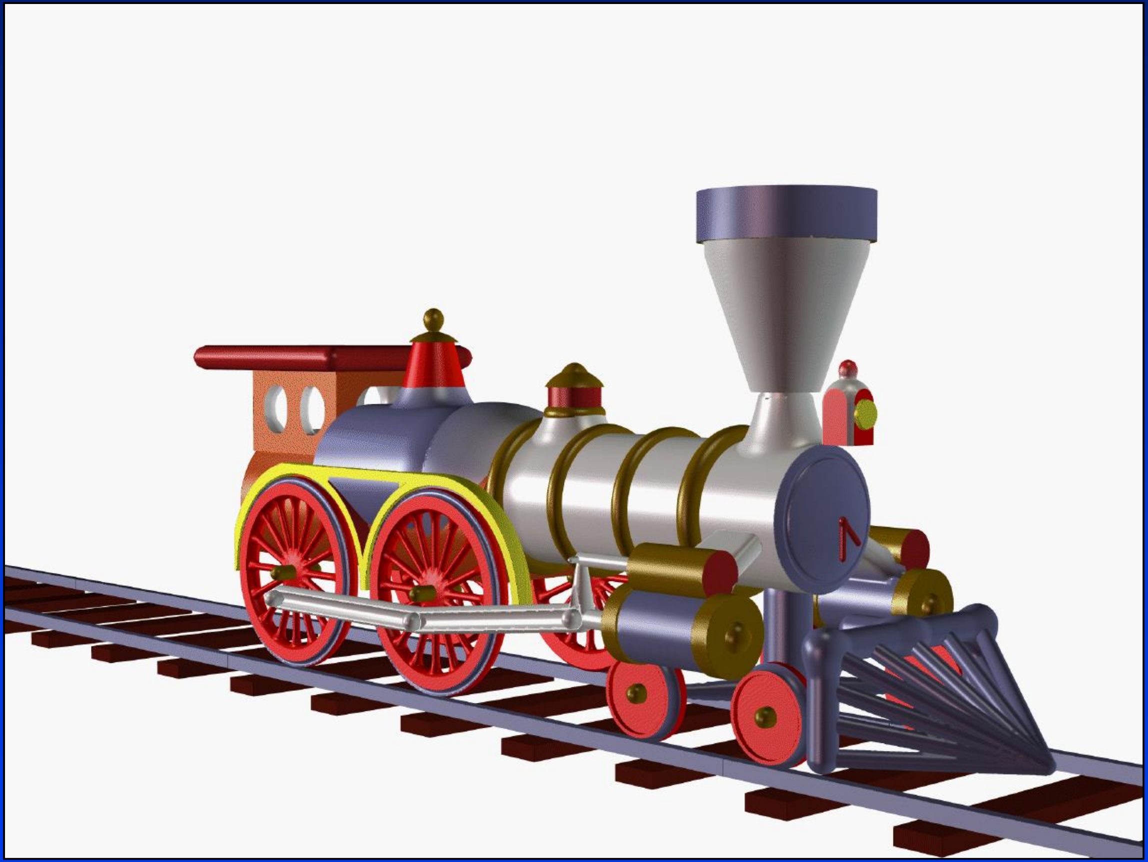
- None of these parameters allow the designer to specify exactly where the surface is actually located.
- A designer only has indirect control over the shape of a blobby implicit surface.
- Blobby models facilitate the design of smooth, complex, organic-appearing shapes.



Example with Blobs

Rendered with POVray. Not everything is a blob, but the characters are.

Examples



Blobby Modeling: Its Utility

- Organic forms and nonlinear shapes
- Scientific modeling (electron orbitals, some medical imaging)
- Muscles and joints with skin
- Rapid prototyping
- CAD/CAM solid geometry

Examples

Mathematics for Blobby Model

- Implicit equation:

$$f(x,y,z) = \sum_{i=1}^{n_{blobs}} w_i g_i(x,y,z) = d$$

- The w_i are weights – just numbers
- The g_i are (scalar) functions, one common choice is:

$$g_i(\mathbf{x}) = e^{\frac{-(\mathbf{x}-c_i)^2}{\sigma_i}}$$

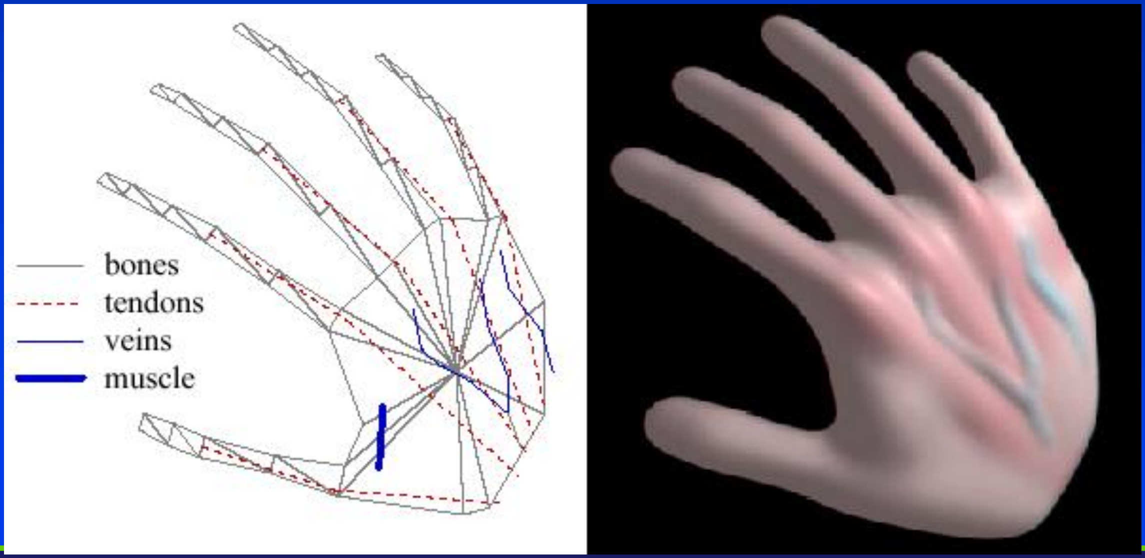
  – c_i and σ_i are parameters

Skeletal Design

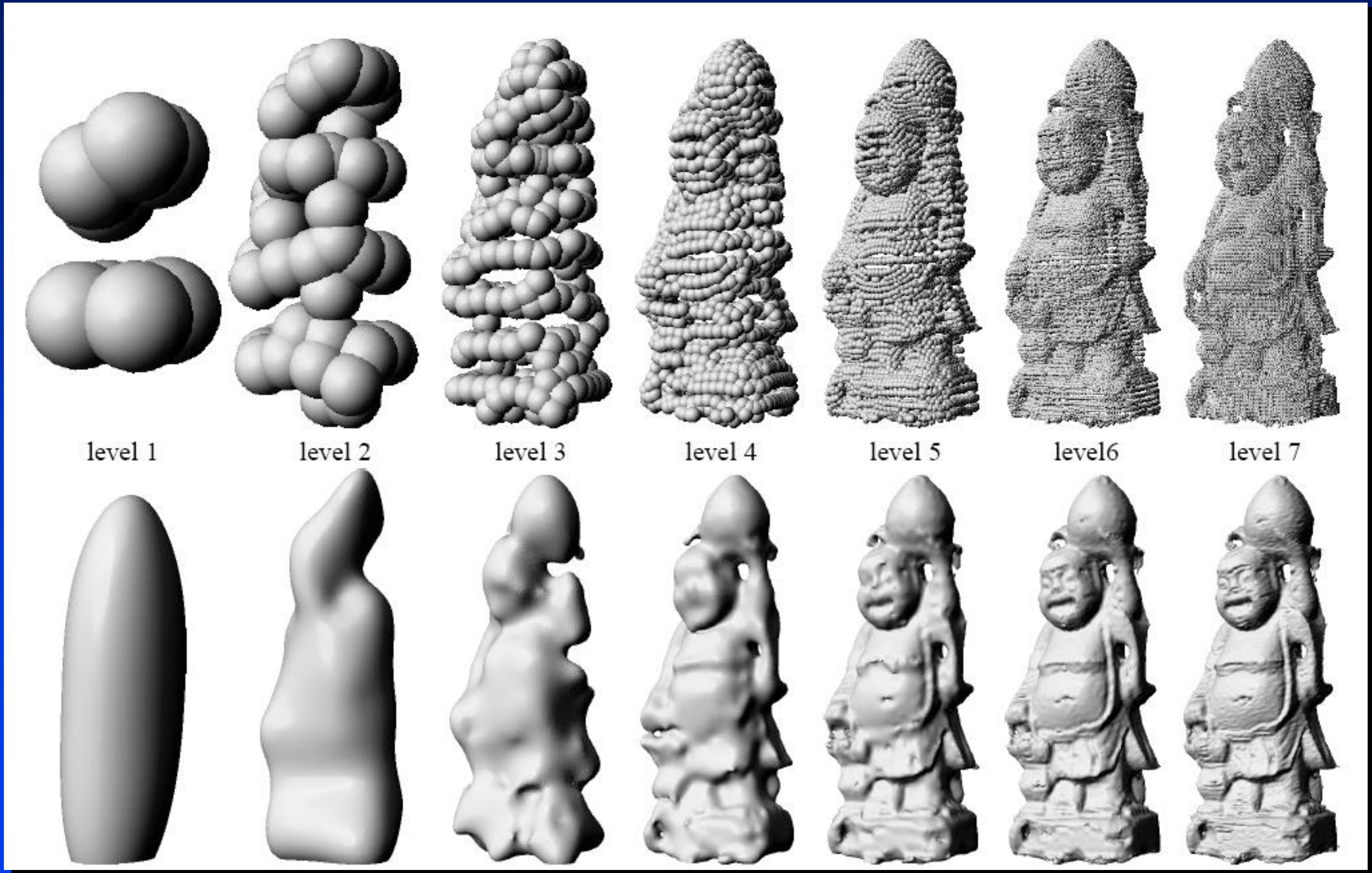
- Use skeleton technique to design implicit surfaces and solids toward interactive speed.
- Each skeletal element is associated with a locally defined implicit function.
- These local functions are blended using a polynomial weighting function.
  - [Bloomenthal and Wyvill 90, 95, 97] defined skeletons consisting of *points*, *splines*, *polygons*.
  - 3D skeletons [Witkin and Heckbert 94] [Chen 01]

Skeletal Design

- Global and local control in three separate ways:
  - Defining or manipulating the skeleton;
  - Defining or adjusting those implicit functions defined for each skeletal element;
  - Defining a blending function to weight the individual implicit functions.



Multi-level Representation



Rendering Implicit Surfaces

- Raytracing or its variants can render them directly
  - The key is to find intersections with Newton’s method
- For polygonal renderer, must convert to polygons
- Advantages:
  - Good for organic looking shapes e.g., human body
  - Reasonable interfaces for design
- Disadvantages:
  - Difficult to render and control when animating
  - Being replaced with subdivision surfaces, it appears

Implicit Surfaces vs Polygons

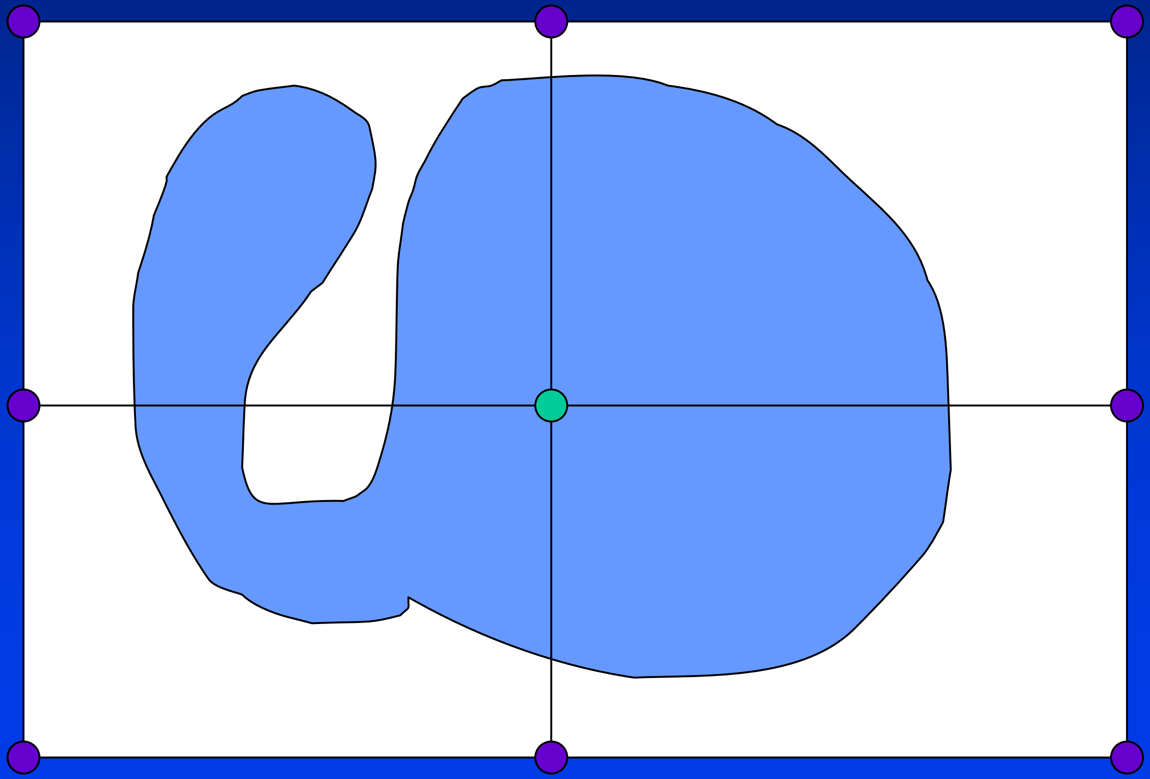
- Advantages
  - Smoother and more precise
  - More compact
  - Easier to interpolate and deform
- Disadvantages
  - More difficult to display in real time

Implicits vs Parameter-Based Representations

- Advantages
  - Implicits are easier to blend and morph
  - Interior/Exterior description
  - Ray-trace
- Disadvantages
  - Rendering
  - Control

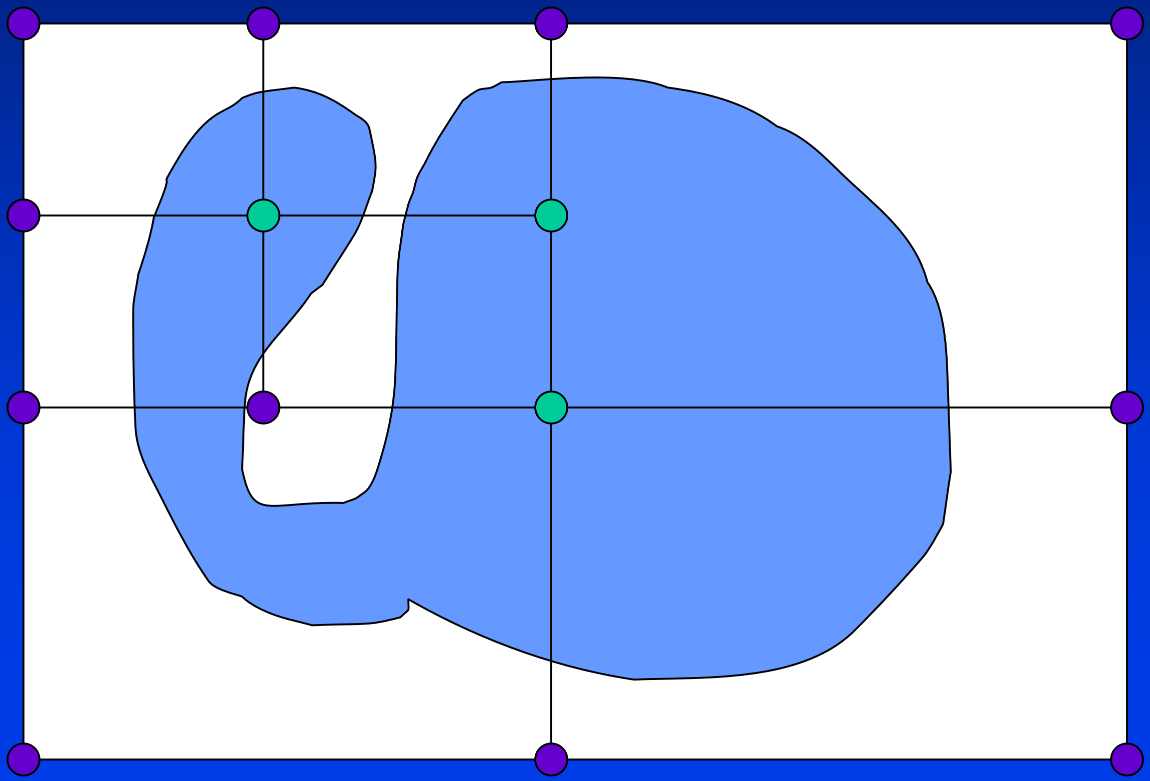
Display Implicit Surfaces

- Recursive subdivision:



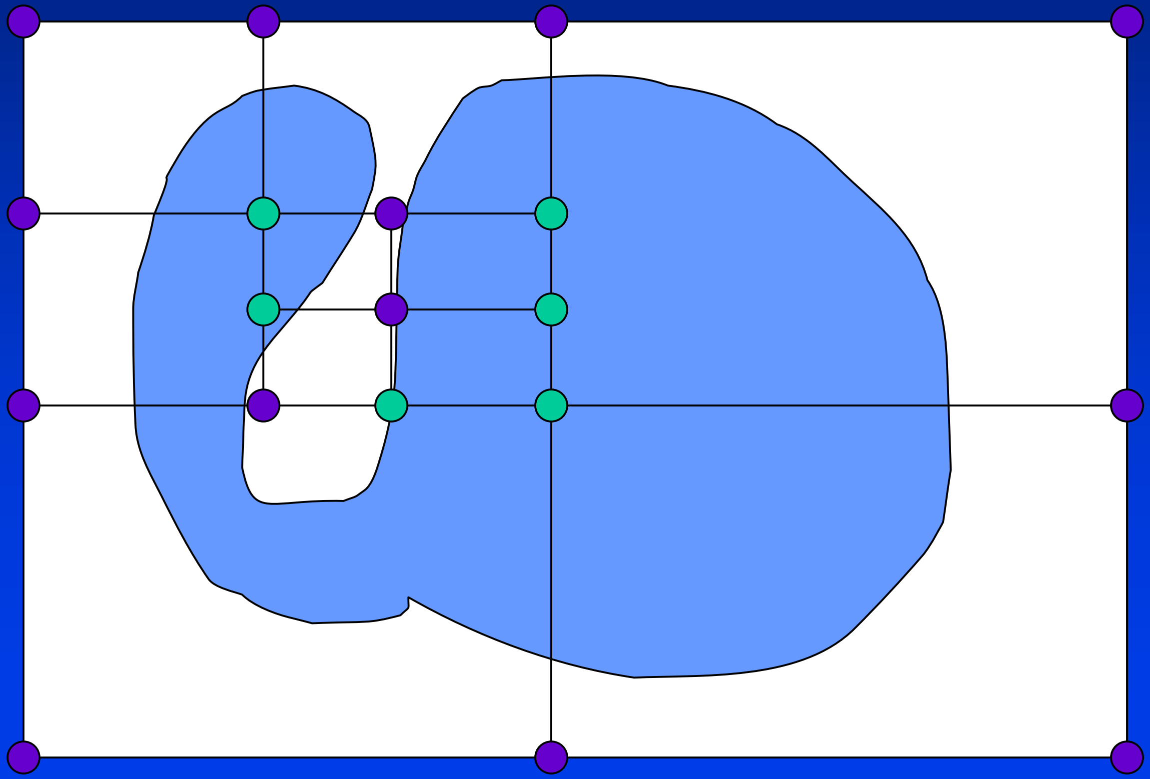
Display Implicit Surfaces

- Recursive subdivision:



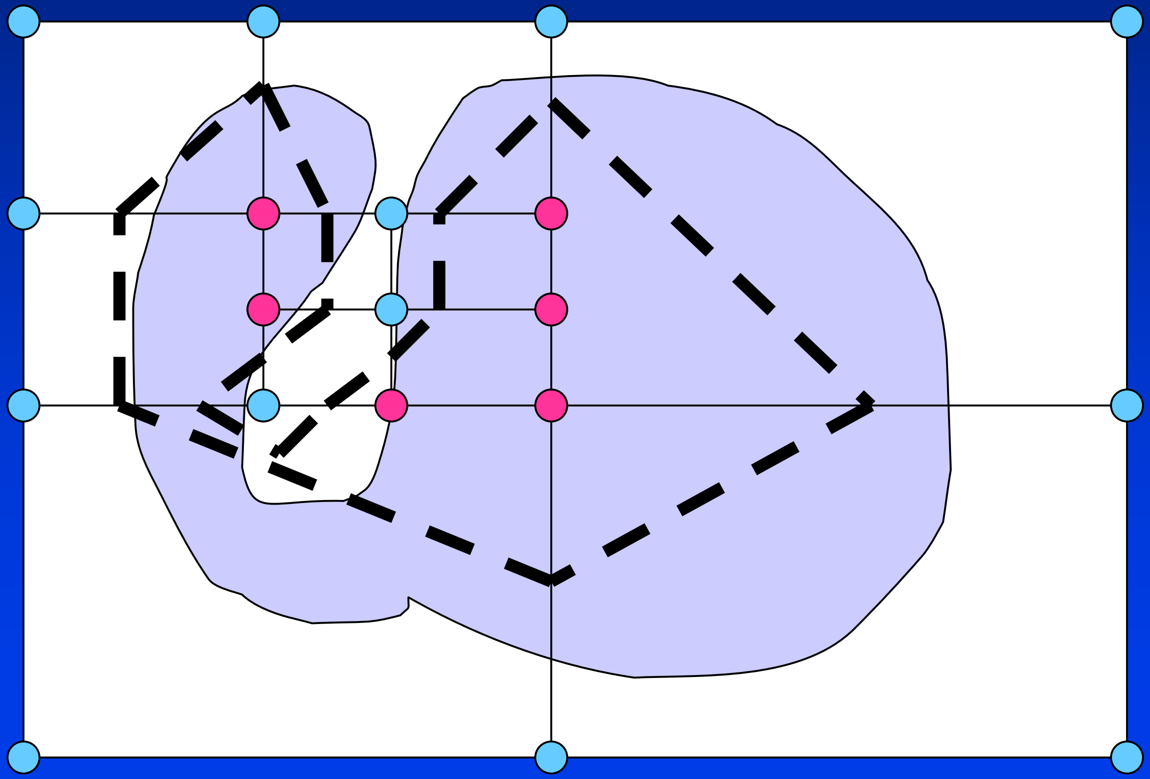
Display Implicit Surfaces

- Recursive subdivision:



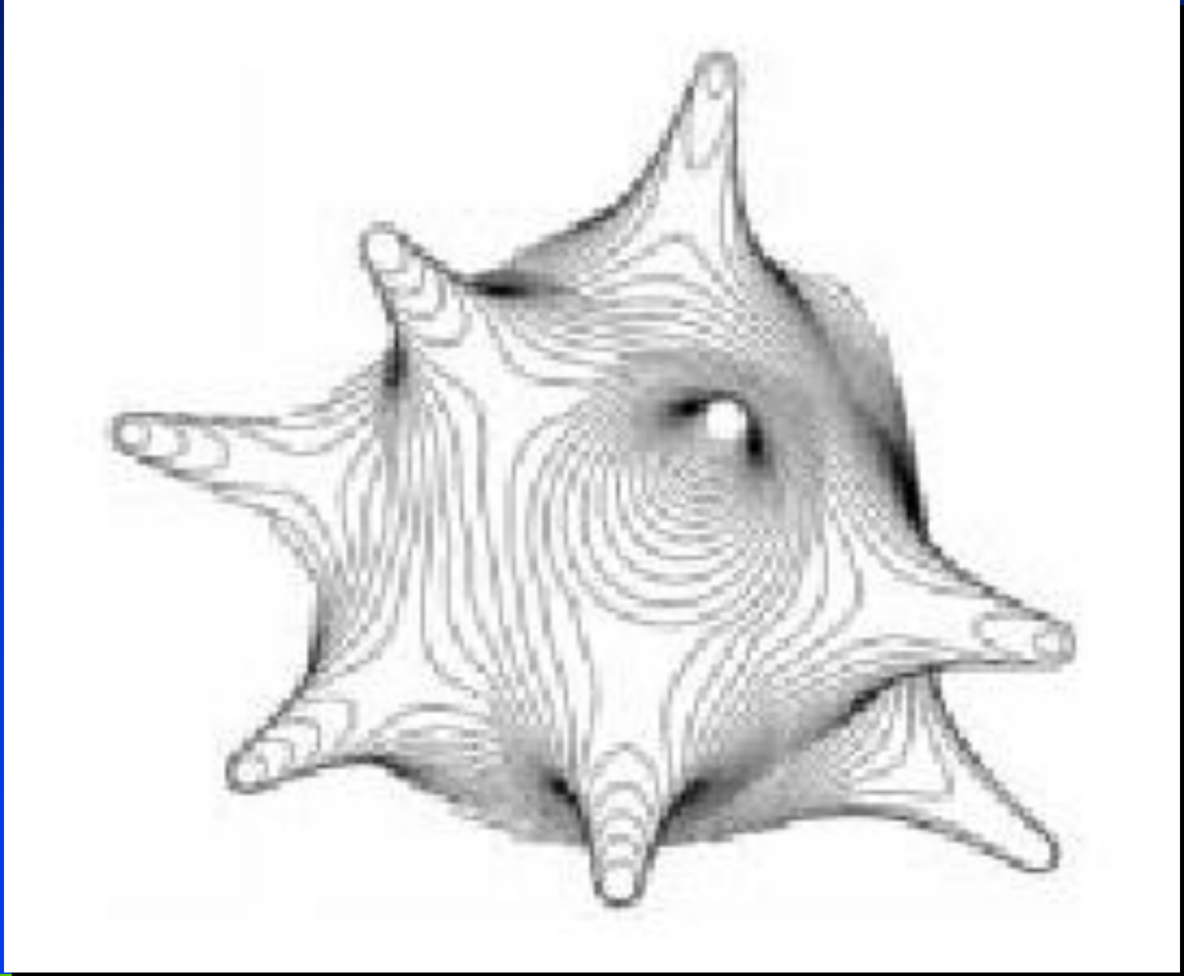
Display Implicit Surfaces

- Find the edges, separating hot from cold:



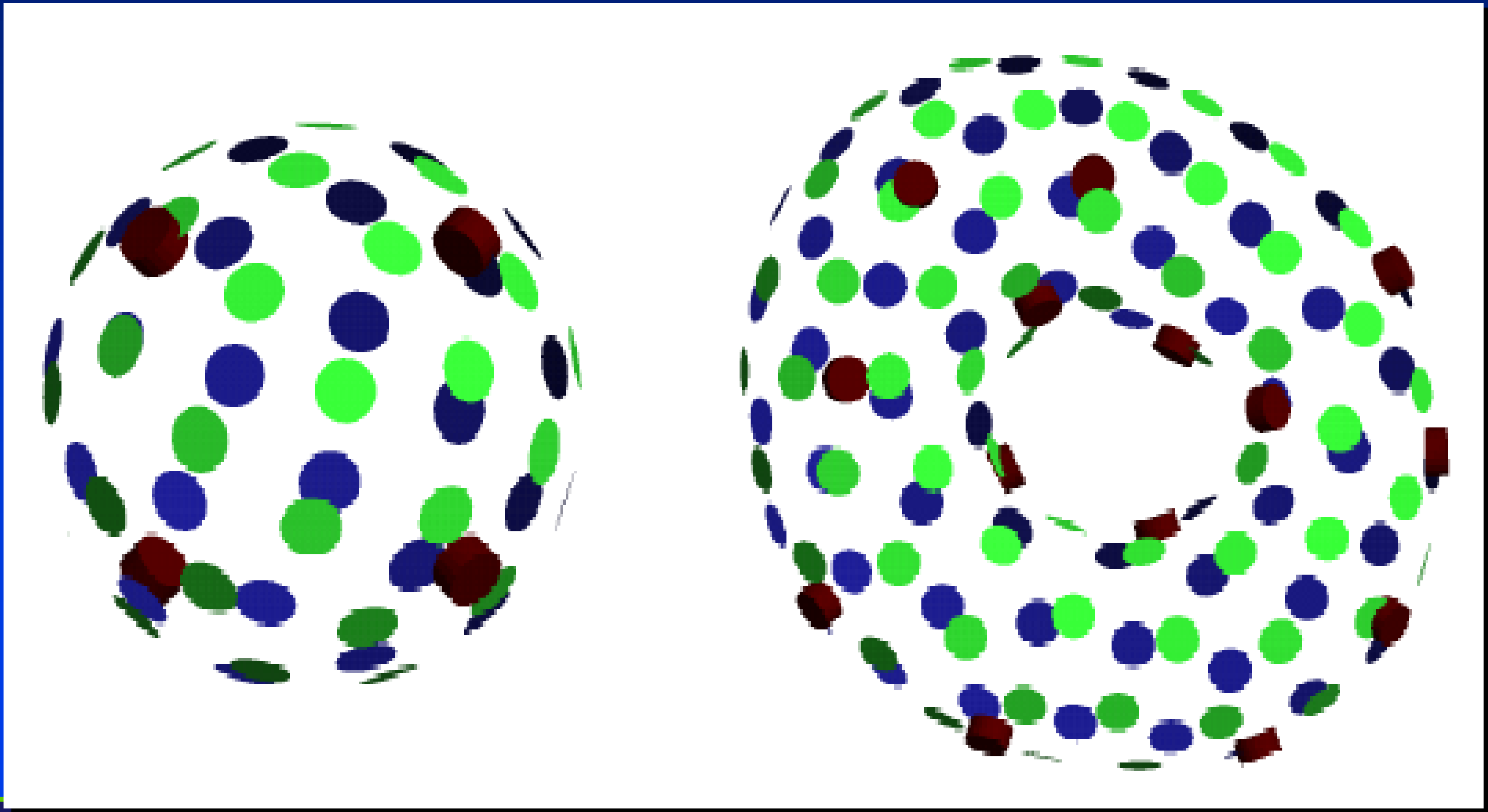
Visualization

- Contours



Visualization

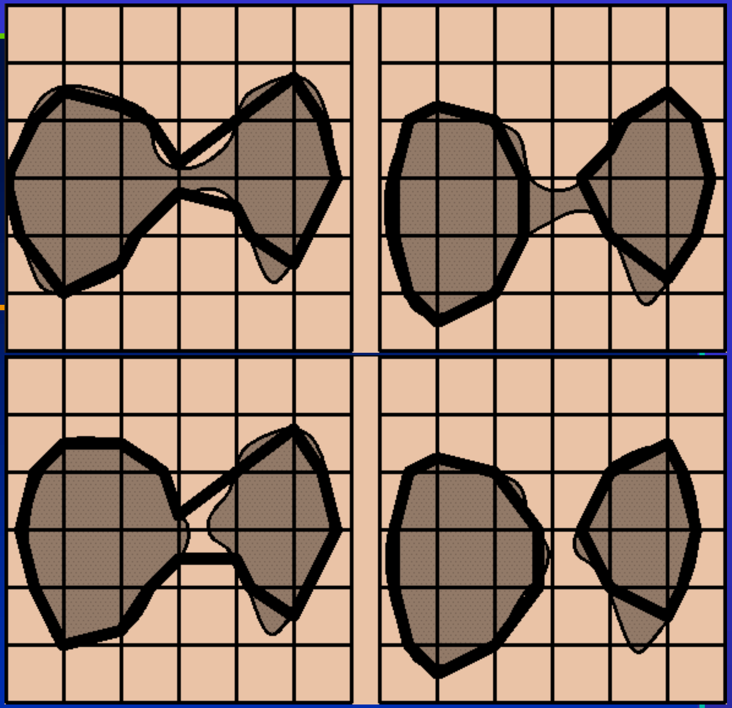
- Particle display



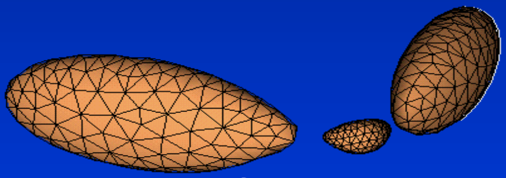
Particle Systems

- Witkin Heckbert S94
- Constrain particle system to implicit surface (Implicit surface $f = 0$ becomes constraint surface $C = 0$)
- Particles exert repulsion forces onto each other to spread out across surface
- Particles subdivide to fill open gaps
- Particles commit suicide if overcrowded
- Display particle as oriented disk
- Constrain implicit surface to particles!

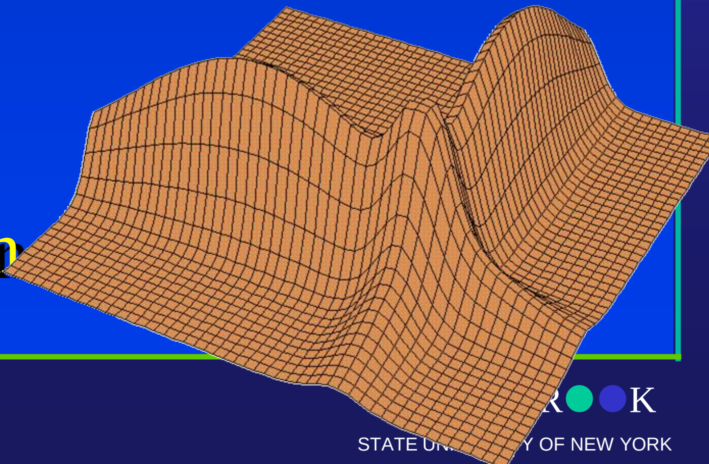
Meshing Particles



- Stander Hart S97
- Use particles as vertices
- Connect vertices into mesh
- Problems:
  - Which vertices should be connected?
  - How should vertices be reconnected when surface moves?

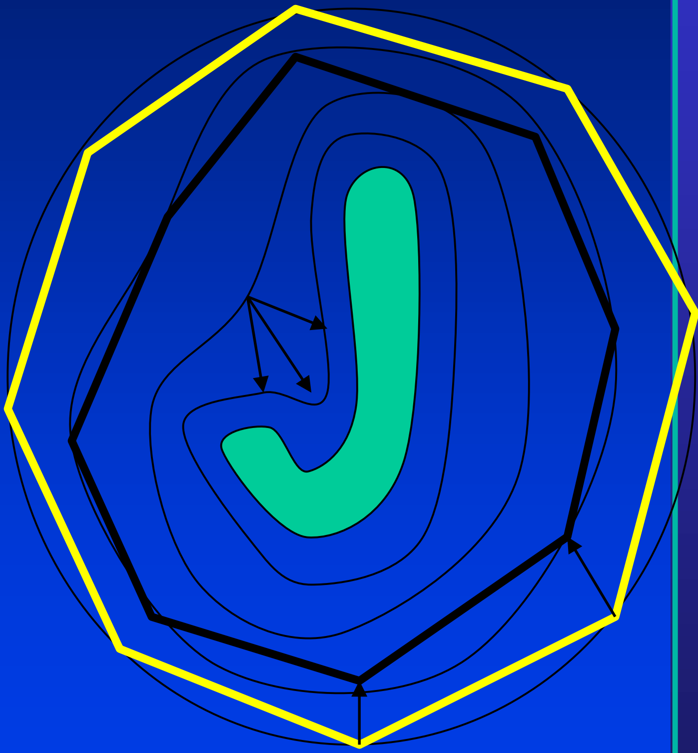


- Solution: Morse theory
- Track/find critical points of function in topology of implicit surface



Shrink-wrapping Mechanism

- Look at family of surfaces $f^{-1}(s)$ for $s > 0$
- For s large, $f^{-1}(s)$ spherical
- Polygonize sphere
- Reduce s to zero

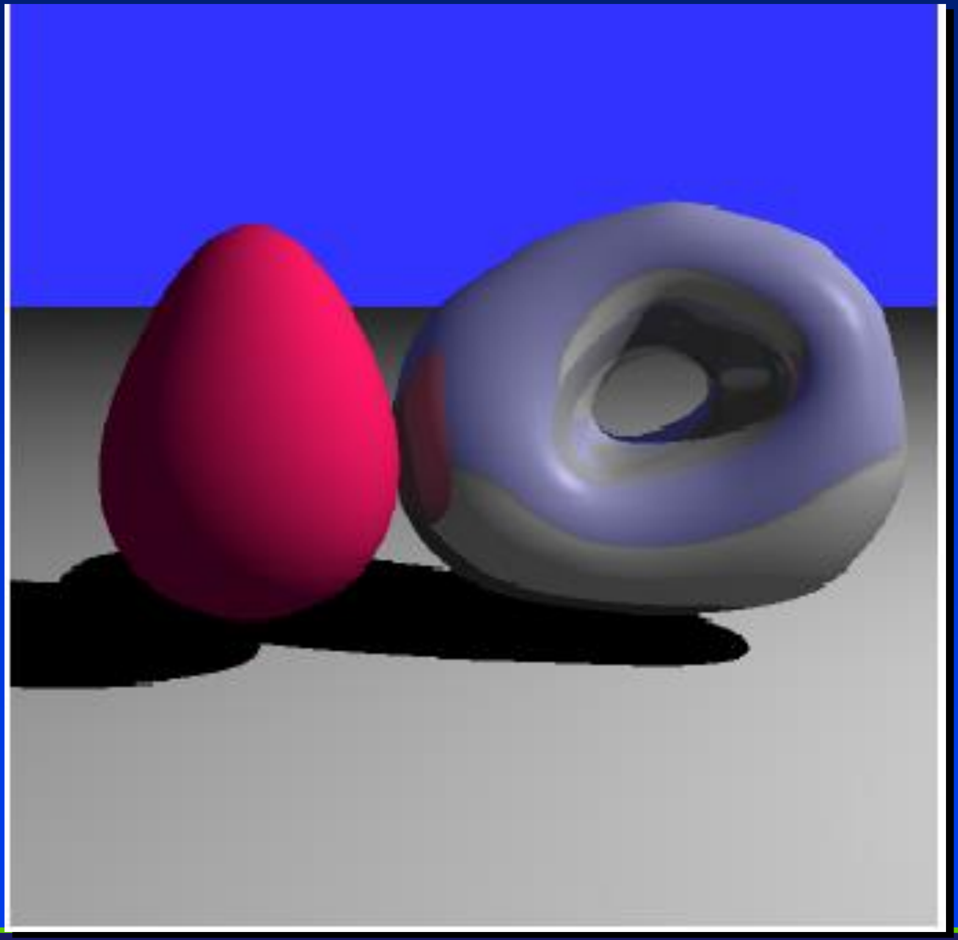


  – Allow vertices to track surface
  – Subdivide polygons as necessary when curvature increases

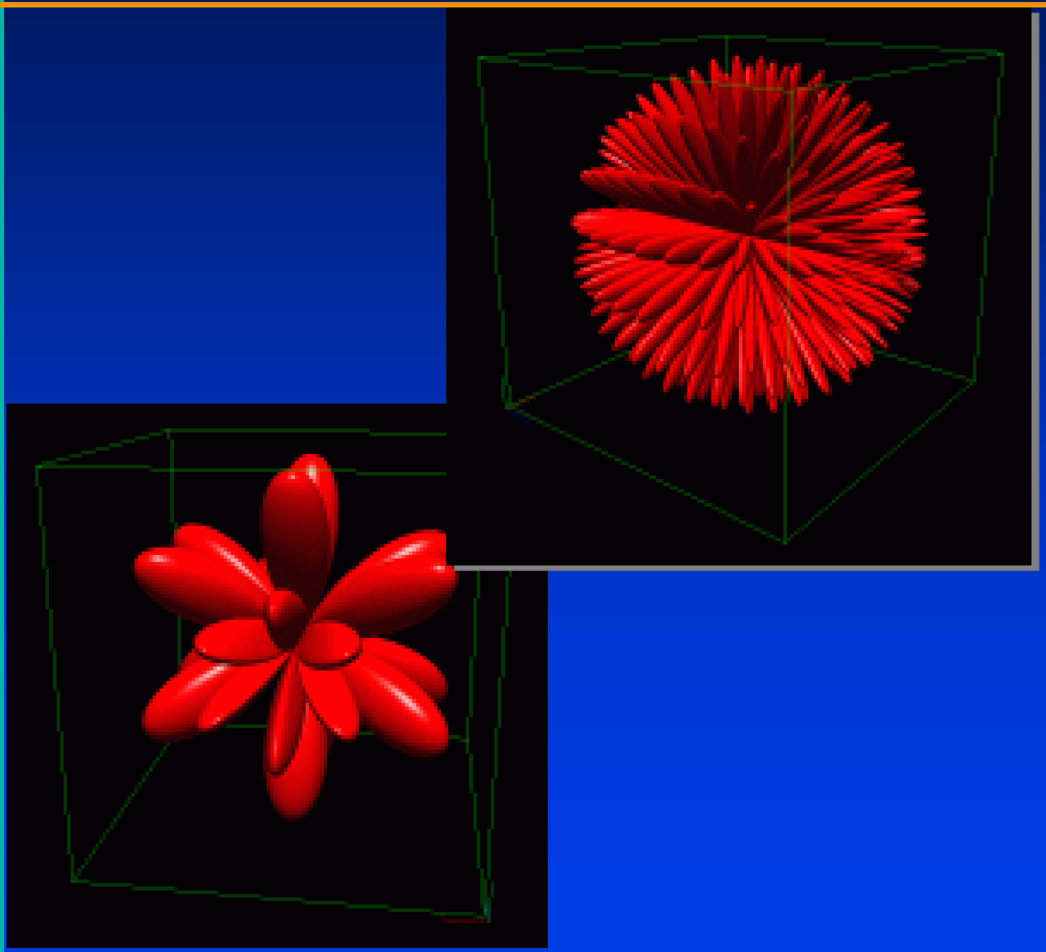
STONY BROOK
STATE UNIVERSITY OF NEW YORK

Visualization

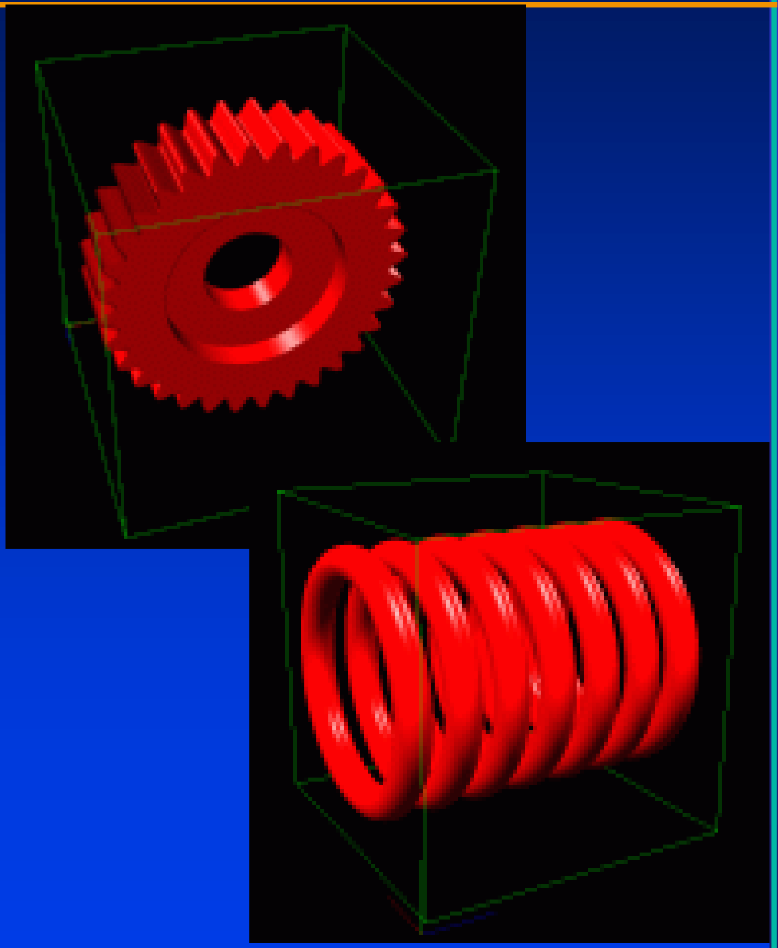
- Ray tracing



Other Coordinate Systems



Spherical Coordinates



Cylindrical Coordinates

Summary

- Surface defined implicitly by $f(p) = 0$; $p=[x,y,z]$
- Easy to test if point is on surface, inside, or outside
- Easy to handle blending, interpolation, and deformation
- Difficult to render

Deformation

- $\mathbf{p}' = D(\mathbf{p})$
- D maps each point in 3-space to some new location
- Twist, bend, taper, and offset