Computer Animation and Games I

CM50244

Today's Lectures

- **Implicit Surface Modeling**
- Subdivision Surface Modeling

Implicit Surface Modeling

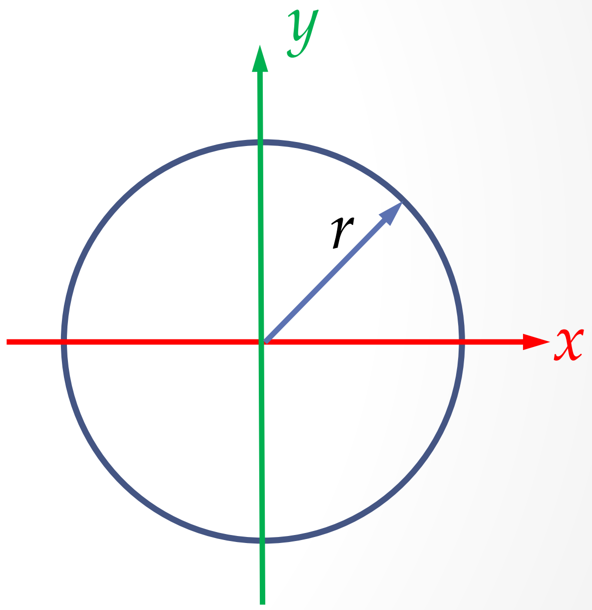
Some slides from Kwang In Kim

Overview

- **2D explicit representation ⇔ implicit representation**
- 3D explicit representation ⇔ implicit representation
- Implicit surface applications
- Implicit surfaces to meshes
  - Sampling implicit function
  - Marching cubes

Recap: 2D Curve Representation

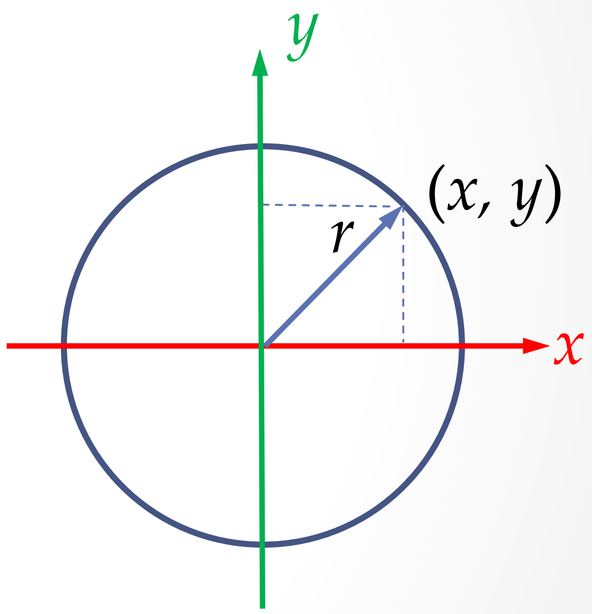
- **Implicit** representation
- **Explicit** representation



2D Curve Representation

- **Implicit** representation $F(x, y) = 0$
  - For any curve point, x and y coordinates should satisfy a single equation.

$$F(x, y) = x^2 + y^2 - r^2 = 0.$$



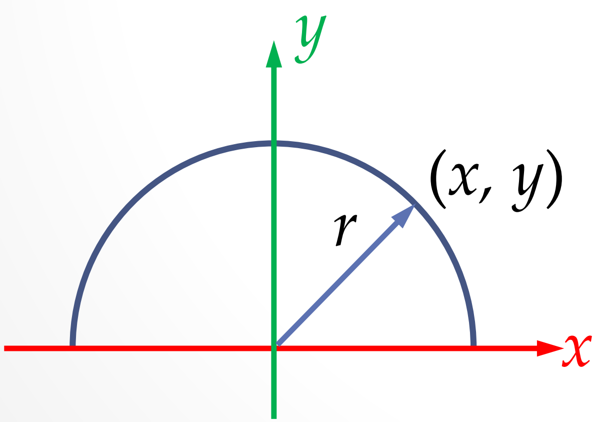
easily check if a point is on/inside/outside curve
by evaluating the function value

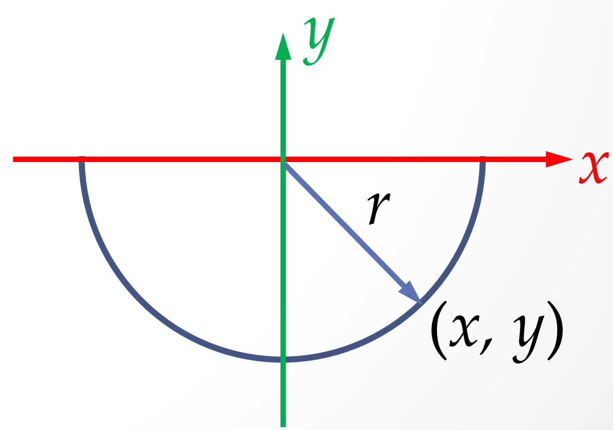
2D Curve Representation

- **Explicit** representation $y = f(x)$
  - y coordinate is explicitly represented as a function of x coordinate.

$$F(x, y) = x^2 + y^2 - r^2 = 0$$

$$f(y) = \sqrt{r^2 - x^2} \qquad\qquad f(y) = -\sqrt{r^2 - x^2}$$





simple since y coord. is explicitly represented by x coord.
but not convenient for multi-valued curve function

From Explicit Rep. to Implicit Rep.

- Explicit representation: $y = f(x)$
  Implicit representation: $F(x, y) = 0$

$$\Rightarrow F(x, y) = f(x) - y = 0.$$

- Straightforward to construct *F* given *f*.

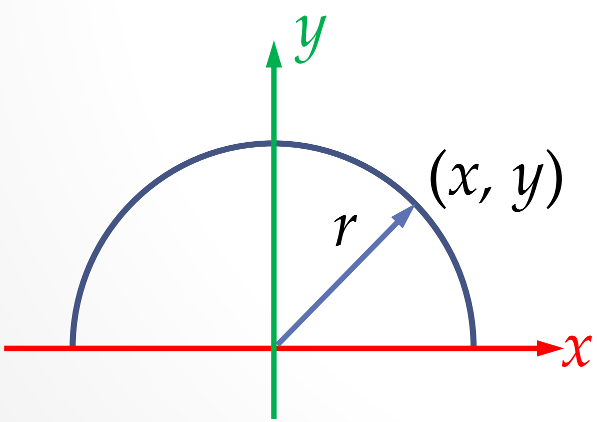
is it always possible to construct *f* given *F*?

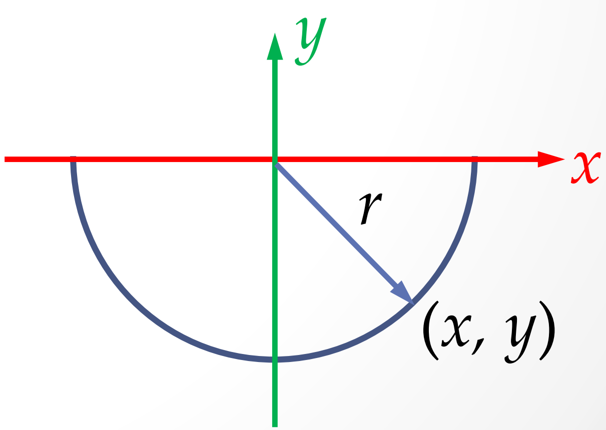
From Implicit Rep. to Explicit Rep.

Is it always possible to **globally** construct *f* given *F*? No.

$$F(x, y) = x^2 + y^2 - r^2 = 0$$

$$f(y) = \sqrt{r^2 - x^2} \qquad f(y) = -\sqrt{r^2 - x^2}$$





have to split the circle since $y = f(x)$ cannot represent multi-value function

From Implicit Rep. to Explicit Rep.

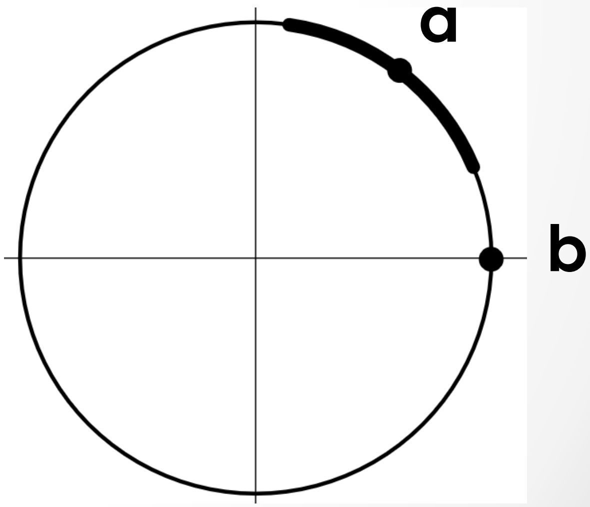
Is it always possible to construct f **locally** given F?

- $y = f(x)$ is defined around **a** = (x_a, y_a).
- $y = f(x)$ is not *well* defined around **b** = (x_b, y_b):

  Derivative $\dfrac{\partial y}{\partial x}$ is infinite at **b**.

- Instead,
  $x = g(y)$ is defined around **b**.

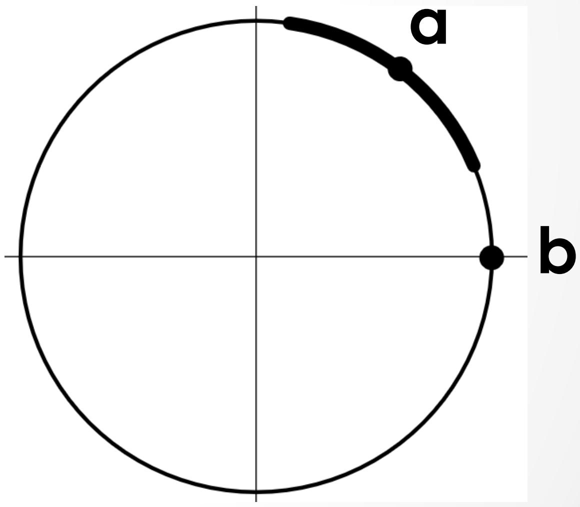
  [roles of x and y exchanged]



From Implicit Rep. to Explicit Rep.

Implicit function theorem:

- Suppose $F(x_a, y_a) = 0$ and $\frac{\partial F}{\partial y}(x_a, y_a) \neq 0$ at **a**=(x_a,y_a).
- Then, it is possible to find *f* around **a**, s.t., y_a=*f*(x_a).



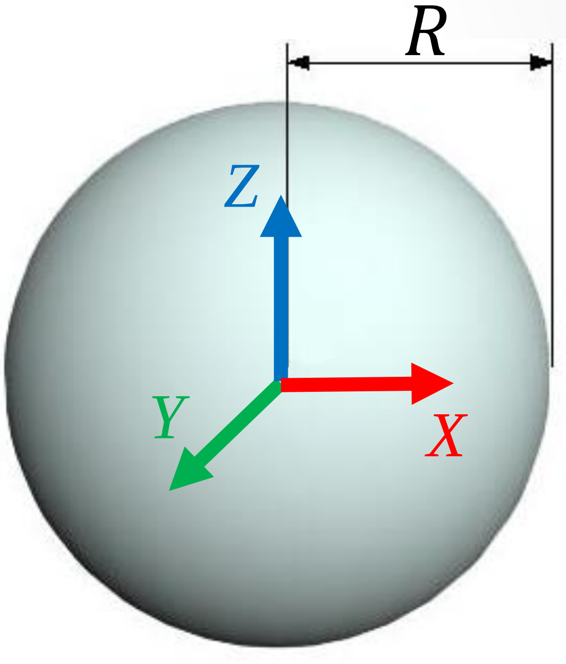
- At **b**=(x_b,y_b), $\frac{\partial F}{\partial x}(x_b, y_b) \neq 0$
- Therefore, we can find *g*: x_b=*g*(y_b)
- In general, if $\nabla F(x_c, y_c) \neq (0{,}0)$,
  it is possible to find *f* (or *g*) around **c**=(x_c,y_c).

Overview

- 2D explicit representation ⇔ implicit representation
- **3D explicit representation ⇔ implicit representation**
- Implicit surface applications
- Implicit surfaces to meshes
  - Sampling implicit function
  - Marching cubes

3D Surface Representation

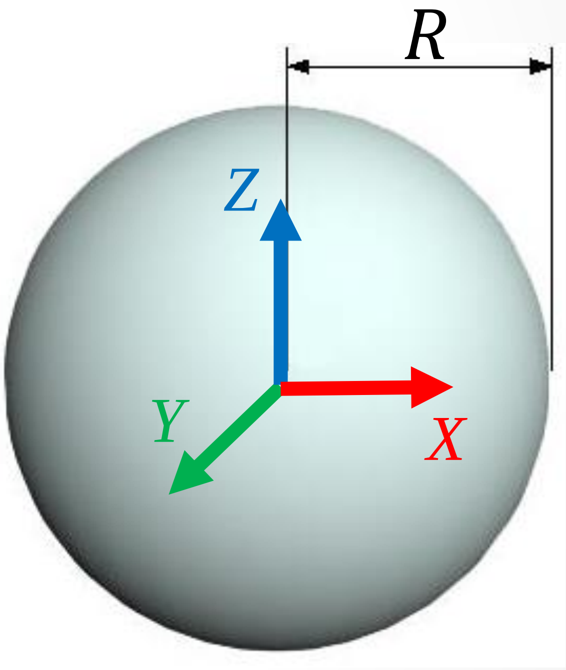
- **Implicit** representation
- **Explicit** representation



3D Surface Representation

- **Implicit** representation $F(x, y, z) = 0$
    - For any surface point, x, y, z coordinates should satisfy a single equation

$$x^2 + y^2 + z^2 - R^2 = 0$$



easily check if a point is on/inside/outside the surface
by evaluating the function value

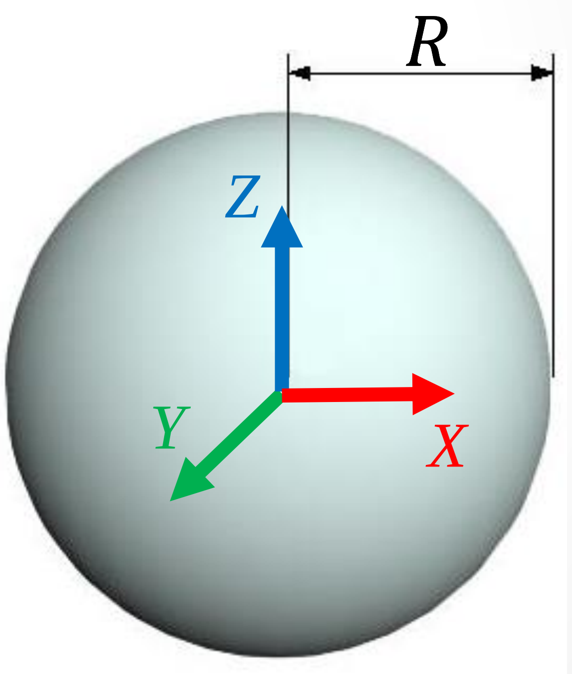
3D Surface Representation

- **Explicit** representation $z = f(x, y)$
  - z coordinate is explicitly represented as a function of x and y coordinates

$$\boxed{x^2 + y^2 + z^2 - R^2 = 0}$$

$$z = \sqrt{R^2 - x^2 - y^2}$$

$$z = -\sqrt{R^2 - x^2 - y^2}$$



simple since z coord. is explicitly represented by x and y coord.
but not convenient for multi-valued surface function

Explicit and Implicit Surfaces

- From explicit rep. to implicit rep.:

$$z = f(x, y)$$
$$\Rightarrow F(x, y, z) = f(x, y) - z = 0.$$

- From implicit rep. to explicit rep.:

if $\dfrac{\partial F(x, y, z)}{\partial z} \neq 0$ at $\mathbf{a}=(a_x,a_y,a_z)$,

one can find f such that $f(x,y)=z$ around $\mathbf{a}$ (similarly for $g(y,z)=x$ and $h(x,z)=y$).

Overview

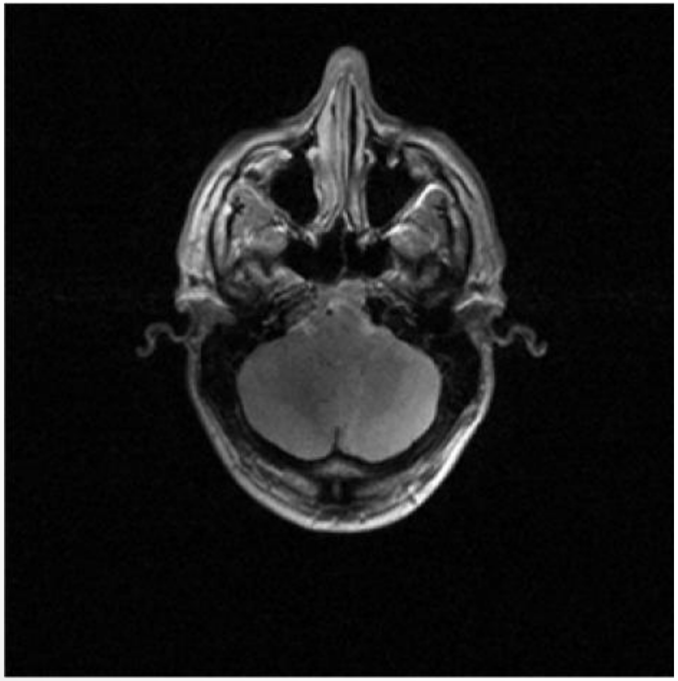
- 2D explicit representation ⇔ implicit representation
- 3D explicit representation ⇔ implicit representation
- **Implicit surface applications**
- Implicit surfaces to meshes
  - Sampling implicit function
  - Marching cubes

Implicit Representation Benefits

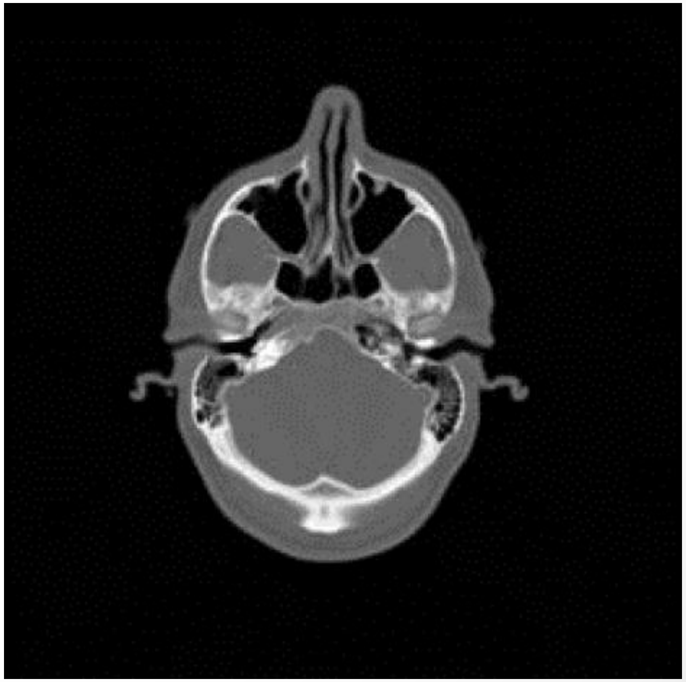
- **Implicit** rep. allows
  - Quick test if a point is on/inside/outside a surface: Easy **collision detection**.
  - Natural representation for **volumetric data**.
  - Suitable representation for **constructive solid geometry**.

Volumetric Data

- CT and MRI scanners produce volume of density values $F(x,y,z)$.
- Individual features (e.g., bone, brain surfaces) are implicit-surfaces of the volume with different iso-value c: $F(x,y,z)=c$.



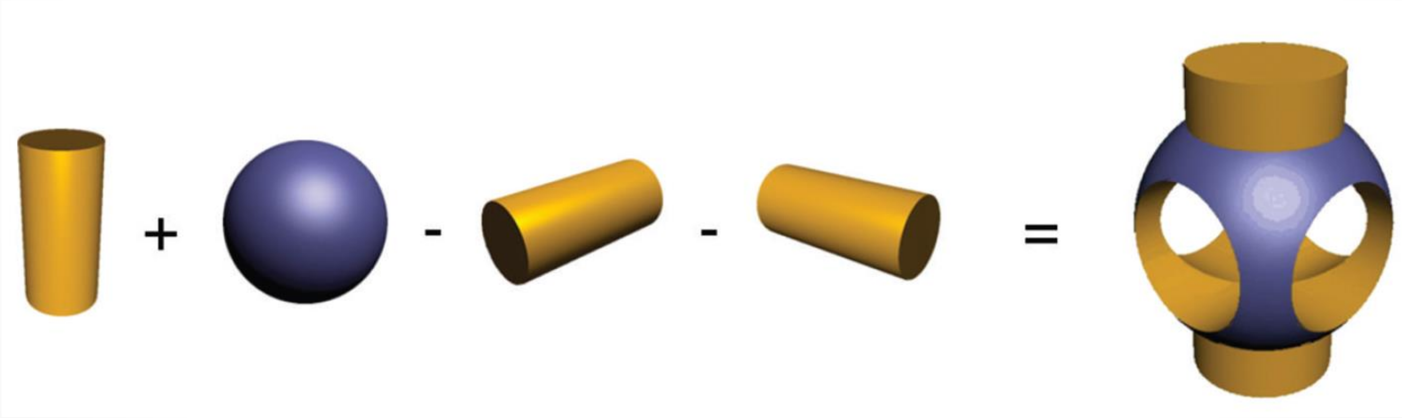
MRI



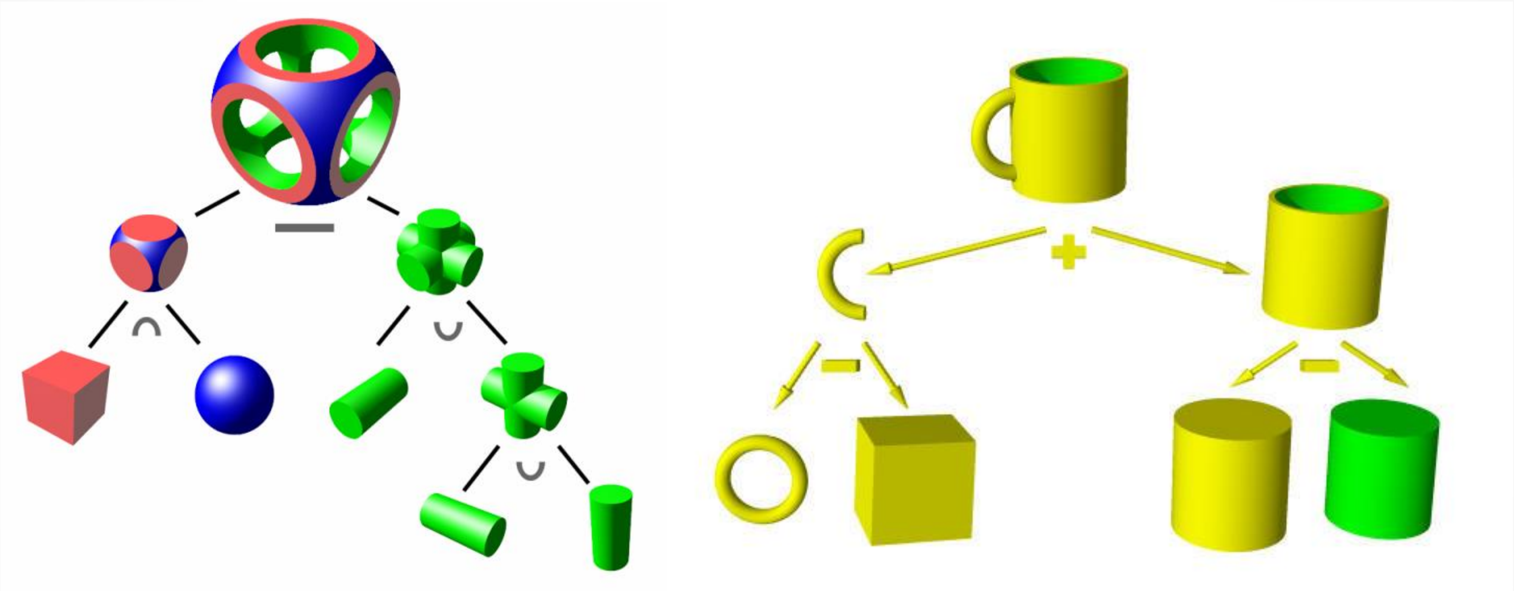
CT

Constructive Solid Geometry (CSG)

- Designers build up a shape by using three dimensional blocks and a selection of **Boolean operations** in which they can **combine**.



Constructive Solid Geometry (CSG)

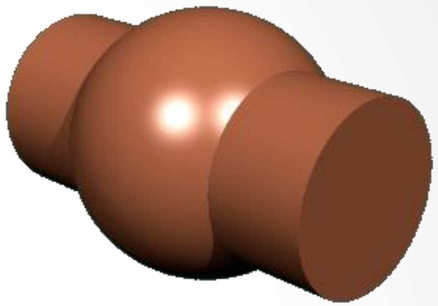


Why is implicit surface modelling suitable for CSG?

Implicit Surface + CSG

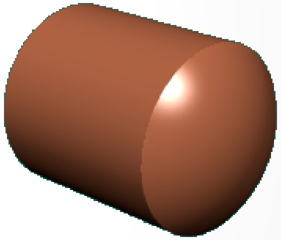
- Union

$$F_{C\cup S}(\cdot) = \min\{F_C(\cdot), F_S(\cdot)\}$$



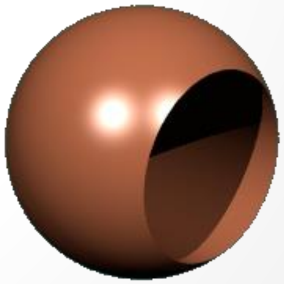
- Intersection

$$F_{C\cap S}(\cdot) = \max\{F_C(\cdot), F_S(\cdot)\}$$



- Difference

$$F_{S\backslash C}(\cdot) = \max\{-F_C(\cdot), F_S(\cdot)\}$$



$F(x,y,z)<0$ means inner, $F(x,y,z)>0$ means outer, $F(x,y,z)=0$ gives implicit surface

Overview

- 2D explicit representation ⇔ implicit representation
- 3D explicit representation ⇔ implicit representation
- Implicit surface applications
- **Implicit surfaces to meshes**
    - Sampling implicit function
    - Marching cubes

Mesh Representation

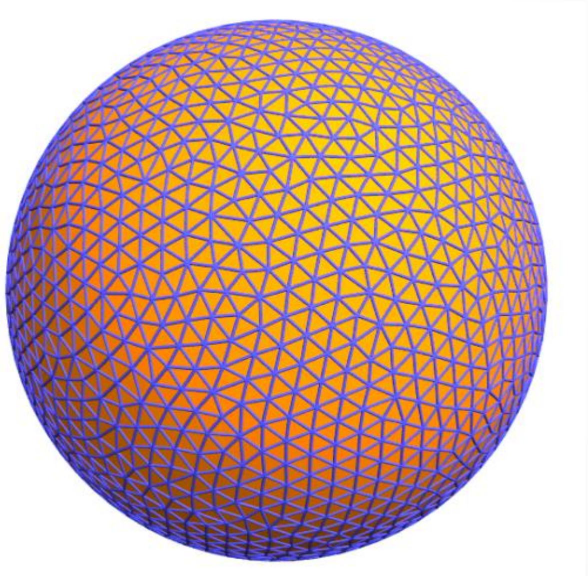
- A **discrete** 3D surface representation

$$M = (V, E, F)$$

$V: mesh\ vertex\ set$

$E: mesh\ edge\ set$

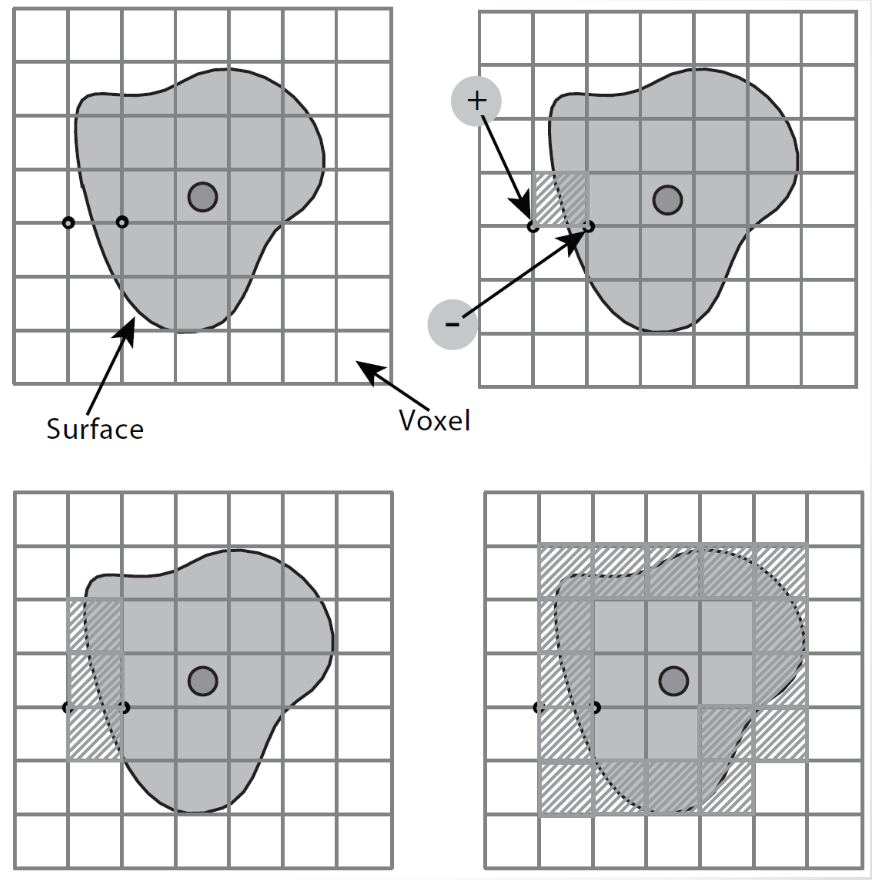
$F: mesh\ face\ set$



How to convert implicat representation for mesh representation?
applications: visualization, simulation, etc.

Sampling Implicit Function

- + sign: a point **x** is outside the curve ($F(\mathbf{x})>0$).
- - sign: inside ($F(\mathbf{x})<0$).
- Classify all squares based on $F(\mathbf{x})$
  - Inside/outside/intersection.
- Intersected cells (squares) are refined.



Marching Square (2D)

1. Classify grid nodes as inside/outside

   Is $F(u,v)>0$ or <0?

2. Classify cell: 2^4 configuration

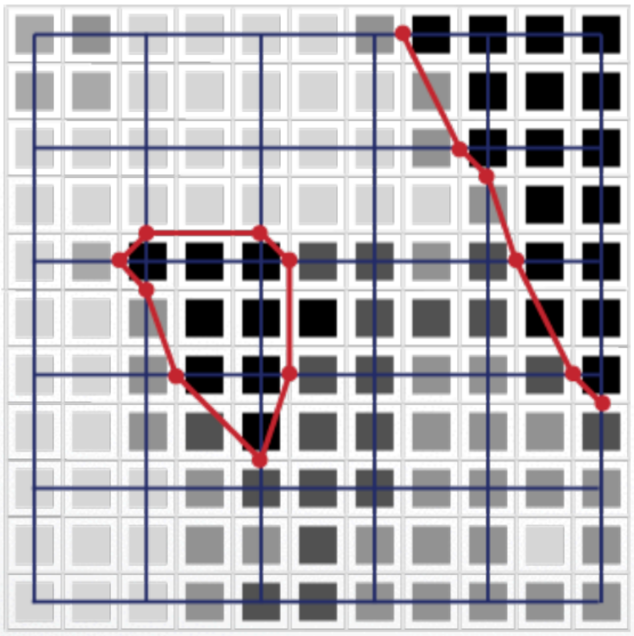
   In/out for each corner.

3. Compute zero-crossing intersection points

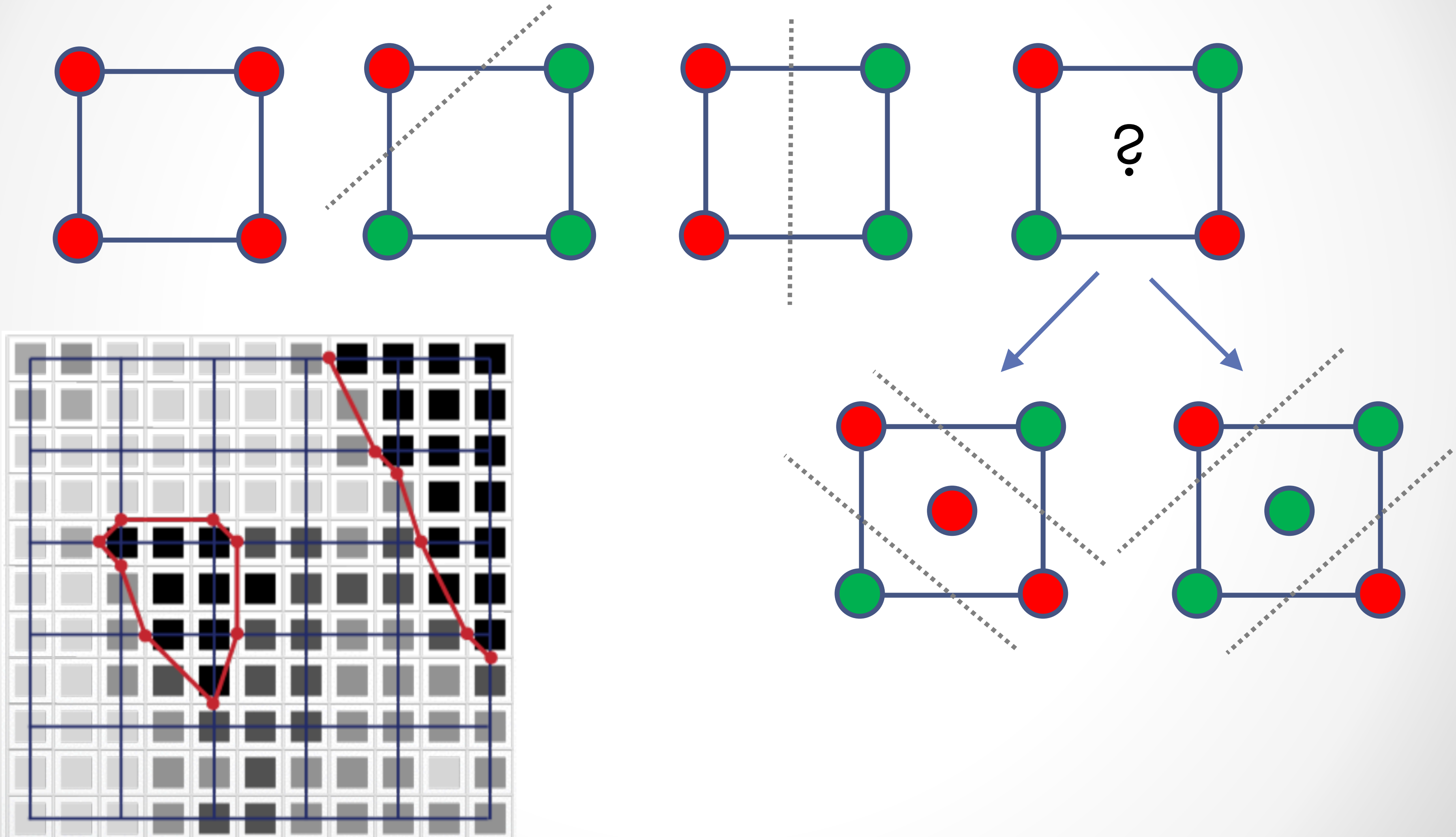
   Linear **interpolation** along edges.

4. Connect zero-crossing by edges

   Look-up table for edge configuration.



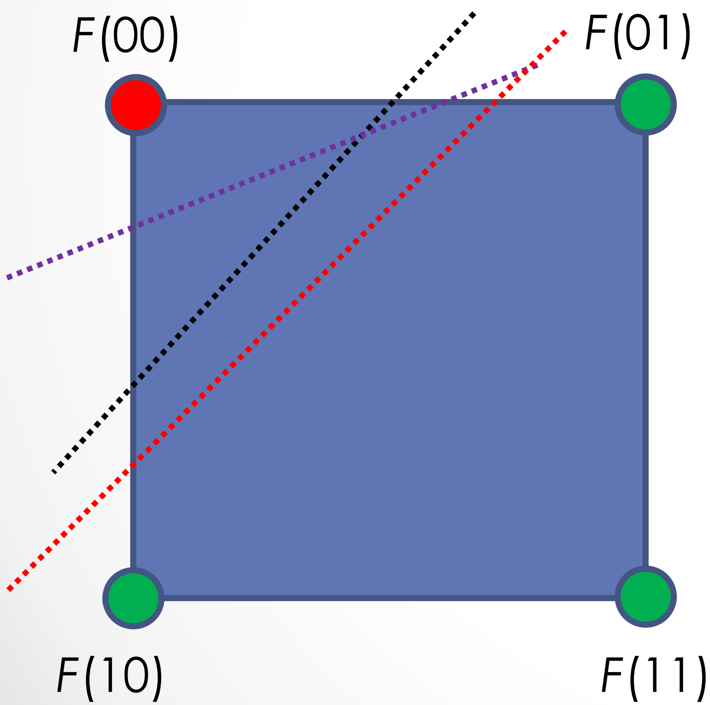
Marching Square (2D)

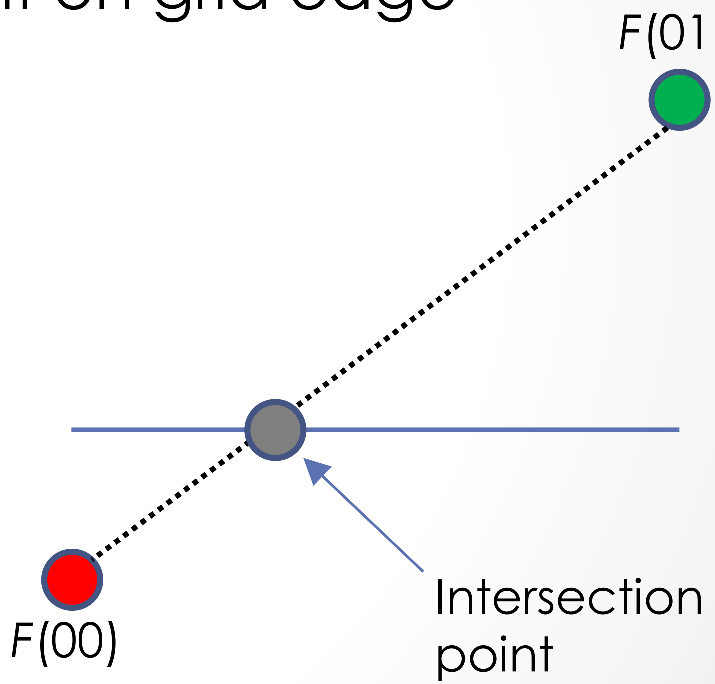


Intersection Points

Regular 2D grid

- Compute implicit function values at nodes.
- Computer zero-crossing point on grid edge





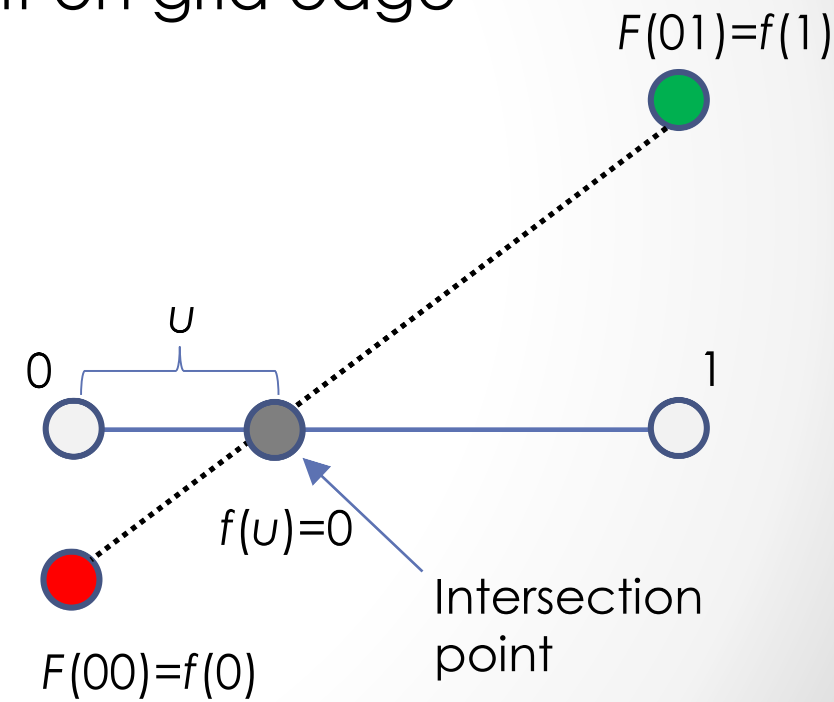
Intersection Points

Regular 2D grid

- Compute implicit function values at nodes.
- Computer zero-crossing point on grid edge

$$f(u) = uf(0) + (1 - \mathrm{u})f(1) = 0$$

$$\Leftrightarrow u = \frac{f(1)}{f(0) - f(1)}$$



Marching Cubes

1. Classify grid nodes as inside/outside

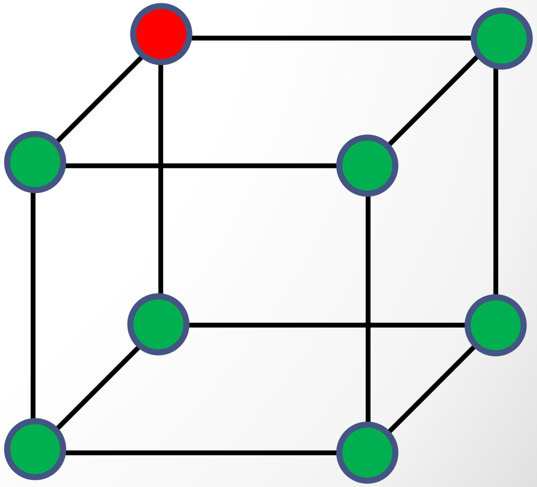
   Is $F(u, v, w)>0$ or <0?
2. Classify cell: 2^8 configuration

   In/out for each corner.
3. Compute boundary intersection points.

   Linear **interpolation** along edges.
4. Connect them by edges.

   Look-up table for edge configuration.

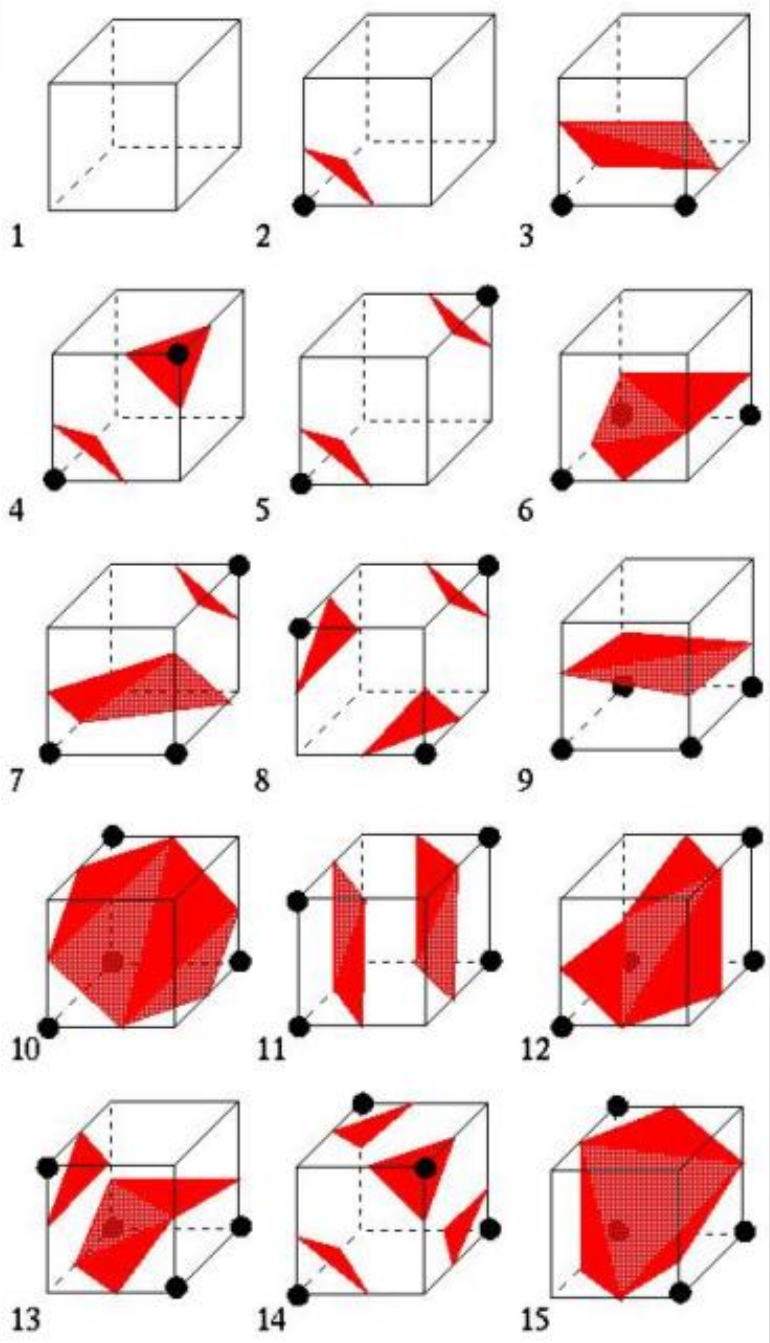
[Lor87]



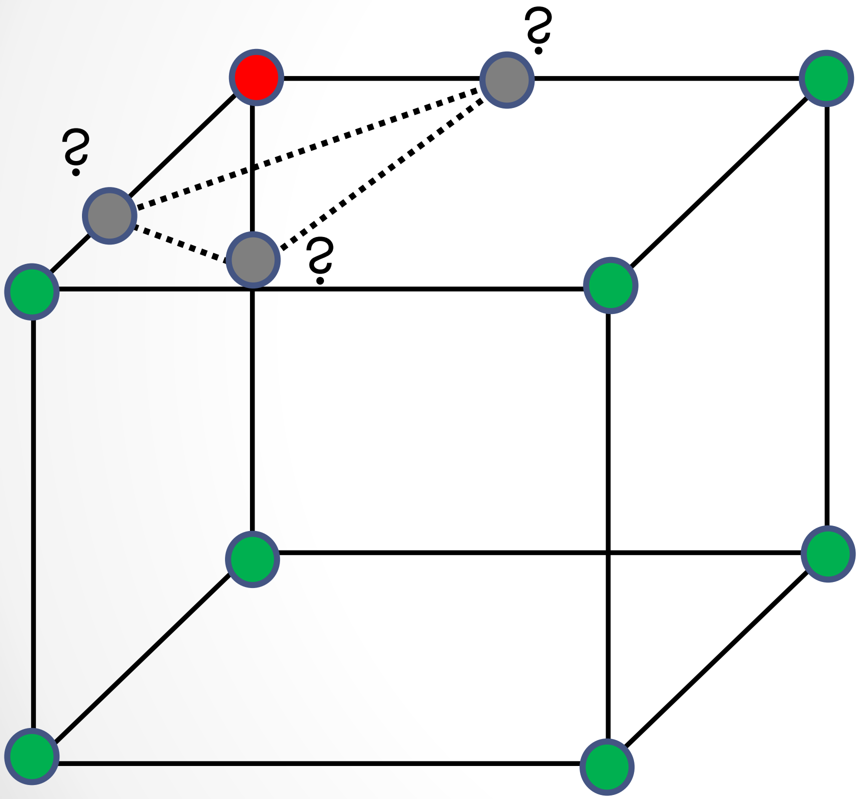
Marching Cubes

Look-up table with 2^8 entries

- 256 cases are reduced to 15 cases based on symmetry

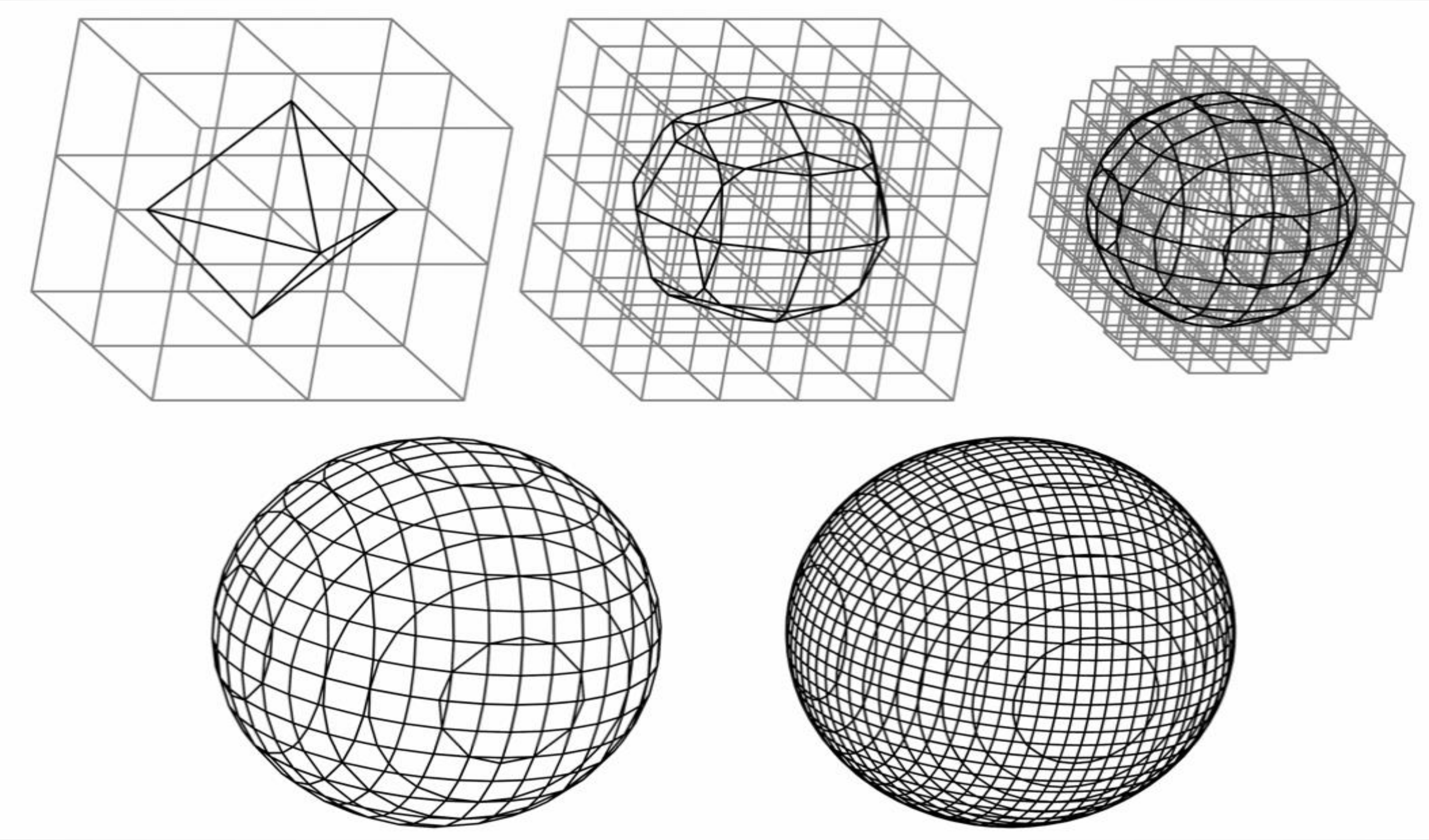


Intersection Points

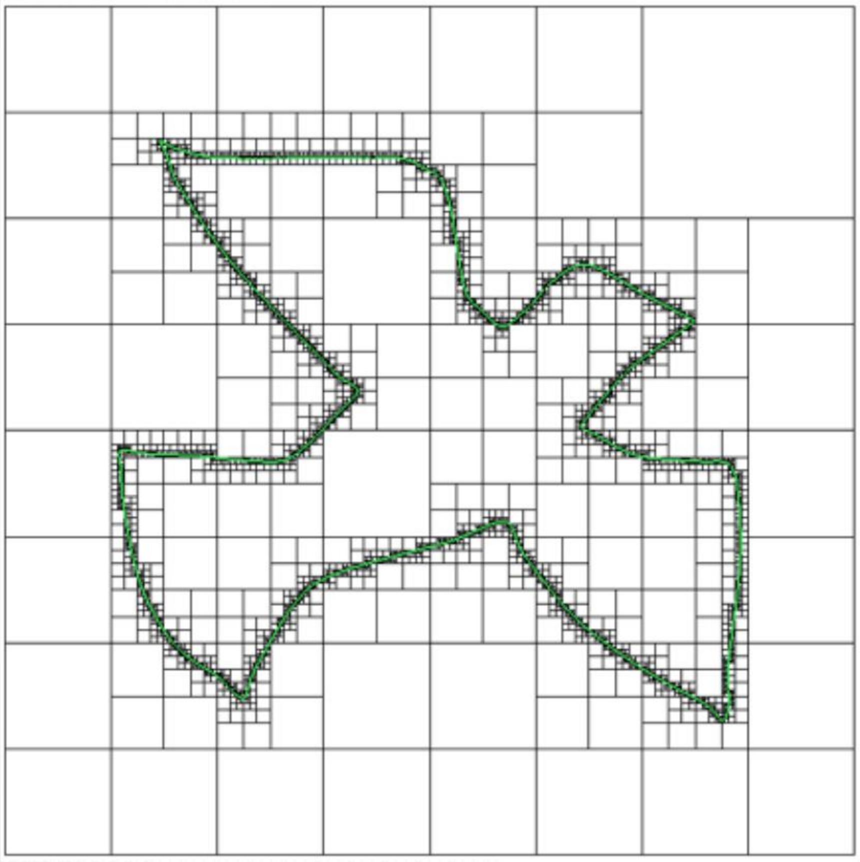


- Linear interpolation along edges.
- Lookup table for mesh face configuration.

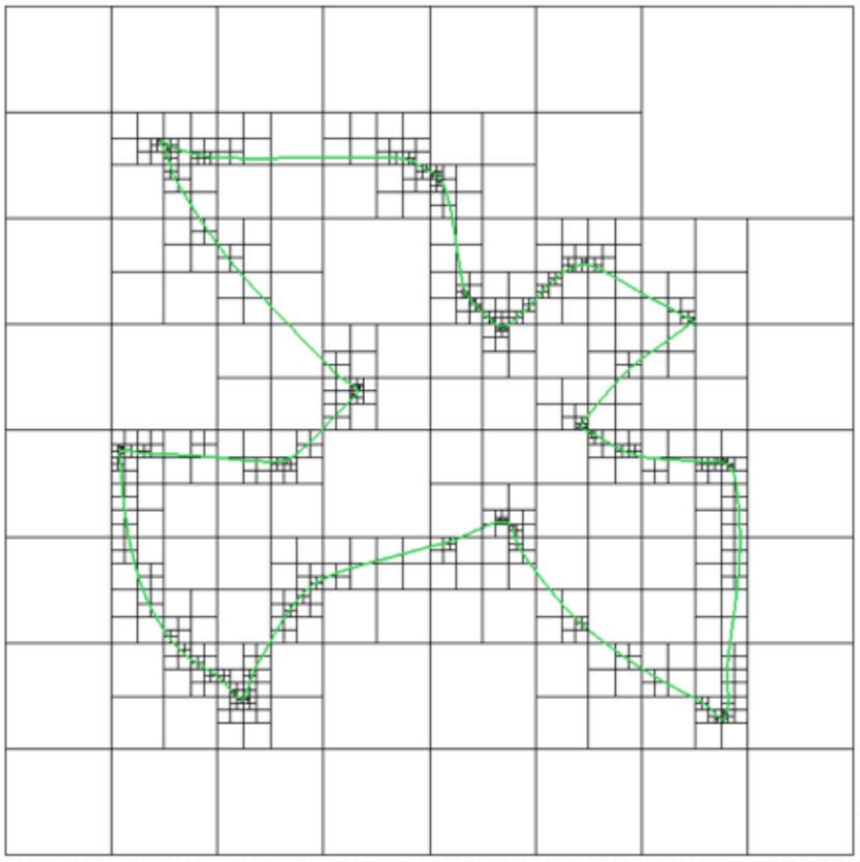
Uniform Sampling



Progressive Sampling



12040 cells



895 cells

- [Li16]

References

[Shi09] Shirley et al., *Fundamentals of Computer Graphics*, 2009.

[PAR12] Parent, *Computer Animation Algorithms & Techniques*, 2012.

[Li16] Li, *Explicit & Implicit Surfaces*.

[Lor87] Lorensen and Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," Computer Graphics, vol. 21, no. 3, pp. 163-169, July 1987