Marching Squares

Objectives

- Nontrivial two-dimensional application
- Important method for
  - Contour plots
  - Implicit function visualization
- Extends to important method for volume visualization
- This lecture is optional
- Material not needed to continue to Chapter 3

Displaying Implicit Functions

- Consider the implicit function

$$g(x,y)=0$$

- Given an x, we cannot in general find a corresponding y
- Given an x and a y, we can test if they are on the curve

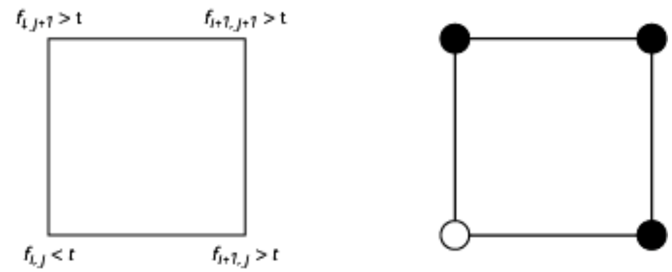
Height Fields and Contours

- In many applications, we have the heights given by a function of the form $z=f(x,y)$
- To find all the points that have a given height c, we have to solve the implicit equation $g(x,y)=f(x,y)-c=0$
- Such a function determines the isosurfaces or contours of f for the isosurface value c

Marching Squares

- Displays isocurves or contours for functions $f(x,y) = t$
- Sample $f(x,y)$ on a regular grid yielding samples $\{f_{ij}(x,y)\}$
- These samples can be greater than, less than, or equal to t
- Consider four samples $f_{ij}(x,y)$, $f_{i+1,j}(x,y)$, $f_{i+1,j+1}(x,y)$, $f_{i,j+1}(x,y)$
- These samples correspond to the corners of a cell
- Color the corners by whether than exceed or are less than the contour value t

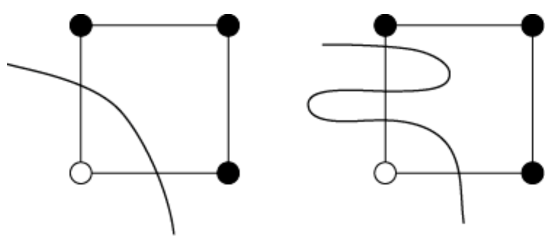
Cells and Coloring



(a) (b)

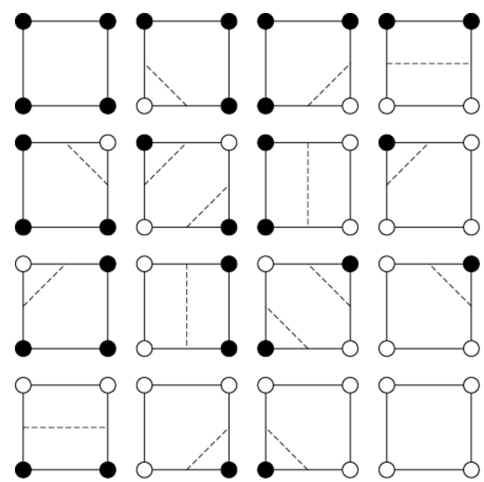
Occam's Razor

- Contour must intersect edge between a black and white vertex an odd number of times
- Pick simplest interpretation: one crossing



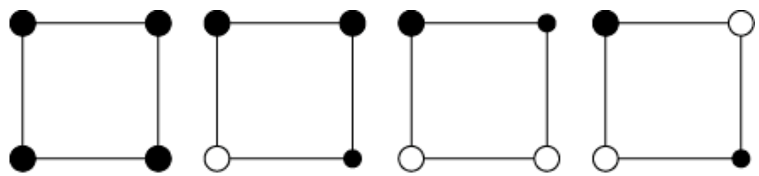
(a) (b)

16 Cases



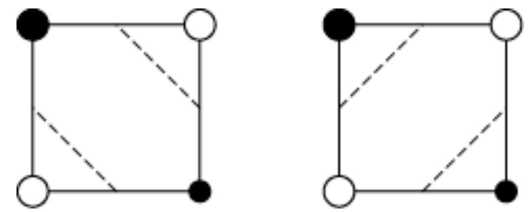
Unique Cases

- Taking out rotational and color swapping symmetries leaves four unique cases
- First three have a simple interpretation



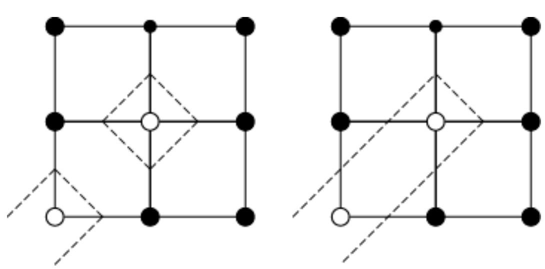
Ambiguity Problem

- Diagonally opposite cases have two equally simple possible interpretations

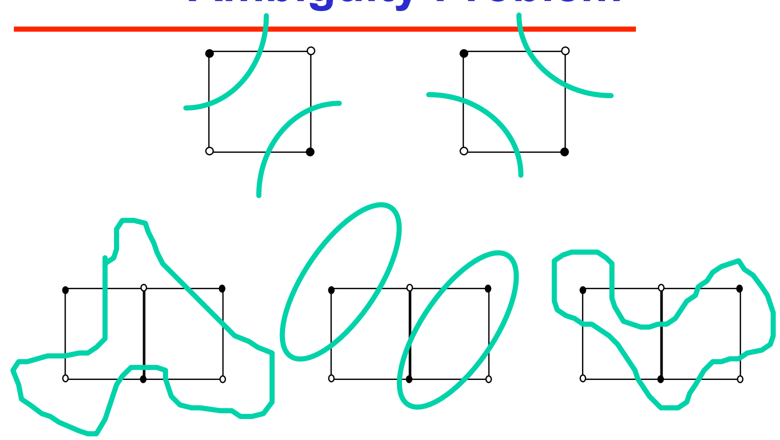


Ambiguity Example

- Two different possibilities below
- More possibilities on next slide



Ambiguity Problem

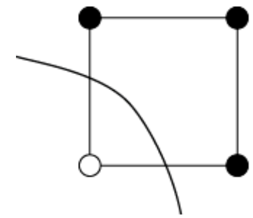


Is Problem Resolvable?

- Problem is a sampling problem
  - Not enough samples to know the local detail
  - No solution in a mathematical sense without extra information
- More of a problem with volume extension (marching cubes) where selecting “wrong” interpretation can leave a hole in a surface
- Multiple methods in literature to give better appearance
  - Supersampling
  - Look at larger area before deciding

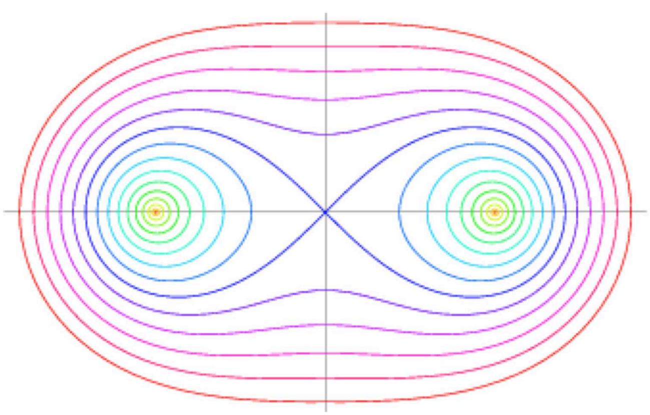
Interpolating Edges

- We can compute where contour intersects edge in multiple ways
  - Halfway between vertices
  - Interpolated based on difference between contour value and value at vertices



Example: Oval of Cassini

$f(x,y)=(x^2+y^2+a^2)^2-4a^2x^2-b^4$

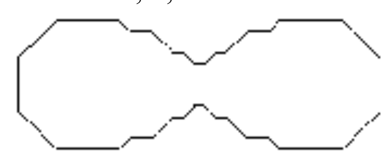


Example: Oval of Cassini

$f(x,y)=(x^2+y^2+a^2)^2-4a^2x^2-b^4$

Depending on a and b we can have 0, 1, or 2 curves

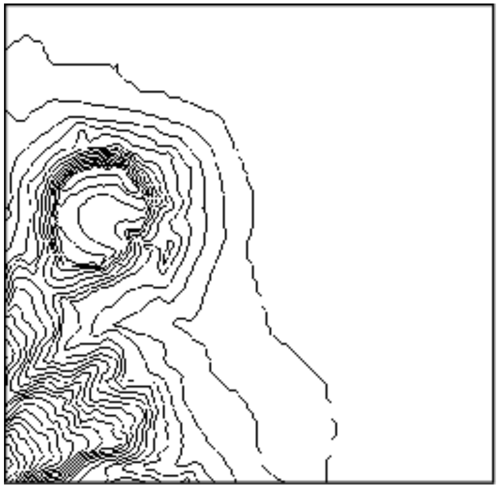
midpoint intersections



interpolating intersections

Contour Map

- Diamond Head, Oahu Hawaii
- Shows contours for many contour values



Marching Cubes

- Isosurface: solution of g(x,y,z)=c
- Same argument to derive method but use cubic cell (8 vertices, 256 colorings)
- Standard method of volume visualization
- Suggested by Lorensen and Kline before marching squares

Marching Cubes

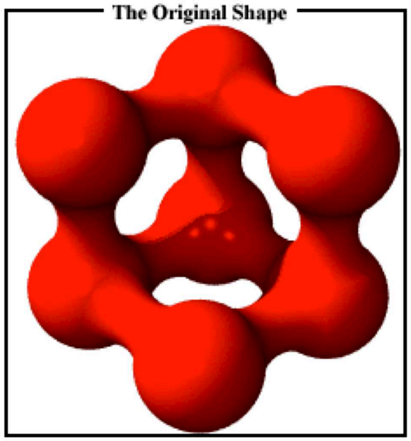
The Marching Cubes algorithm was designed by William E. Lorensen and Harvey E. Cline to extract surface information from a 3d field of values. The input data set can represent anything from medical imaging data to geological scans.

From:

http://www.exaflop.org/docs/marchcubes/ind.html

http://www.essi.fr/~lingrand/MarchingCubes/accueil.html

Surface



Basic Idea

The basic principle behind the marching cubes algorithm is to subdivide space into a series of small cubes.

The algorithm then instructs us to 'march' through each of the cubes testing the corner points and replacing the cube with an appropriate set of polygons.

The sum total of all polygons generated will be a surface that approximates the one the data set describes.

The 3D algorithm

The cubes have 8 corners and therefore a potential 256 possible combinations of corner status.

To simplify the algorithm we can reduce the complexity by taking into account cell combinations that duplicate under the following conditions.

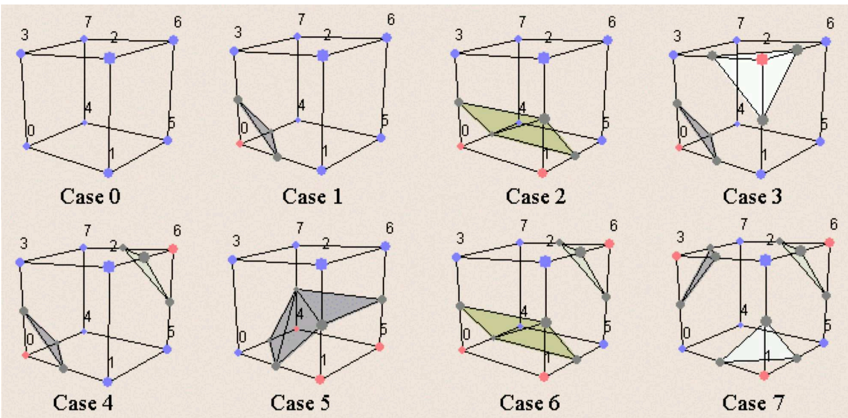
Duplication Conditions

- Rotation by any degree over any of the 3 primary axes
- Mirroring the shape across any of the 3 primary axes
- Inverting the state of all corners and flipping the normals of the relating polygons.

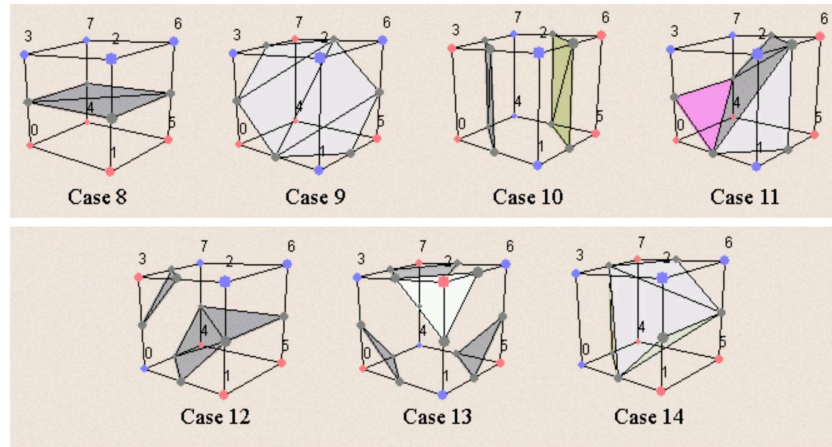
Result

Taking this into account we can resolve the original 256 combinations of cell state down to a total of 15 combinations, with this number it is then easy to create predefined polygon sets for making the appropriate surface approximation.

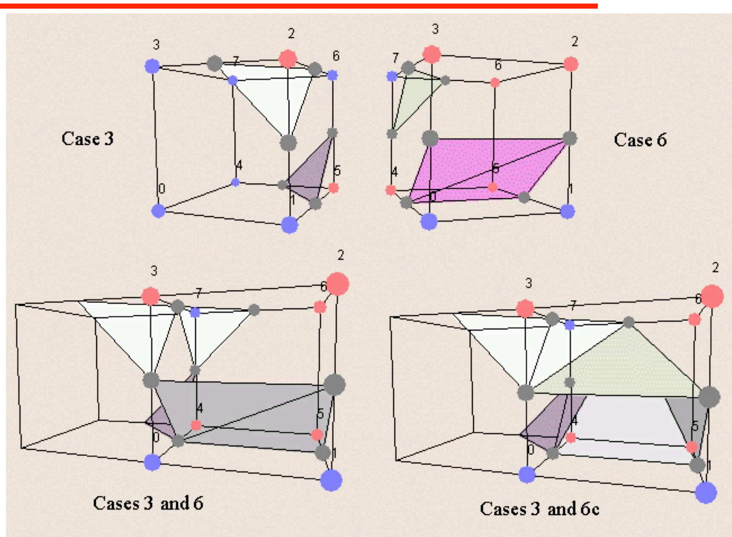
Cases - Marching Cubes



Cases - Marching Cubes



Ambiguous Cases



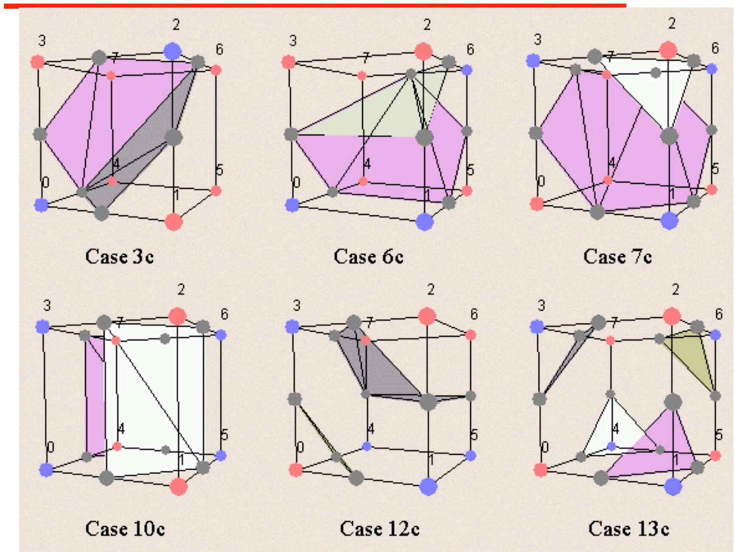
Extensions to cope

To cope with these topology errors (as holes in the 3D model), 6 families have been added to the marching cubes cases.

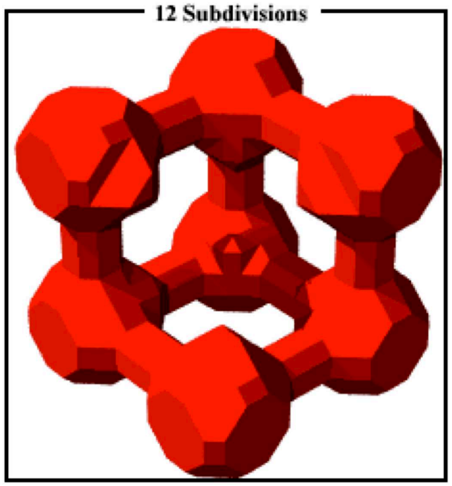
These families have to be used as complementary cases.

For instance, in the previous picture, you have to use the case 6c instead of the standard complementary of the case 6.

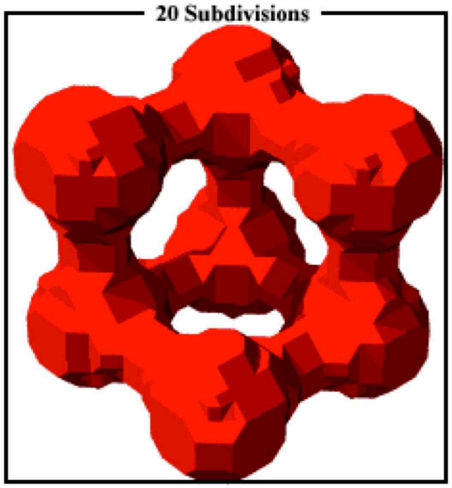
Ambiguous Case Extensions



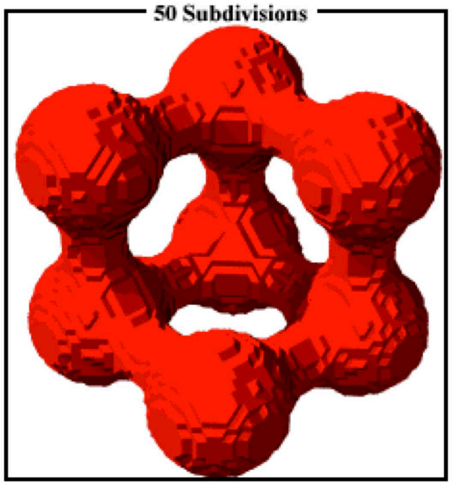
12 Samples/axis



20 Samples/axis



50 Samples/axis



Analysis

Immediately we can see the approximate shape of our test shape even at the lowest sampling frequency, at the top end we can see quite a bit of detail.

Unfortunately we are also seeing the same kind of spatial aliasing as we saw in the 2D example.

Improvements?

- Smoothing?
- Interpolation?
- Other?

If interpolation is used, the algorithm is known as the "Adaptive Marching Cubes" algorithm.

Successive Approximation

We ask the question is point (x,y,z) inside or outside of the shape?

One of the ends of the edge is inside the shape and one is outside.

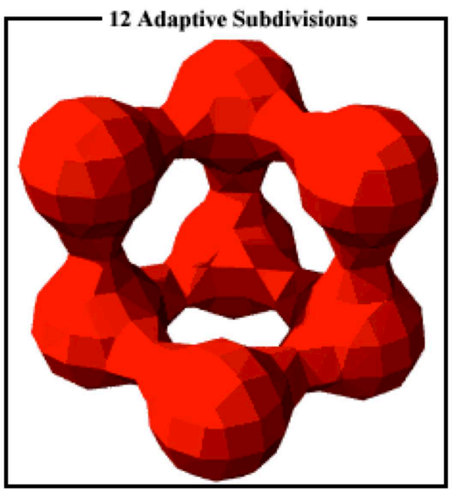
Check an additional point between these two.

The result is a region half the size in which we know the surface resides.

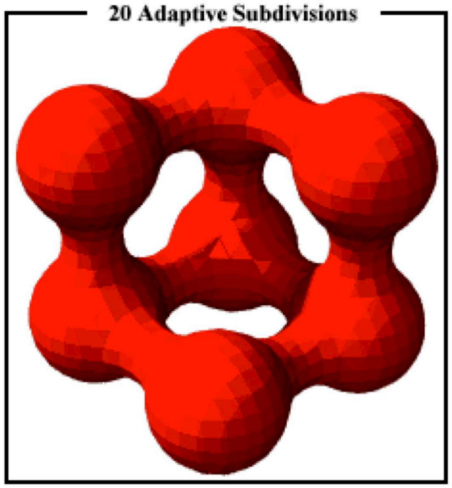
Repeating this process recursively only a few more times gives a very accurate result.

Can only be applied to implicit functions.

Successive Approximation



Successive Approximation



Variations

- **Marching Tetrahedra**
- Maximum of Six Edge Intersections
- Only 3 Unique Entries in Table
- Maximum of Two Triangles/Tetrahedron
- Most Polyhedra Decompose Into Tetrahedra
- Cells Can be Subdivided Into 5, 6, or 24 Tetrahedra Cells
- 5 Tetrahedra case requires flipping adjacent cells, otherwise have anomalous, unconnected surfaces

Tetrahedra Cases