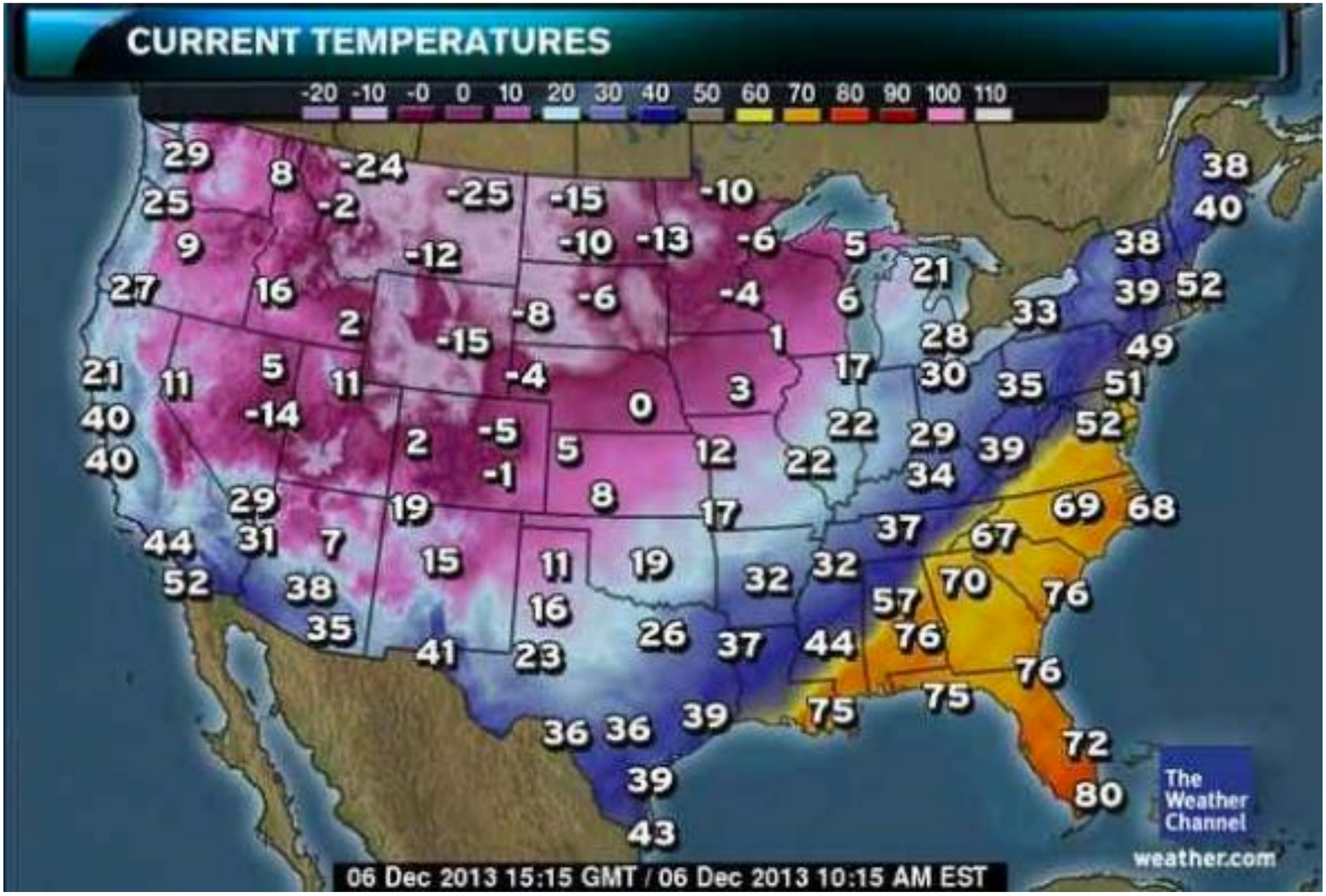
Visualizing Scalar Fields

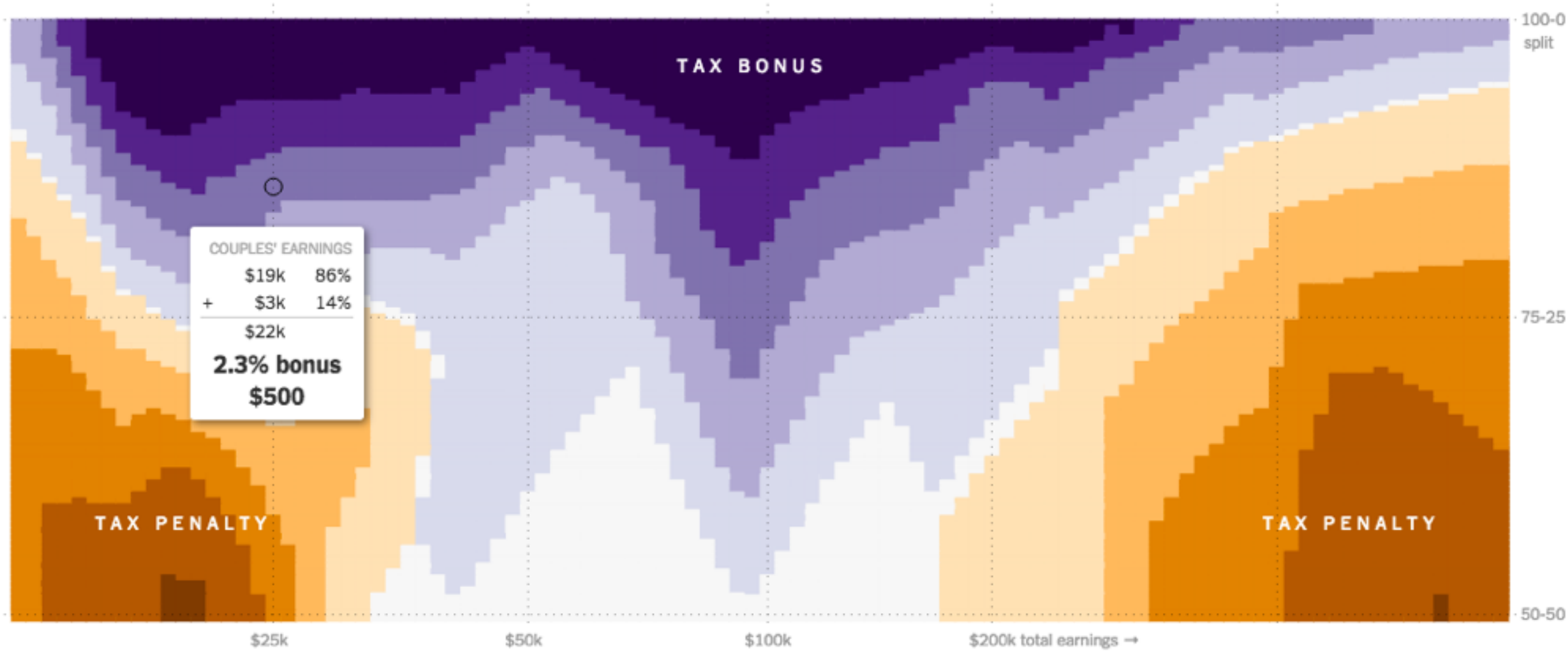
Colormapping

- “Default” strategy:
  - create color scale using the **range** of the function as the **domain** of the scale
  - create a position scale to convert **from the domain of the function to positions on the screen**
  - set the pixel color according to the scale

Colormapping guidelines apply!

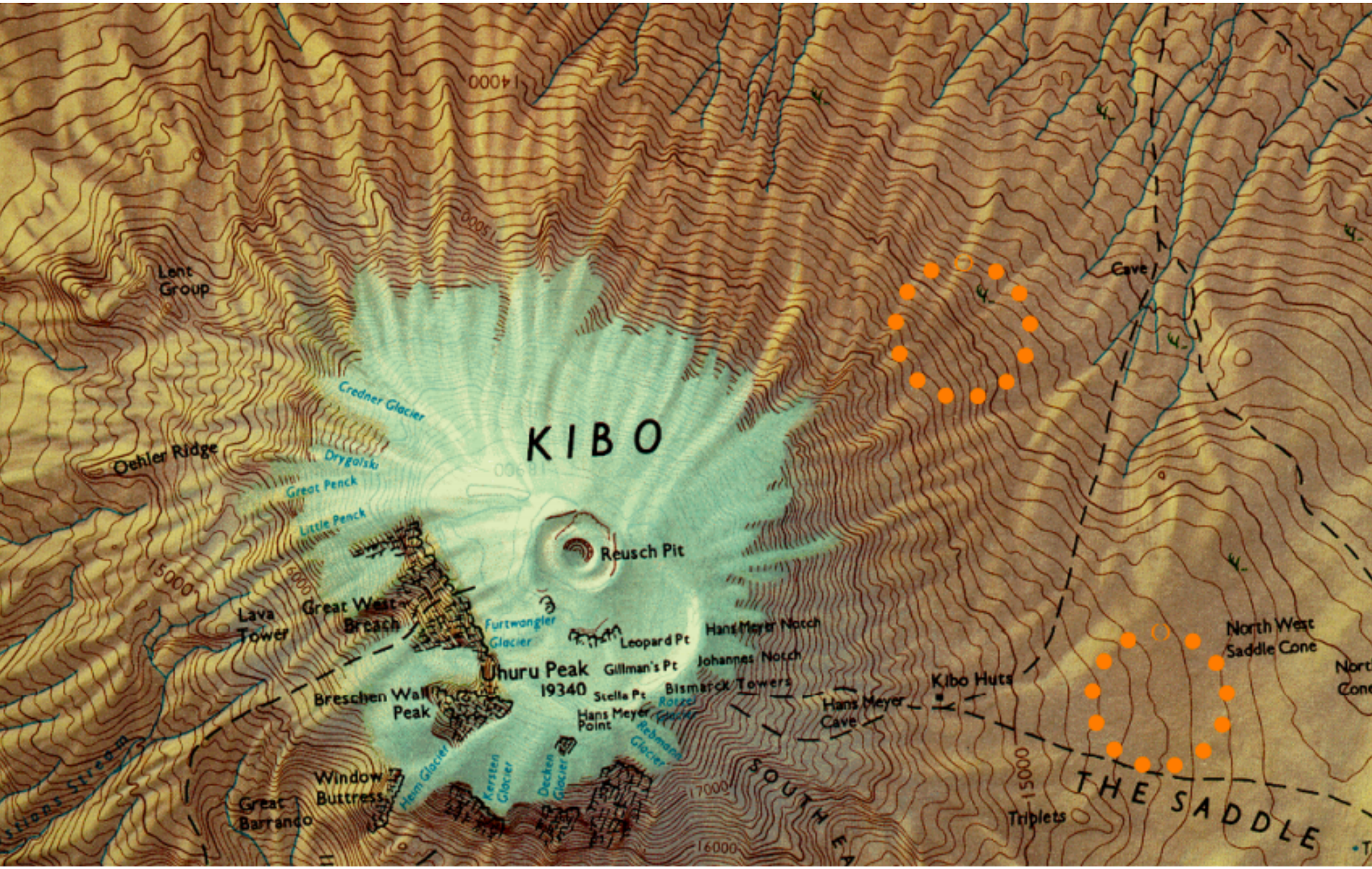


Applies to “abstract” spaces too

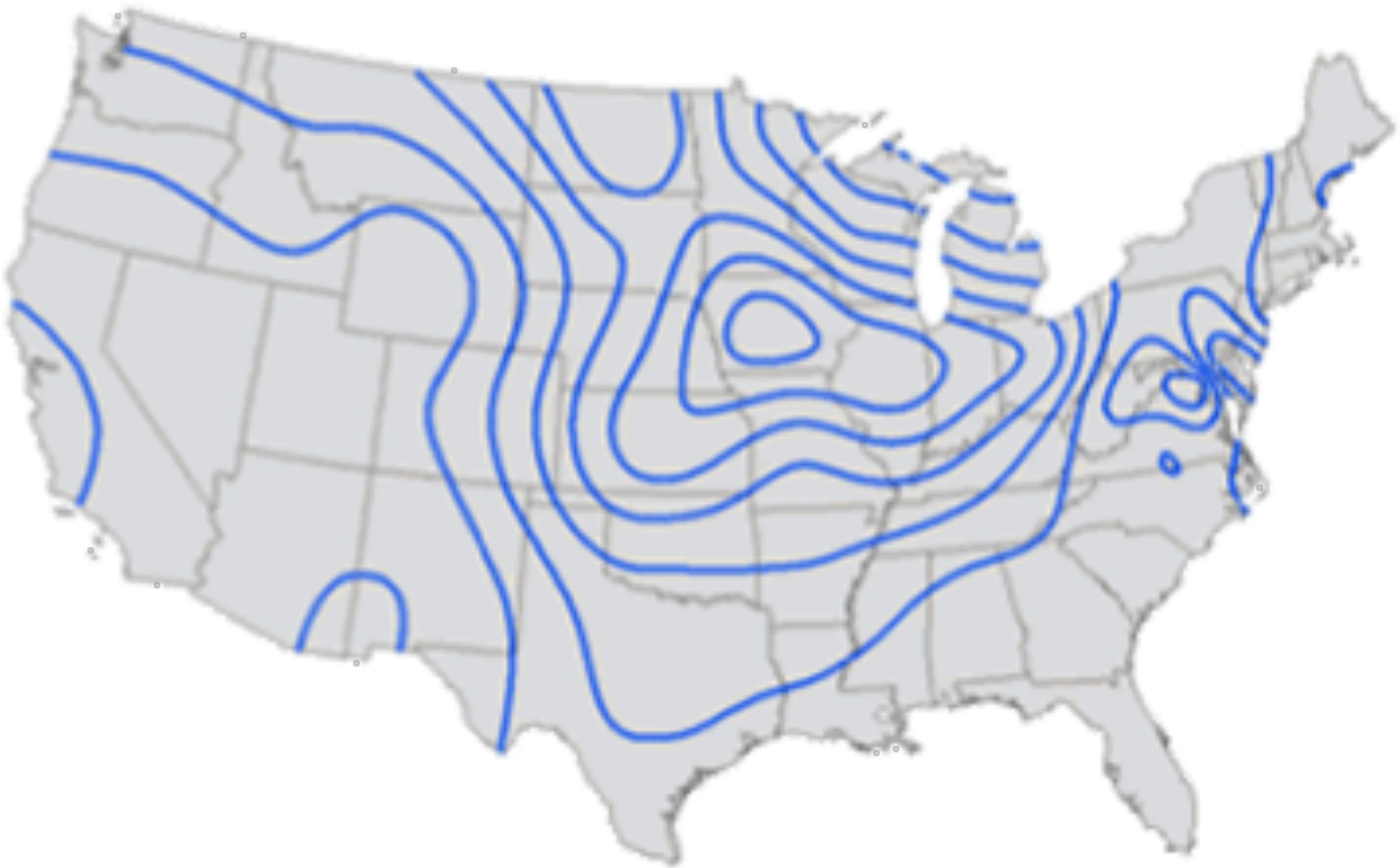


http://www.nytimes.com/interactive/2015/04/16/upshot/marriage-penalty-couples-income.html?abt=0002&abg=0

Contouring: isolines



Contouring: isolines



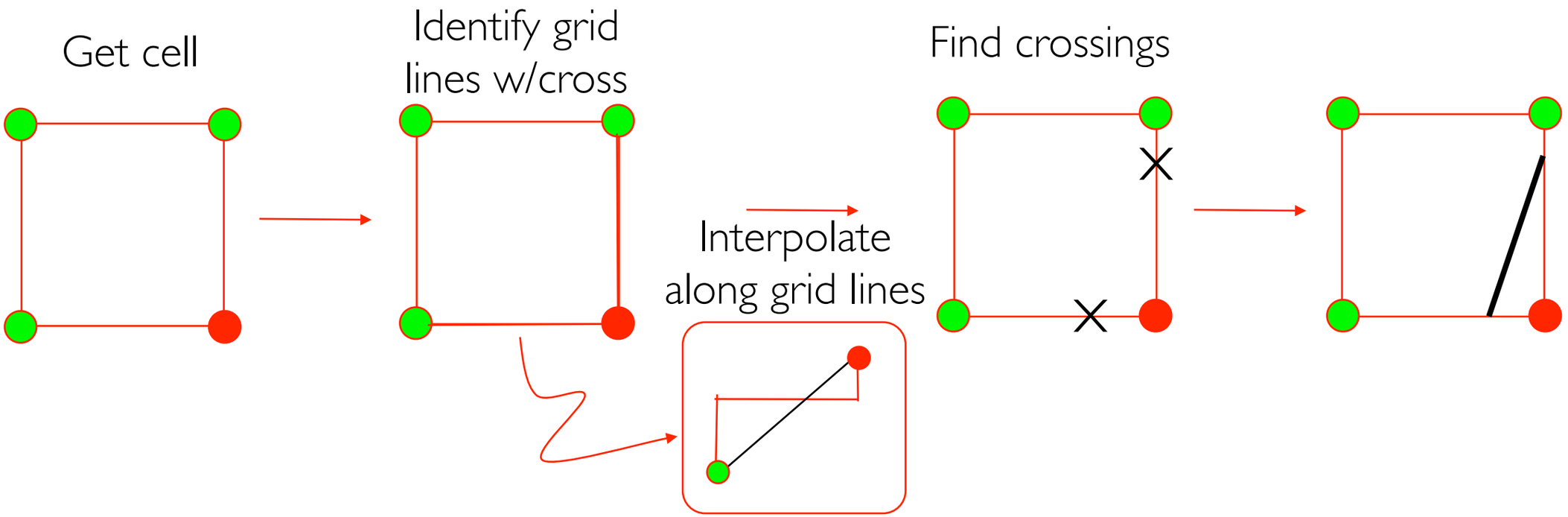
http://ryanhill1.blogspot.com/2011/07/isoline-map.html

Contouring: isolines

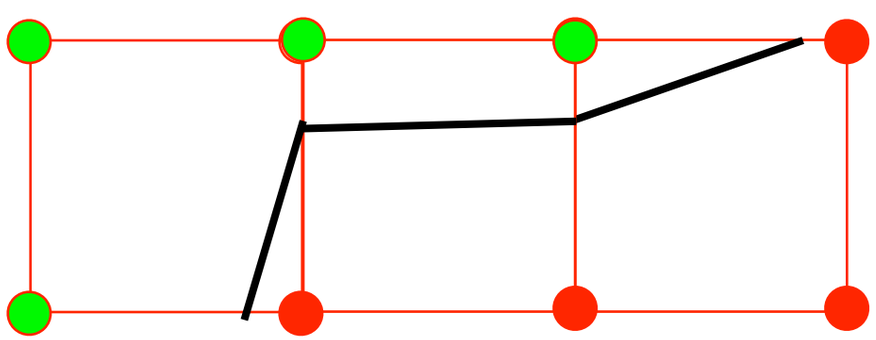
How do we compute them?

Approach to Contouring in 2D

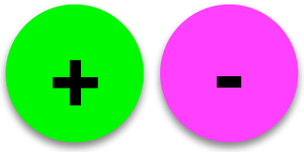
- Contour must cross every grid line connecting two grid points of opposite sign

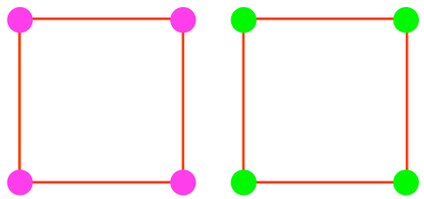
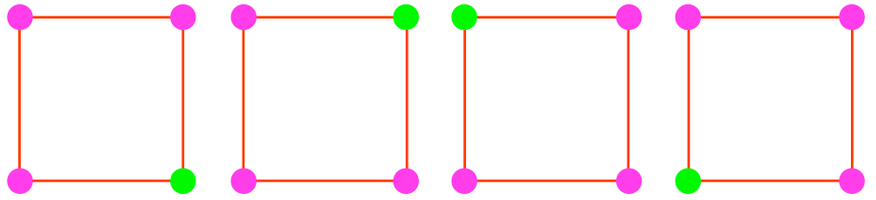
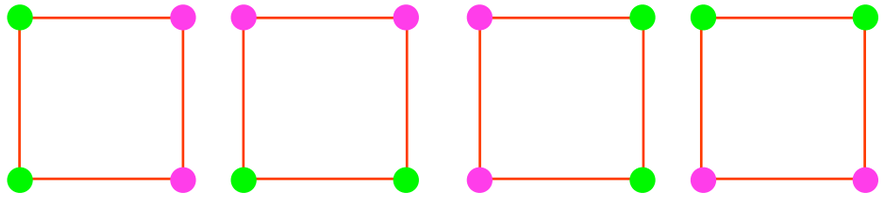
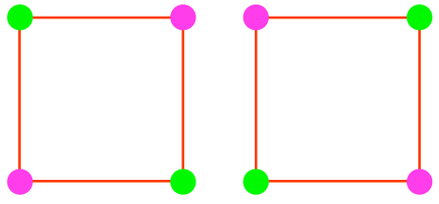


Primitives naturally chain together



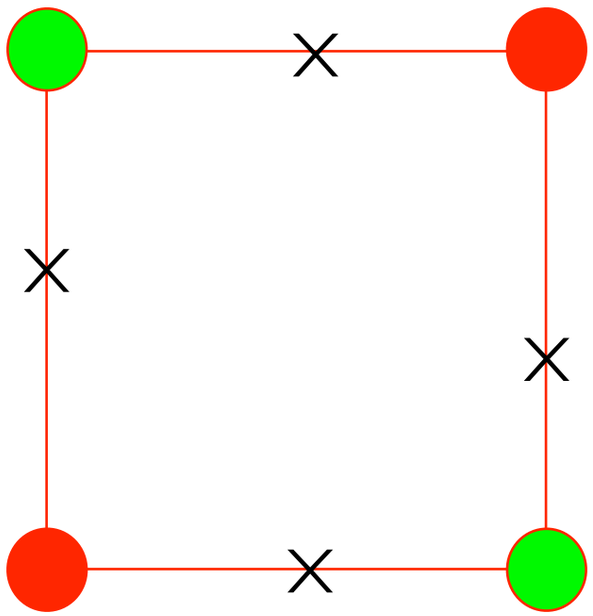
Cases



Case	Polarity	Rotation	Total	
No Crossings	x2		2	
Singlet	x2	x4	8	 (x2 for polarity)
Double adjacent	x2	x2 (4)	4	
Double Opposite	x2	x1 (2)	2	
			$16 = 2^4$	

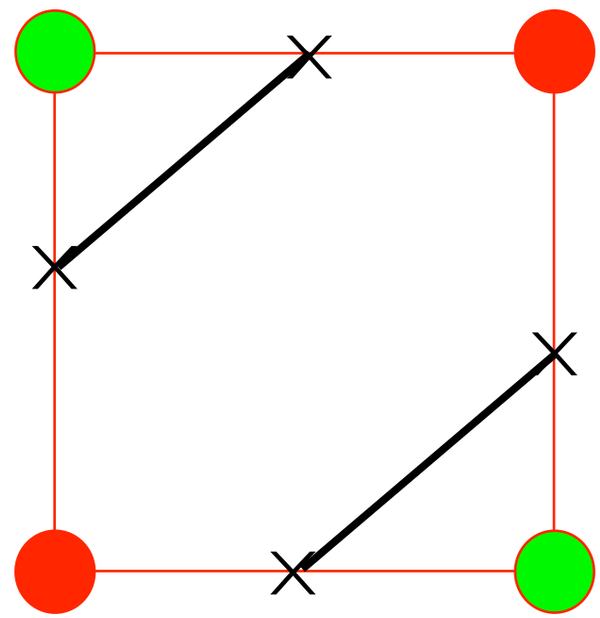
Ambiguities

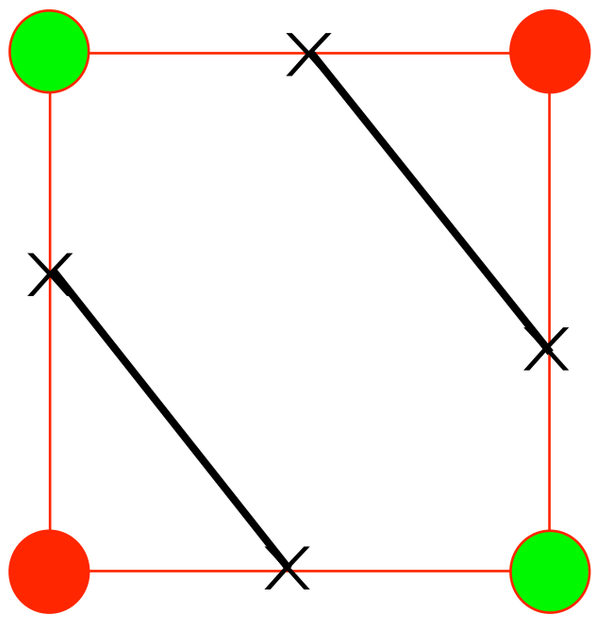
- How to form lines?

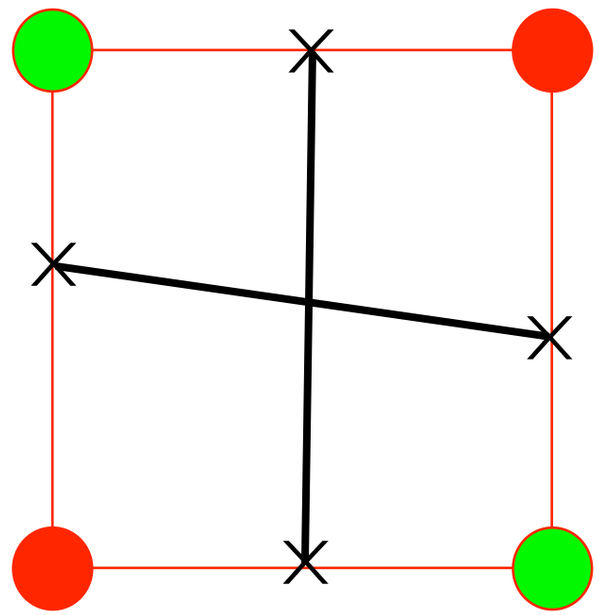


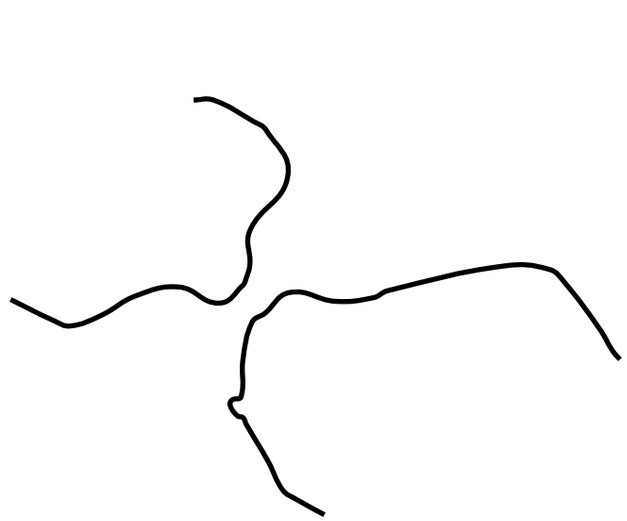
Ambiguities

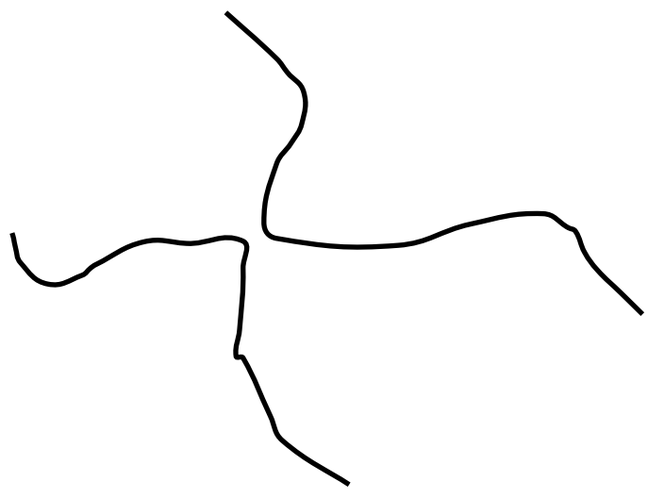
- Right or Wrong?

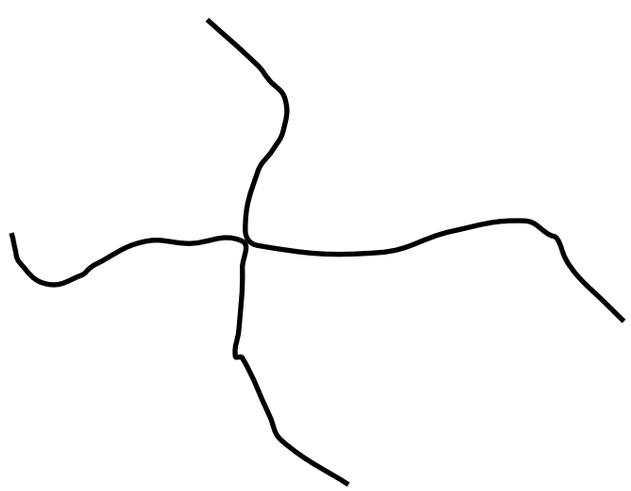












The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes

Gregory M. Nielson Bernd Hamann

Computer Science
Arizona State University
Tempe, AZ 85287-5406

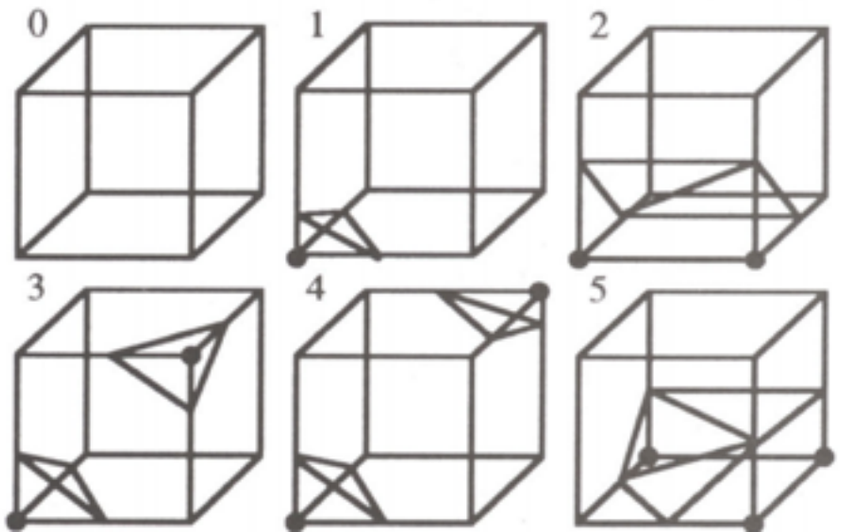
Abstract

A method for computing isovalue or contour surfaces of a trivariate function is discussed. The input data are values of the trivariate function, F_{ijk}, at the cuberille grid points (x_i, y_j, z_k) and the output is a collection of triangles representing the surface consisting of all points where F(x, y, z) is a constant value. The method described here is a modification that is intended to correct a problem with a previous method.

1.0 Introduction

The purpose of this paper is to describe a method for computing contour or isovalue surfaces of a trivariate function F(x, y, z). It is assumed that the function is continuous and that samples over a cuberille grid (see Figure 1) are available. These values are denoted by $F_{ijk} = F(x_i, y_j, z_k)$; i = 1, ... , Nx, j = 1, ... , Ny, k = 1, ... , Nz. The problem is to compute the isovalue or contour surface

$$S_\alpha = \{ (x, y , z) : F(x, y, z) = \alpha\}.$$



marked indicates $F_{ijk} > \alpha$. While there are $2^8 = 256$ possible configurations, there are only 15 shown in Figure 2. This is because some configurations are equivalent with respect to certain operations. First off, the number can be reduced to 128 by assuming two configurations are equivalent if marked grid points and unmarked grid points are switched. This means that we only have to consider cases where there are four or fewer marked grid points. Further reduction to the 15 cases shown is possible by equivalence due to rotations.

$$B(s,t) = (1\text{-}s,\ s)\begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix}\begin{pmatrix} 1\text{-}t \\ t \end{pmatrix}$$

where B_{00}, B_{01}, B_{10} and B_{11} represent the appropriate values of F_{ijk} at the four corner grid points (see Figure 5).

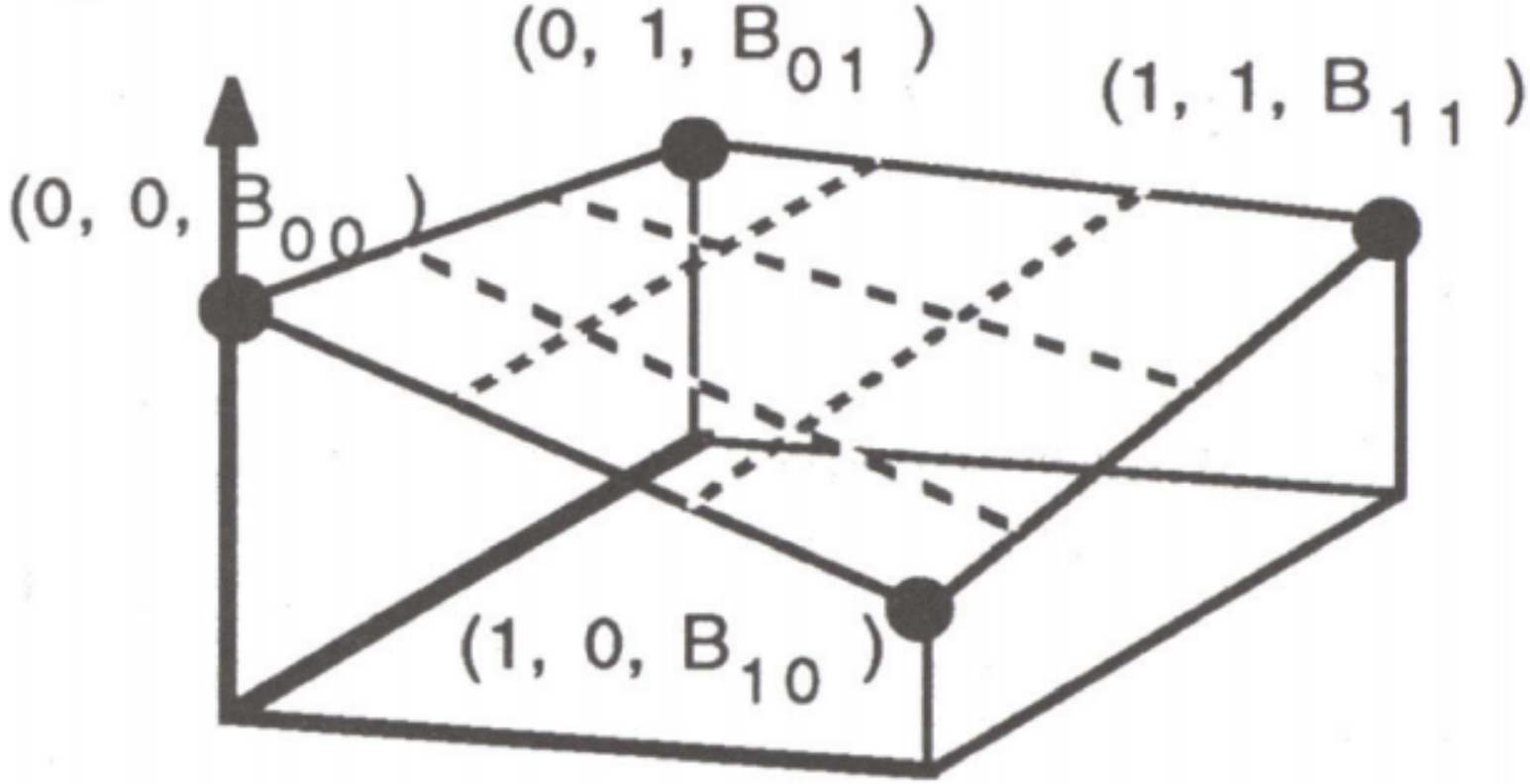
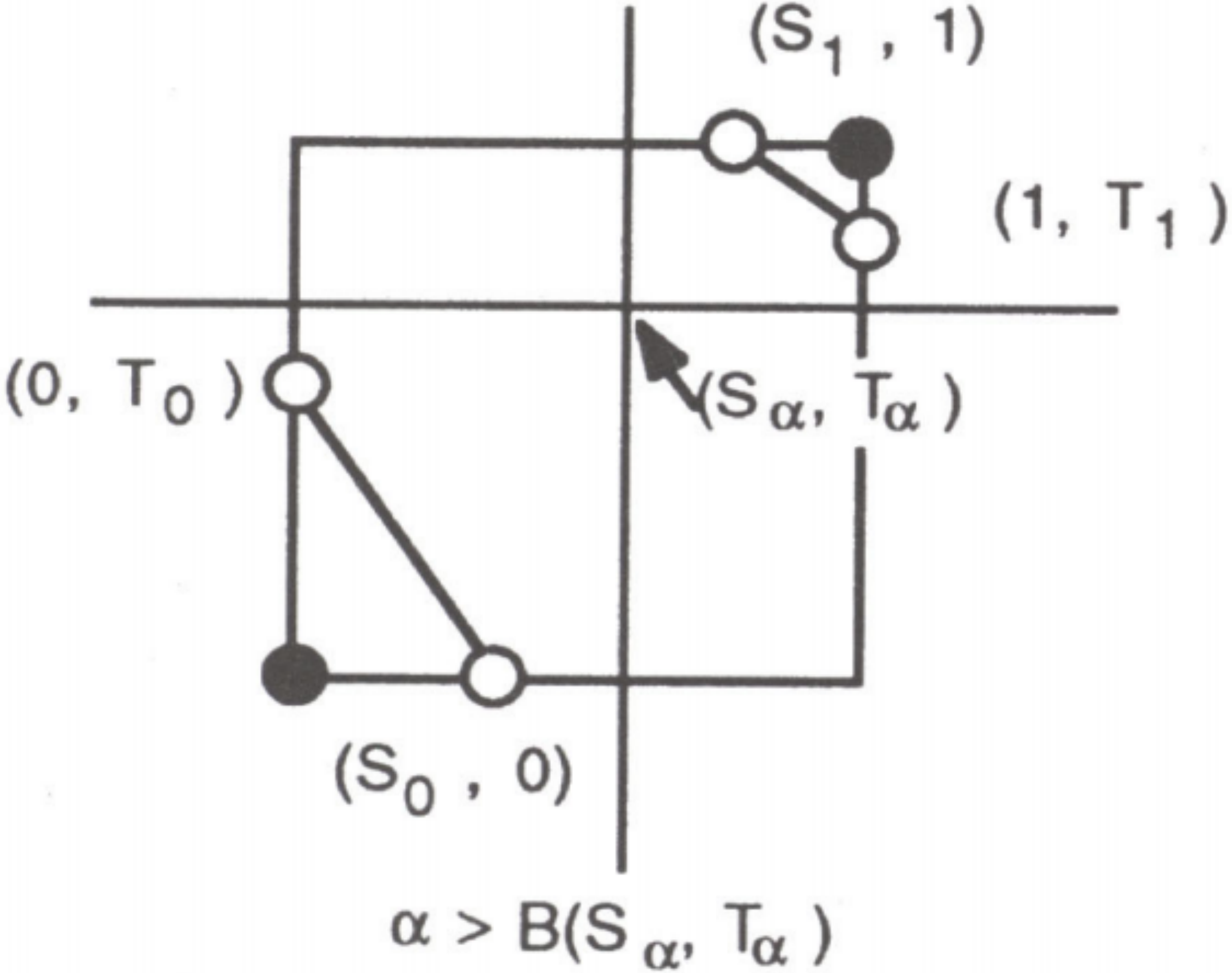
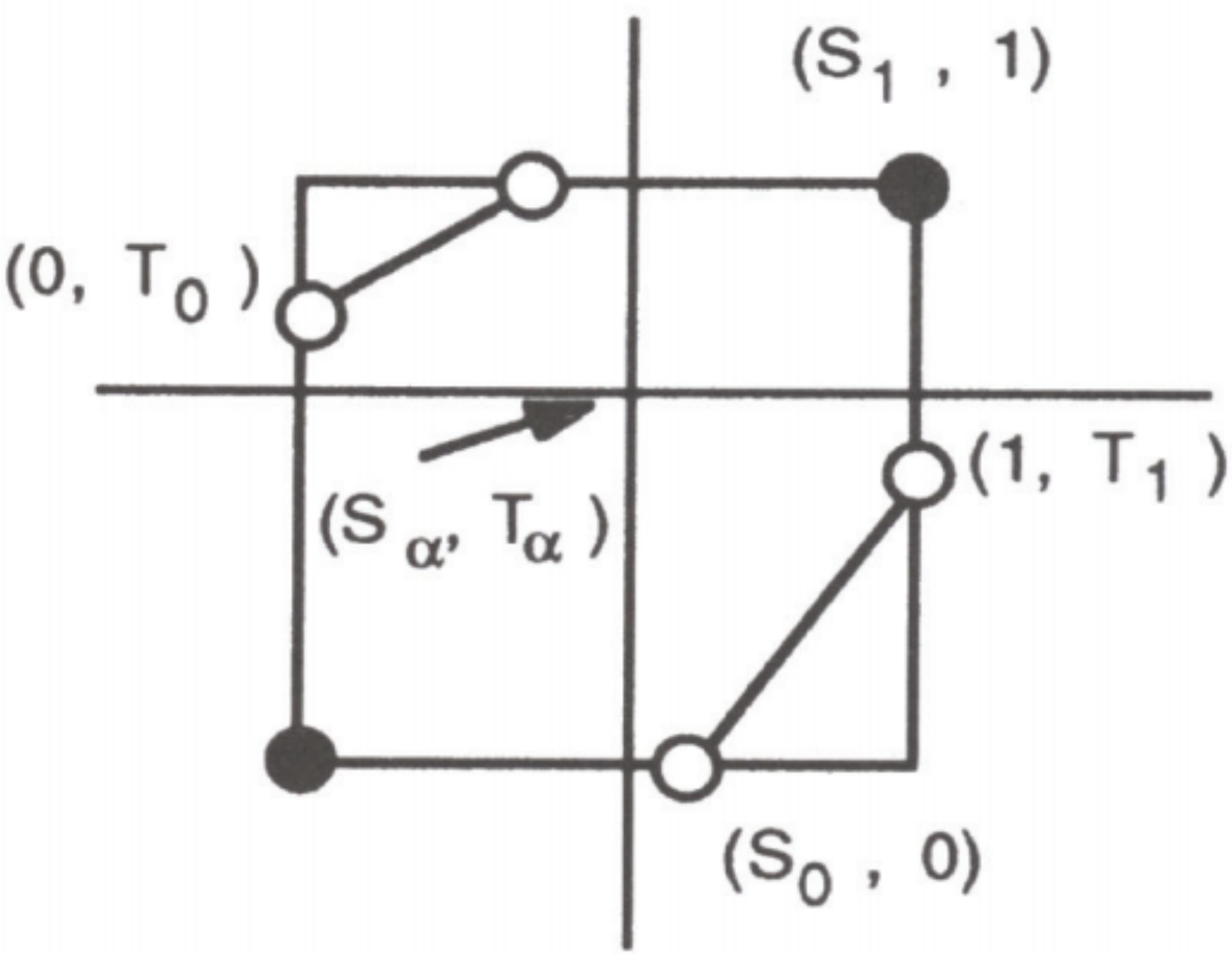


Figure 5. Bilinear interpolation.

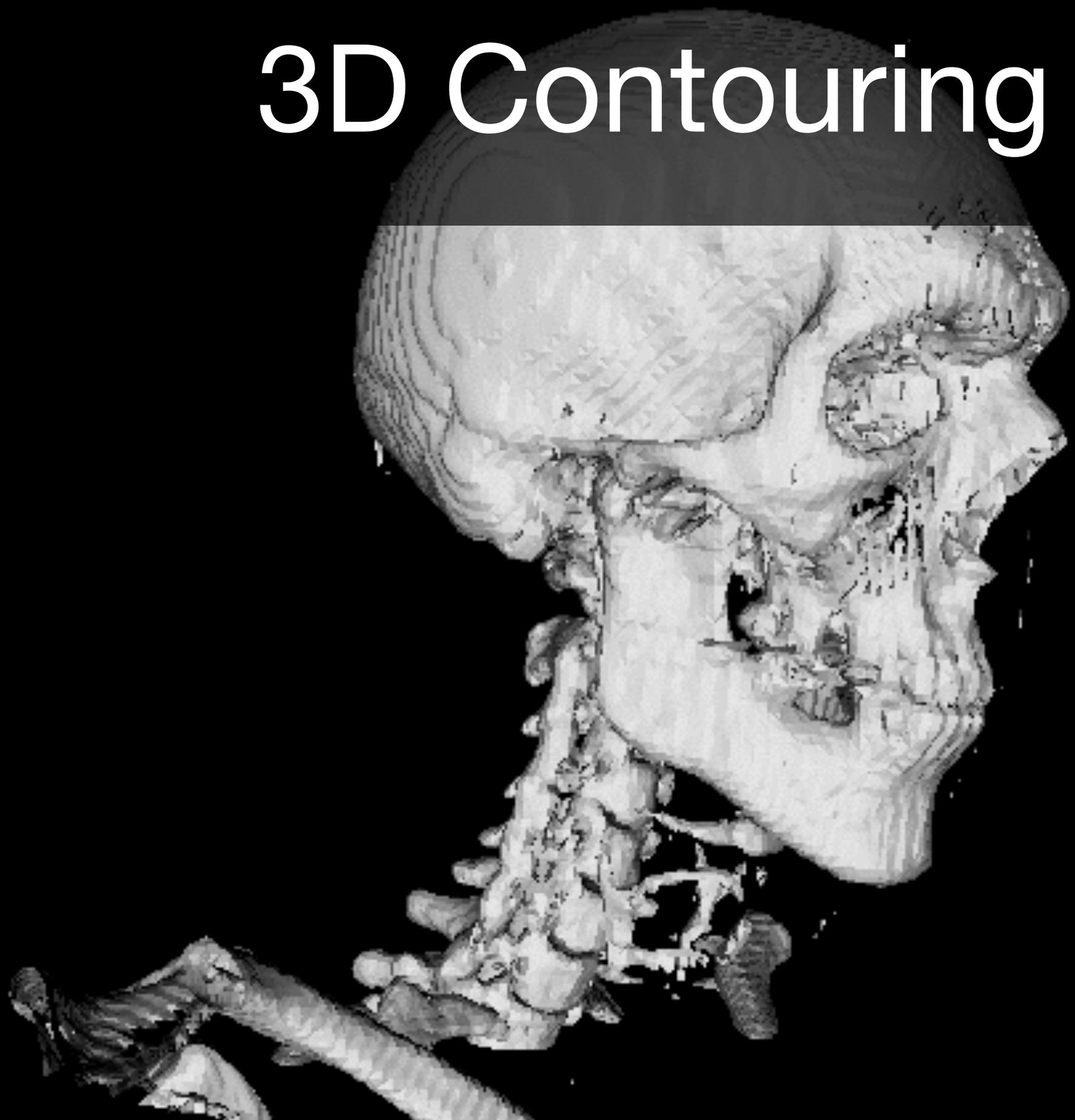
It is easy to verify that the contour curves of B, $\{(s, t) : B(s, t) = \alpha\}$, are hyperbolas. Some


$(S_1, 1)$
$(1, T_1)$
$(0, T_0)$
(S_α, T_α)
$(S_0, 0)$
$\alpha > B(S_\alpha, T_\alpha)$

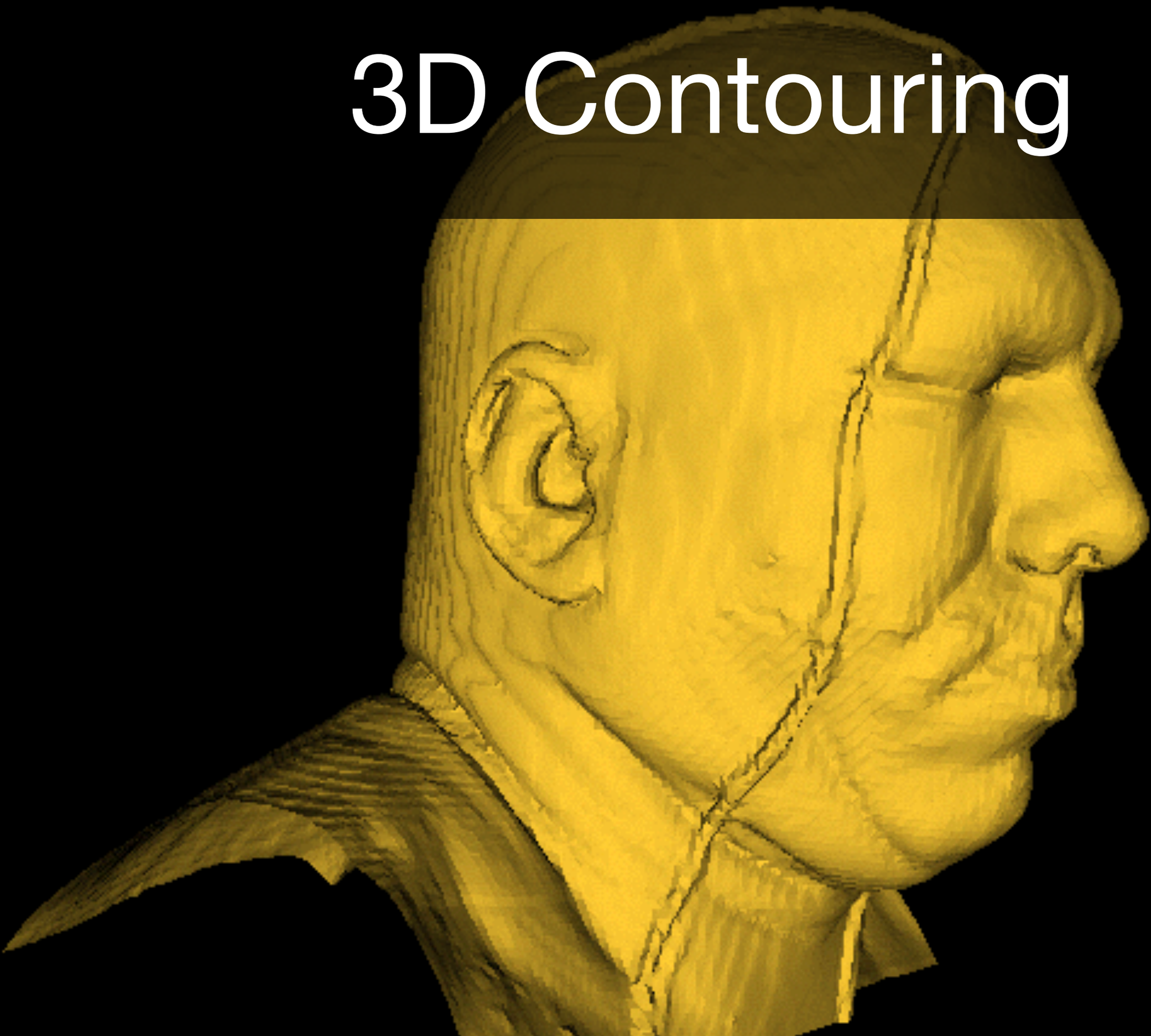


$$\alpha \leq B(S_\alpha, T\alpha)$$

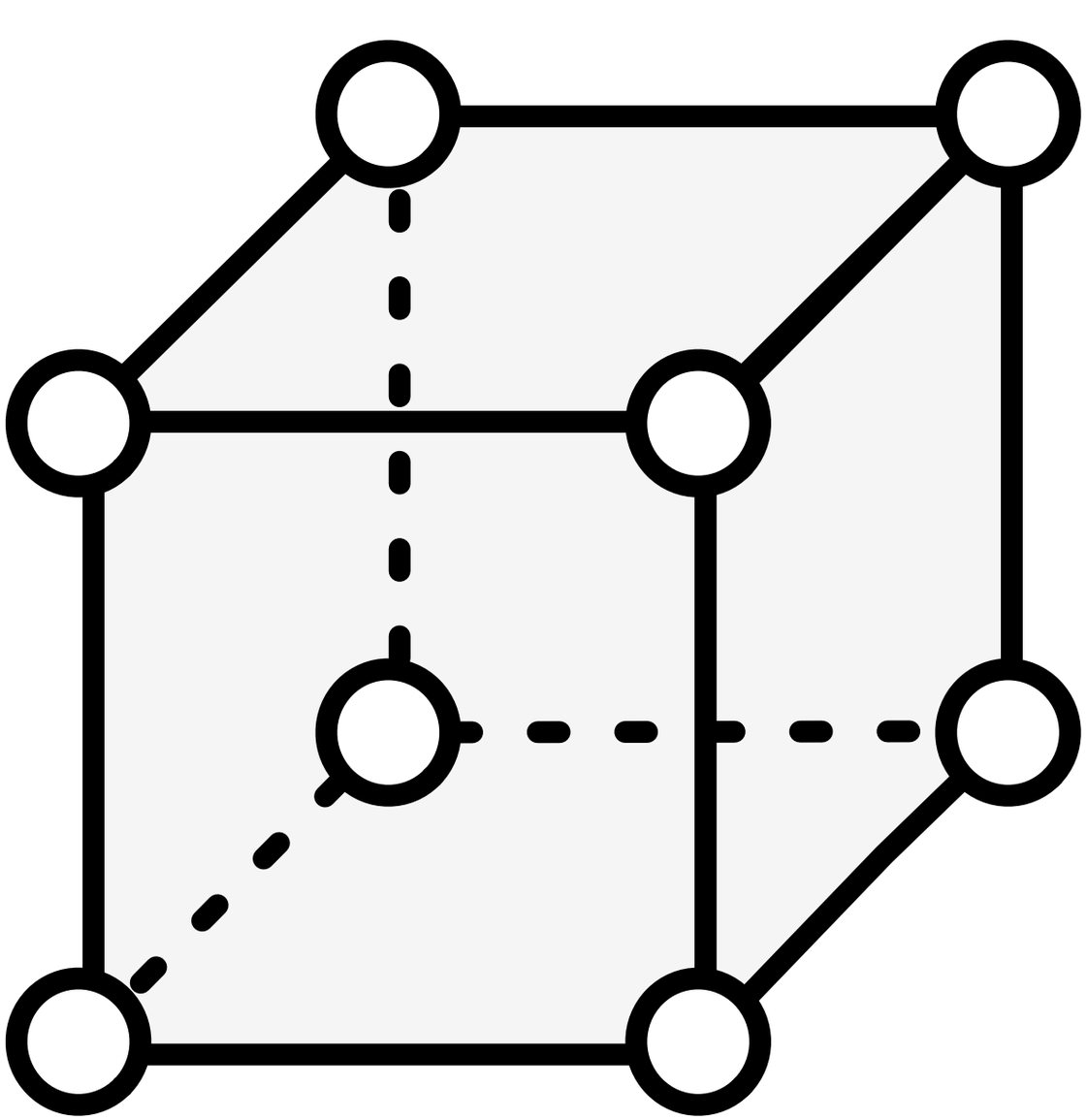
3D Contouring

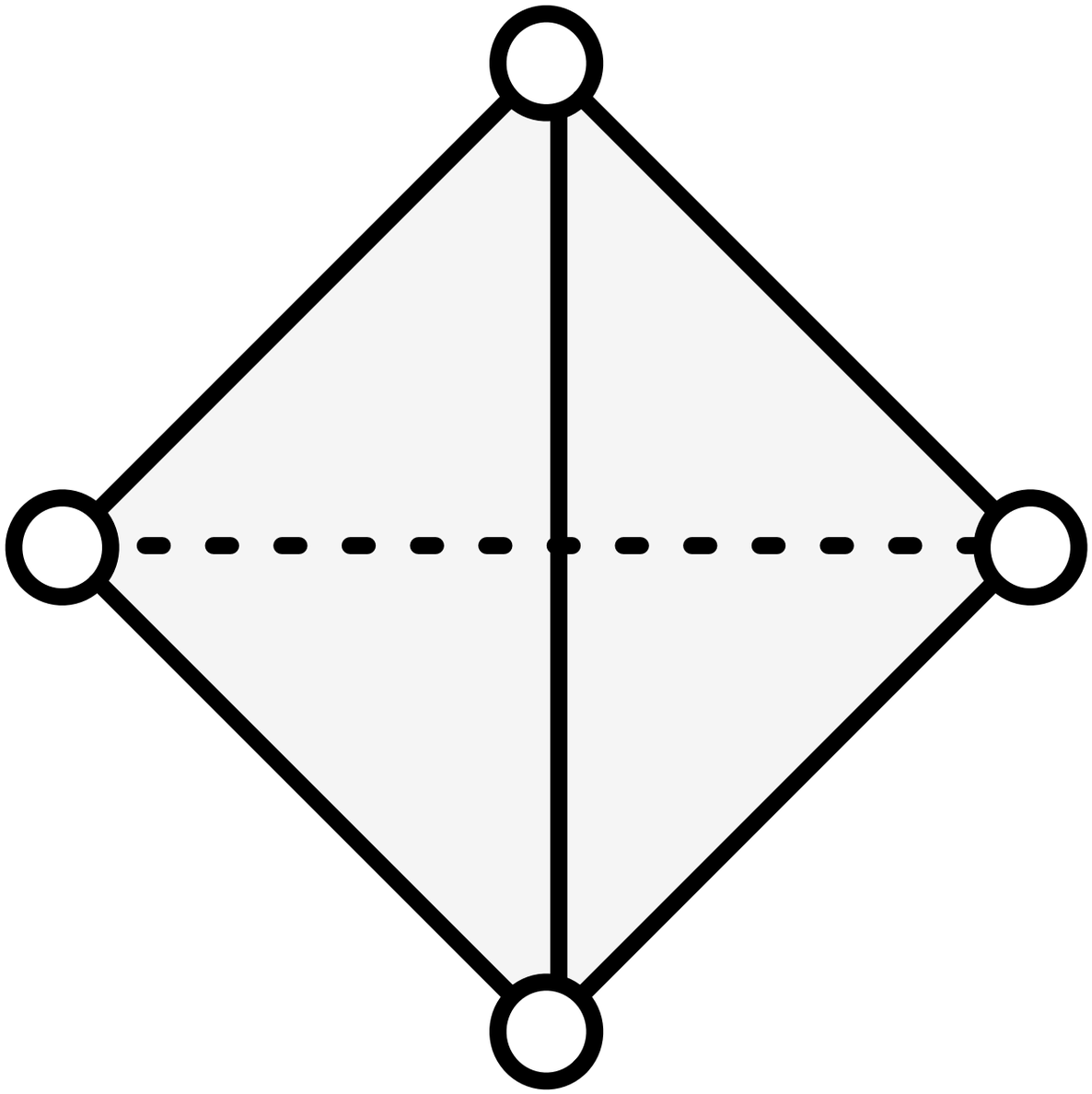


3D Contouring

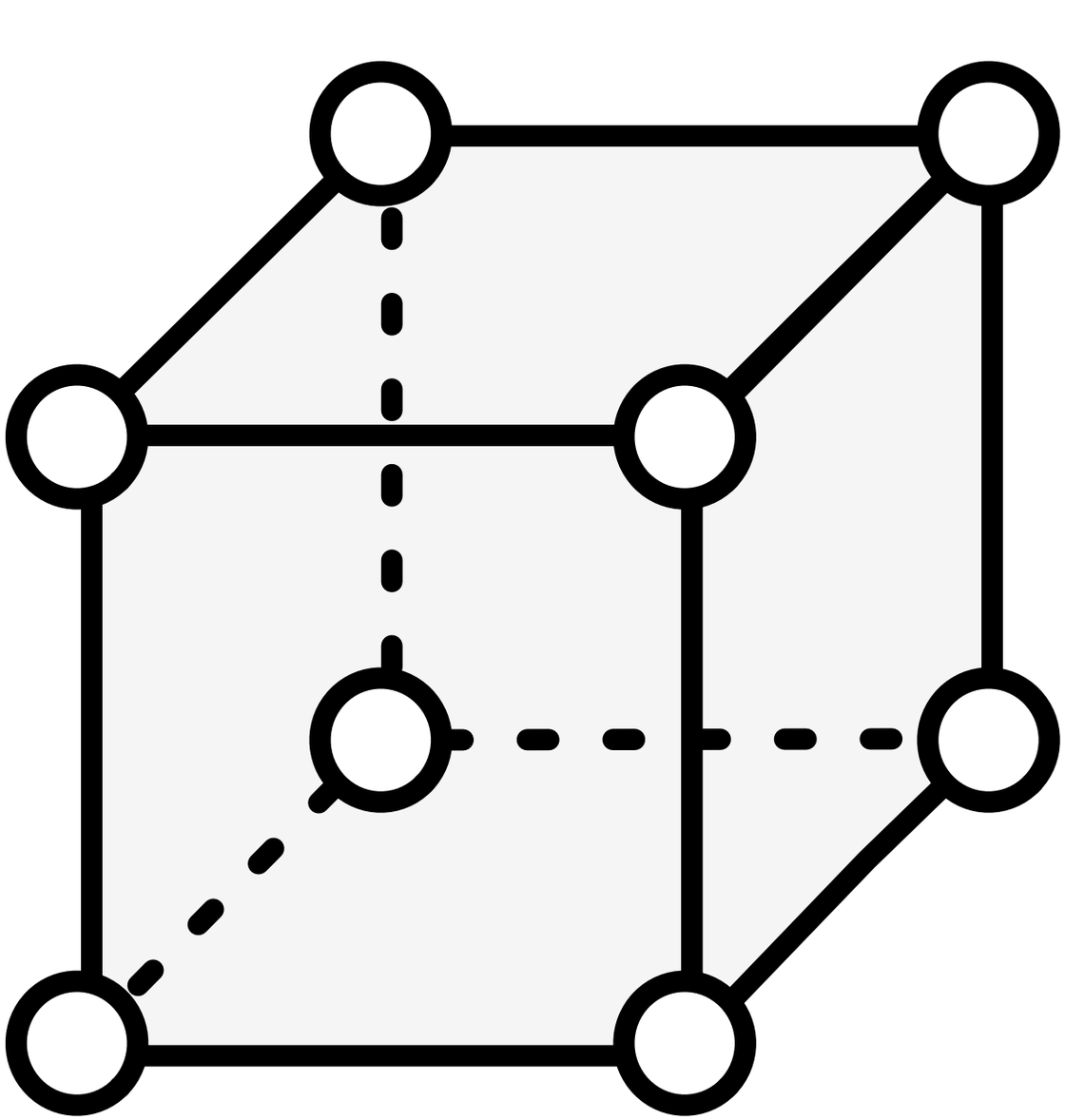


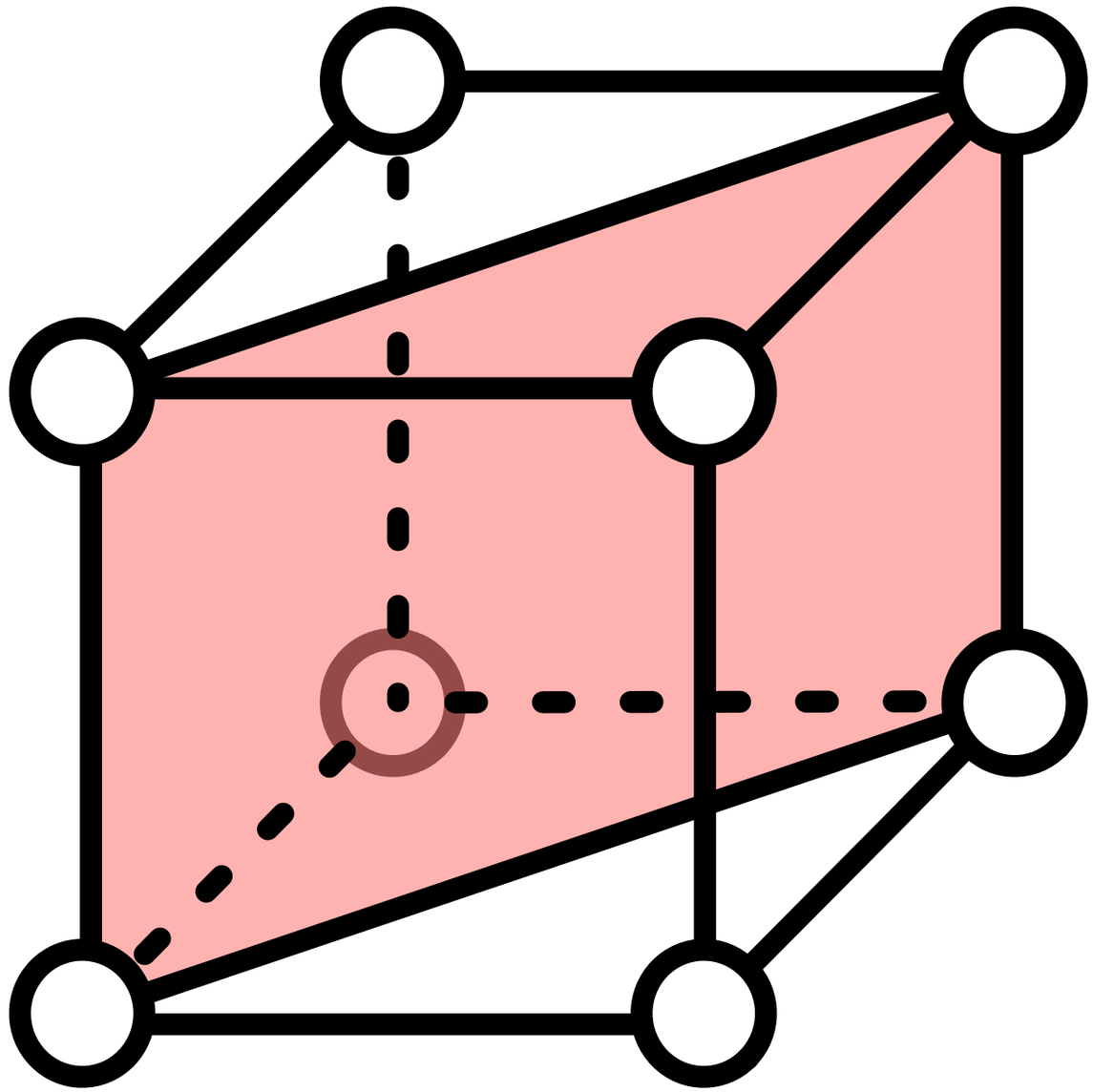
Splitting 3D space into simple shapes



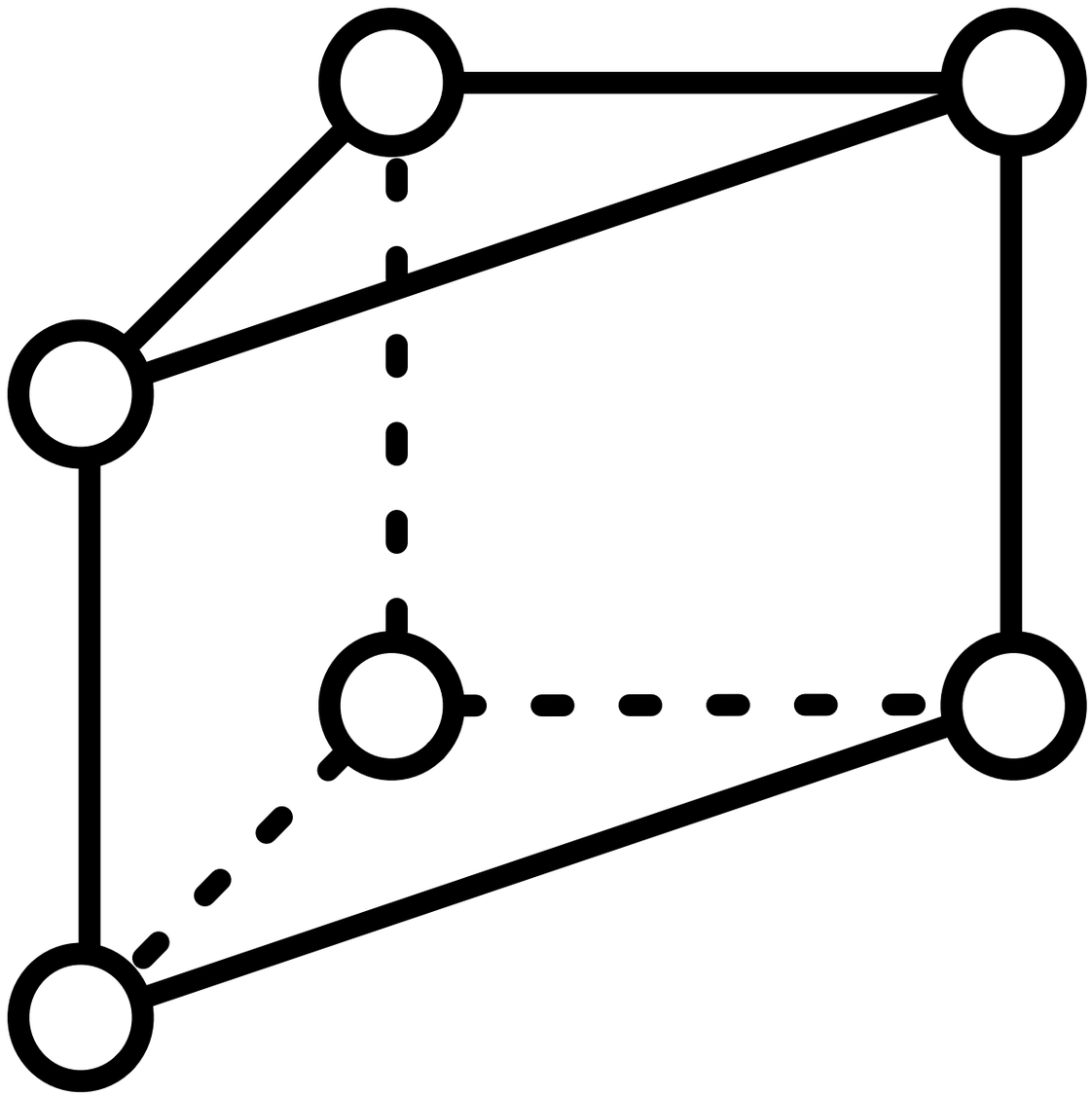


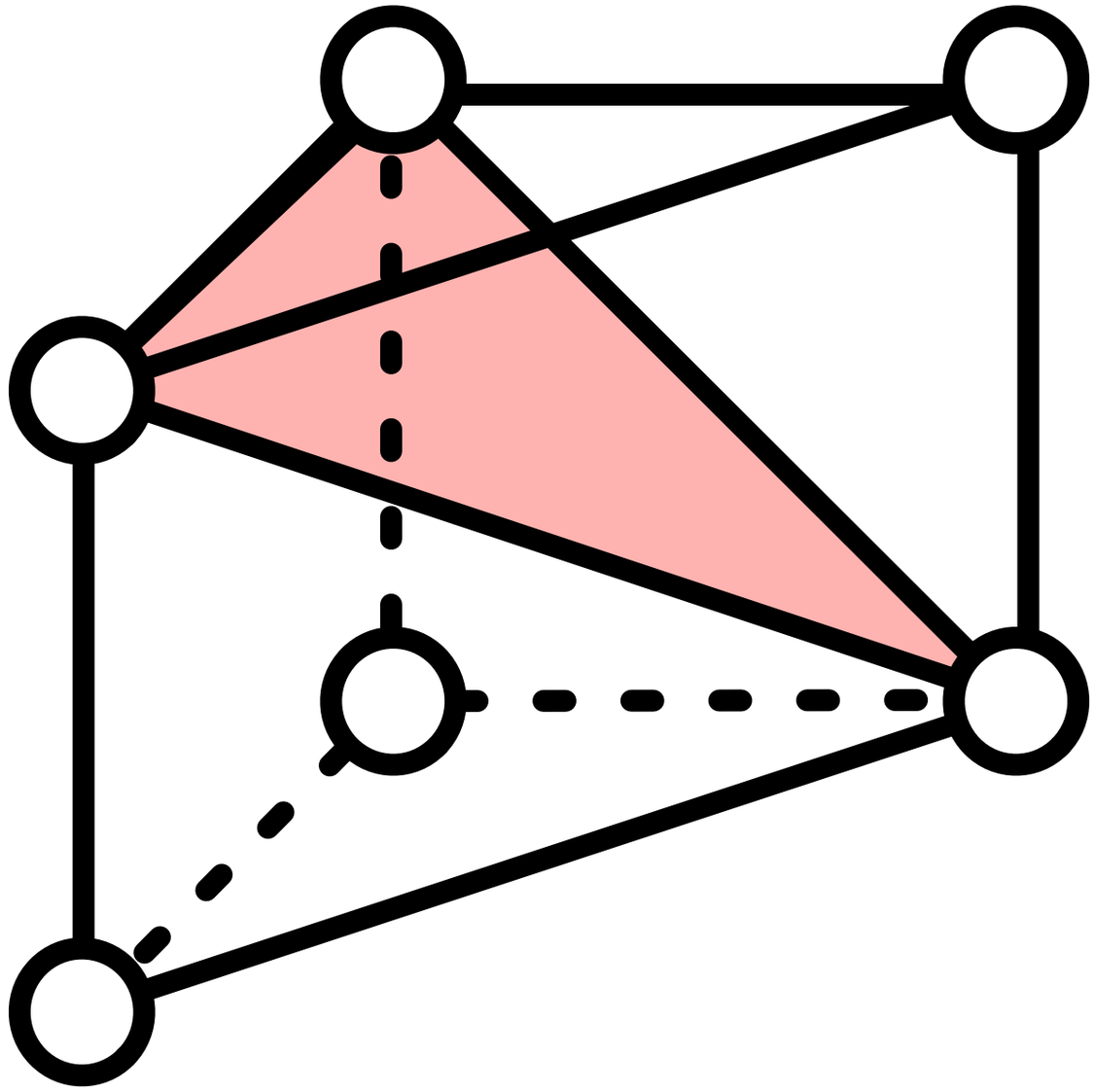
Cube into tetrahedra



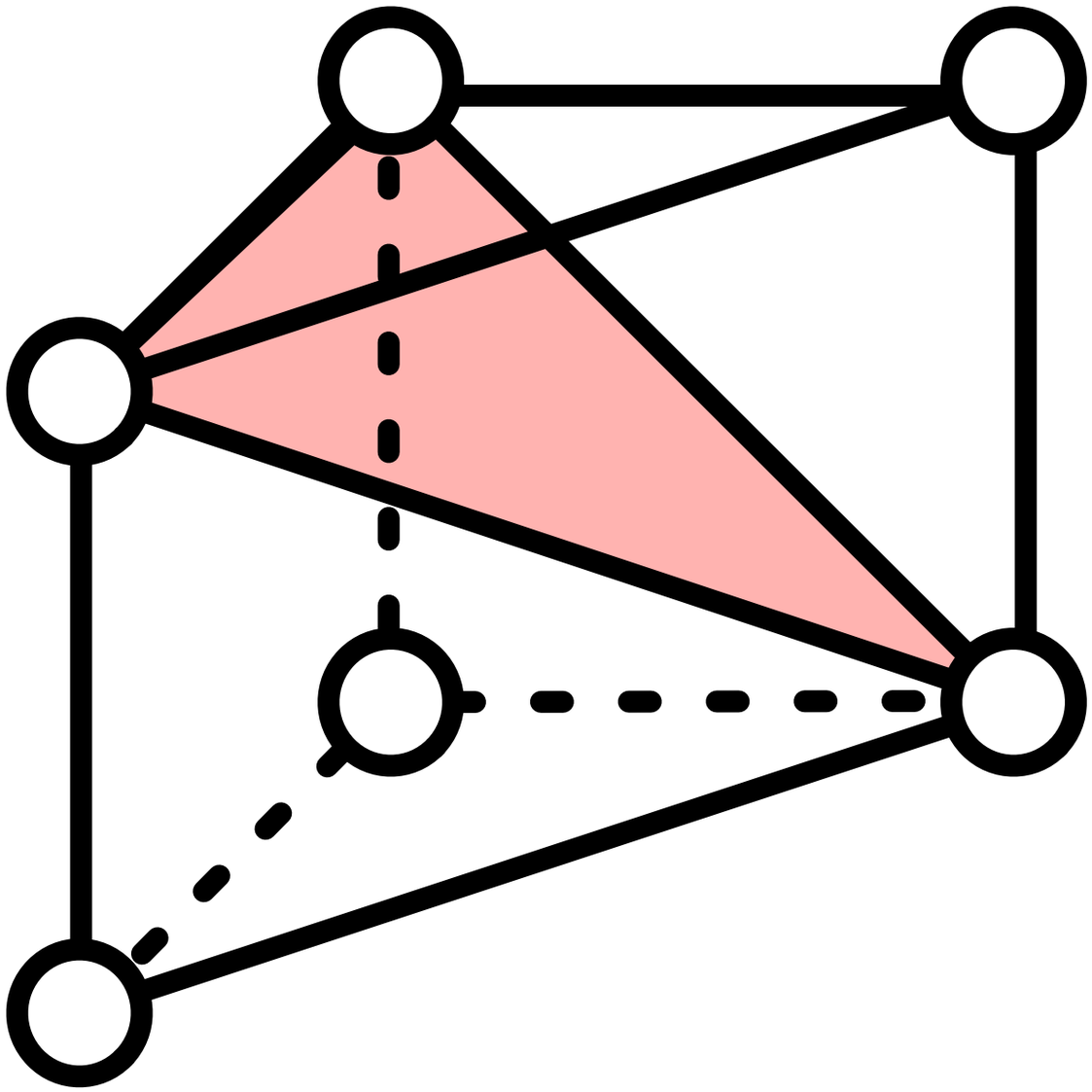


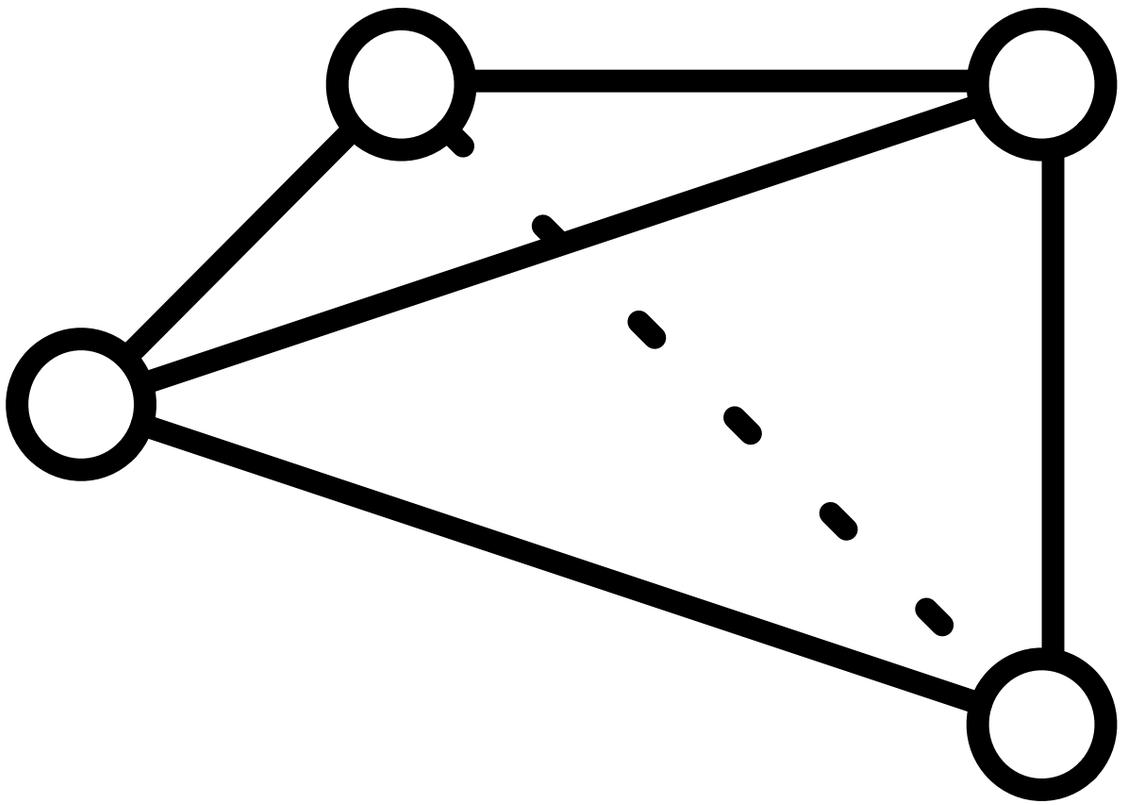
Cube into tetrahedra





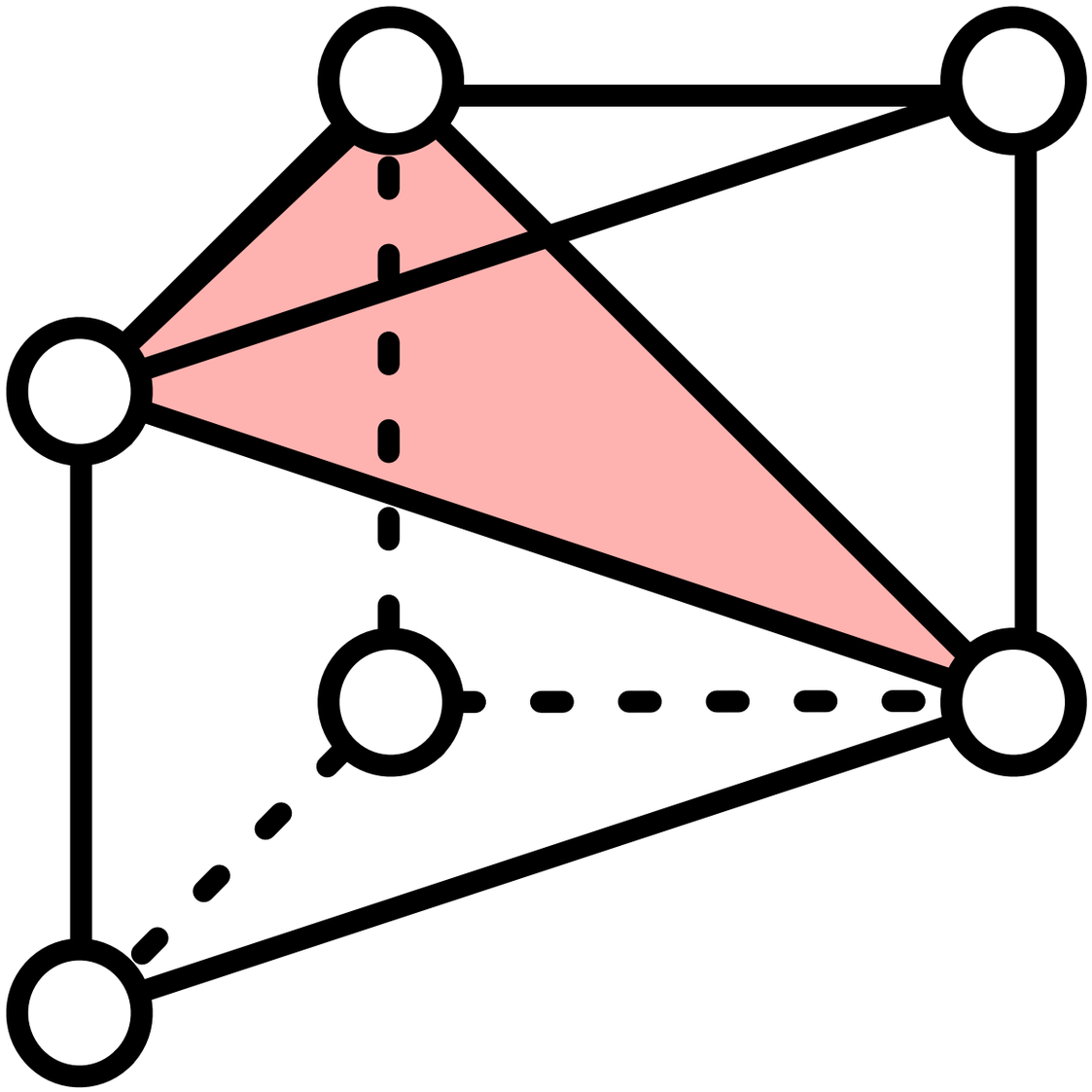
Cube into tetrahedra

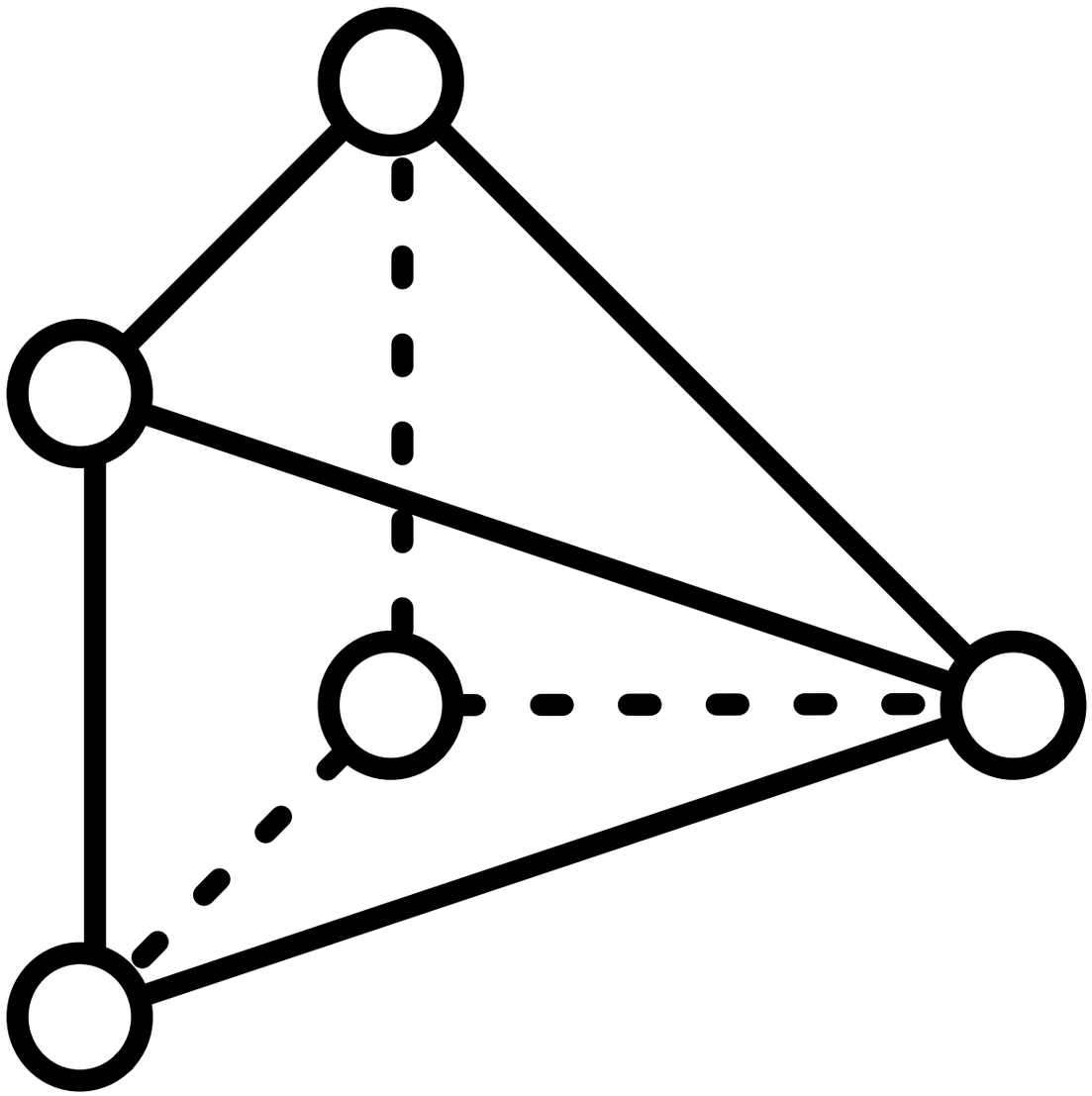




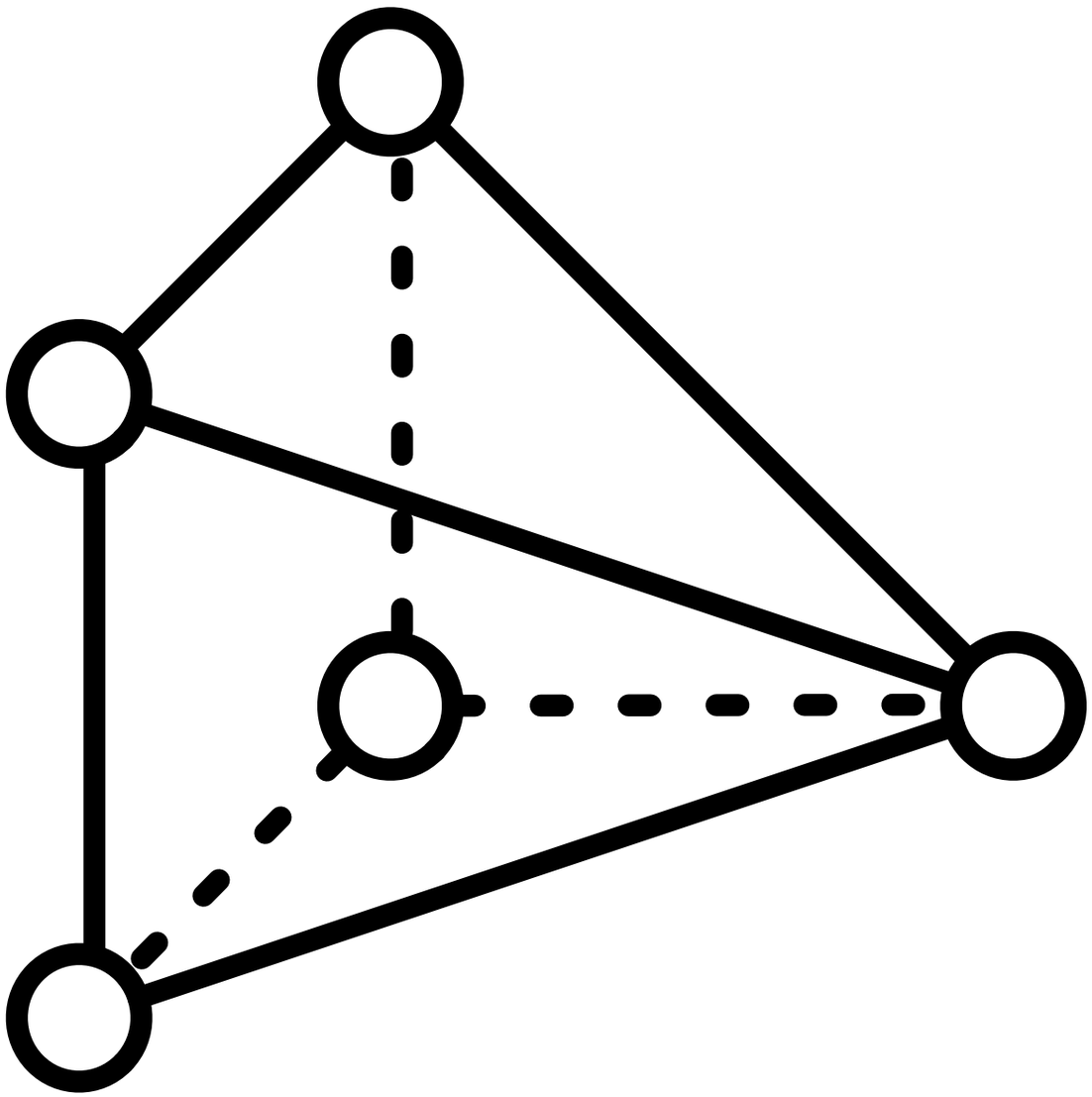
1 tetrahedron,

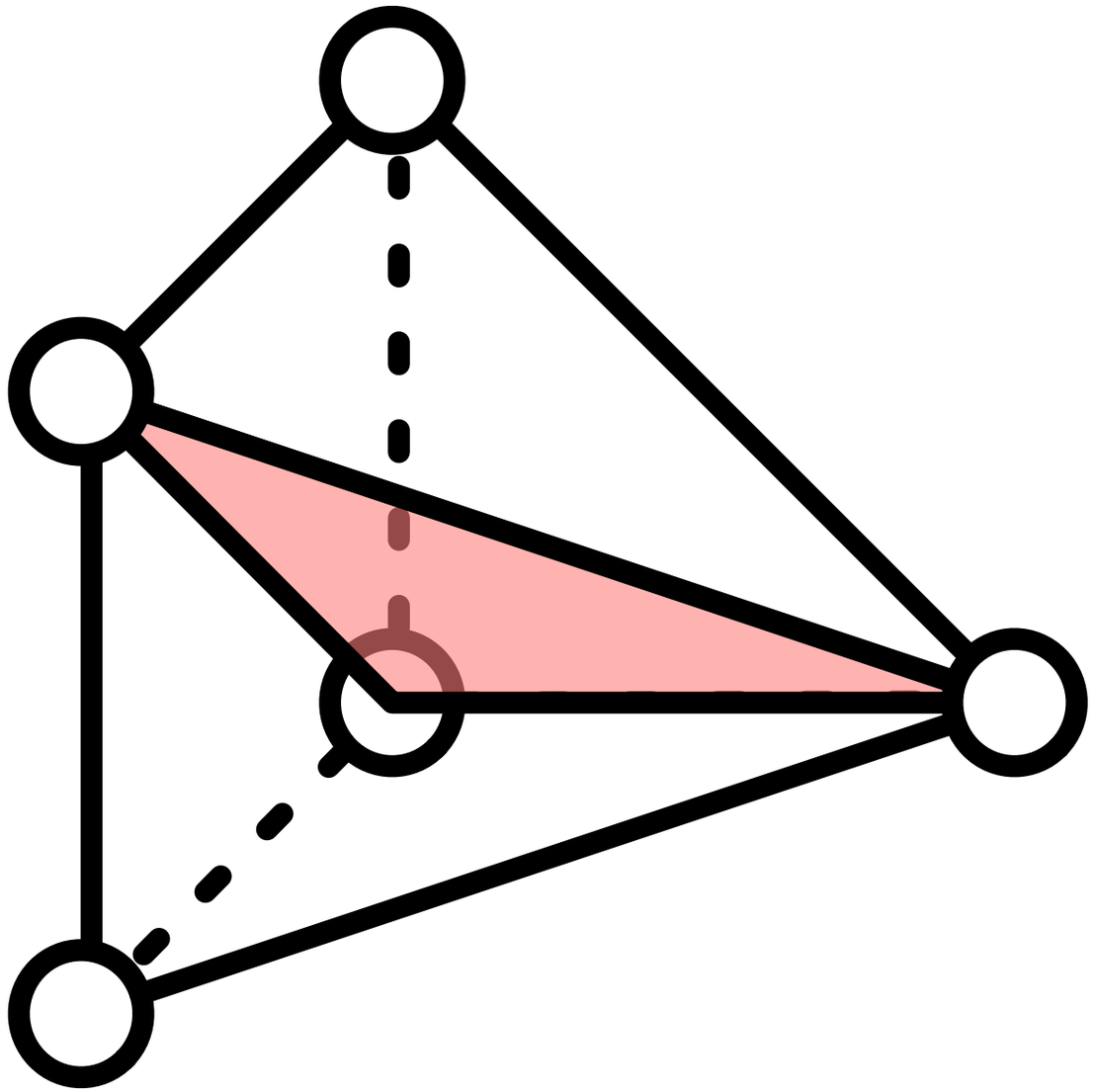
Cube into tetrahedra





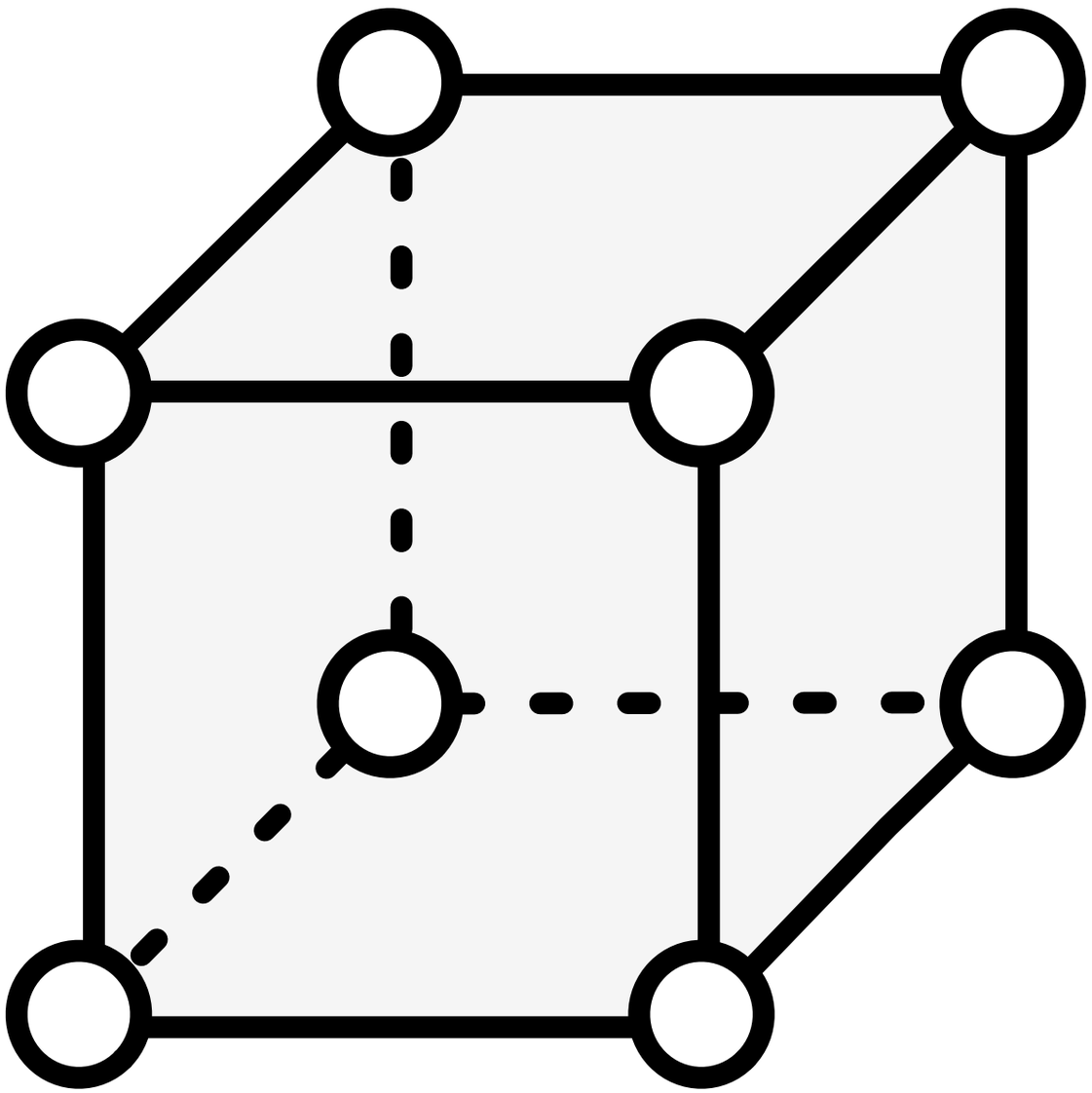
Cube into tetrahedra





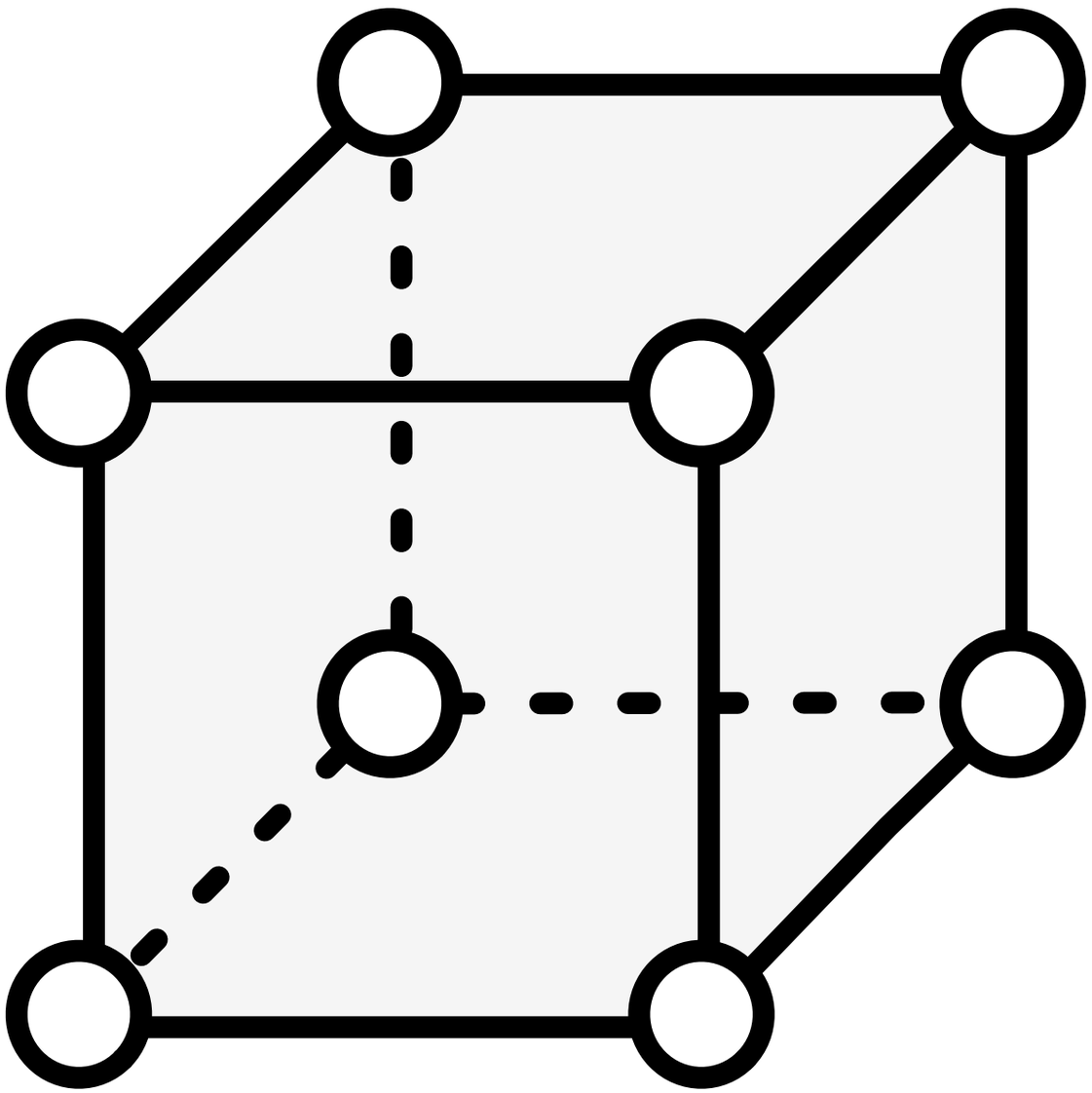
2 tetrahedra

Cube into tetrahedra



1 cube splits into
6 tetrahedra

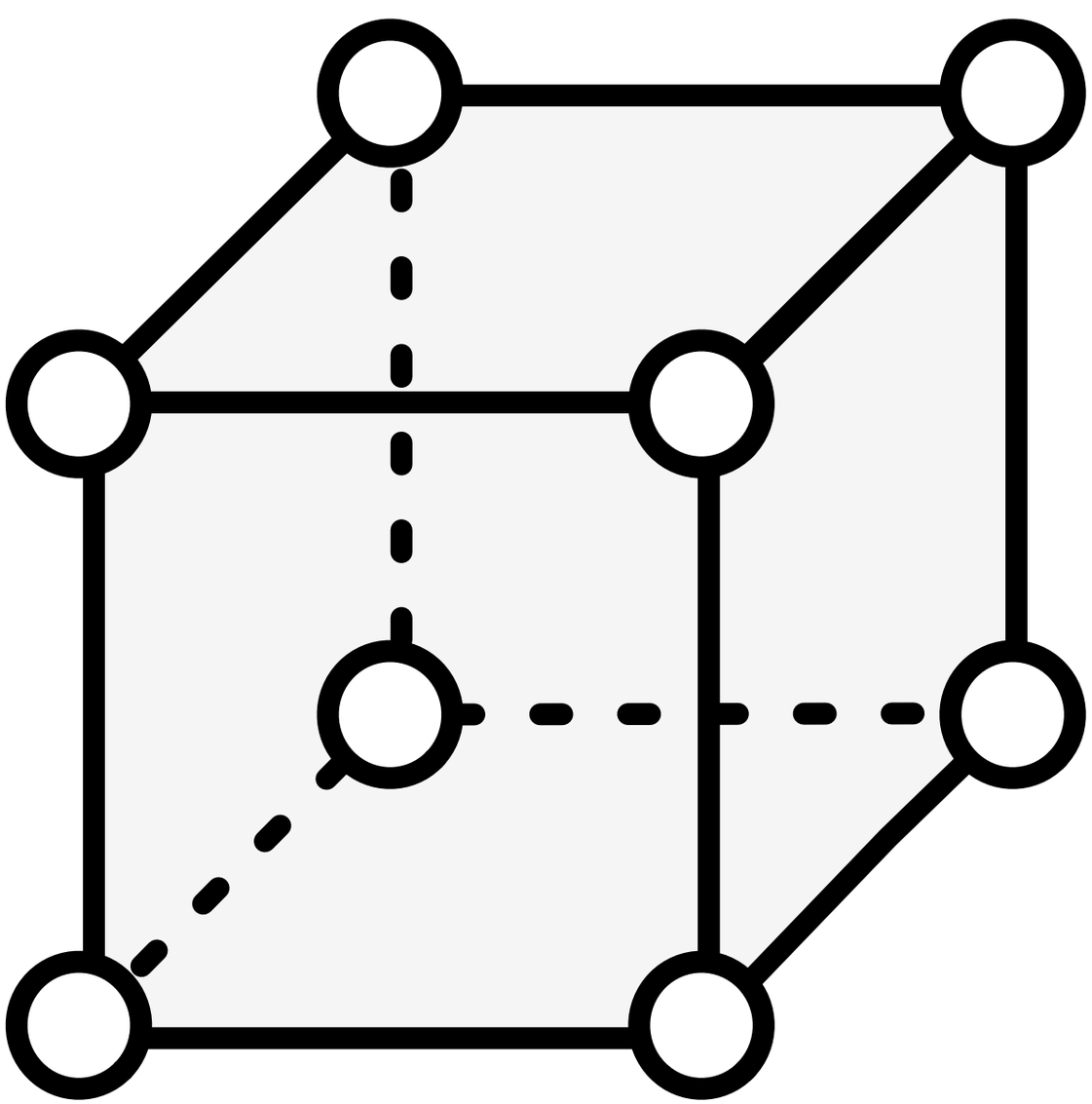
Cube into tetrahedra

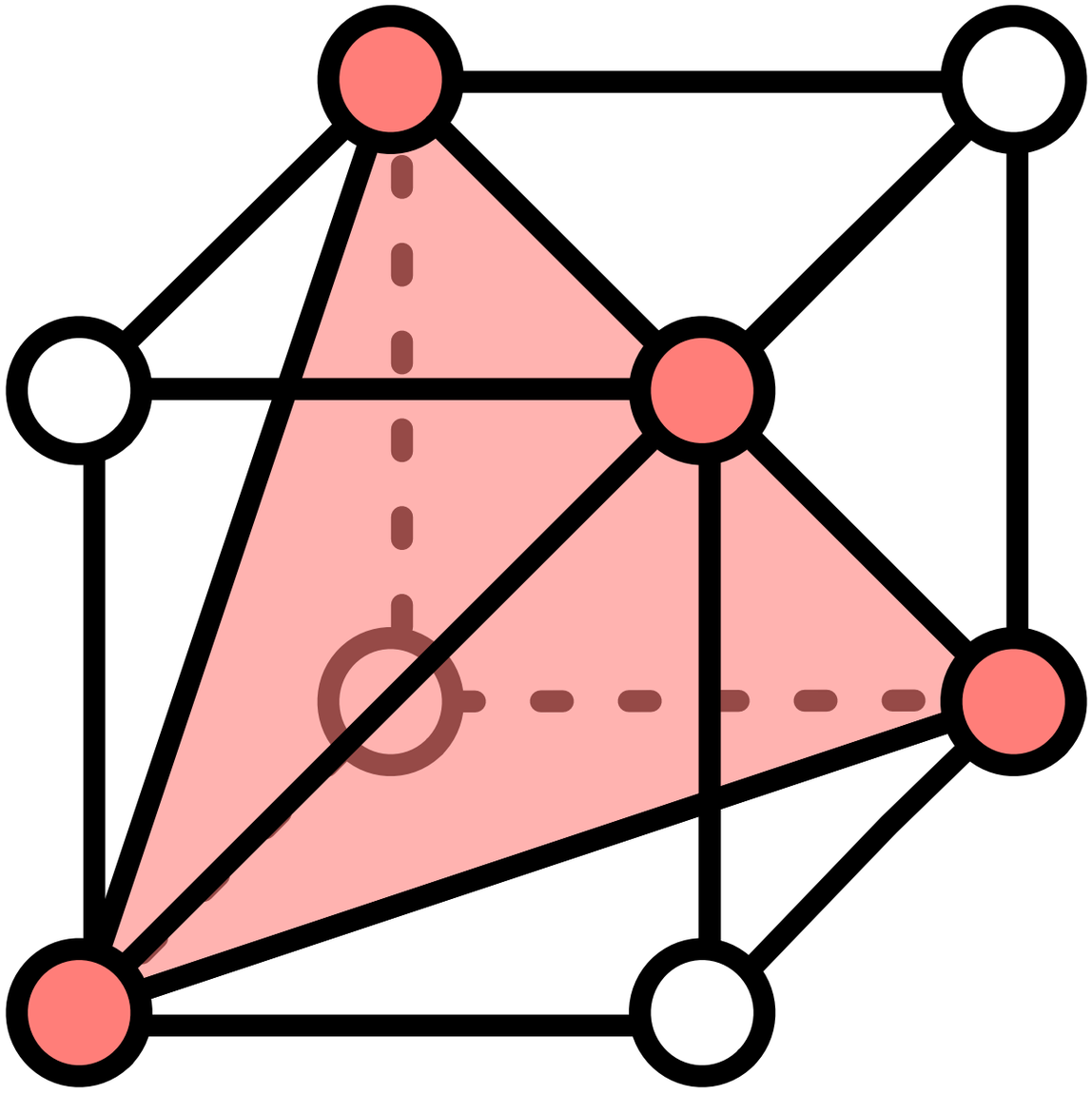


1 cube splits into
6 tetrahedra…

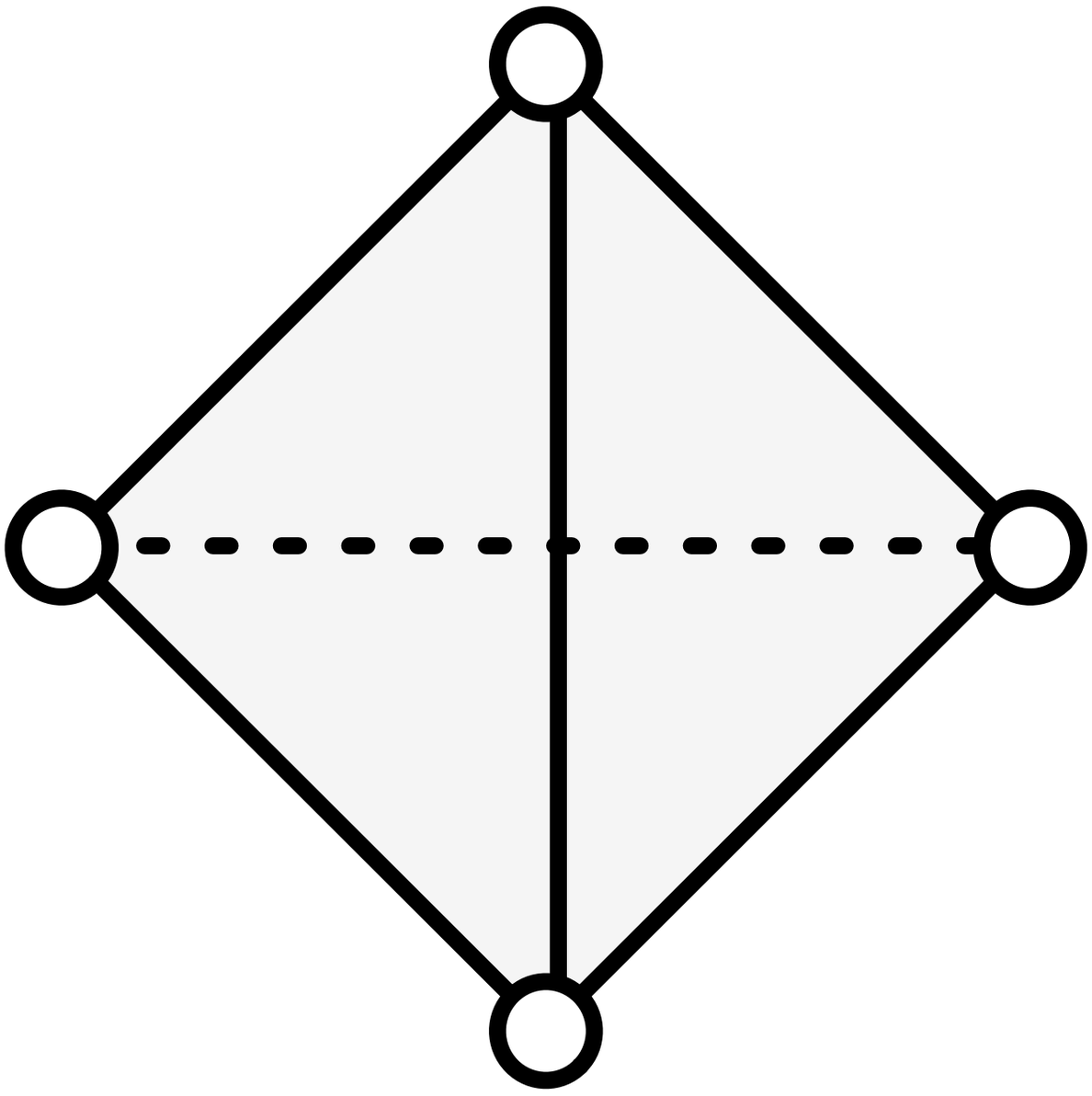
but also into 5 tetrahedra!

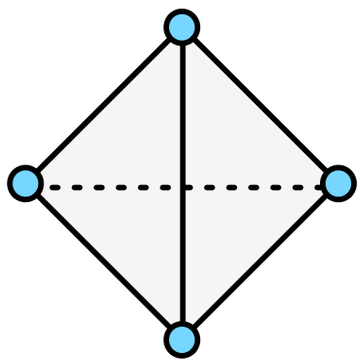
Cube into tetrahedra

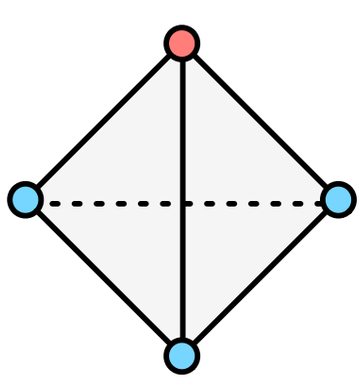


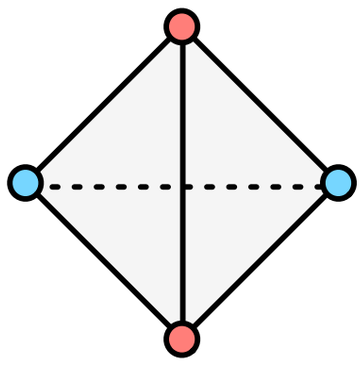


Marching Tetrahedra



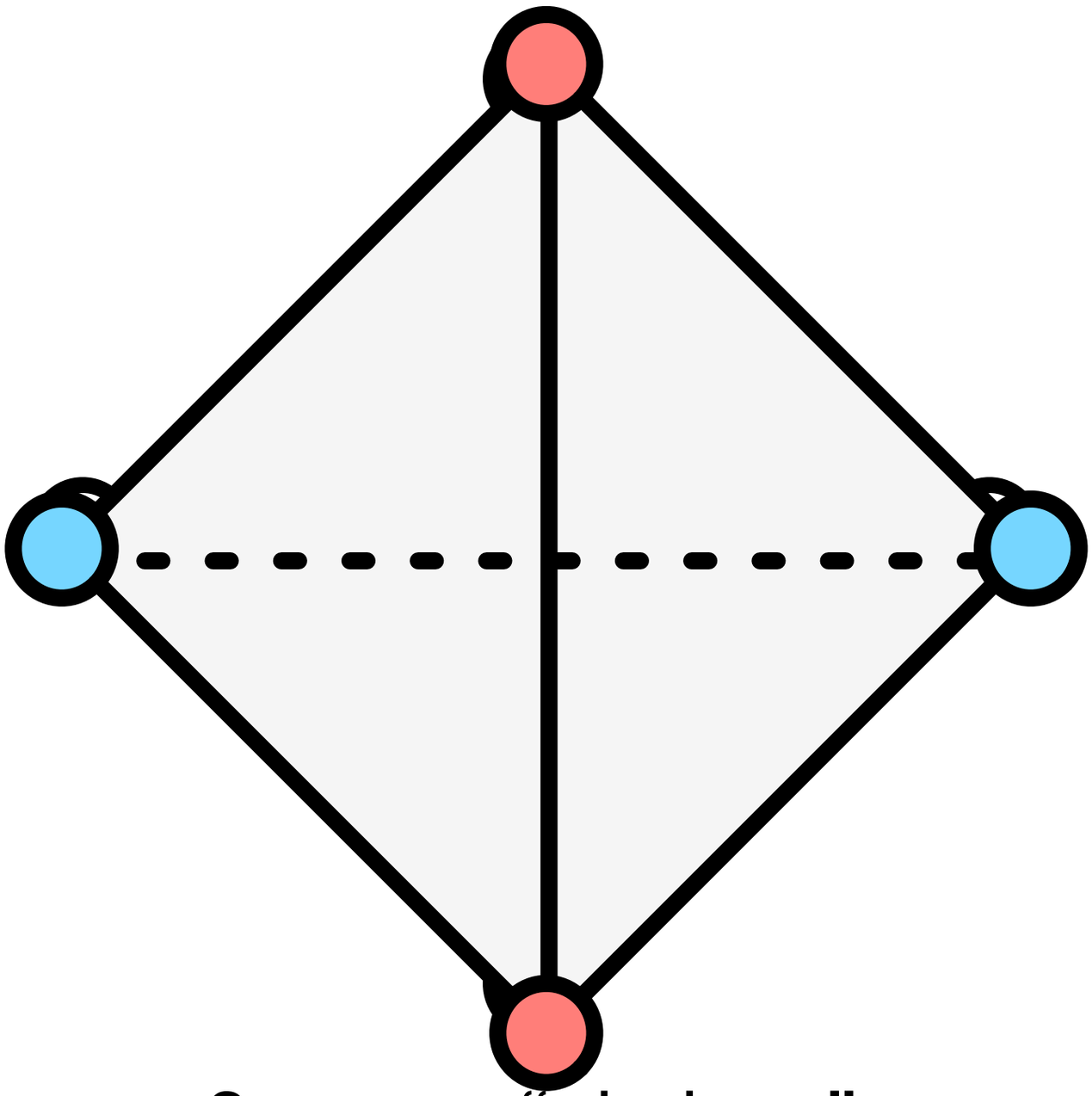


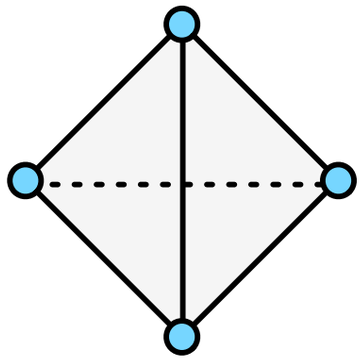


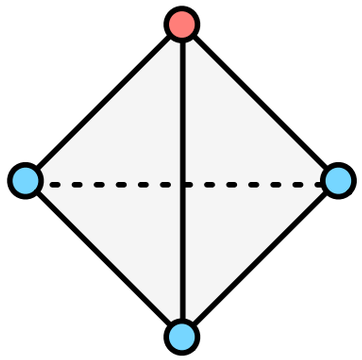


3 cases, “obvious”

Marching Tetrahedra

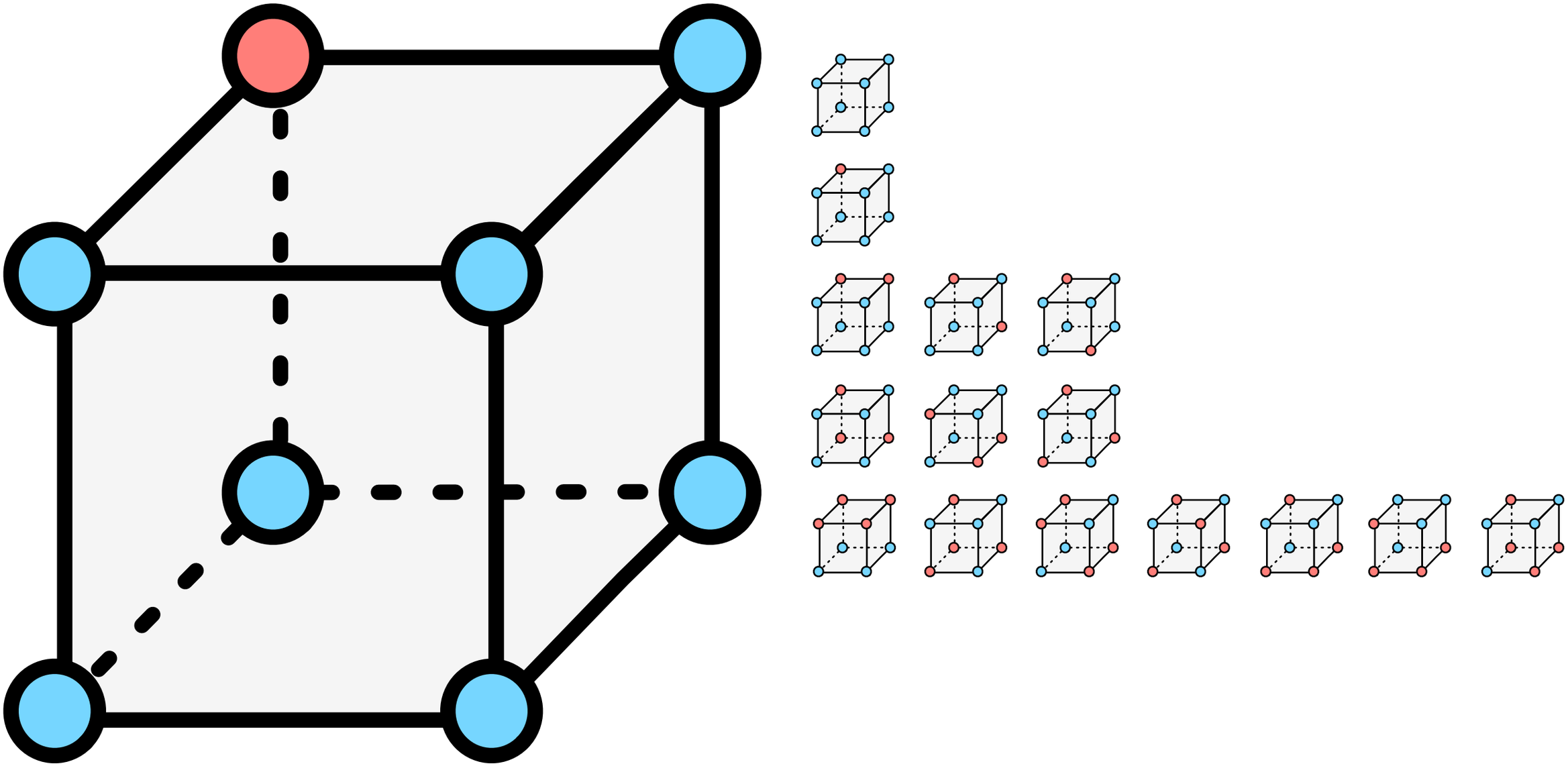






3 cases, “obvious”

3D Contouring



3D Contouring

Computer Graphics, Volume 21, Number 4, July 1987

MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM

William E. Lorensen
Harvey E. Cline

General Electric Company
Corporate Research and Development
Schenectady, New York 12301

Abstract

We present a new algorithm, called *marching cubes*, that creates triangle models of constant density surfaces from 3D medical data. Using a divide-and-conquer approach to generate inter-slice connectivity, we create a case table that defines triangle topology. The algorithm processes the 3D medical data in scan-line order and calculates triangle vertices using linear interpolation. We find the gradient of the origi-

acetabular fractures [6], craniofacial abnormalities [17,18], and intracranial structure [13] illustrate 3D's potential for the study of complex bone structures. Applications in radiation therapy [27,11] and surgical planning [4,5,31] show interactive 3D techniques combined with 3D surface images. Cardiac applications include artery visualization [2,16] and non-graphic modeling applications to calculate surface area and volume [21].

3D Contouring