Introduction to Level Sets for Image Analysis

James McClure

Overview

- Level set representation of surfaces
- Data representation for porous media
- Marching cubes (MC) algorithm
- Porous medium marching cubes algorithm (PMMC)
- Higher order PMMC (HOPMMC)
- Interfacial curvatures
- Examples: Lattice Boltzmann, X-ray tomography

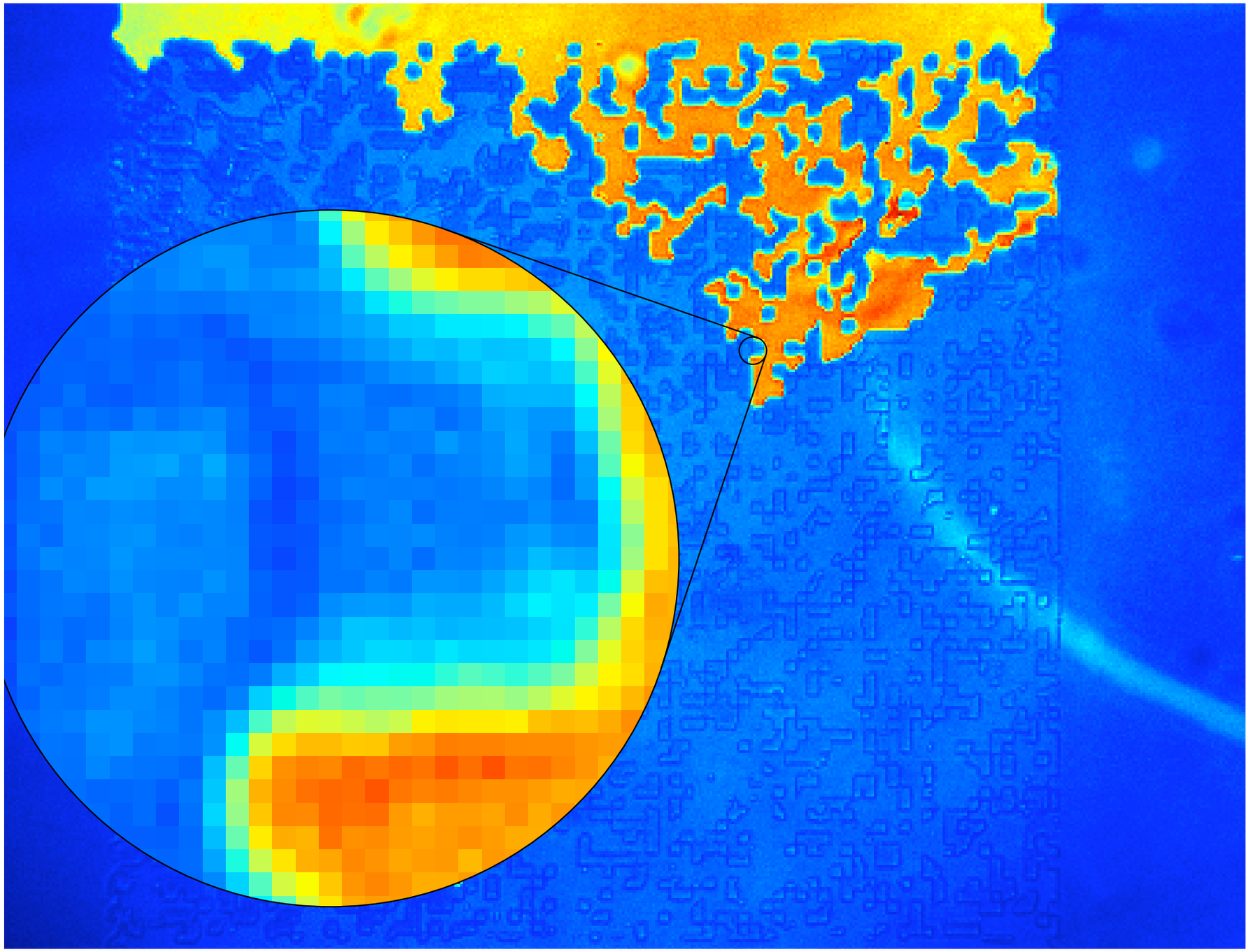
What is a Level Set?

Suppose that you have some continuous function $G(\mathbf{x})$ defined $\forall \mathbf{x} \in \Omega$

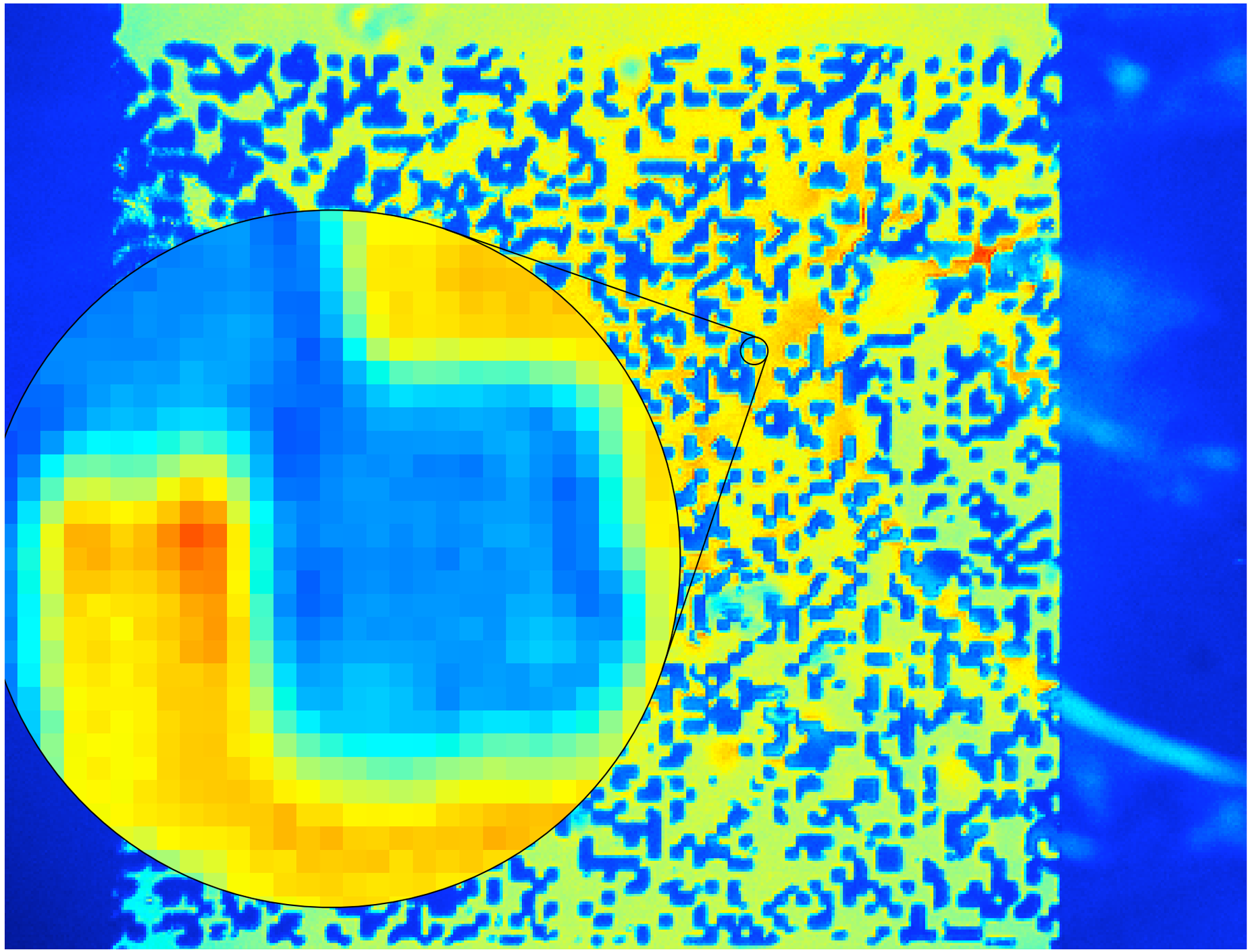
If we set this function to be equal to a constant value ν, $G(\mathbf{x}) = \nu$ defines an "level set" or isosurface.

If we are able to determine $G(\mathbf{x})$ and ν so that this isosurface corresponds to a surface which we are interested in, our job will be easy.

Data Sets



Data Sets



Data Sets

0.38	0.4	0.36	0.33	0.37	0.51	0.64	0.68	0.66	0.67
0.37	0.39	0.36	0.34	0.39	0.52	0.66	0.68	0.67	0.67
0.38	0.39	0.37	0.33	0.4	0.57	0.66	0.66	0.66	0.68
0.39	0.39	0.35	0.34	0.42	0.57	0.67	0.68	0.67	0.67
0.4	0.4	0.37	0.34	0.42	0.57	0.66	0.67	0.68	0.69
0.4	0.4	0.35	0.33	0.43	0.58	0.65	0.67	0.67	0.68
0.42	0.41	0.36	0.33	0.42	0.57	0.65	0.66	0.67	0.68
0.44	0.42	0.37	0.33	0.42	0.56	0.63	0.66	0.66	0.69

Goal

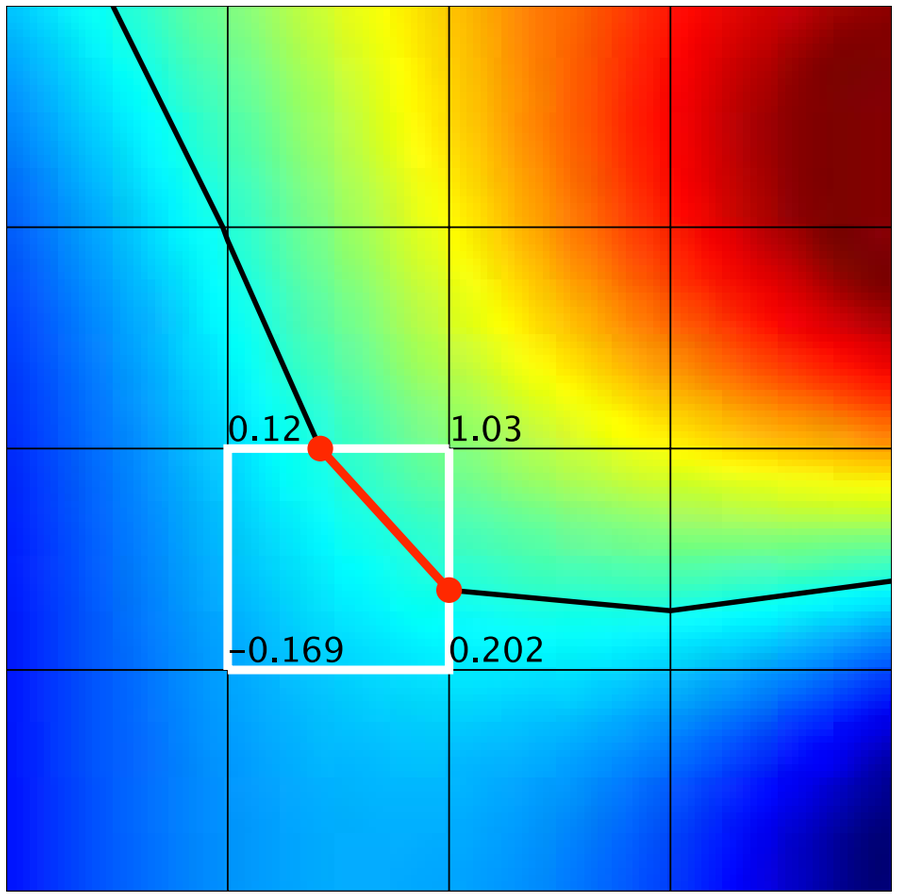
Use this information to determine the position of the phases, interface location, curvature of the interface, other types of information which are of interest

Marching Cubes (MC)

Compute isosurface $G(\mathbf{x}) = \nu$ from continuous data

Here we look at a two dimensional case:

- values of $G(\mathbf{x})$ are known at the corners of squares that cover the domain
- Move through the system one box at a time



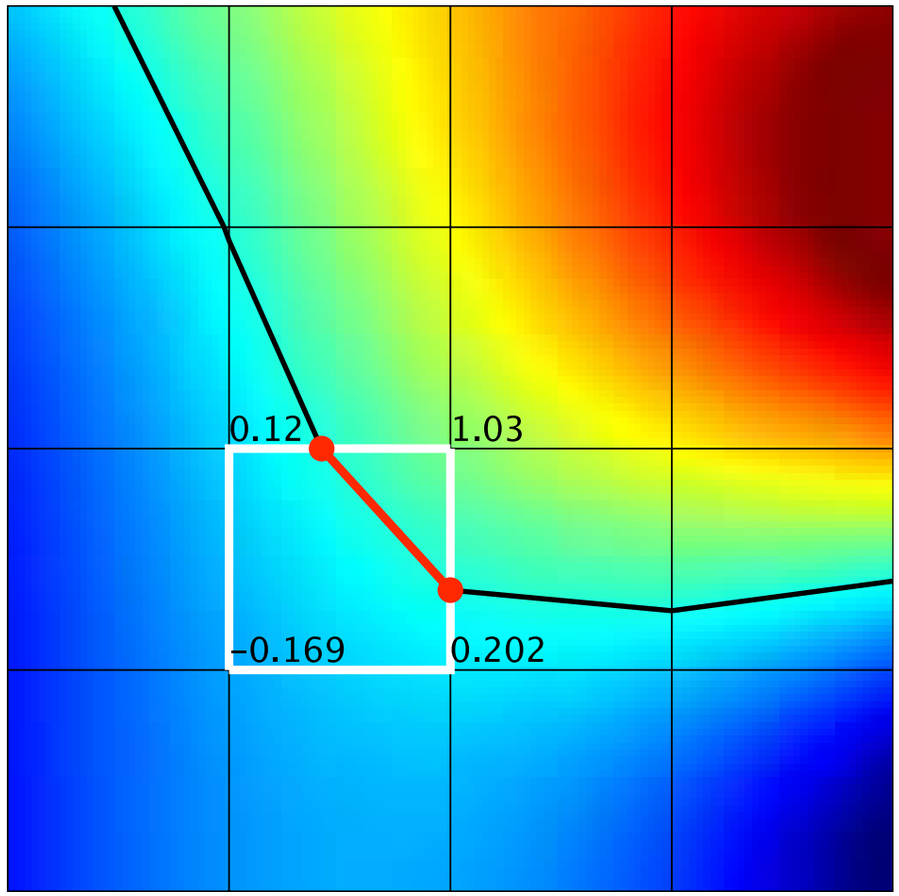
Marching Cubes (MC)

An interface exists along a cube edge if (intermediate value theorem):

$$[G(\mathbf{x}^+) - \nu][G(\mathbf{x}^-) - \nu] \leq 0$$

The interface locations are approximated along the cube edges using linear interpolation:

$$\mathbf{x}_v = \mathbf{x}^+ + \frac{G(\mathbf{x}^+) - \nu}{G(\mathbf{x}^+) - G(\mathbf{x}^-)}(\mathbf{x}^- - \mathbf{x}^+)$$

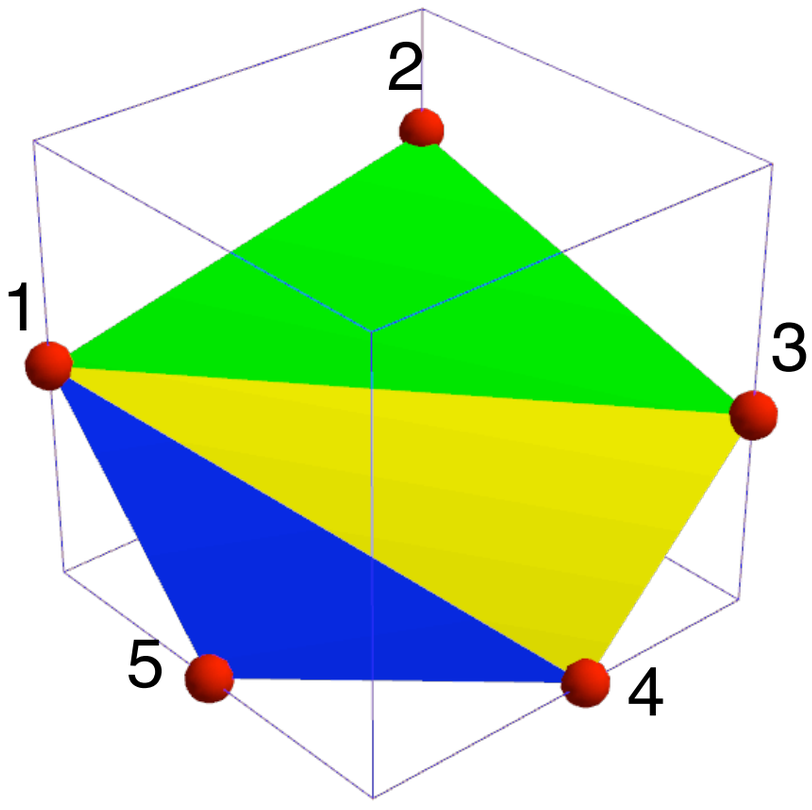


Marching Cubes (MC)

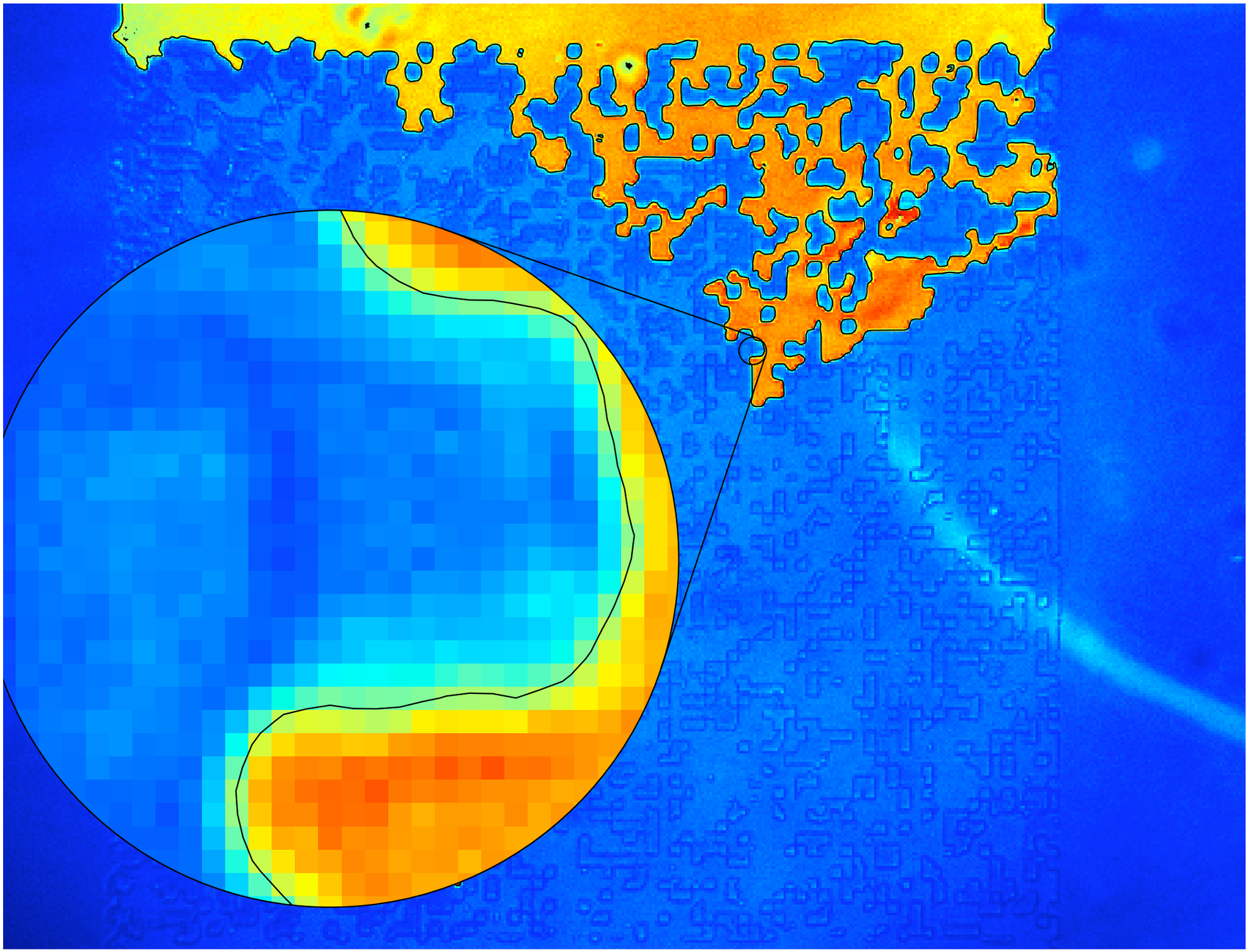
The reason that the MC algorithm is not called the marching squares algorithm is because it is a 3-D algorithm.

Three dimensions makes things more complicated:

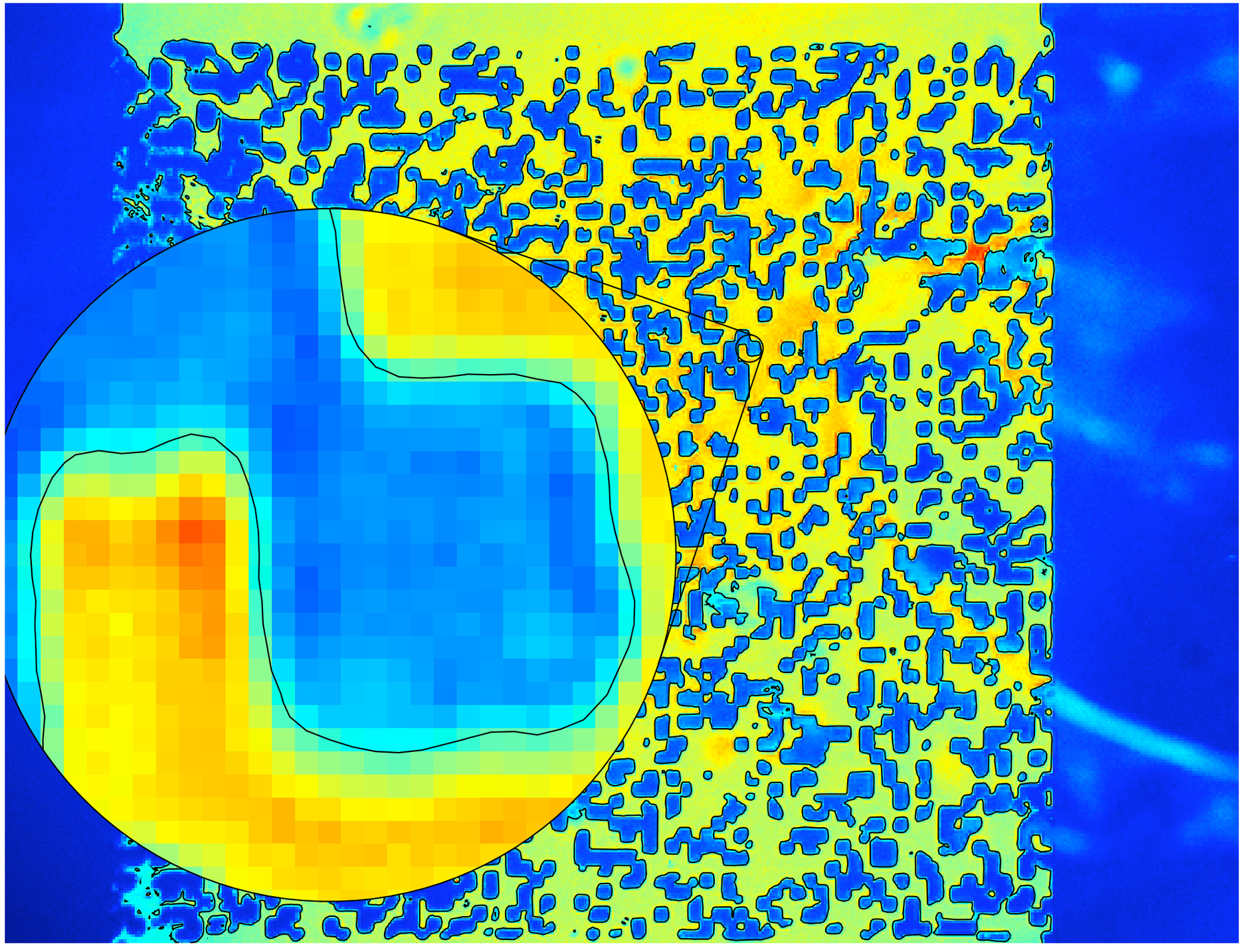
- surfaces are represented as a set of triangles
- you have to figure out a way to specify how to form these triangles



Data Sets with MC



Data Sets with MC



MC and Interfacial Areas

Suppose we want to measure interfacial area from a porous medium data set.
We are interested in the component interfacial areas between one phase and another. For a three phase (two fluids) system there are three such interfaces: $\Omega_{ws}, \Omega_{ns}, \Omega_{wn}$

The MC allows us to extract one interface at a time, which usually means you have to compute the interfacial areas you want using interfaces you can compute easily:

$$a_{wn} = \frac{1}{2}\Big(a_w + a_n - a_s\Big)$$

Porous Medium Marching Cubes (PMMC) Objectives

- Use the ideas developed in the MC to compute the component interfaces explicitly to obtain a precise representation of the system including the common curve.

- Implement in a general framework so that it can be applied to different kinds of data.

Porous Media Marching Cubes (PMMC)

- Solid surface defined by $S(\mathbf{x}) = \nu_s$
- Interface between wetting and non-wetting fluids defined by $F(\mathbf{x}) = \nu_{wn}$
- $S(\mathbf{x})$ is defined $\forall \mathbf{x} \in \Omega$, $F(\mathbf{x})$ may be defined $\forall \mathbf{x} \in \Omega_w \cup \Omega_n$
- Using linear interpolation, construct approximations for the interfaces and common curve within each cube as in the MC algorithm

Porous Media Marching Cubes (PMMC)

Sets needed for the PMMC algorithm.

Set	Description
$\mathcal{N}_{c,l}, \mathcal{N}_{li}, \mathcal{E}_{c,l}, \mathcal{X}_l$	nodes bounding cube, nodes forming all edges i, edges in cube l, and locations of nodes bounding cube
$\mathcal{V}_{wn}, \mathcal{V}_{ns}, \mathcal{V}_{ws}, \mathcal{V}_s$	vertices on the respective surface
$\mathcal{T}_{wn}, \mathcal{T}_{ns}, \mathcal{T}_{ws}, \mathcal{T}_s$	triangles comprised of vertices on the respective surface
$\mathcal{V}_{wns}$	vertices where the three phases meet
$\mathcal{L}_{wns}$	line segments consisting of pairs of vertices from $\mathcal{V}_{wns}$

PMMC Algorithm - Five Primary Cases

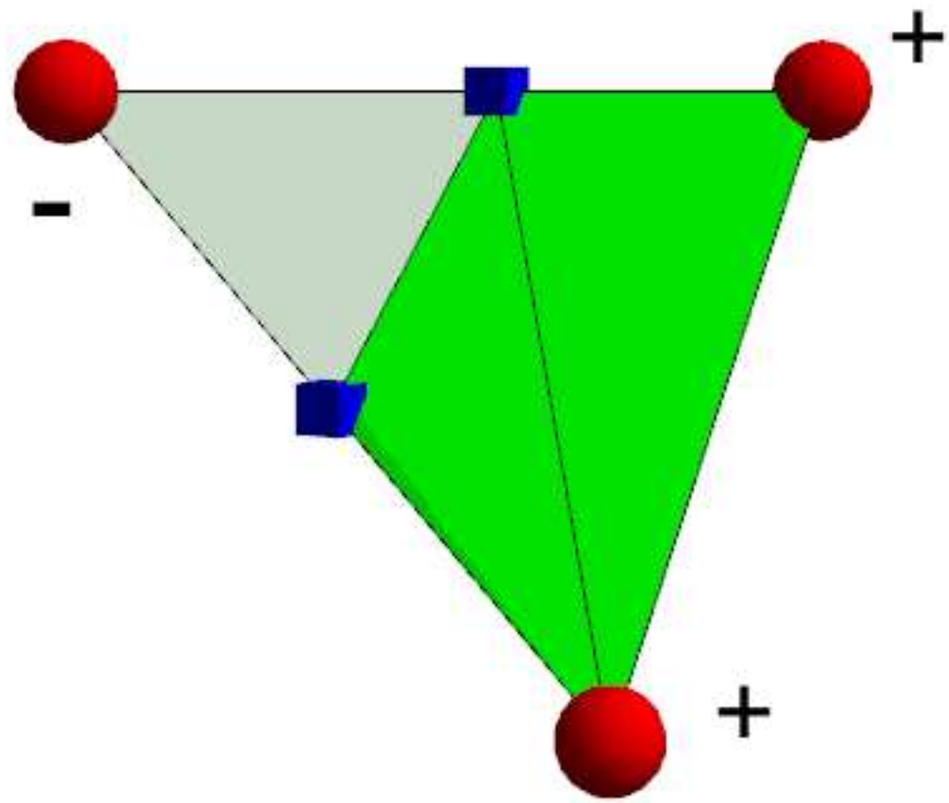
1. All cube corners correspond to single phase; no interface exists.
2. Some corners have $F(\mathbf{x}) \leq \nu_{wn}$, the rest have $F(\mathbf{x}) > \nu_{wn}$; only the wn interface exists.
3. Some corners have $S(\mathbf{x}) \leq \nu_s$, the rest have $F(\mathbf{x}) \leq \nu_{wn}$; only the ws interface exists.
4. Some corners have $S(\mathbf{x}) \leq \nu_s$, the rest have $F(\mathbf{x}) > \nu_{wn}$; only the ns interface exists.
5. Some corners have $S(\mathbf{x}) \leq \nu_s$, some have $F(\mathbf{x}) \leq \nu_{wn}$, the rest have $F(\mathbf{x}) > \nu_{wn}$; all interfaces and common curve are present

Algorithm 1 PMMC Algorithm

for $l = 1, ..., n_c^3$ **do**
  Form $\mathcal{N}_{c,l}, \mathcal{N}_{li}, \mathcal{E}_{c,l}$ and $\mathcal{X}_l$
  if $\mathbf{x}_i \in \Omega_\alpha \forall \mathbf{x}_i \in \mathcal{X}_l, \alpha \in \{w, n, s\}$ **then**
    no surfaces or common curve segments exist in cube l
  else if $\mathbf{x}_i \in \{\Omega_w, \Omega_n\} \forall \mathbf{x}_i \in \mathcal{X}_l$ **then**
    apply the MC algorithm using $F(\mathbf{x}) = \nu_{wn}$ to find $\mathcal{V}_{wn,l}$ and $\mathcal{T}_{wn,l}$
  else if $\mathbf{x}_i \in \{\Omega_w, \Omega_s\} \forall \mathbf{x}_i \in \mathcal{X}_l$ **then**
    apply the MC algorithm using $S(\mathbf{x}) = \nu_s$ to find $\mathcal{V}_{ws,l}$ and $\mathcal{T}_{ws,l}$
  else if $\mathbf{x}_i \in \{\Omega_n, \Omega_s\} \forall \mathbf{x}_i \in \mathcal{X}_l$ **then**
    apply the MC algorithm using $S(\mathbf{x}) = \nu_s$ to find $\mathcal{V}_{ns,l}$ and $\mathcal{T}_{ns,l}$
  else
    apply the MC algorithm using $S(\mathbf{x}) = \nu_s$ to find $\mathcal{V}_s$ and $\mathcal{T}_s$
    find $F(\mathbf{x}) \forall \mathbf{x} \in \mathcal{V}_s$
    if $F(\mathbf{x})$ is defined $\forall \mathbf{x}_i \in \mathcal{X}_l$ **then**
      use linear interpolation to form $\mathcal{V}_{wns,l}$ and $\mathcal{L}_{wns,l}$
    else
      use extrapolation to form $\mathcal{V}_{wns,l}$ and $\mathcal{L}_{wns,l}$
    end if
    use $\mathcal{V}_s$ and $\mathcal{V}_{wns,l}$ to form $\mathcal{V}_{ws}$ and $\mathcal{V}_{ns}$
    form $\mathcal{T}_{ws,l}$ and$\mathcal{T}_{ns,l}$
    form $\mathcal{V}_{wn,l}$ from $\mathcal{E}_{c,l}$
    form $\mathcal{T}_{wn,l}$ from $\mathcal{V}_{wns,l}$ and $\mathcal{V}_{wn,l}$
  end if
  update global sets $\mathcal{V}_{wn}, \mathcal{V}_{ws}, \mathcal{V}_{ns}, \mathcal{V}_{wns}, \mathcal{T}_{wn}, \mathcal{T}_{ws}, \mathcal{T}_{ns},$ and $\mathcal{L}_{wns}$ as needed
end for

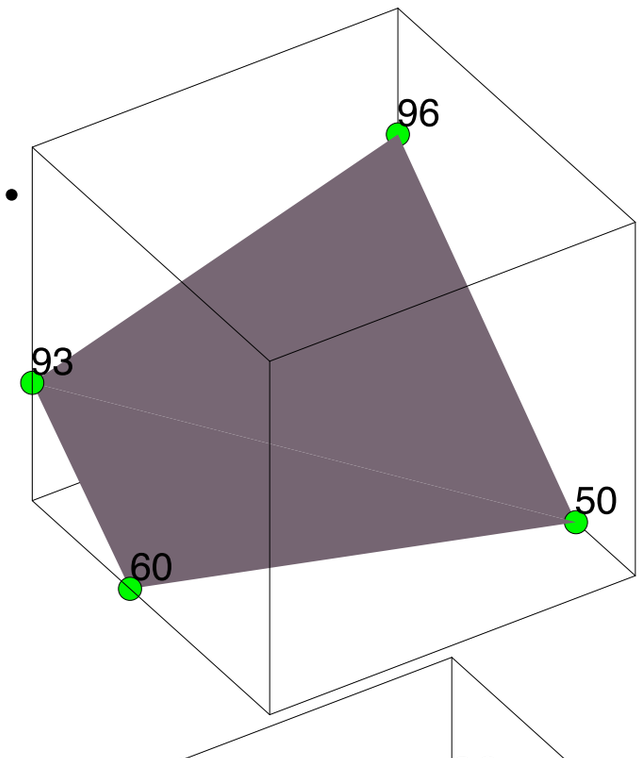
PMMC - Finding the Common Curve

- Values of $F(\mathbf{x})$ are obtained at every point on the solid surface.
- These values may be used to compute the common line along triangle edges in exactly the same fashion vertices are computed along cube edges in the MC algorithm

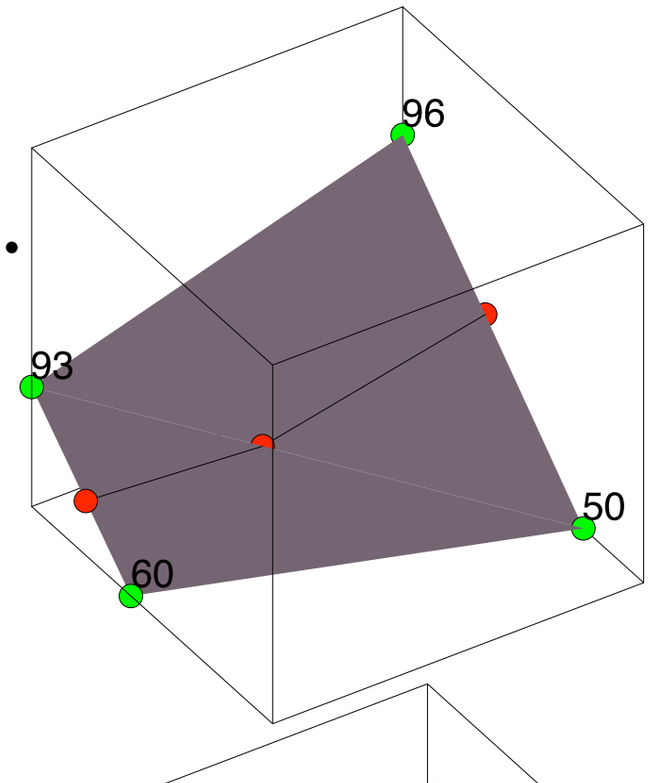


PMMC

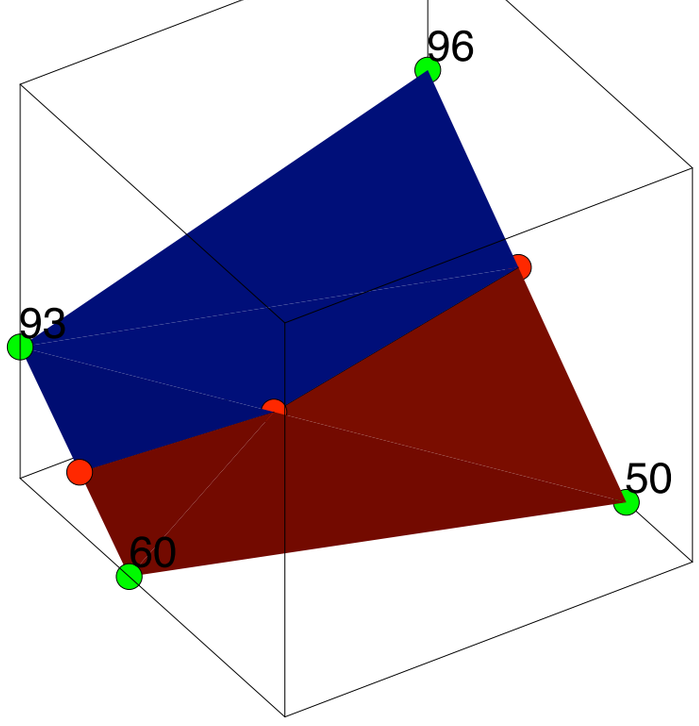
Step 1.



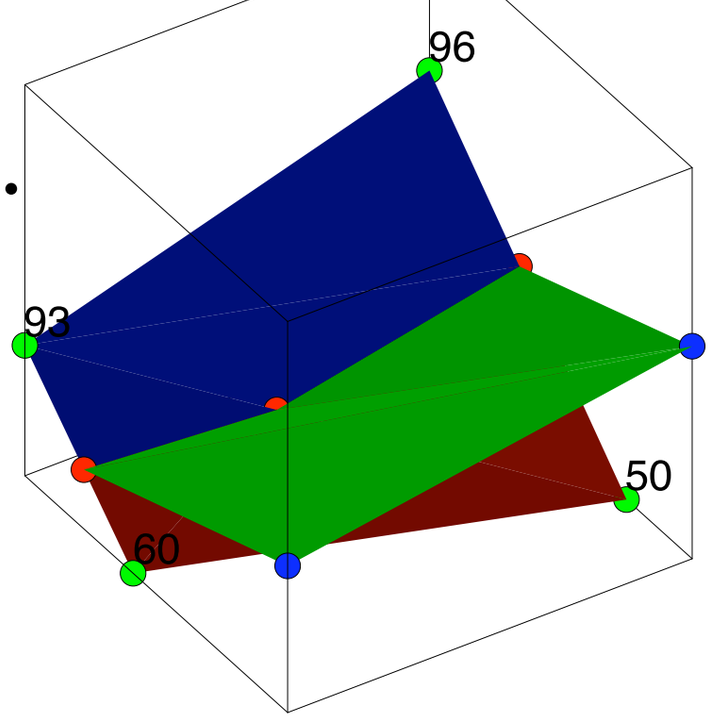
Step 2.



Step 3.



Step 4.



Higher Order Approach (HOPMMC)

Construct a higher order representation of $S(\mathbf{x})$ and $F(\mathbf{x})$ use this to compute the common curve more accurately.

$$\check{F}(\mathbf{x}) = a_0 + a_1 x + a_2 y + a_3 z + a_4 xy + a_5 xz + a_6 yz + a_7 xyz$$

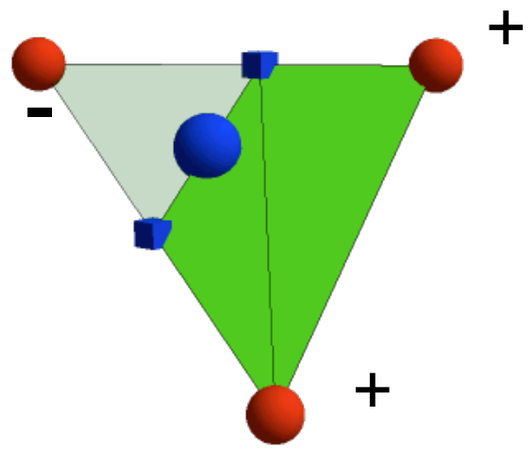
Based on our knowledge of the system, we require

$$\begin{aligned} \check{F}(\mathbf{x}) - \nu_{wn} &= 0 \\ \check{S}(\mathbf{x}) - \nu_s &= 0 \\ \mathbf{n}\cdot(\mathbf{x} - \mathbf{x}_0) &= 0 \end{aligned}$$

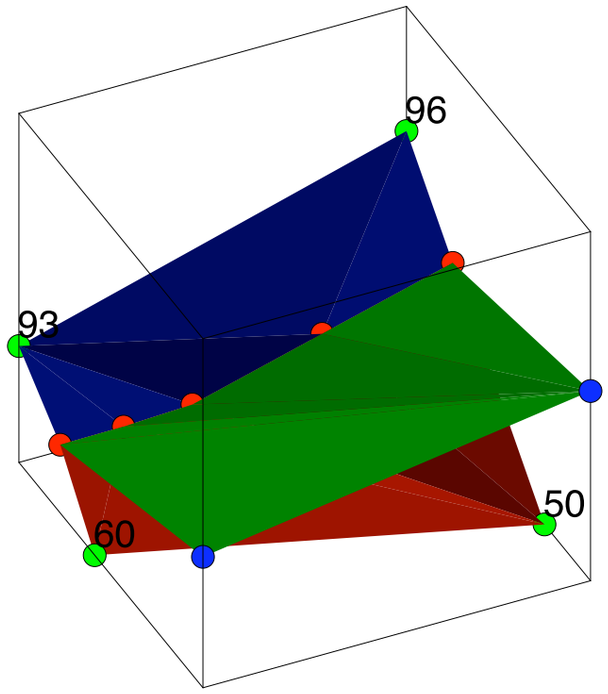
$$\mathbf{n} = [\nabla \check{F}(\mathbf{x}_0) \times \nabla \check{S}(\mathbf{x}_0)] / |\nabla \check{F}(\mathbf{x}_0) \times \nabla \check{S}(\mathbf{x}_0)|$$

HOPMMC

-Subdivide each segment of the common curve to generate initial points



- Use points as an initial guess for a Newton iteration that moves them to the proper position.



- Note that this could be applied to improve the PMMC surfaces also.

PMMC in Application

We have methods to construct the objects we are interested in as long as we have functions $S(\mathbf{x})$ and $F(\mathbf{x})$ and isovalues ν_s and ν_{wn}

- Gives us information about the position of the phases and interfaces as well as other information

- How do we obtain these functions and isovalues for a particular data set?

Test Case 1: Lattice Boltzmann (LB)

The solid phase is composed of a sphere packing so that the geometry is known exactly. $S(\mathbf{x})$ is computed using the signed distance function:

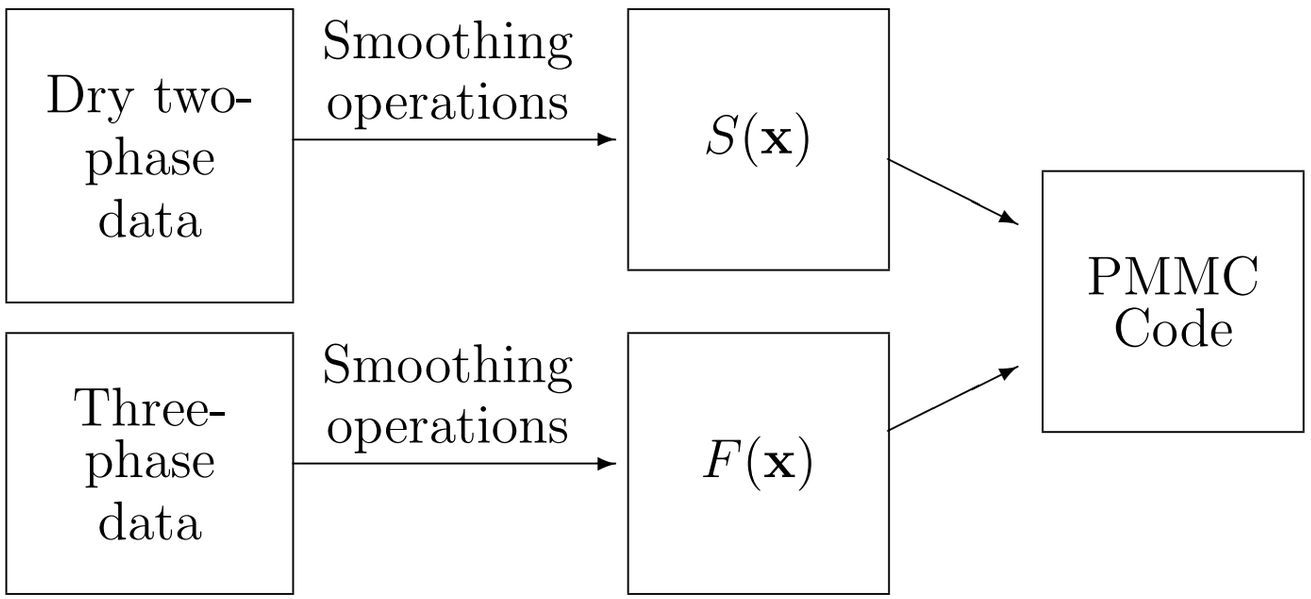
$$S(\mathbf{x}) = \min\left\{||\mathbf{x} - \mathbf{c}_s||_2 - r_s\right\}, \text{ for } s = 1, \ldots, n_S$$

Note that $S(\mathbf{x}) = 0$ by construction at the solid surface.

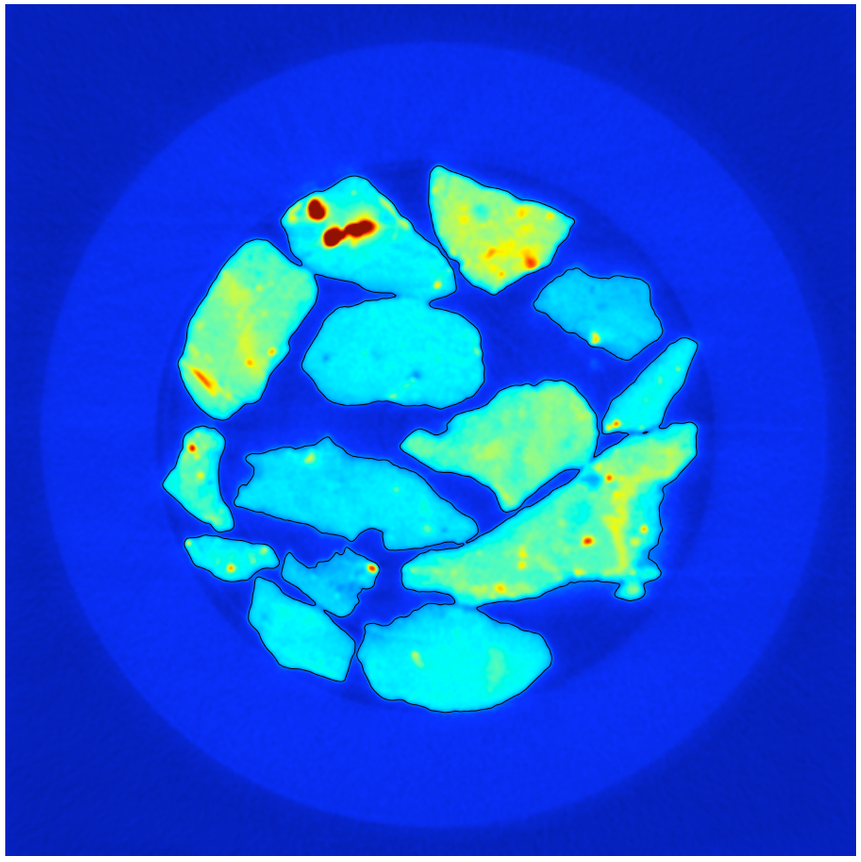
$F(\mathbf{x})$ is chosen to be the fluid density output from the LB simulation. Because it is only defined in the porespace, an extrapolation is necessary to determine its value at the solid surface.

Test Case 2: X-Ray Tomography

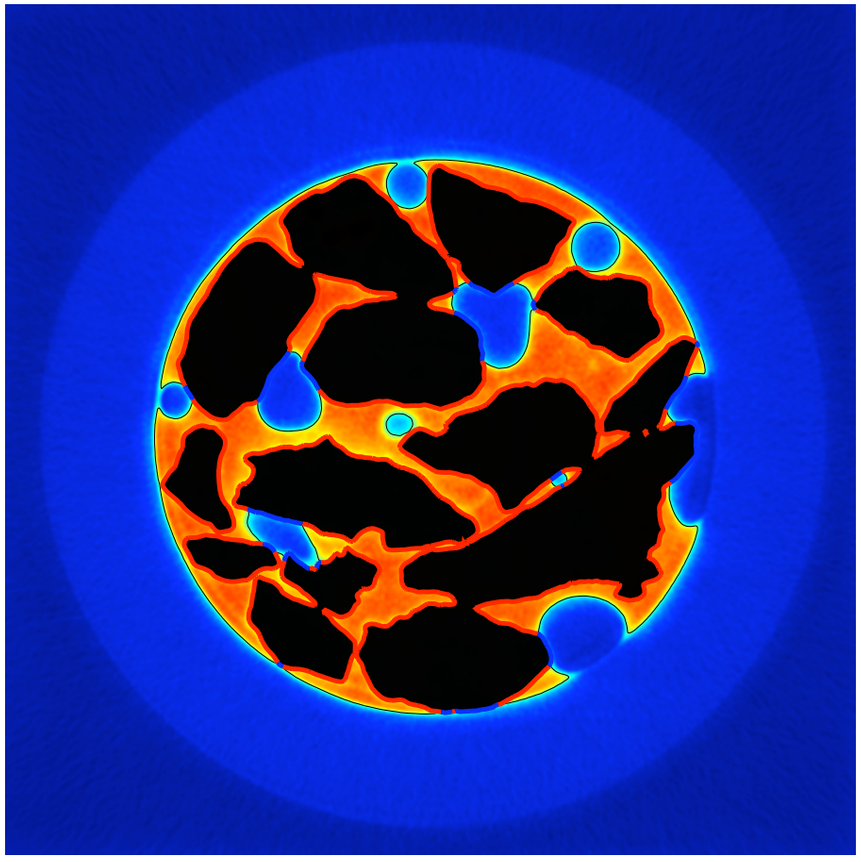
Objective for experimental data: Determine $S(\mathbf{x})$ and $F(\mathbf{x})$, as well as isovalues ν_s and ν_{wn}



Test Case 2: X-Ray Tomography

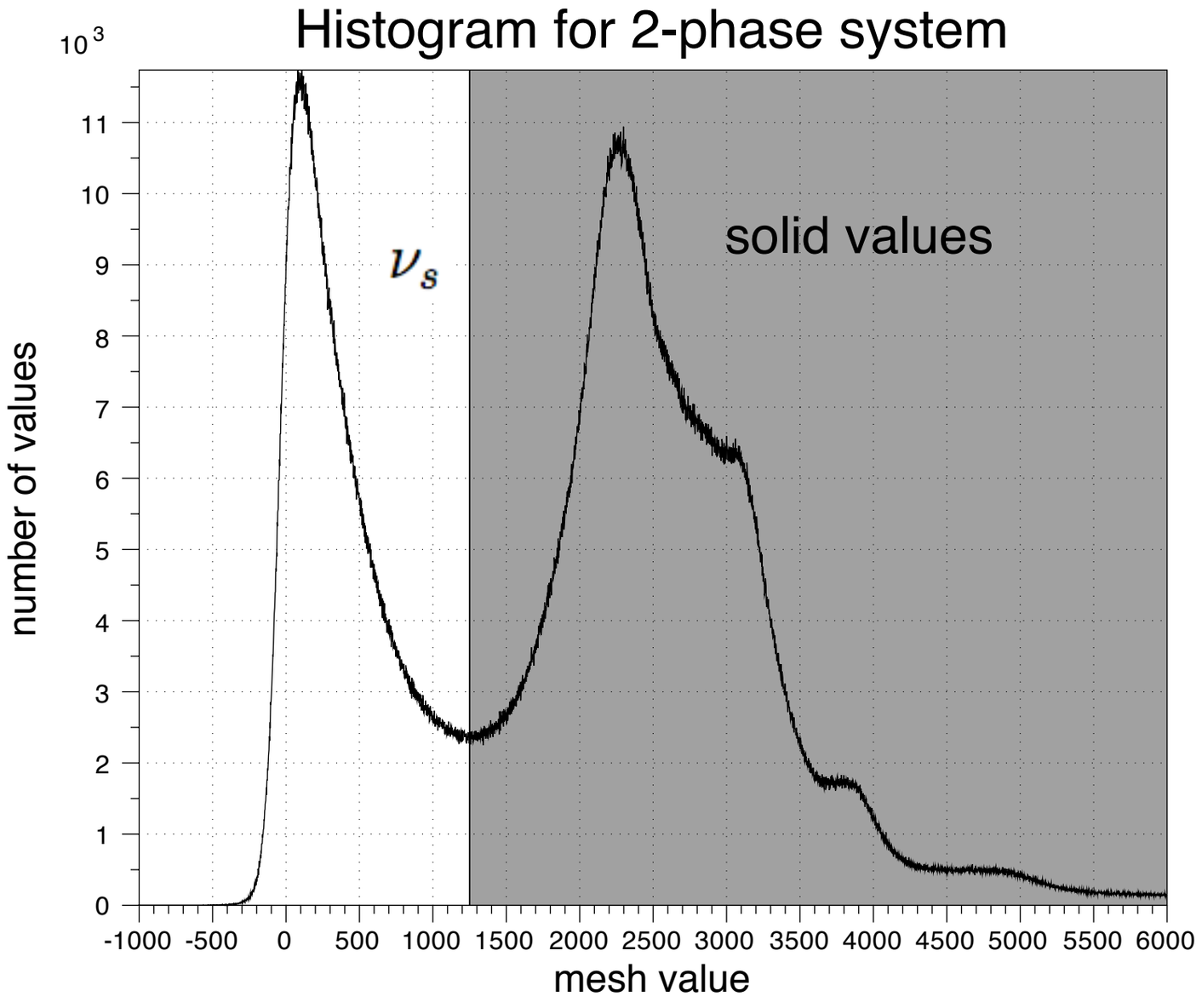


“dry” data

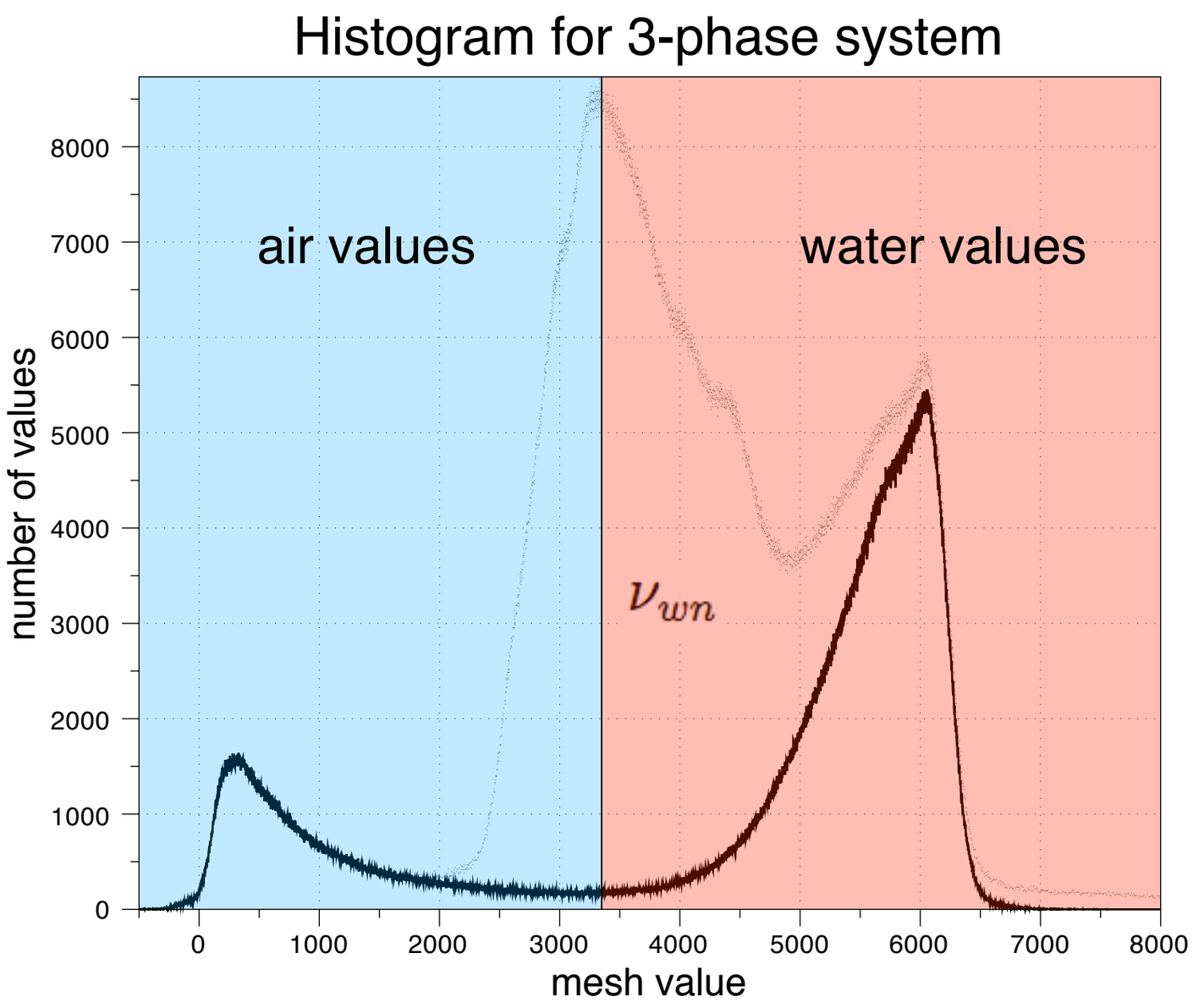


three-phase data

Isovalue determination - solid phase



Isovalue determination - fluid phase

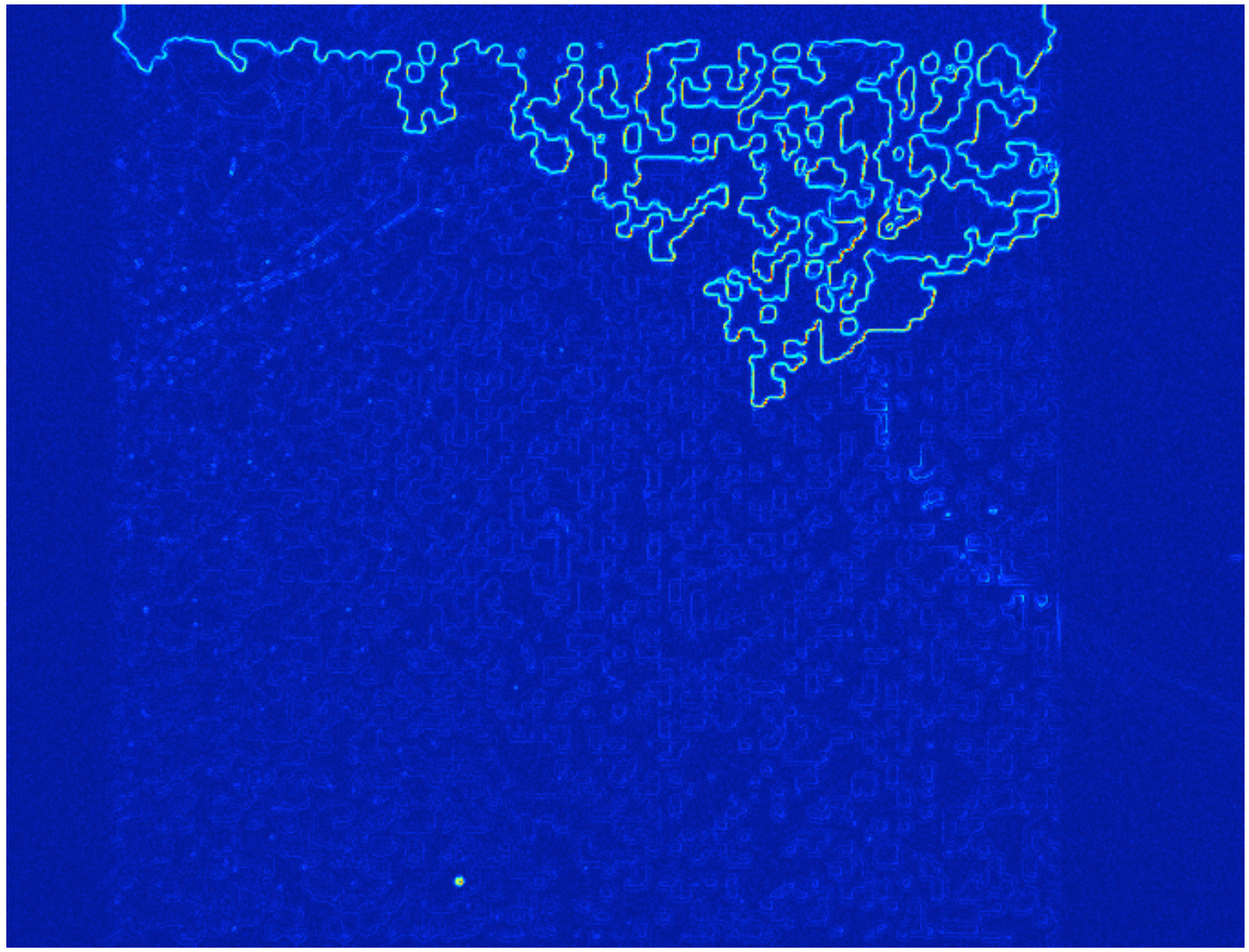


Things to Consider: Isovalue Determination

- If you have any information that you know for sure (ie. porosity, saturation), you should be able to use this information to determine isovalues that will match them.

- Interfacial area estimates should be relatively insensitive to the choice of isovalue.

Other Ideas to Consider



“Edge detection” or “steepest descent”

Edge Detection Algorithms

Take the gradient of the values and look for the regions where they are high.

If you are looking for a maximum in the gradient and wish to find points that do not live on the grid, you need to use a non-linear approximation (not too bad in 2-D).

Data Set Overlap