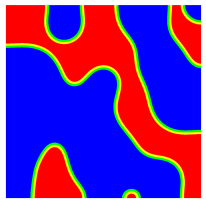


PRISMS-PF

The PRISMS Phase Field Code

Trainers:

Steve DeWitt

David Montiel

Jason Luce

U.S. DEPARTMENT OF ENERGY | Office of Science

UNIVERSITY OF MICHIGAN

Center for PRedictive Integrated Structural Materials Science

PRISMS

Training Objective

Learn enough of PRISMS-PF to allow you to go home and start using it in your work

Plan for the Training

Schedule:

- Brief introductory comments
- Guided walkthrough with exercises interspersed
  - Tour of the file system
  - Running example applications
  - Visualization
  - Tutorial on writing equations in the weak form
  - Walkthrough of the application files
- Dedicated time for individual exercises using PRISMS-PF to modify existing applications and create new applications

Now

Today

Wednesday

One Slide on Phase Field Modeling

- Diffuse interface approach to modeling microstructure evolution
- Used to study phase separation in systems with 2+ free energy minima
- Evolution equations derived from a free energy functional
  - May or may not have a physical basis

$$\frac{\partial \eta}{\partial t} = -L\frac{\delta F}{\delta \eta}$$

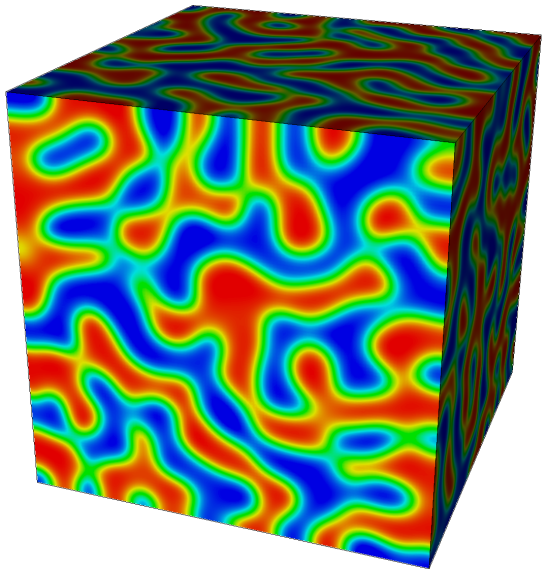
Allen-Cahn Equation
(non-conserved dynamics)

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \left(M \nabla \left(\frac{\delta F}{\delta \phi} \right) \right)$$

Cahn-Hilliard Equation
(conserved dynamics)

- Applications include:

  solidification, precipitation, grain growth, phase separation in batteries, deposition, ferroics

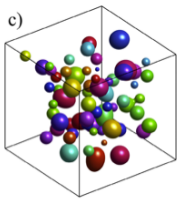


Spinodal Decomposition
(Cahn-Hilliard Equation)

Motivation:
What's hard about writing a phase field code?

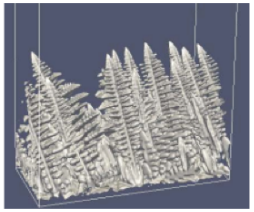
1. Wide diversity of models and coupled physics makes code reuse difficult

Precipitation



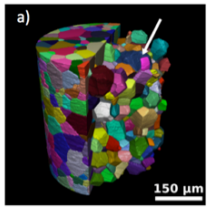
Poulsen,Voorhees, Acta Materialia (2016)

Solidification



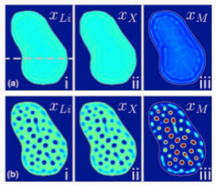
Shimokawabe, et al., SC '11 (2011)

Grain Evolution



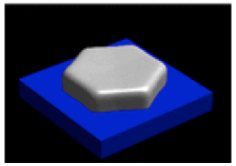
McKenna, et al. Acta Materialia (2014)

Batteries



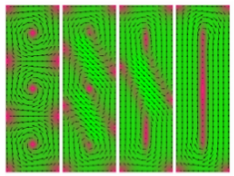
Yu, et al. J. Phase Equil. Diff. (2016)

Deposition



DeWitt, et al. J. Electrochem. Soc. (2016)

Ferroics



Chen, Zheng, Wang, Appl. Phys. Lett. (2012)

No real "typical" governing equations

Large variety of formulations and terms

Motivation:
What's hard about writing a phase field code?

2. Simulating large, physically representative systems is computationally intensive

nersc.gov

Simulations often take days on 10s-100s of cores

Simulations are often done in 2D for tractability

Physical fidelity requires strong numerical performance

Four Principles Guiding PRISMS-PF Development

1. Its computational performance, including parallel scalability, should meet or exceed that of typical phase field codes
2. It should accommodate a wide variety of phase field models and applications
3. The interface for creating or modifying governing equations should be simple, quick, and separate from the numerics
4. It should be open source with a permissive license so it is available to everyone and advances can be shared by the community

PRISMS-PF

An Open Source, Finite Element,
General Purpose Phase-Field Platform
(prisms-center.github.io/phaseField)

Advanced Capabilities:

Matrix-free finite element approach

Higher-order elements

Multi-level parallelization:
MPI/Threads/Vectorization

Adaptive meshing

Explicit nucleus placement

Grain-remapping

Ease of Use:

Simple interface to solve an arbitrary number of coupled PDEs

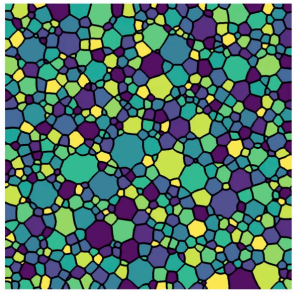
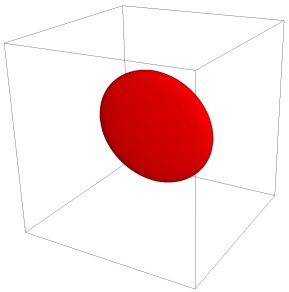
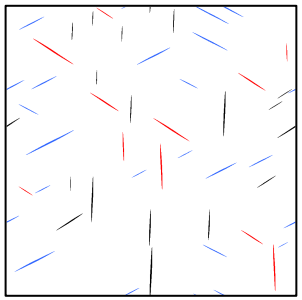
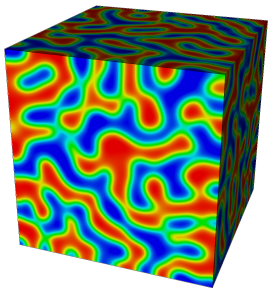
Detailed online user guide

24 pre-built applications

Simple Docker-based installation

nanoHUB tool w/ GUI for educational use

Integrated with Materials Commons


Structure of PRISMS-PF

- Core library
  - Generates mesh, does the finite element calculation, outputs files, etc.
- Apps
  - Each app is a directory that contains an input file and some application files
    - Governing equations, boundary conditions, initial conditions, numerical and model parameters, postprocessing expressions
  - Using an app requires no C++ experience (parsed text input file)
  - Developing an app requires minimal C++ experience and no FEM experience
- Tests
  - Suite of unit and regression tests
  - Continuous integration testing with Travis CI

Three Types of PRISMS-PF Users

1. Uses PRISMS-PF applications
   - No C++ knowledge needed
   - No deal.II knowledge needed
2. Creates PRISMS-PF applications
   - Minimal C++ knowledge needed
   - No deal.II knowledge needed
3. Extends PRISMS-PF itself
   - C++ knowledge needed
   - deal.II knowledge needed

Performance of PRISMS-PF

Comparison vs. Finite Difference

- Finite difference code
  - Fortran with MPI parallelization
  - 2nd order central differences
  - Explicit time stepping
- Two growing particles in 3D
  - ahn

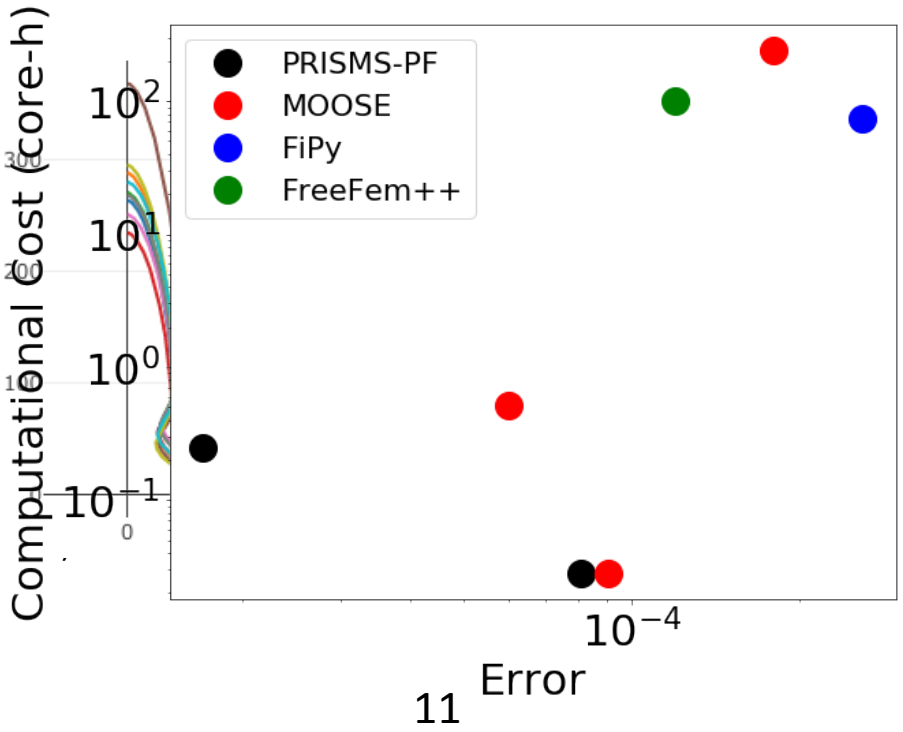
PRISMS-PF Performance at Fixed Error*

Element/Mesh Type	Speedup vs. FD
Linear/Regular	0.028
Quadratic/Regular	0.90
Cubic/Regular	1.9
Cubic/Adaptive	12.0

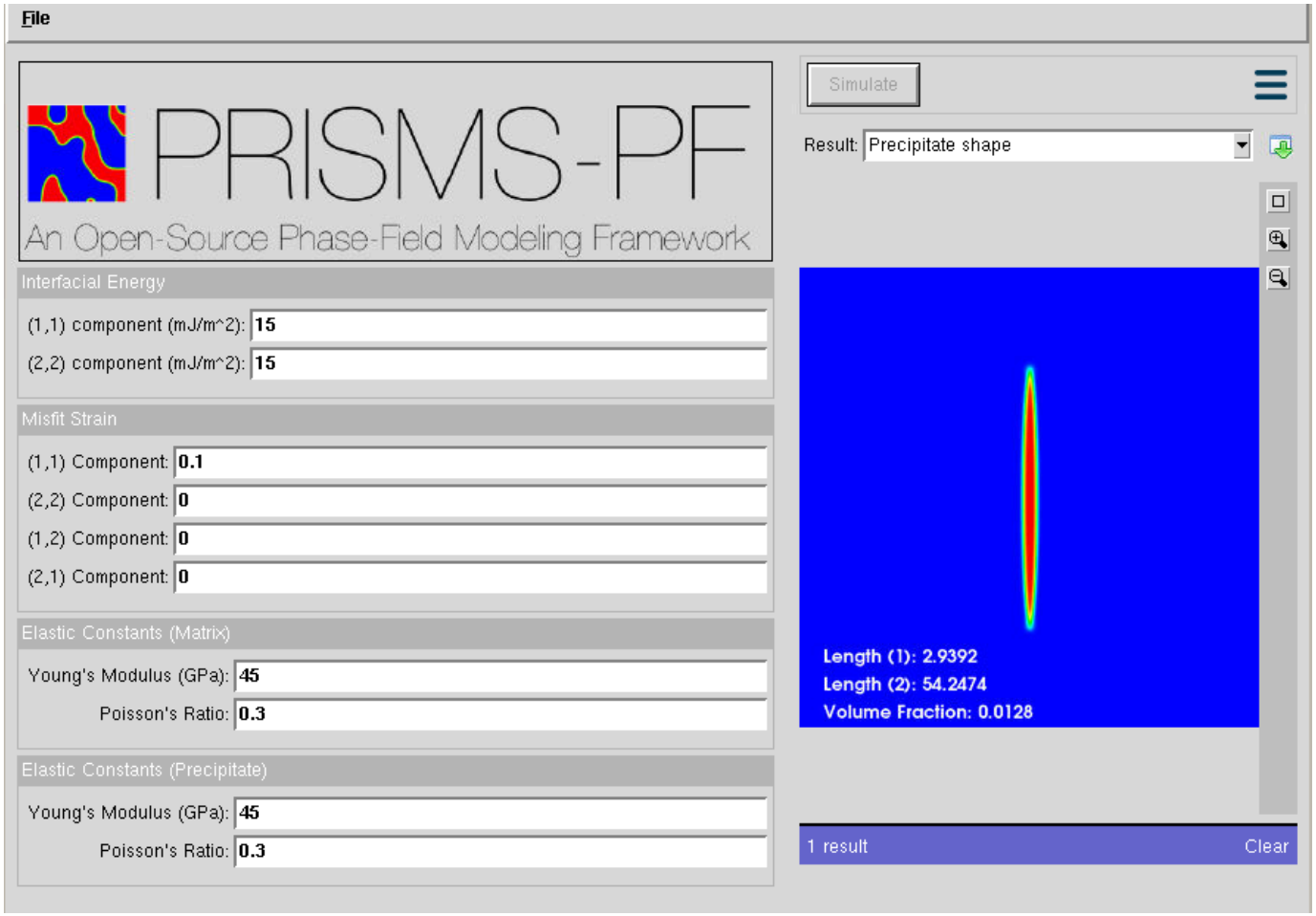
*Equivalent to 5 pts. in the interface for FD

Comparison vs. Open Source Codes

- PFHub dendritic solidification benchmark
  - Developed by the CHiMaD Center (NIST/Northwestern/Argonne)
- Results uploaded to https://pages.nist.gov/pfhub/



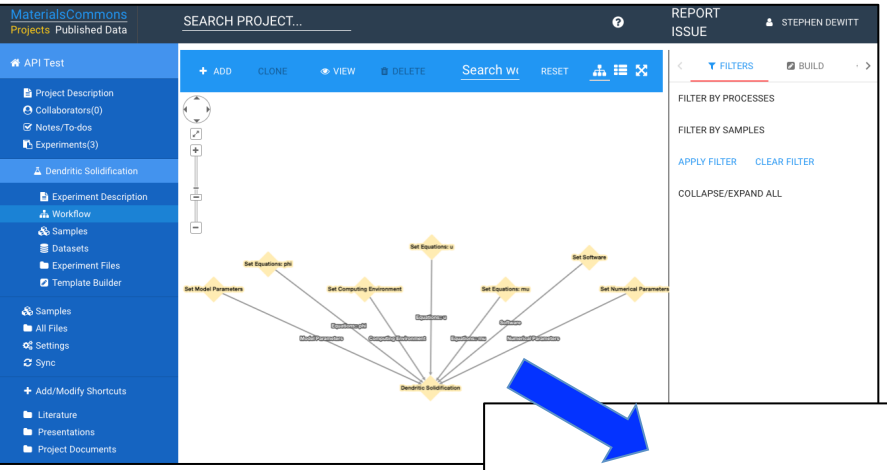
PRISMS-PF on nanoHUB (https://nanohub.org/tools/prismspfmisfit)

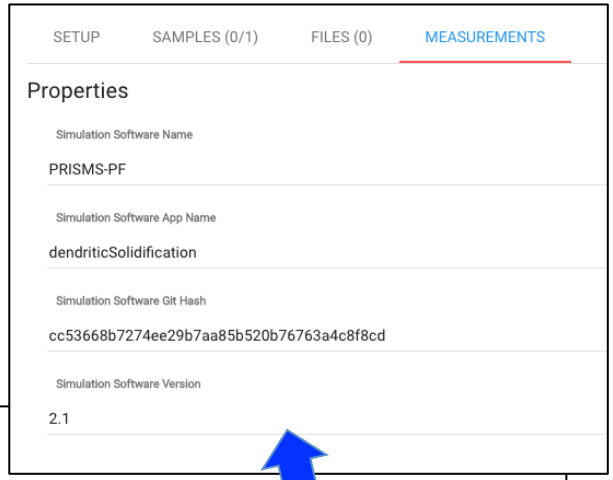


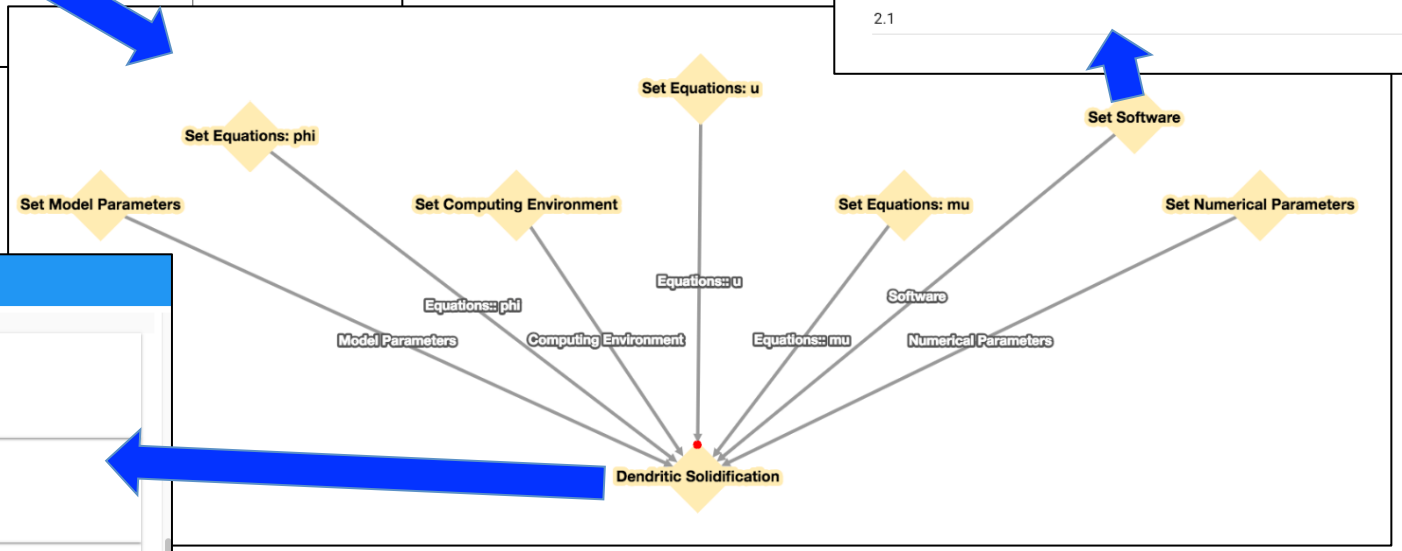
- 2D equilibrium shape of a precipitate, given misfit/interfacial energy/ elastic constants
- Targeted for classroom use
- Companion worksheet being developed by Susan Gentry (UC Davis)

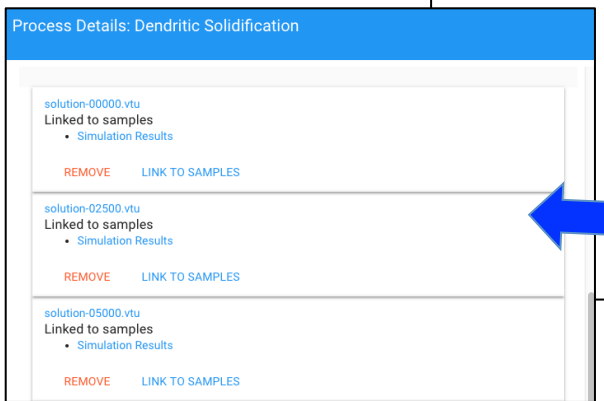
PRISMS-PF and Materials Commons

Command line tool automatically parses the PRISMS-PF app files and uploads the metadata and simulation results to Materials Commons



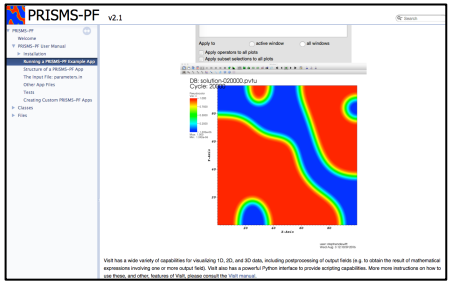


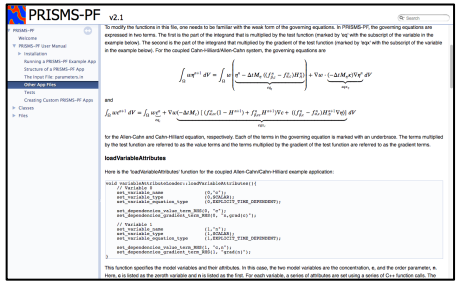




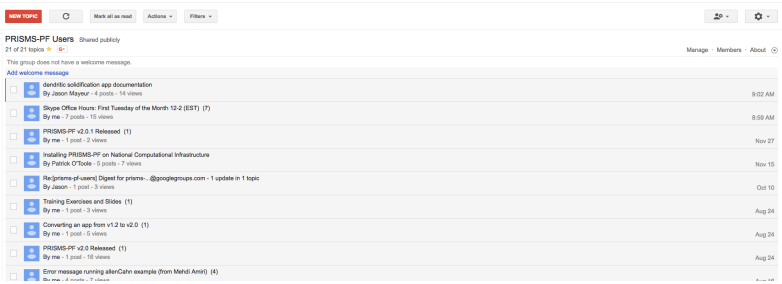
Community Support

- Online user manual





- Online forum



- Monthly Skype office hours

Questions?

Downloading PRISMS-PF

https://github.com/prisms-center/phaseField

Setting Environment Variables

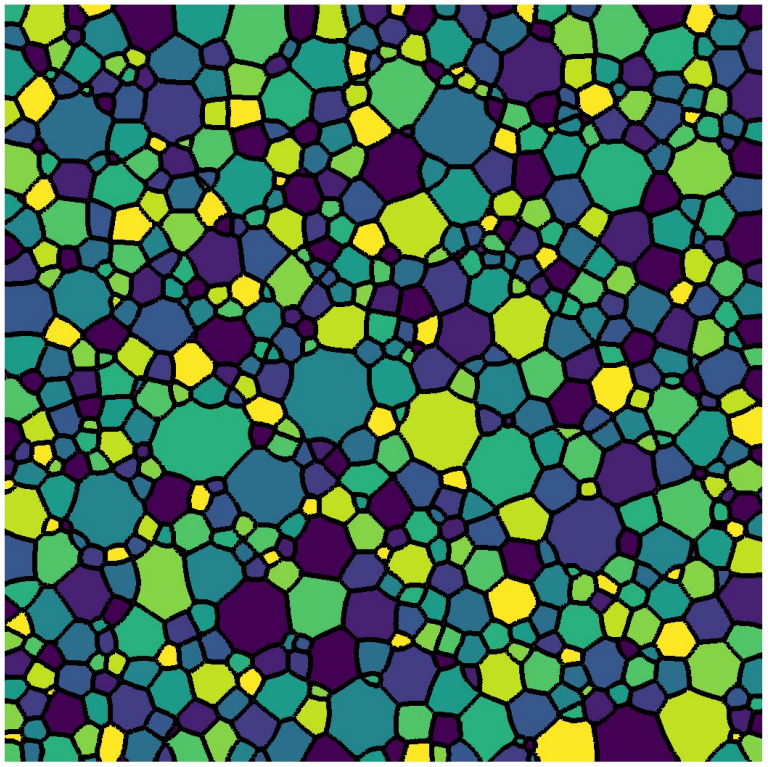
Type this on the command line:

source /afs/umich.edu/user/s/t/stvdwtt/Public/prismspf_script.sh

```
dukenukem% source /afs/umich.edu/user/s/t/stvdwtt/Public/prismspf_script.sh
```

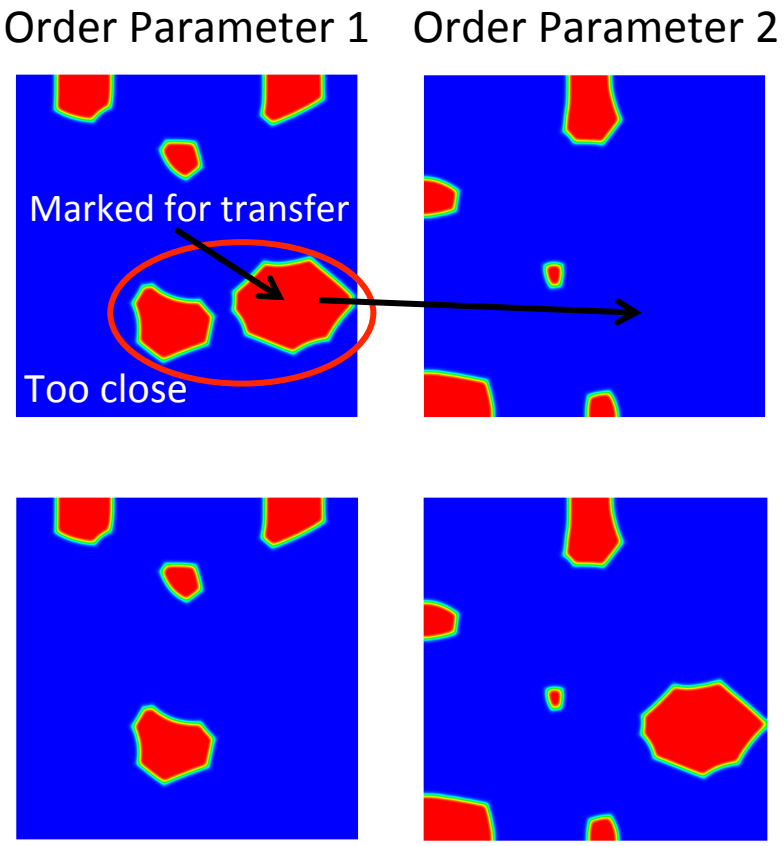
Grain Remapping in PRISMS-PF

- A common use of phase field models is for polycrystalline systems
- Naïve approach: one order parameter per grain (infeasible outside of toy problems)
- Improved approach: Store multiple grains per order parameter and move them if they get close



Initial microstructure imported from DREAM.3D

Grain Remapping in PRISMS-PF



- Each grain is identified using a recursive flood fill
- Simplified representation of each grain is stored
  - Currently a circle/sphere
  - A persistent grain ID is saved for each grain
- Modified greedy coloring algorithm to determine the new home for a grain marked for transfer

Installing the Materials Commons Command Line Interface Tool

Type this on the command line:

source /afs/umich.edu/user/s/t/stvdwtt/Public/install_mccli.sh

Cloning a Materials Commons Project

- Create a directory for your Materials Commons projects and enter it
  - $ mkdir mc_projects
  - $ cd mc_projects
- View current Materials Commons projects
  - $ mc proj
- Clone the project to your local machine
  - $mc clone [id of the project]

Creating a New Dataset

- Enter the project directory
  - $ cd [project name]
- Copy a PRISMS-PF app directory here
  - Either using ‘cp’ or the GUI
- Enter that directory
  - $ cp [app name]
- Parse the input files and upload the results
  - $ mc prismspf simulation --create --full-simulation --proc-name Dendritic Solidification --num-cores 8