PRISMS-PF

The PRISMS Phase Field Code

Trainers:

Steve DeWitt (PRISMS-PF Lead Developer)

Beck Andrews

David Montiel

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
    - Tour of the file system
    - Running example applications
    - Visualization
    - Tutorial on writing equations in the weak form
    - Walkthrough of the application files
  - Brief tutorial on the new plugin for the Materials Commons CLI
  - Individual exercises using PRISMS-PF to modify existing applications and create new applications

Now

This morning

This afternoon and Wednesday

One Slide on Phase Field Modeling

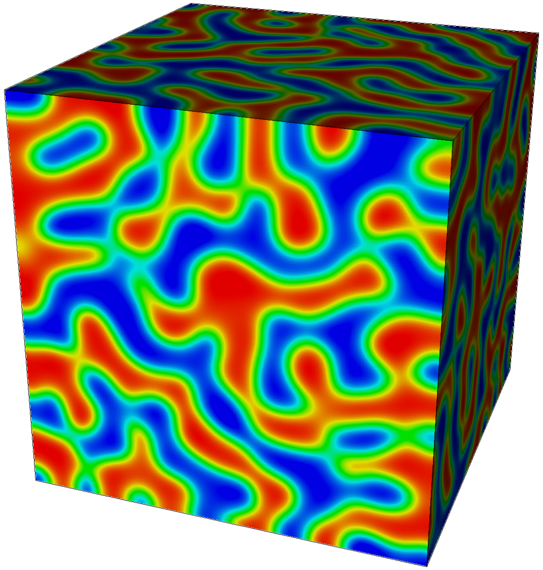
- Diffuse interface approach to modeling microstructure evolution
- Used to study phase separation in systems with 2+ free energy minima
- Evolution equations derived from a free energy functional
  - May or may not have a physical basis

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

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)



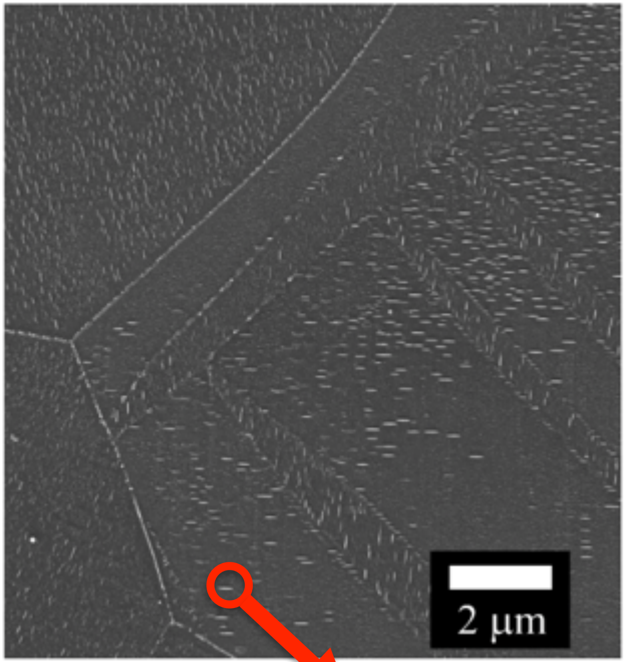
Spinodal Decomposition
(Cahn-Hilliard Equation)

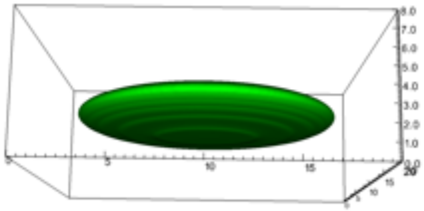
- Applications include:

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

Motivation and Requirements for the PRISMS Phase Field Code

- Microstructure simulations often require very large calculations
- These are **terascale** computational problems with 10^6 to 10^9 DOF
- For grain growth, need the ability to **track 100s of grains**
- Many different governing equations fall under the umbrella of "phase field modeling", need flexibility
- Finite Element Method
  - Easily allows **mesh adaptivity, arbitrary geometries**, and **high order of accuracy**
  - Can build from well-established community codes (e.g. **deal.II**)
  - Allows a shared toolset with PRISMS-Plasticity and PRISMS-RSDFT
  - However, need a **low memory approach** to compete with finite difference





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 permission license so it is available to everyone and can be modified and improved

PRISMS-PF

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

Advanced Capabilities:

Matrix-free finite element approach

High-order elements

Hybrid parallelization: MPI/Threads/Vectorization

Adaptive meshing

Explicit nucleus placement

Grain-remapping

High-Performance:

Ideal scaling for >1,000 processors

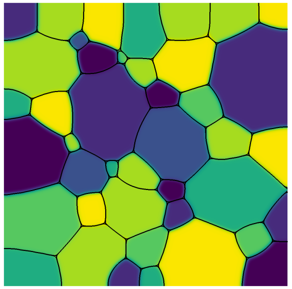
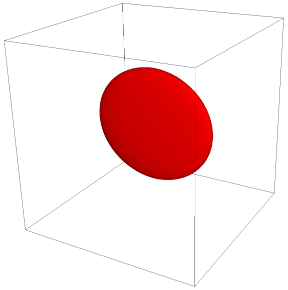
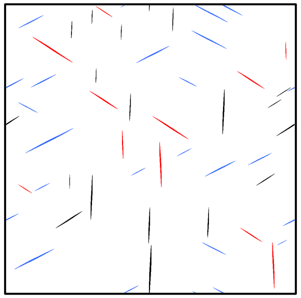
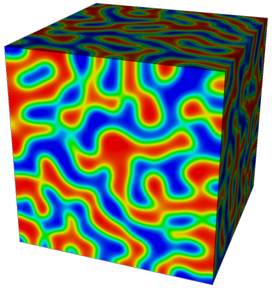
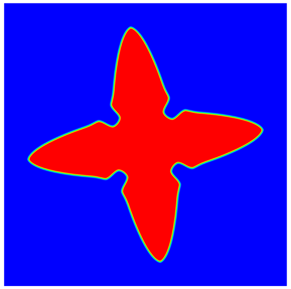
Improved performance with finite difference (10x without adaptive meshing, 100x with adaptive meshing)

User-Friendly:

Simple interface to solve an arbitrary number of coupled PDEs

Detailed online user guide
24 applications to get you started

Simple Docker-based installation

Performance vs. Finite Difference

Test Problem: Coupled Cahn-Hilliard/Allen-Cahn, 2 growing spheres
Comparison Code: Custom FD code written in Fortran w/ MPI

PRISMS-PF element and mesh type	$T_{FD}/T_{PRISMS-PF}$			
	$E_{L_2} = 105.3$ $\lambda \approx 7$ pts.	$E_{L_2} = 142.8$ $\lambda \approx 5$ pts.	$E_{L_2} = 196.3$ $\lambda \approx 4$ pts.	$E_{L_2} = 380.7$ $\lambda \approx 3$ pts.
Linear elements, regular mesh	0.105*	0.0907	0.0721	0.115*
Quadratic elements, regular mesh	6.19	3.30	1.67	1.47*
Cubic elements, regular mesh	11.1	6.13	3.00	1.89
Cubic elements, adaptive mesh	92.6	47.8	21.7	11.7

* Extrapolated value

Structure of PRISMS-PF

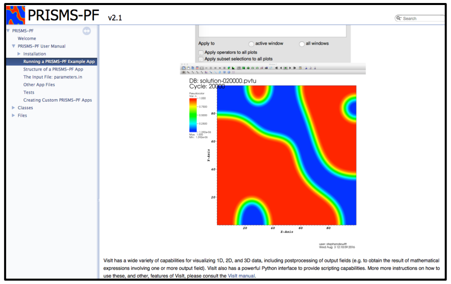
- Core library
  - Generates mesh, does the finite element calculation, outputs files, etc.
- Apps
  - Each app is a directory that contains an input file and some application files
    - Governing equations, boundary conditions, initial conditions, numerical and model parameters, postprocessing expressions
  - Copy and paste an app directory to create a new app
- Tests
  - Suite of unit and regression tests

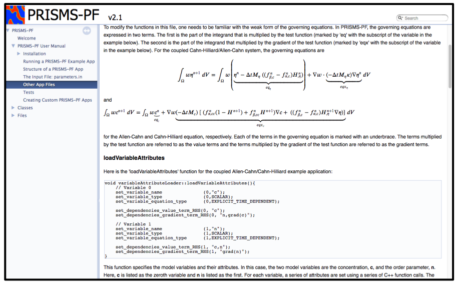
Three Types of PRISMS-PF Users

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

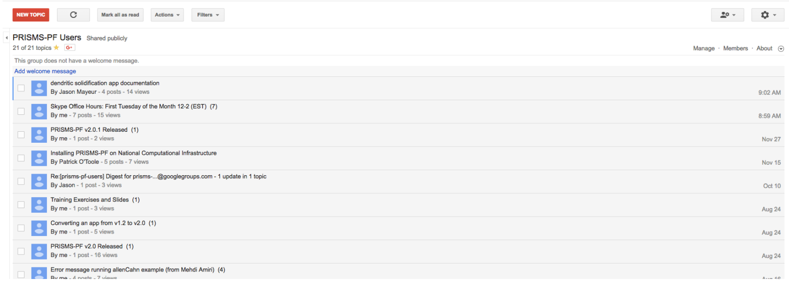
Community Support

- 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
```

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