

# PRISMS-PF Training

## Updated Training Materials / VM setup

[https://github.com/prisms-center/PRISMS-PF\\_Training\\_Materials](https://github.com/prisms-center/PRISMS-PF_Training_Materials)

Frequently used Unix/PRISMS-PF commands and links to resources

Switch to this branch

ICME\_2021 3 branches 0 tags

This branch is 5 commits ahead of master.

Contribute

Go to file Add file Code

David Montiel and David Montiel updated training handout 0d7a879 on Oct 7, 2021 20 commits

| File                           | Description                | Last Commit  |
|--------------------------------|----------------------------|--------------|
| .gitignore                     | ICME Materials first draft | 7 months ago |
| Cheat_Sheet.pdf                | ICME Materials first draft | 7 months ago |
| PRISMS-PF_Exercises.pdf        | ICME Materials first draft | 7 months ago |
| PRISMS-PF_Training_Handout.pdf | updated training handout   | 7 months ago |
| README.md                      | Edited README file         | 7 months ago |

[https://www.tms.org/portal/Meetings\\_\\_Events/2022/ICME2022/ICME2022\\_Workshop\\_Instructions.aspx](https://www.tms.org/portal/Meetings__Events/2022/ICME2022/ICME2022_Workshop_Instructions.aspx)

**Step 1: Install Virtual Box**

**Step 2: Download the ICME 2022 Workshop Tutorials Virtual Box Image**

**Step 4: Launch VirtualBox and Import the VM Image**

**Step 4: Start the VM Image**

user: icme  
password: workshop2021



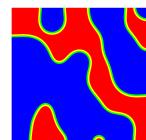
U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRISMS



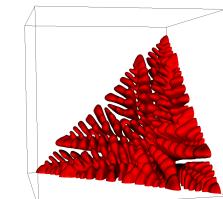
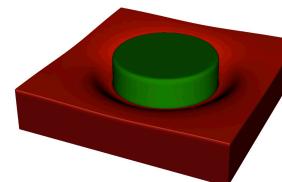
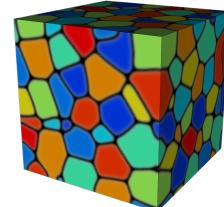
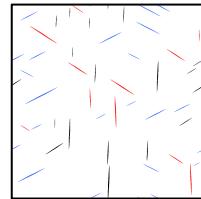
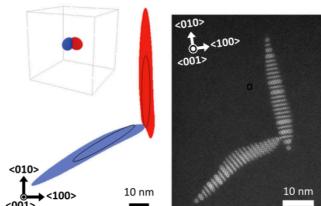
# PRISMS-PF

An Open-Source Phase Field Modeling Framework

## PRISMS-PF Training Session

David Montiel

*Department of Materials Science and Engineering, University of Michigan*



# Table of Contents

|  |    |
|--|----|
| Introduction to the Phase Field Method ..... | 4  |
| PRISMS-PF – Overview and Features .....      | 23 |
| PRISMS-PF – Code Structure .....             | 34 |
| Interactive Session .....                    | 37 |
| Results Analysis and Visualization .....     | 44 |
| Postprocessing Scripts .....                 | 47 |
| Application Development .....                | 49 |
| Materials Commons .....                      | 54 |



# The Phase-Field Method

4



U.S. DEPARTMENT OF  
**ENERGY**

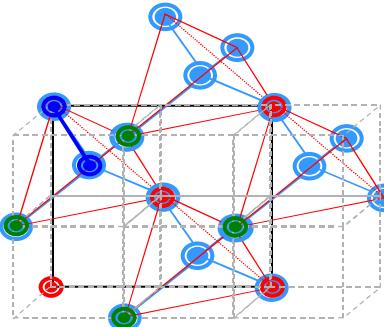
Office of  
Science



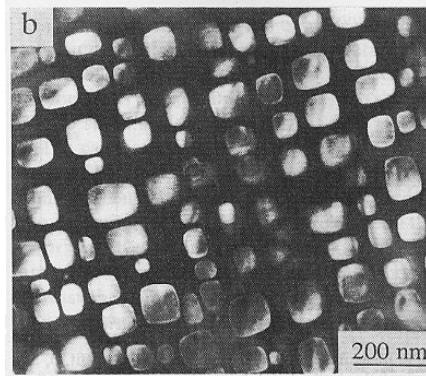
Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

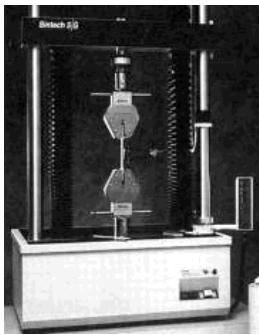
# Motivation - The Paradigm of Materials Science



Processing



Structure



Properties

Fahrmann, et al., Acta Met. 45,1007 (1995)  
Ni<sub>3</sub>Al precipitates in a Ni-Al matrix



Performance

5



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

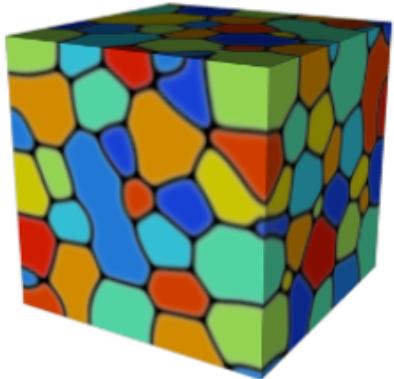
# Phase-Field Modeling

- In essence, phase field modeling is a method to simulate microstructure evolution, and in particular the evolution of interfaces and compositions of different constituents in a material.
- These simulations help us to understand the underlying mechanisms that give rise to different microstructures
- Being able to predict microstructure evolution is a powerful tool in the design processes that will give us a set of desired properties and performance

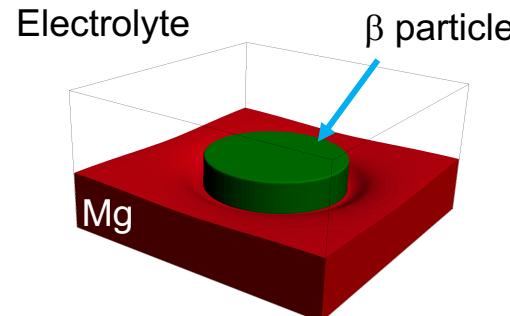


# Examples of Phase Field Applications – Simulations Using PRISMS-PF

## Grain growth in a polycrystal



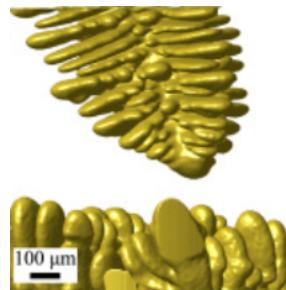
## Migro-galvanic corrosion on the surface of a Mg Alloy



## Alloy solidification

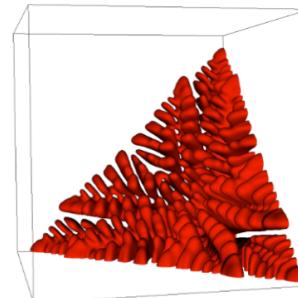
### **Experiment:**

Dendrites formed during solidification of an Al-Cu alloy

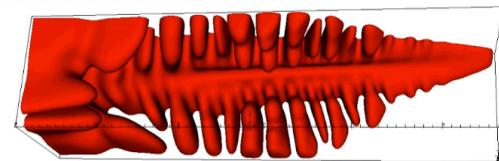


Elder et al., Scripta Mater. (2020)

### **Simulation - Equiaxed**



### **Simulation - Directional**



Zhenjie Yao et al. In preparation (2021)

7



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

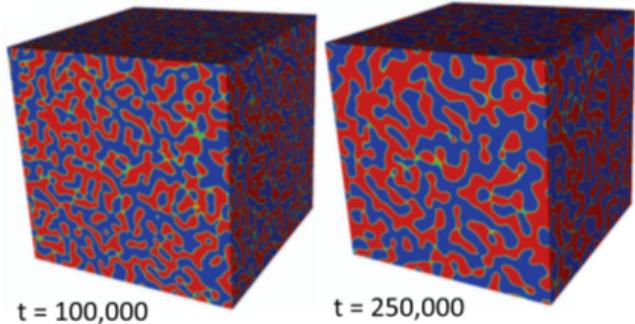


Center for Predictive Integrated  
Structural Materials Science

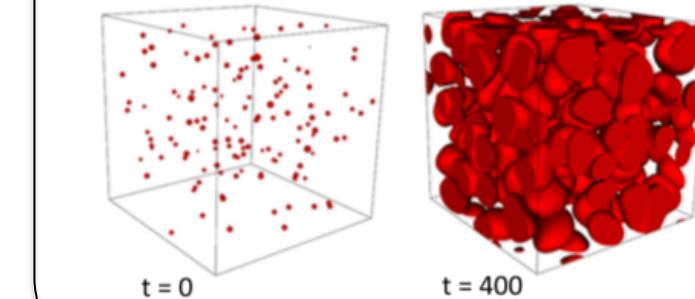
**PRISMS**

# Examples of Phase Field Applications – Simulations Using PRISMS-PF (II)

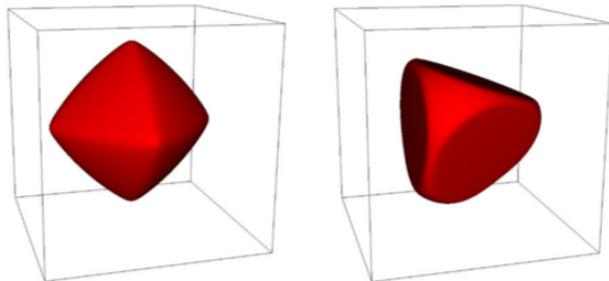
## Spinodal decomposition and coarsening



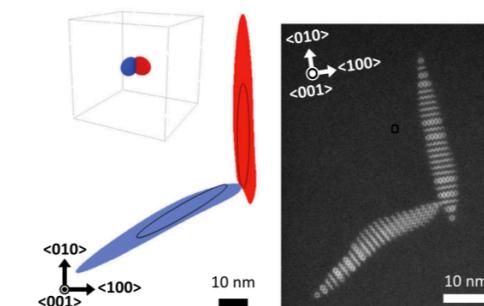
## Nucleation and growth of particles



## Evolution of particles with strong interfacial anisotropy



## Interaction between Mg-Nd alloy precipitates



DeWitt, S. et al., Acta Mater. 136, 378–389 (2017)



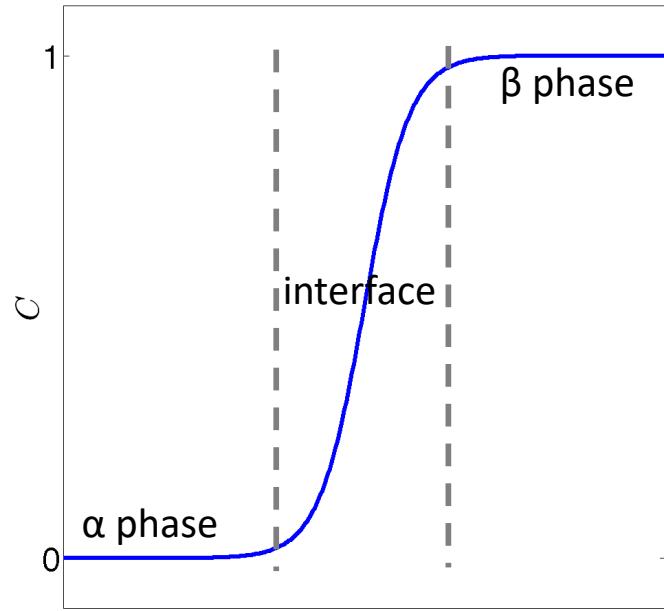
# The Idea Behind the Phase-Field Model

- To model microstructure evolution, we need a method to describe and track interfaces
- If we treat interfaces as sharp 2D surfaces, the interfacial location must be tracked by markers. This is called the sharp interface model
- Challenging free boundary problem - mathematically and computationally intensive.



# The Idea Behind the Phase-Field Model

- The phase-field model considers interfaces to be regions that have a finite volume where bulk quantities (order parameters) change gradually.
- Bulk quantities such as concentration are governed by partial differential equations (PDEs).
- Interfaces are identified as the regions where bulk quantities take their interfacial values
- The same PDEs can be applied everywhere (including the interface regions) thereby eliminating the need to explicitly track the interface



10



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

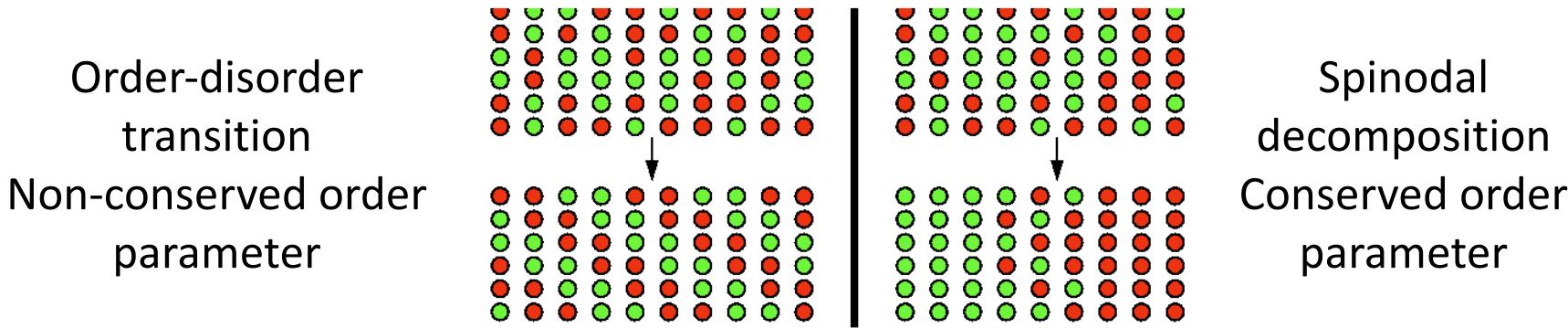


Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

# Order Parameter

- The evolution of interfaces is characterized using order parameters
- An **order parameter** is a continuous field that characterizes each phase
- Each order parameter has a governing equation (PDE) associated with it.
- Order parameters can be classified as conserved or non conserved



11



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

# What Determines Microstructure Evolution?

## Thermodynamics and Kinetics (Dynamics)

- The equilibrium state is the state for which the free energy is a minimum
- For bulk phases, this state can be determined using phase diagrams and is characterized by a uniform chemical potential.
- Systems featuring complex microstructures observed in materials are not in thermodynamic equilibrium
- However, their evolution is still driven by the minimization of the free energy
- The driving force for phase transformation is proportional to the gradient of the chemical potential, but there is kinetic factor that determines how fast this transformation occurs.



# Formulation

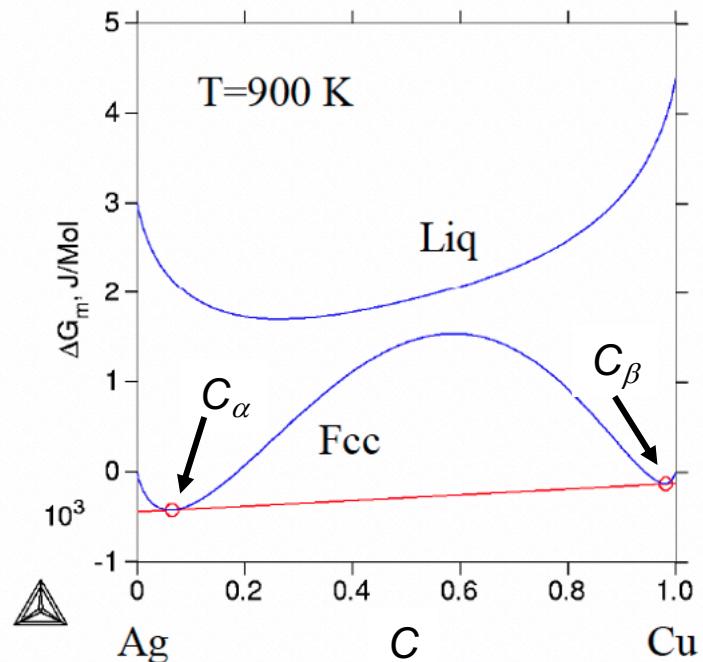
- For example, in a binary system, for which different phases are characterized by different equilibrium concentrations, we can define an order parameter,  $\phi$ , that depends linearly on the local concentration:

$$\phi = \frac{C - C_\alpha}{C_\beta - C_\alpha} \quad C = \text{concentration}$$

- Let  $F\{\phi(r)\}$  be the free energy (we will define this later) of the system which includes bulk and interfacial contributions.

- Chemical potential:  $\mu = \frac{\delta F}{\delta \phi}$

Free energy curves at fixed temperature for an Ag-Cu alloy



# Cahn-Hilliard Dynamics

Flux:

$$J = -M \nabla \mu$$

Mass conservation:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot J$$

Chemical potential:

$$\mu = \frac{\delta F}{\delta \phi}$$

Combine all:

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

The Cahn-Hilliard equation  
(general form)



This equation drives the system towards an energy minimum under the ***constraint of local mass conservation***

The dynamics describe the evolution of the system from a given initial state. The **Cahn-Hilliard** equation needs to be solved numerically in space and time.



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

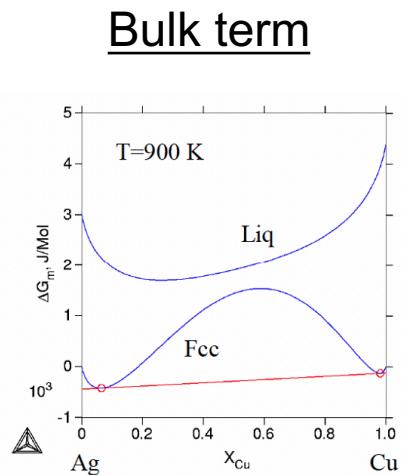
PRiSMS

# Free Energy

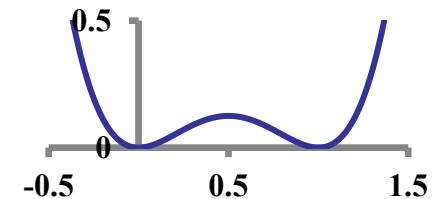
The simplest form of the free energy accounts for two contributions: a “**bulk**” term, which can be constructed from thermodynamic free energies, and a “**gradient energy**” term that account the cost of for inhomogeneities in the system (interfaces)

$$F = \int_{\Omega} \left[ f(\phi) + \frac{1}{2} K |\nabla \phi|^2 \right] dV$$

bulk term                          gradient term



**Approximation:**  
“Double well” free energy with minima at  $\phi = 0$  and  $\phi = 1$



$$f(\phi) = W\phi^2(1 - \phi)^2$$

# Governing Equation

## Free energy

$$F = \int_{\Omega} \left[ f(\phi) + \frac{1}{2} K |\nabla \phi|^2 \right] dV$$

$$f(\phi) = W\phi^2(1 - \phi)^2$$

## Chemical potential (Variational derivative of $F$ )

$$\mu = \frac{\delta F}{\delta \phi} = 2W\phi(\phi - 1)(2\phi - 1) - K\nabla^2\phi$$

...back to CH equation (with constant mobility)

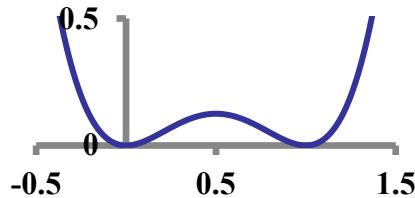
$$\frac{\partial \phi}{\partial t} = \nabla \cdot (M \nabla \mu) = M \nabla^2 [2W\phi(\phi - 1)(2\phi - 1) - K\nabla^2\phi]$$

# Allen-Cahn Dynamics

Free energy functional

$$F = \int_{\Omega} \left[ f(\phi) + \frac{1}{2} K |\nabla \phi|^2 \right] dV$$

$$f(\phi) = W\phi^2(1-\phi)^2$$



“Double well” free energy with minima at  $\phi=0$  and  $\phi=1$

Dynamics  $\frac{\partial \phi}{\partial t} = -M \frac{\delta F}{\delta \phi} = -M \left[ 2W\phi(\phi-1)(2\phi-1) - K\nabla^2\phi \right]$

Allen-Cahn dynamics are similar to Cahn-Hilliard except the system is driven towards an energy minimum **a *without a local conservation constraint*.**



# Finite Element Method

Main ideas:

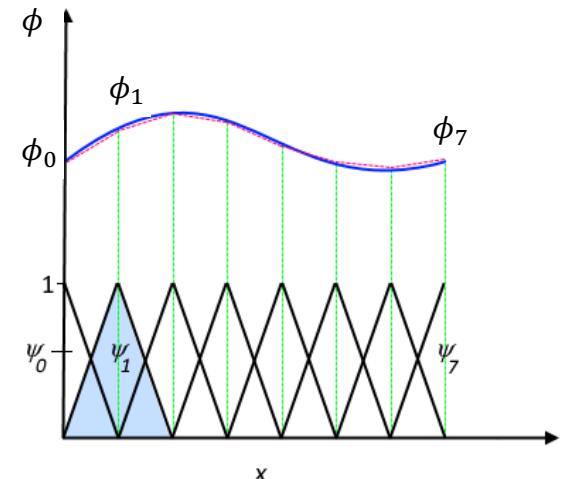
- The solution  $\phi_s(\vec{r})$  of a PDE, is approximated as a weighted sum of local basis functions

$$\phi_s(\vec{r}) \simeq \sum_{j=0}^N U_j \psi_j(\vec{r}) \quad (1)$$

- The PDE is reformulated in its ‘weak’ form. This entails multiplying both sides by an arbitrary test function,  $\omega$

$$\int \omega \frac{\partial \phi}{\partial t} dV = \int \omega \{-M[2W\phi(\phi - 1)(2\phi - 1) - K\nabla^2\phi]\} dV$$

- If  $\phi_s$  is the solution of the original PDE, then the integral above must be true for *any* test function.
- The weak form of the equation becomes  $N$  equations by using each of the  $\psi_j(\vec{r})$  as test functions. The function  $\phi_s$  is replaced by its approximation (Equation 1)
- This will result in a series of  $N$  equations, with  $N$  unknowns. The only terms that remain on the integrals are the  $\psi_j$  functions or products between these functions and their derivatives, which can be computed in advance (**Note: we also need boundary conditions**)



# Time discretization

- In order to calculate the evolution of the fields we also need to **discretize in time**
- The simplest approach is to approximate the time derivative as

$$\frac{\partial}{\partial t} \phi(\vec{r}, t^n) \simeq \frac{\phi(\vec{r}, t^n + \Delta t) - \phi(\vec{r}, t^n)}{\Delta t}$$

- If we assume we know the **initial conditions**,  $\phi(\vec{r}, 0)$  we can use the approximation to the time derivative to calculate all subsequent steps (using the spatial part of the PDE as the left-hand side). The time discretized version of the weak form of the PDE becomes

$$\int \omega \phi^{n+1} dV = \int \omega \{ \phi^n - M \Delta t [2W \phi^n (\phi^n - 1)(2\phi^n - 1) - K \nabla^2 \phi^n] \} dV$$

where  $\phi^n = \phi(t^n)$  and  $\phi^{n+1} = \phi(t^n + \Delta t)$

# Writing Equations in the Weak Form

- In order to be able to use piece-wise linear basis functions, we need to write the weak form of equation into a form that only contains first order derivatives of  $\phi$ .

$$\int \omega \phi^{n+1} dV = \int \omega \{ \phi^n - M\Delta t [2W\phi^n(\phi^n - 1)(2\phi^n - 1) - K\nabla^2\phi^n] \} dV$$

First, we **split the integral**, isolating the term that contains a spatial derivative

$$\int \omega \phi^{n+1} dV = \int \omega \{ \phi^n - M\Delta t [2W\phi^n(\phi^n - 1)(2\phi^n - 1)] \} dV + M\Delta t K \int \omega \nabla^2 \phi^n dV$$

Then we **integrate** the second integral **by parts** [  $\omega \nabla^2 \phi = \nabla \cdot (\omega \nabla \phi) - \nabla \omega \cdot \nabla \phi$  ]

$$\int \omega \nabla^2 \phi dV = \int \nabla \cdot (\omega \nabla \phi) dV - \int \nabla \omega \cdot \nabla \phi dV$$

The **divergence theorem** (we assume that  $\omega$  is zero at the boundaries) gives us

$$\int_{\Omega} \nabla \cdot (\omega \nabla \phi) dV = \int_{\delta\Omega} \omega \nabla \phi \cdot \hat{\mathbf{n}} dS = 0$$

20

# Writing Equations in the Weak Form

Finally, we get

$$\int \omega \phi^{n+1} dV = \underbrace{\int \omega \{ \phi^n - M\Delta t [2W\phi^n(\phi^n - 1)(2\phi^n - 1)] \} dV}_{\text{'value' term}} + \underbrace{\int \nabla \omega \cdot \{-M\Delta t K \nabla \phi^n\} dV}_{\text{'gradient' term}}$$

- The ‘value’ and the ‘gradient’ terms are the inputs that PRISMS-PF needs, in order to solve time-dependent PDEs.



# Remarks

- Most phase field formulations employ either the Cahn-Hilliard or Allen-Cahn dynamics equation (or a combination of both)
- Depending on the system we want to study, we can incorporate different “physics” into the free energy (e.g., elasticity, interaction between different order parameters, interaction with an external electric or magnetic field, etc.)
- Many applications of the phase field model require multiple orders parameter that need to be solved simultaneously



# The PRISMS-PF Framework – Overview and Features

23



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

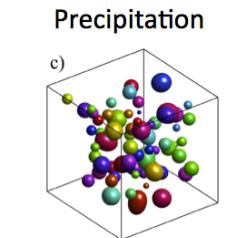


Center for Predictive Integrated  
Structural Materials Science

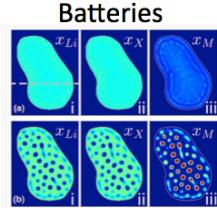
**PRiSMS**

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

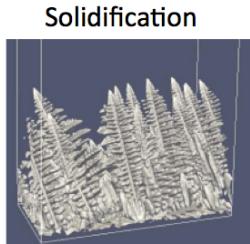
1. Wide diversity of models and coupled physics makes code reuse difficult



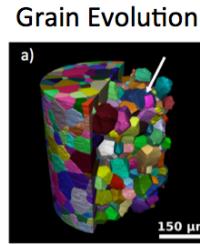
Poulsen, Voorhees, Acta Materialia (2016)



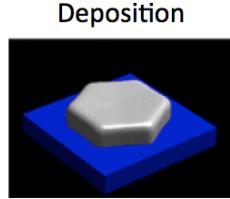
Yu, et al. J. Phase Equil. Diff. (2016)



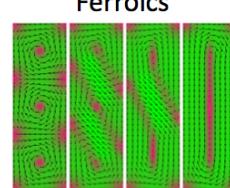
Shimokawabe, et al., SC '11 (2011)



McKenna, et al. Acta Materialia (2014)



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



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

No “typical” governing equations

Large variety of formulations and terms



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Motivation:

## What's hard about writing a phase field code?

2. Simulating large, physically representative systems is computationally intensive



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

Simulations are often done in 2D  
for tractability

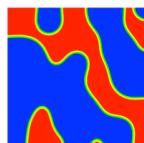
Physical fidelity requires strong  
numerical performance



# 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 numerical implementation.
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 Phase-Field Modeling Framework

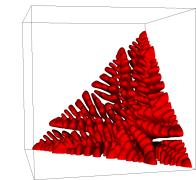
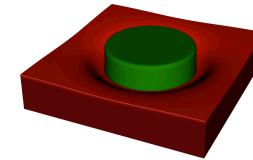
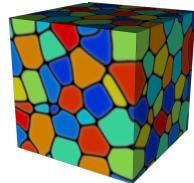
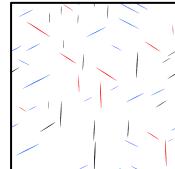
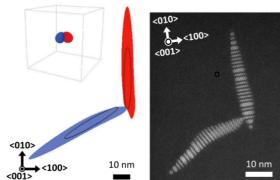
## Advanced capabilities

- Matrix-free finite element approach
- Solution of an arbitrary number of coupled PDEs
  - Higher-order elements
  - Multi-level parallelism
  - Adaptive meshing
  - Explicit nucleus placement
  - Grain-remapping
  - Newton/Picard nonlinear solver
- Implicit time integration capability <- Coming soon!

## Functionalities / Ease of Use

- Simple interface
- Detailed online user guide
- 27 pre-built applications
- Simple Docker-based installation
- nanoHUB tool w/ GUI for educational use
- Integrated with Materials Commons
- Postprocessing scripts for results analysis
- YouTube video tutorials
- Virtual Machine <- Coming soon!

DeWitt et al., *npj Comput. Mater.* 6, 29 (2020)



27



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



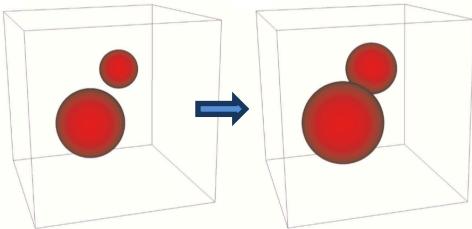
Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

# Performance of PRISMS-PF

## Comparison to finite difference code

- Two growing particles in 3D



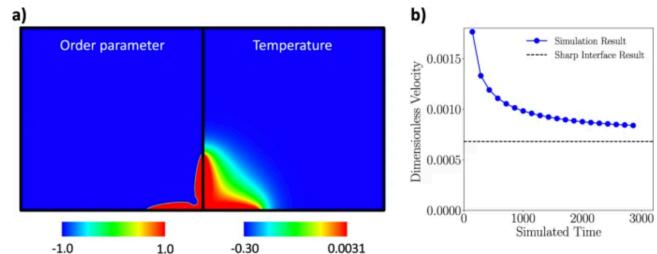
| Element/Mesh Type | Speedup vs. FD* |
|-------------------|-----------------|
| Linear/Regular    | 0.028           |
| Quadratic/Regular | 0.90            |
| Cubic/Regular     | 1.9             |
| Cubic/Adaptive    | 12.0            |

Comparison is at the same error with respect to a highly-resolved simulation in time in space

[DeWitt et al., npj Comput. Mater. 6, 29 (2020)]

## Comparison to other Open Source Codes

- Solidification benchmark problem  
(<https://pages.nist.gov/pfhub/benchmarks/benchmark3.ipynb/>)



- PRISMS-PF performance was compared to MOOSE, AMPE and FiPy
- The PRISMS-PF calculations required **three orders of magnitude fewer** normalized core hours than **AMPE** and **FiPy**, while having similar or lower error.
- The fastest calculations using PRISMS-PF and **MOOSE** have **similar computational cost** and tip velocity error.

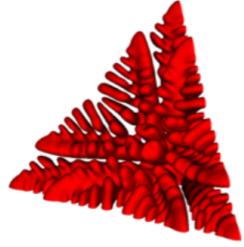
# New and Upcoming Features - Built-in Apps

## Recent

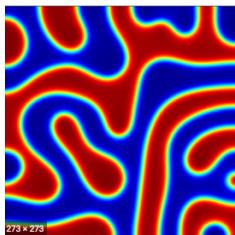
Corrosion



Alloy Solidification

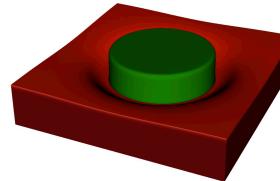


Spinodal Decomposition

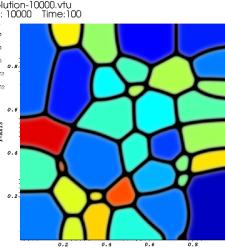


## Coming soon

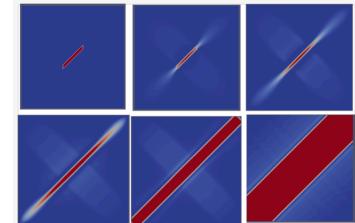
Microgalvanic Corrosion



Grain Growth (with Stored Energy)



Martensitic Transformation\*



\* Pushkar Pandit (IIT Hyderabad, India) and Bonnie Whitney (WPI)

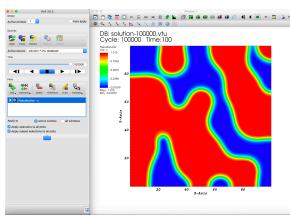


# New and Upcoming Features - YouTube Video Tutorials



PRISMS Center  
219 subscribers

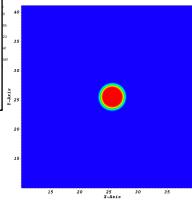
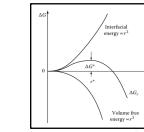
Installing, running, and visualizing results



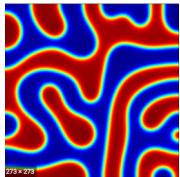
Installation of prerequisites



Nucleation and Growth



Spinodal Decomposition



Coming soon:

Advanced Application



30



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



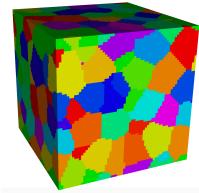
Center for Predictive Integrated  
Structural Materials Science

**PRISMS**

# New and Upcoming Features - Integration Tools

## Integration with DREAM.3D

DREAM.3D

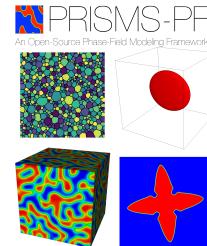


**PRISMS-PF**  
An Open-Source Phase-Field Modeling Framework

Recently upgraded script to import microstructure from DREAM.3D (using the VisIt CLI)

## Postprocessing scripts

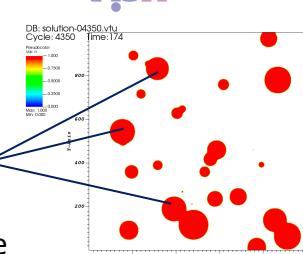
New scripts for results analysis of PRISMS-PF simulations:



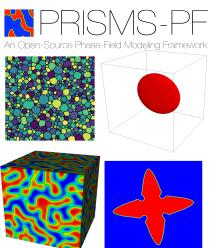
- Phase fraction
- Interface area
- Domain count

... more to come

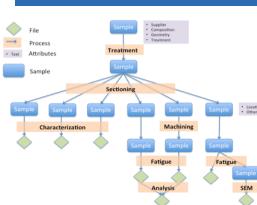
visIt



## Materials Commons CLI



Materials Commons 2.0



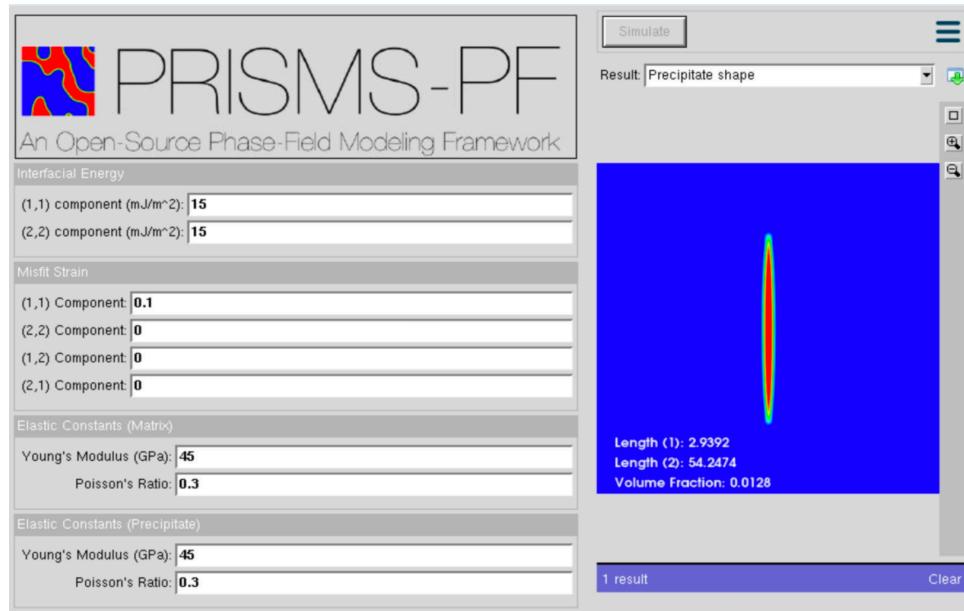
The newly-developed CLI for Materials Commons will facilitate creating scripts to automate the creation of projects, upload data, create and publish datasets, and edit communities of practice



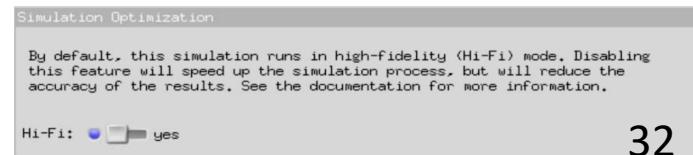
# PRISMS-PF nanoHUB Module

<https://nanohub.org/resources/prismspfmisfit>

## PRISMS-PF: Equilibrium Shape for a Misfitting Precipitate\*



- 2D equilibrium shape of a precipitate taking into account the effects of interfacial and strain energy
- Targeted for classroom use
- Recently optimized to include a low-fidelity option which produces a faster, preliminary calculation



\*S. DeWitt (UM), N. Shuman (UC Davis), and S. Gentry (UC Davis)



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

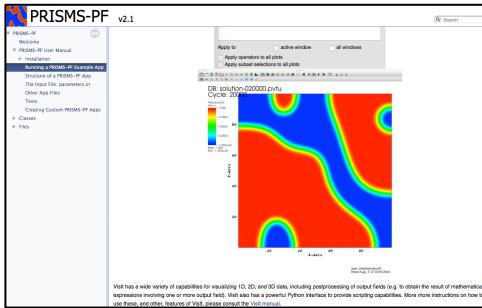


Center for Predictive Integrated  
Structural Materials Science

**PRiSMS**

# Community support

## Online user manual

A screenshot of the PRISMS-PF v2.1 online user manual. It features several sections of text and code. At the top, there's a heading for 'Other App Files' and a section titled 'Tests'. Below that is a large block of mathematical text with equations involving integrals and variables like  $\Delta M$ ,  $f_1$ ,  $f_2$ , and  $V_{\text{eff}}$ . Further down, there's a code snippet in C++ for the 'loadVariableAttributes' function, which defines variables and their attributes. The text explains that each term in the governing equation is marked with an underbrace, and the terms multiplied by the gradient of the test function are referred to as the gradient terms.

Online message board  
(> 100 registered users)

☆ PRISMS-PF Users 99 members 1–30 of 144 < >

Welcome to the PRISMS-PF user forum! This is the place to ask any questions that you have regarding PRISMS-PF ranging from whether it is right for your work, installation, creating new applications, details about the internal workings of the code, or future directions. We work to make sure that postings are responded to within a few business hours. For additional help with PRISMS-PF, we also hold office hours on Skype on the first Tuesday of every month. See the "Skype Office Hours" thread for more details.

- Bryan Kinzer, dmontiel 8 Help with Loading Initial Conditions in Dream3D Example/Trying to add Particulates – Thanks... Feb 26 ☆
- dmontiel Updated vtk conversion scripts within the grainGrowth\_dream3d application – Hello everyone, ... Feb 25 ☆
- Bryan Kinzer, dmontiel 3 CMAKE\_PREFIX\_PATH Not Found Question – Hey David, After clearing out the files when I ran ... Feb 19 ☆
- Bryan Kinzer, ... dmontiel 5 PRISMS-PF is running super slow on Computing Cluster – Hello, Bryan One source of poor per... Feb 17 ☆

## Phase-Field Community of Practice (NEW)

Community: Phase-Field Simulations of Microstructure Evolution Edit Community

Overview Datasets Files Links

Name Phase-Field Simulations of Microstructure Evolution

Description A community of practice for sharing phase-field simulation results of microstructure evolution.

DONE

33



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for Predictive Integrated Structural Materials Science

PRiSMS

# The PRISMS-PF Framework – Code Structure

34



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

# Three Types of PRISMS-PF Users

## 1. Using pre-built applications

- Modifying simulation parameters, boundary conditions, initial conditions.
- No C++ or deal.II knowledge required

## 2. Creating new applications

- Implementing governing equations and defining model parameters
- Basic C++ knowledge required
- No deal.II knowledge required

## 3. Developing new features and functionalities

- Editing the core library
- Knowledge of C++ and deal.II required



# 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



# Interactive Session

37



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Useful Commands in a Linux terminal

- ls      Used to show the contents of the current directory
- pwd    Prints the current working directory (where you are in the file structure)
- cd      Change directory (cd .. to go up one directory, cd [insert directory here] to go into a directory)
- cp      Copy a file or directory (cp [file to be copied] [where to copy it to], cp -r [directory to be copied] [where to copy it to])
- mv      Move a file or directory (mv [file to be moved] [where to move it to], mv -r [directory to be moved] [where to move it to])
- rm      Delete a file or directory (rm [file to be deleted], rm -r [directory to be moved])
- mkdir   Create a new directory (mkdir [directory name])
  
- emacs [filename] -   Open a file with text editor Emacs
  
- vi [filename] -   Open a file with text editor vi



# Prerequisites

Note: In perform the tasks in this training session you will need

- A working installation of deal.II with the MPI and p4est dependencies
- A copy of the PRISMS-PF framework downloaded from the repository
- To have the core library of PRISMS-PF precompiled

(The Virtual Machine for the ICME Workshop will already have this requirements)

39



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for Predictive Integrated  
Structural Materials Science

PRiSMS

# Structure of an application

## Code files (all applications):

- `main.cc` – Parses user input data, builds the fields, loads initial conditions, calls the solver method.
- `parameters.in` – Text file with input data: system dimensions, mesh specification, time step, boundary conditions, output format and frequency, and user-defined simulation parameters.
- `equations.cc` – Declare the expressions (weak form) for the time dependent and time-independent equations
- `ICs_and_BCs.cc` – Set the initial conditions and non-uniform Dirichlet boundary conditions (if applicable).
- `customPDE.h` – Declare user-defined simulation parameters defined in `parameters.in`. Declare application-specific methods, or override methods from the core library.

# Structure of an application

## Code files (optional):

- `postprocess.cc` – Defines fields needed for analysis of the results, but not necessarily for the simulation, e.g., the free energy density.
- `nucleation.cc` – Defines the method to calculate the probability of nucleation (only needed when nucleation is enabled).



# Compiling the Allen Cahn application

- Open a Unix shell terminal
- Go to the PRISMS-PF installation directory (phaseField). For example, in the Virtual Machine (VM):

```
$ cd ~/tools/phaseField/
```

This is the core library directory

- Go to the applications directory and list the contents

```
$ cd applications
```

```
$ ls -l
```

This will list all the application folders, each containing a pre-built application.

- Go to the allenCahn application folder

- Compile the application:

```
$ cmake . (don't forget the dot)
```

```
$ make release
```

You should get a few warnings but no errors. These steps will generate a series of new files and directories, including the executable, main

42



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Running the application

- Run the code using

```
$ mpirun -n 1 main
```

The code should take about a minute to run depending on your system. You may be able to speed this by setting more processors, e.g., “-n 2” or “-n 4”, depending on how many processors you allocated for the VM.

- List the documents on the directory in reverse chronological order (the most recent files should appear in the bottom)

```
$ ls -ltr
```

Notice that there are a new series of files named “solution-\* .vtu”. These files contain the simulation results. Notice also there will be a series of files that labeled “restart.\*”. These are the files PRISMS-PF needs to restart a simulation from the last checkpoint that was generated. We will not use these files in these tutorial. Finally, notice the file integratedFields.txt. This file contains the results of fields that are integrated over the whole system.

# Results Analysis and Visualization

44



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Results Analysis and Visualization (I)

- To view the file integratedFields.txt type:

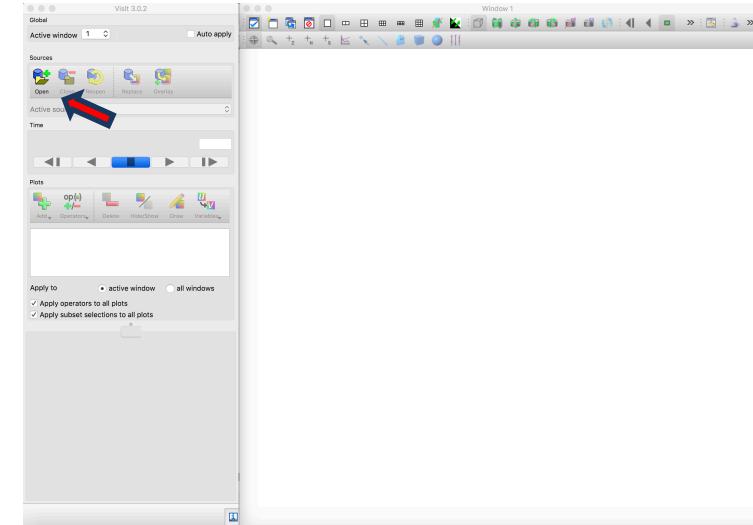
```
$ more integratedFields.txt
```

The first column shows the time and the third column shows the total energy. Note that the total energy decreases monotonically with time, which is consistent with the dynamics.

- Open VisIt by clicking on the  icon on the desktop of the VM

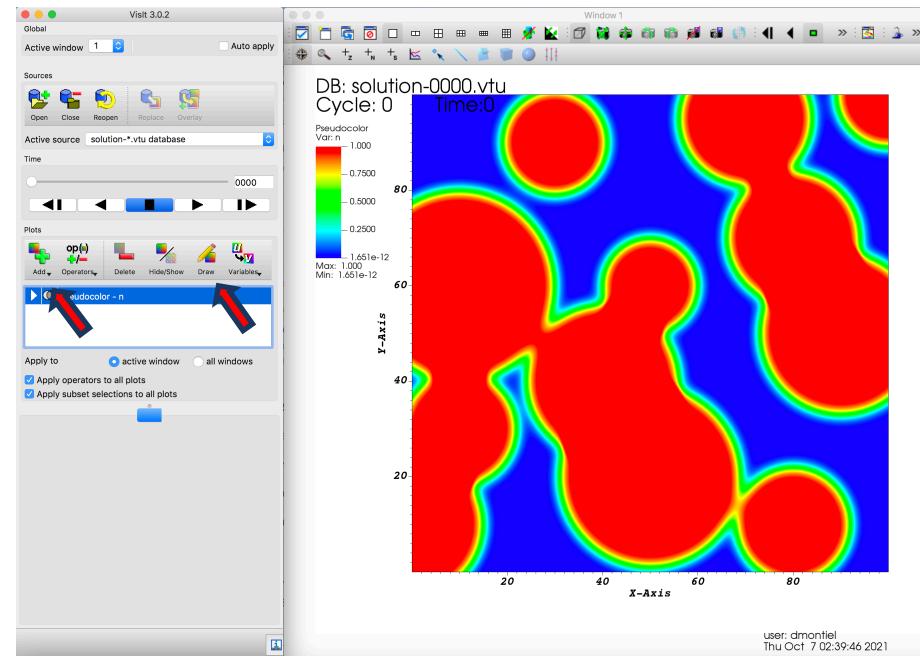
The VisIt GUI will open showing a panel and a window right next to it (see figure).

- Click on the “Open” icon to open the results files
- In the path, input the directory where the files are located (.../applications/allenCahn)
- Select the group of files “solution.vtu database”
- Click “OK” to load the files.



# Results Analysis and Visualization (II)

- On the left panel click on the “Add” icon and hover over the option “Pseudocolor” from the drop-down menu. A new sub-menu will appear with all the output fields from the simulation
- Select the field “n”. This is the order parameter.
- On the same panel, click on “Draw”. A 2D plot will appear in the window to the right showing the initial configuration of the simulation domain.
- Use the arrows below the “Time” slide bar to see each of the time frames of the simulation. Note how, as the simulation progresses, the interfaces become smoother and shorter in length. Close Visit and return to the terminal.



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Postprocessing Scripts

47



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Postprocessing Scripts

- PRISMS-PF contains a suite of postprocessing scripts in Python that calculate the commonly useful properties of the system. These scripts use the VisIt CLI. These scripts are included in the folder phaseField/postprocess\_scripts/ (see: [https://github.com/prisms-center/phaseField/tree/master/postprocess\\_scripts](https://github.com/prisms-center/phaseField/tree/master/postprocess_scripts))
  - **domain\_count.py**: Counts the number of independent domains of the phase characterized by n=1
  - **interface\_area.py**: Calculates the total interfacial area (in 3D) or length (in 2D) between two phases
  - **phase\_fraction.py**: Calculates the phase fraction as the ratio of the volume (area) of the phase defined by n=1 over the volume (area) of the whole system.
- Within the allenCahn application directory, copy all of the scripts into the folder  
\$ cp ../../postprocess\_scripts/\*.py .
- Run each script, one by one as  
\$ visit -cli -py2to3 -s <script.py>  
(Replace <script.py> by any of the script names mentioned above. Type “quit()” or Ctrl-C if a script gets stuck in the Python session).
- Inspect the new output files generated by the scripts

# Application Development

49



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

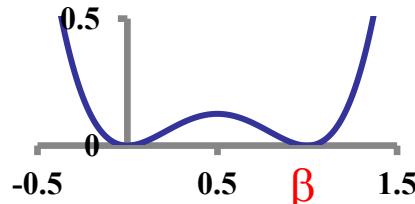
# Example: Adding an undercooling term to the Allen-Cahn application

## Free energy

$$F = \int_{\Omega} \left[ f(\phi) + \frac{1}{2} K |\nabla \phi|^2 \right] dV$$

- **Symmetric** double well: phases  $\alpha$  and  $\beta$  are equally stable

$$f(\phi) = \phi^2(1 - \phi)^2$$



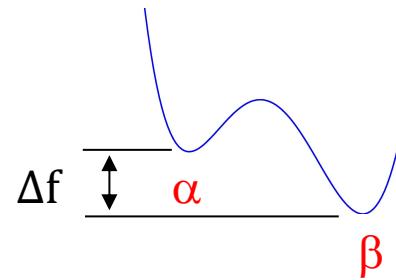
- System domains evolves in such way to minimize interface area until one of the phases disappears

- **Asymmetric** double well: one of the phases becomes more stable than the other

$$f(\phi) = \phi^2(1 - \phi)^2 - \Delta f p(\phi)$$

$\Delta f$  is a constant factor  
 $p(\phi)$  is an interpolation function

$$p(\phi) = \phi^3(10 - 15\phi + 6\phi^2)$$



- Phase  $\beta$  has lower energy than phase  $\alpha$



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for Predictive Integrated  
Structural Materials Science

**PRiSMS**

# Dynamics

## New governing equation

$$\frac{\partial \phi}{\partial t} = -M \frac{\delta F}{\delta \phi}$$

$$\frac{\delta F}{\delta \phi} = \frac{\partial f}{\partial \phi} - K \nabla^2 \phi$$

- We need to include a new term into  $\partial f / \partial \phi$

$$\frac{\partial f}{\partial \phi} = 2W\phi(\phi - 1)(2\phi - 1) - \text{30}\Delta f\phi^2(\phi - 1)^2$$



# Implementation (I)

1) Create a new application starting from the ***allenCahn*** application. Inside the ***applications*** folder type:

```
$ cp -r allenCahn allenCahn_undercooling
```

(you may choose any name you want for the new application)

2) Go to the **allenCahn\_undercooling** directory

3) We will modify the **parameters.prm** to include the constant value for  $\Delta f$ . We will choose  $\Delta f=0.025$

Add the following line to the end of **parameters.prm**:

```
set Model constant delf = 0.025, DOUBLE
```

4) Add a member variable with the same name to the class **customPDE**.

When the code runs it will read the text from **parameters.prm** and assign the value of 0.025 to the new variable. On line 56 of **customPDE.h** add the line:

```
double delf = userInputs.get_model_constant_double("delf");
```

# Implementation (II)

5) We are going to add the new undercooling term to the governing equation for “n” (In the code, the order parameter  $\phi$  is named “n”)

Open the file **equations.cc** and add the new term to line 50, i.e., replace line 50 with:

```
scalarvalueType fnV = 4.0*n*(n-1.0)*(n-0.5) -  
constV(30.0*delf)*n*n*(n-1.0)*(n-1.0);
```

6) Finally, add the corresponding term to the free energy calculation in **postprocess.cc**. Open the file and add the new term in line the 60

```
scalarvalueType f_tilt = -constV(delf)*(n*n*n)*(10.0-  
15.0*n+6.0*n*n);
```

Also add the contribution of `f_tilt` to `f_tot` (close to line 71) :

```
f_tot = f_chem + f_tilt + f_grad;
```

7) Save all changes and close the files.

8) Delete the file **CMakeCache.txt** (the compilation will fail otherwise)

9) Compile and run the code and visualize the results using VisIt.

How do the results differ from those of the allenCahn application?

# Materials Commons

54



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



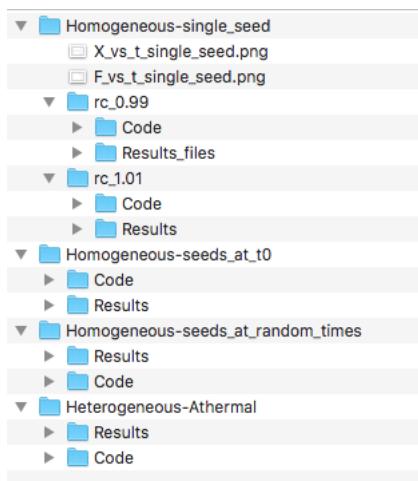
Center for PRedictive Integrated  
Structural Materials Science

PRiSMS

# Materials Commons

- Materials Commons is a site for Materials Scientists to collaborate, store and publish research

## Simulation data in a local directory



## Using the MC CLI to upload data

1) Configure remote (access to MC from your local machine)

2) Initialize project in MC

```
$ mc init
```

3) Upload all files to project

```
$ mc up -r .
```

## Simulation data in a Materials Commons project

| Name                              | Type      | Size | Actions |
|-----------------------------------|-----------|------|---------|
| Heterogeneous-Athermal            | directory | 0 B  | [trash] |
| Homogeneous-seeds_at_random_times | directory | 0 B  | [trash] |
| Homogeneous-seeds_at_t0           | directory | 0 B  | [trash] |
| Homogeneous-single_seed           | directory | 0 B  | [trash] |

<https://materials-commons.github.io/materials-commons-cli/html/index.html>

55

# Resources

Website: <https://prisms-center.github.io/phaseField>

Repository: <https://github.com/prisms-center/phaseField>

PRISMS Center YouTube Channel: <https://www.youtube.com/channel/UCZXc3007JuBCGKDcneD>

Email: [prismsphasefield.dev@umich.edu](mailto:prismsphasefield.dev@umich.edu)

## PRISMS-PF

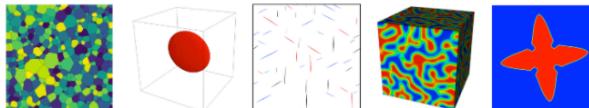
An open-source, general purpose framework for high-performance phase field modeling

GitHub Repository  
User Manual  
User Registration Link  
PRISMS-PF Forum  
YouTube Channel  
Community of Practice



### Overview

PRISMS-PF is a powerful, massively parallel finite element code for conducting phase field and other related simulations of microstructural evolution. The phase field method is commonly used for predicting the evolution of microstructures under a wide range of conditions and material systems. PRISMS-PF provides a simple interface for solving customizable systems of partial differential equations of the type commonly found in phase field models, and has 24 pre-built application modules, including for precipitate evolution, grain growth, dendritic solidification, and spinodal decomposition.



A screenshot of the GitHub repository page for "prisms-center/phaseField". The page shows the repository's structure, recent commits, releases, packages, and contributors. Key details include: 14 branches, 14 tags, 969 commits, and a latest release of Version 2.1.2 from July 31, 2019.



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for Predictive Integrated  
Structural Materials Science<sup>56</sup>

PRiSMS

# Acknowledgments



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

Award No.: DE-SC0008637



**XSEDE**

Extreme Science and Engineering  
Discovery Environment

**M** | ARC ADVANCED  
RESEARCH COMPUTING  
UNIVERSITY OF MICHIGAN



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



Center for PRedictive Integrated  
Structural Materials Science

**PRiSMS**

# Contributors and Developers



Katsuyo Thornton



John Allison



Shiva Rudraraju



Stephen DeWitt



Beck Andrews



Vishwas Goel



Yanjun Lyu



Zhenjie Yao



Bryan Kinzer



Kübra Karayağız



Bonnie Whitney



Pushkar Pandit



Thomas Flint



Xianyue Liu



Nicole Schuman



Susan Gentry

Center for Predictive Integrated  
Structural Materials Science



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

