5/20/2023

**Daniel Schwen**
Computational Methods Development Group Lead

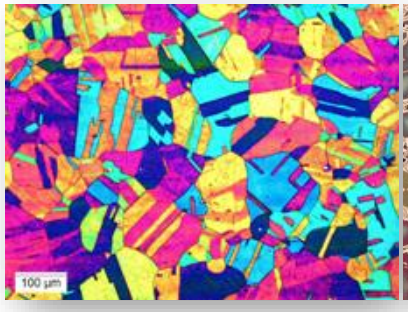
# Materials Modeling with MOOSE

## NSUF User Development Workshop on Irradiation Testing

Idaho National Laboratory

# Materials Microstructure and Properties

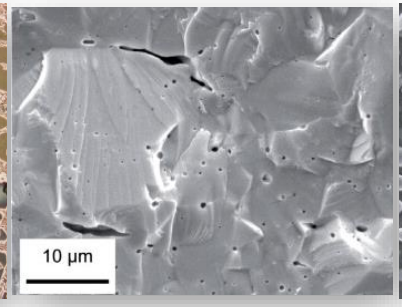
- A key objective of materials science is to understand the impact of microstructure on macroscale material behavior.



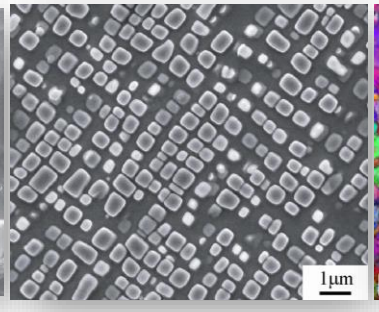

Annealed Stainless Steel



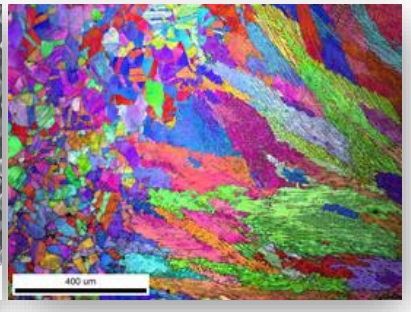
Cast Bronze




Sintered UO2

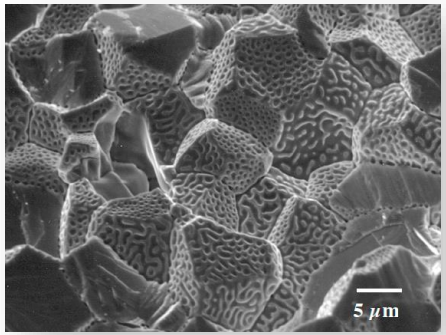



Co-Al-W Superalloy

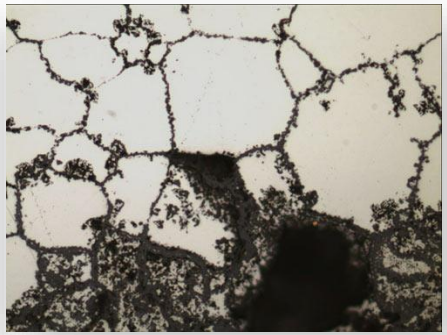



Friction stir welded stainless steel

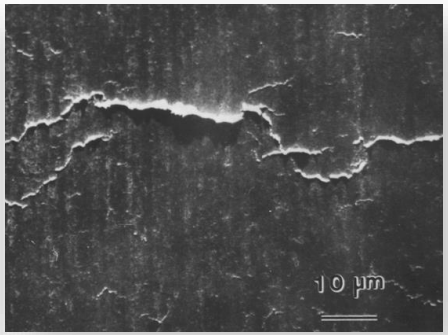
- An essential part of that is predicting the impact of microstructure evolution.



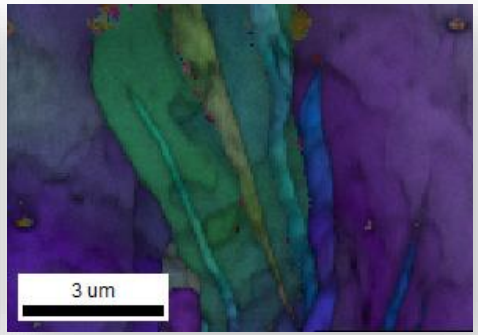

Irradiated $UO_2$ fuel



Corrosion in stainless steel




Micro-cracking in steel




Hydride in Zircaloy

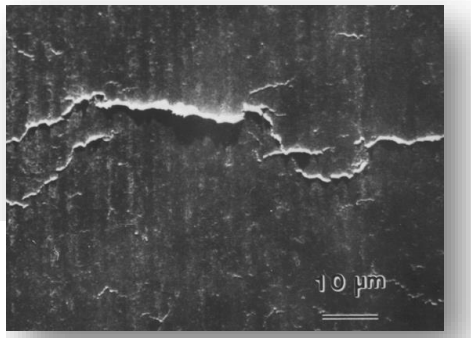
# Material Behavior is Multiphysics

- Material behavior is influenced by many different physics, for example:

## Mechanics

- Dislocations
- Cracking
- Stress-driven Diffusion

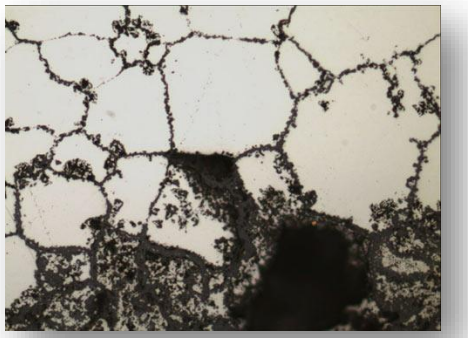
$$\nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}_0) + \mathbf{b} = \mathbf{0} \text{ in } \Omega$$
$$\mathbf{u} = \mathbf{g} \text{ in } \Gamma_{\mathbf{g}}$$
$$\boldsymbol{\sigma} \cdot \mathbf{n} = \boldsymbol{\iota} \text{ in } \Gamma_{\boldsymbol{\iota}}$$




## Chemistry

- Corrosion
- Oxidation
- Reactive transport

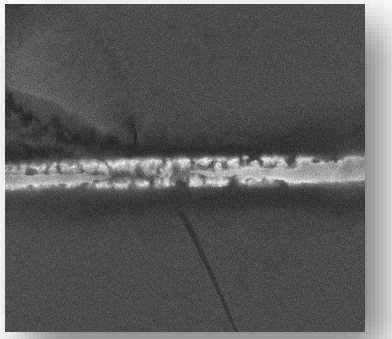
$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F(1-u),$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (F+k)v.$$



## Electricity/Magnetism

- Electromigration
- Ferroelectricity
- Ferromagnetism

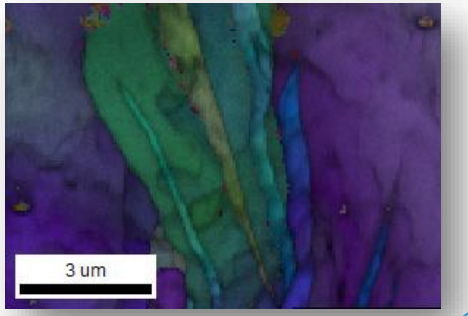
$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$



## Heat Conduction

- Species transport
- Melting
- Precipitation

$$\frac{\partial U}{\partial t} = \nabla^2 U$$



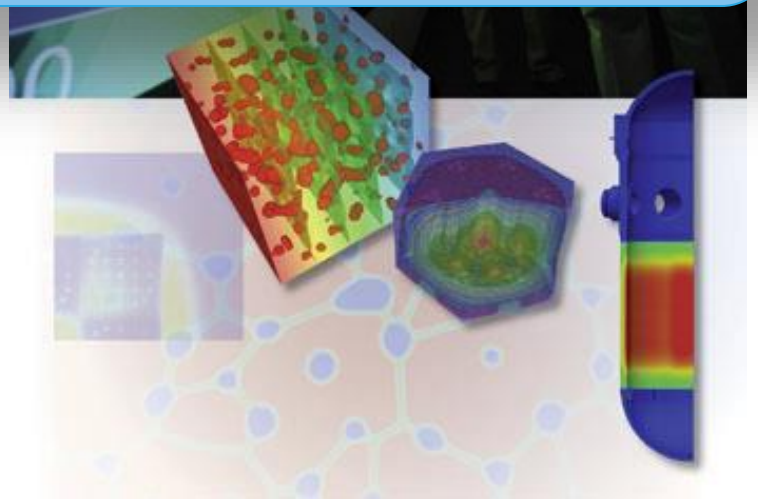
# Multiphysics Object Oriented Simulation Environment

- MOOSE is a finite element, multiphysics framework that **simplifies the development** of advanced numerical applications.
- It provides a high-level interface to **sophisticated nonlinear solvers** and **massively parallel computational capability**.
- Open Source, available at **https://mooseframework.inl.gov**
- Benchmark results at PFHub **https://pages.nist.gov/pfhub**

***MOOSE is in use across the world to solve:***

- **Phase field**
- **Solid mechanics**
- Heat conduction
- Neutronics
- Comp. fluid dynamics
- **Stochastic modeling**
- Thermal hydraulics
- Geomechanics
- Reactive transport
- Corrosion
- Crystal plasticity
- Fracture
- Porous flow
- Electromagnetics

MOOSE



# The MOOSE “Framework”

- MOOSE is not an “application”, but a “framework” that provides
  - C++ infrastructure to write your own…
    - physics (PDE weak forms)
    - Initial and boundary conditions
    - On-the-fly postprocessing (e.g. max/min concentration, precipitate counting, line scans)
  - An input file parser to combine the MOOSE C++ objects into a simulation at runtime, and add/configure
    - Variables
    - Solver and solver parameters
    - Outputs

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 10
  ny = 10
[]

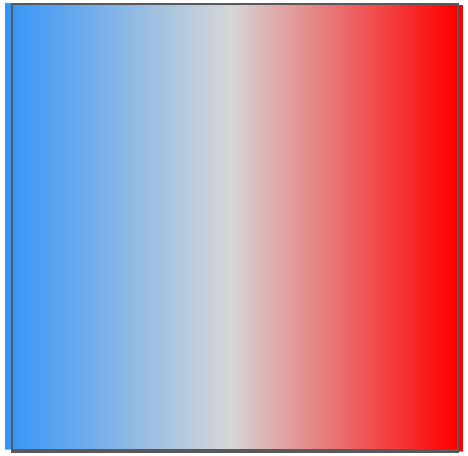
[Variables]
  [u]
  []
[]

[Kernels]
  [diff]
    type = Diffusion
    variable = u
  []
[]

[BCs]
  [left]
    type = DirichletBC
    variable = u
    boundary = left
    value = 0
  []
  [right]
    type = DirichletBC
    variable = u
    boundary = right
    value = 1
  []
[]

[Executioner]
  type = Steady
  solve_type = 'PJFNK'
  petsc_options_iname = '-pc_type -pc_hypre_type'
  petsc_options_value = 'hypre boomeramg'
[]

[Outputs]
  exodus = true
[]
```



# Writing MOOSE Code

## Implementing your physics in MOOSE

1. Derive your PDE term weak form
2. Create small boilerplate C++ .C and .h file
3. Type in weak form



$$\vec{v}\nabla u - \frac{\partial u}{\partial t} = 0$$

$$(\psi, \vec{v}\nabla u) - \left(\psi, \frac{\partial u}{\partial t}\right) = 0$$

```
_test[_i][_qp] * _v[_qp] * _grad_u[_qp];
```

```
registerMooseObject("MyAwesomeApp", ADAdvection);

InputParameters
ADAdvection::validParams()
{
 InputParameters params = ADKernel::validParams();
 params.addRequiredParam<MaterialPropertyName>("velocity_vector",
  "Advection velocity vector");
 return params;
}

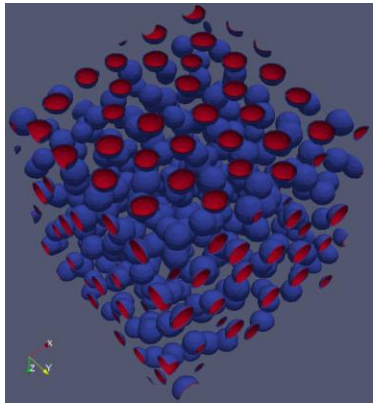
ADAdvection::ADAdvection(const InputParameters & parameters)
 : ADKernel(parameters),
   _v(getADMaterialProperty<RealVectorValue>("velocity_vector"))
{
}

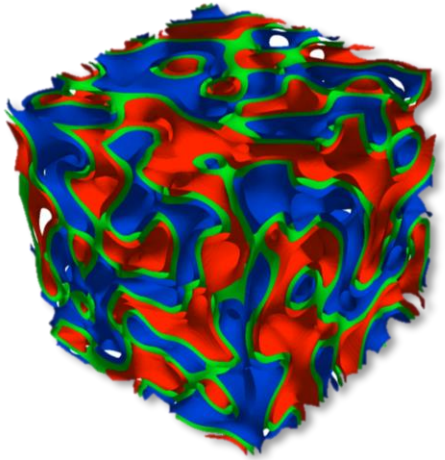
ADReal
ADAdvection ::computeQpResidual()
{
 return _test[_i][_qp] * _v[_qp] * _grad_u[_qp];
}
```

```
[Kernels]
  [convection]
    type = ADCoupledConvection
    variable = u
    velocity_vector = v
  []
  [dt]
    type = ADTimeDerivative
    variable = u
  []
[]
```

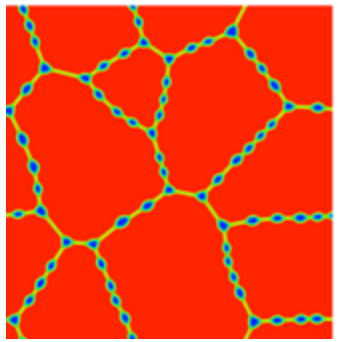
# MOOSE Phase Field Module

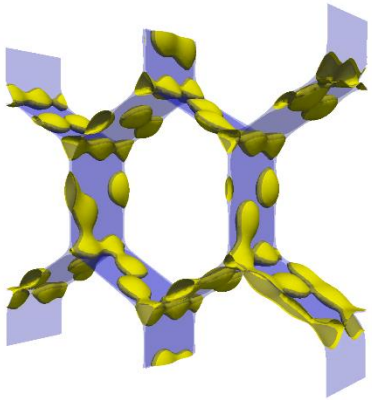
- Provides all the tools necessary to develop a massively parallel phase field code using FEM.
- Base classes for solving Cahn Hilliard equations
  - Direct solution / Split solution
  - Grand Potential, KKS
- Base classes for Allen-Cahn equations
- Grain growth model
- Grain remapping algorithm for efficient models
- Initial conditions
- Postprocessors for characterizing microstructure
- Nucleation
- CALPHAD support

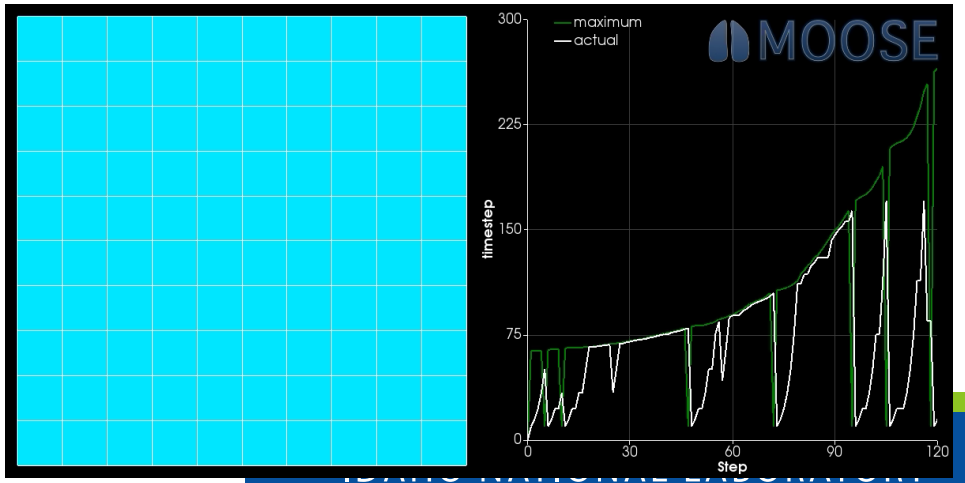






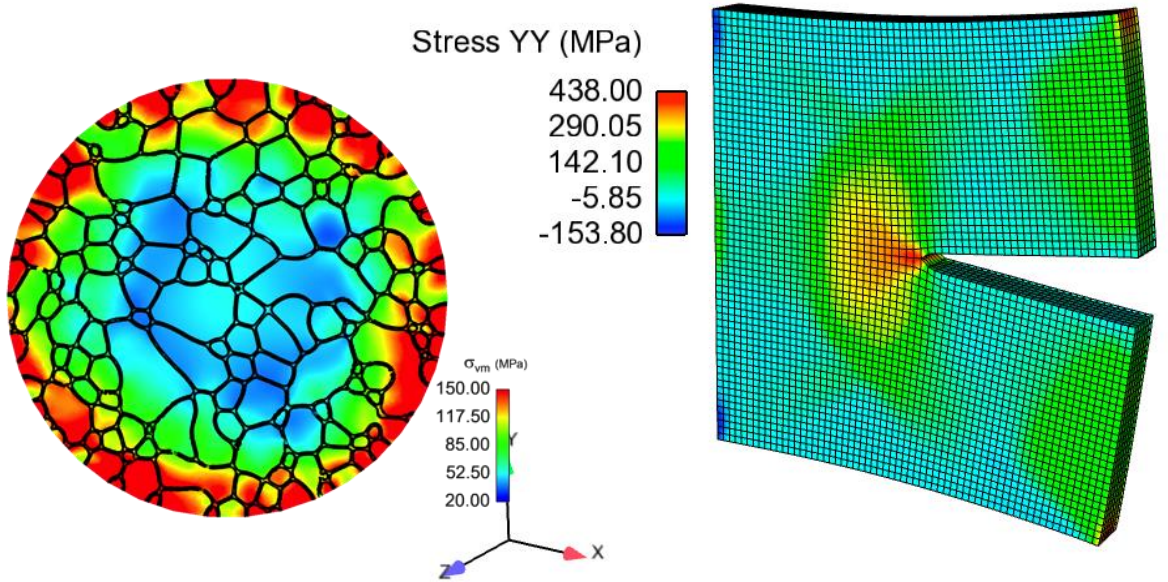


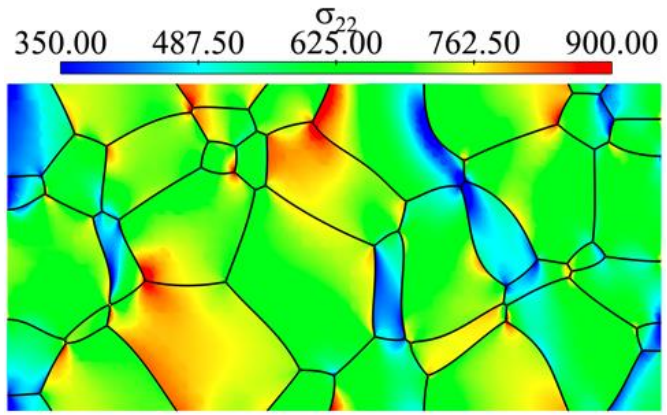


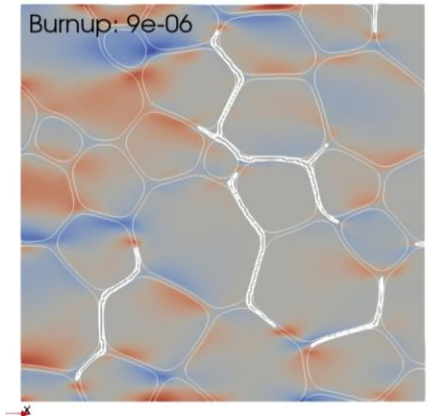


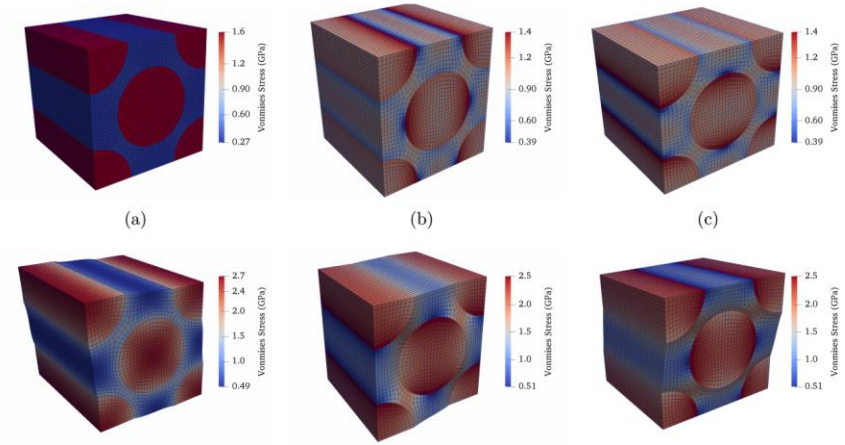
# MOOSE Solid Mechanics Module

- Provides the tools necessary for modeling mechanical deformation and stress at the mesoscale.
- Strain (small, finite, incremental)
- Eigenstrains / Eigenstresses
- Elastic stress, Modular inelastic stress system
  - Creep
  - Plasticity (J2, Crystal Plasticity)
  - Fracture
- Strain periodicity (representative volume elements)
- Fully couplable to phase field
- Coupling to *GPU enabled* constitutive model library NEML2



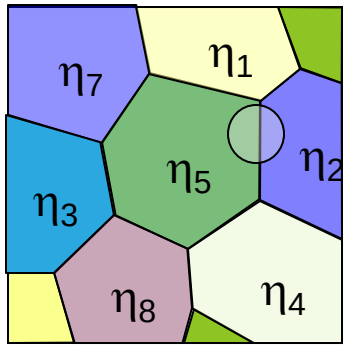



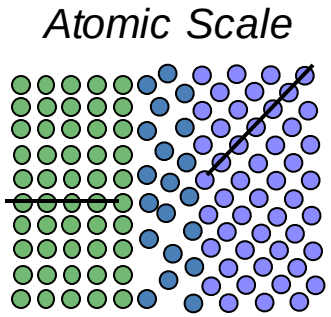



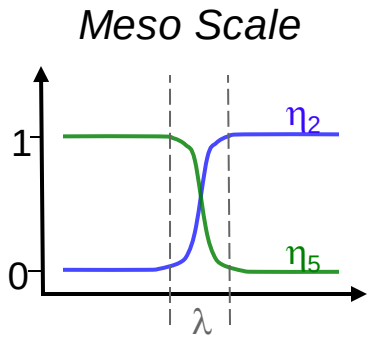



# The Phase Field Method

- Microstructure described by a set of continuous variables…
  - Non-Conserved Order Parameters

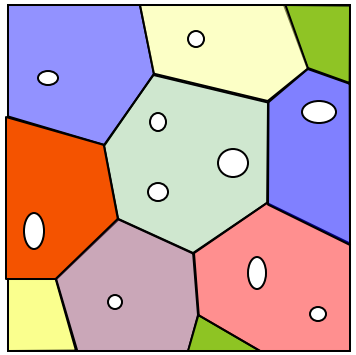


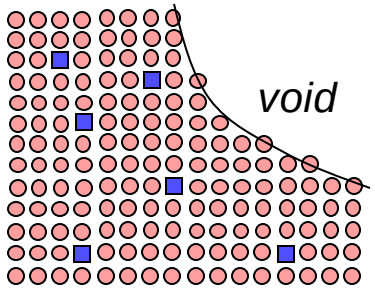



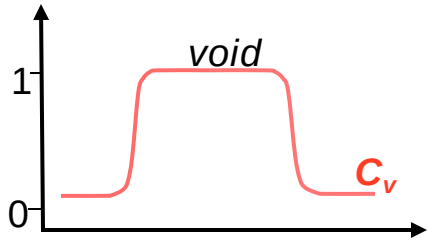




$$\frac{\partial \eta_j}{\partial t} = -L\frac{\delta F}{\delta \eta_j}$$

  - Conserved Concentrations






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

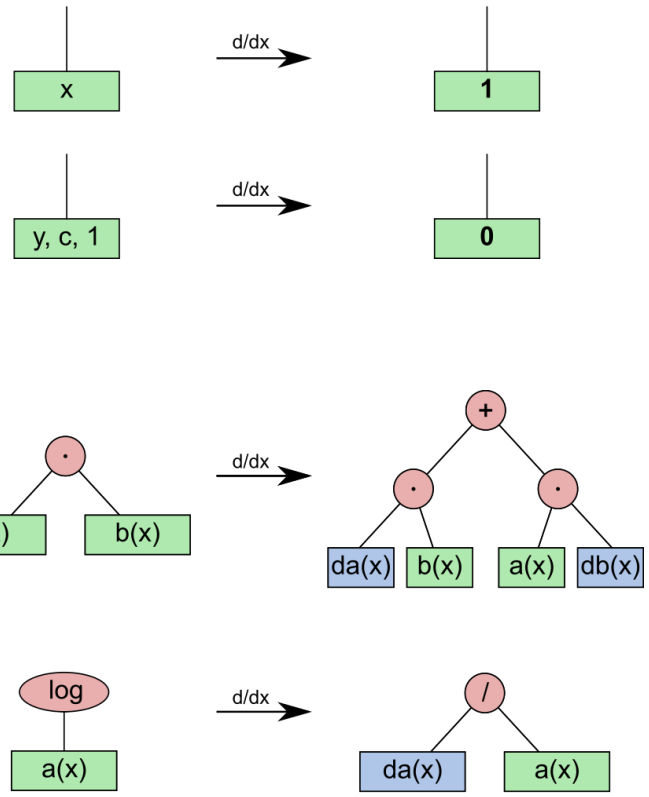
High order derivatives

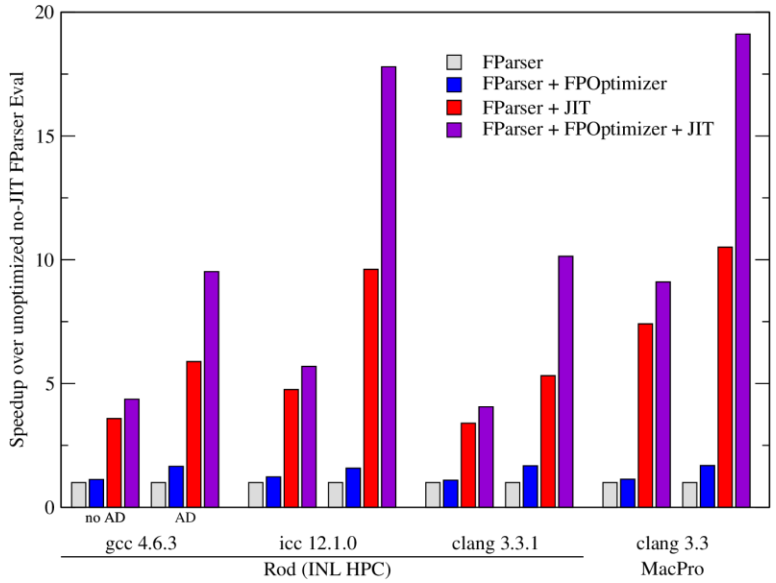
- The variables evolve to minimize a functional $F = \int f(\{c_i, \nabla c_i\}, \{\eta_i, \nabla \eta_i\}) dV$ defining the free energy

# Symbolic Automatic Differentiation

Symbolic differentiation of free energy expressions

- Based on FunctionParser
  `http://warp.povusers.org/FunctionParser/`
  to allow **runtime** specification of mathematical expressions
- Mathematical expressions
  → Tree data structures
- Recursively apply differentiation rules starting at the root of the tree
- Eliminate source of human error
- Conserves developer time
- Just in time compilation
- Independent evaluation of derivatives and function values
  (in contrast to forward mode AD)



# Thermodynamic Database Coupling

CALPHAD approach
⇢ Free energy description of countless systems!
⇢ *de facto* standard TDB file

$$H_a(x) = x(1-x)\left[\sum_{i=0}^{5} a_i (1-2x)^i\right] + a_6 x + a_7(1-x) \quad (1)$$

$$H_{\xi\in\{b,c,d\}}(c) = \sum_{i=0}^{3} \xi_i \cdot (1-x)^i \quad (2)$$

$$H_e(x) = -\frac{H_a(x)}{T_0} + H_c(x)T_0 + H_b(x)\log T_0 + H_d(x)\frac{T_0^2}{2} \quad (3)$$

$$G_0(x) = x(1-x)\left[\sum_{i=0}^{5} f_i (1-2x)^i\right] + f_6 x + f_7(1-x) \quad (4)$$

$$G(x,T) = G_0(x)\frac{T}{T_0} + H_a(x) - H_b(x)T\log T - H_c(x)T^2 + H_d(x)\frac{T^3}{2} + H_e(x)T + Tk_B(x\log x + (1-x)\log(1-x)) \quad (5)$$

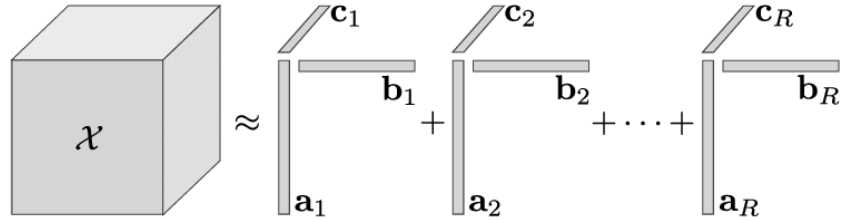
Compound Energy Formalism

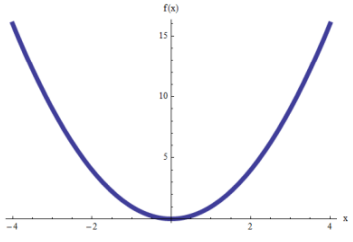
- No sublattices / internal DOFs
  - Directly export free energy expressions
- Solve local equilibrium
  - Concurrently couple thermodynamic software (OpenCalphad, pycalphad, ThermoCalc)
  - Parabolic fits
  - Tabulate free energy (and chemical potentials) in state space
    - Polyadic Tensor decomposition (Moelans, KU Leuven) ⇢ MOOSE
    - Neural Networks
  - Sublattice Kim-Kim-Suzuki model

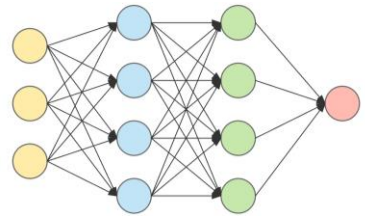
pycalphad

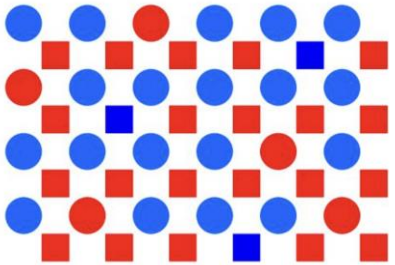
Thermo-Calc Software

OpenCalphad





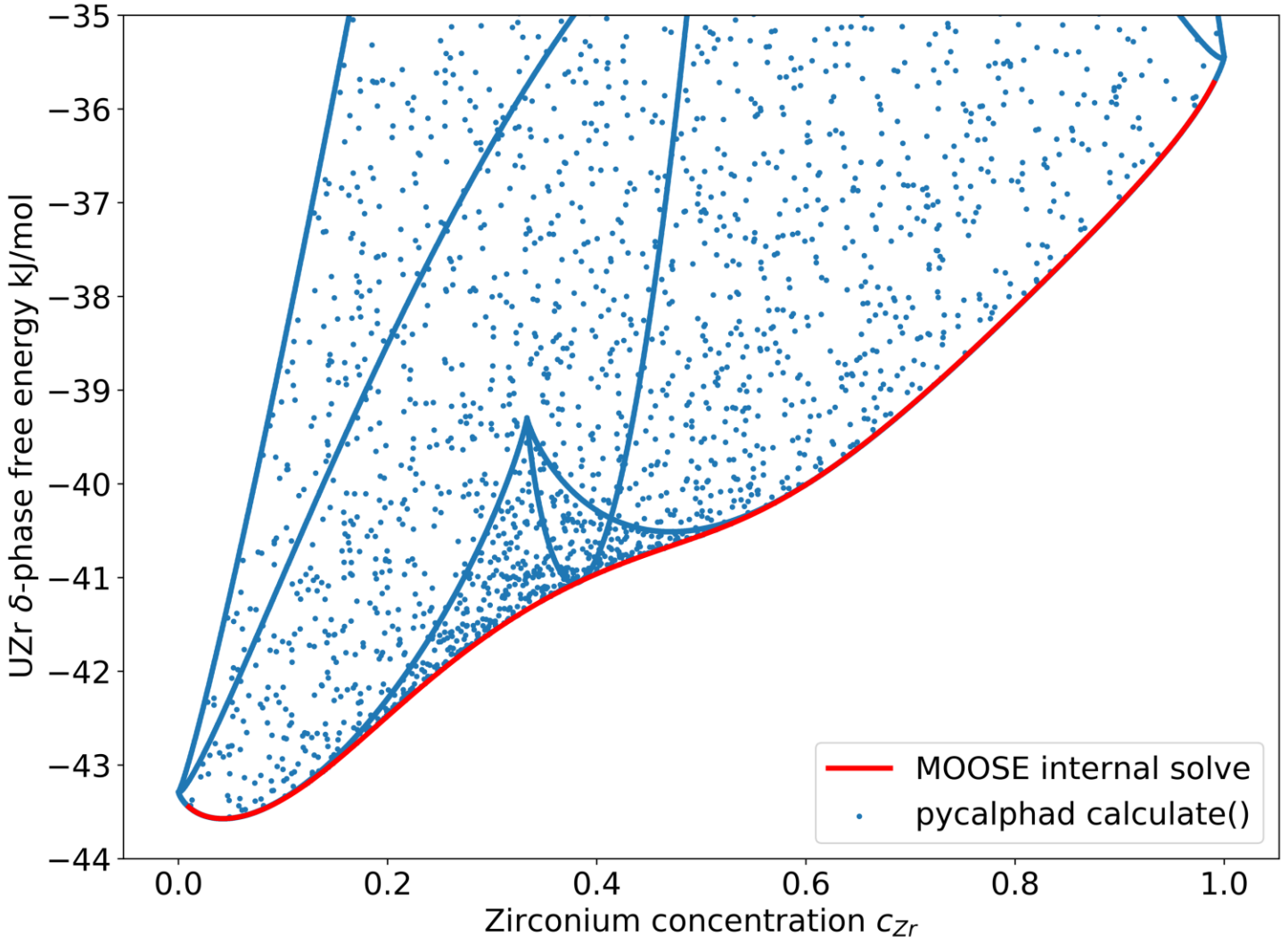


# Internal DOF minimization – Phase free energy

- Solving PDEs for constrained minimization of δ-UZr internal DOFs
- Obtain effective phase free energy for δ-UZr
- Phase chemical potential can be derived as a function of the sublattice chemical potential

$$\frac{1}{a_{jk}}\frac{\partial F_j}{\partial c_{ijk}} = \frac{1}{a_{j'k'}}\frac{\partial F_{j'}}{\partial c_{ij'k'}} = \frac{\partial F_j}{\partial c_{ij}} = \mu_i$$

$$c_i = \sum_j h_j \sum_j a_{jk} c_{ijk}$$

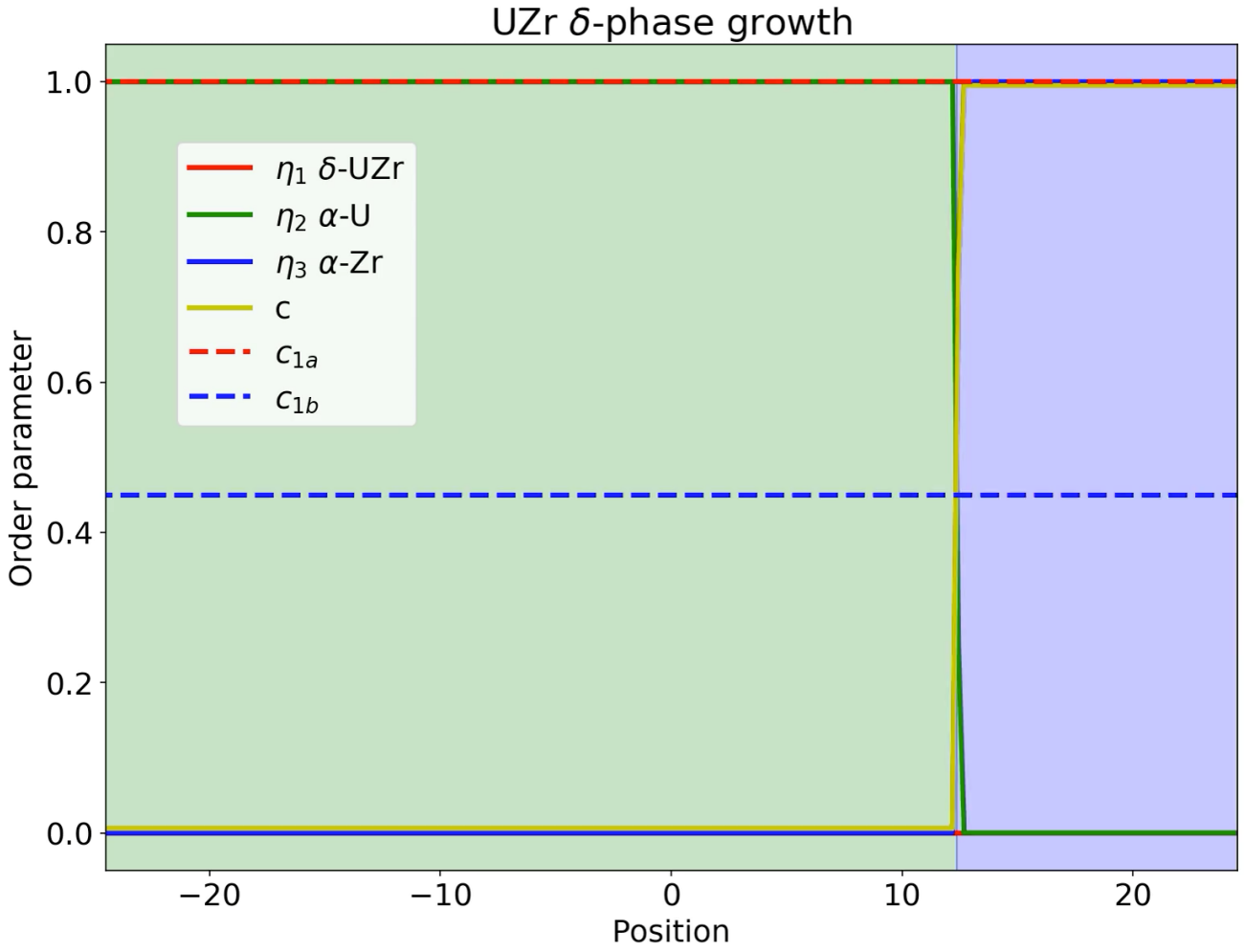


# UZr phase field

- SLKKS phase field model applied to UZr (TAF-ID database data)
- Require multi phase formulation for appropriate switching functions $h_j$
- Thermodynamically consistent model for three phases implemented in MOOSE
- Formulations for n>3 phases exist Implementation WIP

- U/Zr bilayer
- Interdiffusion and formation of δ-UZr

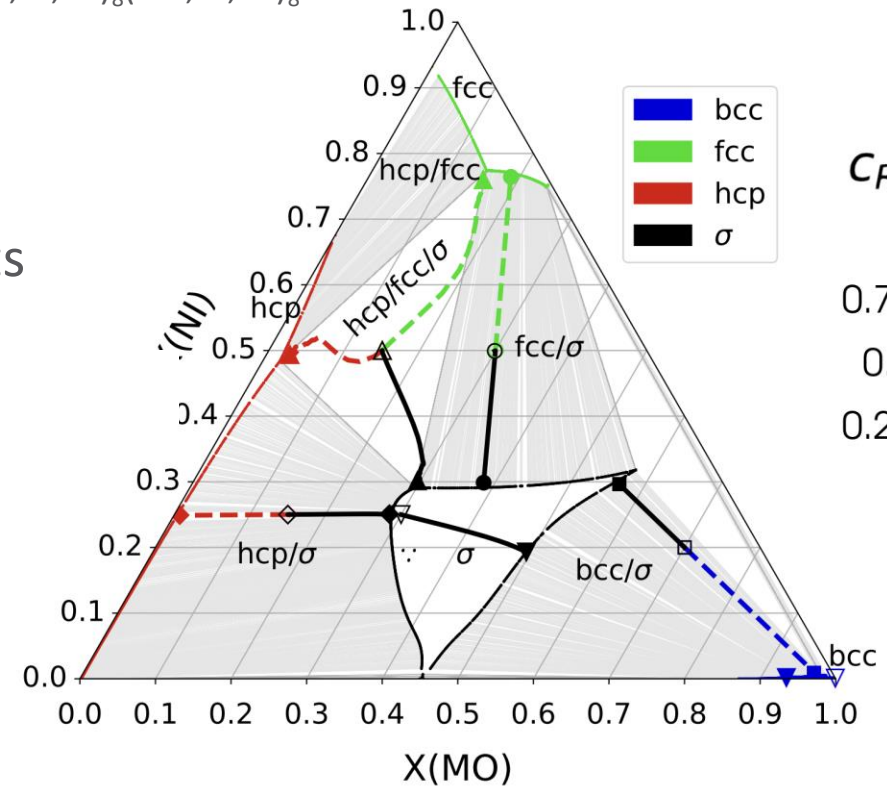
(Interfacial free energies 10mJ/m$^2$ and mobilities not based on physical values, all phases have the same atomic volume)

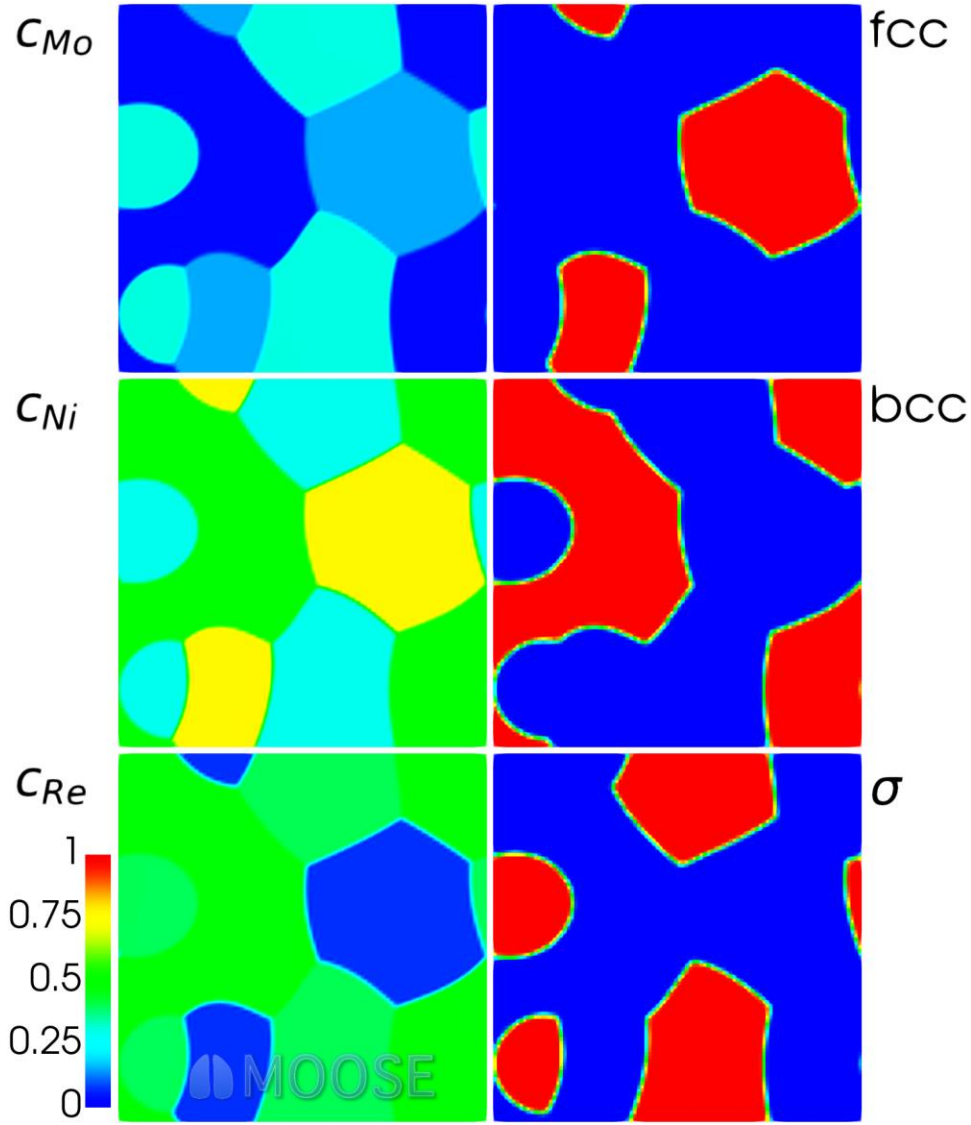


# Mo-Ni-Re phase field

- Mo-Ni-Re High temperature Ni-based superalloy
  J.-C. Crivello, R.Souques, A.Breidi, N.Bourgeois, J.-M.Joubert, Calphad 51(2015) 233-240
- Ternary system
- Five(!) sublattices in the σ-phase
  $(Mo,Ni,Re)_2(Mo,Ni,Re)_4(Mo,Ni,Re)_8(Mo,Ni,Re)_8(Mo,Ni,Re)_8$
- Transient simulation
  Not at equilibrium
- Interfacial energy impacts bulk concentration

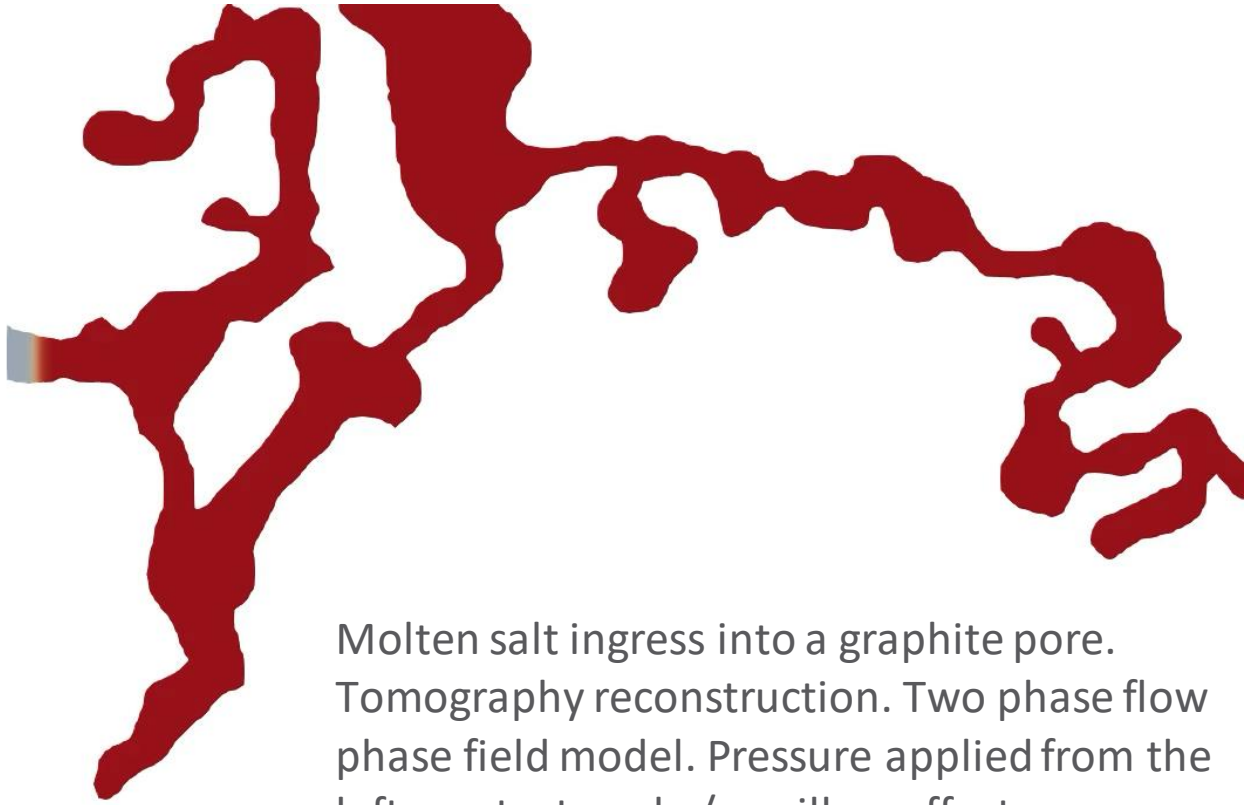
(Interfacial free energies 10mJ/m$^2$ and mobilities not based on physical values, all phases have the same atomic volume)



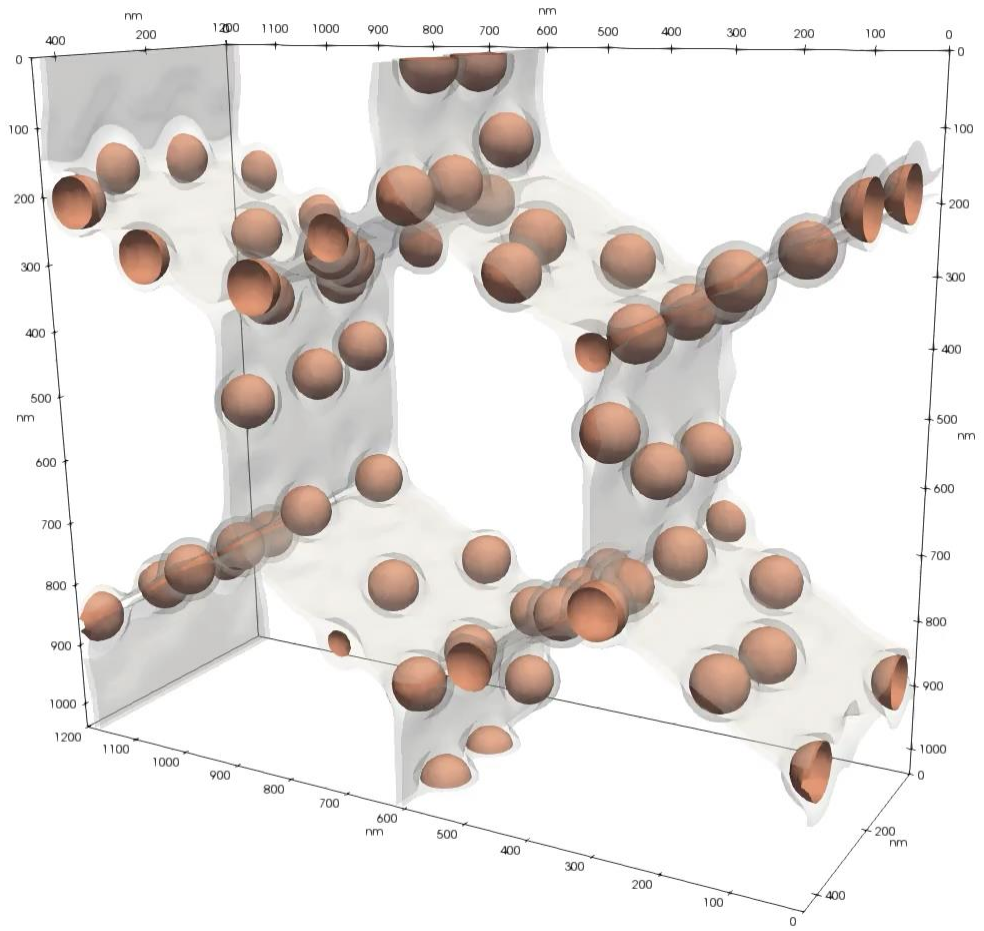




Phase separated microstructure evolved from a homogeneous initial composition of $Mo_3Ni_{10}Re_7$, which is located in the hcp, fcc, and σ-phase coexistence region.

# Phase Field Examples

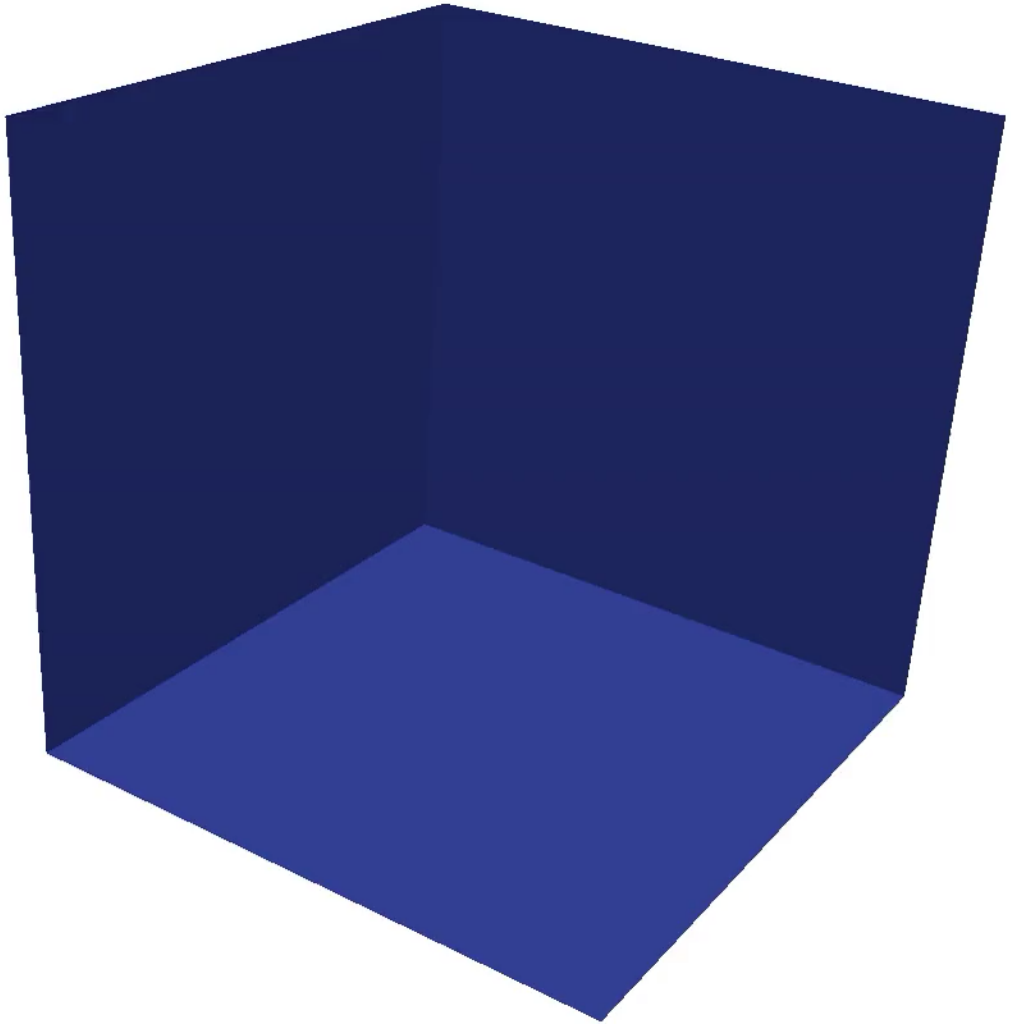


Molten salt ingress into a graphite pore. Tomography reconstruction. Two phase flow phase field model. Pressure applied from the left, contact angle / capillary effects determine ingress.

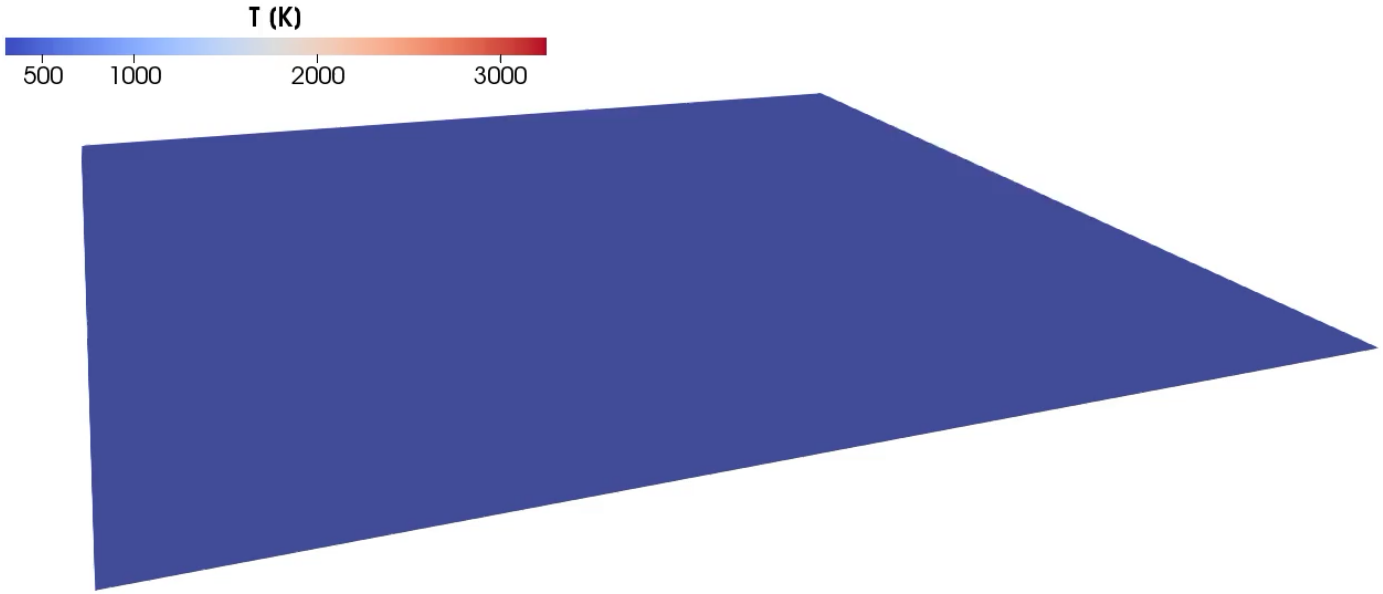



Grain boundaries decorated with growing gas bubbles.

# Phase Field / Laser Melting

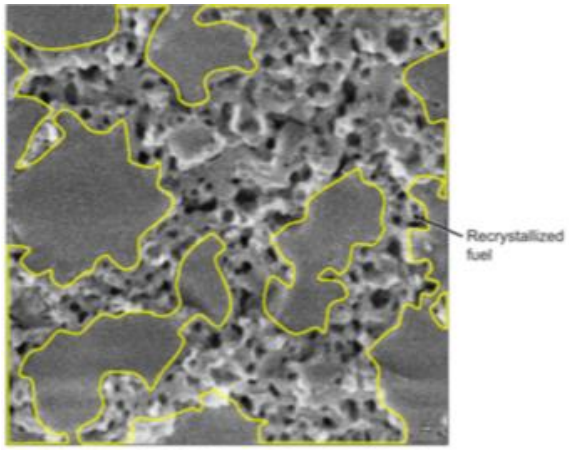


Void lattice formation due to anisotropic interstitial diffusion.

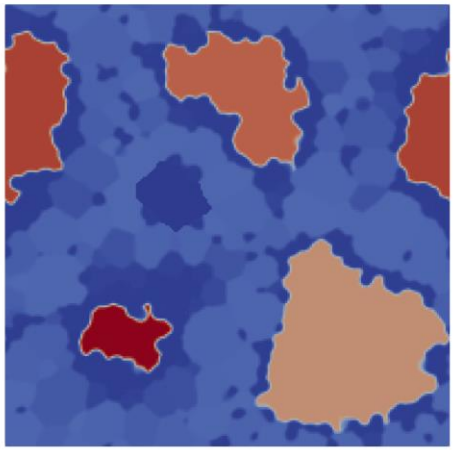



Local laser melting – heat transport, ablation, and Navier-Stokes using ALE.

# High Burnup Structure Formation

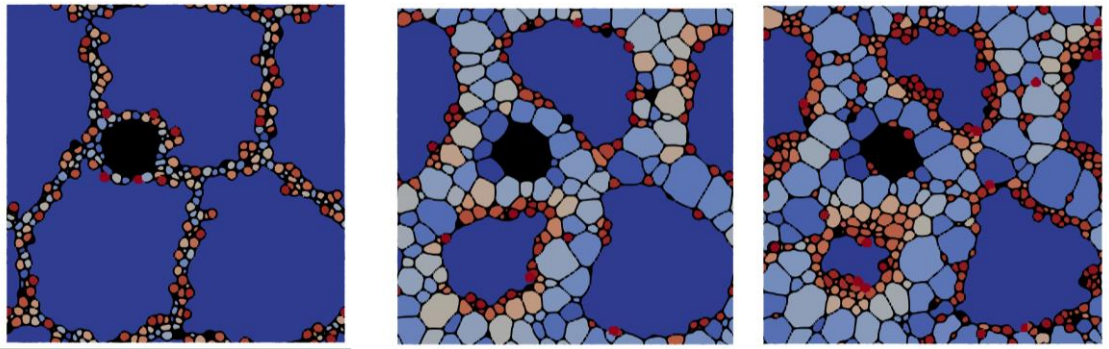



(c) V6001M of RERTR-4 with FD = $4.8 \times 10^{21}$ f/cm$^3$. Fuel volume fraction with recrystallization = 65%.

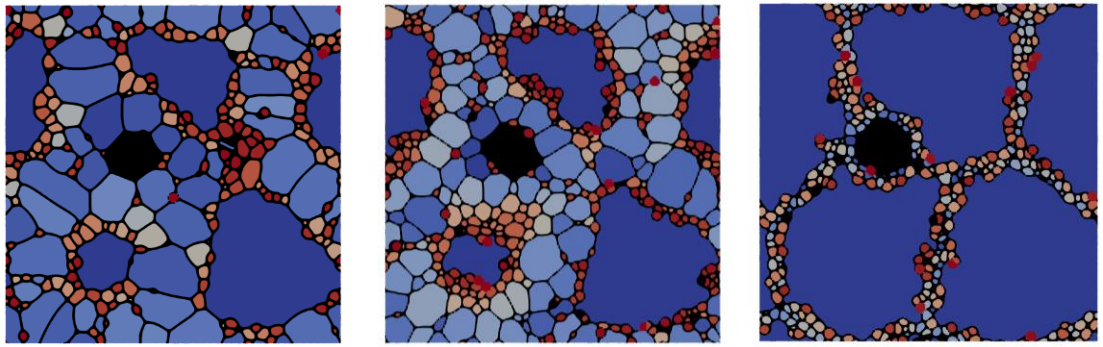


Partial Restructuring

**Burnup Increases**

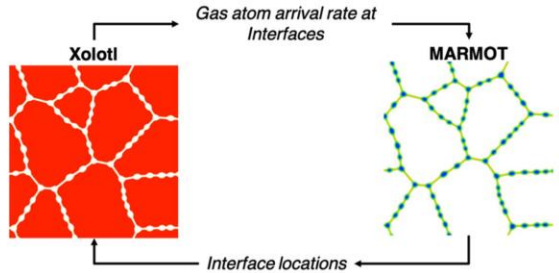


**HBS formation**



1200 K 1100 K 1000 K

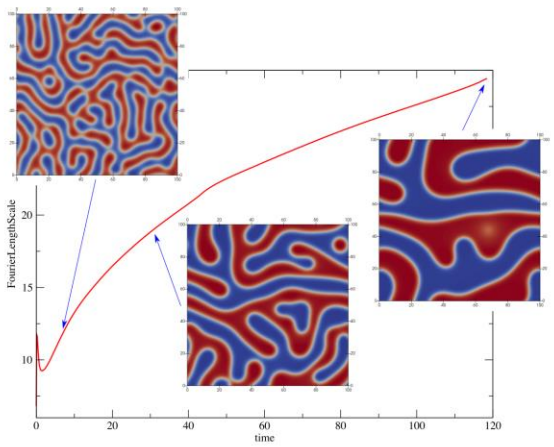
- **Multiscale coupling** to Cluster Dynamics (Xolotl) to resolve intra-granular bubbles



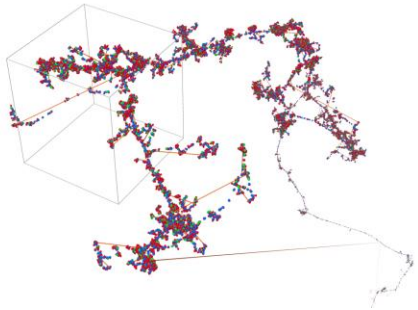

- New grain formation begins around the bubbles and triple junctions and then proceeds to grain boundaries
- Grain formation stage is followed by a coarsening stage
- Restructuring simulations at different temperature demonstrate microstructures with partial HBS formation
- Model captures **partial restructuring** reasonably well, comparable to experimental observations

# Nanoscale ballistic mixing

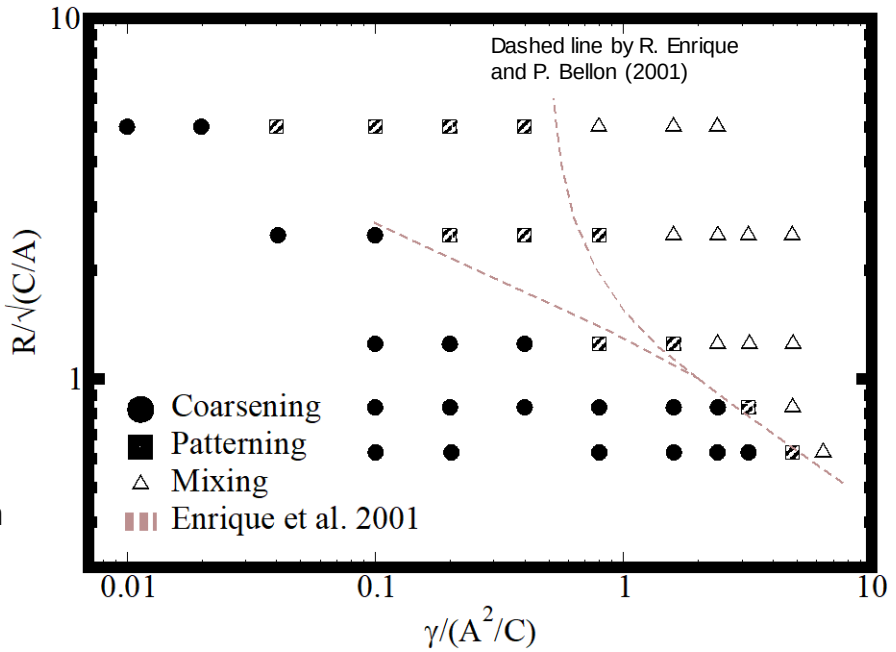
BCMC – Phase Field Coupling

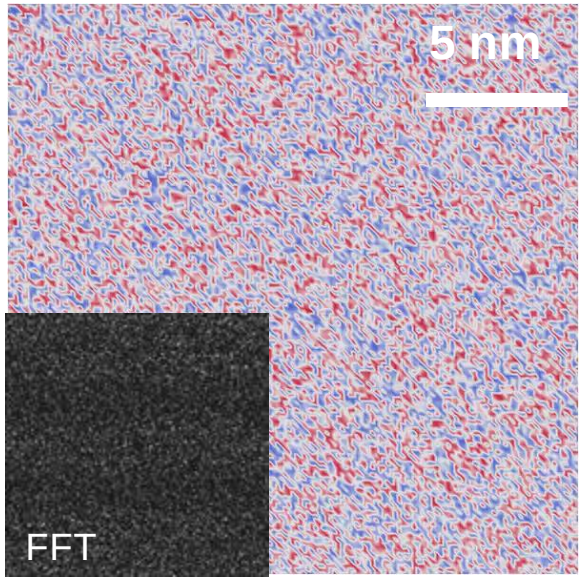



Coupling to FFTW3 to get Fourier space data, such as power spectrum and characteristic length scale.

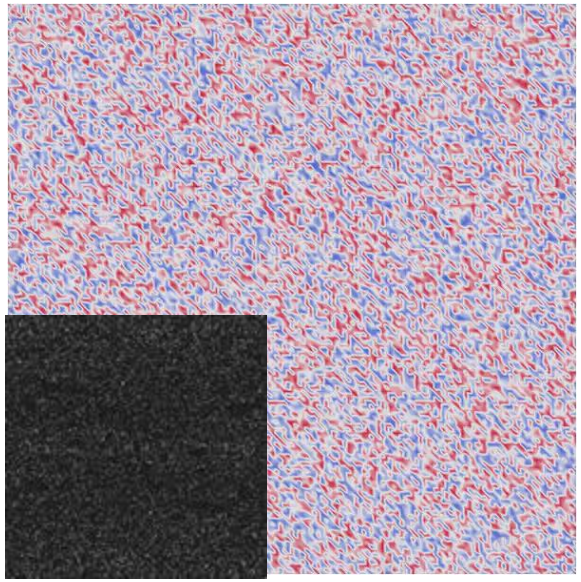


Patterning, mixing, coarsening in an immiscible alloy under irradiation. Axes are displacement rate and distance

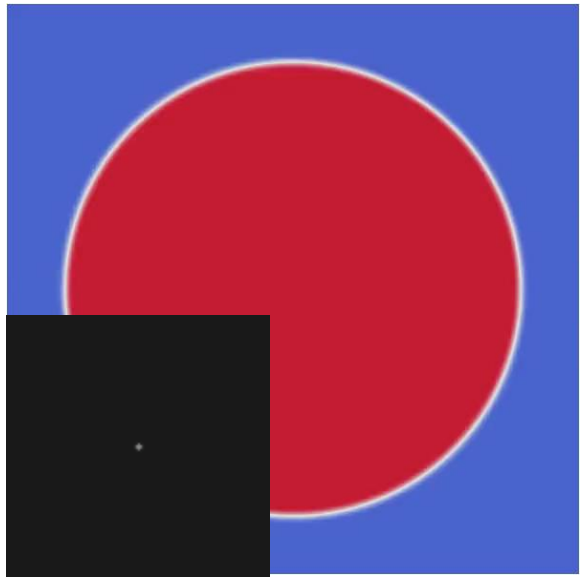



No irradiation

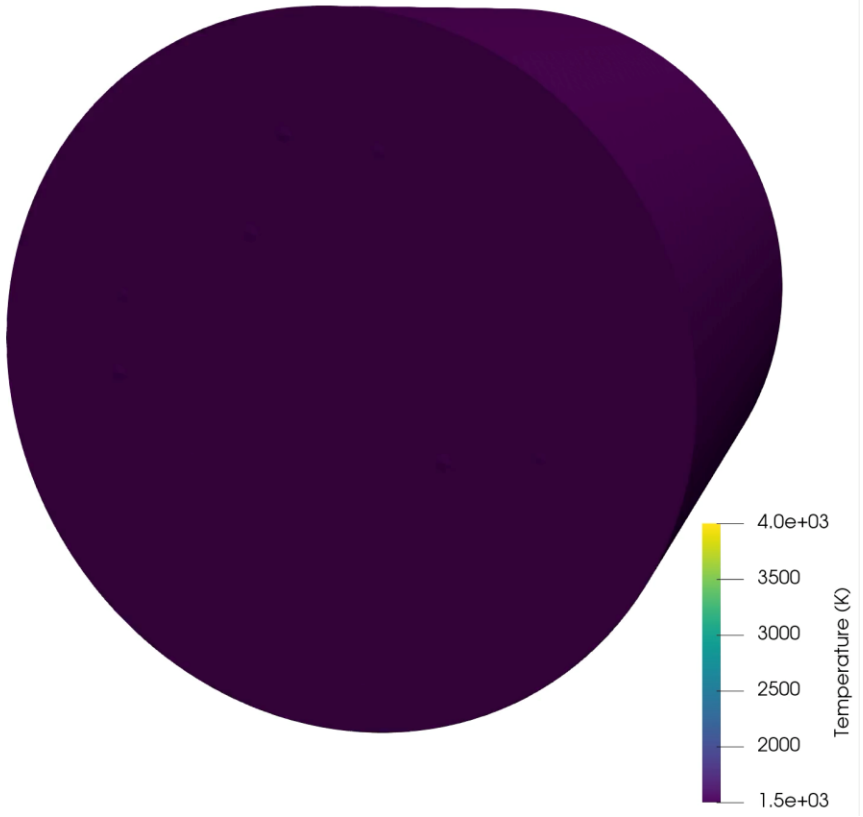


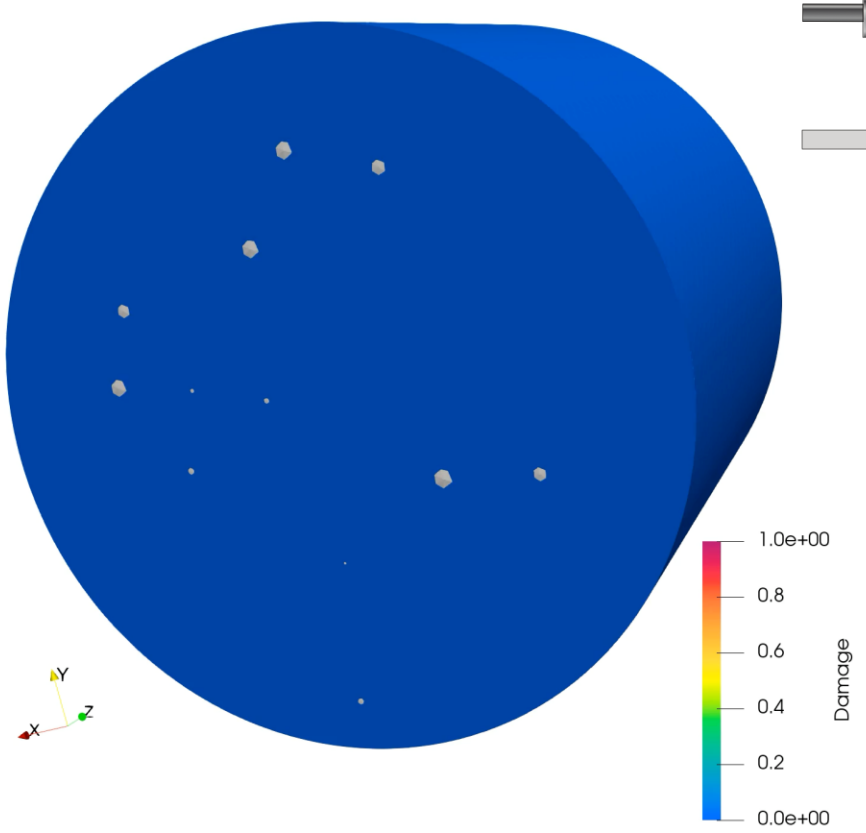
Irradiation



- Phase separating immiscible alloy (CuAg)
- 100,000s of recoil cascades run
  (50 keV Xenon ions)
- Steady-state length scale
- Applications: fission-gas bubbles, FeCrAl precipitates

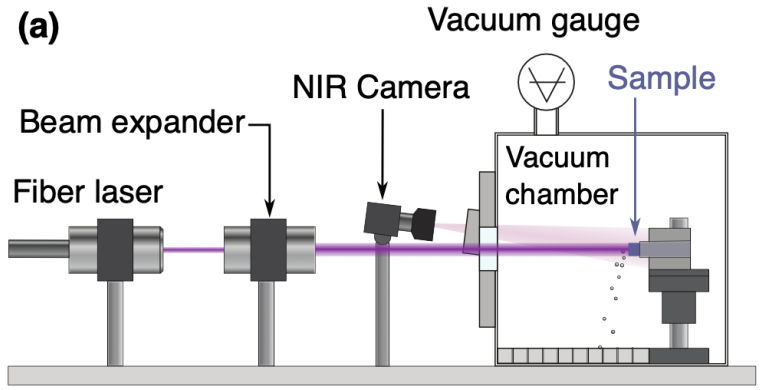
# Conglomerate Laser Ablation

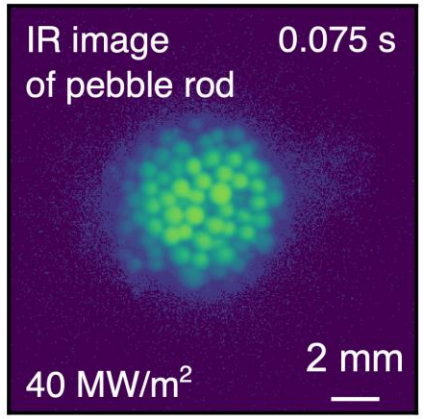


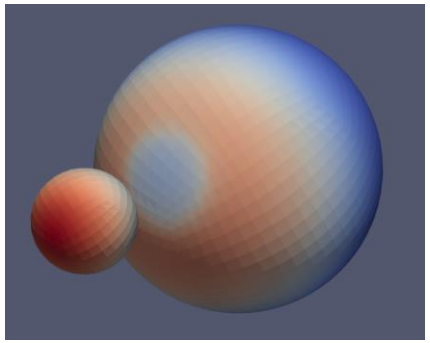




Simulation of carbon spheres embedded in a binder matrix heated by Laser irradiation. Concept for an in-situ extruded renewable first wall material for fusion applications.

- Self shadowing illumination code
- Element deletion
- Smeared cracking
- Feature flood detection



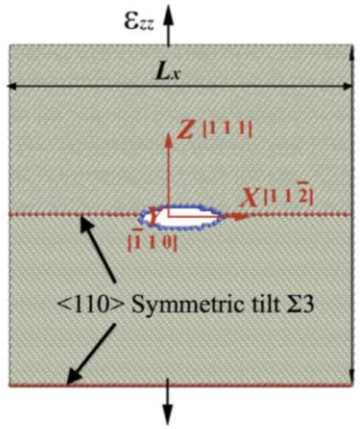
Erick Martinez-Loran[1], Daniel Schwen[2], Benjamin W. Spencer[2], Jose Boedo[1], Eric Hollmann[1] [1)]UCSD, [2)]INL. *In preparation*

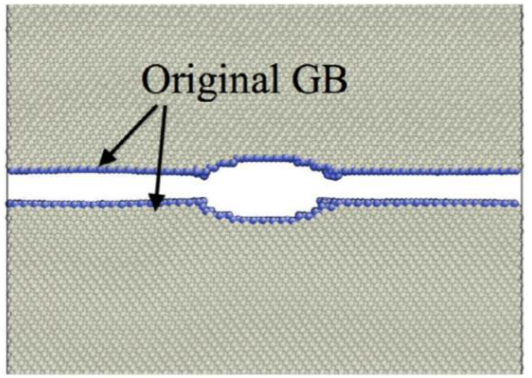
# Fracture strength of porous materials

## Lower length scale informed fracture energy

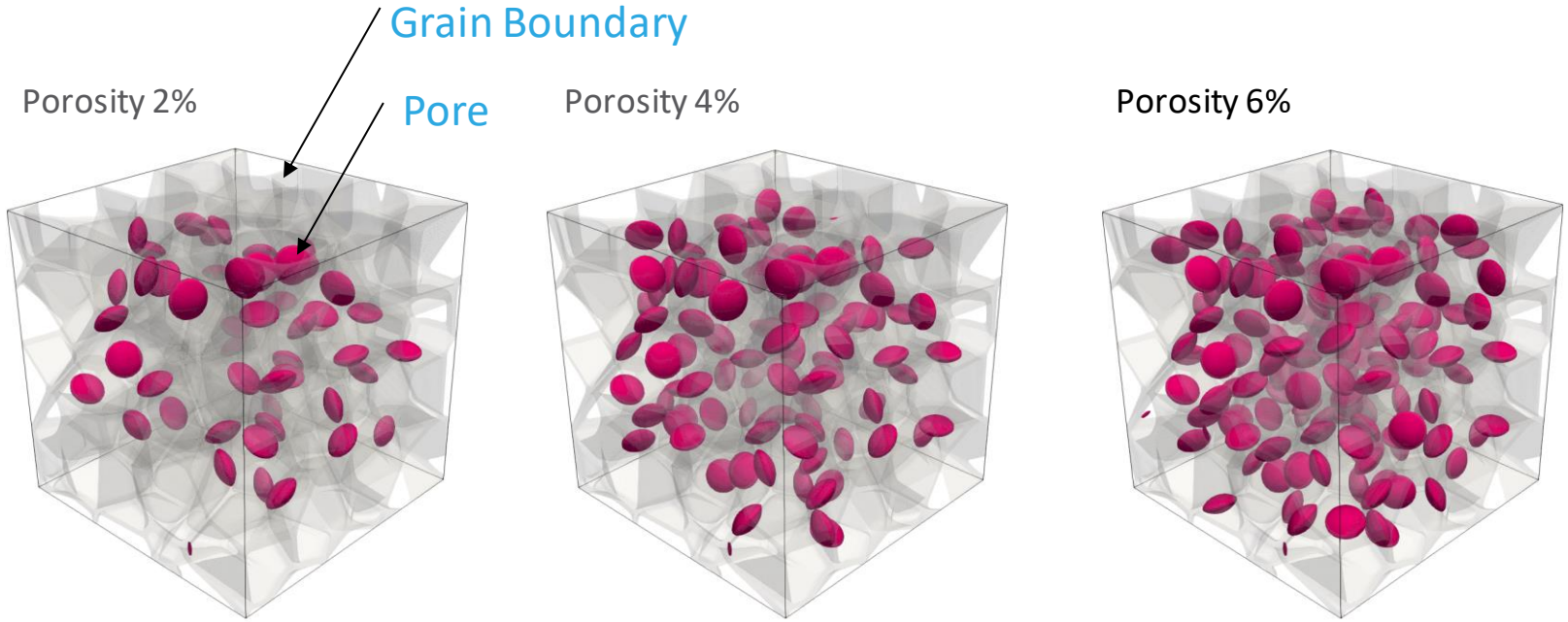
| E(GPa) | GB fracture energy (MPa mm) | Matrix fracture energy (MPa mm) |
|---|---|---|
| 385 | 0.002 | 0.01 |

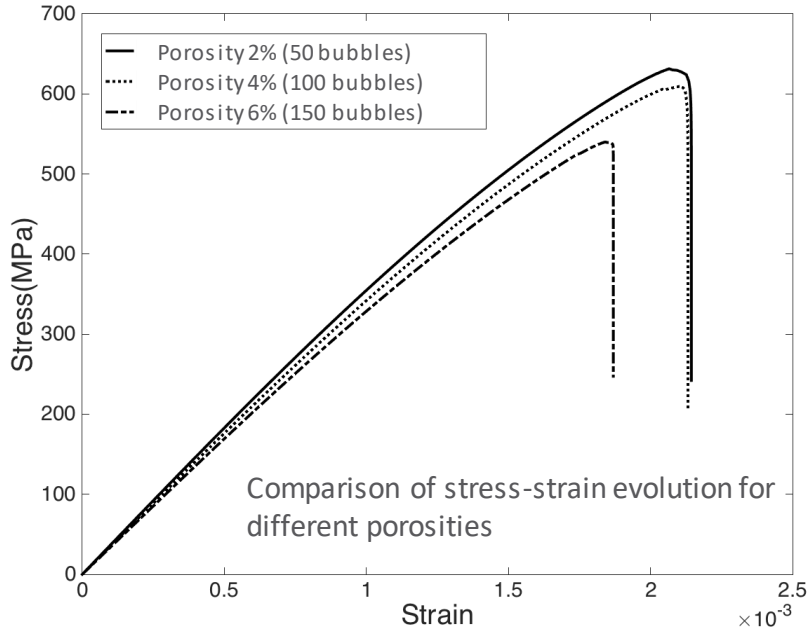
MD




**Phase Field Fracture**

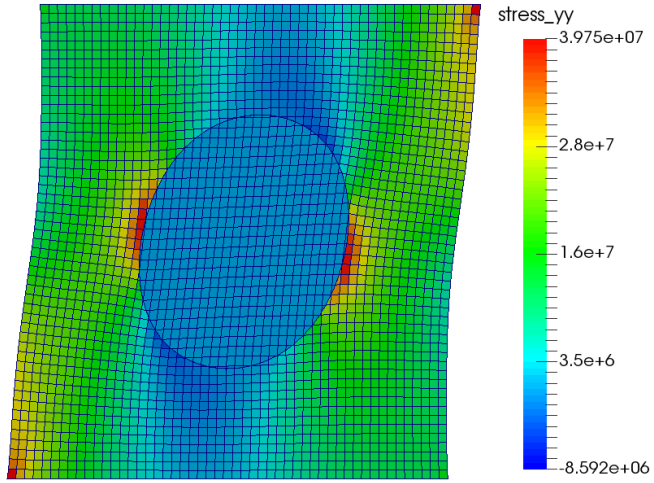


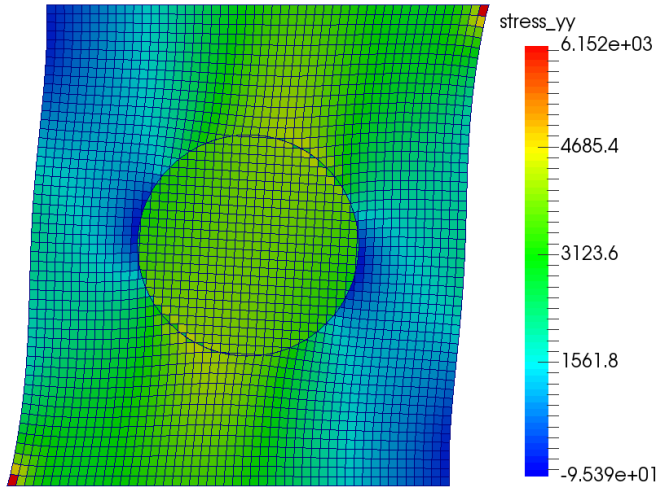
# Extended finite element and fracture modeling

XFEM - Modeling of arbitrary discontinuities in a finite element model that are independent of the mesh

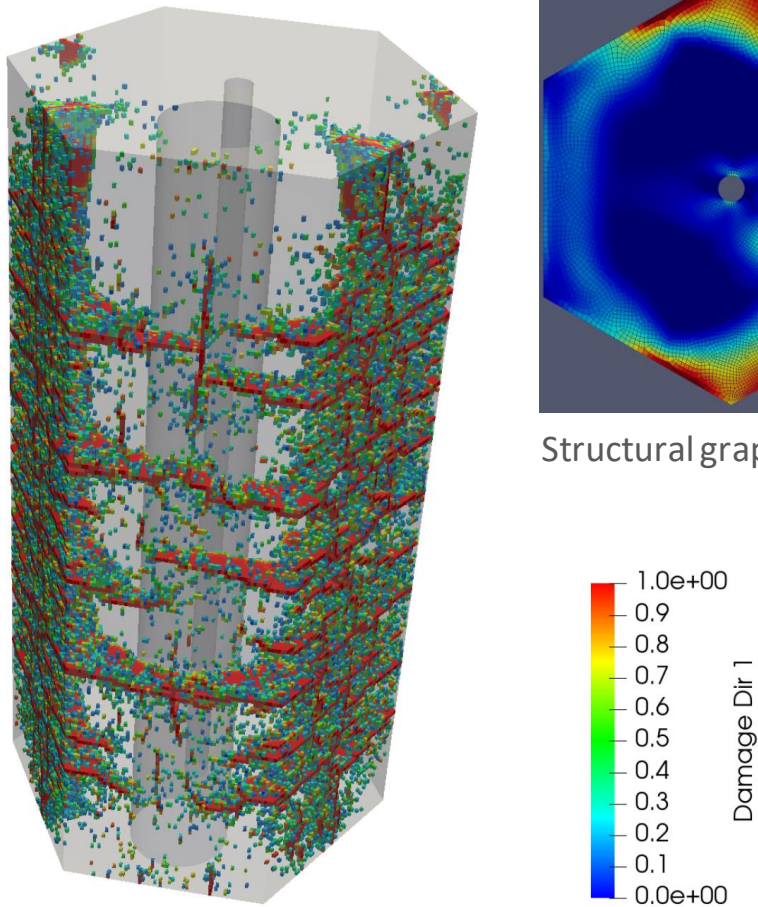
- Stationary and/or moving material interfaces
- The material interface can be represented by a level set function
- Glued and sliding condition can be enforced at interface
- Use existing MOOSE material models, such as plasticity and creep.



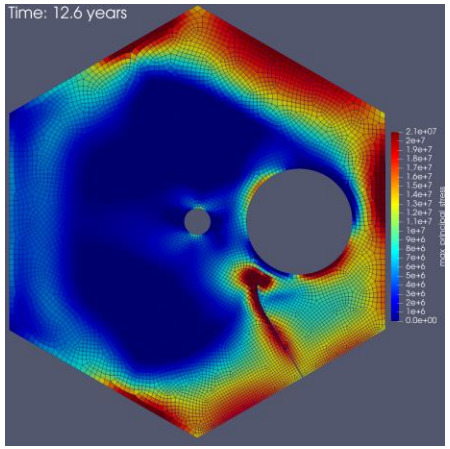

Soft inclusion




Hard inclusion

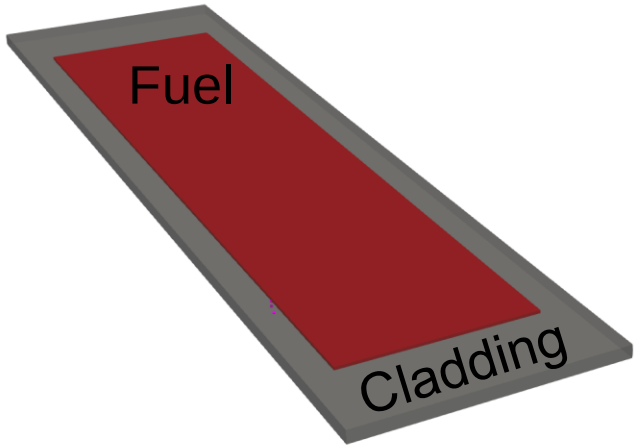


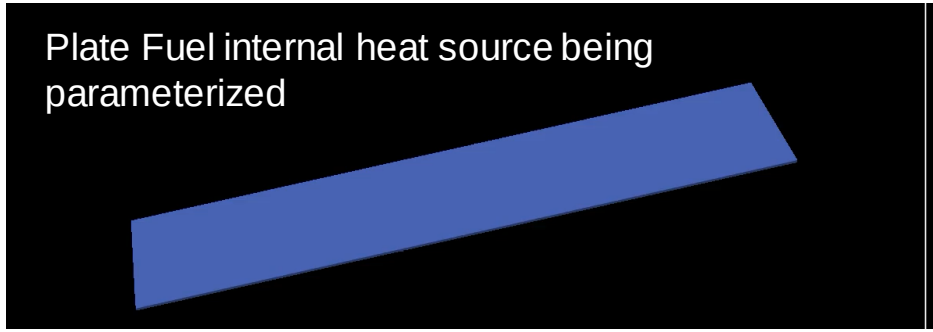

3D Smeared cracking model of structural graphite after 30 years in operation

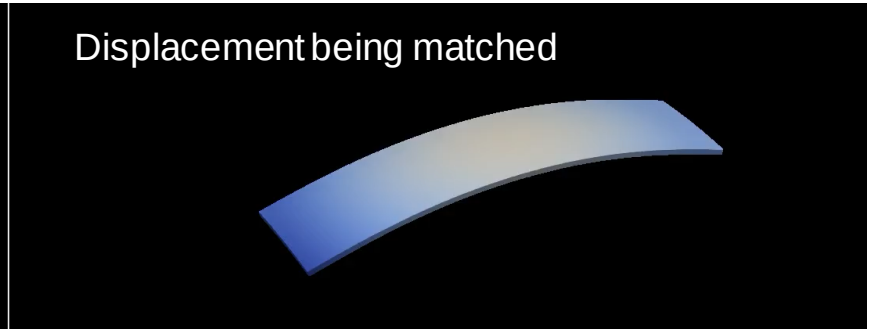



Structural graphite fracture

# Topology- and Inverse Optimization





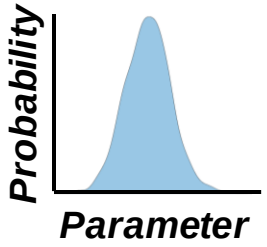

Using Structural isotropic material penalty (SIMP) method to obtain stiffest structure with mass constraint



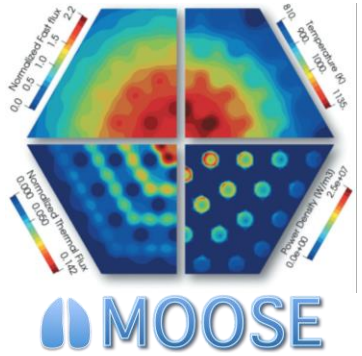
Thermal and mechanical SIMP by combining sensitivities from each physics to obtain a lightweight structure that is stiff and maximizes heat convection

# Bayesian calibration and UQ of computational models

Input parameters:
Fabrication
Material
System

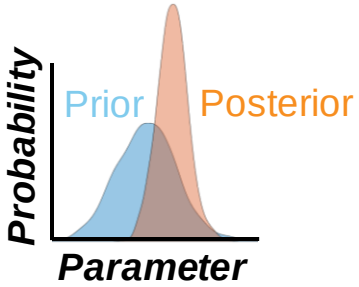


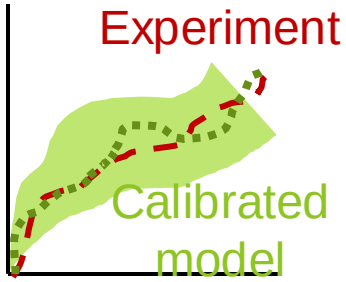

**Step ①:**
model and inputs



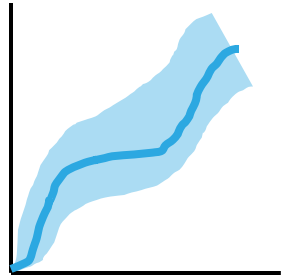

**Step ②:**
validation and calibration





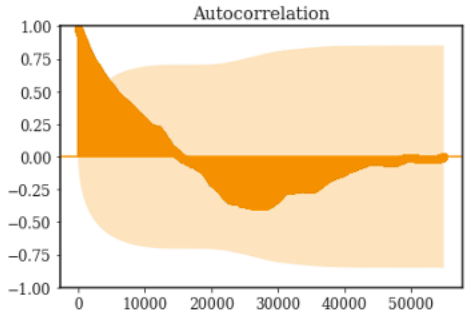




**Step ③:**
forward prediction

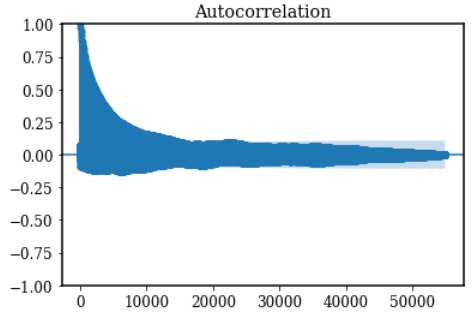


- Identifies and propagates parameter uncertainties
- Reflects model mismatches with experiments during forward simulations
- Permits Bayesian optimal experimental design

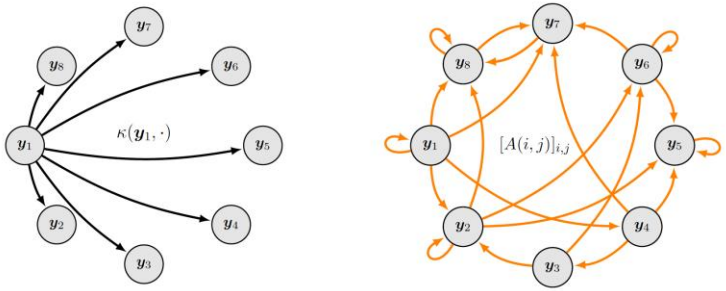
**Independent Metropolis-Hasting**




**Differential Evolution**




**Parallel Metropolis-Hastings**






Idaho National Laboratory

*Battelle Energy Alliance manages INL for the U.S. Department of Energy's Office of Nuclear Energy. INL is the nation's center for nuclear energy research and development, and also performs research in each of DOE's strategic goal areas: energy, national security, science and the environment.*