INL/EXT-15-36754

Light Water Reactor Sustainability Program

Mesoscale modeling of solute precipitation and radiation damage

September 2015

DOE Office of Nuclear Energy

INL/EXT-15-36754

Light Water Reactor Sustainability Program

Mesoscale modeling of solute precipitation and radiation damage

Yongfeng Zhang[1], Daniel Schwen[1], Huibin Ke[1,2], Xianming Bai[1], Jason Hales[1]

[1]Idaho National Laboratory
[2]University of Wisconsin

September 2015

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

EXECUTIVE SUMMARY

This report summarizes the effort during fiscal year 2015 (FY15) to develop mesoscale capabilities for microstructure evolution in reactor pressure vessels. During operation, reactor pressure vessels are subject to hardening and embrittlement caused by irradiation induced defect accumulation and irradiation enhanced solute precipitation. Both defect production and solute precipitation start from the atomic scale, and manifest their eventual effects as degradation in engineering scale properties. To predict the property degradation, multiscale modeling and simulation are needed to deal with the microstructure evolution, and to link the microstructure features to material properties. In this report, the development of mesoscale capabilities for defect accumulation and solute precipitation is summarized. Atomic scale efforts that supply information for the mesoscale capabilities are also included.

CONTENTS

FIGURES

ACRONYMS

BWR	Boiling Water Reactor
CD	cluster dynamics
CRP	copper rich precipitate
DFT	density functional theory
dpa	displacement per atom
FP3DM	Frenkel pair three-dimensional diffusion model
LBP	late blooming phase
LEFM	Linear Elastic Fracture Mechanics
LKMC	lattice kinetic Monte Carlo
MD	molecular dynamics
MNP	Mn/Ni rich precipitate
ODE	ordinary differential equation
PDE	partial differential equation
PF	phase field
PWR	Pressurized Water Reactor
RPV	Reactor Pressure Vessel
RT	rate theory
SPPARKS	Stochastic Parallel PARticle Kinetic Simulator

1 Introduction

The reactor pressure vessel (RPV) in a nuclear reactor contains the reactor core and internals under elevated pressures and temperatures. It serves as the most important pressure boundary component of the nuclear steam supply system, and provides structural support for the reactor core and internals. RPVs are usually made of Fe-based steels, with Ni, Mn and Si being the primary alloying elements, and Cu as either an alloying element or an impurity [4]. During their service life, RPVs are subject to concurrent thermal aging and neutron irradiation, which degrade the mechanical properties of the material from which they are built. In light water reactors (LWRs), RPVs usually operate at a temperature close to 300°C, and they are exposed to neutron fluxes (E>1.0 MeV) in the range of 10^{12} to 10^{15} n/(m^{2}s), corresponding to atomic displacement rates in the range of 10^{-9} to 10^{-11} dpa/s [5]. Here dpa stands for displacement per atom. The flux is usually higher in pressurized water reactors (PWRs) than in boiling water reactors (BWRs). Irradiation by high energy neutrons produces lattice defects in materials, including vacancies (unoccupied lattice sites) and interstitials (extra atoms on lattice sites) and their clusters. At the RPV operating temperatures, these defects can diffuse and agglomerate into extended defects such as voids, loops and other complexities, which together are referred to as lattice features in the context. The lattice features impede dislocation motion, causing hardening and embrittlement [6]. More importantly, the irradiation induced defects enhance solute diffusion and tremendously accelerate the precipitation process, leading to high densities of CRPs (Cu-rich precipitates) and MNPs (Mn/Ni-rich clusters), i.e., the so-called late-blooming-phases (LBPs). Both CRPs and MNPs form in medium or high Cu-concentration (>0.1%Cu) alloys, and only MNPs may form in low (or no) Cu-concentration (<0.1% Cu) alloys [7]. Similar to lattice features, solute precipitates also behave as obstacles for dislocations motion, resulting in hardening and embrittlement [6].

RPVs are extremely expensive, if not impossible, to be replaced or repaired. The integrity of RPVs is one of the important considerations that could limit the service life of current LWRs. To have a reliable assessment of RPV integrity under long term operation, it is required to predict the hardening and embrittlement as functions of irradiation and thermal histories. Due to the slow kinetics involved, the accumulation of lattice features and precipitates, particularly the LBPs, take several or tens of years to develop, making experimental studies difficult. The development of lattice features and solute precipitates depends on both neutron fluence and flux, in addition to temperature. In experimental studies using test reactors, orders higher fluxes than those in LWRs have to be used to reach the same fluence in a short time. The very different fluxes used in the test reactors from those in LWRs may lead to very different microstructure evolution, and thus inaccurate predictions of hardening and embrittlement. On the other hand, surveillance data obtained from commercial reactors utilize the same fluxes as those in LWRs. These data are highly valuable for fitting and validation of empirical or semi-empirical engineering-scale models predicting RPV embrittlement [8, 9]. However, to extrapolate these models into higher fluence levels that will be reached in extended reactor service life is questionable due to the missing physics. As an alternative approach, physics-based multiscale modeling with experimental validations becomes appealing to address this issue.

The build-up of lattice features and solute precipitates starts from the atomic scale, with their sizes in the order of a few nanometers (10^{-9}m) in RPVs. The corresponding timescale spans from fractions of picosecond (10^{-12}s) to years (10^{8}s) [8]. Therefore, to capture the physics involved in microstructure evolution in RPVs requires multiscale modeling and simulations from the atomic to the engineering scale. Under the Grizzly project, we utilize multiscale modeling and simulation to describe the microstructure evolution and the consequent degradation in mechanical properties. In general, the mechanical behavior of a sample of material is determined by its microstructure. For RPV steels, the most important microstructural features are lattice features and solute precipitates, including their volumetric density, size distribution, and spatial distribution. These microstructural features evolve with time, dependent on the irradiation and thermal histories. To predict the hardening and embrittlement in RPVs, models are needed to describe the evolution of microstructural features and to describe the corresponding mechanical behavior. In this report, we focus

on the development of models for microstructure evolution. For lattice features, we use rate theory (RT) and cluster dynamics (CD). We couple lattice kinetic Monte Carlo (LKMC) and phase field (PF) for solute precipitation. For both lattice features and solute precipitates, atomistic simulations such as molecular dynamics (MD) simulations and density functional theory (DFT) calculations are needed to obtain the required material parameters. We note that in reality, the evolutions of lattice features and solute precipitates are coupled with each other, with the detailed interactions not fully clear at this time [10, 11]. Here, they are treated separately. The effect of irradiation-induced defects on solute precipitation is taken into consideration by using vacancy concentration dependent diffusivities for solute elements.

The development of these models spans multiple years. In this document, the effort made in FY15 is summarized. For background on previous work, please refer to previous Grizzly reports [12]. In FY15, we have focused on model development for lattice features and solute precipitates. In this development, we use realistic material parameters for RPV steels when data are available. Otherwise, surrogate materials such as pure Fe are used for simplification and for the purpose of benchmark with literature results. Unless otherwise stated, the model development is made in the Grizzly simulation package within the MOOSE framework. In the following, the PF model for FeCuMnNi alloys is described in Chapter 2, and the coupling of LKMC with phase field in Chapter 3. The RT and CD models for lattice features are presented in Chapter 4, followed by Summary in Chapter 5.

2 Phase field model development for solute precipitation in FeCuMnNi alloys

The phase field method has emerged as a powerful and flexible tool for quantitative modeling of the coevolution of microstructure and physical properties at the mesoscale. In the phase field method, the microstructure is described by a system of continuous variables, where the microstructure interfaces have a finite width over which the variables exhibit smooth transitions. The evolution of the microstructure is defined in terms of the free energy of the system, and can be coupled to other physics to provide a complete view of the material behavior. Phase field simulations range from hundreds of nanometers to hundreds of microns and evolve at diffusive time scales [13].

An iron-copper-manganese-nickel (FeCuMnNi) quaternary two phase free energy and mobility model was implemented in Grizzly. The FeCuMnNi alloy will serve as the surrogate material for reactor pressure vessel (RPV) wall materials to perform microstructure evolution simulations. To enable the simulation of solute precipitation including the precipitate nucleation process, a novel nucleation algorithm was developed to add physical nucleation to the phase field method. A set of underlying capabilities was added to the MOOSE phase field module to enable these developments.

2.1 Capability development

The `phase_field` module in MOOSE contains the necessary tools to solve the partial differential equations for the phase field method that define the microstructure variable evolution to minimize the overall free energy. The evolution of non-conserved order parameters η_i (representing phase regions and grains) is governed by the Allen-Cahn equation (1) and conserved order parameters c_i (representing concentrations) are evolved using the Cahn-Hilliard equation (2).

$$\frac{\partial \eta_j}{\partial t} = -L_j \frac{\delta F}{\delta \eta_j} \tag{1}$$

$$\frac{\partial c_i}{\partial t} = \nabla \cdot M_i \nabla \frac{\delta F}{\delta c_i} \tag{2}$$

F is the total free energy of the modeled system as a function of the phase field variables, which can be formulated as a volume integral

$$F = \int_\Omega \left[f_{\text{loc}}(\vec{c}, \vec{\eta}) + f_{\text{gr}}(\nabla \vec{c}, \nabla \vec{\eta}) + E_{\text{d}} \right] \, dV, \tag{3}$$

over multiple free energy density contributions, where Ω is the simulation domain, f_{loc} is the local free energy density, f_{gr} is the gradient energy contribution, and E_{d} is the contribution of other sources of energy. The $\vec{c}, \vec{\eta}$ and $\nabla \vec{c}, \nabla \vec{\eta}$ indicate a functional dependence on all conserved and non-conserved order parameters in the domain and their gradients, respectively. Executing the variational derivatives in (1) and (2) yields terms containing the derivatives of the local free energy density f_{loc} with respect to all order parameters.

The thermodynamic properties of the modeled system are determined by the thermodynamic potential in f_{loc}. The gradient contribution f_{gr} is the reason the phase field model represents interfaces with a diffuse width and contributes to the interfacial energy. f_{loc} is therefore the primary input needed to formulate a new phase field material model. In the `phase_field` module, the residuals for the generic phase field equations are provided as kernels, while the free energy and its derivatives are supplied by material objects. We use a special material interface to provide material properties for all necessary derivatives of the free energy.

In general, users use the provided kernels without modification, and only create material objects defining different free energies.

The standard MOOSE solver uses the preconditioned Jacobian-free Newton Krylov method (PJFNK), provided by the PETSc library [14]. To improve the convergence of the solve, the chosen preconditioning matrix should be as close as possible to the actual Jacobian of the problem. Computing the Jacobian matrix entries effectively means providing the derivatives of the residual vector with respect to all non-linear variables of the problem, thus requiring additional derivatives (including cross-derivatives) of the free energy functional. Especially for quaternary two phase free energy this amounts to a large number of derivatives to be evaluated.

2.1.1 Automatic differentiation

To create a material object that defines the free energy for a phase field model, code must be written that defines not only the thermodynamic free energy expression, but also its derivatives. For non-conserved order parameters, the 2$^{\text{nd}}$ derivatives are needed and for conserved order parameters, up to the 3$^{\text{rd}}$ derivatives could be required. This is complicated even more when a free energy is a function of multiple variables, because all cross derivatives are also required. To avoid having to take and implement all the derivatives, we have implemented automatic symbolic differentiation.

MOOSE uses the Function Parser library that is included as a third-party plugin in the underlying libMesh finite element library [15]. The Function Parser Library accepts a mathematical function definition given as a plain text string. This string is lexically parsed into an intermediate tree representation and then transformed into a stack machine bytecode. This bytecode can then be executed by the function parser bytecode interpreter module as often as necessary without further transformations.

This intermediate tree representation of the function parser expressions lends itself to algorithmic transformations, such as an automatic differentiation procedure. In this tree structure, leaf nodes can correspond to constants or variables, and internal nodes correspond to mathematical operators and functions with the arguments contained in the respective child nodes or subtrees. The derivative of the leaf nodes yields 0 for all nodes that do not represent the variable the derivative is taken with respect to, and 1 for all nodes that do represent the variable. The derivatives of the internal nodes are constructed recursively according to a set of elementary derivative rules.

Construction of the derivative starts at the root node of the expression tree. For the example expression tree in Figure 1, which represents the expression $x^2(y+5)$, the root node holds the multiplication $N_1 = N_2 \cdot N_3$. To obtain the derivative with respect to x of the given expression we need to calculate the derivative of the root node $d_x N_1$. We set $d_x N_1 = d_x N_2 \cdot N_3 + N_2 \cdot d_x N_3$ according to the product rule. This expression contains derivatives of the nodes N_2 and N_3, which are recursively constructed, until the leaf nodes are reached which have a trivial vanishing 0 derivative in all cases except the $d_x N_4$, which evaluates to 1. The full derivative expression that is constructed this way is $(2x \cdot 1) \cdot (y+5) + x^2(0+0)$.

The function parser library provides a comprehensive algebraic optimizer that groups, reorders, and transforms the function expression into an equivalent but faster to evaluate form. This optimization stage delivers a speedup of a factor of two, on average. The algebraic simplifications are essential to remove the trivial leaf node derivatives which may lead to evaluation errors such as divisions by zero, that can be avoided by simple term cancellations. In the above example, the simplifications reduce the derivative expression to $2x(y+5)$.

2.1.2 Just-in-time compilation

To further improve the performance of the parsed and runtime interpreted functions, we have developed a just-in-time (JIT) compilation module. At runtime, the generated bytecode sequences are automatically

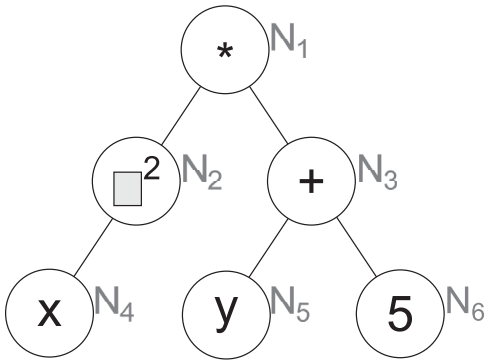

Figure 1: Schematic example of the tree representing the mathematical expression $x^2(y + 5)$. The nodes N1 and N3 represent the multiplication operator and the sum operator with two arguments each, the internal node N2 represents a square function with one argument, and the leaf nodes N4-N6 represent the variables x, y, and the constant 5.

transformed into small C source code files. A compiler is dispatched to compile each function file into a dynamically linkable library, which then is loaded on the fly using the `dlopen` POSIX system call. If at any stage the JIT compilation fails the function evaluation falls back on the bytecode interpreter, otherwise the generated machine code is called. The time overhead of the additional compilation step is on average of the order of 0.1s per function expression or below, depending on the system the simulation is executed on. This is further mitigated by a caching system. A unique hash is computed from the function bytecode and the compiled functions are stored permanently using the hash as a filename. Recompilation will only occur if the bytecode, and thus the function expression, changes. Trivial function changes, namely the modification of constants, will in most cases not trigger a recompilation.

In Figure 2 the performance of unoptimized interpreted function parser evaluations is compared to combinations of optimized and JIT compiled function evaluations for a variety of compilers under Linux and MacOS. Two function sets were used for the comparison. The left data labeled `no AD` were obtained using a set of mathematical expressions as they appear in free energy models. The right bar sets labeled `AD` were obtained by applying the automatic differentiation to the former functions. Two conclusions can be drawn from this comparison, firstly the JIT compilation alone delivers speedups up to a factor of 10. The algebraic optimizer can deliver speedups up to a factor of two on certain functions. The efficacy of the optimizer is largest on the `AD` function set, which contains lots of trivial terms from the leaf node derivatives, such as entire sub terms that end up being multiplied by 0.

Through this automatic differentiation system we achieve a significant reduction in developer time and remove a source of developer errors that are difficult to track down and debug. The resulting models offer optimal convergence properties due to the complete implementation of the full Jacobian matrix.

2.1.3 Expression Builder

To generate functional expressions for free energies and mobilities that can be passed to the automatic differentiation algorithm we have developed the ExpressionBuilder system. ExpressionBuilder is a C++ class that gets added to MOOSE objects such as Materials through inheritance and makes a set of new nested classes available to hold terms and functions. A term is defined as a symbol, a number or a mathematical operator and its parameters. A function is defined as a term with a substitution rule that substitutes the function arguments into the underlying term. In addition ExpressionBuilder provides overloaded operators that accept the new term and function types as parameters and returns augmented terms. Terms are internally stored as tree structures. Node in these tree can represent an operator or mathematical function (such as the

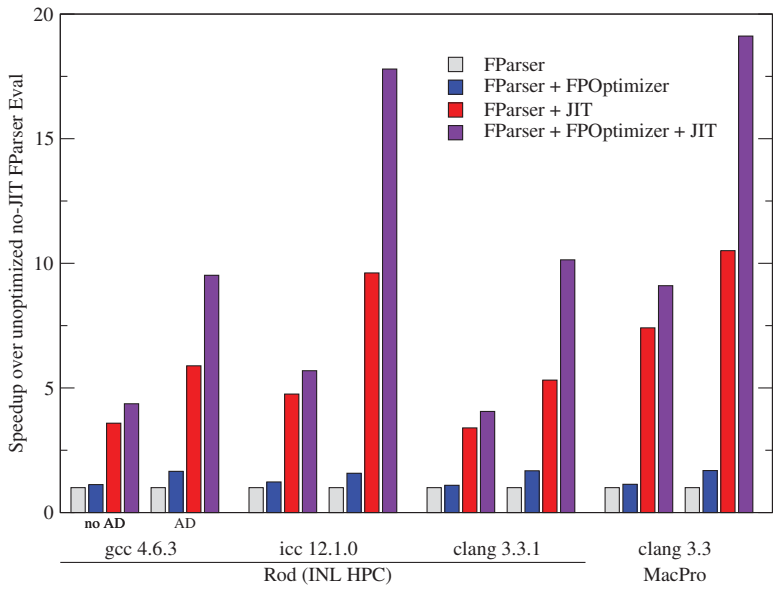

Figure 2: Performance comparison for the parsed functions showing the speed-up gained by the just-in-time compilation (JIT) and the algebraic optimization (FPOptimizer). Various compilers were compared on the INL HPC system and on a desktop computer. The combination of JIT and algebraic optimization can yield speed-ups of up to about a factor of twenty.

logarithm or trigonometric functions) or a symbol or number. Operator and function node have child nodes representing the arguments (e.g. left hand side and right hand side). The overloaded operators assemble a tree representing an entire expression step by step. Named terms and functions can be combined to form new terms and functions, allowing the construction of complex expressions. The syntax of expression builder is designed to naturally match mathematical expressions as they can be found in scientific publications.

2.1.4 Multiphase models

Multiphase model development requires the construction of a global free energy functional spanning the entire phase space of the system. One common approach is utilizing a linear combination of the free energy densities $f_{loc,j}$ of each phase in the system.

$$f_{loc}(\vec{c},\vec{\eta}) = \left[\sum_j h(\eta_j) f_{loc,j}(\vec{c})\right] + W g(\vec{\eta}) \tag{4}$$

A switching function $h(\eta)$ smoothly changes from 0 to 1 as η goes from 0 to 1. The total weight of all phase free energy contributions at each point in the simulation volume is exactly unity, which translates to the need to enforce the constraint $k(\vec{\eta}) = 0$ for

$$k(\vec{\eta}) = \left[\sum_j h(\eta_j)\right] - 1. \tag{5}$$

Constraint enforcement Two phase systems can easily be modeled using a single order parameter η_1 and the explicit constraint $\eta_2 = 1 - \eta_1$, which, for a symmetric switching function with $h(\eta) = 1 - h(1 - \eta)$, satisfies the constraint k. For n-phase systems with $n > 2$ it becomes advantageous to use n order parameters.

In this case the constraint k is not automatically satisfied and needs to be enforced by other means. In the MOOSE phase field module we offer two methods to enforce the switching function sum constraint, a hard constraint utilizing the Lagrange multiplier technique and a soft constraint through a penalty term added to the free energy.

The hard constraint is applied by introducing a Lagrange multiplier λ as a field variable. With $a_j(\vec{\eta}, \vec{c}, v)$ being the weak form (Allen-Cahn) residual for the jth non-conserved order parameter, we need to find $(\vec{\eta}, \lambda)$ satisfying the boundary conditions such that

$$a_j(\vec{\eta}, \vec{c}, v) + \int_\Omega \lambda \frac{\partial k}{\partial \eta_j} v \, dx = 0 \tag{6}$$

$$\int_\Omega q \frac{\partial(\lambda k)}{\partial \lambda} \, dx = 0 \tag{7}$$

holds for every test function v and q. We note that these equations alter the character of the Jacobian matrix of the non-linear problem substantially by introducing a zero block on the Jacobian diagonal. This can complicate the solve substantially. By replacing the constraint k with a modified constraint

$$\bar{k}(\vec{\eta}, \lambda) = k(\vec{\eta}) - \frac{\epsilon}{2}\lambda, \tag{8}$$

the Jacobian fill term $\frac{\epsilon}{2}\lambda$ introduces a small λ dependence in the constraint through an ϵ (which defaults to 10^{-9}). This results in an on-diagonal Jacobian value of $-\epsilon$ in the kernel of Eq. (7), while it drops out in the residual of Eq. (6). This is necessary to force a Jacobian matrix with full rank, avoiding *Zero pivot* PETSc-Errors, and greatly improves convergence. This approach results in a violation of the constraint by about ϵ, though this violation can be kept small by using an ϵ as small as possible.

As an alternative we implemented a soft constraint by constructing a penalty contribution f_p to the free energy as

$$f_p = \chi \left[1 - \sum_j h(\eta_j) \right]^2, \tag{9}$$

where χ is a configurable penalty factor.

2.1.5 Nucleation model

To deal with precipitate nucleation occurring during the aging process of the RPV materials, the phase field method needs to be augmented. Phase field is intrinsically fluctuation free and strictly minimizes the free energy in absence of external driving forces. Nucleation processes depend on thermal fluctuation for nuclei to overcome the Gibbs barrier which results from energy penalty of a newly forming interface between nucleus and matrix. Two classes of approaches for implementing nucleation phenomena in phase field can be found in the literature. One approach is adding thermal fluctuations to the order parameter fields (which is implemented in the MOOSE phase field module through the conserved noise classes). The main drawback of this approach is the timestep reduction incurred by adding the short timescale fluctuations to potentially long timescale diffusion processes. This drawback is avoided by the second type of nucleation model in which nuclei above the critical size are directly inserted into the simulation cell, bypassing the nucleus formation step. When a stable nucleus is inserted into the simulation cell all conserved order parameters must retain their total value. In practice that means a depletion zone must be constructed around the nuclei. It is not straightforward to determine how that zone should look in heterogeneous microstructures.

We have developed a free energy based discrete nucleation model for the phase field method to address the issue of conserving concentrations and establishing physical depletion zones. At a nucleation site the

local free energy density of the system is modified with an additional term that is computed from a list of order parameter (concentration or phase) values and their desired target values for the phase that is supposed to nucleate. As a simple first approach the sum of squares of the differences of an order parameter value c_i and its target values $\bar{c}_i$ is multiplied with a prefactor γ and added to the free energy density. This free energy modification coerces the system to form stable precipitates.

$$f_{\text{nuc}} = \gamma \sum_i (c_i - \bar{c}_i)^2 \tag{10}$$

Nucleus positions are randomly chosen according to a nucleation probability density which can be calculated using classical nucleation theory. The nucleus list is updated at each timestep, adding new nucleation sites (if nucleation events were determined to happen) and removing old nucleation sites once they expire. Each nucleation site is kept in the list for a set hold time. The hold time is chosen sufficiently long for the precipitate to form on the diffusive timescale of the system. As the precipitate forms at the target composition the free energy density f_{nuc} goes down to zero at the nucleation site.

The nucleation algorithm is implemented in MOOSE using two user objects and a material class. One user object manages the global nucleus list, which is synchronized between MPI processes. The second user object uses this list to create a map of all quadrature points in the system that are covered by a nucleation site. A customizable radius can be assigned to the nucleation sites to stabilize precipitates of a given minimum size. The material class accesses the map user object to decide whether the free energy should be modified for a given quadrature point. This approach allows to update the map only if the nucleus list has changed or the mesh was adapted, ensuring the best possible performance

2.2 Material models

We have implemented the two phase FeCuMnNi free energy by Koyama *et al.* [16] which provides concentration and temperature dependent free energy functions for the α (bcc) and γ (fcc) phases. The general structure of these free energies is

$$G_c^\phi(c_i, T) = \sum_i G_i^\phi(T) c_i + {}^E G^\phi(c_i, T) + {}^{mg} G^\phi(T) + RT \sum_i c_i \ln C_i, \tag{11}$$

where $\phi = \{\alpha, \gamma\}$ denotes the phase, ${}^E G^\phi$ is the excess heat of mixing, and ${}^{mg} G^\phi$ is the magnetic contribution to the free energy. G_i^ϕ is the free energy of the pure ϕ phase, which is taken from Dinsdale [17]. ${}^E G^\phi$ is defined as

$${}^E G^\phi = \sum_i \sum_{j>i} L_{i,j}^\phi c_i c_j + \sum_i \sum_{j>i} \sum_{k>j} L_{i,j,k}^\phi c_i c_j c_k, \tag{12}$$

with the coefficients $L_{i,j}^\phi$ and $L_{i,j,k}^\phi$ being functions of T and c_i. The magnetic contribution mgG^ϕ is defined as

$$mgG^\phi = RT \ln\left(\beta^\phi(c_i) + 1\right) \cdot f\left(\frac{T}{T_C^\phi(c_i)}\right), \tag{13}$$

with f defined by Hillert and Jarl [18]. For the parameterization please refer to the original publications.

We implemented the α and γ free energy functions using the expression builder module (Section 2.1.3) in the MOOSE phase field module. This allows us to make use of the symbolic automatic differentiation (Section 2.1.1) while retaining high evaluation performance through the use of the just-in-time compilation module (Section 2.1.2).

The α and γ free energy functions are tied together to form a global free energy density model which adds a non-conserved phase order parameter using the multiphase system (Section 2.1.4).

A snapshot of the precipitate formation in a $Fe_{83}Cu_{15}Mn_1Ni_1$ alloy is shown in Figure 3. The formation of copper core particle with a nickel/manganese shell can be observed. A phase transformation from the α to the γ phase can be observed in the fully formed particles.

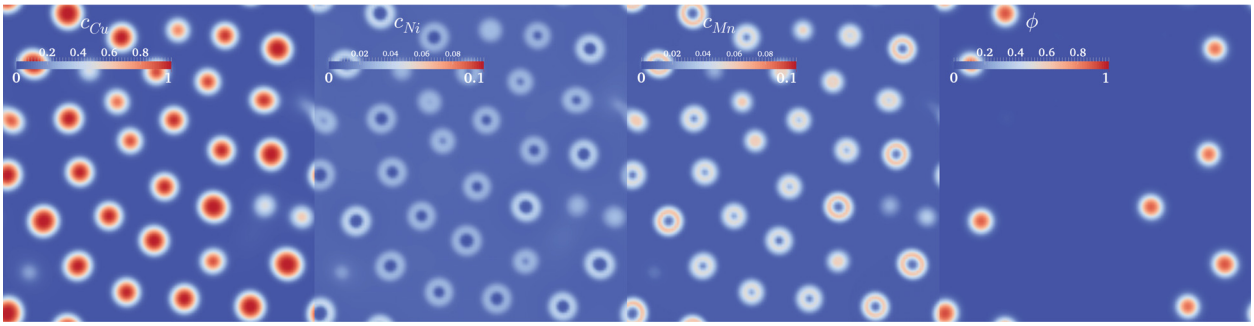

Figure 3: Late stage of precipitation in an $Fe_{83}Cu_{15}Mn_1Ni_1$ alloy. The concentration fields for copper, nickel, manganese, and the phase order parameter distinguishing the α and γ phases are shown in the four panels (left to right).

We utilized the nucleation system (Section 2.1.5) to trigger the nucleation and growth of a copper precipitate in a $Fe_{95}Cu_4Ni_1$ alloy sample at 600 degrees C. Figure 4 shows the evolution of the minimum and maximum of the copper concentration. The maximum Cu concentration reaches 1.0 when a full saturated copper precipitate forms. Figure 5 shows plots of the component concentrations and the free energy densities for the chemical contribution F_c and the nucleation energy penalty F_n in a FeCuNi system as a copper precipitate with a nickel shell is forming.

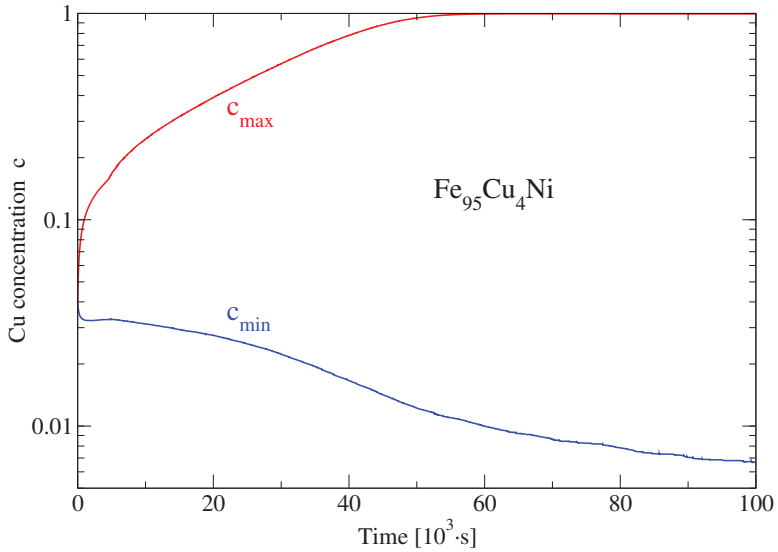

Figure 4: Evolution of the minimum and maximum of the copper concentration in a $Fe_{95}Cu_4Ni_1$ sample during the nucleation and growth of a copper precipitate.

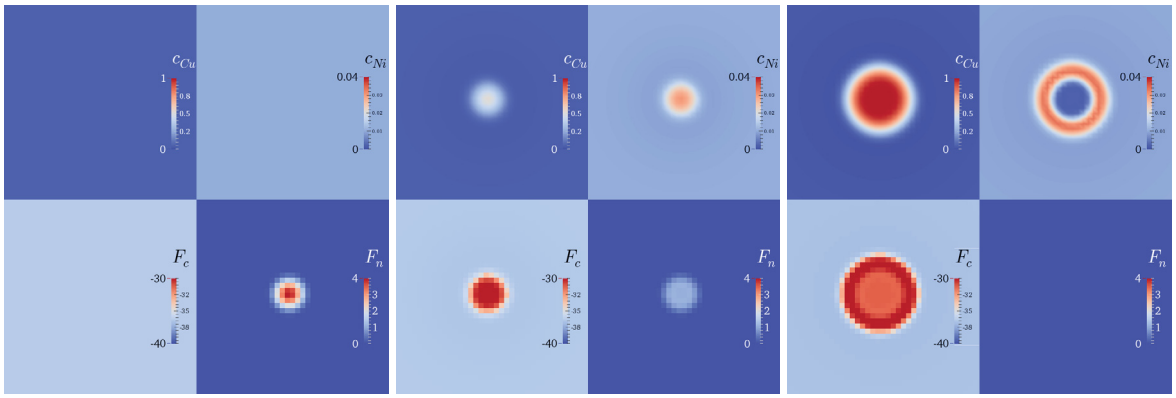

Figure 5: Snapshots at the beginning, middle, and conclusion of the nucleation process in an FeCuNi alloy. The four panels in the three viewgraphs are (clockwise from the top left) copper and nickel concentrations, bulk free energy density, and nucleation penalty energy density F_n in eV/nm^3. As the nucleus is forming, the nucleation penalty vanishes and precipitate continues to grow to minimize the total free energy of the entire system.

3 Coupling of lattice kinetic Monte Carlo and phase field for solute precipitation

3.1 Background

In this section, the coupling of LKMC with PF for solute precipitation is described. The LKMC was developed at INL adopting the SPPARKS framework [19], which was developed at Sandia National Laboratories. For the LKMC method we adopted the residence-time-algorithm where the KMC time t_{KMC} is given by switching of vacancies with surrounding atoms [20]. A detailed description of the LKMC method can be found in our previous reports [12] and the literature. Here only a brief introduction is given. In an LKMC simulation, all atoms (or moving particles) are located on a prescribed, static lattice. Atoms move from one lattice site to a neighboring site via a switching mechanism, usually mediated by vacancies. The interactions between neighboring atoms are governed by pairwise bonds, for which the strength is parameterized using DFT calculations on cohesive energy, energy of mixing, and defect formation and binding energies. At a given temperature, the system evolves by moving of atoms to lower the total energy. Precipitation of solutes takes place by clustering of solute atoms if it is energetically favored. At each LKMC step, only one movement (per processor if in parallel) will be performed, and the KMC time will be advanced by an amount inversely proportional to the total probability of moving summed over all atoms. The KMC time (t_{KMC}) will then be converted to physical time (t_{real}) by rating the vacancy concentration in the simulation over the thermal equilibrium vacancy concentration at the same temperature by: $t_{real} = t_{KMC}\frac{C_v^{KMC}}{C_v^{real}}$. Here C_v^{KMC} is the vacancy concentration used in the simulations, and C_v^{real} is the realistic thermal vacancy concentration, such as that in a thermal aging experiments. At a temperature T, the realistic thermal vacancy concentration can be estimated by $C_v^{real} = exp(-\frac{E^f}{KT})$, with E^f being the vacancy formation energy and K the Boltzmann constant. (In this estimate the contribution of formation entropy, in the order of a few, is ignored.) At each step, the number density, size distribution and spatial distribution of precipitates can be obtained using post-processing.

The LKMC method parameterized using DFT calculations is a powerful tool for modeling coherent precipitation in alloys, such as in the case of RPV steels. However, caution is needed in converting the KMC time (t_{KMC}) to physical time (t_{real}) due to the effect of solute trapping. As solute atoms start to precipitate from the matrix, vacancies become trapped by the precipitates. Once trapped, vacancies hop around the precipitates without contributing to the precipitation kinetics in the matrix, i.e., nucleation of new precipitates or growth of existing ones. Therefore, the physical time should not be advanced since the trapped vacancies can not be taken as thermal vacancies any more. However, the KMC time will still be advanced no matter whether the vacancies are trapped or not. The effect of solute trapping can be corrected by monitoring the fraction of time that each vacancy spends in the matrix in LKMC simulations, fvt [21, 12]. With the correction, the advancement in realistic time t_{real} is now given by $t_{real} = t_{KMC}\frac{fvt*C_v^{KMC}}{C_v^{real}}$. Here fvt measures the fraction of time that vacancies spend in the Fe matrix (sites without Cu atoms being the first or second nearest neighbors).

The method described above can indeed account for the effect of solute trapping regarding converting KMC time to physical time. As shown in Figure 6 (a), soon as the simulation starts, CRPs form immediately, causing substantial dropping in fvt. This means that for the majority of the simulation time, vacancies are trapped by CRPs, without advancing the precipitation process. By including the effect of solute trapping, the LKMC simulation agree well with thermal aging experiments on a Fe-Cu1.34% model alloy on the peak cluster concentration and the time to reach that, as shown in Figure 6(b), indicating the capability of LKMC for modeling the nucleation stage. However, the LKMC simulation fails on the coarsening stage. Due to solute trapping, the precipitation process effectively stopped soon after the nucleation stage, at the end of

which the number of CRPs reaches a peak. As shown in Figure 6(a), due to the very small value of fvt, the realistic time does not evolve any more, meaning that it's nearly impossible to reach the coarsening stage in the LKMC simulations.

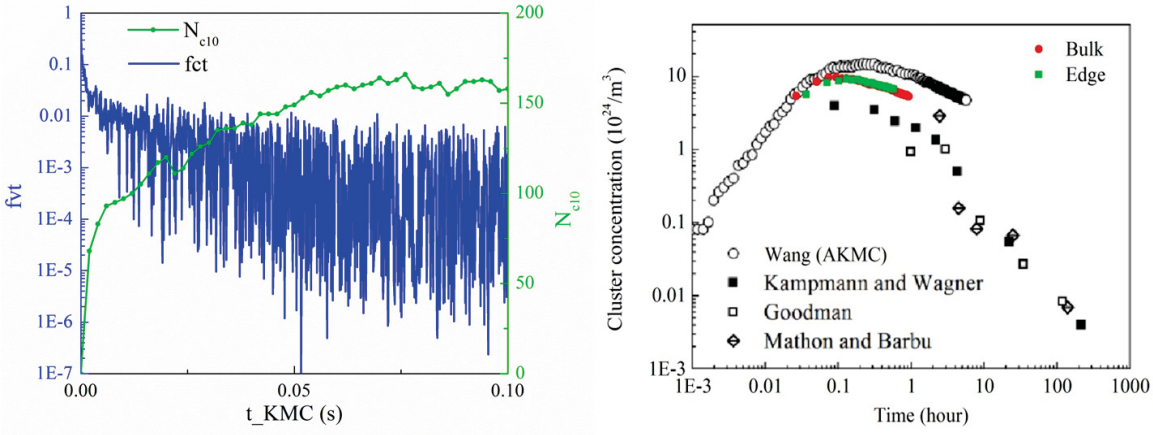

Figure 6: (a) Fraction of time vacancies spent in the Fe matrix and number of CRPs (N_{c10}, CRPs with 10 or more Cu atoms) as functions of KMC time in a LKMC simulation, and (b) Cu cluster concentration as a function of physical time from the LKMC simulation and previous experiments [1, 2, 3]. The simulation was done at 573 K with a Fe-Cu1.34% model alloy. The simulation cell used was 100 by 100 by 100 in unit of lattice constant of Fe, with two million atoms in total.

On the other hand, as a powerful method for microstructural and compositional evolution, the PF method has long been criticized for its performance on nucleation, particularly in systems with dilute concentrations. Even though advanced nucleation models can be used in some cases to facilitate nucleation, PF simulations still reply on other algorithms or theories to obtain the nucleation rate and the critical nucleus size. Therefore, it is highly desired to couple LKMC and PF for solute precipitation. In the coupling, the nucleation of precipitates is described by LKMC. By obtaining the nucleation information from LKMC, including the density, size, and spatial distribution of precipitates, PF will be used to model the subsequent coarsening stage. Such a coupling has been realized in the Grizzly project by coupling SPPARKS and MOOSE. In the following, details about the coupling will be described with a demonstration case using a Fe-Vacancy20% model system.

3.2 Coupling of SPPARKS and MOOSE

The coupling of SPPARKS and MOOSE has been initiated a couple of years ago by another project to couple PF and the Potts model for concurrent grain growth and phase transformation [22]. In that work, a two way information passing has been established between SPPARKS and MOOSE. SPPARKS obtains the concentration field from MOOSE, and uses it to calculate the energy of the system to evolve the spin number and phase index of each lattice site. The spin number can be taken as grain identifiers and the phase index is the order parameter for various phases. They are passed back to MOOSE to be used as parameters in the free energy functional of a PF model for the evolution of the concentration field. SPPARKS and MOOSE interact every MOOSE timestep, with the number of KMC steps determined by a prescribed ratio in the input file.

The aforementioned coupling between MOOSE and SPPARKS has led to interesting results and provided a good example of information passing between these two very different frameworks. SPPARKS is a particle based simulation package adopting discrete lattice sites, while MOOSE solves partial differential equations over continuous domains. Here, the information passing is established by transferring the information on a

lattice site in SPPARKS to a matched node on a MOOSE mesh. Therefore, the information passing is done from a SPPARKS domain to a MOOSE domain of exactly the same size. The coupling has been shown successful in 2D simulations where the square mesh in MOOSE matches the square lattice in SPPARKS.

To meet the needs of the Grizzly project, the two-way coupling has been extended to enable the using of different meshes in MOOSE from the lattice structure in SPPARKS. This was done in the MOOSE framework, and is available to Grizzly as it is based on MOOSE. Bcc Fe (or RPV steel) has a bcc crystal structure. A 3D mesh that matches the bcc lattice currently does not exist in MOOSE. Alternatively, a cubic mesh is used in MOOSE, which matches the corner sites in a bcc lattice unit cell. The coupling needs to be done in 3D to fully represent the bcc crystal structure of Fe. In the previous version of coupling, MOOSE and SPPARKS each solved for different variables: concentrations in MOOSE and spin number and phase index in SPPARKS. After being transferred into MOOSE, the spin number and phase index are treated as auxiliary variables. Here, the concentration fields are solved in both Grizzly and SPPARKS. The transferring can be done on any variables in SPPARKS and Grizzly as long as they are defined in both systems. Moreover, since no physical time is defined in the Potts model [22], the coupling was done on the spatial space only. This is not a problem in coupling LKMC and Grizzly since in both a physical time can be defined. Also, the transfer can be done on a sub-domain of a MOOSE mesh with the SPPARKS simulation domain.

Some changes in the code structure have also been made during the development of the coupling method. Previously, the coupling was done using a UserObject in the ELK directory, which has been removed in the currently version of MOOSE framework. Due to that change, the coupling is now dealt with by a new MOOSE application called Magpie, which was created to handle coupling between any application based on MOOSE and atomistic simulation packages such as SPPARKS and MyTrim [23]. A version of SPPARKS containing the LKMC algorithm has been made publicly available on the GitHub website at `https://github.com/idaholab/SPPARKS` with the permission of the original developers.

In a LKMC simulation, the microstructure information is represented by the atomic configuration on a prescribed lattice, i.e., atoms of various elements on lattice sites. The atomic configuration needs to be translated into concentration fields before transferring into MOOSE. As the lattice sites in LKMC are not one-to-one matched to the nodes in the MOOSE mesh, caution is needed in defining concentration fields in the LKMC model to ensure mass conservation. For the using the cubic mesh in MOOSE and bcc lattice in SPPARKS, a way of defining the atomic concentrations has been derived so that mass conservation is fully satisfied. The way of calculation concentrations fields may need be changed if another mesh structure is used in MOOSE. In SPPARKS, an atomic concentration is defined at each lattice site involving its first and second nearest neighbors. At the site i, the concentration of component α is calculated by:

$$c_i^\alpha = \frac{1}{4} n_i^\alpha + \frac{1}{16} \sum_{j=1}^{8} n_j^\alpha + \frac{1}{24} \sum_{k=1}^{6} n_k^\alpha \tag{14}$$

Here j and k are indexes for the first and second nearest neighbors in a bcc lattice. n_i^α is 1 if the atom at site i is of element α, and 0 otherwise. By defining this way, the concentration at each lattice site has contributions equally from its first and second nearest neighbors. Therefore, either the corner sites or the sites at body center in a bcc lattice can be used to represent the overall concentration field. Transferring the concentration field into Grizzly by matching the corner sites to the nodes of a cubic Grizzly mesh will automatically satisfy mass conservation. The currently LKMC model has been parameterized for bcc Fe with vacancy, Cu, Ni, Mn and Si. For all these elements, a concentration field can be calculated in SPPARKS and transferred into Grizzly. The same capability can be used for any new elements to be included in the future.

To demonstrate the coupling, a simulation using a Fe-Vacancy20% model system is performed at a temperature of 873 K. The 3D simulation domain is 80 by 80 by 4 bcc lattice constant units with 51200 atoms (vacancies). Initially, the vacancies are randomly distributed in the matrix. As shown in Figure 7, at a time of 0.01, the nucleation stage has finished with many nano-sized voids formed. The same configuration is

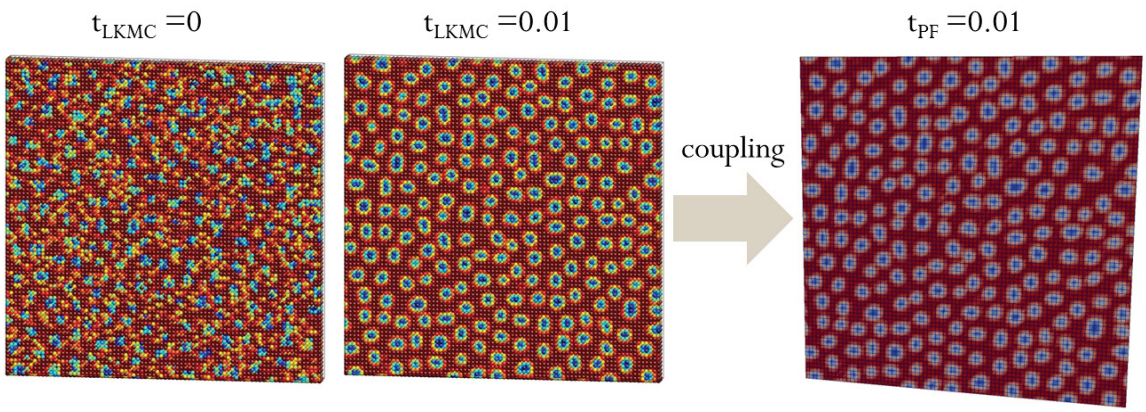

Figure 7: (From left to right) Simulation cells showing the concentration field in LKMC at t=0.0 and t=0.01, and that in Grizzly phase field at t=0.01. The coupling between SPPARKS and Grizzly is done at t=0.01 by transferring the atomic concentration into the concentration field on a Grizzly mesh.

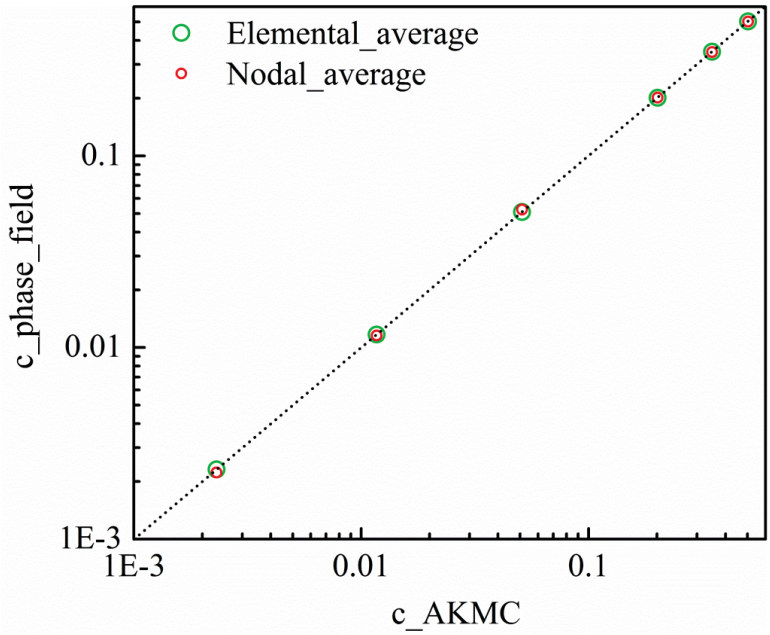

Figure 8: (a) Total mass in Grizzly phase field simulations as a function of that in LKMC simulations showing the mass conservation during coupling.

then transferred into Grizzly for a phase field simulation. As can be seen from the figure, the concentration fields in Grizzly and LKMC agree well with each other.

By defining the concentration using Eq. 14, theoretically mass should be conserved during transferring the atomic concentration from a bcc lattice to a cubic mesh, of which the grid size equals to the lattice constant of the bcc lattice. To further demonstrate this, a few simulations have been done, and the total masses in Grizzly are plotted as functions of those in LKMC simulations. As shown in Figure 8, the masses are accurately conserved during transfers.

3.3 Summary

In summary, in FY15 the previously developed LKMC model has been coupled with Grizzly for phase field simulations of precipitate coarsening. In the coupled framework, LKMC simulations will be used to simulate the nucleation of precipitates, and PF simulations for the subsequent coarsening. The coupling capability has been demonstrated using simulations using the Fe-Vacancy20% model system. Although in the demonstration simulations only transfer from LKMC to Grizzly is shown, the transfer can be two way, and on any properties defined on a bcc lattice. The coupling can be used in both 2D and 3D simulations.

4 Rate theory modeling of radiation damage

In this chapter, we describe our initial efforts to develop the capability to use a rate theory approach to model the radiation damage in RPV materials. In a reactor environment, radiation creates supersaturated point defects such as vacancies and interstitials. These defects are highly diffusive at the reactor operation temperature. If defects diffuse to dislocations, they are annihilated. If they meet defects or clusters of an opposite type (e.g., interstitials meet voids), they recombine with the opposite-type defects. If they meet defects or clusters of same type (e.g., vacancies meet voids), clusters grow. As a result, interstitial loops and voids form during the microstructural evolution under irradiation. The microstructural evolution of radiation damage is a multiscale process, from picoseconds to years. It has been shown that the mean-field rate theory approach is successful for modeling this multiscale process. This method can use the same irradiation conditions as in experiments (e.g., irradiation dose rate, dose, temperature, line dislocation densities in materials) and defect diffusivities from atomistic calculations to predict defect evolution at realistic time scales. The simulation outputs include void and dislocation loop sizes and their size distribution, void swelling, etc., which can be directly compared with experiments. Currently we have implemented two types of rate theory models for modeling radiation damage evolution: a basic Frenkel pair three-dimensional diffusion model (FP3DM) and a more complex cluster dynamics (CD) model.

4.1 FP3DM rate theory modeling

In the FP3DM rate theory model, only isolated Frenkel pairs (mono interstitials and vacancies) are produced and they migrate in a three-dimensional space. Dislocations have bias absorption of interstitials over vacancies. Both voids and interstitial loops are assumed to have fixed densities and their size distribution is uniform. Although this is a basic rate theory model, implementing this FP3DM model is a starting point for developing rate-theory modeling capability in Grizzly. It is also used as a benchmark problem for implementing ordinary differential equation (ODE) based models into partial differential equation (PDE) based MOOSE framework. The full details of the FP3DM model are described in the review paper by Golubov et al. [24]. Here only some key equations are provided to explain our implementation procedure.

In FP3DM, the time evolution of point defect concentration for vacancies (C_v) and interstitials (C_i) can be described by the following rate equations:

$$\frac{\mathrm{d}C_v}{\mathrm{d}t} = K_0 + G_v^{th} - k_v^2 D_v C_v - \mu_R D_i C_i C_v, \tag{15}$$

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = K_0 - k_i^2 D_i C_i - \mu_R D_i C_i C_v, \tag{16}$$

where K_0 represents defect production term which is a product of the dose rate and the cascade efficiency. In FP3DM, equal number of mono interstitials and mono vacancies are produced; G_v^{th} represents thermal vacancy emission rate; k_v^2 and k_i^2 represent total sink strength for vacancies and interstitials, respectively; D_v and D_i are temperature-dependent vacancy and interstitial diffusion coefficients, respectively; μ_R is interstitial-vacancy recombination coefficient. The sink strength has different forms for different types of extended defects. For voids, the sink strength is $k_{void}^2 = 4\pi R_{void}\rho_{void}$, where R_{void} is void radius and ρ_{void} is void density. For line dislocations, $k_d^2 = z_d^{i,v}\rho_d$, where $z_d^{i,v}$ is defect capture efficiency by dislocations. Typically it is larger for interstitials than for vacancies, which is called dislocation bias. For dislocation loops, $k_L^2 = z_L^{i,v} 2\pi R_L \rho_L$, where R_L is loop radius and $z_L^{i,v}$ is defect capture efficiency by dislocation loops. Usually $z_L^{i,v}$ are set to the same values as $z_d^{i,v}$ for interstitials and vacancies, respectively. In this work, all these defect sinks are considered. For recombination coefficient, $\mu_R = \frac{4\pi r_{eff}}{\Omega}$, where r_{eff} is the effective capture radius of a vacancy by an interstitial (i.e., interstitial-vacancy recombination radius) which is about twice of the lattice parameters, Ω is the atomic volume.

By solving the Eqs. (15) and (16), the concentration of interstitials and vacancies as a function of time can be obtained. The void and dislocation loop growth rates can be calculated from the evolution of point defect concentration. For void growth, the rate is

$$\frac{\mathrm{d}R_{void}}{\mathrm{d}t} = \frac{1}{R_{void}}\left[D_v C_v - D_i C_i - D_v C_v^{th} \exp\left(\frac{2\Omega\gamma_{surf}}{R_{void} k_B T}\right)\right], \tag{17}$$

where $C_v^{th} = \exp\left(\frac{-E_f^v}{k_B T}\right)$ is thermal vacancy concentration based on vacancy formation energy E_f^v, γ_{surf} is the surface energy, k_B is Boltzmann constant, and T is temperature. For dislocation loops, both interstitial and vacancy loops can be modeled. However, in this work the vacancy loops are not modeled because it is not clear whether vacancy loops in bcc iron play an important role during irradiation. For interstitial dislocation loop growth, the rate is

$$\frac{\mathrm{d}R_{iL}}{\mathrm{d}t} = \frac{1}{b}\left[z_L^i D_i C_i - z_L^v D_v C_v + z_L^v D_v C_v^{th} \exp\left(-\frac{(\gamma_{sf} + E_{el})b^2}{k_B T}\right)\right], \tag{18}$$

where γ_{sf} is stacking fault energy, b is Burgers vector, and E_{el} is elastic interaction energy between a point defect and a dislocation loop. E_{el} is related to shear modulus, Poisson's ratio, Burgers vector, and loop radius. The expression of the defect-dislocation interaction energy can be find in Ref. [24]. Since the sink strength of voids and dislocation loops is a function of void and loop size, respectively, the growth of voids and loops (Eqs. (17) and (18)) will affect the sink strength dynamically. Therefore, Eqs. (17) and (18) should be fully coupled to Eqs. (15) and (16).

The FP3DM rate theory model has been implemented in Grizzly, using ODE solving capabilities provided by MOOSE. A few new classes are created for implementing this model. Since in FP3DM the equations are ODEs, all the classes for rate theory models are inherited from the ODEKernel class. The base class for FP3DM is called RateTheoryBase which contains some commonly used subroutines such as calculating defect diffusion coefficients at different temperatures, sink strength, thermal vacancy concentration, elastic interaction energy between point defects and dislocation loops, etc. A few child classes derived from the base class are created to calculate the vacancy and interstitial concentration, void and loop sizes (Eqs. (15) - (18)): CvRateTheory, CiRateTheory, VoidGrowthRateTheory, SIALoopGrowthRateTheory. These classes are used as kernels in the input file. Although these kernels are designed to be fully coupled with each other, the coupling can be conveniently turned on or off in the input file. In MOOSE, most differential equations are solved by implicit methods. Therefore Newton method is used to solve the FP3DM equations in Grizzly. Since Eqs. (15) - (18) are fully coupled, both diagonal and off-diagonal elements in the Jacobian matrix are implemented in each kernel. Although the Jacobian matrix is very complex, this implementation helps improve the convergence and accuracy of the solution. A test input file was also created in the Grizzly repository so that users can use the test file as a template to modify the simulation conditions very easily.

To verify the implementation of the FP3DM model in Grizzly, an independent standalone code was also developed to solve the same set of FP3DM equations, but using the explicit 6-order Runge-Kutta integration method. Through the comparison between the explicit-based standalone code and implicit-based Grizzly, we make sure that the FP3DM rate theory model is correctly implemented in Grizzly. In the following we show a few examples of the comparison between Grizzly and standalone code. In these examples, the temperature is 500 K; the dose rate is 1×10^{-5} displacements per atom per second (dpa/s) and the cascade efficiency is 0.366; The dislocation density is 1×10^{13} m^{-2}; The void density and interstitial loop density are both 1×10^{21} m^{-3} and their initial radii are both 1 nm; the defect capture efficiency by dislocations and dislocation loops is 1.04 for interstitials and 1.0 for vacancies.

Figure 9 shows the evolution of the interstitial and vacancy concentration as a function of irradiation dose calculated by Grizzly and standalone code. In this example the growth of voids is turned off but voids still

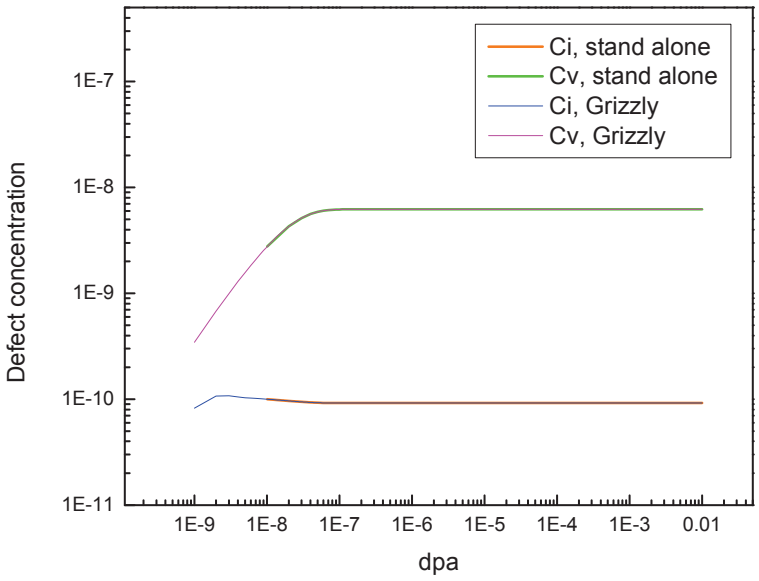

Figure 9: Comparison of the evolution of interstitial and vacancy concentration between Grizzly and standalone code. The rate theory model is based on the FP3DM model.

contribute to the sink strength. Loops are not included in the model. Clearly, Grizzly and the standalone code give identical results despite the fact that they use different integration methods for solving the rate equations. The defect concentration reaches steady state very quickly. The vacancy concentration is typically higher than the interstitial concentration due to the defect bias by dislocations. The steady-state defect concentration is useful for calculating other physical properties such as radiation-enhanced diffusion coefficients.

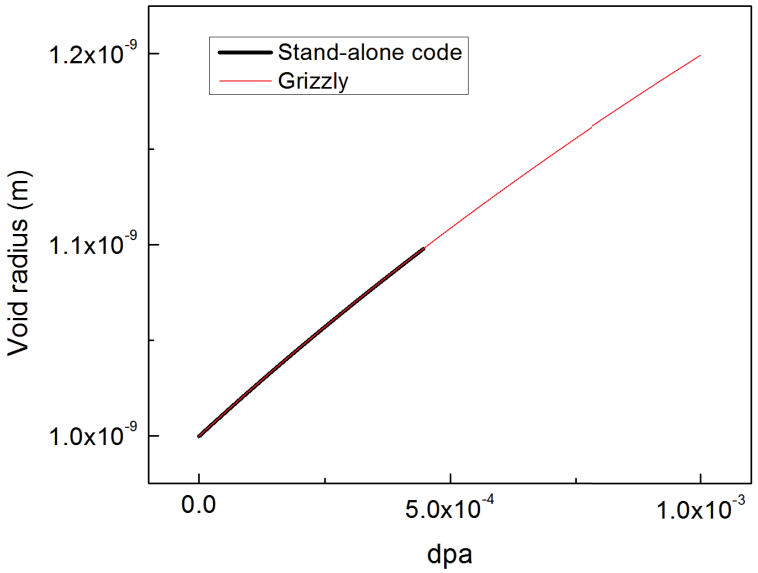

Figure 10: Comparison of the void growth result between Grizzly and standalone code. The rate theory model is based on the FP3DM model.

Figure 10 shows the evolution of void radius as a function of irradiation dose predicted by Grizzly and standalone code. In this example, the void growth is fully coupled with the point defect rate equations. Loops

are not included in the model. Again, the two codes give identical results. The void radius increases almost linearly with the logarithm of irradiation dose. In metals, the void growth contributes to the void swelling directly. Thus from the void growth rate, the volumetric swelling due to void growth can be conveniently calculated.

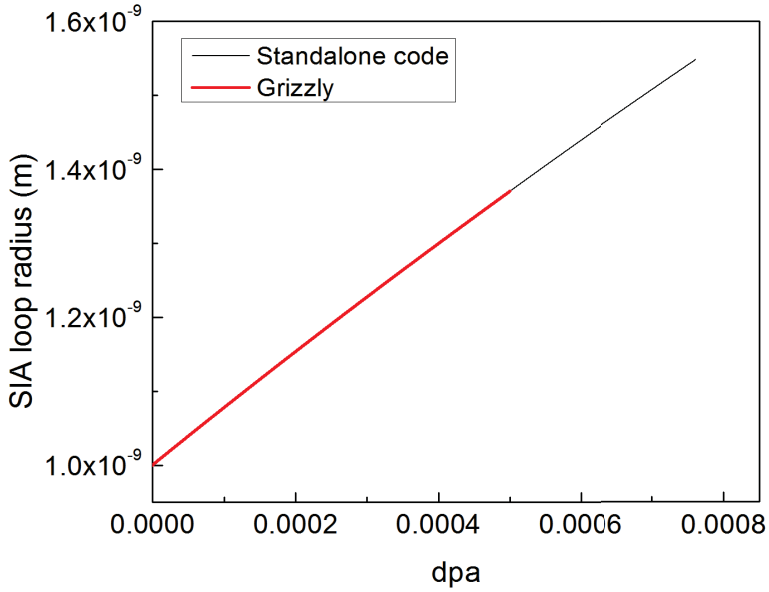

Figure 11: Comparison of the interstitial loop growth result between Grizzly and standalone code. The rate theory model is based on the FP3DM model.

Figure 11 shows the evolution of interstitial loop radius as a function of irradiation dose predicted by Grizzly and standalone code. In this example, both void growth and loop growth are fully coupled to the point defect rate equations. The excellent agreement between Grizzly and standalone code demonstrates that the FP3DM rate theory model implemented in Grizzly is fully verified. The interstitial loop radius increases almost linearly with the irradiation dose within the dose range studied here.

In Grizzly, the input parameters are specified in the input file so that the irradiation conditions and material-specific parameters can be conveniently modified without recompiling the code. Therefore, the code can be conveniently used for studying defect evolution at different irradiation conditions. It also can be extended for studying the radiation damage evolution in other materials if the material-specific parameters are known.

4.2 Cluster dynamics modeling

The FP3DM rate theory model described in the previous section is a simple rate theory model. Although this model has been successfully applied to study radiation damage evolution in many materials, it has many inherent limitations. For example, only point defects (i.e., mono-interstitials and mono-vacancies) are produced in this model. Therefore, this model is suitable for modeling 1 MeV electron irradiation which only produces isolated Frenkel pairs. However, it is well known that under neutron irradiation or heavy-ion irradiation, both point defects and defect clusters are produced in dense cascades. Therefore, it cannot model cascade-induced defect clustering effects. In addition, in the FP3DM model, the void radius and dislocation loop radius are averaged values so that the size distribution of these defect clusters cannot be modeled. Moreover, the nucleation of defect clusters also cannot be modeled. As a result, the defect cluster densities are input parameters and an assumption of the defect cluster densities must be made. To remove these

limitations, a more sophisticated method, cluster dynamics, has been developed. Cluster dynamics is also based on mean-field rate theory, but it can model defect cluster nucleation and growth, defect cluster size distribution, defect clustering from cascades, etc. Therefore, it is a more powerful model than the FP3DM. The cluster dynamics model adopted in this work is based on the model described in Ref. [25]. Here a brief description of this model is provided below.

Figure 12 shows the schematic of the cluster dynamics model for modeling radiation damage evolution. Here cascade can produce both point defects and defect clusters. Mono point defects (interstitials and vacancies) are mobile and they diffuse three dimensionally. Defect clusters are assumed to be immobile for simplicity although they can be mobile in cluster dynamics modeling. When point defects of opposite type meet, recombination occurs. Line dislocations have dislocation bias, i.e., they preferentially absorb interstitials over vacancies. When point defects meet a cluster of the same type, the cluster grows and its size increases by 1. Similarly, when point defects meet a cluster of the opposite type, the cluster shrinks and the cluster size decreases by 1. Vacancy clusters and interstitial clusters also have different thermal stability. Vacancy clusters can thermally emit mono vacancies while interstitial clusters are assumed to be thermally stable.

The cluster evolution can be described as

$$\frac{\partial f(x)}{\partial t} = K(x) + J(x-1) - J(x), x \geq 2, \tag{19}$$

where $f(x)$ is the density of defect cluster for size of x, which is also called the cluster size distribution function (SDF); $K(x)$ is cluster production rate from cascades; $J(x)$ is the cluster flux from size x to x+1. The cluster flux is related to the cluster SDF,

$$J(x) = P(x)f(x) - Q(x+1)f(x+1), \tag{20}$$

where $P(x)$ is cluster absorption rate for absorbing the same type of point defects; $Q(x)$ is the cluster shrinkage rate, either through absorbing defects of opposite type or through thermal emission of the same type of defects. The standard expression of $P(x)$ and $Q(x)$ are provided in Ref. [25]. The equations for mobile point defects are complex. Basically the equations consider defect production by cascades, point defect recombination, defect loss to line dislocations, thermal vacancy emission, defect absorption by all clusters of same type, defect annihilation by all clusters of opposite type, and thermal emission of vacancies from vacancy clusters. The formation and disassociation of dimmers also need special treatment. The full expressions of these processes are described in Ref. [25].

The above cluster dynamics model for radiation damage evolution has been implemented in this work. Similar to the implementation of FP3DM rate theory model, a standalone code has been developed in C/C++ first. In the future, the standalone code can be used as a reference code when the cluster dynamics equations are implemented and solved in Grizzly. The input parameters are the same as those published in Ref. [25], which is for pure iron. In addition, the vacancy formation energy is set to 1.7 eV, which was not provided in Ref. [25]. The maximum cluster size is typically 10,000 (atoms) in our simulations so that there are 20,000 coupled differential equations. These cluster dynamics equations are solved by the SUNDIALS solver [26]. As follows, we present some simulation results from our cluster dynamics modeling.

First the cascade effects are investigated with cluster dynamics, as shown Figure 13. The irradiation temperature is 573 K and the dose rate is 4×10^{-6} dpa/s. When there are no cascades, only Frenkel pairs (mono-interstitials and mono-vacancies) are produced and the cascade efficiency is 1.0 (Note that the total defect production rate is the product of dose rate and cascade efficiency). This situation corresponds to electron irradiation. When there are cascades, the cascade efficiency is 0.4. In addition to the production of Frenkel pairs, defect clusters are also produced by cascades. This scenario represents the radiation damage under neutron irradiation or heavy ion irradiation. Here we use the same assumption as in Ref. [25]: 30% of the total produced defects are in the form of interstitial clusters and vacancy clusters. For interstitial

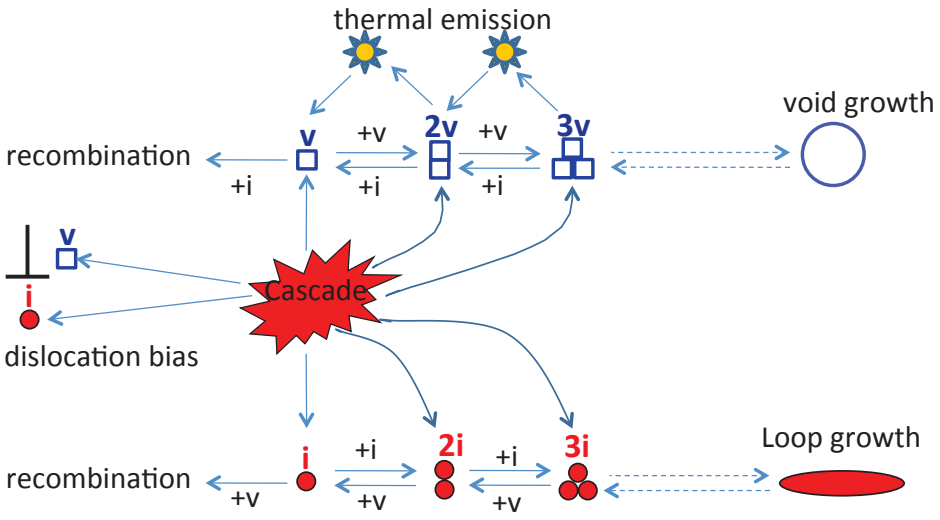

Figure 12: Schematic of the cluster dynamics model for interstitial cluster (loop) and vacancy cluster (void) growth under irradiation.

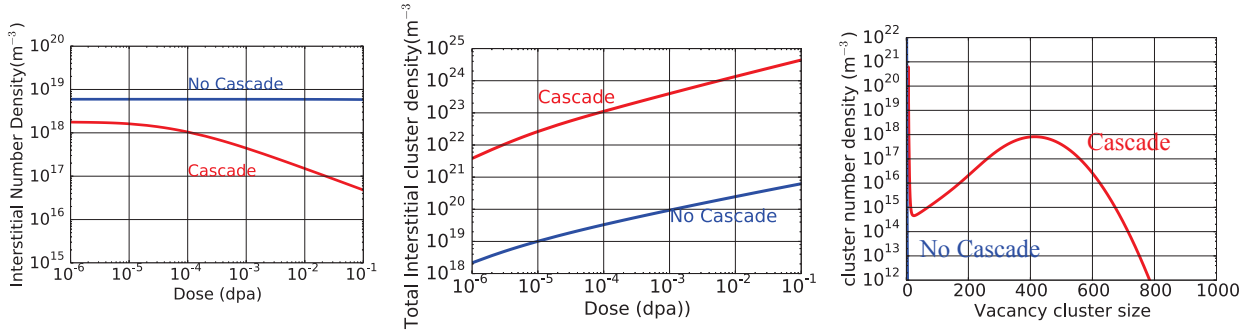

Figure 13: Cascade effects on evolution of radiation damage at 573 K under the dose rate of 4×10^{-6} dpa/s. (a) Evolution of mono-interstitials for two scenarios. (b) Evolution of interstitial clusters. (c) Interstitial cluster size distribution at 0.01 dpa.

clusters, the cluster fractions for 2-interstitial, 3-interstitial, 4-interstitial clusters are 14%, 12%, and 4%, respectively. For vacancy clusters, all the cascade-produced clusters are 6-vacancy clusters. The evolution of mono-interstitials is shown in Figure 13(a) for the two scenarios. When there are no cascades, the concentration of mono-interstitials reaches steady state very quickly. When there are cascades, the concentration of point defects is lower and also decreases with the increasing irradiation dose. However, for the evolution of interstitial clusters (Figure 13(b)), the trend is reversed. The interstitial cluster density is about 3 orders of magnitude higher if cascade effects are included, indicating that the heterogeneous nucleation of defect clusters from cascades facilitates the cluster growth. The cluster size distribution at 0.01 dpa is shown in Figure 13(c) for the two scenarios. If there are no cascades, the interstitial clusters almost have no cluster growth. In contrast, the clusters have a broader size distribution if cascade effects are considered. Therefore, cascades can facilitate cluster growth, which is consistent with our intuitive picture.

The temperature effects on the radiation damage evolution also can be conveniently studied with cluster dynamics. Here three temperatures are investigated: 373 K, 473 K, and 573 K. In all simulations, the dose rate is 4×10^{-6} dpa/s and cascade-induced clustering effects are considered. The density of mono-vacancies

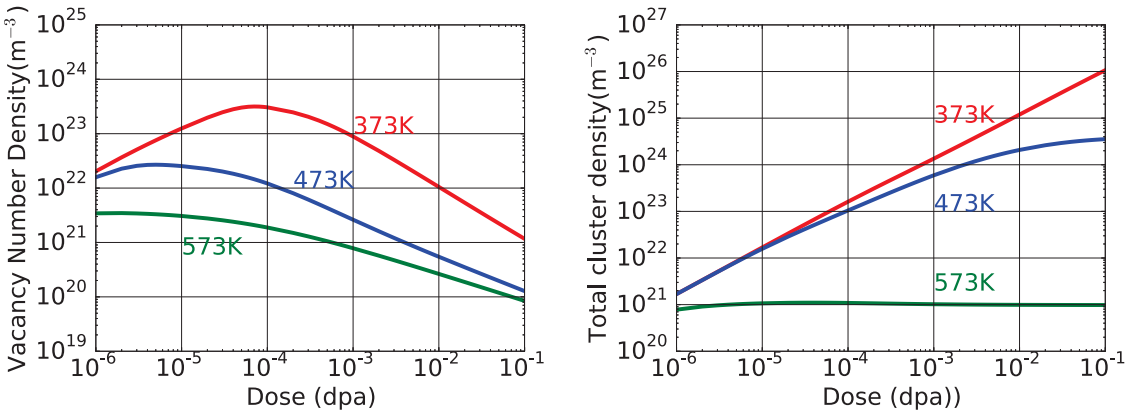

Figure 14: Temperature effects on defect and defect cluster evolution under cascade conditions with the dose rate of 4×10^{-6} dpa/s. (a) Evolution of the density of mono-vacancies at three temperatures. (b) Evolution of the density of vacancy clusters at three temperatures.

as a function of irradiation dose is shown in Figure 14(a) and the evolution of the density of vacancy clusters is shown in Figure 14(b). As temperature increases, the densities of both mono-vacancies and vacancy clusters decrease because the defect recombination and vacancy cluster disassociation also increase. Therefore, the growth of vacancy clusters is suppressed as temperature increases for the present parameter set.

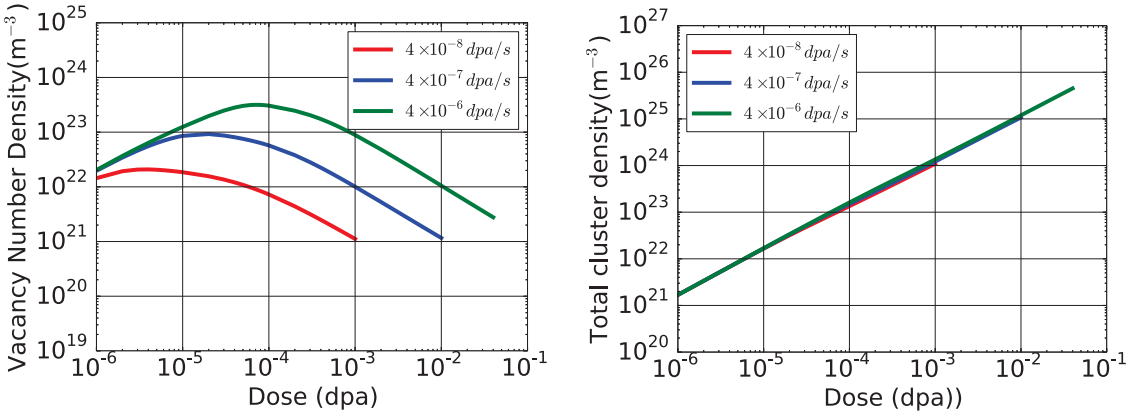

Figure 15: Effects of dose rate on defect and cluster densities under cascade conditions at 373 K. (a) The densities of mono-vacancies as a function of dose at three dose rates. (b) The evolution of the densities of total vacancy clusters at three dose rates.

The effects of dose rate on the damage evolution are also studied with cluster dynamics. Here three dose rates are studied: 4×10^{-8}, 4×10^{-7}, and 4×10^{-6} dpa/s. In all simulations, the temperature is 373 K and the cascade effects are considered. The densities of mono-vacancies as a function of dose at different dose rates are shown in Figure 15(a) and the densities of vacancy clusters as a function of dose are shown in Figure 15(b). As the dose rate increases, the concentration of mono-vacancies also increases. Interestingly, the total density of vacancy clusters is insensitive to the dose rate. Note that this insensitivity may be specific to the present parameter set. In future we will investigate whether this trend changes if different input parameters are used.

One advantage of cluster dynamics over FP3DM rate theory model is that cluster dynamics can model the nucleation of defect clusters. Under cascade conditions, both homogeneous nucleation and heterogeneous in-

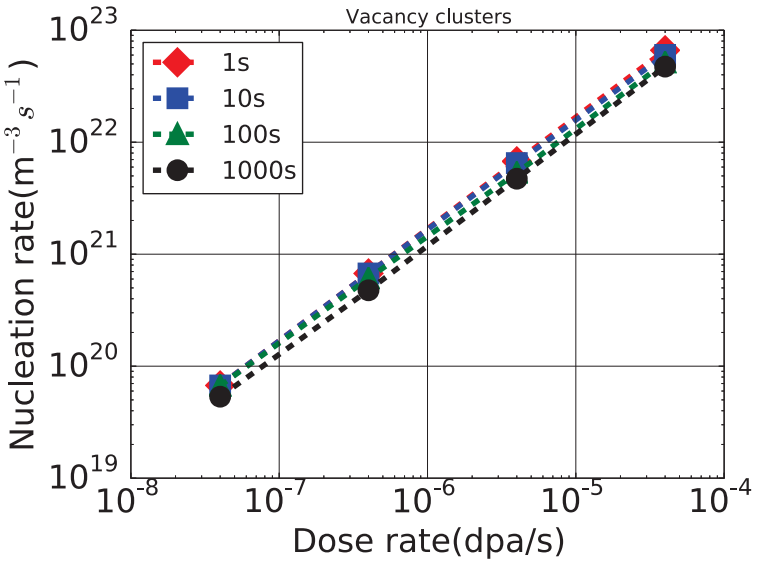

Figure 16: The nucleation rate of total vacancy clusters as a function of dose rate at 373 K under cascade conditions. Both homogeneous and heterogeneous nucleation rates are included in the calculation. At each dose rate, the nucleation rates at four different times are calculated. Note that this is a log-log plot.

cascade nucleation can occur. Cluster dynamics can calculate either the total nucleation rate or the separate nucleation rates. To demonstrate this capability, the total vacancy cluster nucleation rates at four dose rates are calculated under cascade conditions at 373 K, as shown in Figure 16. Here all the clusters with sizes of two or larger are included in the calculation. At each dose rate, the nucleation rates at four different times are calculated. In the log-log plot, the vacancy cluster nucleation rate increases almost linearly with the dose rate. The nucleation rates at different times for a given dose rate are also close to each other, indicating the nucleation reaches nearly a steady state. The nucleation rate calculated from cluster dynamics may be used as input for other methods such as phase field because phase field has difficulties to model the nucleation process when defect concentration is very low (e.g., 1×10^{-8}), which is usually the case for the nucleation of interstitial and vacancy clusters under irradiation.

When vacancies form clusters, they can cause void-induced volumetric swelling. Similar to the FP3DM rate theory model, the void swelling also can be calculated from cluster dynamics. Here we assume that all vacancies in clusters contribute to volumetric swelling. As an example, Figure 17 shows the void swelling as a function of irradiation dose at 473 K under cascade conditions with a dose rate of 4×10^{-6} dpa/s. The swelling increases continuously with dose.

In the standard cluster dynamics modeling, at each cluster size there is one differential equation for each vacancy cluster (same for interstitial clusters). When the maximum cluster size considered in the modeling is large, the computational cost is also high. This will prevent us from studying large clusters or performing simulations for long irradiation times. To overcome these limitations, researchers have developed a grouping method to improve the efficiency of cluster dynamics [27]. In this work, we have implemented this grouping method. Figure 18(a) illustrates the main feature of this grouping method. In this method, the standard discrete method is still used for the cluster size smaller than a threshold size, i.e., one equation for each cluster size. Here the threshold size is set to 100. Beyond this threshold size, several neighboring clusters are grouped into one group. In each group, only the cluster at the upper group boundary has two differential equations. The densities of clusters within each group are linearly extrapolated from the number density of the boundary cluster in that group. To further improve the code efficiency, we also use variable group

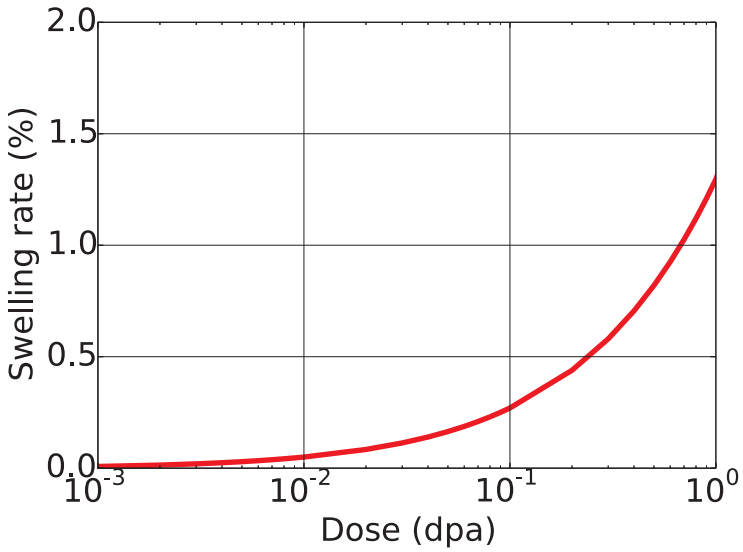

Figure 17: Void swelling as a function of irradiation dose at 473 K under cascade conditions with a dose rate of 4×10^{-6} dpa/s.

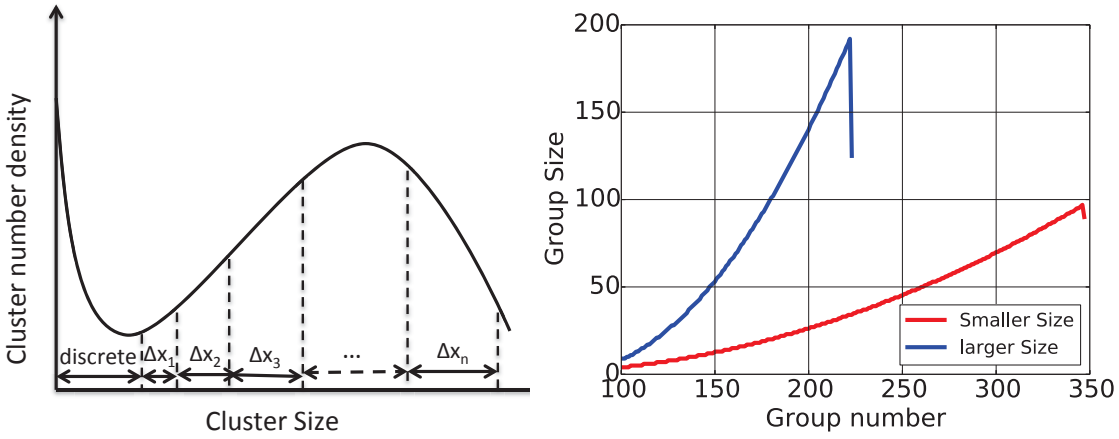

Figure 18: Grouping method for improving the efficiency of solving cluster dynamics equations. (a) Schematic of the grouping method. (b) The group size distributions in two examples.

sizes, i.e., group size increases with group number. In this work, two grouping size distributions are used, as shown in figure 18(b). The grouping method can reduce the number of differential equations to be solved and increase the code efficiency significantly. For example, for the maximum group size of 10,000, the number of differential equations for vacancy clusters is 10,000 in the fully discrete method (same for interstitial clusters). However, for the smaller group size in the grouping method, only 592 differential equations are solved for vacancy clusters. For the larger group size, only 344 differential equations are solved for vacancy clusters.

To demonstrate that the grouping method can reproduce the results from the fully discrete (no grouping) method, we compare the vacancy concentration as a function of dose at three scenarios: no grouping, smaller grouping size, and larger grouping size, as shown in Figure 19(a). Clearly the results are nearly identical in the three scenarios, indicating that the grouping method can reach very good accuracy if the group sizes are chosen appropriately. In addition, the computational times in the two grouping method based simulations are much smaller than the fully discrete method, as shown in Figure 19(b). In particular, the simulation using

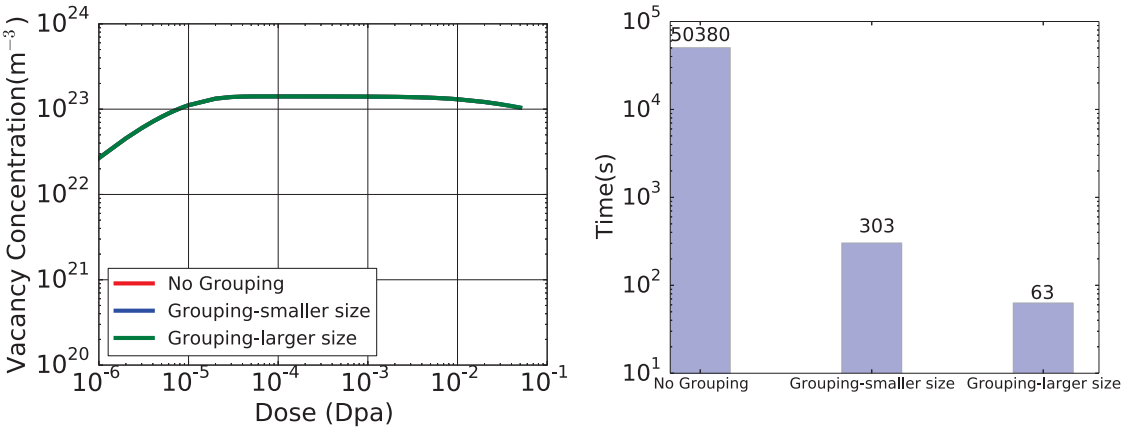

Figure 19: Performance of the grouping method. (a) Comparison of vacancy concentration evolution for three approaches. (b) The computational times for the three approaches. Note that the time is in log scale.

the larger group size is about 800 times faster than the no-grouping method. These results demonstrate that we have successfully implemented the grouping method for improving the efficiency of cluster dynamics modeling.

5 Summary

In FY15, significant progress has been made on model development for microstructure evolution, including rate theory and cluster dynamics for lattice features and coupling lattice kinetic Monte Carlo and phase field for solute precipitation. For lattice features, a rate theory FP3DM model has been developed and integrated into Grizzly for the evolution of the mean size of radiation-induced defects such as interstitial loops and voids. A stand-alone cluster dynamics model has also been developed at INL to obtain the size distribution of lattice features. This model will be merged into the Grizzly package in the future so that engineering scale models can use this information to predict radiation induced hardening and embrittlement. For solute precipitation, a multi-phase, multi-component phase field model has been developed in Grizzly for FeCuMnNi alloys, which are similar to realistic RPV steels. To make up for the deficiency of the phase field method in modeling precipitate nucleation, the previously-developed lattice kinetic Monte Carlo model has been coupled with a Grizzly phase field model to cover both the nucleation and the coarsening stages of precipitation. The coupling has been demonstrated using a Fe-Vacancy20% model system. These developments have substantially advanced the modeling capabilities of Grizzly for RPV microstructure evolution.

In FY16, new efforts on modeling microstructure evolution will focus on: 1) merging the cluster dynamics model into Grizzly; 2) coupling lattice kinetic Monte Carlo and phase field for realistic RPV compositions and operating conditions; 3) investigating radiation enhanced diffusion in RPV steels; 4) coupling phase field and dislocation dynamics for stress-strain behavior in RPV steels; 5) passing information on microstructure evolution to crystal plasticity models for hardening.

6 References

1. R. Kampmann and R. Wagner. *Atomic Transport and Defects in Metals by Neutron Scatterings*. Springer, 1986.

2. S. R. Goodman, S. S. Brenner, and J. R. Low. An fim-atom probe study of the precipitation of copper from iron-1.4 at. percent copper, part-i: field-ion microscopy. *Metall. Trans.*, 4:2363, 1973.

3. M. H. Mathon, A. Barbu, F. Dunstetter, F. Maury, N. Lorenzelli, and C. H. deNovion. Experimental study and modelling of copper precipitation under electron irradiation in dilute fecu binary alloys. *J. Nucl. Mater.*, 245:224, 1997.

4. G.R. Odette and G.E. Lucas. Embrittlement of nuclear reactor pressure vessels. *JOM*, 57:18–22, 2001.

5. R E Stoller. *The Effect of Neutron Flux on Radiation-Induced Embrittlement in Reactor Pressure Vessel Steels*. ASTM International, 2003.

6. D J Bacon, Y N Osetsky, and D Rodney. *DislocationâĂŞObstacle Interactions at the Atomic Level*, chapter 88. Elsevier, 2009.

7. Peter B. Wells, Takuya Yamamoto, Brandon Miller, Tim Milot, James Cole, Yuan Wu, and G Robert Odette. Evolution of manganeseâĂŞnickelâĂŞsilicon-dominated phases in highly irradiated reactor pressure vessel steels. *Acta Mater.*, 80:205, 2014.

8. C. English and J. Hyde. Radiation damage of reactor pressure vessel steels. *Comprehensive Nucl. Mater.*, 4:151–180, 2012.

9. E.D. Eason, G.R. Odette, R.K. Nanstad, and T. Yamamoto. A physically-based correlation of irradiation-induced transition temperature shifts for rpv steels. *J. Nucl. Mater.*, 433:240–254, 2013.

10. B. Radiguet, A. Barbu, and P. Pareige. Understanding of copper precipitation under electron or ion irradiations in fecu0.1 wt% ferritic alloy by combination of experiments and modelling. *J. Nucl. Mater.*, 360:104–117, 2007.

11. Yongfeng Zhang, Paul C. Millett, Michael R. Tonks, Xian-Ming Bai, and S. Bulent Biner. Preferential cu precipitation at extended defects in bcc fe: An atomistic study. *Comp. Mater. Sci.*, 101:181–188, 2015.

12. Benjamin Spencer, Yongfeng Zhang, Pritam Chakraborty, S. Bulent Biner, Marie Backman, Brian Wirth, Stephen Novascone, and Jason Hales. Grizzly year-end progress report benjamin. , Idaho National Laboratory, September 2013.

13. L.Q. Chen. Phase-field models for microstructure evolution. *Annu. Rev. Mater. Res.*, 32:113–40, 2002.

14. Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

15. Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. libMesh: A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254, January 2006.

16. Toshiyuki Koyama, Kiyoshi Hashimoto, and Hidero Onodera. Phase-Field Simulation of Phase Transformation in Fe-Cu-Mn-Ni Quaternary Alloy. *MT*, 47(11):2765–2772, 2006.

17. A. T. Dinsdale. SGTE data for pure elements. *Calphad*, 15(4):317–425, October 1991.

18. Mats Hillert and Magnus Jarl. A model for alloying in ferromagnetic metals. *Calphad*, 2(3):227–238, 1978.

19. S. Plimpton, C. Battaile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, C. Garcia Cardona, and A. Slepoy. Crossing the mesoscale no-man's land via parallel kinetic monte carlo. , Sandia National Laboratory, October 2009.

20. F. Soisson, C. S. Becquart, N. Castin, C. Domain, L. Malerba, and E. Vincent. Atomistic kinetic monte carlo studies of microchemical evolutions driven by diffusion processes under irradiation. *J. Nucl. Mater.*, 406:55–67, 2010.

21. Frederic Soisson and Chu-Chun Fu. Cu-precipitation kinetics in α-fe from atomistic simulations: Vacancy-trapping effects and cu-cluster mobility. *Phys. Rev. B*, 76:214102, 2007.

22. Eric R. Homer, Veena Tikare, Elizabeth, and A. Holm. Hybrid potts-phase field model for coupled microstructuralâĂŞcompositional evolution. *Comp. Mater. Sci.*, 69:414, 2013.

23. D. Schwen, M. Huang, P. Bellon, and R. S. Averback. Molecular dynamics simulation of intragranular Xe bubble re-solution in UO2. *Journal of Nuclear Materials*, 392(1):35–39, 2009.

24. S.I. Golubov, A.V. Barashev, and R.E. Stoller. 1.13 - radiation damage theory. pages 357 – 391, 2012.

25. R.E. Stoller, S.I. Golubov, C. Domain, and C.S. Becquart. Mean field rate theory and object kinetic monte carlo: A comparison of kinetic models. *Journal of Nuclear Materials*, 382(2):77 – 90, 2008.

26. Carol S. Woodward, Daniel R. Reynolds, Alan C. Hindmarsh, and Lawrence E. Banks. SUNDIALS Web page, 2015. https://computation.llnl.gov/casc/sundials/main.html.

27. S. I. Golubov, A. M. Ovcharenko, A. V. Barashev, and B. N. Singh. Grouping method for the approximate solution of a kinetic equation describing the evolution of point-defect clusters. *Philosophical Magazine A*, 81(3):643–658, 2001.