Journal of Computational Physics 229 (2010) 626–641

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A numerical algorithm for the solution of a phase-field model of polycrystalline materials ☆

M.R. Dorr [a,d,*], J.-L. Fattebert [a,d], M.E. Wickett [a,d], J.F. Belak [b,d], P.E.A. Turchi [c,d]

[a] *Center for Applied Scientific Computing, L-561, United States*
[b] *Condensed Matter and Materials Division, L-045, United States*
[c] *Condensed Matter and Materials Division, L-352, United States*
[d] *Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, United States*

ARTICLE INFO

Article history:
Received 18 December 2008
Received in revised form 3 July 2009
Accepted 30 September 2009
Available online 8 October 2009

MSC:
35Q99
65F10
65H10
65M06
65M12
65M20
65Z05

Keywords:
Phase-field model
Polycrystalline microstructure
Method of lines
Newton–Krylov methods

ABSTRACT

We describe an algorithm for the numerical solution of a phase-field model (PFM) of microstructure evolution in polycrystalline materials. The PFM system of equations includes a local order parameter, a quaternion representation of local orientation and a species composition parameter. The algorithm is based on the implicit integration of a semidiscretization of the PFM system using a backward difference formula (BDF) temporal discretization combined with a Newton–Krylov algorithm to solve the nonlinear system at each time step. The BDF algorithm is combined with a coordinate-projection method to maintain quaternion unit length, which is related to an important solution invariant. A key element of the Newton–Krylov algorithm is the selection of a preconditioner to accelerate the convergence of the Generalized Minimum Residual algorithm used to solve the Jacobian linear system in each Newton step. Results are presented for the application of the algorithm to 2D and 3D examples.

1. Introduction

In this paper, we describe an approach for the numerical solution of a phase-field model (PFM) of microstructure evolution in polycrystalline materials. A PFM is a system of equations describing the evolution of continuum representations of material parameters of interest, such as the local state (*e.g.*, liquid or solid), grain orientation and/or species composition. In addition to microstructure evolution [1], PFMs have been applied to a variety of problems involving the evolution of interfaces between spatial domains, including superconductivity [2], phase coexistence [3], solidification [4,5], critical phenomena [6,7], alloy phase ordering [8], recrystallization [9] and martensitic transformation [10]. Correspondingly, PFMs admit

☆ This work performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

* Corresponding author. Address: Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, United States. Tel.: +1 925 422 1064; fax: +1 925 423 2993.

E-mail addresses: dorr1@llnl.gov (M.R. Dorr), fattebert1@llnl.gov (J.-L. Fattebert), wickett1@llnl.gov (M.E. Wickett), belak1@llnl.gov (J.F. Belak), turchi1@llnl.gov (P.E.A. Turchi).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.09.041

many application-dependent variations. The emphasis of this paper is not on models, but instead the numerical solution of a particular PFM representative of models that have been proposed by other investigators.

Rather than explicitly tracking the interfaces between phases, grains, species, etc. to model microstructure development, PFMs evolve diffuse interfaces. The associated equations of motion therefore typically consist of a system of coupled nonlinear diffusion equations. After choosing an appropriate discretization of the spatial variables, the resulting semidiscretized system can be integrated numerically using a method-of-lines approach. Included among the approaches that have been considered for the integration of PFMs are operator splitting, fully explicit methods, semi-explicit methods and fully implicit methods. Operator splitting offers the potential simplification of integrating each equation independently at each time step. However, since a major goal of a PFM integration is to resolve a competition of energy minimization mechanisms represented by each equation in the model, the accuracy and robustness of operator-split approaches are problematic without some reliable means of controlling splitting errors. In addition to these concerns, explicit algorithms suffer the usual time step restrictions imposed by stability requirements. Since PFM diffusion terms result in a time step limit proportional to the inverse square of the mesh size, and a fine mesh may be required to resolve thin interfaces, fully explicit approaches can easily require a prohibitively large number of time steps. For this reason, semi-explicit [11] and fully implicit [12–14] integration algorithms have also been investigated. In [14] it is reported that a second-order implicit time discretization scheme with variable time step size control is considerably faster than an Euler explicit scheme for a 2D binary alloy solidification problem. In [12], a first-order accurate semi-explicit scheme is compared to a fully implicit algorithm in the PFM simulation of dendritic solidification. In spite of the additional significant expense of solving a system of nonlinear algebraic equations at each step, the general conclusion of [12] is that a fully implicit approach is more appropriate for simulations in three dimensions.

A special feature of PFM models of polycrystalline materials is the inclusion of one or more parameters describing local crystallographic orientation. At least two approaches have been employed. In [1], a finite set of orientations is specified, each represented by a different order parameter, effectively one order parameter for each unique solid grain. The total amount of storage required in an implementation of such a model can be reduced by various techniques such as keeping values only for order parameters in regions which have values sufficiently greater than some minimum. Alternatively, local orientation is described by a single parameter whose value may assume a continuous range of orientations. For 2D problems, orientation can be described by a single angle, as developed in [15,16]. This technique was used, *e.g.,* in [17] to simulate austenite to ferrite phase transformation in 2D. In 3D, additional angles can be introduced, or, as described in [18], a quaternion-valued order parameter is employed to avoid issues of singularity and expensive trigonometric function evaluations associated with, *e.g.,* Euler angles.

In this paper, we describe a numerical algorithm for the solution of a PFM that includes a local order parameter, a quaternion representation of local orientation and a species composition. The model and its derivation are presented in Section 2 and the Appendix. The numerical algorithm is described in Section 3. Specifically, we describe the semidiscretization of the PFM system and its implicit integration using a backward difference formula (BDF) temporal discretization combined with a Newton–Krylov algorithm to solve the nonlinear system at each step. The BDF algorithm is combined with a coordinate-projection method to maintain quaternion unit length, which is related to an important solution invariant. A key element of the Newton–Krylov algorithm is the selection of a preconditioner to accelerate the convergence of the Generalized Minimum Residual algorithm used to solve the Jacobian linear system at each step. In Section 4, 2D and 3D example results are presented which illustrate the model capabilities, allow comparison to results published by others, and show some performance metrics. Some conclusions and directions for future work are discussed in Section 5.

2. The phase-field model

On a spatial domain Ω, we begin by introducing the total energy functional

$$F_0 \equiv F_0(\phi, c, \mathbf{q}, T) \equiv \int_\Omega I_0(\phi, c, \mathbf{q}, T) d\Omega, \tag{1}$$

where the energy density, I_0, is

$$I_0(\phi, c, \mathbf{q}, T) \equiv f(\phi, c, T) + \frac{\epsilon_\phi^2}{2}|\nabla\phi|^2 + D_\mathbf{q}(\phi, T)|\nabla\mathbf{q}| + \frac{\epsilon_\mathbf{q}^2}{2}|\nabla\mathbf{q}|^2, \tag{2}$$

where ϕ is a structural order parameter, c is the composition of a particular species (here, we assume a binary material so that $1 - c$ is the composition of the second species), and $\mathbf{q} \equiv (q_1, q_2, q_3, q_4)$ is a quaternion describing local crystallographic orientation (see Appendix A.2), with the normalization

$$\sum_{i=1}^{4} q_i^2 = 1. \tag{3}$$

T is the temperature and is assumed to be uniform across the computational domain Ω in our current model. We further assume that at every point in space we have the possibility of coexistence of a two-phase mixture. We denote these two phases S $(\phi = 1)$ and L $(\phi = 0)$ by reference to the classical problem of solid–liquid mixture, but they can be used to represent various other general two-phase problems. Following the model proposed by Kim et al. [19], we introduce auxiliary variables c_S and c_L that describe the composition in each of the two phases, such that

$$c = h(\phi)c_S + [1 - h(\phi)]c_L, \tag{4}$$

where h is some interpolating monotonic polynomial satisfying $h(0) = 0$ and $h(1) = 1$. For the examples in this paper, we use

$$h(\phi) = \phi^3(10 - 15\phi + 6\phi^2). \tag{5}$$

The free energy density, $f(\phi, c, T)$, in the first term of (2) is defined by the mixture rule

$$f(\phi, c, T) = h(\phi)f^S(c_S, T) + [1 - h(\phi)]f^L(c_L, T) + \omega g(\phi), \tag{6}$$

where f^S and f^L are the free energy densities of the S and L phases, and $g(\phi)$ is a double well potential

$$g(\phi) = 16\phi^2(1 - \phi)^2. \tag{7}$$

The second term of the energy density (2) yields an energy contribution at interfaces between the phases identified by ϕ, with ϵ_ϕ controlling the interface width.

Following [18], the third term of (2) is an orientational free energy where

$$|\nabla \mathbf{q}| = \left(\sum_{i=1}^{4}(\nabla q_i)^2\right)^{1/2} \tag{8}$$

and

$$D_{\mathbf{q}}(\phi, T) \equiv 2HTp(\phi), \tag{9}$$

with H a constant, T the local temperature, and p another interpolating monotonic polynomial satisfying $p(0) = 0$ and $p(1) = 1$. This polynomial should have a positive derivative at $\phi = 1$. We use

$$p(\phi) = \phi^2. \tag{10}$$

We note that we have adopted the opposite convention compared to that used in [18], *i.e.*, we have replaced the polynomial p by $1 - p$.

The final term of (2), involving $|\nabla \mathbf{q}|^2$, is added for numerical reasons, ensuring that interfaces in orientation between different grains are well-resolved, allowing accurate computation of curvature and thus grain boundary motion [15]. The presence and precise form of this term differ in various published models [20,21,15,16,22]. See Appendix A.3 for more information.

We seek to minimize (1) subject to (3). As in [18], we use a Lagrange multiplier to convert the constrained minimization problem to an unconstrained one. This is accomplished by defining the new functional

$$F(\phi, c, \mathbf{q}, T, \lambda) \equiv \int_\Omega I(\phi, c, \mathbf{q}, T, \lambda)d\Omega, \tag{11}$$

where

$$I(\phi, c, \mathbf{q}, T, \lambda) = I_0(\phi, c, \mathbf{q}, T) + \lambda\left(\sum_i q_i^2 - 1\right) \tag{12}$$

is the original energy density (2) augmented by the Lagrange multiplier term. The extrema of (1) then correspond to the critical points of F, which satisfy the Euler–Lagrange equations

$$\frac{\partial F}{\partial \phi} = \frac{\partial F}{\partial c} = \frac{\partial F}{\partial q_i} = \frac{\partial F}{\partial \lambda} = 0. \tag{13}$$

For any particular initial condition $(\phi, c, \mathbf{q}, \lambda)$, it is likely that one or more of the quantities in (13) is non-zero. The essential idea behind the time-evolution phase-field approach is to use the non-zero quantities as source terms in a time-dependent relaxation to a steady state satisfying (13). In particular, for the phase and orientation variables, we postulate the Allen–Cahn equations [3]

$$\dot{\phi} = -M_\phi \frac{\delta F}{\delta \phi}, \tag{14}$$

$$\dot{q}_i = -M_{\mathbf{q}} \frac{\delta F}{\delta q_i}, \quad i = 1, \ldots, 4, \tag{15}$$

where dots denote temporal derivatives, M_ϕ and M_q are mobility coefficients which may be non-constant, and the functional derivatives are computed as described in Appendix A.1. For the composition equation, we postulate the governing equation to be [23]

$$\dot{c} = M_c \nabla \cdot D_c(c, \phi, T) \nabla \frac{\delta F}{\delta c}, \tag{16}$$

where D_c is the diffusivity. In contrast to (14) and (15), this equation evolves the composition c conservatively. We next evaluate the right-hand sides of (14)–(16) individually.

2.1. Phase equation

From (6), (9) and (12), we have

$$\frac{\partial I}{\partial \nabla \phi} = \epsilon_\phi^2 \nabla \phi \tag{17}$$

and

$$\frac{\partial I}{\partial \phi} = -h'(\phi)[f^L(c_L, T) - f^S(c_S, T) - \mu(c_L - c_S)] + \omega g'(\phi) + 2HTp'(\phi)|\nabla \mathbf{q}|. \tag{18}$$

Here, we used the definition of the chemical potential μ from (63) and (70) from Appendix A.4.

Hence, from (14) and the functional derivative formula (57) given in Appendix A.1, we have

$$\dot{\phi} = M_\phi \left\{ \epsilon_\phi^2 \nabla^2 \phi + h'(\phi)[f^L(c_L, T) - f^S(c_S, T) - \mu(c_L - c_S)] - \omega g'(\phi) - 2HTp'(\phi)|\nabla \mathbf{q}| \right\}. \tag{19}$$

In the particular case of a single species material, we have $c = c_L = c_S = 1$ and f^L and f^S are functions of the temperature T only.

In general, M_ϕ may be a function of ϕ itself, as well as its derivatives, and possibly other model variables. For the examples in this paper, M_ϕ will be set to a constant value.

2.2. Orientation equation

From (12), we have, for $i = 1, \ldots, 4$,

$$\frac{\partial I}{\partial \nabla q_i} = \left(\epsilon_\mathbf{q}^2 + \frac{D_\mathbf{q}(\phi)}{|\nabla \mathbf{q}|} \right) \nabla q_i \tag{20}$$

and

$$\frac{\partial I}{\partial q_i} = 2\lambda q_i. \tag{21}$$

Hence, from (15) and (3) we obtain

$$\dot{q}_i - M_\mathbf{q}(\phi) \left\{ \nabla \cdot \left(\epsilon_\mathbf{q}^2 + \frac{D_\mathbf{q}(\phi)}{|\nabla \mathbf{q}|} \right) \nabla q_i - 2\lambda q_i \right\} = 0, \quad i = 1, \ldots, 4, \tag{22}$$

$$\sum_i q_i^2 - 1 = 0. \tag{23}$$

We allow the mobility $M_\mathbf{q}$ to depend on ϕ in order to limit rotation in the ordered phase, further detailed below. Equations (22) and (23) comprise a semi-explicit, differential–algebraic system of index two (see, *e.g.*, [24] for more information about the theory and numerical solution of differential–algebraic systems). Although an algorithm for the integration of such a system could be pursued, it is generally the case that the numerical integration of differential–algebraic systems of index two or higher is facilitated by first reducing the index of the system. In the present case, this is accomplished by replacing (23) by its time derivative and substituting (22), giving

$$0 = 2\sum_i q_i \dot{q}_i = 2\sum_i q_i \left[\nabla \cdot \left(\epsilon_\mathbf{q}^2 + \frac{D_\mathbf{q}(\phi)}{|\nabla \mathbf{q}|} \right) \nabla q_i - 2\lambda q_i \right], \tag{24}$$

which yields an explicit expression for λ. Upon elimination of λ in (22), we obtain the ordinary differential equations

$$\dot{q}_i = M_\mathbf{q}(\phi) \left\{ \nabla \cdot \left(\epsilon_\mathbf{q}^2 + \frac{D_\mathbf{q}(\phi)}{|\nabla \mathbf{q}|} \right) \nabla q_i - \frac{q_i}{\sum_\ell q_\ell^2} \sum_k q_k \nabla \cdot \left(\epsilon_\mathbf{q}^2 + \frac{D_\mathbf{q}(\phi)}{|\nabla \mathbf{q}|} \right) \nabla q_k \right\}, \tag{25}$$

which was originally formulated in [18] (which omits the $\epsilon_\mathbf{q}$ term and includes a noise term).

For any vector $v \equiv (v_1, v_2, v_3, v_4)$, let $\Pi(\mathbf{q})v$ denote the orthogonal projection (with respect to the usual Euclidean inner product) of v onto $\mathbf{q}$. The system (25) can then be written as

$$\dot{\mathbf{q}} = M_{\mathbf{q}}(\phi)[I - \Pi(\mathbf{q})]\nabla \cdot \left(\epsilon_{\mathbf{q}}^2 + \frac{D_{\mathbf{q}}(\phi)}{|\nabla \mathbf{q}|}\right)\nabla \mathbf{q}. \tag{26}$$

In this form, it is clear that solutions of (26) also satisfy the invariant

$$\mathbf{q} \cdot \dot{\mathbf{q}} = 0, \tag{27}$$

which is just a restatement of the differentiated constraint (24). Solutions of (26) with an initial condition on the constraint surface (23) therefore remain on the surface at all times. Differentiation of the constraint (23) has thus replaced the problem of integrating an index two differential–algebraic system with the equivalent problem of enforcing the invariant (27) in the integration of an ordinary differential equation. The equivalent development of an orientation equation of motion via projection onto the tangential plane of the special orthogonal group $SO(3)$ is described in [22].

As with the mobility for the phase equation, the orientation mobility, M_q, may be a general function. It is common to use a functional form that reduces M_q as the phase variable, ϕ, goes to 1 in order to slow or prevent the rotation of ordered grains. For the examples in this paper, we will set

$$M_q(\phi) = M_q^{\min} + m(\phi)\left(M_q^{\max} - M_q^{\min}\right), \tag{28}$$

where $M_q^{\max}$ varies with the problem and $M_q^{\min} = 10^{-6}$, *i.e.*, very near zero, with $m(\phi)$ an interpolating monotonic polynomial satisfying $m(0) = 1$ and $m(1) = 0$. We use

$$m(\phi) = 1 - \phi^3(10 - 15\phi + 6\phi^2). \tag{29}$$

2.3. Composition equation

The particular form of the composition equation depends upon the relationship between the variables (c_S, c_L), and (c, ϕ) in (4). In Appendix A.4, the details for the Kim, Kim and Suzuki (KKS) model are briefly recalled, which results in the following equation of motion

$$\dot{c} = \nabla \cdot D_c^0(\phi, T)\nabla c + \nabla \cdot D_c^0(\phi, T)h'(\phi)(c_L - c_S)\nabla\phi. \tag{30}$$

To actually compute the right-hand side of (30), we need to know $c_S(c, \phi)$ and $c_L(c, \phi)$. For that, we need to know the exact form of f^S and f^L. A specific example for the Hu, Baskes, Stan and Mitchell (HBSM) model of a binary alloy [25] is given in Appendix A.5.

3. Numerical algorithm

We next consider the numerical discretization of the phase-field system given by (19), (26) and (30). Our approach combines a finite-volume spatial discretization with an implicit method of lines temporal discretization.

We begin by introducing a uniform grid on the physical domain Ω and treating the dependent variables ϕ, $\mathbf{q}$ and c as cell-centered quantities with respect to this grid. All divergences are also cell-centered and therefore computable using the divergence theorem and face-centered quantities. The latter are obtained either by averaging the respective cell-centered quantities or by differencing if the quantity is a gradient. The quaternion gradients that appear in the right-hand side of (19) are obtained by averaging face-centered gradients to the cell centers. We note that this sort of finite-volume discretization yields a conservative discretization of the concentration equations (30). For notational convenience in the discussion to follow, we will continue to use continuous spatial operators (*i.e.*, gradients and divergences) to represent their discrete analogs.

Since the diffusion coefficient in (26) becomes unbounded in the limit of small $|\nabla \mathbf{q}|$, we impose a lower bound

$$|\nabla \mathbf{q}| \geqslant \beta > 0 \tag{31}$$

in the evaluation of this coefficient on cell faces. The quantity β is therefore a parameter in the discrete algorithm. The goal in choosing β is to set it small enough so that the diffusion coefficient in (26) is large enough to flatten the components of $\mathbf{q}$ inside the grains (*i.e.*, where ϕ is near unity) while not setting it so small that an unnecessarily fast time scale is introduced. A justification for the use of such bounds based on the theory of semigroups and extended gradient systems is presented in [26]. Although a smoother bound involving a hyperbolic tangent is actually proposed in [26], we have found in our tests that the simpler and less expensive bound (31) is equally effective.

The spatially discretized model can be written as a system of ordinary differential equations

$$\dot{y}(t) = f(t, y(t)), \quad y(0) = y_0, \tag{32}$$

where

$$y(t) \equiv \begin{pmatrix} y_\phi(t) \\ y_{\mathbf{q}}(t) \\ y_c(t) \end{pmatrix} \equiv \begin{pmatrix} \phi(t) \\ \mathbf{q}(t) \\ c(t) \end{pmatrix} \tag{33}$$

and

$$f(t,y(t)) \equiv \begin{pmatrix} f_\phi(t,y_\phi(t),y_{\mathbf{q}}(t),y_c(t)) \\ f_{\mathbf{q}}(t,y_\phi(t),y_{\mathbf{q}}(t),y_c(t)) \\ f_c(t,y_\phi(t),y_{\mathbf{q}}(t),y_c(t)) \end{pmatrix}, \tag{34}$$

where f_ϕ and $f_{\mathbf{q}}$ are the spatially discretized right-hand sides of (19) and (26), respectively, while f_c is the right-hand side of (30).

The inclusion of the orientation components in (32) imposes important requirements for numerical integration. Since the coefficients $D_{\mathbf{q}}(\phi,T)/|\nabla\mathbf{q}|$ in (26) can be large, introducing a potentially fast time scale, an implicit method is recommended for the integration of (32) to avoid the stability-imposed time step limitation of an explicit scheme. Moreover, the solution invariant (27) must be preserved. We note that a simple implicit rule such as backward Euler

$$\mathbf{q}^{n+1} = \mathbf{q}^n + (t^{n+1} - t^n) f_{\mathbf{q}}(t^{n+1},\mathbf{q}^{n+1}), \tag{35}$$

does not preserve this invariant: if $\mathbf{q}^n$ lies on the unit sphere, then $\mathbf{q}^{n+1}$ cannot also lie on the unit sphere and be orthogonal to $f_{\mathbf{q}}(t^{n+1},\mathbf{q}^{n+1})$.

To accommodate the above requirements, we employ variable-order, variable-step backwards difference formulas (BDFs) combined with a coordinate projection. Our choice of integration method is also influenced by the availability of a well-developed software package implementing the algorithms we now summarize. At each discrete time t^n, the use of a BDF results in a nonlinear system to be solved for the discrete solution y^n at time t^n

$$G(y^n) = h_n^{-1}\sum_{i=0}^{k}\sigma_{n,i}y^{n-i} - f(t^n,y^n) = 0, \tag{36}$$

where h_n is the current time step, k is the integration order, and the $\sigma_{n,i}$ are the BDF coefficients. The quantities h_n, k and the $\sigma_{n,i}$ can be chosen adaptively during the integration, based on estimates of the local truncation error and other factors, to maintain stability and achieve accuracy to user-prescribed tolerances. Following the solution of (36), described in more detail below, and the subsequent computation of a corresponding estimate e^n of the local truncation error satisfying a prescribed tolerance, the orientation component $y^n_{\mathbf{q}}$ of the solution y^n is renormalized (projected) onto the unit sphere

$$y^n_{\mathbf{q}} \to y^n_{\mathbf{q}}/|y^n_{\mathbf{q}}| \tag{37}$$

and the orientation component $e^n_{\mathbf{q}}$ of e^n is projected orthogonally onto the subspace orthogonal to the resulting $y^n_{\mathbf{q}}$

$$e^n_{\mathbf{q}} \to e^n_{\mathbf{q}} - y^n_{\mathbf{q}}\cdot e^n_{\mathbf{q}}. \tag{38}$$

The fact that such seemingly *ad hoc* projections do not degrade the stability or accuracy of a BDF integration is proved in [27], in which it is shown that the use of a BDF with coordinate projection is stable if the underlying non-projected method is, and the order of convergence remains the same, including variable order (through sixth-order) and variable step BDFs. For linear multistep methods applied to linear systems, the analysis of [27] concluded that the only error components that matter are those lying in the invariant manifold, so one can (and should) project out the extraneous components of the local truncation error estimates as in (38). In applying coordinate projection to the integration of the particular system (32), since the solution invariant only involves the $y_{\mathbf{q}}$ component, the identity is used in projecting the remaining components y_ϕ and y_c, *i.e.*, the latter components and their corresponding local error estimates are unaffected by the coordinate-projection step.

The nonlinear system (36) is solved using a Newton–Krylov algorithm. Starting with a predicted solution value at the new time step, y^n_{m+1}, an inexact Newton iteration

$$J(\tilde{y}^n)(y^n_{m+1} - y^n_m) = -G(y^n_m) \tag{39}$$

is performed, where $J(\tilde{y})$ is some approximation to the system Jacobian, *i.e.*,

$$J(\tilde{y}) \approx \frac{\partial G}{\partial y}(\tilde{y}) = \frac{\sigma_{n,0}}{h_n}I - \frac{\partial f}{\partial y}(\tilde{y}) \tag{40}$$

evaluated at $\tilde{y}$, which can be the current Newton iterate, y^n_m, an earlier Newton iterate or some other prediction of the solution at time t^n. The Jacobian system (39) is solved using a Generalized Minimum Residual (GMRES) iteration [28]. The advantage of using a Krylov subspace method like GMRES is that only products of the Jacobian matrix $\partial G/\partial y$ times vectors are required, which are computed using finite differences of the system right-hand side f. That is, for an arbitrary vector v

$$\frac{\partial f}{\partial y}(\tilde{y})v \approx \frac{f(\tilde{y}+\sigma v) - f(\tilde{y})}{\sigma} \tag{41}$$

for small σ.

Although the Newton–Krylov approach avoids the need to evaluate and store the Jacobian matrix, a preconditioner is nevertheless required for effective convergence of the GMRES iteration. Since the preconditioner is only required to approximate the system Jacobian, we construct one containing the most dominant terms, which include the diffusive operators whose eigenvalues scale like the inverse square of the mesh size. Specifically, we take

$$P \equiv \begin{pmatrix} P_{\phi,\phi} & 0 & 0 \\ P_{\mathbf{q},\phi} & P_{\mathbf{q},\mathbf{q}} & 0 \\ 0 & 0 & P_{c,c} \end{pmatrix}, \tag{42}$$

where

$$P_{\phi,\phi} \equiv \left\{ \frac{\sigma_{n,0}}{h_n} + M_\phi \omega g''(\tilde{\phi}) \right\} I - M_\phi \epsilon_\phi^2 \nabla^2, \tag{43}$$

$$P_{\mathbf{q},\phi} \equiv (L, L, L, L)^T, \tag{44}$$

$$P_{\mathbf{q},\mathbf{q}} \equiv \begin{pmatrix} K & 0 & 0 & 0 \\ 0 & K & 0 & 0 \\ 0 & 0 & K & 0 \\ 0 & 0 & 0 & K \end{pmatrix}, \tag{45}$$

$$P_{c,c} \equiv \frac{\sigma_{n,0}}{h_n} I - \nabla D_c^0 \nabla, \tag{46}$$

and K and L are the linear operators defined by

$$Kq_i \equiv \frac{\sigma_{n,0}}{h_n} q_i - M_{\mathbf{q}}(\tilde{\phi}) \nabla \cdot \left(\epsilon_{\mathbf{q}}^2 + \frac{D(\tilde{\phi})}{|\nabla \tilde{\mathbf{q}}|} \right) \nabla q_i, \quad i = 1, \ldots, 4, \tag{47}$$

$$L\phi \equiv -M'_{\mathbf{q}}(\tilde{\phi}) \phi \nabla \cdot \left(\epsilon_{\mathbf{q}}^2 + \frac{D(\tilde{\phi})}{|\nabla \tilde{\mathbf{q}}|} \right) \nabla \tilde{\mathbf{q}} - M_{\mathbf{q}}(\tilde{\phi}) \nabla \cdot \frac{D'(\tilde{\phi})\phi}{|\nabla \tilde{\mathbf{q}}|} \nabla \tilde{\mathbf{q}}. \tag{48}$$

Here, $\tilde{\phi}$ and $\tilde{\mathbf{q}}$ denote the components of the vector about which the linearization is being performed. For example, these could be the components of the current Newton iterate or even the solution at a previous time step, depending upon how frequently the preconditioner is being updated. At certain steps in the GMRES algorithm, the solution z of the linear system

$$Pz = r \tag{49}$$

for a given right-hand side r is required, which can be performed using forward block elimination. The only nontrivial step involves the (approximate) inversion of the matrix K. Since K is symmetric and positive definite, a variety of appropriate solvers can be employed. For robustness, we employ a multigrid preconditioned conjugate-gradient algorithm.

We have implemented the algorithm just described in a research code called AMPE, which was used to obtain the example results presented in Section 4. In AMPE, we employ the general-purpose integrator CPODES to integrate the system (32). CPODES solves systems of ordinary differential equations with invariants using the combination of BDF, coordinate projection, and Newton–Krylov type algorithms summarized above. CPODES is closely related to the predecessor CVODE integrator, primarily adding the coordinate-projection capability. Distributed as part of the *Sundials* [29] suite of time integrators and nonlinear solvers, CVODE uses linear multistep methods to integrate stiff or nonstiff systems of ordinary differential equations, automating the problem-independent portions of local error estimation, step size and order selection, and nonlinear solves. Following a beta test period and the creation of appropriate documentation, the recently developed CPODES integrator will also be available as part of the *Sundials* suite. For the solution of the linear systems in (49), we employ a multigrid preconditioned conjugate-gradient solver from the *Hypre* library [30].

A complication in the use of Newton iteration for the solution of the nonlinear Eq. (36) is the presence of the $|\nabla \mathbf{q}|$ factors in the phase Eq. (19) and the diffusion coefficient of the orientation Eq. (26), since these factors are not differentiable at $\nabla \mathbf{q} = 0$. Even when a smooth lower bound is placed on $|\nabla \mathbf{q}|$ to maintain a finite diffusion coefficient in (26), the evaluation of an approximate Jacobian product via finite differencing as in (41) can still result in the generation of poor search directions for the Newton root-finding algorithm. To avoid the finite differencing of the non-differentiable $|\nabla \mathbf{q}|$ factors, we simply suppress them during the computation of the Jacobian-vector products. Specifically, we ensure that both function evaluations in the calculation of the finite difference (41) are performed using the same value of $|\nabla \mathbf{q}|$. This removes the contribution of the $|\nabla \mathbf{q}|$ term in the phase equation to the Jacobian product. Compared with the time scales embedded in the discrete diffusion terms, neglecting the latter term in this manner does not significantly affect the accuracy of the Jacobian approximation. The elimination of this term is also consistent with the preconditioner described above, where the (1,2) block $P_{\phi,\mathbf{q}}$ is zero as well. The latter is important, since it enables (49) to be solved by forward elimination rather than a more complicated block solve. Lagging $|\nabla \mathbf{q}|$ in the quaternion diffusion coefficient results in a term that is linear in $\mathbf{q}$, and therefore easily and accurately (except for the small lagging error) differentiated by the finite difference (41).

4. Numerical examples

4.1. Idealized grain growth and coarsening in 2D

As an example of the use of the full quaternion-orientation parameter, we simulate the growth of a random distribution of multiple non-overlapping solid ($\phi = 1$) grains of randomly varying orientation within a disordered region ($\phi = 0$). The problem geometry is two-dimensional (2D), although the orientation is three-dimensional (3D). The initial condition and two snapshots at later times are shown in Fig. 1. The four components of the quaternion have been mapped to a color map of RGB plus alpha. This is analogous to the thin film example of Fig. 1 in [22].

The physical domain is a square with side length of 2, discretized with 256 cells in each direction. The parameters for this simulation are set to idealized values to display representative behavior, rather than attempting to simulate any particular physical system, and are shown in Table 1. Nominally, values have been set to order 1, then adjusted for balance between phase energy and misorientation energy and adequate resolution of interfaces at this mesh spacing. Orientation mobility is set high enough to avoid impeding the early growth of grains.

There is an initial rapid period of grain growth as the grains expand through solidification into the disordered region. As the grains impinge on one another, the misorientation between them causes interfaces to form. It can be seen from Fig. 1 that the interface width between grains varies with the relative degree of misorientation of those grains. The variation of the orientation mobility with phase from (28) prevents grains from rotating, so the subsequent coarsening of the grain structure leads to some grains shrinking and subsequently disappearing, which happens on a much longer timescale than the initial growth.

4.2. Ostwald ripening of a binary alloy in 2D

We next illustrate the application of our algorithm to the study of structure evolution in binary alloys with a simulation of Ostwald ripening. Ostwald ripening is a phenomenon observed in solids and liquids in which a precipitate grows at the expense of smaller precipitates that have higher solubility into the bulk matrix of the material. Phase-field modeling has proven to be an ideal tool for the study of such processes [31].

Adopting the model and parameters proposed by Hu, Baskes, Stan and Mitchell (HBSM) in [25] (see also Section A.5 of the Appendix), we consider two spherical particles in a face-centered cubic (fcc) phase evolving in a body-centered cubic (bcc) matrix. Parameters related to the misorientation energy are determined in such a way that the single-grain growth are similar to the HBSM model. The growth is diffusion controlled, and the mobilities M_q and M_ϕ are chosen as small as possible, but sufficiently large, such that the grain growth is controlled by the composition equation. Using larger values for those param-

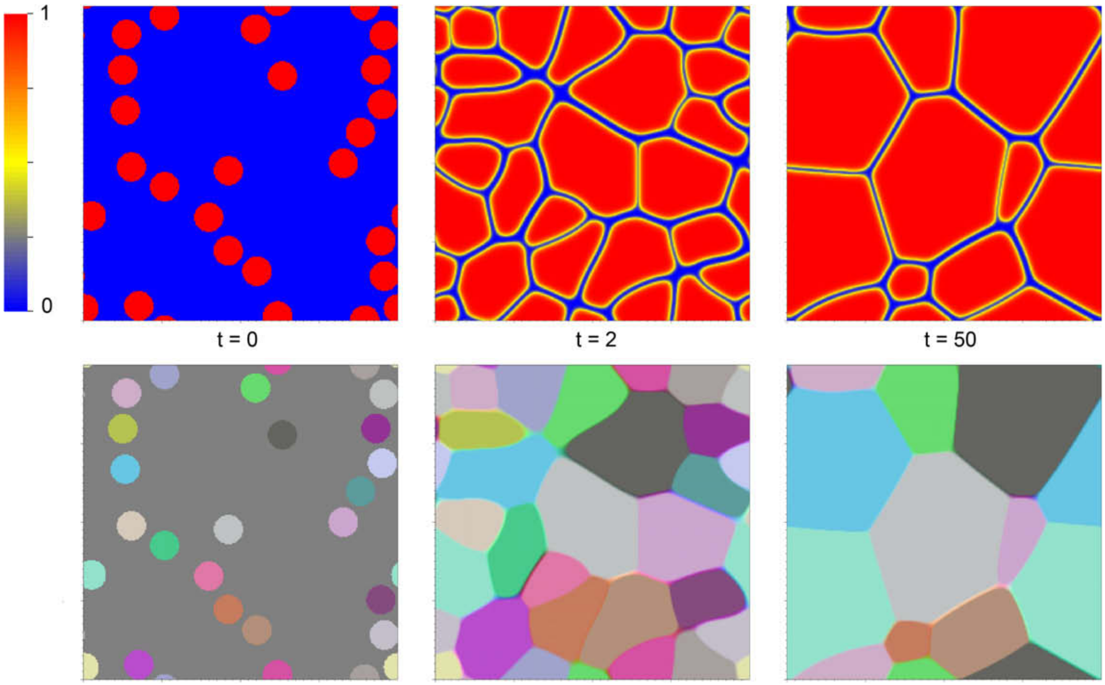

Fig. 1. Idealized 2D grain growth and coarsening: three snapshots of time evolution of phase (top) and orientation (bottom) variable at $t = 0$ (initial condition), $t = 2$, $t = 50$. The different colors indicate different grain orientations.

Table 1
Simulation parameters for idealized grain growth and coarsening. t, E, L, T denote a consistent set of time, energy, length and temperature units.

ϵ_ϕ	$0.1\ (E/L)^{1/2}$
ϵ_q	$0.02\ (E/L)^{1/2}$
f^L	$5.0\ E/L^3$
f^S	$0.0\ E/L^3$
ω	$2.5\ E/L^3$
T	$1.0\ T$
M_ϕ	$1.0\ t^{-1}E^{-1}$
$M_q^{\max}$	$10.0\ t^{-1}E^{-1}$
H	$1.0\ E/TL$

Table 2
Values of parameters used in the binary alloy simulation (adapted from [25] using a molar volume of 1.5×10^{-5} m^3 mol^{-1}).

Parameters used in the binary alloy simulation	
ϵ_ϕ	$0.165\ (\text{pJ}/\mu\text{m})^{1/2}$
ϵ_q	$0.1\ (\text{pJ}/\mu\text{m})^{1/2}$
ω	0.4125 pJ μm^{-3}
D^0_ϵ	$56,000\ \mu\text{m}^2\ s^{-1}$
D^0_δ	$1.3 \times 10^8\ \mu\text{m}^2\ s^{-1}$
Q^0_ϵ	55.29×10^3 J mol^{-1}
Q^0_δ	156.4×10^3 J mol^{-1}
M_ϕ	$200\ s^{-1}$ pJ^{-1}
$M_q^{\max}$	200 s^{-1} pJ^{-1}
H	10^{-3} pJ K^{-1} μm^{-1}
A_ϵ	666 pJ μm^{-3}
A_δ	666 pJ μm^{-3}

eters does not influence the grain growth, but would make the whole problem stiffer, reducing the time step. The diffusion coefficient for the composition equation is given by

$$D^0_c(\phi, T) = h(\phi)\widetilde{D}_\delta(T) + [1 - h(\phi)]\widetilde{D}_\epsilon(T), \tag{50}$$

with

$$\widetilde{D}_\varphi(T) = D^0_\varphi \exp(-Q^0_\varphi/RT) \tag{51}$$

for $\varphi = \delta(\text{fcc}), \epsilon(\text{bcc})$, and where R is the gas constant. Values for $D^0_\delta, D^0_\epsilon, Q^0_\delta$ and Q^0_ϵ have been measured experimentally [32,33]. Parameter values used in our simulation are summarized in Table 2.

The physical domain is a 6.4 μm × 6.4 μm square and is discretized using a 128 × 128 mesh. As mentioned above, the problem is initialized with two fcc grains ($\phi = 1$) embedded in a bcc region ($\phi = 0$). The temperature is fixed at 873 K. The composition c of the alloying element is initialized with its equilibrium fcc phase value inside the grains ($c = 0.10$). The bcc region is initialized with a uniform composition value above equilibrium ($c = 0.06$ vs. 0.05), and thus constitutes a source of the alloying element for the two grains to grow. The two grains are given two of the three possible orientations for fcc phase obtained from bcc through the inverse Bain distortion: $\mathbf{q} = (\cos(\pi/8), \sin(\pi/8), 0, 0)$ and $\mathbf{q} = (\cos(\pi/8), 0, \sin(\pi/8), 0)$, while the bcc region is initialized with $\mathbf{q} = (1, 0, 0, 0)$. Before starting the simulation, we let the variable $\mathbf{q}$ diffuse in the bcc region by evolving the model with $M_\phi = M_c = 0$ and $H = 0$ (see Appendix A.3).

As shown in Fig. 2, the two grains grow until they enter into contact. Subsequently, the larger grain keeps growing while the smaller one shrinks and eventually disappears. At equilibrium, the fraction of each phase (fcc or bcc) is given by the lever rule [34].

4.3. 3D grain growth and coring in a binary alloy

Coring is a well-known phenomenon in alloys [35] that illustrates how the competition between thermodynamic and kinetic driving forces can impact the composition field of off-equilibrium microstructure. Consider the two-phase ($\epsilon + \delta$) region in an A–B binary alloy phase diagram (*e.g.*, liquid + solid or solid + solid as in the present example) illustrated in Fig. 3. Under equilibrium conditions, at T_0 the δ phase starts nucleating and takes from the ϵ phase a disproportionately large amount of the B component, causing the ϵ phase to become richer in A component, compared with the nominal alloy composition c_0. As temperature decreases, the trajectories of the composition field for the ϵ and δ phases evolve along the (solid) lines indicated by single arrows in Fig. 3. The ϵ phase remains richer in A species and the δ phase must also move toward

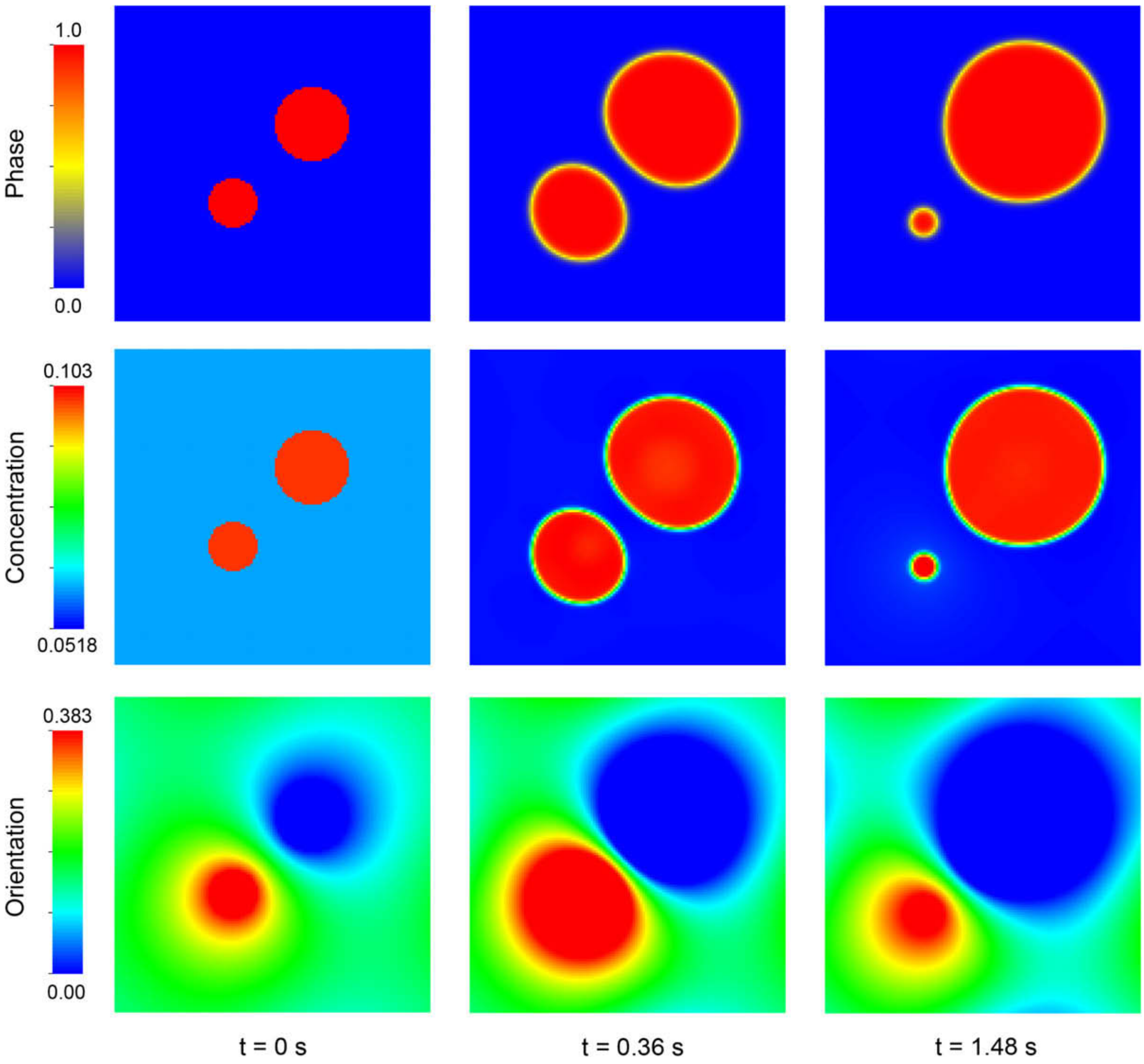

Fig. 2. 2D Ostwald ripening in alloy: phase (top), concentration (middle), 2nd component of orientation (bottom), with snapshots at time $t = 0$, 0.36, and 1.48 s. Phase of 0 and 1 correspond to bcc and fcc phase, respectively.

higher A content, and this can only happen by the preferential adsorption of A from the ϵ phase and the diffusion of A species into the δ phase formed previously at higher temperature. Because of the strong adjustment in composition in the two phases that needs to occur, compared with the nominal alloy composition c_0, departure from equilibrium is to be expected when ordinary cooling rates are used. In the example about to be considered, it can be assumed that homogeneous equilibrium is maintained in the ϵ phase but not in the δ phase where the rate of diffusion is much lower, i.e., $\widetilde{D}_\epsilon \gg \widetilde{D}_\delta$. Hence a cored structure (gradient of composition) develops inside the δ grains with an average composition that evolves along the dotted line (with double arrow) in Fig. 3, whereas the ϵ matrix is compositionally homogeneous.

Before we illustrate the application of PFM to coring, let us first consider the 3D growth of a single spherical fcc (δ) grain of diameter 0.6 μm at $T = 873$ K embedded in a bcc (ϵ) matrix. As in the 2D example, compositions are initially set to 0.10 inside the fcc grain and 0.06 in the bcc region, and the parameter values listed in Table 2 are used. The computational domain is a cube of size 12.8 μm × 12.8 μm × 12.8 μm discretized with a 256 × 256 × 256 uniform mesh. When we let the system evolve, the grain grows and we measure its diameter as a function of time (Fig. 4). After an initial stage during which the initially sharp grain boundary is smoothed out, a growth rate proportional to $t^{1/2}$ is observed, as expected for a phase-field model in 3D [36].

Next we consider the more general case of several fcc grains growing together in a bcc matrix. As an initial condition, we generate twenty non-overlapping spherical grains located at random positions in a 12.8 μm × 12.8 μm × 12.8 μm domain, each with one of the three possible orientations for the δ phase. As in the previous grain growth problem, each grain has an initial nucleated radius of 0.6 μm and a composition $c = 0.10$, while the rest of the domain comprised of ϵ phase has an initial composition $c = 0.06$. The problem is discretized on a 256 × 256 × 256 uniform mesh.

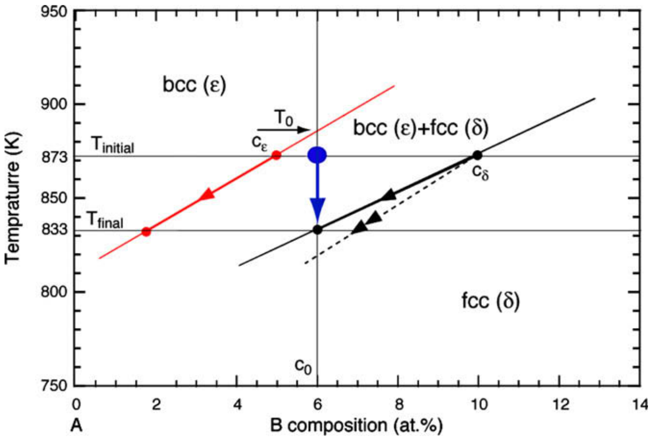

Fig. 3. Schematic representation of coring in a two-phase A–B binary alloy (HBSM model). The data are those that have been used in the PFM-based calculations.

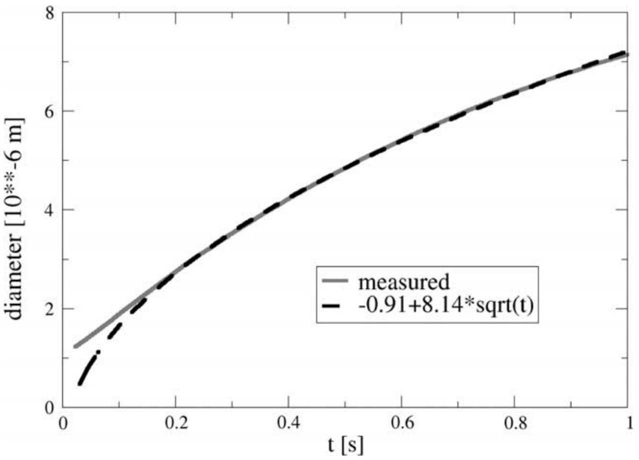

Fig. 4. Binary alloy: growth rate for an fcc spherical grain in a bcc matrix at 873 K.

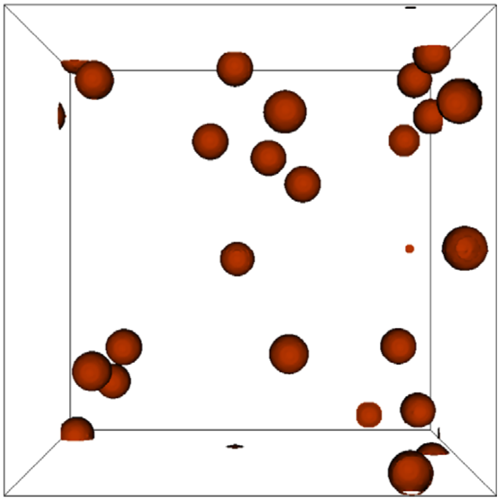

Fig. 5. Initial fcc grains (nuclei) in a bcc matrix for a 3D alloy simulation.

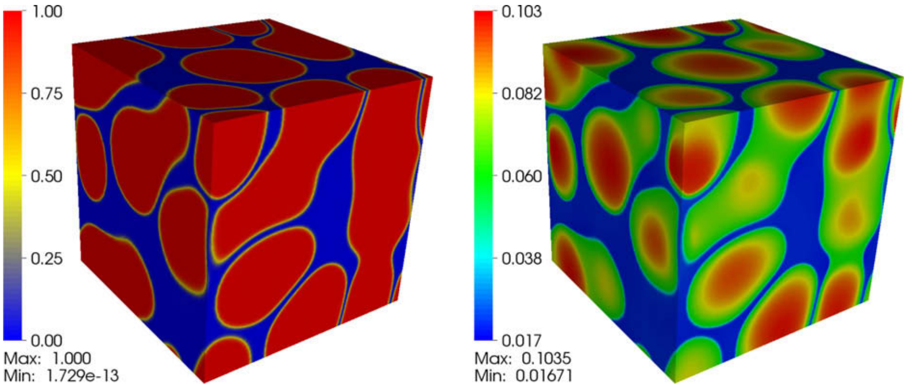

Fig. 6. State after 2 s of a 3D simulation of a binary alloy. Left: phase variable. $\phi = 0$ (resp. $\phi = 1$) corresponds to bcc (resp. fcc) phase. Right: composition variable.

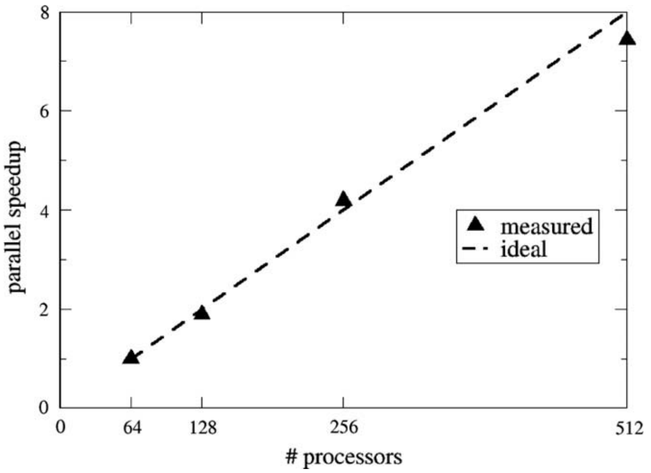

Fig. 7. Parallel speedup for the 3D binary alloy structure evolution problem. The discretization mesh is 256 × 256 × 256. System: *Thunder* Linux cluster, Lawrence Livermore National Laboratory, Quadrics interconnect, Intel IA-64 Itanium 2 1.4 GHz (5.6 GFlops) processors.

The initial temperature is 873 K, which is inside the bcc + fcc region of the phase diagram for a 6% composition. We use a cooling rate of $dT/dt = -20$ K/s and carry out a run for 2 s physical time. At this time the temperature reaches 833 K, which is the boundary between the two-phase bcc + fcc region and the single-phase fcc region in the phase diagram (cf. Fig. 3). This run remains in the temperature range over which the HBSM model is valid. The initial grain distribution is shown in Fig. 5. The phase variable and composition after 2 *s* are shown in Fig. 6. One can observe coring due to the relatively slow diffusion of solute inside the grains. Grains of the same orientation merge together once they impinge. The whole calculation took approximately 1400 time steps and 14 h on 64 processors. [1] We set a maximum BDF integration order of 2, which is the order selected by the integrator for every time step except for the initial two steps, which were order 1. Each time step required an average of 1.4 Newton iteration and 3.8 linear iterations.

Finally we look at the parallel scaling of our code. Our parallel implementation is built on the SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) framework [37,38] using a standard spatial domain decomposition. Fig. 7 shows a strong scaling (speedup) result measured for the 3D alloy simulation described above. For this fixed problem size, there is a roughly 7.5× speedup with an 8× increase in the number of processors from 64 to 512.

5. Conclusions

We have developed a numerical algorithm for the solution of a phase-field model of binary materials that includes a quaternion representation of local crystallographic orientation. Our approach uses a finite-volume spatial discretization combined with an implicit BDF-Newton–Krylov temporal discretization. The BDF algorithm incorporates a coordinate-

[1] Quad AMD Opteron Dual Core 2.4 GHz Infiniband Linux cluster.

projection step to maintain unit quaternion length. Some special considerations arising in the use of Newton iteration to solve the nonlinear system at each time step were addressed, as well as a preconditioning strategy for the GMRES solution of the Jacobian system within each Newton step. We have demonstrated the performance of our algorithm on 2D examples of grain growth and Ostwald ripening as well as 3D alloy examples of single-grain growth and multiple-grain growth during cooling. The scalability of our implementation on massively parallel computer systems was also shown.

Our implementation AMPE (Adaptive Mesh Phase Evolution) of the algorithm described in this paper includes the ability to perform adaptive mesh refinement (AMR). AMR enables the interfaces between grains to be resolved using finer mesh than in the interior of grains. Since the inclusion of AMR results in substantial algorithmic modifications beyond those described here, we defer this discussion to a separate article.

Future directions for expanding the model considered in this paper include the addition of elastic energy, anisotropic mobility, additional phase-order parameters for more than two material phases, and additional composition parameters for more than two material components. Including elastic energy requires the solution of an additional equation for displacements, which are assumed to be in equilibrium with the existing equations [39]. Efficient solution of this additional equation is expected to present further challenges. Including additional components and ordered phases is relatively straightforward in terms of numerical implementation, though doing so significantly increases the complexity of the code implementation.

Acknowledgments

The authors are grateful for discussions with Alan Hindmarsh, Barry Lee, Nele Moelans, Bryan Reed, David Richards, Radu Serban, Frederick Streitz, and Ming Tang at LLNL. We also acknowledge the support of the LLNL Laboratory Directed Research and Development Program.

Appendix A

A.1. Functional derivatives

Let

$$F(u) = F(u_1,\dots,u_n,\nabla u_1,\dots,\nabla u_n) = \int_\Omega I(u_1,\dots,u_n,\nabla u_1,\dots\nabla u_n)d\Omega \tag{52}$$

be a functional on a space of function tuples $u = (u_1,\dots,u_n)$ where the u_i are defined and periodic on the domain Ω. By definition, the functional derivative of F with respect to u_i at any point $\hat{u} = (\hat{u}_1,\dots,\hat{u}_n)$ is the functional whose action is defined by

$$\left\langle \frac{\delta F}{\delta u_i}(\hat{u}), v\right\rangle = \frac{d}{d\epsilon}\int_\Omega I(\hat{u}_1,\dots,\hat{u}_i+\epsilon v,\dots,\hat{u}_n,\nabla\hat{u}_1,\dots,\nabla\hat{u}_i+\epsilon\nabla v,\dots,\nabla\hat{u}_n)d\Omega\bigg|_{\epsilon=0} \quad \text{for all } v. \tag{53}$$

Hence, for all v,

$$\left\langle \frac{\delta F}{\delta u_i}(\hat{u}), v\right\rangle = \int_\Omega \left(\frac{\partial I}{\partial u_i}(\hat{u})v + \frac{\partial I}{\partial \nabla u_i}(\hat{u})\cdot\nabla v\right)d\Omega \tag{54}$$

$$= \int_\Omega \left[\frac{\partial I}{\partial u_i}(\hat{u})v + \nabla\cdot\left(\frac{\partial I}{\partial\nabla u_i}(\hat{u})v\right) - \left(\nabla\cdot\frac{\partial I}{\partial\nabla u_i}(\hat{u})\right)v\right]d\Omega \tag{55}$$

$$= \left\langle \frac{\partial I}{\partial u_i}(\hat{u}) - \nabla\cdot\frac{\partial I}{\partial\nabla u_i}(\hat{u}), v\right\rangle, \tag{56}$$

where the second term in (55) vanishes due to the divergence theorem and the assumed periodicity of v. Thus,

$$\frac{\delta F}{\delta u_i} = \frac{\partial I}{\partial u_i} - \nabla\cdot\frac{\partial I}{\partial\nabla u_i}. \tag{57}$$

A.2. Quaternions

By analogy with complex numbers, a quaternion q can be written as a linear combination

$$q = (a + ib + jc + kd) \tag{58}$$

with three imaginary dimensions i,j,k. In the following, we will assume that all of the quaternions are normalized, *i.e.* $a^2+b^2+c^2+d^2=1$. A rotation by an angle θ around an axis of direction (x,y,z) can be described by a quaternion

$$q = \cos(\theta/2) + i[x*\sin(\theta/2)] + j[y*\sin(\theta/2)] + k[z*\sin(\theta/2)], \tag{59}$$

where (x,y,z) is assumed to be normalized. We denote by

$$q^* = a - ib - jc - kd \tag{60}$$

the conjugate of q. Using the quaternion multiplication rules, *i.e.*, $ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j, i^2 = j^2 = k^2 = -1$, the rotation between two quaternions, q_1 and q_2, can be computed as either $q_1^* q_2$ or $q_2 q_1^*$, depending on an arbitrary convention. Due to the non-commutative nature of the quaternion multiplication, these two are not equivalent, but are the conjugate of each other. They correspond to a rotation by the same angle around an axis of opposite direction.

The formula for the misorientation angle between two unit-length quaternions is given by

$$|d_{21}| = 2\sin(\theta_{12}/4), \tag{61}$$

where $|d_{21}|$ is the distance between q_1 and q_2 in the 4D space, and θ_{12} is the angle between q_1 and q_2 (note the factor of 2). The mapping from unit quaternions to rotations is 2-to-1. Multiplying a quaternion by an overall factor of -1 has no physical effect. Two quaternions on opposite sides of the hypersphere are a distance $|d_{21}| = 2$ apart, which gives $\theta_{12} = 2\pi$, which is no rotation at all. To first order, (61) gives

$$\theta \simeq 2|d_{21}| \tag{62}$$

and justifies using quaternion differences to approximate local misorientation [18].

A.3. Misorientation energy

The third and fourth terms of (2) give the free energy due to variation in crystallographic orientation. As described in [16], the free energy density must depend, to lowest order, linearly in $|\nabla \mathbf{q}|$ in order to have stable grain boundaries of finite width. With linear dependence, the misorientation energy is independent of the width of the boundary, though the variation of the coefficient of the first-order term with ϕ and temperature alters this simple relation. If there is only a linear term, sharp interfaces are energetically favored in some conditions.

Because of this, the fourth term of (2), involving $|\nabla \mathbf{q}|^2$, is added for numerical reasons, ensuring that interfaces in orientation between different grains are well-resolved, allowing accurate computation of curvature and thus grain boundary motion [15]. In [20,21], this term is not present at all. In [15] this term is given as in our Eq. (2), though in later publications, *e.g.* [16,22], this term is scaled by a function of ϕ that goes to zero in disordered (liquid) regions. We find it is important to have a non-vanishing second-order term when ϕ is small in order to ensure adequate resolution of orientation gradients in "wet" interfaces (those interfaces in which the minimum value of ϕ is very near zero). Note that this requires the scaling factor $\epsilon_{\mathbf{q}}$ to be sufficiently small to avoid affecting the overall grain boundary energetics, which in turn requires a problem-dependent minimum resolution to achieve. In practice, this means care must be taken to avoid too large a value of $\epsilon_{\mathbf{q}}$ that would alter grain boundary dynamics, while simultaneously ensuring that gradients in orientation are adequately resolved.

Additionally, without a non-vanishing second-order term involving misorientation, it is necessary on physical grounds to have a mobility function for orientation which approaches infinity as ϕ goes to zero (see [15] or [16] in which mobility is effectively scaled as ϕ^{-2}) for the following reason. Since the value of the orientation in liquid regions is not physically meaningful, it should have no effect on the rate of grain growth into a liquid region, irrespective of the actual value chosen in the liquid. In order for this to be true, the "reorientation" from the liquid region orientation value to the solid region value just ahead of the grain boundary must happen effectively instantaneously, or the grain will grow more slowly than it would if the liquid region value were already the same as the solid value. However, the resulting large changes in mobility near the grain boundary can have large repercussions on numerical error, or equivalently, on the allowable time step. Since the value of the orientation in a disordered region has no physical meaning, the non-vanishing second-order term in misorientation allows this reorientation to happen further in front of the grain interface without resorting to excessively high mobility. Likewise, a smooth initial orientation field may be used in the liquid to allow higher time steps during the early simulation time. This can be done by letting the variable $\mathbf{q}$ diffuse in the disordered region by evolving the model with $M_\phi = M_c = 0$ and $H = 0$. The use of this smoothing should have no effect on the solution after an initial transient in which grain boundaries are formed.

A.4. Derivation of the Kim, Kim, Suzuki (KKS) model

In [19], Kim et al. define the relation between the variables (c_S, c_L) and (c, ϕ) in (4) by imposing the condition

$$\left.\frac{\partial f^S}{\partial c_S}\right|_{c_S = c_S(x,t)} = \left.\frac{\partial f^L}{\partial c_L}\right|_{c_L = c_L(x,t)} = \mu(x,t). \tag{63}$$

This means that the chemical potential is equal for both phases at the infinitesimal point x, and thus there would be no change in free energy by exchanging an infinitesimal amount of species between phases S and L.

In the KKS model, we also define

$$D_c(c,\phi,T) = D_c^0(\phi,T)\left(\frac{\partial^2 f}{\partial c^2}\right)^{-1}. \tag{64}$$

From (6) and (63), we obtain

$$\frac{\partial f}{\partial c} = h(\phi)\frac{\partial f^S}{\partial c_S}\frac{\partial c_S}{\partial c} + [1-h(\phi)]\frac{\partial f^L}{\partial c_L}\frac{\partial c_L}{\partial c} = \mu\left(h(\phi)\frac{\partial c_S}{\partial c} + [1-h(\phi)]\frac{\partial c_L}{\partial c}\right) \tag{65}$$

and from (4)

$$1 = \frac{\partial c}{\partial c} = h(\phi)\frac{\partial c_S}{\partial c} + [1-h(\phi)]\frac{\partial c_L}{\partial c}. \tag{66}$$

From (65) and (66), we then get

$$\left.\frac{\partial f}{\partial c}\right|_{c=c(x,t)} = \mu(x,t) \tag{67}$$

(Eq. (28) of [19]).

From (6), we also have

$$\frac{\partial f}{\partial \phi} = h'(\phi)[f^S(c_S) - f^L(c_L)] + \omega g'(\phi) + h(\phi)\mu\frac{\partial c_S}{\partial \phi} + [1-h(\phi)]\mu\frac{\partial c_L}{\partial \phi}. \tag{68}$$

Noticing that, since c is independent of ϕ,

$$h(\phi)\frac{\partial c_S}{\partial \phi} + [1-h(\phi)]\frac{\partial c_L}{\partial \phi} = \frac{\partial c}{\partial \phi} - h'(\phi)(c_S - c_L) = -h'(\phi)(c_S - c_L), \tag{69}$$

we get

$$\frac{\partial f}{\partial \phi} = -h'(\phi)[f^L(c_L) - f^S(c_S) - \mu(c_L - c_S)] + \omega g'(\phi) \tag{70}$$

(Eq. (27) of [19]). We also get

$$\frac{\partial}{\partial c}\left(\frac{\partial f}{\partial \phi}\right) = -h'(\phi)\left[\frac{\partial f^L}{\partial c_L}\frac{\partial c_L}{\partial c} - \frac{\partial f^S}{\partial c_S}\frac{\partial c_S}{\partial c} - \mu\left(\frac{\partial c_L}{\partial c} - \frac{\partial c_S}{\partial c}\right) - \frac{\partial^2 f}{\partial c^2}(c_L - c_S)\right] = h'(\phi)(c_L - c_S)\frac{\partial^2 f}{\partial c^2} \tag{71}$$

using (63). This is Eq. (30) of [19].

Now, since f is a function of c and ϕ,

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial c} = \frac{\partial^2 f}{\partial c^2}\frac{\partial c}{\partial x} + \frac{\partial^2 f}{\partial c \partial \phi}\frac{\partial \phi}{\partial x}, \tag{72}$$

and we have

$$\nabla \mu = \nabla \frac{\partial f}{\partial c} = \frac{\partial^2 f}{\partial c^2}\nabla c + \frac{\partial^2 f}{\partial c \partial \phi}\nabla \phi. \tag{73}$$

From Eqs. (16), (64), (71), and (73), we obtain

$$\frac{\partial c}{\partial t} = M_c \nabla \cdot D_c^0(\phi, T)\nabla c + M_c \nabla \cdot D_c^0(\phi, T)h'(\phi)(c_L - c_S)\nabla \phi, \tag{74}$$

which is Eq. (33) of [19].

To actually compute the right-hand side of (74), we need to know $c_S(c,\phi)$ and $c_L(c,\phi)$. For that, we need to know an explicit form of f^S and f^L. See, for example, Section A.5.

A.5. Hu, Baskes, Stan, Mitchell (HBSM) model for a binary alloy

In [25], Hu et al. propose a phase-field model for a binary alloy. The two phases are ϵ, or body-centered cubic (bcc), and δ, or face-centered cubic (fcc) (to substitute for L and S, respectively, in the preceding section). The following explicit forms for f^ϵ and f^δ are proposed:

$$f^\epsilon(c_\epsilon, T) = A_\epsilon\left[c_\epsilon - c_\epsilon^{eq}(T)\right]^2, \tag{75}$$

$$f^\delta(c_\delta, T) = A_\delta\left[c_\delta - c_\delta^{eq}(T)\right]^2. \tag{76}$$

Using (63), we obtain

$$A_\epsilon(c_\epsilon - c_\epsilon^{eq}) = A_\delta(c_\delta - c_\delta^{eq}) \tag{77}$$

and thus

$$c_\epsilon(c_\delta, T) = c_\epsilon^{eq} + A_\delta[c_\delta - c_\delta^{eq}(T)]/A_\epsilon, \tag{78}$$
$$c_\delta(c_\epsilon, T) = c_\delta^{eq} + A_\epsilon[c_\epsilon - c_\epsilon^{eq}(T)]/A_\delta. \tag{79}$$

From (4), we then get

$$c_\epsilon(c, \phi, T) = \frac{c - h(\phi)\left[c_\delta^{eq}(T) - \frac{A_\epsilon}{A_\delta} c_\epsilon^{eq}(T)\right]}{[1 - h(\phi)] + h(\phi)\frac{A_\epsilon}{A_\delta}}, \tag{80}$$
$$c_\delta(c, \phi, T) = \frac{c - [1 - h(\phi)]\left[c_\epsilon^{eq}(T) - \frac{A_\delta}{A_\epsilon} c_\delta^{eq}(T)\right]}{h(\phi) + [1 - h(\phi)]\frac{A_\delta}{A_\epsilon}}. \tag{81}$$

These expressions for $c_\epsilon(c, \phi, T)$ and $c_\delta(c, \phi, T)$ can be substituted into (74) to have a right-hand side function that depends explicitly on ϕ and c.

References

[1] L.-Q. Chen, Phase-field models for microstructure evolution, Ann. Rev. Mater. Res. 32 (2002) 113–140.

[2] V.L. Ginzburg, L.D. Landau, On the theory of superconductivity, Dokl. Akad. Nauk SSSR 469 (1954) 469.

[3] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (1979) 1085–1095.

[4] J.S. Langer, Models of pattern formation in first-order phase transitions, in: G. Grinsteirn, G. Mazenko (Eds.), Models of Pattern Formation in First-order Phase Transitions, World Scientific, Singapore, 1986, pp. 165–186.

[5] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Ann. Rev. Mater. Res. 32 (2002) 163–194.

[6] P.C. Hohenberg, B.I. Halperin, Theory of dynamic critical phenomena, Rev. Mod. Phys. 49 (1977) 435–479.

[7] J.D. Gunton, M.S. Miguel, P.S. Sahni, The dynamics of first-order phase transitions, in: C. Domb, J.L. Lebowitz (Eds.), Phase Transitions, vol. 8, Academic Press, New York, 1983, pp. 267–466.

[8] A.J. Bray, Theory of phase-ordering kinetics, Adv. Phys. 43 (1994) 357–459.

[9] M.T. Lusk, A phase-field paradigm for grain growth and recrystallization, Proc. R. Soc. Lond. A 455 (1999) 677–700.

[10] A. Artemev, Y. Jin, A. Khachaturyan, Three-dimensional phase field model of proper martensitic transformation, Acta Mater. 49 (2001) 1165–1177.

[11] L.Q. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to phase field equations, Comput. Phys. Commun. 108 (2-3) (1998) 147–158.

[12] C. Andersson, Time-stepping schemes for phase-field simulation of dendritic solidification, Tech. Rep. TRITA-NA-0216, ISSN 0348-2952, Stockholm University, <http://www.nada.kth.se/christe/publications.html>, 2002.

[13] E. Burman, A. Jacot, M. Picasso, Adaptive finite elements with high aspect ratio for the computation of coalescence using a phase-field model, J. Comput. Phys. 195 (1) (2004) 153–174.

[14] J. Rosam, P. Jimack, A. Mullis, A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification, J. Comput. Phys. 225 (2) (2007) 1271–1287.

[15] R. Kobayashi, J.A. Warren, W.C. Carter, A continuum model of grain boundaries, Phys. D: Nonlinear Phenom. 140 (1-2) (2000) 141–150.

[16] J.A. Warren, R. Kobayashi, A.E. Lobkovsky, W.C. Carter, Extending phase field models of solidification to polycrystalline materials, Acta Mater. 51 (20) (2003) 6035–6058.

[17] C.-J. Huang, D. Browne, S. McFadden, A phase-field simulation of austenite to ferrite transformation kinetics in low carbon steels, Acta Mater. 54 (2006) 11–21.

[18] T. Pusztai, G. Bortel, L. Gránásy, Phase field theory of polycrystalline solidification in three dimensions, EPL (Europhys. Lett.) 71 (1) (2005) 131–137.

[19] S.G. Kim, W.T. Kim, T. Suzuki, Phase-field model for binary alloys, Phys. Rev. E 60 (6) (1999) 7186–7197.

[20] T. Pusztai, G. Bortel, L. Gránásy, Phase field modeling of polycrystalline freezing, Mater. Sci. Eng. A 413–414 (2005) 412–417.

[21] T. Pusztai, G. Tegze, G.I. Tóth, L. Környei, G. Bansel, Z. Fan, L. Gránásy, Phase-field approach to polycrystalline solidification including heterogeneous and homogeneous nucleation, J. Phys.: Condens. Mat. 20 (2008) 404205.

[22] R. Kobayashi, J.A. Warren, Modeling the formation and dynamics of polycrystals in 3D, Phys. A: Statist. Mech. Appl. 356 (2005) 127–132.

[23] A.A. Wheeler, W.J. Boettinger, G.B. McFadden, Phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45 (10) (1992) 7424–7439.

[24] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations, Elsevier Science Publishing Co. Inc., New York, 1989.

[25] S. Hu, M. Baskes, M. Stan, J. Mitchell, Phase-field modeling of coring structure evolution in PuGa alloys, Acta Mater. 55 (11) (2007) 3641–3648.

[26] R. Kobayashi, Y. Giga, Equations with singular diffusivity, J. Statist. Phys. 95 (5/6) (1999) 1187–1220.

[27] E. Eich, Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints, SIAM J. Numer. Anal. 30 (5) (1993) 1467–1482.

[28] Y. Saad, M.H. Schulz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.

[29] <https://computation.llnl.gov/casc/sundials/main.html>.

[30] <https://computation.llnl.gov/casc/hypre/software.html>.

[31] D. Fan, S. Chen, L.-Q. Chen, P. Vorhees, Phase-field simulation of 2-D Ostwald ripening in the high volume fraction regime, Acta Mater. 50 (2002) 1895–1907.

[32] G.R. Edwards, R.E. Tate, E.A. Hakkila, Diffusion in stabilized delta plutonium, I. Gallium, J. Nucl. Mater. 25 (3) (1968) 304–309.

[33] W.Z. Wade, The self-diffusion of plutonium in a Pu/1 wt% Ga alloy, J. Nucl. Mater. 38 (3) (1971) 292–302.

[34] P. Atkins, J. de Paula, Atkins' Physical Chemistry, Oxford University Press, 2002, ISBN 0-19-879285-9.

[35] F. Rhines, Phase diagrams in metallurgy – their development and application, in: R. Mehl, M. Bever (Eds.), Metallurgy and Metallurgical Engineering Series, McGraw-Hill, New York, 1956.

[36] H.V. Atkinson, Theories of normal grain growth in pure single phase systems, Acta Metall. 36 (3) (1988) 469–491.

[37] R.D. Hornung, A.M. Wissink, S.R. Kohn, Managing complex data and geometry in parallel structured AMR applications, Eng. Comput. 22 (3–4) (2006) 181–195.

[38] <https://computation.llnl.gov/casc/SAMRAI>.

[39] P.H. Leo, J.S. Lowengrub, H.J. Jou, A diffuse interface model for microstructural evolution in elastically stressed solids, Acta Mater. 46 (6) (1998) 2113–2130.