

Solving the regularized, strongly anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method

Steven Wise^a, Junseok Kim^b, John Lowengrub^{a,*}

^a *Department of Mathematics, University of California, Irvine, CA 92697-3875, United States*

^b *Department of Mathematics, University of Dongguk, Seoul 100-715, Republic of Korea*

Received 25 August 2006; received in revised form 26 March 2007; accepted 13 April 2007

Available online 6 May 2007

Abstract

We present efficient, second-order accurate and adaptive finite-difference methods to solve the regularized, strongly anisotropic Cahn–Hilliard equation in 2D and 3D. When the surface energy anisotropy is sufficiently strong, there are missing orientations in the equilibrium level curves of the diffuse interface solutions, corresponding to those missing from the sharp interface Wulff shape, and the anisotropic Cahn–Hilliard equation becomes ill-posed. To regularize the equation, a higher-order derivative term is added to the energy. This leads to a sixth-order, nonlinear parabolic equation for the order parameter. An implicit time discretization is used to remove the high-order time step stability constraints. Dynamic block-structured Cartesian mesh refinement is used to highly resolve narrow interfacial layers. A multilevel, nonlinear multigrid method is used to solve the nonlinear equations at the implicit time level. One of the keys to the success of the method is the treatment of the anisotropic term. This term is discretized in conservation form in space and is discretized fully implicitly in time. Numerical simulations are presented that confirm the accuracy, efficiency and stability of the scheme. We study the dynamics of interfaces under strong anisotropy and compare near-equilibrium diffuse interface solutions to the sharp interface Wulff shapes in 2D and 3D. We also simulate large-scale coarsening of a corrugated surface (in 3D) evolving by anisotropic surface diffusion. We show the emergence of long-range order during coarsening and an interesting mechanism of ordered coarsening.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Cahn–Hilliard equation; Strong anisotropy; Regularization; Adaptive mesh refinement; Cartesian grid methods; Nonlinear multigrid methods

1. Introduction

The Cahn–Hilliard equation is an integral part of many physical models describing the evolution of different material phases via an order parameter (or multiple order parameters). The equation was initially derived

* Corresponding author. Tel.: +1 949 824 2655; fax: +1 949 824 7993.

E-mail addresses: swise@math.uci.edu (S. Wise), cfdkim@dongguk.edu (J. Kim), lowengrb@math.uci.edu (J. Lowengrub).

as a model for spinodal decomposition in solid materials [1,2] and has since been extended to many other physical systems. Indeed, an attractive feature of the Cahn–Hilliard framework is that the dynamical equations are derived from an energy functional valid for the whole material and consequently different energy densities may be added easily depending upon the required physics.

In Cahn–Hilliard-based models, sharp interfaces are replaced by narrow transition layers (diffuse interfaces) that result from a competition between the different terms in the energy density. The Cahn–Hilliard equation has been used to develop diffuse interface models for elastic phase transformations in binary alloys, *e.g.*, [3–6]; for surface diffusion of adatoms in stressed epitaxial thin films, *e.g.*, [7–9]; multiphase fluid flow, *e.g.*, [10–12]; stress and electromagnetic driven void migration, *e.g.*, [13–15]; solid tumor growth *e.g.*, [16]; and vesicle dynamics, *e.g.*, [17], just to name a few. See the review articles [18,19] for additional references.

In crystalline solids, the energy of interfaces in certain orientations may be so large that they do not commonly appear in the microstructure. In the sharp interface case, when the surface energy density is a smooth function of the interface normal and the anisotropy is sufficiently strong, it is known that there are missing orientations in the Wulff shape, *i.e.*, the shape that minimizes the total surface energy for a given volume. In 2D missing orientations appear when there is a sign change in the surface “stiffness” $\gamma(\theta) + \gamma''(\theta)$, where γ is the energy density and θ is the interface orientation angle. (The criterion is much more complicated in the 3D case [20].) Correspondingly, the sharp interface geometric evolution equations, *i.e.*, motion by surface diffusion and motion by mean curvature, become ill-posed when there are missing orientations because the surface energy is non-convex [21]. The ill-posedness and loss of smoothness have apparently been avoided by adding a higher-order derivative regularization, *i.e.*, the curvature squared (bending energy)¹ to the surface energy [22–24,21]. In the important context of surface diffusion, this regularization results in sixth-order systems for the interface evolution. Recently, Spencer [25] performed asymptotics for the curvature-regularized sharp interface problem in 2D and demonstrated convergence to the Wulff shape as the regularization parameter tends to zero. Numerical methods for curvature-regularized, strongly anisotropic motion in several 2D and 3D physical problems have been developed using sharp interface (parameterized surface) [26–29], level-set [21], and graph [30] frameworks.

The anisotropic Cahn–Hilliard equation, which may be used as a diffuse interface model of motion by surface diffusion [9], becomes ill-posed when the gradient energy is non-convex. In 2D, this is equivalent with the surface “stiffness” having a sign change (see the Appendix). In the more general diffuse interface context, Taylor and Cahn [31] have suggested that one convexify the gradient energy to keep the equations well-posed. But while convexifying does remove the ill-posedness, there may still be sharp corners appearing on the evolving level sets of the order parameter, what Taylor and Cahn [31] liken to shocks, as well as corners in the equilibrium diffuse interfaces. Naturally, discontinuities of the derivative of the order parameter pose significant computational challenges. Eggleston et al. [32] implemented a convexification scheme for the strongly anisotropic Cahn–Hilliard equation in 2D and used one-sided differencing to deal with the shocks in the evolving order parameter. Their work produced equilibrium solutions whose level curves were remarkably close to the sharp interface Wulff shapes. However, the method in [32] is not mass-conserving and is discretized in time using an explicit method resulting in severe time step restrictions. It has also been pointed out [27] that convexification, whether in the sharp or diffuse interface context, prevents a description of the nucleation of facets in the model.

As in the sharp interface case, it is conjectured that the addition of a bending-energy-type term to the anisotropic Cahn–Hilliard equation will regularize against the formation of sharp corners in the level sets of the order parameter. Du et al. [33–35] developed numerical methods and theory for a diffuse interface analogue of the bending energy in studies of elastic vesicles. Very recently, Rätz et al. [9] proposed using this energy in the development of a regularized diffuse interface model for anisotropic surface diffusion in thin, elastic films. The resulting Cahn–Hilliard-type equation is sixth-order in space. However, they did not propose a numerical scheme or exhibit simulation results for the regularized problem. Using a different high-order regularization (the square of the Laplacian of the order parameter), Wise et al. [8] developed a sixth-order diffuse interface

¹ We note that it is not known theoretically whether this type of regularization is sufficient to always prevent singularity formation. Numerically, it is found that this regularization rounds corners and edges.

model for strongly anisotropic surface diffusion and simulated the development of spatially patterned quantum dots in an epitaxial thin film. Wheeler [36] very recently performed an asymptotic analysis for this regularization in the context of a phase-field model of solidification and determined the profile of rounded corners in 2D and the limiting form of the excess corner energy.

In this paper, we focus on the Laplacian-squared regularization and develop efficient, second-order accurate and adaptive finite-difference methods to solve strongly anisotropic Cahn–Hilliard equation in 2D and 3D. Here, we flesh out the details of the algorithm used in [8] and also present substantial improvements. One of the keys to the success of the method is the treatment of the anisotropic term. This term is discretized in conservation form in space and is fully implicit in time. Non-conservative treatments of the anisotropic term lead to numerical instability while explicit time discretizations of this term lead to strong time step constraints. The implicit discretization of the anisotropic term is found to remove the stiffness arising from anisotropy and allows us to use similar time steps as in the isotropic case.

Other key components of the algorithm are the use of dynamic, block-structured Cartesian mesh refinement (e.g., [37,38]) and the use of an adaptive nonlinear multigrid method to solve the equations at the implicit time level. Neither of these algorithms were used in [8]. Locally refined block-structured Cartesian meshes strike a balance between grid structure and efficiency and are very natural to use together with multilevel multigrid methods. We note that other multilevel multigrid algorithms have been developed as part of the CHOMBO [39,40] and the BEARCLAW [41] software packages. Here, we follow the Full Approximation Scheme (FAS) multigrid methodology presented in Kim et al. [12] for the isotropic Cahn–Hilliard equation, and introduce modifications so as to fit within the framework of the block-structured multilevel adaptive technique (MLAT) developed by Brandt [42]. See Trottenberg et al. [43, Chapter 9].

There have been several previously developed adaptive mesh refinement schemes for the Cahn–Hilliard equation in the finite element framework. See [9] (mentioned above) for the cases of weak surface energy anisotropy and strong kinetic anisotropy and the works of Garcke et al. [4] and Kay and Welford [44] for isotropic surface energies. On the other hand, very few (if any) finite-difference based adaptive methods for the Cahn–Hilliard exist. Furthermore, to our knowledge, [8] briefly describes the only numerical method that has been reported for the regularized, strongly anisotropic Cahn–Hilliard equation.

Our algorithm extends naturally to the case of the diffuse interface version of the bending energy regularization proposed in [9]. Comparisons of the evolution and steady-states for strong surface energy anisotropies using the different regularizations will be presented in a forthcoming work. Here, we demonstrate numerically the second-order convergence of the adaptive scheme and show some relevant computational examples involving strong anisotropy in 2D and 3D. Using simulations of bulk spinodal decomposition followed by domain coarsening, we demonstrate the enormous savings possible when dynamic mesh refinement is performed solely around interfacial regions. In 2D and 3D, we investigate the evolution of closed shapes toward their diffuse interface equilibria and compare against the sharp interface Wulff shapes. We simulate the decomposition of level curves in 2D and level surfaces in 3D into near-facets and subsequent coarsening, where motion is by anisotropic surface diffusion. In particular, we study large-scale coarsening of a pyramidal surface (in 3D) showing the emergence of long-range order during coarsening and an interesting mechanism of ordered coarsening that was observed using a different model of anisotropic surface diffusion [45].

The paper is organized as follows. In Section 2 we derive the regularized, strongly anisotropic Cahn–Hilliard equation. In Section 3 the fully discrete, nonlinear FAS multigrid scheme for the regularized equation is given and in Section 4 we discuss the corresponding block-structured, adaptive MLAT implementation. In Section 5, numerical results are presented. We give some concluding remarks in Section 6. In the Appendix, we discuss the link between the ill-posedness of the sharp and diffuse interface problems.

2. Diffuse interface anisotropy

2.1. Variational derivative and natural boundary conditions

Consider a bounded domain $\Omega \subset \mathbb{R}^d$. Let $c : \Omega \rightarrow \mathbb{R}$ be an order parameter; denote the gradient of c by $\mathbf{p} := \nabla c$ and the corresponding normal by $\mathbf{n} := -\mathbf{p}/|\mathbf{p}|$, provided $|\mathbf{p}| \neq 0$. Following the approach initiated by Kobayashi [46], we consider a free energy functional of the form

$$\mathcal{E}[c] := \int_{\Omega} \left(F(c) + \frac{\epsilon^2}{2} |\gamma(\mathbf{n})\mathbf{p}|^2 \right) d\mathbf{x}, \tag{1}$$

where $F(c) := c^2(1 - c)^2/4$, for example; ϵ is a constant; and γ is the interfacial energy anisotropy function, which can be taken to be that from the sharp interface model. For definiteness, we assume the following form of γ for the derivation of equations:

$$\gamma(\mathbf{n}) := 1 + a\Gamma(\mathbf{n}), \tag{2}$$

where a is the anisotropy parameter and Γ is a smooth function of the normal vector. For cases where γ is non-smooth see [31,47]. We assume that a is constrained so that $\gamma > 0$. If $a = 0$ the energy defined in Eq. (1) is isotropic. In two dimensions γ is typically given as a function of θ , the angle of orientation of the interface relative to the x_1 axis, which satisfies the relation $\tan(\theta) = n_2/n_1 = p_2/p_1$.

We define the anisotropy gradient to be

$$\mathbf{m} := \gamma^2 \mathbf{p} - a\gamma|\mathbf{p}|\mathcal{P}\nabla_{\mathbf{n}}\Gamma, \tag{3}$$

where $\nabla_{\mathbf{n}}\Gamma = \sum_{i=1}^d (\partial\Gamma/\partial n_i)\mathbf{e}_i$, and \mathcal{P} is the tangential projection matrix

$$\mathcal{P} := \mathbf{I} - \mathbf{n} \otimes \mathbf{n}. \tag{4}$$

It follows that the variational derivative of \mathcal{E} with respect to c is

$$\frac{\delta\mathcal{E}}{\delta c} = F'(c) - \epsilon^2 \nabla \cdot \mathbf{m} \tag{5}$$

and the natural boundary condition is

$$\mathbf{m} \cdot \boldsymbol{\zeta} = 0 \quad \text{on } \partial\Omega, \tag{6}$$

where $\boldsymbol{\zeta}$ is the outward-pointing unit normal on the boundary $\partial\Omega$. Note that the normal vector is not defined on the set $\mathcal{N} := \{\mathbf{x} \in \Omega \mid |\mathbf{p}(\mathbf{x})| = 0\}$, but the anisotropy gradient may be continuously defined by setting $\mathbf{m}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} \in \mathcal{N}$.

We say that the derivatives of Γ are *component-wise homogeneous of (at least) degree 1* if

$$\frac{\partial\Gamma}{\partial n_i} = n_i P_i(\mathbf{n}), \quad i = 1, \dots, d, \tag{7}$$

where n_i is the i th component of the normal vector, P_i is a smooth function of the components of the normal that is bounded on compact sets (e.g., a polynomial), and d is the spatial dimension. For instance, this condition is satisfied for the fourfold anisotropy function, where

$$\Gamma(\mathbf{n}) = 4 \sum_{i=1}^d n_i^4 - 3 \quad \text{and} \quad P_i(\mathbf{n}) = 16n_i^2, \tag{8}$$

as well as for the eightfold anisotropy function. If the derivatives of Γ satisfy (7), then the variational derivative may be written as

$$\frac{\delta\mathcal{E}}{\delta c} = F'(c) - \epsilon^2 \nabla \cdot (\mathbf{p} + \mathcal{A}(\mathbf{n})\mathbf{p}), \tag{9}$$

where $\mathcal{A}(\mathbf{n})$ is a diagonal matrix whose entries are

$$\mathcal{A}_{i,i}(\mathbf{n}) = a\gamma(\mathbf{n}) \left[P_i(\mathbf{n}) - \sum_{k=1}^d n_k^2 P_k(\mathbf{n}) \right] + \gamma^2(\mathbf{n}) - 1. \tag{10}$$

To see this, let $\delta_{i,j}$ denote the Krönercker delta. Since the derivatives of Γ are homogeneous of degree 1, the i th component of the anisotropic gradient, m_i , simplifies to

$$\begin{aligned}
 m_i &= -a\gamma|\mathbf{p}| \left[\sum_{j=1}^d n_j P_j (\delta_{ij} - n_i n_j) \right] + \gamma^2 p_i = -a\gamma|\mathbf{p}| n_i \left[P_i - \sum_{j=1}^d n_j^2 P_j \right] + \gamma^2 p_i \\
 &= p_i + \left\{ a\gamma \left[P_i - \sum_{j=1}^d n_j^2 P_j \right] + \gamma^2 - 1 \right\} p_i.
 \end{aligned}
 \tag{11}$$

We say that Ω is a *canonical rectangular domain* provided it is a d -dimensional rectangular box with its face normals in the canonical directions \mathbf{e}_i . The form of the anisotropy gradient in (11) allows for the use of simpler boundary conditions than (6) in the case that Ω is a domain of this special type. Indeed, consider the value of $\mathbf{m} \cdot \boldsymbol{\zeta}$ on the two faces whose normals are $\boldsymbol{\zeta} = -\mathbf{e}_k$ and $\boldsymbol{\zeta} = \mathbf{e}_k$, respectively:

$$\mathbf{m} \cdot \boldsymbol{\zeta} = \pm m_k = \pm \left\{ a\gamma \left[P_k - \sum_{j=1}^d n_j^2 P_j \right] + \gamma^2 \right\} p_k.
 \tag{12}$$

If $\mathbf{p} \cdot \boldsymbol{\zeta} = \pm p_k = 0$ is assumed on these faces, then $\mathbf{m} \cdot \boldsymbol{\zeta} = 0$. In other words $\mathbf{p} \cdot \boldsymbol{\zeta} = 0$ implies $\mathbf{m} \cdot \boldsymbol{\zeta} = 0$ for this special case. The converse is not generally true, and, thus, $\mathbf{p} \cdot \boldsymbol{\zeta} = 0$ on $\partial\Omega$ is not the natural boundary condition, as the condition $\mathbf{m} \cdot \boldsymbol{\zeta} = 0$ is still more general. But, clearly, the former is simpler to implement (numerically) than the latter and still yields the correct variational derivative for canonical rectangular domains Ω .

2.2. Anisotropic Cahn–Hilliard equation

We assume that the order parameter c is conserved, and we define the mass flux of c to be

$$\mathcal{J} := -M(c, \mathbf{p}) \nabla \frac{\delta \mathcal{E}}{\delta c},
 \tag{13}$$

where $M \geq 0$ is a diffusional mobility. The mass conservation equation $\partial c / \partial t = -\nabla \cdot \mathcal{J}$ yields the anisotropic Cahn–Hilliard equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (M(c, \mathbf{p}) \nabla \mu),
 \tag{14}$$

$$\mu := \frac{\delta \mathcal{E}}{\delta c} = F'(c) - \epsilon^2 \nabla \cdot \mathbf{m}.
 \tag{15}$$

We use the natural and no-flux boundary conditions for the evolution of a mass-conserving system:

$$\mathbf{m} \cdot \boldsymbol{\zeta} = \frac{\partial \mu}{\partial \boldsymbol{\zeta}} = 0 \quad \text{on } \partial\Omega,
 \tag{16}$$

where $\boldsymbol{\zeta}$ is the outward-pointing unit normal on $\partial\Omega$ and $\partial f / \partial \boldsymbol{\zeta} := \boldsymbol{\zeta} \cdot \nabla f$. If the derivatives of F are homogeneous as in (7), then the anisotropic Cahn–Hilliard equation is

$$\frac{\partial c}{\partial t} = \nabla \cdot (M(c, \mathbf{p}) \nabla \mu),
 \tag{17}$$

$$\mu = F'(c) - \epsilon^2 \Delta c - \epsilon^2 \nabla \cdot (\mathcal{A}(\mathbf{n}) \nabla).
 \tag{18}$$

If Ω is a canonical rectangular domain then we use the simpler boundary conditions

$$\frac{\partial c}{\partial \boldsymbol{\zeta}} = \frac{\partial \mu}{\partial \boldsymbol{\zeta}} = 0 \quad \text{on } \partial\Omega.
 \tag{19}$$

If c is a sufficiently regular solution to either (14)–(16) or (17)–(19) then the free energy is non-increasing in time. Thus the energy (1) is a Lyapunov functional for either system of equations.

For $M = 1$ above we have bulk diffusion; using $M = c(1 - c)$ or $M = c^2(1 - c)^2$ we can model motion by surface diffusion in the isotropic case ($a = 0$) (e.g., see [48,9]). For an appropriately chosen mobility $M(c, \mathbf{p})$ and weak anisotropy, it may be shown that the Cahn–Hilliard equation above converges asymptotically to motion by (weakly) anisotropic surface diffusion using the techniques in [9].

The difficulty with either system—(14)–(16) or (17)–(19)—is that they may be ill-posed for strong enough anisotropy (*i.e.*, large enough a). Writing $G(\mathbf{p}) = \gamma(\mathbf{n})|\mathbf{p}|$, if $G^2(\mathbf{p})$ is not convex then the term $\nabla \cdot \mathbf{m}$ may be backwards diffusive for some initial data. See the Appendix for further discussion. In particular, in 2D when $\gamma = \gamma(\theta)$ and with some reasonable assumptions on γ , $\nabla \cdot \mathbf{m}$ is backwards diffusive precisely when $\gamma(\theta) + \gamma''(\theta) < 0$. Taylor and Cahn [31] have suggested that one convexify $G^2(\mathbf{p})$ to keep the equations well-posed. Eggleston et al. [32] did essentially that by modifying γ so that $\gamma(\theta) + \gamma''(\theta) \geq 0$ for all θ . But while convexifying G^2 does remove the ill-posedness, there may still be sharp corners appearing on the evolving level sets of the order parameter, what Taylor and Cahn [31] liken to shocks, as well as corners on the corresponding Wulff shape. Naturally, discontinuities of the derivative of the order parameter pose significant computational challenges. But more importantly, as Hausser and Voigt [28] suggest, convexifying G^2 as done in [32] results in a model that cannot properly describe the nucleation of facets. In this paper we obtain well-posedness through regularization by the addition of high-order derivative terms in the energy. This has the effect of smoothing the corners that would otherwise appear in the evolving and equilibrium level sets, and the model retains the ability to nucleate “facets”.

2.3. Regularized anisotropic free energy

Consider a regularized free energy functional of the form

$$\mathcal{E}[c] := \int_{\Omega} \left(F(c) + \frac{\epsilon^2}{2} |\gamma(\mathbf{n})\mathbf{p}|^2 + \frac{\delta^2}{2} \left(\Delta c - \frac{\alpha}{\epsilon^2} F'(c) \right)^2 \right) \mathbf{d}\mathbf{x}, \tag{20}$$

where δ is the corner energy regularization parameter; and α is a non-negative constant. When $\alpha = 1$ the fourth-order term is the diffuse Willmore regularization [34,9]. When $\alpha = 0$ this gives the simpler regularization considered in [8,36]. Although we will focus in this paper on the $\alpha = 0$ case, we present the equations for arbitrary α .

Henceforth, we assume that the derivatives of Γ satisfy the homogeneity condition (7), that Ω is a canonical rectangular domain, and $\mathbf{p} \cdot \boldsymbol{\zeta} = \partial c / \partial \boldsymbol{\zeta} = 0$ on the boundary faces of Ω . Accordingly, the variational derivative may be calculated as

$$\frac{\delta \mathcal{E}}{\delta c} = F'(c) - \epsilon^2 \Delta c - \epsilon^2 \nabla \cdot (\mathcal{A}(\mathbf{n})\mathbf{p}) + \frac{\delta^2}{\epsilon^2} \left(\Delta v - v \frac{\alpha}{\epsilon^2} F''(c) \right), \tag{21}$$

$$v := \epsilon^2 \Delta c - \alpha F'(c), \tag{22}$$

where we have assumed that the following higher-order natural boundary condition holds:

$$\frac{\partial v}{\partial \boldsymbol{\zeta}} = 0 \quad \text{on } \partial\Omega. \tag{23}$$

Note that addition of the higher-order regularization in the energy does not change the calculations that we performed in the previous sections as they are retained setting $\delta = 0$. Defining the mass flux as previously (see Eq. (13)), the regularized, anisotropic Cahn–Hilliard equation becomes

$$\frac{\partial c}{\partial t} = \nabla \cdot (M(c, \mathbf{p})\nabla \mu), \tag{24}$$

$$\mu := \frac{\delta \mathcal{E}}{\delta c} = (1 - \alpha)F'(c) - v - \epsilon^2 \nabla \cdot (\mathcal{A}(\mathbf{n})\mathbf{p}) + \frac{\delta^2}{\epsilon^2} \left(\Delta v - v \frac{\alpha}{\epsilon^2} F''(c) \right), \tag{25}$$

$$v := \epsilon^2 \Delta c - \alpha F'(c). \tag{26}$$

The complete set boundary conditions are expressed as

$$\frac{\partial c}{\partial \boldsymbol{\zeta}} = \frac{\partial \mu}{\partial \boldsymbol{\zeta}} = \frac{\partial v}{\partial \boldsymbol{\zeta}} = 0 \quad \text{on } \partial\Omega. \tag{27}$$

Observe that this is a highly nonlinear, sixth-order equation for c . A calculation shows that the regularized energy is non-increasing in time, *i.e.*, $d\mathcal{E}/dt \leq 0$ along solution paths. We point out that, to the authors’

knowledge, there are currently no rigorous or asymptotic convergence results for the Cahn–Hilliard equation above for either of the cases $\alpha = 0$ or $\alpha = 1$. Hereafter, we focus on the case $\alpha = 0$, although our methods extend straightforwardly to the $\alpha \neq 0$ case, and we assume that $M = M(c)$.

3. Discretization

Before we present the finite-difference discretization of the system (24)–(26) on the adaptive, multilevel mesh, we first consider the discretization on a uniform grid. This enables the description of the nonlinear operator, the source term and the smoothing operations in a simple way. Since the block-structured Cartesian adaptive mesh that we utilize is essentially the union of uniform grids, it is natural to start from this perspective.

We consider the 2D case for simplicity, the extension to 3D is straightforward. In particular, suppose that the equations hold on the rectangular domain $\Omega = (0, L_1) \times (0, L_2)$, such that Ω may be overlain with a regular, uniform $N_1 \times N_2$ grid with grid spacing h . Thus $L_d = hN_d$ for $d = 1, 2$. We use the cell-centered discretizations of the dependent variables defining them at the center of each cell in the grid. The cell centers have the coordinates $(x_i, y_j) = ((i - 1/2)h, (j - 1/2)h)$, for $i = 1, \dots, N_1$ and $j = 1, \dots, N_2$, and we make the identification $f_{i,j} := f(x_i, y_j)$, where f is one of the dependent variables. The east–west cell edges have the coordinates $(x_{i+1/2}, y_j) = (ih, (j - 1/2)h)$, for $i = 0, \dots, N_1$ and $j = 1, \dots, N_2$; the north–south cell edges have the coordinates $(x_i, y_{j+1/2}) = ((i - 1/2)h, jh)$, for $i = 1, \dots, N_1$ and $j = 0, \dots, N_2$. The appropriate east–west cell edge identification is $f_{i+1/2,j} := f(x_{i+1/2}, y_j)$, and similarly for the north–south cell edge.

The edge-difference operators, which act on cell-centered functions, are defined on the cell edges via

$$D_{1,ew}f_{i+1/2,j} = \frac{f_{i+1,j} - f_{i,j}}{h}, \quad (28)$$

$$D_{2,ew}f_{i+1/2,j} = \frac{f_{i+1,j+1} - f_{i+1,j-1} + f_{i,j+1} - f_{i,j-1}}{4h}, \quad (29)$$

$$D_{1,ns}f_{i,j+1/2} = \frac{f_{i+1,j+1} - f_{i-1,j-1} + f_{i+1,j} - f_{i-1,j}}{4h}, \quad (30)$$

$$D_{2,ns}f_{i,j+1/2} = \frac{f_{i,j+1} - f_{i,j}}{h}. \quad (31)$$

The subscripts *ew* and *ns* indicate whether the functions are defined on the east–west or north–south edges. The center-difference operators, which act on edge-centered functions, are defined on the cell center points via

$$d_1f_{i,j} = \frac{f_{i+1/2,j} - f_{i-1/2,j}}{h}, \quad d_2f_{i,j} = \frac{f_{i,j+1/2} - f_{i,j-1/2}}{h}. \quad (32)$$

The edge-average operators, which act on cell-centered functions, are defined on the cell edges via

$$A_{1,ew}f_{i+1/2,j} = \frac{f_{i+1,j} + f_{i,j}}{2}, \quad A_{2,ns}f_{i,j+1/2} = \frac{f_{i,j+1} + f_{i,j}}{2}, \quad (33)$$

and the center-average operators, which act on edge-centered functions, are defined on the cell center points

$$a_1f_{i,j} = \frac{f_{i+1/2,j} + f_{i-1/2,j}}{2}, \quad a_2f_{i,j} = \frac{f_{i,j+1/2} + f_{i,j-1/2}}{2}. \quad (34)$$

The edge operators $D_{k,ew}f_{i+1/2,j}$, $k = 1, 2$, and $A_{1,ew}f_{i+1/2,j}$ are defined for $0 \leq i \leq N_1$ and $1 \leq j \leq N_2$; and $D_{k,ns}f_{i,j+1/2}$, $k = 1, 2$, and $A_{2,ns}f_{i,j+1/2}$ are defined for $1 \leq i \leq N_1$ and $0 \leq j \leq N_2$. The center operators $d_1f_{i,j}$, $a_1f_{i,j}$, $d_2f_{i,j}$ and $a_2f_{i,j}$ are defined for $1 \leq i \leq N_1$ and $1 \leq j \leq N_2$.

By $D_{k,ew}f$, $k = 1, 2$ and $A_{1,ew}f$, we denote the $(N_1 + 1) \times N_2$ arrays whose elements are given by (28), (29), and (33a), respectively; and $D_{k,ns}f$, $k = 1, 2$ and $A_{2,ns}f$ are the $N_1 \times (N_2 + 1)$ arrays whose elements are given by (30), (31), and (33b), respectively. By d_1f , a_1f , a_2f and a_2f we denote the $N_1 \times N_2$ arrays whose elements are given in (32a), (34a), (32b) and (34b), respectively.

Using the difference formulae above we may define the discrete Laplacian operators as

$$A_d c = d_1(D_{1,ew}c) + d_2(D_{2,ns}c), \tag{35}$$

and

$$\nabla_d \cdot (M(c)\nabla_d \mu) = d_1(M(A_{1,ew}c)D_{1,ew}\mu) + d_2(M(A_{2,ns}c)D_{2,ns}\mu). \tag{36}$$

The normal vector is required at the cell edges in this scheme, and its components are defined as

$$(n_{k,ew})_{i+1/2,j} = \frac{D_{k,ew}c_{i+1/2,j}}{\sqrt{(D_{1,ew}c_{i+1/2,j})^2 + (D_{2,ew}c_{i+1/2,j})^2 + \sigma}}, \quad k = 1, 2, \tag{37}$$

$$(n_{k,ns})_{i,j+1/2} = \frac{D_{k,ns}c_{i,j+1/2}}{\sqrt{(D_{1,ns}c_{i,j+1/2})^2 + (D_{2,ns}c_{i,j+1/2})^2 + \sigma}}, \quad k = 1, 2, \tag{38}$$

where $0 < \sigma \ll 1$. With the normal defined on the cell edges we may define

$$\nabla_d \cdot (\mathcal{A}(\mathbf{n})\nabla_d c) = d_1(\mathcal{A}_{1,1}(\mathbf{n}_{ew})D_{1,ew}c) + d_2(\mathcal{A}_{2,2}(\mathbf{n}_{ns})D_{2,ns}c), \tag{39}$$

where \mathbf{n}_{ew} is the $2 \times (N_1 + 1) \times N_2$ array whose elements are given by (37), and \mathbf{n}_{ns} , the $2 \times N_1 \times (N_2 + 1)$ array whose elements are given by (38).

A discretization of the continuous energy (20), with $\alpha = 0$, is given by

$$\mathcal{E}_d = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \left(F(c_{i,j}) + \frac{\epsilon^2}{2} (a_1(g_{1,ew})_{i,j} + a_2(g_{2,ns})_{i,j}) + \frac{\delta^2}{2} (A_d c_{i,j})^2 \right), \tag{40}$$

where $g_{1,ew}$ and $g_{2,ns}$ are edge-centered gradient functions

$$(g_{1,ew})_{i+1/2,j} = (\gamma((\mathbf{n}_{ew})_{i+1/2,j})D_{1,ew}c_{i+1/2,j})^2, \tag{41}$$

$$(g_{2,ns})_{i,j+1/2} = (\gamma((\mathbf{n}_{ns})_{i,j+1/2})D_{2,ns}c_{i,j+1/2})^2. \tag{42}$$

We use a Crank–Nicholson-type discretization in time to approximate Eqs. (24)–(26) as

$$\frac{c_{i,j}^{n+1} - c_{i,j}^n}{s} = \frac{1}{2} [\nabla_d \cdot (M(c_{i,j}^{n+1})\nabla_d \mu_{i,j}^{n+1}) + \nabla_d \cdot (M(c_{i,j}^n)\nabla_d \mu_{i,j}^n)], \tag{43}$$

$$\mu_{i,j}^{n+1} = f(c_{i,j}^{n+1}) - v_{i,j}^{n+1} - \epsilon^2 \nabla_d \cdot (\mathcal{A}(\mathbf{n}_{i,j}^{n+1})\nabla_d c_{i,j}^{n+1}) + \frac{\delta^2}{\epsilon^2} A_d v_{i,j}^{n+1}, \tag{44}$$

$$v_{i,j}^{n+1} = \epsilon^2 A_d c_{i,j}^{n+1}, \tag{45}$$

defined for $1 \leq i \leq N_1$ and $1 \leq j \leq N_2$, where $f(c) = F'(c)$, and s is the time step size. Note that the stencil at (i,j) is compact, sampling data only at (i,j) and its eight nearest neighbors. The stencil remains compact in the extension of the method to 3D, where the stencil size is 27.

One of the key features in our method is that the discretization maintains the conservative form of the equations, as can be seen above; and indeed one may show discrete mass conservation using summation by parts. This is in contrast to the scheme developed by Eggleston et al. [32]. We expect that the scheme will have a solvability restriction on the order of $s \leq Ch^2$, as demonstrated for similar schemes in [49,50]. However in practice, we find that C can be quite large, so that it is possible to take $s \sim h$ in many applications. Discrete energy decrease is difficult to show rigorously, owing to the complicated structure of Eqs. (43)–(45). In our testing we have observed that the discrete energy is non-increasing for time steps for which the nonlinear multigrid iteration converges. The theoretical stability conditions for the method are still being sought. In particular, it is not known whether the solvability condition or the stability condition is more restrictive for the proposed method. We plan to resolve these questions in future work. In particular, we plan to develop a convex splitting of the energy to achieve an unconditionally stable and solvable numerical method (e.g., see [49]). The work in this paper is viewed as a first step in this direction.

To solve the nonlinear system (43)–(45) on the uniform grid we use the Full Approximation Scheme (FAS) Multigrid method. The FAS method has the useful property that the difference between its application on the adaptive mesh (MLAT) and its application on a uniform fixed grid is slight. The first step in the FAS method

is to split the nonlinear equations into operator (**N**) and source (**S**) terms. First, for notational simplicity, we define

$$\boldsymbol{\phi}_{i,j}^* = (c_{i,j}^*, \mu_{i,j}^*, v_{i,j}^*), \quad (46)$$

where * is the place-holder for any superscripted indices. To denote the array of cell-centered values we drop the subscripts. Thus, $\boldsymbol{\phi}^n = (c^n, \mu^n, v^n)$ is the $3 \times N_1 \times N_2$ solution array at time step n . The components of the nonlinear operator are defined as

$$N_{i,j}^{(1)}(\boldsymbol{\phi}^{n+1}) = c_{i,j}^{n+1} - \frac{S}{2} \nabla_d \cdot (M(c_{i,j}^{n+1}) \nabla_d \mu_{i,j}^{n+1}), \quad (47)$$

$$N_{i,j}^{(2)}(\boldsymbol{\phi}^{n+1}) = \mu_{i,j}^{n+1} - f(c_{i,j}^{n+1}) + v_{i,j}^{n+1} - \frac{\delta^2}{\epsilon^2} \Delta_d v_{i,j}^{n+1}, \quad (48)$$

$$N_{i,j}^{(3)}(\boldsymbol{\phi}^{n+1}) = v_{i,j}^{n+1} - \epsilon^2 \Delta_d c_{i,j}^{n+1}, \quad (49)$$

and the components of the source term are defined as

$$S_{i,j}^{(1)}(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n) = c_{i,j}^{n+1} + \frac{S}{2} \nabla_d \cdot (M(c_{i,j}^n) \nabla_d \mu_{i,j}^n), \quad (50)$$

$$S_{i,j}^{(2)}(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n) = -\epsilon^2 \nabla_d \cdot (\mathcal{A}(\mathbf{n}_{i,j}^{n+1}) \nabla_d c_{i,j}^{n+1}), \quad (51)$$

$$S_{i,j}^{(3)}(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n) = 0. \quad (52)$$

Writing, $\mathbf{N} = (N^{(1)}, N^{(2)}, N^{(3)})$ and $\mathbf{S} = (S^{(1)}, S^{(2)}, S^{(3)})$, we must solve the $3 \times N_1 \times N_2$ nonlinear system

$$\mathbf{N}(\boldsymbol{\phi}^{n+1}) = \mathbf{S}(\boldsymbol{\phi}^{n+1}, \boldsymbol{\phi}^n) \quad (53)$$

for each time iteration n . (Here we have neglected the boundary equations for simplicity.) Ideally, the source term would contain only data at the n th time step. One would then “invert” **N** to solve for $\boldsymbol{\phi}^{n+1}$. It is equally valid to include data at the $n + 1$ st time step in the source term, but then a fixed point method is required to solve (53).

This grouping into source and operator terms reflects a natural splitting of $\Delta c + \nabla \cdot (\mathcal{A}(\mathbf{n}) \nabla c)$ into convex (Δc) and non-convex ($\nabla \cdot (\mathcal{A}(\mathbf{n}) \nabla c)$) pieces. It may be possible to split the terms into purely convex and concave pieces, and such a splitting would be expected to yield a more robust algorithm (*e.g.*, see [49] for an algorithm where this is done with $f(c)$). We plan to explore this in a future work. Another option is to lag the non-convex piece in the time iteration, *i.e.*, to replace Eqs. (44) and (51) with

$$\mu_{i,j}^{n+1} = f(c_{i,j}^{n+1}) - v_{i,j}^{n+1} - \epsilon^2 \nabla_d \cdot (\mathcal{A}(\mathbf{n}_{i,j}^n) \nabla_d c_{i,j}^n) + \frac{\delta^2}{\epsilon^2} \Delta_d v_{i,j}^{n+1}, \quad (54)$$

$$S_{i,j}^{(2)}(\boldsymbol{\phi}^n) = -\epsilon^2 \nabla_d \cdot (\mathcal{A}(\mathbf{n}_{i,j}^n) \nabla_d c_{i,j}^n). \quad (55)$$

This results in a relatively stable algorithm; however, it spoils the second-order accuracy of the method.

Our FAS implementation is composed of two pieces, (i) the recursive V-cycle iteration, and (ii) the nonlinear smoother [12,43]. The V-cycle is the outermost iteration, and we consider it first. For practical purposes, one may think of a single V-cycle iteration as a way to obtain an approximate inverse of **N**. Since there are data at the $n + 1$ st time step in the source term, we lag the term in the V-cycle iteration, and we look for the iterated fixed point solution of

$$\mathbf{N}(\boldsymbol{\phi}^{n+1,m+1}) = \mathbf{S}(\boldsymbol{\phi}^{n+1,m}, \boldsymbol{\phi}^n), \quad (56)$$

where m is the index of the V-cycle iteration. Of course the fixed point of (56) is a solution of (53). Provided it converges, we may consider the V-cycle iteration process as simply a method to produce successively better approximations to the fixed point, namely the solution $\boldsymbol{\phi}^{n+1}$, of the system (56). In particular, the outer V-cycle iteration process produces the sequence $\{\boldsymbol{\phi}^{n+1,1}, \dots, \boldsymbol{\phi}^{n+1,m}, \dots\}$, starting from $\boldsymbol{\phi}^{n+1,0} = \boldsymbol{\phi}^n$, and stopping at some value $\boldsymbol{\phi}^{n+1,m'}$ when an error criterion is met. We then set $\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^{n+1,m'}$. Note that here we have abused notation, since $\boldsymbol{\phi}^{n+1}$ is only an approximate solution of $\mathbf{N} = \mathbf{S}$ at the $n + 1$ st time step.

A single iteration of the FAS V-cycle may be represented in operator form as

$$\phi^{n+1,m+1} = \text{FASVcycle}(k = k_{\max}, \phi^{n+1,m}, \mathbf{S}(\phi^{n+1,m}, \phi^n)), \tag{57}$$

which in words is equivalent to the following: find an approximate solution $\phi^{n+1,m+1} \approx \hat{\phi}$ to $\mathbf{N}(\hat{\phi}) = \mathbf{S}(\phi^{n+1,m}, \phi^n)$ using one V-cycle iteration of the nonlinear FAS multigrid method with initial approximation given by $\phi^{n+1,m}$. The index k is the multigrid level, and k_{\max} corresponds to the finest level. The operator FASVcycle is recursive, first called on the finest level. We explore the recursive aspect in detail for the adaptive version of this operator in the next section. The complete V-cycle iteration loop can be summarized as

```

initialize  $\phi^{n=0}$ 
for  $n = 0, n_{\max} - 1$ 
  set  $\phi^{n+1,m=0} = \phi^n$ 
  FAS V-cycle loop: for  $m = 0, m_{\max} - 1$ 
    calculate  $\mathbf{S}(\phi^{n+1,m}, \phi^n)$ 
     $\phi^{n+1,m+1} = \text{FASVcycle}(k = k_{\max}, \phi^{n+1,m}, \mathbf{S}(\phi^{n+1,m}, \phi^n))$ 
    if  $\|\mathbf{S}(\phi^{n+1,m}, \phi^n) - \mathbf{N}(\phi^{n+1,m+1})\|_{l_2, \star} < \text{tol}$  then
      set  $\phi^{n+1} = \phi^{n+1,m+1}$  and exit FAS V-cycle loop
    end for FAS V-cycle loop
  end for

```

The norm in the algorithm is defined as

$$\|\mathbf{r}\|_{l_2, \star} = \sqrt{\frac{1}{3N_1N_2} \sum_{v=1}^3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (r_{i,j}^{(v)})^2}, \tag{58}$$

where v denotes the component of \mathbf{r} . Now that we have defined the V-cycle component of the FAS method, the next piece required is the choice of the nonlinear smoother. The precise details of the role of the smoothing operation in the FAS V-cycle can be found in [43, Section 5.3.4]. A single application of the smoothing iteration can be expressed in operator form as

$$\bar{\phi}_k = \text{Smooth}^\lambda(\phi_k, \mathbf{N}_k, \mathbf{S}_k), \tag{59}$$

where k is the current multigrid level index, λ is number of smoothing passes, ϕ_k is the k th level approximate solution used to start the smoothing cycle, and \mathbf{N}_k and \mathbf{S}_k are the k th level nonlinear operator and source term. For example, to get the k th level operator, we replace each instance of h in (47)–(49) with h_k , the grid spacing on level k . As is the usual case in multigrid, we need only describe the smoothing for the finest level of the mesh, $k = k_{\max}$; the smoothing operation on the coarser level is analogous.

We use the Red–Black Gauss–Seidel-type iteration as the smoother, denoted as GS-RB. However, since the lexicographic Gauss–Seidel-type iteration, denoted as GS-LEX, is algorithmically similar, but much easier to describe, we give the details of the GS-LEX iteration instead. Detailed information on the GS-RB smoother, including typical performance gains over GS-LEX, can be found in [43]. In GS-LEX one proceeds lexicographically through the indices (i, j) from $(1, 1)$ to (N_1, N_2) . A complete traversal through all the indices of the grid constitutes a single smoothing iteration. We index the smoothing iterations by l . For example, $l = 2$ indicates the second smoothing pass, or iteration. The algorithm stops when $l = \lambda$ complete passes have been made through the indices. Once the field variables at (i, j) are updated in the relaxation scheme they are stored in $\phi_{i,j}^{n+1,l+1}$ and are immediately used to update the field variables at nearest neighbor grid sites.

We set

$$M_1^l = M((c_{i,j}^{n+1,l} + c_{i+1,j}^{n+1,l})/2), \quad M_2^l = M((c_{i-1,j}^{n+1,l+1} + c_{i,j}^{n+1,l})/2), \tag{60}$$

$$M_3^l = M((c_{i,j}^{n+1,l} + c_{i,j+1}^{n+1,l})/2), \quad M_4^l = M((c_{i,j-1}^{n+1,l+1} + c_{i,j}^{n+1,l})/2), \tag{61}$$

$$\bar{M}^l = \sum_{e=1}^4 M_e^l, \tag{62}$$

where the M_e^l are treated as known quantities in the smoothing. We linearize the nonlinear energy term f locally in the GS-LEX iteration via

$$f(c_{i,j}^{n+1,l+1}) \approx f(c_{i,j}^{n+1,l}) + f'(c_{i,j}^{n+1,l})(c_{i,j}^{n+1,l+1} - c_{i,j}^{n+1,l}). \tag{63}$$

The GS-LEX relaxation is the solution $\phi_{i,j}^{n+1,l+1}$, for fixed (i,j) , of the following three equations:

$$c_{i,j}^{n+1,l+1} + \frac{s\bar{M}^l}{2h^2} \mu_{i,j}^{n+1,l+1} = S_{i,j}^{(1)} + s \frac{M_1^l \mu_{i+1,j}^{n+1,l} + M_2^l \mu_{i-1,j}^{n+1,l+1}}{2h^2} + s \frac{M_3^l \mu_{i,j+1}^{n+1,l} + M_4^l \mu_{i,j-1}^{n+1,l+1}}{2h^2}, \tag{64}$$

$$-f'(c_{i,j}^{n+1,l})c_{i,j}^{n+1,l+1} + \mu_{i,j}^{n+1,l+1} + \left(1 + \frac{4\delta^2}{\epsilon^2 h^2}\right)v_{i,j}^{n+1,l+1} = S_{i,j}^{(2)} + f(c_{i,j}^{n+1,l}) - f'(c_{i,j}^{n+1,l})c_{i,j}^{n+1,l} + \frac{\delta^2}{\epsilon^2 h^2}(v_{i+1,j}^{n+1,l} + v_{i-1,j}^{n+1,l+1} + v_{i,j+1}^{n+1,l} + v_{i,j-1}^{n+1,l+1}), \tag{65}$$

$$\frac{4\epsilon^2}{h^2}c_{i,j}^{n+1,l+1} + v_{i,j}^{n+1,l+1} = S_{i,j}^{(3)} + \frac{\epsilon^2}{h^2}(c_{i+1,j}^{n+1,l} + c_{i-1,j}^{n+1,l+1} + c_{i,j+1}^{n+1,l} + c_{i,j-1}^{n+1,l+1}). \tag{66}$$

4. Mesh refinement and the adaptive multigrid algorithm

The adaptive process that we use is comprised of two basic steps: starting with a multilevel, block-structured mesh on which we have a computed solution, we (i) construct a new multilevel, block-structured mesh, and (ii) solve the problem on the new mesh using the adaptive version of the FAS multigrid method, called MLAT. The mesh construction is described in Section 4.1 and the MLAT algorithm is given in Section 4.2.

4.1. Construction of the multilevel mesh

Consider a rectangular computational domain $\Omega = (0, L_1) \times (0, L_2)$ and a patch-structured composite mesh (as in Fig. 1a (bottom)) consisting of a hierarchy of levels, indexed by $k = k_{\min}, \dots, -1, 0, 1, \dots, k_{\max}$, that covers Ω . The composite mesh can be decomposed into level- k meshes (as in Fig. 1a (middle)). Each level- k mesh consists of a collection of n_k uniform, rectangular grids $G_{i,k}$ of sizes $N_1^{(i,k)} \times N_2^{(i,k)}$, each with grid spacing h_k . The grids are non-overlapping except possibly along their edges. We use the cell-centered approximation of

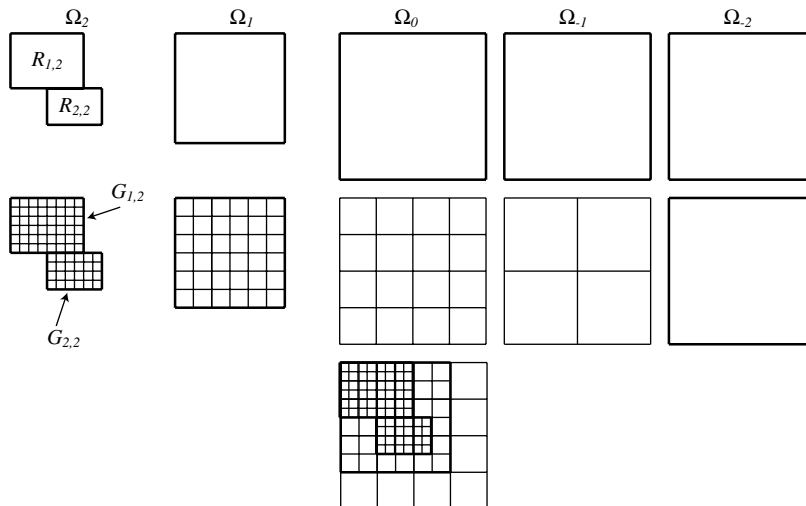


Fig. 1a. An illustration showing the level- k domains, Ω_k , (top), the level- k meshes (middle), for $k = -2, \dots, 2$, and the composite mesh (bottom).

the field variables on each grid $G_{i,k}$. Corresponding to each grid of the level- k mesh is the rectangular patch, denoted $R_{i,k}$, that the grid covers. Specifically, we define $R_{i,k}$ as the interior of the convex hull of $G_{i,k}$:

$$R_{i,k} = ((G_{i,k})_{c.h.})^\circ. \tag{67}$$

The level- k mesh covers a corresponding level- k domain, denoted $\Omega_k \subseteq \Omega$ (as in Fig. 1a (top)), where

$$\Omega_k = \left(\bigcup_{i=1}^{n_k} \bar{R}_{i,k} \right)^\circ, \quad R_{i,k} \cap R_{j,k} = \emptyset, \quad i \neq j, \tag{68}$$

and again n_k is the number of patches (or grids) comprising level k . At each of the levels $k = k_{\min}, \dots, 0$ the level- k mesh consists of a single grid covering the entire computational domain, *i.e.*, $n_k = 1$ and $R_{1,k} = \Omega_k = \Omega$. The grid at level 0 is called the root grid, and level $k = 0$ is called the root level.

The mesh must be nested, meaning the level- k domains satisfy

$$\Omega_k \subseteq \Omega_{k-1}, \quad k = k_{\min} + 1, \dots, k_{\max}. \tag{69}$$

Note that this is more general and less restrictive than the “proper” nesting requirement for a mesh generated in CHOMBO [39]. The grid spacing on level k is related to that of the level below via

$$2h_k = h_{k-1}. \tag{70}$$

Moreover, each level- k grid $G_{i,k}$ completely covers, as in Fig. 1a, some sub-grid $\hat{G}_{i,k}$ in the $k - 1$ level mesh of grid spacing h_{k-1} and of size $\hat{N}_1^{(i,k)} \times \hat{N}_2^{(i,k)}$, where $N_d^{(i,k)} = 2\hat{N}_d^{(i,k)}$, $d = 1, 2$.

For the root-level grid we write $N_d^{(0)} := N_d^{(1,0)}$, $d = 1, 2$. The minimum grid spacing on the composite mesh is denoted by h , and of course $h := h_{k_{\max}}$. The *global uniform grid*, denoted by G_h , is a uniform grid that has grid spacing equal to h , has grid sizes $N_d := 2^{k_{\max}} N_d^{(0)}$, $d = 1, 2$, and covers the entire computational domain Ω . By construction, the level- k_{\max} grids, $G_{i,k_{\max}}$, are sub-grids of G_h . Here, for simplicity, the time step size is the same on every level though this may be generalized through time-subcycling [38].

Construction of the multilevel mesh begins at the root-level grid. Finer resolution grids are added at level 1 to cover those grid points on the root grid where refinement is flagged. This process continues at level 1—finer resolution grids are added to level 2 to cover flagged cells in the grids of level 1—and ends when the predefined finest level, that corresponding to k_{\max} , is reached. It is straightforward to produce a mesh that is nested since refinement at level k can only be triggered at cells in level $k - 1$.

We may tag, or mark, grid cells where refinement is desired by a number of different *a priori* or *a posteriori* criteria. For this work we have used two different tests. The first, the *undivided gradient test*, is very simple and works for most cases in which we are interested. Since it is typically true that much more refinement is needed in the diffuse interface region, where c exhibits a sharp transition, this test marks grid cells where the finite differences of the order parameter are large. In particular, the set of cell-centered points where this test is passed is

$$P_k = \left\{ \mathbf{x}_{i,j} \in \Omega_k \mid \sqrt{(c_{i+1,j} - c_{i-1,j})^2 + (c_{i,j+1} - c_{i,j-1})^2} > C_k \right\}, \tag{71}$$

where C_k is the critical value for level k and where $\mathbf{x}_{i,j}$ are cell center coordinates defined with respect to the level- k grids $G_{i,k}$. The second test is called the *relative truncation error test*. In this scheme the relative truncation error between \mathbf{N}_k and \mathbf{N}_{k-1} is used as a means for populating P_k . In other words, cell center points where the truncation error is greater than C_k are added to the list P_k . More information on this can be found in [43, Sections 5.3.7 and 9.4.1]. In our tests, qualitatively similar meshes were constructed by both schemes; specifically, compare the results in Figs. 3b and 5b.

After P_k is constructed, by either of the two tests above, it may be expanded by buffering. This process, described in [39], increases the list of cells slated for refinement by adding $b \in \{0, 1, 2, \dots\}$ layers of cell-centered points around each point of P_k . This results in the buffered set $B_k \supseteq P_k$. We say a cell in $G_{i,k}$ is tagged for refinement if the coordinates of its cell center are in B_k . Tagged cells are required to be covered by the next finer level domain, *i.e.*, the discrete set B_k satisfies

$$B_k \subset \Omega_{k+1}. \tag{72}$$

The tagged cells are grouped into rectangular patches using the clustering algorithm given in Berger and Rigoutsos [51].

After new grids have been constructed at level $k + 1$ that cover the tagged points B_k , they must be populated with data from the old mesh. This process is illustrated in Fig. 1b. For simplicity in Fig. 1b we assume that two level-1 grids, one old and one new, live simultaneously, where the field variables are to be transferred from old to new. Both level-1 grids share the same parent grid, the root grid (level 0). Data from the old root-level grid are copied into the new root-level grid; note that the root-level grid never changes in size. Cell-centered data contained in the overlap of the old and new level-1 grids are simply copied. The level-1 data in the old grid that are not in the overlap are averaged and copied to the new root grid. The data in the new level-1 grid located where there is no overlap with an old level-1 grid must be generated by interpolation from the cell-centered data in the root grid. The four different symbols in the interpolation stencil are employed to suggest that bilinear interpolation should be used, though piecewise constant interpolation may also be used. The situation gets only slightly more complicated as we continue up in levels.

In order to employ the standard smoothing operation on a generic level- k grid, we must construct a single layer of ghost cells that surrounds the grid. This allows us to use the usual nearest neighbor stencils in the definitions of the discrete derivatives without any modification. Our method is to first fill the ghost-layer values by interpolation (Fig. 1c), then to exchange grid values among neighbor grids that are in direct contact (Fig. 1d). There are several interpolation options available to us for this. We illustrate the I and the II interpolation methods in Fig. 1c. The grid-to-ghost-layer exchange process is illustrated in Fig. 1d. Although neither the I method nor the II method, which is based on the method in [39], is conservative, we find that mass is maintained to a high degree of accuracy in our scheme. We refer the reader to [52] for a different ghost cell interpolation method.

4.2. Adaptive multigrid V-cycle algorithm

The multilevel adaptive technique (MLAT) proposed by [42] is the basis of our adaptive multigrid scheme. MLAT is in essence just a generalization of the FAS method [43], in the sense that the full problem is reconstructed at each of the coarser levels, rather than an equivalent defect problem as in the linear multigrid approach. The MLAT method automatically handles both linear and nonlinear problems, where the nonlinearity is usually handled in the smoothing step. The algorithm described here updates the solution at every level for each time step. The case of time-subcycling—wherein the solution on level Ω_k is updated two times before the solution on Ω_{k-1} is updated—can also be done, but at the expense of some added algorithmic complexity [38].

We assume that the level- k operators N_k and the level- k source terms S_k are given for $k = k_{\min}, \dots, -1, 0, 1, \dots, k_{\max}$. These can be defined as in Section 3 since the mesh on each level consists of the union of rectangular grids. Likewise, the smoothing operation defined in Section 3 translates to the adaptive mesh,

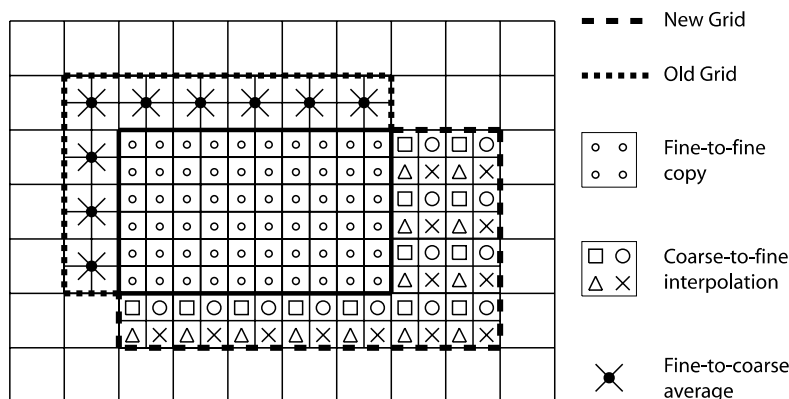


Fig. 1b. Illustration of populating a new level-1 grid with the data from an old level-1 grid and the root-level grid.

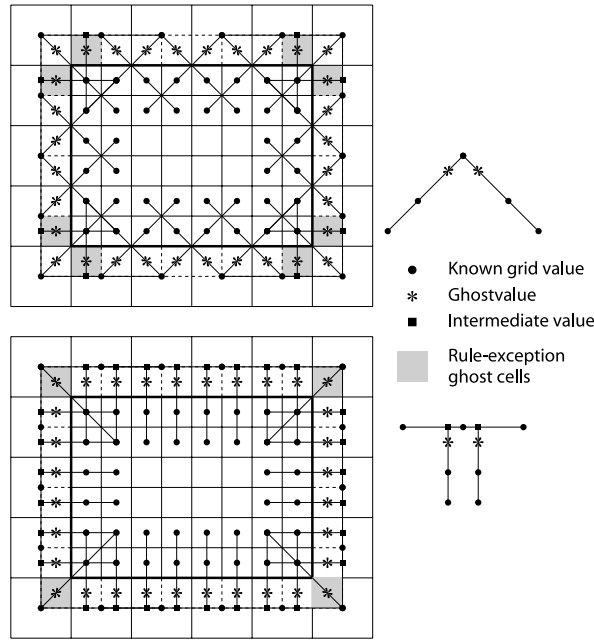


Fig. 1c. The *A* (top) and the *H* (bottom) quadratic interpolation methods for filling points in the ghost layer surrounding a grid. The *A* method requires half the computation of the *H* method, and is generally more accurate; but it has eight rule-exception ghost cells per grid. The *H* method has only four rule-exception cells and extends more easily for refinement ratio greater than 2, which is assumed here. Neither is conservative, generally.

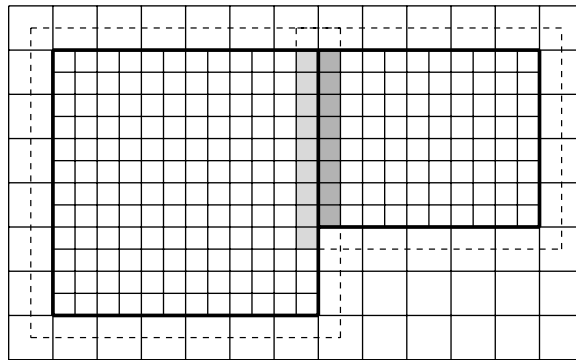


Fig. 1d. Illustration of the exchange of data from a grid to the ghost layer of its neighbor. After the entire ghost layer is filled via interpolation, more accurate values are copied into the locations where the ghost layer overlaps with a neighboring grid. The light grey cells are those located in the ghost layer of the grid on the left. The dark grey cells belong to the ghost layer of the grid on the right.

because smoothing on a level reduces to smoothing on a finite number of rectangular grids. Recall that the mesh at level $k = 0$ consists of a single grid covering the whole computational domain. Below level 0, the root level, the method reduces to the ordinary FAS multigrid scheme. We denote the restriction operator by \mathbf{I}_k^{k-1} , and the prolongation operator by \mathbf{I}_{k-1}^k , for $k = k_{\min} + 1, \dots, 0, \dots, k_{\max}$ [43]. We use averaging for the restriction operator, and either piecewise constant or (mass-corrected) bilinear interpolation for the prolongation operator. (See Trottenberg et al. [43, Section 2.8.4] for more details.) We assume that the pre- and post-smoothing cycles are the same in number, λ , though this can be generalized. We employ the V-cycle algorithm for moving from level to level; thus we begin the associated iteration at the finest level, $k = k_{\max}$. We seed the V-cycle iteration by using the previous time iterate at the starting point: $\phi_k^{n+1, m=0} = \phi_k^n$, for all levels k . The

calculation of a new k th-level iterate $\phi_k^{n+1,m+1}$ from approximations at both the fine level, $\phi_k^{n+1,m}$, and coarse level, $\phi_{k-1}^{n+1,m}$, proceeds as follows:

Recursive MLAT V-cycle operator

$$\phi_k^{n+1,m+1} = \text{MLATVcycle}(k, \phi_k^{n+1,m}, \phi_{k-1}^{n+1,m}, \mathbf{N}_k, \mathbf{S}_k, \lambda) \quad (73)$$

(1) Pre-smoothing:

- Compute a smoothed approximation $\bar{\phi}_k$ on Ω_k :

$$\bar{\phi}_k = \text{Smooth}^\lambda(\phi_k^{n+1,m}, \mathbf{N}_k, \mathbf{S}_k). \quad (74)$$

(2) Coarse-grid correction:

- Compute coarse-level initial iterate:

$$\bar{\phi}_{k-1} = \begin{cases} \mathbf{I}_k^{k-1} \bar{\phi}_k & \text{on } \Omega_{k-1} \cap \Omega_k, \\ \phi_{k-1}^{n+1,m} & \text{on } \Omega_{k-1} - \Omega_k. \end{cases} \quad (75)$$

- Update the ghost cells in the level- $k-1$ grids using quadratic interpolation and data exchange from neighbors for the data $\bar{\phi}_{k-1}$.
- Compute the coarse-level right-hand side:

$$\mathbf{S}_{k-1} = \begin{cases} \mathbf{I}_k^{k-1}(\mathbf{S}_k - \mathbf{N}_k(\bar{\phi}_k)) + \mathbf{N}_{k-1}(\mathbf{I}_k^{k-1} \bar{\phi}_k) & \text{on } \Omega_{k-1} \cap \Omega_k, \\ \mathbf{S}_{k-1} & \text{on } \Omega_{k-1} - \Omega_k. \end{cases} \quad (76)$$

- Compute an approximate solution $\hat{\psi}_{k-1}$ of the following coarse-grid equation on defined on Ω_{k-1} :

$$\mathbf{N}_{k-1}(\hat{\psi}_{k-1}) = \mathbf{S}_{k-1}. \quad (77)$$

- If $k = k_{\min} + 1$ employ λ smoothing steps:

$$\hat{\psi}_{k_{\min}} = \text{Smooth}^\lambda(\bar{\phi}_{k_{\min}}, \mathbf{N}_{k_{\min}}, \mathbf{S}_{k_{\min}}). \quad (78)$$

- If $k > k_{\min} + 1$ get an approximate solution to Eq. (77) using $\bar{\phi}_{k-1}$ as initial guess:

$$\hat{\psi}_{k-1} = \text{MLATVcycle}(k-1, \bar{\phi}_{k-1}, \phi_{k-2}^{n+1,m}, \mathbf{N}_{k-1}, \mathbf{S}_{k-1}, \lambda). \quad (79)$$

- Compute the coarse-grid correction on $\Omega_{k-1} \cap \Omega_k$:

$$\hat{\phi}_{k-1} = \hat{\psi}_{k-1} - \bar{\phi}_{k-1}. \quad (80)$$

- Set the solution on $\Omega_{k-1} - \Omega_k$:

$$\phi_{k-1}^{n+1,m+1} = \hat{\psi}_{k-1}. \quad (81)$$

- Compute the coarse-grid-corrected approximation at level k on Ω_k :

$$\phi_k^{\text{CGC}} = \mathbf{I}_{k-1}^k \hat{\phi}_{k-1} + \bar{\phi}_k. \quad (82)$$

- Update the ghost cells in the level- k grids using quadratic interpolation and data exchange from neighbors for the data ϕ_k^{CGC} .

(3) Post-smoothing:

- Compute $\phi_k^{n+1,m+1}$ by applying λ smoothing steps on Ω_k :

$$\phi_k^{n+1,m+1} = \text{Smooth}^\lambda(\phi_k^{\text{CGC}}, \mathbf{N}_k, \mathbf{S}_k). \quad (83)$$

- Update the ghost cells in the level- k grids using quadratic interpolation and data exchange from neighbors for the data $\phi_k^{n+1,m+1}$.

The complete MLAT V-cycle iteration loop is analogous to the FAS version given in Section 3 and is given in the following:

```

initialize  $\phi_k^{n=0}, k = k_{\min}, \dots, k_{\max}$ 
for  $n = 0, n_{\max} - 1$ 
  set  $\phi_k^{n+1, m=0} = \phi_k^n, k = k_{\min}, \dots, k_{\max}$ 
  MLAT V-cycle loop: for  $m = 0, m_{\max} - 1$ 
    calculate  $S_k(\phi_k^{n+1, m}, \phi_k^n), k = k_{\min}, \dots, k_{\max}$ 
     $\phi_{k_{\max}}^{n+1, m+1} = \text{MLATVcycle}(k = k_{\max}, \phi_{k_{\max}}^{n+1, m}, \phi_{k_{\max}-1}^{n+1, m}, \mathbf{N}_{k_{\max}}, S_{k_{\max}}, \lambda)$ 
    if  $\|S_{k_{\max}}(\phi_{k_{\max}}^{n+1, m}, \phi_{k_{\max}}^n) - \mathbf{N}_{k_{\max}}(\phi_{k_{\max}}^{n+1, m+1})\|_{L_2, \star} < \text{tol}$  then
      set  $\phi_k^{n+1} = \phi_k^{n+1, m+1}, k = k_{\min}, \dots, k_{\max}$ , and exit MLAT V-cycle loop
    end for MLAT V-cycle loop
  end for

```

The norm in the algorithm is defined as

$$\|\mathbf{r}_{k_{\max}}\|_{L_2, \star} = \sqrt{\frac{h_{k_{\max}}^2}{3|\Omega_{k_{\max}}|} \sum_{g=1}^{n_{k_{\max}}} \sum_{v=1}^3 \sum_{i=1}^{N_1^{(g, k_{\max})}} \sum_{j=1}^{N_2^{(g, k_{\max})}} (r_{i,j}^{(v, g)})^2}, \tag{84}$$

where v is the variable index and g is the grid number index.

5. Numerical results

In this section we demonstrate the second-order accuracy of the adaptive method. We then present examples to show the power of the adaptive method in computing the evolution of interfaces with complex morphologies. In all of the following simulations the mesh is adapted at each time step, whereas the time step size s remains fixed in time. Except for the simulation of Figs. 5a and 5b where the relative truncation error test [43, Sections 5.3.7 and 9.4.1] is used, the undivided gradient test in (71) is used to tag cells for refinement. We use the mass-corrected bilinear interpolation as the prolongation operator \mathbf{I}_{k-1}^k in all simulations except that of Fig. 2, where we use piecewise constant interpolation. We use the fourfold anisotropy function $\gamma(\mathbf{n}) = 1 + a\Gamma(\mathbf{n})$, where $\Gamma(\mathbf{n}) = 4\sum_{i=1}^d n_i^4 - 3$, for $d = 2, 3$. In 2D this is equivalent to $\gamma(\theta) = 1 + a\cos(4\theta)$, where $\tan(\theta) = n_2/n_1$. We have also used our method with eightfold anisotropy in 2D and 3D, but do not show results for this case. In the isotropic case ($a = 0$) we consider bulk spinodal decomposition, where a complex microstructure coarsens, and in the anisotropic case we consider strongly anisotropic systems, where corners and near-facets develop during the evolution and regularization is required for well-posedness. We also provide a calculation of the efficiency of our adaptive algorithm.

5.1. Convergence of the adaptive multigrid solver

A convenient way to compare solutions on two different composite meshes is to populate and compare their corresponding global uniform grids. Recall that the global uniform grid, G_h , is a uniform grid that has grid spacing equal to $h := h_{k_{\max}}$, with grid sizes $N_d := 2^{k_{\max}} N_d^{(0)}$, $d = 1, 2$. For example, if the root grid has sizes $N_d^{(0)} = 64$, $d = 1, 2$, and there are $k_{\max} = 2$ levels of refinement, then the global uniform grid G_h has sizes $N_d = 256$, $d = 1, 2$, and grid spacing $h = h_2$. The solution on a composite mesh with $k_{\max} = 2$ is interpolated to the cell centers of the global uniform grid G_h using the prolongation operator \mathbf{I}_{k-1}^k as follows:

$$c^{(h)}(x_1, x_2) = \begin{cases} c(x_1, x_2) & \text{on } \Omega_2, \\ \mathbf{I}_1^2 c(x_1, x_2) & \text{on } \Omega_1 - \Omega_2, \\ \mathbf{I}_1^2 \mathbf{I}_0^1 c(x_1, x_2) & \text{on } \Omega_0 - \Omega_1. \end{cases} \tag{85}$$

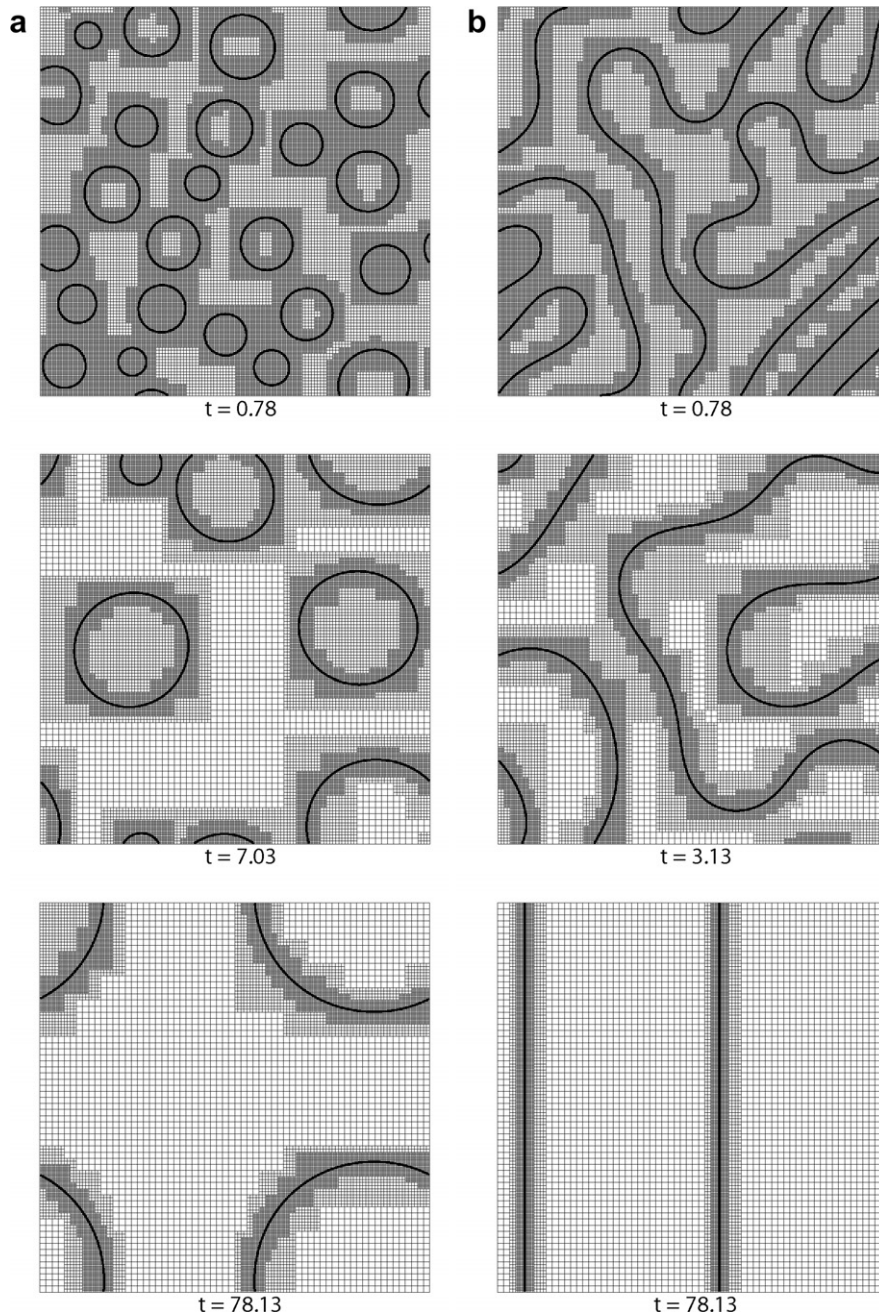


Fig. 2. Evolution of the mesh and the $c(\mathbf{x}, t) = 0.5$ iso-contours under spinodal decomposition and subsequent coarsening. The average concentrations are (a) $c_m = 0.3$ and (b) $c_m = 0.5$. There are two levels of refinement and the effective fine grid resolution is 256×256 .

To estimate the convergence rate simulations are performed using the three different grid spacings $2h_0$, h_0 , and $h_0/2$ in the finest grids of the composite mesh. Therefore, three solutions $c^{(2h_0)}$, $c^{(h_0)}$, and $c^{(h_0/2)}$ are defined at time t_{final} on the global uniform grids G_{2h_0} , G_{h_0} , and $G_{h_0/2}$, respectively. Since $c^{(h_0)}$ is defined at the cell centers we define the “error” between $c^{(h_0)}$ and $c^{(h_0/2)}$ to be

$$e_{i,j}^{h_0:h_0/2} := c_{i,j}^{(h_0)} - \frac{1}{4} \left(c_{2i,2j}^{(h_0/2)} + c_{2i-1,2j}^{(h_0/2)} + c_{2i,2j-1}^{(h_0/2)} + c_{2i-1,2j-1}^{(h_0/2)} \right), \quad (86)$$

and correspondingly for $e^{2h_0:h_0}$. The rate of convergence is defined as

$$\log_2 \left(\frac{\|e^{2h_0:h_0}\|_{l_2}}{\|e^{h_0:h_0/2}\|_{l_2}} \right). \tag{87}$$

Here we examine the convergence of our algorithm in the presence of strong anisotropy. The initial condition is the unit circle described by

$$c(x_1, x_2) = \frac{1}{2} \left[1 - \tanh \left(\frac{\sqrt{(x_1 - 1.6)^2 + (x_2 - 1.6)^2} - 1}{2\sqrt{2}\epsilon} \right) \right]. \tag{88}$$

The domain is $\Omega = (0, 3.2) \times (0, 3.2)$ and the solution is evolved to time $t_{\text{final}} = 1.0$. Here, the anisotropy parameter is $a = 0.2$, $\epsilon = 1.8 \times 10^{-2}$ and $\delta = 1.0 \times 10^{-3}$. The mobility is $M = 1$. We use the fourfold anisotropy function with anisotropy parameter $a = 0.2$. Since $a = 0.2 > 1/15$, there are energetically unfavorable orientations, and the Wulff shape corresponding to the (non-regularized) sharp interface model has sharp corners. We use the condition (71) to tag for refinement with $C_k = 0.01$, $k = 0, 1$. The maximum number of levels of refinement is $k_{\text{max}} = 2$, and the root-level grid sizes are $N_d^{(0)} = 16, 32, 64, 128$, $d = 1, 2$. The time step sizes are $s = 0.05$ h, where, recall, $h := h_{k_{\text{max}}} = h_2$. We use a single pre-smoothing and a single post-smoothing relaxation pass, *i.e.*, $\lambda = 1$. The algorithm terminates at a given time iterate, n , when the scaled l_2 norm (84) of the residual on the finest level mesh is less than $\text{tol} = 1.0 \times 10^{-8}$. The errors and rates of convergence are shown in Table 1. The results suggest that our algorithm is second-order accurate even in the presence of strong anisotropy and missing orientations. The evolution of a circle toward the Wulff shape with missing orientations is studied in more detail in Section 5.3. The energy evolution and equilibrium shapes corresponding to the parameters of the test are presented in Figs. 3c and 3d.

5.2. Isotropic, bulk spinodal decomposition

To show the robustness of the algorithm, we present simulations of spinodal decomposition under isotropic conditions ($a = \delta = 0$) in a periodic domain using our adaptive algorithm. In these simulations we consider the evolution of an initially nearly uniform composition field. In particular, $c(\mathbf{x}, 0) = c_m + \zeta(\mathbf{x})$, where ζ is random and $|\zeta(\mathbf{x})| < 10^{-3}$ and c_m is in the so-called spinodal region, *i.e.*, $F''(c_m) < 0$. Here, we consider two cases, $c_m = 0.3$ (off-critical) and $c_m = 0.5$ (critical). We show that using the adaptive mesh algorithm results in enormous computational savings compared to uniform mesh algorithms.

In both simulations, the computational domain is the unit square $\Omega = (0, 1) \times (0, 1)$ and periodic boundary conditions are used. The mobility is $M = 1$. We take $\epsilon = 2.5 \times 10^{-3}$, $a = 0$, and $\delta = 0$. The root-level grid size is $N_d^{(0)} = 64$, the number of levels of refinement is $k_{\text{max}} = 2$, and the time step size is $s = 0.1$ h, where $h = h_2 = 1/256$. The global grid has size $N_d = 256$, $d = 1, 2$. At the finest level, there are approximately 5–7 grid points across the interface. Refinement is effected through Eq. (71) with $C_k = 0.03$, $k = 0, 1$.

The results for the off-critical composition $c_m = 0.3$ are shown in Fig. 2a while those for the critical composition $c_m = 0.5$ are shown in Fig. 2b. Initially, both systems are unstable and distinct phase regions form where $c \approx 0$ and $c \approx 1$, *e.g.*, see [1]. At early times, large spatial gradients form at small scales (*i.e.*, $\sim \epsilon$) throughout both systems. At later times as the systems coarsen to reduce the total surface energy, the number of second-phase regions decreases and the overall length-scale between regions increases. In the off-critical case, isolated particles rapidly form and coarsen. The mesh adapts around the particle interfaces; away from interfaces the gradients of c are small. The evolution tends to a single circular domain. In the critical case $c_m = 0.5$, a serpentine structure emerges at early times. The mesh adapts around the interfaces as they straighten to

Table 1
Errors and convergence rates for a circle relaxing to the Wulff shape

Root-level grid sizes	32 ² –64 ²	Rate	64 ² –128 ²	Rate	128 ² –256 ²
Error	1.229×10^{-2}	2.064	2.939×10^{-3}	1.995	7.369×10^{-4}

Parameters are given in the text and the initial data are from Eq. (88).

reduce the isotropic surface energy. The evolution tends to a single rectangular second-phase domain bounded by interfaces parallel to the y axis.

For both cases, the number of cells, defined as the number of cells visible in the composite mesh, rapidly decreases by half by time $t = 10$ from 65,536 (256^2) initially as the systems phase separates. At later times, the decrease in the number of cells is not as rapid but is still significant. At the final time, there are approximately 1.5×10^4 cells in the $c_m = 0.3$ case and 1.1×10^4 cells in the $c_m = 0.5$ case. This corresponds to roughly a sixfold decrease in the number of computational cells compared to the initial configuration. Although it not shown, the energy (40) is monotonically decreasing for both simulations. Further, mass is conserved to within 0.2% for $c_m = 0.3$ and to within $4 \times 10^{-4}\%$ for $c_m = 0.5$.

5.3. Anisotropic surface energy

Herein we consider the evolution of shapes toward equilibrium under strong fourfold anisotropy and as a function of the high-order regularization parameter δ . By shapes we mean the $c = 0.5$ level curves (2D) and level surfaces (3D) that divide phase regions where $c \approx 1$ and $c \approx 0$. We present snapshots of the evolving $c = 0.5$ level sets, and in one case the evolution of the total system energy. In the case of closed shapes, we compare the near-equilibrium $c = 0.5$ level sets with the classical sharp interface equilibrium (Wulff) shape, as was done in [32]. We also provide a calculation of the efficiency of our adaptive algorithm using data from the simulation of Figs. 5a and 5b.

For the 2D simulations in Figs. 3a–3d the initial condition is given in Eq. (88), which describes an initially circular shape. The homogeneous Neumann boundary conditions of (27) are used. The anisotropy parameter is $a = 0.2$; the mobility is $M = 1$; the domain size is $\Omega = (0, 3.2)^2$; the root grid size is $N_d^{(0)} = 32$, $d = 1, 2$; and there are $k_{\max} = 3$ levels of refinement. The global uniform grid has size $N_d = 256$, $d = 1, 2$. The time step is $s = 0.08h$, where $h = h_3 = 3.2/256$, and $\epsilon = 1.8 \times 10^{-2}$. There are approximately 8–10 mesh points across the evolving diffuse interface.

Snapshots from the evolution of the initially circular shape with the regularization parameter $\delta = 6.0 \times 10^{-4}$, are shown in Fig. 3a. Very early, two hills and one valley form at the compass points of the circle. This structure emerges because the $(1, 0)$ and surrounding orientations are high energy, and there is a large driving force for them to be removed. A single hill structure emerges at long times. A snapshot of

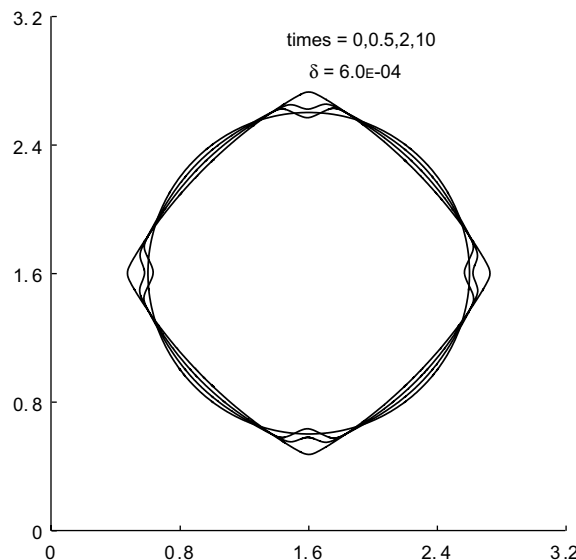


Fig. 3a. Evolution of the $c = 0.5$ level curves by the strongly anisotropic Cahn–Hilliard equation with higher-order regularization. The parameters are given in the text, and the initial shape is circular (Eq. (88)). The simulation was performed with a four-level composite mesh, $k_{\max} = 3$, with a root-level grid of 32×32 , as shown in Fig. 3b at time $t = 1.0$.

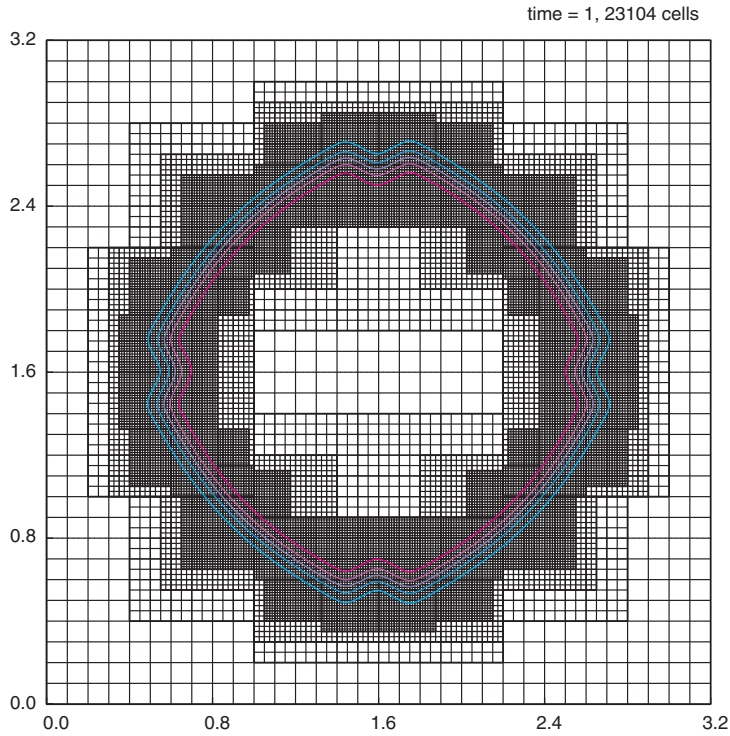


Fig. 3b. Mesh snapshot and five level curves of the order parameter c at $t = 1$ for the simulation shown in Fig. 3a. The contours $c = 0.1, 0.3, 0.5, 0.7, 0.9$ are on the spectrum from cyan ($c = 0.1$) to magenta ($c = 0.9$). The root-level grid has size 32×32 and there are three levels of refinement. The same effective resolution is given by a uniform grid of size 256×256 . Cell refinement is flagged based on (71). (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

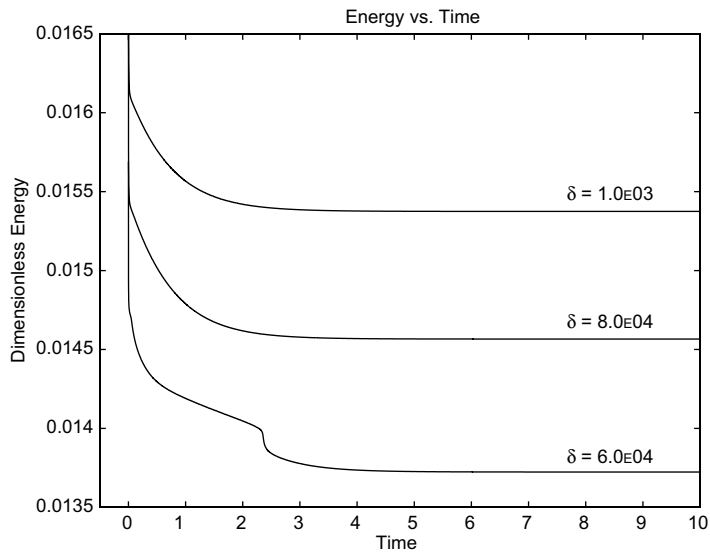


Fig. 3c. Energy profiles for the linearly regularized strongly anisotropic Cahn–Hilliard equation using three values of δ . The initial data are circular and given in Eq. (88). The parameters are given in the text. The near-equilibrium $c = 0.5$ level curves are shown in Fig. 3d. The energy plots suggest that the systems are very close to equilibrium by time $t = 10$.

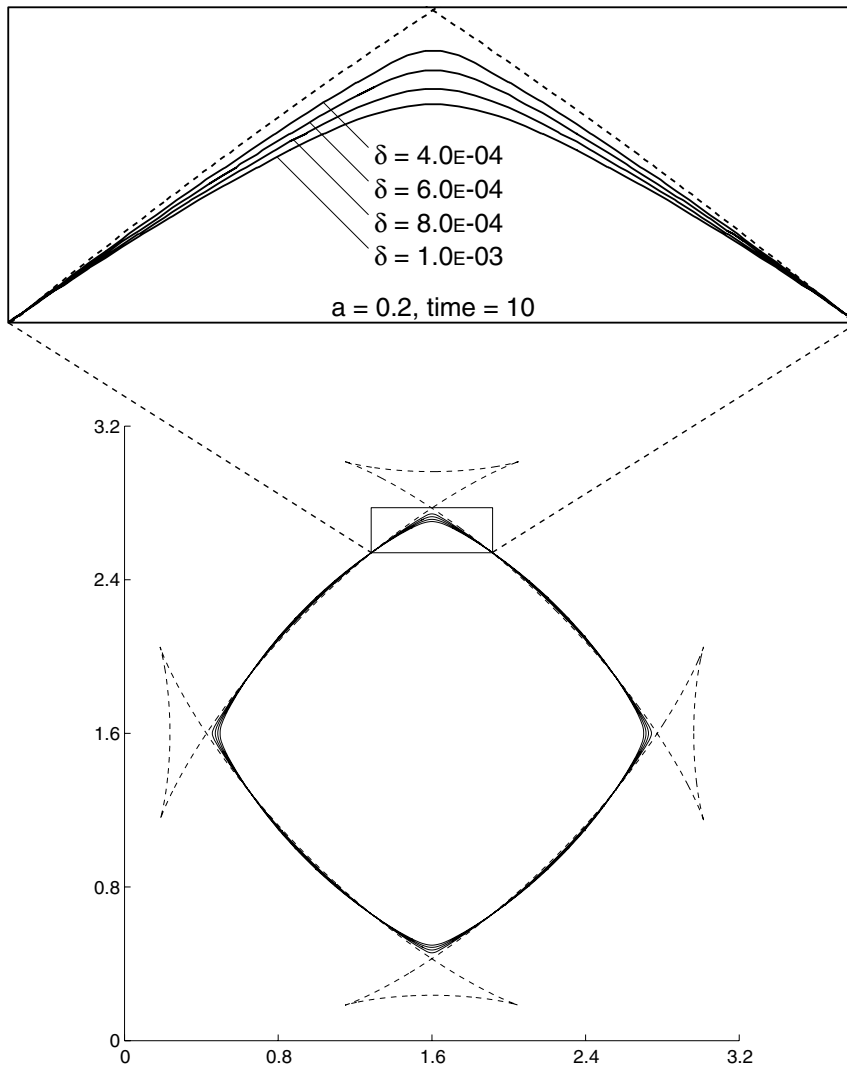


Fig. 3d. Near-equilibrium $c = 0.5$ level curve solutions to the regularized strongly anisotropic Cahn–Hilliard equation. Solutions are plotted at time $t = 10$. The parameters are given in the text. The corresponding energy data in Fig. 3c suggest that the shapes would be very nearly equilibrated by time $t = 8$. The window shows a magnification of approximately $7\times$.

the adaptive mesh at time $t = 1$ is shown in Fig. 3b. The composite mesh has three levels of refinement, $k_{\max} = 3$, and 23,104 cells. This should be compared to 65,536, the number of cells in a uniform 256×256 grid (*i.e.*, the global uniform grid) with equal resolution. In this simulation mass is conserved to within $2.0 \times 10^{-2}\%$.

In Fig. 3c the energy is plotted for the simulation of Figs. 3a and 3b together with the results from two additional values for δ , 8.0×10^{-4} and 1.0×10^{-3} . The large localized decrease in the energy at $t \approx 2.5$ for $\delta = 6.0 \times 10^{-4}$ coincides with the coarsening of the two-hill-one-valley structure to a single hill structure at the four compass points of the evolving shape. For the other two cases, $\delta = 8.0 \times 10^{-4}$, 1.0×10^{-3} , the evolving shapes (not shown) initially form a single hill structure at their compass points, and no coarsening events occur. Since there are no coarsening events, there are no large, localized decreases in the respective energies (apart from $t = 0$). In all cases, it is seen that the energy is non-increasing. In the simulations for the cases $\delta = 8.0 \times 10^{-4}$ and 1.0×10^{-3} mass is conserved to within $1.2 \times 10^{-2}\%$ and $6.5 \times 10^{-3}\%$, respectively.

The energy plots of Fig. 3c suggest that the respective systems are very nearly equilibrated by time $t = 10$. As was done in [32], we may compare the near-equilibrium $c = 0.5$ level curves with the sharp interface (non-regularized) Wulff shape. Here we are particularly interested in the effect of decreasing δ . Recall that the Wulff shape can be found in 2D in parametric form [53] via:

$$x_1(\theta) = A(\gamma(\theta) \cos(\theta) - \gamma'(\theta) \sin(\theta)), \tag{89}$$

$$x_2(\theta) = A(\gamma(\theta) \sin(\theta) + \gamma'(\theta) \cos(\theta)), \tag{90}$$

where A is a constant. As discussed earlier, when $\gamma(\theta) + \gamma''(\theta) < 0$, certain orientations have very high energy and hence are missing from the (sharp interface, non-regularized) Wulff shape. For fourfold symmetric anisotropy, missing orientations occur when $a > 1/15$. In this regime, the shapes described by Eqs. (89) and (90) develop “ears” that include metastable and unstable orientations. Mullins [54] proved that in 2D the Wulff shape is given by the convex shape, with sharp corners, that results after removal of the “ears”. The Wulff shape for $a = 0.2$, together with the ears, is shown as the dotted curves in Fig. 3d.

The near-equilibrium shapes for the simulations whose energy curves are depicted in Fig. 3c are shown in Fig. 3d. We have added also the near-equilibrium $c = 0.5$ level curve for $\delta = 4.0 \times 10^{-4}$, though the evolution of the level curves and energy are not shown for this case. In particular, the $c = 0.5$ level curves for four values of δ are plotted at $t = 10$ along with the (non-regularized) Wulff shape (dashed) corresponding to the anisotropy strength $a = 0.2$. Note that the areas inscribed by the Wulff shape, neglecting the ears, and the solution curves are not necessarily equal. In fact, the areas enclosed by the $c = 0.5$ solution curves are not assumed to be equal for different values of δ . As can be seen in the figure, there is rounding of the tip and deviation from the Wulff shape in the near tip region. However, all the curves appear to be reasonably and qualitatively close to the Wulff shape. More importantly, as δ is decreased, the solution curves tend towards the Wulff shape. Eggleston et al. [32] found that the $c = 0.5$ level curve of the equilibrium diffuse interface solution in the case of finite ϵ and $\delta = 0$ apparently coincides with the sharp interface (non-regularized) Wulff shape. Note that we are not claiming convergence of the near-equilibrium diffuse interface shape to the Wulff shape. Such a convergence would require both $\delta \rightarrow 0$ and $\epsilon \rightarrow 0$.

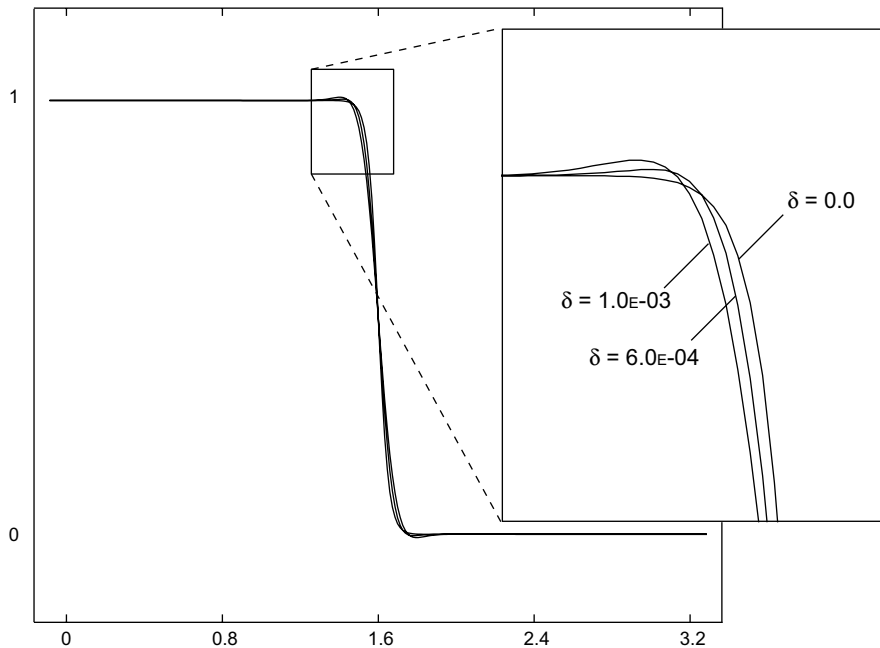


Fig. 4. Near-equilibrium 1D solutions of the Cahn–Hilliard equation with high-order regularization. Here the anisotropy parameter is set to $a = 0$, and $\epsilon = 1.8 \times 10^{-2}$. The effective mesh size is 256. The solution for $\delta = 0.0$ is near to $(1 - \tanh((x - 1.6)/(2\epsilon\sqrt{2}))) / 2$ and is monotonic. The solutions for $\delta > 0$ are not monotonic; extrema appear on the edges of the interfacial regions.

In addition to rounding corners, the Laplacian-squared regularization changes the local hyperbolic tangent profile of the interface (*i.e.*, profile of interface when $\delta = 0$). This phenomenon can be seen in Fig. 4 where a purely 1D problem is considered, *i.e.*, $\mathbf{n} = \mathbf{n}_0$ is fixed. The 1D problem is always well-posed even if $\delta = 0$; see Proposition 7.6. Observe that for $\delta > 0$ the interface profile is no longer locally monotonic, and does not remain bounded between 1 and 0. Our simulations, as well as theory, indicate that this holds for any $a \geq 0$. Notice that as suggested by Wheeler [36], δ behaves like a regular perturbation and 1D solutions converge to the $\delta = 0$ profile (which is nearly a hyperbolic tangent) as δ tends to zero.

In Fig. 5a we show the decomposition of an initially rough curve into near-facets and kinks (corners) and subsequent coarsening due to strong interfacial anisotropy. (For more background on this problem see [55,56,27,30].) The mobility of the Cahn–Hilliard equation models surface diffusion and has the form $M = c(1 - c) + \sigma$, where $0 < \sigma \ll 1$. The anisotropy is fourfold and the anisotropy parameter is $a = 0.5$.

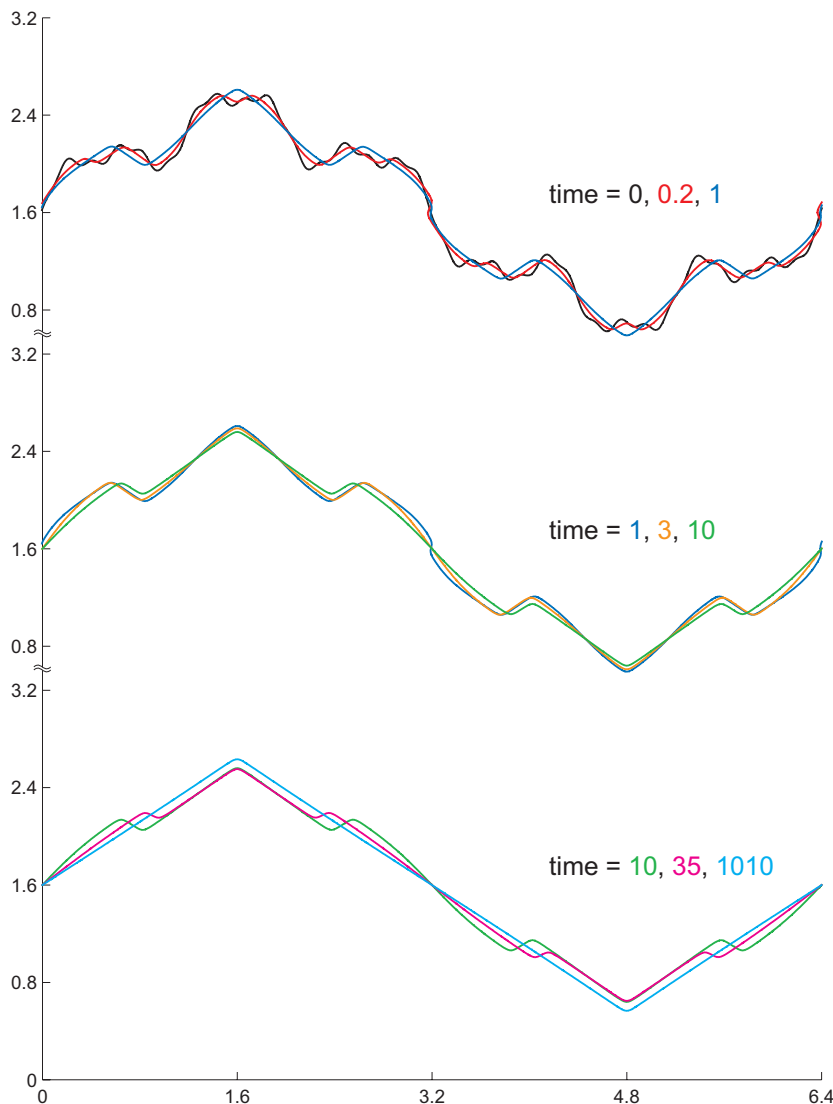


Fig. 5a. Decomposition of a rough surface into near-facets and kinks, followed by coarsening. Shown are the $c = 0.5$ level curves of the computed solution at nine different times. Decomposition occurs on a very fast time scale, whereas coarsening may occur on a very slow one. Note that although the $c = 0.5$ level curve is initially the graph of a function, it does not remain so. (For interpretation of the references in colour, the reader is referred to the web version of this article.)

The regularization parameter is $\delta = 1.0 \times 10^{-3}$, and $\epsilon = 1.8 \times 10^{-2}$. Plotted in Fig. 5a are the $c = 0.5$ level curves at nine times on the computational domain $\Omega = [0, 6.4] \times [0, 3.2]$. The initial data are

$$c(x_1, x_2) = \frac{1}{2} \left[1.0 - \tanh \left(\frac{x_2 - r\left(\frac{\pi x_1}{3.2}\right)}{2\sqrt{2}\epsilon} \right) \right], \tag{91}$$

where

$$r(z) = 1.6 + 0.8 \sin(z) + 0.2 \sin(5z) + 0.1 \sin(11z) + 0.03 \cos(31z). \tag{92}$$

Thus the initial $c = 0.5$ contour is the graph of the function $r(\pi x_1/3.2)$.

For the simulation of Figs. 5a and 5b the domain is periodic in the x_1 direction. The root-level grid has size 32×16 , and four levels of refinement are used, *i.e.*, $k_{\max} = 4$. The finest grid spacing is $h = h_4 = 3.2/256$, and equivalent resolution may be obtained using the global uniform grid, which has size 256×512 . On the time interval $[0, 10]$ the time step size $s = 1.0 \times 10^{-4}$ was used. This was increased to $s = 5.0 \times 10^{-3}$ for $t \in [10, 1010]$. Refinement tagging for this problem is based on the relative truncation error test [43], with a critical error of $C_k = 1.0 \times 10^{-7}$. The mesh at $t = 35$ is shown in Fig. 5b.

On a very fast time scale, the interface profile normal to the contour $c = 0.5$ quickly reaches a local equilibrium and is similar to that shown in Fig. 4. In particular, the profile is non-monotonic. Its precise thickness and shape depend on the interface orientation angle θ , as well as on ϵ and δ . Almost on the same fast time scale near-facets and kinks appear on the solution contours. Observe from the results in Fig. 5a that the solution contours do not necessarily remain the graphs of functions. The corners, which appear because certain ranges of orientations carry very high energy, coarsen away at various time scales. By time $t = 1010$ a one peak and one valley structure has emerged, and coarsening can no longer proceed due to periodicity. Note that the shape at $t = 1010$ is still evolving, as the peak rises and the valley sinks. Coarsening ends at a rather “early” time because the 1-mode is clearly dominant in the initial surface profile, as seen in Eq. (92). We have observed that this is not the case in general, where coarsening of facets may occur on a very slow time scale, perhaps exponentially slow. Note no strict comparison is here made with the extant sharp interface theories, as the precise asymptotic convergence of the diffuse interface model considered here has not been carried out. See

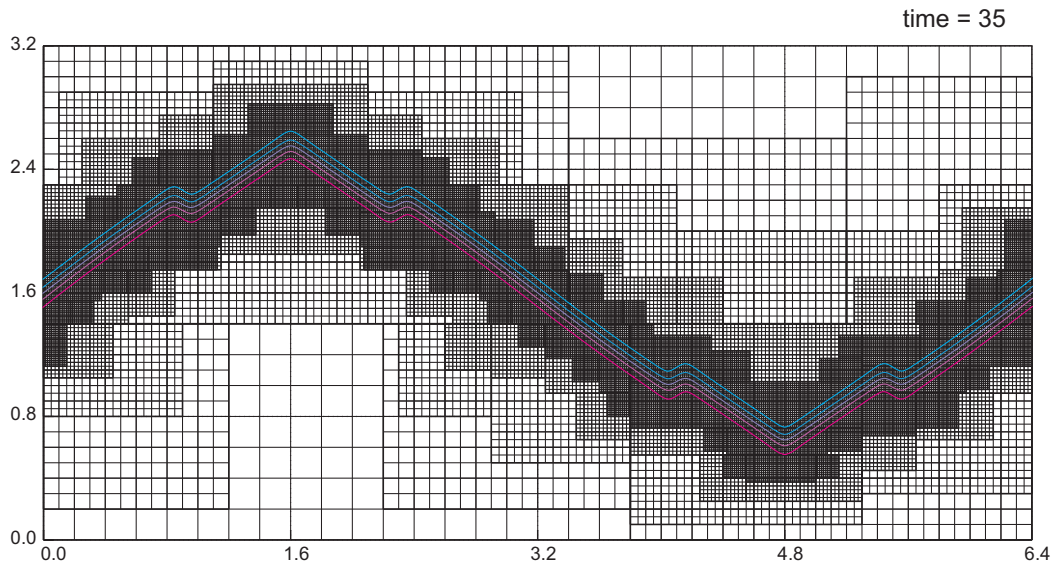


Fig. 5b. A snapshot at $t = 35.0$ of the adaptive mesh and five contours of the computed solution from the simulation in Fig. 5a. The adaptive mesh has four levels of refinement with a root-level grid of size 32×16 . The mesh shown has 37,058 cells, where a uniform mesh with the same resolution of the interface would have 131,072 cells. The contours $c = 0.1, 0.3, 0.5, 0.7, 0.9$ are colored on the spectrum from cyan ($c = 0.1$) to magenta ($c = 0.9$). (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

Wheeler [36] for some very recent asymptotic results for a related strongly anisotropic phase-field model of solidification.

In Fig. 5b a mesh snapshot and five level curves of the order parameter c are shown for the time $t = 35$. The adaptive mesh at $t = 35$ has 37,058 cells, 28.2% of the size of a uniform mesh with equal resolution of the diffuse interface. In Table 2 we report on the efficiency achieved using the adaptive mesh versus a uniform mesh during the first 100 time steps of the simulation reported in Figs. 5a and 5b. The uniform version of the present algorithm is obtained by starting with a root-level grid of size 256×512 and allowing no refinement, and is equivalent to the usual FAS multigrid [43]. I/O was turned off during the tests, which were performed on a Dell Optiplex GX270 with an Intel Pentium 4 cpu running at 3.2 GHz and 1 GB of RAM. Four runs (of 100 time steps) were conducted with both the uniform and adaptive mesh, and an average of the run times was used as the total cpu time, \mathcal{T}_{cpu} . As a reasonable way to compare the cost of calculating on the two meshes, we compare the *basic cell time*, i.e., the cpu time spent calculating on each cell for each V-cycle iteration, $\mathcal{T}_{\text{cpu}}/\mathcal{M}$ as defined in Table 2. The cost is lower for the uniform mesh. This is generally expected, because no time is spent on ghost cell interpolations or remeshing, for example. The efficiency, the ratio of the basic cell time of the uniform mesh algorithm to that of the adaptive mesh algorithm, is calculated for this example to be 0.965. An efficiency close to unity means that any reduction in mesh size with respect to the uniform mesh translates to the same reduction in real cpu time.

In Fig. 6 we show the 3D evolution of an initially three-bulbed surface according to the regularized strongly anisotropic Cahn–Hilliard equation together with the evolution of the mesh. Specifically, the initial $c = 0.5$ isosurface (time $t = 0$) shown in Fig. 6 is the outer envelope of three intersecting spheres. The homogeneous Neumann boundary conditions (27) are assumed. The root-level grid is 32^3 and there are two levels of refinement. The finest resolution is $h = h_2 = 3.2/128$, and the time step size $s = 0.16$ h. We take $\epsilon = 1.8 \times 10^{-2}$, $a = 0.2$, $\delta = 5.0 \times 10^{-4}$, and the mobility is $M = 1$. There are approximately five mesh points across the interface during the simulation. On a very fast time scale the shape forms smoothed corners as high energy orientations on the surface are removed, and a complex form results. At long times, after the coarsening of some surface features, the shape is evolving to an equilibrium structure resembling a smoothed, double-sided pyramid. In particular, the near-equilibrium $c = 0.5$ level surface at $t = 32$ compares qualitatively with the 3D Wulff shape [20, Fig. 2]. Mass is conserved to within $1 \times 10^{-3}\%$ in this simulation.

In Fig. 7a we simulate the coarsening of a surface in 3D. Here we model surface diffusion using the mobility $M = 4c(1 - c) + \sigma$, where as before $0 < \sigma \ll 1$. The computational domain is $(0, 6.4) \times (0, 6.4) \times (0, 3.2)$, the root-level grid is $64 \times 64 \times 32$, and there are two levels of refinement ($k_{\text{max}} = 2$). This degree of resolution puts approximately five mesh points across the interface during the simulation. The time step size is $s = 0.16$ h, where $h = h_2 = 3.2/128$. We take $\epsilon = 1.8 \times 10^{-2}$, $a = 0.2$ and $\delta = 4.0 \times 10^{-4}$. Periodic boundary conditions are used for c , μ and v in the x_1 and x_2 -directions, while homogeneous Neumann boundary conditions are used in the x_3 -direction. The initial $c = 0.5$ isosurface (not shown) is randomly perturbed about $x_3 = 1.6$. At early times, small-scale pyramid structures form as the high energy orientations are removed. Even at early times, the system is attempting to achieve long-range structure. In Fig. 7a, $t = 0.8$, one can see rows and columns forming. At later times, as the system undergoes significant coarsening, the hills (valleys) organize into distinct rows and columns.

At later times there remain partial rows of pyramids; the initial pyramid in one such partial row is indicated by the white arrow in Figs. 7a and 7b. The two surrounding neighbor rows of pyramids are coarsening at the

Table 2

A performance comparison of the adaptive mesh with the uniform mesh having equal resolution of the diffuse interface for the simulation of Figs. 5a and 5b

	$\sum_{i=1}^{100} \mathcal{G}_i$	$\sum_{i=1}^{100} \mathcal{V}_i$	$\mathcal{M} = \sum_{i=1}^{100} \mathcal{G}_i \mathcal{V}_i$	\mathcal{T}_{cpu} (s)	$\frac{\mathcal{T}_{\text{cpu}}}{\mathcal{M}}$ (s)
Uniform grid	131,072	18.1	237,240,320	380.505	1.6039×10^{-6}
Adaptive grid	52,392	18.56	97,211,653	161.510	1.6614×10^{-6}

The data are collected over the time interval $t \in [0, 0.01]$; the time step size is $s = 1 \times 10^{-4}$ and there are 100 time steps. \mathcal{G}_i is the number of mesh cells and \mathcal{V}_i is the number of V-cycle iterations, both at time step i . \mathcal{T}_{cpu} is the total cpu time to execute 100 iterations. The efficiency of the adaptive mesh for this problem, the quotient of the last two column entries, is $1.6039/1.6614 = 0.965$, or 96.5%.

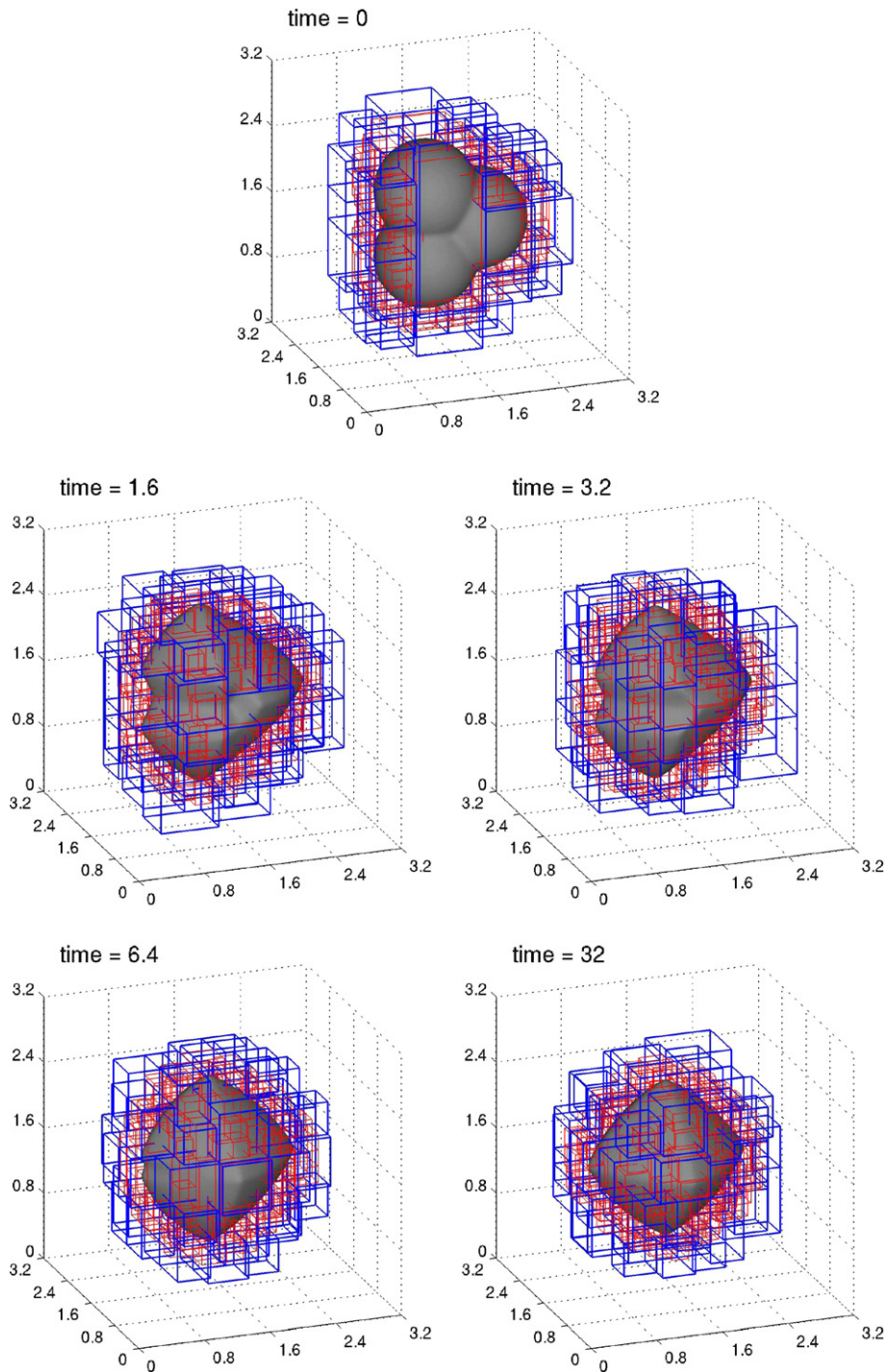


Fig. 6. Snapshots of the 3D evolution of a three-bulbed $c = 0.5$ isosurface by the strongly anisotropic Cahn–Hilliard equation with high-order regularization. Parameters for the simulation are given in the text. The bounding boxes of the $k = 1$ (blue) and $k = 2$ (red) level patches $R_{i,k}$ are shown. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

expense of the partial row of pyramids in the middle, as is shown in Fig. 7b. It is observed that overall coarsening in the surface is strongly mediated by the presence and evolution of these partial rows, which are akin to edge dislocations in a crystalline lattice. Moldovan and Golubovic [45] have reported this phenomenon pre-

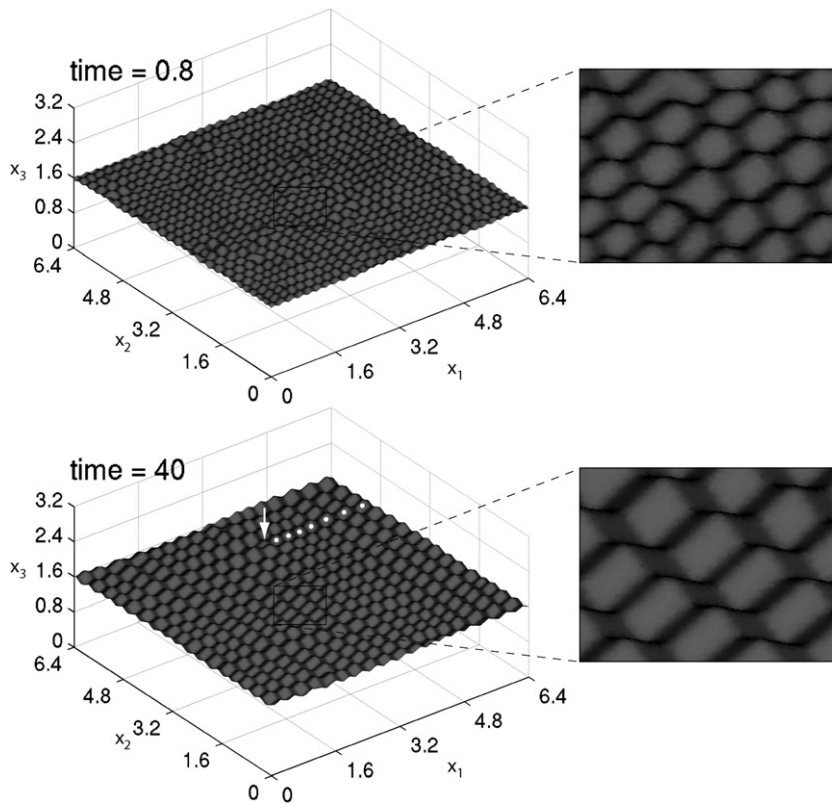


Fig. 7a. Coarsening of a corrugated surface in 3D. The parameters of the simulation are in given in the text, and the initial $c = 0.5$ isosurface coincides with $r(x_1, x_2) = 1.6 + \zeta(x_1, x_2)$, where ζ is a random perturbation of maximum magnitude 0.1. The simulation uses the high-order regularization, and surface diffusion is assumed. The magnification windows show that significant coarsening has occurred by $t = 40.0$. Moreover, long-range order is being established. The white arrow indicates the beginning of an incomplete row of pyramids, indicated by the white dots. The primary mechanism for coarsening is the removal of partial rows as described in the text and Fig. 7b.

viously using a height-function model to describe surface diffusion and deposition in square symmetry (001) surfaces. Characterizing the motion of the partial pyramid rows as dislocation climb, they note that once the average distance between dislocations becomes nearly equal to the size of the periodic domain coarsening essentially stops. In short, the removal of dislocation rows, significantly speeds up coarsening, relative a system in which there are no partial rows of pyramids.² This type of behavior has been observed in our simulations and may be quantified in a future work. We remark that a alternative way to visualize the evolution of these surfaces is via the edge network of the pyramids viewed looking in the negative x_3 -direction. See for example [45, Figs. 8 and 13]. See [45,57] for results on the coarsening of non-fourfold symmetry surfaces. For works modeling the coarsening of faceted surface using $2 + 1$ dimensional approaches see [58–60,45, 61,57].

6. Conclusions

In this paper, we described an adaptive multigrid/finite-difference algorithm for the strongly anisotropic Cahn–Hilliard equation with a high-order regularization in 2D and 3D. The resulting equation is sixth-order. One of the keys to the success of the method is the treatment of the anisotropic term. This term is discretized in conservation form in space and is fully implicit in time. Other key components of the algorithm are the use of dynamic, block-structured Cartesian mesh refinement and the use of an adaptive Full Approximation Storage

² Note that this phenomenon cannot exist in the coarsening of faceted curves in the plane.

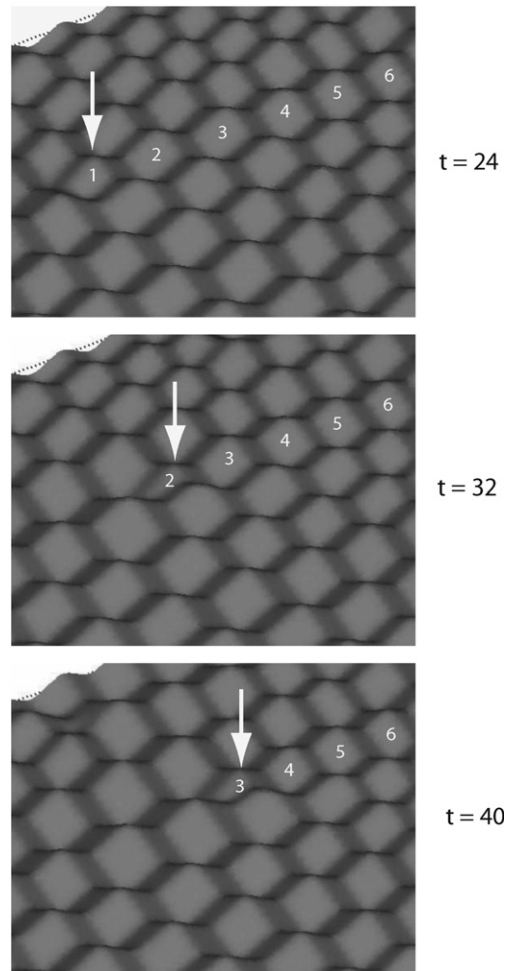


Fig. 7b. Magnified view of the simulation of Fig. 7a showing the removal of a partial row of pyramids during the coarsening of a corrugated surface. The white arrow at $t = 40$ points to the same location as the arrow in Fig. 7a, time 40. The surrounding rows merge as the intermediate partial row is eaten up.

nonlinear multigrid method to solve the nonlinear equations at the implicit time level. The resulting algorithm is robust, efficient and second-order accurate in space and time.

We demonstrated numerically the convergence of the adaptive scheme and showed some relevant computational examples in 2D and 3D. Using simulations of bulk, isotropic spinodal decomposition followed by domain coarsening, we demonstrated the enormous savings possible when dynamic mesh refinement is performed solely around interfacial regions. As in [32], we compared long-time diffuse interface solutions with the sharp interface Wulff shapes in 2D and 3D assuming the presence of strong interfacial anisotropy. For fixed interfacial parameter ϵ , we showed evidence that the closed, near-equilibrium 2D $c = 0.5$ level curves approach the Wulff shape as the regularization parameter δ is decreased. We presented an example of faceting decomposition in an initially rough curve in 2D. Analyzing the adaptive mesh characteristics for this example, we showed the efficiency of our algorithm is very close one. We also simulated large-scale coarsening of a corrugated surface (in 3D) evolving by anisotropic surface diffusion, and showed the emergence of long-range order during coarsening whereby the surface evolved into rows and columns of pyramidal structures. Our results were in qualitative agreement with those in [45]. In particular, we observed that coarsening of the surface is strongly mediated by the presence and evolution of partial pyramid rows, what Moldovan and Golubovic liken to dislocations in a crystal [45]. The dominant mechanism of system-wide coarsening involves two

neighboring rows of pyramids growing at the expense of an intermediate partial row of pyramids. The coarsening rate(s) of the surface obtained by our diffuse interface method may be quantitatively compared with the results in [45] in the future.

Our algorithm extends naturally to the case of the diffuse interface version of the bending energy regularization considered in [9]. Comparisons of the evolution and steady-states for strong surface energy anisotropies using the different regularizations will be presented in a forthcoming work. Initial simulations indicate that the time scales for the two regularizations are very different with the Willmore-type regularization evolving on a significantly longer time scale.

In the future, we also plan to pursue the development of a gradient stable, unconditionally solvable method for the strongly anisotropic, regularized Cahn–Hilliard equation. This involves posing a discrete energy and splitting it into the sum of convex and concave pieces. One may then formulate a discrete gradient method that is conservative, unconditionally solvable, unconditionally stable, and unconditionally discrete energy decreasing [49]. However, deriving and splitting a consistent discrete energy for the continuous, anisotropic energy (20) is non-trivial. This current work has been a first step towards an unconditionally stable and solvable method. Furthermore, we will expand the scope of the physical model to include elasticity and other physics to further study the coarsening of quantum dots in epitaxially strained thin film systems [8]. We also plan to further examine the coarsening of surfaces evolving by strongly anisotropic surface diffusion and deposition, and evolving by adsorption/desorption kinetics [28,29,9].

Acknowledgements

The authors thank Axel Voigt and Jean Taylor for illuminating and enjoyable discussions, and Sorin Miran for making his BEARCLAW package available. SMW thanks Stephen Watson for pointing out several important references on the coarsening of faceted surfaces. The authors also gratefully acknowledge partial support from the University of Minnesota Office of Sponsored Projects, the National Science Foundation Division of Mathematical Sciences (DMS-0612878) and the Division of Materials Research (DMR-0606356 and DMR-0502737), and the European Community through grant STRP-016447.

Appendix. Convexity of the gradient energy density and Ill-posedness

Here, as in Taylor and Cahn [31], we consider a more general form of the energy functional:

$$\mathcal{E}[c] = \int_{\Omega} \left(F(c) + \frac{\epsilon^2}{2} G^2(\nabla c) \right) dx, \quad (93)$$

where $\Omega \subset \mathbb{R}^d$ and where the assumptions on G are that it is bounded from below, *i.e.*, there exists a constant $G_0 \in \mathbb{R}$ such that $G(\mathbf{p}) \geq G_0$, for all $\mathbf{p} \in \mathbb{R}^d$; and it is positively one-homogeneous, *i.e.*, $G(r\mathbf{n}) = rG(\mathbf{n})$ for all $r > 0$ and all unit vectors $\mathbf{n} \in \mathbb{R}^d$. We will use the notation $\mathbf{p} = \nabla c$ herein, with $\mathbf{n} = \mathbf{p}/|\mathbf{p}|$, and will assume that $|\mathbf{p}| \neq 0$. As shown in [31], the variational derivative is

$$\frac{\delta \mathcal{E}}{\delta c} = F'(c) - \frac{\epsilon^2}{2} \mathbf{H}_{\mathbf{p}}(G^2(\mathbf{p})) : \mathbf{H}_{\mathbf{x}}(c(\mathbf{x})), \quad (94)$$

where $\mathbf{H}_{\mathbf{p}}$ and $\mathbf{H}_{\mathbf{x}}$ are d -dimensional Hessian operators. The Cahn–Hilliard gradient flow is

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(M(c, \mathbf{p}) \nabla \left(\frac{\delta \mathcal{E}}{\delta c} \right) \right), \quad (95)$$

where $M(c, \mathbf{p}) \geq 0$, which is ill-posed if $G^2(\mathbf{p})$ is not a convex function with respect to the gradient variable \mathbf{p} . This does not preclude the possibility that for some function spaces the gradient flow problem will be well-posed even if $G^2(\mathbf{p})$ is not convex; and, indeed, we will construct such a problem.

In the following calculations, we will focus on the planar case ($d = 2$), using the specific gradient energy given by $G(\mathbf{p}) = \gamma(\theta)|\mathbf{p}|$, where γ is a smooth, periodic function of θ , with $\tan(\theta) = p_2/p_1$. We will show that the criteria for ill-posedness is the same as that for the 2D sharp interface laws for anisotropic motion by mean curvature and surface diffusion. These are ill-posed if the surface stiffness $\gamma(\theta) + \gamma''(\theta)$ changes sign [23,24,21].

Wheeler has essentially shown this in [62] by considering the characteristic equation for $\delta\mathcal{E}/\delta c = 0$. We will work directly from the perspective of convexity. Note that the 3D case is expected to be much more complicated [20].

Proposition 7.1. *The function $G(\mathbf{p}) := \gamma(\theta)|\mathbf{p}|$ is non-convex if for some angles $\gamma(\theta) + \gamma''(\theta) < 0$, and convex otherwise.*

Proof. We will show that the Hessian matrix is not positive semi-definite if there exist angles for which $\gamma(\theta) + \gamma''(\theta) < 0$, and is positive semi-definite otherwise. One finds that

$$\frac{\partial\theta}{\partial p_1} = -\frac{n_2}{|\mathbf{p}|}, \quad \frac{\partial\theta}{\partial p_2} = \frac{n_1}{|\mathbf{p}|}, \quad \frac{\partial n_i}{\partial p_j} = \frac{1}{|\mathbf{p}|}(\delta_{i,j} - n_i n_j), \tag{96}$$

where $\delta_{i,j}$ is the Krönecker delta, and

$$\frac{\partial G}{\partial p_1} = \gamma'(\theta)(-n_2) + \gamma(\theta)n_1, \quad \frac{\partial G}{\partial p_2} = \gamma'(\theta)n_1 + \gamma(\theta)n_2. \tag{97}$$

The Hessian matrix of G is

$$\mathbf{H}(G) = \frac{\gamma''(\theta) + \gamma(\theta)}{|\mathbf{p}|} \begin{bmatrix} n_2^2 & -n_1 n_2 \\ -n_1 n_2 & n_1^2 \end{bmatrix}. \tag{98}$$

Its eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = (\gamma''(\theta) + \gamma(\theta))/|\mathbf{p}|$, with the associated eigenvectors

$$\mathbf{w}_1 = \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad \text{and} \quad \mathbf{w}_2 = \mathbf{t} = \begin{bmatrix} -n_2 \\ n_1 \end{bmatrix}. \tag{99}$$

If there exist angles for which $\gamma(\theta) + \gamma''(\theta) < 0$, then the eigenvalues are not everywhere non-negative, and the Hessian is not always positive semi-definite. In this case G is not convex. Otherwise, the eigenvalues are everywhere non-negative and the Hessian matrix is always positive semi-definite, and G is convex. \square

Proposition 7.2. *Assuming that $G \geq 0$, if $\mathbf{H}(G)$ is at least positive semi-definite (that is, positive semi-definite or strictly positive definite) then $\mathbf{H}(G^2)$ is at least positive semi-definite.*

Proof. The Hessian of the function G^2 satisfies

$$\frac{1}{2}\mathbf{H}(G^2) = G\mathbf{H}(G) + \nabla_{\mathbf{p}}G \otimes \nabla_{\mathbf{p}}G. \tag{100}$$

Let $\mathbf{p} = (p_1, p_2)^T$ and $\mathbf{q} = (q_1, q_2)^T$ be any non-zero (gradient) vectors, then

$$\frac{1}{2}\mathbf{q}^T\mathbf{H}(G^2(\mathbf{p}))\mathbf{q} = G(\mathbf{p})\mathbf{q}^T\mathbf{H}(G(\mathbf{p}))\mathbf{q} + (\mathbf{q} \cdot \nabla_{\mathbf{p}}G(\mathbf{p}))^2 \geq 0. \quad \square \tag{101}$$

Remark 7.3. The last proposition holds for any $G > 0$. Thus if $\gamma(\theta) > 0$ the proposition holds for the special form $G(\mathbf{p}) := \gamma(\theta)|\mathbf{p}|$. Note that if $|a| < 1$ then $\gamma(\theta) > 0$ using the m -fold anisotropy function $\gamma(\theta) := 1 + a\cos(m\theta)$.

Proposition 7.4. *Take $G(\mathbf{p}) := \gamma(\theta)|\mathbf{p}|$ and assume $\gamma(\theta) > 0$ for all angles. Then if for some interval (θ_1, θ_2) $\gamma''(\theta) + \gamma(\theta) < 0$ for all $\theta \in (\theta_1, \theta_2)$ —which by Proposition 7.1 means G is not convex—and if $\gamma'(\theta_0) = 0$ for some $\theta_0 \in (\theta_1, \theta_2)$, then $G^2(\mathbf{p})$ is not convex.*

Proof. We will show that $\mathbf{H}(G^2)$ is not positive semi-definite, i.e., that there exist (gradient) vectors \mathbf{q} and \mathbf{p} such that

$$\mathbf{q}^T \mathbf{H}(G^2(\mathbf{p})) \mathbf{q} < 0. \quad (102)$$

We choose \mathbf{p} such that $\theta(\mathbf{p}) = \theta_0 \in (\theta_1, \theta_2)$ and then $\mathbf{q} = \mathbf{t} = (-n_2, n_1)^T$, where \mathbf{t} is a unit tangent vector relative to \mathbf{p} . Note that

$$\mathbf{t} \cdot \nabla_{\mathbf{p}} G(\mathbf{p}) = \begin{bmatrix} -n_2 \\ n_1 \end{bmatrix} \cdot \begin{bmatrix} -n_2 \gamma'(\theta) + n_1 \gamma(\theta) \\ n_1 \gamma'(\theta) + n_2 \gamma(\theta) \end{bmatrix} = \gamma'(\theta). \quad (103)$$

Thus

$$\begin{aligned} \frac{1}{2} \mathbf{t}^T \mathbf{H}(G^2(\mathbf{p})) \mathbf{t} &= G(\mathbf{p}) \mathbf{t}^T \mathbf{H}(G(\mathbf{p})) \mathbf{t} + (\mathbf{t} \cdot \nabla_{\mathbf{p}} G(\mathbf{p}))^2 = G(\mathbf{p}) \lambda_2(\mathbf{p}) + (\gamma'(\theta_0))^2 \\ &= \gamma(\theta_0)(\gamma''(\theta_0) + \gamma(\theta_0)) + (\gamma'(\theta_0))^2 < 0. \quad \square \end{aligned} \quad (104)$$

Remark 7.5. If $a > (m^2 - 1)^{-1}$ then the conditions of the last proposition on γ are satisfied by the anisotropy function $\gamma(\theta) := 1 + a \cos(m\theta)$.

Interestingly, the Cahn–Hilliard equation (95) is well-posed for 1D functions, regardless of whether $G^2(\mathbf{p})$ is convex or not.

Proposition 7.6. *If the initial data $c(\mathbf{x}, t = 0)$ are 1D, then (95) reduces to the usual 1D Cahn–Hilliard equation, but with the interfacial parameter $\epsilon^2 \gamma^2(\theta)$ in place of ϵ^2 , and the data remain 1D.*

Proof. Consider $s : \mathbb{R}^2 \rightarrow \mathbb{R}$ defined by $s(\mathbf{x}) = \mathbf{s} \cdot \mathbf{x}$, where \mathbf{s} is an arbitrary unit vector. The function s gives the signed distance from the 1D subspace \mathbf{s}^\perp , and $\nabla s = \mathbf{s}$. Now construct the 1D order parameter

$$c(\mathbf{x}) = C(s(\mathbf{x})). \quad (105)$$

Then $\mathbf{p} = C'(s)\mathbf{s}$, $\mathbf{n} = \mathbf{s}$, $|\mathbf{p}| = C'(s)$, and

$$\mathbf{H}_{\mathbf{x}}(c) = C''(s)\mathbf{s} \otimes \mathbf{s}. \quad (106)$$

Note that

$$\mathbf{n} \cdot \nabla_{\mathbf{p}} G(\mathbf{p}) = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \cdot \begin{bmatrix} -n_2 \gamma'(\theta) + n_1 \gamma(\theta) \\ n_1 \gamma'(\theta) + n_2 \gamma(\theta) \end{bmatrix} = \gamma(\theta). \quad (107)$$

Hence

$$\begin{aligned} \frac{\epsilon^2}{2} \mathbf{H}_{\mathbf{p}}(G^2(\mathbf{p})) : \mathbf{H}_{\mathbf{x}}(c) &= \frac{\epsilon^2}{2} C''(s) (\mathbf{H}_{\mathbf{p}}(G^2(\mathbf{p})) : \mathbf{s} \otimes \mathbf{s}) = \frac{\epsilon^2}{2} C''(s) (\mathbf{s}^T \mathbf{H}_{\mathbf{p}}(G^2(\mathbf{p})) \mathbf{s}) \\ &= \epsilon^2 C''(s) [G(\mathbf{p}) \mathbf{n}^T \mathbf{H}_{\mathbf{p}}(G(\mathbf{p})) \mathbf{n} + (\mathbf{n} \cdot \nabla_{\mathbf{p}} G(\mathbf{p}))^2] = \epsilon^2 \gamma^2(\theta) C''(s). \quad \square \end{aligned} \quad (108)$$

References

- [1] J. Cahn, On spinodal decomposition, *Acta Metall.* 9 (1961) 795–801.
- [2] J. Cahn, J. Hilliard, Free energy of a non-uniform system—I: Interfacial free energy, *J. Chem. Phys.* 28 (1958) 258–267.
- [3] H. Jou, P. Leo, J. Lowengrub, Microstructural evolution in inhomogeneous elastic media, *J. Comput. Phys.* 131 (1997) 109–148.
- [4] H. Garcke, M. Rumpf, U. Weikard, The Cahn–Hilliard equation with elasticity: dinite element approximation and qualitative studies, *Interfaces Free Bound.* 3 (2001) 101–118.
- [5] S. Hu, L. Chen, A phase field-model for evolving microstructures with strong elastic inhomogeneity, *Acta. Mater.* 49 (2001) 1879–1890.
- [6] H. Garcke, U. Weikard, Numerical approximation of the Cahn–Larché equation, *Numer. Math.* 100 (2005) 639–662.
- [7] J. Eggleston, P. Voorhees, Ordered growth of nanocrystals via a morphological instability, *Appl. Phys. Lett.* 80 (2002) 306–308.
- [8] S. Wise, J. Lowengrub, J. Kim, K. Thornton, P. Voorhees, W. Johnson, Quantum dot formation on a strain-patterned epitaxial thin film, *Appl. Phys. Lett.* 87 (2005) 133102.
- [9] A. Rätz, A. Ribalta, A. Voigt, Surface evolution of elastically stressed films under deposition by a diffuse interface model, *J. Comput. Phys.* 214 (2006) 187–208.
- [10] J. Lowengrub, L. Truskinovsky, Quasi-incompressible Cahn–Hilliard fluids and topological transitions, *Proc. Roy. Soc. London A* 454 (1998) 2617–2654.

- [11] V. Badalassi, H. Cenicerros, S. Banerjee, Computation of multiphase systems with phase field models, *J. Comput. Phys.* 190 (2003) 371–397.
- [12] J. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, *J. Comput. Phys.* 193 (2004) 511–543.
- [13] J. Barrett, R. Nürnberg, V. Styles, Finite element approximation of a phase field model for void electromigration, *SIAM J. Numer. Anal.* 42 (2004) 738–772.
- [14] H. Yu, W. Lu, Ordering of nanovoids in an anisotropic solid driven by surface misfit, *J. Comput. Theor. Nano.* 2 (2005) 256–262.
- [15] J. Barrett, H. Garcke, R. Nürnberg, Finite element approximation of a phase field model for surface diffusion of voids in a stressed solid, *Math. Comput.* 75 (2006) 7–41.
- [16] H.B. Frieboes, J.S. Lowengrub, S. Wise, X. Zheng, P. Macklin, E.L. Bearer, V. Cristini, Computer simulation of glioma growth and morphology, *Neuroimage* (2007), doi:10.1016/j.neuroimage.2007.03.008.
- [17] F. Campelo, A. Hernández-Machado, Dynamic model and stationary shapes of fluid vesicles, *Euro. Phys. J. E* 20 (2006) 37–45.
- [18] L. Chen, Phase-field models for microstructure evolution, *Annu. Rev. Mater. Res.* 32 (2002) 113–140.
- [19] D. Anderson, G. McFadden, A. Wheeler, Diffuse-interface methods in fluid mechanics, *Ann. Rev. Fluid Mech.* 30 (1998) 139.
- [20] R. Sekerka, Analytical criteria for missing orientations on three-dimensional equilibrium shapes, *J. Cryst. Growth* 275 (2005) 77–82.
- [21] M. Burger, F. Haußer, C. Stöcker, A. Voigt, A level set approach to anisotropic flows with curvature regularization, *J. Comput. Phys.* (2007), doi:10.1016/j.jcp.2006.11.026.
- [22] C. Herring, Some theorems on the free energies of crystal surfaces, *Phys. Rev.* 82 (1951) 87–93.
- [23] A. DiCarlo, M. Gurtin, P. Podio-Guidugli, A regularized equation for anisotropic motion-by-curvature, *SIAM J. Appl. Math.* 52 (1992) 1111–1119.
- [24] M. Gurtin, M. Jabbour, Interface evolution in three dimensions with curvature-dependent energy and surface diffusion: interface-controlled evolution, phase transitions, epitaxial growth of elastic films, *Arch. Ration. Mech. Anal.* 163 (2002) 171–208.
- [25] B. Spencer, Asymptotic solutions for the equilibrium crystal shape with small corner energy regularization, *Phys. Rev. E* 69 (2004) 011603.
- [26] M. Siegel, M. Miksis, P. Voorhees, Evolution of material voids for highly anisotropic surface energy, *J. Mech. Phys. Solids* 52 (2004) 1319–1353.
- [27] F. Haußer, A. Voigt, A discrete scheme for regularized anisotropic surface diffusion: a 6th order geometric evolution equation, *Interfaces Free Bound.* 7 (2005) 353–369.
- [28] F. Haußer, A. Voigt, Facet formation and coarsening modeled by a geometric evolution law for epitaxial growth, *J. Cryst. Growth* 275 (2005) e47–e51.
- [29] F. Haußer, A. Voigt, A numerical scheme for regularized anisotropic curve shortening flow, *Appl. Math. Lett.* 19 (2006) 691–698.
- [30] M. Burger, Numerical simulation of anisotropic surface diffusion with curvature-dependent energy, *J. Comput. Phys.* 203 (2005) 602–625.
- [31] J. Taylor, J. Cahn, Diffuse interfaces with sharp corners and facets: phase field models with strongly anisotropic surfaces, *Physica D* 112 (1998) 381–411.
- [32] J. Eggleston, G. McFadden, P. Voorhees, A phase-field model for highly anisotropic interfacial energy, *Physica D* 150 (2001) 91–103.
- [33] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, *J. Comput. Phys.* 198 (2004) 450–468.
- [34] Q. Du, C. Liu, R. Ryhman, X. Wang, A phase field formulation of the Willmore problem, *Nonlinearity* 18 (2005) 1249–1267.
- [35] Q. Du, C. Liu, X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions, *J. Comput. Phys.* 212 (2006) 757–777.
- [36] A. Wheeler, Phase-field theory of edges in an anisotropic crystal, *Proc. R. Soc. A* 462 (2006) 3363–3384.
- [37] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512.
- [38] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [39] P. Colella, D.T. Graves, T.J. Ligocki, D.F. Martin, D. Modiano, D.B. Serafini, B.V. Straalen, CHOMBO software package for amr applications: design document, Technical report, Lawrence Berkeley National Laboratory, Applied Numerical Algorithms Group; NERSC Division; Berkeley, CA, USA, 2003.
- [40] D. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, *J. Comput. Phys.* 163 (2000) 271–312.
- [41] S. Mitran, BEARCLAW a code for multiphysics applications with embedded boundaries: user’s manual, Technical report, Department of Mathematics, University of North Carolina. <<http://www.amath.unc.edu/Faculty/mitran/bearclaw.html>> 2006.
- [42] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (1977) 333–390.
- [43] U. Trottenberg, C. Oosterlee, A. Schüller, *Multigrid*, Academic Press, London, UK, 2001.
- [44] D. Kay, R. Welford, A multigrid finite element solver for the Cahn–Hilliard equation, *J. Comput. Phys.* 212 (2006) 288–304.
- [45] D. Moldovan, L. Golubovic, Interfacial coarsening dynamics in epitaxial growth with slope selection, *Phys. Rev. E* 61 (2000) 6190.
- [46] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, *Physica D* 63 (1993) 410–423.
- [47] R. Kobayashi, Y. Giga, Equations with singular diffusivity, *J. Stat. Phys.* 95 (1999) 1187–1220.
- [48] J. Cahn, C. Elliott, A. Novick-Cohen, The Cahn–Hilliard equation with a concentration dependent mobility: motion by minus the laplacian of the mean curvature, *Eur. J. Appl. Math.* 7 (1997) 287.
- [49] D. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, in: J. Bullard, R. Kalia, M. Stoneham, L.-Q. Chen (Eds.), *Computational and Mathematical Models of Microstructural Evolution*, The Materials Research Society, 1998.
- [50] D. Furihata, A stable and conservative finite difference scheme for the Cahn–Hilliard equation, *Numer. Math.* 87 (2001) 675–699.
- [51] M. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, *IEEE Trans. Syst. Man Cybern.* 21 (1991) 1278–1286.

- [52] J. Brown, L. Lowe, Multigrid elliptic equation solver with adaptive mesh refinement, *J. Comput. Phys.* 209 (2005) 582–598.
- [53] W. Burton, N. Cabrera, F. Frank, The growth of crystals and the equilibrium structure of their surfaces, *Trans. R. Soc. Lond. A* 243 (1951) 299–358.
- [54] W. Mullins, Proof that the two dimensional shape of minimum surface free energy is convex, *J. Math. Phys.* 3 (1962) 754–759.
- [55] S. Watson, F. Otto, B. Rubinstein, S. Davis, Coarsening dynamics of the convective Cahn–Hilliard equation, *Physica D* 178 (2003) 127–148.
- [56] S. Watson, Crystal growth, coarsening and the convective Cahn–Hilliard equation, in: P. Colli, C. Verdi, A. Visintin (Eds.), *Free Boundary Problems: Theory and Applications*, International Series of Numerical Mathematics, vol. 147, Birkhauser, 2003, pp. 329–342.
- [57] S. Watson, S. Norris, Scaling theory and porphometrics for a coarsening multiscale surface, via a principle of maximum dissipation, *Phys. Rev. Lett.* 96 (2006) 176103.
- [58] M. Siegert, Coarsening dynamics of crystalline thin films, *Phys. Rev. Lett.* 81 (1998) 5481–5484.
- [59] A. Golovin, S. Davis, A. Nepomnyashchy, A convective Cahn–Hilliard model for the formation of facets and corners in crystal growth, *Physica D* 122 (1998) 202–230.
- [60] A. Golovin, S. Davis, A. Nepomnyashchy, Model for faceting in a kinetically controlled crystal growth, *Phys. Rev. E* 59 (1999) 803–825.
- [61] T. Savina, A. Golovin, S. Davis, A. Nepomnyashchy, P. Voorhees, Faceting of a growing crystal surface by surface diffusion, *Phys. Rev. E* 67 (2003) 021606.
- [62] A. Wheeler, Cahn–Hoffman ξ -vector and its relation to diffuse interface models of phase transitions, *J. Stat. Phys.* 95 (1999) 1245–1280.