Journal of Computational Physics 315 (2016) 52–64

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Improving the accuracy of convexity splitting methods for gradient flow equations

Karl Glasner, Saulo Orizaga [*]

Department of Mathematics, University of Arizona, United States

ARTICLE INFO

Article history:
Received 22 May 2015
Received in revised form 11 February 2016
Accepted 19 March 2016
Available online 24 March 2016

Keywords:
Gradient flows
Cahn–Hilliard
Phase-field crystal
Spectral methods
Eyre splitting

ABSTRACT

This paper introduces numerical time discretization methods which significantly improve the accuracy of the convexity-splitting approach of Eyre (1998) [7], while retaining the same numerical cost and stability properties.

A first order method is constructed by iteration of a semi-implicit method based upon decomposing the energy into convex and concave parts. A second order method is also presented based on backwards differentiation formulas. Several extrapolation procedures for iteration initialization are proposed. We show that, under broad circumstances, these methods have an energy decreasing property, leading to good numerical stability.

The new schemes are tested using two evolution equations commonly used in materials science: the Cahn–Hilliard equation and the phase field crystal equation. We find that our methods can increase accuracy by many orders of magnitude in comparison to the original convexity-splitting algorithm. In addition, the optimal methods require little or no iteration, making their computation cost similar to the original algorithm.

1. Introduction

High order parabolic equations which derive from gradient descent of an energy functional ("gradient flows") are commonly encountered in the study of material microstructure evolution and related phase field models [1,2]. Two extensively studied models in this class are the Cahn–Hilliard (CH) equation [3]

$$u_t = \Delta(-\epsilon^2 \Delta u + u^3 - u), \tag{1}$$

and phase field crystal (PFC) equation [4]

$$u_t = \Delta((\Delta + 1)^2 u - ru + u^3). \tag{2}$$

Numerical methods with explicit time discretization are impractical for these equations because of limitations imposed by numerical stability. This has motivated the development of a variety of implicit and semi-implicit numerical methods [5–11]. A simple and elegant semi-implicit approach was formulated by Eyre [7]. Although it was originally introduced for the Cahn–Hilliard equation, it has been successfully implemented in many different contexts [11–14]. Various improvements and extensions of the method have also been made [15–17].

[*] Corresponding author.
E-mail addresses: kglasner@math.arizona.edu (K. Glasner), sorizaga@math.arizona.edu (S. Orizaga).

http://dx.doi.org/10.1016/j.jcp.2016.03.042
0021-9991/© 2016 Elsevier Inc. All rights reserved.

In the algorithm's most basic form, an energy function $F(u)$ is decomposed into convex and concave parts as $F = F_+ + F_-$. For a gradient flow of the form $u_t = -\nabla F(u)$, the method treats the convex part implicitly, and the concave part explicitly, leading to the formally first order method

$$\frac{u_{n+1} - u_n}{h} = -\nabla F_+(u_{n+1}) - \nabla F_-(u_n), \tag{3}$$

where u_n is an approximation to $u(nh)$ and h is the timestep. For this scheme, it can be shown that the energy decreasing property holds in a discrete way: $F(u_{n+1}) \leq F(u_n)$. An algorithm with this property is said to be *energetically stable*. Under some circumstances, this notion of stability coincides with the usual idea of unconditional numerical stability.

The main advantage to the convexity splitting method is that ∇F_+ can often be chosen to be linear, and moreover $I + h\nabla F_+$ is often easy to invert. This means that (3) can be accomplished very efficiently, without the need for Newton-iteration steps and various sorts of linear algebra typical of fully implicit methods. In addition, practical implementations in phase field modeling appears to produce qualitatively correct features, such as the development and propagation of interfaces. On the other hand, it has been widely observed that the convexity-splitting scheme can have large temporal errors [9,18,19]. *The purpose of this paper is to improve the accuracy of the convexity splitting approach while retaining all of its other benefits.*

Our new methods fall into a broad class of semi-implicit algorithms for high order parabolic equations [9,20–22]. There have been several recent advances for these types of methods in the context of materials science. The approach proposed by Hu et al. [22] used finite differences and a multi-grid approach to solve the PFC equation. Gomez and Hughes [9] introduced a second order accurate in time variational method for the CH equation, using mixed finite elements in space. Second order accurate algorithms have also been used for the PFC equation by Vignal et al. [21]. The work in [21] presented a computational framework that relies on the convex-concave splitting approach. Christlieb et al. [20] suggested that high accuracy can be achieved using fully implicit methods with a pseudo-spectral spatial discretization, employing Newton iterations and a conjugate gradient solver.

In this paper, we utilize the convexity splitting of the free energy as a basis for new algorithms, based upon various combinations of iteration, extrapolation, and higher order discretization. In section 2, we review the variational formulation of abstract gradient flow equations and explain stability of convexity splitting methods in this context. Iterative and higher order algorithms are introduced in section 3, which are shown to be energetically stable under broad circumstances. These methods are tested in section 4 for the Cahn–Hilliard and phase field crystal equations.

2. Generalized gradient flows and their variational characterization

Often gradient flows arise as steepest descent of a functional in a general function space whose geometry is given by an inner product. Suppose S is affine to a Hilbert space H, i.e.

$$S = \{u | u = u_0 + w,\ w \in H\}.$$

Let $\langle \cdot, \cdot \rangle$ be some inner product on that space (we remark that this does not have to correspond to the "natural" inner product of H). Given a smooth functional $F(u) : S \to \mathbb{R}$, the *gradient flow* of F with respect to S is a solution $u(t) \in S \times [0, \infty)$ of the weak equation

$$\langle u_t, w \rangle = -(\delta F(u), w), \quad w \in H. \tag{4}$$

The right hand side denotes the directional derivative

$$(\delta F(u), \varphi) = \lim_{\delta \to 0} \frac{F(u + \delta\varphi) - F(u)}{\delta},$$

which is assumed to always exist in what follows.

There is a variational principle associated with (4) which may be exploited. It is easy to show that (4) arises from minimization of a "Rayleigh" functional

$$R(w) = \frac{1}{2} \langle w, w \rangle + (\delta F(u), w). \tag{5}$$

That is, if a unique critical point w_c of R exists at each time t, then $u_t = w_c$. Informally, this means that a gradient flow seeks the direction of greatest energy decrease, subject to an energy dissipation penalty prescribed by the inner product.

In the context of PDEs, H is typically a Sobolev space such as $H^s(\Omega)$ endowed with an inner product from a weaker space (e.g. $H^q(\Omega)$ with $q < s$). In the case $S = L^2(\Omega)$ is equipped with its usual inner product, equation (4) is just

$$\int_\Omega u_t\, w\, dx = -\int_\Omega \delta F(u)\, w\, dx, \quad w \in L^2(\Omega), \tag{6}$$

which leads equations of the form $u_t = -\delta F(u)$. In the materials science literature, this gradient flow is conventionally called "non-conserved dynamics".

Another common situation is where $H = H_0^1(\Omega)$, where the inner product employed is that of the dual space H^{-1}. For example, for the periodic domain $\Omega = ([0, 2\pi]^d)$, this inner product may be written in terms of the Fourier coefficients $\hat{w}(k)$, $\hat{v}(k)$ as

$$\langle v, w\rangle = \sum_{k\neq 0} \frac{1}{|k|^2}\hat{v}(k)\hat{w}^*(k),$$

which in terms of the inverse Laplacian is

$$\langle v, w\rangle = -\int_\Omega v\phi\, dx, \quad \Delta\phi = w.$$

In this case equation (4) is a weak version of the higher order equation $u_t = \Delta(\delta F(u))$. Conventionally this gradient flow is called "conserved dynamics" since (with suitable boundary conditions) the mean value of u is always equal to u_0.

2.1. Variational algorithms for generalized gradient flows

The variational principle (5) can be used as a basis to construct time discretization methods. The simplest idea is to replace the time derivative with its forward difference, which leads to

$$R_{BE}(v) = \frac{1}{2h}\langle v - u_n, v - u_n\rangle + F(v). \tag{7}$$

Critical points of this functional solve

$$\left\langle \frac{u_{n+1} - u_n}{h}, w\right\rangle = -(\delta F(u_{n+1}), w), \quad w \in H. \tag{8}$$

Notice this is just the standard backward Euler (**BE**) method. Of course, the functional (7) may not have a unique critical point; this is indeed one of the drawbacks to fully-implicit discretizations. On the other hand, if u_{n+1} is a minimizer of R_{BE}, then clearly $F(u_{n+1}) \le R_{BE}(u_{n+1}) \le F(u_n)$. In this case, the method is energetically stable.

The convexity splitting algorithm (3) can be generalized as

$$\left\langle \frac{u_{n+1} - u_n}{h}, w\right\rangle = -(\delta F_+(u_{n+1}), w) - (\delta F_-(u_n), w), \quad w \in H. \tag{\textbf{CS}}$$

It is easy to see formally that this method is the Euler–Lagrange equation for a "discrete" Rayleigh functional

$$R_{CS}(v) = \frac{1}{2h}\langle v - u_n, v - u_n\rangle + F_+(v) + (\delta F_-(u_n), v - u_n) + F_-(u_n). \tag{9}$$

This functional is necessarily convex, since the first two terms are convex and the last two are just the linearization of F_- around u_n. We suppose that F_+, F_- are chosen so that the following property holds.

$$\text{For } u_n \in S, \text{ the functional } R_{CS} \text{ has a minimizer in } S. \tag{10}$$

For most parabolic equations of interest, this property is easily verified using classical results of the calculus of variations. We can show that under these conditions, the time discretization method (**CS**) produces a unique answer, and the method is an energetically stable algorithm.

Proposition 1. *Suppose that (10) holds. Given $u_n \in S$, there exists a unique solution $u_{n+1} \in S$ satisfying (**CS**) and $F(u_{n+1}) \le F(u_n)$.*

Proof. Since (10) provides a minimizer u_{n+1} of (9), usual arguments in the calculus of variations show that (**CS**) also holds. Uniqueness of critical points satisfying (**CS**) follows from the fact that R_{CS} is convex.

Since F_- is concave, its graph sits below the hyperplane tangent to u_n, which is represented by the last two terms in (9). It follows that

$$\begin{aligned} F(u_{n+1}) &= F_+(u_{n+1}) + F_-(u_{n+1}) \\ &\le F_+(u_{n+1}) + (\delta F_-(u_n), u_{n+1} - u_n) + F_-(u_n) \le R_{CS}(u_{n+1}). \end{aligned}$$

On the other hand, since u_{n+1} minimizes R_{CS}, then $R_{CS}(u_{n+1}) \le R_{CS}(u_n) = F(u_n)$. □

The relation between convexity splitting and backward Euler methods is more transparent by comparing the corresponding discrete Rayleigh functionals. In particular, (9) simply replaces the concave part F_- in (7) with its linearization at u_n.

Unless u_n and u_{n+1} are close, this can be a rather crude approximation; this suggests that the differences between **CS** and **BE** methods can be significant. Although both methods are first order, it has been noted empirically that the **CS** method performs substantially worse in terms of accuracy [20]. On the other hand, implementation of the **BE** scheme uses linear and nonlinear solvers which tend to be inefficient.

3. Iterative and higher order convexity splitting methods

It is now shown how to improve the accuracy of the convexity splitting approach, while retaining its simplicity, efficiency and stability properties. We propose the following iterative convexity splitting schemes

$$\begin{gathered} u_{n+1} := u_K^*, \\ \left\langle \frac{u_j^* - u_n}{h}, w \right\rangle = -(\delta F_+(u_j^*), w) - (\delta F_-(u_{j-1}^*), w), \quad w \in H, \quad j = 1, 2, \dots, K, \end{gathered} \tag{$\mathbf{ICS}_K$}$$

where the initial iterate u_0^* will be prescribed later by various extrapolation methods. In less abstract terms, for the specific cases of (non-conserved and conserved) L^2 and H^{-1} gradient flows, the method can be written as

$$u_j^* + h[\delta F_+(u_j^*)] = u_n - h[\delta F_-(u_{j-1}^*)], \quad L^2 \text{ gradient flow} \tag{11}$$

$$u_j^* - h\Delta[\delta F_+(u_j^*)] = u_n + h\Delta[\delta F_-(u_{j-1}^*)], \quad H^{-1} \text{ gradient flow} \tag{12}$$

for $j = 1, 2, \dots, K$.

With the simplest possible initial iterate $u_0^* = u_n$, the **ICS**$_1$ method is the same as the original *CS* scheme. One key to the success of the new methods is a choice of a better initial guess using some kind of extrapolation. A natural choice is simple linear extrapolation,

$$u_0^* = 2u_n - u_{n-1}. \tag{$\mathbf{LINX}$}$$

Of course this requires that two sequential steps be known initially; we explain how this can be accomplished later. Other extrapolation methods based upon the structure of the particular equation are detailed below, and in some cases provide better stability and accuracy.

3.1. Higher order backward difference formula methods

It is natural to replace the first order finite difference approximation of the time derivative in (**ICS**$_K$) with a higher order one. This can be done using standard backward differentiation formulas, giving a new method

$$\begin{gathered} u_{n+1} := u_K^*, \\ \left\langle \frac{\frac{3}{2}u_j^* - 2u_n + \frac{1}{2}u_{n-1}}{h}, w \right\rangle = -(\delta F_+(u_j^*), w) - (\delta F_-(u_{j-1}^*), w), \quad w \in H, \quad j = 1, 2, \dots, K. \end{gathered} \tag{$\mathbf{BDCS}_K$}$$

For the specific inner products discussed above, the method reads

$$\frac{3}{2}u_j^* + h[\delta F_+(u_j^*)] = 2u_n - \frac{1}{2}u_{n-1} - h[\delta F_-(u_{j-1}^*)], \quad L^2 \text{ gradient flow} \tag{13}$$

$$\frac{3}{2}u_j^* - h\Delta[\delta F_+(u_j^*)] = 2u_n - \frac{1}{2}u_{n-1} + h\Delta[\delta F_-(u_{j-1}^*)], \quad H^{-1} \text{ gradient flow} \tag{14}$$

Of course, it is again necessary to specify an initial guess u_0^* for the iteration. In general, this should be done using higher order extrapolation to maintain the same order of accuracy. Formulas are given for test problems below.

3.2. Relationship to fully implicit methods

The methods (**ICS**$_K$) and (**BDCS**$_K$) are related to the fully implicit first and second order backwards difference methods as follows. Suppose that, for either method, as $K \to \infty$, u_j^* converges to u_∞^*. We note that this convergence is not guaranteed, or even a desirable property. Nonetheless, for the first order case it is easy to see that u_∞^* is a critical point of (7). In other words, $u_{n+1} = u_\infty^*$ is a solution of the **BE** method. For the second order method, $u_{n+1} = u_\infty^*$ formally solves

$$\left\langle \frac{\frac{3}{2}u_{n+1} - 2u_n + \frac{1}{2}u_{n-1}}{h}, w \right\rangle = -(\delta F(u^{n+1}), w), \quad w \in H. \tag{15}$$

This is in fact the standard fully implicit second order BDF method.

It should be clearly stated, however, that the goal here is to approximate the actual evolution equation and not simply reproduce the results of a fully implicit scheme. We find below that a small number of iterations with small timestep is often more efficient than many iterations with a larger timestep. In fact, in some cases choosing $K = 1$ is best. Note that with linear extrapolation (for example), the iteration free first order method is really just a predictor–corrector type of algorithm reading

$$\left\langle \frac{u_{n+1} - u_n}{h}, w \right\rangle = -(\delta F_+(u_{n+1}), w) - (\delta F_-(2u_n - u_{n-1}), w), \quad w \in H. \tag{16}$$

3.3. Energetic stability

The iteration steps in (**ICS**$_K$) and (**BDCS**$_K$) are themselves variational in nature, and can in fact by viewed as a relaxation of Rayleigh-type functionals. The first order method, for example, can be seen as attempting to minimize the functional R_{BE} by equation (7). By contrast, the second order method will be seen as a relaxation of

$$R_{BDF}(v) = \frac{1}{3h}\left\langle \frac{3}{2}v - 2u_n + \frac{1}{2}u_{n-1}, \frac{3}{2}v - 2u_n + \frac{1}{2}u_{n-1} \right\rangle + F(v). \tag{17}$$

This characterization of the proposed methods can be used to show that like the original method (3), they can also be energetically stable. We make this idea precise with the following proposition.

Proposition 2. *Suppose that (10) holds. Given u^*_{j-1}, there exists a unique solution u^*_j satisfying either (**ICS**$_K$) or (**BDCS**$_K$). If the initial iteration is chosen so that*

$$R_{BE}(u^*_0) \le F(u_n), \quad R_{BDF}(u^*_0) \le F(u_n), \tag{18}$$

*for (**ICS**$_K$) and (**BDCS**$_K$), respectively, then $F(u_{n+1}) \le F(u_n)$.*

Proof. Note that (**ICS**$_K$) is the Euler–Lagrange equation for

$$R_j(v) = \frac{1}{2h}\langle v - u_n, v - u_n\rangle + F_+(v) + (\delta F_-(u^*_{j-1}), v - u^*_{j-1}) + F_-(u^*_{j-1}). \tag{19}$$

Since R_j is a functional of the form (9), property (10) and convexity guarantee existence and uniqueness of a minimizer which also solves (**ICS**$_K$). Similarly, the iteration u^*_j of (**BDCS**$_K$) are unique minimizers of

$$R_{2j}(v) = \frac{1}{3h}\left\langle \frac{3}{2}v - 2u_n + \frac{1}{2}u_{n-1}, \frac{3}{2}v - 2u_n + \frac{1}{2}u_{n-1} \right\rangle + F_+(v) + (\delta F_-(u^*_{j-1}), v - u^*_{j-1}) + F_-(u^*_{j-1}), \tag{20}$$

which are unique solutions of (**BDCS**$_K$).

By virtue of concavity of F_-, we have $R_{BE}(u^*_j) \le R_j(u^*_j)$, and since u^*_j minimizes R_j, it follows that

$$R_{BE}(u^*_j) \le R_j(u^*_j) \le R_j(u^*_{j-1}) = R_{BE}(u^*_{j-1}).$$

Then it follows from induction that

$$F(u_{n+1}) \le R_{BE}(u_{n+1}) \le \ldots \le R_{BE}(u^*_0) \le F(u_n).$$

The proof for energetic stability of (**BDCS**$_K$) is identical, with R_{2j} replacing R_j. □

Condition (18) entirely depends on the extrapolation procedure used to find u^*_0. Note that (18) can always be satisfied since the trivial choice $u^*_0 = u_n$ gives $R_{BE}(u^*_0) = R_{BDF}(u^*_0) = F(u_n)$. In practice, we find polynomial extrapolations also generally work, so long as the time step h is not too large. Potentially, the condition (18) could be utilized as a heuristic criterion for dynamically adapting the time step h, although we do not pursue this idea here.

3.4. Initialization of multi-step extrapolation

A multi-step extrapolation formula such as (**LINX**) requires accurate knowledge of sequential steps u_n and u_{n-1} to produce a usefully accurate prediction. It is assumed that to begin with, however, only initial data u_0 is prescribed. A simple way of obtaining an accurate next step is to simply use very small timesteps h/N_{small}, where $N_{small} \gg 1$. Then after N_{small} steps of some prescribed one-step method (for example **CS** or **BE**), the solution should be some accurate approximation of u_1. Thereafter, a full sized timestep h can be used.

Unfortunately, even though this process is executed only once, it still can be numerically expensive. We have found a more efficient way to obtain good initial steps based on a repeated doubling of the timestep, which is initially chosen to be very small. Let $u_{n+1} = I(u_{n-1}, u_n; h)$ denote the result of the first order method (with some given K) using linear extrapolation based on the sequential steps u_{n-1} and u_n, and let w_k be an approximation to the solution at $t = h/2^k$. Then we find w_k recursively as

$$w_{k-1} = I(u_0, w_k; h/2^k), \quad k = kmax, kmax - 1, \dots, 1.$$

This process is initialized by setting $w_{kmax} = u_0$ for some maximum number of step doublings *kmax*. At the end of recursion, u_1 is set equal to w_0. In general this seems to provide an accurate enough starting point for the numerical tests described below. Generalization of this idea to higher order multistep methods is straightforward.

4. Numerical implementation and performance

We now describe how the new methods can be utilized in problems of practical interest. Numerical experiments are conducted for the evolution equations (1) and (2). Various combinations of iteration and timestep parameters are investigated to assess accuracy and stability.

The computational time is largely proportional to the total number of iterations of the basic first order (**ICS**$_K$) or second order (**BDCS**$_K$) semi-implicit schemes. In practice, the computational cost per iteration is essentially identical among these schemes since the implicit part is solved within the framework of a Fourier pseudo-spectral discretization. From a perspective of numerical efficiency, the goal is therefore to minimize the total number of iterations for a given level of accuracy.

For the test problems below, our domain is a box $[0, L]^2$. Periodic boundary conditions are assumed, which allows for a standard Fourier pseudo-spectral discretization [23]. For $L = 2\pi$, this approximation takes the form

$$u \approx \sum_{k_x=1}^{N} \sum_{k_y=1}^{N} \hat{u}(k_x, k_y, t) \exp\left[2\pi i\Big(\omega(k_x)x + \omega(k_y)y\Big)\right],$$

where $\omega(k_x)$, $\omega(k_y)$ are the correctly aliased wavenumbers. The discrete Fourier transform is used to compute $\hat{u}$, and can be written as a linear map $\hat{u} = \mathcal{F}u$. The Laplacian is therefore discretized as

$$\Delta u \approx \mathcal{F}^{-1}\Lambda\mathcal{F}u$$

where Λ is just the diagonal multiplication operator $\Lambda\hat{u} = -[\omega(k_x)^2 + \omega(k_y)^2]\hat{u}$. Note that any rational function of the Laplacian can be similarly computed by spectral mapping, for example inverses like

$$(I + h\Delta)^{-1}u = \mathcal{F}^{-1}(1 + h\Lambda)^{-1}\mathcal{F}u.$$

The time discretization methods described are implemented for the exact functional and inner product, after which the spatial approximations are applied. In some cases, it is possible to show that the energy decreasing property is retained for a spatially discrete version of the energy functional (e.g. [11,19]). We also note that spectral methods are known to be exponentially accurate, so it is unlikely that the spatial discretization used here will have a large effect on the overall variational properties of our methods.

4.1. Cahn–Hilliard equation

Cahn and Hilliard [3] introduced a well-studied model (1) for phase separation of two materials. This equation can be cast as a generalized gradient flow in H^{-1} of the free energy

$$F(u) = \int_\Omega \frac{\epsilon^2}{2}|\nabla u|^2 + u^4/4 - u^2/2\, dx. \tag{21}$$

For $\epsilon \ll 1$, random initial data will evolve to create domains where $u \approx \pm 1$, which are separated by diffuse interfaces having the form

$$u \approx \tanh(\sqrt{2}z/\epsilon), \tag{22}$$

where z is a coordinate transverse to the interface.

The nonlinear term can be split into convex and concave parts as

$$\frac{u^4}{4} - \frac{u^2}{2} = \frac{au^2}{2} + \left(\frac{u^4}{4} - [1 + a]\frac{u^2}{2}\right) \equiv f_+(u) + f_-(u),$$

where $a > 0$ is an adjustable parameter. Clearly $f_+''(u) > 0$, and $f_-''(u) < 0$ provided $a > 2$ and $-1 < u < 1$. We remark that the fourth order equation lacks a maximum principle to guarantee the bound on u, but it seems to hold nonetheless. In

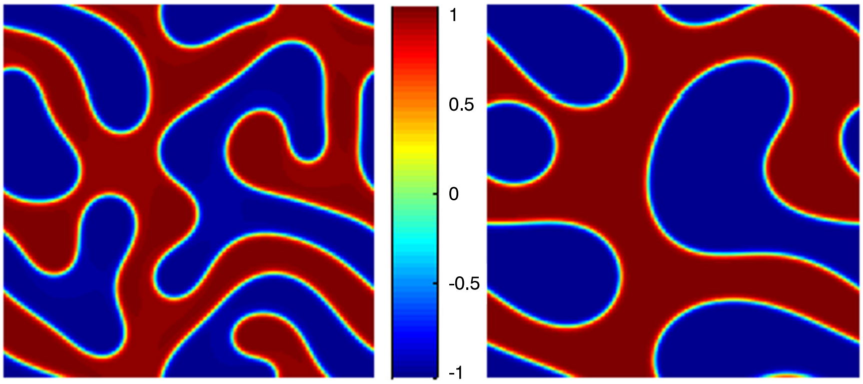

Fig. 1. Simulation of two-dimensional spinodal decomposition was used as our test problem. Left: Snapshot at time $t = 0$ with well-developed phase segregated domains. Right: Configuration at the final time $t = 3.2$.

practice, there is a trade-off between choosing a large for improved stability and choosing a smaller which tends to yield smaller errors. For our tests, we have determined empirically that choosing $a = 3/2$ provides a good trade-off between these two aims. The **ICS**$_K$ method reads in this case

$$\frac{u_j^* - u_n}{h} = (-\epsilon^2\Delta^2 + a\Delta)u_j^* + \Delta\Big[(u_{j-1}^*)^3 - (1+a)u_{j-1}^*\Big].$$

A reasonable initial condition was generated by first evolving random noise to a state where there were well-formed diffuse interfaces. The simulation was performed on a 256 × 256 grid for the spatial domain $\Omega = [0,\pi]^2$ with $\epsilon = 0.02$. Fig. 1 shows the snapshots of the 2D spinodal decomposition simulation of a binary mixture with the initial configuration (left) and the evolved configuration at a fixed final time. An accurate comparison solution u_{ex} was obtained by using a very small step h, and compared at time $t_f = 3.2$ to simulations for various h and K by computing the L^1 norm

$$Error = \int_\Omega |u - u_{ex}|(\cdot, t_f)dx$$

Two extrapolation procedures were investigated: linear extrapolation (**LINX**) and a nonlinear version obtained by applying linear extrapolation to a transformed variable ψ:

$$u_0^* = \alpha\tanh(2\psi_n - \psi_{n-1}), \quad \psi_n = \tanh^{-1}(u_n/\alpha). \qquad (\textbf{NLX})$$

The motivation for this extrapolation formula comes from the structure of the internal layers given by (22). The variable ψ is roughly proportional to z, which is the signed distance to the diffuse interface. For interfaces which are flat and moving at constant velocity V, it follows that $\psi \sim z - Vt$. In this case, linear extrapolation of ψ would be exact, whereas linear extrapolation of u would not. Since there is no guarantee that $|u_n| < 1$, it is necessary to choose α slightly bigger than one so that u_n/α is in the domain of the inverse hyperbolic tangent. The errors tend to increase if α is chosen larger, however. Choosing $\alpha = 1.01$ provides good results in our tests.

Our numerical experiments confirm the predicted energy-decreasing property of the new methods (Fig. 2). The free energy (21) was computed with a standard trapezoid rule and finite differences. The computation employs the particular parameters $h = 0.01$ and $K = 2$ for each of the methods **ICS–LINX**, **ICS–NLX**, **BDCS–QEX** and **BDCS–NQLX**. We find similar results for essentially all the test cases represented in Figs. 3–5, except where instability was observed in the second-order methods. Of course, the conclusion of Proposition 2 is not valid if the extrapolation energy $F(u_0^*)$ is higher than $F(u_n)$, which might occur for extremely large timesteps. Indeed, numerical experiments for timesteps higher than those in Fig. 3 do show a non-monotone evolution of the energy, as well as impractically large errors.

Errors are reported in Fig. 3, as a function of an effective timestep h/K, which is ideally as large as possible for a given level of accuracy. By comparison, the original **CS** method is dramatically worse, with errors roughly 10^3 larger than the method with $K = 1$. On the other hand, the numerical effort of the two methods is nearly identical, except for the computationally cheap extrapolation calculation.

We find that the nonlinear extrapolation (**NLX**) performs slightly better than linear extrapolation. Iteration does improve the overall accuracy of the method, but not enough to justify the computational effort. In all cases it was found that simply reducing the timestep is more efficient to maintain accuracy. As a consequence, the optimal first order method is not iterative at all; it is really a predictor–corrector method like (16).

We also implemented the second order method using the same test problem. The **BDCS**$_K$ method for the Cahn–Hilliard equation amounts to solving

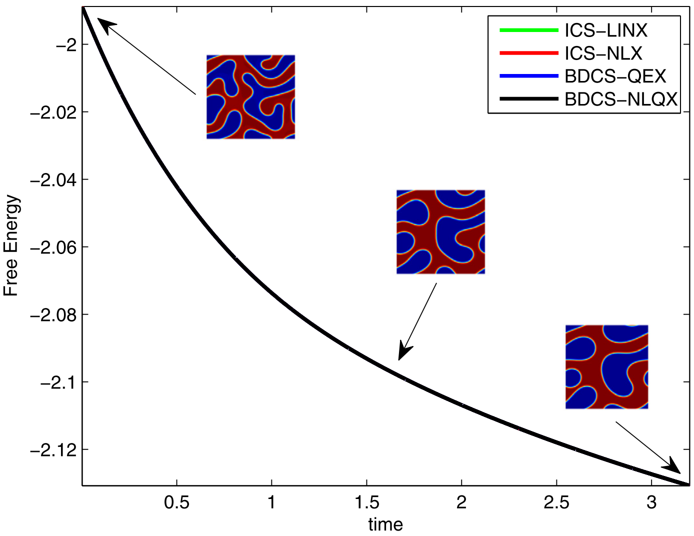

Fig. 2. Time evolution of the free energy functional for the Cahn–Hilliard equation. We have appended to the energy curve snapshots of the solution to illustrate the dynamical process associated with energy decay.

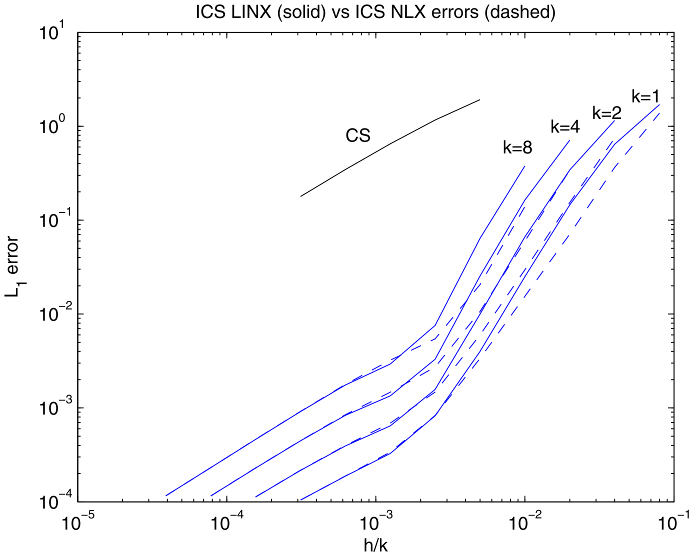

Fig. 3. Accuracy measured as the L^1-error at time $t = 3.2$, versus the effective timestep h/K. Solid (blue) lines indicate errors using linear extrapolation **LINX** and dashed lines correspond to nonlinear extrapolation **NLX**. For comparison, errors for the original **CS** scheme are also reported. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\frac{\frac{3}{2}u_j^* - 2u_n + \frac{1}{2}u_{n-1}}{h} = (-\epsilon^2\Delta^2 + a\Delta)u_j^* + \Delta\left[(u_{j-1}^*)^3 - (1+a)u_{j-1}^*\right].$$

If second order accuracy is to be obtained, then the extrapolation step for u_0^* also needs to be higher order. Two versions based on quadratic polynomial extrapolation were investigated, one linear in u and one nonlinear:

$$u_0^* = 3u_n - 3u_{n-1} + u_{n-2} \tag{QEX}$$

$$u_0^* = \alpha\tanh(3\psi_n - 3\psi_{n-1} + \psi_{n-2}), \quad \psi_n = \tanh^{-1}(u_n/\alpha). \tag{NLQX}$$

Figs. 4 and 5 report numerical errors for the second order scheme **BDCS–QEX** and **BDCS–NLQX** respectively. It can be seen that the scheme **BDCS–QEX** using $k > 1$ is more accurate than **ICS–LINX** provided the effective timestep is approximately less than 10^{-3}. For $K > 2$ and h/K approximately less than $10^{-2.4}$, the **BDCS–NLQX** scheme exceeds the accuracy and performance of the **ICS–LINX** scheme.

For large enough timesteps, the second order method exhibited a mild amount of instability unless iteration was used. This explains the dramatic increase in error in Figs. 4 and 5. In general, instability was easy to control by simply increasing

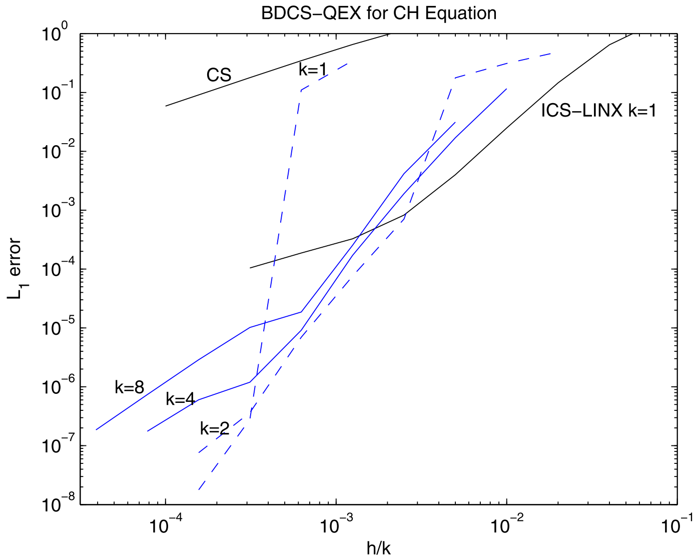

Fig. 4. Accuracy measured as the L^1-error at time $t = 3.2$, versus the effective timestep h/K. Blue lines indicate errors using linear quadratic extrapolation **BDCS–QEX** and the lower solid black line corresponds to the optimal first order method. The upper black line is the error for the original first order CS scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

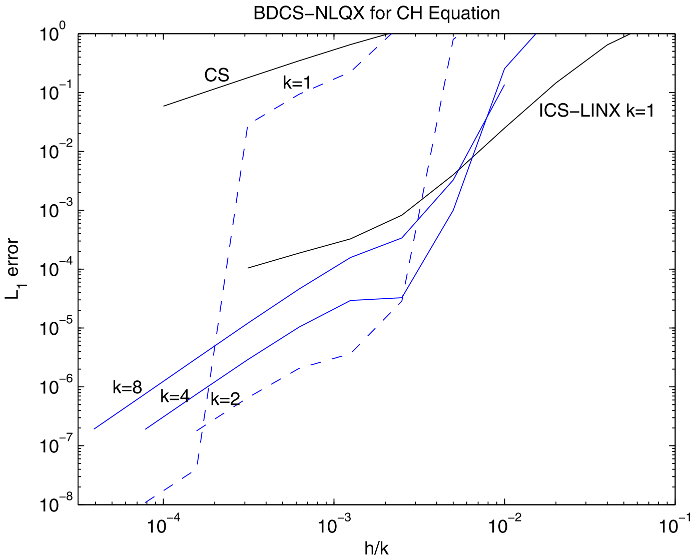

Fig. 5. Accuracy measured as the L^1-error at time $t = 3.2$, versus the effective timestep h/K. Blue lines indicate errors using nonlinear quadratic extrapolation **BDCS–NLQX** and the lower solid black line corresponds to the optimal first order method. The upper black line is the error for the CS scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the number of iterations K by a small amount. We found that nonlinear extrapolation generally had better stability properties than linear extrapolation. For example, with $k = 8$ the method (**QEX**) was unstable when $h > 2.4 \times 10^{-2}$, whereas (**NLQX**) was stable for $h < 8 \times 10^{-2}$.

4.2. Phase field crystal equation

The phase field crystal (PFC) equation (2) was introduced by Elder and Grant [4] as a continuum model of crystalline phase ordering which incorporates descriptions of lattice defects. This equation also represents a gradient flow in H^{-1}, where the free energy is of higher spatial order:

$$F(u) = \int \frac{1}{4}u^4 + \frac{1-r}{2}u^2 - |\nabla u|^2 + \frac{1}{2}(\Delta u)^2 dx.$$

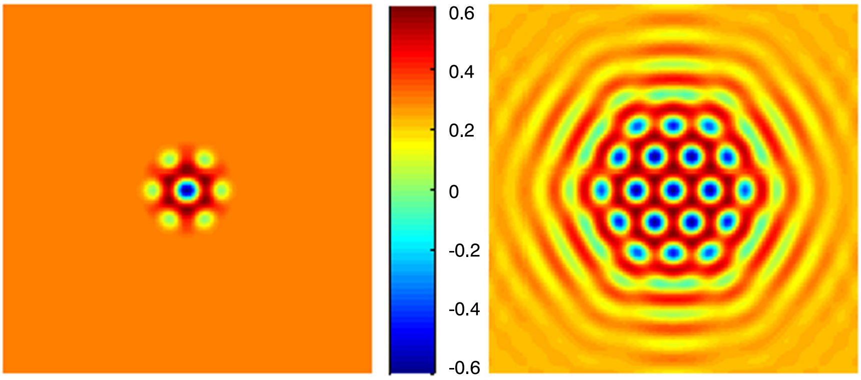

Fig. 6. The simulation of the two-dimensional growth of a crystal in a supercooled liquid is used as our test problem. Snapshots of the numerical simulations for the PFC equation are given. Left: $t = 0$ is the initial seed configuration. Right: $t = 12.8$ is the final time for crystal growth.

For large enough r, there is a pattern forming instability which gives either stripes or hexagonal lattices depending on the conserved spatial average of u.

In this case convexity splitting can be achieved using

$$F_+(u) = \int \frac{1-r+a}{2}u^2 + \frac{1}{2}(\Delta u)^2 dx, \quad F_-(u) = \int \frac{1}{4}u^4 - \frac{a}{2}u^2 - |\nabla u|^2 dx,$$

where if $a > 0$ is large enough then F_- is concave. The term $u^4/4 - au^2/2$ will be concave provided $a > 3||u^2||_\infty$. As with the Cahn–Hilliard equation, there is no maximum principle to bound u. On the other hand, in practice the solution does not get too large and a fixed value for a is sufficient. As before, if the parameter a is too small, stability will be lost and if it is too large, high truncation errors will be observed. When $r = 1/2$ we find that the choice $a = 2$ provides a good compromise.

The corresponding (**ICS**$_K$) method reads

$$\frac{u_j^* - u_n}{h} = \left[(1-r+a)\Delta + \Delta^3\right]u_j^* + \Delta\left[(u_{j-1}^*)^3 - au_{j-1}^* + 2\Delta u_{j-1}^*\right], \tag{23}$$

which was implemented together with the linear first order extrapolation formula (**LINX**). The (**BDCS**$_K$) method in this case is

$$\frac{\frac{3}{2}u_j^* - 2u_n + \frac{1}{2}u_{n-1}}{h} = \left[(1-r+a)\Delta + \Delta^3\right]u_j^* + \Delta\left[(u_{j-1}^*)^3 - au_{j-1}^* + 2\Delta u_{j-1}^*\right], \tag{24}$$

which was used in conjunction with quadratic extrapolation (**QEX**).

For our test problem, we consider the two-dimensional growth of a crystal in a supercooled liquid as it was done in [21] and [4]. The initial seed crystal is represented mathematically with the following initial condition

$$u_o(\mathbf{x}) = \bar{u} + \omega(\mathbf{x})(Au_s(\mathbf{x}))$$

where $u_s(\mathbf{x}) = \cos(qy/\sqrt(3))\cos(qx) - 1/2\cos(2qy/\sqrt(3))$, $\omega(\mathbf{x}) = (1 - (||\mathbf{x} - \mathbf{x_o}||/d_o)^2)^2$ if $||\mathbf{x} - \mathbf{x_o}|| \leq d_o$, 0 otherwise. Here $\mathbf{x_o}$ is the center of the domain, d_o is 1/6 of the domain length in the x-direction. The other parameters were $\epsilon = 0.325$, $\bar{u} = \sqrt{\epsilon}/2$, $A = 4/5(\bar{u} + \sqrt{15\epsilon - 36\bar{u}^2}/3)$ and $q = \sqrt{3}/2$. The simulations for the PFC equation were performed on a 256×256 grid for the spatial domain $\Omega = [0, 40\pi/\sqrt{3}]^2$ with the $r = 1/2$. Similar initial configurations for the PFC equation are found in [24] and [11].

In Fig. 6, we present snapshots of the simulations for the solid crystallite that is initially placed in the center of the liquid domain (left) and also for the evolved crystal at $t = 12.8$ (right). Fig. 7 analyzes the time evolution of the PFC energy functional for the particular values $h = 0.01$ and $K = 1$, although the other tests represented in Figs. 8–9 give nearly the same curves. The results for both the **ICS–LINX** and **BDCS–QEX** methods roughly coincide. The errors for the **ICS–LINX** and **BDCS–QEX** schemes are reported in Figs. 8 and 9, respectively. The experiment shows that **ICS–LINX** exceeds the accuracy of the original **CS** scheme by factors of ≈ 50 for the optimal $K = 1$ scheme. The **BDCS–QEX** scheme is found to be stable for solving the PFC equation regardless of the timestep size or number of iterations implemented, at least over the range of parameters tested. In general, it appears that the second order method is almost always better than the first order method, in the sense of maximizing the effective timestep for a given level of accuracy.

5. Conclusion

This paper presented novel numerical methods for gradient flow evolution equations which have good accuracy properties. The real advantage of these methods is that they retain the efficiency and ease of implementation found in the original

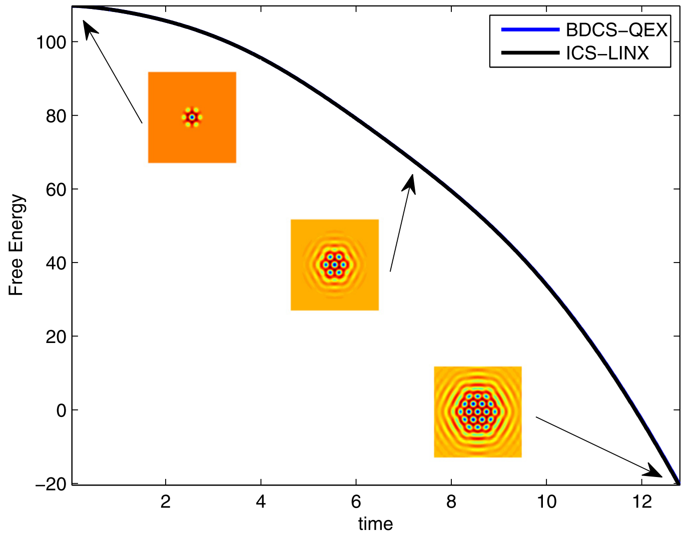

Fig. 7. Evolution of the free energy for the phase field crystal equation. The results for the first and second order methods nearly coincide.

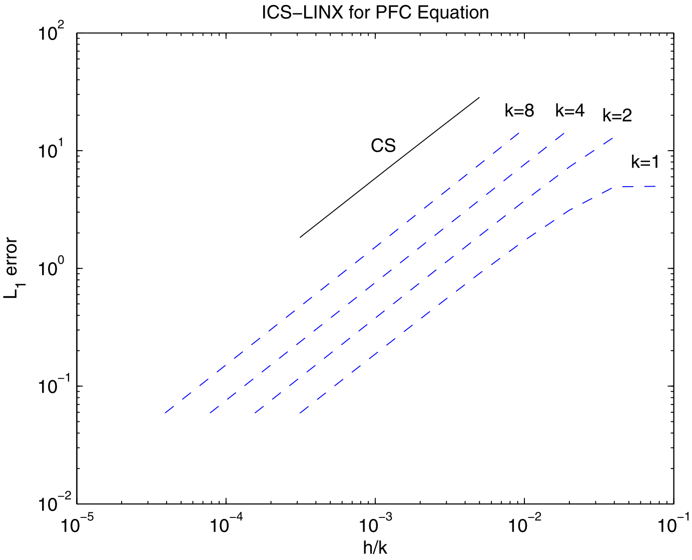

Fig. 8. Accuracy measured as the L^1-error at time $t = 12.8$, versus the effective timestep h/K. Dashed (blue) lines indicate errors for the **ICS–LINX** method and the solid black line corresponds to the original convexity splitting **CS** scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

convexity splitting approach [7], while providing significantly better accuracy. In general, we have found that considerable iteration of our schemes is not necessary, and is less efficient than simply choosing small step sizes. For problems with high accuracy requirements, the second order BDCS method appears to be the best choice of those considered. On the other hand, the stability properties of this method may or may not render it less efficient than the first order method when accuracy requirements are only modest.

We do not have a good assessment of the performance of these new methods in comparison to other semi-implicit [8] or fully implicit [20] approaches for similar problems. Such a comparison would have to control for differences in computational implementation as well as consider the effect of user-selected tolerance parameters (in, for example, iterative linear algebra algorithms). This might make a definitive comparison difficult to obtain.

The robustness and simplicity of our proposed methods suggest that they may be adapted to a wide variety of evolution problems with a variational structure. This includes the vast array of energy-driven models arising in materials science and condensed matter physics [2], as well as optimization and image processing applications (e.g. [25]).

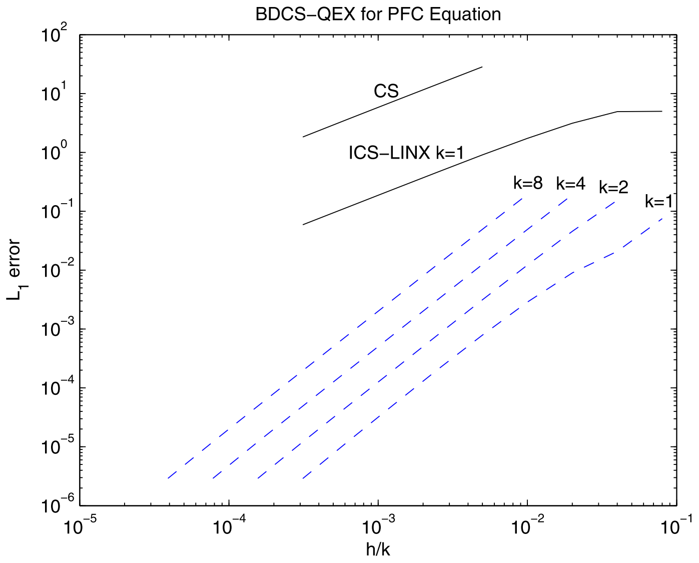

Fig. 9. Accuracy measured as the L^1-error at time $t = 12.8$, versus the effective timestep h/K. Dashed (blue) lines indicate errors for the **BDCS–QEX** method and the lower solid black line corresponds to optimal first order scheme. The upper black line is the error for the CS scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgements

The authors thank the reviewers, who provided valuable comments resulting in improvements in this paper. SO was supported through a NSF-Alliance Postdoctoral award DMS-0946431. KG was supported through NSF award DMS-1514689.

References

[1] Samuel M. Allen, John W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (6) (1979) 1085–1095.

[2] Long-Qing Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Res. 32 (1) (2002) 113–140.

[3] John W. Cahn, John E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28 (2) (1958) 258–267.

[4] K.R. Elder, Martin Grant, Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals, Phys. Rev. E 70 (5) (2004) 051605.

[5] Uri M. Ascher, Steven J. Ruuth, Brian T.R. Wetton, Implicit–explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.

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

[7] David J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, in: MRS Proceedings, vol. 529, Cambridge Univ. Press, 1998, p. 39.

[8] Yinnian He, Yunxian Liu, Tao Tang, On large time-stepping methods for the Cahn–Hilliard equation, Appl. Numer. Math. 57 (5) (2007) 616–628.

[9] Hector Gomez, Thomas J.R. Hughes, Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models, J. Comput. Phys. 230 (13) (2011) 5310–5327.

[10] Mowei Cheng, James A. Warren, An efficient algorithm for solving the phase field crystal model, J. Comput. Phys. 227 (12) (2008) 6241–6248.

[11] Steven M. Wise, Cheng Wang, John S. Lowengrub, An energy-stable and convergent finite-difference scheme for the phase field crystal equation, SIAM J. Numer. Anal. 47 (3) (2009) 2269–2288.

[12] K.B. Glasner, Grain boundary motion arising from the gradient flow of the Aviles-Giga functional, Physica D: Nonlinear Phenom. 215 (1) (2006) 80–98.

[13] John B. Greer, Andrea L. Bertozzi, Guillermo Sapiro, Fourth order partial differential equations on general geometries, J. Comput. Phys. 216 (1) (2006) 216–246.

[14] Vittorio Cristini, Xiangrong Li, John S. Lowengrub, Steven M. Wise, Nonlinear simulations of solid tumor growth using a mixture model: invasion and branching, J. Math. Biol. 58 (4–5) (2009) 723–763.

[15] Franck Boyer, Sebastian Minjeaud, Numerical schemes for a three component Cahn–Hilliard model, ESAIM: Math. Model. Numer. Anal. 45 (4) (2011) 697–738.

[16] Zhengru Zhang, Yuan Ma, Zhonghua Qiao, An adaptive time-stepping strategy for solving the phase field crystal model, J. Comput. Phys. 249 (2013) 204–215.

[17] Daozhi Han, Xiaoming Wang, A second order in time, uniquely solvable, unconditionally stable numerical scheme for Cahn–Hilliard–Navier–Stokes equation, J. Comput. Phys. 290 (2015) 139–156.

[18] Mowei Cheng, James A. Warren, Controlling the accuracy of unconditionally stable algorithms in the Cahn–Hilliard equation, Phys. Rev. E 75 (1) (2007) 017702.

[19] Andrew Christlieb, Keith Promislow, Zhengfu Xu, On the unconditionally gradient stable scheme for the Cahn–Hilliard equation and its implementation with Fourier method, Commun. Math. Sci. 11 (2) (2013) 345–360.

[20] Andrew Christlieb, Jaylan Jones, Keith Promislow, Brian Wetton, Mark Willoughby, High accuracy solutions to energy gradient flows from material science models, J. Comput. Phys. 257 (2014) 193–215.

[21] Philippe Vignal, Lisandro Dalcin, Donald L. Brown, Nathan Collier, Victor M. Calo, An energy-stable convex splitting for the phase-field crystal equation, preprint, arXiv:1405.3488, 2014.

[22] Zhengzheng Hu, Steven M. Wise, Cheng Wang, John S. Lowengrub, Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation, J. Comput. Phys. 228 (15) (2009) 5323–5339.

[23] Lloyd N. Trefethen, Spectral Methods in MATLAB, vol. 10, Siam, 2000.

[24] Hector Gomez, Xesús Nogueira, An unconditionally energy-stable method for the phase field crystal equation, Comput. Methods Appl. Mech. Eng. 249 (2012) 52–61.

[25] Tony F. Chan, Jianhong Jackie Shen, Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods, SIAM, 2005.