Physica A 388 (2009) 1791–1803

Contents lists available at ScienceDirect

Physica A

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

An unconditionally gradient stable numerical method for solving the Allen–Cahn equation

Jeong-Whan Choi, Hyun Geun Lee, Darae Jeong, Junseok Kim [*]

Department of Mathematics, Korea University, Seoul 136-701, Republic of Korea

ARTICLE INFO

Article history:
Received 27 October 2008
Received in revised form 23 December 2008
Available online 6 February 2009

Keywords:
Allen–Cahn equation
Nonlinear multigrid
Finite difference
Unconditionally gradient stable

ABSTRACT

We consider an unconditionally gradient stable scheme for solving the Allen–Cahn equation representing a model for anti-phase domain coarsening in a binary mixture. The continuous problem has a decreasing total energy. We show the same property for the corresponding discrete problem by using eigenvalues of the Hessian matrix of the energy functional. We also show the pointwise boundedness of the numerical solution for the Allen–Cahn equation. We describe various numerical experiments we performed to study properties of the Allen–Cahn equation.

1. Introduction

The Allen–Cahn (AC) equation [1] was originally introduced as a phenomenological model for anti-phase domain coarsening in a binary alloy. It has been applied to a wide range of problems such as phase transitions [2], image analysis [3,4], the motion by mean curvature flows [5], and crystal growth [6]. An efficient and accurate numerical solution of this equation is needed to understand its dynamics. We consider an unconditionally gradient stable algorithm for the AC equation:

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -M(F'(c(\mathbf{x},t)) - \epsilon^2 \Delta c(\mathbf{x},t)), \quad \mathbf{x} \in \Omega, 0 < t \le T, \tag{1}$$

where $\Omega \subset \mathbf{R}^d$ $(d = 1, 2, 3)$ is a domain. The quantity $c(\mathbf{x}, t)$ is defined to be the difference between the concentrations of the two mixtures' components. The coefficient M is a constant mobility. We take $M \equiv 1$ for convenience. The function $F(c)$ is the Helmholtz free-energy density for c. It has a double well form, i.e., $F(c) = 0.25(c^2 - 1)^2$ as in Ref. [7]. Fig. 1 shows the function $F(c)$.

The small constant ϵ is the gradient energy coefficient related to the interfacial energy. The boundary condition is

$$\frac{\partial c}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega, \tag{2}$$

where $\frac{\partial}{\partial \mathbf{n}}$ denotes the normal derivative on $\partial\Omega$. The physical meaning of the condition is that the total free energy of the mixture decreases in time. The AC equation arises from the Ginzburg–Landau free energy,

$$\mathcal{E}(c) := \int_\Omega \left(F(c) + \frac{\epsilon^2}{2} |\nabla c|^2 \right) d\mathbf{x}. \tag{3}$$

[*] Corresponding author. Tel.: +82 2 2 3290 3077.
E-mail address: cfdkim@korea.ac.kr (J. Kim).

0378-4371/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2009.01.026

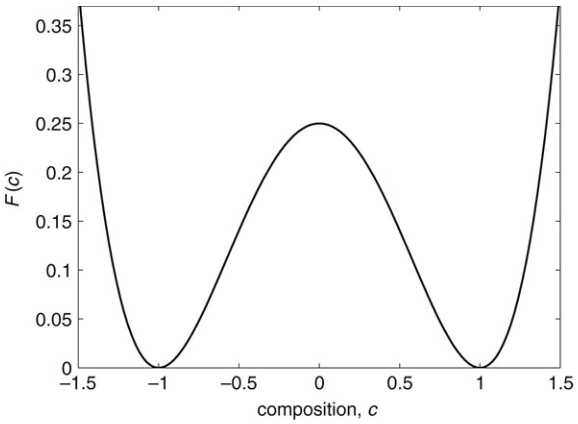


Fig. 1. A double well potential, $F(c) = 0.25(c^2 - 1)^2$.

The AC equation is the L^2-gradient flow of the total free energy $\mathcal{E}(c)$. We differentiate the energy $\mathcal{E}(c)$ to get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(c) &= \int_{\Omega}(F'(c)c_t + \epsilon^2\nabla c\cdot\nabla c_t)\mathrm{d}\mathbf{x}\\ &= \int_{\Omega}(F'(c) - \epsilon^2\Delta c)c_t\mathrm{d}\mathbf{x} = -\int_{\Omega}(c_t)^2\mathrm{d}\mathbf{x} \le 0,\end{aligned} \tag{4}$$

where we have used an integration by parts and the boundary condition (2). Therefore, the total energy is non-increasing in time; that is, the total energy is a Lyapunov functional for solutions of the AC equation. Numerical simulations of the AC equation, using explicit methods, impose severe time-step restrictions requiring the use of implicit type methods [8–10]. Ideally, we would like to use a stable integration algorithm allowing accuracy requirements rather than stability limitations to determine the integration step size. We use an unconditionally gradient stable scheme to solve the resulting discrete equations accurately and efficiently.

Eyre introduced a concept, "an unconditionally gradient stable scheme" in Refs. [8,9]. In Eyre's papers [8,9], he provided the idea and the theory of a scheme, but a little vaguely. We, here, clarify the proof of the scheme. We emphasize that, while the methods allow us to take arbitrarily large time steps, the accuracy of the numerical solution depends on choosing a small enough time step to resolve the dynamics [9].

This paper is organized as follows: In Section 2, we briefly review a derivation of the AC equation, based on gradient dynamics, with a physically motivated functional. In Section 3, we describe the unconditionally gradient stable discrete scheme and its properties such as the total energy decrease and the boundedness of the numerical solution. We present the numerical results in Section 4. In Section 5, we conclude. Appendix follows with details.

2. The Allen–Cahn equation

In this section, we review a derivation of the AC equation as a gradient flow [11,12]. It is natural to seek a law of evolution in the form

$$\frac{\partial c}{\partial t} = -\mathrm{grad}\mathcal{E}(c). \tag{5}$$

The symbol "grad" here denotes the gradient on the manifold in $L^2(\Omega)$ space. Let the domain of definition for the functional $\mathcal{E}$ be $\mathcal{D} = \{c \in H^2(\Omega) | \frac{\partial c}{\partial \mathbf{n}} = 0 \text{ on } \partial\Omega\}$. Let $c, v \in \mathcal{D}$. Then, we have

$$\begin{aligned}(\mathrm{grad}\mathcal{E}(c), v)_{L^2} &= \frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{E}(c+\theta v)\bigg|_{\theta=0} = \lim_{\theta\to 0}\frac{1}{\theta}\left(\mathcal{E}(c+\theta v) - \mathcal{E}(c)\right)\\ &= \int_{\Omega}\left(F'(c) - \epsilon^2\Delta c\right)v\mathrm{d}\mathbf{x} = (F'(c) - \epsilon^2\Delta c, v)_{L^2},\end{aligned}$$

where we have used an integration by parts and the boundary condition (2). We identify $\mathrm{grad}\mathcal{E}(c) \equiv F'(c) - \epsilon^2\Delta c$, then Eq. (5) becomes the AC equation [12].

3. Numerical analysis

In this section, we present an unconditionally gradient stable scheme for time discretization of the AC equation. In addition, we prove a discrete version of the total free energy dissipation for any time step size, which immediately implies the stability of the numerical solution. For simplicity of exposition, we shall discretize the AC equation in one-dimensional space, i.e., $\Omega = (a, b)$. Two and three-dimensional discretizations are defined analogously.

Let N be a positive even integer, $h = (b-a)/N$ be the uniform mesh size, and $\Omega_h = \{x_i = (i-0.5)h,\ 1 \le i \le N\}$ be the set of cell-centers. Let c_i^n be approximations of $c(x_i, n\Delta t)$, where $\Delta t = T/N_t$ is the time step, T is the final time, N_t is the total number of time steps, and $\mathbf{c}^n = (c_1^n, c_2^n, \ldots, c_N^n)$. We first implement the zero Neumann boundary condition, Eq. (2), by requiring that for each n,

$$\nabla_h c^n_{\frac{1}{2}} = \nabla_h c^n_{N+\frac{1}{2}} = 0, \tag{6}$$

where the discrete differentiation operator is $\nabla_h c^n_{i+\frac{1}{2}} = (c^n_{i+1} - c^n_i)/h$. We then define a discrete Laplacian by $\Delta_h c_i = (\nabla_h c_{i+\frac{1}{2}} - \nabla_h c_{i-\frac{1}{2}})/h$ and a discrete l_2 inner product by

$$\langle \mathbf{c}, \mathbf{d} \rangle_h = h \sum_{i=1}^{N} c_i d_i.$$

We also define the discrete norms as $\|\mathbf{c}\|_h^2 = \langle \mathbf{c}, \mathbf{c} \rangle_h$ and $\|\mathbf{c}\|_\infty = \max_{1\le i\le N} |c_i|$. For dissipative dynamics such as the AC equation, a discrete time stepping algorithm is defined to be *unconditionally gradient stable* if the discrete total free energy is nonincreasing for *any* size of a time step Δt.

Eyre's theorem [9] shows that an unconditionally gradient stable algorithm results for the AC equation if we can split the free energy appropriately into contractive and expansive parts,

$$\begin{aligned}\mathcal{E}(c) &= \int_a^b \left[F(c) + \frac{\epsilon^2}{2} c_x^2 \right] \mathrm{d}x \\ &= \int_a^b \left[\frac{c^4+1}{4} + \frac{\epsilon^2}{2} c_x^2 \right] \mathrm{d}x - \int_a^b \frac{c^2}{2} \mathrm{d}x = \mathcal{E}_c(c) - \mathcal{E}_e(c)\end{aligned} \tag{7}$$

and then treat the contractive part $\mathcal{E}_c(c)$ implicitly and the expansive part $-\mathcal{E}_e(c)$ explicitly. We use the nonlinearly stabilized splitting scheme [9] that involves a semi-implicit time and centered difference space discretizations of Eq. (1):

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -(c_i^{n+1})^3 + c_i^n + \epsilon^2 \Delta_h c_i^{n+1} \quad \text{for } i = 1, \ldots, N. \tag{8}$$

The boundary condition is $c_0 = c_1$ and $c_{N+1} = c_N$. This splitting is similar to the one employed in Ref. [13] for the time-depdendent Ginzberg–Landau (TDGL) equation, which treats the TDGL equation as a heat equation with a nonlinear source term resulting in an adaptive mesh refinement scheme which is second-order in space and time.

3.1. The unconditionally gradient stable scheme

The main purpose of this section is to show that the scheme in Eq. (8) inherits characteristic properties such as a decrease in the total energy corresponding to Eq. (4). To show the decrease in the discrete total energy, first, we define a discrete Lyapunov functional,

$$\mathcal{E}^h(\mathbf{c}^n) = \frac{h}{4} \sum_{i=1}^{N} ((c_i^n)^2 - 1)^2 + \frac{\epsilon^2 h}{2} \sum_{i=0}^{N} |\nabla_h c^n_{i+\frac{1}{2}}|^2 \tag{9}$$

for each n. It is convenient to decompose $\mathcal{E}^h(\mathbf{c}^n)$ into three parts:

$$\mathcal{E}^{(1)}(\mathbf{c}^n) = -\frac{h}{2} \sum_{i=1}^{N} (c_i^n)^2,$$

$$\mathcal{E}^{(2)}(\mathbf{c}^n) = \frac{\epsilon^2 h}{2} \sum_{i=0}^{N} \left| \nabla_h c^n_{i+\frac{1}{2}} \right|^2,$$

$$\mathcal{E}^{(3)}(\mathbf{c}^n) = \frac{h}{4} \sum_{i=1}^{N} \left((c_i^n)^4 + 1 \right).$$

We define a decomposition of $\mathcal{E}^h(\mathbf{c}^n)$ as $\mathcal{E}_c^h(\mathbf{c}^n) = \mathcal{E}^{(2)}(\mathbf{c}^n) + \mathcal{E}^{(3)}(\mathbf{c}^n)$ and $\mathcal{E}_e^h(\mathbf{c}^n) = -\mathcal{E}^{(1)}(\mathbf{c}^n)$, i.e., $\mathcal{E}^h(\mathbf{c}^n) = \mathcal{E}_c^h(\mathbf{c}^n) - \mathcal{E}_e^h(\mathbf{c}^n)$. We define grad_h the variational derivative with respect c_i^n, i.e.,

$$\mathrm{grad}_h \mathcal{E}^h(\mathbf{c}^n)_i = \frac{\delta \mathcal{E}^h(\mathbf{c}^n)}{\delta c_i^n} = (c_i^n)^3 - c_i^n - \epsilon^2 \Delta_h c_i^n. \tag{10}$$

We can rewrite the numerical scheme in Eq. (8) in terms of a gradient of the discrete total energy, i.e.,

$$\frac{c_i^{n+1}-c_i^n}{\Delta t} = -\mathrm{grad}_h\mathcal{E}_c^h(\mathbf{c}^{n+1})_i + \mathrm{grad}_h\mathcal{E}_e^h(\mathbf{c}^n)_i, \qquad \text{for } i = 1,\ldots,N. \tag{11}$$

The Hessian of $\mathcal{E}^{(i)}(\mathbf{c})$, denoted by $\mathbf{H}^{(i)}$, is the Jacobian of the $\mathrm{grad}_h\mathcal{E}^{(i)}(\mathbf{c})$ and is thus given for $i = 1, 2, 3$ by

$$\begin{aligned}
\left\{\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \mathbf{H}^{(3)}\right\} &= \left\{J\mathrm{grad}_h\mathcal{E}^{(1)}(\mathbf{c}), J\mathrm{grad}_h\mathcal{E}^{(2)}(\mathbf{c}), J\mathrm{grad}_h\mathcal{E}^{(3)}(\mathbf{c})\right\} \\
&= \left\{ -\begin{pmatrix} 1 & & & & 0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ 0 & & & & 1 \end{pmatrix}, \frac{\epsilon^2}{h^2}\begin{pmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{pmatrix}, 3\begin{pmatrix} c_1^2 & & & & 0 \\ & c_2^2 & & & \\ & & \ddots & & \\ & & & c_{N-1}^2 & \\ 0 & & & & c_N^2 \end{pmatrix} \right\},
\end{aligned}$$

where we have used the boundary condition in Eq. (6). The eigenvalues of $\mathbf{H}^{(1)}$, $\mathbf{H}^{(2)}$, and $\mathbf{H}^{(3)}$ are

$$\lambda_k^{(1)} = -1, \qquad \lambda_k^{(2)} = \frac{4\epsilon^2}{h^2}\sin^2\frac{(k-1)\pi}{2N}, \qquad \lambda_k^{(3)} = 3c_k^2 \quad \text{where } k = 1, 2, \ldots, N. \tag{12}$$

Note that $\lambda_k^{(1)}$ is negative and that $\lambda_k^{(2)}$ and $\lambda_k^{(3)}$ are non-negative. Let $\mathbf{v}_k$ for $k = 1,\ldots,N$ be the orthonormal eigenvector of $\mathbf{H}^{(2)}$ corresponding to the eigenvalue $\lambda_k^{(2)}$ and let $\mathbf{v}_{k,l} = \cos((k-1)\pi(2l-1)/(2N))$ for $l = 1,\ldots,N$ denote the lth component of $\mathbf{v}_k$. We can take the orthonormal eigenvector of $\mathbf{H}^{(1)}$ corresponding to the eigenvalue $\lambda_k^{(1)}$, the same as $\mathbf{v}_k$. Let $\lambda_1^e, \lambda_2^e, \ldots, \lambda_N^e$ be eigenvalues of $J(\mathrm{grad}_h\mathcal{E}_e^h) = -\mathbf{H}^{(1)}$; i.e.,

$$\lambda_k^e = -\lambda_k^{(1)} = 1, \quad k = 1, 2, \ldots, N. \tag{13}$$

We can expand $\mathbf{c}^{n+1} - \mathbf{c}^n$ in a basis of eigenvectors $\mathbf{v}_k$ as follows.

$$\mathbf{c}^{n+1} - \mathbf{c}^n = \sum_{k=1}^{N}\alpha_k\mathbf{v}_k. \tag{14}$$

The decrease of the discrete energy functional is established in the following theorem: If $\mathbf{c}^{n+1}$ is the solution of Eq. (8) with a given $\mathbf{c}^n$, then

$$\mathcal{E}^h(\mathbf{c}^{n+1}) \le \mathcal{E}^h(\mathbf{c}^n). \tag{15}$$

Next, we prove Eq. (15). With an exact Taylor expansion of $\mathcal{E}^h(\mathbf{c}^n)$ about $\mathbf{c}^{n+1}$ up to the second order, we have

$$\mathcal{E}^h(\mathbf{c}^n) = \mathcal{E}^h(\mathbf{c}^{n+1}) + \langle \mathrm{grad}_h\mathcal{E}^h(\mathbf{c}^{n+1}), \mathbf{c}^n - \mathbf{c}^{n+1}\rangle_h + \left\langle \frac{J(\mathrm{grad}_h\mathcal{E}^h)(\boldsymbol{\xi})}{2}(\mathbf{c}^n - \mathbf{c}^{n+1}), \mathbf{c}^n - \mathbf{c}^{n+1}\right\rangle_h,$$

where $\boldsymbol{\xi} = \theta\mathbf{c}^n + (1-\theta)\mathbf{c}^{n+1}$ and $0 \le \theta \le 1$. Rearranging the terms and using $\mathcal{E}^h = \mathcal{E}^{(1)} + \mathcal{E}^{(2)} + \mathcal{E}^{(3)}$, Eq. (11), Eq. (14), and the mean value theorem, we have

$$\begin{aligned}
\mathcal{E}^h(\mathbf{c}^{n+1}) - \mathcal{E}^h(\mathbf{c}^n) &= \left\langle \mathrm{grad}_h\mathcal{E}^h(\mathbf{c}^{n+1}) - \frac{J(\mathrm{grad}_h\mathcal{E}^h)(\boldsymbol{\xi})}{2}(\mathbf{c}^{n+1} - \mathbf{c}^n), \mathbf{c}^{n+1} - \mathbf{c}^n\right\rangle_h \\
&\le \left\langle \mathrm{grad}_h\mathcal{E}^h(\mathbf{c}^{n+1}) - \frac{1}{2}(\mathbf{H}^{(1)} + \mathbf{H}^{(2)})(\mathbf{c}^{n+1} - \mathbf{c}^n), \mathbf{c}^{n+1} - \mathbf{c}^n\right\rangle_h \\
&= \left\langle \mathrm{grad}_h\mathcal{E}^h(\mathbf{c}^{n+1}), \mathbf{c}^{n+1} - \mathbf{c}^n\right\rangle_h - \sum_{j,k=1}^{N}\left\langle \frac{1}{2}(\mathbf{H}^{(1)} + \mathbf{H}^{(2)})\alpha_j\mathbf{v}_j, \alpha_k\mathbf{v}_k\right\rangle_h \\
&= \left\langle \mathrm{grad}_h\mathcal{E}_c^h(\mathbf{c}^{n+1}) - \mathrm{grad}_h\mathcal{E}_e^h(\mathbf{c}^{n+1}) - \frac{1}{\Delta t}(\mathbf{c}^{n+1} - \mathbf{c}^n) - \mathrm{grad}_h\mathcal{E}_c^h(\mathbf{c}^{n+1}) \right. \\
&\quad \left. + \mathrm{grad}_h\mathcal{E}_e^h(\mathbf{c}^n), \mathbf{c}^{n+1} - \mathbf{c}^n\right\rangle_h - \sum_{j,k=1}^{N}\left\langle \frac{1}{2}(\lambda_j^{(1)} + \lambda_j^{(2)})\alpha_j\mathbf{v}_j, \alpha_k\mathbf{v}_k\right\rangle_h \\
&= -\left\langle \mathrm{grad}_h\mathcal{E}_e^h(\mathbf{c}^{n+1}) - \mathrm{grad}_h\mathcal{E}_e^h(\mathbf{c}^n), \mathbf{c}^{n+1} - \mathbf{c}^n\right\rangle_h - \frac{1}{\Delta t}\|\mathbf{c}^{n+1} - \mathbf{c}^n\|_h^2 \\
&\quad - \sum_{j,k=1}^{N}\left\langle \frac{1}{2}(\lambda_j^{(1)} + \lambda_j^{(2)})\alpha_j\mathbf{v}_j, \alpha_k\mathbf{v}_k\right\rangle_h
\end{aligned}$$

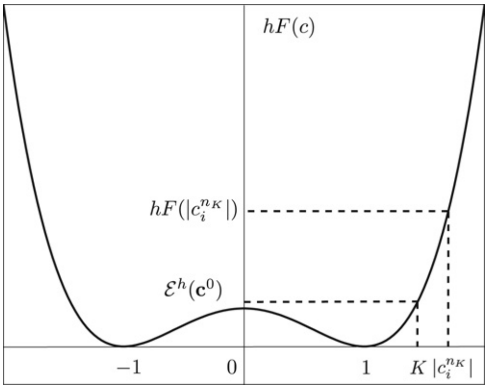


Fig. 2. Graph of $hF(c)$.

$$= -\sum_{j,k=1}^{N} \left\langle \left[J(\mathrm{grad}_h \mathcal{E}_e^h) + \frac{1}{2}(\lambda_j^{(1)} + \lambda_j^{(2)}) I \right] \alpha_j \mathbf{v}_j, \alpha_k \mathbf{v}_k \right\rangle_h - \frac{1}{\Delta t} \|\mathbf{c}^{n+1} - \mathbf{c}^n\|_h^2$$

$$= -\sum_{k=1}^{N} \frac{1}{2} \left(\lambda_k^e + \lambda_k^{(2)} \right) \alpha_k^2 - \frac{1}{\Delta t} \|\mathbf{c}^{n+1} - \mathbf{c}^n\|_h^2 \le 0,$$

where we have used the fact that λ_k^e is positive and $\lambda_k^{(2)}$ is non-negative. Therefore, we have proven the decrease of the discrete total energy. This completes the proof. The theorem holds for any time step Δt; hence, the method is unconditionally gradient stable. It should be emphasized that while the methods will allow us to take arbitrarily large time steps, the accuracy of the numerical solution depends on choosing a small enough time step to resolve the fast-time-scale dynamics.

3.2. Boundedness of the numerical solution

Next, we show that the decrease of the discrete total energy functional implies the pointwise boundedness of the numerical solution for the AC equation. If $\mathbf{c}^n$ is a numerical solution for the discrete Eq. (8), then there exists a constant K, independent of n, such that

$$\|\mathbf{c}^n\|_\infty \le K. \tag{16}$$

We prove Eq. (16) by a contradiction. Assume on the contrary that there is an integer n_K, dependent on K, such that $\|\mathbf{c}^{n_K}\|_\infty > K$ for all K. Then there is an index i $(1 \le i \le N)$ such that $|c_i^{n_K}| > K$. Let K be the largest solution of $hF(K) = \mathcal{E}^h(\mathbf{c}^0)$, i.e., $K = \sqrt{1 + 2\sqrt{\mathcal{E}^h(\mathbf{c}^0)/h}}$. Note that $K \ge 1$. Then, $F(c)$ is a strictly increasing function on (K, ∞) (see Fig. 2). Since the total energy is non-increasing, we have $\mathcal{E}^h(\mathbf{c}^0) = hF(K) < hF(|c_i^{n_K}|) \le \mathcal{E}^h(\mathbf{c}^{n_K}) \le \mathcal{E}^h(\mathbf{c}^0)$. This contradiction implies that Eq. (16) should be satisfied.

4. Numerical experiments

In this section, we perform the following: finding relation between ϵ value and the width of transition layer, comparing the numerical equilibrium solution with the analytic equilibrium solution, investigating properties of AC equation, and checking the total energy decrease. We also implement the unconditionally gradient stable scheme in Eq. (8) with the recently developed adaptive mesh refinement (AMR) methodology. For detailed descriptions of the numerical method used in solving these equations with AMR, we refer to Refs. [14,15] and the references therein. A uniform mesh solution algorithm using a nonlinear multigrid is described in Appendix.

4.1. The relation between the ϵ value and the width of the transition layer

In our first numerical experiment, we consider the relation between the ϵ value and the width of the transition layer for the AC equation. From our choice of the total energy density Eq. (7) and an equilibrium profile $c(x) = \tanh(x/(\sqrt{2}\epsilon))$ on the infinite domain, the concentration field varies from -0.9 to 0.9 over a distance of about $2\sqrt{2}\epsilon \tanh^{-1}(0.9)$. Therefore, if we want this value to be about m grid points, then

$$\epsilon_m = \frac{hm}{2\sqrt{2}\tanh^{-1}(0.9)}. \tag{17}$$

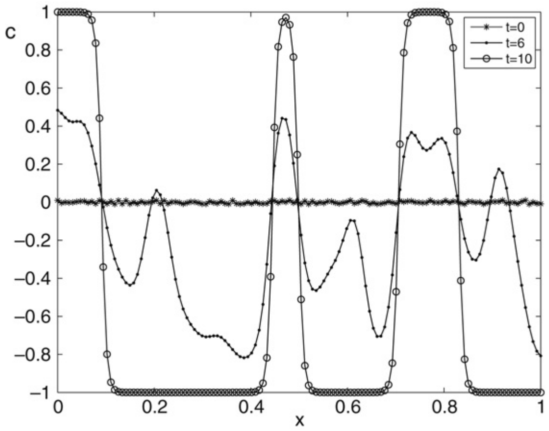


Fig. 3. The evolution of an initial random distribution of concentration, $c(x,0)=0.01\text{rand}(x)$. The concentration profile is shown at $t=0,6$, and 10.

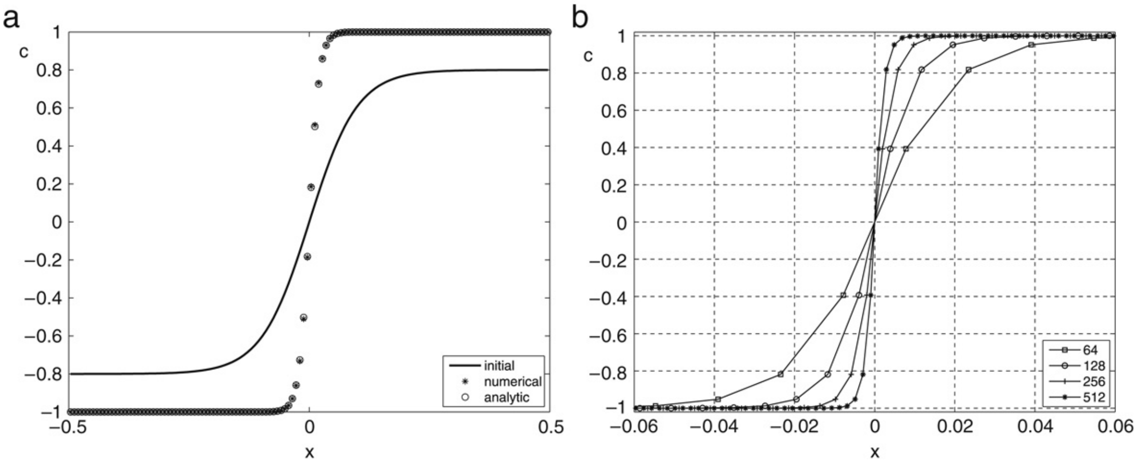


Fig. 4. (a) Lines with circles and stars denote the analytic and numerical equilibrium solutions with an initial concentration $c(x,0)=0.8\tanh(x/(\sqrt{2}\epsilon_8))$, respectively. (b) Numerical equilibrium solutions with an initial concentration $c(x,0)=0.8\tanh(x/(\sqrt{2}\epsilon_8))$ on $\Omega=(-0.5,0.5)$. Lines with the symbols, '$\diamond$', '$\circ$', '$+$', and '$\star$', are numerical results with 64, 128, 256, and 512 mesh sizes with ϵ_4, respectively.

To confirm this, we ran a simulation with the initial condition $c(x,0)=0.01\text{rand}(x)$ on the unit domain $\Omega=(0,1)$ with $h=1/128$, $\Delta t=0.05$, and ϵ_4 (see the line with stars in Fig. 3). Here, $\text{rand}(x)$ is a random number between -1 and 1. In Fig. 3, we see that the transition layer (from $c=-0.9$ to $c=0.9$) is about 4 grid points at time $t=10$.

4.2. An equilibrium profile

Next, we compare numerical equilibrium solutions with analytic ones. We stop the numerical computations when the discrete l_2-norm of the difference between $(n+1)$th and nth time step solutions becomes less than 10^{-10}, $\|c^{n+1}-c^n\|_h\leq 10^{-10}$ and we take c^{n+1} as a numerical equilibrium solution. The initial concentration is $c(x,0)=0.8\tanh(x/(\sqrt{2}\epsilon_8))$ on $\Omega=(-0.5,0.5)$. We take $h=1/128$, ϵ_8, and $\Delta t=0.1$. In Fig. 4(a), the solid, '$*$', and '$\circ$' denoted lines are the initial condition, the numerical equilibrium solution, and the analytic equilibrium solution, respectively. The numerical equilibrium profile matches well with the analytic equilibrium solution $c_{eq}^{\infty}(x)=\tanh(x/(\sqrt{2}\epsilon_8))$ on $\Omega=(-\infty,\infty)$.

In Fig. 4(b), numerical equilibrium solutions with an initial concentration $c(x,0)=0.8\tanh(x/(\sqrt{2}\epsilon_4))$ on $\Omega=(-0.5,0.5)$ with various mesh sizes are shown. Lines with the symbols, '$\diamond$', '$\circ$', '$+$', and '$\star$', denote the numerical results with 64, 128, 256, and 512 mesh sizes with ϵ_4, respectively. In this case, ϵ_4^2/h^2 is constant in Eq. (8); therefore, we have the same discrete equations and have almost the same values with different mesh sizes with respect to grid points from the origin.

4.3. The exact solution

The partial differential equation (1) in one dimensional space may be written as

$$c_t=-c^3+c+\epsilon^2\Delta c. \tag{18}$$

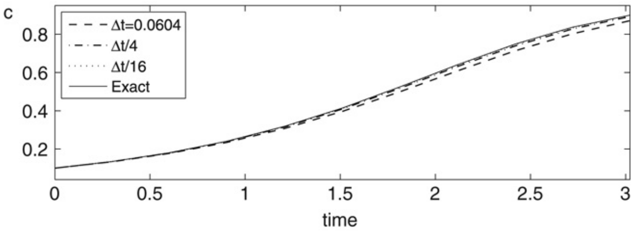


Fig. 5. The evolution of the constant value of $c(x, t)$ with different time steps up to the final time, $T = 3.02257$.

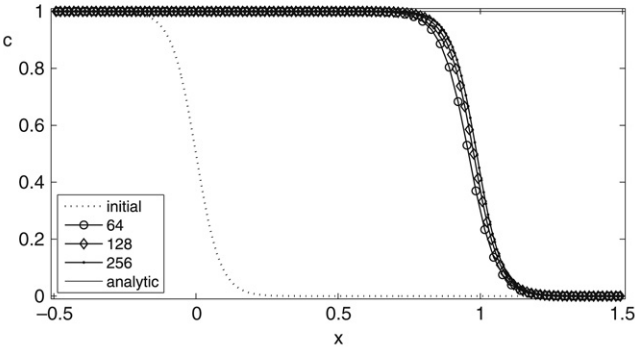


Fig. 6. Numerical traveling wave solutions with an initial profile, $c(x, 0) = \frac{1}{2}(1 - \tanh \frac{x}{2\sqrt{2}\epsilon})$. The final time is $T = 1/s$. The analytic solution is a solid line. Lines with symbols, '$\circ$', '$\diamond$', and '$\cdot$' represent mesh sizes of 64, 128, and 256, respectively.

If we take the initial condition as a constant, i.e., $c(x, 0) = c_0$, then, the exact solution of Eq. (18) is

$$c(x,t) = \frac{c_0 e^t}{\sqrt{1 + c_0^2(e^{2t} - 1)}}. \tag{19}$$

If $c_0 = 0.1$, then in order to find a t that satisfies $c(x, t) = 0.9$, we solve the Eq. (19) and get an approximate value, $t \sim 3.02257$. The initial state is taken to be $c(x, 0) = 0.1$ on the computational domain $\Omega = (0, 1)$ with $h = 1/32$ and ϵ_4. We set the final time $T = 3.02257$ and time step $\Delta t = T/100$. Fig. 5 shows an evolution of the constant concentration $c(x, t)$ with different time steps ($\Delta t = 0.0604$, $\Delta t/4$, $\Delta t/16$) up to the final time. We can see the convergence of the numerical solutions with respect to the time steps.

4.4. Traveling wave solutions

We seek traveling wave solutions of Eq. (1) as a following form,

$$c(x,t) = \frac{1}{2}\left(1 - \tanh \frac{x - st}{2\sqrt{2}\epsilon}\right), \tag{20}$$

where s is the speed of the traveling wave. By substituting Eq. (20) into Eq. (1), we arrive at the following equation.

$$\frac{\sqrt{2}s - 3\epsilon}{8\epsilon} \operatorname{sech}^2\left(\frac{x - \alpha t}{2\sqrt{2}\epsilon}\right) = 0. \tag{21}$$

Therefore, the speed of the traveling wave is $s = 3\epsilon/\sqrt{2}$. In Fig. 6, the numerical traveling wave solutions (where symbols '$\circ$', '$\diamond$', and '$\cdot$' represent mesh sizes of 64, 128, and 256, respectively) with an initial profile, $c(x, 0) = \frac{1}{2}(1 - \tanh \frac{x}{2\sqrt{2}\epsilon})$ and on a computational domain, $\Omega = (-0.5, 1.5)$. The final time is $T = 1/s$ and the fixed time step is $\Delta t = T/400$. The analytic final profile is $c(x, 1/s) = \frac{1}{2}(1 - \tanh \frac{x-1}{2\sqrt{2}\epsilon})$. The convergence of the results with grid refinement is qualitatively evident.

To obtain a quantitative estimate of the rate of convergence, we perform a number of simulations for the same initial problem on a set of increasingly finer grids and time steps. The numerical solutions are computed on the uniform grids and time steps, $h = 2/2^n$ and $\Delta t = 5h^2$, for $n = 6, 7,$ and 8. The errors and rates of convergence are given in Table 1. The results suggest that the scheme is indeed second order accurate in space and first order in time.

In Fig. 7, we show numerical traveling wave solutions with an initial profile, $c(x, 0) = 1$ if $x < 0.2$; $c(x, 0) = 0$, otherwise. Solid, circle, diamond, and star lines are an initial profile, ϵ_4, $2\epsilon_4$, and $4\epsilon_4$, respectively. The results match well with the theoretical prediction of the speed, which depends linearly on the ϵ value.

Table 1
Convergence results.

Case	64	Rate	128	Rate	256
l_2	1.0709e−2	1.9938	2.6887e−3	2.0140	6.6570e−4

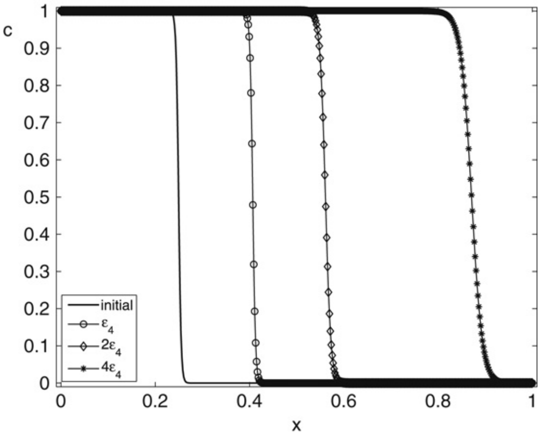


Fig. 7. Numerical traveling wave solutions with an initial profile (solid line), $c(x, 0) = 1$ if $x < 0.2$; $c(x, 0) = 0$, otherwise. The circle, diamond, and star delineated lines represent numerical solutions with ϵ_4, $2\epsilon_4$, and $4\epsilon_4$, respectively. The computational mesh is 512, $\Delta t = 0.1$, and the final time is $T = 40$.

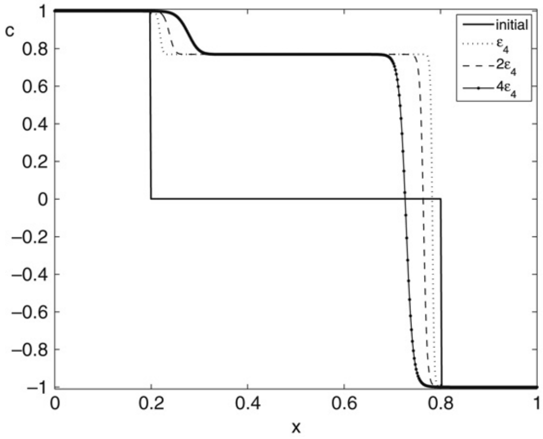


Fig. 8. The evolution of the initial concentration is shown; $c(x, 0) = 1$ if $x < 0.2$ and $c(x, 0) = -1$ if $x > 0.8$; $c(x, 0) = 0.001$, otherwise with ϵ_4, $2\epsilon_4$, and $4\epsilon_4$. The computational mesh is 512, $\Delta t = 1/40$, and the final time is $T = 10$.

Fig. 8 shows evolution of the initial concentration, $c(x, 0) = 1$ if $x < 0.2$, $c(x, 0) = -1$ if $x > 0.8$; $c(x, 0) = 0.001$, otherwise with ϵ_4, $2\epsilon_4$, and $4\epsilon_4$. The computational mesh is 512, $\Delta t = 1/40$, and the final time is $T = 10$. The two transition layers at around $x = 0.2$ and $x = 0.8$ travels like traveling wave solution. On the other hand, in the middle in the domain behaves like a constant solution, Eq. (19). And this constant solution does not depend on ϵ values.

4.5. Metastable states

In Fig. 9(a) and (b) show the evolutions of the concentration $c(x, t)$ with 12 points and 13 points negative one with ϵ_7, respectively. The computational mesh is 128 and $\Delta t = 1/128$. In Fig. 9(c), the open circles denote ϵ_m and the number of points that have constant equilibrium solutions; while the stars indicate parameters that have non-constant equilibrium solutions.

4.6. Linear stability analysis

We perform a linear stability analysis around a spatially constant critical composition solution $c \equiv 0$. Linearizing the partial differential equation (8) about $c \equiv 0$ gives

$$c_t = c + \epsilon^2 \Delta c. \tag{22}$$

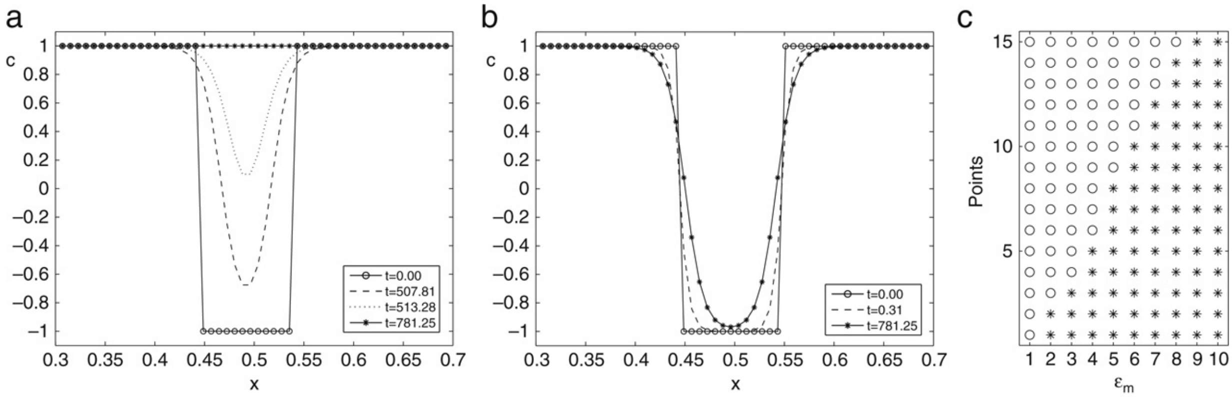


Fig. 9. (a) and (b) are the evolutions of the concentration $c(x, t)$ with 12 points and 13 points negative one with ϵ_7, respectively. (c) The stars denote the pairs of ϵ_m and the number of negative one points that have constant equilibrium solutions as shown in (a), while the open circles indicate parameters that have non-constant equilibrium solutions as shown in (b).

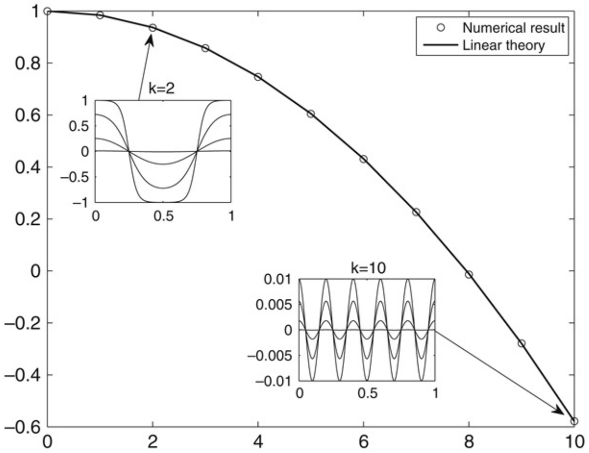


Fig. 10. Numerical and exact λ values for different wave numbers k ($k = 0, 1, \ldots, 10$) and $\epsilon = 0.04$. The insets illustrate time evolutions of the initial profiles.

Next, we let $c = \alpha(t)\cos(k\pi x)$. Then from Eq. (22) we have

$$\alpha'(t)\cos(k\pi x) = \alpha(t)\cos(k\pi x) - (\epsilon k\pi)^2\alpha(t)\cos(k\pi x). \tag{23}$$

By dividing Eq. (23) by $\cos(k\pi x)$, we obtain

$$\alpha'(t) = [1 - (\epsilon k\pi)^2]\alpha(t). \tag{24}$$

The solution of the ordinary differential equation (24) is given by

$$\alpha(t) = \alpha(0)\mathrm{e}^{\lambda t}, \tag{25}$$

where $\lambda = 1 - (\epsilon k\pi)^2$. The numerical growth rate is defined by

$$\tilde{\lambda} = \frac{1}{T}\log\left(\frac{\max\limits_{1\le i\le N}|c_i^{N_t}|}{\alpha(0)}\right). \tag{26}$$

The initial state is taken to be $c(x, 0) = 0.01\cos(k\pi x)$ on the computational domain, $\Omega = (0, 1)$. We use parameters such as the final time $T = 0.1$, time step $\Delta t = 0.01$, $\alpha(0) = 0.01$, $\epsilon = 0.04$, and $N = 256$. In Fig. 10, numerical $\tilde{\lambda}$ ('○') and exact λ (solid line) values for different wave numbers k ($k = 0, 1, \ldots, 10$) are shown and are in good agreement.

4.7. The decrease of the total energy

In order to demonstrate that the numerical scheme is unconditionally gradient stable, we consider the evolution of the discrete total energy. The initial state is taken to be $c(x, y, 0) = 0.01\mathrm{rand}(x, y)$ on the computational domain $\Omega = (0, 1) \times (0, 1)$ with 128×128. $\mathrm{rand}(x, y)$ is a random number between -1 and 1. We use the simulation parameters ϵ_4

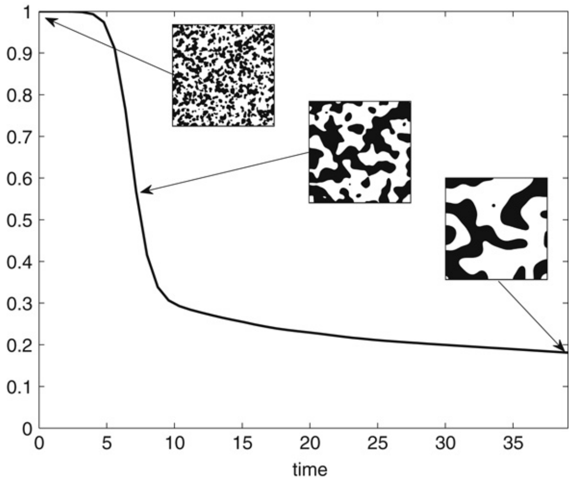


Fig. 11. The time dependent non-dimensional discrete total energy $\mathcal{E}^h(\mathbf{c}^n)/\mathcal{E}^h(\mathbf{c}^0)$ of the numerical solutions with the initial data, $c(x, y, 0) = 0.01\text{rand}(x, y)$.

and $\Delta t = 10/128$. In Fig. 11, the time evolution of the non-dimensional discrete total energy $\mathcal{E}^h(\mathbf{c}^n)/\mathcal{E}^h(\mathbf{c}^0)$ is shown. Also, the inscribed small figures are the concentration fields at the indicated times. The total discrete energy is non-increasing as predicted by Eq. (15).

4.8. Mean curvature flow

After rescaling the time variable, Eq. (1) becomes

$$c_t = -\frac{c(c^2-1)}{\epsilon^2} + \Delta c. \tag{27}$$

It was first formally proved that, as $\epsilon \to 0$, the zero level set of c, denoted by $\Gamma_t^\epsilon := \{\mathbf{x} \in \Omega; c(\mathbf{x}, t) = 0\}$ approaches to a surface Γ_t that evolves according to the geometric law

$$V = -\kappa = -\left(\frac{1}{R_1} + \frac{1}{R_2}\right), \tag{28}$$

where V is the normal velocity of the surface Γ_t at each point, κ is its mean curvature, and R_1, R_2 are the principal radii of curvatures at the point of the surface [1]. In two dimensions, with a single radius of curvature, Eq. (28) becomes $V = -1/R$.

An initial condition is given with a circle centered at the center of a domain $\Omega = (0, 1) \times (0, 1)$. If we set the initial radius of the circle to r_0 and denote the radius at time t as $r(t)$, then Eq. (28) becomes $\mathrm{d}r(t)/\mathrm{d}t = -1/r(t)$. Its solution is given as

$$r(t) = \sqrt{r_0^2 - 2t}. \tag{29}$$

Let $r_0 = 0.25$. Then, the initial condition is

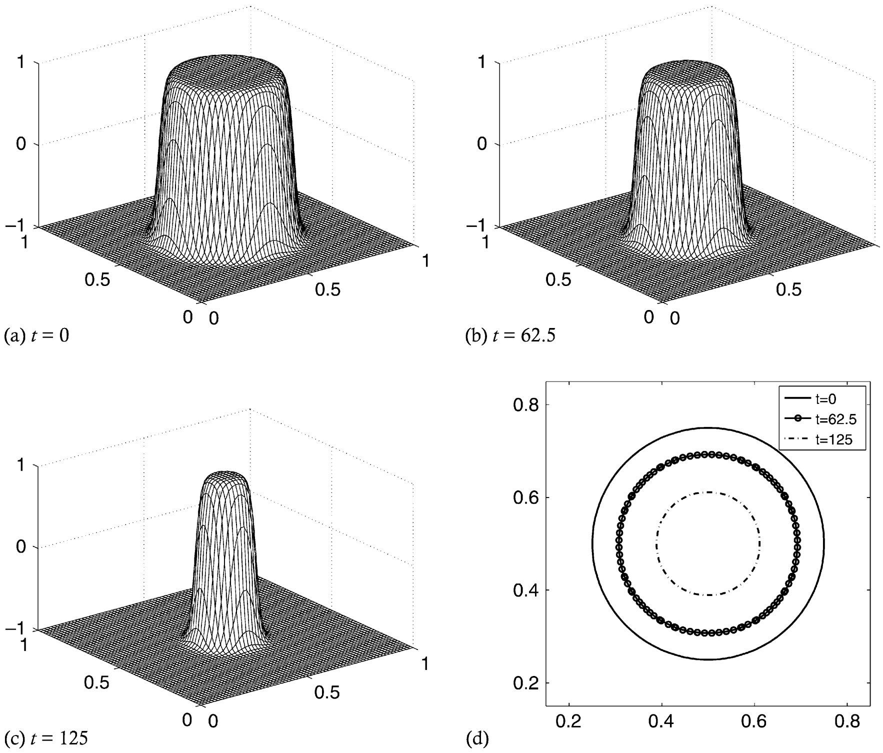
$$c(x, y, 0) = \tanh \frac{0.25 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}. \tag{30}$$

In Fig. 12, (a), (b), and (c) are evolutions of the initial concentration $c(x, y, 0)$, Eq. (30). The times are shown below each figure. (d) Illustrates the zero level contour lines of (a), (b), and (c). We observe that the circle shrinks as theoretically predicted.

In Fig. 13, we show that as the mesh size decreases from $h = 1/64$ ('∘') to $h = 1/128$ ('⋄') and $h = 1/256$ ('·'), the plot of numerical radius of the circle $r(t)$ in time becomes closer and closer to the asymptotic value (solid line) given by Eq. (29). We note that as mesh sizes decrease the actual value of ϵ_4 decreases also. This result confirms the theory that as $\epsilon \to 0$ then $\Gamma_t^\epsilon \to \Gamma_t$.

4.9. Adaptive mesh refinement

Across the spatial interfaces, the solution undergoes an $O(1)$ change over an $O(\epsilon)$ interval. If these interfaces are to be accurately resolved, a fine discretization of space is required. Therefore, an adaptive mesh refinement (AMR) of the space



(a) $t = 0$

(b) $t = 62.5$

(c) $t = 125$

(d)

Fig. 12. (a), (b), and (c) show evolutions of the initial concentration $c(x, y, 0)$, Eq. (30). The times are shown below each figure. (d) Illustrates the zero level contour lines of (a), (b), and (c).

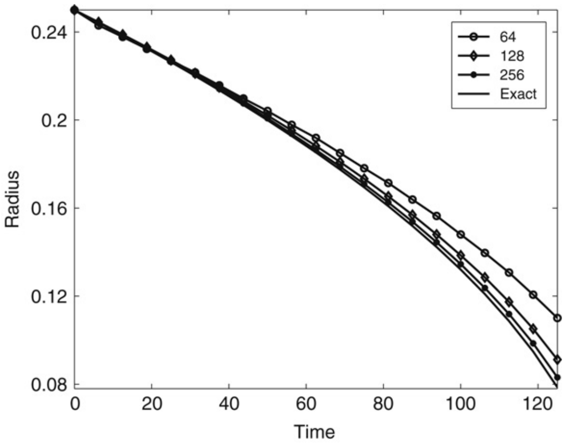
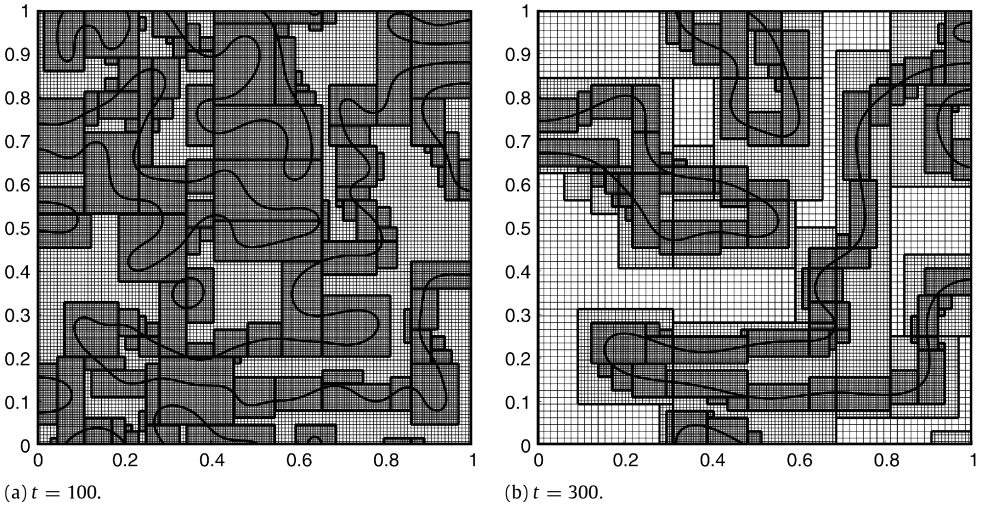


Fig. 13. Mesh refinement with ϵ_4.

is necessary. In this approach, the computational mesh is locally refined in regions where greater accuracy is desired. We use Marsha Berger's and Phillip Colella's block-structured approach; where refinement is organized in logically rectangular regions of the domain [16–20]. We also implement the unconditionally gradient stable scheme in Eq. (8) with the recently developed adaptive mesh refinement methodology. For a detailed description of the numerical method used in solving these equations with AMR, please refer to Refs. [14,15].

In this simulation we consider the evolution of an initial state is taken to be $c(x, y, 0) = 0.01\text{rand}(x, y)$ on the computational domain $\Omega = (0, 1) \times (0, 1)$. We use a base 64×64 mesh with two levels of refinement ratios of 2. Therefore, the effective fine mesh size is 256×256. We use the simulation parameters such as ϵ_4 based on the effective fine mesh and $\Delta t = 0.1$. Fig. 14 shows the evolution of the concentration $c(x, y, t)$ at times $t = 100$ and 300. At $t = 300$, the mesh adapts around the zero level set.



(a) $t = 100$.

(b) $t = 300$.

Fig. 14. The evolution of the mesh and concentration $c(x, y, t) = 0$ under anti-phase domain coarsening. The times are shown below each figure. The effective fine grid resolution for 2 levels of adaptivity is 256 × 256.

5. Conclusions

In this paper, we reviewed a derivation of the AC equation as a gradient flow and showed that a numerical scheme for the AC equation is unconditionally gradient stable by using eigenvalues of the Hessian matrix of the energy functional. We also showed that the decrease of the discrete total energy functional implies the pointwise boundedness of the numerical solution for the AC equation. We investigated a variety of phenomena associated with the AC equation. We have uncovered a traveling wave solution to the AC equation and found that its speed depends linearly on the interfacial energy parameter, i.e., $s = 3\epsilon/\sqrt{2}$.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2008- C1090-0801-0013). The first author (J.-W. Choi) was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-041-C00073). The corresponding author (J.S. Kim) was also supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2008-C00044). The authors thank an anonymous referee for very useful comments on this paper.

Appendix. Numerical solution — A nonlinear multigrid method

In this section, we use a nonlinear Full Approximation Storage (FAS) multigrid method to solve the nonlinear discrete Eq. (8) at the implicit time level. The nonlinearity is treated using one step of Newton's iteration. A pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. See the reference text [21] for additional details and background. The algorithm of the nonlinear multigrid method for solving the discrete AC system is : First, we rewrite Eq. (8) as follows.

$$N(c^{n+1}) = \phi^n, \tag{A.1}$$

where $N(c^{n+1}) = c^{n+1}/\Delta t + (c^{n+1})^3 - \epsilon^2 \Delta_h c^{n+1}$ and the source term is $\phi^n = c^n/\Delta t + c^n$.

In the following description of one FAS cycle, we assume a sequence of grids Ω_k (Ω_{k-1} is coarser than Ω_k by a factor of 2). Given the number β of pre- and post-smoothing relaxation sweeps, an iteration step for the nonlinear multigrid method using the V-cycle is formally written [21]:

FAS multigrid cycle

$$c_k^{m+1} = FAScycle(k, c_k^m, N_k, \phi_k^n, \beta).$$

That is, c_k^m and c_k^{m+1} are the approximation of $c^{n+1}(x_i, y_j)$ before and after an FAScycle. Now, define the FAScycle.

Step (1) *Presmoothing*

$$\bar{c}_k^m = SMOOTH^\beta(c_k^m, N_k, \phi_k^n),$$

which means performing β smoothing steps with the initial approximation c_k^m, source term ϕ_k^n, and *SMOOTH* relaxation operator to get the approximation $\bar{c}_k^m$. Here, we derive the smoothing operator in two dimensions. Since $(c_{ij}^{n+1})^3$ in Eq. (A.1) is nonlinear with respect to c_{ij}^{n+1}, we linearize $(c_{ij}^m)^3$ at c_{ij}^m, i.e.,

$$(c_{ij}^{n+1})^3 \approx (c_{ij}^m)^3 + 3(c_{ij}^m)^2(c_{ij}^{n+1} - c_{ij}^m).$$

After substituting this expression into (A.1), we obtain

$$\left(\frac{1}{\Delta t} + \frac{4\epsilon^2}{h^2} + 3(c_{ij}^m)^2\right) c_{ij}^{n+1} = \phi_{ij}^n + \epsilon^2 \frac{c_{i+1,j}^{n+1} + c_{i-1,j}^{n+1} + c_{i,j+1}^{n+1} + c_{i,j-1}^{n+1}}{h^2} + 2(c_{ij}^m)^3. \tag{A.2}$$

Next, we replace $c_{\alpha\beta}^{n+1}$ in Eq. (A.2) with $\bar{c}_{\alpha\beta}^m$ if $(\alpha \le i)$ or $(\alpha = i$ and $\beta \le j)$; otherwise, with $c_{\alpha\beta}^m$, i.e.,

$$\left(\frac{1}{\Delta t} + \frac{4\epsilon^2}{h^2} + 3(c_{ij}^m)^2\right) \bar{c}_{ij}^{m} = \phi_{ij}^n + \epsilon^2 \frac{c_{i+1,j}^{m} + \bar{c}_{i-1,j}^{m} + c_{i,j+1}^{m} + \bar{c}_{i,j-1}^{m}}{h^2} + 2(c_{ij}^m)^3. \tag{A.3}$$

Step (2) *Coarse grid correction*

- Compute the defect: $\bar{d}_k^m = \phi_k^n - N_k(\bar{c}_k^m)$.
- Restrict the defect and $\bar{c}_k^m$: $\bar{d}_{k-1}^m = I_k^{k-1}(\bar{d}_k^m)$, $\bar{c}_{k-1}^m = I_k^{k-1}(\bar{c}_k^m)$.
- Compute the right-hand side: $\phi_{k-1}^n = \bar{d}_{k-1}^m + N_{k-1}(\bar{c}_{k-1}^m)$.
- Compute an approximate solution $\hat{c}_{k-1}^m$ of the coarse grid equation on Ω_{k-1}, i.e.

$$N_{k-1}(c_{k-1}^m) = \phi_{k-1}^n. \tag{A.4}$$

  If $k = 1$, we apply the smoothing procedure in *Step* (1) to obtain the approximate solution. If $k > 1$, we solve (A.4) by performing a FAS k-grid cycle using $\bar{c}_{k-1}^m$ as an initial approximation:

$$\hat{c}_{k-1}^m = \text{FAScycle}(k-1, \bar{c}_{k-1}^m, N_{k-1}, \phi_{k-1}^n, \beta).$$

- Compute the coarse grid correction (CGC): $\hat{v}_{k-1}^m = \hat{c}_{k-1}^m - \bar{c}_{k-1}^m$.
- Interpolate the correction: $\hat{v}_k^m = I_{k-1}^k \hat{v}_{k-1}^m$.
- Compute the corrected approximation on Ω_k: $c_k^{m,\text{ after } CGC} = \bar{c}_k^m + \hat{v}_k^m$.

Step (3) *Postsmoothing*: $c_k^{m+1} = SMOOTH^{\beta}(c_k^{m,\text{ after } CGC}, N_k, \phi_k^n)$.
This completes the description of a nonlinear FAScycle.

References

[1] S.M. Allen, J.W. Cahn, Acta Metall. 27 (1979) 1085.
[2] M. Gokieli, L. Marcinkowski, Nonlinear Anal. 63 (2005) 1143.
[3] M. Beneš, V. Chalupecký, K. Mikula, Appl. Numer. Math. 51 (2004) 187.
[4] J.A. Dobrosotskaya, A.L. Bertozzi, IEEE Trans. Image Process. 17 (2008) 657.
[5] X. Feng, A. Prohl, Numer. Math. 94 (2003) 33.
[6] A.A. Wheeler, W.J. Boettinger, G.B. McFadden, Phys. Rev. A 45 (1992) 7424.
[7] E.V.L. Melloa, O.T.S. Filho, Physica A 347 (2005) 429.
[8] D.J. Eyre, http://www.math.utah.edu/~eyre/research/methods/stable.ps.
[9] D.J. Eyre, Computational and Mathematical Models of Microstructural Evolution, The Materials Research Society, Warrendale, 1998, p. 39.
[10] Y. He, Y. Liu, T. Tang, Appl. Numer. Math. 57 (2007) 616.
[11] C. Cowan, M.S.Thesis, Simon Fraser University, Canada, 2005.
[12] P.C. Fife, Electron. J. Differential Equations 2000 (2000) 1.
[13] D.F. Martin, P. Colella, M. Anghel, F. Alexander, Comput. Sci. Eng. 7 (2005) 24.
[14] J.S. Kim, J. Korean Phys. Soc. 49 (2006) 1903.
[15] S.M. Wise, J.S. Kim, J.S. Lowengrub, J. Comput. Phys. 226 (2007) 414.
[16] A.S. Almgren, et al., J. Comput. Phys. 142 (1998) 1.
[17] M.J. Berger, P. Colella, J. Comput. Phys. 82 (1989) 64.
[18] P. Colella, M. Dorr, D. Wake, J. Comput. Phys. 152 (1999) 550.
[19] D. Martin, P. Colella, J. Comput. Phys. 163 (2000) 271.
[20] R. Samtaney, S.C. Jardin, P. Colella, D.F. Martin, Comput. Phys. Comm. 164 (2004) 220.
[21] U. Trottenberg, C. Oosterlee, A. Schüller, MULTIGRID, Academic press, 2001.