Adaptive finite element methods for Cahn–Hilliard equations

L’ubomír Baňas*, 1, Robert Nürnberg

Department of Mathematics, Imperial College, London SW7 2AZ, UK

Received 4 February 2007

Abstract

We develop a method for adaptive mesh refinement for steady state problems that arise in the numerical solution of Cahn–Hilliard equations with an obstacle free energy. The problem is discretized in time by the backward-Euler method and in space by linear finite elements. The adaptive mesh refinement is performed using residual based a posteriori estimates; the time step is adapted using a heuristic criterion. We describe the space–time adaptive algorithm and present numerical experiments in two and three space dimensions that demonstrate the usefulness of our approach.

© 2007 Elsevier B.V. All rights reserved.

MSC: 35M10; 74S05

Keywords: Cahn–Hilliard equation; Obstacle free energy; Finite elements; A posteriori estimates; Adaptive numerical methods

1. Introduction

The properties of commercially produced materials depend on microstructures which are generated using special processing techniques, such as phase separation and coarsening mechanisms. Accurate prediction of microstructure or the evolution of pattern formation during phase separation and coarsening are therefore of considerable interest in materials science. In this paper we are concerned with the Cahn–Hilliard equation which was proposed in [6] as a phenomenological model for phase separation in binary alloys. In practice it is important to understand the factors that influence phase separation and coarsening, respectively. As it is difficult to obtain such information by real-life experiments, reliable numerical computations are very important. It is the aim of this paper to introduce a robust and reliable mesh refinement algorithm in two and three space dimensions, that allows for efficient and reliable numerical simulations.

Let Ω be a convex polygonal domain in $\mathbb{R}^d$, $d = 2, 3$. The Cahn–Hilliard equation then takes the following form:

$$\frac{\partial u}{\partial t} = -\frac{1}{\gamma} \Delta u \quad \text{in} \quad \Omega_T := \Omega \times [0, T],$$

$$w = -\gamma \Delta u + \frac{1}{\gamma} \Psi'(u) \quad \text{in} \quad \Omega_T,$$
The phase function $u : \Omega_T \rightarrow [-1, 1]$ represents the difference in the mass fraction of the two alloy components; the values $u = \pm 1$ are associated with the pure materials. The function $w : \Omega_T \rightarrow \mathbb{R}$ represents the chemical potential. We note that (1) can be derived from balance considerations as a gradient flow for the free energy

$$\mathcal{E}(u) := \int_{\Omega} \frac{\gamma}{2} |\nabla u|^2 + \frac{1}{\gamma} \Psi(u) \, dx,$$

with the chemical potential $w := \delta \mathcal{E} / \delta u$ being the variational derivative of the energy $\mathcal{E}$ with respect to $u$.

For notational convenience in (1) it was implicitly assumed that the free energy $\Psi$ is differentiable. However, in this paper we will consider the so-called obstacle free energy, which is defined as follows:

$$\Psi(s) := \begin{cases} \frac{1}{2} (1 - s^2) & \text{if } s \in [-1, 1], \\ \infty & \text{if } s \not\in [-1, 1]. \end{cases}$$

Hence the chemical potential $w$ needs to be computed with the help of a variational inequality, see (3) below. Of course, the obstacle free energy $\Psi$ forces the phase variable $u$ to stay within the interval $[-1, 1]$.

The thickness of the interfacial regions, i.e., the region where $|u| < 1$, is asymptotically of order $\mathcal{O}(\gamma)$. It can be shown that in the sharp interface limit (i.e., when $\gamma \to 0$) the long time dynamics of the Eq. (1) correspond to the Mullins–Sekerka equation, see, e.g., [1,7].

Finite element methods for Eq. (1) have been proposed and analyzed in [5]. In [4] a finite element approximation for a related, so-called degenerate, Cahn–Hilliard equation was considered, and in addition a heuristic adaptive mesh refinement algorithm was used for numerical simulations in two space dimensions, in order to increase the efficiency of the computations. This approximation and the corresponding mesh refinement have recently been extended to three space dimensions in [2]. It is the aim of this paper to construct an adaptive finite element method for (1) based on rigorous a posteriori estimates in two and three space dimensions.

In Section 2 we describe the finite element method for the approximation of (1). We formulate the a posteriori estimate for the approximation error together with a simple heuristic criterion for time step control. In Section 3 we propose an adaptive algorithm for the mesh refinement based on the a posteriori estimate and discuss some aspects of the numerical implementation. Finally, in Section 4 we perform numerical experiments in order to demonstrate the efficiency of the proposed numerical method.

2. Finite element discretization

The time dependent problem (1) is usually discretized in time by the backward-Euler method, which leads to unconditionally stable approximations see [5] for more details. In effect one needs to solve at each time step the following semi-discrete problems: find functions $u \in \mathcal{X} := \{ \psi \in H^1(\Omega); \|\psi\| \leq 1 \text{ a.e. in } \Omega \}$, with $H^1(\Omega)$ being the usual first order Sobolev space, and $w \in H^1(\Omega)$ such that

$$(u, \phi) + \frac{\tau}{\gamma} (\nabla w, \nabla \phi) = (u^\text{old}, \phi) \quad \forall \phi \in H^1(\Omega),$$

$$\gamma (\nabla u, \nabla (\psi - u)) - (w, \psi - u) \geq \frac{1}{\gamma} (u^\text{old}, \psi - u) \quad \forall \psi \in \mathcal{X},$$

where $\tau$ is a given time step size and $\gamma > 0$ is a constant. Moreover, here and throughout $(\cdot, \cdot)$ denotes the $L^2$ inner product over $\Omega$. The function $u^\text{old}$ represents the solution from the previous time level. For simplicity, we consider a homogeneous Neumann boundary condition for $u$ and $w$ on the boundary $\partial \Omega$, recall (1).

We now introduce linear finite elements in space. Let $\mathcal{F}^h$ be a regular partition of $\overline{\Omega}$ into simplices $T$, with $h_T = \text{diam}(T)$. Similarly, for all inter element sides (edges/faces) $e$ we define $h_e = \text{diam}(e)$. With $\mathcal{F}^h$ we associate the space of continuous piecewise linear functions

$$V^h = \{ \phi \in C^0(\overline{\Omega}); \forall T \in \mathcal{F}^h, \phi|_T \in P^1(T) \} \subset H^1(\Omega),$$
and
\[ \mathcal{X}^h = \{ \psi \in V^h; |\psi| \leq 1 \text{ in } \Omega \} \subset \mathcal{X}. \]

The discrete problem then reads as follows. Find \( u_h \in \mathcal{X}^h \) and \( w_h \in V^h \) such that
\[
(u_h, \phi)_h + \frac{\tau}{\gamma} (\nabla w_h, \nabla \phi) = (u_h^{\text{old}}, \phi)_h \quad \forall \phi \in V^h,
\]
\[
\gamma (\nabla u_h, \nabla (\psi - u_h)) - (w_h, \psi - u_h)_h \geq \frac{1}{\gamma} (u_h^{\text{old}}, \psi - u)_h \quad \forall \psi \in \mathcal{X}^h. \tag{4}
\]

with \((\phi, \psi)_h = \int_{\Omega} I_h(\phi \psi)\) for \( \phi, \psi \in V^h \) the usual mass lumped inner product, and \( I_h \) being the usual Lagrange interpolation operator onto \( V^h \).

For details on the existence and uniqueness of solutions to (3) and (4) we refer to [5] and the references therein.

We denote the approximation errors by
\[ e_u = u - u_h, \]
\[ e_w = w - w_h. \tag{5} \]

The domain can be decomposed into
\[ \Omega = \mathcal{C}_h(u_h) \cup \mathcal{F}_h(u_h) \cup \mathcal{N}_h(u_h), \tag{6} \]

where
\[ \mathcal{C}_h(u_h) := \cup\{ T \in \mathcal{T}^h; |u_h| = 1 \text{ on } \mathcal{U}_h(T) \}, \]
\[ \mathcal{N}_h(u_h) := \cup\{ T \in \mathcal{T}^h; |u_h| < 1 \text{ on } T \}, \]
\[ \mathcal{F}_h(u_h) := \Omega \setminus (\mathcal{C}_h(u_h) \cup \mathcal{N}_h(u_h)). \]

In our context, \( \mathcal{C}_h \) denotes the subdomains with pure materials, \( \mathcal{N}_h \) denotes the diffuse interface and \( \mathcal{F}_h \) is the so-called discrete free boundary between \( \mathcal{C}_h \) and \( \mathcal{N}_h \).

The symbol \( \mathcal{U}_h(T) \) stands for a discrete neighborhood of a triangle \( T \), i.e., \( \mathcal{U}_h(T) := \cup\{ T_i \in \mathcal{T}^h; T \cap T_i \neq \emptyset \} \).

Next, we define the discrete residual \( \sigma_h(u_h) \in V^h \) as
\[
(\sigma_h(u_h), \phi)_h = \frac{1}{\gamma} (u_h^{\text{old}}, \phi)_h + (w_h, \phi)_h - \gamma (\nabla u_h, \nabla \phi) \quad \forall \phi \in V^h. \tag{7}
\]

Observe that one can establish for all interior mesh nodes \( p \) that
\[ \sigma_h(u_h; p) = 0 \quad \text{if } |u_h(p)| < 1, \]
see [8, p. 150].

With each inter element side \( e \) we associate the following quantity
\[ [\nabla u_h]_e = \frac{1}{2} (\nabla u_h|_{T_1} - \nabla u_h|_{T_2}) \cdot \nu_{T_2}, \]
i.e., the jump of \( \nabla u_h \) across an inter element boundary, where \( T_1, T_2 \) are two different simplices that contain \( e \).

We have the following local error indicators:
\[
\eta_{T,u}^2 = \left\| h_T \left( \frac{1}{\gamma} u_h^{\text{old}} + w_h - \sigma_h \right) \right\|_{L^2(T)}^2 + \gamma^2 \sum_{e \subset T} \| h_e^{1/2} [\nabla u_h] \|_{L^2(e)}^2,
\]
\[
\eta_{T,w}^2 = \gamma^2 \left\| h_T \frac{u_h - u_h^{\text{old}}}{\tau} \right\|_{L^2(T)}^2 + \sum_{e \subset T} \| h_e^{1/2} [\nabla w_h] \|_{L^2(e)}^2,
\]
\[ \eta^2_{T,\sigma} = \| h_T^2 \nabla \sigma_h(u_h) \|^2, \]
\[ \eta^2_{T,\delta} = \frac{\gamma^2}{\tau^2} h_T^2 \nabla \left( \frac{u_h - u_h^{old}}{\tau} \right)^2. \]

(8)

The indicators \( \eta^2_{T,\sigma}, \eta^2_{T,\delta} \) represent the error due to the use of the inexact inner product \( (\cdot, \cdot)_h \) in (4). The global indicators are defined as the sum of the local error indicators:

\[ \eta_u^2 = \sum_{T \in \mathcal{T}^h} \eta^2_{T,u}, \quad \eta_w^2 = \sum_{T \in \mathcal{T}^h} \eta^2_{T,w}, \]
\[ \eta_\sigma^2 = \sum_{T \in \mathcal{T}^h} \eta^2_{T,\sigma}, \quad \eta_\delta^2 = \sum_{T \in \mathcal{T}^h} \eta^2_{T,\delta}. \]

The following theorem gives a global upper bound for the discretization error.

**Theorem 1.** We have the following a posteriori estimate

\[ \frac{1}{\gamma} \| \nabla e_u \|^2 + \frac{1}{\tau} \| \nabla e_w \|^2 \leq \frac{1}{\gamma^2} \left\{ \frac{C_1}{\tau} (\eta_u^2 + \eta_\sigma^2) + C_2 (\eta_w^2 + \eta_\delta^2) \right\} \]

with the constants \( C_1 \) and \( C_2 \) only depending on the shape regularity of the mesh.

**Proof.** The proof of Theorem 1 can be found in [3].

**Remark 2.** The previous theorem gives an upper bound of the error for both \( u_h \) and \( w_h \). However, in real computations only the estimate of the relative error \( \| \nabla e_u \| / \| \nabla u_h \| \) is of interest. Our experience with numerical computations indicates that the absolute error of \( w_h \) can be much larger compared to the error of \( u_h \). Hence the information provided by the theorem can be unsuitable for practical computations. The theory indicates (see [3]) that the errors \( e_u \) and \( e_w \) are related to the indicators \( \eta_u \) and \( \eta_w \), respectively. Based on this observation, we construct our adaptive algorithm in the next section using only the information provided by the indicator \( \eta_u \).

We use the following quantity to control the time step size \( \tau \):

\[ \eta_\tau = \frac{1}{\gamma} \| u_h - u_h^{old} \|_1. \]

Although there is no rigorous justification for using \( \eta_\tau \) for time step adaptivity, it performed well in our numerical experiments. We note that we only increased the time step size \( \tau \) up to a prescribed threshold \( \tau_{\text{max}} \).

3. Adaptive algorithm and numerical implementation

We use the following adaptive algorithm for the mesh refinement and time step control at every time level. For given tolerances \( \text{TOL} \) and \( \text{TOL}_\tau \), and coarsening/refinement parameters \( \varepsilon_c \) and \( \varepsilon_\tau \) we start with the mesh from the previous time step, i.e., \( \mathcal{T}^h_0 = \mathcal{T}^{h \text{old}} \), and iteratively improve the mesh and the time step size for \( k = 1, \ldots, k_{\text{max}} \) with the following steps:

1. compute \( u_h^k \) and \( \eta_{T,u}^k, \forall T \in \mathcal{T}^h \);
2. if \( \eta_u^k < \text{TOL} \) set \( u_h = u_h^k \) and proceed to the next time step, else continue with step 3;
3. for all \( T \in \mathcal{T}^h_k \), if \( \eta_{T,u}^k > \varepsilon_c \eta_{\text{max}}^k \) mark \( T \) for refinement, if \( \eta_{T,u}^k < \varepsilon_c \eta_{\text{max}}^k \) mark \( T \) for coarsening;
4. refine/coarsen mesh, if \( k < k_{\text{max}} \) proceed with step 1 else proceed with step 5;
5. if \( \eta_\tau > \text{TOL}_\tau \) decrease time step \( \tau := 2^{-1/2} \times \tau \); if \( \eta_\tau < 0.3 \text{TOL}_\tau \) increase time step \( \tau := \min(2^{1/2} \times \tau, \tau_{\text{max}}) \); proceed to the next time level.
Fig. 1. Mesh and \( u_h \) at times \( t = 0, 3 \times 10^{-5}, T = 10^{-4} \).

Fig. 2. Evolution of \( \tau \).

Fig. 3. Degrees of freedom.
We use the notation \( \eta_{\text{max}} = \max_{T \in \mathcal{T}} \eta_{T,0} \). The constants \( \varepsilon_r, \varepsilon_c \) were chosen as 0.6 and 0.05, respectively. The constant \( k_{\text{max}} \) was set to 1 in all our experiments. The parameters that were common for all our experiments are \( \text{TOL} = 0.4 \) and \( \text{TOL}_r = 0.004 \). Further, we used \( \gamma = 1/16\pi \) in the two-dimensional experiments and \( \gamma = 1/12\pi \) in the three-dimensional experiment. The initial time step was set to \( \tau_0 = 10^{-11} \) and the maximum time step was \( \tau_{\text{max}} = 3 \times 10^{-7} \).

We used a Uzawa-multigrid algorithm for the solution of the discrete system of nonlinear algebraic equations arising from (4). For more details on this iterative solver see [2].

4. Numerical experiments

We now demonstrate the practicality and usefulness of our mesh refinement strategy with some exemplary numerical simulations. In all our simulations we assume that phase separation has taken place already, so that the evolution shows the ensuing coarsening. We recall from Section 1 that in the sharp interface limit \( \gamma \to 0 \), our numerical results correspond to solutions of the Mullins–Sekerka equation.
Fig. 6. Degrees of freedom.

Fig. 7. Mesh and $u_h$ at times $t = 0, 10^{-5}, T = 3 \times 10^{-4}$.

Fig. 8. Evolution of $\tau$. 
The first numerical experiment shows the evolution of an ellipse to a circle. The time interval was \([0, T]\) with \(T = 10^{-4}\). The solution and the adaptive mesh are depicted in Fig. 1. The evolutions of the time step size and the number of degrees of freedom are displayed in Figs. 2 and 3, respectively.

In the next experiment we compute the evolution of a square to a circle on the time interval \([0, T]\) with \(T = 2 \times 10^{-5}\). The computed results together with the adaptive mesh can be found in Fig. 4. The evolutions of the time step size and the number of degrees of freedom are depicted in Figs. 5 and 6, respectively.

In the following example we study the evolution of a dumbbell like structure. The results are displayed in Fig. 7. The evolutions of the time step size and the number of vertices are depicted in Figs. 8 and 9.

The last experiment demonstrates the performance of the algorithm for three-dimensional problems. The computational domain was a unit cube, i.e., \(\Omega = (0, 1)^3\). We computed the evolution of a cube with side length 0.5 on the time interval \([0, T]\), with \(T = 3 \times 10^{-5}\). The evolution of the zero level set of the solution and a cut through the mesh at \(x_3 = 0.5\) are depicted in Fig. 10. The evolutions of the time step size and the number of vertices are shown in Figs. 11 and 12, respectively.

The smallest mesh size in all experiments performed here was \(h = \frac{1}{128} \sqrt{2}\). A uniform two-dimensional mesh with \(h = \frac{1}{128} \sqrt{2}\) on a unit square would result in problems with 33,025 unknowns; while for a corresponding uniform mesh in three-dimensional on a unit cube we would have to solve problems with 2,146,689 unknowns. We clearly see from the results, that the proposed adaptive algorithm generates meshes with considerably less unknowns which improves
the efficiency of the computations. We observe, that most of the mesh refinement occurs near the discrete free boundary \( \mathcal{F}_h(u_h) \) between the sets \( \mathcal{C}_h(u_h) \) and \( \mathcal{N}_h(u_h) \), recall (6), i.e., between the sets where \( |u_h| < 1 \) and \( |u_h| = 1 \). The meshes are coarse in \( \mathcal{C}_h(u_h) \) where the solution is constant, i.e., \( u_h \equiv 1 \), which is natural to expect.

**Acknowledgment**

The authors would like to thank the anonymous referee for his comments and suggestions.

**References**

