Commun. Comput. Phys.
doi: 10.4208/cicp.231213.100714a

Vol. **17**, No. 1, pp. 127-145
January 2015

An Efficient Two-Grid Scheme for the Cahn-Hilliard Equation

Jie Zhou[1], Long Chen[2,*], Yunqing Huang[1,3] and Wansheng Wang[4]

[1] *School of Mathematics and Computational Science in Xiangtan University, Xiangtan 411105, China.*
[2] *Department of Mathematics, University of California, Irvine, CA 92697-3875, USA.*
[3] *Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Xiangtan University, Hunan 411105, China.*
[4] *School of Mathematics and Computational Science, Changsha University of Science & Technology, Yuntang Campus, 410114 Changsha, China.*

Received 23 December 2013; Accepted (in revised version) 10 July 2014

Abstract. A two-grid method for solving the Cahn-Hilliard equation is proposed in this paper. This two-grid method consists of two steps. First, solve the Cahn-Hilliard equation with an implicit mixed finite element method on a coarse grid. Second, solve two Poisson equations using multigrid methods on a fine grid. This two-grid method can also be combined with local mesh refinement to further improve the efficiency. Numerical results including two and three dimensional cases with linear or quadratic elements show that this two-grid method can speed up the existing mixed finite method while keeping the same convergence rate.

AMS subject classifications: 65N30, 65N55, 68W25

Key words: Cahn-Hilliard equation, two-grid, adaptivity, mixed method, multigrid.

1 Introduction

We consider numerical solutions of the following Cahn-Hilliard (C-H) equation

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta(-\varepsilon\Delta u + F'(u)) = 0, & x \in \Omega, \\ u(x,0) = u_0(x), & x \in \Omega, \\ \partial_n u = \partial_n(-\varepsilon\Delta u + F'(u)) = 0, & x \in \partial\Omega, \end{cases} \tag{1.1}$$

*Corresponding author. *Email addresses:* xnuzj2004@163.com (J. Zhou), chenlong@math.uci.edu (L. Chen), Huangyq@xtu.edu.cn (Y. Huang), w.s.wang@163.com (W. Wang)

where $\Omega \subset \mathbb{R}^d (d=2,3)$ is a bounded domain, n denotes the unit outward normal of the boundary $\partial\Omega$, $\varepsilon > 0$ is a small but positive constant, and $F(u)$ is a given energy potential. The solution $u(x,t)$ can represent the difference between two concentration, and in most applications $u \in [-1,1]$.

The C-H equation describes the process of phase separation, first introduced by Cahn and Hilliard in the late 1950s [4–6]. Numerical methods for solving the C-H equations provide an important tool on the studying of the dynamics of the C-H equation.

One main difficulty of numerical methods for the C-H equation is the discretization of the fourth order differential operator. For the rectangular domain, finite difference methods or spectral methods [24, 25] can be used. For unstructured grids of a general domain with possible complex geometry structure, finite element methods seems a better choice. However, it is well known that conforming finite-element spaces for fourth order equations is not easy to construct especially in three dimensions. Possible remedy includes non-conforming elements [14, 49] or discontinuous Galerkin methods [1, 39, 43]. Here we consider mixed finite element methods (MFEs) [15, 17, 18], which can give the numerical approximation not only to the concentration u but also to the chemical potential $w = \phi(u) - \epsilon\Delta u$. The price of using the mixed formulation is that the nonlinear system is in the saddle point form which is in general bigger and harder to solve than the symmetric positive definite system obtained by a conforming or non-conforming discretization. In our two grid method to be presented later, we shall overcome this shortcoming.

Another focus of developing accurate and efficient numerical scheme is the energy stable time discretization. It is shown that the implicit Euler method applied to the C-H equation is unconditionally stable and obey the energy law [1]. For full implicit schemes, however, a nonlinear system should be solved at every time step which is usually ten times slower than the first order semi-implicit schemes for which fast Poisson solvers can be applied, see [35]. The semi-implicit scheme (F(u) being explicit and $\Delta^2 u$ being implicit) is conditionally energy stable but with a restrictive constraint for the time step. To remove the requirement of small time step, stabilization [8, 35] or convex splitting scheme [16,27,40] can be applied. Other energy stable schemes can be found in [40,43,49]. Especially the stability and the convergence of mixed finite element methods for the C-H equation was investigated in [17, 18]. We shall not explore more on the stability in this paper.

Instead we are interested in efficient ways to improve the accuracy of numerical approximations. In this paper, we shall apply the two-grid method [44,45] to the C-H equation. The main idea of two-grid methods is solving the C-H equation using a stable mixed finite element method on a coarse grid first, then solving two Poisson equations on the fine grid. The nonlinear system on the coarse grid, because of the small size of the system, can be solved without too much computational cost. On the fine gird, we only need to solve two decoupled Poisson equations, which can be solved efficiently by off-the-shelf solvers such as the multigrid solvers. We shall demonstrate that the two-grid method can achieve the same convergence rate as the standard implicit mixed finite element method on the fine grid but with less computational time.

The two-grid method was first introduced by Xu [44, 45] and then applied to many problems, such as nonlinear elliptic equations [46], nonlinear parabolic equations [11,12], Navier-Stokes equations [23,29], eigenvalue problems [26,47,51] and also Maxwell equation [50]. It is worth noting that our two-grid method is different from the classical two-grid method for evolution problems [11, 32, 37]. We do not apply the fine grid post-processing at every time step. Instead we apply the fine grid post-processing at the time step when a better resolution is needed. In this sense, our two-grid becomes the postprocessing method developed in [2, 19–22, 33, 38, 48].

The outline of this paper is the following. In the next section, we introduce the mixed finite element method for the C-H equation. In Section 3, we propose our two-grid method including adaptive version and provide the convergence result of the method. We present several numerical experiments in the last section. Throughout the paper, we shall use standard notation for Sobolev spaces.

2 Mixed finite element method for Cahn-Hilliard equation

2.1 Energy law

For model equation (1.1), the functional of u

$$E(u)=\int_\Omega \left(\varepsilon|\nabla u|^2+F(u)\right)dx,$$

is called free energy. Notice that the free energy includes two distinct parts, the bulk energy $F(u)$ and interfacial energy $\varepsilon|\nabla u|^2$. The bulk energy does not depend on the spatial gradient of the phase variable, while the interfacial energy does. Further, we have

$$\frac{dE(u)}{dt}=-\int_\Omega \left|\nabla(-\varepsilon\Delta u+F'(u))\right|^2 dx. \tag{2.1}$$

From (2.1), we know that for any $t_1<t_2$, $E(u(t_1))\geq E(u(t_2))$. This can explain that the evolution of the phase variable obey the energy minimization rule. It will be beneficial that this energy minimization is faithfully preserved in the numerical methods.

2.2 Spatial discretization

We focus on the mixed finite element methods in this paper. Let $w=-\varepsilon\Delta u+F'(u)$. The mixed variational formulation of (1.1) is: find (u,w) such that

$$\begin{cases} (\partial_t u,q)+(\nabla w,\nabla q)=0, & \text{for all } q\in H^1(\Omega),\\ (w,v)-\varepsilon(\nabla u,\nabla v)-(F'(u),v)=0, & \text{for all } v\in H^1(\Omega). \end{cases} \tag{2.2}$$

Let $\mathcal{T}_h$ be a quasi-uniform triangulation of Ω. Here h stands for the mesh size. Denote by T an element of $\mathcal{T}_h$ and by S_h the standard Lagrange finite element space defined on

$\mathcal{T}_h$. More precisely, for an integer $r \geq 1$,

$$S_h = \left\{ v_h \in C(\Omega) \,\middle|\, v_h|_T \in P_r, \forall T \in \mathcal{T}_h \right\},$$

where P_r is the space of polynomials of degree at most r. Then we can write the semi-discrete problem of (2.2) as follows: find (u_h, w_h) such that

$$\begin{cases} (\partial_t u_h, q_h) + (\nabla w_h, \nabla q_h) = 0, & \text{for all } q_h \in S_h, \\ (w_h, v_h) - \varepsilon(\nabla u_h, \nabla v_h) - \big(F'(u_h), v_h\big) = 0, & \text{for all } v_h \in S_h. \end{cases} \tag{2.3}$$

2.3 Temporal discretization

We consider the implicit Euler scheme. Given u_h^0, which can be chosen as the interpolation of u_0 in S_h, and a time step $0 < \tau \ll 1$, for $n \geq 0$, find (u_h^{n+1}, w_h^{n+1}) such that

$$\begin{cases} \left(\dfrac{u_h^{n+1} - u_h^n}{\tau}, q_h \right) + \left(\nabla w_h^{n+1}, \nabla q_h \right) = 0, & \text{for all } q_h \in S_h, \\ (w_h^{n+1}, v_h) - \varepsilon(\nabla u_h^{n+1}, \nabla v_h) - \left(F'(u_h^{n+1}), v_h \right) = 0, & \text{for all } v_h \in S_h. \end{cases} \tag{2.4}$$

This formulation is unconditionally energy stable and uniquely solvable [1], and the corresponding error estimate can be found in [17]. In this formulation, we should solve a nonlinear system at every time step. In the two-grid method introduced below, however, we only solve this nonlinear system on a coarse grid with relatively large mesh size.

3 Two-grid algorithm

Let $\mathcal{T}_H$ and $\mathcal{T}_h$ be two triangulations of the domain Ω with different mesh size H and h and $H > h$. Usually $\mathcal{T}_h$ is a refinement of $\mathcal{T}_H$. Their associated finite element spaces are denoted by S_H and S_h, respectively. Denote $S_h^0 = \{u_h \in S_h \,|\, \int_\Omega u_h dx = 0\}$. Based on S_H and S_h, we present the following two-grid method (Algorithm 1) for the C-H equation.

This two-grid algorithm consists of two steps. In the first step, it is a nonlinear system discretized by the mixed finite element method and will be solved by Newton's method. Because this nonlinear system is discretized on the coarse grid, its computation cost is neglectful. Most of the computation work is on the fine grid in the second step.

For the problem in the second step, it consists of two decoupled Poisson equations, for which a lot of fast Poisson solvers, e.g., multigrid methods, can be used. Therefore this two-grid method can save a lot of computation work compared with standard mixed finite element methods. Indeed, in [38], we have proved that the accuracy of the approximation of our two-grid method is of optimal order.

Algorithm 1 Two-grid method

1. Given u_H^0, which can be chosen the interpolation of u_0 in the S_H. Solve the following problem on the coarse grid $\mathcal{T}_H$ for $n=0,1,\cdots,K$.

$$\begin{cases} \left(\frac{u_H^{n+1}-u_H^n}{\tau},q_H\right)+\left(\nabla w_H^{n+1},\nabla q_H\right)=0, & \text{for all } q_H\in S_H, \\ (w_H^{n+1},v_H)-\varepsilon\left(\nabla u_H^{n+1},\nabla v_H\right)-\left(F'(u_H^{n+1}),v_H\right)=0, & \text{for all } v_H\in S_H. \end{cases} \tag{3.1}$$

2. Solve Poisson equations on the fine grid: find $(u^{h,K},w^{h,K})\in(S_h^0)^2$, such that

$$\begin{cases} (\nabla w^{h,K},\nabla q_h)=-\left(\frac{\partial u_H}{\partial t}(t_K),\nabla q_h\right), & \text{for all } q_h\in S_h^0, \\ \varepsilon(\nabla u^{h,K},\nabla v_h)=\left(w_H^K-F'(u_H^K),\nabla v_h\right), & \text{for all } v_h\in S_h^0. \end{cases} \tag{3.2}$$

Theorem 3.1 ([38]). *Let $(u^{h,K},w^{h,K})\in(S_h^0)^2$ be the two-grid approximation to $(u(t_K),w(t_K))$. For linear element $(r=1)$ and quadratic element $(r=2)$, there exist a positive constant C such that*

$$\left\|u(t_K)-u^{h,K}\right\|_1+\left\|w(t_K)-w^{h,K}\right\|_1\leq C(h^r+H^{2r}). \tag{3.3}$$

We give several remarks to discuss variants of this two-grid methods.

Remark 3.1. We apply the Newton's method to solve the nonlinear system (3.1). There are some other robust and efficient approaches to solve this nonlinear system. For example, the full approximation scheme (FAS) [3] is one of the most popular nonlinear multigrid methods for large-scale problems which might be more robust than Newton's method. In the two-grid method, however, the size of the nonlinear problem is small, and for time dependent problem, the solution in the previous time step is always a good initial guess to be used in the Newton's iteration. Therefore we choose the Newton's method instead of nonlinear multi-grid methods.

Remark 3.2. The scheme (3.1) is fully implicit in time. There are some other energy stable discretization, for example, semi-implicit scheme [34], and convex splitting scheme [1, 36]. These schemes can be also used as the coarse grid solver as long as they are energy stable and second order accurate in space.

Remark 3.3. Choosing a suitable time scale is also crucial for the simulation. The choice of the time step depends on three factors: stability, accuracy, and dynamics of the system. The severe constraint of the time step is from the last one. Namely the time step should be small enough [16] to accurately capture the dynamics of the system, especially in the very

beginning. Our numerical experiments show that second order implicit scheme (Crank-Nicolson) is not helpful for choosing a larger time step size. On the other hand, this is not so restrictive for our two-grid method since, again, the coarse grid problem is of small size, the cost of simulation of a small size nonlinear dynamics problem is acceptable.

To further improve the approximation, we propose an adaptive two-grid method. Because the C-H equation describes the process of phase separation, there are lots of interfaces between two components. The adaptive technology is one of the most efficient methods to capture the interface. We combine adaptivity through local mesh refinement and two-grid methods to propose the following adaptive two-grid algorithm.

Algorithm 2 Adaptive two-grid algorithm

1. Given u_H^0, which can be chosen the interpolation of u_0 in the S_H. Solve the following problem on the coarse grid $\mathcal{T}_H$, $n=0,1,\cdots,K$.

$$\begin{cases} \left(\frac{u_H^{n+1}-u_H^n}{\tau},q_H\right)+\left(\nabla w_H^{n+1},\nabla q_H\right)=0, & \text{for all } q_H\in S_H, \\ \left(w_H^{n+1},v_H\right)-\varepsilon\left(\nabla u_H^{n+1},\nabla v_H\right)-\left(F'(u_H^{n+1}),v_H\right)=0, & \text{for all } v_H\in S_H. \end{cases} \tag{3.4}$$

2. Locally refine $\mathcal{T}_H$ then get $\mathcal{T}_h$ in the following way.

   Step 0 Let $i=0$, and denote $\mathcal{T}_{h,0}=\mathcal{T}_H$;

   Step 1 Estimate. Compute an error indicator η_0 on $\mathcal{T}_{h,0}$;

   Step 2 Mark. Select the minimal elements $\tilde{\mathcal{T}}_{h,i}$ on $\mathcal{T}_{h,i}$ such that

$$\sum_{T\in\tilde{\mathcal{T}}_{h,i}}\eta_i|_T\le\theta\eta_i, \quad \text{for some } \theta\in(0,1).$$

   Step 3 Refine. Bisect the marked element $\tilde{\mathcal{T}}_{h,i}$ and get the mesh $\mathcal{T}_{h,i+1}$;

   Step 4 Estimate. Interpolate η_i to $\mathcal{T}_{h,i+1}$, and denote by η_{i+1};

   Step 5 If (i< maximal iteration steps) then
   $i=i+1$ and go to Step 2;
   else $\mathcal{T}_h=\mathcal{T}_{h,i+1}$ and exit.

3. Solve Poisson equations on the fine grid $\mathcal{T}_h$: find $(u^{h,K},w^{h,K})\in(S_h^0)^2$, such that

$$\begin{cases} \left(\nabla w^{h,K},\nabla q_h\right)=-\left(\frac{\partial u_H}{\partial t}(t_K),\nabla q_h\right), & \text{for all } q_h\in S_h^0, \\ \varepsilon\left(\nabla u^{h,K},\nabla v_h\right)=\left(w_H^K-F'(u_H^K),\nabla v_h\right), & \text{for all } v_h\in S_h^0. \end{cases} \tag{3.5}$$

There are three steps in this adaptive two-grid algorithm. We should solve the C-H equation on the coarse grid with mixed finite element first, then refine the mesh $\mathcal{T}_H$ locally and get the mesh $\mathcal{T}_h$. In the last step, we solve two decoupled Poisson equations in the adaptive mesh $\mathcal{T}_h$.

Algorithm 2 differs from Algorithm 1 in the way of generating the fine grid. Thus the key to this adaptive algorithm is the error indicator η_0. Several reliable and efficient error estimators for C-H equation have been constructed; see, e.g. [18, 31]. Based on the software *i*FEM [9], we use the function `estimaterecovery` to compute a recovery type error indicator η_0:

$$\eta_0 = \sum_{\tau \in \mathcal{T}_{h,0}} \eta_{\tau,0}, \qquad \eta_{\tau,0} = \int_\tau |\nabla(R(\nabla u_H))| \, dxdy.$$

The recovery operator R is simply an averaging operator from piecewise constant to piecewise linear function space.

4 Numerical experiments

In this section, we will present some numerical examples to illustrate the efficiency of our algorithm. All examples are implemented in MATLAB using the software library *i*FEM [9] and tested in a laptop with configuration Mac OS 10.8.5, 4G memory, Intel core i5.

In the implementation, all integrals are computed using a quadrature formula which is exact for polynomials of degree 4 (linear element) or 8 (quadratic element). Therefore the integral is exact for polynomial nonlinearity. Our theoretical result, i.e., Theorem 3.1 holds when the nonlinear integral is computed exactly. For other terms, e.g. nonzero source f in Example 4.1, we still use the same quadrature formula such that the integral error will not affect the accuracy of the solution.

4.1 Convergence test

Example 4.1. We test the convergence of the method first. Let us consider the following problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta(-\varepsilon \Delta u + F'(u)) = f, & x \in \Omega, \\ u(x,0) = u_0(x), & x \in \Omega, \\ \dfrac{\partial u}{\partial n} = \dfrac{\partial \Delta u}{\partial n} = 0, & x \in \partial\Omega. \end{cases} \tag{4.1}$$

The computational domain is $\Omega = (0,1)^2$, $\varepsilon = 0.01$, and the exact solution and the nonlinear function $F(u)$ is $u = e^{(-2t)} \sin(\pi x)^2 \sin(\pi y)^2$, $F(u) = (u^2-1)^2/4$, respectively. The initial condition u_0 and the source f can be chosen accordingly.

Table 1: Mixed finite element methods using P1 element and implicit Euler method for Example 4.1, $\tau=$1.0e-5, $t=0.01$.

h	$\|\|u-u_h\|\|_1$	ration	$\|\|w-w_h\|\|_1$	ration
1/16	2.787499e-01		3.589032e-01	
1/32	1.393978e-01	2.00	1.815490e-01	1.98
1/64	6.969114e-02	2.00	9.104220e-02	1.99

Table 2: Two-grid method using P1 element for Example 4.1, $\tau=$1.0e-5, $t=0.01$.

H	h	$\|\|u-u^h\|\|_1$	ration	$\|\|w-w^h\|\|_1$	ration
1/8	1/64	1.617031e-01		1.791149e-01	
1/16	1/256	4.022071e-02	4.02	4.686222e-02	3.82
1/32	1/1024	1.004711e-02	4.00	1.176510e-02	3.98

Table 3: Mixed finite element method using P2 element for Example 4.1, $\tau=$1.0e-5, $t=0.01$.

h	$\|\|u-u_h\|\|_1$	ration	$\|\|w-w_h\|\|_1$	ration
1/16	1.151231e-02		2.534285e-02	
1/32	2.900469e-03	3.97	6.470871e-03	3.92
1/64	7.299357e-04	3.97	1.627038e-03	3.98

Table 4: Two-grid method using P2 element for Example 4.1, $\tau=$1.0e-5, $t=0.01$.

H	h	$\|\|u-u^h\|\|_1$	ration	$\|\|w-w^h\|\|_1$	ration
1/4	1/16	1.105632e-01		5.550984e-02	
1/8	1/64	1.024348e-02	10.79	4.314935e-03	12.86
1/16	1/256	7.635561e-04	13.42	2.848867e-04	15.15

We denote by u_h and u^h the mixed finite element and the two-grid approximation, respectively. We test our two-grid method, using linear element (P1) and quadratic element (P2). We set the time step $\tau=10^{-5}$ in order to test the order of convergence in space. Tables 1-2 present the numerical results at $t=0.01$ obtained by the mixed finite element and the two-grid method respectively using P1 element. Tables 3-4 present the numerical results using P2 element. From the above tables, we can find that the performance is consistent with our theoretical result Theorem 3.1. That is both the two-grid method and the mixed method have the same convergence rate and comparable accuracy.

On the coarse grid, we need to solve a nonlinear equation. The solver we used is the standard Newton's method. In each Newton's iteration, we use the MATLAB built-in direct solver to solve the linearized system. On the fine grid, we need to solve two Poisson equations with Neumann boundary condition. We use MGCG solver implemented in *i*FEM [9]. The MGCG (Multi-Grid based Conjugate Gradient) method is a PCG method

Table 5: CPU time and MGCG iteration in the fine grid with P1 element, $\tau=1.0\text{e-}5$, $t=1.0\text{e-}5$.

H	h	$\|u-u^h\|_1$	time(s)	MGCG iter	h	$\|u-u_h\|_1$	time(s)
1/4	1/16	4.037713e-01	0.297s	11	1/16	2.800709e-1	0.197s
1/8	1/64	1.085241e-01	0.527s	11	1/64	7.036322e-2	0.601s
1/16	1/256	2.681232e-02	4.511s	11	1/256	1.759125e-2	10.521s

Table 6: CPU time and MGCG iteration in the fine grid with P2 element, $\tau=1.0\text{e-}5$, $t=1.0\text{e-}5$.

H	h	$\|u-u^h\|_1$	time(s)	MGCG iter	h	$\|u-u_h\|_1$	time(s)
1/4	1/16	7.522568e-02	0.317s	11	1/16	1.161568e-02	0.27s
1/8	1/64	3.366204e-03	1.527s	11	1/64	7.362690e-04	2.60s
1/16	1/256	1.904476e-04	9.511s	11	1/256	4.596467e-05	87.01s

with a V-cycle MG as the preconditioner. Each V-cycle includes one step of pre-smoothing and post-smoothing.

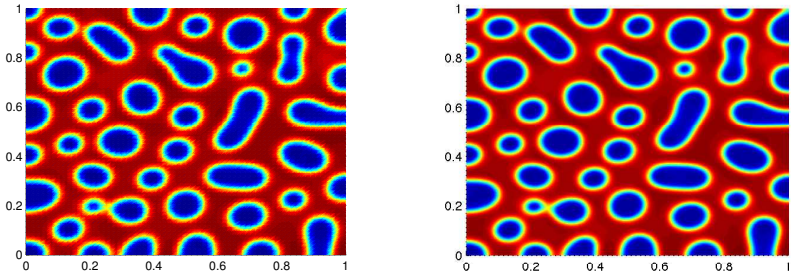
From Tables 5-6, we can observe the number of iterations of the MGCG solver is independent of the mesh size, nearly 11 iterations, to reach the stopping tolerance 1.0e-8. Therefore the two-grid method can save CPU time compared with mixed finite element methods especially for P2 element since the direct solver in MATLAB is less efficient for high order elements. In Table 5, solving the nonlinear system on $h=1/16$ will take 0.197 seconds and the overall cost for the two-grid method with $H=1/16$, $h=1/256$ is 4.511 seconds. That is the nonlinear solver in the coarse grid only takes 5% of the total cost. Similarly Table 6 shows for P2 element, the cost is $0.27s$ vs $9.511s$ and the ratio is 3%. Therefore the cost of the nonlinear solver in the coarse grid is neglectable.

One can apply the nonlinear multi-grid to the C-H equation directly. In [1], the authors propose a nonlinear multigrid solver for this nonlinear system. In their experiments, nearly 10 iterations with one step pre-smoothing and post-smoothing (nonlinear Gauss-Seidel iteration) are also needed to reach the same stopping tolerance 1.0e-8. In our examples, we also need nearly 10 iterations on the fine grid. But this is a linear system for which MG is relatively easier to implement than nonlinear multigrid method and many efficient Poisson solvers are already available.

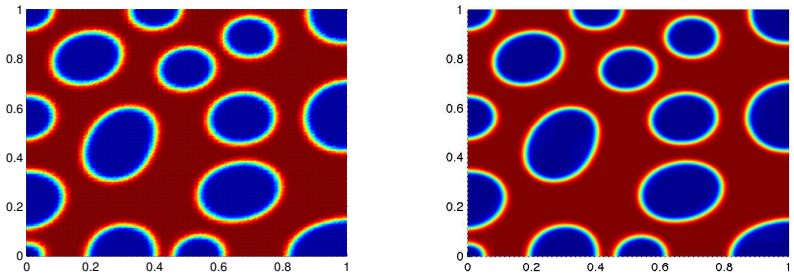
4.2 Spinodal decomposition

Example 4.2. Let us consider the following problem [52]

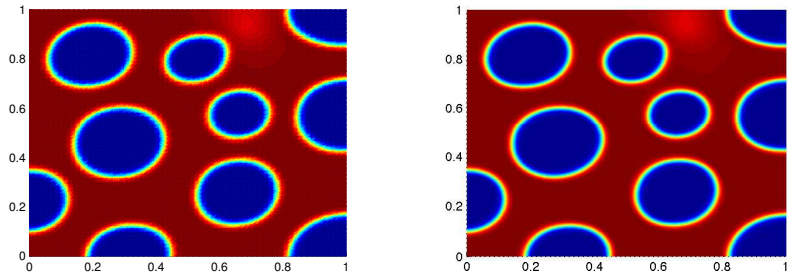
$$\begin{cases} \dfrac{\partial u}{\partial t}-\Delta(-\varepsilon\Delta u+F'(u))=0, & x\in\Omega, \\ u(x,0)=u_0(x), & x\in\Omega, \\ \dfrac{\partial u}{\partial n}=\dfrac{\partial\Delta(-\varepsilon\Delta u+F'(u))}{\partial n}=0, & x\in\partial\Omega. \end{cases} \tag{4.2}$$



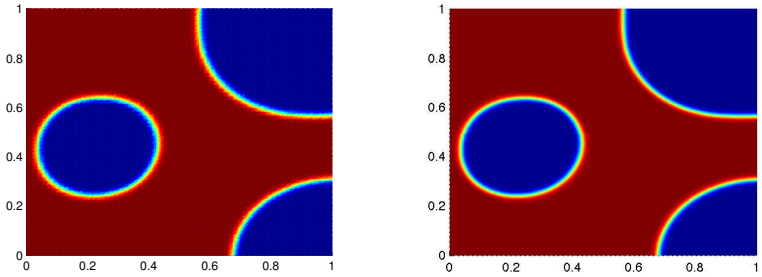
(a) $t=$2.5e-4 (b) $t=$2.5e-4



(c) $t=$1.5e-3 (d) $t=$1.5e-3



(e) $t=$4.0e-3 (f) $t=$4.0e-3



(g) $t=$5.0e-1 (h) $t=$5.0e-1

Figure 1: Comparison of the coarse grid approximation and the fine grid approximation in the two-grid method for Example 4.2. The left figure are computed on the coarse grid, the right figure are computed by the two-grid method in a fine grid. The coarse grid size $H=1/64$ and the fine grid size $h=1/256$. The number of elements on the coarse grid are about 8,192, and on the fine grid are about 131,072.

The computational domain is $\Omega=(0,1)^2$, $\varepsilon=0.01$, $F(u)=100u^2(1-u)^2$ and a random initial value around 0.63 is chosen.

In this example, we apply the two-grid method using the linear element and the time step $\tau=5.0\text{e-}6$. The coarse grid size is $H=1/64$ and the fine grid one is $h=1/256$.

Phase decomposition is shown in the following figures. Figures on the left are computed by the mixed finite element directly on the coarse grid, and figures on the right are solutions in the fine grid computed by our two-grid method. The dynamics in coarse and fine grids are the same but the right figures are with better resolution. From this point of view, our two-grid method can be regard as a postprocessing method to improve the accuracy of the solution on the coarse grid. Fig. 2 shows the free energy associated to u^h decreases sharply in the beginning, then changes slowly, but always decreasing in time.

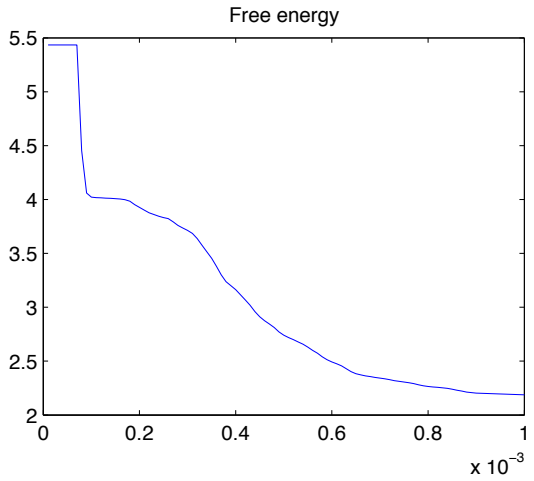


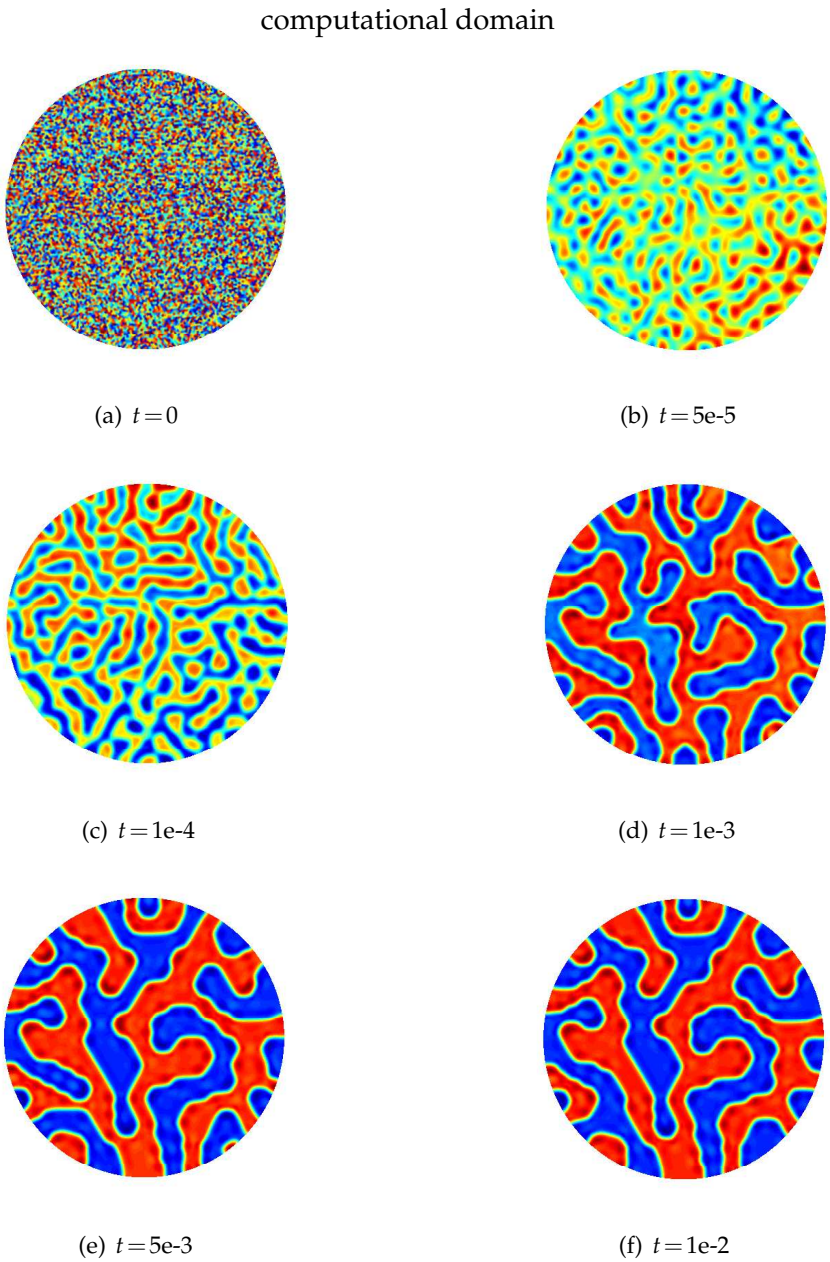
Figure 2: Free energy for Example 4.2, $E(u^h)=\int_\Omega(\varepsilon|\nabla u^h|^2+F(u^h))dx$.

Remark 4.1. In this example, the longest time scale is $t=0.5$. We believe the two-grid algorithm works for longer time scale. We can understand this point from the two-grid algorithm. The fine grid solver can be thought as a post-processing to improving the resolution of the coarse grid solution. The dynamics is captured in the coarse grid.

Example 4.3. Let us consider the following problem [1]

$$\begin{cases} \dfrac{\partial u}{\partial t}-\Delta\Big(-\varepsilon\Delta u+\dfrac{1}{\varepsilon}F'(u)\Big)=0, & x\in\Omega, \\ u(x,0)=u_0(x), & x\in\Omega, \\ \dfrac{\partial u}{\partial n}=\dfrac{\partial\Delta(-\varepsilon\Delta u+\frac{1}{\varepsilon}F'(u))}{\partial n}=0, & x\in\partial\Omega. \end{cases} \tag{4.3}$$

The computational domain is a disk with diameter 2, and centered (0,0). The parameter $\varepsilon=0.0125$, $F(u)=(u^2-1)^2/4$, and a random initial value around 1.0e-3 is chosen.



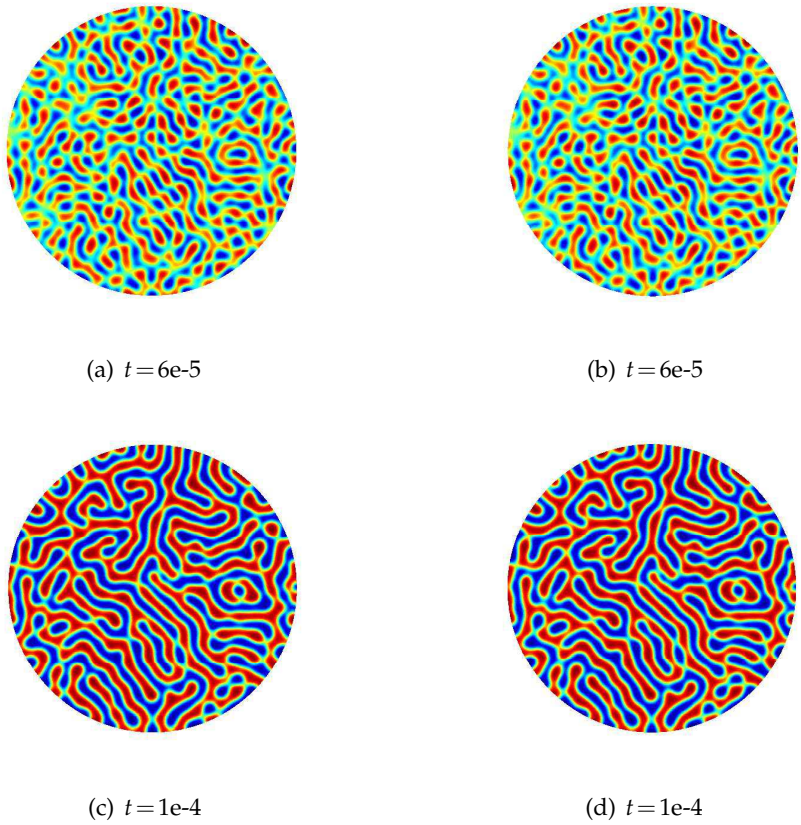
(a) $t=0$ (b) $t=$5e-5

(c) $t=$1e-4 (d) $t=$1e-3

(e) $t=$5e-3 (f) $t=$1e-2

Figure 3: Spinodal decomposition for Example 4.3: $\tau=$1.0e-6, $\varepsilon=0.0125$. The coarse grid contains 2,198 elements, and the fine grid contains 413,696 elements. The figures display two-grid approximations on the fine grid.

We simulate spinodal decomposition using our two-grid Algorithm 1. The coarse grid includes about 2000 elements, and the fine grid about 413,696 elements. The time step is $\tau=$1.0e-6. The computed solution is displayed in Fig. 3 which is consistent with



(a) $t=$6e-5 (b) $t=$6e-5

(c) $t=$1e-4 (d) $t=$1e-4

Figure 4: Spinodal decomposition for Example 4.3: $\tau=$1.0e-6, $\varepsilon=0.0125$. The coarse grid contains 25,856 elements, and the fine grid contains 103,424 elements. The figures display two-grid approximations on the fine grid. The left figures are implemented by using the convex splitting method as the coarse grid solver, the right figures are implemented by using the full implicit method as the coarse grid solver.

results in [41]. Fig. 5 shows that the free energy associated to u^h is decay in time. Fig. 4 shows the find gird solutions using the convex splitting method [1, 36] (left figures) and implicit Euler method (right figures) as the coarse grid solver, receptively. They are almost identical in the sense that the difference is 1.0e-6 in L^2 norm.

Example 4.4. Consider the following three dimensional problem [42]

$$\begin{cases} \dfrac{\partial u}{\partial t}-\Delta(-\varepsilon\Delta u+F'(u))=0, & x\in\Omega, \\ u(x,0)=u_0(x), & x\in\Omega, \\ \dfrac{\partial u}{\partial n}=\dfrac{\partial\Delta(-\varepsilon\Delta u+F'(u))}{\partial n}=0, & x\in\partial\Omega. \end{cases} \tag{4.4}$$

The computational domain $\Omega=[0,1]^3$.

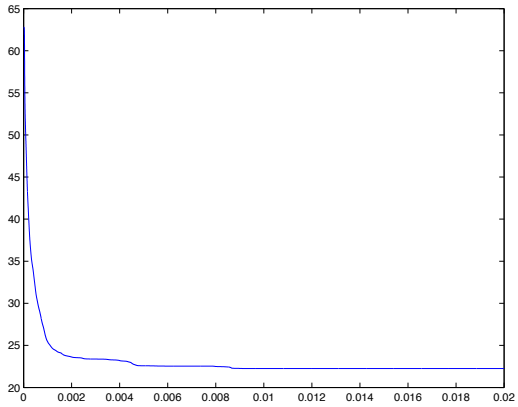


Figure 5: Free energy for Example 4.3, $E(u^h)=\int_\Omega(\varepsilon|\nabla u^h|^2+F(u^h))dx$.

In this example, $\varepsilon=1$, $F(u)=200[u\ln u+(1-u)\ln(1-u)+3u(1-u)]$, the time step $\tau=1.0\text{e-}6$. We use the coarse grid diameter $H=1/16$, including 32,768 elements, and use the full implicit scheme, the fine grid diameter $h=1/64$, including 196,608 elements. We present the spinodal decomposition in Fig. 6. This example shows that the two-grid is feasible in three dimensions. Table 7 shows the performance of MGCG in three dimensions is also stable requiring 14 iterations to reach the stoping tolerance 1.0e-8.

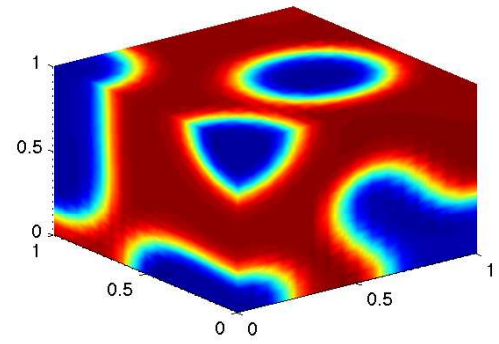
Table 7: Performance of MGCG in three dimensions on a fine grid with $h=1/64$.

t	5.0e-5	1.0e-3	2.0e-3
iterations	14	14	14
CPU time	2.6s	2.5s	2.6s

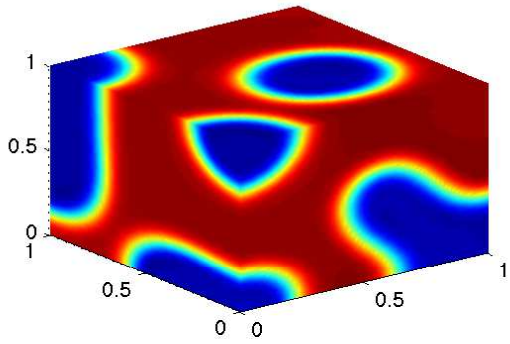
4.3 Adaptive two-grid test

Example 4.5. We use the same Example 4.2.

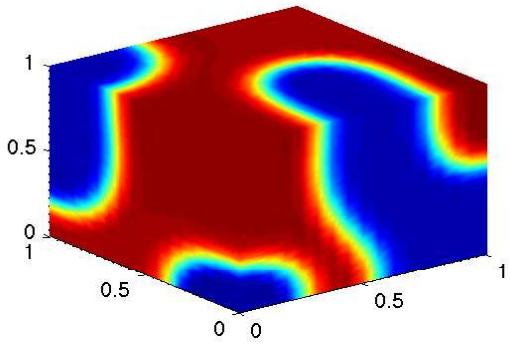
In this test, we apply the adaptive two-grid method using recovery type error estimator and set $\theta=0.5$ and maximal refinement iteration is $|\log H|$. The results are shown in Fig. 7. In order to show the effectiveness of the adaptive two-grid, we compare this example with Example 4.2 to which the standard two-grid method is applied. From the left figures, we can tell that error indicator can catch the interface of the two components accurately, and the number of elements are around one fourth of that in Example 4.2. Furthermore, we find phase positions are consistent with figures in Example 4.2. The MGCG iterations on the fine adaptive gird are about 12 to reach the stoping tolerance 1.0e-8. Considering the fact the solver in the fine grid is the most time consuming part, we conclude that our adaptive two-grid can reach a similar result with less computation work.



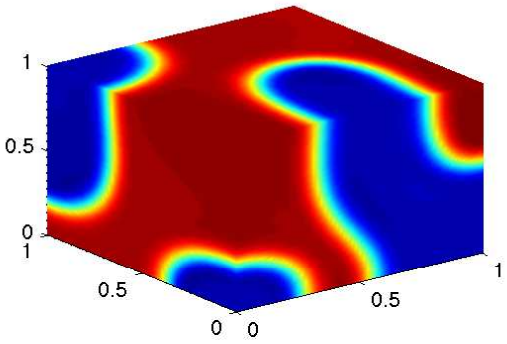
(a) $t=$5.0e-4



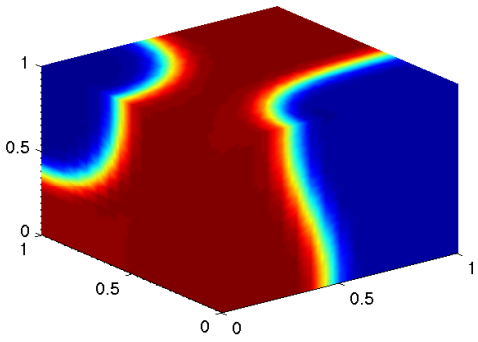
(b) $t=$5.0e-4



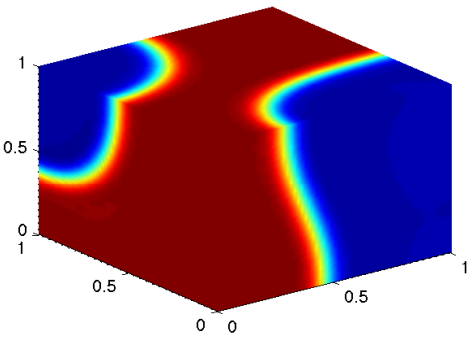
(c) $t=$1.0e-3



(d) $t=$1.0e-3

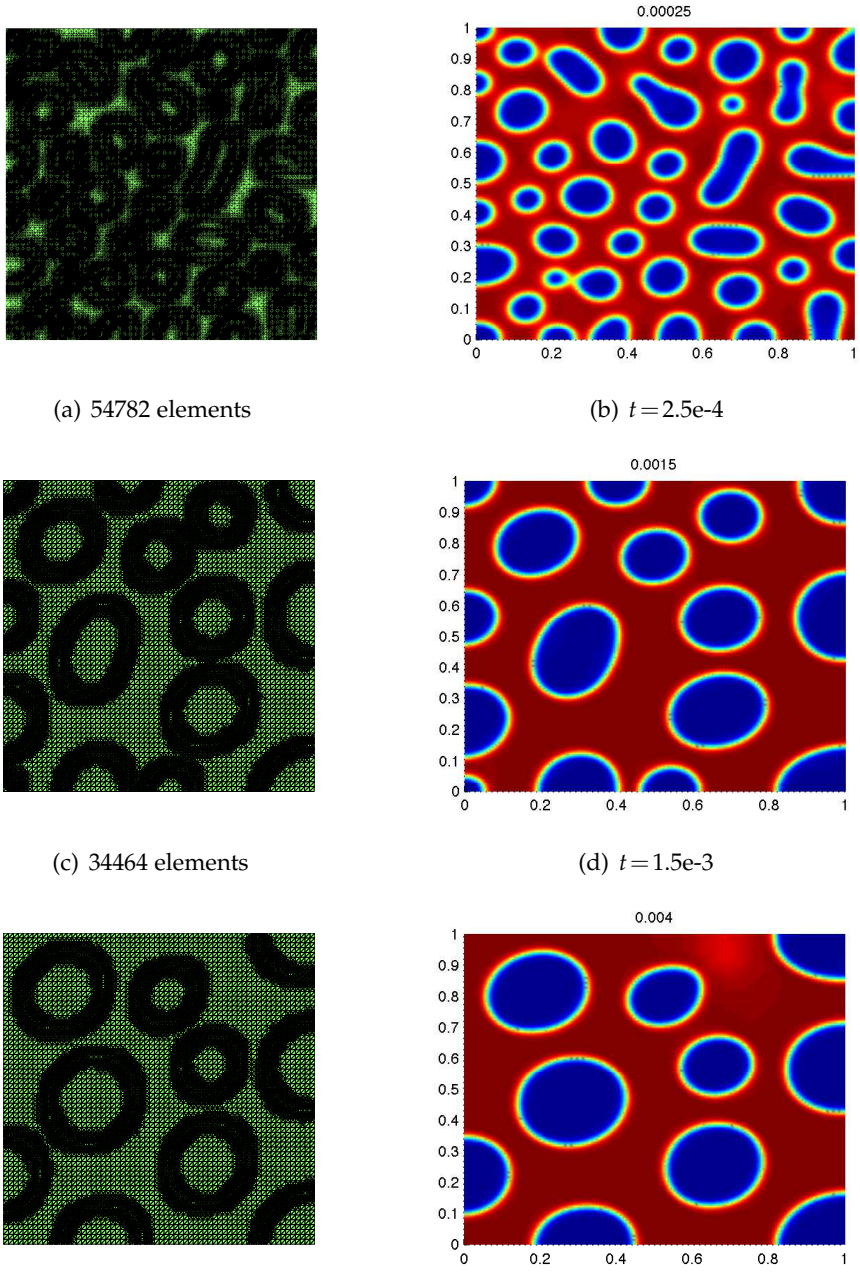


(e) $t=$2.0e-3



(f) $t=$2.0e-3

Figure 6: In Example 4.4, spinodal decomposition in three dimensions, time step $\tau=$1.0e-6. The left figures are computed by the mixed finite element on a coarse grid with 4,913 degree of freedom. The right figures are computed by the two-grid method with a fine grid with 274,625 degree of freedom.



(a) 54782 elements (b) $t=2.5\text{e-}4$

(c) 34464 elements (d) $t=1.5\text{e-}3$

(e) 30305 elements (f) $t=4.0\text{e-}3$

Figure 7: Example 4.5 using Algorithm 2: $\varepsilon=0.01$, $\theta=0.5$, maximal refinement steps $|\log H|$.

5 Conclusion

In this paper, we have proposed a two-grid method for the C-H equation. The highlight of this method is that it can save much computation work while keep the same conver-

gence rate as the standard mixed finite element method. In the numerical experiments, we have presented several examples to illustrate the effectiveness of this method. We have also proposed an adaptive two-grid method based on the local mesh refinement.

Acknowledgments

The research of J. Zhou was supported by Graduate innovation program of Hunan province in China (Grant No. CX2012B240) and China Scholarship Council and Natural Science Foundation of China (Grant No. 11201159). L. Chen was supported by NSF Grant DMS-1115961, and in part by Department of Energy prime award # DE-SC0006903. The research of Y.Q. Huang was partially supported by the National Natural Science Foundation of China (Grant No. 91430213), Program for Changjiang Scholars and Innovative Research Team in University (Grant No. IRT1179), International Science and Technology Cooperation Program of China (Grant No. 2010DFR00700). The research of W.S. Wang was supported by Ky and Yu-Fen Fan Fund Travel Grant from the AMS, the National Natural Science Foundation of China (Grant No. 11001033, 11371074), the Natural Science Foundation for Distinguished Young scholars in Hunan Province, China (Grant No. 13JJ1020), and the Research Foundation of Education Bureau of Hunan Province, China (Grant No. 13A108).

We would like to thank the reviewer for helpful comments and suggestions, especially pointing out the alternative coarse grid discretization such as the convex splitting, and the discussion on the time scale etc.

References

[1] A. C. Aristotelous, O. Karakshian and S. M. Wise, A mixed discontinuous Galerkin, convex splitting scheme for a modified Cahn-Hilliard equation and an efficient nonlinear multigrid solver, Discrete and Continuous Dynamical System-Series B, 18(9):2211–2238, 2013.

[2] B. Ayuso, B. Garcia-Archilla and J. Novo, The postprocssed mixed finite element method for the Navier-Stokes equations, SIAM J. Numer. Anal., 43:1091–1111, 2005.

[3] A. Brandt, Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, St. Augustin: Gesellschaft für Mathematik und Datenverarbeitung, 1984.

[4] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system I: Interfacial free energy, J. Che. Phys., 28:258–267, 1958.

[5] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system II: Thermodynamic basis, J. Che. Phys., 30:1121–1124, 1959.

[6] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system III: Nucleation in a two-component incompressible fluid, J. Che. Phys., 31(3):688–699, 1959.

[7] C. Collins, J. Shen and S. M. Wise, An efficient, energy stable scheme for the Cahn-Hilliard-Brinkman system, in press, 2013.

[8] F. Chen, J. Shen, Efficient energy stable schemes with spectral discretization in space for anisotropic Cahn-Hilliard systems, Comm. Comp. Phys., 13(5):1189–1208, 2013.

[9] L. Chen, *i*FEM: An Integrated Finite Element Methods Package in MATLAB, Technical Report, University of California at Irvine, (2009).

[10] J. deFrutos, B. Garcia-Archilla and J. Novo, The postprocessed mixed finite element method for the Navier-Stokes equations: Improved error bounds, SIAM J. Numer. Math., 46 (2007), 201-230.
[11] C. N. Dawson, M. F. Wheeler, Two-grid methods for mixed finite element approximations of nonlinear parabolic equations, Cont. Math., 180:191–191, 1994.
[12] C. N. Dawson, M. F. Wheeler, A two-grid finite difference scheme for nonlinear parabolic equations, SIAM J. Numer. Anal., 35(2):435–452, 1998.
[13] C. M. Elliott, S. Zheng, On the Cahn-Hilliard equation, Arch. Rational Mech. Anal., 96(4):339–357, 1986.
[14] C. M. Elliott, A. Donald, A nonconforming finite-element method for the two-dimensional Cahn-Hilliard equation, SIAM J. Numer. Anal., 26(4):884–903, 1989.
[15] C. M. Elliott, A. Donald and F. A. Milner, A second order splitting method for the Cahn-Hilliard equation, Numer. Math., 54(5):575–590, 1989.
[16] D. J. Eyre, Unconditionally gradient stable time marching the Cahn-Hilliard equation, MRS. Proc., 529:39–46, 1998.
[17] X. B. Feng, A. Prohl, Error analysis of a mixd finite element method for the Cahn-Hilliard equation, Numer. Math., 99:47–84, 2004.
[18] X. B. Feng, H. J. Wu, A posteriori error estimates for finite element approximation of the Cahn-Hilliard eqatuion and the Hele-Shaw flow, J. Comp. Math., 26(6):767–796, 2008.
[19] J. Frutos, B. Garcia-Archilla and J. Novo, A postprocessed Galerkin method with Chebyshev or Legendre polynomials, Numer. Math., 86(3):419–442, 2000.
[20] J. Frutos, B. Garcia-Archilla and J. Novo, The postprocessed mixed finite-element method for the Navier-Stokes equations: Refined error bounds, SIAM J. Numer. Anal., 46(1):201–230, 2007.
[21] J. Frutos, J. Novo, Postprocessing the linear finite element method, SIAM J. Numer. Anal., 40(3):805–819, 2002.
[22] B. GarcIa-Archilla, E. Titi, Postprocessing the Galerkin Method: The finite-element case, SIAM J. Numer. Anal., 37(2):470–499, 2000.
[23] V. Girault, J. Lions, Two-grid finite-element schemes for the steady Navier-Stokes problem in polyhedra, Port. Math., 58(1):25–58, 2001.
[24] Y. N. He, Y. X. Liu, Stability and convergence of the spectral Galerkin method for the Cahn-Hilliard equation, Numer. Meth. PDE., 24(6):1485–1500, 2008.
[25] Y. N. He, Y. X. Liu and T. Tang, On large time-stepping methods for the Cahn-Hilliard equation, Appl. Numer. Math., 57(5):616–628, 2007.
[26] X. Hu, X. Cheng, Acceleration of a two-grid method for eigenvalue problems, Math. Comp., 80(275):1287–1301, 2011.
[27] Z. Hu, S. Wise, C. Wang and J. Lowengrub, Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation, J. Comp. Phys., 228:5323–5339, 2009.
[28] S. Kaya, B. Riviere, A two-grid stabilization method for solving the steady-state Navier-Stokes equations, Numer. Meth. PDE., 212(1):288–304, 2006.
[29] D. Kay, R. Welfor, A multigrid finite element solver for the Cahn-Hilliard equation, J. Che. Phys., 22:728–743, 2005.
[30] J. S. Langer, M. Bar-on and H. D. Miller, New computational method in the theory of spinodal decomposition, Physical Review A, 11(4):1417–1729, 1975.
[31] B. Lubomir, N. Robert, Adaptive finite element methods for Cahn-Hilliard equations, J. Comp. Appl. Math., 218(1):2–11, 2008.
[32] M. Marion, J. Xu Error estimates on a new nonlinear Galerkin method based on two-grid

finite elements, SIAM J. Numer. Anal., 32(4):1170–1184, 1995.
[33] L. G. Margolin, E. S. Titi and S. Wynne, The postprocessing Galerkin and nonlinear Galerkin methods: A truncation analysis point of view, SIAM J. Numer. Anal., 41:695–714, 2004.
[34] J. Shen, X. F. Yang, Numerical approximations of Allen-Cahn and Cahn-Hilliard equations, Discrete and Continuous Dynamical Systems, Series A, 28:1669–1691, 2010.
[35] J. Shen, X. F. Yang, Energy stable schemes for Cahn-Hilliard phase-field model of two-phase incompressible flows, Chinese Annals of Mathematics, Series B, 31(5):743–758, 2010.
[36] J. Shen, C. Wang, X. M. Wang and S.M. Wise, Second-order convex splitting schemes for gradient flows with Ehrlich-Schwoebel type energy: Application to thin film epitaxy, SIAM J. Numer. Anal., 50(1):105–125, 2012.
[37] W. S. Wang, Long-time behavior of the two-grid finite element method for fully discrete semilinear evolution equations with positive memory, J. Comp. Appl. Math., 250:161–174, 2013.
[38] W. S. Wang, L. Chen and J. Zhou, Postprocessing mixed finite element methods for solving Cahn-Hilliard equation: Methods and error analysis, submit.
[39] G. N. Wells, E. Kuhl and K. Garikipati, A discontinuous Galerkin method for the Cahn-Hilliard equation, J. Comp. Phys., 218(2):860–877, 2006.
[40] S. Wise, C. Wang and J. Lowengrub, An energy-stable and convergent finite-difference scheme for the phase field crystal equation, SIAM J. Numer. Anal., 47:2269–2288, 2009.
[41] S. M. Wise, Unconditionally stable finite difference, nonlinear multigrid simulation of the Cahn-Hilliard-Hele-Shaw system of equations, J. Sci. Comp., 44(1):38–68, 2010.
[42] O. Wodo, G. Baskar, Computationally efficient solution to the Cahn-Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem, J. Comp. Phys., 230(15):6037–6060, 2011.
[43] Y. H. Xia, Y. Xu and C. W. Shu, Local discontinuous Galerkin methods for the Cahn–Hilliard type equations, J. Comp. Phys., 227(1):472–491, 2007.
[44] J. Xu, A new class of iterative methods for nonselfadjoint or indefinite problems, SIAM J. Numer. Anal., 29(2):303–319, 1992.
[45] J. Xu, A novel two-grid method for semilinear elliptic equations, SIAM J. Sci. Comp., 15(1):231–237, 1994.
[46] J. Xu, Two-grid discretization techniques for linear and nonlinear PDEs, SIAM J. Numer. Anal., 33(5):1759–1777, 1996.
[47] J. Xu, A. Zhou, A two-grid discretization scheme for eigenvalue problems, Math. Comp., 70(233):17–25, 2001.
[48] Y. Yan, Postprocessing the finite element-method for semilinear parabolic problems, SIAM J. Numer. Anal., 44:1681-1702, 2006.
[49] S. Zhang, M. Wang, A nonconforming finite element method for the Cahn–Hilliard equation, J. Comp. Phys., 229(14):7361–7372, 2010.
[50] L. Zhong, S. Shu, J. Wang and J.Xu, Two-grid methods for time-harmonic Maxwell equations, Numer. Lin. Alge. Appl., 20(1):93–111, 2013.
[51] J. Zhou, X. Hu, L. Zhong, S. Shu, and L. Chen, Two-grid methods for Maxwell eigenvalue problems, SIAM J. Numer. Anal., 52(4):2027–2047, 2014.
[52] http://fenicsproject.org/documentation/dolfin/dev/python/demo/pde/cahn-hilliard/python/documentation.html.