A Robust Solver for a Mixed Finite Element Method for the Cahn–Hilliard Equation

Susanne C. Brenner  
*Louisiana State University*

Amanda E. Diegel  
*Louisiana State University*

Li Yeng Sung  
*Louisiana State University*

Follow this and additional works at: [https://digitalcommons.lsu.edu/mathematics_pubs](https://digitalcommons.lsu.edu/mathematics_pubs)

**Recommended Citation**  
Brenner, S., Diegel, A., & Sung, L. (2018). A Robust Solver for a Mixed Finite Element Method for the Cahn–Hilliard Equation. *Journal of Scientific Computing*, 77 (2), 1234-1249. [https://doi.org/10.1007/s10915-018-0753-3](https://doi.org/10.1007/s10915-018-0753-3)

This Article is brought to you for free and open access by the Department of Mathematics at LSU Digital Commons. It has been accepted for inclusion in Faculty Publications by an authorized administrator of LSU Digital Commons. For more information, please contact ir@lsu.edu.
A Robust Solver for a Mixed Finite Element Method for the Cahn-Hilliard Equation

Susanne C. Brenner†, Amanda E. Diegel‡, Li-Yeng Sung§

Abstract
We develop a robust solver for a mixed finite element convex splitting scheme for the Cahn-Hilliard equation. The key ingredient of the solver is a preconditioned minimal residual algorithm (with a multigrid preconditioner) whose performance is independent of the spatial mesh size and the time step size for a given interfacial width parameter. The dependence on the interfacial width parameter is also mild.

Keywords Cahn-Hilliard equation; convex splitting; mixed finite element methods; MINRES; block diagonal preconditioner; multigrid.

1 Introduction
Let \( \Omega \subset \mathbb{R}^d \), \( d = 2, 3 \), be an open polygonal or polyhedral domain, and consider the following form of the Cahn-Hilliard energy [7]:

\[
E(\phi) = \int_{\Omega} \left( \frac{1}{4\varepsilon} (\phi^2 - 1)^2 + \frac{\varepsilon}{2} |\nabla \phi|^2 \right) dx,
\]

where \( \varepsilon > 0 \) is a constant, and \( \phi \in H^1(\Omega) \) represents a concentration field. The phase equilibria are represented by \( \phi = \pm 1 \) and the parameter \( \varepsilon \) represents a non-dimensional interfacial width between the two phases.

The Cahn-Hilliard equation, which can be interpreted as the gradient flow of the energy (1.1) in the dual space of \( H^1(\Omega) \), is often represented in mixed form by

\[
\partial_t \phi = \varepsilon \Delta \mu, \quad \text{in } \Omega, \tag{1.2a}
\]

\[
\mu = \varepsilon^{-1} (\phi^3 - \phi) - \varepsilon \Delta \phi, \quad \text{in } \Omega, \tag{1.2b}
\]

together with the boundary conditions \( \partial_n \phi = 0 \) and \( \partial_n \mu = 0 \).

Let \( T \) be a positive number and \( H_N^{-1}(\Omega) \) be the dual space of \( H^1(\Omega) \). A weak formulation of (1.2a)–(1.2b) is to find \((\phi, \mu)\) such that

\[
\phi \in L^\infty \left(0, T; H^1(\Omega)\right) \cap L^4 \left(0, T; L^\infty(\Omega)\right), \tag{1.3a}
\]

\[
\partial_t \phi \in L^2 \left(0, T; H_N^{-1}(\Omega)\right), \tag{1.3b}
\]

\[
\mu \in L^2 \left(0, T; H^1(\Omega)\right), \tag{1.3c}
\]
and, for almost all \( t \in (0, T) \),
\[
\langle \partial_t \phi, \nu \rangle + \varepsilon a(\mu, \nu) = 0 \quad \forall \nu \in H^1(\Omega),
\]
\[
(\mu, \psi) - \varepsilon a(\phi, \psi) - \varepsilon^{-1} (\phi^3 - \phi, \psi) = 0 \quad \forall \psi \in H^1(\Omega).
\]
(1.4a)
(1.4b)

Here \( \langle \cdot, \cdot \rangle \) denotes the duality pairing between the spaces \( H^{-1}_N(\Omega) \) and \( H^1(\Omega) \), \( (\cdot, \cdot) \) is the inner product of \( L^2(\Omega) \), and
\[
a(u, v) = (\nabla u, \nabla v).
\]

The proof for the existence and uniqueness of the weak solution for (1.3)–(1.4) with initial data
\[
\phi(0) = \phi_0 \in H^2_N(\Omega) = \{v \in H^2(\Omega) : \partial v/\partial n = 0 \text{ on } \partial \Omega\}
\]
(1.5)
can be found for example in [30].

The Cahn-Hilliard energy (1.1) along with the system (1.2) was originally developed to model phase separation of a binary fluid [7, 8, 15]. However, variations of the Cahn-Hilliard system are quickly becoming one of the most popular components in what are known as phase field models. The role that the Cahn-Hilliard equation takes in these models may best be described as creating an indicator function so that explicit tracking of the interface between two phases is not required. The growing number of applications include two phase flows, Hele-Shaw flows, copolymer fluids, crystal growth, void electromigration, vesicle membranes and more (cf. [18, 26, 10, 33, 3, 13] and the references therein).

There is a vast literature on numerical methods for the Cahn-Hilliard equation (cf. [31, 12, 25, 35, 37] and the references therein) and solvers based on various numerical schemes were developed in [2, 5, 9, 21, 22, 23, 29, 36, 28, 24, 35]. We will consider the mixed finite element method for (1.3)–(1.5) investigated in [11]. It is based on the convex splitting scheme in time [17] given by
\[
\frac{\phi^m - \phi^{m-1}}{\tau} = \varepsilon \Delta \mu^m,
\]
(1.6a)
\[
\Delta \mu^m = \frac{1}{\varepsilon} ((\phi^m)^3 - \phi^{m-1}) - \varepsilon \Delta \phi^m,
\]
(1.6b)
\[
\text{where } \tau \text{ is the size of the time step, and a spatial discretization that employs Lagrange finite elements. This mixed finite element method is unconditionally stable and has optimal convergence in both time and space. Our goal is to develop a robust solver for this mixed finite element method.}

The remainder of this paper is organized as follows. The mixed finite element method is introduced in Section 2, followed by the construction and analysis of the solver in Section 3. Numerical results that demonstrate the performance of the solver are presented in Section 4 and we end the paper with some concluding remarks in Section 5.

2 A Mixed Finite Element Method

Let \( M \) be a positive integer, \( 0 = t_0 < t_1 < \cdots < t_M = T \) be a uniform partition of \([0, T]\) and \( T_h \) be a quasi-uniform family of triangulations of \( \Omega \) (cf. [6]). The Lagrange finite element space \( S_h \subset H^1(\Omega) \) is given by
\[
S_h = \{v \in C(\bar{\Omega}) : v|_K \in \mathcal{P}_1(K) \forall K \in T_h\},
\]
\[
\text{and we define } \tilde{S}_h = S_h \cap L^2_0(\Omega),
\]
where $L_0^2(\Omega)$ is the space of square integrable functions with zero mean.

The mixed finite element scheme for (1.6) investigated in [1] is defined as follows: For $1 \leq m \leq M$, find $\phi_h^m, \mu_h^m \in S_h$ such that

\[
\begin{align*}
(\delta_t \phi_h^m, \nu) + \varepsilon a(\mu_h^m, \nu) &= 0 &\forall \nu \in S_h, \\
(\mu_h^m, \psi) - \varepsilon^{-1} ((\phi_h^m)^3 - \phi_h^{m-1}, \psi) - \varepsilon a(\phi_h^m, \psi) &= 0 &\forall \psi \in S_h, \\
\phi_h^0 - R_h \phi_0 &= 0.
\end{align*}
\]

Here

\[
\delta_t \phi_h^m = \frac{\phi_h^m - \phi_h^{m-1}}{\tau},
\]

where $\tau = T/M$ is the size for the time step, and the Ritz projection operator $R_h : H^1(\Omega) \rightarrow S_h$ is defined by

\[
a(R_h v, w) = (v, w) \quad \forall w \in S_h, \tag{2.2a}
\]

\[
(R_h v - v, 1) = 0. \tag{2.2b}
\]

**Remark 2.1.** The energy law

\[
E(\phi(t)) + \int_0^t \varepsilon \|\nabla \mu(s)\|^2_{L^2(\Omega)} ds = E(\phi(0)) \quad \forall t \in [0, T]
\]

is a key property of the solution of (1.3)–(1.5). It can be shown [1] that the solution of the finite element method defined by (2.1)–(2.2) also satisfies a similar energy law, which leads to $\phi_h \in L^\infty(0, T; L^\infty(\Omega))$ and $\mu_h \in L^2(0, T; L^\infty(\Omega))$. Moreover, under the assumption that $\phi \in H^2(0, T; L^2(\Omega)) \cap L^\infty(0, T; W^{1,6}(\Omega)) \cap H^1(0, T; H^2(\Omega)), \mu \in L^\infty(0, T; H^1(\Omega)) \cap L^2(0, T; H^2(\Omega))$, and $0 \leq \tau \leq \tau_0$ for a sufficiently small $\tau_0$, the error estimate

\[
\max_{1 \leq m \leq M} \|\nabla \phi^m - \nabla \phi_h^m\|^2_{L^2} + \tau \sum_{m=1}^M \|\nabla \mu^m - \nabla \mu_h^m\|^2_{L^2} \leq C(\varepsilon, T)(\tau^2 + h^2) \tag{2.3}
\]

holds for a positive constant $C$ that depends on $\varepsilon$ and $T$ but does not depend on $\tau$ and $h$.

It follows from (2.1) that $(\phi_h^m, 1) = (\phi_0, 1)$ for $0 \leq m \leq M$, and hence

\[
\phi_h^m = \phi_0 + \phi_h^m \quad \text{for} \quad 0 \leq m \leq M, \tag{2.4}
\]

where $\phi_0 = (\phi_0, 1)/(1, 1)$ is the mean of $\phi_0$ over $\Omega$ and $\phi_h^0 \in \hat{S}_h$. We can also write

\[
\mu_h^m = \mu_h^m + \mu_h^m, \tag{2.5}
\]

where $\mu_h^0$ is a constant function and $\mu_h^m \in \hat{S}_h$.

Using (2.4) and (2.5), we can rewrite (2.1a)–(2.1b) in the following equivalent form: For $1 \leq m \leq M$, find $\phi_h^m, \mu_h^m \in S_h$ such that

\[
\begin{align*}
(\delta_t \phi_h^m, \nu) + \varepsilon a(\mu_h^m, \nu) &= 0 &\forall \nu \in \hat{S}_h, \\
(\mu_h^m, \psi) - \varepsilon^{-1} ((\phi_h^m + \phi_0)^3 - \phi_h^{m-1}, \psi) - \varepsilon a(\phi_h^m, \psi) &= 0 &\forall \psi \in \hat{S}_h,
\end{align*}
\]

where

\[
\delta_t \phi_h^m = \frac{\phi_h^m - \phi_h^{m-1}}{\tau}.
\]
Note that we can recover the constant function \( \overline{\mu}_h^m \) from \( \dot{\phi}_h^m \) and \( \ddot{\phi}_h^{m-1} \) through the relation

\[
(\overline{\mu}_h^m, 1) = \varepsilon^{-1}( (\dot{\phi}_h^m + \overline{\phi}_0)^3 - \dot{\phi}_h^{m-1}, 1 )
\]

that follows from (2.1b), (2.4) and (2.5).

**Remark 2.2.** The nonlinear system (2.6) defines the first order optimality condition at a minimum \( \dot{\phi}_h^m \) of the convex functional \( \Phi : \hat{S}_h \rightarrow \mathbb{R} \) given by

\[
\Phi_h(\varphi) = \frac{\varepsilon}{2\tau} a(\varphi, \varphi) + \frac{1}{4\varepsilon} \| \varphi + \overline{\varphi}_0 \|_{L^4}^4 - \frac{1}{\varepsilon} \left( \dot{\phi}_h^{m-1}, \varphi \right) + \frac{\varepsilon}{2} a(\varphi, \varphi),
\]

where \( \varphi \in \hat{S}_h \) is defined by

\[
\varepsilon a(\varphi, \nu) + \left( \varphi - \dot{\phi}_h^{m-1}, \nu \right) = 0 \quad \forall \, \nu \in \hat{S}_h.
\]

Since this convex minimization problem has a unique minimum by the standard theory [14], the system (2.6) (and hence (2.1)) is also uniquely solvable.

### 3 A Robust Solver

We will solve the nonlinear system (2.6) by Newton’s iteration. Let \( (\dot{\phi}_{h,j}^m, \ddot{\phi}_{h,j}^m) \in \hat{S}_h \times \hat{S}_h \) be the output of the \( j \)-th step. In order to advance the iteration, we need to find \( (\delta_j \mu, \delta_j \phi) \in \hat{S}_h \times \hat{S}_h \) such that

\[
\tau \varepsilon a(\delta_j \mu, \nu) + (\nu, \delta_j \phi) = F_j(\nu) \quad \forall \, \nu \in \hat{S}_h,
\]

\[
(\delta_j \mu, \psi) - \left[ 3\varepsilon^{-1}( (\dot{\phi}_{h,j}^m)^2 \delta_j \phi, \psi) + \varepsilon a(\delta_j \phi, \psi) \right] = G_j(\psi) \quad \forall \, \phi \in \hat{S}_h,
\]

where \( \phi_{h,j}^m = \dot{\phi}_{h,j}^m + \overline{\phi}_0 \) and

\[
F_j(\nu) = \tau \varepsilon a(\dot{\phi}_{h,j}^m, \nu) + (\dot{\phi}_{h,j}^m - \dot{\phi}_{h,j}^{m-1}, \nu),
\]

\[
G_j(\psi) = (\dot{\phi}_{h,j}^m, \psi) - \left[ \varepsilon^{-1}( (\dot{\phi}_{h,j}^m)^3 - \dot{\phi}_{h,j}^{m-1}, \psi) + \varepsilon a(\dot{\phi}_{h,j}^m, \psi) \right].
\]

The next output of the Newton iteration is then given by

\[
(\dot{\phi}_{h,j+1}^m, \ddot{\phi}_{h,j+1}^m) = (\dot{\phi}_{h,j}^m, \ddot{\phi}_{h,j}^m) - (\delta_j \mu, \delta_j \phi).
\]

Below we will construct a robust solver for (3.1).

First we circumvent the inconvenient zero mean constraint by reformulating (3.1) as the following equivalent problem: Find \( (\delta_j \mu, \delta_j \phi) \in S_h \times S_h \) such that

\[
\tau \varepsilon \left[ a(\delta_j \mu, \nu) + (\nu, \delta_j \phi) \right] + (\nu, \delta_j \phi) = \tilde{F}_j(\nu) \quad \forall \, \nu \in S_h,
\]

\[
(\delta_j \mu, \psi) - \left[ 3\varepsilon^{-1}( (\dot{\phi}_{h,j}^m)^2 \delta_j \phi, \psi) + \varepsilon \left[ a(\delta_j \phi, \psi) + (\delta_j \phi, 1)(\psi, 1) \right] \right] = \tilde{G}_j(\psi) \quad \forall \, \psi \in S_h,
\]

where

\[
\tilde{F}_j(\nu) = \begin{cases} F_j(\nu) & \text{if } \nu \in \hat{S}_h \\ 0 & \text{if } \nu = 1 \end{cases} \quad \text{and} \quad \tilde{G}_j(\psi) = \begin{cases} G_j(\psi) & \text{if } \psi \in \hat{S}_h \\ 0 & \text{if } \psi = 1 \end{cases}.
\]
Remark 3.1. It is easy to check that both (3.1) and (3.4) are well-posed linear systems and that the solution $(\delta_j \mu, \delta_j \phi)$ of (3.1)–(3.2) also satisfies (3.4)–(3.5).

Under the change of variables

$$(\delta_j \mu, \nu) \rightarrow \tau^{-\frac{1}{2}} \varepsilon^{-\frac{1}{2}} (\delta_j \mu, \nu) \quad \text{and} \quad (\delta_j \phi, \psi) \rightarrow \tau^{-\frac{1}{2}} \varepsilon^{-\frac{1}{2}} (\delta_j \phi, \psi),$$

the system (3.4) becomes

$$\begin{align*}
\tau^{-\frac{1}{2}} [a(\delta_j \mu, \nu) + (\delta_j \mu, 1)(\nu, 1)] + (\nu, \delta_j \phi) &= \bar{F}_j(\tau^{-\frac{1}{2}} \varepsilon^{-\frac{1}{2}} \nu), \\
(\delta_j \mu, \psi) - (3\tau^{-\frac{1}{2}}(\phi_{h,j}^m \delta_j \phi, \psi) + \tau^{-\frac{1}{2}} \varepsilon^2 [a(\delta_j \phi, \psi) + (\delta_j \phi, 1)(\psi, 1)]) &= \bar{G}_j(\tau^{-\frac{1}{2}} \varepsilon^{-\frac{1}{2}} \psi),
\end{align*}$$

(3.6a)

(3.6b)

for all $(\nu, \psi) \in S_h \times S_h$.

Let $n_h$ be the dimension of $S_h$ and $\varphi_1, \ldots, \varphi_{n_h}$ be the standard nodal basis (hat) functions for $S_h$. The system matrix for (3.6) is given by

$$
\begin{bmatrix}
\tau^{-\frac{1}{2}} (K + cc') & M \\
M & -\tau^{-\frac{1}{2}} J(\phi_{h,j}^m) - \tau^{-\frac{1}{2}} \varepsilon^2 (K + cc')
\end{bmatrix},
$$

(3.7)

where the stiffness matrix $K$ is defined by $K(k, \ell) = (\nabla \varphi_k, \nabla \varphi_\ell)$, the mass matrix $M$ is defined by $M(k, \ell) = (\varphi_k, \varphi_\ell)$, the vector $c$ is defined by $c(k) = (\varphi_k, 1)$, and the matrix $J(\phi_{h,j}^m)$ is defined by

$$J(\phi_{h,j}^m)(k, \ell) = 3((\phi_{h,j}^m)^2 \varphi_k, \varphi_\ell).$$

Note that, since the mixed finite element method is convergent, we can expect $(\phi_{h,j}^m)^2$ to be close to 1 away from an interfacial region with width $\varepsilon$. Therefore, for small $\varepsilon$, we can take $(\phi_{h,j}^m)^2$ to be 1 in the system matrix, i.e., we can replace $J(\phi_{h,j}^m)$ by $3M$ in (3.7). The following result is motivated by this observation.

**Theorem 3.2.** Let the matrices $B$ and $P$ be defined by

$$
B = \begin{bmatrix}
\tau^{-\frac{1}{2}} (K + cc') & M \\
M & -3\tau^{-\frac{1}{2}} M - \tau^{-\frac{1}{2}} \varepsilon^2 (K + cc')
\end{bmatrix},
$$

(3.8)

$$
P = \begin{bmatrix}
\tau^{-\frac{1}{2}} (K + cc') + M & 0 \\
0 & \tau^{-\frac{1}{2}} \varepsilon^2 (K + cc') + M
\end{bmatrix},
$$

(3.9)

where $0 \leq \tau, \varepsilon \leq 1$. There exist two positive constants $C_1$ and $C_2$ independent of $\varepsilon$, $h$ and $\tau$ such that

$$C_2 \max(\tau^{-\frac{1}{2}}, \varepsilon) \leq |\lambda| \leq C_1 \quad \text{for any eigenvalue} \ \lambda \ \text{of} \ P^{-1}B.
$$

(3.10)

**Proof.** A simple calculation shows that

$$
P^{-1}B = \left(\begin{bmatrix}
M & 0 \\
0 & M
\end{bmatrix} \begin{bmatrix}
\tau^{-\frac{1}{2}} \hat{K} + I & 0 \\
0 & \tau^{-\frac{1}{2}} \varepsilon^2 \hat{K} + I
\end{bmatrix}\right)^{-1} \left(\begin{bmatrix}
M & 0 \\
0 & M
\end{bmatrix} \begin{bmatrix}
\tau^{-\frac{1}{2}} \hat{K} & I \\
I & -3\tau^{-\frac{1}{2}} I - \tau^{-\frac{1}{2}} \varepsilon^2 \hat{K}
\end{bmatrix}\right)
$$

$$= \begin{bmatrix}
\tau^{-\frac{1}{2}} \hat{K} + I & 0 \\
0 & \tau^{-\frac{1}{2}} \varepsilon^2 \hat{K} + I
\end{bmatrix}^{-1} \begin{bmatrix}
\tau^{-\frac{1}{2}} \hat{K} & I \\
I & -3\tau^{-\frac{1}{2}} I - \tau^{-\frac{1}{2}} \varepsilon^2 \hat{K}
\end{bmatrix},
$$

where $\hat{K} = M^{-1}(K + cc')$ and $I$ is the $n_h \times n_h$ identity matrix.
By the spectral theorem, there exist \( \mathbf{v}_1, \ldots, \mathbf{v}_{n_h} \in \mathbb{R}^{n_h} \) and positive numbers \( \kappa_1, \ldots, \kappa_{n_h} \) such that

\[
\tilde{K}\mathbf{v}_j = \kappa_j \mathbf{v}_j \quad \text{for } 1 \leq j \leq n_h
\]

and

\[
\mathbf{v}_j^T \mathbf{M} \mathbf{v}_\ell = \begin{cases} 1 & \text{if } j = \ell \\ 0 & \text{if } j \neq \ell \end{cases}.
\]

Observe that the two dimensional space \( V_j \) spanned by \( \begin{bmatrix} \mathbf{v}_j \\ 0 \end{bmatrix} \) and \( \begin{bmatrix} 0 \\ \mathbf{v}_j \end{bmatrix} \) is invariant under \( P^{-1}B \) and

\[
P^{-1}B \left( \alpha \begin{bmatrix} \mathbf{v}_j \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ \mathbf{v}_j \end{bmatrix} \right) = \gamma \begin{bmatrix} \mathbf{v}_j \\ 0 \end{bmatrix} + \delta \begin{bmatrix} 0 \\ \mathbf{v}_j \end{bmatrix},
\]

where

\[
\begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \tau \frac{1}{2} \kappa_j + 1 & 0 \\ 0 & \tau \frac{1}{2} \varepsilon^2 \kappa_j + 1 \end{bmatrix}^{-1} \begin{bmatrix} \tau \frac{1}{2} \kappa_j \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ -3 \tau \frac{1}{2} - \tau \frac{1}{2} \varepsilon^2 \kappa_j \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.
\]

It follows that the eigenvalues of \( P^{-1}B \) are precisely the eigenvalues of the matrix

\[
C_j = \begin{bmatrix} \tau \frac{1}{2} \kappa_j + 1 & 0 \\ 0 & \tau \frac{1}{2} \varepsilon^2 \kappa_j + 1 \end{bmatrix}^{-1} \begin{bmatrix} \tau \frac{1}{2} \kappa_j \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ -3 \tau \frac{1}{2} - \tau \frac{1}{2} \varepsilon^2 \kappa_j \end{bmatrix}
\]

for \( 1 \leq j \leq n_h \). Hence we only need to understand the behavior of the eigenvalues of the matrix

\[
C = \begin{bmatrix} \omega \\ \omega + 1 \\ \omega + 1 \\ \omega^2 + 1 \\ -3 \tau \frac{1}{2} - \omega \varepsilon^2 \\ \omega^2 + 1 \end{bmatrix}
\]

where \( \omega \) is a positive number and \( 0 < \tau, \varepsilon \leq 1 \).

First of all we have

\[
|\lambda| \leq \|C\|_\infty \leq 4 \quad \text{for any eigenvalue } \lambda \text{ of } C,
\]

which implies that the second estimate in (3.10) holds for \( C_1 = 4 \).

A direct calculation shows that

\[
|\det C| = \frac{1 + 3 \tau \frac{1}{2} \omega + \varepsilon^2 \omega^2}{1 + (1 + \varepsilon^2) \omega + \varepsilon^2 \omega^2} \geq \frac{1 + 3 \tau \frac{1}{2} \omega + \varepsilon^2 \omega^2}{1 + 2 \omega + \varepsilon^2 \omega^2}.
\]

On one hand we have

\[
1 + 2 \omega + \varepsilon^2 \omega^2 \leq \tau - \frac{1}{2} (1 + 3 \tau \frac{1}{2} \omega + \varepsilon^2 \omega^2),
\]

for \( 0 \leq \omega \leq 1 \).
which implies
|det C| ≥ \( \tau^2 \).

(3.12)

On the other hand we also have

\[
1 + 2\omega + \varepsilon^2 \omega^2 \leq \varepsilon^{-1}(1 + 2\varepsilon \omega + \varepsilon^2 \omega^2) \leq 2\varepsilon^{-1}(1 + \varepsilon^2 \omega^2) \leq 2\varepsilon^{-1}(1 + 3\tau^2 \omega + \varepsilon^2 \omega^2),
\]

which implies

|det C| ≥ \( \varepsilon \).

(3.13)

Putting (3.11)–(3.13) together we see that

\[
4|\lambda| \geq |det C| \geq \max(\tau^2, \varepsilon/2)
\]

for any eigenvalue \( \lambda \) of \( C \). Therefore the first estimate in (3.10) holds with \( C_2 = 1/8 \).

In our numerical experiments we use the preconditioner \( P_* \) given by

\[
P_* = \begin{bmatrix}
\tau^2 K + M & 0 \\
0 & \tau^2 \varepsilon^2 K + M
\end{bmatrix}.
\]

(3.14)

Since the two symmetric positive definite matrices \( P \) and \( P_* \) are spectrally equivalent, we immediately deduce from Theorem 3.2 that there exist two positive constants \( C_3 \) and \( C_4 \) independent of \( \varepsilon \), \( h \) and \( \tau \) such that

\[
C_4 \max(\tau^2, \varepsilon) \leq |\lambda| \leq C_3
\]

(3.15)

for any eigenvalue \( \lambda \) of \( P_*^{-1}B \).

According to (3.15), the performance of the preconditioned MINRES algorithm (cf. [19, 16]) for systems involving \( B \) is independent of \( \tau \) and \( h \) for a given \( \varepsilon \), and also independent of \( \varepsilon \) and \( h \) for a given \( \tau \). Similar behavior can also be expected for systems involving the matrix in (3.7). Furthermore, the action of \((\gamma K + M)^{-1}\) on a vector can be computed by a multigrid method, which creates large computational savings.

Remark 3.3. Recall the matrix \( B \) is obtained from the matrix in (3.7) by replacing \( J(\phi^{in}) \) by 3M and its justification depends on \( \varepsilon \). Therefore we expect to see some dependence of the performance of the preconditioned MINRES algorithm on \( \varepsilon \) for a given \( \tau \).

Remark 3.4. When \( \tau \) becomes 0, the matrix

\[
B = \begin{bmatrix}
0 & M \\
M & 0
\end{bmatrix}
\]

is well-conditioned. Therefore the performance of the preconditioned MINRES algorithm for systems involving the matrix in (3.7) will improve as the time step size decreases.

Remark 3.5. Block diagonal preconditioners for saddle point systems are discussed in [4, 27] and the references therein.
4 Numerical Experiments

In this section we report the results of eight numerical experiments in two and three dimensions. All computations were carried out using the FELICITY MATLAB/C++ Toolbox [34] unless specified otherwise.

In the first six numerical experiments, we solve (2.1) on the unit square $\Omega = (0,1)^2$ using uniform meshes. The initial mesh $T_0$ is generated by the two diagonals of $\Omega$ and the meshes $T_1, T_2, \ldots$ are obtained from $T_0$ by uniform refinements.

For the first five experiments, we use the initial data

$$\phi^0_h = I_h \left[ \left( \frac{1}{2} \right) \left[ 1 - \cos(2\pi x_1) \right] \left[ 1 - \cos(2\pi x_2) \right] \right] - 1,$$

(4.1)

where $I_h : H^2(\Omega) \rightarrow S_h$ is the standard nodal interpolation operator.

The system (2.1) (or equivalently (2.6)) is solved by the Newton iteration with a tolerance of $10^{-15}$ for $\|\delta_j \phi \|_{L^\infty(\Omega)}$ or a residual tolerance of $10^{-7}$ for (3.4)–(3.5), whichever is satisfied first. It turns out that only one Newton iteration is needed for each time step in all the experiments.

During each Newton iteration, the systems involving (3.7) are solved by a preconditioned MINRES algorithm with a residual tolerance of $10^{-7}$. The systems involving the preconditioner $P$ are solved by a multigrid $V(4,4)$ algorithm that uses the Gauss-Seidel iteration as the smoother (cf. [20, 32]). In all our experiments the maximum number of preconditioned MINRES iterations occurred during the first few time steps after which the number of iterations would decrease and level off.

In the first experiment, we take $\tau = 0.002/64$ with a final time $T = 0.04$ for the two interfacial width parameters $\varepsilon = 0.0625$ and $\varepsilon = 0.001$. In Table 1 we report the average number of preconditioned MINRES iterations over all time steps along with the average solution time per time step as the mesh is refined. (The timing mechanism is the ‘tic toc’ command in MATLAB.) We observe that the performance of the preconditioned MINRES algorithm does not depend on $h$ and the solution time per time step grows linearly with the number of degrees of freedom. Moreover the solution time roughly doubles as $\varepsilon$ decreases from 0.0625 to 0.001, indicating that the performance of the solver only has a mild dependence on $\varepsilon$.

| $h$     | $\varepsilon = 0.0625$ | $\varepsilon = 0.001$ |
|---------|------------------------|------------------------|
| $\sqrt{2}/8$ | 20 | 0.042391 | 28 | 0.01786 |
| $\sqrt{2}/16$ | 21 | 0.070047 | 44 | 0.04537 |
| $\sqrt{2}/32$ | 23 | 0.156576 | 57 | 0.13596 |
| $\sqrt{2}/64$ | 24 | 0.444770 | 71 | 0.50508 |
| $\sqrt{2}/128$ | 25 | 1.752561 | 107 | 3.13307 |
| $\sqrt{2}/256$ | 26 | 6.884936 | 96 | 12.3052 |
| $\sqrt{2}/512$ | 26 | 26.84091 | 97 | 57.2141 |
| $\sqrt{2}/1024$ | 26 | 108.9613 | 100 | 245.456 |

Table 1: The average number of preconditioned MINRES iterations over all time steps together with the average solution time per time step as the mesh is refined ($\Omega = (0,1)^2$, $\tau = 0.002/64$, $T = 0.04$ $\varepsilon = 0.0625$ (left), $\varepsilon = 0.001$ (right)).

Table 2 shows the average solution time per time step for the same problem with $\varepsilon = 0.0625$ using FEniCS [1] on the prebuilt high-performance Docker. The main components of this code are Newton’s method and LU decomposition. The time step size is fixed at $\tau = 0.002/64$. The
residual tolerance is set at $10^{-7}$. The timing mechanism is a start and stop of the python command ‘time.time’. By comparing Table 2 with Table 1, we see that FEniCS appears to be faster for the coarser mesh sizes. However, as the mesh is refined, the advantage of our method is clearly observed.

| $h$     | Avg. Time to Solve (s) |
|---------|-------------------------|
| $\sqrt{2}/8$ | 0.01449                |
| $\sqrt{2}/16$ | 0.02589                |
| $\sqrt{2}/32$ | 0.06675                |
| $\sqrt{2}/64$ | 0.21948                |
| $\sqrt{2}/128$ | 3.94694                |
| $\sqrt{2}/256$ | 44.4569                |

Table 2: The average solution time per time step using FEniCS to run the same test as performed in Table 1 with $\varepsilon = 0.0625$.

In the second experiment, we again take $\tau = 0.002/64$ and a final time $T = 0.04$. The median numbers of the preconditioned MINRES iterations over all time steps for several values of $\varepsilon$ as the mesh is refined are plotted in Figure 1. The performance of our method is independent of the mesh size $h$, and there is some dependence on the interfacial width parameter $\varepsilon$ as expected (cf. Remark 3.3).

![Figure 1: The median number of MINRES iterations over all time steps for several values of $\varepsilon$ as the mesh is refined ($\Omega = (0, 1)^2$, $\tau = 0.002/64$ and $T = 0.04$).](image)

In the third experiment, we fix $h = \sqrt{2}/64$, a final time $T = .04$, $\varepsilon = 0.0625$ and $0.001$, and refine the time step size $\tau$. The average number of the preconditioned MINRES iterations over all time steps is displayed in Table 3 along with the average solution time per time step. The performance is clearly independent of the time step size $\tau$. The solution time roughly triples as $\varepsilon$ decreases from $0.0625$ to $0.001$, indicating again that the performance of the solver only depends mildly on $\varepsilon$. 

![Figure 1: The median number of MINRES iterations over all time steps for several values of $\varepsilon$ as the mesh is refined ($\Omega = (0, 1)^2$, $\tau = 0.002/64$ and $T = 0.04$).](image)
In the fourth experiment, we fix $\varepsilon = 0.0625$ and a final time $T = 0.04$. The median numbers of preconditioned MINRES iterations over all time steps for several values of $\tau$ as the mesh is refined are displayed in Figure 2. The performance of our method is clearly independent of the mesh size $h$ and the time step size $\tau$.

![Median Number of MINRES Iterations](image)

**Figure 2:** The median number of preconditioned MINRES iterations for several time step sizes as the mesh is refined ($\Omega = (0, 1)^2, h = \sqrt{\tau}/64, T = 0.04, \varepsilon = 0.0625$ (left), $\varepsilon = 0.001$ (right)).

In the fifth experiment, we fix the final time $T = 0.04$ and let $\tau = 0.002h/\sqrt{2}$ (cf. (2.3)). The median numbers of preconditioned MINRES iterations over all time steps for several values of $\varepsilon$ as the mesh is refined are displayed in Figure 3. Again, the performance only depends on the interfacial width parameter $\varepsilon$.

![MINRES Iterations vs. Time Step Size](image)

**Figure 3:** The median number of preconditioned MINRES iterations for several values of $\varepsilon$ as the mesh is refined ($\Omega = (0, 1)^2, h = \sqrt{\tau}/64, T = 0.04$).
Figure 3: The median number of preconditioned MINRES iterations for several values of $\varepsilon$ as the mesh and time step are refined ($\Omega = (0, 1)^2$, $\tau = 0.002h/\sqrt{2}$ and $T = 0.04$).

In the sixth experiment, we solve the Cahn-Hillard equation with a random initial condition. We take $h = \sqrt{2}/128$, $\tau = 0.002/128$ and $\varepsilon = 0.0625$. The surface plots for $\phi$ at $t = 0$, $t = 0.0025$, $t = 0.005$, $t = 0.0075$, $t = 0.01$ and $t = 0.0125$ are displayed in Figure 4. For comparison we solve the same problem using FEniCS and display the corresponding surface plots in Figure 5. The two figures are essentially indistinguishable.

Figure 4: Spinodal decomposition of a binary fluid on $(0, 1)^2$ with random initial data. The times displayed are $t = 0, t = 0.0025, t = 0.005$ (top from left to right) and $t = 0.0075, t = 0.01, t = 0.0125$ (bottom from left to right).
Figure 5: Spinodal decomposition of a binary fluid on $(0,1)^2$ with random initial data obtained by FEniCS. The times displayed are $t=0$, $t=0.0025$, $t=0.005$ (top from left to right) and $t=0.0075$, $t=0.01$, $t=0.0125$ (bottom from left to right).

In the seventh experiment, we solve the Cahn-Hilliard equation with a random initial condition on the unit cube $\Omega = (0,1)^3$ using uniform meshes. The initial mesh $T_0$ consists of six tetrahedrons. The meshes $T_1, T_2, \cdots$ are obtained from $T_0$ by uniform refinements. We take $\varepsilon = 0.0625$, $\tau = 0.002/64$, a final time $T = 0.03$ and refine the mesh.

Table 4 displays the maximum, median, and average number of preconditioned MINRES iterations over all time steps along with the average solution time per time step. Again, only one Newton iteration is needed for each time step. We observe that the performance of the preconditioned MINRES algorithm does not depend on $h$ and the solution time per time step grows linearly with the number of degrees of freedom.

| $h$     | MINRES Iterations | Time to Solve (s) |
|---------|-------------------|-------------------|
| $\sqrt{3}/8$ | 33, 28, 27 | 0.0460738 |
| $\sqrt{3}/16$ | 33, 29, 29 | 0.2451147 |
| $\sqrt{3}/32$ | 36, 30, 30 | 1.977465 |
| $\sqrt{3}/64$ | 37, 29, 30 | 15.97806 |
| $\sqrt{3}/128$ | 41, 30, 31 | 169.7146 |

Table 4: The maximum, median and average number of preconditioned MINRES iterations over all time steps together with the average solution time per time step as the mesh is refined ($\Omega = (0,1)^3$, $\varepsilon = 0.0625$, $\tau = 0.002/64$ and $T = 0.03$).

Isocap plots for $\phi$ at $t = 0$, $t = 0.0015625$, $t = 0.003125$, $t = 0.0046875$, $t = 0.00625$ and $t = 0.0078125$ are displayed in Figure 6.
In the eighth experiment, we solve the Cahn-Hilliard equation with a random initial condition. We take $h = \sqrt{3}/32$, $\tau = 0.002/64$ and $\varepsilon = 0.0625$. Isocap plots for $t = 0$, $t = 0.015$ and $t = 0.03$ are displayed in Figure 7. For comparison, we solve the same problem using FEniCS and display the corresponding isocap plots in Figure 8. The two figures are, again, essentially indistinguishable. Furthermore, we achieve considerable savings in time by using our solver. Specifically, the test using our solver completed in under 30 minutes whereas the test using FEniCS required 24 hours to reach the same final stopping time of $T = 0.03$. 

Figure 6: Spinodal decomposition of a binary fluid on $(0,1)^3$. The times displayed are $t = 0, t = 0.0015625, t = 0.003125$ (top from left to right) and $t = 0.0046875, t = 0.00625, t = 0.0078125$ (bottom from left to right).

Figure 7: Spinodal decomposition of a binary fluid on $(0,1)^3$ with random initial data. The times displayed are $t = 0, t = 0.015, t = 0.03$ (from left to right).
5 Concluding Remarks

We have developed a robust solver for a mixed finite element convex splitting scheme for the Cahn-Hilliard equation, where in each time step the Jacobian system for the Newton iteration is solved by a preconditioned MINRES algorithm with a block diagonal multigrid preconditioner. The robustness of our solver is confirmed by numerical tests in two and three dimensions. We have also validated our numerical results through comparisons with the results obtained through FEniCS and observed significant speed-up.

The methodology developed in this paper can be adapted for coupled systems that involve the Cahn-Hilliard equation, such as the Cahn-Hilliard Navier Stokes system (cf. [31] and the references therein). This is the topic for an ongoing research project.

Acknowledgement

Portions of this research were conducted with high performance computational resources provided by Louisiana State University (http://www.hpc.lsu.edu). We would also like to thank Shawn Walker for his valuable advice regarding the FELICITY/C++ Toolbox for MATLAB.

References

[1] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, E. Rognes M, and G. N. Wells. The fenics project version 1.5. Archive of Numerical Software, 3, 2015.

[2] O. Axelsson, P. Boyanova, M. Kronbichler, M. Neytcheva, and X. Wu. Numerical and computational efficiency of solvers for two-phase problems. Comput. Math. Appl., 65:301–314, 2013.

[3] J.W. Barrett, R. Nurnberg, and V. Styles. Finite element approximation of a phase field model for void electromigration. SIAM J. Numer. Anal., 42(2):738–772, 2005.

[4] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. Acta Numerica, 14:1–137, 2005.

[5] P. Boyanova, M. Do-Quang, and M. Neytcheva. Efficient preconditioners for large scale binary Cahn-Hilliard models. Comput. Methods Appl. Math., 12(1):1–22, 2012.
[6] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods (Third Edition).* Springer-Verlag, New York, 2008.

[7] J.W. Cahn. On spinodal decomposition. *Acta Metall.*, 9:795, 1961.

[8] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. I. interfacial free energy. *J. Chem. Phys.*, 28:258, 1958.

[9] H. D. Ceniceros and A. M. Roma. A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation. *J. Comput. Phys.*, 225:1849–1862, 2007.

[10] R. Choksi, M. Maras, and J. F. Williams. 2d phase diagram for minimizers of a Cahn-Hilliard functional with long-range interactions. *SIAM J. Appl. Dyn. Sys.*, 10(4):1344–1362, 2011.

[11] A. Diegel, X. Feng, and S. M. Wise. Analysis of a mixed finite element method for a Cahn-Hilliard-Darcy-Stokes system. *SIAM J. Numer. Anal.*, 53(1):127–152, 2015.

[12] A. Diegel, C. Wang, and S. M. Wise. Stability and convergence of a second-order mixed finite element method for the Cahn-Hilliard equation. *IMA J. Numer. Anal.*, 36:1867–1897, 2016.

[13] Q. Du, M. Li, and C. Liu. Analysis of a phase field Navier-Stokes vesicle-fluid interaction model. *Discrete Contin. Dyn. Syst. Ser. B*, 8(3):539, 2007.

[14] I. Ekeland and R. Téman. *Convex Analysis and Variational Problems.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[15] C.M. Elliott and S. Zheng. On the Cahn-Hilliard equation. *Arch. Ration. Mech. Anal.*, 96:339–357, 1986.

[16] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics.* Oxford University Press, Oxford, second edition, 2014.

[17] D. Eyre. Unconditionally Gradient Stable Time Marching the Cahn-Hilliard Equation. In J W Bullard, R Kalia, M Stoneham, and L Q Chen, editors, *Computational and Mathematical Models of Microstructural Evolution*, volume 53, pages 1686–1712, Warrendale, PA, USA, 1998. Materials Research Society.

[18] X. Feng. Fully discrete finite element approximations of the Navier-Stokes-Cahn-Hilliard diffuse interface model for two-phase fluid flows. *SIAM J. Numer. Anal.*, 44:1049–1072, 2006.

[19] A. Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, Philadelphia, 1997.

[20] W. Hackbusch. *Multi-grid Methods and Applications.* Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1985.

[21] D. Kay and R. Welford. A multigrid finite element solver for the Cahn–Hilliard equation. *J. Comput. Phys.*, 212:288–304, 2006.

[22] J. Kim. A numerical method for the Cahn–Hilliard equation with a variable mobility. *Commun. Nonlinear Sci. Numer. Simul.*, 12:1560–1571, 2007.

[23] J. Kim, K. Kang, and J. Lowengrub. Conservative multigrid methods for Cahn-Hilliard fluids. *J. Comput. Phys.*, 193:511–543, 2004.
[24] C. Lee, D. Jeong, J. Shin, Y. Li, and J. Kim. A fourth-order spatial accurate and practically stable compact scheme for the Cahn–Hilliard equation. *Physica A*, 409:17–28, 2014.

[25] D. Lee, J. Huh, D. Jeong, J. Shin, A. Yun, and J. Kim. Physical, mathematical, and numerical derivations of the Cahn–Hilliard equation. *Comp Mater Sci*, 81:216–225, 2014.

[26] H. G. Lee, J. S. Lowengrub, and J. Goodman. Modeling pinchoff and reconnection in a Hele-Shaw cell. I. the models and their calibration. *Phys. Fluids*, 14:492–513, 2002.

[27] K.-A. Mardal and R. Winther. Preconditioning discretizations of systems of partial differential equations. *Numer. Linear Algebra Appl.*, 18:1–40, 2011.

[28] J. Shin, S. Kim, D. Lee, and J. Kim. A parallel multigrid method of the Cahn–Hilliard equation. *Comp Mater Sci*, 71:89–96, 2013.

[29] R. H. Stogner, G. F. Carey, and B. T. Murray. Approximation of Cahn–Hilliard diffuse interface models using parallel adaptive mesh refinement and coarsening with C1 elements. *Internat. J. Numer. Methods Engrg.*, 76:636–661, 2008.

[30] R. Temam. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*. Springer-Verlag, New York, 1988.

[31] G. Tierra and F. Guillén-González. Numerical methods for solving the Cahn–Hilliard equation and its applicability to related energy-based models. *Arch Comput Method E*, 22:269–289, 2015.

[32] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.

[33] S. van Teeffelen, R. Backofen, A. Voigt, and H. Löwen. Derivation of the phase-field-crystal model for colloidal solidification. *Phys. Rev. E*, 79:051404, 2009.

[34] S.W. Walker. FELICITY: Finite ELement Implementation and Computational Interface Tool for You. [http://www.mathworks.com/matlabcentral/fileexchange/31141-felicity](http://www.mathworks.com/matlabcentral/fileexchange/31141-felicity).

[35] W. Wang, L. Chen, and J. Zhou. Postprocessing mixed finite element methods for solving Cahn–Hilliard equation: Methods and error analysis. *J Sci Comput*, 67:724–746, 2016.

[36] S.M. Wise. Unconditionally stable finite difference, nonlinear multigrid simulation of the Cahn-Hilliard-Hele-Shaw system of equations. *J. Sci. Comput.*, 44:38–68, 2010.

[37] J. Zhou, L. Chen, Y. Huang, and W. Wang. An efficient two-grid scheme for the Cahn-Hilliard equation. *Commun Comput Phys*, 17:127–145, 2015.