SIAM J. SCI. COMPUT.
Vol. 37, No. 5, pp. S216–S243

PRECONDITIONING FOR VECTOR-VALUED CAHN–HILLIARD EQUATIONS*

JESSICA BOSCH† AND MARTIN STOLL†

Abstract. The solution of vector-valued Cahn–Hilliard systems is of interest in many applications. We discuss strategies for the handling of smooth and nonsmooth potentials as well as for different types of constant mobilities. Concerning the nonsmooth systems, the necessary bound constraints are incorporated via the Moreau–Yosida regularization technique. We develop effective preconditioners for the efficient solution of the linear systems in saddle point form. Numerical results illustrate the efficiency of our approach. In particular, we numerically show mesh and phase independence of the developed preconditioner in the smooth case. The results in the nonsmooth case are also satisfying, and the preconditioned version always outperforms the unpreconditioned one.

Key words. Cahn–Hilliard systems, vector-valued problems, preconditioning, Schur complement approximation, Krylov subspace solver, Moreau–Yosida regularization, semismooth Newton method

AMS subject classifications. 65F08, 65F10, 65N22, 65F50, 93C20, 35R35, 74S05, 35K55, 82C26

DOI. 10.1137/14M0973633

1. Introduction.

The Cahn–Hilliard equation is a partial differential equation of fourth order which is used in materials science [50, 32], image processing [21], or chemistry [63]. It was originally introduced to model phase separation in binary alloys [42, 18] that occurs when the temperature of a homogeneous mixture is rapidly quenched below a critical temperature. In practice, often more than two phases occur, see, e.g., [48, 27, 24, 23, 7, 45, 34], and the phase field model has been extended to deal with multicomponent systems. A vector-valued order parameter $\mathbf{u} = (u_1, \ldots, u_N)^T \colon \Omega \times (0,T) \to \mathbb{R}^N$ is introduced, where $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ is a bounded domain, $T > 0$ is an arbitrary but fixed time, and N is the number of phases. Each u_i describes the fraction of one phase, i.e., if $u_i(\mathbf{x}, t) = 0$, then phase i is absent in $\mathbf{x}$, and if $u_i(\mathbf{x}, t) = 1$, only phase i is present there. Hence,

$$\sum_{i=1}^{N} u_i = 1 \tag{1.1}$$

and $u_i \geq 0$ is required, so that admissible states belong to the Gibbs simplex

$$\mathcal{G}^N := \left\{ \mathbf{v} \in \mathbb{R}^N \;\middle|\; \sum_{i=1}^{N} v_i = 1,\ v_i \geq 0 \text{ for } i = 1, \ldots, N \right\}. \tag{1.2}$$

We study a diffuse phase transition, i.e., the region between the phases has a certain width b, the so-called interface (phase field model). There is also the limit case $b \downarrow 0$

*Received by the editors June 18, 2014; accepted for publication (in revised form) March 23, 2015; published electronically October 29, 2015. This research was supported by a research grant of the International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg).
http://www.siam.org/journals/sisc/37-5/M97363.html
†Numerical Linear Algebra for Dynamical Systems, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (bosch@mpi-madgeburg.mpg.de, stollm@mpi-madgeburg.mpg.de).

which gives the sharp interface model [33, 32, 52]. The motion of the interfaces separating N components can be modeled with the Ginzburg–Landau energy

$$\mathcal{E}(\mathbf{u}) = \int_\Omega \left\{ \frac{\varepsilon^2}{2} \sum_{i=1}^{N} |\nabla u_i|^2 + \psi(\mathbf{u}) \right\} \mathrm{d}\mathbf{x},$$

where $\varepsilon > 0$ is the gradient energy coefficient. The potential function $\psi\colon \mathbb{R}^N \to \mathbb{R}_0^+ \cup \{\infty\}$ gives rise to phase separation. It can be modeled by a smooth free energy, e.g., using multiwell potentials [22] such as

$$\psi(\mathbf{u}) := \frac{1}{4} \sum_{i=1}^{N} u_i^2 (1 - u_i)^2, \tag{1.3}$$

or by a nonsmooth multiobstacle potential [4]

$$\psi(\mathbf{u}) := \begin{cases} \psi_0(\mathbf{u}) = -\frac{1}{2}\mathbf{u} \cdot A\mathbf{u}, & \mathbf{u} \in \mathcal{G}^N, \\ \infty, & \text{otherwise}, \end{cases} \tag{1.4}$$

where the symmetric matrix $A \in \mathbb{R}^{N \times N}$ contains constant interaction parameters A_{ij}. From physical considerations, A must have at least one positive eigenvalue. A typical choice is $A = I - \mathbf{1}\mathbf{1}^T$ with $\mathbf{1} = (1, \ldots, 1)^T$ and the identity matrix $I \in \mathbb{R}^{N \times N}$, which means that the interaction between all different components is equal and no self-interaction occurs. Other possible potentials are logarithmic ones; see, e.g., [3]. This work deals with the two types of potential (1.3) and (1.4). Smooth potentials are used for shallow temperature quenches. For the deep quench limit, i.e., a very rapid cooling of the mixture, multiobstacle potentials have to be used, which in turn lead to systems of variational inequalities. Motivated by the work of Hintermüller, Hinze, and Tber [43] as well as our previous studies [12, 11], all of them considering scalar, nonsmooth Cahn–Hilliard systems, we incorporate the bound constraints via the Moreau–Yosida regularization technique and solve the resulting subproblems by a semismooth Newton (SSN) method.

As we show in the course of this paper, the solution of a linear system $\mathcal{K}x = b$ with a real nonsymmetric matrix $\mathcal{K}$ is at the heart of this method. The sparse linear systems are usually of very large dimension, and in combination with three-dimensional experiments, the application of direct solvers such as UMFPACK [20] becomes infeasible. As a result, iterative methods have to be employed (see, e.g.,, [39, 55] for introductions to this field). We propose the use of a Krylov subspace solver. The convergence behavior of the iterative scheme typically depends on the conditioning of the problem and the clustering of the eigenvalues. These properties can be enhanced using preconditioning techniques $\mathcal{P}^{-1}\mathcal{K}x = \mathcal{P}^{-1}b$, where $\mathcal{P}$ is an invertible matrix that is easy to invert and that resembles $\mathcal{K}$. In this paper, we provide efficient preconditioners $\mathcal{P}$ for the solution of Cahn–Hilliard variational (in-)equalities using an effective Schur complement approximation and (algebraic) multigrid developed for elliptic systems [29, 55, 54].

The paper is organized as follows. In section 2, we derive the vector-valued Cahn–Hilliard systems for the use of the smooth potential (1.3). These are discretized in time in section 3, and stability and uniqueness conditions for the time step size are derived. Section 4 presents the systems with the nonsmooth potential (1.4) and their handling with the Moreau–Yosida regularization technique. Section 5 shortly introduces the SSN method to solve the regularized subproblems. The linear systems arising from

the discretization using finite elements are derived in section 6. In section 7, we analyze the linear systems and propose preconditioning strategies for the saddle point problems. The algorithms for the solution and preconditioning procedures are presented in section 8. Section 9 illustrates the efficiency of our approach. In section 10, we discuss alternative approaches, and finally, section 11 summarizes our findings.

2. Derivation. The evolution of $\mathbf{u}$ is governed by the H^{-1}-gradient of the Ginzburg–Landau energy under the constraint (1.1), which has to hold everywhere at any time. Using the smooth potential (1.3), the vector-valued Cahn–Hilliard equations read

$$\partial_t u_i = (L\Delta \mathbf{w})_i, \tag{2.1}$$

$$w_i = f(u_i) + \beta(\mathbf{u}) - \varepsilon^2 \Delta u_i, \tag{2.2}$$

$$\nabla u_i \cdot \mathbf{n} = (L\nabla \mathbf{w})_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega \tag{2.3}$$

for $i = 1, \dots, N$. The matrix $L = (L_{ij})_{i,j=1,\dots,N} \in \mathbb{R}^{N\times N}$ is the mobility matrix and

$$\mathbf{f}(\mathbf{u}) = (f(u_1), \dots, f(u_N))^T := \left(\frac{\partial \psi}{\partial u_1}, \dots, \frac{\partial \psi}{\partial u_N}\right)^T = \frac{\partial \psi}{\partial \mathbf{u}},$$

in which $f(u_i) = u_i^3 - \frac{3}{2}u_i^2 + \frac{1}{2}u_i$, and $\beta(\mathbf{u}) := -\frac{1}{N}\sum_{i=1}^N f(u_i)$. In the process, the chemical potentials $\mathbf{w} = (w_1, \dots, w_N)^T$ result from the variational derivative of the energy $\mathcal{E}$. In doing so, admissible variations $\mathbf{d} = (d_1, \dots, d_N)^T$ of $\mathbf{u}$ have to fulfill $\sum_{i=1}^N d_i = 0$ in order to ensure (1.1). This explains the presence of the term $\beta(\mathbf{u})$; see also [47] for a more detailed calculation. Equation (2.3) contains the natural zero Neumann boundary condition $\nabla u_i \cdot \mathbf{n} = 0$ on $\partial\Omega$ as well as the mass conserving boundary condition $(L\nabla \mathbf{w})_i = 0$ on $\partial\Omega$, $i = 1, \dots, N$. We obtain $\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u_i \,\mathrm{d}\mathbf{x} = 0$, by applying Gauss's theorem in (2.1) together with the latter boundary condition. In other words, the total mass of each phase is conserved. We refer to [24] for a detailed development of the vector-valued Cahn–Hilliard equations.

The coefficients L_{ij} may depend on $\mathbf{u}$ (see, e.g., [23]), but this work deals with constant L_{ij}. In order to ensure the constraint (1.1), a common way in the literature is to assume that L is symmetric and $L\mathbf{1} = \mathbf{0}$ (see, e.g., [24, 9]), since summing (2.1) over $i = 1, \dots, N$ leads then to

$$\frac{\partial}{\partial t}\sum_{i=1}^N u_i = \sum_{i=1}^N \frac{\partial u_i}{\partial t} = \sum_{i=1}^N \nabla \cdot (L\nabla \mathbf{w})_i = \nabla \cdot \sum_{i,j=1}^N L_{ij}\nabla w_j = \nabla \cdot \sum_{j=1}^N \nabla w_j \sum_{i=1}^N L_{ij} = 0.$$

Therefore, (1.1) is fulfilled during the evolution if $\sum_{i=1}^N u_i = 1$ at time 0. It is further assumed that L is positive semidefinite, as differentiating the energy $\mathcal{E}$ gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\mathbf{u}) = -\int_\Omega \sum_{i=1}^N \nabla w_i \cdot (L\nabla \mathbf{w})_i \,\mathrm{d}\mathbf{x} \le 0,$$

where we have used Green's first identity. Therefore, the total energy is nonincreasing in time.

Remark 2.1. We have explained above the common assumption $L\mathbf{1} = \mathbf{0}$. Nevertheless, it is possible to work with $L = I$ for convenience; see, e.g., [46]. Therefore, we also consider this case in our work and our numerical solver simplifies.

As already mentioned, this work focuses on constant mobility matrices. Additionally, we restrict the class of considerations to diagonal and circulant ones. More general forms are a topic of further research and are not discussed in the present paper. As reference examples, we take $L = I$ (used, e.g., in [46]) and $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ (used, e.g., in [23, 38]).

We now want to discretize the problem (2.1)–(2.3) (in weak formulation) in time and give stability and uniqueness conditions.

3. Time step conditions for the smooth potential. Concerning the time, fully implicit discretizations are the most accurate; see, e.g., [8, 17, 12]. Let $\tau > 0$ denote the time step size and $n \in \mathbb{N}$ the time step. We use the backward Euler discretization for the time derivatives $\partial_t u_i$, $i = 1, \dots, N$, and treat all the other terms implicitly. Then, by considering the weak formulation of (2.1)–(2.3), for every time step we have to solve the time-discrete systems

$$\left(u_i - u_i^{(n-1)}, v\right) + \tau \left((L\nabla \mathbf{w})_i, \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{3.1}$$

$$-(w_i, v) + \varepsilon^2 (\nabla u_i, \nabla v) + (f(u_i), v) - \frac{1}{N}\left(\sum_{j=1}^{N} f(u_j), v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{3.2}$$

$i = 1, \dots, N$, where we write $\mathbf{u}^{(n)} = \mathbf{u}$ and $\mathbf{w}^{(n)} = \mathbf{w}$. Here, $(\cdot, \cdot)$ stands for the $L^2(\Omega)$-inner product. Now, we want to give stability and uniqueness conditions for the time step. However, the quartic growth of $\psi(\mathbf{u})$ at infinity introduces various technical difficulties in the analysis. Therefore, we consider a truncated multiwell potential. To be more precise, we restrict the growth of $\psi(\mathbf{u})$ to be quadratic for $u_i \le 1 - M$ and $u_i \ge M$ for a given constant M. This is in fact a common practice (see, e.g., [57] and references therein), and the authors of [57] have proved stability conditions for scalar Allen–Cahn and Cahn–Hilliard equations with a truncated double-well potential. Using this technique, we get the following condition: There exists a constant T such that

$$\max_{\mathbf{s} \in \mathbb{R}^N} \left| \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{s}) \right| \le T \quad \forall i = 1, \dots, N. \tag{3.3}$$

With the use of (3.3), we can prove the following.

THEOREM 3.1. *The solution of* (3.1)–(3.2) *is unique provided that* $\tau < \frac{4\varepsilon^2 \lambda_{\min}}{T^2 \|L\|^2}$, *where* $\|\cdot\|$ *denotes the spectral radius and* $\lambda_{\min}$ *the smallest positive eigenvalue of* L.

THEOREM 3.2. *Under the condition* $\tau < \frac{8\varepsilon^2 \lambda_{\min}}{T^2 \|L\|^2}$, *the time discretization scheme* (3.1)–(3.2) *is energy stable, i.e., its solution satisfies* $\mathcal{E}(\mathbf{u}) \le \mathcal{E}(\mathbf{u}^{(n-1)})$ *for all* $n \ge 1$.

The proofs can be found in Appendices A and B. The resulting time step restrictions are similar to the stability condition obtained in [57]. Although these conditions appear to be quite restrictive for $\varepsilon \ll 1$, the authors of [57] pointed out that they are in fact needed for the sake of convergence. Moreover, note that explicit schemes usually lead to very severe time step restrictions of order $\mathcal{O}(h^4)$.

The final steps to get to the discrete linear systems of equations are straightforward. The nonlinear systems (3.1)–(3.2) are solved by standard Newton methods and discretized by finite element methods; see section 6. In the following, we concentrate on the multiobstacle potential (1.4) and shortly present the used technique that deals with the nonsmoothness.

4. Moreau–Yosida regularization. As motivated in [43, 12, 11], we incorporate the bound constraints $\mathbf{u} \geq \mathbf{0}$ a.e. with the Moreau–Yosida regularization technique. Instead of the energy functional, $\mathcal{E}$ we consider

$$\mathcal{E}_\nu(\mathbf{u}_\nu) = \int_\Omega \left\{ \frac{\varepsilon^2}{2} \sum_{i=1}^N |\nabla u_{\nu,i}|^2 + \psi_0(\mathbf{u}_\nu) + \frac{1}{2\nu} \sum_{i=1}^N |\min(0, u_{\nu,i})|^2 \right\} \mathrm{d}\mathbf{x},$$

where $0 < \nu \ll 1$ denotes the penalty parameter. As done for the smooth potential in the two sections before, we can now derive the time-discrete systems (using an implicit discretization scheme) for the nonsmooth potential:

$$\left(u_{\nu,i} - u_i^{(n-1)}, v\right) + \tau \left((L\nabla \mathbf{w}_\nu)_i, \nabla v\right) = 0 \quad \forall v \in H^1(\Omega), \tag{4.1}$$

$$\begin{aligned}(w_{\nu,i}, v) - \varepsilon^2(\nabla u_{\nu,i}, \nabla v) + ((A\mathbf{u}_\nu)_i, v) - \frac{1}{\nu}(\min(0, u_{\nu,i}), v) \\ + \frac{1}{N} \sum_{j=1}^N \left[\frac{1}{\nu}(\min(0, u_{\nu,j}), v) - ((A\mathbf{u}_\nu)_j, v) \right] = 0 \quad \forall v \in H^1(\Omega),\end{aligned} \tag{4.2}$$

$i = 1, \ldots, N$. Next, we solve (4.1)–(4.2) by an SSN method which is motivated in [43, 12, 11] and shortly summarized in the following section.

5. Semismooth Newton method. For a specified sequence $\nu \to 0$, we solve the system (4.1)–(4.2), which can be compactly written as $\mathbf{F}_\nu(\mathbf{u}_\nu, \mathbf{w}_\nu) = \mathbf{0}$ for every ν by an SSN algorithm; see also [44]. Due to the presence of the minimum operator, $\mathbf{F}_\nu$ is not Fréchet-differentiable. However, the minimum operator satisfies the weaker notion of Newton differentiability; see [44, 43].

DEFINITION 5.1 (Definition 5.1 in [43]). *Let X and Z be Banach spaces, $D \subset X$ an open subset. A mapping $F\colon D \subset X \to Z$ is called Newton-differentiable in $U \subset D$ if there exists a family of mappings $G\colon U \to Z$ such that*

$$\lim_{d \to 0} \frac{\|F(x+d) - F(x) - G(x+d)d\|_Z}{\|d\|_X} = 0 \quad \forall x \in U.$$

The operator G is called a Newton derivative of F on U.

In general, for a Newton-differentiable mapping F with a Newton derivative G, the SSN iteration is given as

$$x^{(k+1)} = x^{(k)} - G\left(x^{(k)}\right)^{-1} F(x^{(k)}), \quad k = 0, 1, \ldots. \tag{5.1}$$

Given a sufficiently close initial guess $x^{(0)}$, [44, Theorem 1.1] shows superlinear convergence of the sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$, generated by (5.1), to the solution of $F(x) = 0$.

Regarding the scalar two-component Cahn–Hilliard equation, the Newton differentiability of the arising mapping F_ν as well as the superlinear convergence of the corresponding SSN iteration are proved in [43] for the semi-implicit time-discrete system and extended to the implicit time-discrete system in [12]. Both works are based on the Newton derivative of the minimum mapping $\min(0, \cdot)\colon H^1(\Omega) \to H^1(\Omega)^*$, which is given as

$$G_{\min}(y)(x) = \begin{cases} 1 & \text{if } y(x) \leq 0, \\ 0 & \text{if } y(x) > 0; \end{cases}$$

see [44, Proposition 4.1] and [43, Lemma 5.3]. We use this result for the Newton derivative of every minimum operator in our mapping $\mathbf{F}_\nu$ given in (4.1)–(4.2). Regarding the ith component, this leads us to the Newton derivative

$$\begin{aligned}\Big\langle G_\nu^{(i)}(\mathbf{u},\mathbf{w})(\delta\mathbf{u},\delta\mathbf{w}),(\phi,\psi)\Big\rangle &= \begin{pmatrix} \tau((L\nabla\delta\mathbf{w})_i,\nabla\phi) \\ (\delta w_i,\psi)-\varepsilon^2(\nabla\delta u_i,\nabla\psi)+((A\delta\mathbf{u})_i,\psi)-\frac{1}{\nu}(\chi_{\mathcal{A}(u_i)}\delta u_i,\psi)\end{pmatrix} \\ &\quad + \begin{pmatrix} (\delta u_i,\phi) \\ \frac{1}{N}\sum_{j=1}^N\left[\frac{1}{\nu}(\chi_{\mathcal{A}(u_j)}\delta u_j,\psi)-((A\delta\mathbf{u})_j,\psi)\right]\end{pmatrix},\end{aligned}$$

where $\chi_{\mathcal{A}(u_i)}$ is the characteristic function of the set

$$\mathcal{A}(u_i) := \{x\in\Omega : u_i(x)<0\}.$$

We now want to discretize the time-discrete problem (4.1)–(4.2) in space and then discuss its efficient solution.

6. Finite element approximation. For the discretization in space we use finite elements [59]. In the following, we assume for simplicity that Ω is a polyhedral domain. Generalizations to curved domains are possible using boundary finite elements with curved faces. Let $\{\mathcal{R}_h\}_{h>0}$ be a triangulation of Ω into disjoint open rectangular elements with maximal element size h, J_h be the set of nodes of $\mathcal{R}_h$, and $p_j\in J_h$ be the coordinates of these nodes. The use of rectangles is motivated by performing the implementation with deal.II [2]. We approximate the infinite-dimensional space $H^1(\Omega)$ by the finite-dimensional space

$$S_h := \left\{\phi\in C^0(\overline{\Omega})\colon \phi|_R\in Q_1(R)\ \ \forall R\in\mathcal{R}_h\right\}\subset H^1(\Omega)$$

of continuous, piecewise multilinear functions. We denote the standard nodal basis functions of S_h by φ_j for all $j\in J_h$. The discretized version of the penalized problem (4.1)–(4.2) consists in finding $(\mathbf{u}_{\nu,h},\mathbf{w}_{\nu,h})\in S_h^N\times S_h^N$ such that

$$\langle\mathbf{F}_{\nu,h}(\mathbf{u}_{\nu,h},\mathbf{w}_{\nu,h}),\mathbf{v}_h\rangle=\mathbf{0}\quad\forall\mathbf{v}_h\in S_h^N, \tag{6.1}$$

where the components are

$$\begin{aligned}\Big\langle F_{\nu,h}^{(1,i)}(\mathbf{u}_{\nu,h},\mathbf{w}_{\nu,h}),v_h\Big\rangle &= \Big(u_{\nu,h,i}-u_{h,i}^{(n-1)},v_h\Big)_h+\tau\left((L\nabla\mathbf{w}_{\nu,h})_i,\nabla v_h\right),\\ \Big\langle F_{\nu,h}^{(2,i)}(\mathbf{u}_{\nu,h},\mathbf{w}_{\nu,h}),v_h\Big\rangle &= (w_{\nu,h,i},v_h)_h-\varepsilon^2(\nabla u_{\nu,h,i},\nabla v_h)+((A\mathbf{u}_{\nu,h})_i,v_h)_h\\ &\quad-\frac{1}{\nu}(\min(0,u_{\nu,h,i}),v_h)_h\\ &\quad+\frac{1}{N}\sum_{j=1}^N\left[\frac{1}{\nu}(\min(0,u_{\nu,h,j}),v_h)_h-((A\mathbf{u}_{\nu,h})_j,v_h)_h\right]\end{aligned}$$

for $i=1,\dots,N$. The semi-inner product $(\cdot,\cdot)_h$ on $C_0(\overline{\Omega})$ is defined by

$$(f,g)_h := \int_\Omega \pi_h(f(\mathbf{x})g(\mathbf{x}))\,\mathrm{d}\mathbf{x}=\sum_{i=1}^m(1,\varphi_i)f(p_i)g(p_i)\quad\forall f,g\in C_0(\overline{\Omega}),$$

where $\pi_h\colon C_0(\overline{\Omega})\to S_h$ is the Lagrange interpolation operator. Within our finite element framework, for a given $(\mathbf{u}_h,\mathbf{w}_h)\in S_h^N\times S_h^N$, every step of the SSN method

for solving (6.1) requires computing $(\delta \mathbf{u}_h, \delta \mathbf{w}_h) \in S_h^N \times S_h^N$ satisfying

$$\begin{aligned}
(\delta u_{h,i}, v_h)_h + \tau\left((L\nabla \delta \mathbf{w}_h)_i, \nabla v_h\right) &= -F_{\nu,h}^{(1,i)}(\mathbf{u}_h, \mathbf{w}_h),\\
(\delta w_{h,i}, v_h)_h - \varepsilon^2(\nabla \delta u_{h,i}, \nabla v_h) + ((A\delta \mathbf{u}_h)_i, v_h)_h - \frac{1}{\nu}(\chi^h_{\mathcal{A}(u_{h,i})}\delta u_{h,i}, v_h)_h&\\
+ \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1}{\nu}(\chi^h_{\mathcal{A}(u_{h,j})}\delta u_{h,j}, v_h)_h - ((A\delta \mathbf{u})_{h,j}, v_h)_h\right]&\\
= -F_{\nu,h}^{(2,i)}(\mathbf{u}_h, \mathbf{w}_h)&
\end{aligned}$$

for all $v_h \in S_h$ and $i = 1, \ldots, N$, where $\chi^h_{\mathcal{A}(u_{h,i})} := \sum_{j=1}^m \chi^h_{\mathcal{A}(u_{h,i})}(p_j)\varphi_j$ with $\chi^h_{\mathcal{A}(u_{h,i})}(p_j) = 0$ if $u_{h,i}(p_j) \geq 0$ and $\chi^h_{\mathcal{A}(u_{h,i})}(p_j) = 1$ otherwise. If we now write a function $v_h \in S_h$ by $v_h = \sum_{j\in J_h} v_{h,j}\varphi_j$ and denote the vector of coefficients by $\mathbf{v}$, the fully discrete linear systems (smooth and nonsmooth) read in matrix form as

$$\begin{bmatrix} I \otimes M & -\mathcal{B} \\ \tau L \otimes K & I \otimes M \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k+1)} \\ \mathbf{u}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ (I \otimes M)\,\mathbf{u}^{(n-1)} \end{bmatrix}, \tag{6.2}$$

where k denotes the Newton step. The first right-hand side is

$$\mathbf{b} = (I \otimes M)\left(-2\left(\mathbf{u}^{(k)}\right)^3 + \frac{3}{2}\left(\mathbf{u}^{(k)}\right)^2\right) + \frac{1}{N}(I \otimes M)\left(\sum_{j=1}^{N} 2\left(u_j^{(k)}\right)^3 - \frac{3}{2}\left(u_j^{(k)}\right)^2\right)\mathbf{1}$$

for the use of the smooth potential and

$$\mathbf{b} = \mathbf{0}$$

for the use of the nonsmooth potential. Further, $K := ((\nabla\varphi_i, \nabla\varphi_j))_{i,j=1,\ldots,m} \in \mathbb{R}^{m\times m}$ is the stiffness matrix, $M := ((\varphi_i, \varphi_j)_h)_{i,j=1,\ldots,m} \in \mathbb{R}^{m\times m}$ is the lumped mass matrix, and $I \in \mathbb{R}^{N\times N}$ is the identity matrix. M is a symmetric positive definite diagonal matrix and K is symmetric and positive semidefinite. For $N = 3$, the block $\mathcal{B}$ is given as

$$\mathcal{B} = \begin{bmatrix} B_{11} & B_2 & B_3 \\ B_1 & B_{22} & B_3 \\ B_1 & B_2 & B_{33} \end{bmatrix},$$

where for $i = 1, \ldots, N$

$$\begin{aligned}
B_{ii} &= \varepsilon^2 K + \left(1 - \frac{1}{N}\right) F_i M F_i,\\
B_i &= -\frac{1}{N} F_i M F_i,\\
F_i &= \operatorname{diag}\left(3(u_i^{(k)}(p_j))^2 - 3u_i^{(k)}(p_j) + \frac{1}{2}\right)
\end{aligned} \tag{6.3}$$

in the smooth system and

$$\begin{aligned}
B_{ii} &= \varepsilon^2 K + \left(1 - \frac{1}{N}\right)\left(\frac{1}{\nu} G_i M G_i - M\right),\\
B_i &= -\frac{1}{N}\left(\frac{1}{\nu} G_i M G_i - M\right),\\
G_i &= \operatorname{diag}\begin{pmatrix} 1 & u_i^{(k)}(p_j) < 0, \\ 0 & \text{otherwise} \end{pmatrix}
\end{aligned} \tag{6.4}$$

in the nonsmooth system. Remember, this work considers the choice $A = I - \mathbf{1}\mathbf{1}^T$ as well as diagonal and circulant mobility matrices L. The two reference examples $L = I$ and $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ are used in the following to demonstrate the preconditioning technique. The system matrix in (6.2) is denoted by $\mathcal{K}$ for the remainder of the paper.

7. Preconditioning. In both cases, smooth and nonsmooth, a linear nonsymmetric system in saddle point form is at the heart of the computation. We propose the block-triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} I \otimes M & 0 \\ \tau L \otimes K & -\hat{\mathcal{S}} \end{bmatrix},$$

motivated by [25, 49], where $\hat{\mathcal{S}}$ is an approximation of the Schur complement $\mathcal{S} = I \otimes M + \tau(L \otimes K)(I \otimes M)^{-1}\mathcal{B}$. The preconditioned matrix becomes

$$\mathcal{P}^{-1}\mathcal{K} = \begin{bmatrix} I & -(I \otimes M)^{-1}\mathcal{B} \\ 0 & -\hat{\mathcal{S}}^{-1}\mathcal{S} \end{bmatrix}.$$

It has Nm eigenvalues at one, and the remaining ones are characterized as the eigenvalues of the matrix $\hat{\mathcal{S}}^{-1}\mathcal{S}$, which has for $\hat{\mathcal{S}}$ being a good approximation only a small number of different eigenvalue clusters. This in turn is known to result in only a few iterations of suitable Krylov subspace solvers until convergence [25, 49]. Therefore, the theoretical ideal choice is $\hat{\mathcal{S}} = \mathcal{S}$, since the generalized eigenvalue problem $\mathcal{K}x = \lambda\mathcal{P}x$ has in this case only two distinct eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -1$. But the application of the preconditioner $\mathcal{P}$ requires the action of the inverses of $I \otimes M$ and of $\hat{\mathcal{S}}$. From this point of view, the ideal choice $\hat{\mathcal{S}} = \mathcal{S}$ is not practical since this is a full matrix. Inverting the block $I \otimes M$ is cheap, as M is a nonsingular diagonal matrix.[1] The remaining task is now to create a Schur complement approximation $\hat{\mathcal{S}}$ that is easy to invert and resembles $\mathcal{S}$. The two difficult points thereby are the nondiagonal block matrices $\mathcal{B}$ and $L \otimes K$ which couple N equations, respectively. Concerning the latter, note that circulant matrices L can be diagonalized using the Fourier matrix; see [19]. We will see in the next section how we can take advantage of this property for the construction of preconditioners. The block matrix $\mathcal{B}$ contains the gradient energy parts (that only arise in the diagonal blocks) as well as the interacting terms coming from the potential. These include, in the case of the nonsmooth potential, the coupling of all penalization terms. In fact, the latter poses the most challenging part; see section 7.2 for details. Regarding the use of the smooth potential, we present in the following an efficient Schur complement approximation. In particular, this approximation is proved to be optimal for a high-quality simplification of the system matrix $\mathcal{K}$ when $L = I$ is used.

7.1. Schur complement preconditioner in the smooth case. The first step for the construction of a practical Schur complement preconditioner consists of the approximation of the nondiagonal block matrix $\mathcal{B}$. The specification of $\mathcal{B}$ given in (6.3) shows that the matrices F_i, $i = 1, \ldots, N$, only depend on the known solution $\mathbf{u}^{(k)}$ from the previous Newton step. From the constraints of the Gibbs simplex (1.2), we know $\mathbf{0} \lesssim \mathbf{u}^{(k)} \lesssim \mathbf{1}$. Therefore, every diagonal entry of F_i, $i = 1, \ldots, N$, approximately ranges from -2.5 up to 3.5. Together with the estimated order of entries $\mathcal{O}(M) = h^2$ for the mass matrix, we approximate $\mathcal{B}$ by $\hat{\mathcal{B}} = \varepsilon^2 I \otimes K$. In other words, we set

[1] For consistent mass matrices, the Chebyshev semi-iteration [36, 37] provides a powerful preconditioner [61, 53].

all blocks $F_i M F_i$, $i = 1, \dots, N$, to zero matrices, as their estimated order of entries ranges from $-3.5\,h^2$ up to $2.5\,h^2$, i.e., a small interval around zero. The resulting approximation $\mathcal{K}_0$ of $\mathcal{K}$ reads as

$$\mathcal{K}_0 = \begin{bmatrix} I \otimes M & -\varepsilon^2 I \otimes K \\ \tau L \otimes K & I \otimes M \end{bmatrix}. \tag{7.1}$$

Note that this is the approximation proposed in [14], which is proven to be of high-quality for small enough time steps. To be more precise, $\mathcal{K}_0^{-1}\mathcal{K}$ has mN eigenvalues at one and the remaining ones are characterized as $\lambda_i = \mu_i + 1$, $i = 1, \dots, mN$, where the μ_i's are eigenvalues of a matrix $\mathcal{Q}$ which can be bounded by $\|\mathcal{Q}\| \leq \frac{\sqrt{\tau}}{2\varepsilon}$. Therefore, μ_i can be made arbitrarily close to zero by choosing the time step $\sqrt{\tau}$ small enough compared to 2ε.

Next, we go over to the construction of the preconditioner $\hat{\mathcal{S}}$ for the Schur complement $\mathcal{S}_0 = I \otimes M + \tau\varepsilon^2 (L \otimes K)(I \otimes M)^{-1}(I \otimes K)$ of $\mathcal{K}_0$. The main idea, as motivated in [51], is the following: Construct a preconditioner of the form $\hat{S} = ABC$ with A, B, and C symmetric positive definite, such that the exact Schur complement is captured as close as possible. Therefore, we propose the following Schur complement preconditioner:

$$\begin{aligned} \hat{\mathcal{S}} = \hat{\mathcal{S}}_1 (I \otimes M)^{-1} \hat{\mathcal{S}}_2 &= \left(I \otimes M + \sqrt{\tau}\varepsilon (L \otimes K)\right) (I \otimes M)^{-1} \left(I \otimes M + \sqrt{\tau}\varepsilon (I \otimes K)\right) \\ &= I \otimes M + \tau\varepsilon^2 (L \otimes K)(I \otimes M)^{-1}(I \otimes K) + \sqrt{\tau}\varepsilon (L \otimes K) + \sqrt{\tau}\varepsilon (I \otimes K). \end{aligned} \tag{7.2}$$

The first two terms in (7.2) match the Schur complement $\mathcal{S}_0$. Due to the balanced distribution of $\tau\varepsilon^2$ in the form of $\sqrt{\tau}\varepsilon$ in both factors, $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$, the influence of both remainder terms in (7.2) is reduced. In the case $L = I$, this approximation turns out to be an optimal Schur complement preconditioner for $\mathcal{S}_0$ (see also [51]).

LEMMA 7.1. *If $L = I$, then the eigenvalues of $\hat{\mathcal{S}}^{-1}\mathcal{S}_0$ are contained within the following interval:*

$$\lambda(\hat{\mathcal{S}}^{-1}\mathcal{S}_0) \in [0.5, 1]\,.$$

Proof. As both $\mathcal{S}_0$ and $\hat{\mathcal{S}}$ are symmetric in the case $L = I$, we may prove the result using a Rayleigh quotient argument. We write

$$\begin{aligned} \frac{\mathbf{v}^\top \mathcal{S}_0 \mathbf{v}}{\mathbf{v}^\top \hat{\mathcal{S}} \mathbf{v}} &= \frac{\mathbf{v}^\top \left(I \otimes M + \tau\varepsilon^2 (I \otimes K)(I \otimes M)^{-1}(I \otimes K)\right) \mathbf{v}}{\mathbf{v}^\top \left(I \otimes M + \tau\varepsilon^2 (I \otimes K)(I \otimes M)^{-1}(I \otimes K) + 2\sqrt{\tau}\varepsilon (I \otimes K)\right) \mathbf{v}} \\ &= \frac{\mathbf{a}^\top \mathbf{a} + \mathbf{b}^\top \mathbf{b}}{\mathbf{a}^\top \mathbf{a} + \mathbf{b}^\top \mathbf{b} + 2\mathbf{a}^\top \mathbf{b}}, \end{aligned}$$

where $\mathbf{a} = (I \otimes M)^{\frac{1}{2}} \mathbf{v}$ and $\mathbf{b} = \sqrt{\tau}\varepsilon (I \otimes M)^{-\frac{1}{2}} (I \otimes K)\mathbf{v}$. From the properties of M and K, it follows $\mathbf{a}^\top \mathbf{a} > 0$ and $\mathbf{b}^\top \mathbf{b}, \mathbf{a}^\top \mathbf{b} \geq 0$, and therefore $\frac{\mathbf{v}^\top \mathcal{S}_0 \mathbf{v}}{\mathbf{v}^\top \hat{\mathcal{S}} \mathbf{v}} \leq 1$. On the other hand, $(\mathbf{a} - \mathbf{b})^\top (\mathbf{a} - \mathbf{b}) \geq 0$, which gives $\frac{\mathbf{v}^\top \mathcal{S}_0 \mathbf{v}}{\mathbf{v}^\top \hat{\mathcal{S}} \mathbf{v}} \geq 0.5$. ∎

Let us now discuss the action of the inverse of $\hat{\mathcal{S}}$ which consists of the action of the inverses of the block matrix $\hat{\mathcal{S}}_1$ and of the diagonal block matrix $\hat{\mathcal{S}}_2$ as well as cheap multiplications with the mass matrix M. The block $\hat{\mathcal{S}}_2$ is of block-diagonal form and contains the same elliptic operator on each diagonal block. Therefore, we approximate the inverse of each diagonal block with one and the same algebraic multigrid (AMG)

preconditioner.[2] Regarding $\hat{\mathcal{S}}_1$, this block is in fact equal to $\hat{\mathcal{S}}_2$ if the mobility matrix $L = I$ is used. In total, for preconditioning the system matrix $\mathcal{K}$, which is of size $2mN$, we need to apply only two AMG preconditioners of size m each plus multiplications with the diagonal mass matrix. As long as the mesh does not change, we do not have to recompute them. In this sense, the application of the preconditioner $\mathcal{P}$ is independent of the number of phases.

Now, let us study the case if $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ and therefore $\hat{\mathcal{S}}_1$ is not of block-diagonal form anymore. But as L is a circulant matrix, it can be diagonalized using the Fourier matrix F, i.e.,

$$L = F \operatorname{diag}(\lambda_1, \dots, \lambda_N)\, F^H;$$

see [19]. This property forms the basis of an efficient fast Fourier transform (FFT) based preconditioner which is used, e.g., in [58] and briefly reviewed in the following. The idea is to diagonalize not only L but also the whole block matrix $\hat{\mathcal{S}}_1$ (which contains L) since the latter is the matrix whose inverse we have to apply. More precisely, if we apply the FFT to the system $\hat{\mathcal{S}}_1 y = g$, we get an equivalent system with the block-diagonal system matrix

$$(F^H \otimes I)\hat{\mathcal{S}}_1(F \otimes I) = I \otimes M + \sqrt{\tau}\varepsilon \operatorname{diag}(\lambda_1, \dots, \lambda_N) \otimes K. \tag{7.3}$$

Inserting the eigenvalues of L, which are $\lambda_1 = 0$ and $\lambda_2 = \cdots = \lambda_N = 1$, we see that the resulting approximation in (7.3) is of block-diagonal form and almost all diagonal blocks are equal. In fact, only two different diagonal blocks occur, M for $\lambda_1 = 0$ and $M + \sqrt{\tau}\varepsilon K$ for all remaining eigenvalues $\lambda_j = 1$. Typically, the extra effort is negligible. As the application of the Fourier transform will in general result in complex valued systems, we formulate the blocks in (7.3) to 2×2 real-valued block systems. In detail, we have to solve two types of systems

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \tilde{y}_r \\ \tilde{y}_c \end{bmatrix} = \begin{bmatrix} \tilde{g}_r \\ \tilde{g}_c \end{bmatrix}$$

and

$$\begin{bmatrix} M + \sqrt{\tau}\varepsilon K & 0 \\ 0 & M + \sqrt{\tau}\varepsilon K \end{bmatrix} \begin{bmatrix} \tilde{y}_r \\ \tilde{y}_c \end{bmatrix} = \begin{bmatrix} \tilde{g}_r \\ \tilde{g}_c \end{bmatrix}.$$

Again, the first of the above systems arises for the diagonal block with $\lambda_1 = 0$ and the second one for all the remaining eigenvalues λ_j. As in [58], we solve these real-valued systems with a fixed number of steps of an inexact Uzawa-type method

$$\tilde{y}^{(l+1)} = \tilde{y}^{(l)} + \omega\, \mathcal{P}_1^{-1} \tilde{r}^{(l)},$$

where $r^{(l)}$ denotes the residual and ω is the relaxation parameter. $\mathcal{P}_1$ is a block-diagonal preconditioner whose inverse is applied by inverting the diagonal, nonsingular block M or by using an AMG approximation of the block $M + \sqrt{\tau}\varepsilon K$. Again, independent of the number of phases, this is the one and the same AMG preconditioner every

[2] AMG methods typically exhibit geometric-like properties for positive definite elliptic type operators but use only algebraic information. This has the advantage that AMG can work well even for complicated geometries and meshes. We refer to [54, 29] for more information on AMG. We also want to emphasize that geometric multigrid (see, e.g., [62, 40]) approximations are also well suited to approximate $\hat{S}_1$ and $\hat{S}_2$ provided they can be readily applied.

time, and in this sense, the circulant approach also leads to a phase-independent preconditioner $\mathcal{P}$. Section 9.2 shows the efficiency of the proposed preconditioning strategy for both cases of L. In particular, we numerically illustrate the independence of $\mathcal{P}$ with respect to the parameters h and N.

Let us now turn to the more challenging case with a nonsmooth potential.

7.2. Schur complement preconditioner in the nonsmooth case. As already mentioned in section 7, the nondiagonal block $\mathcal{B}$ within the Schur complement $\mathcal{S} = I \otimes M + \tau(L \otimes K)(I \otimes M)^{-1}\mathcal{B}$ complicates the construction of a Schur complement approximation whose inverse can be applied in an efficient and easy way. Taking only the smooth potential into consideration, we have seen that we can approximate $\mathcal{B}$ with a block-diagonal matrix by using only the bound constraints from the Gibbs simplex (1.2). However, the inclusion of nonsmoothness involves additional severe penalizations into the system matrix. As can be seen from (6.4), penalized entries are scattered throughout the diagonals of every block of $\mathcal{B}$. The intensity of the penalization can be controlled by the penalty parameter ν. The smaller ν is, the stronger the penalization and the more accurate the numerical approximation of the nonsmoothness. In particular, regarding the nondiagonal blocks of $\mathcal{B}$, the estimated order of those penalized entries is $-\frac{h^2}{N}(\frac{1}{\nu} - 1)$, whereas it is $\frac{h^2}{N}$ for nonpenalized entries. The estimated difference of both types of entries is then of order $\frac{h^2}{N\nu}$, which indicates a severe dependency between h, N, and ν. This implies, e.g., that this difference decreases with decreasing mesh size. However, we have in mind that we want to go over to adaptive mesh strategies in the future. Therefore, the estimated order of penalized entries is usually of large size and highly differs to the order of the remaining nonpenalized entries. So they should not be neglected. That is why an approximation of the matrix $\mathcal{B}$ in block-diagonal form (as it was done in the smooth case) seems not to be of good quality, and our experiences also confirm this observation.

On the other hand, the position of penalized entries is changing with every Newton step. In detail, the places with the penalty parameter in the blocks B_i or B_{ii} depend on the phase u_i. Since all phases are separated in the domain (at least after a few time steps), one cannot expect the penalty parameter to act in the same regions for all phases. Therefore, a common distribution of penalized entries to all phases is not satisfying. That is why an approximation of the matrix $\mathcal{B}$ in form of equal blocks for all phases (as was also done in the smooth case) seems not to be of good quality, and our experiences also confirm this observation. All in all, we end up with keeping the whole block $\mathcal{B}$ within the Schur complement preconditioner. Regarding its construction, we proceed as in the previous section, i.e., we construct a preconditioner in matrix product form such that the exact Schur complement is captured as close as possible. The proposed Schur complement preconditioner is then

$$\begin{aligned}\hat{\mathcal{S}} &= \hat{\mathcal{S}}_1(I \otimes M)^{-1}\hat{\mathcal{S}}_2 \\ &= \left(\frac{N}{N-1}(I \otimes M) + \sqrt{\tau}(L \otimes K)\right)(I \otimes M)^{-1}\left(\frac{N-1}{N}(I \otimes M) + \sqrt{\tau}\mathcal{B}\right) \\ &= I \otimes M + \tau(L \otimes K)(I \otimes M)^{-1}\mathcal{B} + \frac{\sqrt{\tau}\,N}{N-1}\mathcal{B} + \sqrt{\tau}\,\frac{N-1}{N}(L \otimes K), \end{aligned} \tag{7.4}$$

which is similar to the corresponding approximation in the smooth case. The first two terms in (7.4) match the exact Schur complement. Due to the balanced distribution of τ in form of $\sqrt{\tau}$ in both factors, $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$, the influence of both remainder terms

in (7.4) is reduced. Let us discuss the action of the inverses of $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$. The former was already studied in the previous section. Therefore, let us concentrate on the latter now. The factor $\hat{\mathcal{S}}_2$ still contains the complicated, nondiagonal block $\mathcal{B}$, but its diagonal blocks are now shifted by mass matrices. These shifts were not just products of randomness. Without them, the diagonal blocks of $\hat{\mathcal{S}}_2$ would be indefinite. However, we want to apply the action of the inverse of the diagonal blocks of $\hat{\mathcal{S}}_2$ (see below). Solving indefinite systems typically causes problems; see, e.g., [26], which describes the difficulty of solving Helmholtz problems with classical iterative methods. However, shifting the diagonal blocks as proposed in $\hat{\mathcal{S}}_2$ makes them positive definite whenever $\tau < 1$, which is the case for our time discretization scheme. The proposed strategy concerning the solution of the system $\hat{\mathcal{S}}_2 y = g$ is the use of a block Jacobi method with a fixed number of steps:

$$y^{(l+1)} = y^{(l)} + \omega \mathcal{P}_2^{-1} r^{(l)}.$$

$\mathcal{P}_2$ is a block-diagonal preconditioner whose inverse is applied by using AMG approximations of the positive definite diagonal blocks of $\hat{\mathcal{S}}_2$. Unfortunately, this implies the need of N different AMG preconditioners, which have to be recomputed in every Newton step.

All in all, due to the structure of $\mathcal{B}$, preconditioning the nonsmooth system is more complicated than for the smooth one. Nevertheless, section 9.3 presents the performance of the presented preconditioner and shows promising results.

8. Solution algorithm. We now outline the method for the solution of the multicomponent problems with a nonsmooth potential. The basic steps needed to perform in each time step are summarized in Algorithm 1. Additionally, Algorithm 2 illustrates the application of the preconditioner $\mathcal{P}$.

Algorithm 1. Solution of vector-valued nonsmooth Cahn–Hilliard equations.

Input: $\mathbf{u}^{(0)} := \mathbf{u}(\cdot, 0)$, $\mathbf{w}^{(0)} := \mathbf{w}(\cdot, 0)$
Output: approximations $\mathbf{u}^{(t)}$, $\mathbf{w}^{(t)}$ of $\mathbf{u}(\cdot, t)$, $\mathbf{w}(\cdot, t)$
for $n = 1, \ldots, t$ **do**
  if $n \leq n_c$ **then**
    $s_1 = 1$
  else
    $s_1 = s_{\max}$
  end
  Update second right-hand side of the linear system
  for $s = s_1, \ldots, s_{\max}$ **do**
    $\nu = \nu_s$
    for $k = 0, 1, 2, \ldots$ *until convergence* **do**
      Update first right-hand side of the linear system
      Update G_i and the corresponding AMG solver A_i for $i = 1, \ldots, N$
      Calculate Newton residual $\|\mathbf{r}_\nu^{(k)}\|_2$
      Solve the linear system
    end
  end
end

ALGORITHM 2. APPLICATION OF THE PRECONDITIONER $\mathcal{P}$ IN THE NONSMOOTH CASE.

Input: $\mathbf{b}_1$, $\mathbf{b}_2$
Output: $\mathbf{x}_1$, $\mathbf{x}_2$
Solve $(I \otimes M)\mathbf{x}_1 = \mathbf{b}_1$
if $L = I$ **then**
  Solve $\left(\frac{N}{N-1}(I \otimes M) + \sqrt{\tau}(I \otimes K)\right)\mathbf{y}_2 = \tau(I \otimes K)\mathbf{x}_1 - \mathbf{b}_2$
if $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ **then**
  Apply FFT to the system $\hat{\mathcal{S}}_1\mathbf{y}_2 = \tau(L \otimes K)\mathbf{x}_1 - \mathbf{b}_2$
  Solve the resulting block diagonal system in real-valued form with an inexact Uzawa-type method and the use of AMG preconditioners
Solve $\hat{\mathcal{S}}_2\mathbf{x}_2 = \left(\frac{N-1}{N}(I \otimes M) + \sqrt{\tau}\mathcal{B}\right)\mathbf{x}_2 = (I \otimes M)\mathbf{y}_2$ with a block Jacobi method using AMG preconditioners A_i, $i = 1, \ldots, N$ for the diagonal blocks of $\hat{\mathcal{S}}_2$

9. Numerical results. In this section, we show results for the vector-valued Cahn–Hilliard problems. Concerning the regularized subproblems in the case of the nonsmooth potential, we choose the sequence $\nu_1 = 10^{-1} \geq \nu_2 = 10^{-2} \geq \cdots \geq \nu_{\max} = 10^{-7}$ of penalty parameters and solve each corresponding subproblem $\mathbf{F}_{\nu_i}(\mathbf{u}_{\nu_i,h}, \mathbf{w}_{\nu_i,h})$ by the SSN method. In doing so, each Newton method is initialized by the approximate solution of the previous one. If not mentioned otherwise, we fix $\nu = \nu_{\max}$ after the first time step, i.e., from then on it suffices to solve only one SSN method per time step. This is because the initial solution at the beginning might not be a good starting point for the SSN methods. For the (smooth and nonsmooth) Newton method, we use the stopping criterion in [43], given by

$$\|\mathbf{F}_\nu(\mathbf{u}_h^{(k)}, \mathbf{w}_h^{(k)})\|_2 \leq \epsilon_{\text{rel}} \|\mathbf{F}_\nu(\mathbf{u}_h^{(0)}, \mathbf{w}_h^{(0)})\|_2 + \epsilon_{\text{abs}}, \quad k = 1, \ldots, k_{\max},$$

where we set $k_{\max} = 20$, $\epsilon_{\text{rel}} = 10^{-12}$, and $\epsilon_{\text{abs}} = 10^{-6}$ in all examples. In each Newton step, we solve the linear system (6.2) by a Krylov subspace solver. The left preconditioners we have presented can be embedded into various of such iterative solvers. For our nonsymmetric system matrix $\mathcal{K}$, we propose the use of a nonsymmetric short-term recurrence method, namely, BiCG [30], but note that also other solvers such as QMR [31], BiCGStab [60], or GMRES [56] can be used with this preconditioner. We set the BiCG tolerance to be 10^{-7} for the preconditioned relative residual in all examples. The FFT-based preconditioner uses three steps of the inexact Uzawa method, and the block Jacobi preconditioner uses five steps. For the multilevel approximations, we choose Trilinos AMG approximations [41]. For one application of the preconditioner, we take in general 10 steps of a Chebyshev smoother and two V-cycles. The discretization is performed with deal.II [2], which allows the use of the Trilinos library. All numerical experiments listed here are generated with finite elements on rectangles.

Regarding the mesh size, experiments show that it is essential to highly resolve the interface. In particular, as the authors of [10, 6] pointed out, it is essential to ensure that at least eight grid points lie on the interface to avoid mesh effects. Using nonsmooth potentials, this leads to the condition $h_{\min} \leq \frac{\varepsilon\pi}{9}$; see, e.g., [6]. Using smooth potentials, the interfacial profile can be described by means of a tanh term. Following [47], the authors define the interface thickness to be the distance from 0.05 to 0.95, so that the equilibrium interface thickness is given by $\frac{4\sqrt{2}\varepsilon}{\tanh(0.9)}$. If we want

to have again at least eight grid points across the interface, we get the condition $h_{\min} \leq \frac{4\sqrt{2}\varepsilon}{8\tanh(0.9)}$.

Regarding the time step size, the existence and uniqueness of corresponding discrete solutions of the nonsmooth system has been shown in [9, Theorem 2.4] under the condition $\tau < \frac{4\varepsilon^2}{\lambda_A^2 \|L\|}$, where λ_A is the largest positive eigenvalue of A and $\|L\|$ denotes the spectral norm of L. For our choice of matrices A and L, this leads to the bound $\tau < 4\varepsilon^2$. Having the mesh size in mind, this implies a time step restriction of order $h_{\min}^2$. At first view, this seems not to be efficient in terms of long time intervals. But experiments show that the time step restriction is an essential characteristic of the nature of the problem [17, 12]. Using smooth potentials, we have shown stability and uniqueness conditions in section 3 if we pass over to truncated smooth potentials. These time step restrictions are again of order $h_{\min}^2$. In practice, we follow [46], where the authors have numerically determined the largest possible time step which allows stable numerical computations. These are given as $3.2 \cdot 10^{-2}$, $7.8 \cdot 10^{-3}$, $1.7 \cdot 10^{-3}$, and $4.4 \cdot 10^{-4}$ for uniform meshes with size $h = 2^{-i}$, $i = 5, \ldots, 8$, and $\varepsilon = 0.64\, h$. Some calculations show that they can approximately described by $73\varepsilon^2$, and this is the time step size we use in all experiments, if not mentioned otherwise.

The domain used is $[0,1]^2$. The initial state for almost all experiments below is given by 100 randomly distributed circles with different radii.

9.1. Evolution using smooth and nonsmooth potentials. This section demonstrates the distinctive properties of both types of potential. Figure 1 shows the evolution of five phases over 4000 time steps on an uniform mesh of size $h = 2^{-7}$. The parameters are taken as $\varepsilon = 0.0047$ and $\tau = 73\varepsilon^2$ in the smooth case and as $\varepsilon = \frac{9h}{\pi}$ and $\tau = 0.001$ in the nonsmooth case. Here, we fixed $\nu = \nu_{\max}$ from time step three on. Regarding the smooth setting, the CPU time for the whole simulation is $226062s$. In total, 4049 Newton steps were performed, which means that on average, only one Newton iteration is run per time step. The average number of BiCG iterations per time step is 22 with an average CPU time of $54.6s$. Regarding the nonsmooth setting, the CPU time for the whole simulation is $6222330s$. In total, 8178

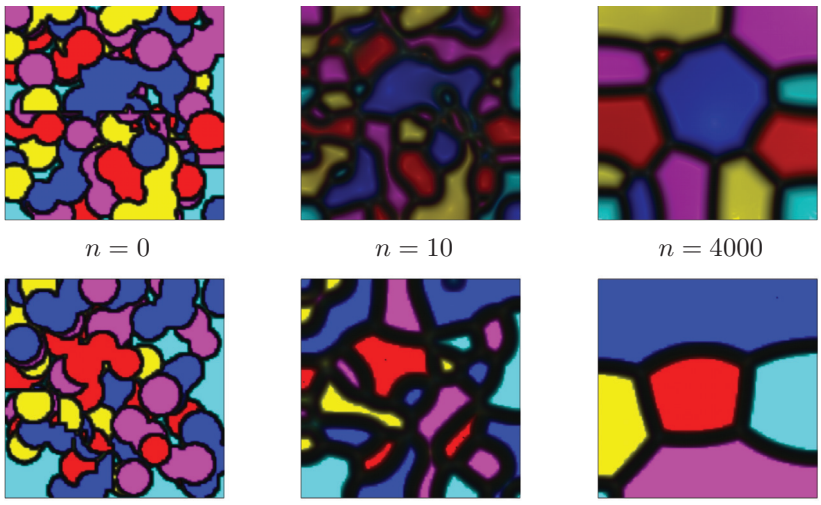
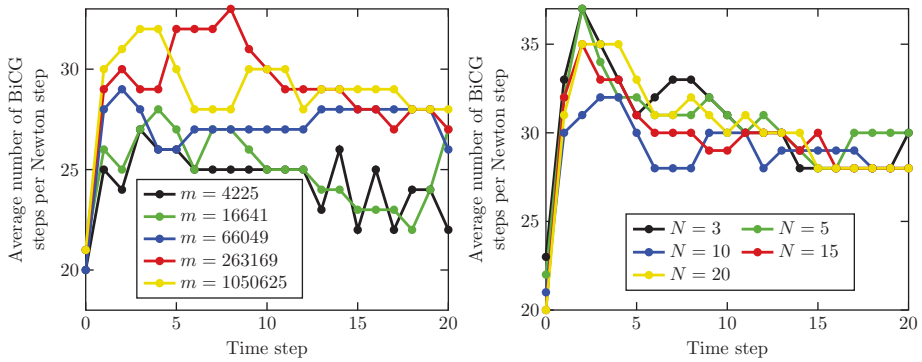


FIG. 1. *Evolution of five phases using the smooth (above) and nonsmooth (below) potential.*

TABLE 1
Minimum and maximum phase values in the smooth and nonsmooth model.

		Time step				
		10	100	1000	2000	4000
min	smooth	−0.0245436	−0.019918	−0.0159193	−0.0185168	−0.011366
	nonsmooth	$-1.3109 \cdot 10^{-7}$	$-1.20302 \cdot 10^{-7}$	$-1.06138 \cdot 10^{-7}$	$-1.05108 \cdot 10^{-7}$	$-1.07309 \cdot 10^{-7}$
max	smooth	1.00641	1.00928	1.01043	1.00639	1.01008
	nonsmooth	1	1	1	1	1



(a) $N = 10$. (b) $m = 1050625$.

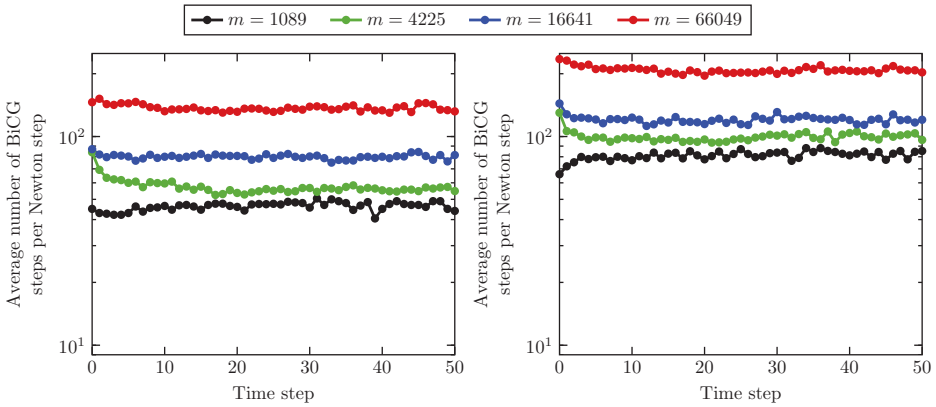
FIG. 2. *Results for the smooth model over* 20 *time steps with* $L = I$.

Newton steps were performed, which means that on average, only two SSN iterations are run per time step. The average number of BiCG iterations per time step is 65 with an average CPU time of 749.1s. Note that one should not compare the timing results, as the evolution with smooth and nonsmooth potentials is very different and distinct parameters are used. Table 1 illustrates the minimum and maximum phase value at some time steps. We observe that the concentrations may exceed one and become less than zero for smooth potentials. However, the overshoots and undershoots are not reported to blow up. In conclusion, both types of potentials are used in many applications. In some of them, like the deep-quench limit, the nonsmooth potential must be used. In other applications, smooth potentials are preferred and produce satisfactory results. Therefore, the development of efficient solvers is of great interest in both cases.

9.2. Iteration numbers with the smooth potential. In this section, we perform simulations for various uniform mesh sizes as well as for different number of phases and compare the number of BiCG iterations and the CPU times. Figure 2 shows the average number of BiCG iterations needed per Newton step over 20 time steps for the smooth model with $\varepsilon = h$ and $L = I$. In the legend of Figure 2(a), the number of grid points m is listed. The computations are done for $N = 10$ phases. The legend of Figure 2(b) shows the number of phases N. Here, the computations are done on uniform meshes of size $h = 2^{-10}$. In all calculations, the number of BiCG iterations does not exceed 37. Note also that the iteration numbers are almost the same when the circulant mobility matrix $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ is used; see Table 2. In Table 2, the resulting CPU times (in seconds) are given. From left to right, the columns indicate the choice of the mobility matrix L, the number of phases, the mesh size,

TABLE 2
Performance of our preconditioner for the smooth potential.

L	N	h	∅ Newton	$\overline{\text{Newton}}$	∅ BiCG	$\overline{\text{BiCG}}$	∅ CPU (s)	CPU (s)
I	10	2^{-6}	2	5	24	29	33.2	1574.8
I	10	2^{-7}	2	4	25	31	126.7	5843.1
I	10	2^{-8}	2	4	27	33	504.5	17843.5
I	10	2^{-9}	1	3	29	33	2650.9	70344.7
I	10	2^{-10}	1	3	29	33	9775.8	229585.0
I	3	2^{-10}	1	3	30	37	3152.3	74489.9
I	5	2^{-10}	1	3	30	37	5197.0	121936.0
I	10	2^{-10}	1	3	29	33	9775.8	229585.0
I	15	2^{-10}	1	3	30	35	16345.8	380718.0
I	20	2^{-10}	1	3	30	35	22830.8	532596.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$	10	2^{-5}	2	5	19	26	34.2	1467.2
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$	10	2^{-6}	2	5	22	28	119.6	5330.7
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$	10	2^{-7}	2	4	24	28	430.1	19707.8
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$	10	2^{-8}	2	4	25	30	1605.6	53587.7



(a) $N = 5$. (b) $N = 20$.

FIG. 3. *Results for the nonsmooth model over* 50 *time steps with* $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$.

the average and maximum number of Newton and BiCG iterations during the whole computation, and the average CPU time needed to solve the linear system during the whole computation as well as the CPU time for the whole computation. The results suggest that the convergence rate of average CPU time is linear with respect to the number of components. All in all, the results show the robustness of our preconditioner for both the mesh size and the number of phases.

9.3. Iteration numbers with the nonsmooth model. Similar computations are done with the nonsmooth model. Again, we consider various uniform mesh sizes and compare the number of BiCG iterations and the CPU times. Figure 3 shows the average number of BiCG iterations needed per SSN step over 50 time steps for the nonsmooth model with $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. The legend shows the number of grid points m. The computations are done for $N = 5$ phases in Figure 3(a) and for 20 phases in Figure 3(b). Unfortunately, the numerical mesh and phase independence, which we have obtained in the smooth setting, has been lost. Nevertheless, in consideration of the complexity of the nonsmooth problem, the gaps in the iteration numbers between

TABLE 3
Performance of our preconditioner for the nonsmooth potential.

L	N	h	∅ Newton	$\overline{\text{Newton}}$	∅ BiCG	$\overline{\text{BiCG}}$	∅ CPU (s)	CPU (s)
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	5	2^{-5}	3	7	46	54	74.7	14066.7
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	5	2^{-6}	4	8	59	99	307.3	67998.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	5	2^{-7}	4	6	79	102	1551.5	361878.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	5	2^{-8}	4	6	137	202	10831.3	2533990.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	20	2^{-5}	4	7	79	105	632.9	153017.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	20	2^{-6}	5	8	100	138	2360.0	636001.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	20	2^{-7}	4	6	118	160	8781.3	2092530.0
$I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$	20	2^{-8}	4	7	205	304	59800.7	13792200.0
I	20	2^{-5}	5	7	64	84	49.8	6639.0*
I	20	2^{-6}	5	7	95	128	1184.2	159206.0*
I	20	2^{-7}	4	6	102	129	3298.8	369418.0*
I	20	2^{-8}	5	7	149	212	15066.5	1838290.0*

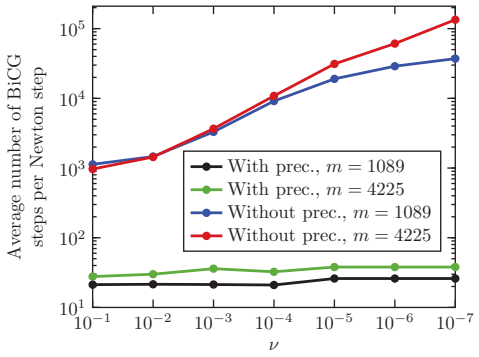
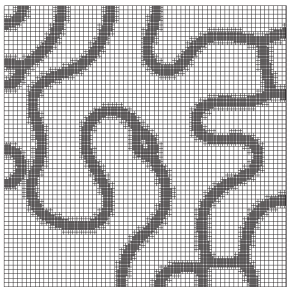


FIG. 4. *Performance of BiCG and preconditioned BiCG for the nonsmooth potential.*

two sequent mesh sizes are satisfying. Note that the whole system size is in fact $2mN$. The maximum (maximum average) BiCG iteration numbers are 255 (219) for five phases and 212 (188) for twenty phases when $L = I$ is used, as well as 202 (158) for five phases and 304 (235) for twenty phases when $L = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ is used. Further improvements are the subject of current research. Additionally, we have tested two more Krylov subspace solvers—GMRES and BiCGStab—but no improvements have been observed.

The iteration numbers for the choice $L = I$ are quite similar to the ones obtained here; see Table 3. In Table 3, the resulting CPU times (in seconds) are given. From left to right, the columns indicate the choice of mobility matrix L, the number of phases, the mesh size, the average and maximum number of SSN and BiCG iterations during the whole computation, and the average CPU time needed to solve the linear system during the whole computation as well as the CPU time for the whole computation. Note that the star sign in the column labeled CPU (s) indicates that the simulations for $L = I$ are performed over 20 time steps instead of 50.

Finally, we compare the average BiCG iteration numbers with and without preconditioning during the first time step in Figure 4. Here, the number of phases is three and $L = I$. As mentioned in the beginning of this section, we solve within this first time step seven SSN methods for the sequence of penalty parameters $\nu_1 = 10^{-1}, \ldots, \nu_7 = 10^{-7}$, respectively. As can be seen from this, although the precondi-



(a) Adaptive mesh.

Time	ν	Grid	m	SSN	BiCG	CPU (s)
0	10^{-1}	16641	166410	8	100	978.5
0	10^{-4}	16641	166410	3	132	1903.3
0	10^{-7}	16641	166410	2	141	2113.6
1	10^{-1}	30947	309470	5	95	1893.9
1	10^{-4}	30947	309470	3	116	2520.4
1	10^{-7}	30947	309470	4	170	4524.5
2	10^{-7}	104496	1044960	10	156	13567.8
4	10^{-7}	106810	1068100	8	130	12237.2
6	10^{-7}	105830	1058300	4	150	14147.8
8	10^{-7}	104838	1048380	3	149	14455.8
10	10^{-7}	104155	1041550	6	129	12066.1

(b) Performance of our preconditioner with $N = 5$ phases.

FIG. 5. *Results using adaptive meshes.*

TABLE 4
Performance of our preconditioner with $N = 3$ phases on an adaptive and uniform mesh.

Time	ν	Grid	m	SSN	SSN_{uni}	BiCG	BiCG_{uni}	CPU (s)	$\text{CPU}_{\text{uni}}(s)$
0	10^{-1}	4225	25350	7	7	99	96	32.2	1764.0
0	10^{-4}	4225	25350	2	2	94	98	39.3	1848.5
0	10^{-7}	4225	25350	1	1	102	101	42.2	2764.1
1	10^{-1}	8508	51048	5	4	89	85	64.7	1591.4
1	10^{-4}	8508	51048	4	2	88	100	70.5	1892.8
1	10^{-7}	8508	51048	4	1	87	108	76.2	2941.6
2	10^{-7}	24721	148326	7	+4	100	97	258.2	2717.3
4	10^{-7}	26771	160626	5	4	89	91	250.1	2536.8
6	10^{-7}	26367	158202	4	4	91	97	233.4	2691.7
8	10^{-7}	26007	156042	4	4	91	91	230.4	2569.1
10	10^{-7}	25520	153120	7	4	90	91	225.5	2567.3

tioned iteration numbers are considerably worse compared to the one in the smooth model, the preconditioned version always outperforms the unpreconditioned method. A factor of 1500 (3500) for $h = 2^{-5}$ ($h = 2^{-6}$) can be observed, and we would expect this to be even more significant if a larger number of phases or grid sizes is used. The average CPU time needed to solve the linear system ranges from $2.0s$ ($7.4s$) in the case ν_1 to $75.63s$ ($986.1s$) in the case ν_7 for $h = 2^{-5}$ ($h = 2^{-6}$) in the unpreconditioned method. Using the preconditioner, the corresponding CPU time ranges from $4.1s$ ($16.8s$) to $5.0s$ ($24.4s$).

9.4. Mesh adaption. As can be seen in section 9.3, the numerical mesh and phase independence, which we have obtained in the smooth setting, have been lost when we pass into the more complex nonsmooth formulation. Therefore, we go over to the development of an adaptively refined and coarsened mesh strategy in order to reduce the system size and accelerate our solver. As already mentioned above, it is essential to ensure that at least eight grid points lie on the interface to avoid mesh effects. We hence refine the interface up to the required level and coarse in areas where the concentrations are constant, as can be seen in Figure 5(a).

Table 4 compares the iteration numbers and CPU times for an example with $N = 3$ phases on an adaptive mesh with the results we get on an uniform mesh. The parameters are taken as $\varepsilon = 0.01$, $\tau = 0.0003$, and $L = I$. The minimum and maximum mesh sizes in the adaptive setting are set to be $h_{\min} = \frac{\varepsilon\pi}{9}$ and $h_{\max} = 10\, h_{\min}$.

The uniform setting uses $h = 2^{-8}$ as the mesh size.[3] From left to right, the columns of Table 4 indicate the time step, penalty parameter, and, for the adaptive setting, the number of grid points as well as the number of unknowns. The columns labeled SSN, BiCG, and costs show for the adaptive setting the number of SSN iterations, the average number of BiCG iterations per SSN step, and the average CPU time (in seconds) needed to solve the linear system with BiCG. The columns labeled BiCG_{uni} and $\text{costs}_{\text{uni}}$ show the same numbers when the uniform mesh (with 66049 grid points) is used instead. We observe that, instead of solving a linear system of 396294 unknowns (uniform setting) in each time step, we can use the adaptive mesh strategy to reduce this number to sizes of order 153120. This is a reduction of a factor 2.5. Considering that the phases will coarsen in the course of the ongoing process, the interfacial regions will shrink further until the steady state is reached. Therefore, the system size will be more reduced over time. Besides the reduction of unknowns, the average CPU time needed to solve the linear system has been reduced by a factor of 11. Besides these reductions, a slight increase in the number of SSN steps is observed when we pass to the adaptive setting. Note, the plus sign in the column labeled SSN_{uni} indicates that we have used the sequence of penalty parameters $\nu_1, \ldots, \nu_7$ until time step three in the uniform setting. In contrast, we have already fixed $\nu = \nu_{\max}$ from time step two on in the adaptive setting. That means that the number of SSN iterations marked with the plus sign might be slightly higher if we would have fixed $\nu = \nu_{\max}$ also here from time step two on. Nevertheless, the slight increase in SSN steps does not prevent the decrease of costs needed per time step. To show some more results using adaptive meshes, we have repeated the same experiment using $N = 5$ phases, $\varepsilon = 0.007$ and $\tau = 0.0001$; see Figure 5(b). An important observation is the further increase of SSN iteration numbers. These can be reduced, for example, if we use the sequence of penalty parameters for some more time steps at the beginning, since here we fixed again $\nu = \nu_{\max}$ from time step two on. Another reason might be a too tight choice of the time step size τ here. Remember that we need the uniqueness condition $\tau < 4\varepsilon^2 = 0.000196$. A slight decrease of our choice of τ should usually slightly reduce the SSN iterations.

9.5. Comparison of different constant mobilities. In this section, we visually compare various constant mobilities. We test the choices $L_1 = I$ as well as the two different circulant mobilities $L_2 = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ and

$$L_3 = \begin{bmatrix} 0.5 & -0.25 & 0 & -0.25 \\ -0.25 & 0.5 & -0.25 & 0 \\ 0 & -0.25 & 0.5 & -0.25 \\ -0.25 & 0 & -0.25 & 0.5 \end{bmatrix}.$$

The test example is taken from [47, section 4.6] and considers four phases. We use their initial state on the domain $\Omega = [0,1]^2$ with uniform mesh size $h = 2^{-7}$, as illustrated in Figure 6(a). The expression $u_i^{(0)}$, $u_j^{(0)}$ indicates that the initial phase $u_i^{(0)}$ is set to be randomly between 0.5 and 0.51 and the initial phase $u_j^{(0)}$ is set to be $1 - u_i^{(0)}$ in the considered rectangle. The results after 50 time steps are illustrated in Figure 6(b)–(d). There are no differences in the morphologies; only small time differences seem to occur. The same observations are made with a smooth potential. In [23] and references therein, the requirement of concentration dependent mobilities for many

[3] This is the required refinement in order to resolve the interface with about eight grid points in the uniform setting.

(a) Initial state. (b) L_1. (c) L_2. (d) L_3.

FIG. 6. *Simulation results for different mobilities after* 50 *time steps.*

applications is mentioned, for example, if the mobility in the interface is larger than in the pure phases. This motivates us to consider concentration dependent mobilities in the future in order to model other physical situations.

10. Other available solvers. In this section, we compare our solution technique to existing solution methods. A block preconditioning strategy for multicomponent Cahn–Hilliard problems using smooth potentials is proposed by Boyanova and Neytcheva [14]. They have extended their techniques, developed regarding to scalar systems [13, 1], to the multicomponent case. In general, these works study Cahn–Hilliard equations with and without convection. Similar to us, they have considered so far some special forms of the mobility matrix L, namely, diagonal ones. The numerical method consists of a time discretization scheme, where during each time step a nonlinear system is solved by a quasi-Newton method. Each Newton step involves the solution of a nonsymmetric linear system with the Jacobian matrix. The preconditioning technique is based on two steps: First, the system matrix is simplified by neglecting small entries,[4] so that it is replaced by a high-quality and symmetric matrix approximation. Second, they prove that the resulting approximation can be preconditioned by an optimal preconditioner. To be more precise, the proposed preconditioner (applied to the system matrix in (6.2) with $L = I$) is given by

$$\hat{\mathcal{A}}_0 = \begin{bmatrix} I \otimes M & -\varepsilon^2 (I \otimes K) \\ \tau (I \otimes K) & I \otimes M + 2\sqrt{\tau}\varepsilon (I \otimes K) \end{bmatrix}$$

and proved to be an optimal preconditioner for the simplified system matrix $\mathcal{K}_0$ in (7.1). In fact, the spectrum of $\hat{\mathcal{A}}_0^{-1}\mathcal{K}_0$ is contained in the interval $[0.5, 1]$. Solutions with $\hat{\mathcal{A}}_0$ require four steps: one solution with $I \otimes M$, two with $I \otimes M + \sqrt{\tau}\varepsilon(I \otimes K)$, and some matrix-vector operations. As shown in section 7.1, our smooth Schur complement based preconditioner $\mathcal{P}$ is also proved to be optimal in the case $L = I$ with the same eigenvalue bound $\lambda\left(\mathcal{P}^{-1}\mathcal{K}_0\right) \in [0.5, 1]$. In fact, solutions with $\mathcal{P}$ require three steps, namely, the same first three steps which are performed by the application of $\hat{\mathcal{A}}_0$. In total, the application of our proposed preconditioner requires one step less, which is the one performing some matrix-vector operations. But as these are cheap operations, they are negligible. Table 5 contains comparisons of the performance of

[4] Remember that we have applied the same technique in order to construct our proposed Schur complement approximation.

TABLE 5
Performance of our preconditioner compared to the one proposed in [14].

		Preconditioner in [14]				Our preconditioner			
N	h	∅ BiCG	$\overline{\text{BiCG}}$	∅ CPU (s)	CPU (s)	∅ BiCG	$\overline{\text{BiCG}}$	∅ CPU (s)	CPU (s)
3	2^{-5}	19	25	2.7	134.2	24	31	3.3	194.1
3	2^{-6}	22	27	8.9	427.6	24	29	9.3	479.4
3	2^{-7}	22	30	32.2	1588.4	26	35	37.1	1888.9
3	2^{-8}	22	28	133.2	4882.3	27	31	153.7	5574.3
3	2^{-9}	23	29	635.2	18303.4	29	36	773.0	21415.4
3	2^{-10}	27	31	2819.3	66465.0	31	37	3152.3	74489.9
10	2^{-5}	19	27	9.0	434.5	21	31	10.1	474.3
10	2^{-6}	21	27	29.4	1463.5	24	29	33.2	1574.8
10	2^{-7}	22	30	112.9	5151.4	25	31	126.7	5843.1
10	2^{-8}	23	30	456.4	17715.4	27	33	504.5	17843.5
10	2^{-9}	25	28	2356.5	62074.1	29	33	2650.9	70344.7
10	2^{-10}	25	29	8902.8	209855.0	29	33	9775.8	229585.0

the preconditioner proposed by us (concerning the smooth potential) with the one proposed in [14]. We have applied both strategies to the solution of the time-discrete systems (3.1)–(3.2), which are solved by the standard Newton method. The numerical solutions are computed on the uniform meshes of size $h = 2^{-i}$ for $i = 5, \dots, 10$. For each case, $\varepsilon = h$ and $\tau = 73\varepsilon^2$ is used, 20 time steps are performed and $N = 3$, and 10 phases are considered. The mobility matrix L is set to be the identity matrix. From left to right, the columns of Table 5 indicate the number of phases, the mesh size, the average and maximum number of BiCG iterations during the whole computation, and the average CPU time (in seconds) needed to solve the linear system during the whole computation as well as the CPU time for the whole computation. Table 5 confirms the high efficiency of both rigorous analyzed preconditioners, the one in [14] as well as the one proposed by us. Moreover, it illustrates a slightly better performance in terms of iteration numbers as well as CPU times with the strategy proposed in [14], when considering diagonal mobility matrices. The case of more general forms of the mobility matrix are in both cases a topic of further research and is of great interest. At least for circulant matrices, our preconditioner is shown to be numerically mesh and phase independent.

A nonlinear multigrid method is proposed by Lee and co-authors [46, 47], who consider smooth potentials. The first work uses the mobility matrix $L = I$, and a practically unconditionally gradient stable scheme is presented which is based on a nonlinear splitting method. This allows them to decouple the N-component Cahn–Hilliard system into $N - 1$ scalar Cahn–Hilliard equations. The efficiency of the approach is shown by means of the average CPU time whose convergence rate is linear with respect to the number of phases. The second work [47] uses a concentration dependent mobility matrix and Crank–Nicolson's method for the discretization in time. The authors develop a full approximation storage multigrid method with a pointwise Gauß–Seidel relaxation scheme as a smoother. The nonlinearity is treated using one Newton step. The second-order accuracy of the numerical scheme is demonstrated. They also visually compare phase separation of four phases with a constant and a degenerate concentration dependent mobility, and the differences in morphologies and evolution dynamics can be seen.

Gräser, Kornhuber, and Sack [38] propose globally convergent nonsmooth Schur–Newton methods (NSNMG) for the solution of discrete multicomponent Cahn–Hilliard systems. They consider logarithmic as well as obstacle potentials. NSNMG can be

formulated in primal-dual form and results in a preconditioned Uzawa method. Each step consists first of the update of the primal variable which includes the direct work with the inverse $(A + \partial\varphi)^{-1}$. Here, A is a symmetric positive definite matrix and $\partial\varphi$ is the subdifferential of the nonsmooth part that includes the indicator function $\sum_{i=1}^{N} \chi_{[0,\infty)}(u_i)$. The second step of NSNMG is to compute the dual variable, which can be done by solving a truncated linear saddle point problem and updating the step size for the Uzawa method. The authors solve the linear systems by a preconditioned GMRES method with restart after 50 steps. They numerically investigated local mesh independence of NSNMG as well as a robust convergence speed of NSNMG and the truncated nonsmooth Newton method for different numbers of phases.

In our previous works [12, 11] dealing with scalar smooth and nonsmooth Cahn–Hilliard systems, we have already tested the use of finite differences combined with an FFT-based solver against finite element methods. As the FFT basis functions are eigenvectors of the finite difference operators, which form the discrete Laplacian, FFT methods rapidly solve diffusion equations on simple domains. For the scalar Cahn–Hilliard equation using a double-well potential, Eyre [28] presents an FFT method for the fast inversion of the preconditioner. He uses a finite difference scheme and suggests unconditionally gradient stable methods. In order to solve the preconditioned system, Eyre proposes the use of a conjugate gradient squared method. The overall effort for solving the linear system is dominated by the FFTs and is $m \log(\sqrt{m})$. Another use of FFT methods appears in [35, 5, 16] for the smooth, scalar Cahn–Hilliard inpainting problem. They propose a two-dimensional FFT method and achieve fast inpainting. In fact, regarding the computational time, FFT methods are hard to beat, but with respect to more complex problems, spectral methods on complicated domains are difficult; see, e.g., [15]. Extending the idea of using an FFT-based solution scheme to the nonsmooth problem is a challenge. An efficient FFT-based implementation employing a nonsmooth potential would typically suffer from the nonconstant, nonsmooth term that originates in the discretization of the penalization term. In general, it holds for spectral methods: the smoother the function is, the faster the convergence. It is shown in [12, 11] how the iteration numbers increase with the nonsmoothness obtained by varying the penalty parameter ν. This is the motivation for this work to focus on a discretization via finite elements especially for the nonsmooth problem.

11. Conclusions. In this paper, we have analyzed the linear systems arising in smooth and nonsmooth vector-valued Cahn–Hilliard systems. For the latter, we have applied a semismooth Newton method combined with a Moreau–Yosida regularization technique for handling the pointwise constraints. In order to make the semismooth Newton method more efficient, we have used a Krylov subspace solver. We have introduced and studied block-triangular preconditioners using an efficient Schur complement approximation. This approximation can be done using multilevel techniques, such as AMG (as in our case), and the numerical results justify this choice. In particular, the proposed preconditioner is proved to be optimal for a high-quality approximation of the system matrix when the smooth potential together with the identity mobility matrix $L = I$ is used.

Appendix A. Proof of Theorem 3.1. Throughout the following, we will make use of the scalar product

$$(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} = \sum_{i=1}^{N} (u_i, v_i)$$

in $L^2(\Omega)^N$ with norm $\|\cdot\|$, of the $H^1(\Omega)$-inner product $(\cdot,\cdot)_1$ and of the scalar product

$$(\mathbf{u},\mathbf{v})_1 = (\mathbf{u},\mathbf{v}) + (\nabla\mathbf{u},\nabla\mathbf{v}) = \sum_{i=1}^{N}\left((u_i,v_i) + (\nabla u_i,\nabla v_i)\right) = \sum_{i=1}^{N}(u_i,v_i)_1$$

in $H^1(\Omega)^N$ with norm $\|\cdot\|_1$.

Proof. Assume that there exist two solutions $(\mathbf{u},\mathbf{w})$ and $(\tilde{\mathbf{u}},\tilde{\mathbf{w}})$ of (3.1)–(3.2). Then we get

$$(\mathbf{u}-\tilde{\mathbf{u}},\mathbf{v}) + \tau\,(L\nabla(\mathbf{w}-\tilde{\mathbf{w}}),\nabla\mathbf{v}) = 0 \tag{A.1}$$

$$-(\mathbf{w}-\tilde{\mathbf{w}},\mathbf{v}) + \varepsilon^2\,(\nabla(\mathbf{u}-\tilde{\mathbf{u}}),\nabla\mathbf{v}) + (\mathbf{f}(\mathbf{u})-\mathbf{f}(\tilde{\mathbf{u}}),\mathbf{v}) - \frac{1}{N}\left(\sum_{j=1}^{N}(f(u_j)-f(\tilde{u}_j))\,\mathbf{1},\mathbf{v}\right) = 0 \tag{A.2}$$

for all $\mathbf{v}\in H^1(\Omega)^N$. Choosing $\mathbf{v}=\mathbf{w}-\tilde{\mathbf{w}}$ in (A.1) gives

$$0 \geq (\mathbf{u}-\tilde{\mathbf{u}},\mathbf{w}-\tilde{\mathbf{w}}) + \tau\lambda_{\min}\|\nabla(\mathbf{w}-\tilde{\mathbf{w}})\|^2 \tag{A.3}$$

(see also [4]). Choosing $\mathbf{v}=\mathbf{u}-\tilde{\mathbf{u}}$ in (A.2) gives

$$0 = -(\mathbf{u}-\tilde{\mathbf{u}},\mathbf{w}-\tilde{\mathbf{w}}) + \varepsilon^2\|\nabla(\mathbf{u}-\tilde{\mathbf{u}})\|^2 + (f(\mathbf{u})-f(\tilde{\mathbf{u}}),\mathbf{u}-\tilde{\mathbf{u}}) - \frac{1}{N}\left(\sum_{j=1}^{N}(f(u_j)-f(\tilde{u}_j))\,\mathbf{1},\mathbf{u}-\tilde{\mathbf{u}}\right). \tag{A.4}$$

The last term in (A.4) is zero since we can reorder this term to

$$\begin{aligned}\left(\sum_{j=1}^{N}(f(u_j)-f(\tilde{u}_j))\,\mathbf{1},\mathbf{u}-\tilde{\mathbf{u}}\right) &= \sum_{i=1}^{N}\left(\sum_{j=1}^{N}(f(u_j)-f(\tilde{u}_j)),u_i-\tilde{u}_i\right)\\ &= \sum_{j=1}^{N}\left(f(u_j)-f(\tilde{u}_j),\sum_{i=1}^{N}(u_i-\tilde{u}_i)\right)\end{aligned}$$

and $\sum_{i=1}^{N}(u_i-\tilde{u}_i)=0$ for a.e. $\mathbf{x}\in\Omega$. The second last term in (A.4) can be reformulated using the Taylor expansion of the potential

$$\psi(\mathbf{u}) = \psi(\tilde{\mathbf{u}}+\mathbf{u}-\tilde{\mathbf{u}}) = \psi(\tilde{\mathbf{u}}) + \mathbf{f}(\tilde{\mathbf{u}})\cdot(\mathbf{u}-\tilde{\mathbf{u}}) + \frac{1}{2}\sum_{i=1}^{N}f'(s_i)(u_i-\tilde{u}_i)^2,$$

$$\psi(\tilde{\mathbf{u}}) = \psi(\mathbf{u}+\tilde{\mathbf{u}}-\mathbf{u}) = \psi(\mathbf{u}) + \mathbf{f}(\mathbf{u})\cdot(\tilde{\mathbf{u}}-\mathbf{u}) + \frac{1}{2}\sum_{i=1}^{N}f'(\tilde{s}_i)(u_i-\tilde{u}_i)^2,$$

where $f'(s_i) = \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{s})$ and $\mathbf{s}, \tilde{\mathbf{s}}$ lie between $\mathbf{u}$ and $\tilde{\mathbf{u}}$. Subtracting these two equations gives

$$(f(\mathbf{u}) - f(\tilde{\mathbf{u}})) \cdot (\mathbf{u} - \tilde{\mathbf{u}}) = \frac{1}{2} \sum_{i=1}^{N} \left(f'(s_i) + f'(\tilde{s}_i)\right) (u_i - \tilde{u}_i)^2 \overset{(3.3)}{\geq} -T \sum_{i=1}^{N} (u_i - \tilde{u}_i)^2.$$

Therefore, we obtain in (A.4)

$$0 \geq -\left(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{w} - \tilde{\mathbf{w}}\right) + \varepsilon^2 \left\|\nabla(\mathbf{u} - \tilde{\mathbf{u}})\right\|^2 - T \left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2. \tag{A.5}$$

The last equation we need is the one we obtain with the choice $\mathbf{v} = \mathbf{u} - \tilde{\mathbf{u}}$ in (A.1)

$$\begin{aligned} 0 &= \left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2 + \tau \left(L\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \nabla(\mathbf{u} - \tilde{\mathbf{u}})\right) \qquad \text{(A.6)} \\ &= T \left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2 + \left(\frac{\tau T}{\sqrt{2}\varepsilon} L\nabla(\mathbf{w} - \tilde{\mathbf{w}}), \sqrt{2}\varepsilon \nabla(\mathbf{u} - \tilde{\mathbf{u}})\right) \\ &\geq T \left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2 - \frac{\tau^2 T^2}{4\varepsilon^2} \left\|L\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2 - \varepsilon^2 \left\|\nabla(\mathbf{u} - \tilde{\mathbf{u}})\right\|^2 \qquad \text{(A.7)} \\ &\geq T \left\|\mathbf{u} - \tilde{\mathbf{u}}\right\|^2 - \frac{\tau^2 T^2 \|L\|^2}{4\varepsilon^2} \left\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2 - \varepsilon^2 \left\|\nabla(\mathbf{u} - \tilde{\mathbf{u}})\right\|^2, \qquad \text{(A.8)} \end{aligned}$$

where we have used Young's inequality in (A.7). Now, adding (A.3), (A.5), and (A.8), we get

$$\begin{aligned} 0 &\geq \tau\lambda_{\min} \left\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2 - \frac{\tau^2 T^2 \|L\|^2}{4\varepsilon^2} \left\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2 \\ &= \tau \left(\lambda_{\min} - \frac{\tau T^2 \|L\|^2}{4\varepsilon^2}\right) \left\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\right\|^2. \end{aligned}$$

Hence, we obtain uniqueness if $\lambda_{\min} - \frac{\tau T^2 \|L\|^2}{4\varepsilon^2} > 0 \Leftrightarrow \tau < \frac{4\varepsilon^2 \lambda_{\min}}{T^2 \|L\|^2}$. Since then, it follows that $\|\nabla(\mathbf{w} - \tilde{\mathbf{w}})\| = 0$, which implies that $(\mathbf{w} - \tilde{\mathbf{w}}) = \text{const}$. Using this, (A.1) yields $(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) = 0$ for all $\mathbf{v} \in H^1(\Omega)^N$ and therefore $\mathbf{u} = \tilde{\mathbf{u}}$ a.e. Finally, (A.2) then gives $\mathbf{w} = \tilde{\mathbf{w}}$ a.e. ∎

Appendix B. Proof of Theorem 3.2.

Proof. Choosing $\mathbf{v} = \mathbf{w}$ in (3.1) gives

$$0 \geq \left(\mathbf{u} - \mathbf{u}^{(n-1)}, \mathbf{w}\right) + \tau\lambda_{\min} \left\|\nabla \mathbf{w}\right\|^2. \tag{B.1}$$

Choosing $\mathbf{v} = \mathbf{u} - \mathbf{u}^{(n-1)}$ in (3.2) gives

$$\begin{aligned} 0 = &-\left(\mathbf{u} - \mathbf{u}^{(n-1)}, \mathbf{w}\right) + \frac{\varepsilon^2}{2} \left(\left\|\nabla \mathbf{u}\right\|^2 - \left\|\nabla \mathbf{u}^{(n-1)}\right\|^2 + \left\|\nabla(\mathbf{u} - \mathbf{u}^{(n-1)})\right\|^2\right) \qquad \text{(B.2)} \\ &+ \left(f(\mathbf{u}), \mathbf{u} - \mathbf{u}^{(n-1)}\right) - \frac{1}{N} \left(\sum_{j=1}^{N} f(u_j)\, \mathbf{1}, \mathbf{u} - \mathbf{u}^{(n-1)}\right). \end{aligned}$$

As in the proof before, we can show that the last term in (B.2) is zero and the second to last term in (B.2) can be reformulated using the Taylor expansion of the potential

$$\begin{aligned}\mathbf{f}(\mathbf{u})\cdot\left(\mathbf{u}-\mathbf{u}^{(n-1)}\right) &= \psi(\mathbf{u})-\psi(\mathbf{u}^{(n-1)})+\frac{1}{2}\sum_{i=1}^{N}f'(s_i)(u_i-u_i^{(n-1)})^2\\ &\overset{(3.3)}{\geq} \psi(\mathbf{u})-\psi(\mathbf{u}^{(n-1)})-\frac{T}{2}\sum_{i=1}^{N}(u_i-u_i^{(n-1)})^2,\end{aligned}$$

where $\mathbf{s}$ lies between $\mathbf{u}$ and $\mathbf{u}^{(n-1)}$. Therefore, we obtain in (B.2)

$$\begin{aligned}0\geq -\left(\mathbf{u}-\mathbf{u}^{(n-1)},\mathbf{w}\right)+\frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2-\left\|\nabla\mathbf{u}^{(n-1)}\right\|^2+\left\|\nabla(\mathbf{u}-\mathbf{u}^{(n-1)})\right\|^2\right)\\ +(\psi(\mathbf{u}),1)-\left(\psi(\mathbf{u}^{(n-1)}),1\right)-\frac{T}{2}\left\|\mathbf{u}-\mathbf{u}^{(n-1)}\right\|^2.\end{aligned}\tag{B.3}$$

The last equation we need is the one we obtain with the choice $\mathbf{v}=\mathbf{u}-\mathbf{u}^{(n-1)}$ in (3.1),

$$\begin{aligned}0&=\|\mathbf{u}-\tilde{\mathbf{u}}\|^2+\tau\left(L\nabla\mathbf{w},\nabla(\mathbf{u}-\mathbf{u}^{(n-1)})\right)\\ &=\frac{T}{2}\left\|\mathbf{u}-\mathbf{u}^{(n-1)}\right\|^2+\left(\frac{\tau T}{2\varepsilon}L\nabla\mathbf{w},\varepsilon\nabla(\mathbf{u}-\mathbf{u}^{(n-1)})\right)\\ &\geq\frac{T}{2}\left\|\mathbf{u}-\mathbf{u}^{(n-1)}\right\|^2-\frac{\tau^2T^2}{8\varepsilon^2}\|L\nabla\mathbf{w}\|^2-\frac{\varepsilon^2}{2}\left\|\nabla(\mathbf{u}-\mathbf{u}^{(n-1)})\right\|^2 \qquad \text{(B.4)}\\ &\geq\frac{T}{2}\left\|\mathbf{u}-\mathbf{u}^{(n-1)}\right\|^2-\frac{\tau^2T^2\|L\|^2}{8\varepsilon^2}\|\nabla\mathbf{w}\|^2-\frac{\varepsilon^2}{2}\left\|\nabla(\mathbf{u}-\mathbf{u}^{(n-1)})\right\|^2, \qquad \text{(B.5)}\end{aligned}$$

where we have used Young's inequality in (B.4). Now, adding (B.1), (B.3), and (B.5), we get

$$\begin{aligned}0\geq\tau\left(\lambda_{\min}-\frac{\tau T^2\|L\|^2}{8\varepsilon^2}\right)\|\nabla\mathbf{w}\|^2+\frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2-\left\|\nabla\mathbf{u}^{(n-1)}\right\|^2\right)\\ +(\psi(\mathbf{u}),1)-\left(\psi(\mathbf{u}^{(n-1)}),1\right).\end{aligned}$$

Now, we can bound the energy

$$\begin{aligned}\mathcal{E}(\mathbf{u})-\mathcal{E}(\mathbf{u}^{(n-1)})&=\frac{\varepsilon^2}{2}\left(\|\nabla\mathbf{u}\|^2-\left\|\nabla\mathbf{u}^{(n-1)}\right\|^2\right)+(\psi(\mathbf{u}),1)-\left(\psi(\mathbf{u}^{(n-1)}),1\right)\\ &\leq\tau\left(\frac{\tau T^2\|L\|^2}{8\varepsilon^2}-\lambda_{\min}\right)\|\nabla\mathbf{w}\|^2.\end{aligned}$$

Hence, we obtain energy stability if $\frac{\tau T^2\|L\|^2}{8\varepsilon^2}-\lambda_{\min}\leq 0\Leftrightarrow\tau\leq\frac{8\varepsilon^2\lambda_{\min}}{T^2\|L\|^2}$. ∎

Acknowledgments. The authors would like to thank Luise Blank and Harald Garcke as well as the anonymous referees for their helpful comments and suggestions.

REFERENCES

[1] 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 (2013), pp. 301–314.

[2] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.II—A general purpose object oriented finite element library*, ACM Trans. Math. Software, 33 (2007),

[3] J. W. Barrett and J. F. Blowey, *An error bound for the finite element approximation of a model for phase separation of a multi-component alloy*, IMA J. Numer. Anal., 16 (1996), pp. 257–287.

[4] J. W. Barrett and J. F. Blowey, *Finite element approximation of a model for phase separation of a multi-component alloy with non-smooth free energy*, Numer. Math., 77 (1997), pp. 1–34.

[5] A. L. Bertozzi, S. Esedoḡlu, and A. Gillette, *Inpainting of binary images using the Cahn–Hilliard equation*, IEEE Trans. Image Process., 16 (2007), pp. 285–291.

[6] L. Blank, M. Butz, and H. Garcke, *Solving the Cahn–Hilliard variational inequality with a semi-smooth Newton method*, ESAIM Control Optim. Calc. Var., 17 (2011), pp. 931–954.

[7] L. Blank, H. Garcke, L. Sarbu, and V. Styles, *Nonlocal Allen–Cahn systems: Analysis and a primal-dual active set method*, IMA J. Numer. Anal., 33 (2013), pp. 1126–1155.

[8] L. Blank, H. Garcke, L. Sarbu, and V. Styles, *Primal-dual active set methods for Allen–Cahn variational inequalities with nonlocal constraints*, Numer. Methods Partial Differential Equations, 29 (2013), pp. 999–1030.

[9] J. F. Blowey, M. I. M. Copetti, and C. M. Elliott, *Numerical analysis of a model for phase separation of a multicomponent alloy*, IMA J. Numer. Anal., 16 (1996), pp. 111–139.

[10] J. F. Blowey and C. M. Elliott, *Curvature dependent phase boundary motion and parabolic double obstacle problems*, in Degenerate Diffusions, W.-M. Ni, L. A. Peletier, and J. L. Vazquez, eds., IMA Vol. Math. Appl. 47, Springer, Berlin, 1993, pp. 19–60.

[11] J. Bosch, D. Kay, M. Stoll, and A. J. Wathen, *Fast solvers for Cahn–Hilliard inpainting*, SIAM J. Imaging Sci., 7 (2014), pp. 67–97.

[12] J. Bosch, M. Stoll, and P. Benner. *Fast solution of Cahn–Hilliard variational inequalities using implicit time discretization and finite elements*, J. Comput. Phys., 262 (2014), pp. 38–57.

[13] P. Boyanova, M. Do-Quang, and M. Neytcheva, *Efficient preconditioners for large scale binary Cahn–Hilliard models*, Comput. Methods Appl. Math., 12 (2012), pp. 1–22.

[14] P. Boyanova and N. Neytcheva, *Efficient numerical solution of discrete multi-component Cahn–Hilliard systems*, Comput. Math. Appl., 67 (2014), pp. 106–121.

[15] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover, Mineola, NY, 2001.

[16] M. Burger, L. He, and C.-B. Schönlieb, *Cahn–Hilliard inpainting and a generalization for grayvalue images*, SIAM J. Imaging Sci., 2 (2009), pp. 1129–1167.

[17] M. Butz, *Computational Methods for Cahn–Hilliard Variational Inequalities*, Ph.D. thesis, University of Regensburg, Regensburg, Germany, 2012.

[18] J. W. Cahn and J. E. Hilliard, *Free energy of a nonuniform system.* I. *Interfacial free energy*, J. Chem. Phys., 28 (1958), pp. 258–267.

[19] M. Chen, *On the solution of circulant linear systems*, SIAM J. Numer. Anal., 24 (1987), pp. 668–683.

[20] T. A. Davis, *UMFPACK Version* 5.7.0 *User Guide*, Technical report TR-04-003 (revised), Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 2013.

[21] I. C. Dolcetta, S. F. Vita, and R. March, *Area-preserving curve-shortening flows: From phase separation to image processing*, Interfaces Free Bound., 4 (2002), pp. 325–343.

[22] C. M. Elliott, *The Cahn–Hilliard model for the kinetics of phase separation*, in Mathematical Models for Phase Change Problems, J. F. Rodrigues, ed., Internat. Ser. Numer. Math. 88, Birkhäuser, Basel, 1989, pp. 35–73.

[23] C. M. Elliott and H. Garcke, *Diffusional phase transitions in multicomponent systems with a concentration dependent mobility matrix*, Phys. D, 109 (1997), pp. 242–256.

[24] C. M. Elliott and S. Luckhaus, *A Generalised Diffusion Equation for Phase Separation of a Multi-component Mixture with Interfacial Free Energy*, preprint 195, University of Bonn, Bonn, Germany, 1991.

[25] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numer. Math. Sci. Comput. 21, Oxford University Press, Oxford, 2005.

[26] O. G. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl, eds., Lect. Notes Comput. Sci. Eng. 83, Springer, Berlin, 2012, pp. 325–363.
[27] D. J. Eyre, *Systems of Cahn–Hilliard equations*, SIAM J. Appl. Math., 53 (1993), pp. 1686–1712.
[28] D. J. Eyre, *An Unconditionally Stable One-Step Scheme for Gradient Systems*, Technical report, Department of Mathematics, University of Utah, Salt Lake City, UT, 1998.
[29] R. D. Falgout, *An introduction to algebraic multigrid computing*, Comput. Sci. Eng., 8 (2006), pp. 24–33.
[30] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, G. A. Watson, ed., Lecture Notes in Math. 506, Springer, Berlin, 1976, pp. 73–89.
[31] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.
[32] H. Garcke, *Mechanical effects in the Cahn–Hilliard model: A review on mathematical results*, in Mathematical Methods and Models in Phase Transitions, A. Miranville, ed., Nova Science Publishers, New York, 2005, pp. 43–77.
[33] H. Garcke, B. Nestler, and B. Stoth, *On anisotropic order parameter models for multi-phase systems and their sharp interface limits*, Phys. D, 115 (1998), pp. 87–108.
[34] H. Garcke, B. Nestler, and B. Stoth, *A multiphase field concept: Numerical simulations of moving phase boundaries and multiple junctions*, SIAM J. Appl. Math., 60 (1999), pp. 295–315.
[35] A. Gillette, *Image Inpainting Using a Modified Cahn–Hilliard Equation*, Ph.D. thesis, University of California, Los Angeles, 2006.
[36] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods.* I, Numer. Math., 3 (1961), pp. 147–156.
[37] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods.* II, Numer. Math., 3 (1961), pp. 157–168.
[38] C. Gräser, R. Kornhuber, and U. Sack, *Nonsmooth Schur–Newton methods for multicomponent Cahn–Hilliard systems*, IMA J. Numer. Anal., 35 (2015), pp. 652–679.
[39] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
[40] W. Hackbusch, *Multi-grid methods and applications*, Springer Ser. Comput. Math. 4, Springer, Berlin, 1985.
[41] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An Overview of Frilinos*, Technical report SAND2003-2927, Sandia National Laboratories, Albuquerque, NM, 2003.
[42] J. E. Hilliard and J. W. Cahn, *An evaluation of procedures in quantitative metallography for volume-fraction analysis*, Trans. Am. Inst. Min. Metall. Eng., 221 (1961), pp. 344–352.
[43] M. Hintermüller, M. Hinze, and M. H. Tber, *An adaptive finite-element Moreau–Yosida-based solver for a non-smooth Cahn–Hilliard problem*, Optim. Methods Softw., 26 (2011), pp. 777–811.
[44] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.
[45] D. A. Kay and A. Tomasi, *Color image segmentation by the vector-valued Allen–Cahn phase-field model: A multigrid solution*, IEEE Trans. Image Process., 18 (2009), pp. 2330–2339.
[46] H. G. Lee, J.-W. Choi, and J. Kim, *A practically unconditionally gradient stable scheme for the N-component Cahn–Hilliard system*, Phys. A, 391 (2012), pp. 1009–1019.
[47] H. G. Lee and J. Kim, *A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system*, Phys. A, 387 (2008), pp. 4787–4799.
[48] J. E. Morral and J. W. Cahn, *Spinodal decomposition in ternary systems*, Acta Metallurgica, 19 (1971), pp. 1037–1045.
[49] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
[50] A. Novick-Cohen, *The Cahn–Hilliard equation: Mathematical and modeling perspectives*, Adv. Math. Sci. Appl., 8 (1998), pp. 965–985.

[51] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.
[52] R. L. Pego, *Front migration in the nonlinear Cahn–Hilliard equation*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 422 (1989), pp. 261–278.
[53] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 17 (2010), pp. 977–996.
[54] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
[55] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[56] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[57] J. Shen and X. Yang, *Numerical approximations of Allen–Cahn and Cahn–Hilliard equations*, Discrete Contin. Dyn. Syst., 28 (2010), pp. 1669–1691.
[58] M. Stoll, *One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem*, IMA J. Numer. Anal., (2013).
[59] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, 2nd ed., Wellesley Cambridge Press, Wellesley, MA, 2008.
[60] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.
[61] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2009), pp. 125–135.
[62] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, Chichester, UK, 1992.
[63] X.-F. Wu and Y. A. Dzenis, *Phase-field modeling of the formation of lamellar nanostructures in diblock copolymer thin films under inplanar electric fields*, Phys. Rev. E (3), 77 (2008), 031807.