A PRECONDITIONER FOR THE OHTA–KAWASAKI EQUATION

PATRICK E. FARRELL* AND JOHN W. PEARSON†

Abstract. We propose a new preconditioner for the Ohta–Kawasaki equation, a nonlocal Cahn–Hilliard equation that describes the evolution of diblock copolymer melts. We devise a computable approximation to the inverse of the Schur complement of the coupled second-order formulation via a matching strategy. The preconditioner achieves mesh independence: as the mesh is refined, the number of Krylov iterations required for its solution remains approximately constant. In addition, the preconditioner is robust with respect to the interfacial thickness parameter if a timestep criterion is satisfied. This enables the highly resolved finite element simulation of three-dimensional diblock copolymer melts with over one billion degrees of freedom.

Key words. Ohta–Kawasaki equation, preconditioner, Schur complement, nonlocal Cahn–Hilliard equation

AMS subject classifications. 65F08, 65M60, 35Q99, 82D60

1. Introduction. The Ohta–Kawasaki equation [14] models the evolution of diblock copolymer melts. A diblock copolymer is a polymer consisting of two subchains of different monomers that repel each other but are joined by a covalent bond. A large collection of these molecules is termed a melt. These melts are of scientific and engineering interest because they undergo phase separation of their different constituent monomers, allowing for the design of nanostructures with particular desirable properties. The numerical simulation of this equation is an essential tool in exploring the associated phase diagram [4].

The Ohta–Kawasaki functional describes the free energy of a diblock copolymer melt:

$$E[u] = \frac{1}{2} \int_\Omega \left(\varepsilon^2 \left|\nabla u\right|^2 + \frac{1}{2}\left(1 - u^2\right)^2 + \sigma \left|(-\Delta_N)^{-1/2}(u - m)\right|^2 \right) \, \mathrm{d}x, \qquad (1.1)$$

where $u = \pm 1$ denotes the two pure phases, Ω is the domain under study ($(0,1)^2$ or $(0,1)^3$), $\varepsilon \ll 1$ is the interfacial thickness between regions of the pure phases, σ is the nonlocal energy coefficient, m is the (conserved) average value of u in Ω, and Δ_N denotes the Laplacian with homogeneous Neumann boundary conditions; a mean-zero constraint is applied to eliminate the nullspace. Assuming that the dynamics are governed by a $H^{-1}(\Omega)$ gradient flow,

$$(u_t, \phi)_{H^{-1}(\Omega)} + E'[u; \phi] = 0 \quad \forall \phi \in \{v \in H^1(\Omega) : \int_\Omega v \, \mathrm{d}\Omega = 0\}, \qquad (1.2)$$

the resulting Ohta–Kawasaki dynamic equation on $\Omega \times (0, T]$ in strong second-order form is given by [16]

$$u_t - \Delta w + \sigma(u - m) = 0, \tag{1.3a}$$

$$w + \varepsilon^2 \Delta u - u(u^2 - 1) = 0, \tag{1.3b}$$

with homogeneous Neumann boundary conditions

$$\nabla u \cdot n = 0 \text{ and } \nabla w \cdot n = 0, \tag{1.4}$$

and with initial condition $u(x, 0) = u_0(x)$.

2. Approximating the Schur complement. After applying a finite element discretization in space and the θ-method in time, a discrete nonlinear problem must be solved at each timestep. Each Newton iteration involves solving a linear system of the form

$$J \begin{bmatrix} \delta u \\ \delta w \end{bmatrix} = \begin{bmatrix} (1 + \Delta t \theta \sigma) M & \Delta t \theta K \\ -\varepsilon^2 K - M_E & M \end{bmatrix} \begin{bmatrix} \delta u \\ \delta w \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}. \tag{2.1}$$

where J is the Jacobian, M is the standard mass matrix with entries of the form $\int_\Omega \phi_i \phi_j \, d\Omega$, K is the standard discretization of the Neumann Laplacian with entries $\int_\Omega \nabla \phi_i \cdot \nabla \phi_j \, d\Omega$, M_E is a mass matrix involving a spatially varying coefficient with entries $\int_\Omega (3u^2 - 1) \phi_i \phi_j \, d\Omega$, δu is the update for u, δw is the update for w, f_1 and f_2 gather the source term and contributions from previous time levels, and Δt is the timestep. As the discretization is refined and the dimension of (2.1) increases, it becomes impractical to employ direct solvers and preconditioned Krylov methods must be used instead. As the matrix in (2.1) is nonsymmetric, a suitable iterative solver such as GMRES [18] is required to compute its solution.

We note however that there are structures within the system that can be exploited within a solver. For example, M is symmetric positive definite, K is symmetric positive semidefinite (with one zero eigenvalue corresponding to the nullspace of constants), and M_E is symmetric.

Preconditioners for block-structured matrices typically involve approximating the Schur complement of the system. Let the Jacobian J be partitioned as

$$J = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{2.2}$$

Consider the preconditioner

$$P^{-1} = \begin{bmatrix} A & 0 \\ C & S \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix}, \tag{2.3}$$

where $S = D - CA^{-1}B$ is the Schur complement with respect to A. If exact inner solves are used for A^{-1} and S^{-1}, then the preconditioned operator $P^{-1}J$ has minimal polynomial degree two and GMRES converges exactly in two iterations [9, 13]. Here, A is a scaled mass matrix and is thus straightforward to solve with standard techniques such as Chebyshev semi-iteration [19]. Thus, the efficient preconditioning of S is the key to the fast solution of the Newton step (2.1), and thus of the dynamic Ohta–Kawasaki equations (1.3a)–(1.3b).

The Schur complement of (2.1) is

$$S = M + \varepsilon^2 c K M^{-1} K + c M_E M^{-1} K, \tag{2.4}$$

with $c = (\Delta t\theta)/(1 + \Delta t\theta\sigma) > 0$. In general it is very difficult to precondition the sum of different matrices. The approach adopted here is the *matching strategy* of Pearson and Wathen [17, 3]: the sum is approximated by the product of matrices, carefully chosen to match as many terms of the sum as possible. We propose the approximation

$$S \approx \tilde{S} = \hat{S}M^{-1}\hat{S}, \tag{2.5}$$

with

$$\hat{S} = M + \varepsilon\sqrt{c}K. \tag{2.6}$$

This approximation $\tilde{S}$ is the product of three invertible matrices, and so its inverse action can be efficiently computed by

$$\tilde{S}^{-1} = \hat{S}^{-1}M\hat{S}^{-1}. \tag{2.7}$$

The action of $\hat{S}^{-1}$ can be efficiently approximated with algebraic multigrid techniques [8] to yield a computationally cheap preconditioner.

Expanding the approximation (2.5), we find that it matches the first two terms of the Schur complement exactly:

$$\tilde{S} = M + \varepsilon^2 cKM^{-1}K + 2\varepsilon\sqrt{c}K. \tag{2.8}$$

Recall that M_E depends on the current estimate of the solution u. With this matching strategy, the term involving M_E in the Schur complement (2.4) has been neglected, so that $\tilde{S}$ does not vary between Newton iterations and no reassembly or algebraic multigrid reconstruction is required. The Schur complement approximation $\tilde{S}$ is straightforward and feasible to apply, and its effectiveness within a preconditioner will depend to a large extent on the effect of the neglected third term from S. In the next section we present some analysis to explain why we expect our approximation to work well as a preconditioner.

We note in passing that it is possible to rearrange (2.1) to the symmetric form

$$\begin{bmatrix} \frac{\Delta t\theta}{1+\Delta t\theta\sigma}K & M \\ M & -\varepsilon^2 K - M_E \end{bmatrix} \begin{bmatrix} \delta w \\ \delta u \end{bmatrix} = \begin{bmatrix} \frac{1}{1+\Delta t\theta\sigma}f_1 \\ f_2 \end{bmatrix}. \tag{2.9}$$

It is likely that a similar approach could be used to precondition this rearranged system, providing one takes into account the singular $(1,1)$-block. For instance, provided $\varepsilon^2 K + M_E$ is positive semidefinite, one could apply a recently proposed augmented preconditioner [12]. Furthermore, as discussed in the next section, it is more straightforward to prove the rates of convergence of iterative methods for symmetric systems. However, whereas the $(1,1)$-block A of (2.1) may be well approximated using Chebyshev semi-iteration, the stiffness matrix K arising in the $(1,1)$-block of the rearranged system would require a more expensive method such as a multigrid process. We therefore prefer to solve the nonsymmetric system (2.1) due to the ease with which we may compute the approximate action of A^{-1}.

3. Analysis. We now wish to justify why our preconditioner is likely to be effective for the problem being solved. It is well-known that for nonsymmetric matrix systems it is extremely difficult to provide a concrete proof for the rate of convergence of an iterative method, as opposed to symmetric systems for which convergence

is controlled only by the eigenvalues of the preconditioned system $P^{-1}J$. For nonsymmetric operators the spectrum alone does not determine the convergence of GMRES [7]; furthermore, the usual techniques for establishing eigenvalue bounds do not apply to the nonsymmetric case [3]. However, in practice, the tight clustering of eigenvalues for the preconditioned system frequently leads to strong convergence properties, even though this is not theoretically guaranteed. We present an analysis that establishes spectral equivalence of the proposed preconditioner, and corroborate this analysis with numerical experiments in section 4 which demonstrate mesh independence.

Observe that when applying the preconditioner

$$\begin{bmatrix} \tilde{A} & 0 \\ C & \tilde{S} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{A}^{-1} & 0 \\ 0 & \tilde{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -C\tilde{A}^{-1} & I \end{bmatrix} \tag{3.1}$$

for the Jacobian J, the crucial steps are applying $\tilde{A}^{-1}$ and $\tilde{S}^{-1}$. For the matrix system (2.1), the inverse of the sub-block $A = (1 + \Delta t\theta\sigma)M$ may be approximated accurately and cheaply using a Jacobi iteration or Chebyshev semi-iteration [5, 6, 19]. It is therefore instructive to consider the preconditioned system

$$\begin{bmatrix} A & 0 \\ C & \tilde{S} \end{bmatrix}^{-1} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A^{-1} & 0 \\ 0 & \tilde{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & A^{-1}B \\ 0 & \tilde{S}^{-1}S \end{bmatrix}, \tag{3.2}$$

where the inexactness of our preconditioner arises from the stated approximation $\tilde{S}$ of the Schur complement S. The eigenvalues of this system, which serve as a guide as to the effectiveness of the preconditioner, are either equal to 1 or correspond to the eigenvalues of $\tilde{S}^{-1}S$.

We therefore examine the spectrum of the preconditioned Schur complement $\tilde{S}^{-1}S$, the matrix which governs the effectiveness of our algorithm, in the ideal setting where the matrix $M + \varepsilon\sqrt{c}K$ is inverted exactly. We first present a short result concerning the reality of the eigenvalues.

LEMMA 3.1. *The eigenvalues of $\tilde{S}^{-1}S$ are real.*

Proof. If $v \in \text{null}(K)$ and $v \neq 0$, then

$$Sv = \lambda\tilde{S}v \tag{3.3}$$

implies

$$Mv = \lambda Mv, \tag{3.4}$$

and hence $\lambda = 1 \in \mathbb{R}$.

On the other hand, if $v \notin \text{null}(K)$, then (3.3) implies

$$KM^{-1}Sv = \lambda KM^{-1}\tilde{S}v \quad \Rightarrow \quad Fv = \lambda Gv \quad \Rightarrow \quad v^*Fv = \lambda v^*Gv, \tag{3.5}$$

where the matrices F and G are given by

$$F = K + \varepsilon^2 cKM^{-1}KM^{-1}K + cKM^{-1}M_EM^{-1}K, \tag{3.6}$$

$$G = K + \varepsilon^2 cKM^{-1}KM^{-1}K + 2\varepsilon\sqrt{c}KM^{-1}K. \tag{3.7}$$

Using the symmetry of F and G, it follows that v^*Fv and v^*Gv are real (with v^*Gv also positive), and hence $\lambda \in \mathbb{R}$. $\square$

To motivate why $\tilde{S}$ should serve as an effective approximation of S, we now make use of the above analysis to present a result on the eigenvalues of $\tilde{S}^{-1}S$.

LEMMA 3.2. *The eigenvalues of $\tilde{S}^{-1}S$ satisfy:*

$$\lambda(\tilde{S}^{-1}S) \in \left[\frac{1}{2} - \frac{1}{4\varepsilon}\sqrt{c}, 1 + \frac{1}{4\varepsilon}\sqrt{c}\lambda_+\right], \tag{3.8}$$

with $\lambda_+ = \max\{\lambda_{\max}(M^{-1}M_E), 0\}$, and $\lambda_{\max}$ the maximum eigenvalue of a matrix.

Proof. As for Lemma 3.1, we split the analysis into two cases. The setting where $v \in \text{null}(K)$ is again straightforward, and the resulting eigenvalue is $\lambda = 1$.

In the case $v \notin \text{null}(K)$, we examine (3.5) once more. Then,

$$\lambda = \frac{v^*Fv}{v^*Gv} = 1 - \underbrace{\frac{2\varepsilon\sqrt{c}\,v^*KM^{-1}Kv}{v^*(K + \varepsilon^2cKM^{-1}KM^{-1}K + 2\varepsilon\sqrt{c}KM^{-1}K)v}}_{\mathcal{R}_1}$$

$$+ c\underbrace{\frac{v^*KM^{-1}M_EM^{-1}Kv}{v^*(K + \varepsilon^2cKM^{-1}KM^{-1}K + 2\varepsilon\sqrt{c}KM^{-1}K)v}}_{\mathcal{R}_2}.$$

First observe that $\mathcal{R}_1 > 0$. To find an upper bound on $\mathcal{R}_1$, we note that $\mathcal{R}_1 = (a^*b + b^*a)/(a^*a + b^*b + a^*b + b^*a)$, where $a = K^{1/2}v$, $b = \varepsilon\sqrt{c}K^{1/2}M^{-1}Kv$, and therefore $\mathcal{R}_1 \leq \frac{1}{2}$ by simple algebraic manipulation. Therefore $\mathcal{R}_1 \in (0, \frac{1}{2}]$.

Next, observe that

$$\mathcal{R}_2 = c\underbrace{\frac{v^*KM^{-1}M_EM^{-1}Kv}{v^*KM^{-1}Kv}}_{\mathcal{R}_{2,1}} \cdot \underbrace{\frac{v^*KM^{-1}Kv}{v^*(K + \varepsilon^2cKM^{-1}KM^{-1}K + 2\varepsilon\sqrt{c}KM^{-1}K)v}}_{\mathcal{R}_{2,2}}.$$

As $\mathcal{R}_{2,2} = \frac{1}{2\varepsilon\sqrt{c}}\mathcal{R}_1$, we see that $\mathcal{R}_{2,2} \in (0, \frac{1}{4\varepsilon\sqrt{c}}]$. Further, writing $\bar{v} = M^{-1/2}Kv \neq 0$, we observe that

$$\mathcal{R}_{2,1} = \frac{\bar{v}^*M^{-1/2}M_EM^{-1/2}\bar{v}}{\bar{v}^*\bar{v}} \in \left[\lambda_{\min}(M^{-1}M_E), \lambda_{\max}(M^{-1}M_E)\right]$$

by considering cases where $\mathcal{R}_{2,1}$ could be positive or negative. Writing the function associated with v as $v_h = \sum_i v_i\phi_i$, where v_i denote the entries of v and ϕ_i denote the basis functions of the finite element discretization, we see that

$$v^*M_Ev = \int_\Omega (3u^2 - 1)v_h^2 \, \mathrm{d}\Omega \geq -\int_\Omega v_h^2 \, \mathrm{d}\Omega = -v^*Mv,$$

and hence that $\lambda_{\min}(M^{-1}M_E) \geq -1$. Combining the derived bounds, we see that $\mathcal{R}_2 \in [-\frac{1}{4\varepsilon}\sqrt{c}, \frac{1}{4\varepsilon}\sqrt{c}\lambda_+]$, which leads to bounds for $\lambda(\tilde{S}^{-1}S)$ as above. $\square$

REMARK 3.1. *If each Newton iterate satisfies $u \leq 1$, as guided by the values at the two pure phases, it also holds that $\lambda_{\max}(M^{-1}M_E) \leq 2$.*

We also note that we observe the largest eigenvalue of $M^{-1}M_E$ to be mesh independent, and that the bounds for $\lambda(\tilde{S}^{-1}S)$ can therefore be driven tighter by decreasing Δt. Therefore, as the dimension of the matrix system is increased by taking a finer discretization in space or time, we predict that our preconditioner should not worsen in performance. A necessary requirement for the choice of Δt is to ensure that the smallest eigenvalue of $\tilde{S}^{-1}S$ is bounded away from 0. This is equivalent to writing

$$\frac{1}{4\varepsilon}\sqrt{c} < \frac{1}{2} \quad \Leftrightarrow \quad \sqrt{c} < 2\varepsilon \quad \Leftrightarrow \quad \Delta t < \frac{4\varepsilon^2}{\theta}(1 + \Delta t\theta\sigma),$$

which is guaranteed as $\varepsilon \to 0$ provided $\Delta t < \frac{4\varepsilon^2}{\theta}$. We uphold this condition on Δt in our numerical tests.

Furthermore, if Δt is chosen to scale like ε^2, then the eigenvalue bounds asymptote to a constant as the interfacial thickness parameter $\varepsilon \to 0$. Hence, we also expect the preconditioner to be robust to changes in this parameter. Note that the scaling $\Delta t \sim \varepsilon^2$ is typically necessary for stability in time discretization schemes [1], and thus this criterion does not represent an additional restriction on the timestep size. As this scaling ensures that $\lambda(\tilde{S}^{-1}S)$ is bounded away from 0, it also ensures that the eigenvalues of S itself are bounded away from 0, therefore guaranteeing that the Schur complement is invertible.

In the next section we demonstrate how our proposed solver performs for a practical test problem, and examine whether the predicted robustness is achieved.

3.1. Comment on the symmetric matrix system. We now wish to detail how our preconditioning strategy could be adapted to the symmetrized matrix system (2.9). As the matrix K is not invertible due to the Neumann boundary conditions imposed, one might instead devise a saddle point preconditioner for the perturbed matrix

$$J_D = \begin{bmatrix} \frac{\Delta t \theta}{1+\Delta t \theta \sigma} K_D & M \\ M & -\varepsilon^2 K_D - M_E \end{bmatrix}.$$

This preconditioner may then be used to solve the original system (2.9). Here, K_D denotes some perturbed version of K which is symmetric positive definite, but also resembles closely the properties of K itself—for instance, one may achieve this by artificially imposing a Dirichlet boundary condition somewhere in the domain.

Then, one may take as a preconditioner

$$P_D = \begin{bmatrix} \frac{\Delta t \theta}{1+\Delta t \theta \sigma} K_D & 0 \\ 0 & \tilde{S}_D \end{bmatrix},$$

where $\tilde{S}_D$ approximates the negative Schur complement $S_D = \frac{1}{c}MK_D^{-1}M + \varepsilon^2 K_D + M_E$ of J_D. Using a similar matching strategy to that of Section 2 leads to the choice

$$\tilde{S}_D = \left(\frac{1}{\sqrt{c}}M + \varepsilon K_D\right) K_D^{-1} \left(\frac{1}{\sqrt{c}}M + \varepsilon K_D\right) = \frac{1}{c}MK_D^{-1}M + \varepsilon^2 K_D + \frac{2\varepsilon}{\sqrt{c}}M.$$

Eigenvalue bounds for $\tilde{S}_D^{-1}S_D$ are identical to those in Lemma 3.2. In more detail, they may be described using

$$\lambda = \frac{v^T S_D v}{v^T \tilde{S}_D v} = 1 - \underbrace{\frac{\frac{2\varepsilon}{\sqrt{c}} v^T M v}{v^T \tilde{S}_D v}}_{\bar{\mathcal{R}}_1} + \underbrace{\frac{v^T M_E v}{v^T M v}}_{\bar{\mathcal{R}}_{2,1}} \cdot \underbrace{\frac{v^T M v}{v^T \tilde{S}_D v}}_{\bar{\mathcal{R}}_{2,2}}.$$

It can then be shown that $\bar{\mathcal{R}}_1 \in (0, \frac{1}{2}]$ using the same argument as in Lemma 3.2, except with $a = \frac{1}{\sqrt{c}}K_D^{-1/2}Mv$, $b = \varepsilon K_D^{1/2}v$, and one can then use that $\bar{\mathcal{R}}_{2,2} = \frac{\sqrt{c}}{2\varepsilon}\bar{\mathcal{R}}_1$. Putting all the pieces together then produces the same bounds as before.

The advantage of this approach is that convergence with MINRES [15] would depend solely on the eigenvalues of $P_D^{-1}J_D$.[1] A disadvantage is that one must choose

---

[1] Some research has been carried out on analyzing the convergence of iterative methods for non-symmetric systems in a similar way, such as the field-of-values analysis in [2, 10]. Even if this approach were possible for the nonsymmetric matrix J discussed in this paper, it would require a wholly different analysis.
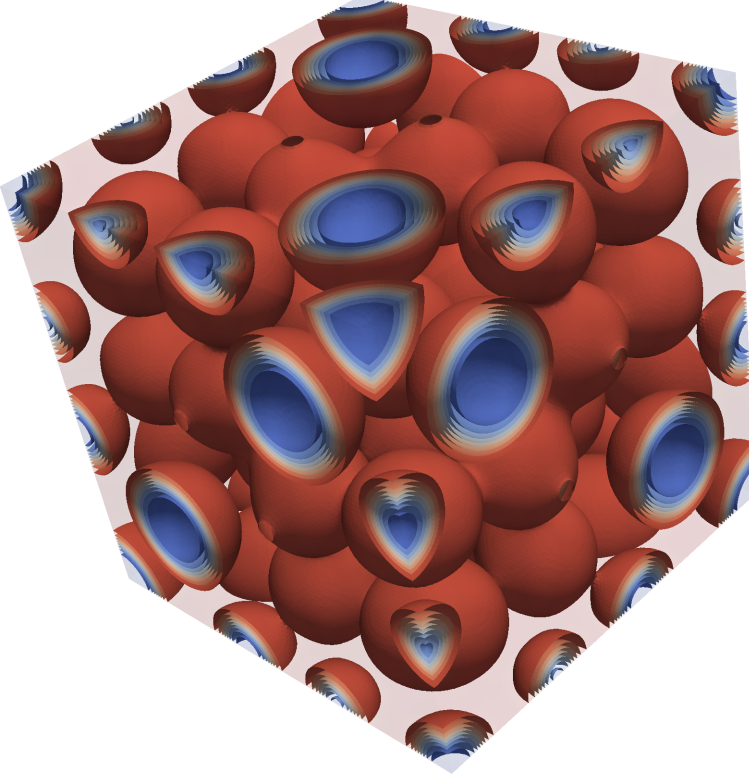
FIG. 4.1. *The final concentration u of the simulation described in section 4. The solution consists of regions of negative material ($u \approx -1$, in blue) embedded within a background of positive material ($u \approx 1$, in red). The figure shows the isosurfaces corresponding to constant values of u.*

precisely how to define K_D, and must assemble an additional matrix.

4. Numerical results. The solver was implemented in 338 lines of Python using version 1.6 of the FEniCS finite element library [11][2]. The experiment conducted used $\Omega = (0,1)^3$, $m = 0.4$, $\varepsilon = 0.02$, $\sigma = 100$, and an initial condition of

$$u_0(x, y, z) = m + p(x, y, z), \tag{4.1}$$

where the perturbation p must be chosen to have integral zero and satisfy $\nabla p \cdot n = 0$ on the boundary. In this experiment we chose

$$p(x, y, z) = \frac{\cos(2\pi x)\cos(2\pi y)\cos(2\pi z)}{50}. \tag{4.2}$$

The initial condition for w was computed from u_0 via (1.3b). In this parameter regime the solution is expected to consist of spherical regions of negative material ($u \approx -1$) embedded within a background of positive material ($u \approx 1$), see Figure 4.1.

The discretization used standard piecewise linear finite elements for u and w, a timestep $\Delta t = \varepsilon^2$, a final time $T = 300\Delta t$, and an implicitness parameter $\theta = 0.5$. The mesh was generated to achieve approximately 2.5×10^5 degrees of freedom per

---

[2]The code is available at http://bitbucket.org/pefarrell/ok-solver.

Degrees of freedom ($\times 10^6$)	Number of cores	Iterations
0.265	1	8.2
2.060	8	8.0
16.24	64	8.0
128.9	512	8.0
1027	4096	8.0

TABLE 4.1

Average number of GMRES iterations per Newton step with preconditioner (2.3) and Schur complement approximation (2.8) as the problem is weakly scaled. The number of iterations required remains constant as the mesh is refined.

ε	Δx	Δt	Iterations
0.02	0.01	0.0004	8.0
0.01	0.005	0.0001	8.0
0.005	0.0025	0.000025	8.0

TABLE 4.2

Average number of GMRES iterations per Newton step with preconditioner (2.3) and Schur complement approximation (2.8) as the interfacial thickness ε is varied (and with it the spatial discretization Δx and timestep Δt). The performance of the preconditioner does not change as $\varepsilon \to 0$.

core, to investigate how the number of Krylov iterations required to solve (2.1) varies as the mesh is refined. The preconditioner (2.3) approximated the action of A^{-1} with ten Chebyshev semi-iterations with SOR preconditioning, and approximated the action of S^{-1} with two Richardson iterations of $\tilde{S}^{-1}$. Each action of $\hat{S}^{-1}$ was in turn approximated with five V-cycles of the BoomerAMG algebraic multigrid solver [8]. GMRES was used as the outer Krylov solver. Each timestep typically took one or at most two Newton iterations.

The essential properties of a good preconditioner are that the Krylov iteration counts are low, they grow slowly (if at all) with mesh refinement, and they are robust to variation in parameters. We therefore examined the average number of Krylov iterations required per Newton step in two numerical experiments: in the first, all parameters were fixed, and only the mesh was refined; in the second, the interfacial thickness ε was reduced, and with it the spatial discretization (to resolve the interface) and the temporal discretization (to retain stability). The experiments were conducted on Hexagon, a Cray XE6m-200 hosted at the University of Bergen, and ARCHER, a Cray XC30 hosted at the University of Edinburgh.

The results of the first experiment are shown in Table 4.1. The average number of Krylov iterations per Newton step remains very close to 8, even as the number of degrees of freedom is increased by four orders of magnitude. The results of the second experiment are shown in Table 4.2. The average number of Krylov iterations required per Newton step does not vary as $\varepsilon \to 0$.

5. Conclusions. We have presented a new preconditioner for the Ohta–Kawasaki equation that models diblock copolymer melts. An approximation to the Schur complement was derived using a matching strategy. The proposed preconditioner yields mesh independent convergence, is robust to changes in interfacial thickness if a timestep criterion is satisfied, and requires no reassembly or reconstruction between

Newton steps. This enables the solution of very fine discretizations with billions of degrees of freedom.

References.

[1] B. Benešová, C. Melcher, and E. Süli, *An implicit midpoint spectral approximation of nonlocal Cahn–Hilliard equations*, SIAM J. Numer. Anal., 52 (2014), pp. 1466–1496.

[2] M. Benzi and M. A. Olshanskii, *Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem*, SIAM J. Numer. Anal., 49 (2011), pp. 770–788.

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

[4] R. Choksi, M. A. Peletier, and J. F. Williams, *On the phase diagram for microphase separation of diblock copolymers: an approach via a nonlocal Cahn–Hilliard functional*, SIAM J. Appl. Math., 69 (2009), pp. 1712–1738.

[5] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, Part I*, Numer. Math., 3 (1961), pp. 147–156.

[6] ——, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, Part II*, Numer. Math., 3 (1961), pp. 157–168.

[7] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.

[8] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.

[9] I. C. F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[10] A. Klawonn and G. Starke, *Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis*, Numer. Math., 81 (1999), pp. 577–594.

[11] A. Logg, K. A. Mardal, G. N. Wells, et al., *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2011.

[12] B. Morini, V. Simoncini, and M. Tani, *Spectral estimates for unreduced symmetric KKT systems arising from interior point methods*, Numer. Linear Alg. Appl., (2016).

[13] 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.

[14] T. Ohta and K. Kawasaki, *Equilibrium morphology of block copolymer melts*, Macromolecules, 19 (1986), pp. 2621–2632.

[15] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[16] Q. Parsons, *Numerical Approximation of the Ohta–Kawasaki Functional*, master's thesis, University of Oxford, Oxford, UK, 2012.

[17] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Alg. Appl., 19 (2012), pp. 816–829.

[18] Y. Saad and M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[19] 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.