

NKS Method for the Implicit Solution of a Coupled Allen-Cahn/Cahn-Hilliard System*

Chao Yang¹, Xiao-Chuan Cai², David E. Keyes³, and Michael Pernice⁴

1 Coupled Allen-Cahn/Cahn-Hilliard system

Coupled Allen-Cahn/Cahn-Hilliard (AC/CH) systems, often found in phase-field simulations, are prototype systems that admit simultaneous ordering and phase separation. Numerical methods to solve coupled AC/CH systems are studied in e.g., [2, 6, 8, 9, 10, 11]. However, except for [9] and [10], the above works are based on explicit methods that require very small time step size to advance the solution and need many time steps for long time integrations. Fully implicit methods enjoy an advantage that the stability limit on the time step size is greatly relaxed. The purpose of this paper is to study efficient and scalable algorithms based on domain decomposition methods for the fully implicit solution of a coupled AC/CH system.

There are several different ways to couple the AC and the CH equations. Among them we restrict our study to the original form introduced in [3], which is

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot c(u, v) \nabla \frac{\delta E(u, v)}{\delta u}, \\ \frac{\partial v}{\partial t} = -\frac{c(u, v)}{\rho} \frac{\delta E(u, v)}{\delta v}. \end{cases} \quad (1)$$

where u and v are functions of $\mathbf{x} \in \Omega \subset \mathbf{R}^2$ and $t \in [0, +\infty)$. Both u and v are bounded with restrictions: $u \in [0, 1]$, $v \in [-1/2, 1/2]$ and $(u \pm v) \in [0, 1]$. Here the first equation in (1) is the Cahn-Hilliard equation in which u represents a conserved concentration field for the phase separation; the second equation in (1) is the Allen-Cahn equation where v denotes a non-conserved order parameter for the anti-phase coarsening.

In (1), the mobility $c(u, v) = u(1-u)(1/4 - v^2)$ is degenerate at pure phases and the density ρ is a positive constant. The free energy functional $E(u, v)$ reads

¹ Chao Yang, Institute of Software, Chinese Academy of Sciences, Beijing 100190, and State Key Laboratory of High Performance Computing, Changsha 410073, China, e-mail: yangchao@iscas.ac.cn ² Xiao-Chuan Cai, Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA, e-mail: cai@cs.colorado.edu ³ David E. Keyes, CEMSE Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, e-mail: david.keyes@kaust.edu.sa ⁴ Michael Pernice, Idaho National Laboratory, Idaho Falls, ID 83415, USA e-mail: michael.pernice@inl.gov

* This work was supported in part by DE-FC02-06ER25784. The first author also received supports from NSFC under 61170075, 91130023 and 61120106005, and from 973 Program of China under 2011CB309701.

$$E(u, v) = \int_{\Omega} \left\{ \frac{\gamma}{2} (|\nabla u|^2 + |\nabla v|^2) + \theta (\Phi(u+v) + \Phi(u-v)) + \frac{\alpha}{2} u(1-u) - \frac{\beta}{2} v^2 \right\} d\mathbf{x}, \quad (2)$$

where $\Phi(z) = z \ln z + (1-z) \ln(1-z)$ and $\gamma, \theta, \alpha, \beta$ are all positive constants. It then follows that

$$\begin{aligned} \frac{\delta E}{\delta u} &= -\gamma \Delta u + \theta \Phi'(u+v) + \theta \Phi'(u-v) - \alpha(u-1/2), \\ \frac{\delta E}{\delta v} &= -\gamma \Delta v + \theta \Phi'(u+v) - \theta \Phi'(u-v) - \beta v. \end{aligned} \quad (3)$$

In the current study we consider periodic boundary conditions for both u and v . Other boundary conditions lead to similar numerical results and the performance of our proposed solver is not sensitive to them. The AC/CH system (1) is closed with the above boundary conditions and an initial condition $u = u^0, v = v^0$ at $t = 0$.

2 Discretizations

We restrict our study in this paper to the case of a 2-dimensional square domain Ω . A second-order accurate cell-centered finite difference (CCFD) scheme on a uniform mesh is applied to the system. The details of the CCFD scheme is omitted here due to the page limit.

Special attention should be paid when considering the time integration of the AC/CH system (1). Because of the high-order spatial differentiation in the system, explicit methods become impractical due to the severe restriction on the time step size. In order to relax the restriction and obtain the steady-state solution in an efficient way, we use the fully implicit backward Euler scheme. We remark that due to the co-existence of both diffusive and anti-diffusive terms in the AC/CH system, the backward Euler scheme is not unconditionally stable. Other more efficient and accurate implicit schemes will be studied in a forthcoming paper.

After spatially discretizing the AC/CH system, u and v are replaced with their cell-centered values U and V respectively. Denote the spatial discretizations of the right-hand-sides in the two equations in (1) as $M(U, V)$ and $N(U, V)$ respectively, the nonlinear algebraic system arising at each time step reads

$$\begin{cases} \mathcal{M}_k(U^{k+1}, V^{k+1}) := \frac{U^{k+1} - U^k}{\Delta t^k} - M(U^{k+1}, V^{k+1}) = 0, \\ \mathcal{N}_k(U^{k+1}, V^{k+1}) := \frac{V^{k+1} - V^k}{\Delta t^k} - N(U^{k+1}, V^{k+1}) = 0, \end{cases} \quad (4)$$

where Δt^k is the step size and U^{k+1}, V^{k+1} are the solutions for the k -th time step. Due to the multiple temporal scales admitted by the AC/CH system, Δt^k is adaptively controlled by a method that is analogous to the switched evolution/relaxation method [5, 7]. More specifically, we start with a relatively small time step size Δt^0 and adjust its value according to

$$\begin{aligned} \Delta t^k &= \max(1/r, \min(r, s)) \Delta t^{k-1}, \\ s &= \left(\frac{\|(\mathcal{M}_{k-1}(U^{k-1}, V^{k-1}), \mathcal{N}_{k-1}(U^{k-1}, V^{k-1}))^T\|_2}{\|(\mathcal{M}_k(U^k, V^k), \mathcal{N}_k(U^k, V^k))^T\|_2} \right)^p, \end{aligned} \quad (5)$$

for $k = 1, 2, 3, \dots$, where we use $r = 1.5$ and $p = 0.75$.

3 Newton-Krylov-Schwarz solver

An inexact Newton method is applied to solve the nonlinear system (4) at each time step. We denote the solution of (4) at the k -th time step as $W^{k+1} = (U^{k+1}, V^{k+1})^T$. The initial guess $X_0 = W^k$ is set to be the solution of the previous time step, then the approximate solution X_{n+1} is obtained by

$$X_{n+1} = X_n + \lambda_n S_n, \quad n = 0, 1, \dots \quad (6)$$

Here λ_n is the steplength determined by a linesearch procedure and S_n is the Newton correction vector. To calculate S_n for each Newton iteration, a right-preconditioned linear system

$$J_n M^{-1} (M S_n) = -F_k(X_n) \quad (7)$$

is constructed and solved approximately by using a GMRES method that restarts every 30 iterations. Here $F_k(X_n) = (\mathcal{M}_k(X_n), \mathcal{N}_k(X_n))^T$ is the nonlinear residual and

$$J_n = \frac{\partial F_k(X_n)}{\partial X_n} \quad (8)$$

is the Jacobian matrix.

In (4) M^{-1} is an additive Schwarz preconditioner. We first partition Ω into np non-overlapping subdomains Ω_p , $p = 1, 2, \dots, np$. An overlapping decomposition is obtained by extending each subdomain with δ mesh layers. Denote the overlapping subdomain as Ω_p^δ . The one-level restricted additive Schwarz (RAS, [4]) preconditioner is

$$M^{-1} = \sum_{p=1}^{np} (R_p^0)^T \text{inv}(B_p) R_p^\delta. \quad (9)$$

Here R_p^δ and $(R_p^0)^T$ serve as a restriction operator and an interpolation operator respectively; their detailed definitions can be found in [4].

In (9), $\text{inv}(B_p)$ is either an exact or approximate inverse of the subdomain problem defined by B_p . Choosing proper boundary conditions for the subdomain problems has a great impact on the convergence of the RAS preconditioner. Since the AC/CH system (1) contains two differential equations with different orders, it is natural to impose different boundary conditions. For the first equation in (1) we follow [12] by employing the following homogeneous boundary conditions

$$u = (\nabla u) \cdot \mathbf{n} = 0, \quad \partial \Omega_p^{\delta+1} \setminus \partial \Omega, \quad (10)$$

where \mathbf{n} is the outward normal of $\partial\Omega_p^{\delta+1}$. For the second equation in (1), the boundary conditions are simply

$$v = 0, \quad \partial\Omega_p^\delta \setminus \partial\Omega. \quad (11)$$

We remark that the above boundary conditions for the subdomain problems are essential for the success of the NKS solver. Other boundary conditions are also tested but only lead to poor convergence of GMRES. To solve the subdomain problems, we use either a sparse LU factorization or a sparse incomplete LU (ILU) factorization. In doing the factorization, we use a point-block ordering for the subdomain matrix and keep the coupling between the two components at each mesh cell. Within each time step, the factorization is only done once at the first Newton iteration and is reused thereafter.

4 Numerical experiments

We carry out numerical experiments on a Dell supercomputer located at the University of Colorado Boulder. The computer consists of 1368 compute nodes, with two hex-core 2.8Ghz Intel Westmere processors and 24GB local memory in each node. Our algorithm is implemented based on the Portable, Extensible Toolkits for Scientific computations (PETSc, [1]) library. In the numerical experiments we use all 12 cores in each node and assign one subdomain per processor core. The relative stopping conditions for the Newton and GMRES iteration are respectively 1×10^{-6} and 1×10^{-5} .

4.1 Steady-state solution

The test case we study here is taken from [11]. The initial condition for the test is a randomly distributed state $(U^0, V^0) = (0.05 + \delta_u, \delta_v)$, where $\max(\|\delta_u\|_\infty, \|\delta_v\|_\infty) \leq 0.05$. The parameters are set as: $\alpha = 4$, $\beta = 2$, $\gamma = 0.005$, $\theta = 0.1$, $\rho = 0.001$.

We run the test case on a 256×256 mesh with an initial time step size $\Delta t^0 = 0.001$. The time step size is then adaptively controlled by using (5). Thanks to the fully implicit method and the adaptive time stepping strategy, we are able to obtain the steady-state solution at about $t = 100$, as seen in Figure 1 and 2. From the figures we observe that when $t < 1.4$ both the spinodal decomposition and the order-disorder type instability occur but after that the order parameter quickly tends to zero as the conserved concentration field coarsens to a stabilized state. Provided in Figure 3 is the evolution history of the time step size and the total free energy. It can be seen that by using the adaptive strategy, the time step is successfully adjusted by several orders of magnitude. The total free energy decays and finally approaches to its minimizer when the solution arrives at the steady-state.

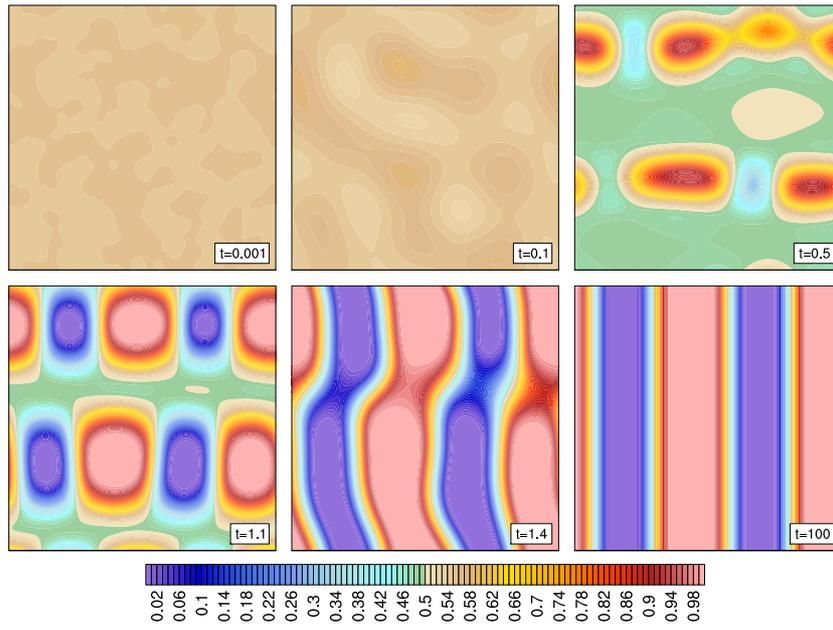


Fig. 1 Contour plots of the conserved concentration field u .

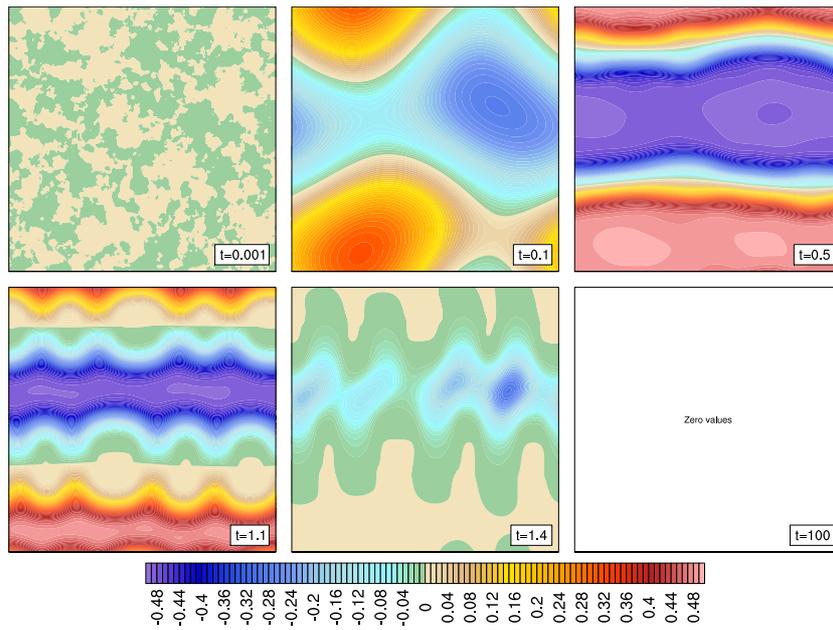


Fig. 2 Contour plots of the non-conserved order parameter v .

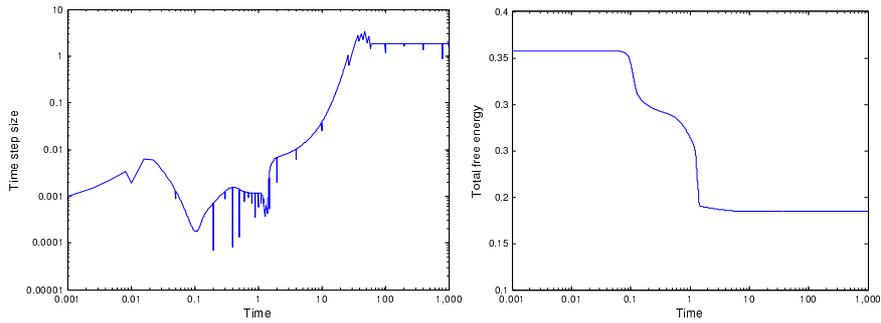


Fig. 3 Evolution history of the time step size (left panel) and the total free energy (right panel).

We remark that because of the severe stability restriction on the time step size, it is often difficult to obtain the steady-state solution when an explicit method is used. In [11], although similar tests are conducted, no steady-state solutions are obtained due to the explicit time stepping.

4.2 Parameters in the NKS solver

To understand how the parameters in the Schwarz preconditioner affect the performance, in the following experiments we run the test case on a 1152×1152 mesh with 144 processor cores by using a fixed the time step size $\Delta t = 1.0 \times 10^{-5}$ for only the first 20 steps.

We first examine the effects of different subdomain solvers. The overlap size is fixed at $\delta = 2$. In Table 1 we show the total numbers of Newton and GMRES iterations as well as the total compute time. Results for both LU and ILU with different fill-in levels are provided. From the table we find surprisingly that GMRES

Table 1 Effects of different subdomain solvers. Here “n/c” means no convergence.

	ILU(2)	ILU(4)	ILU(8)	LU	LU-blk	LU-blk-reuse
#Newton	n/c	n/c	n/c	41	41	41
#GMRES	n/c	n/c	n/c	1225	1225	1243
Time (s)	n/c	n/c	n/c	138.6	89.2	65.6

doesn’t converge when ILU is the subdomain solver, even with large fill-in levels. When a sparse LU factorization is used as subdomain solver, although the point-block version doesn’t change the number of iterations, the compute time is saved by around 35% compared to the non-block version. To reduce the compute time, we perform the subdomain LU factorization only once per time step, and reuse it for all the Newton iterations within the same time step. By reusing the LU factorization the

total compute time is cut by around 26% despite of the slight increase of the number of GMRES iterations. Based on the above observations, for all the following tests, we use the point-block version of sparse LU factorization and reuse the factorization within each time step.

We next investigate the performance of the NKS solver with different overlap δ . Table 2 shows the total numbers of Newton and GMRES iterations as well as the total compute time for $\delta = 0, 1, \dots, 6$. It is observed from the table that: (1) the

Table 2 Results on using different overlaps.

δ	0	1	2	3	4	5	6
#Newton	41	41	41	41	41	41	41
#GMRES	21274	2482	1243	840	642	513	440
Time (s)	205.6	95.3	65.6	55.7	45.7	50.8	52.0

number of Newton iterations does not change as δ varies; (2) the number of GMRES iterations reduces when δ becomes larger; and (3) the total compute time is optimal for $\delta = 4$ in the test. Therefore we use $\delta = 4$ in our scalability tests.

4.3 Parallel scalability

In the parallel scalability tests, we fix the overlap size to be $\delta = 4$ and choose the point-block version of the sparse LU factorization (reused with each time step) as the subdomain solver. We run the tests on a 1152×1152 mesh for 20 time steps with $\Delta t = 1.0 \times 10^{-5}$ and gradually double the number of processor cores. As shown in Figure 4, when the number of processor cores is increased the total number of Newton iterations stays unchanged while the total number of GMRES iterations increases slightly. Further from Figure 5 we observe that the total compute time is reduced almost linearly as more processor cores are used. A total of 12.35 speedup is achieved when the number of processor cores increases from 144 to 2304, leading to a parallel efficiency of 78.1%.

References

- Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)
- Barrett, J.W., Blowey, J.F.: Finite element approximation of a degenerate Allen-Cahn/Cahn-Hilliard system. *SIAM J. Numer. Anal.* **39**, 1598–1624 (2002)
- Cahn, J., Novick-Cohen, A.: Evolution equations for phase separation and ordering in binary alloys. *J. Statist. Phys.* **76**, 877–909 (1994)

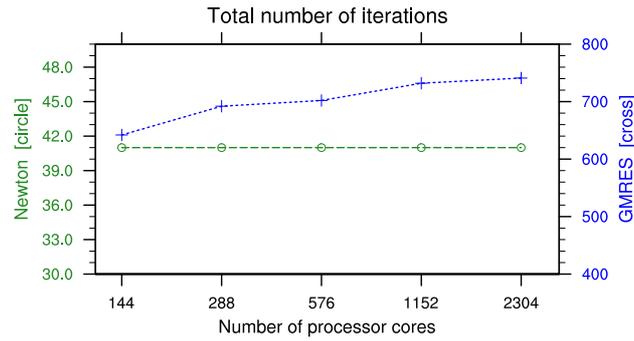


Fig. 4 Total numbers of Newton and GMRES iterations for the first 20 time steps.

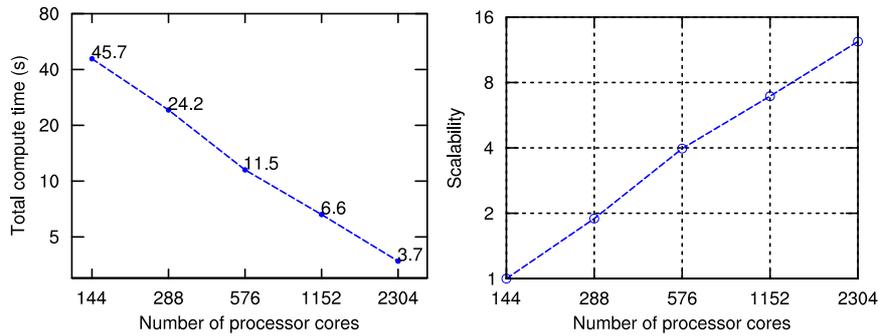


Fig. 5 Total compute time (left) and parallel scalability (right) results.

4. Cai, X.-C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.* **21**, 792–797 (1999)
5. Gropp, W.D., Kaushik, D.K., Keyes, D.E., Smith, B.: Performance modeling and tuning of an unstructured mesh CFD application. In: *Proc. Supercomputing 2000*. IEEE Computer Society (2000)
6. Millett, P.C., Rokkam, S., El-Azab, A., Tonks, M., Wolf, D.: Void nucleation and growth in irradiated polycrystalline metals: a phase-field model. *Modelling Simul. Mater. Sci. Eng.* **17**, 064,003 (2009)
7. Mulder, W.A., Leer, B.V.: Experiments with implicit upwind methods for the Euler equations. *J. Comput. Phys.* **59**, 232–246 (1985)
8. Rokkam, S., El-Azab, A., Millett, P., Wolf, D.: Phase field modeling of void nucleation and growth in irradiated metals. *Modelling Simul. Mater. Sci. Eng.* **17**, 064,002 (2009)
9. Tonks, M.R., Gaston, D., Millett, P.C., Andrs, D., Talbot, P.: An object-oriented finite element framework for multiphysics phase field simulations. *Comput. Mater. Sci.* **51**, 20 – 29 (2012)
10. Wang, L., Lee, J., Anitescu, M., Azab, A.E., McInnes, L.C., Munson, T., Smith, B.: A differential variational inequality approach for the simulation of heterogeneous materials. In: *Proc. SciDAC 2011* (2011)
11. Xia, Y., Xu, Y., Shu, C.W.: Application of the local discontinuous Galerkin method for the Allen-Cahn/Cahn-Hilliard system. *Commun. Comput. Phys.* **5**, 821–835 (2009)
12. Yang, C., Cai, X.-C., Keyes, D.E., Pernice, M.: Parallel domain decomposition methods for the 3D Cahn-Hilliard equation. In: *Proc. SciDAC 2011* (2011)