

# A Scalable Implicit Solver for Phase Field Crystal Simulations

Chao Yang<sup>\*†</sup>, Xiao-Chuan Cai<sup>†</sup>

<sup>\*</sup> *Institute of Software, Chinese Academy of Sciences, Beijing 100190, P. R. China*

*Email: yangchao@iscas.ac.cn*

<sup>†</sup> *Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA*

*Email: chao.yang@colorado.edu, cai@cs.colorado.edu*

**Abstract**—The phase field crystal equation has become a popular model for simulating micro-structures in materials science but is very computationally expensive to solve. A highly scalable solver for phase field crystal modeling is presented in this paper. The equation is discretized with a stabilized implicit finite difference method and the time step size is adaptively controlled to obtain physically meaningful solutions. The nonlinear system arising at each time step is solved by using a parallel Newton-Krylov-Schwarz algorithm. In order to achieve good performance, low-order homogeneous boundary conditions are imposed on subdomain interfaces in the Schwarz preconditioner. Experiments are carried out to exploit optimal choices of the preconditioner type, the subdomain solver and the overlap size. Numerical results are provided to show that the solver is scalable to thousands of processor cores.

**Keywords**-phase field crystal equation; domain decomposition method; Newton-Krylov-Schwarz; restricted additive Schwarz; preconditioner; parallel scalability

## I. INTRODUCTION

Modeling micro-structural properties of crystalline materials poses a severe computational challenge, due in large part to the complexity in the non-equilibrium dynamics. The phase field crystal (PFC) equation, introduced by Elder et. al. [8], [9], has proven to be a versatile approach to model complicated micro-structures (e.g., defects) that most crystals exhibit, and has become increasingly popular in materials science; see, e.g., [10], [11], [18], [20], [22], [23].

The basis of the PFC equation is the free-energy functional that originates from the more advanced density functional theory of Hohenberg and Kohn [16]. In a PFC model, the density wave structure of a crystalline material is modeled by a high-order partial differential equation (PDE) that is valid at atomic-length level and is evolved with diffusive time-scale. Compared to molecular dynamics that are also accurate to describe microstructures on the atomic scale, the PFC equation allows time-scales that are typically magnitudes larger.

It is not easy to obtain an analytic solution of the PFC equation due to the nonlinearity of the problem. Therefore the PFC equation is usually solved numerically. Obtaining physically meaningful solutions of the PFC equation is computationally expensive because: (1) the PFC equation contains time scales that change in magnitudes during the time evolution; and (2) material scientists are often interested

in the long-time dynamics of crystalline materials. It is therefore of great importance to study scalable parallel algorithms for the PFC equation. Although numerical methods for the PFC equation have been investigated in a number of publications, e.g., [5], [14], [17], [26], [28], works dedicated to parallel algorithms are not yet to be seen. There are some successful studies on scalable parallel algorithms for some other phase-field problems such as the Cahn-Hilliard equation [29], [31] and the coupled Allen-Cahn/Cahn-Hilliard equations [24], [27], [30]. However, it is not clear if those algorithms can be applied to the PFC equation, because the PFC equation, compared to other phase-field models, contains solutions that are highly oscillatory due to the sixth-order differential term in the PDE.

In this paper a highly scalable parallel solver for PFC simulations is presented. In the solver we discretize the PFC equation with a stabilized implicit finite difference method and adaptively control the time step size during the simulation. A parallel Newton-Krylov-Schwarz algorithm is then applied to solve the nonlinear system arising at each time step. Several key issues in the method, including the type of the Schwarz preconditioner, the interface conditions for subdomain problems, the overlap size and the solver used to solve subdomain problems, are discussed and tested. Numerical experiments reveal that the PFC solver based on the Newton-Krylov-Schwarz algorithm performs well on a supercomputer with thousands of processor cores.

The rest of the paper is organized as follows. In Section II we introduce the PFC equation and the numerical methods to effectively discretize it in space and time. A detailed introduction of the Newton-Krylov-Schwarz algorithm to solve the nonlinear system at each time step is given in Section III. We then provide in Section IV some numerical results including experiments on the homogeneous crystallization in a supercooled liquid, performance tuning of the Schwarz preconditioner and scalability tests with thousands of processor cores. The paper is concluded in Section V.

## II. PHASE-FIELD CRYSTAL EQUATION

Non-equilibrium dynamics modeled in phase field simulations are often based on the minimization of a free energy functional. A typical free energy functional often found in

PFC simulations exhibits the following dimensionless form:

$$E(\phi) = \int_{\Omega} \left\{ \frac{1}{4}\phi^4 + \frac{1+\gamma}{2}\phi^2 - |\nabla\phi|^2 + \frac{1}{2}(\nabla^2\phi)^2 \right\} d\mathbf{x}, \quad (1)$$

where  $\gamma < 0$  is a parameter representing the quench depth for supercooling the material and  $\phi$  stands for a periodic order parameter (i.e., probability density) that is a function of the spatial variable  $\mathbf{x} \in \Omega \subset \mathbf{R}^2$  and the temporal variable  $t \in [0, +\infty)$ .

In PFC modeling, the density distribution of  $\phi$  is considered to be conserved during the non-equilibrium process. Inserting (1) into the system of conserved dynamics

$$\frac{\partial\phi}{\partial t} = \nabla^2 \frac{\delta E(\phi)}{\delta\phi},$$

we obtain the PFC equation

$$\frac{\partial\phi}{\partial t} = \nabla^2 [\phi^3 + (1+\gamma)\phi + 2\nabla^2\phi + \nabla^4\phi], \quad (2)$$

which is a sixth-order parabolic PDE. The PFC equation (2) is closed with periodic boundary conditions and an initial condition  $\phi = \phi^0$  at  $t = 0$ .

A cell-centered finite difference scheme on a uniform mesh is employed to spatially discretize the PFC equation. Due to existence of the sixth-order differential term in the equation, the stencil for the finite difference scheme exhibits a diamond shape with stencil width 3. In other words, the finite difference on mesh cell  $(i, j)$  depends on the values on mesh cell  $(i', j')$ , where  $|i' - i| + |j' - j| \leq 3$ . Omitting the lengthy details of the scheme, we denote the cell-centered values of  $\phi$  as  $\phi_{i,j}$  ( $i, j = 1, 2, \dots, N$ ) and the discrete form of the Laplacian operator  $\nabla^2$  as  $\nabla_h^2$ .

Special care should be taken when choosing method to integrate the PFC equation in the temporal direction. Explicit methods are too expensive to apply due to the severe stability limit on the time step size, which roughly reads

$$\Delta t \leq Ch^{-6},$$

where  $h$  is the mesh size and  $C$  is a constant. On the other hand, although greatly relaxed, fully implicit methods also suffer from stability issues because of the simultaneously diffusive-antidiffusive property of the PFC equation. To construct a stable scheme, we use the method introduced by Eyre [12], [13] in which the free energy functional is splitted into a convex part

$$E_1(\phi) = \int_{\Omega} \left\{ \frac{1}{4}\phi^4 + \frac{1+\gamma}{2}\phi^2 + \frac{1}{2}(\nabla^2\phi)^2 \right\} d\mathbf{x},$$

and a concave part

$$E_2(\phi) = - \int_{\Omega} |\nabla\phi|^2 d\mathbf{x}.$$

Then the terms related to the convex part are treated implicitly and the rest explicitly. By using the convex splitting

method, we obtain the following stabilized implicit scheme

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^k}{\Delta t^k} = \nabla_h^2 [f(\phi_{i,j}^{k+1}) + 2\nabla_h^2\phi_{i,j}^k + \nabla_h^4\phi_{i,j}^{k+1}], \quad (3)$$

for  $k = 0, 1, 2, \dots$ . Here  $f(\phi) = \phi^3 + (1+\gamma)\phi$  is a nonlinear function,  $\Delta t^k$  is the step size and  $\phi_{i,j}^k$  is the solution at the  $k^{\text{th}}$  time step. By using the above scheme, the energy of the discretized equation decays, which is consistent with the energy dissipation property of the PFC equation.

The PFC equation admits time scales that change in magnitudes as the dynamical system evolves. Therefore it is often impractical to use a fixed time step size during the entire simulation, especially when the long-time dynamics of a crystalline material are of interest. In order to conduct physically meaningful simulations and reduce the computational cost, we adaptively control the time step size  $\Delta t^k$  by using a strategy that is analogous to the switched evolution/relaxation method [15], [19]. More specifically, we start with a relatively small time step size  $\Delta t^0$  and adjust its value according to

$$\Delta t^k = \max(1/\alpha, \min(\alpha, \beta)) \Delta t^{k-1} \quad (4)$$

for  $k = 1, 2, \dots$ . Here  $\alpha > 1$  is a safeguard to avoid excessive change of the time step size between any two immediate time steps and  $\beta$  is obtained from

$$\beta = (\|r^{k-1}\|_2 / \|r^k\|_2)^p,$$

where  $0 < p < 1$  is used to control the adaptivity (larger  $p$  results in more aggressive adjustment of the time step size) and  $r^k$  is the residual with its components calculated by

$$r_{i,j}^k = \nabla_h^2 [f(\phi_{i,j}^k) + 2\nabla_h^2\phi_{i,j}^k + \nabla_h^4\phi_{i,j}^k].$$

In our PFC solver, we set the adaptivity parameters to be  $\alpha = 1.5$  and  $p = 0.75$ .

### III. NEWTON-KRYLOV-SCHWARZ ALGORITHM

After applying the stabilized implicit scheme (3), the PFC equation (2) is discretized into a nonlinear system

$$F(X) = 0 \quad (5)$$

at each time step. We employ a Newton-Krylov-Schwarz (NKS) algorithm [3] to solve (5) efficiently on parallel supercomputers. The NKS algorithm consists of three important components: (i) an inexact Newton method for the nonlinear system; (ii) a Krylov iterative method for the linear Jacobian system at each Newton iteration; and (iii) a Schwarz preconditioner for the linear solver.

In the NKS algorithm, an inexact Newton method is applied to solve the nonlinear system (5) at each time step. Choosing the initial guess  $X_0$  for the inexact Newton iteration may have a great impact on the convergence of the iteration. For a time-dependent problem such as the PFC equation, the solution of the previous time step serves as a

good initial guess. Given the current approximate solution  $X_n$  for (5), the inexact Newton method seeks the next approximate solution  $X_{n+1}$  via

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

Here  $\lambda_n$  is the steplength determined by a linesearch procedure (see, e.g., [6, Sec. 6.3]). The purpose of using the linesearch is to ensure that at each Newton iteration a local minimizer is found along the search direction.

In (6),  $S_n$  is the Newton correction vector, which is obtained by approximately solving the linear Jacobian system

$$J_n S_n = -F(X_n), \quad (7)$$

where  $J_n = \frac{\partial F(X_n)}{\partial X_n}$  is the Jacobian matrix. Compared to the classical Newton method, the inexact method is superior especially when the number of unknowns is large (e.g., of the order of millions or larger) due to the reason that the linear Jacobian system is solved approximately instead of exactly, leading to a substantial reduction of the computational cost. In the NKS algorithm, the linear Jacobian system (7) is solved by using a Krylov subspace methods. In practice, the Generalized Minimal RESidual (GMRES) method that restarts every 30 iterations is employed in our PFC solver.

To accelerate the convergence of the linear solver, we solve the right-preconditioned linear system

$$J_n M^{-1} (M S_n) = -F(X_n) \quad (8)$$

instead of the original one (7). Here the preconditioner  $M^{-1}$  is the key to the success of the linear solver. In large-scale parallel computing, additive Schwarz preconditioners based on domain decomposition theory [21], [25] not only help in improving the convergence but also are beneficial to the scalability of the linear solver.

To define the preconditioner  $M^{-1}$ , We first partition the computational domain  $\Omega$  into  $np$  non-overlapping subdomains  $\Omega_p (p = 1, 2, \dots, np)$ , then extend each subdomain by  $\delta$  mesh layers to form an overlapping decomposition  $\Omega = \cup_{p=1}^{np} \Omega_p^\delta$ . The classical additive Schwarz (AS, [7]) preconditioner is defined as

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

Here  $R_p^\delta$  serves as a restriction operator which restricts a vector to a new one that is defined in the overlapping subdomain  $\Omega_p^\delta$ , by discarding the components outside  $\Omega_p^\delta$ ;  $(R_p^\delta)^T$  represents an extension operator that maps a vector defined in the overlapping subdomain  $\Omega_p^\delta$  to a new one defined in the whole domain, by putting zeros at the components outside  $\Omega_p^\delta$ .

There are two modified versions of the AS preconditioner that may have some potential advantages. The first one is

the left restricted additive Schwarz (left-RAS, [4]) preconditioner that reads

$$M_{0\delta}^{-1} = \sum_{p=1}^{np} (R_p^0)^T \text{inv}(A_p) R_p^\delta. \quad (10)$$

The only difference between the left-RAS preconditioner and the AS preconditioner is the extension operator. Instead of  $(R_p^\delta)^T$ , the left-RAS preconditioner uses  $(R_p^0)^T$  which puts zeros not only outside  $\Omega_p^\delta$  but also outside  $\Omega_p$ . The other modification to the AS preconditioner is the right restricted additive Schwarz (right-RAS, [2]) preconditioner that is given by

$$M_{\delta 0}^{-1} = \sum_{p=1}^{np} (R_p^\delta)^T \text{inv}(A_p) R_p^0. \quad (11)$$

The only difference between the right-RAS preconditioner and the AS preconditioner is the restriction operator. Instead of  $R_p^\delta$ , the right-RAS preconditioner uses  $R_p^0$  which ignores the entries outside  $\Omega_p$  when doing the extension.

In (10),  $\text{inv}(A_p)$  represent a solution of the subdomain problem defined by  $A_p$ . Here the subdomain matrix can be directly generated as

$$A_p = R_p^\delta A (R_p^\delta)^T, \quad A = J_n. \quad (12)$$

However, the cost of (12) could be high because of the usage of the global matrix  $A$  in the formula. Therefore, instead of using (12), we choose to generate  $A_p$  from the discretization of the subdomain problem. Except for the boundaries that coincide with the boundary of the whole domain, there are interfaces between subdomains that require extra boundary conditions. Choosing different interface conditions for the subdomain problems can lead to very different convergence results. In our approach, we find that low-order homogeneous boundary conditions work well for this type of problem, because they are not only easy to implement, but also helpful in reducing the number of linear iterations. More precisely, we employ the following interface condition

$$u = \frac{\partial u}{\partial \mathbf{n}} = \frac{\partial^2 u}{\partial \mathbf{n}^2} = 0, \quad \partial \Omega_p^{\delta+3/2} \setminus \partial \Omega, \quad (13)$$

where  $\mathbf{n}$  is the outward normal of  $\partial \Omega_p^{\delta+3/2}$ . Here  $\Omega_p^{\delta+3/2}$  is a domain that is obtained by further extending the overlapping subdomain  $\Omega_p^\delta$  by one and a half mesh layers. The purpose of using  $\partial \Omega_p^{\delta+3/2}$  instead of  $\Omega_p^\delta$  is to ensure when solving the subdomain problem, all mesh points in  $\Omega_p^\delta$  have sufficient information, including the ghost points (i.e., halos) to perform the stencil calculations. We remark here that other interface conditions are also tested but only lead to poor convergence of linear solver. Similar observations were reported for the solution of the Cahn-Hilliard equation in [31]. After defining suitable interface conditions for the subdomain problems, we then solve them either directly by

using a sparse LU factorization or approximately by using a sparse incomplete LU (ILU) factorization.

A great advantage of the additive Schwarz preconditioners is that communication only occurs between neighboring subdomains during the restriction and extension processes. The major cost of the additive Schwarz preconditioners is the subdomain solves which are done sequentially without any inter-process communication. Therefore the locality of the additive Schwarz preconditioners are naturally good for massive parallel as long as the number of iterations is kept low. We further remark that compared to the classical AS preconditioner, the communication in the two restricted versions is reduced approximately by half because only the restriction or the extension step requires communication. This may further improve the performance of the preconditioner.

#### IV. NUMERICAL EXPERIMENTS

Numerical experiments are performed on Janus, a Dell supercomputer located at the University of Colorado Boulder. The computing nodes of Janus are interconnected via a non-blocking QDR Infiniband high performance network, 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 to each processor core. The Newton iteration stops when the 2-norm of the nonlinear residual is smaller than  $1 \times 10^{-7}$  or at least  $1 \times 10^{-6}$  smaller than the residual of the first Newton iteration. The relative stopping condition for the GMRES iteration is set to  $1 \times 10^{-3}$ .

##### A. Homogeneous crystallization in a supercooled liquid

The generation of polycrystal in a supercooled homogeneous liquid is studied in the test. The simulation is conducted on a periodic square domain  $\Omega = [0, 128]^2$  with a random initial condition  $\phi^0 = 0.07(1 + \delta_\phi)$ , where  $\|\delta_\phi\|_\infty \leq 1$ . The quench depth takes the value of  $\gamma = -0.025$ . Similar experiments were reported in, e.g., [5], [17].

We perform the simulation on a  $384 \times 384$  mesh with an initial time step size  $\Delta t^0 = 0.001$ . The time step size is then adaptively controlled by using (4). Contour plots of the density distribution are shown in Fig. 1, from which we observe: (1) from  $t = 0$  to around  $t = 5000$  the fluid quickly crystallizes under the supercooling; (2) from around  $t = 5000$  to around  $t = 20000$  the crystallized material gradually stabilizes as a solid lattice with periodic hexagonal pattern. We show in Fig. 2 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 nearly four magnitudes. The total free energy decreases

monotonically to the minimizer as the solution evolves to the steady-state.

##### B. Performance tuning

There are several parameters in the Schwarz preconditioner that have major impact on the performance of the NKS solver. In order to find the optimal choice of those parameters, we run the test on a  $1728 \times 1728$  mesh with 576 processor cores. To avoid the excessively small time scale at the early stage of the simulation, we take the numerical solution at  $t = 5$  as the initial condition. Then we fix the time step size as  $\Delta t = 0.1$  for the rest of the test. We examine the total numbers of Newton and GMRES iterations as well as the total compute time for the first 20 time steps.

First we examine the performance of the NKS solver when different subdomain solvers are utilized. To focus on the influence of the subdomain solvers, we limit the test to the classical AS preconditioner (9) and fix the overlap size to  $\delta = 4$ . The subdomain solvers that we try in the test include ILU factorizations with 2, 4 and 8 levels of fill-in and LU factorizations with and without a reuse strategy. The results are summarized in Table I, where the total numbers of Newton and GMRES iterations as well as the total compute time are provided (In the table “n/c” means no convergence of the GMRES solver). We observe from

Table I  
Performance results with different subdomain solvers.

	ILU(2)	ILU(4)	ILU(8)	LU	LU-reuse
#Newton	n/c	n/c	n/c	40	40
#GMRES	n/c	n/c	n/c	2196	2200
Time (s)	n/c	n/c	n/c	73.7	52.8

the table that GMRES doesn’t converge when ILU is used as the subdomain solver, even when the fill-in level is large. After replacing the ILU with the LU factorization, GMRES converges without any problem; the averaged number of GMRES iteration per Newton iteration is about 110. At each time step, since the Jacobian matrices of the Newton iteration have very similar structures, it is possible to save the compute time by only performing the LU factorization once and reusing the factorized matrices within the same time step. It can be seen from the table that with the reuse strategy, although the total number of GMRES iterations slight increases, the total number of Newton iterations stays unchanged and the total compute time is save by nearly 30%.

We then investigate the performance of the NKS solver when different types of the AS preconditioners are employed and when different overlaps are taken. Based on the previous test, we use the sparse LU factorization as the default subdomain solver and apply the reuse strategy throughout the test. We test the classical-AS (9), the left-RAS (10) and the right-RAS (11) preconditioners. For each case, the overlap size  $\delta$  is gradually increased from 1 to 5. The

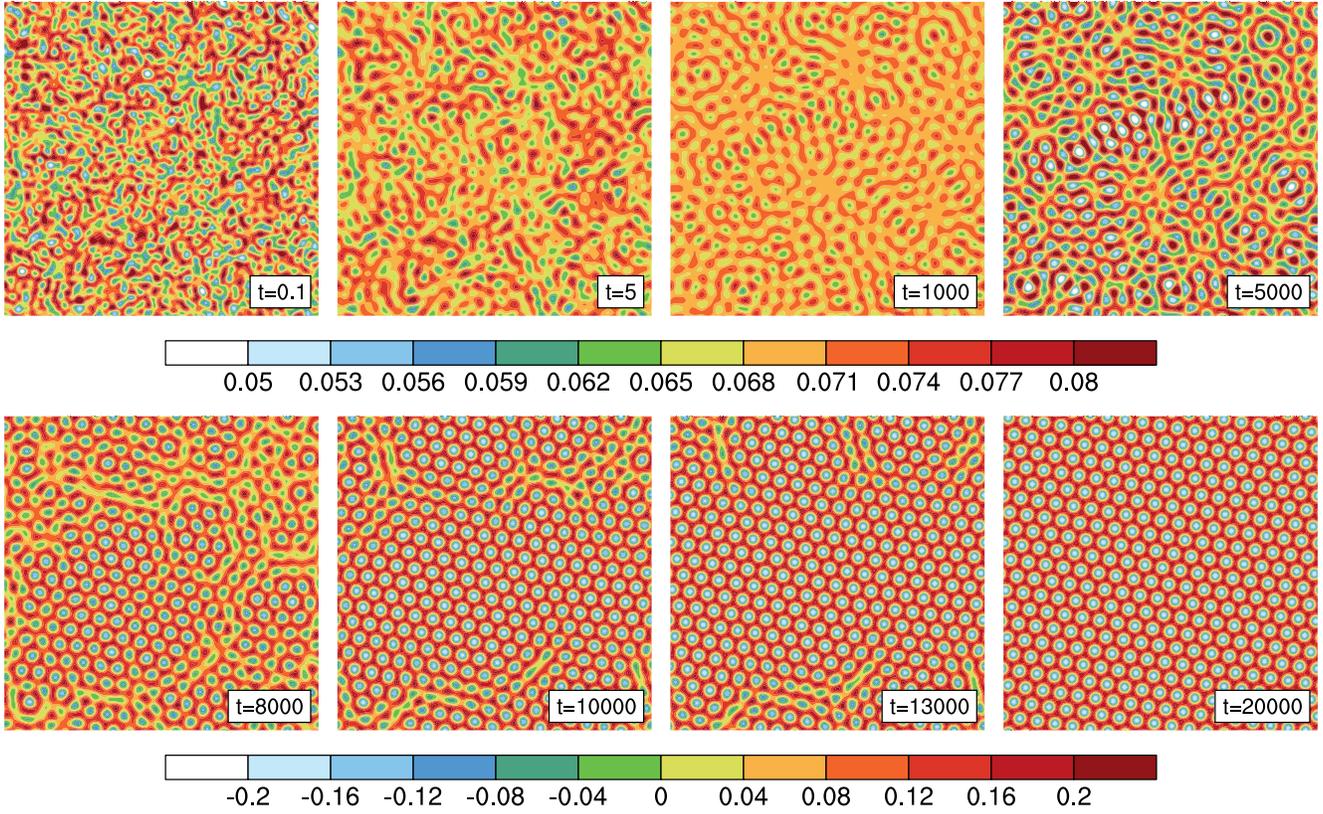


Figure 1. Results of homogeneous crystallization in a supercooled liquid. Shown in the pictures are contour plots of the density distribution.

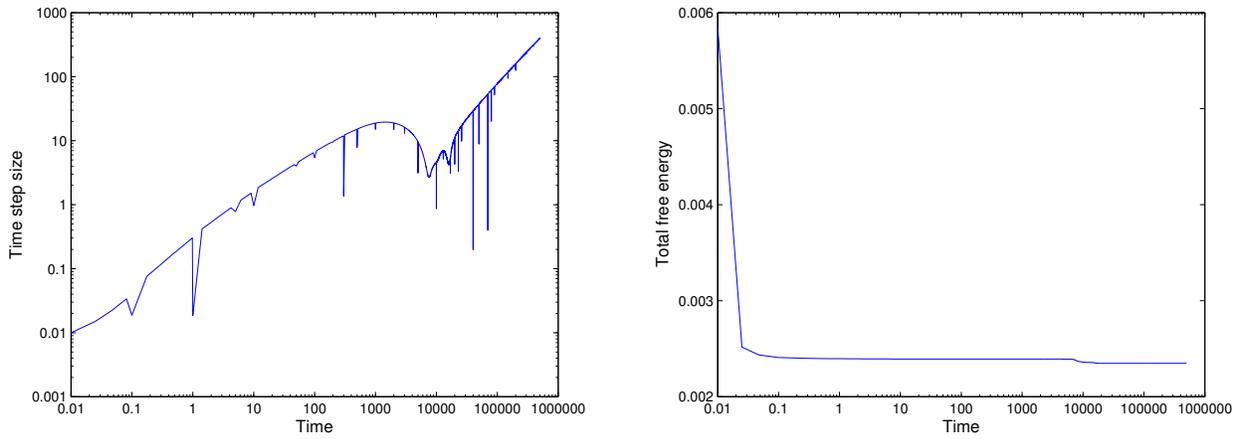


Figure 2. The evolution history of the time step size (left panel) and the total free energy (right panel).

Table II  
Performance results with different overlaps.

$\delta$	classical-AS					left-RAS					right-RAS				
	0, 1, 2	3	4	5	6	0, 1, 2	3	4	5	6	0, 1, 2	3	4	5	6
#Newton	n/c	40	40	40	40	n/c	40	40	40	40	n/c	40	40	40	40
#GMRES	n/c	2200	1000	740	600	n/c	6001	1434	900	682	n/c	1422	819	600	460
Time (s)	n/c	52.8	39.2	41.0	48.3	n/c	123.9	46.5	47.1	48.7	n/c	39.7	36.0	40.1	44.2

special case with zero overlap, which is equivalent to a block Jacobian preconditioner, is also examined in the test. The performance results are shown in Table II, where the total numbers of Newton and GMRES iterations as well as the total compute time are listed. We notice from the table that when the overlap size is smaller than the stencil width, i.e., when  $\delta < 3$ , no convergence result is obtained, no matter which type of the AS preconditioner is applied. This indicates that to precondition the PFC equation, it is crucial to include all information including the ghost points (i.e., halos) for subdomain problems; otherwise, the linear solver may not even converge. For  $\delta \geq 3$ , observations can be made from the table that: (1) the number of Newton iterations is always insensitive to  $\delta$  in the tests; (2) for a given type of the AS preconditioner, the number of GMRES iterations decreases as  $\delta$  becomes larger and the fastest compute time is obtained when  $\delta = 4$ ; (3) for a fixed  $\delta$ , among the three AS preconditioners, the left-RAS preconditioner performs the worst while the right-RAS preconditioner performs the best, in terms of both the number of GMRES iterations and the total compute time.

### C. Parallel scalability

Based on the observations from the above tests, we use the right-RAS preconditioner with a fixed overlap  $\delta = 4$  and employ the sparse LU factorization with the reuse strategy as the subdomain solver. The scalability tests are also performed on a fixed  $1728 \times 1728$  mesh for 20 time steps with  $\Delta t = 0.1$ , with the initial condition obtained from the numerical solution at  $t = 5$  in a previous simulation.

Provided in Figure 3 are the results on the numbers of Newton and GMRES iterations. From the figure we observe that when the number of processor cores becomes larger the total number of Newton iterations does not change while the total number of GMRES iterations slowly increases. Figure 4 shows the results on the total compute time and the parallel scalability. Despite the increase of the GMRES iterations, it can be seen from the figure that the total compute time decreases almost linearly, as the number of processor cores increases. We believe this is because only one LU factorization is performed within each time step due to the reuse strategy. The overall speedup from 144 to 2304 cores is around 16.7, which indicates an ideal parallel efficiency.

## V. CONCLUSIONS

In this paper, we present a highly scalable parallel solver for phase field crystal modeling. To conduct stable and physically meaningful simulations, the phase field crystal equation is discretized with a stabilized implicit finite difference method and integrated using an adaptive time step control strategy. A parallel Newton-Krylov-Schwarz algorithm is then applied to solve the nonlinear system arising at each time step. To obtain good parallel performance, we

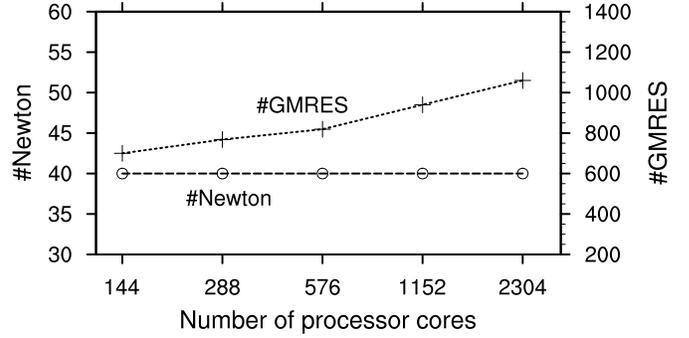


Figure 3. The total numbers of Newton and GMRES iterations in the scalability tests.

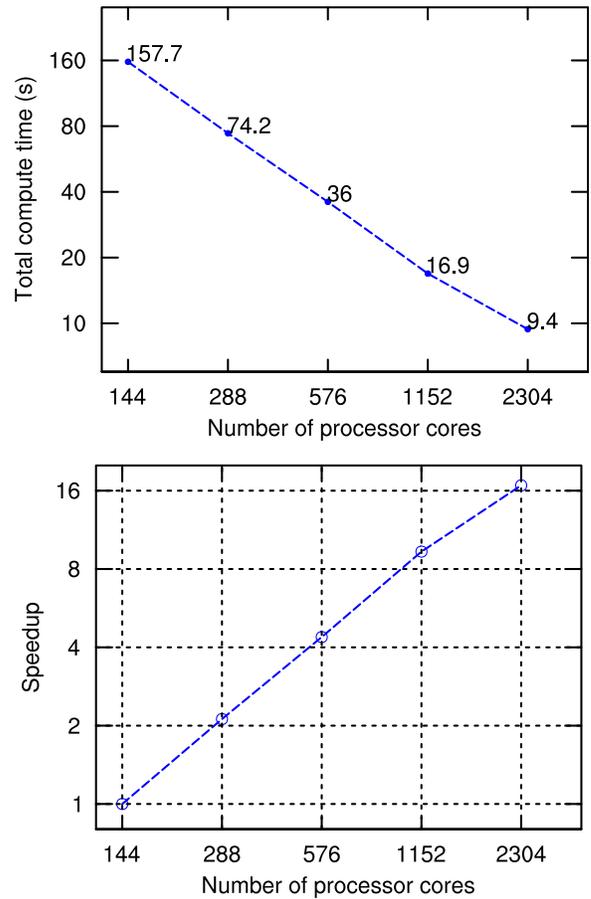


Figure 4. The total compute time and the relative scalability in the scalability tests.

study several important issues in the Schwarz preconditioner, including the type of the Schwarz preconditioner, the overlap size, the interface condition and the subdomain solver. Numerical results are provided to show that the proposed solver performs well on a supercomputer with thousands of processor cores.

#### ACKNOWLEDGMENT

This research was supported in part by DE-FC02-06ER25784. The work of CY also received supports from NSFC under 61170075 & 91130023 and from the 973 Program of China under 2011CB309701.

#### REFERENCES

- [1] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.3, 2012.
- [2] X.-C. Cai, M. Dryja, and M. Sarkis, "Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems," *SIAM J. Numer. Anal.*, vol. 41, pp. 1209–1231, 2003.
- [3] X.-C. Cai, W. D. Gropp, D. E. Keyes, and M. D. Tidriri, "Newton-Krylov-Schwarz methods in CFD," in *Proceedings of the International Workshop on the Navier-Stokes Equations, Notes in Numerical Fluid Mechanics*, R. Rannacher, Ed. Braunschweig: Vieweg Verlag, 1994, pp. 123–135.
- [4] X.-C. Cai and M. Sarkis, "A restricted additive Schwarz preconditioner for general sparse linear systems," *SIAM J. Sci. Comput.*, vol. 21, pp. 792–797, 1999.
- [5] M. Cheng and J. A. Warren, "An efficient algorithm for solving the phase field crystal model," *J. Comput. Phys.*, vol. 227, pp. 6241–6248, 2008.
- [6] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia: SIAM, 1996.
- [7] M. Dryja and O. B. Widlund, "Domain decomposition algorithms with small overlap," *SIAM J. Sci. Comput.*, vol. 15, pp. 604–620, 1994.
- [8] K. Elder and M. Grant, "Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals," *Phys. Rev. E*, vol. 70, p. 051605, 2004.
- [9] K. Elder, M. Katakowski, M. Haataja, and M. Grant, "Modeling elasticity in crystal growth," *Phys. Rev. Lett.*, vol. 88, p. 245701, 2002.
- [10] H. Emmerich, L. Gránásy, and H. Löwen, "Selected issues of phase-field crystal simulations," *Eur. Phys. J. Plus*, vol. 126, pp. 1–18, 2011.
- [11] H. Emmerich, H. Löwen, R. Wittkowski, T. Gruhn, G. I. Tóth, G. Tegze, and L. Gránásy, "Phase-field-crystal models for condensed matter dynamics on atomic length and diffusive time scales: an overview," *Adv. Phys.*, vol. 61, pp. 665–743, 2012.
- [12] D. J. Eyre, "Unconditionally gradient stable time marching the Cahn-Hilliard equation," *Mater. Res. Soc. Sympos. Proc.*, vol. 529, pp. 39–46, 1998.
- [13] —, "An unconditionally stable one-step scheme for gradient systems," 1998, (Unpublished article).
- [14] H. Gomez and X. Nogueira, "An unconditionally energy-stable method for the phase field crystal equation," *Comput. Methods Appl. Mech. Engrg.*, vol. 249-252, pp. 52–61, 2012.
- [15] W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. Smith, "Performance modeling and tuning of an unstructured mesh CFD application," in *Proc. Supercomputing 2000*. IEEE Computer Society, 2000.
- [16] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev. B*, vol. 136, pp. 864–871, 1964.
- [17] Z. Hu, S. M. Wise, C. Wang, and J. S. Lowengrub, "Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation," *J. Comput. Phys.*, vol. 228, pp. 5323–5339, 2009.
- [18] A. Jaatinen, C. V. Achim, K. R. Elder, and T. Ala-Nissila, "Thermodynamics of bcc metals in phase-field-crystal models," *Phys. Rev. E*, vol. 80, p. 031602, 2009.
- [19] W. A. Mulder and B. V. Leer, "Experiments with implicit upwind methods for the euler equations," *J. Comput. Phys.*, vol. 59, pp. 232–246, 1985.
- [20] N. Provatas, J. Dantzig, B. Athreya, P. Chan, P. Stefanovic, N. Goldenfeld, and K. Elder, "Using the phase-field crystal method in the multi-scale modeling of microstructure evolution," *J. Miner. Metals Mater. Soc.*, vol. 59, pp. 83–90, 2007.
- [21] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. New York, NY: Cambridge University Press, 1996.
- [22] S. Tang, R. Backofen, J. Wang, Y. Zhou, A. Voigt, and Y.-M. Yu, "Three-dimensional phase-field crystal modeling of fcc and bcc dendritic crystal growth," *J. Crystal Growth*, vol. 334, pp. 146–152, 2011.
- [23] G. Tegze, G. Bansel, G. I. Tóth, T. Pusztai, Z. Fan, and L. Gránásy, "Advanced operator splitting-based semi-implicit spectral method to solve the binary phase-field crystal equations with variable coefficients," *J. Comput. Phys.*, vol. 228, pp. 1612–1623, 2009.
- [24] M. R. Tonks, D. Gaston, P. C. Millett, D. Andrs, and P. Talbot, "An object-oriented finite element framework for multiphysics phase field simulations," *Comput. Mater. Sci.*, vol. 51, pp. 20–29, 2012.
- [25] A. Toselli and O. Widlund, *Domain Decomposition Methods – Algorithms and Theory*. Berlin: Springer-Verlag, 2005.
- [26] C. Wang and S. M. Wise, "An energy stable and convergent finite-difference scheme for the modified phase field crystal equation," *SIAM J. Numer. Anal.*, vol. 49, pp. 945–969, 2011.
- [27] L. Wang, J. Lee, M. Anitescu, A. E. Azab, L. C. McInnes, T. Munson, and B. Smith, "A differential variational inequality approach for the simulation of heterogeneous materials," in *Proc. SciDAC 2011*, 2011.

- [28] S. M. Wise, C. Wang, and J. S. Lowengrub, “An energy-stable and convergent finite-difference scheme for the phase field crystal equation,” *SIAM J. Numer. Anal.*, vol. 47, pp. 2269–2288, 2009.
- [29] O. Wodo and B. Ganapathysubramanian, “Computationally efficient solution to the Cahn-Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem,” *J. Comput. Phys.*, vol. 230, pp. 6037–6060, 2011.
- [30] C. Yang, X.-C. Cai, D. E. Keyes, and M. Pernice, “NKS method for the implicit solution of a coupled Allen-Cahn/Cahn-Hilliard system,” in *Proc. the 21st Intl. Conf. on Domain Decomposition Methods, Lecture Notes in Computational Science and Engineering*, 2012, to appear.
- [31] —, “Parallel domain decomposition methods for the 3D Cahn-Hilliard equation,” in *Proc. SciDAC 2011*, 2011.