



10-12-2017

Method of Lines Transpose: Energy Gradient Flows Using Direct Operator Inversion for Phase-Field Models

Matthew Causley

Hana Cho

Andrew Christlieb

Follow this and additional works at: https://digitalcommons.kettering.edu/mathematics_facultypubs



Part of the [Mathematics Commons](#)

METHOD OF LINES TRANSPOSE: ENERGY GRADIENT FLOWS USING DIRECT OPERATOR INVERSION FOR PHASE FIELD MODELS

MATTHEW CAUSLEY, HANA CHO, AND ANDREW CHRISTLIEB

ABSTRACT. In this work, we develop an $\mathcal{O}(N)$ implicit real space method in 1D and 2D for the Cahn Hilliard (CH) and vector Cahn Hilliard (VCH) equations, based on the Method Of Lines Transpose (MOL^T) formulation. This formulation results in a semi-discrete time stepping algorithm, which we prove is gradient stable in the H^{-1} norm.

The spatial discretization follows from dimensional splitting, and an $\mathcal{O}(N)$ matrix-free solver, which applies fast convolution to the modified Helmholtz equation. We propose a novel factorization technique, in which fourth order spatial derivatives are incorporated into the solver. The splitting error is included in the nonlinear fixed point iteration, resulting in a high order, logically Cartesian (line-by-line) update. Our method is fast, but not restricted to periodic boundaries like the fast Fourier transform (FFT).

The basic solver is implemented using the Backward Euler formulation, and we extend this to both backward difference (BDF) stencils, implicit Runge Kutta (SDIRK) and spectral deferred correction (SDC) frameworks to achieve high orders of temporal accuracy. We demonstrate with numerical results that the CH, and VCH equations maintain gradient stability in one and two spatial dimensions. We also explore time-adaptivity, so that metastable states and ripening events can be simulated both quickly and efficiently.

Keywords: Method of Lines Transpose, Rothe's method, Implicit Methods, Boundary Integral Methods, Alternating Direction Implicit Methods, ADI schemes, Unconditionally gradient schemes, Cahn-Hilliard, Functionalized Cahn-Hilliard.

1. INTRODUCTION

Many materials such as metals, ceramics, and polymers have physical properties which depend strongly on their microstructure. The morphology evolves during various physical processes, such as solidification, solid-state phase transformation, grain coarsening and grain growth. These processes are governed complex and nonlinear chemical reactions, and so the location and motion of phase interfaces typically evolve over long time scales. Consequently the phase-field method has become a consolidated tool for simulating microstructure evolution. See [8] for a survey on the subject.

One of the most well-known phase field models is the Cahn-Hilliard (CH) equation. Cahn and Hilliard [3] introduced it to explain the phase separation by predicting the interfacial free energy between two coexisting phases. The free energy is modeled using an order parameter u , and a reaction term $f(u)$. The reaction function is defined as the derivative of a double-well potential with two minima, $u = \pm 1$, which represent each phase of the composite material. The phases are separated by *transition layers*, which are of width $\mathcal{O}(\epsilon)$. The CH equation has been used extensively to model phenomena in material science; and several generalizations have been developed, including the vector Cahn-Hilliard (3) and functionalized Cahn-Hilliard equations.

In this paper, we will develop numerical schemes for phase field models based on the Method of Lines Transpose (MOL^T) framework. The MOL^T formulation, which is a practical approach to Rothe’s method [21], starts by discretizing a PDE in time, and then developing efficient approaches to solve the resulting set of coupled boundary value problems. The traditional formulation, proposed by Rothe [21], resulted in large matrices that were cumbersome; hence the method did not get a great deal of attention. In the late 2000s Rothe’s method received more attention, due to the development of spectral deferred correction methods and Krylov subspace techniques [17]. Soon, other fast summation techniques such as the fast multipole method [19] were employed based on potential theory and boundary integral techniques.

Even more recently, MOL^T algorithms for PDEs have been developed in the context of wave propagation [4, 7, 5], in which the resulting Helmholtz boundary value problem is solved with a dimensionally split algorithm, similar to alternating direction implicit (ADI) methods [12, 15]. Here the spatial operator is inverted one dimension at a time through fast convolution, in a logically Cartesian fashion. In [7], successive convolution was introduced, leading to an A-stable method of arbitrary order in time for hyperbolic problems. This led the current authors to revisit this idea in context of linear and nonlinear (second order) parabolic problems [6], and successive convolution was inserted into the resolvent expansion of pseudo differential operators to achieve L-stable real space solvers which scale as $\mathcal{O}(N)$ for N spatial points.

Here we turn our attention to parabolic problems of higher order, specifically the CH equation and its extensions. To this end, we highlight the relevant new features of our approach:

- i We combine our previous work on the MOL^T formulation [4, 7, 5, 6] with traditional implicit time stepping methods (BDF, SDIRK and IDC methods, with time adaptivity);
- ii we directly invert the linear part of the differential operator using the Green’s function, combined with fast $\mathcal{O}(N)$ convolution;
- iii we use dimensional splitting to avoid boundary integrals, and the need for global inversion;
- iv we directly include the splitting error in our formulation by explicitly incorporating it into the nonlinear iterations;
- v we remain competitive with explicit methods in terms of wall clock time needed to solve each fixed point iteration.

It is worth pointing out that the MOL^T formulation is really targeted at developing new paradigms for parallel multicore computing. The point to this paper, as well as our previous work, is to establish the theoretical and practical underpinnings necessary to address a wide class of PDEs. In particular, our task is to investigate the implementation for higher order PDEs. Parallel implementation will be the focus of our upcoming work in this area.

The rest of this is organized as follows. In Section 2, we present several models which are of interest in this work. In Section 3, we derive a first order scheme for CH equation in a basic 1D setting.

In Section 4, we modify the traditional time stepping scheme, to achieve higher orders of accuracy and present temporal refinement studies. In addition, we extend 1D schemes to multiple spatial dimension in Section 5. Finally, we describe an adaptive time stepping

strategy in Section 6 and present numerous numerical results including 1D, 2D solutions to the CH equations and the vector CH model in Section 7, followed by some concluding remarks.

2. MODELS

In this paper, we present the proposed Method of lines transpose (MOL^T) scheme for Cahn-Hilliard (CH) equation

$$u_t = -\Delta [\epsilon^2 \Delta u - f(u)], \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad (1)$$

with the appropriate initial and boundary conditions. The phase function $u \in H^1(\Omega)$ describes the volume fraction of one component of a binary mixture where $H^1(\Omega)$ is the standard Sobolev space. The reaction function $f(u)$ is the derivative of classical Ginzburg-Landau double-well potential $F(u) = \frac{1}{4}(u^2 - 1)^2$ whose local minima is at $u = \pm 1$. The small parameter $0 < \epsilon \ll 1$ is the width of the interfacial transition layer. In a 1D setting ($\Omega \equiv [a, b] \subset \mathbb{R}$), the Laplacian Δ in the above equations is replaced by ∂_{xx} .

The CH equation (1) describes the H^{-1} gradient flow of the CH free energy [3]

$$\mathcal{E}(u) = \int_{\Omega} \frac{\epsilon^2}{2} |\nabla u|^2 + F(u) d\mathbf{x}. \quad (2)$$

If we assume zero-flux boundary conditions, the CH free energy is dissipative

$$\frac{d}{dt} \mathcal{E}(u) = \left\langle u_t, \frac{\delta \mathcal{E}}{\delta u} \right\rangle_{L^2} \leq 0.$$

We emphasize that the physical property of energy stability is vital, and any consistent numerical scheme for the CH model must also possess this property. Many such schemes have been proposed, such as the operator splitting approach introduced by Eyre [14], the semi-implicit spectral deferred correction method by Shen, [22, 20], and even fully implicit schemes utilizing nonlinear solvers, such as the conjugate gradient method [9]. We shall use a linearly implicit fixed point method to solve the CH equation, which is stabilized by shifting the phase field about the background state $u = -1$.

We also consider vector version of CH (VCH) equation [9]. For $\mathbf{u} = (u_1, u_2)$,

$$\mathbf{u}_t = -\Delta [\epsilon^2 \Delta \mathbf{u} - \nabla_{\mathbf{u}} W(\mathbf{u})] \quad (3)$$

where the reaction term is the derivative of the potential function,

$$W(\mathbf{u}) = \prod_{i=1}^3 |\mathbf{u} - \mathbf{z}_i|^2, \quad \mathbf{z}_i : \text{cube roots of unity in the } (u_1, u_2) \text{ plane.} \quad (4)$$

The potential W is non-negative and its minimum values are attained at three vectors $\{\mathbf{z}_i\}$, so as to model a three-phase physical system. This model can be seen as the higher order volume preserving version of the vector-valued Ginzburg-Landau equation [2], which suggests a model for three phase boundary motion, such as the grain-boundary motion in alloys. The energy functional is a vector version of (2) such that

$$\mathcal{E}_{VCH}(\mathbf{u}) = \int_{\Omega} \frac{\epsilon^2}{2} |\nabla \mathbf{u}|^2 + W(\mathbf{u}) d\mathbf{x}. \quad (5)$$

3. FIRST ORDER SCHEME IN ONE SPATIAL DIMENSION

Before employing a time discretization to CH equation (1), we first introduce a new transformed variable $v = u + 1$. Since pure states $u = \pm 1$ dominates the solution during ripening, we see that equivalently $v = 0, 2$. This transformation is partially motivated by [11], in which the same approach is combined with operator splitting to produce a contractive operator. Similarly, in [9] the linearized variable $v \approx u + 1$ is used as a preconditioner for the full problem. Our work is distinct, however in that we are not solving the linearized equation, but instead the fully nonlinear equation

$$v_t = -\epsilon^2 v_{xxxx} + f(v-1)_{xx}, \quad f(v-1) = v^3 - 3v^2 + 2v, \quad (6)$$

which follows from the transformation $v = u + 1$ inserted into the CH equation (1), with $\Omega = (a, b)$. Similarly, the energy functional (2) becomes

$$\mathcal{E}(v) = \int_a^b \frac{\epsilon^2}{2} |v_x|^2 + F(v-1) dx, \quad F(v-1) = \frac{1}{4} v^4 - v^3 + v^2 \quad (7)$$

In Section 3.1, we will formulate our first-order MOL^T scheme using above equation (6), and will show that this transformation makes our scheme gradient stable in Section 3.1.1.

3.1. Semi-discrete scheme for 1D CH equation. We utilize the MOL^T by discretizing (6) in time as the Backward Euler (BE) scheme,

$$\frac{v^{n+1} - v^n}{\Delta t} = -\epsilon^2 \partial_{xxxx} v^{n+1} + 2\partial_{xx} v^{n+1} + \partial_{xx} \tilde{f}^{n+1}, \quad \tilde{f}(v) = v^3 - 3v^2, \quad (8)$$

where $v^n = u(x, t^n) + 1$ and $\Delta t = t^{n+1} - t^n$. This scheme (8) is first-order in time but still continuous in space. We note that the equation contains both linear and nonlinear implicit terms, and so an efficient iterative scheme is required to construct a fully implicit solution. We will suggest two nonlinear iteration schemes to solve the solution v^{n+1} in Section 3.1.2. We first prove that the semi-discrete solution of (8) is unconditionally gradient stable.

3.1.1. Energy stability. Our proof is similar to those which have appeared elsewhere [22, 14], in that we utilize the H^{-1} , under reasonable assumptions. However, we investigate the fully implicit scheme (8), which alters the linear part of the PDE. We first make useful observations.

- The operator $\Delta^{-1} : \{v \in L^2 \mid \int_{\Omega} v dx = 0\} \rightarrow H^2(\Omega)$ is defined [11] as

$$\Delta^{-1} v = h \iff \langle v, q \rangle = \langle \Delta h, q \rangle, \quad \forall q \in L^2(\Omega). \quad (9)$$

- For any $p, q \in L^2(\Omega)$, the following identity holds

$$\langle p, p - q \rangle = \frac{1}{2} (\|p\|_0^2 - \|q\|_0^2 + \|p - q\|_0^2) \geq \frac{1}{2} (\|p\|_0^2 - \|q\|_0^2), \quad (10)$$

where $\langle \cdot, \cdot \rangle$ is standard L^2 inner product and $\|\cdot\|_0$ is the L^2 norm.

We will consider the first-order scheme (8) in general spatial dimension ($\Omega \in \mathbb{R}^d, d \in \mathbb{N}$).

Lemma 3.1. *Under the following assumption,*

$$0 \leq v \leq 2, \quad v = u + 1$$

the fully-implicit scheme (8) satisfies the discrete energy law for any time step Δt :

$$\mathcal{E}(v^{n+1}) \leq \mathcal{E}(v^n), \quad \forall n \geq 0.$$

where v^n is approximation of $v(x, t^n) \equiv u(x, t^n) + 1$ of Cahn-Hilliard equation (1).

Proof. The solution v^{n+1} satisfies the following weak formulation in $H^2(\Omega)$,

$$\frac{1}{\Delta t} \langle v^{n+1} - v^n, \phi \rangle = -\epsilon^2 \langle \Delta v^{n+1}, \Delta \phi \rangle + 2 \langle v^{n+1}, \Delta \phi \rangle + \langle \tilde{f}^{n+1}, \Delta \phi \rangle, \quad (11)$$

for all $\phi \in H^2(\Omega)$. Utilizing (9), we choose the test function $\phi = \Delta^{-1}(v^{n+1} - v^n)$, so that

$$\begin{aligned} -\frac{1}{\Delta t} \|\nabla \cdot \Delta^{-1}(v^{n+1} - v^n)\|_0^2 &= \epsilon^2 \langle \nabla v^{n+1}, \nabla(v^{n+1} - v^n) \rangle \\ &\quad + 2 \langle v^{n+1}, v^{n+1} - v^n \rangle + \langle \tilde{f}^{n+1}, v^{n+1} - v^n \rangle, \end{aligned} \quad (12)$$

where integration by parts is employed. Next we apply the identity (10),

$$0 \geq \frac{\epsilon^2}{2} (\|\nabla v^{n+1}\|_0^2 - \|\nabla v^n\|_0^2) + (\|v^{n+1}\|_0^2 - \|v^n\|_0^2 + \|v^{n+1} - v^n\|_0^2) + \langle \tilde{f}^{n+1}, v^{n+1} - v^n \rangle. \quad (13)$$

We replace the last term with its Taylor expansion

$$\langle \tilde{F}^{n+1} - \tilde{F}^n, 1 \rangle = \langle \tilde{f}^{n+1}, v^{n+1} - v^n \rangle - \langle \frac{\tilde{f}'(\xi^{n+1})}{2} (v^{n+1} - v^n), v^{n+1} - v^n \rangle,$$

so that the inequality (13) becomes

$$\begin{aligned} \mathcal{E}(v^{n+1}) - \mathcal{E}(v^n) &= \frac{\epsilon^2}{2} (\|\nabla v^{n+1}\|_0^2 - \|\nabla v^n\|_0^2) + (\|v^{n+1}\|_0^2 - \|v^n\|_0^2) + \langle \tilde{F}^{n+1} - \tilde{F}^n, 1 \rangle \\ &\leq \left(\frac{\|\tilde{f}'\|_\infty}{2} - 1 \right) \|v^{n+1} - v^n\|_0^2 \leq 0, \end{aligned}$$

where $\|\cdot\|_\infty$ is L^∞ norm. The last inequality holds because $\tilde{f}' = 3v^2 - 6v \leq 0$ for $0 \leq v \leq 2$. Therefore the discrete energy is non-increasing, and the implicit scheme (8) is unconditionally gradient stable. \square

Remark 1. In [22, 20], Shen et.al. introduced a stabilizing term $S(u^{n+1} - u^n)$ for their semi-implicit scheme for the AC and CH equations. They choose the constant to satisfy $S \geq \frac{\max_{u \in \mathbb{R}} |\tilde{f}'(u)|}{2}$ with the truncated potential $-1 \leq u \leq 1$. Equivalently, we assume that $0 \leq v \leq 2$. However, we alter the linear part of the differential operator to ensure stability, rather than introducing an additional term. The distinction is that the transformation we employ modifies the linear part of the differential operator, rather than introducing an additional term, for stabilization.

3.1.2. *Factorization of the linear operator.* We will now propose a simple fixed point iterative solver for equation (8),

$$(I - (2\Delta t \partial_{xx} - \epsilon^2 \Delta t \partial_{xxxx})) v^{n+1, k+1} = v^n + \Delta t \partial_{xx} \tilde{f}^{n+1, k}, \quad (14)$$

where k indicates the iteration index. By lagging the the nonlinear term $\tilde{f}^{n+1, k}$, we can make the update explicit through formal *analytic* inversion of the fourth-order operator

$$v^{n+1, k+1} = (I - 2\Delta t \partial_{xx} + \epsilon^2 \Delta t \partial_{xxxx})^{-1} \left[v^n + \Delta t \partial_{xx} \tilde{f}^{n+1, k} \right]. \quad (15)$$

However, the Green's function of this fourth order operator contain both decaying and oscillatory components, which would not produce an efficient method. Instead we propose a novel

factorization technique, which produces two second order decaying (modified Helmholtz) differential operators, and in doing so leverage the fast computation methods previously employed in [6]. After some algebra, we rewrite (15) as

$$v^{n+1,k+1} = \left(I - \frac{\partial_{xx}}{\alpha_1^2} \right)^{-1} \left(I - \frac{\partial_{xx}}{\alpha_2^2} \right)^{-1} \left[v^n + \Delta t \partial_{xx} \tilde{f}^{n+1,k} \right], \quad (16)$$

where

$$\frac{1}{\alpha_1^2} = \Delta t + \sqrt{\Delta t^2 - \epsilon^2 \Delta t}, \quad \frac{1}{\alpha_2^2} = \Delta t - \sqrt{\Delta t^2 - \epsilon^2 \Delta t},$$

so that $\alpha_1, \alpha_2 \in \mathbb{R}$ when $\Delta t \geq \epsilon^2$. This implies convolution with the Green's functions $e^{-\alpha_1|x|}$ and $e^{-\alpha_2|x|}$. For $\Delta t < \epsilon^2$, these parameters become complex, and the Green's functions are oscillatory. To overcome this difficulty, we switch to an alternate factorization to solve v^{n+1} :

$$v^{n+1,k+1} = \left(I - \sqrt{\epsilon^2 \Delta t} \partial_{xx} \right)^{-2} \left[v^n + \Delta t \partial_{xx} \left(\tilde{f}^{n+1,k} - 2 \left(1 + \sqrt{\frac{\epsilon^2}{\Delta t}} \right) v^{n+1,k} \right) \right], \quad (17)$$

which is based on completing the square, and lagging linear terms which are incorporated into \tilde{f} . In this case, we convolve twice with the Green's function $e^{-\alpha|x|}$, where $\alpha = \frac{1}{\epsilon\sqrt{\Delta t}}$. This switching is summarized by the bifurcation diagram shown in Figure 1.

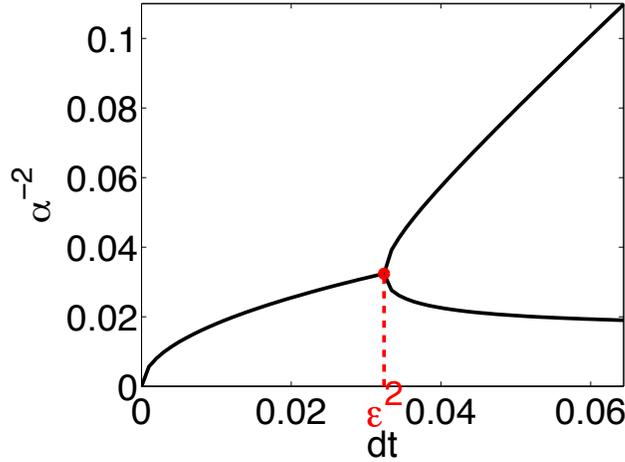


FIGURE 1. Bifurcation of the parameter $\frac{1}{\alpha^2}$ in the modified Helmholtz operator \mathcal{L} .

Since both equations (16) and (17) converge to (8) as the iterations $k \rightarrow \infty$, the overall scheme will be gradient stable. Thus, we invert the operator based on the parameters Δt and ϵ as follows:

$$\begin{aligned} \text{Factorization method } (\Delta t \geq \epsilon^2) : & \quad \text{Invert} \left(I - \frac{\partial_{xx}}{\alpha_1^2} \right) \left(I - \frac{\partial_{xx}}{\alpha_2^2} \right) \\ \text{Completed square method } (\Delta t < \epsilon^2) : & \quad \text{Invert} \left(I - \frac{\partial_{xx}}{\alpha^2} \right)^2. \end{aligned}$$

Remark 2. The second iterative scheme (17) can be equivalently viewed as introducing a term of the form $S\partial_{xx}(v^{n+1,k+1} - v^{n+1,k})$ into the first (16), with $S = 2\sqrt{\epsilon^2\Delta t}$. Numerical results confirm that both regimes, namely $\Delta t < \epsilon^2$ and $\Delta t > \epsilon^2$, exhibit gradient stability.

Both iteration schemes require two inversions of a modified Helmholtz operator in [6], hence

$$\mathcal{L} = I - \frac{\partial_{xx}}{\alpha^2}, \quad (18)$$

with the different values of α . As we have indicated above, formal inversion of this operator [6] yields:

$$\mathcal{L}^{-1}[v](x) = \underbrace{\mathcal{I}[v](x)}_{\text{Particular Solution}} + \underbrace{B_a e^{-\alpha(x-a)} + B_b e^{-\alpha(b-x)}}_{\text{Homogeneous Solution}}, \quad a \leq x \leq b,$$

where the particular solution is a convolution with the Green's function,

$$\mathcal{I}[v](x) = \frac{\alpha}{2} \int_a^b e^{-\alpha|x-x'|} v(x') dx', \quad (19)$$

and the coefficients B_a and B_b are determined by applying boundary conditions at $x = a, b$ in Section 3.2.

Remark 3. We observe that the Green's function will be convolved with terms of the form $\partial_{xx}f$, and so we will use the following result below

$$\partial_{xx} = \alpha^2(I - \mathcal{L}) \implies \mathcal{L}^{-1}[\partial_{xx}v] = \alpha^2(\mathcal{L}^{-1} - I)[v],$$

where we use the definition (18) of the modified Helmholtz operator \mathcal{L} .

3.2. Fully-discrete solution to 1D CH equation. It remains to discretize the convolution integral (18), and obtain a fully discrete algorithm. In our previous works [4, 7, 5, 6], we accomplish this with fast convolution; for convenience, we will summarize the fast algorithm here.

First, the particular solution (19) is split into $\mathcal{I}[v](x) = \mathcal{I}^L(x) + \mathcal{I}^R(x)$, where

$$\mathcal{I}^L(x) = \frac{\alpha}{2} \int_a^x e^{-\alpha(x-x')} v(x') dx', \quad \mathcal{I}^R(x) = \frac{\alpha}{2} \int_x^b e^{-\alpha(x'-x)} v(x') dx'.$$

The domain $\Omega \equiv [a, b]$ is partitioned into N subdomains $[x_{j-1}, x_j]$

$$a = x_0 < x_1 < \dots < x_N = b, \quad h_j = x_j - x_{j-1},$$

we then evaluate the convolution operator at each grid point through $\mathcal{I}_j \equiv \mathcal{I}[v](x_j) = \mathcal{I}_j^L + \mathcal{I}_j^R$. Since each integral satisfies exponential recursion [7, 5, 6],

$$\mathcal{I}_j^L = e^{-\alpha h_j} \mathcal{I}_{j-1}^L + \mathcal{J}_j^L, \quad \mathcal{J}_j^L = \frac{\alpha}{2} \int_{x_{j-1}}^{x_j} e^{-\alpha(x_j-x')} v(x') dx', \quad (20)$$

$$\mathcal{I}_j^R = e^{-\alpha h_{j+1}} \mathcal{I}_{j+1}^R + \mathcal{J}_j^R, \quad \mathcal{J}_j^R = \frac{\alpha}{2} \int_{x_j}^{x_{j+1}} e^{-\alpha(x'-x_j)} v(x') dx'. \quad (21)$$

Those recursive updates are still exact, and only require computing the "local" convolutions \mathcal{J}_j^L and \mathcal{J}_j^R . In [6], we derived a spatial quadrature of general order $M \geq 2$ for computing these local integrals. For example, the second-order accurate ($M = 2$) quadrature on a uniform grid ($h_j = \Delta x$) is given by

$$J_j^L \approx Pv(x_j) + Qv(x_{j-1}) + R(v(x_{j+1}) - 2v(x_j) + v(x_{j-1})) \quad (22)$$

$$J_j^R \approx Pv(x_j) + Qv(x_{j+1}) + R(v(x_{j+1}) - 2v(x_j) + v(x_{j-1})) \quad (23)$$

where defining $\nu = \alpha\Delta x$ and $d = e^{-\nu}$, the quadrature weights are given by

$$\begin{aligned} P &= 1 - \frac{1-d}{\nu}, \\ Q &= -d + \frac{1-d}{\nu}, \\ R &= \frac{1-d}{\nu^2} - \frac{1+d}{2\nu}. \end{aligned}$$

Likewise, for higher order methods ($M > 2$), the quadrature weights are pre-computed, so that the fast convolution algorithm is achieved as $\mathcal{O}(MN)$ per time step with user-defined order M in space. In every simulation in this work, we choose $M = 4$ or $M = 6$.

Next, the homogeneous solution (18) is used to enforce various boundary conditions. For example periodic boundary conditions lead to

$$v^n(a) = v^n(b), \quad v_x^n(a) = v_x^n(b), \quad \forall n \in \mathbb{N}.$$

and evaluation of (18) at $x = a, b$, produces a 2×2 system for the unknowns B_a and B_b :

$$\begin{aligned} \mathcal{L}^{-1}[v^n](a) = \mathcal{L}^{-1}[v^n](b) &\iff \mathcal{I}_0 + B_a + B_b\mu = \mathcal{I}_N + B_a\mu + B_b, \\ \mathcal{L}_x^{-1}[v^n](a) = \mathcal{L}_x^{-1}[v^n](b) &\iff \alpha(\mathcal{I}_0 - B_a + B_b\mu) = \alpha(-\mathcal{I}_N + B_a\mu + B_b). \end{aligned}$$

Solving this linear system yields

$$B_a = \frac{\mathcal{I}_N}{1-\mu}, \quad B_b = \frac{\mathcal{I}_0}{1-\mu}, \quad (24)$$

where $\mu = e^{-\alpha(b-a)}$ and $\mathcal{I}_0 = \mathcal{I}[v](a)$ and $\mathcal{I}_N = \mathcal{I}[v](b)$. Other boundary conditions (e.g. Neumann, Dirichlet) follow the analogous procedure that requires solving a simple 2×2 linear system. (cf. [5])

4. HIGHER ORDER SCHEMES

In this Section we formulate second and third order time accurate methods by combining the ideas of MOL^T formulation with backward difference formulas (BDF), singly diagonal implicit Runge-Kutta (SDIRK), and spectral deferred correction (SDC) methods. We will present several refinement studies to compare these approaches.

4.1. MOL^T with BDF. The BDF time stepping methods are one of the most commonly used implicit linear multi-step methods [16]. To achieve the desired order of accuracy, we discretize the time derivative using BDF formulas as follows:

$$\text{BDF2:} \quad \left(I + \frac{2}{3}\epsilon^2\Delta t\partial_{xxxx} \right) v^{n+1} = \frac{4}{3}v^n - \frac{1}{3}v^{n-1} + \frac{2}{3}\Delta t\partial_{xx}f^{n+1}, \quad (25)$$

$$\text{BDF3:} \quad \left(I + \frac{6}{11}\epsilon^2\Delta t\partial_{xxxx} \right) v^{n+1} = \frac{18}{11}v^n - \frac{9}{11}v^{n-1} + \frac{2}{11}v^{n-2} + \frac{6}{11}\Delta t\partial_{xx}f^{n+1}, \quad (26)$$

where we now require two and three previous time steps, respectively. Similar to the first-order scheme in Section 3.1, we define two fixed-point iterations, depending on the time step Δt . First, energy-stable factorization iteration which has a time step lower bound for real-valued Green's functions. Second, completed square version by adding $S(v^{n+1,k+1} - v^{n+1,k})$ term, which has a time step upper bound for energy stability. Specifically, the BDF 2 formulation (25) utilizes

$$\begin{aligned} \Delta t \geq \frac{3}{2}\epsilon^2 : \quad \mathcal{L} &= I - \frac{4}{3}\Delta t \partial_{xx} + \frac{2}{3}\epsilon^2 \Delta t \partial_{xxxx} = \left(I - \frac{\partial_{xx}}{\alpha_1^2} \right) \left(I - \frac{\partial_{xx}}{\alpha_2^2} \right), \\ \Delta t < \frac{3}{2}\epsilon^2 : \quad \mathcal{L} &= I - 2\sqrt{\frac{2}{3}\epsilon^2 \Delta t} \partial_{xx} + \frac{2}{3}\epsilon^2 \Delta t \partial_{xxxx} = \left(I - \frac{\partial_{xx}}{\alpha^2} \right)^2, \end{aligned}$$

where

$$\frac{1}{\alpha_1^2} = \frac{2}{3}\Delta t + \sqrt{\left(\frac{2}{3}\Delta t\right)^2 - \frac{2}{3}\epsilon^2 \Delta t}, \quad \frac{1}{\alpha_2^2} = \frac{2}{3}\Delta t - \sqrt{\left(\frac{2}{3}\Delta t\right)^2 - \frac{2}{3}\epsilon^2 \Delta t}, \quad \frac{1}{\alpha^2} = \sqrt{\frac{2}{3}\epsilon^2 \Delta t}.$$

A similar result is obtained for the BDF 3 method. The main advantage of BDF schemes is that the extension of first-order methods to higher-order is very straightforward. However, they also require initialization of the first few time steps. In next section, we will look at Ruge-Kutta (RK) methods, which do not have this requirement.

4.2. MOL^T with SDIRK. Singly implicit RK methods (SDIRK) are those which have the same diagonal elements in the Butcher Table, [1]. To obtain a P^{th} order method, RK methods, P intermediate steps are computed at each time step. The second order method, SDIRK2, is as follows

$$\begin{aligned} (I + \eta\epsilon^2 \Delta t \partial_{xxxx}) K_1 &= -\epsilon^2 \partial_{xxxx} v^n + \partial_{xx} f(v^n + \eta \Delta t K_1), \quad \eta = 1 - \frac{\sqrt{2}}{2}, \\ (I + \eta\epsilon^2 \Delta t \partial_{xxxx}) K_2 &= -\epsilon^2 \partial_{xxxx} (v^n + (1 - \eta) \Delta t K_1) + \partial_{xx} f(v^n + (1 - \eta) \Delta t K_1 + \eta \Delta t K_2) \\ v^{n+1} &= v^n + \Delta t \{(1 - \eta) K_1 + \eta K_2\}, \end{aligned}$$

where η is the root of the polynomial $\frac{1}{2} - 2\eta + \eta^2$, as derived from the order conditions [1]. Next, we construct two intermediate solution K_1 and K_2 , using nonlinear iterative schemes based on Section 4.1. First, if $\Delta t \geq \frac{\epsilon^2}{\eta}$,

$$\begin{aligned} K_1^{n+1,k+1} &= \mathcal{L}_1^{-1} \mathcal{L}_2^{-1} \left[-\epsilon^2 \partial_{xxxx} v^n + \partial_{xx} \left(\tilde{f} \left(v^n + \eta \Delta t K_1^{n+1,k} \right) + 2v^n \right) \right], \quad \mathcal{L}_i = I - \frac{\partial_{xx}}{\alpha_i^2} \\ K_2^{n+1,k+1} &= \mathcal{L}_1^{-1} \mathcal{L}_2^{-1} \left[-\epsilon^2 \partial_{xxxx} \left(v^n + (1 - \eta) \Delta t K_1^{n+1,k} \right) \right. \\ &\quad \left. + \partial_{xx} \left(\tilde{f} \left(v^n + (1 - \eta) \Delta t K_1^{n+1,k} + \eta \Delta t K_2^{n+1,k} \right) + 2(v^n + (1 - \eta) \Delta t K_1^{n+1,k}) \right) \right], \end{aligned}$$

where $\frac{1}{\alpha_1^2} = \eta\Delta t + \sqrt{(\eta\Delta t)^2 - \epsilon^2\eta\Delta t}$ and $\frac{1}{\alpha_2^2} = \eta\Delta t - \sqrt{(\eta\Delta t)^2 - \epsilon^2\eta\Delta t}$. If $\Delta t < \frac{\epsilon^2}{\eta}$,

$$\begin{aligned} K_1^{n+1,k+1} &= \mathcal{L}^{-2} \left[-\epsilon^2 \partial_{xxxx} v^n + \partial_{xx} \left(f \left(v^n + \eta\Delta t K_1^{n+1,k} \right) - 2\sqrt{\eta\epsilon^2\Delta t} K_1^{n+1,k} \right) \right], \\ K_2^{n+1,k+1} &= \mathcal{L}^{-2} \left[-\epsilon^2 \partial_{xxxx} \left(v^n + (1-\eta)\Delta t K_1^{n+1,k} \right) \right. \\ &\quad \left. + \partial_{xx} \left(f \left(v^n + (1-\eta)\Delta t K_1^{n+1,k} + \eta\Delta t K_2^{n+1,k} \right) - 2\sqrt{\eta\epsilon^2\Delta t} K_2^{n+1,k} \right) \right], \end{aligned}$$

where $\mathcal{L} \equiv I - \sqrt{\eta\epsilon^2\Delta t}\partial_{xx}$ is used. Below we will also make use of SDIRK3, which can be derived in a similar fashion.

Remark 4. *Similar to Remark 3, the fourth derivative can be calculated as follows,*

$$\partial_{xxxx} = \alpha_1^2 \alpha_2^2 (I - \mathcal{L}_1)(I - \mathcal{L}_2) \implies \mathcal{L}_1^{-1} \mathcal{L}_2^{-1} [\partial_{xxxx} v] = \alpha_1^2 \alpha_2^2 (\mathcal{L}_1^{-1} - I) (\mathcal{L}_2^{-1} - I) [v].$$

In Section 4.4 we will present refinement studies for both the BDF and SDIRK methods. We note that for a small enough time step Δt , both methods converge as expected. However for the corresponding third-order methods (BDF3 and SDIRK3), as Δt increases the order of convergence begins to plateau. To resolve this issue, we will suggest another higher order method.

4.3. MOL^T with SDC. The spectral deferred correction (SDC) methods are a class of time integrators [13]. First a prediction to the solution ("level 0") is computed using low order schemes (e.g. Backward Euler), and then the error (defect) is corrected at subsequent stages, using higher order integrators. Our SDC procedure follows the presentation in [10].

(1) **Prediction step (level [0]):** Subdivide the time interval $[0, T]$ with uniform Δt :

$$0 \equiv t^0 < t^1 < \dots < t^{N_t} \equiv T.$$

Compute $\{v_m^{[0]}(x)\}_{0 \leq m \leq N_t}$ via Backward Euler approximation in (8) for CH equation, which is first-order approximation to the exact solution $\{y_m(x)\}_{0 \leq m \leq N_t}$.

(2) **Correction step (level [1]):** Assume that $v^{(0)}(x, t)$ is a polynomial interpolant, approximating the exact solution $y(x, t)$ satisfying

$$v^{(0)}(x, t^m) \equiv v_m^{[0]}(x), \quad m = n+1, n, \dots, n+1-P,$$

where P will be specified later. The error equation is defined

$$e(x, t) = y(x, t) - v^{(0)}(x, t),$$

and the residual (or "defect") is

$$\gamma(x, t) = v_t^{(0)} - \mathcal{F}_{\text{CH}}(v^{(0)}), \quad \mathcal{F}_{\text{CH}}(v) = -\epsilon^2 \partial_{xxxx} v + \partial_{xx} f(v).$$

We take the derivative of the error equation with respect to t , and rewrite it using the residual definition,

$$\begin{aligned} e_t(x, t) &= y_t(x, t) - v_t^{(0)}(x, t) \equiv \mathcal{F}_{\text{CH}}(y(x, t)) - \mathcal{F}_{\text{CH}}(v^{(0)}(x, t)) - \gamma(x, t), \\ \iff e_t(x, t) + \gamma(x, t) &= \mathcal{F}_{\text{CH}}((e + v^{(0)})(x, t)) - \mathcal{F}_{\text{CH}}(v^{(0)}(x, t)), \\ \iff \frac{\partial}{\partial t} \left(v^{(1)}(x, t) - \int_0^t \mathcal{F}_{\text{CH}}(v^{(0)}(x, \tau)) d\tau \right) &= \mathcal{F}_{\text{CH}}(v^{(1)}(x, t)) - \mathcal{F}_{\text{CH}}(v^{(0)}(x, t)), \end{aligned}$$

where we assume that the initial condition $e(x, 0) = 0$, and $v^{(1)} = v^{(0)} + e$. Hence, our updating "level [1]" solution $\{v_m^{[1]}\}$ is found by approximating the above differential equation with the same method as "level 0". We apply the backward Euler scheme, thus

$$v_{n+1}^{[1]} - v_n^{[1]} = \Delta t \left(\mathcal{F}_{\text{CH}}(v_{n+1}^{[1]}) - \mathcal{F}_{\text{CH}}(v_{n+1}^{[0]}) \right) + \int_{t^n}^{t^{n+1}} \mathcal{F}_{\text{CH}}(v^{(0)}(x, \tau)) d\tau. \quad (27)$$

(3) **Correction step (level [j]):** The process is then iterated by generalizing (27),

$$v_{n+1}^{[j]} - \Delta t \mathcal{F}_{\text{CH}}(v_{n+1}^{[j]}) = v_n^{[j]} - \Delta t \mathcal{F}_{\text{CH}}(v_{n+1}^{[j-1]}) + \int_{t^n}^{t^{n+1}} \mathcal{F}_{\text{CH}}(v^{(j-1)}(x, \tau)) d\tau, \quad (28)$$

where the terms including updating solution $v^{[j]}$ have been collected on the left hand side, and the old solution $v^{[j-1]}$ are on the right hand side.

To complete SDC, we must consider an approximation of the integral in (28):

$$\int_{t^n}^{t^{n+1}} \mathcal{F}_{\text{CH}}(v^{(j-1)}(x, \tau)) d\tau = \begin{cases} \Delta t \sum_{i=0}^P \tilde{q}_i \mathcal{F}_{n+1-i}^{[j-1]}, & \text{if } n \geq P-1, \\ \Delta t \sum_{i=0}^P \tilde{q}_i \mathcal{F}_i^{[j-1]} & \text{if } n < P-1, \end{cases} \quad (29)$$

where $\mathcal{F}_n^{[j-1]} = \mathcal{F}_{\text{CH}}(v_n^{[j-1]})$, and \tilde{q}_i are quadrature weights (cf. [10]). Note that the number of terms in the sum (29) is $P+1$, where P is the order of polynomial interpolation $v^{(j-1)}$. The integral must be approximated with increasing accuracy as the level increases, so that $P \geq j$ at "level [j]". In Section 4.4, we will show that this order P affects the asymptotic region of stability of SDC3 method.

We now combine MOL^T scheme with these higher-order SDC methods. For instance, the second-order SDC scheme (SDC2) only requires one more correction (level [1]). If $\Delta t \geq \epsilon^2$,

$$v_{n+1, k+1}^{[1]} = \mathcal{L}_1^{-1} \mathcal{L}_2^{-1} \left[v_n^{[1]} + \Delta t \partial_{xx} \left(\tilde{f}_{n+1, k}^{[1]} - \frac{f_{n+1}^{[0]} - f_n^{[0]}}{2} \right) + \frac{\epsilon^2 \Delta t}{2} \partial_{xxxx} \left(v_{n+1}^{[0]} - v_n^{[0]} \right) \right] \quad (30)$$

where quadrature weights of (29) are $\tilde{q}_1 = \tilde{q}_2 = \frac{1}{2}$ ($P=1$: trapezoidal rule) and \mathcal{L}_i are same with the first-order scheme in Section 3.1. Similarly, if $\Delta t < \epsilon^2$,

$$v_{n+1, k+1}^{[1]} = \mathcal{L}^{-2} \left[v_n^{[1]} + \Delta t \partial_{xx} \left(f_{n+1, k}^{[1]} - 2\sqrt{\frac{\epsilon^2}{\Delta t}} v_{n+1, k}^{[1]} - \frac{f_{n+1}^{[0]} - f_n^{[0]}}{2} \right) + \frac{\epsilon^2 \Delta t}{2} \partial_{xxxx} \left(v_{n+1}^{[0]} - v_n^{[0]} \right) \right] \quad (31)$$

4.4. **Refinement studies for higher order methods.** We solve (1) with periodic boundary conditions, with the same initial condition used in [9],

$$u_0(x) = \cos(2x) + \frac{1}{100} e^{\cos(x + \frac{1}{10})}, \quad x \in [0, 2\pi]. \quad (32)$$

We integrate up to a final time T_{final} using each second-order method. Since an exact solution is not available, we perform time refinement using successive approximations, and the approximate error with step size Δt is $\|u_{\Delta t} - u_{\Delta t/2}\|_{\infty}$, measured in the maximum norm. Successive errors for each second-order method is presented in Table 1. The parameters used for all computations were:

$$\epsilon = 0.18, \quad \Delta x = \frac{2\pi}{512} \approx 0.0123, \quad T_{\text{final}} = 1, \quad N_{\text{tol}} = 10^{-12}, \quad (33)$$

where N_{tol} is a tolerance for fixed point iterations such that $\|v^{n+1,k+1} - v^{n+1,k}\|_{\infty} < N_{\text{tol}}$. We use 6th-order spatial quadrature, so that the dominant error was temporal.

TABLE 1. Refinement studies of second-order methods for 1D CH equation with periodic BC.

Δt	BDF2		SDIRK2		SDC2	
	L^{∞} error	order	L^{∞} error	order	L^{∞} error	order
0.0500	2.9454×10^{-5}	—	4.7787×10^{-6}	—	6.3763×10^{-5}	—
0.0250	7.2771×10^{-6}	2.0170	1.2050×10^{-6}	1.9876	1.5412×10^{-5}	2.0487
0.0125	1.9755×10^{-6}	1.8812	3.0081×10^{-7}	2.0021	3.7618×10^{-6}	2.0345
0.0063	5.3869×10^{-7}	1.8747	7.5076×10^{-8}	2.0024	9.2786×10^{-7}	2.0194
0.0031	1.4352×10^{-7}	1.9082	1.8732×10^{-8}	2.0029	2.3032×10^{-7}	2.0103

The factorization of the Helmholtz operators is switched based on the time step size, and is different for each method:

$$\text{BDF2: } \Delta t = \frac{3}{2}\epsilon^2 = 0.0486, \quad \text{SDIRK2: } \Delta t = \frac{\epsilon^2}{1 - \frac{\sqrt{2}}{2}} \approx 0.1106, \quad \text{SDC2: } \Delta t = \epsilon^2 = 0.0324.$$

Table 1 shows that all methods exhibit second order convergence. Moreover in Figure 2(a), we see that the total energy of each method decays during this time evolution.

In Figure 2(b), we plot the total iteration history of each scheme for the tolerance $N_{\text{tol}} = 10^{-12}$. The wall clock time (seconds) of BDF2, SDC2, SDIRK2 is 0.9725(s), 1.9003(s), and 2.6468(s), respectively, for a final simulation time $T_{\text{final}} = 1$; so the BDF2 method computationally the most efficient. We observe that SDIRK2 is the slowest, as a nonlinear solve is required for both stages K_1 and K_2 , at each time step.

Next, we compare the corresponding third-order schemes for the CH equation, with the same parameters in (33), and with the following considerations:

- **[BDF3]** Need two initial steps: SDIRK2 used. (Switch criterion: $\Delta t_{\text{switch}} = \frac{11}{6}\epsilon^2$)
- **[SDIRK3]** Need to compute K_1, K_2, K_3 at each update. ($\Delta t_{\text{switch}} = \frac{\epsilon^2}{\eta}, \eta \approx 0.4359$)

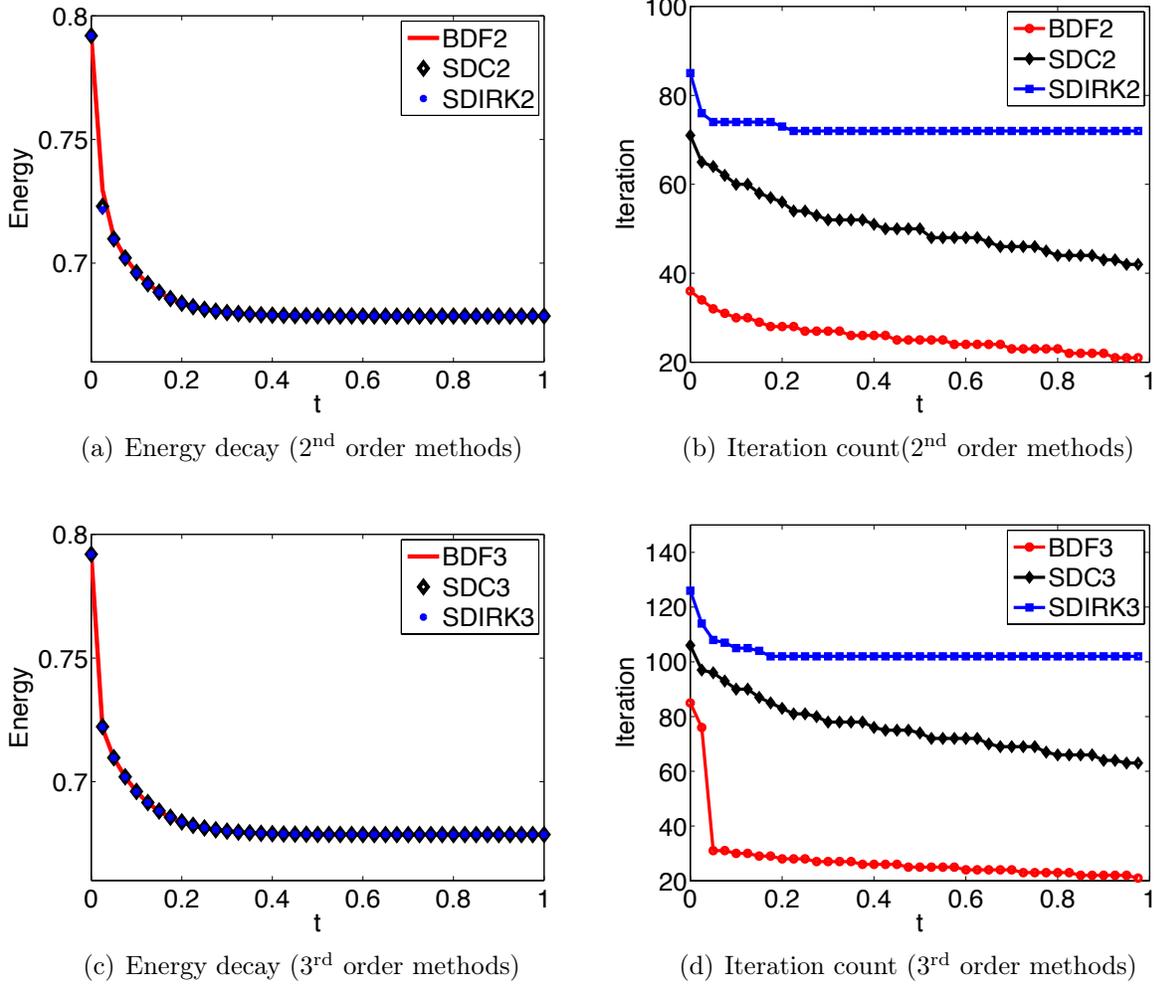


FIGURE 2. Energy descent and nonlinear iteration count history of all second- and third- order methods with fixed time step $\Delta t = 0.025$ and parameters (33).

- **[SDC3]** Compute from level [0] to level [2] with the quadrature order P in (29): First, choose $P = 2$ at level [2], then weights are $\tilde{q}_0 = \frac{5}{12}, \tilde{q}_1 = \frac{8}{12}, \tilde{q}_2 = -\frac{1}{12}$. ($\Delta t_{\text{switch}} = \epsilon^2$)

The energy and iteration history of third order schemes are shown in Figures 2(c) and 2(d). Again we see that the numerical energy decays. The beginning iteration number of BDF3 is larger than that of BDF2, since the initialization requires SDIRK2. But BDF3 is still the most efficient method of the third order schemes. The wall clock time (seconds) of BDF3, SDC3, SDIRK3 is 1.0355(s), 2.8539(s), and 3.6613(s), respectively, for a final simulation time $T_{\text{final}} = 1$.

The successive errors of third-order methods are shown in Table 2. Each method achieves third order convergence for small Δt ; but the order of convergence begins to plateau for larger time steps. To this end, SDC3 has a slight advantage. Indeed upon raising the polynomial

TABLE 2. Refinement studies of third-order methods for the 1D CH equation with periodic BC.

Δt	BDF3		SDIRK3		SDC3	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
0.0500	1.6001×10^{-5}	—	1.7259×10^{-6}	—	1.2669×10^{-5}	—
0.0250	5.2832×10^{-6}	1.5987	3.6118×10^{-7}	2.2566	2.1002×10^{-6}	2.5927
0.0125	1.1756×10^{-6}	2.1680	6.2823×10^{-8}	2.5234	3.1123×10^{-7}	2.7545
0.0063	1.9314×10^{-7}	2.6057	9.6163×10^{-9}	2.7077	4.2615×10^{-8}	2.8686
0.0031	2.6506×10^{-8}	2.8652	1.3011×10^{-9}	2.8858	5.4185×10^{-9}	2.9754
0.0016	3.2519×10^{-9}	3.0270	1.7447×10^{-10}	2.8987	6.6924×10^{-10}	3.0173

order (29) from $P = 2$ to $P = 3$, which requires one more correction at each time step, we obtain third order accuracy even for large time steps, as shown in Table 3.

TABLE 3. Refinement study of SDC3 ($P = 3$) for the 1D CH equation.

Δt	SDC3 ($P = 3$)	
	L^∞ error	order
0.0500	9.9462×10^{-6}	—
0.0250	1.3061×10^{-6}	2.9289
0.0125	1.4142×10^{-7}	3.2072
0.0063	1.6155×10^{-8}	3.1299
0.0031	1.7851×10^{-9}	3.1779
0.0016	2.2767×10^{-10}	2.9710

5. MULTIPLE SPATIAL DIMENSION

We now extend the 1D solver to multiple spatial dimensions via dimensional splitting [7, 5, 6]. We first write the 2D modified Helmholtz operator as

$$\mathcal{L} = I - \frac{\Delta}{\alpha^2} = \left(I - \frac{\partial_{xx}}{\alpha^2} \right) \left(I - \frac{\partial_{yy}}{\alpha^2} \right) - \frac{\partial_{xx}\partial_{yy}}{\alpha^4} \equiv \mathcal{L}_x \mathcal{L}_y - \frac{\partial_{xx}\partial_{yy}}{\alpha^4}, \quad (34)$$

where \mathcal{L}_x and \mathcal{L}_y are univariate operators, and so the fourth order term represents a splitting error. Now \mathcal{L}_x^{-1} is applied for fixed y , and vice versa for \mathcal{L}_y^{-1} in a line-by-line fashion, similar to alternating direction implicit (ADI) type method [12, 15]. Our key observation is that if we include the splitting error term from equation (34) in the fixed point iteration, then we can simultaneously solve the nonlinear problem, and correct the splitting error. More detailed analysis of the 2D CH equation (1) and 2D CH vector equations (4) will be presented below.

5.1. Semi-discrete solution to 2D CH equation. We now construct the semi-discrete scheme for the 2D CH equation, using the Backward Euler method. The analogous higher order schemes follow accordingly. Starting from the 1D scheme (17), we replace ∂_{xx} with the

Laplacian operator $\Delta = \partial_{xx} + \partial_{yy}$. If $\Delta t < \epsilon^2$, we complete the square of the operator, and have

$$\left(I - \frac{\Delta}{\alpha^2}\right)^2 [v^{n+1,k+1}] = v^n + \Delta t \Delta \left(f^{n+1,k} - 2\sqrt{\frac{\epsilon^2}{\Delta t}} v^{n+1,k}\right), \quad (35)$$

where $\frac{1}{\alpha^2} = \sqrt{\epsilon^2 \Delta t}$. Plugging the identity in (34), by lagging the mixed derivative term along with the nonlinear term,

$$(\mathcal{L}_x \mathcal{L}_y)^2 [v^{n+1,k+1}] = v^n + \Delta t \Delta \left(f^{n+1,k} - 2\sqrt{\frac{\epsilon^2}{\Delta t}} v^{n+1,k}\right) + \left(2\frac{\partial_{xx}\partial_{yy}}{\alpha^4} - \left(\frac{\partial_{xx}\partial_{yy}}{\alpha^4}\right)^2\right) v^{n+1,k} \quad (36)$$

Note that Laplacian operator and mixed derivative can be replaced as follows:

$$\Delta = \alpha^2 \left(I - \mathcal{L}_x \mathcal{L}_y + \frac{\partial_{xx}\partial_{yy}}{\alpha^4}\right), \quad \frac{\partial_{xx}\partial_{yy}}{\alpha^4} = (\mathcal{L}_x - I)(\mathcal{L}_y - I).$$

Now we formally invert both operators to the right hand side of (36),

$$v^{n+1,k+1} = (\mathcal{L}_x \mathcal{L}_y)^{-2} [v^n] + (\mathcal{L}_x \mathcal{L}_y)^{-1} \mathcal{C}_1 [\alpha^2 \Delta t (f^{n+1,k} - 2\sqrt{\frac{\epsilon^2}{\Delta t}} v^{n+1,k})] + \mathcal{C}_2 [v^{n+1,k}] \quad (37)$$

where

$$\mathcal{C}_1 = (\mathcal{L}_x \mathcal{L}_y)^{-1} - I + \mathcal{D}_x \mathcal{D}_y, \quad \mathcal{D}_\gamma = I - \mathcal{L}_\gamma^{-1}, (\gamma = \{x, y\}) \quad (38)$$

$$\mathcal{C}_2 = 2\mathcal{D}_x \mathcal{D}_y - (\mathcal{D}_x \mathcal{D}_y)^2. \quad (39)$$

As shown, every mixed-derivative splitting error term can be controlled by applying \mathcal{D}_γ , ($\gamma = \{x, y\}$) operators, which can be constructed in a line-by-line fashion. We emphasize that this allows us to remove splitting error, which is $O(\frac{1}{\alpha^4}) = O(\Delta t)$. Similar treatment of higher order BDF, SDIRK and SDC methods shows that the corresponding splitting errors can also be removed in this manner. Also, the fully discrete scheme follows from line-by-line spatial discretization, as outlined in Section 3.2.

We now consider the standard benchmark initial states in the 2D setting [9, 23], to confirm the temporal order of accuracy. The initial condition is

$$u_0(x, y) = 2e^{(\sin(x)+\sin(y)-2)} + 2.2e^{(-\sin(x)-\sin(y)-2)} - 1, \quad (x, y) \in [0, 2\pi]^2, \quad (40)$$

with the periodic boundary conditions, and the following parameters

$$\epsilon = 0.18, \quad \Delta x = \frac{2\pi}{128} \approx 0.0491, \quad T_{\text{final}} = 1 \quad (0 \leq t \leq T_{\text{final}}), \quad N_{\text{tol}} = 10^{-6}. \quad (41)$$

We use a 4th-order spatial quadrature to ensure that temporal error is dominant.

The temporal refinement study of each second-order scheme for the 2D CH equation is shown in Table 4. We observe the expected second-order of convergence for all three methods.

We also confirm the total energy, shown in Figure 3(a), is non-increasing during time evolution. Based on the iteration count, we see that BDF2 is the most efficient scheme, as was the case in 1D.

TABLE 4. Refinement studies of second-order methods for the 2D CH equation.

Δt	BDF2		SDIRK2		SDC2	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
0.1000	3.7891×10^{-3}	—	1.0250×10^{-3}	—	2.3334×10^{-3}	—
0.0500	8.2626×10^{-4}	2.1972	2.6950×10^{-4}	1.9272	5.0570×10^{-4}	2.2061
0.0250	2.0319×10^{-4}	2.0238	6.9796×10^{-5}	1.9491	1.1000×10^{-4}	2.2007
0.0125	4.6909×10^{-5}	2.1149	1.7809×10^{-5}	1.9706	2.8312×10^{-5}	1.9581

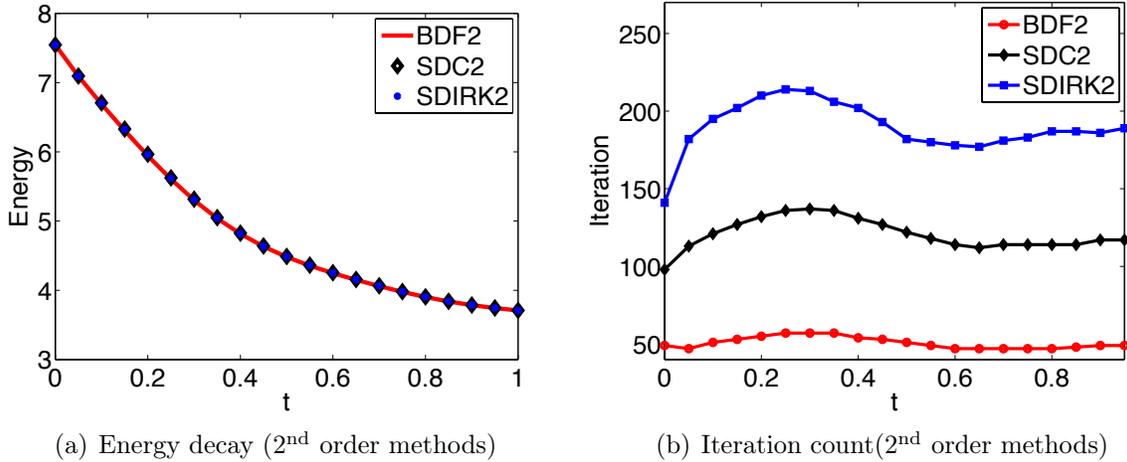


FIGURE 3. Energy descent and nonlinear iteration count history of all second-order 2D methods with fixed time step $\Delta t = 0.05$ and with parameters (41)

In principle, our method extends to even higher orders of accuracy in both time and space, as well as 3 spatial dimensions, since the splitting error technique in (38) can remove additional splitting errors.

5.2. Semi-discrete solution to 2D vector CH equation. We now extend our algorithm to vector CH (VCH) model (3). This model consists of two coupled variables u_1 and u_2 with local dynamic $\nabla_{\mathbf{u}}W$, comprised of partial derivatives of 6th- order polynomials $W(u_1, u_2)$ defined in (4).

Before employing the MOL^T formulation to this system, we translate the vector (u_1, u_2) about its equilibrium state, which is based on the functional $W(u_1, u_2)$,

$$(u_1^*, u_2^*) = \left\{ (\cos(\theta_i), \sin(\theta_i)) \mid \theta_1 = 0, \theta_2 = \frac{2\pi}{3}, \theta = -\frac{2\pi}{3} \right\} \equiv \left\{ (1, 0), \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right), \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right) \right\}$$

which are cube roots of unity in (u_1, u_2) plane. A straightforward calculation yields the Jacobian of the potential $\nabla_{\mathbf{u}}W$ evaluated at these points

$$\mathcal{J}_{\nabla_{\mathbf{u}}W}(u_1^*, u_2^*) \equiv \begin{bmatrix} \frac{\partial^2 W}{\partial u_1^2} & \frac{\partial^2 W}{\partial u_1 \partial u_2} \\ \frac{\partial^2 W}{\partial u_2 \partial u_1} & \frac{\partial^2 W}{\partial u_2^2} \end{bmatrix}_{(u_1^*, u_2^*)} = \begin{bmatrix} 18 & 0 \\ 0 & 18 \end{bmatrix}, \quad (42)$$

hence all points are stable equilibrium solutions [23]. We subtract $\mathbf{u} = (u_1, u_2)$ of the background state $\mathbf{z}_3 \equiv (-\frac{1}{2}, -\frac{\sqrt{3}}{2})$, and introduce the new transformed vector $\mathbf{v} = (v_1, v_2) = \mathbf{u} - \mathbf{z}_3$ into the original system (3),

$$\mathbf{v}_t = -\epsilon^2 \Delta^2 \mathbf{v} + \Delta \nabla_{\mathbf{v}} W(\mathbf{v} + \mathbf{z}_3) \equiv -\epsilon^2 \Delta^2 \mathbf{v} + \Delta \left(\nabla_{\mathbf{v}} \tilde{W}(\mathbf{v}) + 18\mathbf{v} \right) \quad (43)$$

where $\tilde{W}_{v_1}(\mathbf{v}) := W_{v_1}(\mathbf{v} + \mathbf{z}_3) - 18v_1$ and $\tilde{W}_{v_2}(\mathbf{v}) := W_{v_2}(\mathbf{v} + \mathbf{z}_3) - 18v_2$.

The backward Euler scheme applied to the transformed system (43) results in

$$(I - 18\Delta t \Delta + \epsilon^2 \Delta t \Delta^2) \mathbf{v}^{n+1, k+1} = \mathbf{v}^n + \Delta \nabla_{\mathbf{v}} \tilde{W}^{n+1, k} \quad (44)$$

where k denotes the iteration index. We again introduce two factorizations of the left-hand side operator:

$$\Delta t \geq \frac{\epsilon^2}{81} : I - 18\Delta t \Delta + \epsilon^2 \Delta t \Delta^2 = \left(I - \frac{\Delta}{\alpha_1^2} \right) \left(I - \frac{\Delta}{\alpha_2^2} \right), \quad (45)$$

$$\Delta t < \frac{\epsilon^2}{81} : I - 2\sqrt{\epsilon^2 \Delta t \Delta} + \epsilon^2 \Delta t \Delta^2 = \left(I - \frac{\Delta}{\alpha^2} \right)^2, \quad (46)$$

where

$$\frac{1}{\alpha_i^2} = 9\Delta t \pm \sqrt{81\Delta t^2 - \epsilon^2 \Delta t}, \quad \frac{1}{\alpha^2} = \sqrt{\epsilon^2 \Delta t}.$$

Inversion of the above operators follows from the same strategy as in Section 5.1. As expected, the stabilized fixed point iteration (45) permits larger time steps without limiting the foregoing energy stability property (5).

6. ADAPTIVE TIME STEPPING

For phase-field models, adaptive time stepping is a crucial feature for an efficient and accurate numerical solution. For instance, the solution of CH equation (1) evolves on various time scales. During *spinodal* evolution, transition layers are developed in $O(1)$ time. Subsequently, they slowly evolve and merge on a longer time scale, $O(e^{C/\epsilon})$ for 1D, which is called *ripening* process. Simulating these phenomena with a fixed time stepping necessarily becomes inefficient, and so we must incorporate adaptive time into our above MOL^T schemes.

The adaptive time step size control is based on the Local Truncation Error (LTE) η_e at each time level $t = t^n$. The LTE can be approximated by $\eta_e \approx \eta = \|u^* - u^n\|_\infty$, where u^* is an explicit predictor solution, typically using the Forward Euler (FE) or Adams Bashforth (AB) schemes [9]. On the other hand, Richardson extrapolation (known as step-doubling) or embedded Runge-Kutta pairs can also be used [10]. Below we will compare LTE estimates based on Richardson extrapolation, as well BDF2 and SDC2 predictors. The time step size selection criteria presented in [9] will be used, which is summarized in algorithm 1 below.

In practice, this procedure leads to small time steps during spinodal evolution, or at the ripening event, to maintain a consistent LTE. On the other hand, during slow coarsening (metastable states), small time steps are unnecessary and so Δt is increased within the upper bound for fixed-point iteration count.

Algorithm 1 Adaptive time step-size control

- (1) Starting at $t = t^n$, approximate the local truncation error. For Richardson extrapolation, the solution is estimated twice: once with a full step Δt (denoted by $u_{\Delta t}^{n+1}$), and again with two half steps ($u_{\Delta t/2}^{n+1}$). The difference between the two numerical approximations gives an estimate for the LTE of u^{n+1}

$$\text{(Richardson extrapolation)} \quad \eta_e \approx \eta := \frac{1}{2} \|u_{\Delta t}^{n+1} - u_{\Delta t/2}^{n+1}\|_{\infty}.$$

- (2) Define a tolerance σ_{tol} for the above LTE. If the accuracy criterion is not met ($\eta > \sigma_{\text{tol}}$), then the time step is reduced. If the desired accuracy is achieved, then we test for the following criteria,

- (I) $\eta \leq \sigma_{\text{tol}}$:

$$\begin{aligned} \frac{N_{\text{it}}}{N_{\text{max it}}} < 0.7 : \quad \Delta t^{n+1} &= \Delta t^n \cdot \min \left(\theta \sqrt{\frac{\sigma_{\text{tol}}}{\eta}}, \gamma \right), \quad \theta = 0.8, \quad \gamma = 1.3 > 1 \\ 0.7 \leq \frac{N_{\text{it}}}{N_{\text{max it}}} < 1 : \quad \Delta t^{n+1} &= \Delta t^n \cdot \min \left(\theta \sqrt{\frac{\sigma_{\text{tol}}}{\eta}}, 1 \right), \\ \frac{N_{\text{it}}}{N_{\text{max it}}} \geq 1 : \quad \text{Step fails.} \quad \text{Reduce time step} \quad \Delta t^n &= \Delta t^n \cdot \frac{1}{\gamma} \end{aligned}$$

- (II) $\eta > \sigma_{\text{tol}}$: Step fails. Reduce time step

$$\begin{aligned} \frac{\eta}{\sigma_{\text{tol}}} > 2 : \quad \Delta t^n &= \Delta t^n \cdot \frac{1}{\gamma}, \\ \frac{\eta}{\sigma_{\text{tol}}} \leq 2 : \quad \Delta t^n &= \Delta t^n \cdot \theta \sqrt{\frac{\sigma_{\text{tol}}}{\eta}} \end{aligned}$$

where γ and θ are the same safety factors defined in [9].

7. NUMERICAL RESULTS

In this section we present adaptive time stepping results using the previously developed MOL^T schemes.

7.1. 1D CH model. We first solve the 1D CH equation (1) with a stiff initial condition (32) ($\epsilon = 0.18$). The second perturbation term of this initial state creates two intervals, $u = -1$ and $u = +1$, which are asymmetric, so that a finite number of transition layers are formed during spinodal evolution. After a long ripening process, such layers are eventually absorbed into one region, at the so-called ripening time [9]. The aim of this simulation is to accurately capture all time scales using both fixed and adaptive time stepping strategies. (The spatial mesh size is fixed at $\Delta x = \frac{2\pi}{128} \approx 0.05$.)

In the first experiment, we implement our various time stepping schemes, with small fixed time step ($\Delta t = 0.01$). The ripening time T_r is defined as that for which the midpoint value $u(\pi, t)$ changes from positive to negative. The fixed point iteration has residual tolerance 10^{-11} at each step, and the ripening times are presented in Table 5. Our results agree well with the reference time $T_r = 8318.63$ in [9].

TABLE 5. Ripening time of 1D CH equation with small fixed time ($\Delta t = 0.01$)

	Ripening time
BE	8317.81
BDF2	8318.70
BDF3	8318.74
SDC2	8318.99
SDC3	8318.84

We also compare the ripening time using several schemes, with larger fixed time steps in Table 6. Among the three methods, BDF2 is the most efficient, and provides better estimates

TABLE 6. Ripening time of 1D CH equation with larger fixed time step sizes.

	time step	Ripening time	Times(s)
BE	10	8250.00	313.38
	1	8296.00	351.52
	0.05	8311.85	1312.16
SDC2	10	9050.00	717.26
	1	8582.00	692.65
	0.05	8319.80	2779.61
BDF2	10	8290.00	231.40
	1	8303.00	281.56
	0.05	8315.75	1288.68

of the true ripening time, even for larger Δt . In the most extreme instance of $\Delta t = 10$, we note that the first-order scheme (BE) predicts ripening too soon, and that SDC2 is too late; but BDF2 is still fairly accurate. However, to capture the ripening moment accurately, we still require small fixed time steps ($\Delta t \leq 0.05$), which is too expensive for long time simulations.

TABLE 7. Ripening time of 1D CH equation with adaptive time step size.

	δ_{tol}	Ripening time	Times(s)
BE (Richardson)	10^{-3}	8292.54	224.03
	10^{-4}	8276.08	226.06
	10^{-5}	8308.23	235.44
BE-SDC2	10^{-3}	8311.08	233.38
	10^{-4}	8311.47	239.58
	10^{-5}	8312.91	226.43
BE-BDF2	10^{-3}	8320.03	184.71
	10^{-4}	8319.91	196.82
	10^{-5}	8317.87	208.18

Thus, we consider adaptive time stepping for the same problem. As indicated in Section 6, the local truncation error (LTE) can be approximated with Richardson extrapolation, or a higher order solver such as BDF2 or SDC2.

We implement three methods with the same fixed point residual tolerance 10^{-11} , $N_{\max it} = 600$ in Algorithm 1, but with various error tolerance δ_{tol} . The performance of the time-adaptive scheme is shown in Table 7. We see that time adaptivity is superior to fixed time stepping both in terms of accuracy and time to solution. Additionally, the BDF2 method is the most efficient predictor.

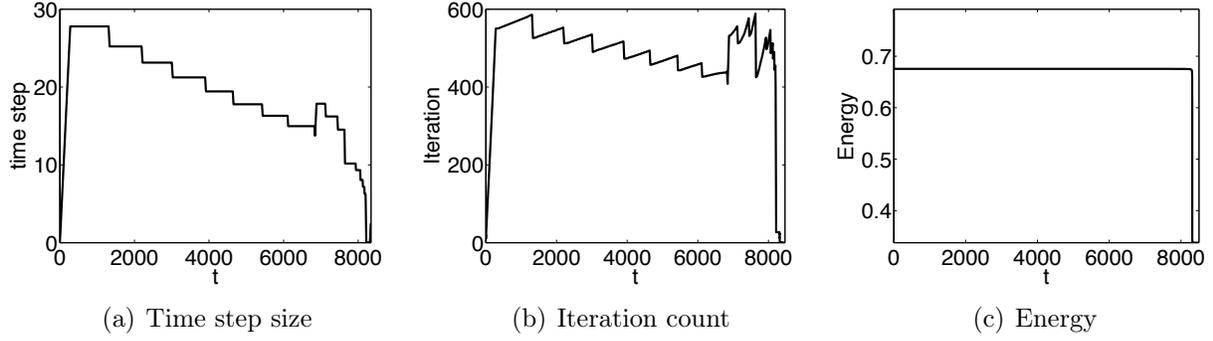


FIGURE 4. Adaptive time stepping with the BE-BDF2 strategy.

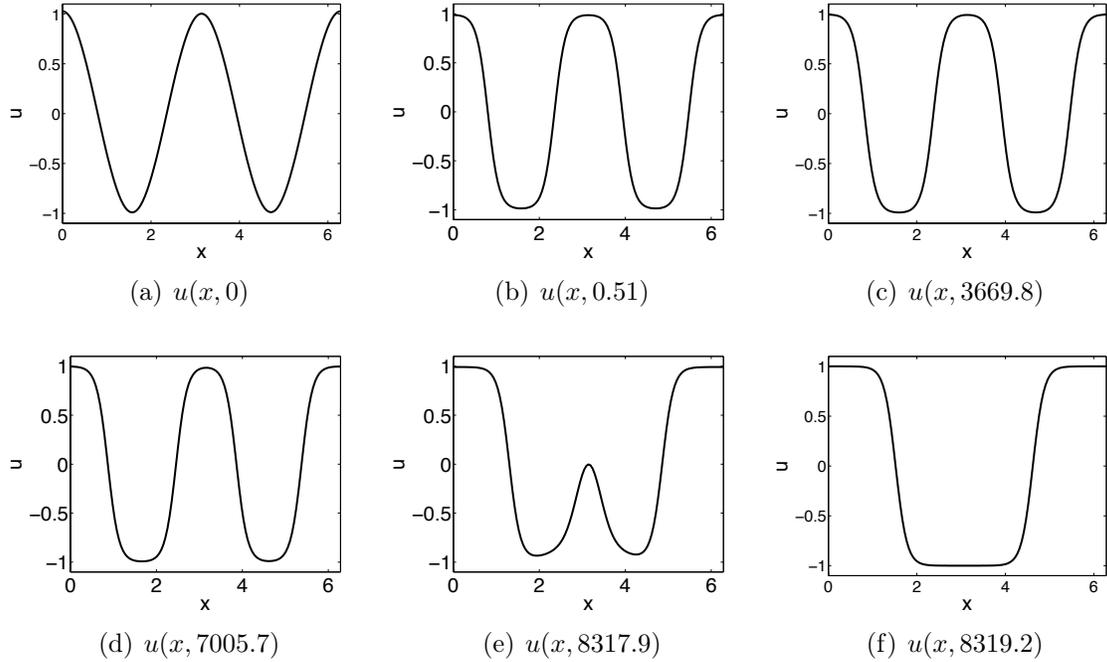


FIGURE 5. Spinodal evolution for the 1D CH equation using adaptive time stepping (BE-BDF2).

In Figure 4 we plot the time step size, fixed-point iteration history and energy of numerical solutions obtained by the BE-BDF2 strategy ($\delta_{\text{tol}} = 10^{-5}$). As shown in Figure 4(a), small time steps are used at early stage (spinodal evolution) but increase when the coarsening

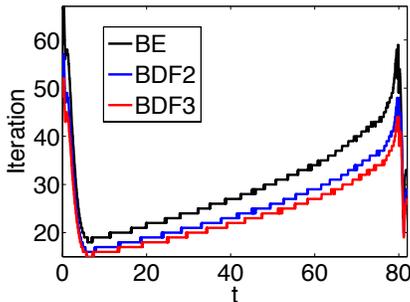
process starts, which speeds up the simulation without a loss of accuracy. If the iteration count is too large ($N_{\text{it}} \geq N_{\text{maxit}}$), we reject the solution and compute $u(x, t)$ again with reduced Δt . We see this behavior in Figure 4(b), where time steps are decreased whenever $N_{\text{it}} \approx N_{\text{maxit}}$. We also observe in Figure 4(c) that adaptive time stepping does not affect energy decay.

The phase function $u(x, t)$ obtained by our numerical scheme is shown in Figure 5. The initial state 5(a) quickly moves to the metastable state 5(b), and then finally reaches the stable state 5(f) at which two layers are merged together after a long time. This simulation also shows that the ripening event happens over a very fast time scale in Figure 5(e).

7.2. 2D CH model. We next solve the CH equation (1) in two spatial dimensions. The parameters are

$$\epsilon = 0.18, \quad \Delta x = \Delta y = \frac{2\pi}{128} \approx 0.0491, \quad N_{\text{maxit}} = 300, \quad N_{\text{tol}} = 10^{-6},$$

and the initial condition is (40). We first use BE, BDF2, and BDF3 with a small time step ($\Delta t = 0.05$) and check the ripening time, which is defined such that $u(\frac{\pi}{2}, \frac{\pi}{2})$ changes from positive to negative. Based on the results summarized in Figure 6, we can define such ripening time as $T_r = 80.10$. In addition, we note that raising the order of the scheme reduces the number of iterations per time step, and so the overall computational time is lower.



	Ripening time	Run Time (s)
BE	79.95	630.16
BDF2	80.10	526.19
BDF3	80.10	481.86

FIGURE 6. Ripening time of 2D CH equation with small fixed time ($\Delta t = 0.01$)

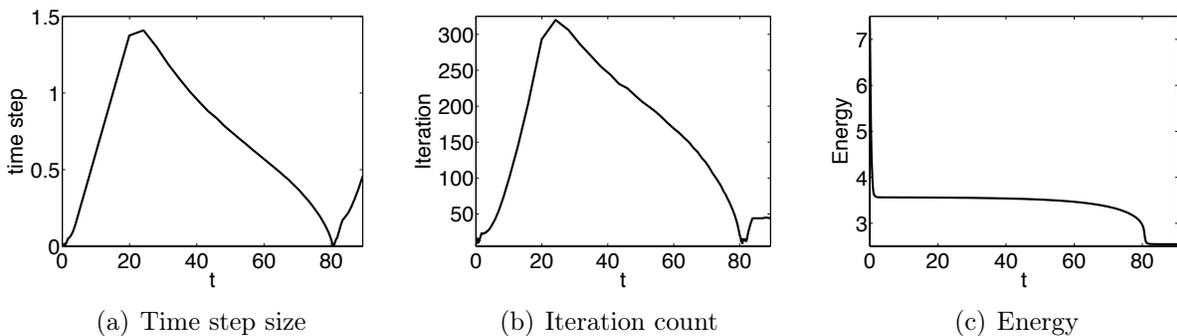


FIGURE 7. Adaptive time stepping for 2D, with the BE-BDF2 strategy.

Next, we simulate the same problem with adaptive time step method (BE-BDF2). The time step history, number of nonlinear iterations, and energy history are presented in Figure 7. The 2D results are comparable to those from 1D in Section 7.1, in that larger time step sizes are used during coarsening, and smaller steps are required only to capture ripening (see Figure 7(a)). The energy in Figure 7(c) indicates that there are two sharp transitions in the energy \mathcal{E} ; early on, and at ripening. Finally, we also observe the desired energy decay.

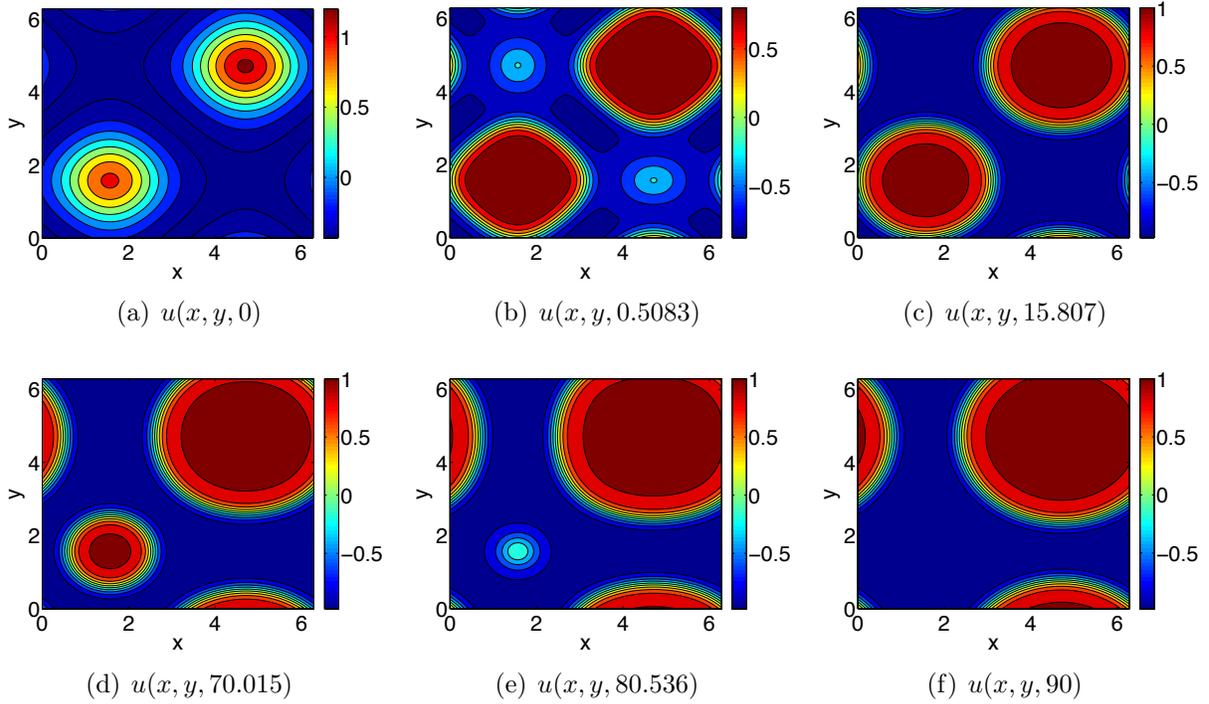


FIGURE 8. Temporal evolution of the 2D CH solution with initial condition (40).

The contour plots of time evolution of the phase function $u(x, y, t)$ are shown in Figure 8. The initial states 8(a) quickly reaches the metastable state, where we see two circular formations. Eventually the larger one absorbs the smaller, although over a very long time scale, shown in 8(c) - 8(e). The final state is shown in 8(f), where the larger region has fully consumed the smaller. In all plots, the total volume is preserved.

7.3. 2D CH vector model. We next apply our scheme from Section 5.2 to the 2D CH system (3), with our adaptive time stepping strategy. We use the same initial condition (40) for u_1 and

$$u_2(x, y, 0) = \sin(y), \quad (x, y) \in [0, 2\pi]^2 \quad (47)$$

for u_2 in the reference [9], and observe the long time behavior of the phase function $\mathbf{u} = (u_1, u_2)$. With the parameters

$$\epsilon = 0.32, \Delta x = \Delta y = \frac{2\pi}{64} \approx 0.0982, \quad N_{\max\text{it}} = 400, \quad N_{\text{tol}} = 10^{-6}, \quad \delta_{\text{tol}} = 1e - 4$$

we implement BE-BDF2 adaptive scheme for VCH model. The contour plots of $\cos(\arg(u_1 + iu_2))$ are shown in Figure 10. Instead of plotting u_1 and u_2 separately, [9] we define the angle

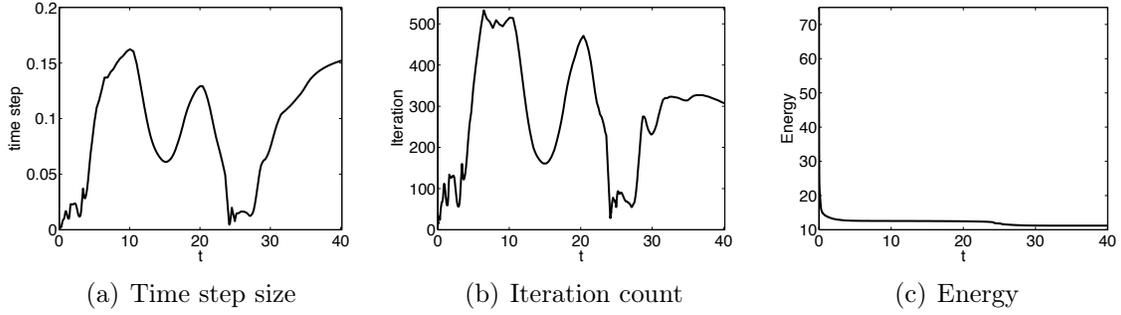


FIGURE 9. The time step, iteration count, and energy history of VCH solution.

$\theta \equiv \arg(u_1 + iu_2)$ at a triple junction, and plot $\cos(\theta)$ to avoid any discontinuities. We follow this benchmark plot using our numerical solution. After some initial ripening in Figure 10(b), the interfaces dividing the three states $\mathbf{u} = \mathbf{z}_i$ ($i = 1, 2, 3$) form around $T = 0.5$. In Figure 10(c) - 10(i), two of the values have $\cos(2\pi/3)$ (light blue in the plots) but separated by dark blue lines. Then the ripening process begins, and occurs over a long time scale, ending around $T = 24.286$ in Figure 10(f). Again, we remark that the volume is preserved over all time steps. For a more involved discussion of this simulation, we refer the interested reader to [9]. Our goal here is to demonstrate that our MOL^T scheme captures the correct ripening behavior.

7.4. 2D 6th order model. We also consider the 6th order problem:

$$u_t = \Delta [(\epsilon^2 \Delta - f'(u) + \epsilon^2 \eta)(\epsilon^2 \Delta u - f(u))], \quad f(u) = u^3 - u. \quad (48)$$

where ϵ and η are given positive constants. This problem is motivated by the functionalized Cahn-Hilliard (FCH) equation [18] which models interfacial energy in amphiphilic phase-separated mixtures. We follow a similar approach as before, and introduce the transformed variable $v = u + 1$. Substitution into (48) results in

$$v_t = \Delta [(\epsilon^2 \Delta - f'(v) + \epsilon^2 \eta)(\epsilon^2 \Delta v - f(v))], \quad f(v) = v^3 - 3v^2 + 2v. \quad (49)$$

We again apply the backward Euler (BE) scheme for time discretization of (49), so that

$$(I - \Delta t \epsilon^4 \Delta^3) v^{n+1} = v^n - \Delta t \Delta (\epsilon^2 \Delta f^{n+1} - \epsilon^2 (f' \Delta v)^{n+1} + (f f')^{n+1} + \eta \epsilon^4 \Delta v^{n+1} - \eta \epsilon^2 f^{n+1})$$

Hence, we invert the 6th order operator to solve v^{n+1} . One can invert this by completing the cube such that

$$\begin{aligned} (I - \sqrt[3]{\Delta t \epsilon^4 \Delta})^3 v^{n+1} &= v^n - \Delta t \Delta (\epsilon^2 \Delta f^{n+1} - \epsilon^2 (f' \Delta v)^{n+1} + (f f')^{n+1} + \eta \epsilon^4 \Delta v^{n+1} - \eta \epsilon^2 f^{n+1}) \\ &\quad - \left(3 \sqrt[3]{\Delta t \epsilon^4 \Delta} - 3 (\sqrt[3]{\Delta t \epsilon^4})^2 \Delta^2 \right) v^{n+1}. \end{aligned} \quad (50)$$

We can now solve v^{n+1} by applying the triple inversion of our modified Helmholtz operator $\mathcal{L} = I - \sqrt[3]{\Delta t \epsilon^4 \Delta}$ by the same procedure in Section 5.1.

For simplicity, we apply the first-order BE scheme (50) with a fixed time step $\Delta t = 0.1$ to solve the 6th order problem (49) ($\eta = 1$). By starting with the same initial condition (40)

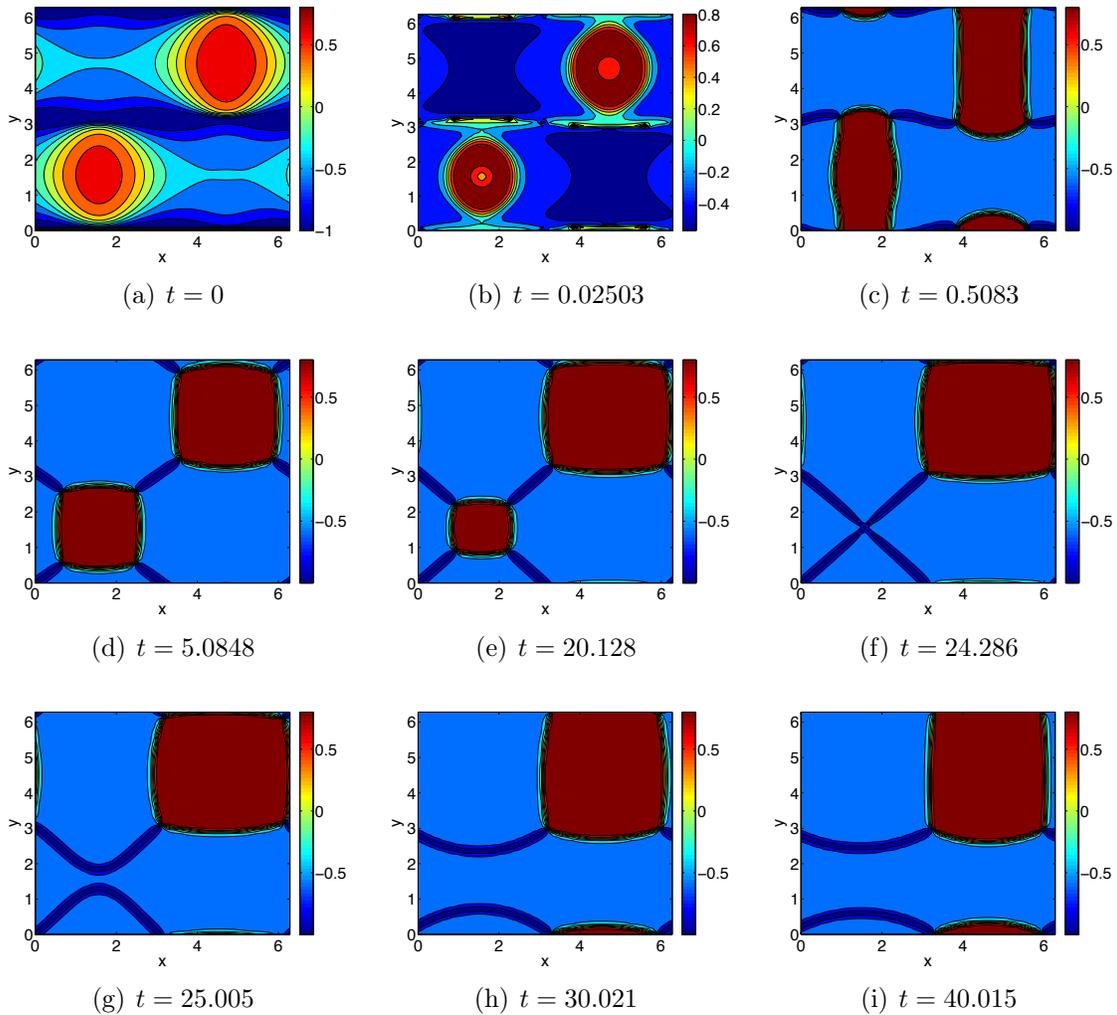


FIGURE 10. Temporal evolution of the 2D CH vector solution with the initial (40) for u_1 and (47) for u_2 . Contours of $\cos(\arg u_1 + i u_2)$ are plotted.

($\epsilon = 0.18$), and with the following parameters:

$$\Delta x = \Delta y = \frac{2\pi}{128} \approx 0.05, \quad N_{\max\text{it}} = 200, \quad N_{\text{tol}} = 10^{-6}.$$

The contour plots of our numerical solutions of (48) are shown in Figure 11. As expected [9], we can confirm that the final state becomes a regular array in Figure 11(f).

8. CONCLUSION

In this paper we have illustrated how the method of lines transpose (MOL^T) can be used to solve nonlinear phase field models, particularly the Cahn Hilliard, vector Cahn-Hilliard, and functionalized Cahn-Hilliard equations. We utilize a novel factorization of the semi-discrete equation to ensure gradient stability, permitting large time steps. When combined with time

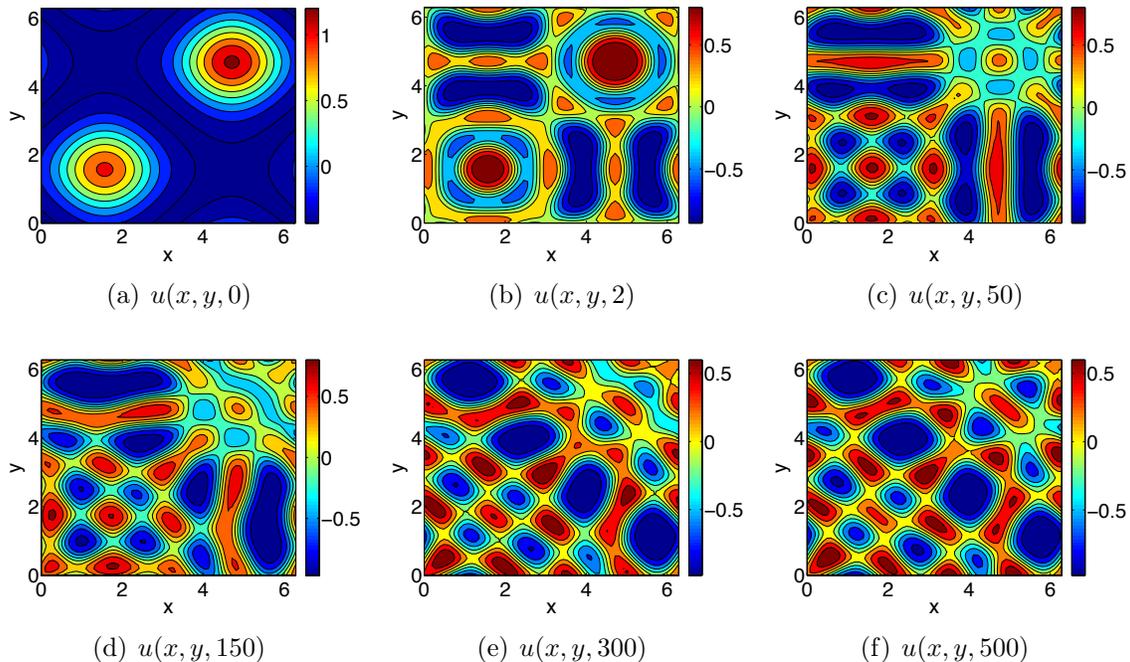


FIGURE 11. Temporal evolution of the 2D FCH equation with initial condition (40).

adaptivity, we are able to resolve rapid events such as spinodal evolution, which occur after long meta stable states.

The spatial solver is matrix free, $O(N)$, and logically Cartesian, and the splitting error is directly incorporated into the nonlinear fixed point iterations. We have considered higher order time stepping, such as backward difference formulas, implicit Runge-Kutta, and spectral deferred correction methods. Of all possible configurations, the time adaptive backward Euler-backward difference 2 (BE-BDF2) method is the most efficient in terms of accuracy and time to solution.

REFERENCES

- [1] R. Alexander. Diagonally implicit runge-kutta methods for stiff ode's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.
- [2] L. Bronsard and F. Reitich. On three-phase boundary motion and the singular limit of a vector-valued ginzburg-landau equation. *Archive for Rational Mechanics and Analysis*, 124(4):355–379, 1993.
- [3] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.
- [4] M. Causley, A. Christlieb, B. Ong, and L. Van Groningen. Method of lines transpose: An implicit solution to the wave equation. *Mathematics of Computation*, 83(290):2763–2786, 2014.
- [5] M. Causley, A. Christlieb, and E. Wolf. Method of lines transpose: an efficient a-stable solver for wave propagation. *Journal of Scientific Computing*, pages 1–26, 2016.
- [6] M. F. Causley, H. Cho, A. J. Christlieb, and D. C. Seal. Method of lines transpose: High order l-stable $o(n)$ schemes for parabolic equations using successive convolution. *SIAM Journal on Numerical Analysis*, 54(3):1635–1652, 2016.

- [7] M. F. Causley and A. J. Christlieb. Higher order a-stable schemes for the wave equation using a successive convolution approach. *SIAM Journal on Numerical Analysis*, 52(1):220–235, 2014.
- [8] L.-Q. Chen. Phase-field models for microstructure evolution. *Annual review of materials research*, 32(1):113–140, 2002.
- [9] A. Christlieb, J. Jones, K. Promislow, B. Wetton, and M. Willoughby. High accuracy solutions to energy gradient flows from material science models. *Journal of Computational Physics*, 257:193–215, 2014.
- [10] A. Christlieb, C. Macdonald, B. Ong, and R. Spiteri. Revisionist integral deferred correction with adaptive step-size control. *Communications in Applied Mathematics and Computational Science*, 10(1):1–25, 2015.
- [11] A. Christlieb, K. Promislow, and Z. Xu. On the unconditionally gradient stable scheme for the cahn-hilliard equation and its implementation with fourier method. *Commun. Math. Sci*, 11:345–360, 2013.
- [12] J. Douglas, Jr. On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods. *Journal of the society for industrial and applied mathematics*, 3(1):42–65, 1955.
- [13] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.
- [14] D. J. Eyre. An unconditionally stable one-step scheme for gradient systems. *Unpublished article*, 1998.
- [15] G. Fairweather and A. Mitchell. A new computational procedure for adi methods. *SIAM Journal on Numerical Analysis*, 4(2):163–170, 1967.
- [16] A. Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.
- [17] J. Jia and J. Huang. Krylov deferred correction accelerated method of lines transpose for parabolic problems. *Journal of Computational Physics*, 227(3):1739–1753, 2008.
- [18] N. Kraitzman and K. Promislow. An overview of network bifurcations in the functionalized cahn-hilliard free energy. In *Mathematics of Energy and Climate Change*, pages 191–214. Springer, 2015.
- [19] M. C. A. Kropinski and B. D. Quaife. Fast integral equation methods for the modified helmholtz equation. *Journal of Computational Physics*, 230(2):425–434, 2011.
- [20] F. Liu and J. Shen. Stabilized semi-implicit spectral deferred correction methods for allen-cahn and cahn-hilliard equations. *Math. Methods Appl. Sci*, 2013.
- [21] E. Rothe. Zweidimensionale parabolische randwertaufgaben als grenzfall eindimensionaler randwertaufgaben. *Mathematische Annalen*, 102(1):650–670, 1930.
- [22] J. Shen and X. Yang. Numerical approximations of allen-cahn and cahn-hilliard equations. *Discrete Contin. Dyn. Syst*, 28(4):1669–1691, 2010.
- [23] M. R. Willoughby. *High-order time-adaptive numerical methods for the Allen-Cahn and Cahn-Hilliard equations*. PhD thesis, University of British Columbia, 2011.

MATHEMATICS DEPARTMENT, KETTERING UNIVERSITY, FLINT, MI 48504
E-mail address: mcausley@kettering.edu

DEPARTMENT OF MATHEMATICS, MICHIGAN STATE UNIVERSITY, EAST LANSING, MI 48824
E-mail address: chohana@math.msu.edu

DEPARTMENT OF MATHEMATICS, MICHIGAN STATE UNIVERSITY, EAST LANSING, MI 48824
E-mail address: andrewch@math.msu.edu