METHOD OF LINES TRANSPOSE: HIGH-ORDER SCHEMES FOR PARABOLIC PROBLEMS

By

Hana Cho

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics - Doctor of Philosophy

2016

ABSTRACT

METHOD OF LINES TRANSPOSE: HIGH-ORDER SCHEMES FOR PARABOLIC PROBLEMS

By

Hana Cho

In the dissertation, we mainly consider developing efficient numerical schemes for Allen-Cahn and Cahn-Hilliard equations, which are the origin of the phase-field equations.

In the first part of the disseration, we present a new solver for nonlinear second-order parabolic problems that is L-stable and achieves high order accuracy in space and time. The solver is built by first constructing a single-dimensional heat equation solver that uses fast $\mathcal{O}(N)$ convolution. This fundamental solver is based on the use of the Green's function to invert a modified Helmholtz equation. Higher orders of accuracy in time are then constructed through a novel technique known as successive convolution, which facilitate our proofs of stability and convergence, and permit us to construct schemes that have provable stiff decay. The multi-dimensional solver is built by repeated application of dimensionally split independent fundamental solvers. We also solve nonlinear parabolic problems by using the integrating factor method, where we apply the basic scheme to invert linear terms (that look like a heat equation), and make use of Hermite-Birkhoff interpolants to integrate the remaining nonlinear terms. Our solver is applied to several linear and nonlinear equations including heat, Allen-Cahn, and the Fitzhugh-Nagumo system of equations in one and two dimensions.

In the second part of the disseration, we extend our Method Of Lines Transpose (MOL^{T}) scheme to Cahn Hilliard (CH) and vector Cahn Hilliard (VCH) equations. Our first step is

to establish the gradient stability for CH models. This procedure is just a simple change of variables where one of the fixed points is subtracted from the original variable. We prove that in the semi-analytic setting, using Backwards Euler time stepping.

After discretizing in time, we proceed to our spatial solver for inverting the linear part of the semi-analytic operator onto the non-linear part to construct an efficient fixed point method. This is done by factoring the fourth into the modified Helmholtz operators, which defined above. By including the splitting error into the right hand side of the fixed-point method, we arrive at a non-split scheme. We also combine the MOL^{T} formulation with existing time stepping for high order time stepping methods, and numerically demonstrated the gradient stable property in 1D and 2D in all simulations run. Time adaptive methods are shown to be more efficient than using the same method with large fixed time steps.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, professor Andrew Christlieb, for his support and encouragement during five years. Andrew has provided me with tremendous knowledge, great insight, patient guidance and pleasant research environment. It was such an amazing opportunity to discuss to our group members every week and to build a friendship. I especially thank professor Matt Causley, who gave me and our group valuable guidance about MOL^{T} scheme. He was always willing to spend his time to provide me academic knowledge and personal mentoring.

I thank to professor Keith Promislow, for his enlightening advices through his classes and discussions. His strong analysis of phase-field models has been valuable. I also thank the rest of committee members, professor Yingda Cheng, for teaching me great knowledge in numerical analysis, professor Shanker Balasubramaniam and professor Jose Perea, for valuable feedback and comments. I also thank to my Korean advisor, professor Jungho Yoon, for his continuous support academically and personally for ten years.

I thank to all my dear friends at MSU, EWHA, Youngran, Jiguchon and Lansing church, for their help and prayers. I have no doubt we will always be friends wherever we are.

I am eternally grateful to my family: my parents, my parents-in-law, my sister Eunhye (congrats your wedding!), my sister-in-law's family, and my grandparents, for their love, unconditional support, and prayers. My deepest gratitude should go to my husband, Sijung Oh, for his endless love. I wouldn't have made it any of this work without his patience, love, and support. He is my best friend, my best teacher, and my *Broom tree* of my life.

Last but not least, I am beyond words in my gratitude but forever thank to my Jesus, my Lord. *The Lord is my shepherd, I lack nothing.* All glory belongs to my Savior. Amen.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Phase-field method

Certain materials that we use in daily life, such as metals, alloys, ceramics or polymers, have properties that depend on their microstructure. Microstructure can be defined, in general, as compositional and structural inhomogeneities that arise during material processing [15]. Specifically, microstructures include grain arrays, phase distributions, precipitate dispersions, and dislocation networks, each of which produces different physical phenomena.

Since the microstructure of a material is strongly related to it's physical properties, composition, or performance, studying and investigating its evolution is crucial in material science. This investigation, in fact, has led to practical use in many areas, such as fluid mechanics [4], thermodynamics [7], and morphology [21] and so forth.

The process of microstructure evolution, however, is often very complex due to energy interactions. Specifically, this microstructure process is driven by the decrease of total free energy, which includes chemical energy, interfacial energy, elastic strain energy, electrostatic energy and magnetic energy [15]. In this context, the phase-field model, which allows studying the total free energy, has become a consolidated tool for simulating microstructure evolution for decades. The phase-field model has two major features, *interface* and *order parameter*, thus it is worth to review of the literature for interface and order parameter briefly.

An interface is defined as a surface which forming a common boundary between two

different phases of matter. A history of nature of the interface can be found in [4] and here, we summarize the essence of the literature. In the early 1800s, the interface was represented by a zero thickness surface (*sharp interface*) and it was assumed that physical quantities (e.g. density) were discontinuous across the interface. Poisson, Maxwell and Gibbs realized that the interface represented a rapid but smooth transition of physical quantities [30]. In the late 1800s, Lord Rayleigh and Van der Waals introduced the first concept of the non-zero thickness interface (*diffuse interface*), which circumvents certain difficulties in explicit-tracking in the sharp-interface model. These original concepts of the interface have been refined and developed for centuries.

In 1935, Landau and Lifshitz introduced a phase transition in a ferromagnetic crystal [43] such that there is an intermediate region where the magnetic moments change gradually from one direction to the opposite. This is the origin of the order parameter, space and time dependent fuction which is zero at the disordered phase and nonzero at the ordered phase. Later, Landau and Ginzburg extended the idea to the equilibrium of a superconductor; the free energy of a superconductor near transition can be described in terms of the spatial variation of an order parameter. Landau's work has been crucial for the development of modern theory of phase transition and order parameter became a foundation of phase-field models.

Based on this literature, the origin of the phase-field equation lies in the works of Cahn and Hilliard [8] on the interfacial free energy of non-uniform systems, and Allen and Cahn [3] on the antiphase boundary motion in binary alloy. Without explicitly tracking the interface positions, both models are able to predict corresponding complex microstructure, and thus have been extensively discussed in many areas, such as solidification, solid-state phase transformation, grain growth, and many examples in [15].

An important generalization of Cahn-Hilliard model is the functionalized Cahn Hilliard (FCH) model, which has been proposed by Promislow et.al. in [29, 21, 40]. The FCH free energy describes a phase separation in blends of amphiphilic polymers and solvents. Moreover, the FCH gradient flows are comprised of long-lived *network morphologies* of distinct co-dimension, and the authors analyized their geometric evolution, bifurcation and competition. A great review of the details of FCH model is in [40].

1.2 Review of previous works

In this Section, we review the previous works related to numerical methods for the phase-field models, especially for the Allen-Cahn(AC) and Cahn-Hilliard(CH) equations. Both equations have some numerical challenges, such as the presence of the small parameter ϵ which describes the interfacial width, the nonlinearity introduced by the free energy, and various time scales between each stage of temporal evolutions. There have been many contributions to resolve such numerical challenges, but we mainly focus on following representative methods: (1) the method of lines (MOL), and (2) the method of lines transpose (MOL^{T}). A review of more diverse numerical approximations of the CH model can be found in [56].

1.2.1 Method of Lines (MOL)

The Method of Lines (MOL) is a classical technique [52] for time-dependent partial differential equations (PDEs). MOL first discretizes the spatial derivatives, using various methods such as finite differences, and leaves the time variable continuous. This leads to a system of coupled ordinary differential equations (ODEs), with the corresponding initial and boundary conditions. The basic idea of the MOL is to replace the spatial (boundary value) derivatives

in the PDE with *algebraic* approximations.

We will review some numerical approximations that are based on the MOL literature.

1.2.1.1 Eyre's operator splitting

In 1998, Eyre proposed a convex splitting scheme [26] for the CH equation (3.1). After MOL discretization (using second-order centered finite differences in space), he split the potential function in the CH equation, producing the convex and non-convex functions

$$F(U_j) = F_c(U_j) + F_e(U_j), \qquad F_c \text{ and } -F_e \text{ are strictvely convex,}$$

where U_j approximates $u(x_j, t)$. He then treated the convex term implicitly and the non-convex term explicitly,

$$f(U_j^n, U_j^{n+1}) = F_c'(U_j^{n+1}) + F_e'(U_j^n) \equiv (U_j^{n+1})^3 - U_j^n,$$

where $f(u) = F'(u)$. This operator splitting guarantees unconditional energy stability, and solvability for any time step Δt. Thus many numerical schemes for phase-field models utilize Eyre's splitting, such as [18], where convex splitting is combined with a direct iterative solver.

From the point of view of solver efficiency, the splitting methods might be ideal. However, [19, 37] indicate that Eyre's splitting can lead to disproportionately large temporal errors during ripening, and can result in poor dynamics when time-accurate solutions are required.

1.2.1.2 The stabilized semi-implicit SDC method

The stabilized semi-implicit spectral deferred correction (stabilized SISDC) method was proposed by Shen et.al. for CH equation and related systems [46].

The authors used a spectral-Galerkin method (Legendre-Galerkin) in space which can address non-periodic boundary conditions, and applied the spectral deferred correction (SDC) for time discretization, where the basic solver (prediction) is a stabilized semi-implicit scheme [54] which is the author's previous work. The reaction term $f(u)$ was treated explicitly, but an extra dissipative term $\frac{S}{\epsilon^2}(u^{n+1} - u^n)$ was introduced. To ensure unconditional energy stability, the authors also considered a truncated potential $\tilde{F}$ such that $|\tilde{F}''(u)|$ is uniformly bounded in the maximum norm, and chose S depending on the upper bound of the $|\tilde{F}''(u)|$.

This approach guarantees high order unconditional energy stable methods for general gradient flow models, including a thin film model, and computational efficiency since the method does not require solving a nonlinear system. The additional stabilizing term they introduced in the first-order scheme (prediction), however, leads to the accuracy lost so that it might be only restricted to SDC framework to get higher order of accuracy in time. Moreover, [18] mentions that this semi-implicit scheme should impose a time step restriction to ensure unique solvability of CH equation.

1.2.1.3 Fully implicit scheme with conjugate gradient iteration

In [19], Wetton et.al. presented a new approach, a fully implicit time stepping scheme with a conjugate gradient (CG) spatial solver, for energy gradient flows from several models including AC, CH, higher-order derivative model and vector variants.

Specifically, they applied a standard Fourier pseudo-spectral discretization in space, but this produced a dense Jacobian matrix, which in [19] was preconditioned efficiently based on physics. They could get high order of accuracy in time using the Backward difference formula (BDF) as well as time adaptivity for long time coarsening process. Their high order scheme was able to easily extend to sixth order problem and vector problem.

One of the main contributions of this work is developing several benchmark problems for moving interface models in 1D, 2D and 3D. In this dissertation, we will also reproduce their benchmark problems of CH, vector CH and sixth order phase-field models in Chapter 3. However, since their scheme utilizes a Fourier spectral method, the problem is restricted to the periodic boundary conditions. Moreover, even though their numerical simulations show energy decaying solution for CH models, they do not give an analytical proof for energy stability.

1.2.1.4 Exponential time differencing (ETD) method

The exponential time differencing (ETD) [20], sometimes called the exponential integrator, is a classical numerical method for stiff systems. The ETD scheme is employed by the exact integration of the governing equations followed by an approximation of an integral involving the nonlinear terms.

In [37], a physics-based stable schemes for CH and FCH model were proposed. After a Fourier pseudo-spectral discretization, the ETD method was employed. In particular, the implicit high order Runge-Kutta (IRK) and the backward taylor expansions were considered. This led to a parallelizable code, which was implemented on Graphics Processing Units (GPUs).

A main contribution of the physics-based ETD scheme is implementations of various CH and FCH problems (2D, 3D) quickly and efficiently as a parallel computing, and the scheme can predict various geometric events of FCH models.

However, the physics-based ETD scheme is also limited to periodic boundary conditions on rectangular domains. Another issue is that the scheme suffers from a loss of convergence order for large time steps (called *freeze out* solution), prohibiting to use of large time steps

during coarseng process of models.

1.2.2 Method of Lines Transpose ($\mathbf{MOL^T}$)

As shown in Section 1.2.1, several schemes based on MOL discretization have had many contributions to the numerical solution of phase-field models. However, since MOL scheme discretizes in space first, the boundary conditions must be chosen immediately and any proof of stability depends on that discretization. The $\mathrm{MOL^T}$ scheme, referred to as Rothe's method [38], or the horizontal line method, is an alternative to MOL formulation.

The $\mathrm{MOL^T}$ starts by discretizing a PDE in time but continuous in space. Hence, any stability properties are independent of the spatial discretization and the PDE is transformed to a coupled set of elliptic boundary value problems (BVPs), which can be solved through *analytic* inversion. The challenge, then, is to approximate the integral equations. For instance, a naive approach leads to a direct solver formed by a dense matrix-vector product,

$$U^n(x_j) = \int_\Omega Gf\, dx \approx \sum_{i=1}^{N} G(x_i - x_j) f_i, \qquad i = 1, \cdots, N.$$

where N is number of spatial grid points, and $G(\cdot)$ is a Green's function (kernel). Then this summation yields $\mathcal{O}(N^2)$ complexity, which is impractical in large number of N.

Due to the lack of efficient integral solvers, the $\mathrm{MOL^T}$ scheme required greater storage than other methods, thus the scheme did not get a great deal of attention before the early 1980s. However, in 1987, Greengard and Rokhlin first suggested a potential theory and developed a fast algorithm for integral equation based on the theory [31], since then the $\mathrm{MOL^T}$ scheme has received more attention recently with continuous progress of the fast algorithm for integral equations.

In this Section, we will review some representative works based on the MOL$^{\mathrm{T}}$ framework.

1.2.2.1 Fast multipole method (FMM)

The fast multipole method (FMM) was first introduced by Greengard and Rokhlin to evaluate the Coulomb potential and force fields in particle system [31]. The FMM has since been extended to include other potential functions, such as the Yukawa potential [32], which is also called a a screened Coulomb potential.

The basic idea of FMM is an adaptive quad-tree structure in order to impose a hierarchy of refinements on the computational domain. For each particle, the *nearby* particles to the potential field are handled directly by summation, but *far-field* (non-neighbor) interactions are handled using multipole expansions [42]. In other words, the FMM first forms the mulipole expansions for all of nodes in the quad-tree structure, and then constructs a local expansion for each node in far-field. Specifically, Kropinski et.al. presented FMM for the modified Helmholtz equation which appears in the semi-discrete solution of the Heat, the Navier-Stokes, and linearized Poisson-Boltzmann equation in [42]. Based on FMM, the presented solution computes as rapid $\mathcal{O}(N)$ operations, where N is the number of nodes in the discretization of the boundary, even in the highly complex domain.

The FMM is one of the most efficient and robust methods, which can speed up the calculation; however, this scheme might be inefficient for modern parallel processing because of global coupling of the boundary terms.

1.2.2.2 Fourier continuation-alternating direction (FC-AD) method

In 2010, Bruno and Lyon developed a new method for general PDEs, which is FC-AD (Fourier-Continuation Alternating-Direction) in [6, 48]. The well-known Alternating Direc-

tion Implicit (ADI) approach is pioneered by Douglas, Peaceman and Rachford [22, 27].

First, the FC (Gram) solver is the method of spatial differentiation, which enables high order/spectral convergence of Fourier expansions of non-periodic functions. The high order FC-AD algorithm yields unconditional stability for general domains at an $\mathcal{O}(Nlog(N))$ cost per time step.

The exact Green's function is replaced with a dimensionally split Greens function. As a result, the boundary conditions are no longer globally coupled. The scheme also enables efficient parallel implementations for general domains and does not suffer both Gibbs phenomenon and CFL restrictions in hyperbolic problems. However, This is at the expense of introducing splitting error.

1.2.2.3 Dimensionally splitting-fast convolution method

Recently, Christlieb's group has presented the fast convolution strategy based on dimensional splitting MOL$^{\mathrm{T}}$ schemes in [11, 13, 10, 12]. Each of these schemes is able to make analytically solve simpler, one-dimensional boundary value problem, and the subsequent solution can be constructed through dimensional sweeps. This is also unconditionally stable numerical scheme with computational cost and coding complexity comparable to explicit schemes.

The resulting dimensionally split MOL$^{\mathrm{T}}$ method has been demonstrated on a range of problems with complex geometry and boundary conditions. In [13] successive convolution was introduced, leading to an A-stable method of arbitrary order in time for hyperbolic problems. Moreover, [10, 58] include the development of an embedded boundary method for Neumann boundary conditions on a complex boundary geometry and several numerical examples, such as Maxwells equation. The contributions of the present author to [12] extend the dimensionally split MOL$^{\mathrm{T}}$ to high order in space and time for linear and non-linear

second-order parabolic problems, which will be reproduced in this dissertation. This work is ongoing to high order derivative phase-field models, such as CH and FCH equations. In the second part of the dissertation, we turn our attention to above models.

In each case, the spatial operator if factored, so that the boundary value problem is solved using a composition of one-dimensional solves. Each solver is $\mathcal{O}(N)$ and matrix-free, and the complexity is comparable to explicit.

1.3 Outline of the dissertation

In the dissertation, we mainly consider the efficient numerical schemes for Allen-Cahn and Cahn-Hilliard equations. First, we would like to solve the equations in such a way as to achieve high order of accuracy in time and space. Second, we would like to maintain the energy stability when the method is used to model any microstructural evolution, such as coarsening process in binary alloy.

In Chapter 2, we propose a numerical method for the parabolic equations, include the linear heat equation and nonlinear equations of reaction-diffusion type. This Chapter is organized as follows: In Section 2.2, we derive the basic scheme for the one-dimensional heat equation, which is L-stable and can achieve high orders of accuracy in space and time. In Section 2.3, we describe how to obtain an arbitrary order discretization in a single dimension with resolvent expansions. In Section 2.4.1, we describe how this can be extended to multiple dimensions, and we present results for linear heat in one and two dimensions in Section 2.3.3 and 2.4.2, respectively. In Section 2.5, we describe how our approach can handle nonlinear source terms, and we present numerous numerical results including Allen-Cahn and the Fitzhugh-Nagumo system of equations.

In Chapter 3, we extend the dimensionally split MOL^{T} for the 1D and 2D Cahn-Hilliard and vector variant. In Section 3.1, we introduce the CH model and vector CH model, which are of interest in this Chapter. In Section 3.2, we derive a first order scheme for CH equation in a basic 1D setting and the energy stability proof in semi-discrete setting in Section 3.2.1. In Section 3.3, we modify the traditional time stepping scheme to achieve higher orders of accuracy in time and present the temporal refinement studies in Section 3.3.4. In addition, we extend our 1D solver to multiple spatial dimension in Section 3.4 with the refinement studies in 2D in Section 3.4.3, as well as extend to vector model in Section 3.5. Finally, we describe an adaptive time stepping strategy in Section 3.6 and present numerous numerical results including 1D, 2D and vector model in Section 3.7.

Chapter 2

MOL^T for Parabolic Equations

2.1 Motivation

The prototypical parabolic differential equation is the heat equation.

$$u_t = \Delta u, \tag{2.1}$$

subject to appropriate initial and boundary condition. Here $t > 0$ and $x \in \Omega$, where $\Omega \subset \mathbb{R}^n$ is open. The Laplacian Δ is taken with respect to the spatial variables $x = (x_1, x_2, \cdots, x_n)$: $\Delta u = \sum_{i=1}^{n} u_{x_i x_i}$. It forms a cornerstone of mathematics and physics, and its understanding is essential for defining more complicated mathematical models.

Fourier introduced this equation as a means to describe transient heat flow. Fick quickly recognized its importance to particle and chemical concentrations. As a result, parabolic equations are now ubiquitous in describing diffusion processes, which are found in a vast array of physical problems, among which are reaction-diffusion models of chemical kinetics [3, 8], phase field models describing morphology and pattern formation in multiphase fluids and solids [5, 9, 21], and even the volatility of stocks and bonds in mathematical finance [55].

Numerical solutions of (linear and nonlinear) diffusion equations have been the subject of active research for many decades. As early as the 1950's and 60's, it was recognized that due to the parabolic scaling, method of lines (MOL) discretizations of the heat equation lead to

numerically stiff systems of equations. In principal, the numerical stiffness can be subsided by taking larger time steps, which are only stable if fully implicit solvers are used. But in practice, the memory of early computers was extremely limited, making full matrix inversions difficult and costly. Thus, alternate dimensionally implicit (ADI) splitting methods [22, 27], which utilize dimensional splitting and tridiagonal solvers, quickly gained popularity.

Later on, memory constraints no longer defined the bottleneck for computing, and attention shifted toward methods that focused on reducing floating point operations (FLOPs), albeit with additional memory constraints. Most notable among these are Krylov methods [35], boundary integral methods in Sections 1.2.2.1, and quadrature methods [47, 36]. However, with the advent of GPU processors, it appears that we are yet again seeing a paradigm shift towards methods that should emphasize small memory footprints, even at the expense of incurring a higher operation count. Thus, ADI-like methods, which can efficiently decompose larger problems and limit overhead communication, warrant further investigation, and these features are the motivating factor for this work.

We now start to propose a novel numerical method, MOL^{T} in Section 2.2 for obtaining solutions to the linear heat equation (2.1).

2.2 First order scheme for 1D Heat equation

We begin by forming a semi-discrete solution to the 1D heat equation using MOL^{T} scheme. Let $u = u(x,t)$ satisfy

$$u_t = \gamma u_{xx}, \quad (x,t) \in (a,b) \times [0,T], \tag{2.2}$$

with constant diffusion coefficient γ, and appropriate initial and boundary conditions. The MOL^T amounts to employing a finite difference scheme for the time derivative, and collocating the Laplacian term at time levels $t = t^n$ and $t = t^{n+1}$, so that collocation has the form

$$\frac{u^{n+1} - u^n}{\Delta t} = \gamma \partial_{xx} \left(u^n + \frac{u^{n+1} - u^n}{\beta^2} \right), \qquad \beta > 0,$$

where we intoduced a free parameter $\beta > 0$. Next, we introduce the differential operator corresponding to the modified Helmholtz equation, defined by

$$\mathcal{L} = I - \frac{\partial_{xx}}{\alpha^2}, \quad \alpha = \frac{\beta}{\sqrt{\gamma \Delta t}}. \tag{2.3}$$

After some algebra, we find that the scheme can be written as

$$\mathcal{L}[u^{n+1} - (1 - \beta^2)u^n] = \beta^2 u^n. \tag{2.4}$$

We note that there are at least two reasonable strategies for choosing β:

1. **Maximize the order of accuracy.** For example, if we choose $\beta^2 = 2$, then the discretization is the trapezoidal rule, which is second order accurate and A-stable.

2. **Enforce stiff decay.** For example, if we choose $\beta^2 = 1$, then the discretization is the Backward Euler scheme, which is first order accurate, L-stable, yet does not maximize the order of accuracy.

Here and below, we opt for the second strategy, as the stiff decay of numerical solutions of the heat equation is of paramount importance. In Section 2.3.2, we develop this discussion in the context of higher order schemes that relies on a careful selection of β as well as repeated

applications of a single inverse operator.

Upon solving equation (2.4) for u^{n+1}, we find that the equation for the update is

$$u^{n+1} = (1 - \beta^2)u^n + \beta^2 \mathcal{L}^{-1}[u^n], \tag{2.5}$$

that requires inverting a modified Helmholtz operator. We will accomplish this *analytically* by using Green's function. We shall define $\mathcal{L}^{-1}$ in the ensuing discussion.

2.2.1 Inversion of the modified Helmholtz operator

In this Section, we seek the free-space Green's function $G(x|y)$ as the solution to

$$\mathcal{L}[G](x) = \delta(x - y), \quad -\infty < x, y < \infty, \tag{2.6}$$

where δ is the delta distribution and $\mathcal{L}$ is the differential operator in (2.3). We first obtain the solution G, which is stated in the following theorem.

Theorem 2.2.1. *The free space Green's function satisfying* (2.6) *is given by*

$$G(x|y) = \frac{\alpha}{2} e^{-\alpha|x-y|}, \qquad \alpha = \frac{\beta}{\sqrt{\gamma \Delta t}}. \tag{2.7}$$

Proof. We first consider $x \neq y$, and upon solving the homogeneous differential equation, $\mathcal{L}[G] = 0$,

$$G(x|y) = \begin{cases} Ae^{-\alpha(x-y)} + Be^{\alpha(x-y)}, & -\infty < x < y, \\ Ce^{-\alpha(x-y)} + De^{\alpha(x-y)}, & y < x < \infty. \end{cases}$$

To ensure that G remains bounded for all x and y, we set $A = D = 0$. Then it remains to determine B and C. We demand that G be continuous at $x = y$, but allow for the derivative of G to be discontinuous. The size of the jump discontinuity is obtained by integrating the equation over the interval $[y - \epsilon, y + \epsilon]$ and letting $\epsilon \to 0$, from which we find

$$[G_x]_{x=y} = \alpha^2.$$

Imposing that the continuity and jump conditions yields the following set of equations

$$C - B = 0, \quad -\alpha(C + B) = -\alpha^2.$$

The solution is then given by $B = C = \dfrac{\alpha}{2}$, and the result (2.7) follows. □

We note that the modified Helmholtz equation has a non-oscillatory Green's function, and especially it is exponential decaying property. Now, we consider a differential equation of the form

$$\mathcal{L}[u^{n+1}](x) = u^n(x), \quad a \le x \le b, \tag{2.8}$$

where u^n is an approximation of $u(x, t^n)$, $(n \in N)$. Upon multiplying by G and integrating over $[a, b]$, we find after integration by parts that

$$u^{n+1}(x) = \frac{\alpha}{2} \int_a^b e^{-\alpha|x-y|} u^n(y) dy + \frac{1}{2\alpha} \left[u(y) \frac{\partial}{\partial y} e^{-\alpha|x-y|} - e^{-\alpha|x-y|} \frac{\partial}{\partial y} u(y) \right]_{y=a}^{y=b}.$$

The boundary terms can also be evaluated explicitly, using

$$\lim_{y \to a} \frac{\partial}{\partial y} e^{-\alpha|x-y|} = \alpha e^{-\alpha(x-a)}, \quad \lim_{y \to b} \frac{\partial}{\partial y} e^{-\alpha|x-y|} = -\alpha e^{-\alpha(b-x)},$$

which gives

$$\frac{1}{2\alpha}\left[u(y)\frac{\partial}{\partial y}e^{-\alpha|x-y|}-e^{-\alpha|x-y|}\frac{\partial}{\partial y}u(y)\right]_{y=a}^{y=b}=B_a e^{-\alpha(x-a)}+B_b e^{-\alpha(b-x)},$$

for some coefficients B_a and B_b. Moreover, the boundary terms satisfy the homogeneous differential equation $\mathcal{L}[u]=0$.

Based on this result, an inversion of the modified Helmholtz operator (2.3) is defined that

$$\mathcal{L}^{-1}[u^n]=\left(I-\frac{\partial_{xx}}{\alpha^2}\right)^{-1}[u^n]:=\frac{\alpha}{2}\int_a^b e^{-\alpha|x-y|}u^n(y)dy+B_a e^{-\alpha(x-a)}+B_b e^{-\alpha(b-x)}, \quad (2.9)$$

and the coefficients B_a and B_b are determined by applying prescribed boundary conditions at $x=a,b$ which we describe in Section 2.2.3.

Remark 1. *Alternatively, had we followed the method of lines (MOL) and first discretized* (2.2) *in space, then the differential operator $\mathcal{L}$ would be replaced by an algebraic operator L, and would be inverted numerically.*

Remark 2. *Although the update* (2.5) *(with $\beta\neq 1$) is only first order accurate, we describe in Section 2.3 how to extend our procedure to arbitrary order in time.*

Remark 3. *Since dimensional splitting is used, all spatial quantities are computed according to a one-dimensional convolution integral of the form* (2.9)*, which is performed on a line-by-line basis, following so-called "dimensional sweeps". Since the discrete convolution is computed in $\mathcal{O}(N)$ complexity, the full solver scales linearly in the number of spatial points (assuming each sweep is performed in parallel).*

Thus, we find that the inversion operator consists of two parts, the particular solution and the homogeneous solution. The convolution operator is comprised of a particular solution,

which is defined by the convolution integral

$$I[u^n](x) := \frac{\alpha}{2}\int_a^b e^{-\alpha|x-y|}u^n(y)dy, \qquad n \in N, \tag{2.10}$$

and a homogeneous solution

$$B_a e^{-\alpha(x-a)} + B_b e^{-\alpha(b-x)}, \tag{2.11}$$

both of which can be constructed in $\mathcal{O}(N)$ operations using fast convolution.

Furthremore, a fully discrete scheme is obtained after a spatial discretization of (2.9). The domain $[a, b]$ is partitioned into N subdomains $[x_{j-1}, x_j]$, with $a = x_0 < x_1 < \ldots x_N = b$. We now describe each of these in turn, starting with the first.

2.2.2 Particular solution

The particular solution is first split into $I[u](x) = I^L(x) + I^R(x)$, where

$$I^L(x) = \frac{\alpha}{2}\int_a^x e^{-\alpha(x-y)}u(y)dy, \quad I^R(x) = \frac{\alpha}{2}\int_x^b e^{-\alpha(y-x)}u(y)dy.$$

We now make a few key observations about the particular solution, which will be used extensively in the ensuing discussion.

2.2.2.1 Characteristic decomposition for particular solution

Each of these parts independently satisfy the first order "initial value problems"

$$(I^L)'(y) + \alpha I^L(y) = \frac{\alpha}{2}u(y), \quad a < y < x, \quad I^L(a) = 0,$$
$$(I^R)'(y) - \alpha I^R(y) = -\frac{\alpha}{2}u(y), \quad x < y < b, \quad I^R(b) = 0,$$

where the prime denotes spatial differentiation. From the integrating factor method, the integral satisfies the following identity, known as exponential recursion

$$I^L(x_j) = e^{-\nu_j} I^L(x_{j-1}) + J^L(x_j), \quad \text{where} \quad J^L(x_j) = \frac{\alpha}{2}\int_{x_{j-1}}^{x_j} e^{-\alpha(x_j - y)}u(y)dy, \tag{2.12}$$

$$I^R(x_j) = e^{-\nu_{j+1}} I^R(x_{j+1}) + J^R(x_j), \quad \text{where} \quad J^R(x_j) = \frac{\alpha}{2}\int_{x_j}^{x_{j+1}} e^{-\alpha(y - x_j)}u(y)dy, \tag{2.13}$$

and

$$\nu_j = \alpha h_j, \qquad h_j = x_j - x_{j-1}.$$

This expression is still exact, and indicates that only the "local" integrals J^L and J^R needs to be approximated. By symmetry, the scheme for J^R follows from that of J^L, which we describe.

2.2.2.2 A spatial solver for particular solution

We first consider a projection of $u(y)$ onto P_M, the space of polynomials of degree M. A local approximation

$$u(x_j - zh_j) \approx p_j(z), \quad z \in [0,1],$$

is accurate to $\mathcal{O}(h_j^M)$, and defines a quadrature of the form

$$J^L(x_j) = \frac{\nu_j}{2} \int_0^1 e^{-\nu_j z} u(x_j - h_j z) dz \approx \frac{\nu_j}{2} \int_0^1 e^{-\nu_j z} p_j(z) dz. \tag{2.14}$$

If standard Lagrange interpolation is used, then the polynomials can be factorized as

$$p_j(z) = \sum_{k=-\ell}^{r} p_{jk}(z) u_{j+k} = z^T A_j^{-1} u_j^M, \tag{2.15}$$

where $z = [1, z, \ldots, z^M]^T$, and $u_j^M = [u_{j-\ell}, \ldots, u_j, \ldots, u_{j+r}]^T$, and A_j is the Vandermonde matrix corresponding to the points x_{j+k}, for $k = -\ell \ldots r$,

$$A_j = \begin{bmatrix} \left(\frac{x_j - x_{j-l}}{h_j}\right)^0 & \cdots & \left(\frac{x_j - x_{j-l}}{h_j}\right)^M \\ \vdots & \ddots & \vdots \\ \left(\frac{x_j - x_{j+r}}{h_j}\right)^0 & \cdots & \left(\frac{x_j - x_{j+r}}{h_j}\right)^M \end{bmatrix} \in \mathbb{R}^{M+1, M+1}$$

The values of ℓ and r are such that $\ell + r = M$, and are centered about j except near the boundaries, where a one-sided stencil is required.

Substituting this factorization into (2.14) and integrating against an exponential, we find that

$$J^L(x_j) \approx J_j^L := \sum_{k=-\ell}^{r} w_{jk} u_{j+k},$$

where the weights $w_j = [w_{j,-\ell}, \dots w_{j,r}]$ satisfy

$$w_j^T = \phi_j^T A_j^{-1} \tag{2.16}$$

and where

$$\phi_{jk} = \frac{\nu_j}{2} \int_0^1 e^{-\nu_j z} z^k dz = -\frac{e^{-\nu_j}}{2} + \frac{k!}{2\nu_j^k} \left(1 - e^{-\nu_j} \sum_{p=0}^{k-1} \frac{(\nu_j)^p}{p!} \right) \tag{2.17}$$

If the weights are pre-computed, then the fast convolution algorithm scales as $\mathcal{O}(MN)$ per time step, and achieves a user-defined $\mathcal{O}(M)$ in space. In every example shown in this work, we choose $M = 2, 4$ or $M = 6$.

2.2.3 Homogeneous solution

The homogeneous solution in (2.11) is used to enforce boundary conditions. We first observe that all dependence on x in the convolution integral, $I(x) := I[u^n](x)$, in (2.10) is on the Green's function, which is a simple exponential function. Through direct differentiation, we obtain

$$I_x(a) = \alpha I(a), \quad I_x(b) = -\alpha I(b). \tag{2.18}$$

Various boundary conditions at $x = a$ and $x = b$ can be enforced by solving a simple 2×2 system for the unknowns B_a and B_b.

Remark 4. *The cases of applying different boundary conditions at $x = a$ and $x = b$ are not considered here, but the details follow from an analogous procedure to that demonstrated below.*

2.2.3.1 Dirichlet boundary conditions

Let us begin with Dirichlet boundary conditions, which we shall define by $u^n(a) = w_a$ and $u^n(b) = w_b$ (at each discrete time step, $t = t^n$). Then

$$w_a = I(a) + B_a + B_b\mu, \quad w_b = I(b) + B_a\mu + B_b,$$

where $\mu = e^{-\alpha(b-a)}$. This system of equations can be solved for the B_a and B_b, yielding

$$B_a = \frac{(w_a - I_a) - \mu(w_b - I_b)}{1 - \mu^2}, \quad B_b = \frac{(w_b - I_b) - \mu(w_a - I_a)}{1 - \mu^2}, \tag{2.19}$$

where $I_a = I(a)$ and $I_b = I(b)$.

2.2.3.2 Neumann boundary conditions

To apply the Neumann boundary conditions, we use the identities (2.18), and denote $u^n_x(a) = \alpha w'_a$ and $u^n_x(b) = \alpha w'_b$. This yields

$$\alpha w'_a = \alpha\left(I'(a) - B_a + B_b\mu\right), \quad \alpha w'_b = \left(-I'(b) - B_a\mu + B_b\right),$$

and this system of equations can be solved for the B_a and B_b,

$$B_a = \frac{(I'_a - w'_a) + \mu(w'_b + I'_b)}{1 - \mu^2}, \quad B_b = \frac{(w'_b + I'_b) + \mu(I'_a - w'_a)}{1 - \mu^2}. \tag{2.20}$$

2.2.3.3 Periodic boundary conditions

We can also impose Periodic boundary conditions, by assuming that

$$u^n(a) = u^n(b), \quad u_x^n(a) = u_x^n(b), \qquad \forall n \in \mathbb{N}. \tag{2.21}$$

We next enforce this assumption to hold on the scheme (2.9),

$$\begin{aligned}
\mathcal{L}^{-1}[u^n](a) = \mathcal{L}^{-1}[u^n](b) \quad &\Longleftrightarrow \quad I(a) + B_a + B_b\mu = I(b) + B_a\mu + B_b, \\
\mathcal{L}_x^{-1}[u^n](a) = \mathcal{L}_x^{-1}[u^n](b) \quad &\Longleftrightarrow \quad \alpha\left(I(a) - B_a + B_b\mu\right) = \alpha\left(-I(b) - B_a\mu + B_b\right),
\end{aligned}$$

where the identities (2.18) are used to find $\mathcal{L}_x^{-1}$. Solving this linear system yields

$$B_a = \frac{I_b}{1-\mu}, \quad B_b = \frac{I_a}{1-\mu}. \tag{2.22}$$

2.3 Higher order scheme for 1D Heat equation

In our group's recent work [13], Causley et.al. applied a successive convolution approach to derive high order A-stable solvers for the wave equation. We extend this successive convolution idea based on new approach, resolvent expansion, for the heat equation.

The key idea is to recognize the fact that, in view of the modified Helmholtz operator (2.3), the second derivative can be factored as

$$\left(-\frac{\partial_{xx}}{\alpha^2}\right) = \mathcal{L} - I = \mathcal{L}\left(I - \mathcal{L}^{-1}\right) := \mathcal{L}\mathcal{D}, \tag{2.23}$$

where

$$\mathcal{D} = I - \mathcal{L}^{-1}, \qquad \mathcal{L} = (I - \mathcal{D})^{-1}. \tag{2.24}$$

Substitution of the second expression into (2.23) determines the second derivative completely in terms of this new operator

$$\left(-\frac{\partial_{xx}}{\alpha^2}\right) = (I - \mathcal{D})^{-1}\,\mathcal{D} = \sum_{p=0}^{\infty} \mathcal{D}^p. \tag{2.25}$$

This shows that second order partial derivatives of a sufficiently smooth function $u(x)$ can be approximated by truncating a resolvent expansion based on successively applying $\mathcal{D}$ to $u(x)$, which is a linear combination of successive convolutions (2.9). We also note that for implementation, each operator is applied successively, and is defined by

$$\mathcal{D}^{(p+1)}[u] := \mathcal{D}[\mathcal{D}^p[u]], \quad \mathcal{D}^0[u] := u.$$

Now, we consider a solution $u(x,t)$ to the heat equation (2.2), that for simplicity we take to be infinitely smooth. We perform a Taylor expansion on $u(x, t + \Delta t)$, and then use the Cauchy-Kovalevskaya procedure [53] to exchange temporal and spatial derivatives to yield

$$u(x, t + \Delta t) = \sum_{p=0}^{\infty} \frac{(\Delta t \partial_t)^p}{p!} u(x,t) = \sum_{p=0}^{\infty} \frac{(\gamma \Delta t \partial_{xx})^p}{p!} u(x,t) =: e^{\gamma \Delta t \partial_{xx}} u(x,t). \tag{2.26}$$

The term $e^{\gamma \Delta t \partial_{xx}}$ is a spatial pseudo-differential operator, and it compactly expresses the full Taylor series. Our goal is to make use of the formula (2.25) to convert the Taylor series into a resolvent expansion [24, 25]. This can be performed term-by-term, and requires rearranging a doubly infinite sum. However, if we instead work directly with the operator

defining the Taylor series, then

$$e^{\gamma \Delta t \partial_{xx}} = e^{-\beta^2\left(-\frac{\partial_{xx}}{\alpha^2}\right)} = e^{-\beta^2 (I-\mathcal{D})^{-1}\mathcal{D}}.$$

The last exponential form of the operator reminds us a generating function of orthogonal polynomials, which is the generating function of the *generalized Laguerre* polynomials $L_p^{(\lambda)}(z)$,

$$\sum_{p=0}^{\infty} L_p^{(\lambda)}(z) t^p = \frac{1}{(1-t)^{\lambda+1}} e^{-\frac{tz}{1-t}}, \qquad z > 0, \tag{2.27}$$

and this bears a striking resemblance to our expansion. Indeed, if we take $\lambda = -1$, substitute $z = \beta^2$ and $t = \mathcal{D}$, then

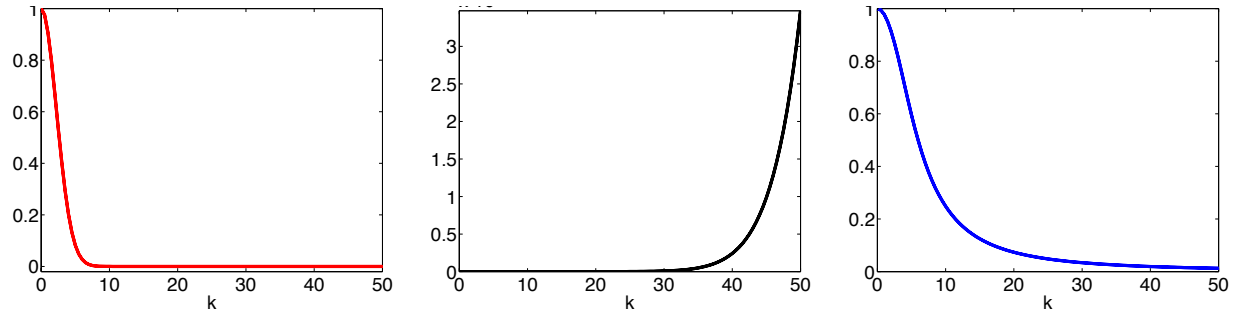
$$e^{-\beta^2 (I-\mathcal{D})^{-1}\mathcal{D}} = \sum_{p=0}^{\infty} L_p^{(-1)}(\beta^2)\mathcal{D}^p = I + \sum_{p=1}^{\infty} L_p^{(-1)}(\beta^2)\mathcal{D}^p. \tag{2.28}$$

Therefore, the Laguerre expansions via our successive convolutions of $\mathcal{D}$ operator and the Taylor expansion are exactly same with our exponential form of operator, $e^{-\beta^2 (I-\mathcal{D})^{-1}\mathcal{D}}$, in the limiting sense. However, we do prefer the Laguerre expansion for the stability reason.

To compare the Laguerre expansion with the Taylor expansion, we first consider the traditional Fourier stability analysis. Before replacing them in a Fourier mode, we should truncate each series at some finite order in order to develop a numerical scheme. Thus, we assume that truncate each series at the same order P, which gives us

$$\text{Taylor: } \sum_{p=0}^{P} \frac{(-\gamma \Delta k^2)^p}{p!}, \qquad \text{Laguerre: } \sum_{p=0}^{P} L_p(\beta^2)\hat{\mathcal{D}}^p, \quad \hat{\mathcal{D}} = 1 - \frac{1}{1+\gamma \Delta k^2},$$

where k is the wave number. Then, based on the previous identities (2.26) and (2.28),



(a) $e^{-\gamma \Delta t k^2}$ (b) Taylor ($P = 6$) (c) Laguerre ($P = 6$)

Figure 2.1: Fourier modes of (a) the expnential operator, (b) the truncated Taylor expansion ($P = 6$), and (c) the Truncated Laguerre expansion ($P = 6$). The common parameters $\gamma = 1$ and $\Delta t = 0.1$ is chosen.

two partial sums will converge to the same solution $e^{-\gamma \Delta t k^2}$, that is exponential decaying function in the Fourier mode. We specially chose $P = 6$ and draw each Fourier mode of the expansions in Figure 2.1. As shown in Figure 2.1b, the truncated Taylor expansion diverges for larger wave number k, however, the truncated Laguerre expansion at the same order P is nicely decaying, which as shown in Figure 2.1c. This is because the Fourier mode of convolution operator satisfies the boundedness, $|\hat{\mathcal{D}}| \leq 1$, whatever the wave number k is.

Because of this boundedness of our $\mathcal{D}$ operator, we do prefer the Laguerre expansion for our higher order timestepping scheme. Furthermore, this has a theoretical proof about the rate of convergence of partial sum, which will be discussed in the following Section.

2.3.1 Convergence

This expansion has been considered in the context of C_0- semigroups [33, 1], where $\left(-\frac{\partial_{xx}}{\alpha^2}\right)$ is replaced with a general closed operator A on a Hilbert space X. In our notation, we restate part (ii) of Theorem 5.2 in [1], which is proven therein.

Theorem 2.3.1. *Let the C_0- semigroup*

$$T(\beta^2) = e^{-\beta^2\left(-\frac{\partial_{xx}}{\alpha^2}\right)} = \sum_{p=0}^{\infty} L_p^{(-1)}(\beta^2)\mathcal{D}^p = \sum_{p=0}^{\infty} L_p^{(-1)}(\beta^2)\mathcal{L}^{-p}\left(\mathcal{L}-I\right)^p,$$
$$= \sum_{p=0}^{\infty} L_p^{(-1)}(\beta^2)\left(I-\frac{\partial_{xx}}{\alpha^2}\right)^{-p}\left(-\frac{\partial_{xx}}{\alpha^2}\right)^p,$$

where $L_p^{(-1)}$ are generalized Laguerre polynomials. This can be approximated by P^{th} partial sum of the series,

$$T_P(\beta^2) = \sum_{p=0}^{P} L_p^{(-1)}(\beta^2)\mathcal{D}^p.$$

Then, for $u \in C^{2P+2}(\Omega)$, there exists for each $\beta^2 > 0$ an integer m_0 such that for all integers $2 \le k \le P+1$, with $P \ge m_0$,

$$\left\| T(\beta^2)u - T_P(\beta^2)u \right\| \le \frac{c(\beta^2,k)}{P^{k/2-1}} \left\| \left(-\frac{\partial_{xx}}{\alpha^2}\right)^k u \right\|,$$

where $c(\beta^2,k)$ is a constant that depends only on β^2 and k.

Remark 5. *The salient point of the theorem is that, in consideration of α (c.f. Eqn. (2.3)), the approximation error is of the form $c\Delta t^{P+1}\left\|u^{(2P+2)}(x)\right\|$, which matches the form given by a typical Taylor method.*

Finally, we truncate the resolvent expansion (2.28) at order $p = P$. For the heat equation, this defines the numerical method as

$$u(x,t+\Delta t) = u(x,t) + \sum_{p=1}^{P} L_p^{(-1)}(\beta^2)\mathcal{D}^p[u](x,t), \tag{2.29}$$

which has a truncation error of the form

$$\tau := L_{P+1}^{(-1)}(\beta^2)\mathcal{D}^{P+1}[u](x,t) + \mathcal{O}(\Delta t^{P+2}). \tag{2.30}$$

For $P = 1, 2, 3$, these schemes (evaluated at $t = t^n$) are

$$u^{n+1} = u^n - \beta^2\mathcal{D}[u^n], \tag{2.31}$$

$$u^{n+1} = u^n - \beta^2\mathcal{D}[u^n] - \left(\beta^2 - \frac{\beta^4}{2}\right)\mathcal{D}^2[u^n], \tag{2.32}$$

$$u^{n+1} = u^n - \beta^2\mathcal{D}[u^n] - \left(\beta^2 - \frac{\beta^4}{2}\right)\mathcal{D}^2[u^n] - \left(\beta^2 - \beta^4 + \frac{\beta^6}{6}\right)\mathcal{D}^3[u^n], \tag{2.33}$$

where we used the Rodrigues formula for generalized Laguerre polynomials

$$\begin{aligned} L_0^{(-1)}(\beta^2) &= 1, \\ L_1^{(-1)}(\beta^2) &= -\beta^2, \\ L_p^{(-1)}(\beta^2) &= -\frac{ze^z}{p!}\frac{d^p}{dz^p}\left(e^{-z}z^{p-1}\right)\Big|_{z=\beta^2}, \quad p > 1. \end{aligned}$$

Now, the next natural question becomes how we can choose the parameter β for our time stepping method. This parameter actually guarentees the *stiff decay* of our numerical scheme. This will be discussed in the next Section.

2.3.2 Stability

There remains one critical issue: the choice of the free parameter β. In 1974, Nørsett studied a similar single-step multiderivative method for the heat equation [49] and he too, had a free parameter in his solver. We follow his lead on the Von-Neumann analysis based on his MOL

discretization, but in this work we optimize β to obtain stiff decay, whereas Nørsett chose β to maximize the order of accuracy of the solver.

Consider the linear test problem

$$\frac{dy}{dt} = \lambda y, \qquad y(0) = 1, \quad \lambda \in \mathbb{C},$$

whose exact solution $y(t)$ satisfies

$$y(t^n + \Delta t) = e^z y(t^n) \equiv \sum_{p=0}^{\infty} \frac{z^p}{p!} y(t^n), \qquad z = \lambda \Delta t \in \mathbb{C}.$$

On the other hand, our approximation solution y^{n+1} is obtained by P^{th} partial sum of the resolvent series (2.28). Application of (2.29) to this test problem results in

$$y^{n+1} = \sum_{p=0}^{P} L_p^{(-1)}(\beta^2) \bar{D}^p(z, \beta^2)\, y^n, \qquad \bar{D}(z, \beta^2) = 1 - \left(1 - \frac{z}{\beta^2}\right)^{-1} = \frac{z}{z - \beta^2}.$$

In other words, we can express our approximation as

$$y^{n+1} = \phi_{P,\beta}(z) y^n = \left(\phi_{P,\beta}(z)\right)^2 y^{n-1} = \cdots = \left(\phi_{P,\beta}(z)\right)^{n+1},$$

where $\phi_{P,\beta}(z)$ is the stability function of the presented P^{th}-order time stepping method,

$$\phi_{P,\beta}(z) = \sum_{p=0}^{P} L_p^{(-1)}(\beta^2) \bar{D}^p(z, \beta^2), \tag{2.34}$$

which depends on the free parameter β (determined later) and the temporal order P.

Following standard definitions, we say that a numerical scheme is *A-stable*, provided

$|\phi| \leq 1$ in the left-half of the complex plane $z \in \mathbb{C}^-$. Likewise, a scheme exhibits *stiff decay* if $\phi(z) \to 0$ as $Re(z) \to -\infty$. If an A-stable method also exhibits stiff decay, it is L-stable. We first note the following recurrence relation of generalized Laguerre polynomials, in order to calculate the limit of stability functions at negative infinity. The generalized Laguerre polynomials satisfy many identities, the following of which is the most pertinent:

$$L_{p+1}^{(0)}(x) - L_p^{(0)}(x) = L_{p+1}^{(-1)}(x) = \left(\frac{x}{p+1}\right) \frac{d}{dx} L_{p+1}^{(0)}(x). \tag{2.35}$$

Here, $L_p^{(0)}(x)$ is the standard Laguerre polynomial $L_p(x)$.

Now, observing that $\bar{D} \to 1$ as $Re(z) \to -\infty$, we find that

$$\lim_{Re(z)\to-\infty} \phi_{P,\beta}(z) = \sum_{p=0}^{P} L_p^{(-1)}(\beta^2) = L_0^0(\beta^2) + \sum_{p=1}^{P} \left(L_p^0(\beta^2) - L_{p-1}^0(\beta^2) \right) = L_P^0(\beta^2), \tag{2.36}$$

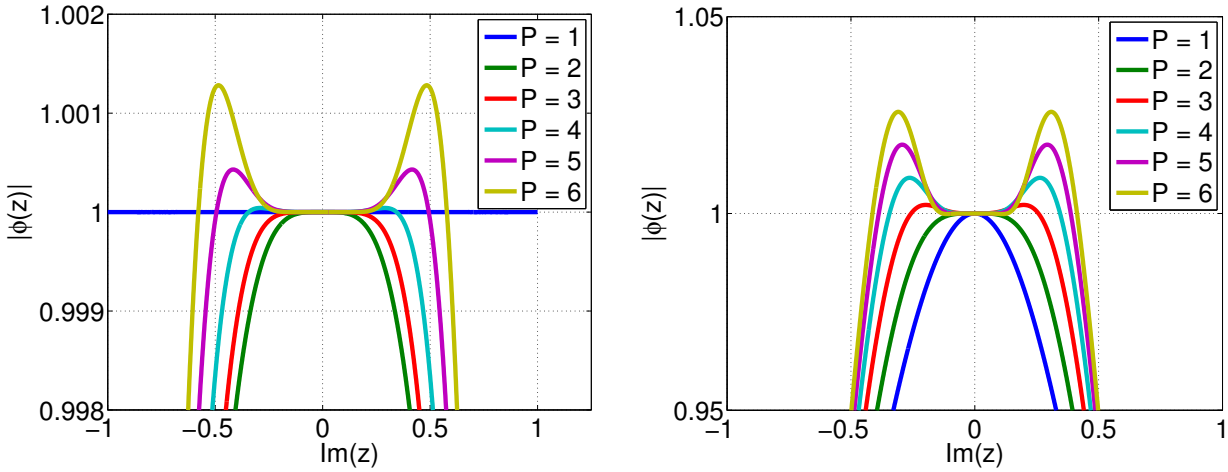
where we have used the first two expressions in (2.35) to introduce a telescoping sum. We are now prepared to prove the following.

Theorem 2.3.2. *Let $u(x,t)$ be an approximate solution to the heat equation* (2.2)*, given by the successive convolution scheme* (2.29)*. Then,*

1. *If $\beta^2 = x_1^{(P)}$ is chosen as the smallest root of $L'_{P+1}(x) = (L_{P+1}^{(0)}(x))'$, then the scheme achieves order $P+1$, but does not exhibit stiff decay.*

2. *If $\beta^2 = x_1^{(P)}$ is chosen as the smallest root of $L_P(x) = L_P^{(0)}(x)$, then the scheme achieves order P, and exhibits stiff decay.*

3. *Following the first strategy, the schemes are A-stable for $P = 1, 2, 3$, whereas the second strategy ensures L-stability. For both strategies, $A(\alpha)$-stability is achieved for $P > 3$,*



(a) Maximal Order (b) Stiff Decay

Figure 2.2: Maximum amplification factors $|\phi(iy)|$ for the first few orders P, with (a) maximal order, or (b) stiff decay. When maximizing order, the first 3 schemes exhibit A-stability, whereas ensuring stiff decay leads to L-stable schemes. For $P > 3$, both schemes become $A(\alpha)$-stable.

with large values of $\alpha \approx \pi/2$.

Proof. The proof follows by applying the maximum modulus principle coupled with (2.36). For part 1, upon examining the truncation error (2.30), we see that an additional order of accuracy can be gained if we choose

$$L_{P+1}^{(-1)}(\beta^2) = \left(\frac{\beta^2}{P+1}\right) L'_{P+1}(\beta^2) = 0.$$

However, $L_P(\beta^2) \neq 0$ for this choice, and so stiff decay does not hold. For part 2, we instead enforce stiff decay, but then the truncation error is of order P. Finally, part 3 is demonstrated by the maximum amplification factors ϕ along the imaginary axis, as shown for both strategies in Figure 2.2. In particular, we observe that $|\phi_{P,\beta}(iy)| \leq 1$ for $P = 1, 2, 3$.

	Stiff Decay		Maximal Order	
P	β^2	$L_P(\beta^2)$	β^2	$L_P(\beta^2)$
1	1.0000	0	2.0000	-1.0000
2	0.5858	0	1.2679	-0.7320
3	0.4158	0	0.9358	-0.6304
4	0.3225	0	0.7433	-0.5768
5	0.2636	0	0.6170	-0.5436
6	0.2228	0	0.5277	-0.5211

Table 2.1: Values of β^2 chosen for orders $P = 1, 2, \ldots 6$. The first column are those used in our schemes, and uniquely guarantee stiff decay and $A(0)$-stability. For comparison, we also display the values in Nørsett [49] which give optimal order $P+1$, at the expense of stiff decay.

Specifically, the explicit forms of stability functions are

$$\phi_{1,\beta} = 1 - \beta^2 \left(\frac{z}{z - \beta^2} \right), \qquad \phi_{2,\beta} = \frac{\left(1 - 2\beta^2 + \beta^4/2\right) z^2 + \left(\beta^4 - 2\beta^2\right) z + \beta^4}{\left(z - \beta^2\right)^2},$$
$$\phi_{3,\beta} = \frac{\left(1 - 3\beta^2 + \frac{3}{2}\beta^4 - \frac{1}{6}\beta^6\right) z^3 + \left(-3\beta^2 + 3\beta^4 - \frac{1}{2}\beta^6\right) z^2 + (3\beta^4 - \beta^6) z - \beta^6}{\left(z - \beta^2\right)^3}.$$

□

Remark 6. *In [49], the scheme was chosen to maximize the order of accuracy, implicitly leading to eliminating the first term in the truncation error* (2.30)*, which is equivalent to the first strategy. However, in this work we follow the second strategy, and choose* β^2 *as the smallest root of* $L_P(x)$ *to ensure stiff decay.*

For comparison we record the values of β^2 chosen for each order $1 \leq P \leq 6$, to those of Nørsett in Table 2.1. For all of our solvers, we choose β to be the largest possible value that still yields provable stiff decay.

	$P=1$		$P=2$		$P=3$	
Δt	L^∞ error	order	L^∞ error	order	L^∞ error	order
0.1	1.8405×10^{-4}	–	1.6255×10^{-6}	–	2.4225×10^{-8}	–
0.05	9.2121×10^{-5}	0.9985	4.0841×10^{-7}	1.9928	3.0620×10^{-9}	2.9839
0.025	4.6084×10^{-5}	0.9993	1.0236×10^{-7}	1.9964	3.8501×10^{-10}	2.9915
0.0125	2.3048×10^{-5}	0.9996	2.5622×10^{-8}	1.9982	4.8402×10^{-11}	2.9918
0.00625	1.1525×10^{-5}	0.9998	6.4097×10^{-9}	1.9990	6.2021×10^{-12}	2.9642

Table 2.2: Refinement studies for 1D Heat equation (2.2) ($(x,t) \in (0, 2\pi) \times [0, 4]$) with Periodic boundary conditions. P indicates the order of resolvent expansion in (2.28).

2.3.3 Numerical result: Refinement study of 1D Heat equation

We first illustrate the accuracy of our method for the 1D heat equation defined in (2.2). We consider initial conditions $u(x, 0) = \sin(x)$, for $x \in [0, 2\pi]$ with periodic boundary conditions. We integrate up to a final time of $T = 4$, and set $\gamma = 0.18^2$. We use the fast convolution algorithm that is fourth order accurate in space ($M = 4$), and set the spatial grid size to be $\Delta x = \frac{2\pi}{1024} \approx 0.0061$. This ensures that the dominant error in the solution is temporal. We compute errors by the L^∞-norm, and compare against the exact solution

$$u(x, T) = e^{-\gamma T} u_0(x) \equiv e^{-\gamma T} sin(x). \tag{2.37}$$

The result of a temporal refinement study for $P = 1, 2$ and 3 is presented in Table 2.2, computed at $T = 4$. This example serves to illustrate that our scheme achieves the designed order of accuracy after appropriate truncations of the resolvent expansions.

2.4 MOL^T for Multi-D Heat equation

We extend the 1D solver to multiple spatial dimensions through the use of dimensional splitting. Our key observation is that we can use the factorization property of the exponential

to perform the series expansion. For instance, in three dimensions, we have

$$e^{\gamma \Delta t \nabla^2} = e^{-\beta^2\left(-\frac{\partial_{xx}}{\alpha^2}\right)} e^{-\beta^2\left(-\frac{\partial_{yy}}{\alpha^2}\right)} e^{-\beta^2\left(-\frac{\partial_{zz}}{\alpha^2}\right)}. \tag{2.38}$$

Now, we first replace each term with the identity (2.28) dimension by dimension, and then truncate the expansions which will be in terms of the univariate operators $\mathcal{L}_\gamma^{-1}$ and $\mathcal{D}_\gamma$ for $\gamma = \{x, y, z\}$ as defined by (2.3), and (2.29) acting on a function $u^n(x, y, z)$. This infinite sum with three indices must then be truncated to order P, and after a change of indices we find

$$E_P = \sum_{p=0}^{P} \sum_{q=0}^{P-p} \sum_{r=0}^{P-(p+q)} L_p^{(-1)}(\beta^2) L_q^{(-1)}(\beta^2) L_r^{(-1)}(\beta^2) \mathcal{D}_x^p \mathcal{D}_y^q \mathcal{D}_z^r, \tag{2.39}$$

in 3D, with the corresponding 2D operator given by

$$E_P = \sum_{p=0}^{P} \sum_{q=0}^{P-p} L_p^{(-1)}(\beta^2) L_q^{(-1)}(\beta^2) \mathcal{D}_x^p \mathcal{D}_y^q. \tag{2.40}$$

Here we adopt that sums are taken over all non-negative indices that sum to P.

2.4.1 Stability

The proof of stability for the multi-dimensional algorithm follows directly from that of the one-dimensional case, with the same approach applied to each spatial dimension. For in-

	$P=1$		$P=2$		$P=3$	
Δt	L^∞-error	order	L^∞-error	order	L^∞-error	order
0.1	9.8182×10^{-5}	–	8.6717×10^{-7}	–	1.2925×10^{-8}	–
0.05	4.9143×10^{-5}	0.9985	2.1788×10^{-7}	1.9928	1.6354×10^{-9}	2.9825
0.025	2.4584×10^{-5}	0.9992	5.4608×10^{-8}	1.9963	2.0791×10^{-10}	2.9756
0.0125	1.2295×10^{-5}	0.9996	1.3672×10^{-8}	1.9979	2.9204×10^{-11}	2.8317

Table 2.3: Refinement study for a 2D Heat equation ($(x, y) \in (0, 2\pi) \times (0, 2\pi)$) with Periodic boundary conditions, ($T = 1$).

stance, for the 2D case, the stability function can be defined as

$$\phi(z) = \sum_{p=0}^{P} \sum_{q=0}^{P-p} L_p^{(-1)}(\beta^2) L_q^{(-1)}(\beta^2) \bar{D}_x^p \bar{D}_y^q, \qquad \bar{D}_x = \bar{D}_y = 1 - \left(1 - \frac{z}{\beta^2}\right)^{-1},$$

thus, similar with the 1D stability function,

$$\lim_{Re(z) \to -\infty} \phi(z) = \left(L_P^0(\beta^2)\right)^2,$$

where we have used that $\bar{D}_x$, $\bar{D}_y$ approach to 1 as $Re(z) \to -\infty$ and telescoping sum of Laguerre polynomial. 3D case is also similar with low spatial dimension case.

2.4.2 Numerical result: Refinement study of 2D Heat equation

As a second example, we present results for the 2D heat equation. We consider initial conditions $u(x, y, 0) = \sin(x)\sin(y)$, for $(x, y) \in [0, 2\pi] \times [0, 2\pi]$ with periodic boundary conditions. We use a uniform mesh of size $\Delta x = \Delta y = 2\pi/512 \approx 0.0123$. Likewise, the L^∞-error is computed by comparing against the exact solution $u(x, y, T) = e^{-2\gamma T} u_0(x, y)$ at the final time $T = 1$. In Table 2.3, we present results for a temporal refinement study for orders $P = 1, 2$, and 3.

2.5 MOL^T for Nonlinear equations

We next extend our method to nonlinear reaction-diffusion systems of the form

$$\mathbf{u}_t = \mathbf{D}\nabla^2\mathbf{u} + \mathbf{F}(\mathbf{u}), \quad (\mathbf{x}, t) \in \Omega \times (0, T], \tag{2.41}$$

where $\mathbf{u} = (u_1, u_2, \cdots, u_N)$, with $u_i = u_i(\mathbf{x}, t)$, $\mathbf{D}$ is a diffusion coefficient matrix, and the reaction term $\mathbf{F} := (f_1, f_2, \cdots, f_N)$ is a function of u_i, $(i = 1, 2, \cdots, N)$. In the above, $\Omega \subset \mathbb{R}^N$ is a bounded domain, and we assume appropriate initial values and boundary conditions. We shall view the diffusion as being the linear part of the differential operator, and invert this linear part analytically, using successive convolution. To derive the scheme, we use operator calculus to first write

$$\left(\partial_t - \mathbf{D}\nabla^2\right)\mathbf{u} = \mathbf{F} \quad \Longrightarrow \quad \left(e^{-\mathbf{D}t\nabla^2}\mathbf{u}\right)_t = e^{-\mathbf{D}t\nabla^2}\mathbf{F}, \tag{2.42}$$

where $e^{-\mathbf{D}t\nabla^2}$ is a pseudo-differential operator. Upon integrating (2.42) over the interval $[t, t + \Delta t]$, we arrive at the update equation

$$\begin{aligned}\mathbf{u}(t + \Delta t) - e^{\mathbf{D}\Delta t\nabla^2}\mathbf{u}(t) &= \int_t^{t+\Delta t} e^{\mathbf{D}(t+\Delta t-\tau)\nabla^2}\mathbf{F}(\tau)d\tau \\ &= \int_0^{\Delta t} e^{\mathbf{D}(\Delta t-\tau)\nabla^2}\mathbf{F}(t + \tau)d\tau,\end{aligned} \tag{2.43}$$

where we have made use of the abbreviated notation, $\mathbf{F}(t) := \mathbf{F}(\mathbf{u}(\mathbf{x}, t))$. On the left hand side, the diffusion terms have been collected by this pseudo-differential operator, and will be approximated using the successive convolution techniques developed above. The reaction terms on the right hand side (2.43) are fully nonlinear, and we must consider nonlinear

stability when choosing a method of discretization.

We first consider approximating the integral on the right hand side (2.43) with the trapezoidal rule. This defines a single-step update equation, which will be second order accurate

$$\mathbf{u}(t+\Delta t) - e^{\mathbf{D}\Delta t\nabla^2}\mathbf{u}(t) = \frac{\Delta t}{2}\left[e^{\mathbf{D}\Delta t\nabla^2}\mathbf{F}(t) + \mathbf{F}(t+\Delta t)\right]. \tag{2.44}$$

We may also obtain a single-step third order scheme, using multiderivative integration [53]. By replacing the integrand (2.43) with a third order Hermite-Birkhoff interpolant and performing exact integration of the resulting function, we arrive at

$$\mathbf{u}(t+\Delta t) - e^{\mathbf{D}\Delta t\nabla^2}\mathbf{u}(t) = e^{\mathbf{D}\Delta t\nabla^2}\left[\frac{2\Delta t}{3}\mathbf{F}(t) + \frac{\Delta t}{6}\left(-\mathbf{D}\Delta t\nabla^2\mathbf{F}(t) + \Delta t\frac{d\mathbf{F}}{d\mathbf{t}}(t)\right)\right] + \frac{\Delta t}{3}\mathbf{F}(t+\Delta t), \tag{2.45}$$

where $\frac{d\mathbf{F}}{dt}(t) = \frac{d\mathbf{F}}{d\mathbf{u}}(t)\cdot(\mathbf{D}\nabla^2\mathbf{u}(t) + \mathbf{F}(t))$. The Hermite-Birkhoff interpolant that matches the integrand in (2.43) at times $\tau = 0$, and $\tau = \Delta t$, as well as its derivative at time $\tau = 0$ produces the quadrature rule in (2.45).

Upon perusing the third order update equation (2.45), we will need to use successive convolution to replace the psuedo-differential operator $\exp\left(\mathbf{D}\Delta t\nabla^2\right)$, as well as the Laplacian operator ∇^2. This latter point has been detailed in [13], and so we comment briefly on it here. Using the one-dimensional expansion (2.25), we observe that the two-dimensional Laplacian is similarly given by

$$-\frac{\nabla^2}{\alpha^2} = -\frac{\partial_{xx}}{\alpha^2} - \frac{\partial_{yy}}{\alpha^2} = \sum_{p=1}^{\infty}\left(\mathcal{D}_x^p + \mathcal{D}_y^p\right),$$

and can be truncated at the appropriate accuracy $p = P$. Here, the subscripts indicate that the convolution is only in one spatial direction, and the other variable is held fixed. Thus,

$\mathcal{D}_x$ is applied along horizontal lines for fixed y-values, and likewise for $\mathcal{D}_y$.

Remark 7. *The proposed schemes in* (2.44) *and* (2.45) *produce nonlinear equations for* $\mathbf{u}(x, t + \Delta t)$ *that need to be solved at each time step. Therefore, any implicit solver will necessarily be problem dependent.*

For the problems examined in this work, we make use of simple fixed-point iterative schemes. We stabilize our iterative solvers by linearizing $\mathbf{F}(\mathbf{u})$ about a background state $\mathbf{F}_{\mathbf{u}}(\mathbf{u}^*)$, which depends on the problem under consideration.

2.5.1 Allen-Cahn Equation

We examine in greater detail the application of our schem to the Allen-Cahn (AC) equation [3],

$$u_t = \epsilon^2 \nabla^2 u + f(u), \qquad (x, t) \in \Omega \times (0, T], \tag{2.46}$$

where the reaction term is $f(u) = u - u^3$, and $\Omega \subset \mathbb{R}^d$ is a bounded domain, and u satisfies homogeneous Neumann boundary conditions.

For our fixed point iteration, we linearize f about the stable fixed points $u^* = \pm 1$, which satisfy $f'(u^*) = 0$. For example, the second order scheme from (2.44) becomes

$$(1 + \Delta t)\, u^{n+1,k+1} = e^{\epsilon^2 \Delta t \nabla^2} \left(u^n + \frac{\Delta t}{2} f^n \right) + \frac{\Delta t}{2} \left(f^{n+1,k} + 2u^{n+1,k} \right), \tag{2.47}$$

where n indicates the time step as before, and now k is the iteration index. By lagging the nonlinear term $f^{n+1,k}$, the fixed point update is made explicit. Likewise, the third order scheme from (2.45) becomes

	$P=1$		$P=2$		$P=3$	
Δt	L^∞ error	order	L^∞ error	order	L^∞ error	order
0.025	2.8216×10^{-4}	–	1.3895×10^{-5}	–	2.6060×10^{-6}	–
0.0125	1.4419×10^{-4}	0.9686	3.6115×10^{-6}	1.9439	3.9417×10^{-7}	2.7249
0.0063	7.2874×10^{-5}	0.9845	9.2164×10^{-7}	1.9703	5.5010×10^{-8}	2.8411
0.0031	3.6632×10^{-5}	0.9923	2.3294×10^{-7}	1.9842	7.3122×10^{-9}	2.9113
0.0016	1.8365×10^{-5}	0.9961	5.8695×10^{-8}	1.9886	9.5714×10^{-10}	2.9335

Table 2.4: Refinement study for the 1D Allen-Cahn (AC) equation with an initial $u_{AC}(x,0)$ in (2.49) and with the homogeneous Neumann boundary conditions.

$$\left(1+\frac{2\Delta t}{3}\right)u^{n+1,k+1} = e^{\epsilon^2\Delta t\nabla^2}\left[u^n+\frac{2\Delta t}{3}f^n+\frac{\Delta t}{6}\left(-\epsilon^2\Delta t\nabla^2 f^n+\Delta t f_t^n\right)\right] + \frac{\Delta t}{3}\left(f^{n+1,k}+2u^{n+1,k}\right). \tag{2.48}$$

Here, $e^{\epsilon^2\Delta t\nabla^2}$ is again understood by replacing it with a resolvent expansion, which is a truncated series of successive convolution operators.

2.5.1.1 Numerical result: 1D Allen-Cahn test

We demonstrate the accuracy of our proposed schemes by simulating a well-known traveling wave solution [14, 54, 44],

$$u_{AC}(x,t) = \frac{1}{2}\left(1-\tanh\left(\frac{x-T_s-st}{2\sqrt{2}\epsilon}\right)\right), \qquad x\in\Omega=[0,4], \quad 0\le t\le T. \tag{2.49}$$

Here, $s = \frac{3\epsilon}{\sqrt{2}} = 0.09$ is the speed of the traveling wave, and we choose $\epsilon = 0.03\sqrt{2}$. Additionally, we choose the delay time $T_s := 1.5 - sT$, so that the exact solution satisfies $u_{AC}(1.5,T) = 0.5$. Results for a final time of $T = 8$ are shown in Figure 2.3, with two different time steps. The solutions agree well with the exact solution.

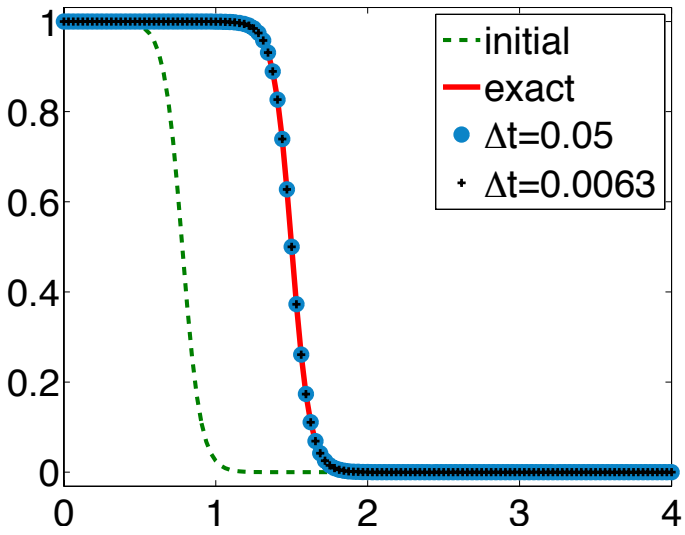


Figure 2.3: The numerical solutions with two different time step sizes, $\Delta t = 0.05$ and $\Delta t = 0.0063$ (common parameters: $\Delta x = 2^{-6}$ and $\epsilon = 0.03\sqrt{2}$). Solid (red) line: the exact profile in (2.49) at $T = 8$.

In Table 2.4, we present the L^∞-error in the numerical solution at a final time $T = 1$, using the exact solution $u_{AC}(x, T)$ (2.49). We observe first order accuracy from the Backward Euler method, and the expected orders of accuracy from the second (2.47) and third (2.48) order schemes. To ensure that the temporal error is dominant, we have used the fourth order accurate scheme (eq. (2.14) with $M = 4$), with $\Delta x = 2^{-9}$ to perform spatial integration in the successive convolutions.

In principle, we can achieve higher orders accuracy in space and time. The latter would require using higher order Hermite-Birkhoff interpolation to discretize the reaction term in (2.43).

2.5.1.2 Numerical result: 2D Allen-Cahn test

We next solve the Allen-Cahn equation in two spatial dimensions. A standard benchmark problem involves the motion of a circular interface [14, 54, 44], to which an exact solution is

known in the limiting case $\epsilon \to 0$. The radially symmetric initial conditions are defined by

$$u(x, y, 0) = \tanh\left(\frac{0.25 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right), \tag{2.50}$$

which has an interfacial circle ($u(x, y, 0) = 0$) centered at $(0.5, 0.5)$, with a radius of $R_0 = 0.25$. This interfacial circle is unstable, and will shrink over time, as determined by the mean curvature [3],

$$V = \frac{dR}{dt} = -\frac{1}{R}. \tag{2.51}$$

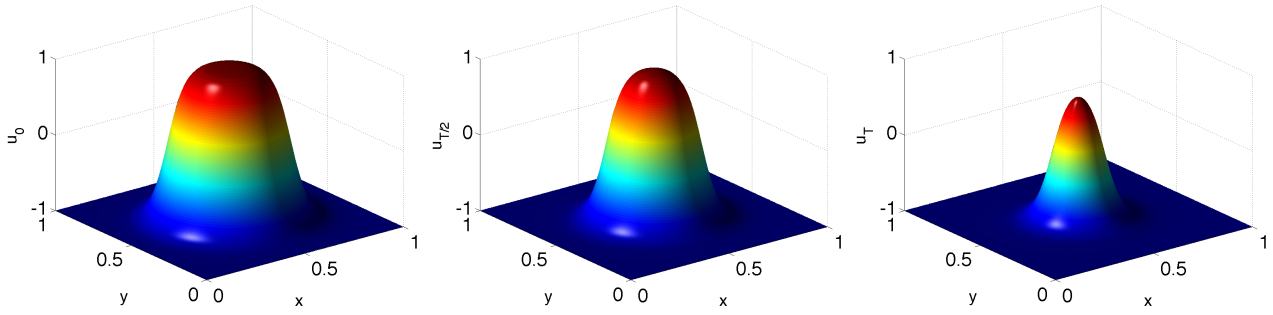
Here V is the velocity of the moving interface, and R is the radius of the interfacial circle at time t (i.e., it is the radius of the curve defined by $u(x, y, t) = 0$). By integrating (2.51) with respect to time, we solve for the radius as a function of time

$$R(t) = \sqrt{R_0^2 - 2\epsilon^2 t}. \tag{2.52}$$

The location where ϵ is placed in equation (2.46) differs from other references [14, 44, 54]. Therefore, we point out that our time scales have been appropriately rescaled for comparison.

The moving interface problem was simulated using $\epsilon = 0.05$, $\Delta t = \frac{6.4 \times 10^{-4}}{\epsilon^2} = 0.0256$, and $\Delta x = \Delta y = 2^{-8} \approx 0.0039$, which are based on the parameters used in [44]. The numerical solution is displayed in Figure 2.4, and we observe that the interfacial circle shrinks, as is expected.

In Figure 2.5, we plot compare the evolution of the radii obtained by our second order scheme with the exact radius (2.52), for two different values of the diffusion parameter ϵ. The radius is measured by taking a slice of the solution along $y = 0$, and then solving for the spatial point where $u = 0$ using linear interpolation between the two closest points that



(a) $u(x, y, 0)$ (b) $u(x, y, \frac{T}{2})$ (c) $u(x, y, T)$

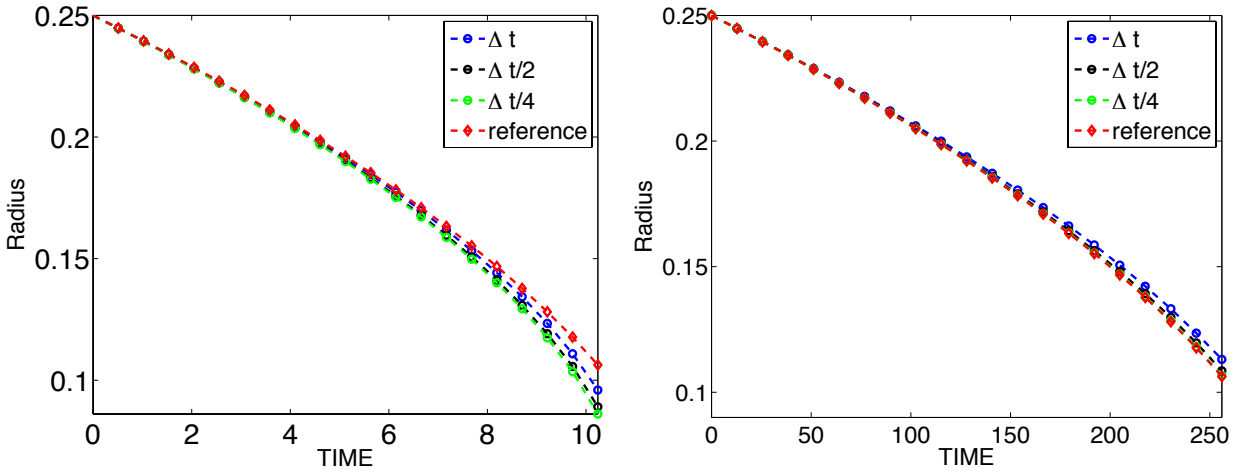
Figure 2.4: Time evolution of the 2D AC numerical solution with initial condition (2.50), up to time $T = \frac{0.0256}{\epsilon^2} = 10.24$. (parameters: $\Delta t = 0.0256, \Delta x = \Delta y = 0.0039, \epsilon = 0.05$)

satisfy $u(x, 0, t) < 0$ and $u(x, 0, t) > 0$. Refinement is performed with a fixed spatial mesh $\Delta x = \Delta y = 2^{-8}$, and time steps of $\Delta t = 0.2560, 0.1280$, and 0.0640. Because the radius is derived as an exact solution in the limit (i.e., $\epsilon \to 0$) [3], we observe that the smaller value of ϵ is indeed more accurate.

We next perform a refinement study for the Allen-Cahn equations, but this time in two spatial dimensions. To do so, we must incorporate the multivariate successive convolution algorithms in (2.40) and (2.25) into the second (2.47) and third (2.48) order schemes. Given that we do not have an exact solution, we compute successive errors in an L^∞-norm. That is, we compute $||u_{\Delta t} - u_{\frac{\Delta t}{2}}||_\infty$ for each time step Δt. Results are as expected, and are presented in Table 2.5. The parameters used are $\epsilon = 0.05$, $\Delta x = \Delta y = 2^{-9}$, and the final computation time is $T = 0.5$. Again, the quadrature method is fourth order accurate in space, so that the dominant source of error is temporal.

2.5.2 FitzHugh-Nagumo System

Finally, we solve a well known reaction diffusion system that arises in the modeling of neurons, the Fitzhugh-Nagumo (FHN) model [28, 39]. The FHN system consists of an



(a) $\epsilon = 0.05$ (b) $\epsilon = 0.01$

Figure 2.5: Radii of the numerical interfacial circle as a function of time ($0 \leq t \leq \frac{0.0256}{\epsilon^2}$) with various time step sizes. (a) $\epsilon = 0.05$ (b) $\epsilon = 0.01$ (common parameter: $\Delta x = 2^{-8}$). Red line: the exact radius R in (2.52).

	$P = 1$		$P = 2$		$P = 3$	
Δt	L^∞ error	order	L^∞ error	order	L^∞ error	order
0.0063	6.5941×10^{-4}	–	1.1740×10^{-4}	–	5.1744×10^{-6}	–
0.0031	3.3065×10^{-4}	0.9959	3.2637×10^{-5}	1.8468	7.8351×10^{-7}	2.7234
0.0016	1.6563×10^{-4}	0.9973	8.6726×10^{-6}	1.9120	1.0811×10^{-7}	2.8574
0.0008	8.2894×10^{-5}	0.9987	2.2389×10^{-6}	1.9537	1.2961×10^{-8}	3.0602

Table 2.5: Refinement studies for 2D Allen-Cahn equataion with the initial (2.50) and with homogeneous Neumann boundary conditions.

activator u and an inhibitor v, which are coupled via nonlinear reaction diffusion equations

$$\begin{aligned} u_t &= D_u \nabla^2 u + \frac{1}{\delta} h(u, v), \\ v_t &= D_v \nabla^2 v + g(u, v), \end{aligned} \tag{2.53}$$

where D_u, D_v are the diffusion coefficients for u and v, respectively, and $0 < \delta \ll 1$ is a real parameter. We use the classical cubic FHN local dynamics [39], that are defined as

$$\begin{aligned} h(u,v) &= Cu(1-u)(u-a) - v, \\ g(u,v) &= u - dv, \end{aligned} \tag{2.54}$$

where C, a and d are dimensionless parameters. The parameters we use are the same as in [19, 50]: $D_u = 1$, $D_v = 0$, $a = 0.1$, $C = 1$, $d = 0.5$, and $\delta = 0.005$. The diffusion coefficient for the inhibitor is $D_v = 0$, identical to the work found in [41, 59].

The second order scheme from (2.47) is applied to each variable u and v separately. This defines the numerical scheme as

$$\begin{aligned} u^{n+1} &= e^{\Delta t \nabla^2}\left(u^n + \frac{\Delta t}{2\delta}h^n\right) + \frac{\Delta t}{2\delta}h^{n+1}, \\ v^{n+1} &= \left(v^n + \frac{\Delta t}{2}g^n\right) + \frac{\Delta t}{2}g^{n+1}, \end{aligned} \tag{2.55}$$

where $h^n = h(u^n, v^n)$ and $g^n = g(u^n, v^n)$. We again use a stabilized fixed point iteration to address the nonlinear reaction terms. Because $(u^*, v^*) = (0, 0)$ is the only stable excitable fixed point of equation (2.53) simply construct the Jacobian of $\mathbf{F} = (h, g)$ of (2.54) about this point:

$$\mathcal{J}_{\mathbf{F}}(u^*, v^*) \cdot \begin{bmatrix} u - u^* \\ v - v^* \end{bmatrix} \equiv \frac{\partial(h,g)}{\partial(u,v)}|_{(0,0)} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -Ca & -1 \\ 1 & -d \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -Cau - v \\ u - dv \end{bmatrix}. \tag{2.56}$$

The resulting second order scheme is

$$\begin{bmatrix} 1+\frac{Ca\Delta t}{2\delta} & \frac{\Delta t}{2\delta} \\ -\frac{\Delta t}{2} & 1+\frac{d\Delta t}{2} \end{bmatrix} \begin{bmatrix} u^{n+1,k+1} \\ v^{n+1,k+1} \end{bmatrix} = \begin{bmatrix} e^{\Delta t\nabla^2}\left(u^n+\frac{\Delta t}{2\delta}h^n\right) \\ v^n+\frac{\Delta t}{2}g^n \end{bmatrix} + \frac{\Delta t}{2}\begin{bmatrix} \frac{1}{\delta}\left(h^{n+1,k}+Cau^{n+1,k}+v^{n+1,k}\right) \\ g^{n+1,k}-u^{n+1,k}+dv^{n+1,k} \end{bmatrix}, \tag{2.57}$$

where k is the iteration number, and n is the time level.

2.5.2.1 Numerical result: 2D FitzHugh-Nagumo test

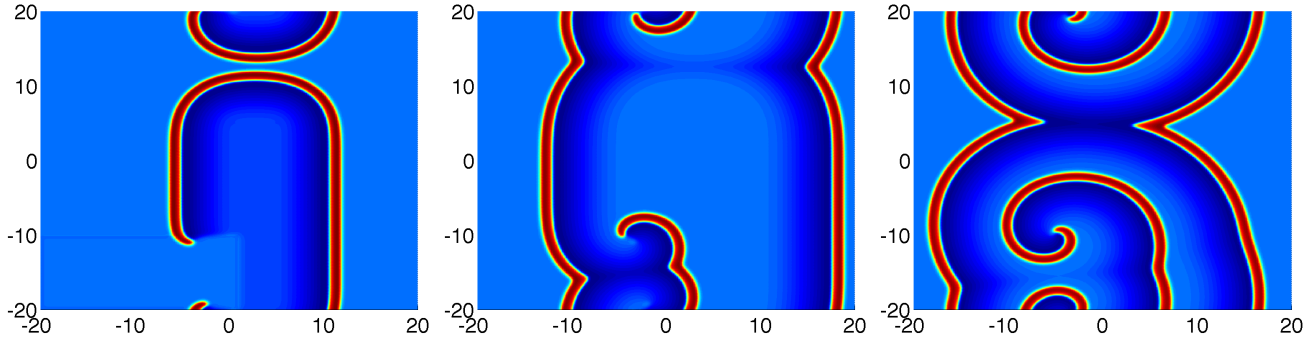
We solve the FitzHugh-Nagumo system over the domain $\Omega = [-20, 20] \times [-20, 20]$ with Periodic boundary conditions. Using initial conditions

$$u(x,y,0) = \begin{cases} 0 & \text{if } \{x<0\}\cup\{y>5\} \\ \frac{1}{(1+e^{4(|x|-5)})^2}+\frac{1}{(1+e^{4(|x|-1)})^2} & \text{otherwise} \end{cases} \tag{2.58}$$

and

$$v(x,y,0) = \begin{cases} 0.15 & \text{if } \{x<1\}\cup\{y<-10\} \\ 0 & \text{otherwise} \end{cases} \tag{2.59}$$

we present the numerical evolution of the activator u in Figure 2.6. We observe similar spiral waves that form in other recent work from the literature [19].



(a) $T = 1$ (b) $T = 2$ (c) $T = 4$

Figure 2.6: Contour plot: Temporal evolution of the concentration of activator u using the second-order scheme in (2.57) with parameters: $\Delta t = 0.01, \Delta x = \Delta y = 0.2, M = 4$ (fourth order in space).

Chapter 3

MOL^T for Cahn-Hilliard Equation

3.1 Models

In this Chapter, we present the proposed method of lines transpose (MOL$^{\mathrm{T}}$) scheme for Cahn-Hilliard (CH) equation

$$u_t = -\Delta \left[\epsilon^2 \Delta u - f(u) \right], \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \tag{3.1}$$

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 eqution is replaced by ∂_{xx}.

In 1893, van der Waals first developed a local free energy density function of binary mixtures, which became a key component in a phase-field model. Over decades, in 1950s, Cahn and Hilliard rediscovered this functions role and extended this concept to model spinodal decomposition based on their analysis. Since then, this continuum equation, Cahn-Hilliard nonlinear diffusion equation, has been extensively used to predict the evolution of arbitrary morphologies and complex microstructures without explicitly tracking the positions of inter-

faces [15].

The CH equation (3.1) describes the H^{-1} gradient flow of the CH free energy [8]

$$\mathcal{E}(u) = \int_\Omega \frac{\epsilon^2}{2}|\nabla u|^2 + F(u)d\mathbf{x}. \tag{3.2}$$

Subject to zero-flux boundary condition, the CH equation dissipates the CH free energy

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

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 [26], the semi-implicit spectral deferred correction method by Shen, [54, 46], and even fully implicit schemes utilizing nonlinear solvers, such as the conjugate gradient method [16]. 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 $u = -1$.

In addition, we also consider vector version of CH (VCH) equation [16]. For $\mathbf{u} = (u_1, u_2)$,

$$\mathbf{u}_t = -\Delta \left[\epsilon^2 \Delta \mathbf{u} - \nabla_{\mathbf{u}} W(\mathbf{u})\right] \tag{3.3}$$

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

$$W(\mathbf{u}) = \Pi_{i=1}^3 |\mathbf{u} - \mathbf{z}_i|^2, \qquad \mathbf{z}_i : \text{cube roots of unity in the } (u_1, u_2) \text{ plane.} \tag{3.4}$$

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 Ginzber-Landau equation [5] , 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 (3.2) such that

$$\mathcal{E}_{VCH}(\mathbf{u}) = \int_\Omega \frac{\epsilon^2}{2}|\nabla \mathbf{u}|^2 + W(\mathbf{u})d\mathbf{x}. \tag{3.5}$$

3.1.1 transformed model

Before employing a time discretization to CH equation (3.1), 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 [18], in which the same approach is combined with operator splitting to produce a contractive operator. Similarly, in [16] 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,

$$v_t = -\epsilon^2 v_{xxxx} + f(v-1)_{xx}, \quad f(v-1) = v^3 - 3v^2 + 2v, \tag{3.6}$$

which follows from the transformation $v = u + 1$ inserted into the CH equation (3.1), with $\Omega = (a, b)$. Similarly, the energy functional (3.2) becomes

$$\mathcal{E}(v) = \int_a^b \frac{\epsilon^2}{2}|v_x|^2 + F(v-1)d\mathbf{x}, \quad F(v-1) = \frac{1}{4}v^4 - v^3 + v^2 \tag{3.7}$$

In Section 3.2, we will formulate our first-order MOL$^{\mathrm{T}}$ scheme using above equation (3.6), and will show that this transformation makes our scheme gradient stable in Section 3.2.1.

In the previous Chapter 2, we developed MOL$^{\mathrm{T}}$ scheme for Heat equation and Allen-Cahn equation. We will revisit this previous work when necessary.

3.2 First order scheme for 1D CH equation

We utilize the MOL^{T} by discretizing (3.6) in time as the Backward Euler (BE) scheme,

$$\frac{v^{n+1}-v^{n}}{\Delta t} = -\epsilon^{2}\partial_{xxxx}v^{n+1} + 2\partial_{xx}v^{n+1} + \partial_{xx}\tilde{f}^{n+1}, \quad \tilde{f}(v) = v^{3} - 3v^{2}, \tag{3.8}$$

where $v^{n} = u(x, t^{n}) + 1$ and $\Delta t = t^{n+1} - t^{n}$. This scheme (3.8) 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 3.2.2. We first would like to give a simple proof that the semi-discrete solution of (3.8) is unconditionally gradient stable without considering the spatial discretization in the following Section.

3.2.1 Energy stability

In this Section, we will give a simple proof that the first order scheme (3.8) is gradient stable in the H^{-1} norm in the semi-analytic setting under the reasonable assumptions. The proof is very similar to the stabilized semi-implicit scheme in [54] and operator splitting scheme in [26], except our scheme is fully implicit. The salient point is that one can prove such stability

when the original system is modified through our simple change of variable in Section 3.1.1.

We first make useful observations which will be used to prove stability.

- The operator $\Delta^{-1} : \left\{ v \in L^2 | \int_\Omega v dx = 0 \right\} \to H^2(\Omega)$ is defined [18]

$$\Delta^{-1} v = h \quad \Longleftrightarrow \quad < v, q >=< \Delta h, q >, \quad \forall q \in L^2(\Omega). \tag{3.9}$$

- For any $p, q \in L^2(\Omega)$, one can easily prove following identity,

$$< p, p - q >= \frac{1}{2} \left(\|p\|_0^2 - \|q\|_0^2 + \|p - q\|_0^2 \right) \geq \frac{1}{2} \left(\|p\|_0^2 - \|q\|_0^2 \right), \tag{3.10}$$

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

We will consider the first-order scheme (3.8) in general spatial dimension ($\Omega \in \mathbb{R}^d, d \in \mathbb{N}$).

Lemma 1. *Under the following assumption,*

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

the fully-implicit scheme (3.8) *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* (3.1)*.*

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

$$\frac{1}{\Delta t} < v^{n+1} - v^n, \phi >= -\epsilon^2 < \Delta v^{n+1}, \Delta \phi > + 2 < v^{n+1}, \Delta \phi > + < \tilde{f}^{n+1}, \Delta \phi >, \tag{3.11}$$

for all $\phi \in H^2(\Omega)$. We choose $\phi = \Delta^{-1}(v^{n+1} - v^n)$, then this weak form becomes

$$-\frac{1}{\Delta t}\|\nabla \cdot \Delta^{-1}(v^{n+1} - v^n)\|_0^2 = \epsilon^2 < \nabla v^{n+1}, \nabla(v^{n+1} - v^n) > \\ + 2 < v^{n+1}, v^{n+1} - v^n > + < \tilde{f}^{n+1}, v^{n+1} - v^n >, \tag{3.12}$$

by the definition of operator Δ^{-1} in (3.9), and employing integration by parts. We now apply the property (3.10),

$$0 \geq \frac{\epsilon^2}{2}(\|\nabla v^{n+1}\|_0^2 - \|\nabla v^n\|_0^2) + (\|v^{n+1}\|_0^2 - \|v^n\|_0^2 + \|v^{n+1} - v^n\|_0^2) + < \tilde{f}^{n+1}, v^{n+1} - v^n > . \tag{3.13}$$

We replace the last term with its Taylor expansion

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

so that the inequality (3.13) becomes

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

where $\|\cdot\|_\infty$ is L^∞ norm. The last inequality holds because $\tilde{f}' = 3v^2 - 6v \leq 0$ with our assumption v. Therefore, with this physical assumption, the implicit scheme (3.8) is unconditionally gradient stable. □

Remark 8. *In [54, 46], Shen et.al. introduced a stabilizing term $S(u^{n+1} - u^n)$ for their semi-implicit scheme for the AC and CH equations. If $S \geq \frac{\max_{u\in\mathbb{R}}|\tilde{f}'(u)|}{2}$ with the truncated*

potential, then their scheme guarantees unconditionally stability. In practice, if they are truncating $F(u)$ outside the interval $[-1, 1]$ such that $\max_{u\in\mathbb{R}} |f'(u)| = 2$. Essentially, our cut-off of v is equivalent to this assumption. Moreover, our transformed variable has same stabilizing effect, without introducing an additional term.

We proved the weak formulation of our first-order scheme. In practice, we solve CH equation using a strong formulation. Even though the proof we gave it here is only illustrative, all numerical simulations we are using allows to take large time step size while maintaining gradient stability.

3.2.2 Two nonlinear iterative schemes

In this Section, we will give direct iteration solvers to solve the nonlinear problem (3.8). We first set up a simple fixed point iteration to solve v^{n+1} of the first-order scheme (3.8)

$$\Big(I - (2\Delta t\partial_{xx} - \epsilon^2\Delta t\partial_{xxxx})\Big)\, v^{n+1,k+1} = v^n + \Delta t\partial_{xx}\tilde{f}^{n+1,k}, \tag{3.14}$$

where k indicates the iteration index. By lagging the the nonlinear term $\tilde{f}^{n+1,k}$, the iteration update is made explicit. Now, we invert the fourth-order operator *analytically*, yielding the update equation

$$v^{n+1,k+1} = \Big(I - 2\Delta t\partial_{xx} + \epsilon^2\Delta t\partial_{xxxx}\Big)^{-1}\Big[v^n + \Delta t\partial_{xx}\tilde{f}^{n+1,k}\Big]. \tag{3.15}$$

If we employ the Green's function method directly to this fourth order operator, then the Green's function would contain both decaying and oscillatory computations, which would not produce an efficient method. However, we can instead factor this into a product of

second order differential operators (modified Helmholtz equations) which have a nonoscillatory Green's function, respectively, and in doing so leverage the fast computation methods previously employed in the previous Chapter 2.

After some algebra, we rewrite (3.15) as the product of two second order differential operators (modified Helmholtz),

$$v^{n+1,k+1} = \left(I - \frac{\partial_{xx}}{\alpha_1^2}\right)^{-1} \left(I - \frac{\partial_{xx}}{\alpha_2^2}\right)^{-1} \left[v^n + \Delta t \partial_{xx} \tilde{f}^{n+1,k}\right], \tag{3.16}$$

where $\frac{1}{\alpha_1^2} = \Delta t + \sqrt{\Delta t^2 - \epsilon^2 \Delta t}$ and $\frac{1}{\alpha_2^2} = \Delta t - \sqrt{\Delta t^2 - \epsilon^2 \Delta t}$, so that $\alpha_1, \alpha_2 \in \mathbb{R}$ when $\Delta t \geq \epsilon^2$. For $\Delta t \geq \epsilon^2$, the inverse operators both correspond to Green's function which decay exponentially. However, for $\Delta t < \epsilon^2$, the corresponding Green's function exhibits oscillatory behavior.

To overcome this difficulty, we suggest another fixed-point iteration to solve v^{n+1}:

$$v^{n+1,k+1} = \left(I - \sqrt{\epsilon^2 \Delta t}\partial_{xx}\right)^{-2} \left[v^n + \Delta t \partial_{xx} \left(f^{n+1,k} - 2\sqrt{\frac{\epsilon^2}{\Delta t}} v^{n+1,k}\right)\right], \tag{3.17}$$

which is based on completing the square, and has required the addition of the term $2\sqrt{\epsilon^2 \Delta t}\left(v^{n+1,k+1} - v^{n+1,k}\right)$. The purpose for adding the extra term is to ensure that the required Green's function maintains decaying property.

We should point out that even though the first iteration (3.16) has a time step size restriction to have a real-valued Green's function, we gave a simple proof that this scheme is gradient stable in semi-analytic setting in Section 3.2.1. Moreover, we will present numerical evidence that the scheme converges even with very large Δt, which is of paramount importance for numerical simulation of CH type equations.

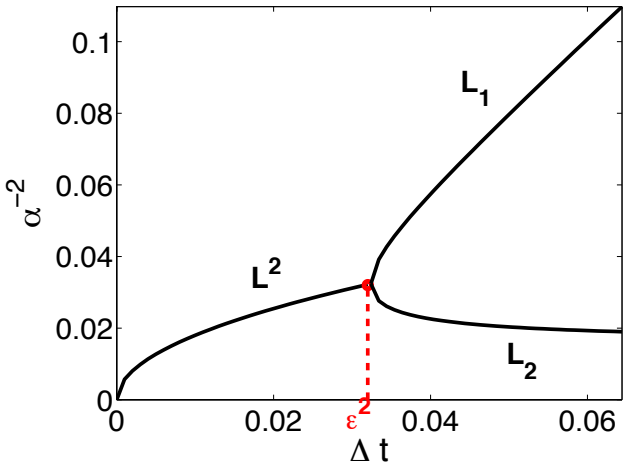


Figure 3.1: Bifurcation of the parameter $\frac{1}{\alpha^2}$ in the modified Helmholtz operator $\mathcal{L}$.

When $\Delta t \leq \epsilon^2$, alternatively, we can transition to (3.17). Since both iteration schemes converge to one solution with the error of some order of Δt, and are exactly equivalent at the transition point $\Delta t = \epsilon^2$ as shown in Figure 3.1. Therefore, we summarize as following:

$$\text{Factorization method}(\Delta t \geq \epsilon^2): \quad \text{Invert} \left(I - \frac{\partial_{xx}}{\alpha_1^2}\right)\left(I - \frac{\partial_{xx}}{\alpha_2^2}\right)$$

$$\text{Completed square method } (\Delta t < \epsilon^2): \quad \text{Invert} \left(I - \frac{\partial_{xx}}{\alpha^2}\right)^2$$

where the constants α and $\alpha_i (i = 1, 2)$ are determined by each scheme in Figure 3.1. We should point out that the main reason we choose to change the operator is because the solution for v is between 0 and 2 and a non-oscillatory kernel helps ensure that the integral solution we compute is also non-oscillatory.

Remark 9. *The second iterative scheme* (3.17) *was defined by introducing a second-order term,* $S\partial_{xx}(v^{n+1,k+1} - v^{n+1,k})$ *where* $S = 2\sqrt{\epsilon^2 \Delta t}$. *Since our variable* v *was already transformed by the steady stae,* $u = -1$, *this additional term could lower the stabilizing effect. Indeed, we have checked that this second iteration guarantees a decrease in energy when*

$\Delta t \leq \epsilon^p$ *where* $p \approx 1.5$*, so that we can safely transfer the iteration with our criteria* $\Delta t \leq \epsilon^2$.

Both iterations are made by two inversions of a modified Helmholtz operator in (2.3) where $\beta = 1$ and $\gamma = \epsilon^2$, hence

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

with the different values of α. Formal inversion of this operator in Section 2.2.1 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,

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

and the coefficients B_a and B_b are determined by applying boundary conditions at $x = a, b$ in Section 3.2.3.

Remark 10. *We additionally can calculate the second derivative of* u *as follows*

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

where we use the definition of differential operator (2.3).

3.2.3 Fully discrete solution

Our semi-discrete scheme is made by double inversion of modified Helmholtz operators, defined in Section 3.2.2. It remains to discretize the convolution integral $\mathcal{I}[v](x)$, and obtain a fully discrete algorithm. In the previous Chapter 2, we have accomplished this inversion

with the fast convolution algorithm for our inversion operator (2.9); for convenience, we will point out the essence of this fast algorithm in this Section.

For example, we review the spatial quadrature on a uniform grid ($h_j = \Delta x$), as the second-order accurate ($M = 2$) scheme. First of all, the domain $\Omega \equiv [a, b]$ is partitioned into N subdomains $[x_{j-1}, x_j]$

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

we then should evaluate the convolution operator at each grid point through $\mathcal{I}_j \equiv \mathcal{I}[v](x_j) = \mathcal{I}_j^L + \mathcal{I}_j^R$ in Section 2.2.2.1. However, because $\mathcal{I}^L$ and $\mathcal{I}^R$ are satisfying the recursive relation in (2.12) and (2.13) respectively, this evaluation can be simplified through following approximation

$$\begin{aligned}
\mathcal{I}_j^L &= e^{-\nu}\mathcal{I}_{j-1}^L + \mathcal{J}_j^L, \quad \mathcal{J}_j^L = \frac{\nu}{2}\int_0^1 e^{-\nu z} v(x_j - \Delta x z) dz, \\
\mathcal{I}_j^R &= e^{-\nu}\mathcal{I}_{j+1}^R + \mathcal{J}_j^R, \quad \mathcal{J}_j^R = \frac{\nu}{2}\int_0^1 e^{-\nu z} v(x_j + \Delta x z) dz,
\end{aligned}$$

where $\mathcal{J}_j^L$ and $\mathcal{J}_j^R$ are "local" integrals in Section 2.2.2.2 and $\nu = \alpha \Delta x$. The standard Lagrange interpolation is used to approximate these local integrals,

$$\begin{aligned}
\mathcal{J}_j^L &\approx \int_0^1 e^{-\nu z} p_j^L dz, \quad p_j^L = \left(\frac{z^2 + z}{2}\right) v_{j-1} + \left(1 - z^2\right) v_j + \left(\frac{z^2 - z}{2}\right) v_{j+1}, \\
\mathcal{J}_j^R &\approx \int_0^1 e^{-\nu z} p_j^R dz, \quad p_j^R = \left(\frac{z^2 - z}{2}\right) v_{j-1} + \left(1 - z^2\right) v_j + \left(\frac{z^2 + z}{2}\right) v_{j+1}.
\end{aligned}$$

Integrating against an exponential, we find that

$$\mathcal{J}_j^L \approx w_{-1} \cdot v_{j-1} + w_1 \cdot v_j + w_2 \cdot v_{j+1}, \quad \mathcal{J}_j^R \approx w_1 \cdot v_{j-1} + w_0 \cdot v_j + w_{-1} \cdot v_{j+1},$$

where the quadrature weights are given by

$$\begin{aligned} w_{-1} &= \frac{1}{2}\left(\phi_{j,2} + \phi_{j,1}\right) = \frac{1}{2}\left(d + \frac{1-3d}{2\nu} + \frac{1-d}{\nu^2}\right), \\ w_0 &= \left(\phi_{j,0} - \phi_{j,2}\right) = \frac{1}{2} + \frac{d}{\nu} - \frac{1-d}{\nu^2}, \\ w_1 &= \frac{1}{2}\left(\phi_{j,2} - \phi_{j,1}\right) = \frac{1}{2}\left(\frac{1-d}{\nu^2} - \frac{1+d}{2\nu}\right). \end{aligned}$$

where $d = e^{-\nu}$ and $\phi_{j,k}$ is defined in (2.17).

For higher-order spatial schemes ($M > 2$), the quadrature weights are pre-computed as the similar way, which presented in Section 2.2.2.2, so that the fast convolution algorithm is achieved as $\mathcal{O}(MN)$ per time step where user-defined order M in space. In every simulation in this work, we choose $M = 4$ or $M = 6$.

Next, the homogenous solution (2.11) is used to enforce various boundary conditions in Section 2.2.3.3. 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 (2.11) at $x = a, b$, produces a 2×2 system for the unknown B_a and B_b. Solving this linear system yields

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

where $\mu = e^{-\alpha(b-a)}$ and $\mathcal{I}_0 = \mathcal{I}[v](a)$ and $\mathcal{I}_N = \mathcal{I}[v](b)$.

3.3 Higher order scheme for 1D CH equation

In this Section, we show how to make second and third order time accurate methods by combining the ideas of MOL$^{\mathrm{T}}$ formulation with Backward Difference Formulas (BDF), Singly Diagonal Implicit Runge Kutta (SDIRK), and Spectral Differed Correction (SDC) methods. We will present several refinement studies to compare those methods.

3.3.1 MOL^T with Backward Difference Formula (BDF)

The BDF time stepping methods are one of the most commonly used implicit Linear Multi-step Methods [34]. To achieve the designed order of accuracy, we discretize a time derivative using BDF formulas as follows:

$$\text{BDF2:} \quad \left(I + \frac{2}{3}\epsilon^2 \Delta t \partial_{xxxx}\right) v^{n+1} = \frac{4}{3}v^n - \frac{1}{3}v^{n-1} + \frac{2}{3}\Delta t \partial_{xx} f^{n+1}, \tag{3.18}$$

$$\text{BDF3:} \quad \left(I + \frac{6}{11}\epsilon^2 \Delta t \partial_{xxxx}\right) v^{n+1} = \frac{18}{11}v^n - \frac{9}{11}v^{n-1} + \frac{2}{11}v^{n-2} + \frac{6}{11}\Delta t \partial_{xx} f^{n+1}, \tag{3.19}$$

where we now require two or three previous time steps, respectively. Similar with the first-order scheme in Section 3.2, we define two fixed-point iterations. 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. With that, BDF2 iterations for (3.18)

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

where $\frac{1}{\alpha_1^2} = \frac{2}{3}\Delta t + \sqrt{(\frac{2}{3}\Delta t)^2 - \frac{2}{3}\epsilon^2\Delta t}$, $\frac{1}{\alpha_2^2} = \frac{2}{3}\Delta t - \sqrt{(\frac{2}{3}\Delta t)^2 - \frac{2}{3}\epsilon^2\Delta t}$ and $\frac{1}{\alpha^2} = \sqrt{\frac{2}{3}\epsilon^2\Delta t}$. For BDF3 in (3.19),

$$\Delta t \geq \frac{11}{6}\epsilon^2: \quad \mathcal{L} = I - \frac{12}{6}\Delta t\partial_{xx} + \frac{6}{11}\epsilon^2\Delta t\partial_{xxxx} = \left(I - \frac{\partial_{xx}}{\alpha_1^2}\right)\left(I - \frac{\partial_{xx}}{\alpha_2^2}\right),$$
$$\Delta t < \frac{11}{6}\epsilon^2: \quad \mathcal{L} = I - 2\sqrt{\frac{6}{11}\epsilon^2\Delta t}\partial_{xx} + \frac{6}{11}\epsilon^2\Delta t\partial_{xxxx} = \left(I - \frac{\partial_{xx}}{\alpha^2}\right)^2,$$

where $\frac{1}{\alpha_1^2} = \frac{6}{11}\Delta t + \sqrt{(\frac{6}{11}\Delta t)^2 - \frac{6}{11}\epsilon^2\Delta t}$, $\frac{1}{\alpha_2^2} = \frac{6}{11}\Delta t - \sqrt{(\frac{6}{11}\Delta t)^2 - \frac{6}{11}\epsilon^2\Delta t}$ and $\frac{1}{\alpha^2} = \sqrt{\frac{6}{11}\epsilon^2\Delta t}$.

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

3.3.2 MOL^T with Singly Diagonal Implicit Runge Kutta (SDIRK)

The SDIRK method is an implicit Runge Kutta (RK) method, which has the same diagonal element in the Butcher Table in [2]. For the P^{th} order method, RK methods need P intermediate steps in one time stepping. It is somewhat computationally expensive, but it is desirable for the initial step of multi step methods. Here, we derive the second-order SDIRK

(SDIRK2) method for CH equation, but employing SDIRK3 is similar.

$$\left(I+\eta\epsilon^2\Delta t\partial_{xxxx}\right)K_1 = -\epsilon^2\partial_{xxxx}v^n + \partial_{xx}f\left(v^n+\eta\Delta tK_1\right), \qquad \eta = 1-\frac{\sqrt{2}}{2},$$
$$\left(I+\eta\epsilon^2\Delta t\partial_{xxxx}\right)K_2 = -\epsilon^2\partial_{xxxx}\left(v^n+(1-\eta)\Delta tK_1\right) + \partial_{xx}f\left(v^n+(1-\eta)\Delta tK_1+\eta\Delta tK_2\right)$$
$$v^{n+1} = v^n + \Delta t\{(1-\eta)K_1+\eta K_2\},$$

where η is the roolt of the polynomial $\frac{1}{2}-2\eta+\eta^2$, which is deriven from the order conditions [2]. Next, we solve two intermediate solution K_1 and K_2 using similar nonlinear iterative schemes in Section 3.2.2. First, if $\Delta t \geq \dfrac{\epsilon^2}{\eta}$,

$$K_1^{n+1,k+1} = \mathcal{L}_1^{-1}\mathcal{L}_2^{-1}\left[-\epsilon^2\partial_{xxxx}v^n + \partial_{xx}\left(\tilde{f}\left(v^n+\eta\Delta tK_1^{n+1,k}\right)+2v^n\right)\right], \quad \mathcal{L}_i = I-\frac{\partial_{xx}}{\alpha_i^2}$$
$$K_2^{n+1,k+1} = \mathcal{L}_1^{-1}\mathcal{L}_2^{-1}\Bigg[-\epsilon^2\partial_{xxxx}\left(v^n+(1-\eta)\Delta tK_1^{n+1,k}\right)$$
$$+\partial_{xx}\left(\tilde{f}\left(v^n+(1-\eta)\Delta tK_1^{n+1,k}+\eta\Delta tK_2^{n+1,k}\right)+2(v^n+(1-\eta)\Delta tK_1^{n+1,k})\right)\Bigg],$$

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

$$K_1^{n+1,k+1} = \mathcal{L}^{-2}\left[-\epsilon^2\partial_{xxxx}v^n + \partial_{xx}\left(f\left(v^n+\eta\Delta tK_1^{n+1,k}\right)-2\sqrt{\eta\epsilon^2\Delta t}K_1^{n+1,k}\right)\right],$$
$$K_2^{n+1,k+1} = \mathcal{L}^{-2}\Bigg[-\epsilon^2\partial_{xxxx}\left(v^n+(1-\eta)\Delta tK_1^{n+1,k}\right)$$
$$+\partial_{xx}\left(f\left(v^n+(1-\eta)\Delta tK_1^{n+1,k}+\eta\Delta tK_2^{n+1,k}\right)-2\sqrt{\eta\epsilon^2\Delta t}K_2^{n+1,k}\right)\Bigg],$$

where $\mathcal{L} \equiv I-\sqrt{\eta\epsilon^2\Delta t}\partial_{xx}$ is used.

Remark 11. *Similar to Remark 10, the fourth derivative can be calculated as follows,*

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

In Section 3.3.4, we will present refinement studies for both the BDF and SDIRK methods. We note that for 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.

3.3.3 MOL^T with Spectral Deferred Correction (SDC)

The SDC methods are a class of time integrators [23]. Deferred Correction (DC) methods first compute a prediction to the solution ("level 0") using low order schemes (e.g. Backward Euler), and then compute one or more corrections at the subsequent levels, to get higher order of accuracy. We present the traditional SDC procedure in [17].

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

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

Compute $\{v_m^{[0]}(x)\}_{0\le m\le N_t}$, via Backward Euler approximation in (3.8) for CH equation, which is first-order approximation to the exact solution $\{y(x,t^m)\equiv y_m(x)\}_{0\le m\le N_t}$.

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

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

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

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

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

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

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 as the same method as "level 0". In our case, we apply Backward Euler,

$$v_{n+1}^{[1]} - v_n^{[1]} = \Delta t \left(\mathcal{F}_{\text{CH}}(v_{n+1}^{[1]}) - \mathcal{F}_{\text{CH}}(v_{n+1}^{[0]}) \right) + \int_{t^n}^{t^{n+1}} \mathcal{F}_{\text{CH}}(v^{(0)}(x,\tau))d\tau. \quad (3.20)$$

3. **Correction step (level [j]):** The process is then iterated by generalizing (3.20),

$$v_{n+1}^{[j]} - \Delta t \mathcal{F}_{\mathrm{CH}}(v_{n+1}^{[j]}) = v_n^{[j]} - \Delta t \mathcal{F}_{\mathrm{CH}}(v_{n+1}^{[j-1]}) + \int_{t^n}^{t^{n+1}} \mathcal{F}_{\mathrm{CH}}(v^{(j-1)}(x,\tau))d\tau, \quad (3.21)$$

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 method, we must consider an approximation of the integral in (3.21):

$$\int_{t^n}^{t^{n+1}} \mathcal{F}_{\mathrm{CH}}(v^{(j-1)}(x,\tau))d\tau = \begin{cases} \Delta t \sum_{i=0}^{P} \tilde{q}_i \mathcal{F}_{n+1-i}^{[j-1]}, & \text{if} \quad n \geq P-1, \\ \Delta t \sum_{i=0}^{P} \tilde{q}_i \mathcal{F}_{i}^{[j-1]} & \text{if} \quad n < P-1, \end{cases} \quad (3.22)$$

where $\mathcal{F}_n^{[j-1]} = \mathcal{F}_{\mathrm{CH}}(v_n^{[j-1]})$, and $\tilde{q}_i$ are quadrature weights (cf. [17]). For example, if $P = 2$ and $n > P-1$, the function values $\mathcal{F}_{n+1}^{[1]}$, $\mathcal{F}_n^{[1]}$, and $\mathcal{F}_{n-1}^{[1]}$ are used for the quadrature and

$$\tilde{q}_i = \int_{t^n}^{t^{n+1}} \prod_{k=0, k\neq i}^{P=2} \frac{(t - t^{n+1-k})}{(t^{n+1-i} - t^{n+1-k})} dt, \quad i = 0, 1, 2.$$

Note that the number of terms in the sum (3.22) is $P+1$, where P is the order of polynomial interpolation $v^{(j-1)}$. The integral must be approximated with increasing accuracy as level increases, so that $P \geq j$ at "level $[j]$". In Section 3.3.4, 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 $\Delta t \geq \epsilon^2$,

$$v_{n+1,k+1}^{[1]} = \mathcal{L}_1^{-1}\mathcal{L}_2^{-1}\left[v_n^{[1]} + \Delta t\partial_{xx}\left(\tilde{f}_{n+1,k}^{[1]} - \frac{f_{n+1}^{[0]} - f_n^{[0]}}{2}\right) + \frac{\epsilon^2\Delta t}{2}\partial_{xxxx}\left(v_{n+1}^{[0]} - v_n^{[0]}\right)\right] \tag{3.23}$$

where quadrature weights of (3.22) 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 3.2. Similarly, if $\Delta t < \epsilon^2$,

$$\begin{aligned} v_{n+1,k+1}^{[1]} = \mathcal{L}^{-2}&\left[v_n^{[1]} + \Delta t\partial_{xx}\left(f_{n+1,k}^{[1]} - 2\sqrt{\frac{\epsilon^2}{\Delta t}}v_{n+1,k}^{[1]} - \frac{f_{n+1}^{[0]} - f_n^{[0]}}{2}\right)\right] \\ &+ \mathcal{L}^{-2}\left[\frac{\epsilon^2\Delta t}{2}\partial_{xxxx}\left(v_{n+1}^{[0]} - v_n^{[0]}\right)\right] \end{aligned} \tag{3.24}$$

We will now present refinement studies for second- and third-order BDF, SDIRK and SDC methods in the following Section.

3.3.4 Numerical test: refinement studies of 1D CH solutions

In this Section, we will check the order of accuracy of presented time stepping methods and compare these schemes. Starting from an initial data in reference [16],

$$u_0(x) = \cos(2x) + \frac{1}{100}e^{\cos\left(x+\frac{1}{10}\right)}, \qquad x \in [0, 2\pi] \tag{3.25}$$

and with the periodic boundary conditions, we integrate up to a final time T_{final} using each second-order method. If we denote $u_{\Delta t}$ as the computed solution with fixed time-step size Δt, then recompute solution $u_{\Delta t/2}$ with halved time step up to T_{final}. Given that we do

	BDF2		SDIRK2		SDC2	
Δt	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

Table 3.1: Refinement studies of second-order methods for 1D CH equation with periodic BC.

not have an exact solution of (3.1), we estimated the error for Δt as the successive error in maximum norm, that is $\|u_{\Delta t} - u_{\Delta t/2}\|_\infty$, and again halved the time step to compute the corresponding error repeatedly. Those successive errors for three second-order methods are presented in Table 3.1. We used the parameters in common:

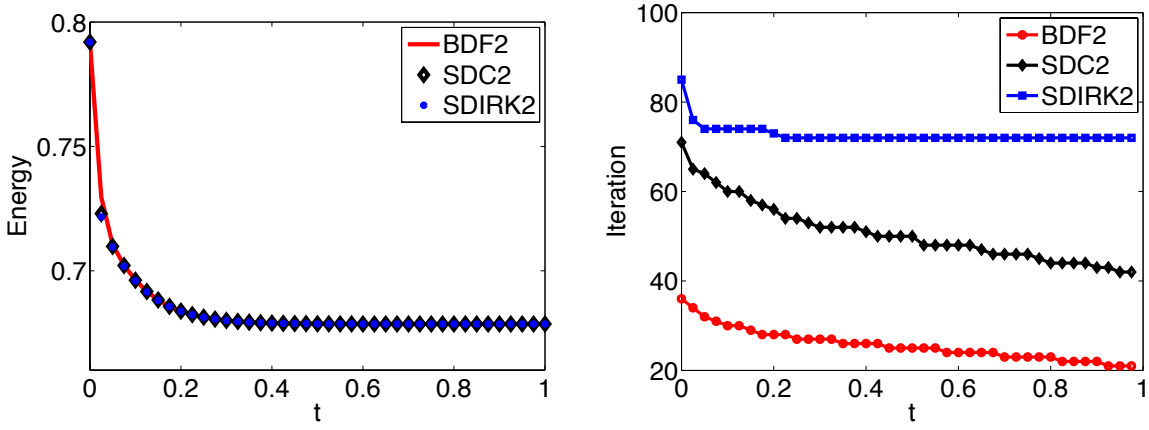
$$\epsilon = 0.18, \quad \Delta x = \frac{2\pi}{512} \approx 0.0123, \quad T_{\text{final}} = 1, \quad N_{\text{tol}} = 10^{-12}, \tag{3.26}$$

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 6^{th}-order spatial quadrature ($M = 6$), so that the dominant error was temporal.

As explained, each method has a criterion for time step size, which leads to switch iterative schemes:

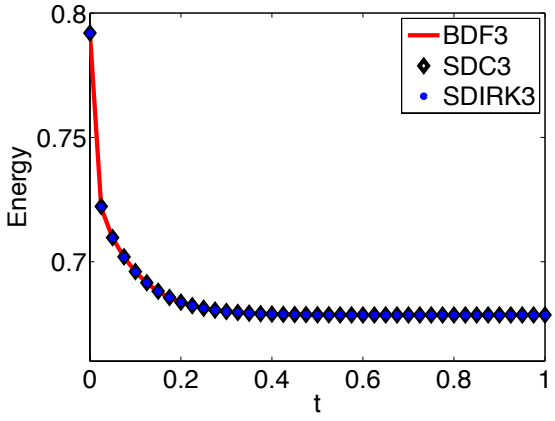
$$\text{BDF2: } \Delta t = \frac{3}{2}\epsilon^2, \quad \text{SDIRK2: } \Delta t = \frac{\epsilon^2}{1 - \frac{\sqrt{2}}{2}}, \quad \text{SDC2: } \Delta t = \epsilon^2.$$

With this criteria, we switched the corresponding fixed-point iterations, and the Table 3.1 shows that each of the methods exhibits second order convergence, respectively. This result guarantees the safe adaptive switching control. Moreover in Figure 3.2a, we see that the total energy of each fixed time stepping method decays during the time evolution.

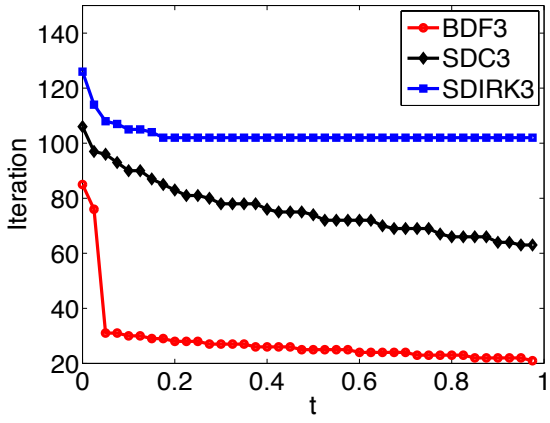


(a) Energy decay (2^{nd} order methods)

(b) Iteration count(2^{nd} order methods)



(c) Energy decay (3^{rd} order methods)



(d) Iteration count (3^{rd} order methods)

Figure 3.2: Energy descent and nonlinear iteration count history of all second- and third-order methods with fixed time step $\Delta t = 0.025$ and parameters (3.26).

In Figure 3.2b, we plot the iteration count at each time level of all second-order schemes (with the same tolerance $N_{\text{tol}} = 10^{-12}$). Since SDIRK2 is comprised ot two intermediate steps, this requires two separate nonlinear iterations. Hence, SDIRK2 methods is more expansive than the other two methods. In fact, using our MATLAB codes, the computational time of BDF2, SDC2, SDIRK2 is $0.9725(s)$, $1.9003(s)$, and $2.6468(s)$, respectively, to get final time $T_{\text{final}} = 1$. Therefore, we can conclude that BDF2 is most computationally efficient among these schemes.

Next, we similarly implemented the corresponding third-order schemes for CH equation, with the same parameters in (3.26), and with the following considerations:

- **[BDF3]** Need two initial step: 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. ($\Delta t_{\text{switch}} = \frac{\epsilon^2}{\eta}, \eta \approx 0.4359$)
- **[SDC3]** Compute from level [0] to level [2] with the quadrature order P in (3.22): If we 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 hisotry of third order schemes are shown in Figure 3.2c and Figure 3.2d. Again we see that the discrete energy of each solution decays. The very beginning iteration number of BDF3 is larger than BDF2's since we initialized using SDIRK2. But BDF3 is still the most efficient method than other third order schemes. Using MATLAB built-in function, the computational time of BDF3, SDC3, SDIRK3 is $1.0355(s)$, $2.8539(s)$, and $3.6613(s)$, respectively, to compute up to the final time $T_{\text{final}} = 1$. The successive errors of the third-order methods are shown in Table 3.2.

Each method achieves third order convergence for small Δt; but the order of convergence begins to plateau for larger time steps. However, we point out that the plateau is less severe

	BDF3		SDIRK3		SDC3	
Δt	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	3.1124×10^{-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	5.4193×10^{-9}	2.9752
0.0016	3.2511×10^{-9}	3.0273	1.7447×10^{-10}	2.8986	6.6913×10^{-10}	3.0177

Table 3.2: Refinement studies of third-order methods for the 1D CH equation with periodic BC.

	SDC3 ($P = 3$)	
Δt	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	1.6156×10^{-8}	3.1298
0.0031	1.7852×10^{-9}	3.1779
0.0016	2.2753×10^{-10}	2.9720

Table 3.3: Refinement study of SDC3 ($P = 3$) for the 1D CH equation with periodic BC.

for SDC3 than BDF3 or SDIRK3; the convergence order $C_{\mathrm{R}} := \log_2 \left(\frac{\|u_{\Delta t} - u_{\Delta t/2}\|_\infty}{\|u_{\Delta t/2} - u_{\Delta t/4}\|_\infty} \right)$ of SDC3 starts around 2.59, which is closer to the expected order, compared to BDF3 ($C_{\mathrm{R}} = 1.6$) and SDIRK3 ($C_{\mathrm{R}} = 2.26$) when $\Delta t = 0.05$.

Furthermore, if we use higher quadrature of order P in (3.22) for SDC3, then the anticipated rate of convergence is observed even for large time steps Δt. For instance, if we use $P = 3$, (the quadrature weights are $\tilde{q}_0 = \frac{9}{24}, \tilde{q}_1 = \frac{19}{24}, \tilde{q}_2 = -\frac{5}{24}$ and $\tilde{q}_3 = \frac{1}{24}$), then the rate is roughly third order even with the larger time steps, shown in Table 3.3.

We summarize this Section that we can achieve expected order of accuracy through (short-time) refinement studies of second and third order schemes. From the point of view of solver efficiency, higher-order BDF schemes (BDF2 and BDF3) are ideal. High-order

quadrature in SDC3 shows interesting result, which performs well on larger time steps.

3.4 MOL^T for Multi-D CH equation

We now extend the 1D solver to multiple spatial dimension via dimensional splitting [13, 10, 12] which also discussed in Section 2.4. Each time stepping method presented thus far requires the inversion of a modified Helmholtz operator, with independently defined α. For simplicity, in 2D,

$$\mathcal{L} = I - \frac{\Delta}{\alpha^2} = \left(I - \frac{\partial_{xx}}{\alpha^2}\right)\left(I - \frac{\partial_{yy}}{\alpha^2}\right) - \frac{\partial_{xx}\partial_{yy}}{\alpha^4} \equiv \mathcal{L}_x\mathcal{L}_y - \frac{\partial_{xx}\partial_{yy}}{\alpha^4}, \tag{3.27}$$

where $\mathcal{L}_x$ and $\mathcal{L}_y$ are our $\mathcal{O}(N)$ 1D solvers and the subscripts denote the spatial component of univariate modified Helmholtz operators. Later, we should formally invert both operators. Now $\mathcal{L}_x^{-1}$ is done 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 [22, 27]. However, this factorization (3.27) yields a mixed derivative term, which is $O(\frac{1}{\alpha^4})$. Our key observation is that if we include this term in our fixed point iteration, then we can simultaneously solve the nonlinear problem and remove the splitting error. More detaild analysis of the 2D CH equation (3.1) and 2D CH vector equation (3.4) will be present in next several Sections.

3.4.1 First order scheme for 2D CH equation

In this Section, we will present the semi-discrete scheme for 2D CH equation, using the Backward Euler method. The analogous higher-order schemes follow accordingly. Starting from the 1D scheme (3.14), we replace ∂_{xx} with the Laplacian operator $\Delta = \partial_{xx} + \partial_{yy}$. If

$\Delta t > \epsilon^2$, we factorize the operator, and have

$$\left(I - \frac{\Delta}{\alpha_1^2}\right)\left(I - \frac{\Delta}{\alpha_2^2}\right)[v^{n+1,k+1}] = v^n + \Delta t \Delta \tilde{f}^{n+1,k}, \quad \tilde{f} = f(v) - 2v \equiv v^3 - 3v^2, \quad (3.28)$$

where $\dfrac{1}{\alpha_i^2} = \Delta t \pm \sqrt{\Delta t^2 - \epsilon^2 \Delta t}$, $i = 1, 2$ respectively, which are same parameters with 1D scheme. Plugging the identity in (3.27), by lagging the mixed derivative term along with the nonlinear term,

$$\begin{aligned}\left(\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)[v^{n+1,k+1}] &= v^n + \Delta t \Delta \tilde{f}^{n+1,k} \\ &+ \left(\left(\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)\frac{\partial_{xx}\partial_{yy}}{\alpha_2^4} + \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)\frac{\partial_{xx}\partial_{yy}}{\alpha_1^4} - \left(\frac{\partial_{xx}\partial_{yy}}{\alpha_1^4}\right)\left(\frac{\partial_{xx}\partial_{yy}}{\alpha_2^4}\right)\right) v^{n+1,k},\end{aligned} \quad (3.29)$$

where $\mathcal{L}_{i,x} = I - \dfrac{\partial_{xx}}{\alpha_i^2}$, $\quad \mathcal{L}_{i,y} = I - \dfrac{\partial_{yy}}{\alpha_i^2}$, $\quad (i = 1, 2)$. Note that Laplacian operator and mixed derivative can be replaced as following:

$$\Delta = \alpha^2\left(I - \mathcal{L}_x\mathcal{L}_y + \frac{\partial_{xx}\partial_{yy}}{\alpha^4}\right), \quad \frac{\partial_{xx}\partial_{yy}}{\alpha^4} = (\mathcal{L}_x - I)(\mathcal{L}_y - I). \quad (3.30)$$

Now we formally invert both operators $\left(\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)$ to the right hand side of (3.29),

$$v^{n+1,k+1} = \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)^{-1}[v^n] + \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1[\alpha_1^2 \Delta t \tilde{f}^{n+1,k}] + \mathcal{M}_{1,2}\left[v^{n+1,k}\right] \quad (3.31)$$

where

$$\mathcal{C}_i = \left(\mathcal{L}_{i,x}\mathcal{L}_{i,y}\right)^{-1} - I + \mathcal{D}_{i,x}\mathcal{D}_{i,y}, \quad \mathcal{D}_{i,\gamma} = I - \mathcal{L}_{i,\gamma}^{-1}, (\gamma = \{x, y\}), \tag{3.32}$$

$$\mathcal{M}_{i,j} = \mathcal{D}_{i,x}\mathcal{D}_{i,y} + \mathcal{D}_{j,x}\mathcal{D}_{j,y} - \left(\mathcal{D}_{i,x}\mathcal{D}_{i,y}\mathcal{D}_{j,x}\mathcal{D}_{j,y}\right). \tag{3.33}$$

As shown, every mixed-derivative splitting error term can be controlled by applying $\mathcal{D}_\gamma$, $(\gamma = \{x, y\})$ operators, which can also be constructed in a line-by-line fashion. We emphasize that this allows us to remove splitting error $\mathcal{O}(\frac{1}{\alpha^4}) = O(\Delta t)$.

Similar operators' extension $\mathcal{L}_\gamma^{-1}$ and $\mathcal{D}_\gamma$ holds for the other case, $\Delta t \leq \epsilon^2$:

$$\left(I - \frac{\Delta}{\alpha_0^2}\right)^2 [v^{n+1,k+1}] = v^n + \Delta\left(\Delta t f^{n+1,k} - \frac{2}{\alpha_0^2} v^{n+1,k}\right), \tag{3.34}$$

where $\frac{1}{\alpha_0^2} = \sqrt{\epsilon^2 \Delta t}$. This completed square form is replaced by the identity in (3.27),

$$\begin{aligned}\left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^2 [v^{n+1,k+1}] = v^n + \Delta &\left(\Delta t f^{n+1,k} - \frac{2}{\alpha_0^2} v^{n+1,k}\right) \\ &+ \left(2\left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)\frac{\partial_{xx}\partial_{yy}}{\alpha_0^4} - \left(\frac{\partial_{xx}\partial_{yy}}{\alpha_0^4}\right)^2\right) v^{n+1,k}.\end{aligned} \tag{3.35}$$

Finally, we invert the modified Helmholtz operators and use the above operators (3.32) and (3.33), then

$$v^{n+1,k+1} = \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-2} [v^n] + \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1} \mathcal{C}_0 \left[\alpha_0^2 \Delta t f^{n+1,k} - 2v^{n+1,k}\right] + \mathcal{M}_{0,0}\left[v^{n+1,k}\right]. \tag{3.36}$$

We now have first-order switching-scheme with the time-step criterion $\Delta t = \epsilon^2$, whcih is same formulation with the 1D in Section 3.2.2. In addition, we can extend the fully-

discrete solution by applying the same procedure in Section 3.2.3 via line-by-line fashion. This straightforward extension is one of main advantages of dimensional-split algorithm. We also point out that we have already proved energy stability of above first-order scheme in Section 3.2.1.

3.4.2 Higher order scheme for 2D CH equation

The higher-order time stepping methods' extensions on 2D are analogous to the first order scheme in previous Section 3.4.1. Hence, we will briefly state the formulas for second-order 2D CH schemes in this Section.

- BDF2 $\left(\frac{1}{\alpha_0^2} = \sqrt{\frac{2}{3}\epsilon^2\Delta t}, \quad \frac{1}{\alpha_i^2} = \frac{2}{3}\Delta t \pm \sqrt{\frac{4}{9}\Delta t^2 - \frac{2}{3}\epsilon^2\Delta t} \quad (i = 1, 2)\right)$

$$\begin{aligned}
\Delta t \leq \frac{3}{2}\epsilon^2: \quad v^{n+1,k+1} &= \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-2}\left[\frac{4}{3}v^n - \frac{1}{3}v^{n-1}\right] \\
&\quad + \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1}\mathcal{C}_0\left[\frac{2}{3}\alpha_0^2\Delta t f^{n+1,k} - 2v^{n+1,k}\right] + \mathcal{M}_{0,0}\left[v^{n+1,k}\right]. \\
\Delta t > \frac{3}{2}\epsilon^2: \quad v^{n+1,k+1} &= \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)^{-1}\left[\frac{4}{3}v^n - \frac{1}{3}v^{n-1}\right] \\
&\quad + \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[\frac{2}{3}\alpha_1^2\Delta t \tilde{f}^{n+1,k}\right] + \mathcal{M}_{1,2}\left[v^{n+1,k}\right].
\end{aligned}$$

- SDIRK2 $\left(\frac{1}{\alpha_0^2} = \sqrt{\eta\epsilon^2\Delta t}, \quad \frac{1}{\alpha_i^2} = \eta\Delta t \pm \sqrt{(\eta\Delta t)^2 - \eta\epsilon^2\Delta t}, \quad \eta = 1 - \frac{\sqrt{2}}{2}\right)$

$$\Delta t \le \frac{\epsilon^2}{\eta}: \quad K_1^{n+1,k+1} = -\frac{1}{\eta\Delta t}\mathcal{C}_0^2\left[v^n\right] + \mathcal{M}_{0,0}\left[K_1^{n+1,k}\right]$$
$$+ \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1}\mathcal{C}_0\left[\alpha_0^2 f(v^n + \eta\Delta t K_1^{n+1,k}) - 2K_1^{n+1,k}\right].$$
$$K_2^{n+1,k+1} = -\frac{1}{\eta\Delta t}\mathcal{C}_0^2\left[v^n + (1-\eta)\Delta t K_1^{n+1,k}\right] + \mathcal{M}_{0,0}\left[K_2^{n+1,k}\right]$$
$$+ \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1}\mathcal{C}_0\left[\alpha_0^2 f(v^n + (1-\eta)\Delta t K_1^{n+1,k} + \eta\Delta t K_2^{n+1,k}) - 2K_2^{n+1,k}\right].$$
$$v^{n+1,k+1} = v^n + (1-\eta)\Delta t K_1^{n+1,k+1} + \eta\Delta t K_2^{n+1,k+1}.$$

$$\Delta t > \frac{\epsilon^2}{\eta}: \quad K_1^{n+1,k+1} = -\frac{1}{\eta\Delta t}\mathcal{C}_1\mathcal{C}_2\left[v^n\right] + \mathcal{M}_{1,2}\left[K_1^{n+1,k}\right]$$
$$+ \alpha_1^2\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[\tilde{f}(v^n + \eta\Delta t K_1^{n+1,k}) + 2v^n\right].$$
$$K_2^{n+1,k+1} = -\frac{1}{\eta\Delta t}\mathcal{C}_1^2\left[v^n + (1-\eta)\Delta t K_1^{n+1,k}\right] + \mathcal{M}_{1,2}\left[K_2^{n+1,k}\right]$$
$$+ \alpha_1^2\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[\tilde{f}(v^n + (1-\eta)\Delta t K_1^{n+1,k} + \eta\Delta t K_2^{n+1,k})\right]$$
$$+ \alpha_1^2\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[2(v^n + (1-\eta)\Delta t K_1^{n+1,k})\right].$$
$$v^{n+1,k+1} = v^n + (1-\eta)\Delta t K_1^{n+1,k+1} + \eta\Delta t K_2^{n+1,k+1}.$$

- SDC2: The numerical solution at prediction step (level [0]) $\{v_n^{[0]}\}_{n\in\mathbb{N}}$ is obtained by the first-order BE schemes in Section 3.4.1. The following is the correction step (level [1]) $\{v_n^{[1]}\}$ depending on the time step size.
$$\left(\frac{1}{\alpha_0^2} = \sqrt{\epsilon^2\Delta t}, \quad \frac{1}{\alpha_i^2} = \Delta t \pm \sqrt{\Delta t^2 - \epsilon^2\Delta t} \quad (i = 1, 2)\right)$$

$$\Delta t \le \epsilon^2: \quad v^{[1]}_{n+1,k+1} = \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-2}\left[v^{[1]}_n\right] + \left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1}\mathcal{C}_0\left[\alpha_0^2\Delta t f^{[1]}_{n+1,k} - 2v^{[1]}_{n+1,k}\right]$$
$$+ \mathcal{M}_{0,0}\left[v^{[1]}_{n+1,k}\right] - \frac{1}{2}\mathcal{C}_0^2\left[v^{[0]}_{n+1} - v^{[0]}_n\right] + \frac{\alpha_0^2\Delta t}{2}\left(\mathcal{L}_{0,x}\mathcal{L}_{0,y}\right)^{-1}\mathcal{C}_0\left[f^{[0]}_{n+1} - f^{[0]}_n\right].$$
$$\Delta t > \epsilon^2: \quad v^{[1]}_{n+1,k+1} = \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\mathcal{L}_{1,x}\mathcal{L}_{1,y}\right)^{-1}\left[v^{[1]}_n\right] + \left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[\alpha_1^2\Delta t \tilde{f}^{[1]}_{n+1,k}\right]$$
$$+ \mathcal{M}_{1,2}\left[v^{[1]}_{n+1,k}\right] - \frac{1}{2}\mathcal{C}_1\mathcal{C}_2\left[v^{[0]}_{n+1} - v^{[0]}_n\right] + \frac{\alpha_1^2\Delta t}{2}\left(\mathcal{L}_{2,x}\mathcal{L}_{2,y}\right)^{-1}\mathcal{C}_1\left[f^{[0]}_{n+1} - f^{[0]}_n\right].$$

With these formulas, we will implement refinement studies for 2D CH equation in the next Section.

3.4.3 Numerical test: refinement studies of 2D CH solutions

In this Section, we present results for the 2D CH equation. We consider the standard benchmark initial states in [16, 57] to confirm the temporal order of accuracy in 2D setting.

$$u_0(x,y) = 2e^{(\sin(x)+\sin(y)-2)} + 2.2e^{(-\sin(x)-\sin(y)-2)} - 1, \qquad (x,y) \in [0,2\pi]^2 \tag{3.37}$$

with the periodic boundary conditions and with the following parameters,

$$\epsilon = 0.18, \quad \Delta x = \frac{2\pi}{128} \approx 0.0491, \quad T_{\text{final}} = 1 \ (0 \le t \le T_{\text{final}}), \quad N_{\text{tol}} = 10^{-6}, \tag{3.38}$$

where a $4^{\text{th}}-$order spatial quadrature ($M = 4$) is used.

We present a temporal refinement study of each second-order scheme is presented in Table 3.4. This refinement study shows the second-order of convergence for all three methods.

Furthermore, the total energy of each solution is decreased during time evolution, as shown in Figure 3.3a. The dimensional split algorithm guarantees the energy-decent solution in multiple spatial dimensions. By comparing the iteration count at each time level in Figure

3.3b, the BDF2 method is the most efficient, as was the case in 1D.

We note that we addressed the splitting error directly in the dimensional splitting formulation by explicitly incorporating the mixed-derivative term (via $\mathcal{D}_\gamma$ operators in (3.32)) into the fixed-point method used to solve the nonlinear problem. This numerical studies guarentee that we can achieve higher orders of accuracy in space and time by removing splitting errors. This is important feature of the presented dimensional split algorithm.

We will show that our solver can be easily applied to the vector model in next Section.

3.5 MOL^T for vector CH (VCH) equation

We now extend our dimensional split MOL^T algorithm to vector CH (VCH) model (3.3). This model consists of two coupled variables u_1 and u_2 with local dynamic $\nabla_{\mathbf{u}} W$, comprised of partial derivatives of $6^{\text{th}}-$ order polynomials $W(u_1, u_2)$ defined in (3.4).

Before we employ MOL$^{\text{T}}$ formulation to this system, we transform the vector (u_1, u_2) about its background state. To do that, we first need to find the equilibrium points of the functional $W(u_1, u_2)$,

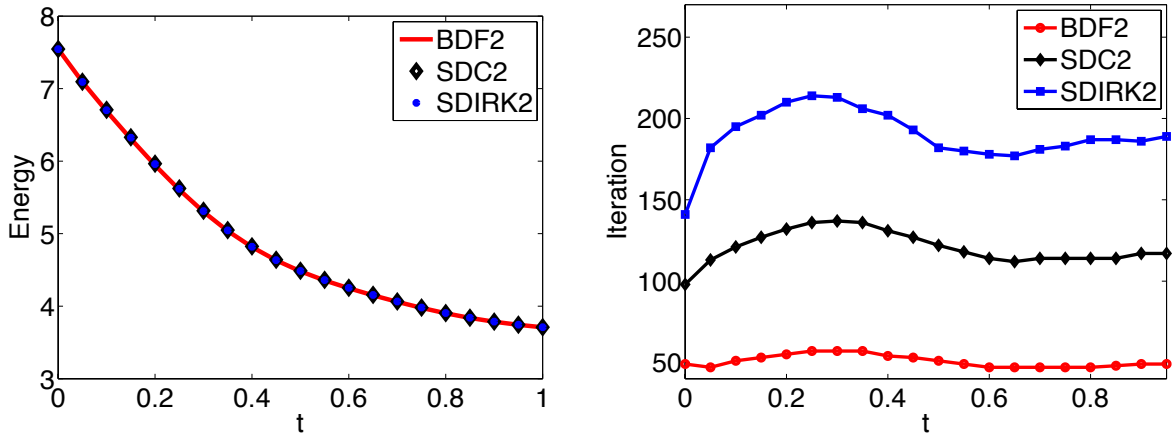
$$(u_1^*, u_2^*) = \{(\cos(\theta_i), \sin(\theta_i))\,|\theta_1 = 0, \theta_2 = \frac{2\pi}{3}, \theta = -\frac{2\pi}{3}\} \equiv \{(1,0), (-\frac{1}{2}, \frac{\sqrt{3}}{2}), (-\frac{1}{2}, -\frac{\sqrt{3}}{2})\}$$

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

$$\mathcal{J}_{\nabla_{\mathbf{u}} W}(u_1^*, u_2^*) \equiv \begin{bmatrix} \frac{\partial^2 W}{\partial u_1^2} & \frac{\partial^2 W}{\partial u_1 u_2} \\ \frac{\partial^2 W}{\partial u_2 u_1} & \frac{\partial^2 W}{\partial u_2^2} \end{bmatrix}_{(u_1^*, u_2^*)} = \begin{bmatrix} 18 & 0 \\ 0 & 18 \end{bmatrix}, \tag{3.39}$$

	BDF2		SDIRK2		SDC2	
Δt	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

Table 3.4: Refinement studies of second-order methods for the 2D CH equation with periodic BC.



(a) Energy decay (2^{nd} order methods)

(b) Iteration count(2^{nd} order methods)

Figure 3.3: Energy descent and nonlinear iteration count history of all second-order 2D methods with fixed time step $\Delta t = 0.05$ and with parameters (3.38).

thus, all points are stable equilibrium solutions [57]. We subract $\mathbf{u} = (u_1, u_2)$ 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 (3.3),

$$\mathbf{v}_t = -\epsilon^2 \Delta^2 \mathbf{v} + \Delta \nabla_{\mathbf{v}} W(\mathbf{v} + \mathbf{z_3}) \equiv -\epsilon^2 \Delta^2 \mathbf{v} + \Delta \left(\nabla_{\mathbf{v}} \tilde{W}(\mathbf{v}) + 18\mathbf{v} \right) \tag{3.40}$$

where $\tilde{W}_{v_1}(\mathbf{v}) := W_{v_1}(\mathbf{v} + \mathbf{z_3}) - 18v_1$ and $\tilde{W}_{v_2}(\mathbf{v}) := W_{v_2}(\mathbf{v} + \mathbf{z_3}) - 18v_2$. We will develop the time marching scheme using this transformed system (3.40) in the following Section.

3.5.1 Two nonlinear iterative schemes

We first apply the Backward Euler scheme to the transformed system (3.40), then

$$\left(I - 18\Delta t \Delta + \epsilon^2 \Delta t \Delta^2 \right) \mathbf{v}^{n+1,k+1} = \mathbf{v}^n + \Delta \nabla_{\mathbf{v}} \tilde{W}^{n+1,k} \tag{3.41}$$

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

$$\Delta t \geq \frac{\epsilon^2}{81} : \quad I - 18\Delta t \Delta + \epsilon^2 \Delta t \Delta^2 = \left(I - \frac{\Delta}{\alpha_1^2} \right) \left(I - \frac{\Delta}{\alpha_2^2} \right), \frac{1}{\alpha_i^2} = 9\Delta t \pm \sqrt{81\Delta t^2 - \epsilon^2 \Delta t}, \tag{3.42}$$

$$\Delta t < \frac{\epsilon^2}{81} : \quad I - 2\sqrt{\epsilon^2 \Delta t}\Delta + \epsilon^2 \Delta t \Delta^2 = \left(I - \frac{\Delta}{\alpha^2} \right)^2, \qquad \frac{1}{\alpha^2} = \sqrt{\epsilon^2 \Delta t}. \tag{3.43}$$

We can handle above operators with the same strategy in Section 3.4.1. The higher order expansions are analogous in Section 3.4.2 to each component v_1 and v_2 respectively. As expected, the stabilized fixed point iteration (3.42) also allow us to take large time steps

without forgoing the energy stability property (3.5).

3.5.2 Numerical test: refinement studies of 2D VCH solutions

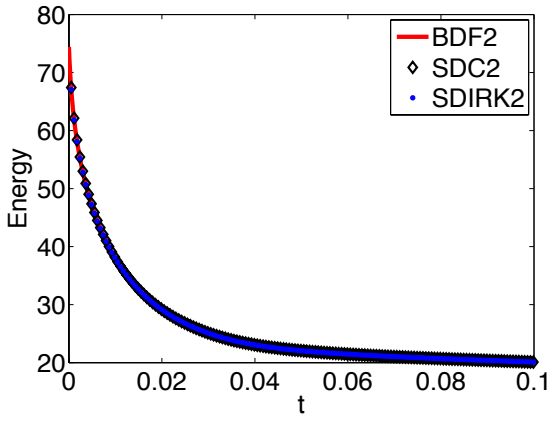
In this Section, we implement our second-order scheme to 2D system (3.3) for vector CH model. In order to look spinodal evolution of phase function $\mathbf{u} = (u_1, u_2)$, we begin with the same initial condition (3.37) for u_1 and

$$u_2(x, y, 0) = \sin(y), \qquad (x, y) \in [0, 2\pi]^2 \tag{3.44}$$

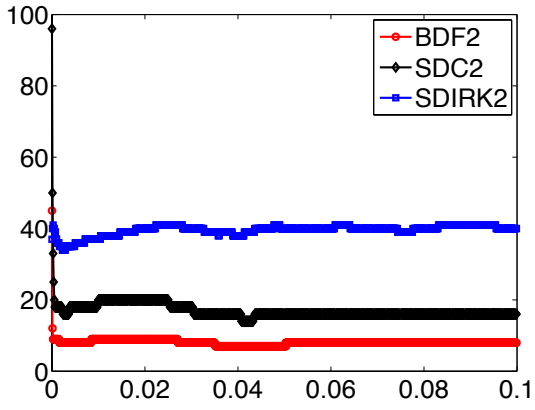
for u_2 in the reference [16]. With the parameters

$$\epsilon = 0.32, \quad \Delta x = \Delta y = \frac{2\pi}{128} \approx 0.0491, \quad N_{\max\,\mathrm{it}} = 500, \quad N_{\mathrm{tol}} = 10^{-6}, \tag{3.45}$$

we implement BDF2, SDC2, and SDIRK2 methods for VCH system during $0 \le t \le T_{\mathrm{final}} = 0.1$ with fixed time step size $\Delta t = 0.000125$.



(a) Energy decay (2^{nd} order methods)



(b) Iteration count(2^{nd} order methods)

Figure 3.4: Energy descent and nonlinear iteration count history of all second-order VCH 2D methods with fixed time step $\Delta t = 0.000125$ and with parameters (3.45).

As shown in Figure 3.4a, every scheme preserves the discrete form of the energy law during the spinodal evolution of vector CH model. Figure 3.4b is obtained by iteration count at each time such that $\max\left(\|u_1^{n+1,k+1} - u_1^{n+1,k}\|_\infty, \|u_2^{n+1,k+1} - u_2^{n+1,k}\|_\infty\right) < N_{\text{tol}}$. The result indicates that BDF2 is most efficient, as expected, thus it requires least number of nonlinear iteration while all have same convergence tolerance $N_{\text{tol}} = 10^{-6}$. In fact, computational time using built-in MATLAB function (tic-toc) of BDF2 simulation was 161.09 seconds, while SDC2 is 373.74 second and SDIRK2 is 803.15 seconds.

In this simulation, we used small time step size to accrately simulate the spinodal phase. However, we also present numerically that our schemes allow large time steps and iterative schemes converge in Section 3.7.3.

3.6 Time adaptive strategy

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 (3.1) 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 model, which is called *ripening* process. Simulating these phenomena with a fixed time stepping necessarily become inefficient, and so we suggest an adaptive time stepping strategy combined with presented 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 in [16]. On the other hand, the Richardson extrapolation (known

Algorithm 1 Adaptive time step-size control

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

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

2. Define a tolerance σ_{tol} for the above LTE. If accuracy fails ($\eta > \sigma_{\text{tol}}$), then the time step fails and repeat with the reduced time step. If accuracy success, then we also test the following criteria for time step-size selection,

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

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

   (II) $\eta > \sigma_{\text{tol}}$: Step fails. Try with shrink step-size

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

   where γ and θ are safety factors in [16].

as step-doubling) or embedded Runge-Kutta pairs can be used [17]. We adopt Richardson extrpolation as well our second-order scheme, BDF2 and SDC2, to approximate the LTE. We also use the time step-size selection criteria presented in [16], which is summarized below Algorithm 1.

In practice, the 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.

3.7 Numerical tests

In this Section, we present adaptive time stepping results using the previously developed dimensional-split MOL$^{\mathrm{T}}$ schemes. As mentioned in Section 1.2.1.3, we will reproduce novel benchmark problems in [16] using our schemes. We will compare several fixed time-stepping schemes with the adaptive time stepping schemes via numerical results.

3.7.1 1D Cahn-Hilliard Model

In this Section, we first solve the 1D CH equation (3.1) with a stiff initial condition (3.25) ($\epsilon = 0.18$). The second perturbation term of this initial state creates two intervals, $u = -1$ and $u = +1$, are asymmetric, so that 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 [16]. The aim of this simulation is to accurately capture all time scales using both fixed and adaptive time stepping strategies. (The spatial step-size is fixed with $\Delta x = \frac{2\pi}{128} \approx 0.05$.)

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 $N_{\mathrm{tol}} = 10^{-11}$ at each step, and the ripening times are presented in Table 3.5. Our results agree will with the reference time $T_{\mathrm{r}} = 8318.63$ in [16].

We also compare the ripening time using several schemes, with larger fixed time steps in Table 3.6. Among the three methods, BDF2 is the most efficient, and provides better estimates 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 gives fairly accurate evolution. 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.

Thus, we consider adaptive time stepping for the same problem. In Section 3.6, we explained that there might be several ways to approximate the local truncation error (LTE). With our schemes, we use three different methods for the LTE approximation: Richardson extrapolation based on first-order BE scheme (in Algorithm 1); or BE combined with BDF2 (BE-BDF2) or combined with SDC2 (BE-SDC2). We note that we do not consider SDIRK2, since it has already proved that this scheme is inefficient than two others in Section 3.3.4.

We implement three methods with the same fixed point residual tolerance $N_{\mathrm{tol}} = 10^{-11}$, $N_{\max \mathrm{it}} = 600$ in Algorithm 1, but with various error tolerance δ_{tol}. The performance of the time-adaptive scheme is shown in Table 3.7, which are more accurate and efficient than previous fixed time-stepping methods. In particular, BE-BDF2 combination for LTE approximation is the fastest, as well as the most accurate, in this simulation.

The phase function $u(x, t)$ obtained by our adaptive time stepping (BE-BDF2) scheme

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

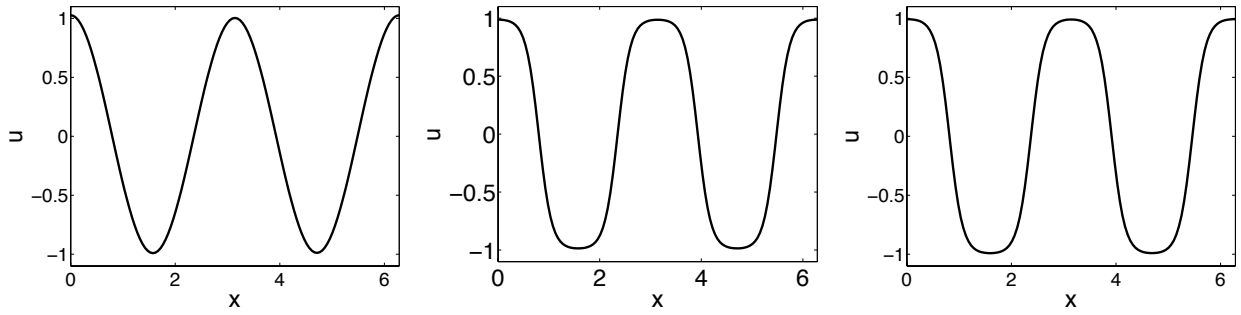
Table 3.5: Ripening time of 1D CH equation with small fixed time step size ($\Delta t = 0.01$) and with the periodic BC.

	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

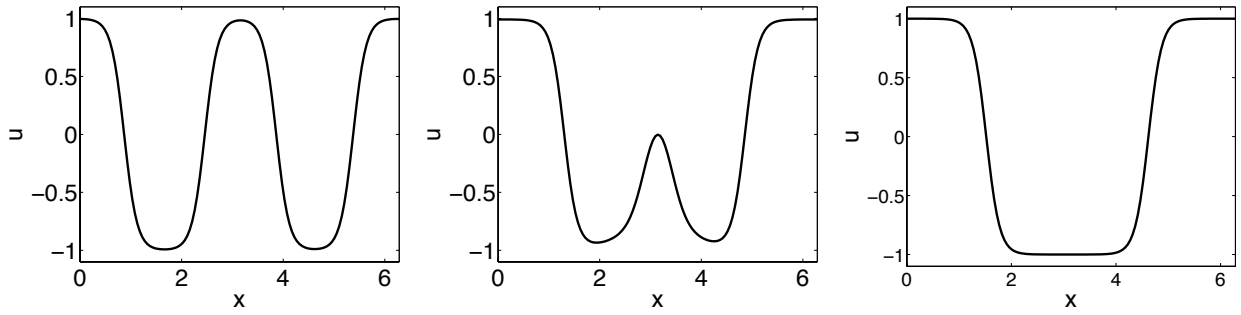
Table 3.6: Ripening time of 1D CH equation with large time steps ($\Delta t = 0.05, 1, 10$) and with the periodic BC.

	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

Table 3.7: Ripening time of 1D CH equation with adaptive time step size and with the periodic BC.



(a) $u(x, 0)$ (b) $u(x, 0.51)$ (c) $u(x, 3669.8)$

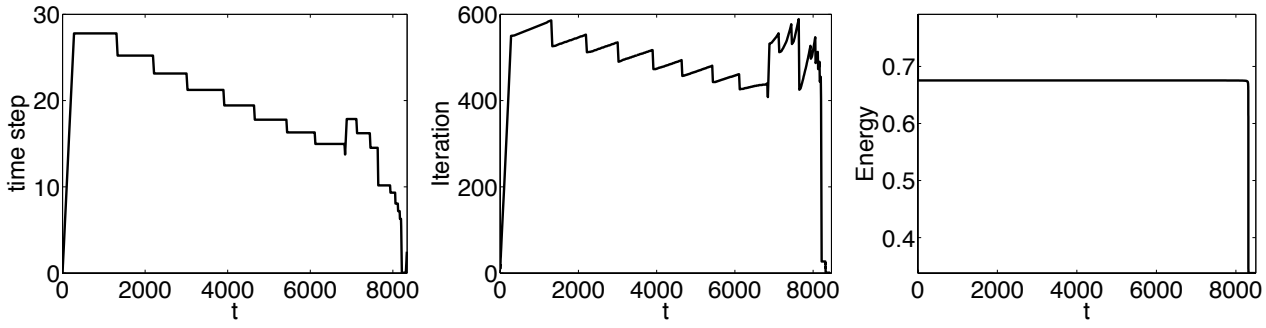


(d) $u(x, 7005.7)$ (e) $u(x, 8317.9)$ (f) $u(x, 8319.2)$

Figure 3.5: Temporal evolution of the 1D CH solution from the adaptive time step (BE-BDF2) scheme.

is shown in Figure 3.5. The initial state 3.5a quickly moves to the metastable state 3.5b, and then finally reaches the stable state 3.5f at which two layers are merged together after a very slow time scale. This simulation also shows that the ripening event happens over a very fast time scale in Figure 3.5e.

In Figure 3.6, we also plot the history of time step size, iteration count at each time level, and discrete energy during time evolutions of our numerical solution which is obtained by adaptive time stepping method (BE-BDF2). As shown in Figure 3.6a, small time steps are used at early stage (spinodal evolution) but increase when the coarsening process starts, which speed up the simulation. Also, if iteration count is too large ($\frac{N_{\text{it}}}{N_{\max \text{it}}} \geq 1$), we reject the solution and compute $u(x, t)$ again with the reduced time step Δt. We see this behavior in Figure 3.6b, where time steps are decreased whenver $N_{\text{it}} \approx N_{\max \text{it}}$. We also observe in



(a) Time step size (b) Iteration count (c) Energy

Figure 3.6: The time step, iteration count, and energy history of our adaptive time step (BE-BDF2) scheme for 1D CH equation.

Figure 3.6c that adaptive time stepping does not affect energy decay.

3.7.2 2D Cahn-Hilliard Model

We next solve the CH equation (3.1) in two spatial dimension. With the parameters

$$\epsilon = 0.18, \quad \Delta x = \Delta y = \frac{2\pi}{128} \approx 0.0491, \quad N_{\max\,\mathrm{it}} = 500, \quad N_{\mathrm{tol}} = 10^{-7},$$

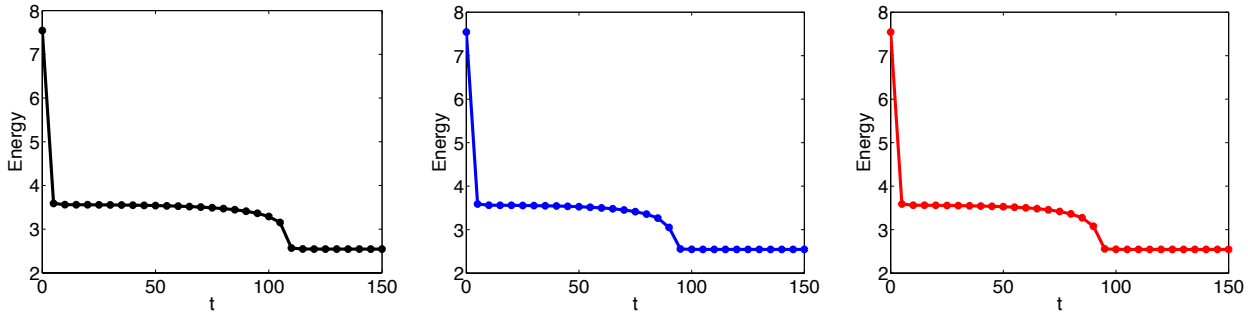
we implement the initial condition (3.37), using both the fixed and adaptive (BE-BDF2) time stepping methods.

We first implement fixed time stepping by using the first-order (BE), second-order (BDF2), and the third-order (BDF3) methods first with a small time step ($\Delta t = 0.01$) and check the ripening time. The numerical ripening time is defined at which $u(\frac{\pi}{2}, \frac{\pi}{2})$ changes from positive to negative, because the point $(\frac{\pi}{2}, \frac{\pi}{2})$ is the center of the smaller circular region of $u = 1$, which will be consumed by the larger one after a long time scale. Based on numerical results in Table 3.8, we can define such ripening time as $T_{\mathrm{r}} = 80.07$.

Each time stepping method combined with dimensional-splitting guarantees that the

	Time stepping	Ripening time	Total iteration	Times(s)
$\Delta t = 0.01$	BE	80.04	167,086	2025.96
	BDF2	80.07	149,500	1883.72
	BDF3	80.07	142,240	1793.91

Table 3.8: Numerical ripening time of the 2D CH equation with fixed time stepping methods ($\Delta t = 0.01$, $N_{\text{tol}} = 10^{-7}$) and with the periodic BCs.

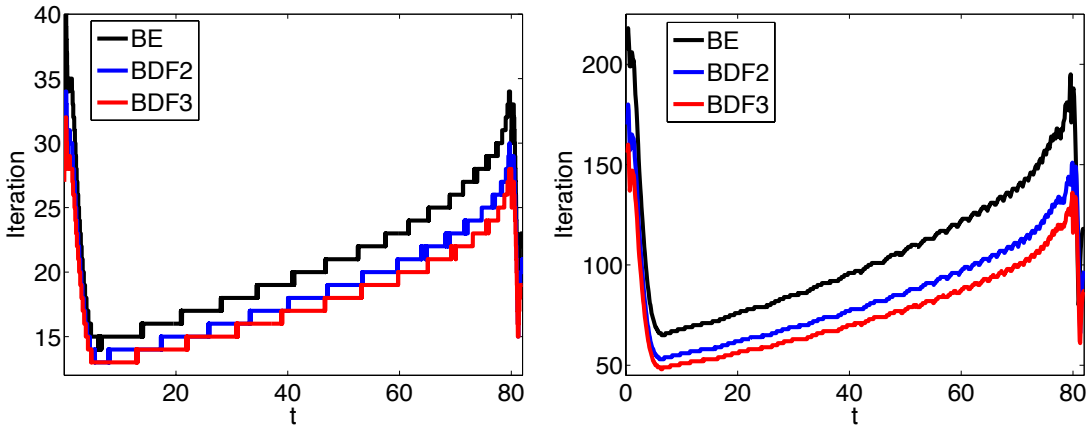


(a) Energy (BE) (b) Energy (BDF2) (c) Energy (BDF3)

Figure 3.7: The discrete energy of fixed time stepping schemes ((a)BE, (b)BDF2, and (c)BDF3) for 2D CH equation. The common parameters are used: $\Delta t = 5$, and $\Delta x = \Delta y = 0.05$

solution is decaying in energy, which shown in Figure 3.7, even where the CFL is 100, ($\Delta t = 5$, and $\Delta x = \Delta y = 0.05$). By these results, we strongly believe our higher-order dimensional splitting MOL$^{\text{T}}$ methods are unconditional gradient stable in practice.

In addition, it is important to note that that raising the order of the scheme (temporal accuracy) reduces the number of iteration per time step, especially at the ripening moment, as shown in Figure 3.8. Thus, the overall computation time of higher-order method is lower than low-order schemes, summarized in Table 3.8. We also implement the same time stepping methods with a larger time step $\Delta t = 0.1$. As shown in Figure 3.8b, each of iteration counts per time step is higher than the corresponding one with smaller time step (in Figure 3.8a). We believe that this is reasonable results because of the accuracy requirement. Moreover, the higher-order time stepping scheme exhibits the better efficiency like the smaller time



(a) $\Delta t = 0.01$ (b) $\Delta t = 0.1$

Figure 3.8: Iteration count of 2D CH solution obtained by the fixed time steppping method (a) ($\Delta t = 0.01$), and (b) ($\Delta t = 0.1$). Black line is obtained by Backward Euler(BE) scheme; blue is BDF2; and the red one is BDF3 at both plots.

step case. With these tests, we strongly believe that higher-order time stepping method is preferable to resolve the interfacial structure of these types of the problems, regardless of time step sizes.

However, $\Delta t = 0.1$ or even $\Delta t = 0.01$ is still large for this CH problem, especially during spinodal composition, so that it is possible to damp out high frequency contributions in the evolving field, which is somewhat unphysical. Thus, we next simulate the same problem with adaptive time stepping method (BE-BDF2). We first compute the numerical ripening time, total iteration number during temporal evolution, and the computational time of our adaptive time stepping method, shown in Table 3.9. When we compare this result with the fixed time stepping methods, the adaptive scheme is better with respect to the accracy, as expected. The reason why the computational time of adaptive scheme is a little larger than higher-order schemes is it uses very small time step size during spinodal evolution. Thus, we realize that the adaptive time stepping method is necessary for the CH model in multiple

	Time stepping	Ripening time	Total iteration	Times(s)
$\Delta t = 0.1$	BE	79.80	84,522	1125.69
	BDF2	80.10	68,322	844.39
	BDF3	80.10	61,548	770.67
Adaptive time	BE-BDF2	80.083	66,085	918.53

Table 3.9: Numerical ripening time of the 2D CH equation (the periodic BCs): Comparison between the fixed time stepping methods ($\Delta t = 0.1$, $N_{\text{tol}} = 10^{-7}$) and the adaptive time stepping method (BE-BDF2).

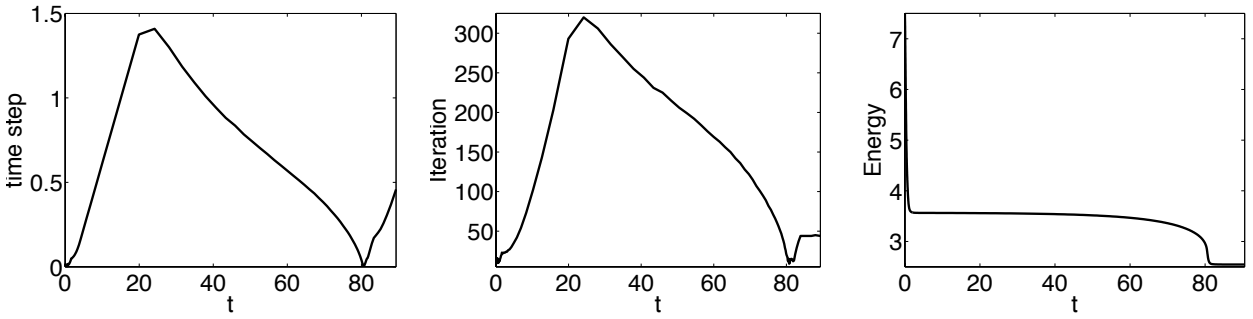
spatial dimension, as well.

Like 1D case, the time step size, iteration count, and energy history are presented in Figure 3.9. Our results agree with those of Section 3.7.1, in that larger time step sizes are used where the transition layers are varying slowly, and smaller steps are used where the layers vary rapidly (see Figure 3.9a), so that we believe our scheme captures various time scales of 2D CH solution accurately. The energy in Figure 3.9c indicates that there are two sharp transitions in the energy $\mathcal{E}$; early on, and at ripening. Finally, we also observe the desired energy decaying property of our numerical solution.

The contour plots of time evolution of the phase function $u(x, y, t)$ are shown in Figure 3.10. The initial states 3.10a 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 3.10c - 3.10e. The final state is shown in 3.10f, where the larger region has fully consumed the smaller. In all plots, the total volume looks to be preserved. The computational ripening time is $T_{\text{r}} \approx 80.0834$ in this simulation.

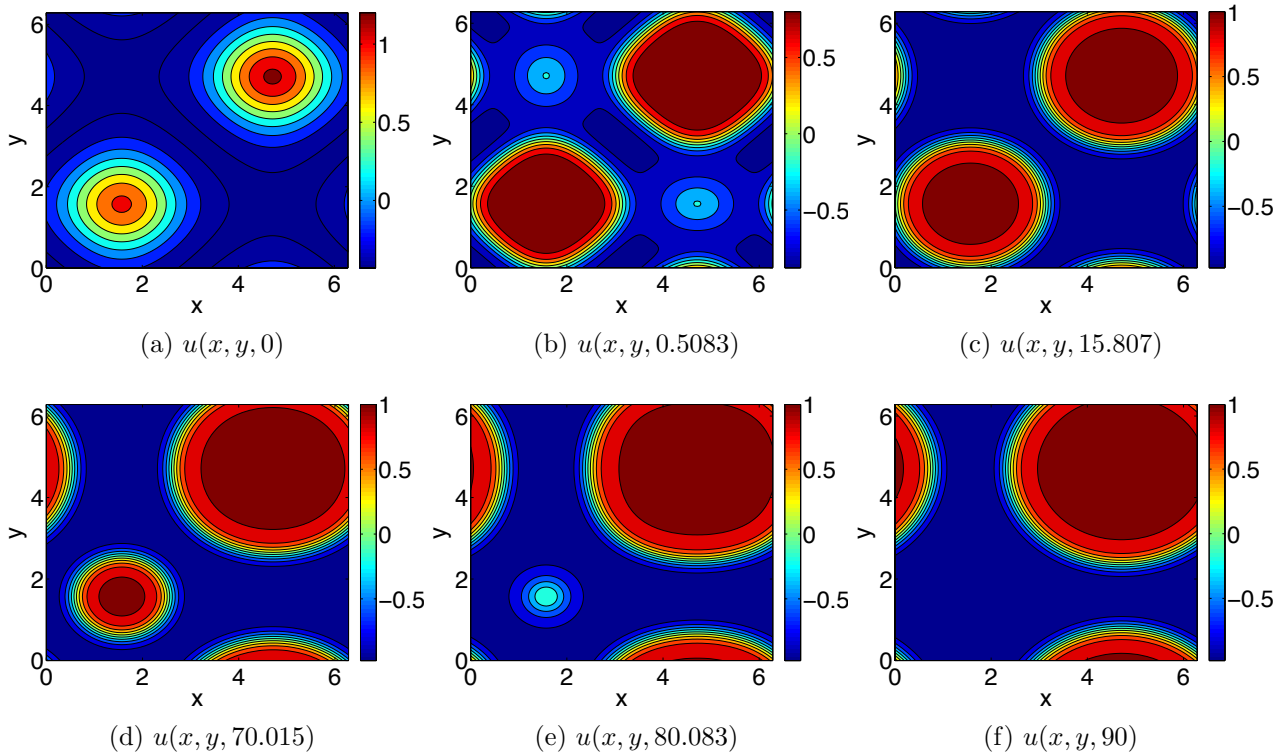
3.7.3 2D vector Cahn-Hilliard Model

We now apply our first- and second-order time stepping methods from Section 3.5 to the 2D vector CH system (3.3) combined with our adaptive time stepping (BE-BDF2) strategy. In



(a) Time step size (b) Iteration count (c) Energy

Figure 3.9: The time step, iteration count, and energy history of our adaptive time step (BE-BDF2) scheme for 2D CH equation.



(a) $u(x, y, 0)$ (b) $u(x, y, 0.5083)$ (c) $u(x, y, 15.807)$

(d) $u(x, y, 70.015)$ (e) $u(x, y, 80.083)$ (f) $u(x, y, 90)$

Figure 3.10: Temporal evolution of the 2D CH solution with the initial (3.37).

.

this Section, we observe the long time behavior of the phase function $\mathbf{u} = (u_1, u_2)$, using the same initial condition for u_1 and u_2 in Section 3.5.1. With the parameters

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

we implement the adaptive time stepping method (BE-BDF2) for vector CH (VCH) model.

First of all, the time step, number of nonlinear iteration and energy history results of 2D VCH solution are presented in Figure 3.11. Our results agree with previous Sections, we also observe the desired energy decay in Figure 3.11c.

In addition, the contour plots of $\cos(\arg(u_1 + iu_2))$ are shown in Figure 3.12. Instead of plotting u_1 and u_2 separately, [16] we define the angle $\theta \equiv \arg(u_1 + iu_2)$ at a triple juction, and plot $\cos(\theta)$ to avoid any discontinuities. We follow this benchmark plot using our numerical solution.

After some initial ripening in Figure 3.12b, the interfaces dividing the three states $\mathbf{u} = \mathbf{z_i}$ $(i = 1, 2, 3)$ form around $T = 0.5$. In Figure 3.12c - 3.12f, 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 3.12e. Again, we believe that the volume is preserved over all time steps. For a more involved discussion of this simulation, we refer the interested reader to [16]. Our goal here is to reproduce the same results in the reference, but based on our dimensional-split MOL^{T} scheme and confirm that our scheme can capture the correct ripening behavior in multiple spatial domain. Moreover, the numerical results what we have done in this Section prove that our scheme can be extensible to vector system. Future work will investigate the extension of these second- and third- order time stepping schemes of vector model to apply the multicomponent bilayer

structures arise in membrane in biology [51].

3.7.4 2D sixth order Model

In this Section, we introuduce one of our ongoing research projects. We also consider the sixth order phase-field problem [16]:

$$u_t = \Delta\left[(\epsilon^2\Delta - f'(u) + \epsilon^2\eta)(\epsilon^2\Delta u - f(u))\right], \quad f(u) = u^3 - u. \tag{3.46}$$

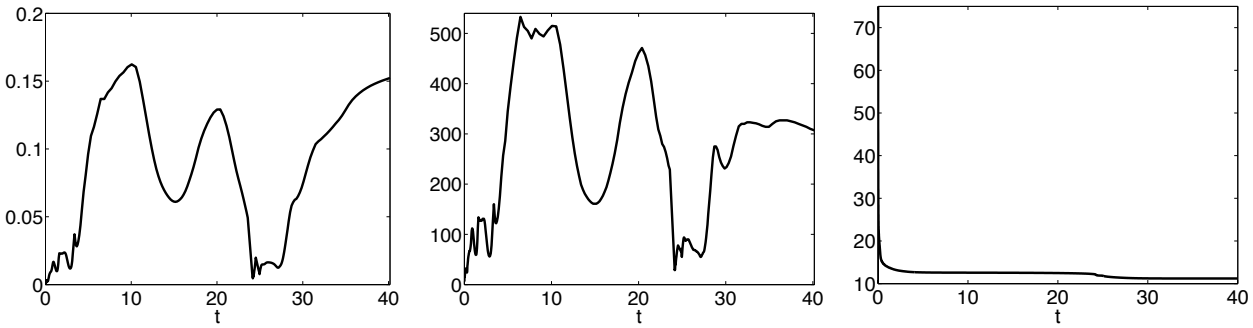
where ϵ and η are given positive constants. (The sign of η is important since minus η can form the phase interface) This problem is motivated by the functionalized Cahn-Hilliard (FCH) equation [29, 21, 37, 40] which models interfacial energy in amphiphilic phase-separated mixtures. In the original FCH model, $\eta_1 > 0$ and $\eta_2 \in \mathbb{R}$, which are functionalization term that are analogous to the surface and volume energies typical of models of charged solutes, but we have simplified to the case $\eta = \epsilon\eta_1 = \epsilon\eta_2$. Later, we will consider the solution of original FCH equation with our MOL$^{\mathrm{T}}$ formulation.

With the similar approaches to CH type equations, we first introduce the transformed variable $v = u + 1$ and substitues in (3.46),

$$v_t = \Delta\left[(\epsilon^2\Delta - f'(v) + \epsilon^2\eta)(\epsilon^2\Delta v - f(v))\right], \quad f(v) = v^3 - 3v^2 + 2v. \tag{3.47}$$

Apply the backward Euler (BE) scheme for time discretization of (3.47),

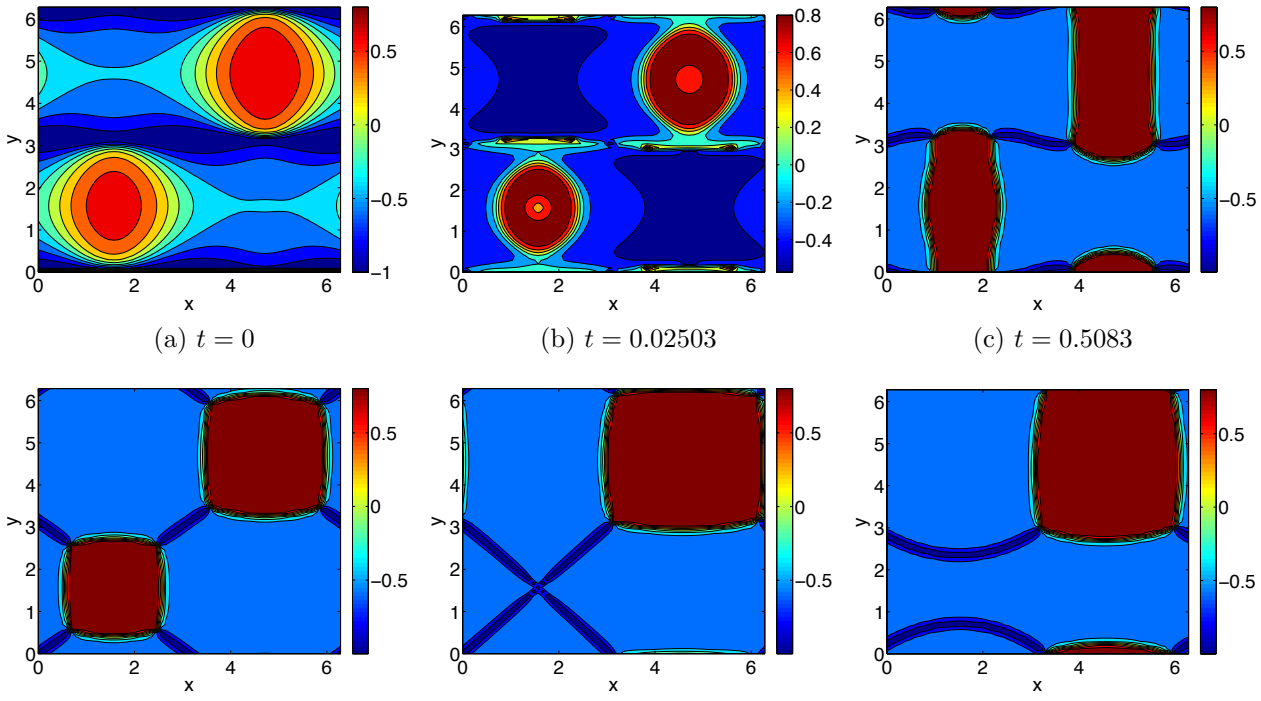
$$\begin{aligned}\left(I - \Delta t\epsilon^4\Delta^3\right)v^{n+1} = v^n - \Delta t\Delta\left(\epsilon^2\Delta f^{n+1} - \epsilon^2(f'\Delta v)^{n+1} + (ff')^{n+1}\right)\\ - \eta\epsilon^2\Delta t\Delta\left(\epsilon^2\Delta v^{n+1} - f^{n+1}\right).\end{aligned}$$



(a) Time step size (b) Iteration count (c) Energy

Figure 3.11: The time step, iteration count, and energy history of VCH solution

.



(a) $t = 0$ (b) $t = 0.02503$ (c) $t = 0.5083$

(d) $t = 5.0848$ (e) $t = 24.286$ (f) $t = 30.021$

Figure 3.12: Temporal evolution of the 2D CH vector solution with the initial (3.37) for u_1 and (3.44) for u_2. Contours of $\cos(\arg u_1 + iu_2)$ are plotted. Two phases $\mathbf{u} = \mathbf{z_2}$ and $\mathbf{u} = \mathbf{z_3}$ have same cosine values, $-\frac{1}{2}$ (light blue in the plots) but are separated by dark blue lines.

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

$$\left(I - \sqrt[3]{\Delta t\epsilon^4}\Delta\right)^3 v^{n+1} = v^n - \Delta t\Delta\left(\epsilon^2\Delta f^{n+1} - \epsilon^2(f'\Delta v)^{n+1} + (ff')^{n+1}\right) - \eta\epsilon^2\Delta t\Delta\left(\epsilon^2\Delta v^{n+1} - f^{n+1}\right) - \left(3\sqrt[3]{\Delta t\epsilon^4}\Delta - 3(\sqrt[3]{\Delta t\epsilon^4})^2\Delta^2\right)v^{n+1}. \tag{3.48}$$

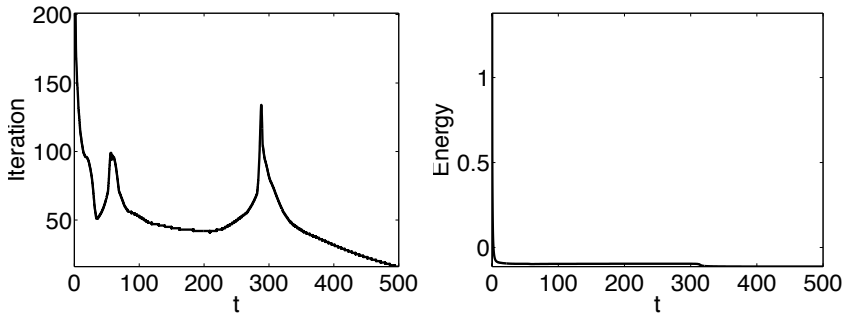
We can now solve v^{n+1} by applying the triple inversion of our modified Helmhotlz operator $\mathcal{L} = I - \sqrt[3]{\Delta t\epsilon^4}\Delta$ by the same procedure in Section 3.4.1.

For simplicity, we apply the first-order BE scheme (3.48) with the fixed time step $\Delta t = 0.1$ to solve the 6^{th} order problem (3.47) ($\eta = 1$). By starting with the same initial condition (3.37) ($\epsilon = 0.18$), and with the following parameters:

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

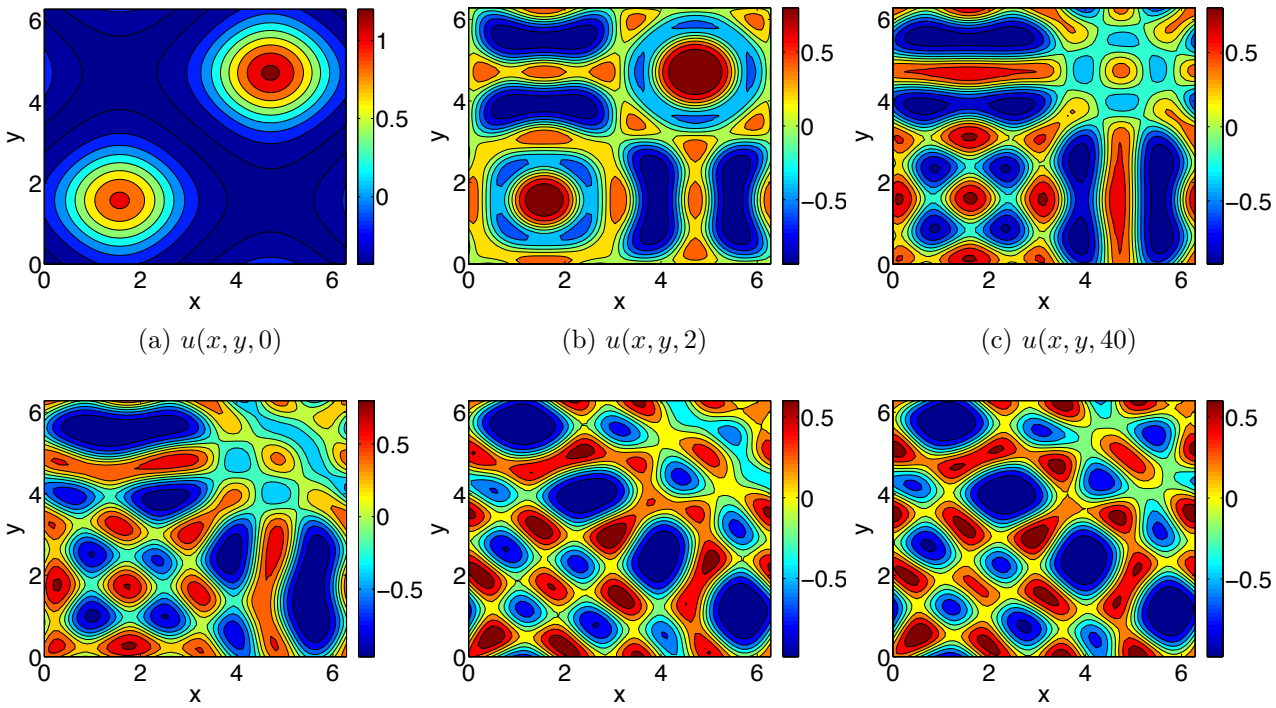
The contour plots of the temporal evolutions of our numerical solution $u(x, y, t)$ of (3.46) are shown in Figure 3.14. After solution varies rapidly, then solution seems to move very slowly. As expected [16], we believe that our numerical solution reaches to the final steady state around $t \approx 320$, which has formed the regular array in Figure 3.14f and the corresponding discrete energy decreased at that moment in Figure 3.13b.

Future work will consider the switching-scheme for the sixth-order problem, and extend to the adaptive time stepping strategy, similar with the previously presented CH solutions.



(a) Iteration count (b) Energy

Figure 3.13: The iteration count and energy history of the first-order solution of sixth-order equation (3.47).



(a) $u(x, y, 0)$ (b) $u(x, y, 2)$ (c) $u(x, y, 40)$

(d) $u(x, y, 150)$ (e) $u(x, y, 300)$ (f) $u(x, y, 500)$

Figure 3.14: Temporal evolution of the 2D sixth order model's solution with the initial (3.37).

Chapter 4

Conclusion and Future work

In this work, we have proposed the Method of lines transpose (MOL^{T}) formulation for nonlinear problems such as Allen-Cahn and Cahn-Hilliard equations. The reason we have done is that the semi-discrete formulation gives better favorable stability property than a traditional Method of lines (MOL) formulation. Then, we have proved the successive convolution (resolvent expansion) could achieve arbitrary order in time, giving us great time accuracy. Moreover, we have compared more traditional time stepping algorithms, such as backward difference formula (BDF), singly diagonally implicit Runge kutta (SDIRK), and spectral deferred correction (SDC), to achieve high order of accuracy for Cahn-Hilliard models. Furthermore, the adaptive time stepping is still favorable to capture the scale separation of CH model. Specifically, small time steps are used at the spinodal phase, but large steps are used at the coarsening process in which phase-separated domains merge into larger domains.

In order to handle the spatial component of the problems, typically the fast Fourier method are chosen, but this only guarantees the periodic boundary conditions. Instead, we have introduced a dimensional-split kernel, which is efficient $\mathcal{O}(N)$, and matrix-free scheme. Not only we can address other boundary conditions, we can raise arbitrary order of accuracy in space. This also promises the parallel efficiency for modern parallel multicore computing because of the dimensional splitting strategy.

Especially, the dimensional splitting also has a framework that we can extend our 1D

	Tools	Results
Time stepping	MOL^{T} discretization Successive convolution BDF/SDIRK/SDC Time-adaptive	Stability in semi-discrete setting High-order in time Compare strategies Capture various time scales
Spatial scheme	Fast Convolution	Flexibility with BCs Arbitrary order of accuracy in space Fast $\mathcal{O}(N)$ computation Parallel efficiency
Multi-D scheme	Dimensional splitting	Easy extension to multi-D scheme High-order space and time

Table 4.1: Summarization of numerical results of presented scheme, dimensional split MOL^{T} scheme.

scheme to multiple spatial dimensions. In 2D, we have achieved the same results with 1D, i.e. flexible, efficient, can achieve high-order of accuracy, and address various boundary conditions. One might wonder the splitting error introduced by the dimensional splitting, however, we have also addressed the splitting error in our iterative solver. We summarize the numerical strategies we have done in this work and resulting positive effects what we have got in the Table 4.1, which is easy to see a a glance our works.

In future work, parallel implementation will be the focus of our upcoming work in this area. Moreover, we will investigate employing our scheme to the higher-order derivative models, such as (2D and 3D) FCH models in [40]. In this work, we have done fixed time stepping method for the sixth order parabolic equations, and the next goal should employ adaptive time stepping, which can allow increased time steps to solve such higher-order derivative models. We also want to make use of the implicit solvers ability to handle complex boundary geometries of various phase-field models.

We have not yet tried comparison study between other boundary integral approximation to solve the modified Helmholtz equations in the MOL^{T} semi-discrete schemes, such as FFM

in Section 1.2.2.1 or Treecode algorithm [45]. Besides our dimensional splitting algorithm, we will implement other boundary integral solvers, and compare which solvers gain efficiency and accuracy in parallel computing.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] L. Abadias and P. J. Miana. c_0-semigroups and resolvent operators approximated by laguerre expansions. *arXiv preprint arXiv:1311.7542*, 2013.

[2] R. Alexander. Diagonally implicit runge-kutta methods for stiff ode's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[3] S. M. Allen and J. W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

[4] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual review of fluid mechanics*, 30(1):139–165, 1998.

[5] 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.

[6] O. P. Bruno and M. Lyon. High-order unconditionally stable fc-ad solvers for general smooth domains i. basic elements. *Journal of Computational Physics*, 229(6):2009–2033, 2010.

[7] J. W. Cahn. On spinodal decomposition. *Acta metallurgica*, 9(9):795–801, 1961.

[8] 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.

[9] J. Carr and R. L. Pego. Metastable patterns in solutions of ut= 2uxx- f (u). *Communications on pure and applied mathematics*, 42(5):523–576, 1989.

[10] M. Causley, A. Christlieb, Y. Güçlü, and E. Wolf. Method of lines transpose: an efficient a-stable solver for wave propagation. *arXiv preprint arXiv:1511.01013*, 2015.

[11] 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.

[12] 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.

[13] 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.

[14] L. Chen and J. Shen. Applications of semi-implicit fourier-spectral method to phase field equations. *Computer Physics Communications*, 108(2):147–158, 1998.

[15] L.-Q. Chen. Phase field models for microstructure evolution. *Annual review of materials research*, 32(113–140), 2002.

[16] 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.

[17] 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.

[18] 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.

[19] A. J. Christlieb, Y. Liu, and Z. Xu. High order operator splitting methods based on an integral deferred correction framework. *Journal of Computational Physics*, 294:224–242, 2015.

[20] S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

[21] S. Dai and K. Promislow. Geometric evolution of bilayers under the functionalized cahn–hilliard equation. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 469, page 20120505. The Royal Society, 2013.

[22] 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.

[23] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.

[24] K.-J. Engel and R. Nagel. *One-parameter semigroups for linear evolution equations*, volume 194. Springer Science & Business Media, 2000.

[25] L. C. Evans. *Partial Differential Equations: Second Edition*. American Mathematical Society, 2010.

[26] D. J. Eyre. An unconditionally stable one-step scheme for gradient systems. *Unpublished article*, 1998.

[27] G. Fairweather and A. Mitchell. A new computational procedure for adi methods. *SIAM Journal on Numerical Analysis*, 4(2):163–170, 1967.

[28] P. C. Fife. *Mathematical aspects of reacting and diffusing systems*, volume 28. Springer Science & Business Media, 2013.

[29] N. Gavish, J. Jones, Z. Xu, A. Christlieb, and K. Promislow. Variational models of network formation and ion transport: applications to perfluorosulfonate ionomer membranes. *Polymers*, 4(1):630–655, 2012.

[30] J. W. Gibbs, H. A. Bumstead, and W. R. Longley. *The collected works of J. Willard Gibbs*, volume 1. Longmans, Green and Company, 1928.

[31] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

[32] L. F. Greengard and J. Huang. A new version of the fast multipole method for screened coulomb interactions in three dimensions. *Journal of Computational Physics*, 180(2):642–658, 2002.

[33] V. Grimm and M. Gugat. Approximation of semigroups and related operator functions by resolvent series. *SIAM Journal on Numerical Analysis*, 48(5):1826–1845, 2010.

[34] A. Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.

[35] 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.

[36] S. Jiang, L. Greengard, and S. Wang. Efficient sum-of-exponentials approximations for the heat kernel and their applications. *Advances in Computational Mathematics*, 41(3):529–551, 2015.

[37] J. S. Jones. *Development of a fast and accurate time stepping scheme for the functionalized Cahn-Hilliard equation and application to a graphics processing unit.* PhD thesis, MICHIGAN STATE UNIVERSITY, 2013.

[38] J. Kačur. *Method of Rothe in evolution equations.* Springer, 1986.

[39] J. P. Keener and J. Sneyd. *Mathematical physiology*, volume 1. Springer, 1998.

[40] 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.

[41] V. Krinsky and A. Pumir. Models of defibrillation of cardiac tissue. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(1):188–203, 1998.

[42] 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.

[43] L. D. Landau and E. Lifshitz. On the theory of the dispersion of magnetic permeability in ferromagnetic bodies. *Phys. Z. Sowjetunion*, 8(153):101–114, 1935.

[44] H. G. Lee and J.-Y. Lee. A semi-analytical fourier spectral method for the allen–cahn equation. *Computers & Mathematics with Applications*, 68(3):174–184, 2014.

[45] K. Lindsay and R. Krasny. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *Journal of Computational Physics*, 172(2):879–907, 2001.

[46] F. Liu and J. Shen. Stabilized semi-implicit spectral deferred correction methods for allen-cahn and cahn-hilliard equations. *Math. Methods Appl. Sci*, 2013.

[47] C. Lubich and R. Schneider. Time discretization of parabolic boundary integral equations. *Numerische Mathematik*, 63(1):455–481, 1992.

[48] M. Lyon and O. P. Bruno. High-order unconditionally stable fc-ad solvers for general smooth domains ii. elliptic, parabolic and hyperbolic pdes; theoretical considerations. *Journal of Computational Physics*, 229(9):3358–3381, 2010.

[49] S. P. Nørsett. One-step methods of hermite type for numerical integration of stiff systems. *BIT Numerical Mathematics*, 14(1):63–77, 1974.

[50] D. Olmos and B. D. Shizgal. Pseudospectral method of solution of the fitzhugh–nagumo equation. *Mathematics and Computers in Simulation*, 79(7):2258–2278, 2009.

[51] K. Promislow and Q. Wu. Geometric evolution of quasi-bilayers in multicomponent functionalized cahn-hilliard equation. *arXiv preprint arXiv:1510.08467*, 2015.

[52] W. E. Schiesser. *The numerical method of lines: integration of partial differential equations*. Elsevier, 2012.

[53] D. C. Seal, Y. Güçlü, and A. J. Christlieb. High-order multiderivative time integrators for hyperbolic conservation laws. *Journal of Scientific Computing*, 60(1):101–140, 2014.

[54] J. Shen and X. Yang. Numerical approximations of allen-cahn and cahn-hilliard equations. *Discrete Contin. Dyn. Syst*, 28(4):1669–1691, 2010.

[55] J. M. Starobin and C. Starmer. Common mechanism links spiral wave meandering and wave-front–obstacle separation. *Physical Review E*, 55(1):1193, 1997.

[56] G. Tierra and F. Guillén-González. Numerical methods for solving the cahn–hilliard equation and its applicability to related energy-based models. *Archives of Computational Methods in Engineering*, 2(269–289):2015, 22.

[57] M. R. Willoughby. High-order time-adaptive numerical methods for the allen-cahn and cahn-hilliard equations. Master's thesis, University of British Columbia, 2011.

[58] E. M. Wolf. *A particle-in-cell method for the simulation of plasmas based on an unconditionally stable wave equation solver*. PhD thesis, MICHIGAN STATE UNIVERSITY, 2015.

[59] F. Xie, Z. Qu, J. N. Weiss, and A. Garfinkel. Coexistence of multiple spiral waves with independent frequencies in a heterogeneous excitable medium. *Physical Review E*, 63(3):031905, 2001.