1-1-2013

The Compact Implicit Integration Factor Scheme For the Solution of Allen-Cahn Equations

Meshack K. Kiplagat
University of South Carolina - Columbia

The Compact Implicit Integration Factor Scheme for the Solution of Allen-Cahn Equations

by

Meshack K. Kiplagat

Bachelor of Science
University of South Dakota, 2010

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Mathematics

College of Arts and Sciences

University of South Carolina

2013

Accepted by:

Lili Ju, Director of Thesis

Xinfeng Liu, Reader

Lacy Ford, Vice Provost and Dean of Graduate Studies

Dedication

To my family, both old and new: thank you for all your support the past two years.

Acknowledgments

A special thanks to Prof. Lili Ju for not only providing the valuable guidance that led me to complete my thesis, but for also being so generous with his time; so much so that even when the fall break was in session he nevertheless found the time to answer the many questions I had. Thank you for being patient beyond measure. I would also like to express my gratitude towards Prof. Xinfeng Liu for taking the time to read and review this thesis.

Abstract

In this thesis we apply the compact implicit integration factor (cIIF) scheme towards solving the Allen-Cahn equations with zero-flux or periodic boundary conditions. The Allen-Cahn equation is a second-order nonlinear PDE which has been the focus of many applications spanning a wide range of fields, such as in material science where it was first introduced to model the phase separation of two metallic alloys, and in biology to study population dynamics, just to name a few. The compact implicit integration method works by first transforming the PDE into a system of ODEs by discretizing the spatial derivatives using the central differencing scheme. This yields a semi-discretized form which produces a nice compact representation to the original PDE. The resulting system is then integrated with respect to time, thereby treating the linear component of the PDE exactly. The transformed nonlinear portion which represents the integrand is then approximated by a Lagrange interpolation polynomial of order r and then integrated exactly, with $r \in \{0, 1, 2\}$ in our study. Altogether, the cIIF scheme is a fully discrete scheme which is second-order accurate in space and $(r + 1)$-order accurate in time. Experiments are also performed to numerically demonstrate the stability and convergence properties of the proposed scheme.

Table of Contents

List of Figures

Chapter 1

Introduction

Let $\Omega \in \mathbb{R}^d$ be an open, rectangular and bounded region with $d \in \mathbb{N}$ and $T > 0$. In this paper, we shall discuss a scheme for the solution of the Allen-Cahn equations where

$$\begin{cases} \frac{\partial u}{\partial t} = D\Delta u - f(u), & \mathbf{x} \in \Omega \subset \mathbb{R}^2, t \in [0, T], \\ u|_0 = u_{t=0} & \mathbf{x} \in \Omega, \end{cases} \tag{1.1}$$

Ω is an rectangular set, and either Neumann or periodic boundary conditions are imposed.

The Allen-Cahn equation was first introduced by John W. Cahn and his graduate student Sam Allen to describe the motion of anti-phase boundaries in metallic alloys[1]. Specifically, it was proposed as a simple model for phase separation of metallic components within a binary alloy at a fixed temperature [9]. The function u, known as the phase field function, assumes values between -1 and 1 where the endpoints represent volumes with pure states. Similarly, values in the region $-1 < u < 1$ are yielded where mixtures arise.

This phenomena of differences in concentration can also be explained in terms of the crystal lattice arrangements (see [15]). The anti-phase boundary of thickness $\epsilon > 0$ separates two regions, where the arrangement of crystal lattices in one domain resembles another except for a relative displacement. Upon heating and subsequent cooling of the solid has the potential to not only move the boundary separating the two domains, but also cause further disordering among the crystal lattices in the solid. These regions with the same ordering may also coalesce into larger regions, a

phenomenon known as domain coarsening. In this context, the function u models the amount of order in a localized region. For a more detailed review of the Allen-Cahn equation, its derivation and related equations refer to [1, 13, 4].

The parameter $\epsilon > 0$ is the interface width, and in its initial application denoted the interaction thickness of the interface separating the two anti-phase domains. The reaction term is such that $f(u) = F'(u)$, where F is the potential energy function, and in most contexts is taken to be double-welled [9, 5]. It is well known that the Allen-Cahn equation is a gradient flow in L_2 of the Ginzburg-Landau energy functional equal to

$$E(u) = \int_\Omega \left\{ \frac{1}{2}|\nabla u|^2 + \frac{1}{4\epsilon^2}\left(u^2 - 1\right)^2 \right\} d\mathbf{x}, \tag{1.2}$$

where $E(u)$ is a decreasing function of time. Beyond its application to various material science problems, the Allen-Cahn equation has proved essential in the fields of image processing [11, 3]; in fluid dynamics where it is used to solve the mean flow curvature problem [9]; in biology where population dynamics are studied; and in geology where the phase-field model is applied to the study of geological grain microstructures [17], just to name a few.

This thesis will primarily deal with the development of the compact Implicit Integration Factor (**cIIF**) discretization scheme for the solution of the Allen-Cahn equation. The technique will first require a rectangular mesh of the spatial domain based on which the PDE is transformed into a system of ODEs after a spatial discretization. The two broadly used methods to achieve this are finite difference formulas and spectral methods [2]. Due to the nature of the space, and ease of implementation, we shall use the central difference method to approximate the Laplacian at the predetermined points of our rectangular grid.

The remaining nonlinear forcing term is simply evaluated at the points of our mesh. This yields a semi-discrete approximation consisting of two parts which together forms a coupled system of ODEs with respect to the independent time variable

t. It is this nice, compact representation mostly attributable to the spatial discretization from which the scheme derives its name. It is also a technique borrowed from a representation developed originally for solving a two-dimensional Poission's equation [16]. The resulting compact, semi-discrete system is thereafter multiplied by the appropriate exponential matrices via the integrating factors method.

As this paper will show, the proposed scheme works by integrating the system which treats the approximation to the linear portion of the PDE exactly. The remaining nonlinear component in the integrand is approximated by the implicit Lagrange interpolation polynomial method. This integration allows us to approximate values for the function u at the points of the spatial mesh and ahead one time step.

Overall the scheme is attractive first for the efficiency with which the approximated solutions are computed, with a cost of only $O(N^2 \log_2 N)$ per time step in our two dimensional case through the use of FFTs. The cIIF scheme excels in its utility of storage by requiring only $O(N^2)$ storage, as opposed to $O(N^4)$ storage via a "non-compact" approach. The implicit treatment of the nonlinear component also grants us a level of flexibility with regards to the time step sizes we can use. Lastly, the nonlinear solution process can be done pointwisely, thus becoming very efficient when used in conjunction with Newton-based algorithms.

The paper is organized as follows: In Chapter 2 we will propose the cIIF scheme in the cases where periodic or zero-flux boundary conditions are imposed. In chapter 3 we will discuss the stability properties of the scheme. Chapter 4 will contain the numerical testing of the scheme where the accuracy and convergence of the scheme will be demonstrated, and will be followed by our conclusions in Chapter 5.

Chapter 2

Compact Implicit Integration Factor Methods

Let Ω be an open rectangular domain in $\mathbb{R}^2$ with $T > 0$. Consider the Allen-Cahn equation where $D > 0$ is the diffusion constant, and the nonlinear forcing term is defined as

$$f(u) = \frac{1}{\epsilon^2}\left(u^3 - u\right) \tag{2.1}$$

The problems we shall concern ourselves with will have either periodic or zero-flux boundary conditions.

Suppose that $\Omega = \{(x, y)|x_b < x < x_e, y_b < y < y_e\}$. By replacing the Laplace operator, equation (1.1) can then be written concisely as

$$\frac{\partial u}{\partial t} = D\left(u_{xx} + u_{yy}\right) - f(u), \qquad (x, y) \in \Omega, t \in (0, T) \tag{2.2}$$

Discretizing the PDE will require the evaluation of u at discrete points of a uniform mesh $\Omega' = \{(x_i, y_j) \mid x_i = x_0 + i \cdot h_x,\ y_j = y_0 + j \cdot h_y\}$, where

$$h_x = \frac{x_e - x_b}{N_x}, \qquad h_y = \frac{y_e - y_b}{N_y} \tag{2.3}$$

for N_x, $N_y \in \mathbb{N}$.

2.1 The model problem with periodic boundary conditions

We derive the cIIF method for equation (1.1) We also define the periodic boundary condition as one having the following set of conditions satisfied,

$$\begin{cases} u(x_b, y) = u(x_e, y), u_x(x_b, y) = u_x(x_e, y), & y \in [y_b, y_e], t \in [t_0, t_0 + T] \\ u(x, y_b) = u(x, y_e), u_y(x, y_b) = u_y(x, y_e), & x \in [x_b, x_e], t \in [t_0, t_0 + T] \end{cases} \tag{2.4}$$

With Ω' defined as our mesh we shall use the second-order accurate central differencing scheme to approximate the spatial derivatives. Set $u_{i,j} = u_{i,j}(t) \approx u(t, x_i, y_j)$ for $0 \leq i \leq N_x - 1$ and $0 \leq j \leq N_y - 1$. Denote the set of unknowns as

$$\mathbf{U} = (u_{i,j})_{N_x \times N_y} = \begin{pmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,N_y-1} \\ u_{1,0} & u_{1,1} & \dots & u_{1,N_y-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_x-1,0} & u_{N_x-1,1} & \dots & u_{N_x-1,N_y-1} \end{pmatrix} \tag{2.5}$$

and define

$$\mathbf{G_{P\times P}} = \begin{pmatrix} -2 & 1 & 0 & \dots & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & & 0 \\ 0 & 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}$$

It is obvious in this case that $\mathbf{G}^T = \mathbf{G}$. Furthermore, let us define the matrices $\mathbf{A}$ and $\mathbf{B}$ equal to

$$\mathbf{A} = \frac{D}{h_x^2}\mathbf{G}_{N_x \times N_x}, \qquad \mathbf{B} = \frac{D}{h_y^2}\mathbf{G}_{N_y \times N_y}$$

We then obtain the following semi-discretized and compact representation of the model equation

$$\frac{d\mathbf{U}}{dt} = \mathbf{AU} + \mathbf{UB} - \mathcal{F}(\mathbf{U}) \tag{2.6}$$

where $\mathcal{F}(U) = (f(u_{i,j}))\,,\ 0 \leq i \leq N_x - 1,\ 0 \leq j \leq N_y - 1.$

Since the matrices A and B are diagonalizable we can rewrite the matrices $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = \mathbf{P}_x\mathbf{D}_x\mathbf{P}_x^{-1}, \qquad \mathbf{B} = \mathbf{P}_y\mathbf{D}_y\mathbf{P}_y^{-1} \tag{2.7}$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are the diagonal matrices holding as its diagonal elements the eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ respectively. It bears noting that in the general case, $\mathbf{P}_x^{-1} = \mathbf{P}_x^T$ and $\mathbf{P}_y^{-1} = \mathbf{P}_y^T$ provided $\mathbf{P}_x$, $\mathbf{P}_y$ are orthonormal.

Let us define $\mathbf{V} = \mathbf{P}_x^{-1}\mathbf{U}\mathbf{P}_y$. Then substituting $\mathbf{V}$ for $\mathbf{U}$ in (2.6) we get

$$\frac{d\mathbf{V}}{dt} = \mathbf{D}_x\mathbf{V} + \mathbf{V}\mathbf{D}_y - \mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U})\mathbf{P}_y. \tag{2.8}$$

Suppose that the diagonal matrices are defined equivalently as $\mathbf{D}_x = diag[d_1^x, \ldots, d_{N_x}^x]$, $\mathbf{D}_y = diag[d_1^y, \ldots, d_{N_y}^y]$. At this point we perform a fundamental step derived from Integrating Factor and Exponential Time Differencing methods, which is to multiply the left and right of both sides of equality (2.8) by $e^{-\mathbf{D}_x t}$ and $e^{-\mathbf{D}_y t}$ respectively. This gives us

$$e^{-\mathbf{D}_x t}\left(\frac{d\mathbf{V}}{dt}\right)e^{-\mathbf{D}_y t} = e^{-\mathbf{D}_x t}\left(\mathbf{D}_x\mathbf{V} + \mathbf{V}\mathbf{D}_y - \mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U})\mathbf{P}_y\right)e^{-\mathbf{D}_y t}, \tag{2.9}$$

or equivalently

$$e^{-\mathbf{D}_x t}\left(\frac{d\mathbf{V}}{dt} - \mathbf{D}_x\mathbf{V} - \mathbf{V}\mathbf{D}_y\right)e^{-\mathbf{D}_y t} = -e^{-\mathbf{D}_x t}\left(\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U})\mathbf{P}_y\right)e^{-\mathbf{D}_y t}. \tag{2.10}$$

The left of (2.10) can be rearranged so that it is the derivative with respect to t of a single matrix $\mathbf{X}$. To do so, we introduce some new notations. First, we set $\mathbf{H} = (h_{i,j})_{(N_x-1)\times(N_y-1)}$, where $h_{i,j} = d_i^x + d_j^y$ and $0 \le i \le N_x - 1$, $0 \le j \le N_y - 1$. Also, we define the operation e^* such that

$$(e^*)^{\mathbf{H}} = \left(e^{h_{i,j}}\right)_{(N_x-1)\times(N_y-1)}.$$

We also define a second operator $\odot$ as the element by element multiplication of two matrices of similar structure, or

$$(\mathbf{M} \odot \mathbf{L}) = (m_{i,j}l_{i,j}),$$

where $\mathbf{M} = (m_{i,j})_{\mathbf{P}\times\mathbf{P}}$ and $\mathbf{L} = (l_{i,j})_{\mathbf{P}\times\mathbf{P}}$.

With these operations defined, direct calculations on (2.10) leads to a nice representation where the left portion holds the linear portion of the original PDE, and the right holds the nonlinear portion, or

$$\frac{d(\mathbf{V} \odot (e^*)^{-\mathbf{H}t})}{dt} = -e^{-\mathbf{D}_x t} \left(\mathbf{P}_x^{-1} \mathcal{F}(\mathbf{U}) \mathbf{P}_y \right) e^{-\mathbf{D}_y t}. \tag{2.11}$$

We partition the time interval as follows. Setting $t_0 = 0$, let $t_n = t_0 + n\Delta t$, where $\Delta t = T/N_t$, $0 \leq n \leq N_t$, and N_t is the number of equal partitions of the time interval. At this point, we integrate (2.11) from t_n to t_{n+1}, in effect treating the linear component of the equation exactly. This leaves us with

$$\mathbf{V}_{n+1} \odot (e^*)^{-\mathbf{H}\Delta t} - \mathbf{V}_n = -\int_0^{\Delta t} \left(\mathbf{P}_x^{-1} \mathcal{F}\left(\mathbf{U}(t_n + \tau)\right) \mathbf{P}_y \right) \odot (e^*)^{-\mathbf{H}\tau} \, d\tau, \tag{2.12}$$

or equivalently,

$$\mathbf{V}_{n+1} = \mathbf{V}_n - \left(\int_0^{\Delta t} \left(\mathbf{P}_x^{-1} \mathcal{F}\left(\mathbf{U}(t_n + \tau)\right) \mathbf{P}_y \right) \odot (e^*)^{-\mathbf{H}\tau} \, d\tau \right) \odot (e^*)^{\mathbf{H}\Delta t} \tag{2.13}$$

From this a general compact approximation scheme for solving $\mathbf{U}_{n+1}$,

$$\mathbf{U}_{n+1} = \mathbf{P}_x \left((\mathbf{P}_x^{-1} \mathbf{U}_n \mathbf{P}_y) \odot (e^*)^{\mathbf{H}\Delta t} - \int_0^{\Delta t} (\mathbf{P}_x^{-1} \mathcal{F}(\mathbf{U}(t_n + \tau))) \right. \tag{2.14}$$

$$\left. \odot (e^*)^{\mathbf{H}(\Delta t - \tau)} d\tau \right) \mathbf{P}_y^{-1}. \tag{2.15}$$

Having dealt with the linear component, we are left with the integral

$$\int_0^{\Delta t} (\mathbf{P}_x \mathcal{F}(\mathbf{U}(t_n + \tau)) \mathbf{P}_y) \odot (e^*)^{\mathbf{H}(\Delta t - \tau)} d\tau,$$

to which we implement an implicit scheme which not only approximates the integrand, but also allows for an appropriate level of stability while doing so. One such approach is the Lagrange interpolation polynomial of degree r at the nodes $t_{n+1}, t_n, \ldots, t_{n+1-r}$ applied to the integrand

$$\mathbf{F}(t_n + \tau) = (\mathbf{P}_x^{-1} \mathcal{F}(\mathbf{U}(t_n + \tau)) \mathbf{P}_y) \odot (e^*)^{\mathbf{H}(\Delta t - \tau)},$$

as proposed in [14]:

$$P_r^F(t_n+\tau) = \sum_{s=-1}^{r-1} \omega_{r,s}(\tau)\mathbf{F}(t_{n-s}) \tag{2.16}$$

where

$$\omega_{r,s}(\tau) = \prod_{\substack{l=-1 \\ l\neq s}}^{r-1} \frac{\tau + l\Delta t}{(l-s)\Delta t}.$$

It is easy then to see that, after approximating,

$$\begin{aligned}
\int_0^{\Delta t} (\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U}(t_n+\tau))\mathbf{P}_y) \odot (e^*)^{\mathbf{H}(\Delta t-\tau)} d\tau \\
&\approx \Delta t \sum_{s=-1}^{r-1} \beta^{(r,s)} (\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U}_{n-s})\mathbf{P}_y) \odot (e^*)^{(s+1)\mathbf{H}\Delta t} \\
&= \Delta t \sum_{s=0}^{r-1} \beta^{(r,s)} (\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U}_{n-s})\mathbf{P}_y) \odot (e^*)^{(s+1)\mathbf{H}\Delta t} \\
&\quad + \Delta t\, \beta^{(r,-1)} (\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U}_{n+1})\mathbf{P}_y)
\end{aligned} \tag{2.17}$$

with $\beta^{r,s} = \int_0^{\Delta t} \omega_{r,s}(\tau)dt$, which is (r+1)-th order accurate. For simplicity, we again only present the values we only present the values of $\{\beta_{i,j}^{(r,s)}\}$ for $r = 0, 1, 2$ below:

$$\begin{aligned}
r = 0: &\quad \beta^{(0,-1)} = 1; \\
r = 1: &\quad \beta^{(1,-1)} = \tfrac{1}{2}, \quad \beta^{(1,0)} = \tfrac{1}{2}; \\
r = 2: &\quad \beta^{(2,-1)} = \tfrac{5}{12}, \quad \beta^{(2,0)} = \tfrac{2}{3}, \quad \beta^{(2,1)} = -\tfrac{1}{12},
\end{aligned}$$

where it's important to note that $\{\beta_{i,j}^{(r,s)}\}$ is constant across all points. Now, putting (2.17) into (2.14) yields a compact implicit integration factor scheme that is second order accurate in space and $(r+1)$-th order accurate in time:

$$\begin{aligned}
\mathbf{U}_{n+1} = \mathbf{P}_x \Bigg((\mathbf{P}_x^{-1}\mathbf{U}_n\mathbf{P}_y) \odot (e^*)^{\mathbf{H}\Delta t} - \Delta t \sum_{s=0}^{r-1} \beta^{(r,s)} (\mathbf{P}_x^{-1}\mathcal{F}(\mathbf{U}_{n-s})\mathbf{P}_y) \\
\odot (e^*)^{(s+1)\mathbf{H}\Delta t} \Bigg) \mathbf{P}_y^{-1} - \Delta t \beta^{(r,-1)} \mathcal{F}(\mathbf{U}_{n+1}).
\end{aligned} \tag{2.18}$$

Remark 1. In the nonlinear case, where $f(u) \neq 0$, the scheme (2.18) is evaluated pointwisely. As a result it it very efficient especially when used in conjunction with some Newton-type iterative method for scalar methods.

Remark 2. In the cases where $r \in \{0, 1, 2\}$, we get a first-order cIIF accurate in time scheme for $r = 0$ as

$$\mathbf{U}_{n+1} = \mathbf{P}_x \left(\left(\mathbf{P}_x^{-1} \mathbf{U}_n \mathbf{P}_y \right) \odot (e^*)^{\mathbf{H}\Delta t} \right) \mathbf{P}_y^{-1} - \Delta t \mathcal{F}(\mathbf{U}_{n+1}), \tag{2.19}$$

a second-order accurate in time cIIF scheme for $r = 1$ as

$$\mathbf{U}_{n+1} = \mathbf{P}_x \left((\mathbf{P}_x^{-1}(\mathbf{U}_n - \frac{\Delta}{2}\mathcal{F}(\mathbf{U}_n))\mathbf{P}_y) \odot (e^*)^{\mathbf{H}\Delta t} \right) \mathbf{P}_y^{-1} - \frac{\Delta t}{2}\mathcal{F}(\mathbf{U}_{n+1}), \tag{2.20}$$

or a third-order accurate in time cIIF scheme corresponding to $r = 2$ as

$$\begin{aligned} \mathbf{U}_{n+1} = \mathbf{P}_x \Bigg(& (\mathbf{P}_x^{-1}(\mathbf{U}_n - \frac{2\Delta t}{3}\mathcal{F}(\mathbf{U}_n))\mathbf{P}_y) \odot (e^*)^{\mathbf{H}\Delta t} \\ & + \frac{\Delta t}{12}(\mathbf{P}_x^{-1}(\mathcal{F}(\mathbf{U}_{n-1}))\mathbf{P}_y) \odot (e^*)^{2\mathbf{H}\Delta t} \Bigg) - \frac{5\Delta t}{12}\mathcal{F}(\mathbf{U}_{n+1}) \end{aligned}$$

2.2 The model problem with no-flux boundary conditions

In this case, we define the zero-flux boundary condition as

$$\frac{\partial u}{\partial \mathbf{n}} = 0, \qquad x \in \partial\Omega,\ t \in (0, T) \tag{2.21}$$

Working with our rectangular mesh, we may then set

$$\mathbf{U} = (u_{i,j})_{(N_x+1)\times(N_y+1)} = \begin{pmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,N_y} \\ u_{1,0} & u_{1,1} & \dots & u_{1,N_y} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_x,0} & u_{N_x,1} & \dots & u_{N_x,N_y} \end{pmatrix}_{(N_x+1)\times(N_y+1)} \tag{2.22}$$

and the corresponding matrix $\mathbf{G}$ defined as

$$\mathbf{G}_{\mathbf{P}\times\mathbf{P}} = \begin{pmatrix} -2 & 2 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & & 0 \\ 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & \dots & 0 & 2 & -2 \end{pmatrix}_{P\times P} \tag{2.23}$$

Note that in the zero flux case $\mathbf{G}^T \neq \mathbf{G}$, unlike the periodic case. We thereafter set

$$\mathbf{A} = \frac{D}{h_x^2}\mathbf{G}_{(N_x+1)\times(N_y+1)}, \qquad \mathbf{B} = \frac{D}{h_y^2}\mathbf{G}_{(N_y+1)\times(N_y+1)}. \tag{2.24}$$

With the matrices defined as shown, the cIIF scheme for the problem with no-flux boundary conditions follows the exact same form as the scheme (2.18).

Remark 3. As discussed in [10] the computation complexities of $\mathbf{P}_x\mathbf{V}$, $\mathbf{P}_x^{-1}\mathbf{V}$, $\mathbf{P}_y\mathbf{V}$, and $\mathbf{P}_y^{-1}\mathbf{V}$ for any $(N_x + 1) \times (N_y + 1)$ matrix in the zero-flux case, or $N_x \times N_y$ matrix in the periodic case, can be reduced from $O(N^3)$ to $O(N^2 \log_2 N)$ for $N = \max\{N_{x'}, N_{y'}\}$ if through the use of Fast Fourier Transforms we choose the eigenvalues d_i^x, d_j^y appropriately.

Chapter 3

Stability Analysis

3.1 The nonlinear system solver and convergence

We go back to the definition of the Allen-Cahn equation. Let us briefly write (2.18) as

$$\mathbf{U}_{n+1} = \mathbf{CON} - \Delta t \beta^{(r,-1)} \mathcal{F}(\mathbf{U}_{n+1}) \tag{3.1}$$

where

$$\mathbf{CON} = \mathbf{P}_x \left((\mathbf{P}_x^{-1} \mathbf{U}_n \mathbf{P}_y) \odot (e^*)^{\mathbf{H}\Delta t} - \Delta t \sum_{s=0}^{r-1} \beta^{(r,s)} \ (\mathbf{P}_x^{-1} \mathcal{F}(\mathbf{U}_{n-s}) \mathbf{P}_y) \odot (e^*)^{(s+1)\mathbf{H}\Delta t} \right) \mathbf{P}_y^{-1}$$

can be calculated using results from previous time steps. In approximating the original equation (1.1), our goal will be to apply the scheme (2.18) towards solving

$$\mathbf{U}_{n+1} = \mathbf{CON} - \frac{\Delta t \beta^{r,-1}}{\epsilon^2} (\mathbf{U}_{n+1} \odot \mathbf{U}_{n+1} \odot \mathbf{U}_{n+1} - \mathbf{U}_{n+1}) \tag{3.2}$$

which is equivalent to solving $\mathcal{G}((\mathbf{U}_{n+1})_{i,j}) = 0$ where

$$\mathcal{G}((\mathbf{U}_{n+1})_{i,j}) = (\mathbf{U}_{n+1})_{i,j}^3 + (\rho - 1)(\mathbf{U}_{n+1})_{i,j} - \rho(\mathbf{CON})_{i,j} \tag{3.3}$$

and

$$\rho = \frac{\epsilon}{\Delta t \beta^{(r,-1)}}. \tag{3.4}$$

Thus, in implementing Newton's algorithm for $\mathcal{G}((\mathbf{U}_{n+1})_{i,j}) = 0$ we get that

$$(\mathbf{U}_{n+1})_{i,j}^{(k+1)} = (\mathbf{U}_{n+1})_{i,j}^{(k)} - \frac{\mathcal{G}((\mathbf{U}_{n+1})_{i,j}^{(k)})}{\mathcal{G}'((\mathbf{U}_{n+1})_{i,j}^{(k)})}. \tag{3.5}$$

It is easy to show that for $\rho \geq 1$, the nonlinear system (3.2) is a contraction, thus implying the Newton's method is stable. Consequently, when $r = 0$ we require $\Delta t \leq \epsilon^2$; $r = 1$ requires $\Delta t \leq 2\epsilon^2$; and $r = 2$ requires $\Delta t \leq \frac{12}{5}\epsilon^2$.

Chapter 4

Numerical Testing

In this chapter, we shall study the stability and convergence properties of the cIIF scheme, based on its application to approximating a mean flow curvature problem and a traveling wave solution. The computations were performed on a PC with Intel Core i7 3.2GHz processor and 6GB RAM. MATLAB was also used to write the programs on which the testing was done.

The experiments in this section will be based on a spatial domain $\Omega \in \mathbb{R}^2$ and a time interval $[0, T]$. The number of partitions along each spatial axis, as well as the the time discretization size N_t will be varied in order to study the convergence and accuracy of the scheme. We note that the cIIF scheme is presumed to be stable as the values chosen for the interaction term ϵ are small enough that they do not cause major deviations from the true solution as time progresses.

4.1 Traveling Wave Solution

It is known that the Allen-Cahn equation in the whole space has a traveling wave solution. In this application, we consider the space $\Omega = [-0.5, 1.5]^2 \subset \mathbb{R}^2$ where we have

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u - \frac{1}{\epsilon^2}(u^3 - u), & (x, y) \in \Omega,\ t \in [0, T] \\ u(0, x, y) = \frac{1}{2}\left(1 - \tanh\left(\frac{x}{2\sqrt{2}\epsilon^2}\right)\right), & (x, y) \in \Omega. \end{cases} \tag{4.1}$$

Additionally, the zero-Neumann boundary condition $\frac{\partial u}{\partial t}|_{\partial\Omega} = 0$ is imposed to allow for an approximate exact solution (for $\epsilon << 1$) of the form

$$u(t,x,y) = \frac{1}{2}\left(1 - \tanh\left(\frac{x - st}{2\sqrt{2}\epsilon}\right)\right) \tag{4.2}$$

with the constant s set to $\frac{3}{\sqrt{2}\epsilon}$.

The numerical results were run based on the benchmark tests of [12, 10]. We set out to numerically confirm both the accuracy of the space discretization as finer meshes are used, as well as the $(r+1)$-th order accuracy of the time discretization as finer partitions along $[0,T]$ were implemented.

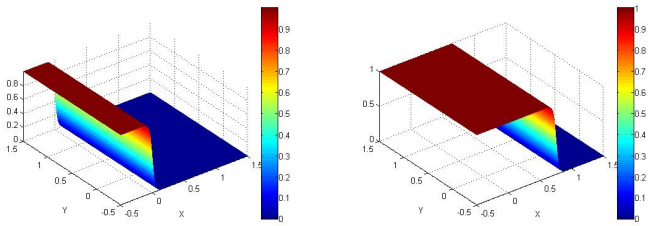
The test of the accuracy of space discretization with $r = 2$ yielded the first set of results in Table 4.1. The convergence rates as the meshes of the spatial domain were made finer were observed, and the results prove that the cIIF scheme is second-order accurate in space with respect to both the L^2 and L^∞ error norms.

While performing the accuracy of space discretization test it was noted that the further refinement of the spatial mesh into 512×512 and 1024×1024 grid while holding N_t constant yielded results which might have been interpreted as inconsistent with the convergence rates already calculated. However, this is due to the the errors attributable to the spatial domain initially decreasing in orders of magnitude as finer meshes are used. Past a point, the error associated with the space domain did not decrease as quickly with finer meshes, while the contribution to the error by the time approximations remained comparable.

Numerical tests were also performed to study the accuracy of time discretization. In the three cases corresponding to $r = 0, 1, 2$ the results confirmed that the cIIF scheme is indeed first, second, and third order accurate in time respectively. Testing was also done to calculate the energy based on the cIIF result. Consistent with the case of the exact solution, the energy decreased as time increased as is illustrated in Figure 4.2. The plots of the exact solution and the cIIF scheme at time $T = 0.75s$ are also given by Figures 4.1a and 4.1b.

Table 4.1: Errors and convergence rates at the final time $T = \frac{3}{\sqrt{2}\epsilon}$ of the 2D traveling wave problem using the cIIF scheme.

$(N_x \times N_y) \times N_t$	L_2 Error	CR	L_∞ Error	CR
Accuracy test of space discretization, $r = 2$				
$(32^2) \times 1024$	3.6271e-01	-	8.2217e-01	-
$(64^2) \times 1024$	1.2385e-01	1.55	3.3532e-01	1.29
$(128^2) \times 1024$	3.2897e-02	1.91	9.5657e-02	1.81
$(256^2) \times 1024$	8.1367e-03	2.02	2.3901e-02	2.00
$(512^2) \times 1024$	2.1819e-03	1.90	5.1100e-03	2.23
Accuracy test of time discretization, $r = 0$				
$(1024^2) \times 16$	1.2073	-	9.9999e-01	-
$(1024^2) \times 32$	1.2070	0.00	9.9970e-01	0.00
$(1024^2) \times 64$	4.4498e-01	1.44	8.64791e-01	0.21
$(1024^2) \times 128$	1.9683e-01	1.18	5.4035e-01	0.68
Accuracy test of time discretization, $r = 1$				
$(1024^2) \times 16$	5.3609e-01	-	8.9058e-01	-
$(1024^2) \times 32$	9.2209e-02	2.54	2.6730e-01	1.74
$(1024^2) \times 64$	2.2568e-02	2.03	6.7590e-02	1.98
$(1024^2) \times 128$	6.1184e-03	1.88	1.8196e-02	1.89
Accuracy test of time discretization, $r = 2$				
$(1024^2) \times 16$	1.9736e-01	-	5.1906e-01	-
$(1024^2) \times 32$	2.7285e-02	2.85	8.0851e-02	2.68
$(1024^2) \times 64$	4.3940e-03	2.63	1.2883e-02	2.65



(a) (b)

Figure 4.1: Plots of the traveling wave solution at time (a) t=0 and (b) $t = \frac{3}{4s}$ with $s = \frac{3}{\sqrt{2}\epsilon}$, using the cIIF results with grid $(512)^2 \times 1024$, third-order accurate in time, with interaction term $\epsilon = 0.015$

.

4.2 Mean Curvature Flow

In this example, we shall consider the mean curvature flow problem based on [13, 8].

Let $\Omega = [-0.5, 0.5]^2$. The mean flow problem we shall simulate is described in the

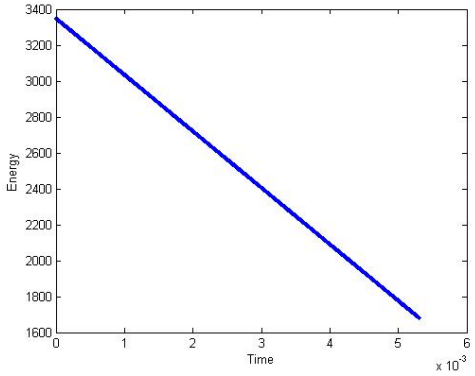


Figure 4.2: Evolution of the energy functional $E(u)$ by the cIIF results with mesh $(512)^2 \times 1024$ with $\epsilon = 0.015$ and third-order accurate in time as time increases.

following:

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u - \frac{1}{\epsilon^2}(u^3 - u), & \mathbf{x} \in \Omega, t \in [0, T] \\ u(0, x, y) = \tanh\left(\frac{R_0 - \sqrt{x^2+y^2}}{\sqrt{2}\epsilon}\right), & \mathbf{x} \in \Omega. \end{cases} \tag{4.3}$$

with $R_0 = 0$. For a more detailed exposition on this test example, see [18, 6, 7]. Additionally, a periodic boundary condition is imposed. With the conditions defined thus far, the PDE models the shrinking process of a circle in the plane. The inital radius of the circle is denoted R_0. Let the radius of the circular region at time t be denoted by $R(t)$ and the area by $A(t)$. As $\epsilon \to 0$, we know that the theoretic limit radius satisfies [18, 12]

$$\frac{dR_{lim}}{dt} = -\frac{1}{R_{lim}}, \tag{4.4}$$

hence we have

$$R_{lim}(t) = \sqrt{R_0^2 - 2t} \tag{4.5}$$

Correspondingly, it also holds that

$$A_{lim}(t) = \pi(R_0^2 - 2t), \tag{4.6}$$

We set the final time to $T = 0.075$. Thus, consistent with 4.5 it follows that $R_{lim}(T) = 0.1$. Numerical testing was performed to approximate the radius for the different cases using the cIIF scheme and the results were recorded in Table 4.2. Since the exact

radius solutions for the provided final time T was not readily available we opted to use the radii results based on the mesh $(1024)^2 \times 1024$. Denoting as R_ϵ the values for $\epsilon = 0.04,\ 0.02$, we used these values as approximates to the exact solutions at the final time T to calculate the convergence rates. The results are reported in Table 4.3.

The numerical results show that the cIIF scheme is second-order accurate in space in both the cases of $\epsilon = 0.04$ and $\epsilon = 0.02$. Further tests were completed to test the accuracy of time discretization with respect to the spatial mesh 1024×1024. The errors are shown to be monotonically decreasing for all orders of accuracy $(r+1)$, $r = 0, 1, 2$. The convergence rates yielded by our results especially for the first- and second-order accuracies agree with the theoretical results of Chapter 2. It is noted, however, that the convergence rates associated with the third-order accuracy tests are highly oscillatory and do not approach the theoretical value of 3 consistently. This is due to the sizes of the grids used, as testing for higher orders of accuracies require finer meshes both in the spatial and temporal domain in order to achieve optimum convergence. This is also especially true when small values of ϵ are used for the experiments. Finally, we see that $|R_\epsilon - R_{lim}|$ is equal to $9.2900e-04$ and $2.790e-04$ for $\epsilon = 0.04$ and 0.02 respectively. This yielded a convergence rate of 1.7354 with respect to the interaction term ϵ.

The shrinking of the circle as time progresses is captured in the Figure 4.4. It is seen that the radius does shrink as we march forward in time. Furthermore, the evolution of the energy and area as computed by the cIIF scheme is seen to decrease linearly, as Figure 4.5 shows.

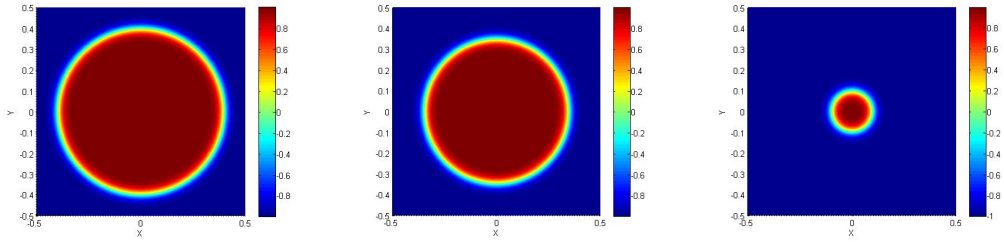


Figure 4.3: Visualization of the shrinking circle at times $t = 0, 0.02$ and 0.075 respectively (from left to right) using the cIIF scheme with grid $(1024)^2 \times 512$, interface thickness $\epsilon = 0.02$, and third-order accurate in time.

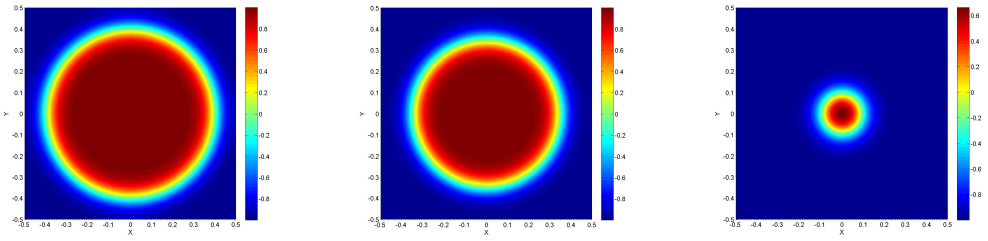


Figure 4.4: Visualization of the shrinking circle at times $t = 0, 0.02$ and 0.075 respectively (from left to right) using the cIIF scheme with grid $(1024)^2 \times 512$, interface thickness $\epsilon = 0.04$, and third-order accurate in time.

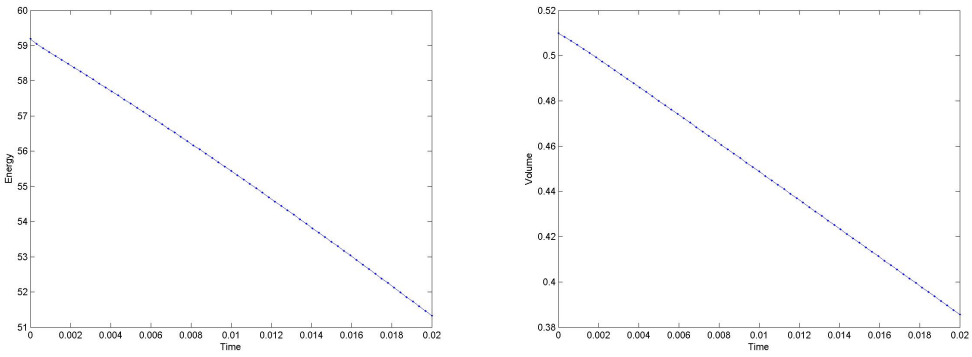


Figure 4.5: Evolution of the energy and area (left and right respectively) of the shrinking circle along time using the cIIF scheme with grid $(1024)^2 \times 512$, interaction thickness $\epsilon = 0.04$ and third-order accurate in time.

Table 4.2: Numerical results for the radius of the mean curvature flow shrinking circle experiment at time $T = 0.075$ using the cIIF Scheme.

$(N_x \times N_y) \times N_t$	$\epsilon = 0.04$		$\epsilon = 0.02$	
	R	$R - R_{lim}$	R	$R - R_{lim}$
Accuracy Test of Space Discretization $r = 2$				
$(64)^2 \times 1024$	0.1054	5.4070e-03	0.1219	-2.1860e-02
$(128)^2 \times 1024$	0.1007	6.6905e-04	0.1054	5.3700e-03
$(256)^2 \times 1024$	0.09945	-5.4620e-04	0.1011	1.0817e-04
$(512)^2 \times 1024$	0.0991	-8.5212e-04	0.1000	-6.0941e-06
$(1024)^2 \times 1024$	**0.0991**	-9.2873-04	**0.0997**	-2.7914e-04
Accuracy Test of Time Discretization $r = 0$				
$(1024)^2 \times 16$	0.5642	4.6419e-01	0.5642	4.6419e-01
$(1024)^2 \times 32$	0.5642	4.6419e-01	0.5642	4.6419e-01
$(1024)^2 \times 64$	0.0981	-1.8916e-03	0.5642	4.6419e-01
$(1024)^2 \times 128$	0.0986	-1.3563e-03	0.5642	4.6419e-01
$(1024)^2 \times 256$	0.0989	-1.1379e-03	0.0989	-1.1064e-03
$(1024)^2 \times 512$	0.0990	-1.0332e-03	0.0993	-6.6446e-04
Accuracy Test of Time Discretization $r = 1$				
$(1024)^2 \times 16$	0.5642	4.6421e-01	0.5650	4.6500e-01
$(1024)^2 \times 32$	0.0963	-3.7100e-03	0.5642	4.6417e-01
$(1024)^2 \times 64$	0.0985	-1.5246e-03	0.5642	4.6418e-01
$(1024)^2 \times 128$	0.0989	-1.0691e-03	0.0996	-4.4414e-04
$(1024)^2 \times 256$	0.0990	-9.6320e-04	0.0997	-3.1580e-04
$(1024)^2 \times 512$	0.0991	-9.3735e-04	0.0997	-2.8878-04
Accuracy Test of Time Discretization $r = 2$				
$(1024)^2 \times 16$	0.5701	4.7006e-01	0.5949	4.9402e-01
$(1024)^2 \times 32$	0.0982	-1.8280e-03	0.5856	4.8556e-01
$(1024)^2 \times 64$	0.0990	-9.6272e-04	0.5669	4.6692e-01
$(1024)^2 \times 128$	0.0991	-9.2061e-04	0.09967	-3.3355e-04
$(1024)^2 \times 256$	0.0991	-9.2632e-04	0.0997	-2.8464e-04

Table 4.3: Errors and convergence rates associated with the numerical results of Table 4.2 using the cIIF scheme.

$(N_x \times N_y) \times N_t$	$\epsilon = 0.04$		$\epsilon = 0.02$	
	$\lvert R - R_\epsilon \rvert$	CR	$\lvert R - R_\epsilon \rvert$	CR
Accuracy Test of Space Discretization $r = 2$				
$(64)^2 \times 1024$	6.3360e-03	-	2.2139e-02	-
$(128)^2 \times 1024$	1.5980e-03	1.99	5.6490e-03	1.97
$(256)^2 \times 1024$	3.8300e-04	2.06	1.3610e-03	2.05
$(512)^2 \times 1024$	7.7000e-05	2.31	2.7300e-04	2.32
Accuracy Test of Time Discretization $r = 0$				
$(1024)^2 \times 16$	4.6512e-01	-	4.6447e-01	-
$(1024)^2 \times 32$	4.6512e-01	0.00	4.6447e-01	0.00
$(1024)^2 \times 64$	9.6300e-04	8.92	4.6447e-01	0.00
$(1024)^2 \times 128$	4.2700e-04	1.17	4.6447e-01	0.00
$(1024)^2 \times 256$	2.0900e-04	1.03	8.2700e-04	9.13
$(1024)^2 \times 512$	1.0400e-04	1.01	3.8500e-04	1.10
Accuracy Test of Time Discretization $r = 1$				
$(1024)^2 \times 16$	4.6514e-01	-	4.6528e-01	-
$(1024)^2 \times 32$	2.7810e-03	7.39	4.6445e-01	0.00
$(1024)^2 \times 64$	5.9600e-04	2.22	4.6446e-01	0.00
$(1024)^2 \times 128$	1.4000e-04	2.09	1.6500e-04	11.46
$(1024)^2 \times 256$	3.4000e-05	2.04	3.7000e-05	2.16
$(1024)^2 \times 512$	8.0000e-06	2.09	9.0000e-06	2.04
Accuracy Test of Time Discretization $r = 2$				
$(1024)^2 \times 16$	4.7099e-01	-	4.9518e-01	-
$(1024)^2 \times 32$	8.9900e-04	9.03	4.8581e-01	0.03
$(1024)^2 \times 64$	3.3000e-05	4.77	4.6720e-01	0.06
$(1024)^2 \times 128$	8.0000e-06	2.04	5.5000e-05	13.05
$(1024)^2 \times 256$	3.0000e-06	1.42	6.6000e-06	3.20

Chapter 5

Conclusion

In this thesis, the compact Implicit Integration Factor was studied and proposed for the solution of two dimensional Allen-Cahn equations with periodic and no-flux boundary conditions. The efficiency with which the spatial derivatives are computed, due largely to the compact treatment of the Laplacian by the cIIF scheme yielded a scheme that is second-order accurate in space and up to third-order accurate in time. While the examples we tested on dealt with domains in the plane, further work could be done to create an implementation of the cIIF scheme for dimensions three or higher. Also, further work can be done to extended the scheme to support higher orders of accuracy both in the space and time domains. Lastly, while our work focused on examples of the Allen-Cahn equations, reseach can be done to show how the cIIF scheme could be used in other types of reaction-diffusion PDEs.

Bibliography

[1] S. Allen and J.W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27:1084–1095, 1979.

[2] H. Ashi. *Numerical Methods for Stiff Systems*. University of Nottingham, June 2008.

[3] M. Beneš, V. Chalupecký, and K. Mikula. Geometrical image segmentation by the allen-cahn equation. *Applied Numerical Mathematics*, 51(2-3):187–205, 2004.

[4] J. W. Cahn and A. Novick-Cohen. Limiting motion for an Allen-Cahn/Cahn-Hilliard system. In *Free boundary problems, theory and applications (Zakopane, 1995)*, volume 363 of *Pitman Res. Notes Math. Ser.*, pages 89–97. Longman, Harlow, 1996.

[5] L. Calatroni and P. Colli. Global solution to the allen-cahn equation with singular potentials and dynamic boundary conditions. *Nonlinear Analysis: Theory, Methods and Applications*, 79(0):12–27, 2013.

[6] L.Q. Chen and J. Shen. Applications of semi-implicit fourier-spectral method to phase field equations. *Computer Physics Communications*, 108(2-3):147 – 158, 1998.

[7] Q. Du and W. Zhu. Stability analysis and application of the exponential time differencing schemes. *Journal of Computational Mathematics*, 22(2):200–209, 2004. Special issue dedicated to the 70th birthday of Professor Zhong-Ci Shi.

[8] L.C. Evans and J. Spruck. Motion of level sets by mean curvature. *Journal of Differential Geometry*, (33):635–681, 1991.

[9] X. Feng and A. Prohl. Numerical analysis of the allen-cahn equation and approximation for mean curvature flows. *Numerische Mathematik*, 94(1):33–65, 2003.

[10] L. Ju, J. Zhang, L. Zhu, and Q. Du. Fast explicit integration factor methods for semilinear parabolic equations. Submitted.

[11] D. A. Kay and A. Tomasi. Color image segmentation by the vector-valued allen-cahn phase-field model: A multigrid solution. *IEEE Transactions on Image Processing*, 18(10):2330 – 2339, 2009.

[12] Y. Li, H. G. Lee, D. Jeong, and J. Kim. An unconditionally stable hybrid numerical method for solving the Allen-Cahn equation. *Computers & Mathematics with Applications*, 60(6):1591–1606, 2010.

[13] P. Motoni and M. Schatzman. Geometrical evolution of developed interfaces. *Transactions of the American Mathematial Society*, (347):1533–1589.

[14] Q. Nie, F. Y. M. Wan, Y.-T Zhang, and X. Liu. Compact integration factor methods in high spatial dimensions. *Journal of Computational Physics*, 227(10):5238–5255, 2008.

[15] John Rumsey. The stability of certain localized structures in the perturbed allen-cahn and gierer-meinhardt systems, 2009.

[16] F. Y. M. Wan. An in-core finite difference method for separable boundary value problems on a rectangle. *Studies in Applied Mathematics*, LII(2):103–113, 1973.

[17] F. Wendler, J. K. Becker, B. Nestler, P. D. Bons, and N. P. Walte. Phase-field simulations of partial melts in geological materials. *Computers & Geosciences*, 35(9):1907–1916, 2009.

[18] J. Zhang and Q. Du. Numerical studies of discrete approximations to the allen-cahn equation in the sharp interface limit. *SIAM Journal on Scientific Computing*, 31(4):3042–3063, 2009.