ALTERNATING DIRECTION IMPLICIT METHOD FOR THE ELECTRO-CARDIOLOGY MODELS

by

Zeinab Rammal

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Arts and Sciences
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Mathematics

Sharjah, United Arab Emirates

May 2017

Approval Signatures

We, the undersigned, approve the Master's Thesis of Zeinab Rammal.

Thesis Title: Alternating Direction Implicit Method for the Electro-Cardiology Models

Signature	**Date of Signature** (dd/mm/yyyy)
_____	_____
Dr. Youssef Belhamadia Assistant Professor Thesis Advisor	
_____	_____
Dr. Ali Sayfy Professor Thesis Committee Member	
_____	_____
Dr. Rim Gouia Assistant Professor Thesis Committee Member	
_____	_____
Dr. Hana Sulieman Head of the Department of Mathematics	
_____	_____
Dr. James Griffin CAS Graduate Programs Director	
_____	_____
Dr. Mahmoud Anabtawi Dean of CAS	
_____	_____
Dr. Khaled Assaleh Interim Vice Provost for Research and Graduate Studies	

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Youssef Belhamadia whose expertise, immense knowledge, and motivation led to the completion of writing this thesis. I attribute this completion to his constructive criticism and support.

Besides my advisor, I would like to thank the teaching faculty of the graduate program at American University of Sharjah. Their insightful comments incentivised me to widen my mathematical understanding from various perspectives.

I would also appreciate the remarkable support from my friend and sister Rana Itani. I thank her for stimulating me to peruse my graduate studies, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last three years. Without her precious care it would not have been possible to finalize this work.

Last but not the least, I would like to thank my students and fellow colleagues at Sharjah American International School, and my family for supporting me spiritually throughout writing this thesis and my career life in general.

I must acknowledge my husband and best friend, Hassan, without his love and encouragement, I would not have finished this thesis.

Abstract

Electrophysiology is an area of science that led to innovative experimentations between clinicians and scientists. Studying and understanding cardiac dynamics plays a key role in designing therapies, preoperative planning, and studying arrhythmias, fibrillations and other cardiac anomalies. Exploring the heart dynamics in three dimensions in vivo is difficult, and hence an alternative is needed. Mathematical modelling offers a valuable, yet computationally expensive tool for such exploration. The mathematical models of the electrical activity of the heart consist of a system of nonlinear partial differential equations coupled with a system of stiff ordinary differential equations. Numerical simulation of this coupled system requires accurate space and time discretization. In this work, an Alternating Direction Implicit Method *(ADI)* is presented as a new numerical scheme for solving the electrical model of the heart. Two-dimensional numerical results are presented illustrating the advantages and robustness of this proposed method. Comparison with the Crank-Nicolson Adams Bashforth ($CNAB$) method is demonstrated and the advantages of ADI are explained in terms of run time and memory consumption.

Contents

Abstract **4**

List of Figures **6**

List of Tables **7**

1 Introduction **8**
1.1 Biological Background 8
1.2 Mathematical Work 9
1.3 Thesis Structure 12

2 Mathematical Models **13**
2.1 Bidomain Model 13
2.2 Monodomain Model 15
2.3 Aliev-Panfilov Model 16

3 The Alternating Direction Implicit Method **17**
3.1 Solving the Heat Equation using *ADI* 17
3.1.1 Dirichlet boundary conditions 18
3.1.2 Neumann boundary conditions 23
3.2 *ADI* Applied to Aliev-Panfilov Monodomain Model 26

4 Numerical Results **29**
4.1 Analytical Solution 29
4.2 Numerical Results for the Aliev-Panfilov Monodomain Model . . . 30
4.2.1 The *CNAB* scheme 31
4.2.2 *ADI* v.s. *CNAB* 32

5 Conclusions and Future Work **36**

Bibliography **38**

Vita **44**

List of Figures

1.1 Simplified diagram of the heart 8
1.2 Heart's electrical system 9

3.1 Finite-difference mesh on the x-y plane 17
3.2 Implicit x direction for ADI 19
3.3 Implicit y direction for ADI 20
3.4 Virtual points 23

4.1 Evolution of transmembrane potential over time for the 2D Aliev-Panfilov monodomain model. 33
4.2 Memory Consumption of the ADI and $CNAB$ Methods 34
4.3 Run Time of the ADI and $CNAB$ Methods 35

List of Tables

4.1 Verification of time order 30
4.2 Verification of space order 30
4.3 Parameter values for the Aliev-Panfilov Monodomain model 32

Chapter 1: Introduction

1.1 Biological Background

The heart is a muscular organ about the size of a fist, located just behind and slightly left of the breastbone. The heart pumps blood by contracting and expanding about 2.5 billion times during a human life. The heart has four chambers: right atrium, left atrium, right ventricle and left ventricle. The chambers are separated by atrioventricular valves (see Fig. 1.1).

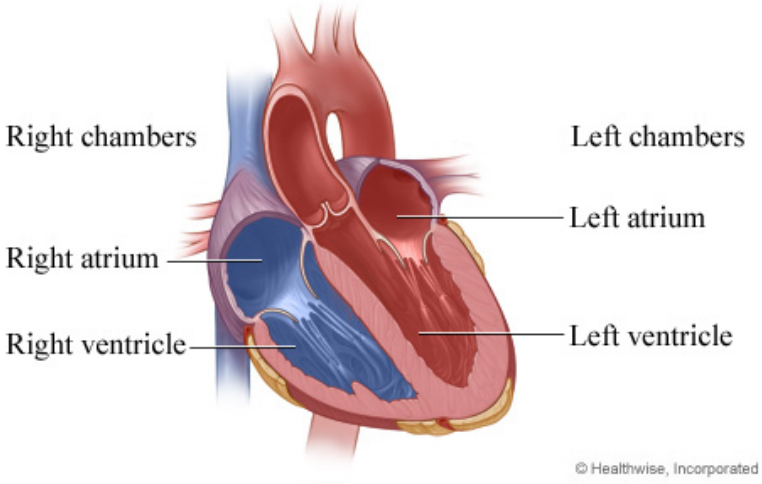


Figure 1.1: Simplified diagram of the heart

www.img.webmd.com

The heart's electrical system is responsible for creating the signals that trigger the heart to beat. The electrical signal originates from the sinoatrial node (SA) node which is located just below the superior vena cava on the right atrium. The action potential that is generated by the (SA) node continues propagation through the atrioventricular (AV) node located at the base of the atria. As the action potential exists the (AV) node, it propagates through the bundle of His (see Fig. 1.2). This signal prompts the hearts muscle to contract. With each contraction, blood is pumped to the lungs and the rest of the body.

Many heart problems result from irregularities in the flow of cardiac electricity. Studying these irregularities motivated developing mathematical models of the electrical activity in the cardiac tissue.

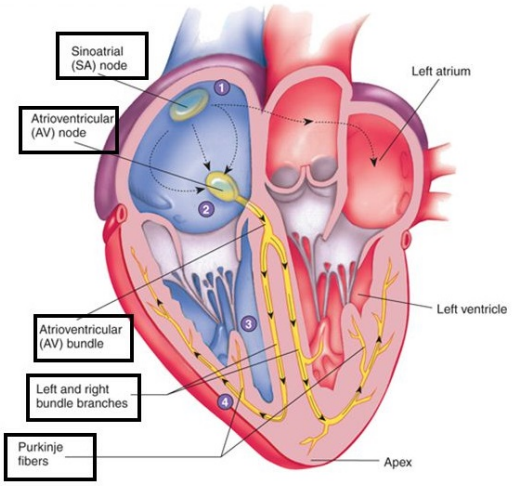


Figure 1.2: Heart's electrical system

www.images.slideplayer.com

1.2 Mathematical Work

Mathematical modeling for single myocardial cell can be represented using stiff ordinary differential equations (*ODEs*). Describing the electrical activity in the whole heart requires coupling a single cell model with a system of partial differential equations (*PDEs*) that describes the flow of electricity in a network of cells. One of these models, known as the bidomain model, consists of two *PDEs* coupled at each point in space with a system of stiff *ODEs*.

The bidomain model was proposed first by Tung and Geselowitz in 1978 [1], and is now widely used to model the electrical behaviour of the cardiac tissue due to the fact that it uses realistic physiological parameter values based on experimental measurement (for a review of the bidomain model see S. Kandel. [2]).

The solution of the bidomain equations in three dimensional geometry is a computationally challenging problem. The main source causing the computational load in solving the bidomain model is the rapid dynamics of the cellular reactions described by the system of *ODEs* and the ionic current term in the *PDEs*. The fast reactions of the calcium ion results in narrowing the wavefront of activation that propagates through the cardiac tissue. Grasping these fast variations necessitates a high resolution in time and space. Consequently, a large systems of equations must be solved for each time step.

The first decision to take when developing a numerical scheme for solving any partial differential equation *PDE* is the choice of discretization in time. In the literature, different approaches were implemented for time discretization for the bidomain model. Explicit discretization was used by Puwal and Roth [3]. Fully explicit methods are simple to code, but the time step used in such discretization suffers from stability restrictions.

Fully implicit discretization was used by Murillo and Cai [4], Ying et al. [5], Dal et al. [6], and Belhamadia [7]. The choice of this type of discretization removes any stability restrictions on the time step used. However, it requires solving the entire coupled system simultaneously which requires solving a massive nonlinear system.

Semi-implicit discretization was implemented by Keener and Bogar [8], Franzone and Pavarino [9], and Bourgault and Ethier [10]. The latter suggested the Crank-Nicolson Adams Bashforth (*CNAB*) scheme as an impressive scheme that provides second-order accuracy in time.

Operator splitting is another common approach for time discretization that was implemented by Trangenstein and Kim [11], Schroll et al. [12], Spiteri and Ziaratgahi [13], and Sundnes et al. [14]. This method is used for solving the bidomain model to reduce the computational load resulting from the coupling of the *PDEs* and the system of *ODEs*. In the latter work, a second-order in time operator splitting method to solve the bidomain method was formulated. Due to the complexity accompanied with this method, it is hard to have a detailed mathematical analysis of its accuracy and stability, and "few of results of this kind have been reported in the literature" as stated in Linge et al. [15].

In addition to choosing an appropriate time discretization scheme, the use of a space discretization scheme remains important to improve the spatial approximation. In the literature, variety of numerical methods were used for space discretization using uniform meshes. Franzone et al. [16], Pathmanathan et al. [17], Bordas et al. [18], and Scacchi [19] used second order finite element method.

An alternative to using uniform meshes is implementing adaptive mesh, as it is extremely helpful to improve the accuracy of the simulations. Whiteley [20]

used adaptive scheme combined with the multi-grid method whereas Cherry et al. [21] used finite volume scheme enriched by a fully adaptive multi-resolution method for the monodomain and bidomain models. Adaptive mesh refinement algorithm ($AMRA$) that was developed by Berger and Colella [22] was applied by Trangenstein and Kim [11] to simulate single propagating pulse in different monodomain models. Belhamadia [23] proposed, in the context of electro-cardiology, an enhancement of the adaptive methods by considering anisotropic mesh adaptation techniques, where the mesh elements can be elongated along some appropriate directions. (For more details see [24] and [25]).

Higher order methods were also used for enhancing the space simulation. However, to our knowledge only Heidenreich et al. [26] worked on developing fourth order compact schemes for solving the bidomain model.

Based on the above review, it is noticed that accurate numerical schemes are needed to solve the bidomain model. Yet, the proposed schemes for solving suffered from run-time and memory restrictions. In our case, we suggest implementing the Alternating Direction Implicit Method (ADI) to solve the electro-cardiology model. This method has the advantages of reducing memory intake and run-time and can be easily improved to higher orders in space.

ADI methods reduce the solution of multidimensional problems to a set of one-dimensional problems such that each of them can be solved efficiently. The solution of the multidimensional problems is then obtained iteratively from the solutions of the one-dimensional problems. The reduction in the structure decreases memory intake and run-time, and hence, triumphs the ADI over the introduced schemes.

Alternating direction implicit (ADI) methods were first introduced as a finite difference method by Peaceman and Rachford in 1955 [27] for solving two dimensional parabolic and elliptic Partial differential equations. ADI has been proven to be second-order in time by Douglas [28].

Since their formulation, ADI methods have been studied extensively for solving variety of problems. Thomas et al. [29] described the ADI scheme as a cost effective technique with stability and accuracy, in comparison to other standard

finite-element method using analytical solutions for two problems approximating different stages in steel ingot processing. Abimbola and Bright [30] used *ADI* for solving the Laplace's equation governing the two dimensional heat conduction in a metallic bar. Aderito et al. [31] solved a two-dimensional hyperbolic diffusion problem, where it is assumed that both convection and diffusion are responsible for flow motion, using *ADI* method and showcased its accuracy. Wang and Chen [32] employed the *ADI* scheme to solve transient thermal problems and Karaa and Zhang [33] used it to study unsteady convection-diffusion problems. Fernandes et al. [34] used an *ADI* extrapolated Crank-Nicolson orthogonal spline collocation method to solve nonlinear reaction-diffusion systems. Their method was tested on the Schnakenberg model and demonstrated numerically that it is second-order accurate in time and of optimal accuracy in space.

Alternating Direction Implicit Method's special accuracy has been also a field of interest. Using the standard fourth-order Pade scheme, Gao and Xie [35] developed a fourth-order accurate in space and second-order accurate in time *ADI* method to solve two-dimensional Schrodinger equations. Based on higher order compact (*HOC*) difference schemes Xu and Zhang [36] proposed a spatially fourth-order and temporally second-order accurate *HOC-ADI* method to solve the two-dimensional cubic non linear Schrodinger equations.

This review is not exhaustive, as *ADI* has been widely used in the literature. However, to our knowledge, there is no work that implements an *ADI* method to solve the cardiac models, and this work aims at laying the foundation for such implementation.

1.3 Thesis Structure

The remainder of this thesis is structured into five chapters. In Chapter 2, the mathematical models used for representing the electrical activity of the heart is explained and the *ADI* method, the numerical approach chosen to solve the model, is introduced in Chapter 3. The numerical verification of *ADI* along with the comparison between *ADI* and *CNAB* are illustrated in Chapter 4 and Chapter 5 is devoted to results and conclusion.

Chapter 2: Mathematical Models

In this chapter, we explain the formulation of the mathematical models used to describe the cardiac electrical activity. The bidomain model is discussed in section 2.1. A common, albeit unrealistic, simplification of the bidomain model, known as the monodomain model, is presented in section 2.2, and the coupling between the monodomain model and the Aliev-Panfilov ionic model is introduced in section 2.3.

2.1 Bidomain Model

The heart tissue is classified into two domains, the intracellular domain and the extracellular domain. The bidomain model treats the intracellular domain and the extracellular domain as overlapping by considering that every point in the myocardium lies in both the intracellular and the extracellular domains, and accordingly, possess two electrical potentials: intracellular potential ϕ_e and extracellular potential ϕ_i, as well as two electrical currents G_i and G_e. The difference between the two electrical potentials is known as the transmembrane potential, V, given by the relation

$$V = \phi_i - \phi_e$$

. The derivation of the bidomain model begins with Ohm's Law. For each domain, the ohmic relationship between the current and the potential results in the following equations:

$$G_i = -\sigma_i \nabla \phi_i \tag{2.1}$$

$$G_e = -\sigma_e \nabla \phi_e \tag{2.2}$$

with σ_i and σ_e denoting the conductivity tensors.

Considering that the heart is in isolation; any current leaving one domain must enter the other. Thus, a change in current density will be of equal magnitude in both domains but with opposite sign, which translates into the following equation:

$$\text{-}\nabla G_i = \nabla G_e \tag{2.3}$$

The current flow across the membrane must be equal to either side of equation (2.3) which may be seen as a time-dependent capacitive current along with an ionic current. Hence, using Kirchhoff's current law, we obtain:

$$-\nabla G_i = \nabla G_e = \chi(C_m \frac{\partial V}{\partial t} + I_{ion}), \tag{2.4}$$

where C_m is the capacitance of the cell membrane, I_{ion} is the ionic current across the cell membrane, and χ is the area of the cell membrane per unit to volume.

Using equations (2.1) and (2.2) with equation (2.4) we obtain:

$$\nabla \cdot (\sigma_i \nabla \phi_i) = \chi \left(C_m \frac{\partial V}{\partial t} + I_{ion} \right) \tag{2.5}$$

$$\nabla \cdot (\sigma_e \nabla \phi_e) = -\chi \left(C_m \frac{\partial V}{\partial t} + I_{ion} \right) \tag{2.6}$$

The difference in signs between (2.5) and (2.6) is due to defining the positive direction of current flow to be from the intracellular domain to the extracellular domain. Accordingly, (2.5) and (2.6) result in:

$$\nabla \cdot (\sigma_i \nabla \phi_i) + \nabla \cdot (\sigma_e \nabla \phi_e) = 0 \tag{2.7}$$

Using the definition of V we can simplify equations (2.5) and (2.7) by eliminating the intracellular potential. This would result in the standard formulation of the bidomain model:

$$\begin{cases} \nabla \cdot (\sigma_i \nabla V) + \nabla \cdot (\sigma_i \nabla \phi_e) = \chi \left(C_m \dfrac{\partial V}{\partial t} + \chi I_{ion} \right) \\ \nabla \cdot (\sigma_i \nabla V) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e) = 0 \end{cases} \tag{2.8}$$

Typically, ordinary differential equations *ODEs* system representing the ionic model at single cell level is coupled with (2.8) via cellular states, W_n. Some of these *ODEs* models include Hodgkin-Huxley model [37], FitzHugh-Nagumo model [38], Luo-Rudy model [39], and Aliev-Panfilov model [40]. The choice of the *ODEs* system will be discussed later in this chapter. However, we the general

form of the bidomain model including the ionic model can be written as:

$$\begin{cases} \nabla \cdot (\sigma_i \nabla V) + \nabla \cdot (\sigma_i \nabla \phi_e) = \chi \left(C_m \dfrac{\partial V}{\partial t} + I_{ion} \right) \\ \nabla \cdot (\sigma_i \nabla V) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e) = 0 \\ \dfrac{\partial W_n}{\partial t} = g\,(V, W_n) \end{cases} \tag{2.9}$$

where $g(V, W_n)$ depends on the ionic model selected.

2.2 Monodomain Model

A simplified cardiac tissue model is the monodomain model where less analysis and computations are required (the reader is referred to [41] and the references therein). The computational cost of using the monodomain model reduces the cost by a factor of one-half to one-tenth of that obtained when using the bidomain model. The range of this computational cost depends on the complexity of the cell model used (see [42] for further details). The simplification from the bidomain model to the monodomain model occurs by assuming that: $\sigma_e = \lambda \sigma_i$, where $\lambda \in \mathbb{R}$, i.e., an assumption of equal anisotropy is applied. Replacing $\sigma_e = \lambda \sigma_i$ in the bidomain model gives:

$$\nabla \cdot (\sigma_i \nabla V) + \nabla \cdot (\sigma_i \nabla \phi_e) = \chi \left(C_m \frac{\partial V}{\partial t} + I_{ion} \right) \tag{2.10}$$

$$\nabla \cdot (\sigma_i \nabla V) + (1 + \lambda) \nabla \cdot (\sigma_i \nabla \phi_e) = 0 \tag{2.11}$$

Equation (2.11) implies:

$$\nabla \cdot (\sigma_i \nabla \phi_e) = -\frac{1}{1 + \lambda} \nabla \cdot (\sigma_i \nabla V)\,. \tag{2.12}$$

Using Equation (2.12) in (2.10) results in a single *PDE*. Rearranging the terms, we obtain the standard formulation of the monodomain model:

$$\begin{cases} -\nabla \cdot (D \nabla V) = \chi \left(C_m \dfrac{\partial V}{\partial t} + \chi I_{ion} \right) \\ \dfrac{\partial W_n}{\partial t} = g(V, W_n) \end{cases} \tag{2.13}$$

where $D = \sigma_i \frac{\lambda}{1+\lambda}$ and $g(V, W_n)$ depends on the ionic model selected.

2.3 Aliev-Panfilov Model

As discussed perviously, the choice of an *ODEs* system representing the ionic model at single cell level to be coupled with the *PDE* is still needed. In this thesis we will use the Aliev-Panfilov model [40]. Although it is a simplified model, it still reproduces realistic shape of the cardiac action potential. The model consists of the following equations:

$$I_{ion}(V, W) = kV(V - \alpha)(1 - V) - VW, \tag{2.14}$$

$$g(V, W) = \left(\epsilon + \frac{\mu_1}{\mu_2 + V}\right)(-W - kV\,(V - \alpha - 1))\,. \tag{2.15}$$

The monodomain equations using Aliev-Panfilov model therefore consists of a single *PDE* and a single *ODE* and takes the following form:

$$\begin{cases} \dfrac{\partial V}{\partial t} - \nabla \cdot (D \nabla V) = I_{ion}(V, W) \\ \dfrac{\partial W}{\partial t} = g(V, W), \end{cases} \tag{2.16}$$

The next chapter introduces the numerical scheme for solving this system of equations.

Chapter 3: The Alternating Direction Implicit Method

This chapter is devoted to introducing the detailed scheme of the Alternating Direction Implicit Method (ADI). The idea when solving using this method is to complete the time stepping $[t_n, t_{n+1}]$ with two sub-time steps: firstly on $[t_n, t_{n+\frac{1}{2}}]$ and secondly on $[t_{n+\frac{1}{2}}, t_{n+1}]$. In this chapter, the ADI scheme will first be implemented for the heat equation and then we will adjust the work for the monodomain equations.

3.1 Solving the Heat Equation using *ADI*

We will consider the heat equation that has the following form:

$$\begin{cases} \dfrac{\partial U}{\partial t} - \dfrac{\partial^2 U}{\partial x^2} - \dfrac{\partial^2 U}{\partial y^2} = f(x,y,t) \qquad in \quad \Omega \\ \qquad U(x,y,0) = U_0 \qquad in \quad \Omega \\ \qquad U(x,y,t) = g(x,y,t) \qquad at \quad \partial\Omega \end{cases} \tag{3.1}$$

The ADI method is a type of finite difference methods, and thus the continuous space domain should be discretized into a mesh with a finite number of grid points. As illustrated in figure 3.1, the variable $U(x,y,t)$ at each point will be replaced by $U(i\Delta_x, j\Delta_y, n\Delta_t)$, which shall be denoted as $U^n_{i,j}$ for the remainder of this thesis.

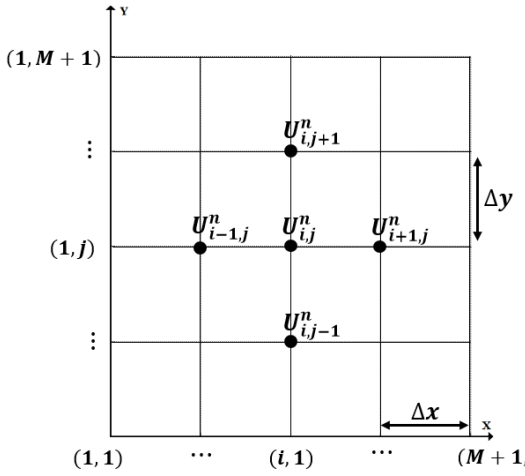


Figure 3.1: Finite-difference mesh on the x-y plane

The central difference representation of the second-partial derivative of U with respect to x can be written as:

$$\frac{\partial^2 U}{\partial x^2}(i\Delta_x, j\Delta_y, n\Delta_t) = \frac{U^n_{i+1,j} - 2U^n_{i,j} + U^n_{i-1,j}}{\Delta_x^2} \tag{3.2}$$

Similarly, the central difference representation of the second-partial derivative of U with respect to y can be written as:

$$\frac{\partial^2 U}{\partial y^2}(i\Delta_x, j\Delta_y, n\Delta_t) = \frac{U^n_{i,j+1} - 2U^n_{i,j} + U^n_{i,j-1}}{\Delta_y^2} \tag{3.3}$$

which are second order in space.

We first discuss the *ADI* scheme with Dirchilet boundary condition and afterwards we modify our work to Neumann boundary conditions.

3.1.1. Dirichlet boundary conditions.

ADI method is a second-order in time method. To obtain the solution $U^{n+1}_{i,j}$ while preserving this time-order, the following time-dependent algorithm will be followed:

Step 1: $[t_n, t_{n+\frac{1}{2}}]$

In this step, we first get a solution at the mid time $t_{n+\frac{1}{2}}$, based on the following equation:

For each j from 2 to M

$$\begin{aligned} \frac{U^{n+\frac{1}{2}}_{ij} - U^n_{ij}}{\Delta_t} &= \frac{U^{n+\frac{1}{2}}_{i-1,j} - 2U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i+1,j}}{2\Delta_x^2} + \\ \frac{U^n_{i,j-1} - 2U^n_{i,j} + U^n_{i,j+1}}{2\Delta_y^2} &+ \frac{f^{n+\frac{1}{2}}_{i,j}}{2} \quad 2 \leqslant i \leqslant M \end{aligned} \tag{3.4}$$

For $j = 1$ and $j = M + 1$ we use Dirichlet boundary conditions:

$$U^{n+\frac{1}{2}}_{ij} = g^{n+\frac{1}{2}}_{i,j} = g\left(i\Delta_x, j\Delta_y, \left(n + \frac{1}{2}\right)\Delta_t\right) \quad 1 \leqslant i \leqslant M+1 \tag{3.5}$$

Similarly, for $i = 1$ and $i = M + 1$, we use the exact solution.

The above step applies an implicit update in the x-direction and explicit

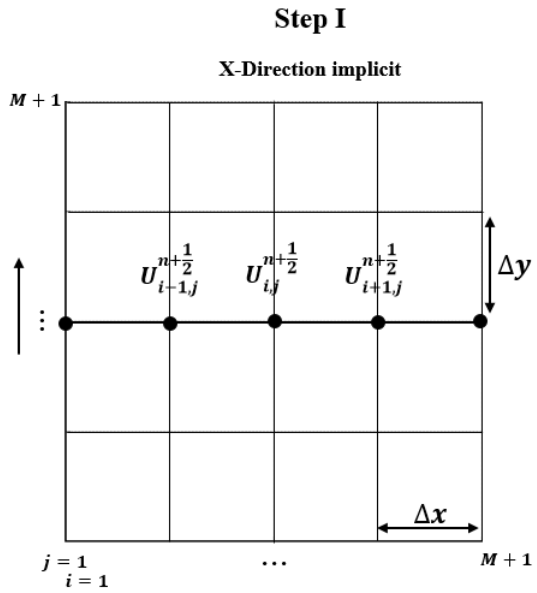


Figure 3.2: Step I: the x-direction is implicit and the y-direction is explicit

update in the y-direction on the time interval $[t_n, t_{n+\frac{1}{2}}]$ (see figure 3.2).

Step 2: $[t_{n+\frac{1}{2}}, t_{n+1}]$

Now, the final solution at the time step t_{n+1} is calculated using the following equation

For each i from 2 to M

$$\begin{aligned} \frac{U_{ij}^{n+1} - U_{ij}^{n+\frac{1}{2}}}{\Delta_t} &= \frac{U_{i-1,j}^{n+\frac{1}{2}} - 2U_{i,j}^{n+\frac{1}{2}} + U_{i+1,j}^{n+\frac{1}{2}}}{2\Delta_x^2} + \\ \frac{U_{i,j-1}^{n+1} - 2U_{i,j}^{n+1} + U_{i,j+1}^{n+1}}{2\Delta_y^2} &+ \frac{f_{i,j}^{n+\frac{1}{2}}}{2} \quad 2 \leqslant j \leqslant M \end{aligned} \tag{3.6}$$

For $i = 1$ and $i = M + 1$ we use Dirichlet boundary conditions:

$$U_{ij}^{n+\frac{1}{2}} = g_{i,j}^{n+1} = g\left(i\Delta_x, j\Delta_y, (n+1)\,\Delta_t\right) \quad 1 \leqslant j \leqslant M+1 \tag{3.7}$$

Similarly, for $j = 1$ and $j = M + 1$, we use the exact solution.

In this step, the method applies an implicit update in the y-direction and explicit update in the x-direction on the time interval $[t_{n+\frac{1}{2}}, t_{n+1}]$ (see figure 3.3).

To obtain a clear algorithm that can be easily implemented in terms of

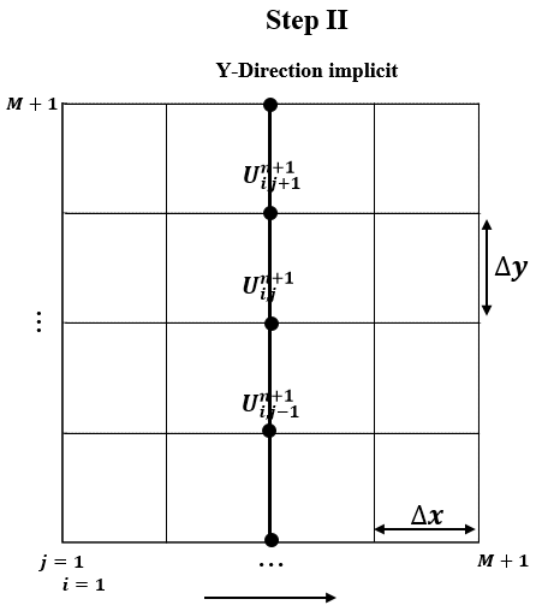


Figure 3.3: Step II: the y-direction is implicit and the x-direction is explicit

matrices we need to simplify the equations obtained. For this simplification, we set $r_x = \dfrac{\Delta_t}{\Delta_x^2}$ and $r_y = \dfrac{\Delta_t}{\Delta_y^2}$. Therefore, the algorithm translates to the following two steps:

Step 1: $[t_n, t_{n+\frac{1}{2}}]$

for $j = 1$

$$U_{i,1}^{n+\frac{1}{2}} = U_{exact}^{n+\frac{1}{2}}(i,1) \quad 1 \leqslant i \leqslant M+1 \tag{3.8}$$

for $j = 2$ to M

$$U_{1,j}^{n+\frac{1}{2}} = U_{exact}^{n+\frac{1}{2}}(1,j)$$

$$\begin{aligned} &-r_x U_{i-1,j}^{n+\frac{1}{2}} + (2+2r_x)U_{i,j}^{n+\frac{1}{2}} - r_x U_{i+1,j}^{n+\frac{1}{2}} \\ &= r_y U_{i,j-1}^{n} + (2-2r_y)U_{i,j}^{n} + r_y U_{i,j+1}^{n} + \tfrac{\Delta_t}{2} f_{i,j}^{n+\frac{1}{2}} \quad 2 \leqslant i \leqslant M \end{aligned} \tag{3.9}$$

$$U_{M+1,j}^{n+\frac{1}{2}} = U_{exact}^{n+\frac{1}{2}}(M+1,j)$$

for $j = M+1$

$$U_{i,M+1}^{n+\frac{1}{2}} = U_{exact}^{n+\frac{1}{2}}(i,M+1) \quad 1 \leqslant i \leqslant M+1 \tag{3.10}$$

Step 2: $[t_{n+\frac{1}{2}}, t_{n+1}]$

for $i = 1$

$$U_{1,j}^{n+1} = U_{exact}^{n+1}(1,j) \quad 1 \leqslant j \leqslant M+1 \tag{3.11}$$

for $i = 2$ to M

$$U_{i,1}^{n+1} = U_{exact}^{n+1}(i,1)$$

$$\begin{aligned} &-r_y U_{i-1,j}^{n+1} + (2+2r_y)U_{i,j}^{n+1} - r_y U_{i+1,j}^{n+1} \\ &= r_x U_{i,j-1}^{n+\frac{1}{2}} + (2-2r_x)U_{i,j}^{n+\frac{1}{2}} + r_x U_{i,j+1}^{n+\frac{1}{2}} + \tfrac{\Delta_t}{2} f_{i,j}^{n+\frac{1}{2}} \quad 2 \leqslant j \leqslant M \end{aligned} \tag{3.12}$$

$$U_{i,M+1}^{n+1} = U_{exact}^{n+1}(i,M+1)$$

for $i = M+1$

$$U_{M+1,j}^{n+1} = U_{exact}^{n+1}(M+1,j) \quad 1 \leqslant j \leqslant M+1 \tag{3.13}$$

The two steps listed in equations (3.8),(3.9),(3.10), (3.11),(3.12),and (3.13) can be now translated to a matrix vector form.

Step 1: $[t_n, t_{n+\frac{1}{2}}]$

for each $j = 1$ to $M+1$, $U_{i,j}^{n+\frac{1}{2}}$ are solutions of the following system:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & ... & 0 \\ -r_x & 2+2r_x & -r_x & 0 & 0 & ... & 0 \\ 0 & -r_x & 2+2r_x & -r_x & 0 & ... & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & -r_x & 2+2r_x & -r_x \\ 0 & 0 & ... & ... & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} U_{1,j}^{n+\frac{1}{2}} \\ U_{2,j}^{n+\frac{1}{2}} \\ \vdots \\ \vdots \\ \vdots \\ U_{M,j}^{n+\frac{1}{2}} \\ U_{M+1,j}^{n+\frac{1}{2}} \end{pmatrix} = B$$

where $B =$

$$\begin{pmatrix} U_{exact}^{n+\frac{1}{2}}(1,j) \\ r_y U_{2,j-1}^n + (2-2r_y)U_{2,j}^n + r_y U_{2,j+1}^n + \frac{\Delta_t}{2} f_{2,j}^{n+\frac{1}{2}} \\ \vdots \\ \vdots \\ \vdots \\ r_y U_{M,j-1}^n + (2-2r_y)U_{M,j}^n + r_y U_{M,j+1}^n + \frac{\Delta_t}{2} f_{M,j}^{n+\frac{1}{2}} \\ U_{exact}^{n+\frac{1}{2}}(M+1,j) \end{pmatrix}$$

Step 2: $[t_{n+\frac{1}{2}}, t_{n+1}]$

for each $i = 1$ to $M+1$, $U_{i,j}^{n+1}$ are solutions of the following system:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & ... & 0 \\ -r_y & 2+2r_y & -r_y & 0 & 0 & ... & 0 \\ 0 & -r_y & 2+2r_y & -r_y & 0 & ... & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & -r_y & 2+2r_y & -r_y \\ 0 & 0 & ... & ... & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} U_{i,1}^{n+1} \\ U_{i,2}^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ U_{i,M}^{n+1} \\ U_{i,M+1}^{n+1} \end{pmatrix} = D$$

where $D =$

$$\begin{pmatrix} U_{exact}^{n+1}(j,1) \\ r_x U_{i-1,2}^{n+\frac{1}{2}} + (1-2r_x)U_{i,2}^{n+\frac{1}{2}} + r_x U_{i+1,2}^{n+\frac{1}{2}} + \frac{\Delta_t}{2} f_{i,2}^{n+\frac{1}{2}} \\ \vdots \\ \vdots \\ \vdots \\ r_x U_{i-1,M}^{n+\frac{1}{2}} + (1-2r_x)U_{i,M}^{n+\frac{1}{2}} + r_x U_{i+1,M}^{n+\frac{1}{2}} + \frac{\Delta t}{2} f_{i,M}^{n+\frac{1}{2}} \\ U_{exact}^{n+1}(j,M+1) \end{pmatrix}$$

Dirichlet boundary conditions was smoothly implemented. On the contrary, im-

plementing the Neumann boundary conditions necessitates more details to be included, all of which will be introduced in the next section.

3.1.2. Neumann boundary conditions.

For the Neumann boundary conditions, instead of having fixed values on the bounded surfaces, it uses the derivatives i.e. $\frac{\partial U}{\partial x} = g(x, y, t)$. In order to achieve the second-order accuracy, the central-difference approximation will be used to discretize the boundary conditions. This implies the introduction of the virtual points $U^n_{0,j}$, $U^n_{M+2,j}$, $U^n_{i,0}$, and $U^n_{i,M+2}$ by expanding the region with Δ_x to the left and right and with Δ_y to the bottom and top as illustrated in figure (3.4).

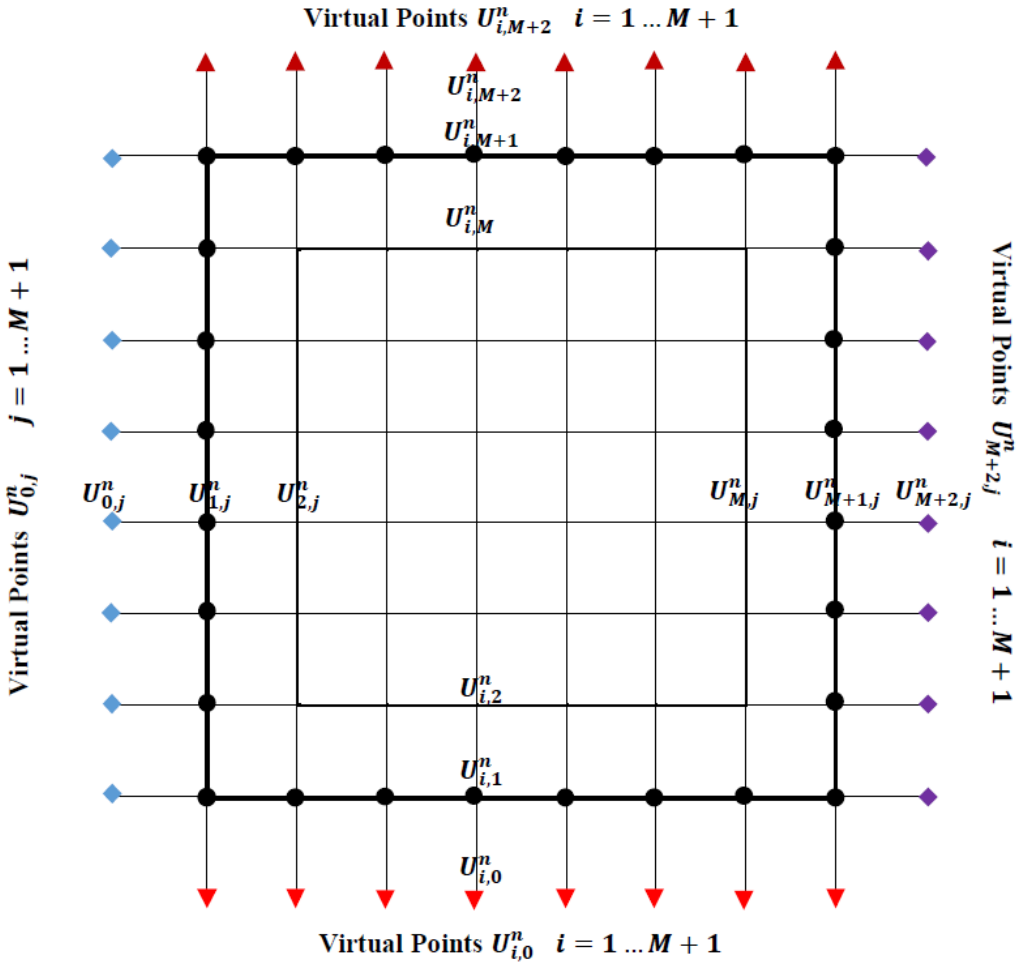


Figure 3.4: Virtual points

For example, at $i = 1$ and time step $n+\frac{1}{2}$ the boundary condition becomes:

$$\frac{U_{2,j}^{n+\frac{1}{2}} - U_{0,j}^{n+\frac{1}{2}}}{2\Delta_x} = g(x)_{1,j}^{n+\frac{1}{2}}$$

where

$$g(x)_{1,j}^{n+\frac{1}{2}} = g\left(1\Delta_x, j\Delta_y, \left(n+\frac{1}{2}\right)\Delta_t\right)$$

And thus, the virtual point $U_{0,j}^{n+\frac{1}{2}}$ can be expressed as:

$$U_{0,j}^{n+\frac{1}{2}} = U_{2,j}^{n+\frac{1}{2}} - 2\Delta_x(gx)_{1,j}^{n+\frac{1}{2}} \tag{3.14}$$

The other virtual points, $U_{M+2,j}^{n+\frac{1}{2}}$, $U_{i,0}^{n+\frac{1}{2}}$, and $U_{i,M+2}^{n+\frac{1}{2}}$ can be obtained using the same way. These virtual points can then be used in the difference equations (3.8),(3.9),(3.10), (3.11),(3.12),and (3.13) whenever needed.

The difference equation at the point $(1, j)$ in equation (3.9) has the form:

$$\begin{aligned} &-r_x U_{0,j}^{n+\frac{1}{2}} + (2+2r_x)U_{1,j}^{n+\frac{1}{2}} - r_x U_{2,j}^{n+\frac{1}{2}} \\ &= r_y U_{1,j-1}^{n} + (2-2r_y)U_{1,j}^{n} + r_y U_{1,j+1}^{n} + \Delta_t f_{1,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.15}$$

The virtual point $U_{0,j}^{n+\frac{1}{2}}$ can be replaced with its equivalent derived in (3.14), and thus equation (3.15) at the points $(1, j)$ for the boundary conditions becomes:

$$\begin{aligned} &(2+2r_x)U_{1,j}^{n+\frac{1}{2}} - 2r_x U_{2,j}^{n+\frac{1}{2}} \\ &= r_y U_{1,j-1}^{n} + (2-2r_y)U_{1,j}^{n} + r_y U_{1,j+1}^{n} + \Delta_t f_{1,j}^{n+\frac{1}{2}} - 2r_x\Delta_x(gx)_{1,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.16}$$

The remaining difference equations on the boundaries $i = M+1$, $j = 1$, and $j = M+1$ can be obtained similarly. Therefore the *ADI* algorithm with Neumann boundary conditions simplifies to:

Step 1: $[t_n, t_{n+\frac{1}{2}}]$ for $j = 1$ to $M+1$:

for $i = 1$:

$$\begin{aligned} &(2+2r_x)U_{1,j}^{n+\frac{1}{2}} - 2r_x U_{2,j}^{n+\frac{1}{2}} \\ &= r_y U_{1,j-1}^{n} + (2-2r_y)U_{1,j}^{n} + r_y U_{1,j+1}^{n} + \Delta_t f_{1,j}^{n+\frac{1}{2}} - 2r_x\Delta_x(gx)_{1,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.17}$$

for $i = 2$ to M

$$\begin{aligned} &-r_x U_{i-1,j}^{n+\frac{1}{2}} + (2 + 2r_x) U_{i,j}^{n+\frac{1}{2}} - r_x U_{i+1,j}^{n+\frac{1}{2}} \\ &= r_y U_{i,j-1}^{n} + (2 - 2r_y) U_{i,j}^{n} + r_y U_{i,j+1}^{n} + \tfrac{\Delta_t}{2} f_{i,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.18}$$

for $i = M + 1$

$$\begin{aligned} &-2r_x U_{M,j}^{n+\frac{1}{2}} + (2 + 2r_x) U_{M+1,j}^{n+\frac{1}{2}} \\ &= r_y U_{M+1,j-1}^{n} + (2 - 2r_y) U_{M+1,j}^{n} + r_y U_{M+1,j+1}^{n} + \Delta_t f_{M+1,j}^{n+\frac{1}{2}} + 2r_x \Delta_x (gx)_{M+1,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.19}$$

Step 2: $[t_{n+\frac{1}{2}}, t_{n+1}]$ for $i = 1$ to $M + 1$:
for $j = 1$:

$$\begin{aligned} &(2 + 2r_y) U_{i,1}^{n+1} - 2r_y U_{i,2}^{n+1} \\ &= r_x U_{i-1,1}^{n+\frac{1}{2}} + (2 - 2r_x) U_{i,1}^{n+\frac{1}{2}} + r_x U_{i+1,1}^{n+\frac{1}{2}} + \Delta_t f_{i,1}^{n+\frac{1}{2}} - 2r_y \Delta_y (gy)_{i,1}^{n+1} \end{aligned} \tag{3.20}$$

for $j = 2$ to M

$$\begin{aligned} &-r_y U_{i-1,j}^{n+1} + (2 + 2r_y) U_{i,j}^{n+1} - r_y U_{i+1,j}^{n+1} \\ &= r_x U_{i,j-1}^{n+\frac{1}{2}} + (2 - 2r_x) U_{i,j}^{n+\frac{1}{2}} + r_x U_{i,j+1}^{n+\frac{1}{2}} + \tfrac{\Delta_t}{2} f_{i,j}^{n+\frac{1}{2}} \end{aligned} \tag{3.21}$$

for $j = M + 1$

$$\begin{aligned} &-2r_y U_{i,M}^{n+1} + (2 + 2r_y) U_{i,M+1}^{n+1} \\ &= r_x U_{i-1,M+1}^{n+\frac{1}{2}} + (2 - 2r_x) U_{i,M+1}^{n+\frac{1}{2}} + r_x U_{i+1,M+1}^{n+\frac{1}{2}} \Delta_t f_{i,M+1}^{n+\frac{1}{2}} + 2r_y \Delta_y (gy)_{i,M+1}^{n+1} \end{aligned} \tag{3.22}$$

Since the monodomain equations using Aliev-Panfilov model consist of a source term that is U-dependent, and to avoid having a nonlinear system, a different discretization will be applied for $f(U)$ (see [**?**] for further explanation).

The discretization will take the form:

$$f_{i,j}^{n+\frac{1}{2}} = f\left(\frac{3U_{i,j}^{n} - U_{i,j}^{n-1}}{2}\right) \tag{3.23}$$

3.2 *ADI* Applied to Aliev-Panfilov Monodomain Model

Recall that, the Aliev-Panfilov monodomain model reads as:

$$\begin{cases} \dfrac{\partial V}{\partial t} = \nabla.(\mathbf{D})\nabla V + f(V,W) \\ \dfrac{\partial W}{\partial t} = g(V,W) \end{cases} \tag{3.24}$$

where

$$\begin{cases} f(V,W) = -kV(V-\alpha)(V-1) - VW \\ g(V,W) = (\epsilon + \dfrac{\mu_1 W}{\mu_2 + V})(-W - kV(V-\alpha-1)) \end{cases} \tag{3.25}$$

where μ_1, μ_2, α, ϵ, and k are physical parameters and are determined in the next chapter. The boundary conditions considered are homogeneous Neumann boundary conditions.

The implementation of the *ADI* method for the Aliev-Panfilov monodomain model is similar to the heat equation using Neumann boundary conditions presented in the previous section and thus the first step is implemented on the time interval $[t_n, t_{n+\frac{1}{2}}]$. This step will enable finding both $V^{n+\frac{1}{2}}$ and $W^{n+\frac{1}{2}}$. The difference equations governing this step therefore become:

Step 1:

for $i = 1$:

$$\begin{aligned} &(2+2r_x)V_{1,j}^{n+\frac{1}{2}} - 2r_x V_{2,j}^{n+\frac{1}{2}} \\ &= r_y V_{1,j-1}^{n} + (2-2r_y)V_{1,j}^{n} + r_y V_{1,j+1}^{n} + \frac{\Delta_t}{2}\cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.26}$$

$$W_{1,j}^{n+\frac{1}{2}} = W_{1,j}^{n} + \frac{\Delta_t}{4}\cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.27}$$

for $i = 2$ to M:

$$\begin{aligned} &-r_x V_{i-1,j}^{n+\frac{1}{2}} + (2+2r_x)V_{i,j}^{n+\frac{1}{2}} - r_x V_{i+1,j}^{n+\frac{1}{2}} \\ &= r_y V_{i,j-1}^{n} + (2-2r_y)V_{i,j}^{n} + r_y V_{i,j+1}^{n} + \frac{\Delta_t}{2}\cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.28}$$

$$W_{i,j}^{n+\frac{1}{2}} = W_{i,j}^{n} + \frac{\Delta_t}{4}\cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.29}$$

for $i = M+1$:

$$\begin{aligned} &-2r_x V_{M,j}^{n+\frac{1}{2}} + (2+2r_x)V_{M+1,j}^{n+\frac{1}{2}} \\ &= r_y V_{M+1,j-1}^{n} + (2-2r_y)V_{M+1,j}^{n} + r_y V_{M+1,j+1}^{n} + \frac{\Delta_t}{2}\cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.30}$$

$$W_{M+1,j}^{n+\frac{1}{2}} = W_{M+1,j}^{n} + \frac{\Delta_t}{4}\cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.31}$$

$V^{n+\frac{1}{2}}$ and $w^{n+\frac{1}{2}}$ will now be used in the second step to find V^{n+1} and W^{n+1} on the time interval $[t_{n+\frac{1}{2}}, t_{n+1}]$. The difference equations that enable this evaluation are:

Step 2:

for $j = 1$:

$$\begin{aligned} &(2+2r_y)V_{i,1}^{n+1} - 2r_y V_{i,2}^{n+1} \\ &= r_x V_{i-1,1}^{n+\frac{1}{2}} + (2-2r_x)V_{i,1}^{n+\frac{1}{2}} + r_x V_{i+1,1}^{n+\frac{1}{2}} + \frac{\Delta_t}{2}\cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.32}$$

$$W_{i,1}^{n+1} = W_{i,1}^{n+\frac{1}{2}} + \frac{\Delta_t}{4}\cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.33}$$

for $j = 2$ to M

$$\begin{aligned} &-r_y V_{i-1,j}^{n+1} + (2+2r_y)V_{i,j}^{n+1} - r_y V_{i+1,j}^{n+1} \\ &= r_x V_{i,j-1}^{n+\frac{1}{2}} + (2-2r_x)V_{i,j}^{n+\frac{1}{2}} + r_x V_{i,j+1}^{n+\frac{1}{2}} + \frac{\Delta_t}{2}\cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.34}$$

$$W_{i,j}^{n+1} = W_{i,j}^{n+\frac{1}{2}} + \frac{\Delta_t}{4}\cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.35}$$

for $j = M + 1$

$$\begin{aligned} &-2r_y V_{i,M}^{n+1} + (2 + 2r_y)V_{i,M+1}^{n+1} \\ &= r_x V_{i-1,M+1}^{n+\frac{1}{2}} + (2 - 2r_x)V_{i,M+1}^{n+\frac{1}{2}} + r_x V_{i+1,M+1}^{n+\frac{1}{2}} + \frac{\Delta_t}{2} \cdot \tilde{f}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \end{aligned} \tag{3.36}$$

$$W_{i,M+1}^{n+1} = W_{i,M+1}^{n+\frac{1}{2}} + \frac{\Delta_t}{4} \cdot \tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) \tag{3.37}$$

Where:

$$\tilde{f}\left(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}\right) =$$

$$-k\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2}\right)\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2} - \alpha\right)\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2} - 1\right) - \tag{3.38}$$

$$\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2}\right)\left(\frac{3W_{i,j}^n - W_{i,j}^{n-1}}{2}\right)$$

and

$$\tilde{g}(V^{n+\frac{1}{2}}, W^{n+\frac{1}{2}}) =$$

$$\left(\epsilon + \frac{\mu_1 \frac{3W_{i,j}^n - W_{i,j}^{n-1}}{2}}{\mu_2 + \frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2}}\right) \cdot \tag{3.39}$$

$$\left(-\frac{3W_{i,j}^n - W_{i,j}^{n-1}}{2} - k\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2}\right)\left(\frac{3V_{i,j}^n - V_{i,j}^{n-1}}{2} - \alpha - 1\right)\right)$$

The next chapter presents the numerical simulations illustrating the presented method.

Chapter 4: Numerical Results

In this chapter, we will show the performance of the ADI method for solving the Aliev-Panfilov monodomain model. We will first use an analytical solution to verify the order of time and space. Then, we will show the efficiency of the ADI method in comparison to the Crank-Nicolson Admas-Bashforth ($CNAB$) scheme. The comparison is done in terms of run-time and memory consumption.

4.1 Analytical Solution

In this section, we present numerical tests for a two dimensional heat equation problem. We present a problem for which we are able to determine the exact solution in order to compute the error for the time and space orders. Therefore, the following problem is considered:

$$\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2} - \frac{\partial^2 U}{\partial y^2} = f(x, y, t) \qquad x \in [0, 1], \quad y \in [0, 1], \quad t \in [0, 1] \tag{4.1}$$

with initial condition

$$U(x, y, 0) = e^{-x^2-y^2} \quad (x, y) \in [0, 1] \times [0, 1] \tag{4.2}$$

The exact solution is given by:

$$U(x, y) = e^{-x^2-y^2-t^2} \tag{4.3}$$

$f(x, y, t)$ and the boundary conditions are adjusted based on the analytical solution.

To verify the order of the method, p, we use the formula:

$$p \approx \frac{\ln |E(2h)| - \ln |E(h)|}{\ln(2)} \tag{4.4}$$

where $E(h)$ is the infinity error in the approximate solution obtained when we solve the above problem with step size h.

Tables 4.1 and 4.2 shows the details of the simulation. As can be seen, the

value of p is approximately 2 as the step size is doubled which verifies the expected second order of the method both in space and time.

Table 4.1: Results of the analytical solution solved with ADI method showing the second order in time

Space steps M	**Time steps** N	$E(h)$	p
1000	10	0.0044	
1000	20	0.0011	2.00
1000	40	$2.695 \times 10^{-}4$	2.00
1000	80	$6.759 \times 10^{-}5$	2.00
1000	160	$1.713 \times 10^{-}5$	2.00

Table 4.2: Results of the analytical solution solved with ADI method showing the second order in space

Time steps N	**Space steps** M	$E(h)$	p
1000	10	0.0032	
1000	20	$7.983 \times 10^{-}4$	2.01
1000	40	$1.997 \times 10^{-}4$	1.9
1000	80	$5.023 \times 10^{-}5$	1.9
1000	160	$1.288 \times 10^{-}5$	1.9

4.2 Numerical Results for the Aliev-Panfilov Monodomain Model

To verify the efficiency of the ADI method we prosed, we refer to the study and comparison done by Ethier and Bourgault in [10]. The authors presented a study and comparison of different numerical schemes for solving the cardiac model. These schemes were first order methods such as forward-backward Euler, Crank-Nicolson-forward Euler, IMEX first order Gear, and backward Euler, second order methods such as Crank-Nicolson/Adams-Basforth ($CNAB$), modified Crank-Nicolson/Adams-Basforth, second-order semi-implicit backward differentiation ($SBDF$), and implicit gear, and third order methods such $SBDF$.

The results published in this paper verified that the Crank-Nicolson/Adams-Basforth ($CNAB$) is the best scheme available to solve the cardiac model. To show the performance of ADI method in this thesis, our focus is on verifying the improvement in the memory consumption and the run-time over the Crank Nicolson Adams Bashforth ($CNAB$) scheme when both methods are applied to the Aliev-Panfilov monodomain model.

We will first briefly illustrate the $CNAB$ scheme implemented for comparison reasons.

4.2.1. The *CNAB* scheme. $CNAB$ was developed by taking the average of the simple explicit update and the simple implicit update on the right-hand side of the heat equation (see [43]).

The discretized equation of the Aliev-Panfilov monodomain model takes the form:

$$\begin{cases} \dfrac{V_{i,j}^{n+1}-V_{i,j}^{n}}{\Delta_t} = \dfrac{V_{i-1,j}^{n+1}-2V_{i,j}^{n+1}+V_{i+1,j}^{n+1}}{2\Delta_x^2}+ \\ \qquad \dfrac{V_{i,j-1}^{n+1}-2V_{i,j}^{n+1}+V_{i,j+1}^{n+1}}{2\Delta_y^2}+ \\ \qquad \dfrac{V_{i-1,j}^{n}-2V_{i,j}^{n}+V_{i+1,j}^{n}}{2\Delta_x^2}+ \\ \qquad \dfrac{V_{i,j-1}^{n}-2V_{i,j}^{n}+V_{i,j+1}^{n}}{2\Delta_y^2}+ \\ \qquad \dfrac{3f(V_{i,j}^{n},W_{i,j}^{n})}{2}-\dfrac{f(V_{i,j}^{n-1},W_{i,j}^{n-1})}{2} \\ \dfrac{W_{i,j}^{n+1}-W_{i,j}^{n}}{\Delta_t} = \dfrac{3g(V_{i,j}^{n},W_{i,j}^{n})}{2}-\dfrac{g(V_{i,j}^{n-1},W_{i,j}^{n-1})}{2} \end{cases} \tag{4.5}$$

The above equation can be written as: for $i = 2$ to M and $j = 2$ to M;

$$\begin{cases} -r_x V_{i-1,j}^{n+1} - r_y V_{i,j-1}^{n+1} + 2(1 + r_x + r_y)V_{i,j}^{n+1} - r_x V_{i+1,j}^{n+1} - r_y V_{i,j+1}^{n+1} = \\ r_x V_{i-1,j}^{n} + r_y V_{i,j-1}^{n} + 2(1 - r_x - r_y)V_{i,j}^{n} + r_x V_{i+1,j}^{n} + r_y V_{i,j+1}^{n} + \\ 3\Delta_t f(V_{i,j}^n, W_{i,j}^n) - \Delta_t f(V_{i,j}^{n-1}, W_{i,j}^{n-1}) \\ \\ W_{i,j}^{n+1} = W_{i,j}^n + \frac{3}{2}\left(\Delta_t g(V^n, W^n) - \Delta_t g(V^{n-1}, W^{n-1})\right) \end{cases} \tag{4.6}$$

where $r_x = \dfrac{\Delta_t}{\Delta_x^2}$, $r_y = \dfrac{\Delta_t}{\Delta_y^2}$, and

$$\begin{cases} f(V_{i,j}^n, W_{i,j}^n) = -k(V_{i,j}^n)(V_{i,j}^n - \alpha)(V_{i,j}^n - 1) - (V_{i,j}^n)(W_{i,j}^n) \\ \\ g(V_{i,j}^n, W_{i,j}^n) = \left(\epsilon + \dfrac{\mu_1 W_{i,j}^n}{\mu_2 + V_{i,j}^n}\right)(-W_{i,j}^n - k(V_{i,j}^n)(V_{i,j}^n - \alpha - 1)) \end{cases} \tag{4.7}$$

The method is proven to be unconditionally stable with a truncation error of $O[(\Delta_t)^2, (\Delta_x)^2, (\Delta_y)^2]$ (for more details the reader is referred to [44]).

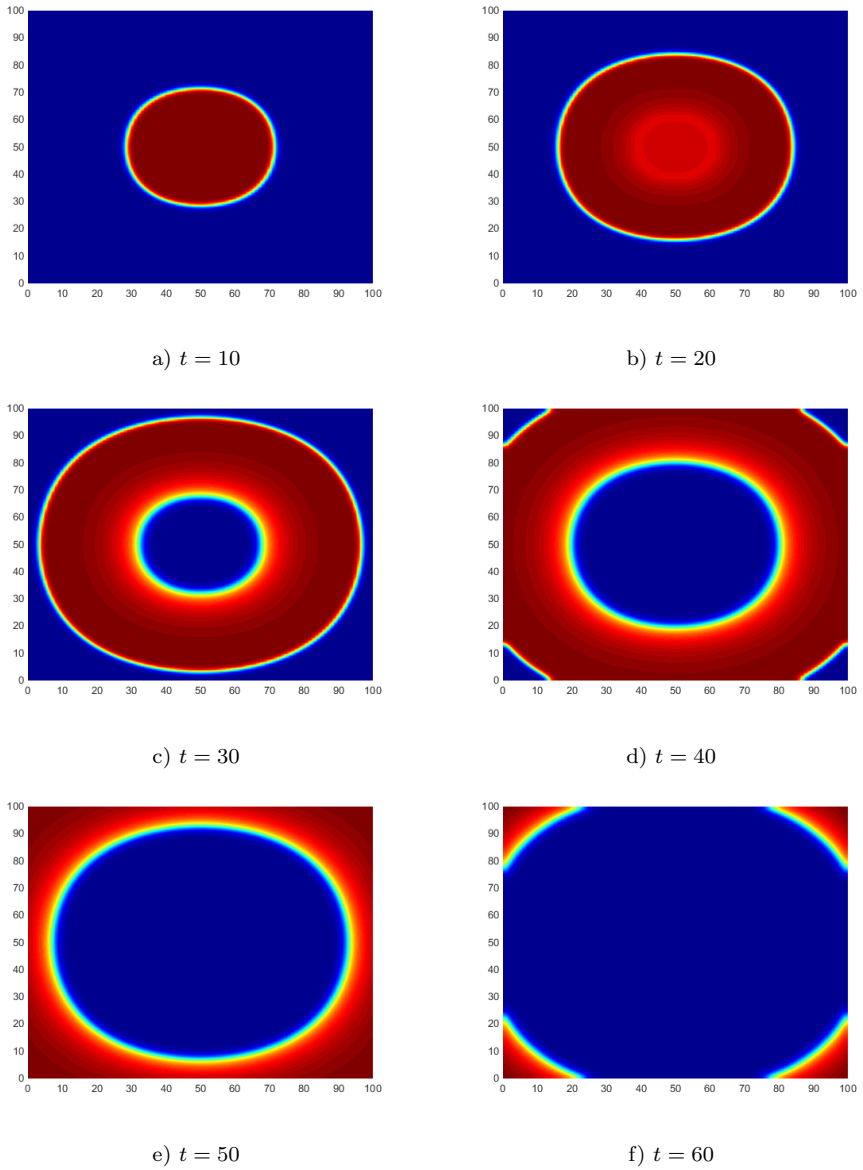
4.2.2. *ADI* v.s. *CNAB*. This section is devoted to presenting the results of *ADI* and *CNAB* methods upon implementing them on the Aliev-Panfilov monodomain model. Both *ADI* and *CNAB* methods were executed using MATLAB language, and were run using the same computer with Intel(R) Core(TM) i7 − 4790 CPU processor, and 8.00 GB installed memory functioning with a 64-bit operating system. This guaranteed a fair comparison between the two methods in terms of run-time and memory consumption.

For the Aliev-Panfilov model the physical parameters used are fixed using values taken from the literature (see [40]) and are given by table 4.3.

Table 4.3: Parameter values for the Aliev-Panfilov Monodomain model

parameter	k	α	ϵ	μ_1	μ_2	D
value	8	0.15	0.002	0.2	0.3	1

Figure 4.1 shows the evolution of transmembrane potential over time for the two dimensional case. The solution is generated with $dt = 0.1$, $dx = 1$, and $dy = 1$. As can be seen, the wave is initiated as a spherical wave and propagates in a similar manner till it reaches the boundaries, an expected behaviour of this model.



a) $t = 10$ b) $t = 20$

c) $t = 30$ d) $t = 40$

e) $t = 50$ f) $t = 60$

Figure 4.1: Evolution of transmembrane potential over time for the 2D Aliev-Panfilov monodomain model.

To illustrate the efficiency of the *ADI* scheme applied to the Aliev-Panfilov monodomain model, a comparison versus *CNAB* scheme is presented in terms of memory and run time. Both methods are tested using the same number of time

iterations to obtain the same numerical solution.

First, figure 4.2 shows the comparison between the memory consumption of the *CNAB* and *ADI* methods.

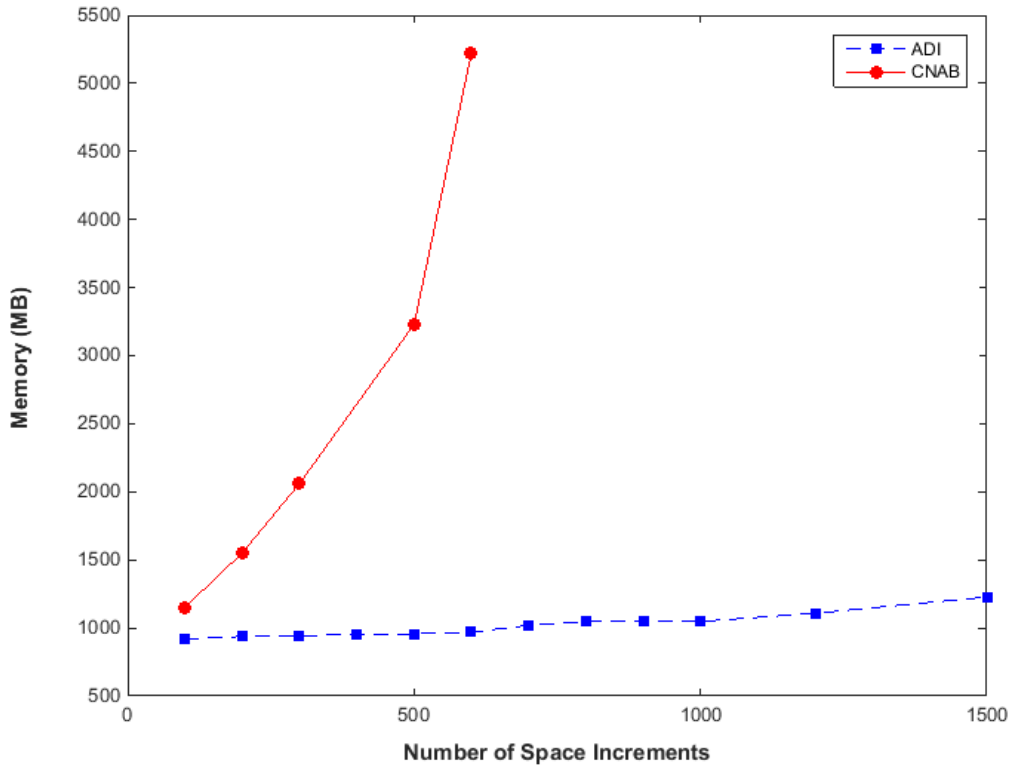


Figure 4.2: Memory Consumption of the *ADI* and *CNAB* Methods

As can be seen in this figure, the memory consumption of the *CNAB* increases dramatically. This is justified by the relation that ties the size of the matrix used for storing the coefficients and the number of space steps, M. The *CNAB* method consumed huge memory. With such consumption of memory, the simulation of the Aliev-Panfilov monodomain model at large step sizes will be limited depending on the memory available for MATLAB on a given computer, a point the *ADI* scheme easily overcomes. The *ADI* keeps running even with 1000 space steps, and while doing so, it is noticed that the memory consumption remains less than what is consumed by *CNAB* using only 100 space steps. The fact that *ADI* scheme remains smoothly running even with high number of space steps while avoiding massive memory consumptions solidifies the expected benefits of

applying *ADI* to Aliev-Panfilov monodomain model.

Second, the comparison between the run time of the *CNAB* scheme and the *ADI* methods is displayed in figure 4.3. The time required to run *CNAB* with

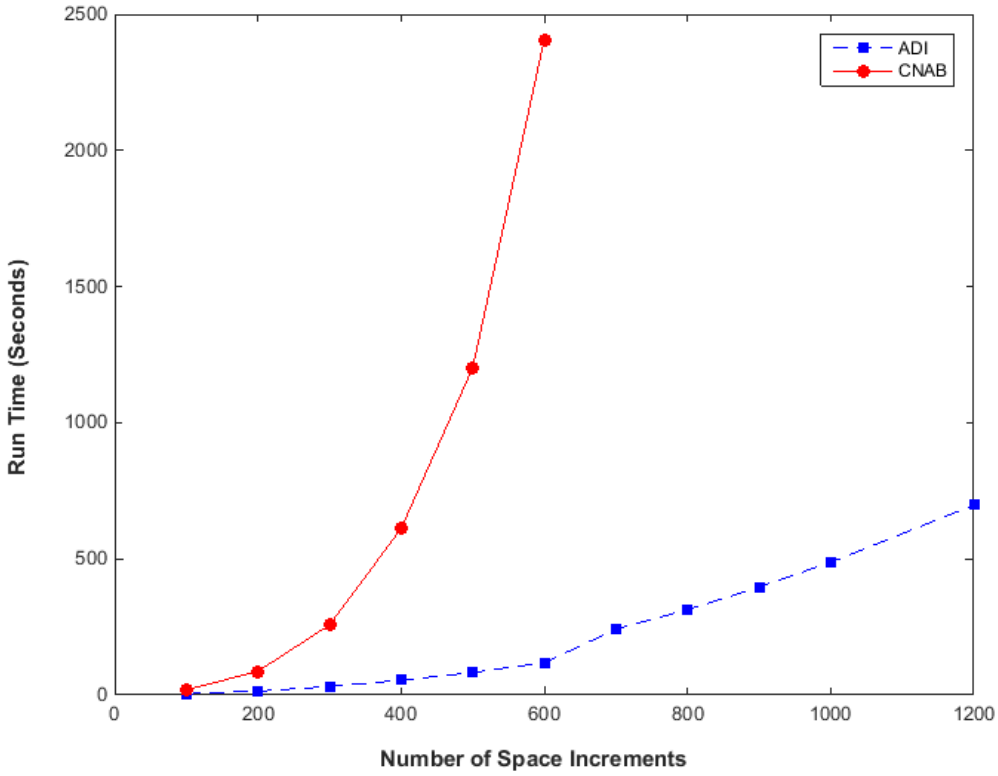


Figure 4.3: Run Time of the *ADI* and *CNAB* Methods

400 space steps is close to that needed by the *ADI* to run with 1200 space steps. A slight increase in the number of space steps, M, used causes a rapid increase in the run time of *CNAB*, whereas the increase in the run-time in *ADI* is relatively slight when increasing the space step.

The slow run-time of the *CNAB* is again linked to the size of the matrix used to store the coefficients. The *CNAB* uses a matrix of size $(M+1)(M+1)$, whereas that of the *ADI* is $(M+1)$. The reduction in the matrix size of the *ADI* scheme fastened run-time. The rapid results provided by the *ADI* is a powerful advantage for a model that aims at capturing the behaviour of the cardiac electrical signal, a signal that propagates for only fractions of a second.

Chapter 5: Conclusions and Future Work

Electrical activity is responsible for the heart functioning. Many heart problems result from irregularities in the cardiac electrical signal. To enable the study of these conditions non-invasively and consequently generate appropriate treatments, is what triggered mathematical modelling of the electrical activity in cardiac tissue.

The bidomain model is one of these mathematical models that targets capturing the electrical flow in a cardiac tissue. A briefing of the numerical schemes available in the literature that were used to solve this model was presented in the introduction. The review enabled us to notice the concerns related to the memory consumption and run-time required by these schemes. Our suggestion was to implement the Alternating Direction Implicit method to partially overcome these issues.

In chapter 2, the formulation of the bidomain model was discussed along with the simplification that led to the monodomain model. We restricted our study to the monodomain model coupled with Aliev-Panfilov Ionic model.

The detailed scheme of the *ADI* method used was introduced in chapter 3. We discussed the implementation of this method for both Dirichlet boundary conditions and Neumann boundary conditions. The algorithm was first discussed in terms of the heat equation and then generalized to the Aliev-Panfilov monodomain model.

A numerical verification of the time and space order of the *ADI* method through an analytical solution was presented in chapter 4. Moreover, a validation of the efficiency of the *ADI* algorithm applied to the Aliev-Panfilov Monodomain model was provided though demonstrating the difference in the run-time and memory requirements between this method and *CNAB*. The numerical simulations has shown that the *ADI* algorithm not only speeds up the run-time over the *CNAB* method, but also reduces memory usage and overcomes running out of memory that finite difference methods such as *CNAB* result in.

Our work was done using a basic *ADI* method but still requires more

stability analysis to achieve methods with better stability properties such as the Douglas-Gunn ADI method. Space order can be another area where this work can progress due to the simplicity the ADI methods provide in advancing to higher orders in space.

Bibliography

[1] L. Tung, "A Bidomain Model for Describing Ischemic Myocardial D-C Potentials", PhD Thesis, Massachusetts Institute of Technology, 1978.

[2] S. Kandel, "The Electrical Bidomain Model: A Review", *Scholars Academic Journal of Biosciences*, vol. 3, pp. 633-639, 2015.

[3] S. Puwal and B. Roth, "Forward Euler Stability of the Bidomain Model of Cardiac Tissue", *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 951-955, 2007.

[4] M. Murillo and X. Cai, "A Fully Implicit Parallel Algorithm for Simulating the Non Linear Electrical Activity of the Heart", *Numerical Linear Algebra with Applications*, vol. 11, pp. 261-277, 2004.

[5] W. Ying, D. Rose and C. Henriquez, "Efficient Fully Implicit Time Integration Methods for Modeling Cardiac Dynamics", *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 2701-2711, 2008.

[6] H. Dal, S. Goktepe, M. Kaliske, and E. Kuhl, "A Fully Implicit Finite Element Method for Bidomain Models of Cardiac Electromechanics", *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 323-336, 2013.

[7] Y. Belhamadia, "An Efficient Computational Method for Simulation of the Two Dimensional Electrophysiological Waves", in *Proc. IEEE Engineering in Medice and Biology Society*, pp. 5922-5928, 2008.

[8] J. Keener and K. Bogar, "A Numerical Method for the Solution of the Bidomain Equations in Cardiac Tissue", *Chaos*, vol. 1, pp. 234-241, 1998.

[9] P. Franzone and L. Pavarino, "A Parallel Solver for Reaction Diffusion Systems in Computational Electrocardiology", *Mathematical Models and Methods in Applied Sciences*, vol. 14, pp. 883-911, 2004.

[10] M. Ethier and Y. Bourgault, "Semi-Implicit Time-Discretization Schemes for the Bidomain Model", *Journal on Numerical Analysis*, vol. 46, pp. 2443-2468, 2008.

[11] J. Trangenstein and C. Kim, "Operator Splitting and Adaptive Mesh Refinement for the LuoRudy I Model", *Journal of Computational Physics*, vol. 196, pp. 645-679, 2004.

[12] H. Schroll, G. Lines and A. Tveito, "On the Accuracy of Operator Splitting for the Monodomain Model of Electrophysiology", *International Journal of Computer Mathematics*, vol. 84, pp. 871-885, 2007.

[13] R. Spiteri and S. Ziaratgahi, "Operator Splitting for the Bidomain Model Revisited", *Journal of Computational and Applied Mathematics*, vol. 296, pp. 550-563, 2016.

[14] J. Sundnes, G. Lines, and A. Tveito, "An Operator Splitting Method for Solving the Bidomain Equations Coupled to a Volume Conductor Model for the Torso", *Mathematical Biosciences*, vol. 194, pp. 233-248, 2005.

[15] S. Linge, J. Sundnes, M. Hanslien, G. Lines, and A. Tveito, "Numerical Solution of the Bidomain Equations", *Philosophical Transcations of the Royal Society*, vol. 367, pp. 1931-1950, 2009.

[16] P. Franzone, L. Pavarino and S. Scacchi, "A Comparison of Coupled and Uncoupled Solvers for the Cardiac Bidomain Model", *Mathematical Modelling and Numerical Analysis*, vol. 47, pp. 1017-1035, 2013.

[17] P. Pathmanathan, M. Bernabeu, R. Bordas, J. Cooper, A. Garny, J. Pitt-Francis, J. Whiteley, and D. Gavaghan, "A Numerical Guide to the Solution of the Bidomain Equations of Cardiac Electrophysiology", *Progress in Biophysics and Molecular Biology*, vol. 102, pp. 136-155, 2010.

[18] R. Bordas, B. Carpentieri, G. Fotia, F. Maggio, R. Nobes, J. Pitt-Francis and J. Southern, "Simulation of Cardiac Electrophysiology on Next-Generation

High-Performance Computers", *Philosophical Transcations of the Royal Society*, vol. 367, pp. 1951-1969, 2009.

[19] S. Scacchi, "A Multilevel Hybrid NewtonKrylovSchwarz Method for the Bidomain Model of Electrocardiology", *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 717-725, 2011.

[20] J. Whiteley, "Physiology Driven Adaptivity for the Numerical Solution of the Bidomain Equations", *Annals of Biomedical Engineering*, vol. 35, pp. 1510-1520, 2007.

[21] E. Cherry, H. Greenside, and C. Henriquez, "Efficient Simulation of Three Dimensional Anisotropic Cardiac Tissue using an Adaptive Mesh Refinement Method", *Chaos*, vol. 13, pp. 853-865, 2003.

[22] M. Berher and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics", *Journal of Computational Physics*, vol. 82, pp. 64-89, 1989.

[23] Y. Belhamadia, "A Time-Dependent Adaptive Remeshing for Electrical Waves of the Heart", *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 443-452, 2008.

[24] Y. Belhamadia, A. Fortin, and Y. Bourgault, "Towards Accurate Numerical Method for Monodomain Equations Using a Realistic Heart Geometry", *Mathematical Biosciences*, vol. 220, pp. 89-101, 2009.

[25] Y. Belhamadia, A. Fortin, and Y. Bourgault, "On the Performance of Anisotropic Mesh Adaptation for Spiral and Scroll Wave Turbulence Dynamics in Reaction-Diffusion Systems", *Journal of Computational and Applied Mathematics*, vol. 271, pp. 233-246, 2014.

[26] E. Heidenreich, F. Gaspar, J. Ferrero and J. Rodriguez, "Compact Schemes for Anisotropic ReactionDiffusion Equations with Adaptive Time Step", *International Journal for Numerical Methods in Engineering*, vol. 82, pp. 1022-1043, 2009.

[27] D. Peaceman and H. Rachford, "The Numerical Solution of Parabolic and Elliptic Differential Equations", *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, pp. 28-41, 1955.

[28] J. Douglas, "On the Numerical Integration of $u_{xx} + u_{yy} = u_t$ by Implicit Methods", *Journal of the Society of Industrial and Applied Mathematics*, vol. 3, pp. 42-65, 1955.

[29] B. Thomas, I. Samarasekera, and J. Brimacombe, "Comparison of Numerical Modeling Techniques for Complex, Two Dimensional, Transient heat-Conduction Problems", *Metallurgical Transactions B*, vol. 15, pp. 307-318, 1984.

[30] A. Abimbola and S. Bright, "Alternating-Direction Implicit Finite-Difference Method for Transient 2D Heat Transfer in a Metal Bar using Finite Difference Method", *International Journal of Scientific and Engineering Research*, vol. 6, pp. 105-108, 2015.

[31] A. Araujo, C. Neves, and E. Sousa, "An Alternating Direction Implicit Method for a Second-Order Hyperbolic Diffusion Equation with Convection", *Applied Mathematics and Computation*, vol. 239, pp. 17-28, 2014.

[32] T. Wang and C. Chen, "Thermal-ADIA Linear-Time Chip-Level Dynamic Thermal-Simulation Algorithm Based on Alternating-Direction-Implicit (ADI) Method", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, pp. 691-700, 2003.

[33] S. Karaa and J. Zhang, "High Order ADI Method for Solving Unsteady ConvectionDiffusion Problems", *Journal of Computational Physics*, vol. 198, pp. 1-9, 2004.

[34] R. Fernandesa, B. Bialeckib, and G. Fairweatherc, "An ADI Extrapolated CrankNicolson Orthogonal Spline Collocation Method for Nonlinear ReactionDiffusion Systems on Evolving Domains", *Journal of Computational Physics*, vol. 299, pp. 561-580, 2015.

[35] Z. Gao and S. Xie, "Fourth-Order Alternating Direction Implicit Compact Finite Difference Schemes for Two Dimensional Schrodinger equations", *Applied Numerical Mathematics*, vol. 61, pp. 593-614, 2011.

[36] Y. Xu and L. Zhang, "Alternating Direction Implicit Method for Solving Two Dimensional Cubic Nonlinear Schrodinger Equation", *Computer Physics Communications*, vol. 183, pp. 1082-1093, 2012.

[37] A. Hodgkin and A. Huxley, "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve", *The Journal of physiology*, vol. 117, pp. 500-544, 1952.

[38] R. FitzHugh, "Impulses and Physiological States in Theoretical Models of Nerve Membrane", *Biophysical Journal*, vol. 1, pp. 445-466, 1961.

[39] C. Luo and Y. Rudy, "A Model of the Ventricular Cardiac Action Potential Depolarization, Repolarization, and their Interaction", *Journal of the American Heart Association*, vol. 68, pp. 1501-1526, 1991.

[40] R. Aliev and A. Panfilov, "A Simple Two-variable Model of Cardiac Excitation", *Chaos, Solirons and Fractals*, vol. 7, pp. 293-301, 1996.

[41] J. Sundnes, G. Lines, X. Cai, B. Nielsen, K. Mardal, and A. Tveito, "Computing the Electrical Activity in the Heart", *Monographs in Computational Science and Engineering*, vol. 1, 2006.

[42] J. Sundnes, B. Nielsen, K. Mardal, X. Cai, G. Lines, and A. Tveito, "On the Computational Complexity of the Bidomain and the Monodomain Models of Electrophysiology", *Annal of Biomedical Engineering*, vol. 24, pp. 1088-1097, 2006.

[43] J. Crank and E. Nicolson, "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type", *Advances in Computational Mathematics*, vol. 6, pp. 207-226, 1996.

[44] N. Sweilam, M. Khader, and S. Mahdy, "Crank-Nicolson Finite Difference Method for Solving Time-Fractional Diffusion Equation", *Journal of Fractional Calculus and Applications*, vol. 2, pp. 1-9, 2012.

Vita

Zeinab Rammal was born on October 13, 1988, in Odeisse, Lebanon. She was educated in private secondary schools till she earned the Lebanese Baccalaureate in General Sciences in 2006. After that she joined the Lebanese International University, and graduated with honor in 2010 with a Bachelors Degree of Science in Mathematics.

Mrs. Zeinab got married and moved with her husband to Dubai, United Arab Emirates in 2010. She started her career path as a high school math teacher for almost seven years, teaching in different international American schools. In 2015, Mrs. Zeinab enrolled in the masters program of Applied Mathematics at the American University of Sharjah where she also worked for the Department of Mathematics and Statistics as a graduate teaching assistant.