Fast simulations of models of cardiac electrophysiology

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Yoseph Tereda

Permission to Use

Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

Cardiac electrophysiology studies the electrical activity of the heart. Researchers from different fields are working together to model and simulate the electrical activity in the heart. Such simulations may lead to help cardiologists to treat a patient's heart condition with better techniques and diagnoses.

Mathematical models of cardiac electrophysiology are described by a system of partial differential equations (PDEs) and a non-linear system of ordinary differential equations (ODEs). One way to reach real-time simulation of the electrical activity of the heart is through more efficient time integration of these ODEs. Larger time steps lead to more efficient computations provided the solutions remain accurate enough. Producing the largest stable time step for solving these ODEs is a daunting task. Usually, trial and error is used to get the largest stable time step, but it is time consuming. In this thesis, we propose a new efficient method to find the largest stable time step to solve these ODEs efficiently.

In this thesis, we present thirty-seven cardiac cell models. We compare, the forward Euler method, the explicit midpoint method, the four-stage, fourth-order Runge–Kutta method, the two-stage, first-order Runge–Kutta–Chebyshev method, the three-stage, first-order Runge–Kutta–Chebyshev method, the three-stage, third-order strong-stability-preserving Runge–Kutta method, and the four-stage, third-order strong-stability-preserving Runge–Kutta method based on their largest stepsize, mixed root mean square errors, and CPU time to solve the cardiac cell models. From the theoretical largest stepsize results, the forward Euler method outperforms all the other methods considered on all thirty-seven cardiac cell models. Next to the forward Euler method, the two-stage, first-order Runge–Kutta–Chebyshev method outperforms all the other methods considered on thirty-seven cardiac cell models.

From the experimental largest stepsize results, the forward Euler method outperforms all the other methods considered on all thirty-seven cardiac cell models. Next to the forward Euler method, the two-stage, first-order Runge–Kutta–Chebyshev and the explicit midpoint methods outperform all the other methods considered on thirty-six cardiac cell models and one cardiac cell model, respectively.

Also, we solve the monodomain problem coupled with FitzHugh–Nagumo model using the forward Euler, the two-stage, first-order Runge–Kutta–Chebyshev method, and the three-stage, first-order Runge–Kutta–Chebyshev method in one, two, and three dimensions. We compare these methods based on their theoretical largest stepsize, experimental largest stepsize, CPU time, and mixed root mean square errors. From these results, the three-stage, first-order Runge–Kutta–Chebyshev method outperforms all the other methods over a one-dimensional problem, a two-dimensional problem, and a three-dimensional problem.

Acknowledgements

First and foremost, I praise God, the almighty, for the opportunity and for granting me the capability to proceed successfully. Thank my supervisor, Dr. Raymond Spiteri, for his help and advice with this MSc. Thank my thesis committee, Dr. Tom Steele, Dr. Mark Keil, and Dr. Lingling Jin, for their insightful comments and suggestions. I want to thank Dr. Martha Smith and Dr. Ryan Walker support helped me overcome many crises and finish this thesis. Finally, I would like to express my sincere gratitude to my parent. Without their tremendous understanding and encouragement in the past few years, it would be impossible to complete my study.

This thesis is dedicated to my parents for their love and measureless support.

Contents

List of Tables

List of Figures

List of Abbreviations

FE	Forward Euler
BE	Backward Euler
IVP	Initial-Value Problem
MRMS	Mixed Root Mean Square
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
RK	Runge–Kutta
RKC	Runge–Kutta–Chebyshev
RKC$(s, 1)$	s-stage, first-order Runge–Kutta–Chebyshev
EMP	Explicit Midpoint
SSP$(s, 3)$	s-stage, third-order Strong-Stability-Preserving Runge–Kutta
RK4	Four-stage, fourth-order Runge–Kutta
$\Delta t_{\max}$	Largest stable time step
$\lambda_{\min}$	The minimum eigenvalue

1 Introduction

Every year people die from cardiovascular disease. In 2008, an estimated 30% of all deaths worldwide were due to cardiovascular disease [1, 35]. Currently, due to the coronavirus (COVID-19) pandemic, thousands of people are dying worldwide. This virus can infect the heart muscle; see Figure 1.1. The death rate for COVID-19 patients with pre-existing heart disease is far higher than among other patients [34]. As a result of COVID-19, the worldwide mortality rate associated with cardiovascular diseases will likely increase from the percentage mentioned above.

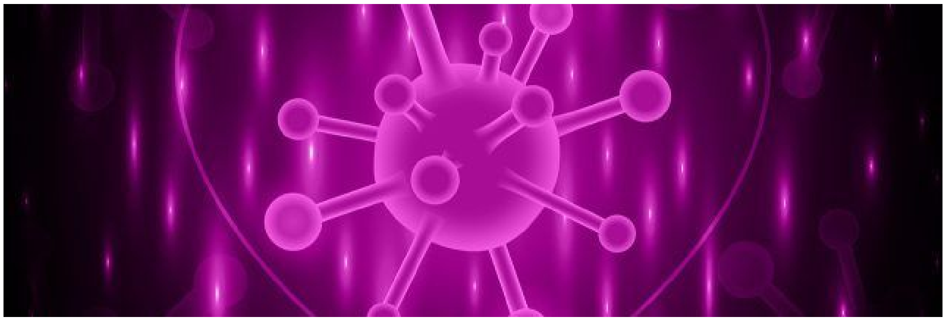


Figure 1.1: Illustration of a neon purple coronavirus cell in the center of a heart outline [4].

These days, we can design the shape of planes and automobiles using finite element simulations; see Figure 1.2. These simulations predict how the planes and automobiles behave in wind and thunder instead of putting them in the wind and thunder. These results are the motivation for the researchers from computer science, engineering, mathematics, and physiologists to work together to model and simulate the electrical activity in cardiac cells. This computer simulation may lead to help the cardiologists to treat a patient's heart condition with better techniques and diagnosis, for example, in the form of more accurate electrocardiogram (ECG) analysis.

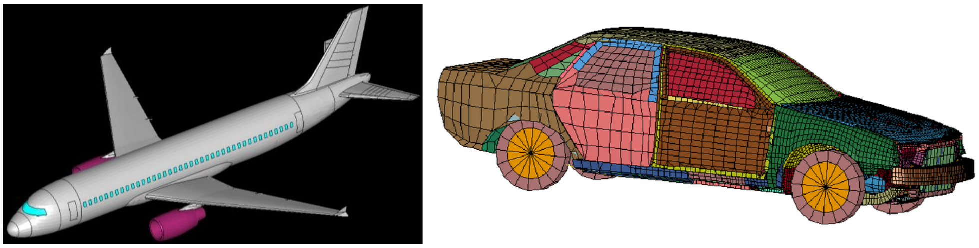


Figure 1.2: Finite element models of airbus and Ford taurus car [56, 58]

Cardiac electrophysiology studies the electrical activity of the heart. Mathematical models and numerical simulations of cardiac electrophysiology are valuable for studying heart development and disorder. These models are described by a system of partial differential equations (PDEs) and a non-linear system of ordinary differential equations (ODEs). These systems of ODEs are used to model the electrical activity in the cardiac cells. This thesis proposes a new efficient method to minimize the time required to perform cardiac simulations.

Over the most recent two decades, the modeling and simulating the electrical activity in the heart have advanced quickly. Currently, high-resolution simulation of the human heart with great detail and accuracy level can now be carried out. However, it is still slower than real-time despite using 16,384 computer nodes and 1,572,864 supercomputer cores [43, 45].

One way to reach the real-time simulation of the electrical activity of the heart is through more efficient time integration. Usually, time integration is done with a constant step size, and to maximize the size of the step while maintaining sufficient accuracy, trial and error is used, but it is inefficient and leads to sub-optimal stepsizes. These sub-optimal stepsizes are either too large and lead to a divergence of the time integration, or they are too small and lead to solutions that are too accurate, and this leads to long computational times. To overcome these problems, we propose a new efficient method to find the largest stepsize that will yield a desired accuracy.

1.1 Contributions of the thesis

In this thesis, we examine time integration methods for cardiac cell models. These methods are the forward Euler (FE), the explicit midpoint (EMP), the four-stage, fourth-order Runge–Kutta (RK4), the two-stage, first-order Runge–Kutta–Chebyshev (RKC$(2,1)$), the two-stage, first-order Runge–Kutta–Chebyshev (RKC$(3,1)$), the three-stage, third-order strong-stability-preserving (SSP$(3,3)$), and the four-stage, third-order strong-stability-preserving (SSP$(4,3)$) methods. We propose a new efficient methodology to find the largest constant stepsize of these methods to meet a given tolerance. Also, we solved the monodomain model using the FE method, the RKC$(2,1)$, and the RKC$(3,1)$ based on their theoretical largest stepsize and experimental largest stepsize. We determine the largest stepsize of the time integration method based on its stability region and the eigenvalues computed from the Jacobian matrix of a system of ODEs. This avoids trial and error to find the experimental largest stepsize.

1.2 Structure of the thesis

In chapter 2, we present background on the electrical activity in the heart and the mathematical model of cardiac cell and myocardial tissue. In chapter 3, we present the numerical methods used to solve cardiac cell models and the monodomain model. We also give the procedure to find the largest stable time step of the method to solve the systems of ODEs. In chapter 4, we present a numerical experiment for solving cardiac cell models using the FE, the RKC$(2,1)$, the RKC$(3,1)$, the SSP$(3,3)$, the SSP$(4,3)$, the EMP, and RK4 methods and simulation of the monodomain model using FE, RKC$(2,1)$, and RKC$(3,1)$ methods. In chapter 5, we summarize the conclusions of these works and make suggestions for future work. Appendix A provides complete results of Table 4.2, Table 4.3, and Table 4.4. Appendix B provides reference solutions used for one-dimensional, two-dimensional, and three-dimensional monodomain problem.

2 Background

2.1 Heart

The heart is an important organ because it carries oxygen and nutrients to different parts of the body; see Figure 2.1. For instance, the brain is the central organ of the human nervous system. It gets a supply of oxygen and nutrients when the heart carries out its responsibility. So without the heart, these capacities would fall flat. If the heart fails, the entire body will shut down rapidly.

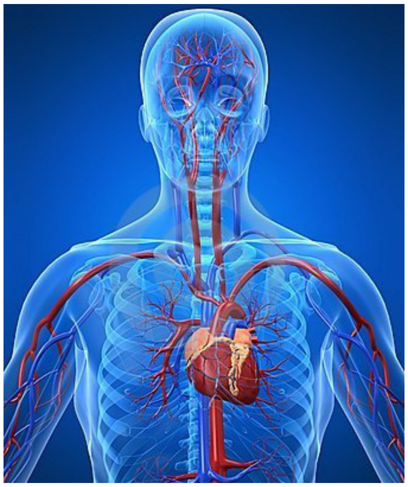


Figure 2.1: Heart [3]

The right and left atria and the right and left ventricles are the chambers of the heart. The electrical heart system is fundamental for the heart to work those activities efficiently [23]. The heart's pumping action is regulated by pacemaker cells that administer regular, spontaneous electrical impulses. These impulses are transmitted to nearby muscle cells, which then contract together as a unit to make the heart pump.

2.2 Cardiac muscle cells

Cardiac muscle is made up of specialized muscle cells, called cardiomyocytes. The cells appear Y-shaped in the Figure 2.2 because they are connected end to end, and some are branched. They are all electrically connected. One of the cells is stimulated, that signal will travel to all of the other cells. There are two types of cells in the cardiac muscle known as cardiomyocytes and cardiac pacemaker cells. The cardiomyocytes are found in the chambers of the heart. They contract and relax to allow blood to enter and exit the heart. Cardiac pacemaker cells are specifically found on the walls of the right atria. They are responsible for initiating action potentials at a regular pace. The five phases of the action potential (see Figure 2.3) in the membrane of cardiac cell are presented below [5]:

1. **Phase 4 (Diastole):** This is known as resting membrane potential.
2. **Phase 0 (Depolarization):** The sodium channels open, and sodium ions ($\mathbf{Na}^+$) enter the cell.
3. **Phase 1 (Early repolarization):** The potassium channels open, and potassium ions ($\mathbf{K}^+$) exit the cell.
4. **Phase 2 (Plateau phase):** Both the potassium channels and calcium channels are open. Potassium ions ($\mathbf{K}^+$) exit the cell at the same time calcium ions ($\mathbf{Ca}^{2+}$) enter the cell. This restores the potential.
5. **Phase 3 (Delayed repolarization):** Calcium channels close slowly while potassium channels remain open. Potassium ions ($\mathbf{K}^+$) continue to move out of the cell, causing repolarization.

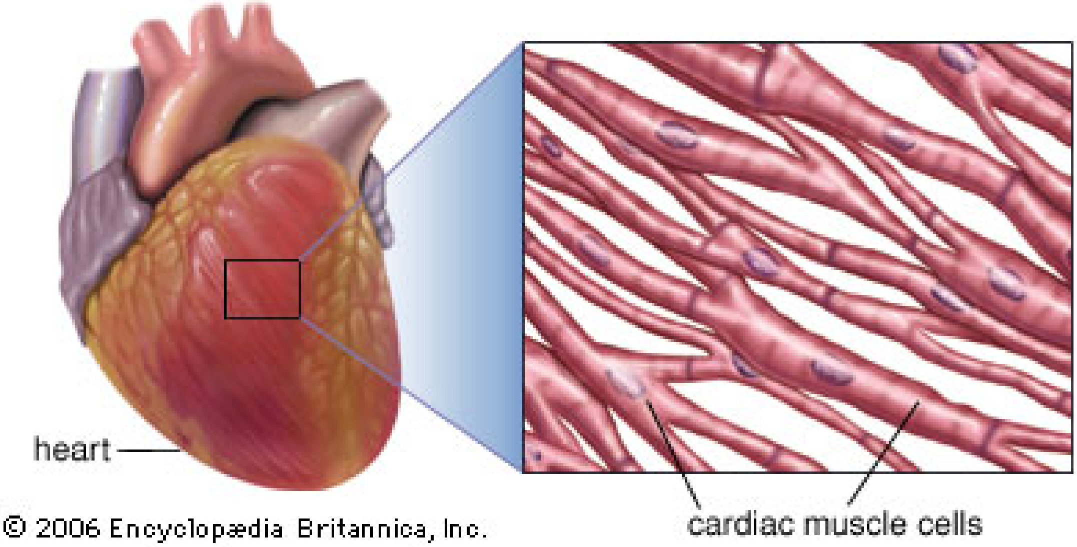


Figure 2.2: Cardiac muscle cell [61].

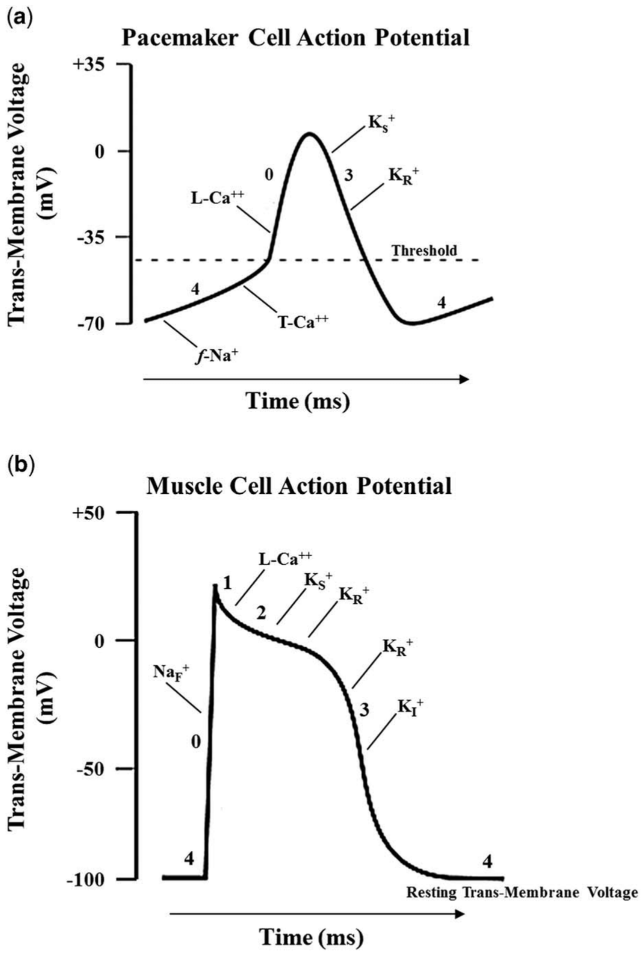


Figure 2.3: The action potential in the membrane of the cardiac cell [59]. Pacemaker cells and muscle cells show different kinds of action potential patterns.

2.2.1 Cell models

In this thesis, we present 37 cardiac cell models of different species, for example, human, pig, and mouse. Each model describe the electrical activity of the cardiac cells. FitzHugh–Nagumo (FHN) [22] is a model of nerve membrane and the simplest model with two ODEs. Bondarenko et al. [11] is the model of mouse ventricular myocyte action potential and the largest model with 41 ODEs.

The Luo-Rudy (1991 model)

Luo and Rudy developed the Luo–Rudy model [36] in 1991. They updated the preexisting guinea pig ventricular action potentials model of Beeler–Reuter (1977) [10]. The Luo–Rudy model (1991) is the most widely used cardiac cell model in research. It makes it possible to simulate cardiac myocytes' excitation-contraction in different conditions; for more details see [37, 38]. Figure 2.4 shows the ionic current flows across the cell membrane of the Luo–Rudy model.

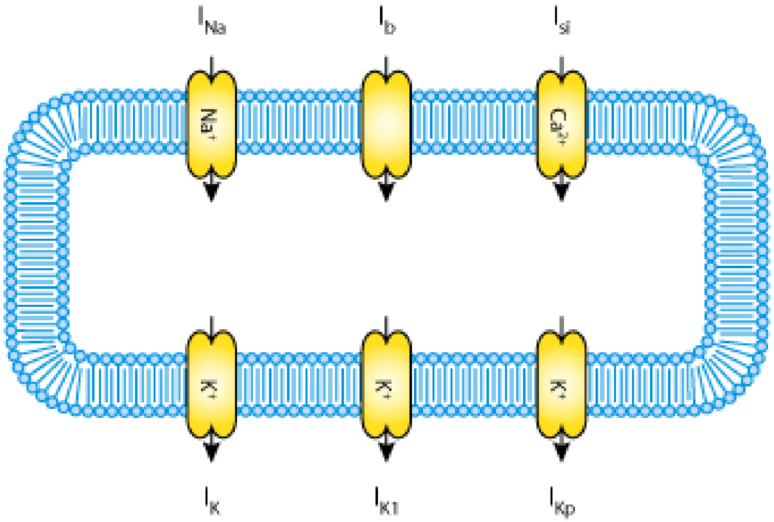


Figure 2.4: The ionic current flows across the cell membrane of Luo-Rudy(1991) model [2].

Winslow et al. (1999) model

Winslow, Rice, Jafri, Marban, and O'Rourke developed the model of Winslow et al. [68] in 1999. They developed a model of canine ventricular tissue. The model of Winslow et al. (1999) are among the more advanced cardiac cell models available; for more details see [68]. Figure 2.5 shows the ionic current flows across the cell membrane of the model of Winslow et al. (1999).

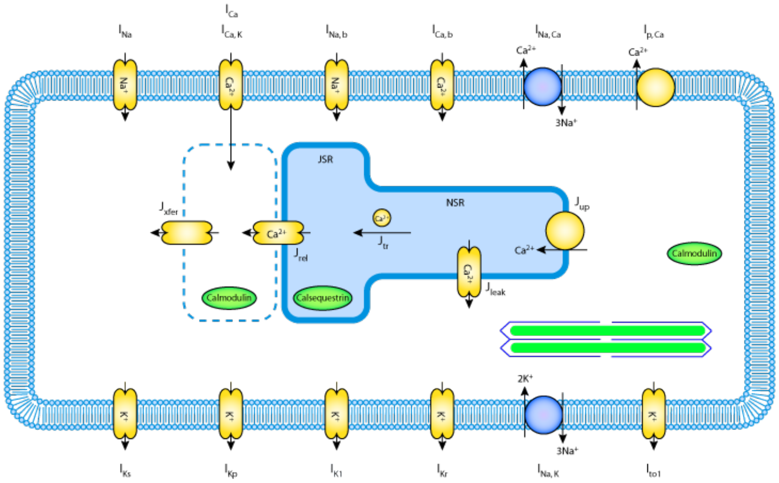


Figure 2.5: The ionic current flows across the cell membrane of the model of Winslow et al. (1999) [2].

2.3 Bidomain model

The bidomain model represents the electrical activity of the whole heart. The bidomain domain consists of two parts: the cardiac cell called the intracellular domain, and the extracellular domain which is the space surrounding the intracellular domain. The cell membrane divides these domains. The bidomain model describes the potential difference across this cell membrane. This model consists of an ODE, a nonlinear PDE, and a linear PDE and is given by [63]

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{2.1}$$

$$\chi C_{me}\frac{\partial v}{\partial t} + \chi \boldsymbol{I}_{ion}(\boldsymbol{s}, v, t) + \boldsymbol{I}_{stim}(\boldsymbol{x}, t) = \nabla \cdot (\boldsymbol{M}_I \nabla v) + \nabla \cdot (\boldsymbol{M}_I \nabla u_E), \tag{2.2}$$

$$0 = \nabla \cdot (\boldsymbol{M}_I \nabla v) + \nabla \cdot ((\boldsymbol{M}_I + \boldsymbol{M}_E) \nabla u_E), \tag{2.3}$$

with boundary conditions on $\partial\Omega$ is given by

$$\begin{aligned} \hat{\boldsymbol{n}} \cdot (\boldsymbol{M}_I \nabla v + \boldsymbol{M}_I \nabla u_E) &= 0, \\ \hat{\boldsymbol{n}} \cdot (\boldsymbol{M}_E \nabla u_E) &= 0, \end{aligned} \tag{2.4}$$

where $\hat{\boldsymbol{n}}$ is the unit outward normal of the domain and $\partial\Omega$ is the domain of the boundary.

Table 2.1: Parameters of the bidomain model.

v	the transmembrane potential
u_E	the extracellular potential
M_I and M_E	conductivity tensors
$\boldsymbol{s}$	a vector of state variables from the cell model
χ	the area of cell membrane per unit volume
C_{me}	the capacitance of the cell membrane per unit area
I_{stim}, $\boldsymbol{f}$, and I_{ion}	dependent on the particular cell model used

2.4 Monodomain model

The bidomain model is challenging for mathematical computation and analysis. We can simplify this model to a monodomain model by assuming $\boldsymbol{M}_I = \lambda \boldsymbol{M}_E$. This monodomain model is given by

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{2.5a}$$

$$\chi C_{me}\frac{\partial v}{\partial t} + \chi \boldsymbol{I}_{ion}(\boldsymbol{s}, v, t) + \boldsymbol{I}_{stim}(\boldsymbol{x}, t) = \frac{\lambda}{1+\lambda} \nabla \cdot (\boldsymbol{M}_I \nabla v), \tag{2.5b}$$

with boundary conditions on $\partial\Omega$ is given by

$$\hat{\boldsymbol{n}} \cdot (\boldsymbol{M}_I \nabla v) = 0,$$

where $\hat{\boldsymbol{n}}$ is the unit outward normal of the domain and $\partial\Omega$ is the domain of the boundary.

It is not easy to find λ that satisfies the assumption mentioned above in real life. This assumption hides critical electrophysiological phenomena because it assumes tissue is isotropic (the conductivities are homogeneous in all directions) [63].

3 Numerical Methods

In this chapter, we present the time integration methods used to solve the ODEs and the finite element method to solve PDEs.

3.1 Explicit Runge–Kutta Methods

Explicit Runge–Kutta (RK) methods are used in the approximation solutions of systems of ODEs [8]. The general form of an s-stage explicit RK method applied to an initial-value problem (IVP),

$$\frac{d\boldsymbol{y}}{dt} = f\left(t, \boldsymbol{y}\right), \quad \boldsymbol{y}\left(t_0\right) = \boldsymbol{y}_0, \quad \text{where} \quad \boldsymbol{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{pmatrix}, \tag{3.1}$$

given as follows:

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{k}_i, \quad \text{where} \quad \boldsymbol{k}_i = f\left(t_n + c_i \Delta t, \boldsymbol{y}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \boldsymbol{k}_j\right), \quad i = 1, 2, ..., s. \tag{3.2}$$

The general form of (3.2) can be represented by its Butcher tableau [30].

$$\begin{array}{c|ccccc} 0 & 0 & 0 & \dots & 0 & 0 \\ c_2 & a_{21} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} & 0 \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

3.1.1 Stability analysis

To illustrate the stability of time integration methods, we use the following test function:

$$\frac{dy}{dt} = \lambda y, \quad y\left(t_0\right) = y_0 \text{ and } \lambda \in \mathbf{C}. \tag{3.3}$$

We implement ERK to (3.3) yields

$$y_n = R(z)^n y_0, \text{ where } R(z) = 1 + z\boldsymbol{b}^{\mathsf{T}}(\mathbf{I} - z\mathbf{A})^{-1}(1, 1, \dots, 1)^{\mathsf{T}} \text{ is a stability function and } \; z = \lambda \Delta t.$$

To have a bounded numerical solution, we must ensure $|R(z)| \leq 1$ and the stability domain of the time integration method given as follow

$$\mathbb{S} := \{z \in \mathbf{C} : |R(z)| \leq 1\}.$$

In this thesis, we are dealing with 37 non-linear systems of cardiac cell models, so we have generalized the stability analysis from equation (3.3) to the non-linear system equation (3.1). The right-hand side of the non-linear systems of equation (3.1) are linearized by using a Taylor expansion about the point $(t^*, \boldsymbol{y}^*)$

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{u} - \boldsymbol{y}^*) + O(\mid \boldsymbol{u} - \boldsymbol{y} \mid^2). \tag{3.4}$$

By ignoring the higher-order terms, the equation (3.4) becomes

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{u} - \boldsymbol{y}^*). \tag{3.5}$$

In the stability analysis of (3.1), we check the difference of two solutions of (3.1) is bounded. Let $\boldsymbol{v}$ and $\boldsymbol{w}$ satisfy (3.5). Then

$$\frac{d\boldsymbol{v}}{dt} = \boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{v} - \boldsymbol{y}^*) \tag{3.6}$$

and

$$\frac{d\boldsymbol{w}}{dt} = \boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{w} - \boldsymbol{y}^*). \tag{3.7}$$

The difference of (3.6) and (3.7) becomes

$$\frac{d\boldsymbol{v}}{dt} - \frac{d\boldsymbol{w}}{dt} = \left[\boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{v} - \boldsymbol{y}^*)\right] - \left[\boldsymbol{f}(t^*, \boldsymbol{y}^*) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{w} - \boldsymbol{y}^*)\right], \tag{3.8}$$

after the cancellation of the term $\boldsymbol{f}(t^*, \boldsymbol{y}^*)$ from right hand side of (3.8) and it becomes

$$\frac{d\boldsymbol{v}}{dt} - \frac{d\boldsymbol{w}}{dt} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)(\boldsymbol{v} - \boldsymbol{w}). \tag{3.9}$$

Let $\tilde{\boldsymbol{y}} = \boldsymbol{v} - \boldsymbol{w}$ and (3.9) becomes

$$\frac{d\tilde{\boldsymbol{y}}}{dt} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)\tilde{\boldsymbol{y}}. \tag{3.10}$$

The partial derivative $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t^*, \boldsymbol{y}^*)$ represent the Jacobian matrix

$$\mathbf{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}} = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial y_1} & \cdots & \frac{\partial f_m}{\partial y_m} \end{pmatrix}, \text{ where } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \text{ and } \boldsymbol{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}.$$

We assume $\mathbf{J}$ is diagonalizable. Therefore, we can write $\mathbf{J}$ as

$$\mathbf{J} = \mathbf{PDP}^{-1}, \tag{3.11}$$

where $\mathbf{D}$ is a diagonal matrix with the eigenvalues of $\mathbf{J}$ and $\mathbf{P}$ is an invertible matrix consisting of the eigenvectors of $\mathbf{D}$. We replaced $\mathbf{J}$ by $\mathbf{PDP}^{-1}$ and (3.10) becomes

$$\frac{d\tilde{\boldsymbol{y}}}{dt} = \mathbf{PDP}^{-1}\tilde{\boldsymbol{y}}. \tag{3.12}$$

Let $\tilde{\tilde{\boldsymbol{y}}} = \mathbf{P}^{-1}\tilde{\boldsymbol{y}}$. Then (3.12) can be rearrange as follows

$$\frac{d(\mathbf{P}\tilde{\tilde{\mathbf{y}}})}{dt} = \mathbf{P}\mathbf{D}\tilde{\tilde{\boldsymbol{y}}}. \tag{3.13}$$

Using $\mathbf{P}^{-1}\mathbf{P} = \mathbf{I}$, then (3.13) becomes

$$\frac{d\tilde{\tilde{\boldsymbol{y}}}}{dt} = \boldsymbol{D}\tilde{\tilde{\boldsymbol{y}}}. \tag{3.14}$$

All of the differential equations in the system (3.14) are fully decoupled. Then we can write each component in the form

$$\frac{d\tilde{\tilde{y}}_i}{dt} = \lambda_i\tilde{\tilde{y}}_i, \text{ where } \lambda_i \text{ eigenvalue of } \mathbf{D}. \tag{3.15}$$

The equation (3.15) has the same form of equation (3.3), so we can apply stability analysis of (3.3) to (3.15).

3.1.2 FE Method

The forward Euler (FE) method is the simplest explicit RK method. The order, Butcher tableau, stability region, and stability formula of this method are given in Table 3.1.

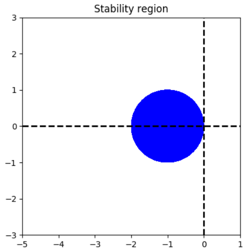
FE			
Order	Butcher Tableau	Stability Region	Stability Formula
One	$\begin{array}{c\|c} 0 & 0 \\ \hline & 1 \end{array}$		$\{z \in \mathbf{C} : \|z+1\| \leq 1\}$

Table 3.1: The order, Butcher tableau, stability region, and stability formula of FE.

3.1.3 EMP Method

The explicit midpoint (EMP) method is also known as the modified FE method [8]. The order, Butcher tableau, stability region, and stability formula of this method are given in Table 3.2.

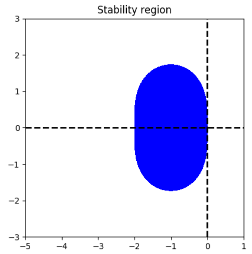
<table>
<tr><td>EMP</td><td></td><td></td><td></td></tr>
<tr><td>Order</td><td>Butcher Tableau</td><td>Stability Region</td><td>Stability Formula</td></tr>
<tr><td>Two</td><td>$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$</td><td>
</td><td>$\left\{z \in \mathbf{C} : \left|\frac{1}{2}z^2 + z + 1\right| \leq 1\right\}$</td></tr>
</table>

Table 3.2: The order, Butcher tableau, stability region, and stability formula of EMP.

3.1.4 RK4 Method

The four-stage, fourth-order RK (RK4) is known as the classical RK method [8]. The order, Butcher tableau, stability region, and stability formula of this method are given in Table 3.3.

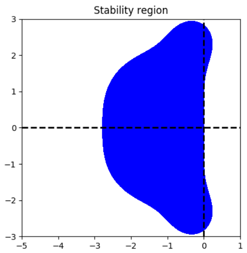
<table>
<tr><td>RK4</td><td></td><td></td><td></td></tr>
<tr><td>Order</td><td>Butcher Tableau</td><td>Stability Region</td><td>Stability Formula</td></tr>
<tr><td>Four</td><td>$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$</td><td>
</td><td>$\left\{z \in \mathbf{C} : \left|\frac{1}{24}z^4 + \frac{1}{6}z^3 + \frac{1}{2}z^2 + z + 1\right| \leq 1\right\}$</td></tr>
</table>

Table 3.3: The order, Butcher tableau, stability region, and stability formula of RK4.

3.1.5 SSP Methods

The strong-stability-preserving (SSP) ERK methods were previously known as total-variation-diminishing time integration methods [25]. These time integration methods preserve the strong stability property $\|\boldsymbol{y}_{n+1}\| \leq \|\boldsymbol{y}_n\|$, of the FE method [26]. These methods are useful, especially in solving hyperbolic partial differential equations, because they have non-linear stability properties [60]. We can write an s-stage explicit SSP methods (3.2) in the following form:

$$\begin{aligned}
\boldsymbol{y}_0 &= \boldsymbol{y}_n, \\
\boldsymbol{y}_i &= \sum_{j=0}^{i-1} \left(\alpha_{ij}\boldsymbol{y}_j + \Delta t \beta_{ij} f(\boldsymbol{y}_j)\right), \quad \text{for} \quad i = 1, 2, \ldots, s, \\
\boldsymbol{y}_{n+1} &= \boldsymbol{y}_s,
\end{aligned} \tag{3.16}$$

where all α_{ij} and β_{ij} are non-negative and $\alpha_{ij} = 0$ only if $\beta_{ij} = 0$, and for consistency $\sum_{j=0}^{j-1} \alpha_{ij} = 1$. This form is known as Shu–Osher form [25]. Because the α_{ij} and β_{ij} are non-negative, the equation (3.16) is a convex combination of forward Euler step, Δt is replaced with modified stepsizes of $\frac{\beta_{ij}}{\alpha_{ij}}\Delta t$. The coefficient β_{ij} can be negative, in which case the function $\boldsymbol{f}$ must be modified to account for a downwinded discretization; for more details see [57]. Next, we present two SSP RK methods.

SSP$(3,3)$ method

The three-stage, third-order strong-stability-preserving (SSP$(3,3)$) method is broadly known as the Shu–Osher method [25]. It is probably the most popular and commonly used SSP method [25]. The order, Butcher tableau, stability region, stability formula, coefficients of α_{ij} and β_{ij} [60] of this method are given in Table 3.4.

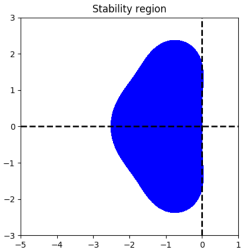
<table>
<tr><td>SSP(3,3)</td><td></td><td></td><td></td></tr>
<tr><td>Order</td><td>Butcher Tableau</td><td>Stability Region</td><td>Stability Formula</td></tr>
<tr><td>Three</td><td>$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ \hline & 1/6 & 1/6 & 2/3 \end{array}$</td><td>
</td><td>$\left\{z \in \mathbf{C} : \left\lvert \frac{1}{6}z^3 + \frac{1}{2}z^2 + z + 1 \right\rvert \leq 1\right\}$</td></tr>
<tr><td>Stage</td><td>α_{ij}</td><td>β_{ij}</td><td>Consistency</td></tr>
<tr><td>Three</td><td>$\begin{matrix} 1 & 0 & 0 \\ 3/4 & 1/4 & 0 \\ 1/3 & 0 & 2/3 \end{matrix}$</td><td>$\begin{matrix} 1 & 0 & 0 \\ 3/4 & 1/4 & 0 \\ 1/3 & 0 & 2/3 \end{matrix}$</td><td>$\sum_{j=0}^{j-1} \alpha_{ij} = 1, \quad i = 1, 2, \ldots, s$</td></tr>
</table>

Table 3.4: The order, Butcher tableau, stability region, stability formula, α_{ij}, and β_{ij} of SSP$(3,3)$.

SSP$(4,3)$ method

The SSP$(4,3)$ gives improved stability over SSP$(3,3)$. The order, Butcher tableau, stability region, stability formula, coefficients of α_{ij} and β_{ij} [60] of SSP$(4,3)$ method are given in Table 3.5.

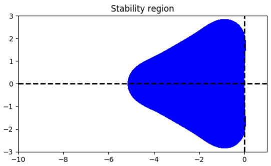
SSP(4,3)			
Order	Butcher Tableau	Stability Region	Stability Formula
Three	$\begin{array}{c\|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/6 & 1/6 & 1/6 & 0 \\ \hline & 1/6 & 1/6 & 1/6 & 1/2 \end{array}$		$\left\{z \in \mathbf{C} : \left\| \frac{1}{48}z^4 + \frac{1}{6}z^3 + \frac{1}{2}z^2 + z + 1 \right\| \leq 1\right\}$
Stage	α_{ij}	β_{ij}	Consistency
Four	$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{array}$	$\begin{array}{cccc} 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/6 & 0 \\ 0 & 0 & 0 & 1/2 \end{array}$	$\sum_{j=0}^{j-1} \alpha_{ij} = 1, \quad i = 1, 2, \ldots, s$

Table 3.5: The order, Butcher tableau, stability region, stability formula, α_{ij}, and β_{ij} of SSP$(4,3)$.

3.1.6 RKC Methods

The Runge–Kutta–Chebyshev (RKC) methods are explicit stabilized RK methods [29]. These methods are explicit one-step methods with extended stability domains along the negative real axis.

For stiff problems, the standard RK methods are not efficient because they have severe stability constraints. For semi-discrete parabolic problems, implicit methods can be computationally expensive in higher dimensions. However, RKC methods are useful for such problems because they extend the real stability interval with a length proportional to s^2, where s is the number of stages, using the shifted Chebyshev polynomial as the stability polynomial. This polynomial is given as follows:

$$P_s(z) = T_s\left(1 + \frac{z}{s^2}\right), \quad \beta_r = 2s^2,$$

where $[-\beta_r, 0]$ is the largest segment of the negative x-axis belongs to the stability region $\mathbb{S}$, and $T_s(z)$ is the Chebyshev polynomial defined by

$$T_s(z) = \cos(s \arccos(z)), \quad z \in \mathbb{C}.$$

The s-stage RKC method [29] is written as

$$\begin{aligned}
&\boldsymbol{y}_{n0} = \boldsymbol{y}_n, \\
&\boldsymbol{y}_{n1} = \boldsymbol{y}_n + \tilde{\mu}_1 \Delta t \boldsymbol{f}(t_n + c_k \Delta t, \boldsymbol{y}_{n0}), \\
&\boldsymbol{y}_{nj} = (1 - \mu_j - \nu_j)\boldsymbol{y}_n + \mu_j \boldsymbol{y}_{n,j-1} + \nu_j \boldsymbol{y}_{n,j-2} + \tilde{\mu}_j \Delta t \boldsymbol{f}(t_n + c_k \Delta t, \boldsymbol{y}_{n,j-1}) + \tilde{\gamma}_j \boldsymbol{y}_{n+1}, \quad \text{for} \quad j = 2, 3, ..., s, \\
&\boldsymbol{y}_{n+1} = \boldsymbol{y}_{ns},
\end{aligned}$$

and,

$$\tilde{\mu}_1 = b_1 u_1,\ \mu_j = \frac{2b_j u_0}{b_{j-1}},\ \nu_j = \frac{-b_j}{b_{j-2}}, \tilde{\mu}_j = \frac{2b_j u_1}{b_{j-1}},\ \tilde{\gamma}_j = -a_{j-1}\tilde{\mu}_j,\ a_j = 1 - b_j T_j(u_0).$$

For an s-stage, first-order RKC method, the parameters w_0, w_1, and b_j, are defined as,

$$w_0 = 1 + \epsilon/s^2, \quad w_1 = T_s(w_0)/T_s'(w_0),\ \text{where } \epsilon \text{ is the damping parameter,}$$

and,

$$b_j = \frac{1}{T_j(w_0)}.$$

Damping parameters implemented to the RKC method to overcome a small imaginary perturbation on z might lead to instability [29]. These damping parameters extended the stability region and reducing the intersection of the stability region with the negative x-axis; see Figure 3.1 and Figure 3.2. For practical problems, a convenient choice for ϵ studied in [29] is 0.05.

RKC$(2,1)$

The order, Butcher tableau, stability region, and stability formula of two-stage, first-order Runge–Kutta–Chebyshev (RKC$(2,1)$) method are given in Table 3.6.

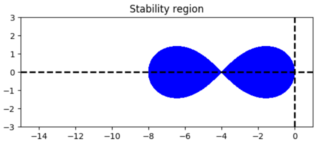
RKC(2,1)			
Order	Butcher Tableau	Stability Region	Stability Formula
One	$\begin{array}{c\|cc} 0 & 0 & 0 \\ 1/4 & 1/4 & 0 \\ \hline & 1/2 & 1/2 \end{array}$		$\left\{z \in \mathbf{C} : \left\lvert\frac{1}{8}z^2 + z + 1\right\rvert \leq 1\right\}$

Table 3.6: The order, Butcher tableau, stability region, stability formula of RKC$(2,1)$.

RKC$(3,1)$

The order, Butcher tableau, stability region, and stability formula of three-stage, first-order Runge–Kutta–Chebyshev (RKC$(3,1)$) method are given in Table 3.7.

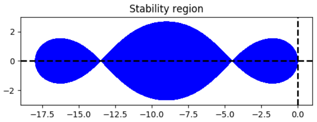
RKC(3,1)			
Order	Butcher Tableau	Stability Region	Stability Formula
One	$\begin{array}{c\|ccc} 0 & 0 & 0 & 0 \\ 1/9 & 1/9 & 0 & 0 \\ 4/9 & 2/9 & 2/9 & 0 \\ \hline & 1/3 & 4/9 & 2/9 \end{array}$		$\left\{z \in \mathbf{C} : \left\lvert \frac{4}{729}z^3 + \frac{4}{27}z^2 + z + 1 \right\rvert \leq 1\right\}$

Table 3.7: The order, Butcher tableau, stability region, stability formula of RKC$(3, 1)$.

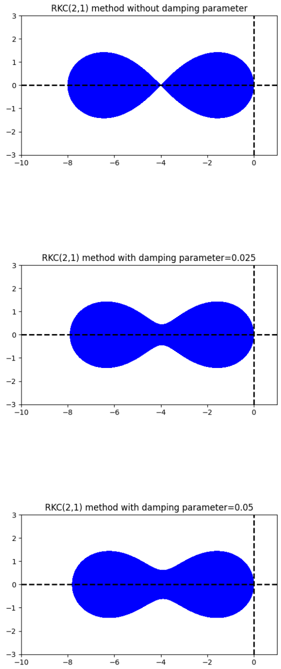


Figure 3.1: Stability regions of RKC$(2, 1)$ method without damping parameter and with damping parameter.

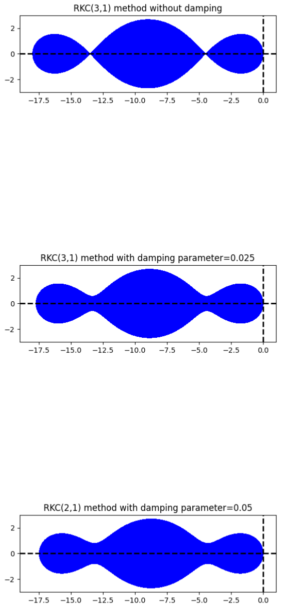


Figure 3.2: Stability regions of RKC$(3, 1)$ method without damping parameter and with damping parameter.

3.2 The theoretical largest stepsize

In this subsection, we propose a new efficient method to find the theoretical largest stepsize of a time integration method to solve the given system ODEs.

The stability region of the time integration method is $|R(z)| \leq 1$ where $|R(z)|$ is a complex polynomial, for example, see Figure 3.3. We determine the theortical largest stepsize of the time integration method based on its linear stability region and the eigenvalues computed from the Jacobian matrix of a system of ODEs.

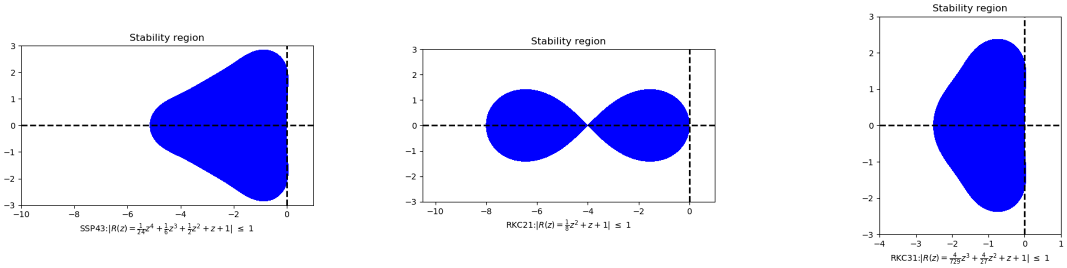


Figure 3.3: Stability regions and their complex polynomials of time integration methods.

The procedure used to find the largest time step is summarized in the following steps:

Step 1: **Compute boundary values of stability region**. The boundary of the stability region is the set of all z such that $R(z)$ is on the unit circle.

$$R(z) = e^{i\theta}, \tag{3.17}$$

for some $\theta \in [0, 2\pi]$. To compute the values of z of equation (3.17) analytically is a tedious task. However, we can solve it easily numerically using MATLAB's built-in *roots* function. This gives us the boundary values of the stability region.

Step 2: **Compute eigenvalues from the Jacobian matrix**. We use the following procedure to find the eigenvalues:

1. We generate the reference solution for (3.1) by using MATLAB's built-in *ode15s* function. This function used the following syntax:

$$[T, \boldsymbol{y}] = \textit{ode15s}(@\boldsymbol{f}(t, \boldsymbol{y}), [t_0{:}\Delta t^*{:}t_f], \text{'RelTol'}, \text{1e–12}, \text{'AbsTol'}, \text{1e–12}),$$

where $\boldsymbol{f}(t, \boldsymbol{y})$ is the RHS of (3.1), t_0 is the initial time, t_f is the final time, and Δt^* is the time interval at which eigenvalues are stored. This value is chosen small enough to capture the evolution of the eigenvalues.

2. We use the reference solution to compute the Jacobian matrix. Then, we compute the eigenvalues from the Jacobian matrix.

```
**for** i=$t_0$:$\Delta t^*$:$t_f$ **do**
    Get the current $\boldsymbol{y}$ values from the reference solution,
        $\boldsymbol{y}_{\text{ref}} = \boldsymbol{y}(i + 1, :).$
```

We use the current reference solution to compute $\boldsymbol{f}(t, \boldsymbol{y})$,

$$\boldsymbol{f}_{\text{current}} = \boldsymbol{f}(t^*, \boldsymbol{y}_{\text{ref}}). \text{ where } t^* = t_0 + i\Delta t^*.$$

We compute the current Jacobian matrix using MATLAB's built-in *numjac* function,

$$[\text{jac}, \text{fac}] = \mathit{numjac}(@\boldsymbol{f}(t, \boldsymbol{y}), t^*, \boldsymbol{y}_{\text{ref}}, \boldsymbol{f}_{\text{current}}, \text{'thresh'}, 1\text{e–}12, \text{fac}, [\ ], \text{S}, 0),$$

which numerically approximates the Jacobian matrix of $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}$. The column vector `thresh` gives a threshold of importance $\boldsymbol{y}_{\text{ref}}$ (i.e, $\boldsymbol{y}_{\text{ref}}(i)$ with $|\boldsymbol{y}_{\text{ref}}(i)| \leq$ `thresh`(i) is considered not important). The column fac is working storage. On the first call, we assign `fac` to []. This does not change the returned value between calls. The matrix S is a non-empty sparse matrix of zeros and ones. A value of 0 for S(i, j) means that element j of the vector $\boldsymbol{f}(t, \boldsymbol{y})$ does not depend on element i of vector $\boldsymbol{y}$.

We compute eigenvalues using MATLAB's built-in *eig* function on the Jacobian matrix,

$$\text{eigenvalues}(i + 1, :) = \mathit{eig}(\text{jac}).$$

end for

Step 3: **Scaling the eigenvalues**. We scale each eigenvalue to the nearest boundary of the stability region.

1. The stability region is a set of values of $z = \lambda\Delta t$ such that $|R(z)| \leq 1$. We are trying to bring the value of $\lambda\Delta t$ to the boundary of the stability region of the time integration method because this is the largest stable time step. We scale each eigenvalue λ_i, $i = 1, 2, \ldots, mn^*$, where $n^* = \left(\frac{t_f - t_0}{\Delta t^*} + 1\right)$ is the number of steps taken, computed from Step 2 to the boundary of stability region using the formula, $\Delta t_i = \left|\frac{z_j}{\lambda_i}\right|$, $j = 1, 2, ..., n$ (n is the number of numerical roots of (3.17) used for plotting the boundary of the stability region). We use the following procedure to get stable time steps by bringing the eigenvalues to the nearest boundary of the stability region:

   a. **Find the nearest point**. We use MATLAB's built-in *dsearchn* function to find the nearest point of the eigenvalue to boundary values of the stability region. The *dsearchn*(boundary values of the stability region, λ_i) gives the indices of the nearest point.

   b. **Determine the stable step size**. We get the stable step size by scaling the eigenvalues to the nearest boundary; see Figure 3.4.

$$\Delta t_i = \left|\frac{\text{The nearest point on the boundary to } \lambda_i}{\lambda_i}\right| \tag{3.18}$$

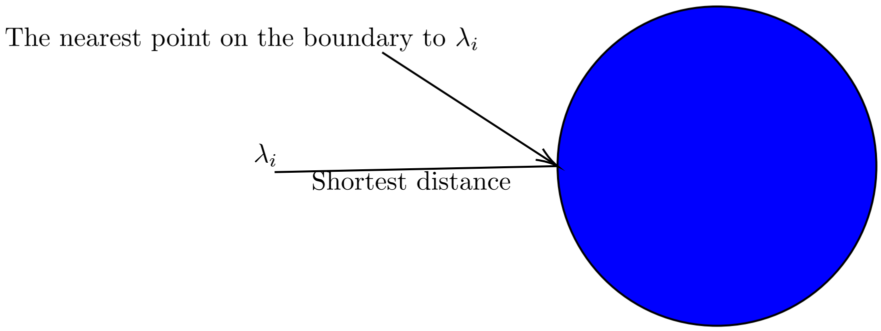


Figure 3.4: The figure shows the shortest distance between the eigenvalue and the boundary values of the stability region of the time integration method.

2. The minimum of stable time steps computed from equation (3.18) gives us the theoretical largest stable stepsize. The theoretical largest stepsize is

$$\Delta t_{\max} = \min(\Delta t_i), \quad i = 1, 2, \ldots, mn^*. \tag{3.19}$$

3.2.1 Measuring the accuracy

In this subsection, we are interested in measuring the accuracy of the numerical solution. In this thesis, we use the following accuracy measurement, which is called MRMS error [41]:

$$[\mathrm{MRMS}]_y := \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{y}_i - y_i}{1 + |\hat{y}_i|}\right)^2}, \tag{3.20}$$

where y_i is the numerical solution and $\hat{y}_i$ is the reference solution to y_i at time point i and N is the number of solution points. The $[\mathrm{MRMS}]_v$ measure is based on a combination of relative error and absolute error. This error norm gives more satisfactory results than the *Relative Root Mean Squared* (RRMS) error [41]. We used an $[\mathrm{MRMS}]_v$ of 5% as acceptable for the cardiac cell model simulation and the monodomain simulation, The procedure used to an acceptable $[\mathrm{MRMS}]_v$ is summarized in the following steps:

Step 1: **Reference solution**: We generate a reference solution by using a MATLAB's built-in *ode15s* function and lowering the error tolerances for successive approximations until two approximations are identical for at least ten significant digits at 100 equally spaced output points in time.

Step 2: **Numerical solution**: Using the theoretical largest stepsize ($\Delta t_{\max}$), we generate a numerical solution.

Step 3: **Interpolation of the numerical solution**: Using linear interpolation, we construct 100 new data points from the numerical solution.

Step 4: **Calculate MRMS**: We calculate $[\mathrm{MRMS}]_v$ by using equation (3.20).

Step 5: **Checking the accuracy**: If the $[\mathrm{MRMS}]_v$ is less than or equal to 0.05, then we have achieved an acceptable $[\mathrm{MRMS}]_v$.

Step 6: **Extrapolation**: If the $[\text{MRMS}]_v$ greater than 0.05, then we scale the $\Delta t_{\max}$ by using extrapolation:

$$\Delta t_{\max_{New}} = \Delta t_{\max} \left(\frac{0.05}{[\text{MRMS}]_v} \right)^{\frac{1}{p}}, \quad p \text{ is the order of time integration method.}$$

Step 7: **Repeats steps**: Repeat Step 2 to Step 6 until we satisfy Step 5.

3.3 Experimental largest stepsize

In this section, we present a method to find the experimental largest stepsize of a time integration method to solve a given system ODEs to within a specified $[\text{MRMS}]_v$.

The procedure used to find the experimental largest stepsize is summarized in the following steps:

Step 1: **Reference solution**: We generate a reference solution by using a MATLAB's built-in *ode15s* function and lowering the error tolerances for successive approximations until two approximations are identical for at least ten significant digits at 100 equally spaced output points in time.

Step 2: **Starting time step:** We use the theoretical largest stepsize as a starting time step to compute the experimental largest stepsize.

Step 3: **Tune time step:** The time step value is incremented or decremented until the $[\text{MRMS}]_v$ tolerance is satisfied. If the time step value has d_{be} digits before the decimal point and d digits after the decimal point (say $0.a_1a_2\ldots a_d$), then the time step is changed by 10^{-d} $(0.00\ldots1_d)$. If the $[\text{MRMS}]_v$ is greater than 0.05, we decrement the time step value by 10^{-d}. If the $[\text{MRMS}]_v$ is less than 0.05, we increment the time step by 10^{-d}. This gives us a new time step value of $(0.a_1a_2\ldots(a_d-1))$ or $(0.a_1a_2\ldots(a_d+1))$ and using this time step, we do one simulation and check the accuracy. If the accuracy is satisfied, we are done. If not, we repeat the process by decrementing or incrementing the new time step.

Step 4: **Numerical solution**: Using a given time step, we generate a numerical solution.

Step 5: **Interpolation of the numerical solution**: Using linear interpolation, we construct 100 new data points from the numerical solution.

Step 6: **Calculate MRMS**: We calculate $[\text{MRMS}]_v$ by using the linear interpolation of the numerical solution and reference solution.

Step 7: **Experimental largest stepsize**:

```
if the $[\text{MRMS}]_v$ is equal to 0.05 then
    this gives us the experimental largest stepsize such that $[\text{MRMS}]_v$ equal to 0.05.
else if the $[\text{MRMS}]_v$ is less than 0.05 then
    i. the time step is incremented as described in Step 3. We increment the time step to check if
       there is a larger stable time step with less than 0.05 $[\text{MRMS}]_v$.
    ii. Using the new time step, repeat Step 4 to Step 6 to check MRMS error greater than 0.05.
        if the MRMS error is less than 0.05 then
            we increment the time step as described in Step 3 and repeat Step 4 to Step 6.
        end if
else
    i. The time step is decremented as described in Step 3. We decrement the time step to check if there is a smaller time step with less than 0.05 $[\text{MRMS}]_v$.
    ii. Using the new time step, repeat Step 4 to Step 6 to check the $[\text{MRMS}]_v$ less than 0.05.
        if the $[\text{MRMS}]_v$ is greater than 0.05 then
            we decrement time step as described in Step 3 and repeat Step 4 to Step 6.
        end if
        this new time step is the experimental largest stepsize such that $[\text{MRMS}]_v$ is less than 0.05.
end if
```

3.4 Fully discrete first-order Godunov operator-splitting method for the monodomain model

The monodomain model consists of systems of non-linear ODEs of equation (2.5a) coupled with a PDE equation (2.5b). Solving this model numerically is a daunting task because of its complexity. Using the operator splitting method, this complex system of the model can be split into two smaller parts that can be easier to solve. There are operator splitting methods higher than order two, but they need negative time stepsizes [70]. These time stepsizes may lead to instability for operator splitting methods [55]. For this thesis, we apply a first-order operator splitting method called Godunov splitting. This method solves the monodomain PDE and the coupled ODE model into the two steps [63]:

1. Using the initial conditions v_n and $\boldsymbol{s}_n$, solve the system of ODEs arising from the cell models for $t \in [t_n, t_n + \Delta t]$ to obtain $v_{n+1}^{(1)}$ and $\boldsymbol{s}_{n+1}^{(1)}$:

$$\frac{\partial \boldsymbol{s}}{\partial t} = \boldsymbol{f}(\boldsymbol{s}, v, t), \tag{3.21a}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{C_{me}} \boldsymbol{I}_{ion}(\boldsymbol{s}, v, t). \tag{3.21b}$$

2. Solve the PDE, using $v_{n+1}^{(1)}$ as the initial condition, for $t \in [t_n, t_n + \Delta t]$ and the boundary conditions

$\hat{\boldsymbol{n}} \cdot (\boldsymbol{M}_I \nabla v) = 0$, to obtain v_{n+1}:

$$\chi C_{me} \frac{\partial v}{\partial t} = \frac{\lambda}{1+\lambda} \nabla \cdot (\boldsymbol{M}_I \nabla v), \tag{3.22}$$

3.4.1 Spatial discretization

In this subsection, we semi-discretize (3.22) in space, using the finite element method.

The meshes $(x_0 < x_1, \ldots, < x_N)$ is quasi uniform if there exists a number $C > 0$, independent of $h = \frac{x_N - x_0}{N}$ and i such that

$$\frac{h}{h_i} \leq C, \text{ as } N \to \infty, \text{ where } i = 1, 2, \ldots, N, \text{ and } h_i = x_i - x_{i-1}.$$

We first define the $L_2(\Omega)$ space over the open and bounded domain Ω as follows:

$$L_2(\Omega) = \left\{ w : \int_\Omega w(X) w(X) \, \mathrm{d}X < \infty \right\}.$$

The L_2 inner product over the domain Ω is defined as follows:

$$\langle w, u \rangle = \int_\Omega w(X) u(X) \, \mathrm{d}X, \quad \forall w, u \in L_2(\Omega).$$

The Sobolev space $H^1(\Omega)$ is defined as follows:

$$H^1(\Omega) = \{ w \in L_2(\Omega) : \boldsymbol{\nabla} w \in L_2(\Omega) \},$$

and the variational formulation of the problem (3.22) is defined as follows: find $v(x,t) \in H^1$, for each t, such that

$$\left\langle \chi C_{me} \frac{\partial v}{\partial t}, w \right\rangle = \left\langle \boldsymbol{M}_I \nabla v, \frac{\lambda}{1+\lambda} \nabla w \right\rangle, \quad \forall w \in H^1.$$

Finite element method

We use a continuous, piecewise-linear finite element over a quasi-uniform mesh of Ω, so $v_h : [0, T] \to K_h$ satisfies

$$\left\langle \chi C_{me} \frac{\partial v_h}{\partial t}, w \right\rangle = \left\langle \frac{\lambda}{1+\lambda} \nabla \cdot (\boldsymbol{M}_I \nabla v_h), w \right\rangle, \quad \forall w \in K_h, \tag{3.23}$$

where, $K_h = \{ u \in H^1(\Omega) : u|_k \in \text{polynomial of degree} \leq K \}$. This K_h space is spanned by basis functions Φ_i, $i = 0, 1, 2, \ldots, N_x + 1$, where N_x is the total number of interior nodes, defined as follows:

$$\Phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i], \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise}, \end{cases}$$

$$\Phi_0(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_0}, & x \in [x_0, x_1], \\ 0, & \text{otherwise.} \end{cases}$$

and,

$$\Phi_{N_x+1}(x) = \begin{cases} \frac{x - x_{N_x}}{x_{N_x+1} - x_{N_x}}, & x \in [x_{N_x}, x_{N_x+1}], \\ 0, & \text{otherwise.} \end{cases}$$

The one-dimensional, two-dimensional, and three-dimensional semi-discrete schemes for the equation (3.23) given as follows:

1. In one dimension, we write the v_h for each fixed t, of the basis functions as follows:

   $$v_h(x,t) = \sum_{i=0}^{N_x+1} v_i(t)\Phi_i(x),$$

   where $\{\Phi_i\}_{i=1}^{N_x+1}$ a basis for K_h where N_x is the total number of interior nodes. We insert this basis function into equation (3.23) and choose $w = \Phi_j$ to find the following equivalent equation

   $$\chi C_{me} \sum_{i=0}^{N_x+1} \frac{\partial v_i}{\partial t} \Big\langle \Phi_i, \Phi_j \Big\rangle = - \sum_{i=0}^{N_x+1} v_i(t) \Big\langle \boldsymbol{M}_I \left(\frac{d\Phi_i}{dx} \right), \frac{\lambda}{1+\lambda} \left(\frac{d\Phi_j}{dx} \right) \Big\rangle, \text{ for } j = 0, 1, \ldots, N_x+1. \quad (3.24)$$

   Define one vector and two matrices $\boldsymbol{v}$, $\boldsymbol{A}_x$, and $\boldsymbol{A}_{I_x}$ as follows:

   $$\boldsymbol{v} = (v_i)_{(N_x+2)\times 1}, \quad [\boldsymbol{A}_x]_{ij} = \Big\langle \Phi_i, \Phi_j \Big\rangle_{(N_x+2)\times(N_x+2)}, \quad [\boldsymbol{A}_{I_x}]_{ij} = \Big\langle \boldsymbol{M}_I \left(\frac{d\Phi_i}{dx} \right), \frac{\lambda}{1+\lambda} \left(\frac{d\Phi_j}{dx} \right) \Big\rangle_{(N_x+2)\times(N_x+2)}.$$

   Therefore, equation (3.24) becomes

   $$\chi C_{me} \frac{\partial \boldsymbol{v}}{\partial t} \boldsymbol{A}_x = -\boldsymbol{v}(t) \boldsymbol{A}_{I_x}. \quad (3.25)$$

2. In two dimensions, we write the v_h for each fixed t, of the basis functions as follows:

   $$v_h(x,y,t) = \sum_{i=0}^{N_y+1} \sum_{j=0}^{N_x+1} v_{i,j}(t)\Phi_i(x)\phi_i(y),$$

   where $\{\Phi_i\}_{i=0}^{N_x+1}$ and $\{\phi_j\}_{j=0}^{N_y+1}$ basis for K_h where N_x and N_y is the total number of interior nodes in x-direction and y-direction. We insert this basis function into equation (3.23) and choose $w = \Phi_l \phi_k$ to find the following equivalent equation

   $$\begin{aligned} \chi C_{me} \sum_{i=0}^{N_y+1} \sum_{j=0}^{N_x+1} \frac{\partial v_{i,j}}{\partial t} \Big\langle \Phi_j \phi_j, \Phi_l \phi_k \Big\rangle &= - \sum_{i=0}^{N_y+1} \sum_{j=0}^{N_x+1} v_{i,j}(t) \Big\langle \boldsymbol{M}_I \left(\frac{d\Phi_i}{dx} \right) \phi_j, \frac{\lambda}{1+\lambda} \left(\frac{d\Phi_l}{dx} \right) \phi_k \Big\rangle \\ &\quad - \sum_{i=0}^{N_y+1} \sum_{j=0}^{N_x+1} v_{i,j}(t) \Big\langle \boldsymbol{M}_I \Phi_i \left(\frac{d\phi_j}{dy} \right), \frac{\lambda}{1+\lambda} \Phi_l \left(\frac{d\phi_k}{dy} \right) \Big\rangle, \end{aligned} \quad (3.26)$$

   for $l = 0, 1, \ldots, N_x + 1$ and $k = 0, 1, \ldots, N_y + 1$.

Define one vector and four matrices $\boldsymbol{v}$, $\boldsymbol{A}_x$, $\boldsymbol{A}_{I_x}$, $\boldsymbol{A}_y$, and $\boldsymbol{A}_{I_y}$ as follows:

$$\boldsymbol{v} = (v_{i,j})_{\big((N_x+2)(N_y+2)\big)\times 1}, \quad [\boldsymbol{A}_x]_{il} = \Big\langle \Phi_i, \Phi_l \Big\rangle_{(N_x+2)\times(N_x+2)}, \quad [\boldsymbol{A}_{I_x}]_{il} = \Big\langle \boldsymbol{M}_I \left(\frac{d\Phi_i}{dx}\right), \frac{\lambda}{1+\lambda}\left(\frac{d\Phi_l}{dx}\right) \Big\rangle_{(N_x+2)\times(N_x+2}$$

$$[\boldsymbol{A}_y]_{jk} = \Big\langle \phi_j, \phi_k \Big\rangle_{(N_y+2)\times(N_y+2)}, \quad [\boldsymbol{A}_{I_y}]_{jk} = \Big\langle \boldsymbol{M}_I \left(\frac{d\phi_j}{dy}\right), \frac{\lambda}{1+\lambda}\left(\frac{d\phi_k}{dy}\right) \Big\rangle_{(N_y+2)\times(N_y+2)}.$$

Therefore, equation (3.26) becomes

$$\chi C_{me} \frac{\partial \boldsymbol{v}}{\partial t} \left(\boldsymbol{A}_x \otimes \boldsymbol{A}_y\right) = -\boldsymbol{v}(t) \left(\boldsymbol{A}_{I_x} \otimes \boldsymbol{A}_y\right) - \boldsymbol{v}(t) \left(\boldsymbol{A}_x \otimes \boldsymbol{A}_{I_y}\right). \tag{3.27}$$

where $\otimes$ denotes the Kronecker product of two matrices [6].

3. In three dimensions, we write the v_h for each fixed t, of the basis functions as follows:

$$v_h(x,y,z,t) = \sum_{i=0}^{N_z+1} \sum_{j=0}^{N_y+1} \sum_{\ell=0}^{N_x+1} v_{i,j,\ell}(t) \Phi_i(x) \phi_i(y) \varphi_\ell(z),$$

where $\{\Phi_i\}_{i=0}^{N_x+1}$, $\{\phi_j\}_{j=0}^{N_y+1}$, and $\{\varphi_\ell\}_{\ell=0}^{N_z+1}$ basis for K_h where N_x, N_y, and N_z is the total number of interior nodes in the x-direction, y-direction, and z-direction. We insert this basis function into equation (3.23) and choose $w = \Phi_l \phi_k \varphi_p$ to find the following equivalent equation

$$\begin{aligned} \chi C_{me} \sum_{i=0}^{N_z+1} \sum_{j=0}^{N_y+1} \sum_{\ell=0}^{N_z+1} \frac{\partial v_{i,j,\ell}}{\partial t} \Big\langle \Phi_i \phi_j \varphi_l, \Phi_l \phi_k \varphi_p \Big\rangle = &- \sum_{i=0}^{N_z+1} \sum_{j=0}^{N_y+1} \sum_{\ell=0}^{N_z+1} v_{i,j,\ell}(t) \Big\langle \boldsymbol{M}_I \left(\frac{d\phi_j}{dx}\right) \phi_j \varphi_\ell, \frac{\lambda}{1+\lambda}\left(\frac{d\phi_l}{dx}\right) \phi_k \varphi_p \Big\rangle \\ &- \sum_{i=0}^{N_z+1} \sum_{j=0}^{N_y+1} \sum_{\ell=0}^{N_z+1} v_{i,j,\ell}(t) \Big\langle \boldsymbol{M}_I \Phi_i \left(\frac{d\phi_j}{dy}\right) \varphi_\ell, \frac{\lambda}{1+\lambda} \Phi_l \left(\frac{d\phi_k}{dy}\right) \varphi_p \Big\rangle \\ &- \sum_{i=0}^{N_z+1} \sum_{j=0}^{N_y+1} \sum_{\ell=0}^{N_z+1} v_{i,j,n}(t) \Big\langle \boldsymbol{M}_I \Phi_i \phi_j \left(\frac{d\varphi_\ell}{dz}\right), \frac{\lambda}{1+\lambda} \Phi_l \phi_k \left(\frac{d\varphi_p}{dz}\right) \Big\rangle, \\ &\text{for } l = 0,1,\ldots,N_x+1,\ k = 0,1,\ldots,N_y+1, \text{ and} \\ &p = 0,1,\ldots,N_z+1. \end{aligned} \tag{3.28}$$

Define one vector and six matrices $\boldsymbol{v}_m$, $\boldsymbol{A}_x$, $\boldsymbol{A}_{I_x}$, $\boldsymbol{A}_y$, $\boldsymbol{A}_{I_y}$, $\boldsymbol{A}_z$, and $\boldsymbol{A}_{I_z}$as follows:

$$\begin{aligned} &\boldsymbol{v} = (v_{i,j,n})_{\big((N_x+2)(N_y+2)(N_z+2)\big)\times 1}, \\ &[\boldsymbol{A}_x]_{il} = \Big\langle \Phi_i, \Phi_l \Big\rangle_{(N_x+2)\times(N_x+2)}, \quad [\boldsymbol{A}_{I_x}]_{il} = \Big\langle \boldsymbol{M}_I \left(\frac{d\Phi_i}{dx}\right), \frac{\lambda}{1+\lambda}\left(\frac{d\Phi_l}{dx}\right) \Big\rangle_{(N_x+2)\times(N_x+2)}, \\ &[\boldsymbol{A}_y]_{jk} = \Big\langle \phi_j, \phi_k \Big\rangle_{(N_y+2)\times(N_y+2)}, \quad [\boldsymbol{A}_{I_y}]_{jk} = \Big\langle \boldsymbol{M}_I \left(\frac{d\phi_j}{dy}\right), \frac{\lambda}{1+\lambda}\left(\frac{d\phi_k}{dy}\right) \Big\rangle_{(N_y+2)\times(N_y+2)}, \\ &[\boldsymbol{A}_z]_{np} = \Big\langle \varphi_n, \varphi_p \Big\rangle_{(N_z+2)\times(N_z+2)}, \quad [\boldsymbol{A}_{I_z}]_{np} = \Big\langle \boldsymbol{M}_I \left(\frac{d\varphi_n}{dz}\right), \frac{\lambda}{1+\lambda}\left(\frac{d\varphi_p}{dz}\right) \Big\rangle_{(N_z+2)\times(N_z+2)}. \end{aligned}$$

Therefore, equation (3.28) becomes

$$\chi C_{me}\frac{\partial \boldsymbol{v}}{\partial t}\left(\boldsymbol{A}_x\otimes\boldsymbol{A}_y\otimes\boldsymbol{A}_z\right) = -\boldsymbol{v}(t)\left(\boldsymbol{A}_{I_x}\otimes\boldsymbol{A}_y\otimes\boldsymbol{A}_z\right) - \boldsymbol{v}(t)\left(\boldsymbol{A}_x\otimes\boldsymbol{A}_{I_y}\otimes\boldsymbol{A}_z\right) \\ - \boldsymbol{v}(t)\left(\boldsymbol{A}_x\otimes\boldsymbol{A}_y\otimes\boldsymbol{A}_{I_z}\right). \tag{3.29}$$

3.4.2 FE method

To define the one-dimensional, two-dimensional, and three-dimensional fully discrete scheme of the first-order Godunov operator-splitting method for the monodomain model of equations (3.21) and (3.22), we discretize equation (3.21) and the semi-discretization of (3.25), (3.27), and (3.29) using the FE method defined by the Butcher tableau in Table 3.1.

3.4.3 RKC$(2,1)$ method

To define the one-dimensional, two-dimensional, and three-dimensional fully discrete scheme of the first-order Godunov operator-splitting method for the monodomain model of equations (3.21) and (3.22), we discretize equation (3.21) and the semi-discretization of (3.25), (3.27), and (3.29) using the RKC$(2,1)$ method defined by the Butcher tableau in Table 3.6.

3.4.4 RKC$(3,1)$ scheme

To define the one-dimensional, two-dimensional, and three-dimensional fully discrete scheme of the first-order Godunov operator-splitting method for the monodomain model of equations (3.21) and (3.22), we discretize equation (3.21) and the semi-discretizzation of (3.25), (3.27), and (3.29) using the RKC$(3,1)$ method defined by the Butcher tableau in Table 3.7.

3.4.5 Richardson Extrapolation

We generate a reference solution for equations (3.21) and (3.22) by using Richardson extrapolation [12]. Using this method, we generate high-order accuracy results of monodomain problems using the RKC$(3,1)$ method. The procedure used to generate the reference solution of the monodomain problem is summarized in the following steps:

step 1: **Numerical solution of half step size :** First, we solve the monodomain problems using the RKC$(3,1)$ method in a step-doubling manner.

$$R(i,0) = v(X,\Delta t/2^i),\quad i = 1,2,\ldots,r,\ \text{where } r \text{ is the number of time step refinements.}$$

Each time we increase the accuracy of the numerical solution by one order, at the cost of one new $R(i,0)$ computation with a smaller Δt.

step 2: **Extrapolation:** We use the extrapolation formula to improve the first-order numerical solution to more accurate solutions. For each fixed j ($j = 1, 2, \ldots, r$), we computed

$$R(i,j) = \frac{2^j R(i,j-1) - R(i-1,j-1)}{2^j - 1},\ i = 1, 2, \ldots, r.$$

3.4.6 The theoretical largest stepsize

In this subsection, we present a method to find the theoretical largest stepsize of a time integration method to solve the monodomain problem within a specified MRMS error.

The procedure used to find the theoretical largest stepsize is summarized in the following steps:

Step 1: **Compute eigenvalues**. We computed the eigenvalues from cell model consists of systems of non-linear ODEs of equation (3.21) and a PDE equation (3.22).

1. **Reference solution**. We generate a reference solution for equations (3.21) and (3.22) by using Richardson extrapolation [12] method.

2. **Compute eigenvalues from cell model**. The Jacobian matrix is computed from the right-hand side of the non-linear systems of ODEs of equation (3.21) using the reference solution and MATLAB's built-in *numjac* function over the time integration interval. Then, the numerical eigenvalues are computed using MATLAB's built-in *eig* function on the Jacobian matrix. The Jacobians of the coupled systems have size $2(N_x + 2)$, $2(N_x + 2)(N_y + 2)$, and $2(N_x + 2)(N_y + 2)(N_z + 2)$ for the 1D monodomain problem, 2D monodomain problem, and 3D monodomain problem, respectively.

3. **Compute the eigenvalues from PDE**. We implemented the finite element method to discretize the PDE equation (3.22) in space to obtain the system of ODEs. These system ODEs were rearranged to put into the matrix on the right-hand side. Then, the eigenvalues are computed using MATLAB's built-in *eig* function on the matrix.

Step 2: We implemented the techniques of Step 1 and Step 3 presented on Section 3.2 to get the theoretical largest step stepsize.

3.4.7 The experimental largest stepsize

In this subsection, we will present a method to find the experimental largest stepsize of a time integration method to of the monodomain problem.

The procedure used to find the experimental largest stepsize is summarized in the following steps:

Step 1: **Reference solution**: We generate a reference solution for equations (3.21) and (3.22) by using Richardson extrapolation [12].

a. For the monodomain simulation, the reference solution converged to $D = 4$ decimal places of accuracy at N points in space and time. We used $N_x = 21$ ($\Omega = [0, 1]$), $N_x = N_y = 21$ ($\Omega = [0, 1] \times [0, 1]$), and $N_x = N_y = N_z = 21$ ($\Omega = [0, 1] \times [0, 1] \times [0, 1]$) equally spaced points in the spatial direction for one-dimensional problems, two-dimensional problems, three-dimensional problems respectively. Also, we used $N_t = 201$ ($T = [0, 20]$) equally spaced points in the time direction.

b. The reference solution of the spiral wave problem converged to $D = 3$ decimal places of accuracy at $N_xN_yN_t$ points in space and time. We used $N_x = N_y = 51$ ($\Omega = [0, 2.5] \times [0, 2.5]$) equally spaced points in space and $N_t = 20001$ ($T = [0, 1000]$) equally spaced points in time. We were unable to compute a reference solution for the scroll wave simulation.

Step 2: **Starting time stepsize:** We use the theoretical largest stepsize as a starting time stepsize to compute the experimental largest stepsize.

Step 3: **Tune time step:** The time step value is incremented or decremented until the $[\text{MRMS}]_v$ tolerance is satisfied. If the time step value has d_{be} digits before the decimal point and d digits after the decimal point (say $0.a_1a_2 \ldots a_d$), then the time step is changed by 10^{-d} ($0.00 \ldots 1_d$). If the $[\text{MRMS}]_v$ is greater than 0.05, we decrement the time step value by $1o^{-d}$. If the $[\text{MRMS}]_v$ is less than 0.05, we increment the time step by 10^{-d}. This gives us a new time step value of $(0.a_1a_2 \ldots (a_d - 1))$ or $(0.a_1a_2 \ldots (a_d + 1))$ and using this time step, we do one simulation and check the accuracy. If the accuracy is satisfied, we are done. If not, we repeat the process by decrementing or incrementing the new time step.

Step 4: **Numerical solution**: Using a given time step, we generate a numerical solution.

Step 5: **Interpolation of the numerical solution**: Using linear interpolation, we construct new data points from the numerical solution.

a. For the monodomain simulation, we construct $N_x \times N_t = 21 \times 201$, $N_x \times N_y \times N_t = 21 \times 21 \times 201$, and $N_x \times N_y \times N_z \times N_t = 21 \times 21 \times 21 \times 201$ new data points from the numerical solution using MATLAB's built-in *interp2* function, *interp3* function, and *interpn* function for one-dimensional problems, two-dimensional problems, three-dimensional problems respectively.

b. For spiral simulation, we construct $N_x \times N_y \times N_t = 51 \times 51 \times 20001$ new data points from the numerical solution using MATLAB's built-in *interp3* function for two-dimensional spiral problem.

Step 6: **Calculate MRMS**: We calculate $[\text{MRMS}]_v$ by using the linear interpolation of the numerical solution and reference solution.

Step 7: **Experimental largest stepsize**:

if the $[\text{MRMS}]_v$ is equal to 0.05 **then**

```
  this gives us the experimental largest stepsize such that $[\mathrm{MRMS}]_v$ equal to 0.05.
else if the $[\mathrm{MRMS}]_v$ is less than 0.05 then
   i.  the time step is incremented as described in Step 3. We increment the time step to check if
       there is a larger stable time step with less than 0.05 $[\mathrm{MRMS}]_v$
  ii.  Using the new time step, repeat Step 4 to Step 6 to check $[\mathrm{MRMS}]_v$ greater than 0.05.
         if the $[\mathrm{MRMS}]_v$ is less than 0.05 then
           we increment the time step as described in Step 3 and repeat Step 4 to Step 6.
         end if
else
   i.  The time step is decremented as described in Step 3. We decrement the time step to check
       if there is a smaller time step with less than 0.05 $[\mathrm{MRMS}]_v$.
  ii.  Using the new time step, repeat Step 4 to Step 6 to check the $[\mathrm{MRMS}]_v$ less than 0.05.
         if the $[\mathrm{MRMS}]_v$ error is greater than 0.05 then
           we decrement time step as described in Step 3 and repeat Step 4 to Step 6.
         end if
         this new time step is the experimental largest stepsize such that $[\mathrm{MRMS}]_v$ is less than 0.05.
end if
```

4 Numerical Experiments

In this chapter, we provide the numerical experiments that should enable us to have a better understanding of the theories presented in Chapter 2 and Chapter 3.

In Section 4.2, we present thirty-seven cardiac cell models and use them to examine the time integration methods. In Section 4.3, we present results for the numerical solutions of all thirty-seven cardiac cell models. We used all the time integration methods presented in Section 3 to solve these models. We compare these methods based on their theoretical largest stepsize, experimental largest stepsize, CPU time, and $[\text{MRMS}]_v$. In Section 4.4, we present results for the numerical solutions of the monodomain model coupled with the FHN model problem presented in Section 3.4. We used all the first-order time integration methods presented Section 3 to solve this model. We compare these methods based on their theoretical largest stepsize, experimental largest stepsize, CPU time, and $[\text{MRMS}]_v$. In Section 4.5, we present the simulation of the monodomain model coupled with the modified FHN model using the theoretical largest stepsizes of all the first-order time integration presented Section 3. We present the spiral and scroll waves propagation snapshots captured in the simulation of the monodomain model coupled with the modified FHN model.

4.1 Machine used

The simulations are implemented on a laptop with an Intel(R) Pentium(R) 2 core(s) 1.90GHz and 4 GB RAM running 64-bit Windows 10.

4.2 Cardiac cell models

In this thesis, we focus on thirty-seven cardiac cell models. These models are obtained from the CellML model repository [2] and presented in Table 4.1.

4.3 Cell model simulation

In this section, we examine the time integration methods applied to the thirty-seven cardiac cell models. We compare these methods based on their theoretical largest stepsizes, experimental largest stepsizes, MRMS errors, and CPU times. To compute CPU time, we solved the cardiac cell model fifty times using the time integration method and recorded each running times. The minimum of all recorded running times gives us the CPU time to solve the cardiac cell model.

Table 4.1: Details on 37 validated cardiac cell models [40]. Models labeled with an asterisk (*) indicate three cell variants (endocardial cell, epicardial cell, and M-cell). The model referred as Winslow31 is a reduced model with 31 variables compared to its original form.

Model	Reference	Number of variables	Description
Beeler–Reuter (1977)	[10]	8	Mammalian ventricular model
Bondarenko et al. (2004)	[11]	41	Mouse ventricular model
Courtemanche et al. (1998)	[13]	21	Human atrial model
Demir et al. (1994)	[17]	27	Rabbit sinoatrial node model
Demir et al. (1999)	[16]	29	Rabbit sinoatrial node model
DiFrancesco–Noble (1985)	[18]	16	Mammal Purkinje fibre model
Dokos et al. (1996)	[19]	18	Rabbit sinoatrial node model
Faber–Rudy (2000)	[21]	19	Guinea pig ventricular model
FitzHugh–Nagumo (1961)	[22, 44]	2	Nerve membrane model
Fox et al. (2002)	[24]	13	Canine ventricular model
Hilgemann–Noble (1987)	[27]	15	Rabbit atrial model
Hund–Rudy (2004)	[28]	29	Canine ventricular model
Jafri et al. (1998)	[31]	31	Guinea pig ventricular model
Luo–Rudy (1991)	[36]	8	Guinea pig ventricular model
Maleckar et al. (2008)	[39]	30	Human atrial model
McAllister et al. (1975)	[42]	10	Canine Purkinje fibre model
Noble (1962)	[46]	4	Mammal Purkinje fibre model
Noble–Noble (1984)	[47]	15	Rabbit sinoatrial node model
Noble et al. (1991)	[48]	17	Guinea pig ventricular model
Noble et al. (1998)	[49]	22	Guinea pig ventricular model
Nygren et al. (1998)	[50]	29	Human atrial model
Pandit et al. (2001)	[51]	26	Rat left-ventricular model
Pandit et al. (2003)	[52]	26	Rat left-ventricular model
Puglisi–Bers (2001)	[53]	17	Rabbit ventricular model
Sakmann et al. (2000)*	[54]	21	Guinea pig ventricular model
Stewart et al. (2009)	[62]	20	Human Purkinje fibre model
Ten Tusscher et al. (2004)*	[64]	17	Human ventricular model
Ten Tusscher et al. (2006)*	[65]	19	Human ventricular model
Wang–Sobie (2008)	[66]	35	Neonatal mouse ventricular model
Winslow31 (1999)	[68]	31	Canine ventricular model
Zhang et al. (2000)	[71]	15	Rabbit sinoatrial node model

4.3.1 Theoretical results

We implemented the time integration methods, the accuracy measure of the numerical solution, and the theoretical largest stepsize techniques from the Chapter 3 on cardiac cell models to produce the theoretical largest time steps, $[\text{MRMS}]_v$, and CPU time. These results are presented in Table 4.2. We now present some analysis based on the results of Table 4.2; see Table A.1 for complete results.

1. If the theoretical largest stepsize also produces a sufficiently accurate solution, FE will be the most efficient method because it is less expensive per step than any method with more than one stage. The FE method outperforms the RKC$(2,1)$, RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods on all thirty-seven cardiac cell models. The additional stages of the RKC$(2,1)$, RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods add extra costs for each step compared to the FE method.

2. After the FE method, RKC$(2,1)$ method is the most efficient method, outperforming the RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, and RK4 methods on all 37 cardiac cell models. The RKC$(2,1)$ method outperforms the EMP method on thirty-six cardiac cell models. Both the EMP and RKC$(2,1)$ methods are two-stage methods, but the improvements of the time step sizes compensate for the extra costs it takes per step of the RKC$(2,1)$ method. However, the EMP method outperformed the RKC$(2,1)$ method on the FHN model. This is because the larger theoretical stepsize of the EMP method produces a more accurate solution than the RKC$(2,1)$ method.

3. The RKC$(3,1)$ method outperforms the SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods on twenty-eight cardiac cell models. The RKC$(3,1)$ method outperforms the EMP method on twenty-nine cardiac cell models. This was because of the improvements of the time stepsizes compensate for the extra cost it takes per step of the RKC$(3,1)$ method. However, the EMP method outperformed the RKC$(3,1)$ method on eight cardiac cell models. This is because the larger theoretical stepsize of the EMP method produces a more accurate solution than the RKC$(3,1)$ method.

4. The SSP$(4,3)$ method outperforms the SSP$(3,3)$ on twenty-five cardiac cell models. This was because of the improvements of the time step sizes compensate for the extra costs it takes per step of the SSP$(4,3)$ methods. However, the SSP$(3,3)$ method outperformed the SSP$(4,3)$ method on twelve cardiac cell models. This is because the larger theoretical stepsize of the SSP$(3,3)$ method produces a more accurate solution than the SSP$(4,3)$ method.

5. Figure 4.1 shows that the maximum of stable stepsizes gives us the theoretical largest stepsize. These stable stepsizes are computed by scaling all calculated time-sampled eigenvalues with non-positive real parts for the FHN cell model to lie inside the stability region of the FE method.

   From section 3.2, we calculate the stable stepsizes by scaling each eigenvalue to the boundary of the stability region of the time integration method. The minimum of all stable stepsizes gives us the largest theoretical stepsize. This largest theoretical stepsize brings all eigenvalues with a non-positive real part inside and on the stability region of the time integration method, and the numerical solution does not diverge. However, for the FHN model, the maximum of all stable stepsizes keep all eigenvalue inside the stability region of the FE method (see Figure 4.1) because all eigenvalues with non-positive real parts lie inside the stability region of the FE method. Therefore, we use the following steps to determine the largest theoretical stepsize:

Step 1: First, we must check whether or not all eigenvalues with non-positive real parts computed from the cell model lie inside the stability region of the time integration method. This is because the stability region of each time integration method is not the same.

Step 2: We calculate the stable stepsizes by scaling all eigenvalues with non-positive real parts computed from the cell model to the boundary of the stability region of the time integration method.

Step 3: The maximum scaling factor that brings all eigenvalues with a non-positive real part inside and on the stability region of the time integration method is the theoretical largest stepsize.

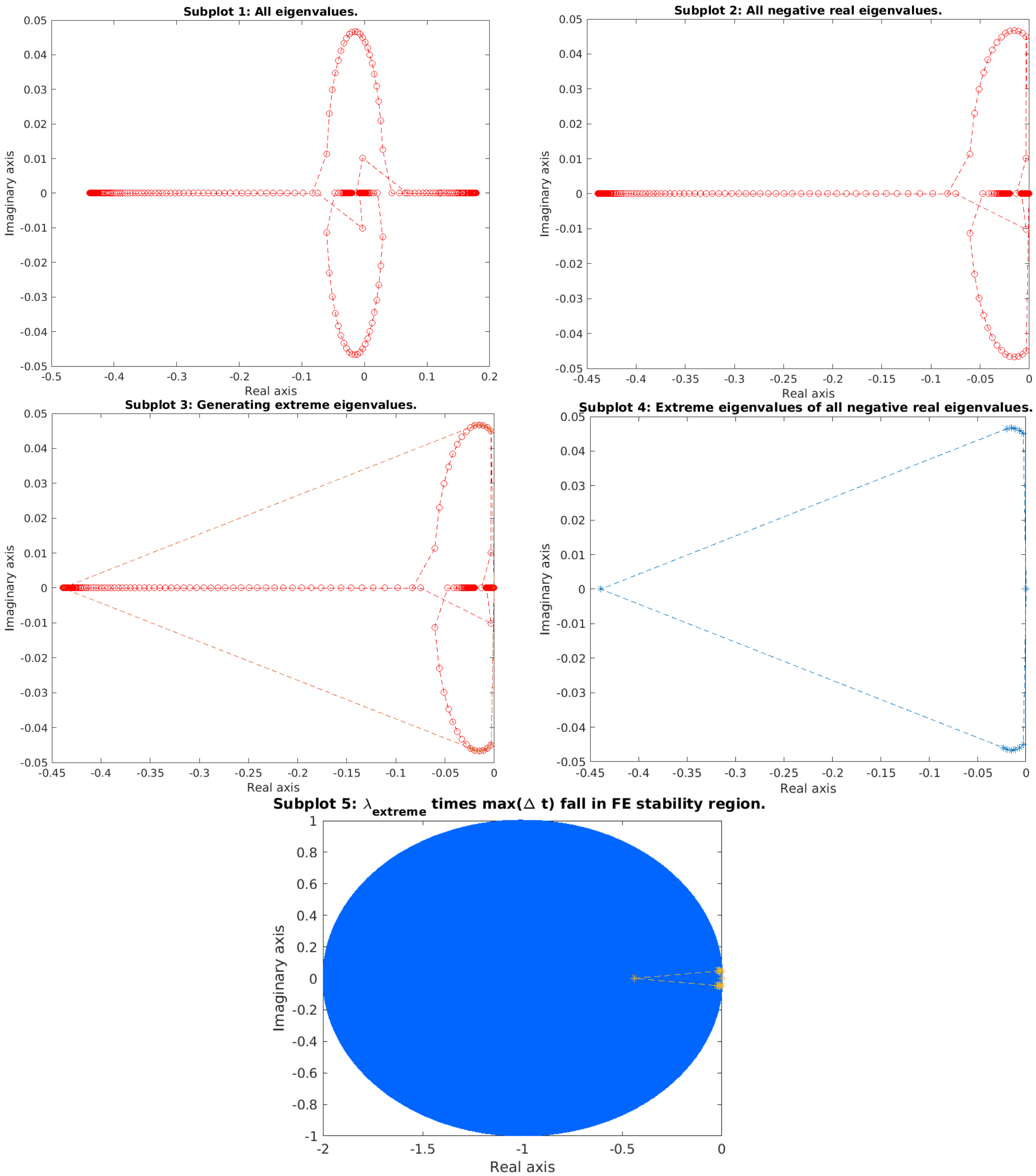


Figure 4.1: Subplot 1 shows all the computed time-sampled eigenvalues spanning over the integration interval for the FHN model. Subplot 2 shows all eigenvalues with non-positive real parts because we do not take eigenvalue with positive real parts for finding the theoretical largest stepsizes. Subplot 3 shows finding a bounding polygon of those eigenvalues. Subplot 4 shows the extreme values of the eigenvalues with non-positive real parts. Subplot 5 shows the maximum of stepsizes (the value of the maximum of all stepsizes is 9.9972e–01) times all extreme eigenvalues ($\lambda_{extreme}$) lie inside the stability region of the FE method (it means, $|1 + \max(\Delta t)\lambda_{extreme}| \leq 1$).

Table 4.2: Δt_{theo} represents the theoretical largest time step and CPU time represents the minimum of fifty times running of the FE method with less than 5% $[\text{MRMS}]_v$.

Model	FE		
	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$
Beeler–Reuler (1977)	2.4388e–02	1.5517e–01	4.8211e–03
Bondarenko et al. (2004)	2.1068e–04	1.3519e+00	1.1229e–03
Courtemanche et al. (1998)	1.5517e–02	4.7248e+00	4.5076e–03
Demir et al. (1994)	5.2293e–05	1.8012e–02	1.7997e–02
Demir et al. (1999)	5.2293e–05	1.6240e–02	2.1097e–02
DiFrancesco–Noble (1985)	7.6176e–05	1.3916e–01	8.7797e–03
Dokos et al. (1996)	6.6838e–05	8.0463e–02	9.6235e–03
FitzHugh-Nagumo (1961)	2.7217e–03	4.7620e–03	5.0000e–02
Faber–Rudy (2000)	1.0874e–02	3.5192e–01	1.6327e–02
Fox et al. (2002)	4.5596e–03	3.3172e–01	5.3733e–03
Hilgemann–Noble (1987)	6.1571e–05	5.6775e–02	4.3602e–03
Hund–Rudy (2004)	7.7965e–03	1.9462e–01	4.8996e–02
Jafri et al. (1998)	4.3300e–04	5.3558e+00	8.0899e–04
Luo–Rudy (1991)	1.3287e–02	9.1163e–02	5.4646e–03
Maleckar et al. (2009)	4.8053e–05	8.7255e–02	1.3231e–03
McAllister et al. (1975)	1.0934e–02	1.2491e–01	2.1598e–03
Noble (1962)	9.4880e–05	4.1900e–03	1.2260e–02
Noble–Noble (1984)	1.6059e–04	4.9362e–02	2.9266e–02
Noble et al. (1991)	5.1412e–05	1.0832e–01	1.7163e–03
Noble et al. (1998)	5.5473e–05	9.7216e–02	6.1817e–03
Nygren et al. (1998)	4.9589e–05	2.4881e–01	3.0497e–03
Pandit et al. (2001))	2.8897e–07	3.3149e+00	1.2442e–04
Pandit et al. (2003)	2.6523e–08	3.3880e+01	6.9372e–03
PuglisiBers	1.0454e–02	7.0989e–01	2.9473e–02
Sakmann et al. (2000) – Endocardial	6.7440e–05	6.5490e–02	3.6162e–02
Sakmann et al. (2000) – Epicardial	6.7000e–05	5.9875e–02	4.1447e–02
Sakmann et al. (2000) – M-cell	6.7049e–05	1.2193e–01	3.5619e–03
Stewart et al. (2009)	1.4459e–02	4.4209e–01	2.7666e–03
Ten Tusscher et al. (2004) –Endocardial	1.7057e–03	1.6552e+00	4.3223e–04
Ten Tusscher et al. (2006) –Endocardial	1.5900e–03	7.5807e–01	9.8490e–04
Ten Tusscher et al. (2004) –Epicardial	1.7055e–03	1.4744e+00	3.8554e–04
Ten Tusscher et al. (2006) –Epicardial	2.1191e–03	6.0086e–01	5.3524e–04
Ten Tusscher et al. (2004) –M-cell	1.7115e–03	1.1081e+00	2.0687e–04
Ten Tusscher et al. (2006) –M-cell	2.0379e–03	9.6348e–01	5.1864e–04
Wang–Sobie (2008)	1.6311e–02	3.3048e–01	7.1810e–03
Winslow et al. (1999) (31 eqn)	1.0400e–04	1.3163e+01	1.4362e–03
Zhang et al. (2000)	9.0103e–05	6.1924e–02	4.9013e–03

4.3.2 Experimental results

We implemented the time integration methods and the experimental largest stepsize techniques from the Chapter 3 on cardiac cell models to produce the experimental largest stepsize, $[\mathrm{MRMS}]_v$, and CPU time. These results are presented in Table 4.3. We now present some analysis based on the results of Table 4.3; see Table A.2 for complete results.

1. The FE method outperforms the RKC$(2,1)$, RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods on all thirty-seven cardiac cell models. The additional stages of the RKC$(2,1)$, RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods add extra costs for each step compared to the FE method. Therefore, the FE method becomes the most efficient method.

2. After the FE method, RKC$(2,1)$ method is the most efficient method because it outperforms the RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods on all thirty-seven cardiac cell models. The additional stages of the RKC$(3,1)$, SSP$(3,3)$, SSP$(4,3)$, and RK4 methods bring extra costs for each step than the RKC$(2,1)$ method. Both the EMP and RKC$(2,1)$ methods have two stages, but the improvements of the time step sizes compensate for the extra cost per step of the RKC$(2,1)$ method.

3. The RKC$(3,1)$ method outperforms the SSP$(3,3)$, SSP$(4,3)$, EMP, and RK4 methods on all thirty-seven cardiac cell models. This was because of improvements of the time step sizes compensate for the extra cost per step of the RKC$(3,1)$ method.

4. The SSP$(4,3)$ method outperforms the SSP$(3,3)$ on thirty-three cardiac cell models. This is because it is able to produce a sufficiently accurate solution with a stepsize that is large enough to compensate for the increase in cost per step. However, the SSP$(3,3)$ method outperform the SSP$(4,3)$ method on three cardiac cell models. This is because SSP$(4,3)$ cannot produce a sufficiently accurate solution with a stepsize that is large enough to compensate for the increase in cost per step.

4.3.3 A comparison of theoretical largest stepsize and experimental largest stepsize of cardiac cell models

In this subsection, we compare the theoretical largest stepsize presented in Table 4.2 (see Table A.1 for complete results) and the experimental largest stepsize results presented in Table 4.3 (see Table A.2 for complete results). To compare, we used relative accuracy of the theoretical largest stepsize to measure how close the theoretical largest stepsize to the experimental largest step size. The formula of relative accuracy of the theoretical largest stepsize is given by

$$\text{Relative accuracy of the theoretical largest stepsize } = \frac{\Delta t_{\text{theo}}}{\Delta t_{\text{exp}}} \times 100\%.$$

Table 4.3: Δt_{exp} represents the experimental largest time step and CPU time represents the minimum of fifty times running of the FE method with less than 5% $[\text{MRMS}]_v$.

Model	FE		
	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$
Beeler–Reuler (1977)	2.5298e–02	2.4791e–02	5.2120e–03
Bondarenko et al. (2004)	2.1300e–04	2.4416e+00	1.1353e–03
Courtemanche et al. (1998)	1.9400e–02	3.0747e–01	6.2901e–03
Demir et al. (1994)	5.9599e–05	1.6659e–02	2.2081e–02
Demir et al. (1999)	5.9607e–05	1.9908e–02	2.4060e–02
DiFrancesco–Noble (1985)	7.7276e–05	8.3383e–02	4.5145e–02
Dokos et al. (1996)	7.0200e–05	8.0562e–02	2.2554e–02
FitzHugh–Nagumo (1961)	2.7200e–03	7.6938e–01	4.9968e–02
Faber–Rudy (2000)	1.1294e–02	1.3211e–01	1.7008e–02
Fox et al. (2002)	4.6200e–03	3.5088e–01	5.4455e–03
Hilgemann–Noble (1987)	6.2500e–05	8.4361e–02	4.7582e–03
Hund–Rudy (2004)	7.8874e–03	2.2220e–01	4.9939e–02
Jafri et al. (1998)	5.7600e–04	4.5764e+00	1.8154e–02
Luo–Rudy (1991)	1.3572e–02	8.5973e–02	5.6083e–03
Maleckar et al. (2009)	5.0200e–05	2.9673e–01	1.6159e–02
McAllister et al. (1975)	2.4700e–02	3.5631e–02	4.9527e–02
Noble (1962)	2.0228e–04	3.0909e–03	4.9586e–02
Noble–Noble (1984)	2.0400e–04	5.7262e–03	3.8972e–02
Noble et al. (1991)	5.1500e–05	1.8201e–02	1.6961e–03
Noble et al. (1998)	5.5600e–05	3.5936e–02	6.1962e–03
Nygren et al. (1998)	5.3589e–05	7.7448e–02	3.0280e–02
Pandit et al. (2001)	2.9100e–07	5.3835e+00	1.2482e–04
Pandit et al. (2003)	2.6523e–08	3.6539e+01	6.9372e–03
PuglisiBers	1.0884e–02	1.9327e–01	3.2953e–02
Sakmann et al. (2000) – Endocardial	6.9000e–05	5.2694e–01	3.7504e–02
Sakmann et al. (2000) – Epicardial	6.8000e–05	2.0159e–01	4.3111e–02
Sakmann et al. (2000) – M-cell	6.8600e–05	3.3911e–02	3.9258e–03
Stewart et al. (2009)	1.5212e–02	4.1142e–01	4.9869e–02
Ten Tusscher et al. (2004) –Endocardial	1.7829e–03	1.1957e+00	1.5526e–03
Ten Tusscher et al. (2006) –Endocardial	1.6200e–03	8.9863e–01	9.9918e–04
Ten Tusscher et al. (2004) –Epicardial	1.7835e–03	1.1548e+00	1.5568e–03
Ten Tusscher et al. (2006) –Epicardial	2.1400e–03	8.6557e–01	5.3646e–04
Ten Tusscher et al. (2004) –M-cell	1.7667e–03	9.0396e–01	4.3927e–04
Ten Tusscher et al. (2006) –M-cell	2.0619e–03	6.6438e–01	4.7968e–04
Wang–Sobie (2008)	1.6611e–02	8.4543e–02	8.4128e–03
Winslow et al. (1999) (31 eqn)	1.0700e–04	1.2650e+01	1.4777e–03
Zhang et al. (2000)	9.9925e–05	7.3992e–02	5.4361e–03

The relative accuracy of the theoretical largest time step of the time integration methods for solving the cardiac cell models presented in Table 4.4; see Table A.3 for complete results. We now present some analysis based on the results of Table 4.4.

1. From Figure 4.2, the FE method has the highest relative accuracy of the theoretical largest stepsize for only 2 cell models.

2. From Figure 4.2, the SSP$(4, 3)$ has the highest relative accuracy of the theoretical largest stepsize for 14 cell models.

3. From the Table 4.4 (see Table A.3 for complete results), the 100 percent relative accuracy of the theoretical largest stepsize value shows that the experimental largest stepsize is the same as the theoretical largest stepsize. This shows that in such cases the experimental largest stepsize can be achieved with a one-time simulation using the theoretical largest stepsize.

4. From Figure 4.2, the higher-order time integration methods are more accurate than the low-order ones in predicting their experimental largest stepsize for solving the cardiac cell models.

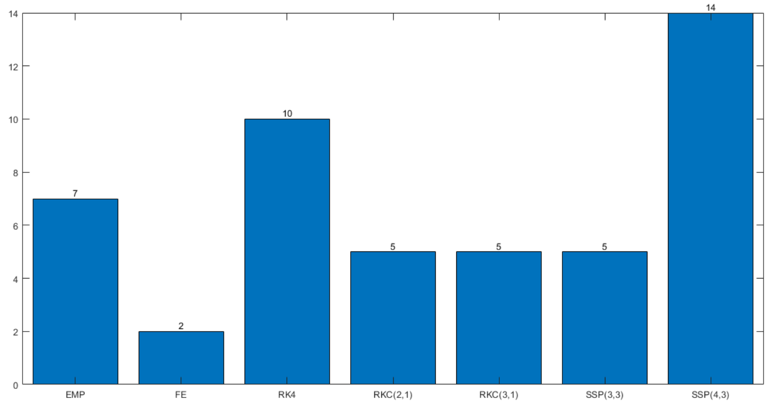


Figure 4.2: The blue bar represents the number of the highest relative accuracy of the theoretical largest stepsize of each time integration method out of 37 cardiac cell models. We compare each time integration method based on the relative accuracy of the theoretical largest stepsize for predicting their experimental largest stepsize for 37 cardiac cell models.

Table 4.4: This table shows the relative accuracy of the theoretical time step to the experimental time step of the FE method. Δt_{theo} represents the theoretical largest time step and Δt_{exp} represents the experimental largest time step.

Model	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of FE
Beeler–Reuler (1977)	96.40
Bondarenko et al. (2004)	98.91
Courtemanche et al. (1998)	79.99
Demir et al. (1994)	87.74
Demir et al. (1999)	87.74
DiFrancesco–Noble (1985)	98.58
Dokos et al. (1996)	95.21
FitzHugh-Nagumo (1961)	100.00
Faber–Rudy (2000)	96.28
Fox et al. (2002)	98.69
Hilgemann–Noble (1987)	98.51
Hund–Rudy (2004)	98.85
Jafri et al. (1998)	75.17
Luo–Rudy (1991)	97.90
Maleckar et al. (2009)	95.72
McAllister et al. (1975)	44.27
Noble (1962)	46.91
Noble–Noble (1984)	78.72
Noble et al. (1991)	99.83
Noble et al. (1998)	99.77
Nygren et al. (1998)	92.54
Pandit et al. (2001)	99.30
Pandit et al. (2003)	100.00
PuglisiBers	96.05
Sakmann et al. (2000) – Endocardial	97.74
Sakmann et al. (2000) – Epicardial	98.53
Sakmann et al. (2000) – M-cell	97.74
Stewart et al. (2009)	95.05
Ten Tusscher et al. (2004) –Endocardial	95.67
Ten Tusscher et al. (2006) –Endocardial	98.15
Ten Tusscher et al. (2004) –Epicardial	95.63
Ten Tusscher et al. (2006) –Epicardial	99.02
Ten Tusscher et al. (2004) –M-cell	96.88
Ten Tusscher et al. (2006) –M-cell	98.84
Wang–Sobie (2008)	98.19
Winslow et al. (1999) (31 eqn)	97.20
Zhang et al. (2000)	90.17

4.4 Monodomain simulation

In this section, we examine the FE, the RKC$(2,1)$, and the RKC$(3,1)$ methods using the monodomain model coupled with the FHN model. We compare these methods based on their theoretical largest stepsizes, experimental largest stepsizes, CPU time, and $[\text{MRMS}]_v$.

4.4.1 One-dimensional experiment

The monodomain model was simulated in one dimension with the FHN model [22]. The model of FHN has 2 cellular state variables. The FHN model is

$$\frac{\partial s}{\partial t} = f(t, v, s), \tag{4.1a}$$

$$\frac{\partial v}{\partial t} = \frac{1}{C_{me}} I_{ion}(v, s), \tag{4.1b}$$

where $f(t, v, s) = -k(v - \text{Vrest})(s + (v - \text{Vthreshold})(v - \text{Vpeak})) - \text{Istim}$ and $I_{ion}(t, v, s) = l(v - \text{Vrest}) - bs$. The values of the parameters of the FHN model listed in Table 4.5.

Table 4.5: Parameters of the FHN model.

C_{me}	1
k	0.00004
Vrest	-85
Vthreshold	-70
Vpeak	40
l	0.63
b	0.013
Istim	If time ≥ 0 and time ≤ 0.5, then Istim $= 80$. else Istim=0.

We implemented the FE, RKC(2,1), and RKC(3,1) methods presented in Subsection 3.4.2, Subsection 3.4.3, Subsection 3.4.4, and the theories presented in Section 3.2 and Subsection 3.4.7 on the monodomain problem (3.22) coupled with systems of ODEs (4.1) to produce the theoretical largest stepsizes, the experimental largest stepsizes, and MRMS error. The initial conditions used on the domain $[0, 1]$ are: $v(x, 0) = \sin(\pi x)$ and $\boldsymbol{s}(x, 0) = \boldsymbol{s_0}$, where $\boldsymbol{s_0}$ is is the default resting state value for the FHN model.

From Table 4.6a and Table 4.6b, the theoretical largest stepsize of the FE, RKC$(2,1)$, and RKC$(3,1)$ methods are 99%, 97%, and 94% of their corresponding experimental largest stable time step sizes for solving the monodomain model coupled with the FHN model. From these results, we conclude that the theoretical method to find the theoretical largest stable stepsize is quite effective at avoiding trial-and-error to get the experimental largest stepsize.

As we can see from Table 4.6a and Table 4.6b, the RKC$(3,1)$ method outperforms the FE and RKC$(3,1)$ methods. Also, RKC$(2,1)$ method outperforms the FE method. These results were because of the improve-

Table 4.6: Results of 1D monodomain model simulation.

	Δt_{theo} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	2.4510e–02	6.0322e–03	1.5506e–02
$\text{RKC}(2,1)$	9.5691e–02	2.8909e–02	6.2357e–03
$\text{RKC}(3,1)$	1.9702e–01	3.6950e–02	4.7593e–03

(a) Δt_{theo} represents the theoretical largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, $\text{RKC}(2,1)$, and $\text{RKC}(3,1)$ methods for solving the monodomain model coupled with the FHN model.

	Δt_{exp} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	2.4777e–02	4.9619e–02	1.5476e–02
$\text{RKC}(2,1)$	9.9115e–02	4.9951e–02	5.9370e–03
$\text{RKC}(3,1)$	2.0999e-01	4.9999e-02	4.4740e-03

(b) Δt_{exp} represents the experimental largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, $\text{RKC}(2,1)$, and $\text{RKC}(3,1)$ methods for solving the monodomain model coupled with the FHN model.

ments of the time step size through the multistage $\text{RKC}(2,1)$ and $\text{RKC}(3,1)$ methods. These multistage methods extend the real stability domain with a length proportional to the square of their number of stages. The time step size of $\text{RKC}(2,1)$ and $\text{RKC}(3,1)$ methods are 3.9 and 8.0 times the time step size of the FE method respectively. The improvements of these time step sizes compensate for the extra costs taken per step of $\text{RKC}(2,1)$ and $\text{RKC}(3,1)$ methods. From this, we conclude that $\text{RKC}(2,1)$ and $\text{RKC}(3,1)$ methods obtain advantages over FE method. This in the future might lead us toward the real-time simulation of the electrical simulation of the heart.

4.4.2 Two-dimensional experiment

The monodomain model was simulated in two dimensions with the FHN model [22]. We implemented the FE, RKC(2,1), and RKC(3,1) methods presented in Subsection 3.4.2, Subsection 3.4.3, Subsection 3.4.4 and the theories presented in Section 3.2 and Subsection 3.4.7 on the monodomain problem (3.22) coupled with systems of ODEs (4.1) to produce the theoretical largest stepsizes, the experimental largest stepsizes, and MRMS error. The initial conditions used on the domain $[0,1] \times [0,1]$ are: $v(x,y,0) = \sin(\pi x)\sin(\pi y)$ and $\boldsymbol{s}(x,y,0) = \boldsymbol{s_0}$, where $\boldsymbol{s_0}$ is is the default resting state value for the FHN model.

From Table 4.7a and Table 4.7b, the theoretical largest stable time step size of the FE, $\text{RKC}(2,1)$,

Table 4.7: Results of 2D monodomain model simulation.

	Δt_{theo} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	1.2255e–02	1.2204e–03	2.1669e–01
RKC(2, 1)	4.7846e–02	1.0587e–02	9.4249e–02
RKC(3, 1)	1.0716e–01	1.5131e–02	6.4316e–02

(a) Δt_{theo} represents the theoretical largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, RKC(2, 1), and RKC(3, 1) methods for solving the monodomain model coupled with the FHN model.

	Δt_{exp} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	1.2343e–02	4.9820e–02	2.1638e–01
RKC(2, 1)	4.9373e–02	4.9956e–02	9.2007e–02
RKC(3, 1)	1.1113e–01	4.9989e–02	6.3118e–02

(b) Δt_{exp} represents the experimental largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, RKC(2, 1), and RKC(3, 1) methods for solving the monodomain model coupled with the FHN model.

and RKC(3, 1) methods are 99%, 97%, and 96% of their corresponding experimental largest stepsizes for solving the monodomain model coupled with the FHN model. From these results, we conclude that the theoretical method to find the theoretical largest stepsize can be used to largely avoid trial-and-error to get the experimental largest stepsize.

As we can see from Table 4.7a and Table 4.7b, RKC(3, 1) method outperforms FE and RKC(2, 1) methods. Also, RKC(2, 1) method outperforms FE method. These results were because of the improvements of the time step size through the multistage RKC(2, 1) and RKC(3, 1) methods. These multistage methods extend the real stability domain with a length proportional to the square of their number of stages. The time step size of RKC(2, 1) and RKC(3, 1) methods are 3.9 and 8.7 times the time step size of the FE method respectively. The improvements of these time step sizes compensate for the extra costs taken per step of RKC(2, 1) and RKC(3, 1) methods. From this, we conclude that RKC(2, 1) and RKC(3, 1) methods obtain advantages over the FE method. This in the future might lead us toward the real-time simulation of the electrical simulation of the heart.

4.4.3 Three-dimensional experiment

The monodomain model was simulated in three dimensions with the FHN model [22]. We implemented the FE, RKC(2,1), and RKC(3,1) methods presented in Subsection 3.4.2, Subsection 3.4.3, Subsection 3.4.4, and the theories presented in Section 3.2 and Subsection 3.4.7 on the monodomain problem (3.22) coupled with systems of ODEs (4.1) to produce the theoretical largest stepsizes, the experimental largest stepsizes, and MRMS error. The initial conditions used on the domain $[0,1] \times [0,1] \times [0,1]$ are: $v(x,y,z,0) = \sin(\pi x)\sin(\pi y)\sin(\pi z)$ and $s(x,y,z,0) = \boldsymbol{s}_0$, where $\boldsymbol{s_0}$ is is the default resting state value for the FHN model.

Table 4.8: Results of 3D monodomain model simulation.

	Δt_{theo} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	8.1699e–03	2.5750e–03	2.2806e+02
RKC$(2,1)$	3.1897e–02	4.3865e–03	1.1005e+02
RKC$(3,1)$	7.1438e–02	1.4979e–03	7.5010e+01

(a) Δt_{theo} represents the theoretical largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, RKC$(2,1)$, and RKC$(3,1)$ methods for solving the monodomain model coupled with the FHN model.

	Δt_{exp} (ms)	$[\text{MRMS}]_{\text{v}}$	CPU time (s)
FE	8.2185e–03	4.9325e–02	2.1655e+02
RKC$(2,1)$	3.2875e–02	4.9326e–02	1.0634e+02
RKC$(3,1)$	7.3984e–02	4.9479e–02	6.9984e+01

(b) Δt_{exp} represents the experimental largest stepsize used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$, and the CPU time represents the minimum time running of the FE, RKC$(2,1)$, and RKC$(3,1)$ methods for solving the monodomain model coupled with the FHN model.

From Table 4.8a and Table 4.8b, the theoretical largest stepsize of the FE, RKC$(2,1)$, and RKC$(3,1)$ methods are 99%, 97%, and 97% of their corresponding experimental largest stepsizes for solving the monodomain model coupled with the FHN model. From these results, we conclude that the theoretical method to find the theoretical largest stepsize can be used to largely avoid trial-and-error to get the experimental largest sepsize.

As we can see from Table 4.8a and Table 4.8b, RKC$(3,1)$ method outperforms FE and the RKC$(2,1)$ methods. Also, RKC$(2,1)$ method outperforms the FE method. These results were because of the improvements of the time step size through the multistage RKC$(2,1)$ and RKC$(3,1)$ methods. These multistage methods extend the real stability domain with a length proportional to the square of their number of stages. The time step size of RKC$(2,1)$ and RKC$(3,1)$ methods are 3.9 and 8.7 times of the time step size of the FE method respectively. The improvements of the time step sizes compensate for the extra costs taken per step of the RKC$(2,1)$ and RKC$(3,1)$ methods. From this, we conclude that the RKC$(2,1)$ and RKC$(3,1)$ methods obtain advantages over FE method. This in the future might lead us toward the real-time simulation of the electrical simulation of the heart.

4.5 Spiral and scroll wave simulation in the monodomain model

A spiral wave is one of the types of traveling waves that circulate the core in a spiral [9]. Spiral waves occur in many excitable and oscillatory systems [15, 33]. Scroll waves are the three-dimensional extensions of spiral waves. They rotate away from the curves of the filaments. Ventricular tachycardia and ventricular fibrillation are two of the causes of unexpected cardiac death. These are related to the creation of spiral and scroll wave of electrical activation in cardiac tissue. To simulate a spiral and scroll waves, we use the modified FHN model [67] coupled with the monodomain model.

4.5.1 Spiral wave simulation

The modified FHN model is

$$\frac{\partial s}{\partial t} = f(t, v, s), \tag{4.2}$$

$$\frac{\partial v}{\partial t} = \frac{1}{C_{me}} I_{ion}(v, s), \tag{4.3}$$

where $C_{me} = 1$, $f(t, v, s) = \epsilon(\beta v - \sigma - \gamma s)$ with $\epsilon = 0.01,\ \beta = 0.5,\ \gamma = 1$, and $I_{ion}(t, v, s) = v(1-v)(v-a)-s$ with $a = 0.1$. This model is coupled with the monodomain model and uses the following initial conditions [67] on the square region $[0, 2.5] \times [0, 2.5]$ to initiate a spiral wave

$$v(x, y, 0) = \begin{cases} 1, & 0 < x < 1.25, \quad 0 < y < 1.25, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$s(x, y, 0) = \begin{cases} 0.1, & 0 < x \leq 1.25, \quad 1.25 \leq y < 2.5, \\ 0.1, & 1.25 \leq x < 2.5, \quad 1.25 \leq y < 2.5, \\ 0, & \text{otherwise}. \end{cases}$$

The following time interval $[0, 1000]$ is selected because it shows the full propagation of a spiral wave [67]. To begin a spiral wave, we applied external potential $v(x, y, 0)$ at $t = 0$ on the left boundary over bounded region of $x \in [0, 1.25]$ and $y \in [0, 1.25]$. To get the numerical solution of Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.8, Figure 4.9, Figure 4.10, Figure 4.11, Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16, we implemented the FE. RKC$(2, 1)$, RKC$(3, 1)$ methods and the theoretical largest stepsizes of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods (see Figure 4.9) on the monodomain model coupled with the modified FHN model. As we can see from Figure 4.3, Figure 4.5, Figure 4.8, Figure 4.10, Figure 4.13 and Figure 4.15, a single spiral wave and a clockwise runs of spiral wave are seen at $t = 169.3$, $t = 966.0$, $t = 169.4$, $t = 966.1$, $t = 169.5$ and $t = 966.3$, respectively.

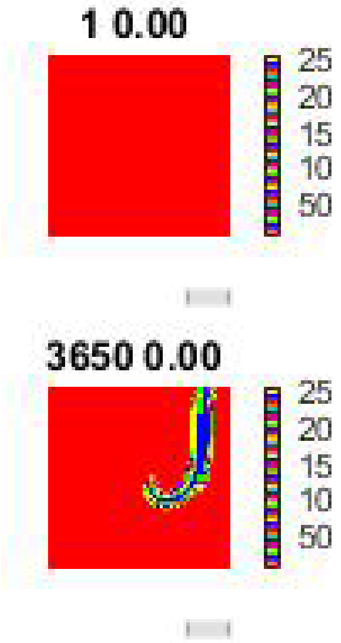


Figure 4.3: $t = 0$, and $t = 169.3$.

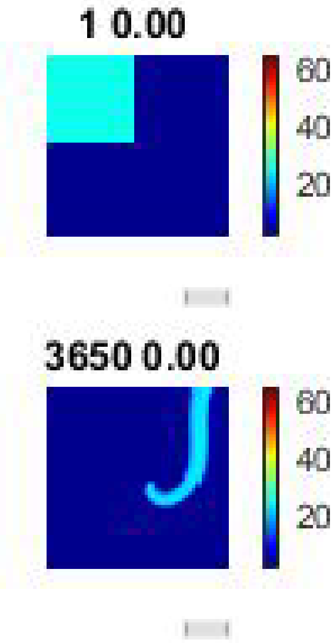


Figure 4.4: $t = 0.0$, and $t = 169.3$.

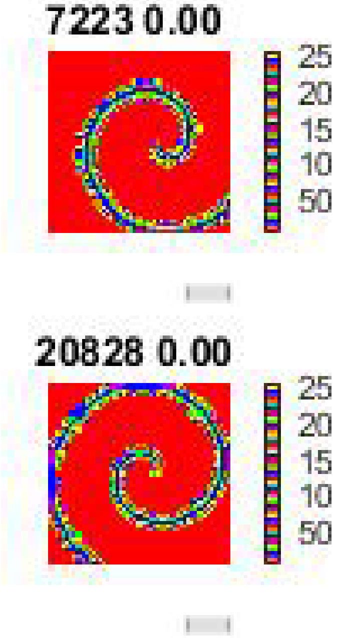


Figure 4.5: $t = 335.0$, and $t = 966.0$.

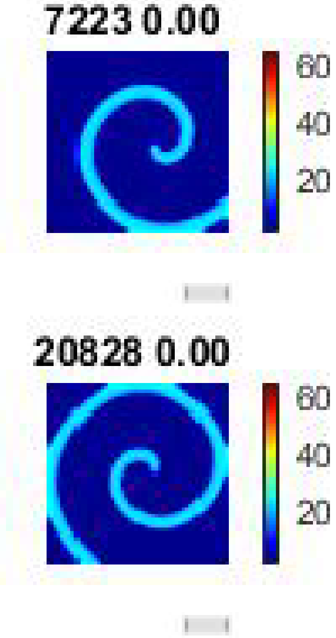


Figure 4.6: $t = 335.0$, and $t = 966.0$.

Figure 4.7: The figures show the generation of a spiral wave. The numerical solution of Figure 4.3, Figure 4.4, Figure 4.5, and Figure 4.6 are obtained by using the FE method. Figure 4.3 and Figure 4.5 show the value of v at different time levels. Figure 4.4 and Figure 4.6 show the value of s at different time levels.

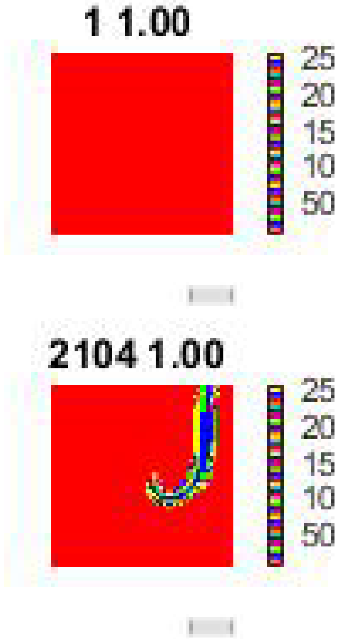


Figure 4.8: $t = 0$, and $t = 169.4$.

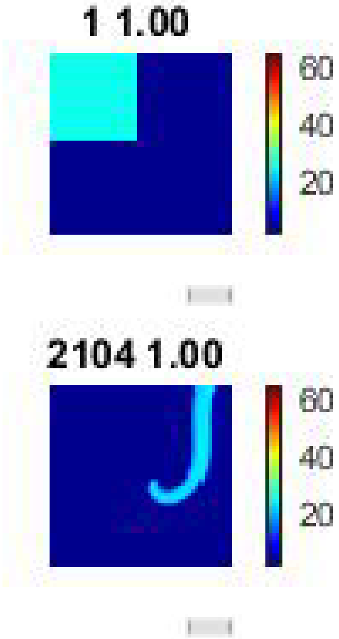


Figure 4.9: $t = 0.0$, and $t = 169.4$.

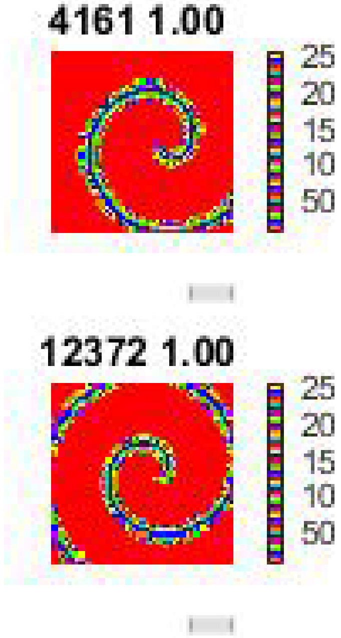


Figure 4.10: $t = 335.0$, and $t = 966.1$.

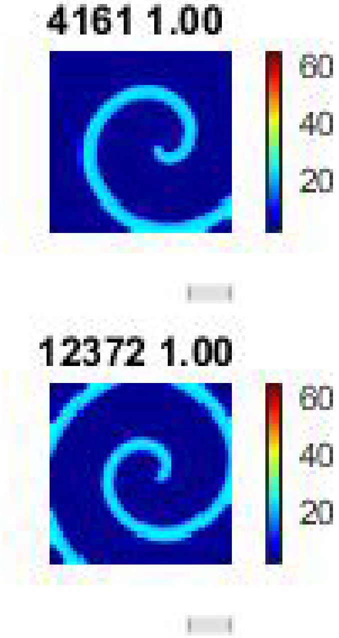


Figure 4.11: $t = 335.0$, and $t = 966.1$.

Figure 4.12: The figures show the generation of a spiral wave. The numerical solution of Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11 are obtained by using the RKC$(2, 1)$ method. Figure 4.8 and Figure 4.10 show the value of v at different time levels. Figure 4.9 and Figure 4.11 show the value of s at different time levels.

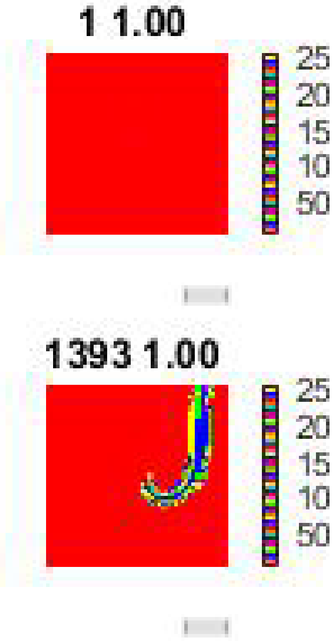


Figure 4.13: $t = 0,$ and $t = 169.5$.

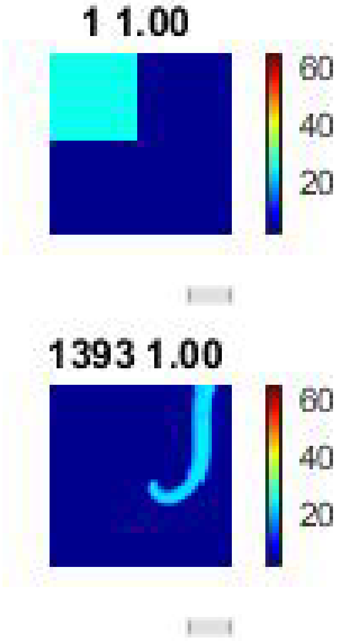


Figure 4.14: $t = 0.0,$ and $t = 169.5$.

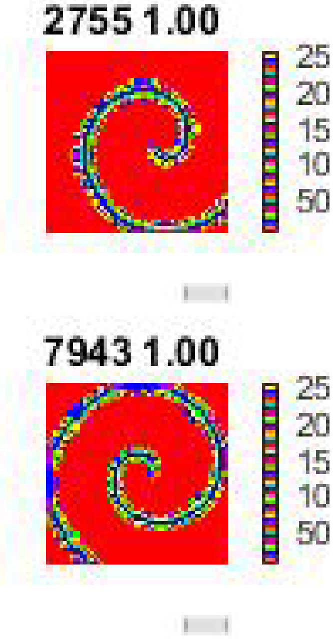


Figure 4.15: $t = 335.2,$ and $t = 966.3$.

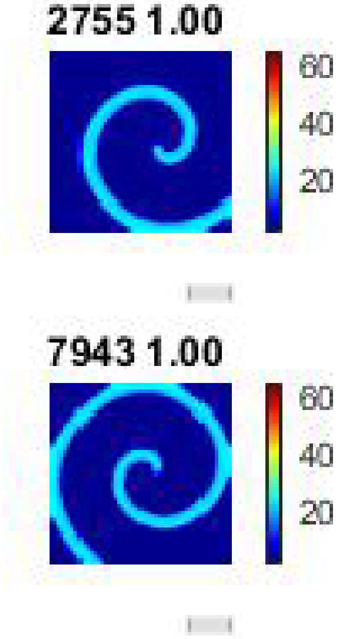


Figure 4.16: $t = 335.2,$ and $t = 966.3$.

Figure 4.17: The figures show the generation of a spiral wave. The numerical solution of Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16 are obtained by using the RKC$(3,1)$ method. Figure 4.13 and Figure 4.15 show the value of v at different time levels. Figure 4.14 and Figure 4.16 show the value of s at different time levels.

	FE	RKC(2, 1)	RKC(3, 1)
Δt_{theo} (ms)	4.6382e–02	8.0511e–02	1.2165e–01
Δt_{exp} (ms)	5.2190e–02	1.2108e–01	1.3600e–01

Table 4.9: Δt_{theo} and Δt_{exp} represent the theoretical largest time step and the experimental largest time step used to get the numerical solutions with less than 5% $[\text{MRMS}]_v$ of the FE, the RKC$(2, 1)$, and the RKC$(3, 1)$ methods.

All eigenvalues computed from the modified FHN cell model and the PDE equation (3.22) lie inside the stability regions of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods. To compute theoretical largest stepsizes (see Table 4.9), we used the technique described in Subsection 4.3.1.

From Table 4.9, the theoretical largest stable time step size of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods are 89%, 67%, and 90% of their corresponding experimental largest stepsizes for solving the monodomain model coupled with the modified FHN model. From these results, we conclude that the theoretical method to find the theoretical largest stepsize can be used to largely avoid trial-and-error to get the experimental largest stepsize.

4.5.2 Scroll wave simulation

We use the following modified FHN model

$$\frac{\partial s}{\partial t} = f(t, v, s), \tag{4.4}$$

$$\frac{\partial v}{\partial t} = \frac{1}{C_{me}} I_{ion}(v, s), \tag{4.5}$$

where $C_{me} = 1$, $f(t, v, s) = \epsilon(\beta v - \sigma - \gamma s)$ with $\epsilon = 0.01$, $\beta = 0.5$, $\gamma = 1$, and $I_{ion}(t, v, s) = v(1-v)(v-a)-s$ with $a = 0.1$. This model coupled with the monodomain model and use the following initial conditions on the cubic region $[0, 2.5] \times [0, 2.5] \times [0, 2.5]$ to initiate a scroll wave

$$v(x, y, z, 0) = \begin{cases} 1, & 0 < x < 1.25, \quad 0 < y < 1.25, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$s(x, y, z, 0) = \begin{cases} 0.1, & 0 < x \leq 1.25, \quad 1.25 \leq y < 2.5, \\ 0.1, & 1.25 \leq x < 2.5, \quad 1.25 \leq y < 2.5, \\ 0, & \text{otherwise}. \end{cases}$$

The following time interval $[0, 1000]$ is selected because it shows the full propagation of a scroll wave.

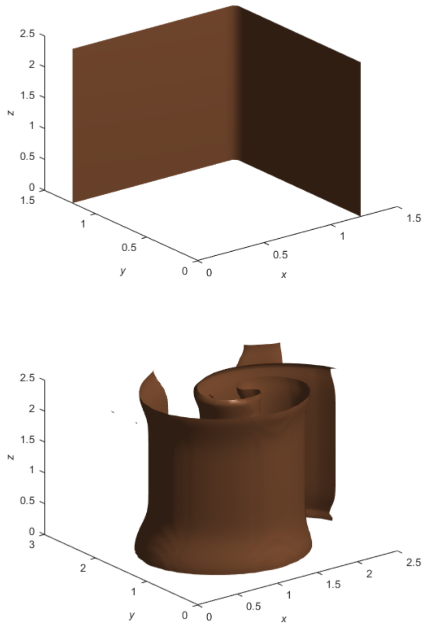


Figure 4.18: $t = 0,$ and $t = 252.9.$

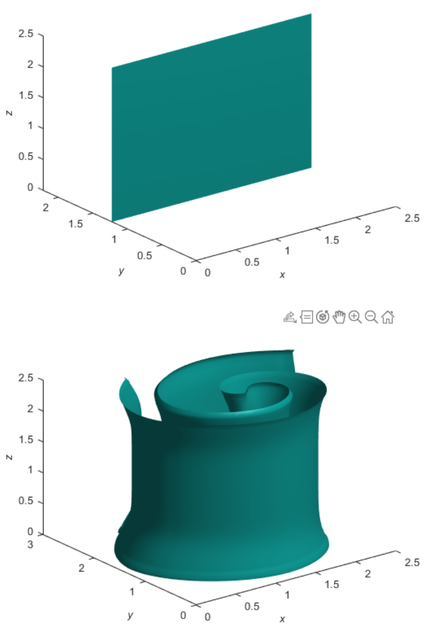


Figure 4.19: $t = 0,$ and $t = 252.9.$

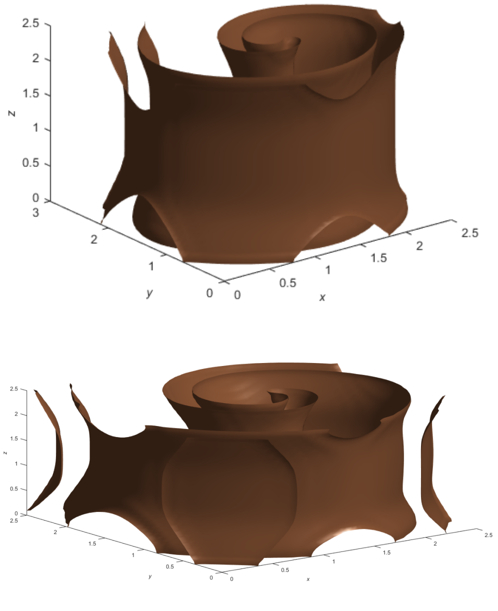


Figure 4.20: $t = 679.7,$ and $t = 999.9.$

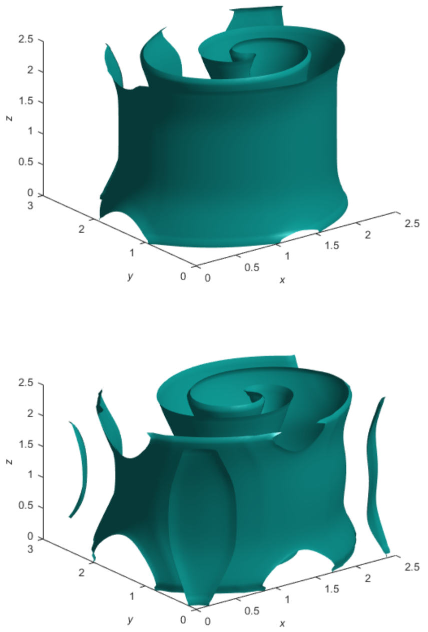


Figure 4.21: $t = 679.7,$ and $t = 999.9.$

Figure 4.22: The figures show the generation of a scroll wave. The numerical solution of Figure 4.18, Figure 4.20, Figure 4.19, and Figure 4.21 are obtained by using the FE method. Figure 4.18 and Figure 4.20 show the value of v at different time levels. Figure 4.19 and Figure 4.21 show the value of s at different time levels.

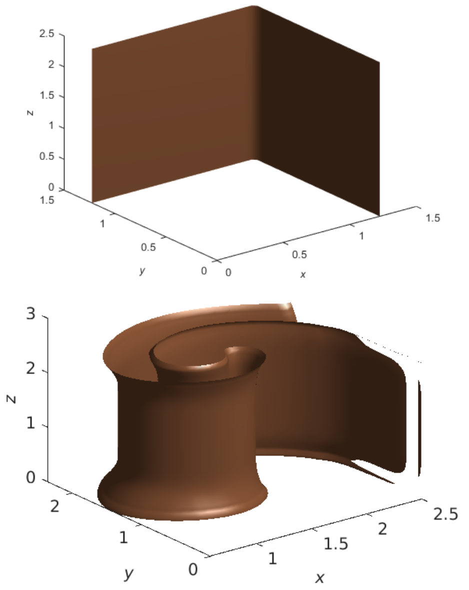


Figure 4.23: $t = 0,$ and $t = 242.6.$

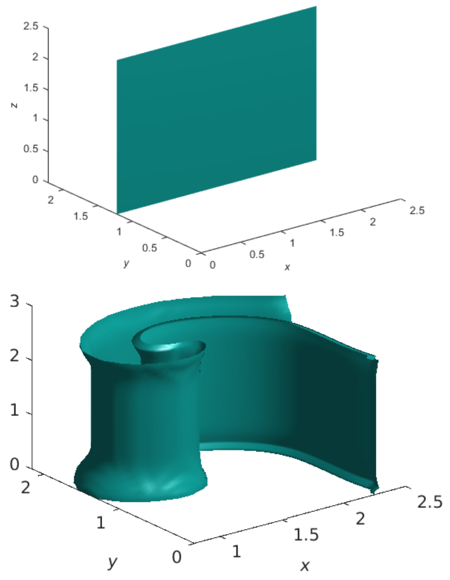


Figure 4.24: $t = 0,$ and $t = 242.6.$

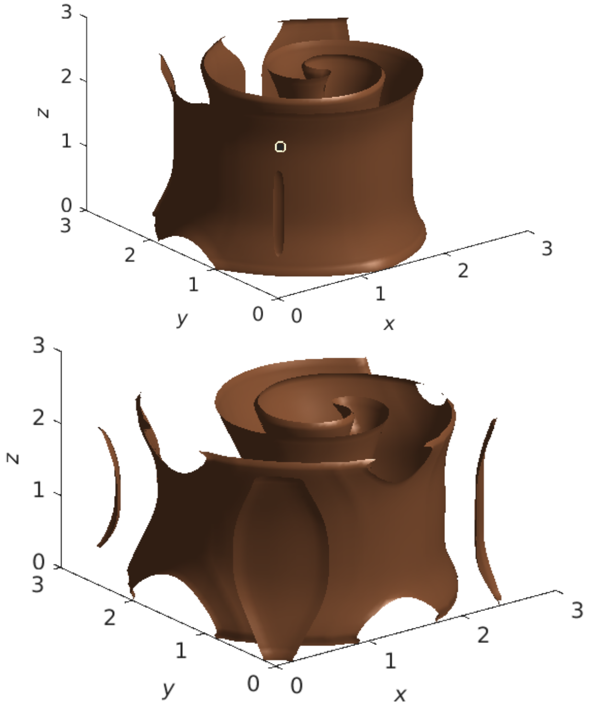


Figure 4.25: $t = 727.8,$ and $t = 999.2.$

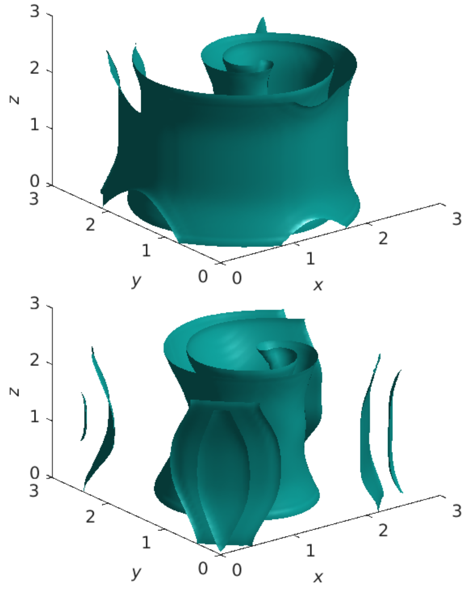


Figure 4.26: $t = 727.8,$ and $t = 999.2.$

Figure 4.27: The figures show the generation of a scroll wave. The numerical solution of Figure 4.23, Figure 4.25, Figure 4.24, and Figure 4.26 are obtained by using the RKC(2, 1) method. Figure 4.23 and Figure 4.25 show the value of v at different time levels. Figure 4.24 and Figure 4.26 show the value of s at different time levels.

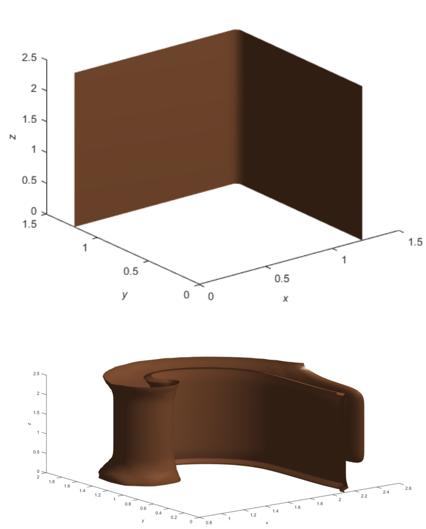


Figure 4.28: $t = 0,$ and $t = 251.6.$

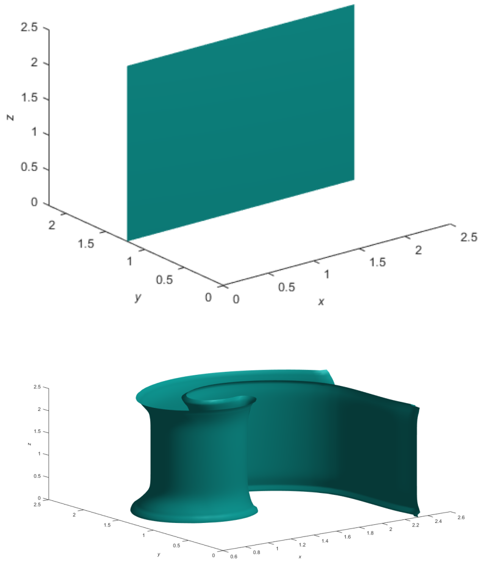


Figure 4.29: $t = 0,$ and $t = 251.6.$

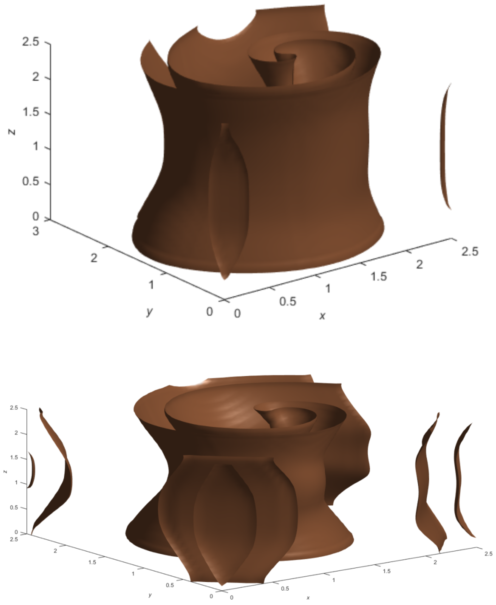


Figure 4.30: $t = 880.7,$ and $t = 1000.$

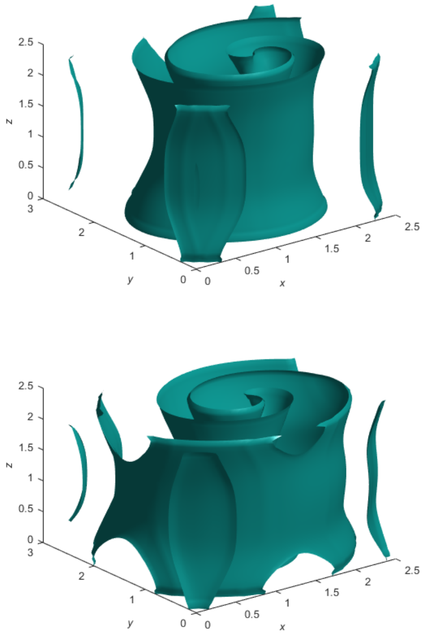


Figure 4.31: $t = 880.7,$ and $t = 1000.$

Figure 4.32: The figures show the generation of a scroll wave. The numerical solution of Figure 4.28, Figure 4.29, Figure 4.30, and Figure 4.31 are obtained by using the RKC$(3, 1)$ method. Figure 4.28 and Figure 4.30 show the value of v at different time levels. Figure 4.29 and Figure 4.31 show the value of s at different time levels.

To create a scroll wave in three dimensions, we used $[0, 2.5] \times [0, 2.5] \times [0, 2.5]$ spatial domain over $[0, 1000]$ time interval. To begin a scroll wave, we applied external potential $v(x, y, z, 0)$ at $t = 0$ on the left boundary over bounded region of $x \in [0, 1.25]$, $y \in [0, 1.25]$, and $z \in [0, 2.5]$. We can see single scroll wave and a clockwise runs of scroll wave from Figure 4.20, Figure 4.19, Figure 4.21, Figure 4.25, Figure 4.24, Figure 4.26, Figure 4.29, Figure 4.30, and Figure 4.31, respectively.

	FE	RKC$(2, 1)$	RKC$(3, 1)$
Δt_{theo} (ms)	1.0671e+00	1.4356e+00	1.3980e+00

Table 4.10: Δt_{theo} represents the theoretical largest time step of the FE, the RKC$(2, 1)$, and the RKC$(3, 1)$ methods. We did not check whether the numerical solutions produced with the largest stepsize of the FE, the RKC$(2, 1)$, and the RKC$(3, 1)$ satisfied the acceptable $[\text{MRMS}]_{\text{v}}$ because we were unable to compute a reference solution.

We computed the theoretical largest step sizes of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods (see Table 4.10) using the eigenvalues of only the PDE (3.22) because the largest eigenvalues come from this PDE. This allows us to avoid computing reference solutions to find eigenvalues of the FHN cell models. Because all eigenvalues lie inside the stability regions of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods, we used the technique described in Subsection 4.3.1 to compute the theoretical largest stepsizes (see Table 4.10) of the FE, RKC$(2, 1)$, and RKC$(3, 1)$ methods. From Table 4.10, we observe that the theoretical largest stepsize for RKC$(3, 1)$ is smaller than that for RKC$(2, 1)$. This result might be due to the approximation of not considering the eigenvalues from the FHN cell model.

5 Conclusion and Future Work

Cardiac simulation is essential for studying heart development and disorder. Computer simulations of the electrical activity in the heart may help electrophysiologists to treat a patient's heart condition with better techniques and diagnosis.

The contributions of this thesis can be summarized as follows:

1. We compared the performance of seven time integration methods for solving thirty-seven cardiac cell models. From the theoretical largest stepsizes and the experimental largest stepsizes results listed in Subsection 4.3.1 and Subsection 4.3.2, the FE method outperformed the RKC$(2, 1)$, the RKC$(3, 1)$, the SSP$(3, 3)$, the SSP$(4, 3)$, the EMP, and the RK4 methods on all cardiac cell models tested.

2. We determined the largest stepsize of the time integration method based on its stability region and the eigenvalues computed from the Jacobian matrix of a system of ODEs. To the best of our knowledge, this is a new methodology to find the largest stepsize. From the results listed in Subsection 4.4.1, Subsection 4.4.2, and Subsection 4.4.3, the theoretical method used to find the theoretical largest stepsize of the time integration methods with an average of the ratio of predicted to experimental is 96%. This method avoids trial-and-error to find the experimental largest stepsize.

3. We solved the monodomain model coupled with the FHN model using FE, RKC$(2, 1)$, and RKC$(3, 1)$ in one dimensions, two dimensions, and three dimensions. The RKC$(3, 1)$ method outperforms the FE and RKC$(2, 1)$ methods for all problems. The improvements of the time stepsizes more than compensate for the extra costs it takes per step of the RKC$(3, 1)$ method. From this, we conclude that the RKC$(3, 1)$ method has advantages over the FE and RKC$(2, 1)$ methods. This is a promising result. This in the future might lead us toward the real-time simulation of the electrical stimulation of the heart.

4. Using the largest stepsize of the FE method, we simulated the monodomain model coupled with the modified FHN model and captured the excitation of spiral and scroll waves propagation in cardiac tissue.

   Possible future work includes the following:

   1. It is of interest to investigate the performance of the SSP$(s, 1)$ methods for $s > 1$ for solving the cardiac cell models and the monodomain model. From results listed in Subsection 4.3.1 and Subsection 4.3.2, the RKC$(2, 1)$ and RKC$(3, 1)$ were the second-most and third-most efficient methods for solving the cardiac cell models. From results listed in Subsection 4.4.1, Subsection 4.4.2, and

Subsection 4.4.3, however, the RKC$(3, 1)$ and RKC$(2, 1)$ were the most and second-most efficient methods for solving the monodomain model. From results listed in Subsection 4.3.1 and Subsection 4.3.2, the SSP$(4, 3)$ method outperformed the SSP$(3, 3)$ method on twenty-five and thirty-three cardiac cell models, respectively. These results show that increasing the number of stages of the SSP method might benefit the simulation efficiency of some cardiac cell models and the monodomain model.

2. The space-fractional monodomain model has an excellent performance in describing the excitation of the electrical potential wave propagation in heterogeneous heart tissue [14, 69]. An efficient higher-order numerical method should appropriately describe the excitation of the electrical potential wave propagation in the heart tissue in high resolution. To solve this model, we plan to implement the high-order discontinuous Petrov–Galerkin method [7, 32] in time and the high-order continuous Ritz–Galerkin method [20] in space.

References

[1] Cardiovascular diseases (cvds). https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). [Online; accessed January 13, 2020].

[2] The CellML Project. `http://cellml.org/`.

[3] Class page - 8th grade science circo-respiratory systems. https://www.quia.com/pages/mcdsscience/8th, 2015. [Online; accessed January 10, 2020].

[4] Illustration of a neon purple coronavirus cell in the center of a heart outline. https://www.yahoo.com/news/covid-19-study-suggests-lasting-003754956.html, 2020. [Online; accessed August 18, 2020].

[5] Cardiac action potential, https://en.wikipedia.org/wiki/Cardiac_action_potential. [Online; accessed July 2, 2021].

[6] Kronecker product, https://en.wikipedia.org/wiki/Kronecker_product. [Online; accessed September 3, 2020].

[7] Petrov–galerkin method, https://en.wikipedia.org/wiki/Petrov. [Online; accessed July 4, 2021].

[8] Runge–kutta methods, https://en.wikipedia.org/wiki/Runge–Kutta_methods. [Online; accessed July 9, 2021].

[9] Spiral wave, https://en.wikipedia.org/wiki/Spiral_wave. [Online; accessed July 9, 2021].

[10] G.W. Beeler and H. Reuter. Reconstruction of the action potential of ventricular myocardial fibres. *The Journal of Physiology*, 268(1):177, 1977.

[11] V.E. Bondarenko, G.P. Szigeti, G.C.L. Bett, S. Kim, and R.L. Rasmusson. Computer model of action potential of mouse ventricular myocytes. *American Journal of Physiology-Heart and Circulatory Physiology*, 287(3):H1378–H1403, 2004.

[12] R.L. Burden, J.D. Faires, and A.M. Burden. *Numerical analysis.* Cengage Learning, 2016.

[13] M. Courtemanche, R.J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology-Heart and Circulatory Physiology*, 275(1):H301–H321, 1998.

[14] N. Cusimano and L. Gerardo-Giorda. A space-fractional monodomain model for cardiac electrophysiology combining anisotropy and heterogeneity on realistic geometries. *Journal of Computational Physics*, 362:409–424, 2018.

[15] J.M. Davidenko, A.V. Pertsov, R. Salomonsz, W. Baxter, and J. Jalife. Stationary and drifting spiral waves of excitation in isolated cardiac muscle. *Nature*, 355(6358):349–351, 1992.

[16] S.S. Demir, J.W. Clark, and W.R. Giles. Parasympathetic modulation of sinoatrial node pacemaker activity in rabbit heart: a unifying model. *American Journal of Physiology–Heart and Circulatory Physiology*, 276(6):H2221–H2244, 1999.

[17] S.S. Demir, J.W. Clark, C.R. Murphey, and W.R. Giles. A mathematical model of a rabbit sinoatrial node cell. *American Journal of Physiology-Cell Physiology*, 266(3):C832–C852, 1994.

[18] D. DiFrancesco and D. Noble. A model of cardiac electrical activity incorporating ionic pumps and concentration changes. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 307(1133):353–398, 1985.

[19] S. Dokos, B. Celler, and N. Lovell. Ion currents underlying sinoatrial node pacemaker activity: a new single cell mathematical model. *Journal of Theoretical Biology*, 181(3):245–272, 1996.

[20] A. Ern and J.L. Guermond. *Theory and practice of finite elements.* Springer, 2004.

[21] G.M. Faber and Y. Rudy. Action potential and contractility changes in $[\mathrm{Na}^+]_i$ overloaded cardiac myocytes: a simulation study. *Biophysical Journal*, 78(5):2392–2404, 2000.

[22] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

[23] R.N. Fogoros. How the heart's electrical system works, https://www.verywellhealth.com/cardiac-electrical-system-how-the-heart-beats-1746299. [Online; accessed July 2, 2021].

[24] J.J. Fox, J.L. McHarg, and R.F. Gilmour. Ionic mechanism of electrical alternans. *American Journal of Physiology-Heart and Circulatory Physiology*, 282(2):H516–H530, 2002.

[25] S. Gottlieb, D. Ketcheson, and C. Shu. *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations.* World Scientific Publishing Co., Inc., USA, 2011.

[26] S. Gottlieb, C. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.

[27] D.W. Hilgemann and D. Noble. Excitation-contraction coupling and extracellular calcium transients in rabbit atrium: reconstruction of basic cellular mechanisms. *Proceedings of the Royal Society of London B: Biological Sciences*, 230(1259):163–205, 1987.

[28] T.J. Hund and Y. Rudy. Rate dependence and regulation of action potential and calcium transient in a canine cardiac ventricular cell model. *Circulation Research*, 110(20):3168–3174, 2004.

[29] W. H. Hundsdorfer and J. G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations.* Springer, 2011.

[30] Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2 edition, 2008.

[31] M.S. Jafri, J.J. Rice, and R.L. Winslow. Cardiac Ca^{2+} dynamics: the roles of ryanodine receptor adaptation and sarcoplasmic reticulum load. *Biophysical Journal*, 74(3):1149–1168, 1998.

[32] S. Larsson and Thomeée Vidar. *Partial differential equations with numerical methods.* Springer, 2009.

[33] J Lechleiter, S Girard, E Peralta, and D Clapham. Spiral calcium wave propagation and annihilation in xenopus laevis oocytes. *Science*, 252(5002):123–126, 1991.

[34] D.K.L. Lewis. Covid-19 and heart: what have we learned?, https://www.health.harvard.edu/blog/covid-19-and-the-heart-what-have-we-learned-2021010621603. [Online; accessed July 2, 2021].

[35] D. Longo, A. Fauci, D. Kasper, S. Hauser, J. Jameson, and J. Loscalzo. *Harrison's Principles of Internal Medicine, 18th Edition.* 07 2011.

[36] C. Luo and Y. Rudy. A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction. *Circulation Research*, 68(6):1501–1526, 1991.

[37] C. Luo and Y Rudy. A dynamic model of the cardiac ventricular action potential. i. simulations of ionic currents and concentration changes. *Circulation Research*, 74(6):1071–1096, 1994.

[38] C. Luo and Y Rudy. A dynamic model of the cardiac ventricular action potential. ii. afterdepolarizations, triggered activity, and potentiation. *Circulation Research*, 74(6):1097–1113, 1994.

[39] M.M. Maleckar, J.L. Greenstein, N.A. Trayanova, and W.R. Giles. Mathematical simulations of ligand-gated and cell-type specific effects on the action potential of human atrium. *Progress in Biophysics and Molecular Biology*, 98(2):161–170, 2008.

[40] M.E. Marsh. An assessment of numerical methods for cardiac simulation. Master's thesis, Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, 2012.

[41] M.E. Marsh, S. Torabi, and R.J. Spiteri. The secrets to the success of the Rush–Larsen method and its generalizations. *IEEE Transactions on Biomedical Engineering*, 59(9):2506–2515, 2012.

[42] R.E. McAllister, D. Noble, and R.W. Tsien. Reconstruction of the electrical activity of cardiac purkinje fibres. *The Journal of Physiology*, 251(1):1, 1975.

[43] A.A. Mirin, D.F. Richards, J.N. Glosli, E.W. Draeger, B. Chan, J. Fattebert, W.D. Krauss, T. Oppelstrup, J.J. Rice, J.A. Gunnels, et al. Toward real-time modeling of human heart ventricles at cellular resolution: Multi-hour simulation of drug-induced arrhythmias. Technical report, Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2012.

[44] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[45] S. Niederer, L. Mitchell, N. Smith, and G. Plank. Simulating human cardiac electrophysiology on clinical time-scales. *Front Physiol*, 2:1–7, 01 2011.

[46] D. Noble. A modification of the hodgkin–huxley equations applicable to purkinje fibre action and pacemaker potentials. *The Journal of Physiology*, 160(2):317, 1962.

[47] D. Noble and S.J. Noble. A model of sino-atrial node electrical activity based on a modification of the difrancesco–noble (1984) equations. *Proceedings of the Royal Society of London B: Biological Sciences*, 222(1228):295–304, 1984.

[48] D. Noble, S.J. Noble, G.C.L. Bett, Y.E. Earm, W.K. Ho, and I.K. So. The role of sodium-calcium exchange during the cardiac action potential. *Annals of the New York Academy of Sciences*, 639(1):334–353, 1991.

[49] D. Noble, A. Varghese, P. Kohl, and P. Noble. Improved guinea-pig ventricular cell model incorporating a diadic space, IKr and IKs, and length-and tension-dependent processes. *Canadian Journal of Cardiology*, 14(1):123–134, 1998.

[50] A. Nygren, C. Fiset, L. Firek, J.W. Clark, D.S. Lindblad, R.B. Clark, and W.R. Giles. Mathematical model of an adult human atrial cell. *Circulation Research*, 82(1):63–81, 1998.

[51] S.V. Pandit, R.B. Clark, W.R. Giles, and S.S. Demir. A mathematical model of action potential heterogeneity in adult rat left ventricular myocytes. *Biophysical Journal*, 81(6):3029–3051, 2001.

[52] S.V. Pandit, W.R. Giles, and S.S. Demir. A mathematical model of the electrophysiological alterations in rat ventricular myocytes in type-I diabetes. *Biophysical Journal*, 84(2):832–841, 2003.

[53] J.L. Puglisi and D.M. Bers. LabHEART: an interactive computer model of rabbit ventricular myocyte ion channels and Ca transport. *American Journal of Physiology-Cell Physiology*, 281(6):C2049–C2060, 2001.

[54] B.F.A.S. Sakmann, A.J. Spindler, S.M. Bryant, K.W. Linz, and D. Noble. Distribution of a persistent sodium current across the ventricular wall in guinea pigs. *Circulation Research*, 87(10):910–914, 2000.

[55] Q. Sheng. Solving linear partial differential equations by exponential splitting. *IMA J. Numer. Anal*, 1989.

[56] S. Shenoy and H. Lankarani. Energy absorption of a car roof reinforced with a grid stiffened composite panel in the event of a rollover [electronic resource] /. 07 2020.

[57] C.W. Shu. Total-variation-diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084, 1988.

[58] A. Siefert and F.O. Henkel. Nonlinear analysis of commercial aircraft impact on a reactor building—comparison between integral and decoupled crash simulation. *Nuclear Engineering and Design*, 269:130–135, 04 2014.

[59] D. Singh, A. Akingbola, N. Ross-Ascuitto, and R. Ascuitto. Electrocardiac effects associated with lithium toxicity in children: An illustrative case and review of the pathophysiology. *Cardiology in the young*, 26:1–9, 09 2015.

[60] R.J. Spiteri and S.J. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM Journal on Numerical Analysis*, 40(2):469–491, 2002.

[61] W. Jacob Stanley, F.Oliver Michael, and L. Entman Mark. *Human cardiovascular system.*

[62] P. Stewart, O.V. Aslanidi, D. Noble, P.J. Noble, M.R. Boyett, and H. Zhang. Mathematical models of the electrical action potential of purkinje fibre cells. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1896):2225–2255, 2009.

[63] J. Sundnes, G. T. Lines, and A. Tveito. An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso. *Mathematical Biosciences*, 194:233–248, 2005.

[64] K.H.W.J. Ten Tusscher, D. Noble, P.J. Noble, and A.V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology-Heart and Circulatory Physiology*, 286(4):H1573–H1589, 2004.

[65] K.H.W.J. ten Tusscher and A.V. Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):H1088–H1100, 2006.

[66] L.J. Wang and E.A. Sobie. Mathematical model of the neonatal mouse ventricular action potential. *American journal of physiology. Heart and circulatory physiology*, 294(6):H2565, 2008.

[67] Y. Wang, L. Cai, X. Luo, W. Ying, and H. Gao. Simulation of action potential propagation based on the ghost structure method. *Scientific Reports*, 9(1), 2019.

[68] R.L. Winslow, J. Rice, S. Jafri, E. Marban, and B. O'Rourke. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II. *Circulation Research*, 84(5):571–586, 1999.

[69] W. Yongheng, C. Li, F. Xiaobing, L. Xiaoyu, and G. Hao. A ghost structure finite difference method for a fractional fitzhugh-nagumo monodomain model on moving irregular domain. *Journal of Computational Physics*, 428:110081, 2021.

[70] Haruo Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5):262–268, 1990.

[71] H. Zhang, A.V. Holden, I. Kodama, H. Honjo, M. Lei, T. Varghese, and M.R. Boyett. Mathematical models of action potentials in the periphery and center of the rabbit sinoatrial node. *American Journal of Physiology-Heart and Circulatory Physiology*, 279(1):H397–H421, 2000.

Appendix A

Simulation Results

Table A.1: Δt_{theo} represents the theoretical largest time step, and the CPU time represents the minimum of fifty times running of the FE, the RKC$(2,1)$, the RKC$(3,1)$, the SSP$(3,3)$, the SSP$(4,3)$, the EMP, and the RK4 methods with less than 5% $[\text{MRMS}]_v$.

Model	FE			RKC$(2,1)$			RKC$(3,1)$			SSP$(3,3)$			SSP$(4,3)$			EMP			RK4		
	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{theo} (ms)	CPU time (s)	$[\text{MRMS}]_v$
Beeler–Reuler (1977)	2.4388e–02	1.5517e–01	4.8211e–03	9.5217e–02	3.5663e–01	1.5549e–02	2.1325e–01	3.8467e–01	3.7030e–02	3.0949e–02	1.0088e+00	2.0496e–03	6.2794e–02	9.3206e–01	4.1414e–03	2.4388e–02	6.0067e–01	1.4650e–03	3.3964e–02	1.8402e+00	2.1359e–03
Bondarenko et al. (2004)	2.1068e–04	1.3519e+00	1.1229e–03	8.2252e–04	1.9617e+01	4.4698e–03	1.8422e–03	1.8255e+01	1.0026e–02	2.6735e–04	1.0687e+02	1.5736e–03	5.4244e–04	7.7829e+01	3.1899e–03	2.1068e–04	6.1032e+01	1.2402e–03	2.9340e–04	1.3111e+02	1.7267e–03
Courtemanche et al. (1998)	1.5517e–02	4.7248e+00	4.5076e–03	6.0580e–02	1.3621e+01	1.5657e–02	1.3568e–01	1.7716e+01	3.7874e–02	1.9691e–02	1.3311e+01	2.3073e–04	3.9951e–02	6.8069e+00	1.4191e–03	1.5517e–02	2.2090e+00	5.3323e–04	2.1609e–02	5.3159e+00	4.3219e–04
Demir et al. (1994)	5.2293e–05	1.8012e–02	1.7997e–02	1.9534e–04	1.0189e–01	4.9918e–02	2.0875e–04	2.1896e–01	4.9998e–02	6.6361e–05	1.2604e+01	1.4833e–05	1.3464e–04	1.7770e+00	4.1373e–05	5.2293e–05	2.8714e–01	6.4110e–05	7.2826e–05	9.8023e–01	1.7133e–05
Demir et al. (1999)	5.2293e–05	1.6240e–02	2.1097e–02	1.6551e–04	1.3096e–01	4.9991e–02	1.7648e–04	3.5549e–01	5.0000e–02	6.6364e–05	6.3046e+00	1.2990e–05	1.3464e–04	2.1037e+00	3.8603e–05	5.2293e–05	4.6029e–01	7.3613e–05	7.2826e–05	1.1232e+00	1.4929e–05
DiFrancesco–Noble (1985)	7.6176e–05	1.3916e–01	8.7797e–03	2.9740e–04	1.6617e+00	2.7500e–02	5.0563e–04	1.6135e+00	4.8359e–02	9.6667e–05	3.6133e+00	2.0273e–02	1.9612e–04	2.1106e+00	5.1745e–05	7.6171e–05	8.4802e–01	7.8744e–05	1.0608e–04	2.2281e+00	3.1190e–05
Dokos et al. (1996)	6.6838e–05	8.0463e–02	9.6235e–03	2.6095e–04	1.9307e–01	3.0310e–02	4.1274e–04	1.5522e–01	4.7214e–02	8.4818e–05	8.0628e–01	9.8628e–06	1.7209e–04	1.1566e+00	6.4063e–05	6.6838e–05	3.0696e–01	9.9930e–05	9.3082e–05	8.1761e–01	3.8085e–05
FitzHugh-Nagumo (1961)	2.7217e–03	4.7620e–03	5.0000e–02	2.6940e–03	1.5777e+00	4.9999e–02	2.6890e–03	2.8700e+00	4.9999e–02	2.6140e–03	2.6883e+00	4.9997e–02	2.6141e–03	2.6159e+00	4.9999e–02	2.6143e–03	1.2819e+00	5.0000e–02	2.6142e–03	2.7481e+00	5.0000e–02
Faber–Rudy (2000)	1.0874e–02	3.5192e–01	1.6327e–02	4.0646e–02	9.3878e–01	4.9508e–02	3.5438e–02	2.0589e+00	3.8926e–02	1.3799e–02	6.4960e+00	1.4017e–03	2.7998e–02	3.0988e+01	2.4258e–03	1.0874e–02	2.0897e+00	1.1467e–03	1.5144e–02	4.5945e+00	1.2999e–03
Fox et al. (2002)	4.5596e–03	3.3172e–01	5.3733e–03	1.7802e–02	9.8529e–01	1.8886e–02	3.9869e–02	9.8121e–01	4.5025e–02	5.7862e–03	6.1257e+00	2.8119e–03	1.1740e–02	4.3085e+00	5.7604e–03	4.5596e–03	3.5079e+00	2.2097e–03	6.3499e–03	8.1134e+00	3.0899e–03
Hilgemann–Noble (1987)	6.1571e–05	5.6775e–02	4.3602e–03	2.4038e–04	1.8134e–01	1.1942e–02	1.2224e–04	3.6132e–01	6.2921e–03	7.8134e–05	7.9424e–01	6.0361e–04	1.5853e–04	7.1143e–01	8.7678e–04	6.1571e–05	3.3012e–01	4.4528e–04	8.5746e–05	7.9979e–01	9.2930e–04
Hund–Rudy (2004)	7.7965e–03	2.1382e–01	4.8996e–02	9.7927e–03	2.6599e+00	4.6699e–02	9.3866e–03	1.1529e+01	4.1533e–02	1.3003e–02	1.3746e+01	3.0134e–03	2.6382e–02	6.8219e+00	2.2549e–03	1.0246e–02	2.6602e+00	4.9819e–04	1.4270e–02	6.7317e+00	3.2548e–03
Jafri et al. (1998)	4.3300e–04	5.3558e+00	8.0899e–04	1.6883e–03	1.8107e+01	2.4143e–03	3.7600e–03	3.2805e+01	5.2780e–03	5.4877e–04	1.2480e+02	2.2398e–04	1.0179e–03	1.1533e+02	4.0834e–04	3.9535e–04	7.5503e+01	1.5245e–04	6.4004e–04	1.7567e+02	3.4178e–04
Luo–Rudy (1991)	1.3287e–02	9.1163e–02	5.4646e–03	5.1876e–02	2.5316e–01	1.7954e-02	1.1618e–01	5.0030e–01	4.3569e–02	1.6862e–02	2.6092e+00	1.5905e–04	3.4211e–02	1.5423e+00	1.7533e–05	1.3287e–02	8.0670e–01	8.2108e–05	1.8504e–02	2.1740e+00	2.7365e–07
Maleckar et al. (2009)	4.8053e–05	8.7255e–02	1.3231e–03	1.8761e–04	5.6364e–01	4.3223e–03	4.2018e–04	6.2885e–01	9.0645e–03	6.0980e–05	3.6579e+00	8.4020e–04	1.2372e–04	2.9905e+00	1.4523e–03	4.8053e–05	1.8397e+00	2.1775e–04	6.6921e–05	4.7585e+00	5.1124e–04
McAllister et al. (1975)	1.0934e–02	1.2491e–01	2.1598e–03	4.2690e–02	5.6409e–01	6.3421e–03	9.5610e–02	4.8466e–01	1.3715e–02	1.2876e–02	3.6550e+00	4.7942e–07	2.8153e–02	3.4354e+00	2.5661e–06	1.0934e–02	2.2030e+00	9.5369e–06	1.5228e–02	5.3292e+00	9.2902e–08
Noble (1962)	9.4880e–05	4.1900e–03	1.2260e–02	1.1077e–04	5.8033e–02	1.0781e–02	1.0526e–04	1.2011e–01	9.6324e–03	1.6245e–04	8.3532e–02	1.0462e–03	3.8370e–04	8.4302e–02	4.0605e–03	2.0410e–04	5.7850e–02	3.3654e–03	2.0025e–04	1.5323e–01	1.5930e–03
Noble–Noble (1984)	1.6059e–04	4.9362e–02	2.9266e–02	3.6874e–04	6.8159e–02	4.9981e–02	3.9383e–04	1.3232e–01	4.9980e–02	2.0379e–04	4.9581e–01	3.6845e–05	4.1348e–04	5.6403e–01	1.6032e–04	1.6059e–04	1.6251e–01	4.6419e–04	2.2365e–04	4.4397e–01	5.6692e–05
Noble et al. (1991)	5.1412e–05	1.0832e–01	1.7163e–03	2.0072e–04	5.2958e–01	3.4262e–03	4.4955e–04	9.1534e–01	6.7517e–03	6.4593e–05	1.3587e+00	2.3214e–04	1.3237e–04	1.0922e+00	4.9678e–04	5.1412e–05	2.9327e–01	1.4937e–04	7.1599e–05	9.2946e–01	2.7498e–04
Noble et al. (1998)	5.5473e–05	9.7216e–02	6.1817e–03	1.7839e–04	2.7056e–01	1.2515e–02	1.7506e–04	7.1063e–01	1.1756e–02	6.9695e–05	2.9400e+00	4.6083e–04	1.4283e–04	2.2861e+00	5.0710e–04	5.5473e–05	6.2276e–01	7.0186e–04	7.7254e–05	1.5425e+00	7.2074e–04
Nygren et al. (1998)	4.9589e–05	2.4881e–01	3.0497e–03	1.9361e–04	1.3914e+00	9.6006e–03	4.3361e–04	1.8218e+00	3.7892e–02	6.2929e–05	5.1327e+00	6.7847e–05	1.2768e–04	7.0054e+00	2.4360e–04	4.9589e–05	1.9430e+00	1.4596e–04	6.9060e–05	5.0188e+00	3.6091e–06
Pandit et al. (2001))	2.8897e–07	3.3149e+00	1.2442e–04	1.1282e–06	1.7107e+01	3.6273e–04	2.5267e–06	2.0713e+01	7.6377e–04	3.6305e–07	1.2585e+02	3.0361e–07	7.4401e–07	1.1671e+02	5.6234e–07	2.8897e–07	6.9375e+01	3.5921e–08	4.0243e–07	1.7711e+02	3.8200e–07
Pandit et al. (2003)	2.6523e–08	3.3880e+01	6.9372e–03	1.0355e–07	1.7944e+02	6.9958e–03	2.3192e–07	1.8303e+02	7.0939e–03	3.3000e–08	1.3307e+03	6.9069e–03	6.8203e–08	1.1159e+03	6.9069e–03	2.6523e–08	8.2199e+02	6.9069e–03	3.6938e–08	1.9196e+03	6.9069e–03
PuglisiBers	1.0454e–02	7.0989e–01	2.9473e–02	1.6822e–02	1.0760e+00	4.7114e–02	1.6602e–02	2.5711e+00	4.7971e–02	1.3267e–02	5.1939e+00	4.4702e–02	1.4888e–02	4.1489e+00	4.9395e–02	1.0454e–02	3.1416e+00	3.3901e–02	1.4559e–02	7.3737e+00	4.8580e–02
Sakmann et al. (2000) – Endocardial	6.7440e-05	6.5490e–02	3.6162e–02	1.0836e–04	5.3471e–01	4.7079e–02	1.6216e–04	6.2157e–01	7.0137e–02	8.5582e–05	1.3748e+00	2.7620e–03	1.7364e–04	2.1750e+00	5.4049e–03	6.7440e–05	5.7687e–01	3.4538e–03	9.3920e–05	1.3848e+00	2.5036e–03
Sakmann et al. (2000) – Epicardial	6.7000e–05	5.9875e–02	4.1447e–02	1.0097e–04	4.2843e–01	4.8653e–02	2.9937e–06	3.1092e+01	5.5979e–04	8.5643e–05	1.4081e+00	3.1994e–03	1.7371e–04	2.4310e+00	5.8307e–03	6.7466e–05	5.8817e–01	3.7727e–03	9.3957e–05	1.5080e+00	2.7794e–03
Sakmann et al. (2000) – M-cell	6.7049e–05	1.2193e–01	3.5619e–03	2.6177e–04	4.0051e–01	4.2459e–03	5.8628e–04	5.7791e–01	1.7006e–02	8.5086e–05	1.4087e+00	2.0517e–03	1.7263e–04	4.1988e+00	4.2390e–03	6.7049e–05	5.7300e–01	2.0841e–03	9.3376e–05	1.3976e+00	1.8459e–03
Stewart et al. (2009)	1.4459e–02	4.4209e–01	2.7666e–03	5.6451e–02	2.0499e+00	7.3510e–03	1.2643e–01	2.1530e+00	1.4448e–02	1.8349e–02	1.2167e+01	6.1833e–03	3.7228e–02	8.2702e+00	5.9501e–06	1.4459e–02	5.0047e+00	9.5991e–06	2.0136e–02	1.2684e+01	2.4421e–07
Ten Tusscher et al. (2004) –Endocardial	1.7057e–03	1.6552e+00	4.3223e–04	6.6595e–03	5.7068e+00	1.4745e–03	1.4915e–02	5.9435e+00	3.3340e–03	2.1646e–03	4.1463e+01	5.0955e–04	4.3628e–03	3.0338e+01	7.3680e–04	1.6945e–03	1.8440e+01	6.8065e–05	2.3598e–03	4.4641e+01	9.6544e–05
Ten Tusscher et al. (2006) –Endocardial	1.5900e–03	7.5807e–01	9.8490e–04	6.1924e–03	3.4789e+00	3.0752e–03	1.3869e–02	4.0404e+00	6.7679e–03	2.0128e–03	3.1421e+01	2.0840e–04	4.0837e–03	2.3064e+01	4.2646e–04	1.5861e–03	1.6584e+01	1.6256e–04	2.2088e–03	4.2175e+01	2.2910e–04
Ten Tusscher et al. (2004) –Epicardial	1.7055e–03	1.4744e+00	3.8554e–04	6.6588e–03	6.4227e+00	1.1390e–03	1.4913e–02	6.7111e+00	2.4889e–03	2.1644e–03	4.1625e+01	3.6098e–04	4.3628e–03	2.6398e+01	1.3202e–04	1.6945e–03	1.7598e+01	4.9603e–05	2.3598e–03	4.3503e+01	7.0644e–05
Ten Tusscher et al. (2006) –Epicardial	2.1191e–03	6.0086e–01	5.3524e–04	8.2734e–03	3.0580e+00	1.6770e–03	1.8529e–02	2.5948e+00	3.7201e–03	2.5000e–03	2.1643e+01	9.3349e–05	5.4561e–03	1.5586e+01	2.0699e–04	2.1191e–03	1.7212e+01	7.7445e–05	2.9512e–03	4.4649e+01	1.1034e–04
Ten Tusscher et al. (2004) –M-cell	1.7115e–03	1.1081e+00	2.0687e–04	6.6820e–03	5.0505e+00	5.7034e–04	1.4965e–02	5.2963e+00	1.1857e–03	2.1719e–03	3.4899e+01	5.6612e–05	4.4067e–03	2.2226e+01	1.1371e–04	1.7115e–03	1.4962e+01	4.3613e–05	2.3835e–03	3.8744e+01	6.1649e–05
Ten Tusscher et al. (2006) –M-cell	2.0379e–03	9.6348e–01	5.1864e–04	7.9565e–03	3.5900e+00	1.6037e–03	1.7820e–02	4.1434e+00	3.4888e–03	2.5862e–03	2.9407e+01	1.0175e–04	5.2472e–03	1.5964e+01	1.9272e–04	2.0379e–03	1.2095e+01	7.3340e–05	2.8381e–03	3.0826e+01	1.0333e–04
Wang–Sobie (2008)	1.6311e–02	3.3048e–01	7.1810e–03	6.3683e–02	1.5232e+00	2.2595e–02	1.1994e–01	7.3131e–01	4.6219e–02	2.0699e–02	3.5582e+00	1.1400e–03	4.1997e–02	5.3548e+00	3.8562e–03	1.6311e–02	1.2537e+00	1.7217e–03	2.2716e–02	3.8948e+00	2.3545e–03
Winslow et al. (1999) (31 eqn)	1.0400e–04	1.3163e+01	1.4362e–03	4.1278e–04	7.6932e+01	2.5991e–03	9.2448e–04	8.0854e+01	7.4232e–03	1.3417e–04	5.1014e+02	5.6852e–03	2.6978e–04	3.8861e+02	9.5513e–03	1.0478e–04	2.9860e+02	4.8139e–03	1.4592e–04	7.2392e+02	7.0014e–04
Zhang et al. (2000)	9.0103e–05	6.1924e–02	4.9013e–03	3.5178e–04	4.8803e–01	1.4544e–02	7.8786e–04	6.1441e–01	3.1339e–02	1.1434e–04	1.5728e+00	1.1459e–05	2.3199e–04	6.9150e–01	2.7911e–05	9.0103e–05	5.5132e–01	3.1884e–05	1.2548e–04	1.2841e+00	1.7238e–06

Table A.2: Δt_{exp} represents the experimental largest time step and CPU time represents the minimum of fifty times running of the FE, the RKC$(2,1)$, the RKC$(3,1)$, the SSP$(3,3)$, the SSP$(4,3)$, the EMP, and the RK4 methods with less than 5% $[\text{MRMS}]_v$.

Model	FE			RKC$(2,1)$			RKC$(3,1)$			SSP$(3,3)$			SSP$(4,3)$			EMP			RK4		
	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$	Δt_{exp} (ms)	CPU time (s)	$[\text{MRMS}]_v$
Beeler–Reuler (1977)	2.5298e–02	2.4791e–02	5.2120e–03	1.0121e–01	1.0664e–01	1.7150e–02	2.1585e–01	1.0075e–01	4.8472e–02	3.1559e–02	1.4632e+00	2.3551e–03	6.7006e–02	6.4874e–01	2.3586e–02	2.4630e–02	4.5274e–01	1.4632e–03	3.4158e–02	1.0171e+00	1.4848e–03
Bondarenko et al. (2004)	2.1300e–04	2.4416e+00	1.1353e–03	8.5182e–04	1.4903e+01	4.6280e–03	1.9252e–03	1.3704e+01	1.0474e–02	2.6745e–04	9.9434e+01	1.5742e–03	5.5141e–04	6.9712e+01	3.2427e–03	4.0300e–04	2.9767e+01	2.3709e–03	2.9540e–04	1.2811e+02	1.7386e–03
Courtemanche et al. (1998)	1.9400e–02	3.0747e–01	6.2901e–03	8.1700e–02	4.4933e–01	1.9742e–02	1.6232e–01	4.2629e–01	4.9851e–02	2.4341e–02	3.0391e+00	6.8178e–04	4.9485e–02	2.7204e+00	2.4199e–03	1.8400e–02	1.9110e+00	7.3792e–04	2.6400e–02	4.2439e+00	2.5034e–04
Demir et al. (1994)	5.9599e–05	1.6659e–02	2.2081e–02	1.9551e–04	8.4736e–02	4.9978e–02	2.0875e–04	1.9210e–01	4.9998e–02	7.1524e–05	4.9149e–01	1.0548e–03	1.3851e–04	3.8060e–01	2.4342e–02	5.5293e–05	4.1262e–01	3.9599e–04	7.5493e–05	7.5214e–01	2.2562e–03
Demir et al. (1999)	5.9607e–05	1.9908e–02	2.4060e–02	1.6552e–04	1.2456e–01	4.9993e–02	1.7648e–04	2.9459e–01	5.0000e–02	7.1764e–05	6.5279e–01	1.8245e–02	1.3860e–04	4.1587e–01	5.3154e–03	5.5393e–05	3.3214e–01	3.9717e–04	7.5626e–05	8.0026e–01	4.9487e–02
DiFrancesco–Noble (1985)	7.7276e–05	8.3383e–02	4.5145e–02	3.0640e–04	2.3743e–01	3.4776e–02	5.0873e–04	3.2096e–01	4.8544e–02	9.7467e–05	1.6269e+00	4.9700e–02	1.9984e–04	1.7458e+00	4.9811e–02	7.6638e–05	8.9125e–01	9.2129e–05	1.0654e–04	2.3961e+00	1.8557e–04
Dokos et al. (1996)	7.0200e–05	8.0562e–02	2.2554e–02	2.7000e–04	1.6345e–01	3.1396e–02	4.3452e–04	2.3803e–01	4.9945e–02	8.7018e–05	6.5181e–01	8.9834e–03	1.7552e–04	4.3388e–01	1.0259e–04	6.9338e–05	3.3600e–01	1.1757e–04	9.5438e–05	8.0437e–01	6.8507e–05
FitzHugh–Nagumo (1961)	2.7200e–03	7.6938e–01	4.9968e–02	2.6937e–03	1.1895e+00	4.9994e–02	2.6891e–03	1.9282e+00	5.0000e–02	2.6135e–03	2.1985e+00	4.9987e–02	2.6141e–03	2.6159e+00	4.9999e–02	2.6143e–03	1.2819e+00	5.0000e–02	2.6142e–03	2.7481e+00	5.0000e–02
Faber–Rudy (2000)	1.1294e–02	1.3211e–01	1.7008e–02	4.0966e–02	4.1099e–01	4.9997e–02	4.3788e–02	8.9999e–01	4.9999e–02	1.4089e–02	3.2388e+00	1.4373e–03	2.9634e–02	1.2905e+01	3.1579e–03	1.1244e–02	1.6168e+00	1.0018e–03	1.5564e–02	3.9347e+00	1.1878e–03
Fox et al. (2002)	4.6200e–03	3.5088e–01	5.4455e-03	1.8000e–02	6.5123e–01	1.9077e–02	4.1200e–02	7.0721e–01	4.6519e–02	5.7962e–03	4.7237e+00	2.8173e–03	1.1843e–02	3.5261e+00	5.8110e–03	4.6200e–03	2.7766e+00	2.2392e–03	6.5200e–03	6.7926e+00	3.2090e–03
Hilgemann–Noble (1987)	6.2500e–05	8.4361e–02	4.7582e–03	2.4400e–04	1.0039e–01	1.0436e–02	4.2187e–04	1.0606e–01	4.9947e–02	7.8334e–05	5.7160e–01	8.4811e–04	1.5923e–04	4.3931e–01	1.0700e–03	6.2071e–05	3.1958e–01	7.2888e–04	8.6246e–05	7.0635e–01	1.1697e–03
Hund–Rudy (2004)	7.8874e–03	2.2220e–01	4.9939e–02	1.0440e–02	2.9393e+00	4.9973e–02	1.1003e–02	2.6024e+00	4.9784e–02	1.3963e–02	4.2422e+00	3.0650e–03	2.8505e–02	3.4895e+00	8.7876e–04	1.1117e–02	4.4086e+00	3.6066e–04	1.5444e–02	5.7985e+00	3.0478e–04
Jafri et al. (1998)	5.7600e–04	4.5764e+00	1.8154e–02	1.7500e–03	1.8212e+01	2.4939e–03	3.7600e–03	3.2805e+01	5.2780e–03	7.9972e–04	9.2747e+01	4.9545e–02	1.2727e–03	9.2116e+01	6.6143e–03	4.2600e–04	6.7364e+01	1.6968e–04	7.4600e–04	1.3251e+02	4.6231e–03
Luo–Rudy (1991)	1.3572e–02	8.5973e–02	5.6083e–03	5.4400e–02	1.9396e–01	1.8982e–02	1.1800e–01	2.1622e–01	4.5668e–02	1.7032e–02	1.3568e+00	4.6923e–04	3.4426e–02	1.4179e+00	5.0656e–05	1.3422e–02	8.0194e–01	8.4156e–05	1.8642e–02	2.0074e+00	1.7647e–07
Maleckar et al. (2009)	5.0200e–05	2.9673e–01	1.6159e–02	1.9000e–04	5.5468e–01	3.4263e–03	4.3628e–04	4.3383e–01	4.8524e–02	6.1980e–05	3.1770e+00	5.9248e–03	1.2582e–04	2.7435e+00	4.2425e–02	5.0053e–05	1.7369e+00	1.7378e–02	6.8221e–05	4.9233e+00	1.5111e–02
McAllister et al. (1975)	2.4700e–02	3.5631e–02	4.9527e–02	9.6000e–02	2.3019e–01	4.6948e–02	9.8600e–02	5.201e–01	1.4162e–02	1.3835e–02	3.8038e+00	6.1926e–07	2.8274e–02	3.2698e+00	3.0749e–06	1.1014e–02	2.0484e+00	9.7051e–06	1.5318e–02	5.3200e+00	2.3573e–05
Noble (1962)	2.0228e–04	3.0909e–03	4.9586e–02	3.5769e–04	2.7038e–02	4.9878e–02	3.9082e–04	4.0913e–02	4.9951e–02	2.5085e–04	4.7646e–02	4.9795e–02	5.1912e–04	5.0400e–02	4.9336e–02	2.1298e–04	4.4086e+00	1.3490e–03	2.7078e–04	6.9977e–02	4.9914e–02
Noble–Noble (1984)	2.0400e–04	5.7262e–03	3.8972e–02	3.6874e–04	6.3294e–02	4.9981e–02	3.9405e–04	1.4796e–01	4.9999e–02	2.3636e–04	2.2660e–01	2.1897e–02	4.3938e–04	1.0305e–01	3.6892e–02	1.7848e–04	8.1128e–02	3.5450e–02	2.4108e–04	2.3326e–01	4.9071e–02
Noble et al. (1991)	5.1500e–05	1.8201e–02	1.6961e–03	2.0100e–04	1.2543e–01	3.4581e–03	4.5000e–04	1.3260e–01	6.6311e–03	6.4735e–05	1.5370e+00	2.4792e–04	1.3249e–04	4.4325e–01	4.9153e–04	5.1500e–05	4.9358e–01	1.3620e–04	7.3799e–05	7.0791e–01	3.8207e–03
Noble et al. (1998)	5.5600e–05	3.5936e–02	6.1962e–03	1.8000e–04	2.8651e–01	1.2508e–02	1.8700e–04	6.1719e–01	1.2933e–02	7.0213e–05	1.1216e+00	7.5883e–04	1.4300e–04	9.8792e–01	5.0792e–04	5.6973e–05	6.1927e–01	2.6946e–03	7.9154e–05	1.5082e+00	5.3382e–03
Nygren et al. (1998)	5.3589e–05	7.7448e–02	3.0280e–02	2.0521e–04	3.8420e–01	1.0284e–02	4.4691e–04	4.2804e–01	4.5291e–02	6.6129e–05	3.4139e+00	8.0505e–04	1.3250e–04	2.2046e+00	7.5250e–04	5.2989e–05	1.7679e+00	9.6730e–04	7.1989e–05	4.7359e+00	3.3392e–03
Pandit et al. (2001)	2.9100e–07	5.3835e+00	1.2482e–04	1.1300e–06	2.0256e+01	3.6281e–04	2.5400e–06	2.0255e+01	7.6641e–04	3.6305e–07	1.2967e+02	3.0360e–07	7.4401e–07	9.1504e+01	5.6234e–07	2.8897e–07	6.7539e+01	8.4664e–08	4.0243e–07	1.7448e+02	3.6367e–07
Pandit et al. (2003)	2.6523e–08	3.6539e+01	6.9372e–03	1.0355e–07	1.8203e+02	6.9958e–03	2.3192e–07	1.9140e+02	7.0939e–03	3.3323e–08	1.3018e+03	6.9069e–03	3.6938e–08	1.8973e+03	6.9069e–03	2.6523e–08	7.4224e+02	6.9069e–03	3.6938e–08	1.9196e+03	6.9069e–03
PuglisiBers	1.0884e–02	1.9327e–01	3.2953e–02	1.6842e–02	1.1701e+00	4.9781e–02	1.6642e–02	2.9553e+00	4.9027e–02	1.3457e–02	3.5549e+00	4.6365e–02	1.5359e–02	4.6629e+00	4.9673e–02	1.2474e–02	1.4543e+00	4.9307e–02	1.4589e–02	4.1141e+00	4.8432e–02
Sakmann et al. (2000) – Endocardial	6.9000e–05	5.2694e–01	3.7504e–02	1.1000e–04	5.0272e–01	4.9120e–02	1.1881e–04	8.2267e–01	4.9895e–02	8.6233e–05	1.1288e+00	2.6263e–03	1.7466e–04	8.2207e–01	6.4168e–03	6.7440e–05	5.5610e–01	3.4538e–03	9.3920e–05	1.6529e+00	2.5035e–03
Sakmann et al. (2000) – Epicardial	6.8000e–05	2.0159e–01	4.3111e–02	1.0200e–04	4.6299e–01	4.9550e–02	1.0800e–04	8.7030e–01	4.8896e–02	8.6343e–05	1.0991e+00	3.0370e–03	1.7469e–04	1.0235e+00	6.7507e–03	6.8066e–05	5.1905e–01	3.9557e-03	9.4557e–05	1.5265e+00	2.8068e–03
Sakmann et al. (2000) – M-cell	6.8600e–05	3.3911e–02	3.9258e–03	2.7200e–04	1.9863e–01	7.9652e–03	5.9308e–04	1.8133e–01	4.9495e–02	8.5821e–05	1.9528e+00	2.7661e–03	1.7358e–04	8.6923e–01	4.5381e–03	6.7649e–05	7.2961e–01	9.7024e–04	9.3976e–05	2.2848e+00	2.3383e–03
Stewart et al. (2009)	1.5212e–02	4.1142e–01	4.9869e–02	6.0986e–02	1.9369e+00	4.8662e–02	1.3539e–01	1.5576e+00	2.7832e–02	1.9132e–02	8.5850e+00	4.8325e–02	3.7366e–02	6.8836e+00	3.3738e–05	1.4559e–02	5.0458e+00	9.4102e–06	2.0236e–02	1.3133e+01	3.3947e–05
Ten Tusscher et al. (2004) –Endocardial	1.7829e–03	1.1957e+00	1.5526e–03	6.9345e–03	4.2900e+00	1.6259e–03	1.6235e–02	4.2864e+00	4.0746e–03	2.2396e–03	3.1663e+01	1.5221e–03	4.5586e–03	2.3303e+01	1.4644e–03	1.6959e–03	2.0071e+01	2.8498e–04	2.3609e–03	4.7466e+01	3.9775e–04
Ten Tusscher et al. (2006) –Endocardial	1.6200e–03	8.9863e–01	9.9918e–04	6.4300e–03	3.5449e+00	3.1929e–03	1.4200e–02	3.5092e+00	6.8820e–03	2.0297e–03	2.6447e+01	1.9958e–04	4.4915e–03	1.7539e+01	7.0503e–04	1.6200e–03	1.4310e+01	1.6531e–04	2.2200e–03	3.4823e+01	2.3036e–04
Ten Tusscher et al. (2004) –Epicardial	1.7835e–03	1.1548e+00	1.5568e–03	7.1548e–03	4.2930e+00	2.0351e–03	1.6043e–02	4.1409e+00	3.0533e–03	2.2364e–03	3.1805e+01	1.4412e–03	4.3663e–03	2.4756e+01	1.3177e–04	1.6965e–03	1.7430e+01	5.0079e–05	2.3625e–03	4.3102e+01	7.0650e–05
Ten Tusscher et al. (2006) –Epicardial	2.1400e–03	8.6557e–01	5.3646e–04	8.5600e–03	2.8914e+00	1.7278e–03	1.8700e–02	3.1167e+00	3.7086e–03	2.6888e–03	2.0917e+01	1.5317e–04	5.5085e–03	1.5399e+01	2.3682e–04	2.1251e–03	1.1069e+01	7.7695e–05	2.9582e–03	2.6920e+01	1.1004e–04
Ten Tusscher et al. (2004) –M-cell	1.7667e–03	9.0396e–01	4.3927e–04	7.0820e–03	4.1688e+00	7.4036e–04	1.5904e–02	3.5254e+00	1.3192e–03	2.5872e–03	2.1481e+01	1.0537e–04	4.4501e–03	2.0321e+01	1.1544e–04	1.7305e–03	1.5244e+01	4.5085e–05	2.4075e–03	3.5862e+01	6.2223e–05
Ten Tusscher et al. (2006) –M-cell	2.0619e–03	6.6438e–01	4.7968e–04	8.2505e–03	2.6742e+00	1.6142e–03	1.8520e–02	2.9897e+00	3.6150e–03	2.5790e–03	2.0657e+01	8.6791e–05	5.3011e–03	1.5359e+01	1.7609e–04	2.0400e–03	1.0333e+01	7.3263e–05	2.8474e–03	2.5446e+01	1.0686e–04
Wang–Sobie (2008)	1.6611e–02	8.4543e–02	8.4128e–03	6.6443e–02	3.5429e–01	2.4299e–02	1.2634e–01	5.7322e–01	4.9967e–02	2.0789e–02	2.2012e+00	1.1371e–03	4.2732e–02	1.4439e+00	2.5577e–03	1.6381e–02	1.0174e+00	1.6877e–03	2.2771e–02	2.4985e+00	2.3341e–03
Winslow et al. (1999) (31 eqn)	1.0700e–04	1.2650e+01	1.4777e–03	4.1400e–04	7.3222e+01	2.6094e–03	9.5500e–04	6.8827e+01	1.9762e–02	1.3452e–04	5.2316e+02	5.5230e–03	2.7344e–04	3.6913e+02	4.8072e–02	1.0488e–04	2.9238e+02	1.3882e–02	1.4612e–04	7.0635e+02	5.0418e–03
Zhang et al. (2000)	9.9925e–05	7.3992e–02	5.4361e–03	3.8080e–04	1.2287e–01	1.5437e–02	8.6990e–04	1.1898e–01	3.4806e–02	1.2155e–04	8.6181e–01	4.8513e–04	2.3846e–04	1.1920e+00	4.1124e–02	9.5103e–05	5.2296e–01	1.5793e–03	1.3028e–04	1.2183e+00	4.4012e–02

Table A.3: This table shows the relative accuracy of the theoretical time step to the experimental time step of the FE, the RKC$(2,1)$, the RKC$(3,1)$, the SSP$(3,3)$, the SSP$(4,3)$, the EMP, and the RK4 methods.

Model	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of FE	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of RKC$(2,1)$	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of RKC$(3,1)$	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of SSP$(3,3)$	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of SSP$(4,3)$	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of EMP	$\frac{\Delta t_{theo}}{\Delta t_{exp}} \times 100\%$ of RK4
Beeler–Reuler (1977)	96.40	94.08	98.80	98.07	93.71	99.02	99.43
Bondarenko et al. (2004)	98.91	96.56	95.69	99.63	98.37	52.28	99.32
Courtemanche et al. (1998)	79.99	74.15	83.58	80.90	80.73	84.33	81.85
Demir et al. (1994)	87.74	99.91	100.00	92.78	97.21	94.57	96.47
Demir et al. (1999)	87.74	100.00	100.00	92.48	97.14	94.40	96.30
DiFrancesco–Noble (1985)	98.58	97.06	99.39	99.18	98.14	99.39	99.57
Dokos et al. (1996)	95.21	96.65	94.99	97.47	98.05	96.39	97.53
FitzHugh-Nagumo (1961)	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Faber–Rudy (2000)	96.28	99.22	80.93	97.94	94.48	96.71	97.30
Fox et al. (2002)	98.69	98.90	96.77	99.83	99.13	98.69	97.39
Hilgemann–Noble (1987)	98.51	98.52	28.98	99.75	99.56	99.19	99.42
Hund–Rudy (2004)	98.85	93.80	85.31	93.13	92.55	92.17	92.39
Jafri et al. (1998)	75.17	96.47	100.00	68.62	79.98	92.80	85.80
Luo–Rudy (1991)	97.90	95.36	98.46	99.00	99.38	98.99	99.26
Maleckar et al. (2009)	95.72	98.74	96.31	98.39	98.33	96.00	98.09
McAllister et al. (1975)	44.27	44.47	96.97	93.07	99.57	99.27	99.41
Noble (1962)	46.91	30.97	26.93	64.76	73.91	95.83	73.95
Noble–Noble (1984)	78.72	100.00	99.94	86.17	94.11	89.98	92.77
Noble et al. (1991)	99.83	99.86	99.90	99.78	99.91	99.83	97.02
Noble et al. (1998)	99.77	99.11	93.61	99.26	99.88	97.31	97.60
Nygren et al. (1998)	92.54	94.35	97.02	95.16	96.36	93.58	95.93
Pandit et al. (2001)	99.30	99.84	99.48	100.00	100.00	100.00	100.00
Pandit et al. (2003)	100.00	100.00	100.00	99.03	100.00	100.00	100.00
PuglisiBers	96.05	100.00	94.99	98.59	96.93	83.81	99.79
Sakmann et al. (2000) – Endocardial	97.74	98.51	97.30	99.25	99.42	100.00	100.00
Sakmann et al. (2000) – Epicardial	98.53	98.99	27.72	99.19	99.43	99.11	99.37
Sakmann et al. (2000) – M-cell	97.74	96.24	98.85	99.14	99.45	99.11	99.36
Stewart et al. (2009)	95.05	92.57	93.38	95.91	99.63	99.36	99.53
Ten Tusscher et al. (2004) –Endocardial	95.67	96.03	91.87	96.65	95.71	99.92	99.95
Ten Tusscher et al. (2006) –Endocardial	98.15	96.31	97.67	99.17	90.92	97.91	99.50
Ten Tusscher et al. (2004) –Epicardial	95.63	93.07	92.96	96.78	99.92	99.88	99.89
Ten Tusscher et al. (2006) –Epicardial	99.02	96.65	99.09	92.98	99.05	99.72	99.79
Ten Tusscher et al. (2004) –M-cell	96.88	94.35	94.10	97.88	99.03	98.90	99.00
Ten Tusscher et al. (2006) –M-cell	98.84	96.44	96.22	100.00	98.98	99.90	99.67
Wang–Sobie (2008)	98.19	95.85	94,93	99.57	98.28	99.57	99.76
Winslow et al. (1999) (31 eqn)	97.20	99.71	96.80	99.74	98.67	99.91	99.86
Zhang et al. (2000)	90.17	92.38	90.57	94.07	97.29	94.74	96.32

Appendix B

Reference Solutions

We present the reference solutions used in Section 4.4 to compute the MRMS error of the numerical solution of the one-dimensional, two- dimensional, and three-dimensional monodomain problem. The solution presented at different time levels and this solution obtained by applying RKC$(3,1)$ method on monodomain model problem.

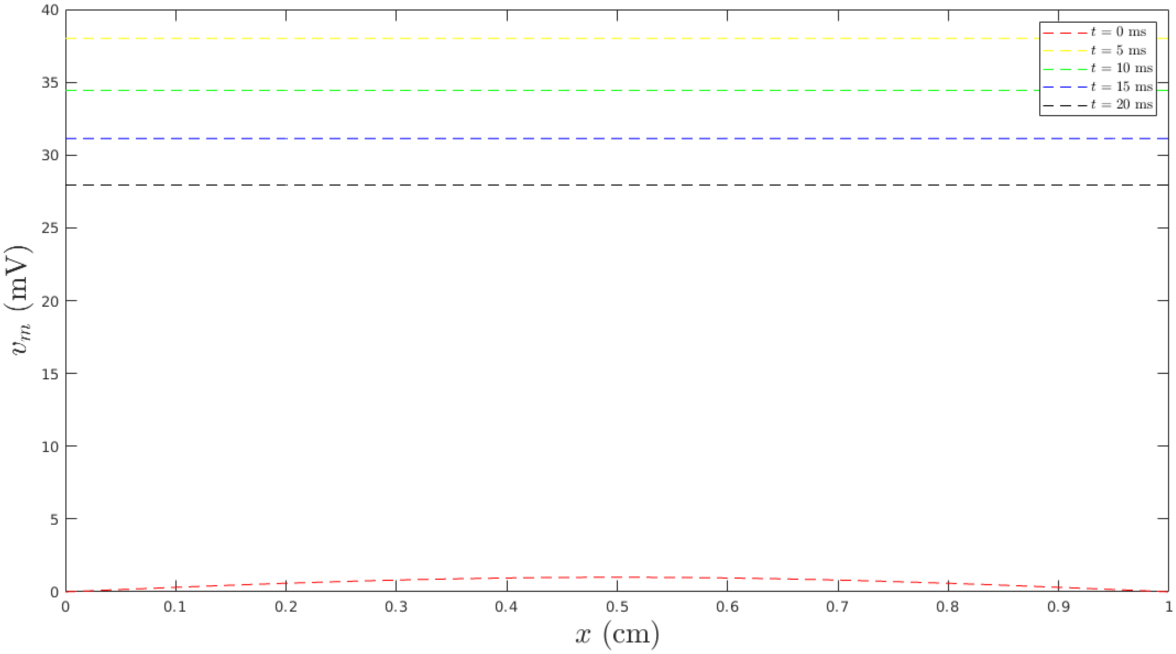


Figure B.1: Electrical potential from RKC$(3,1)$ method at $t = 0$ ms, $t = 5$ ms, $t = 10$ ms, $t = 15$ ms and $t = 20$ ms for the one-dimensional monodmain model coupled with the FHN model.

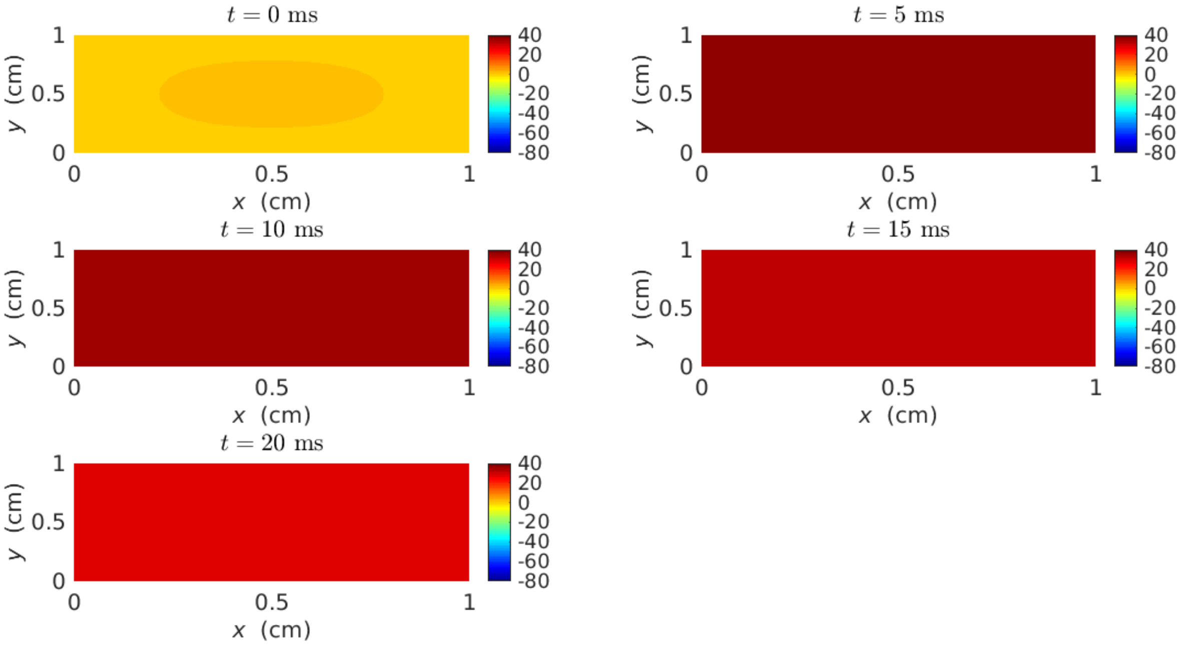


Figure B.2: Electrical potential from RKC$(3, 1)$ method at $t = 0$ ms, $t = 5$ ms, $t = 10$ ms, $t = 15$ ms and $t = 20$ ms for the two-dimensional monodmain model coupled with the FHN model.

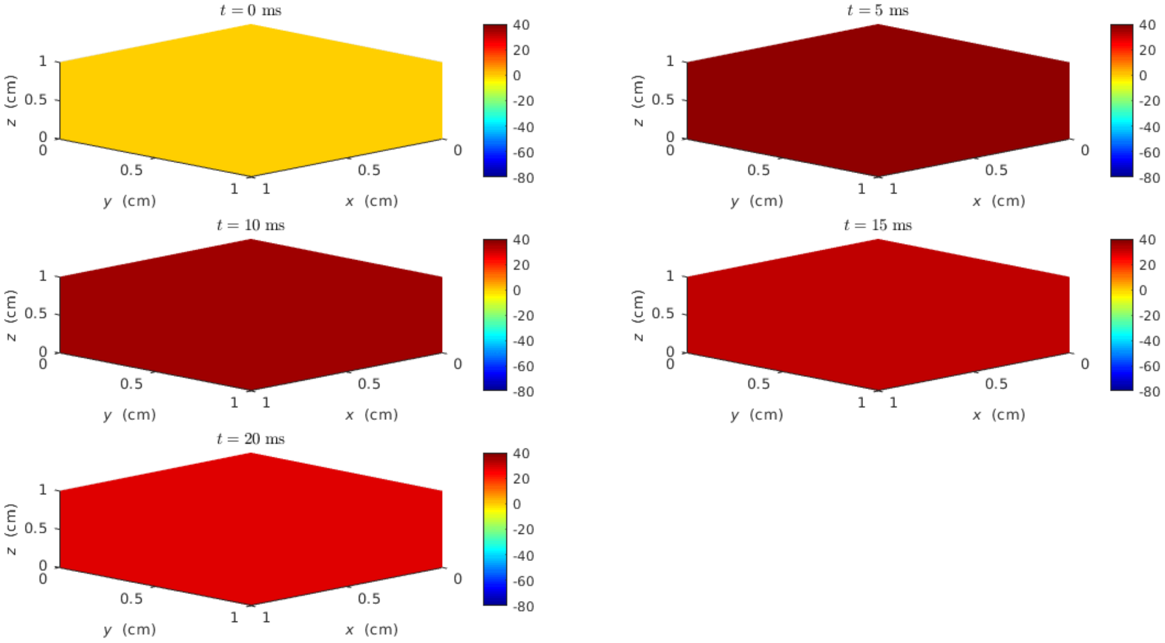


Figure B.3: Electrical potential from RKC$(3, 1)$ method at $t = 0$ ms, $t = 5$ ms, $t = 10$ ms, $t = 15$ ms and $t = 20$ ms for the three-dimensional monodmain model coupled with the FHN model.