

IMPLICIT-EXPLICIT NUMERICAL METHODS IN MODELS
OF CARDIAC ELECTRICAL ACTIVITY

By
Ryan C. Dean

ABSTRACT

Mathematical models of electric activity in cardiac tissue are becoming an increasingly powerful tool in the study of the heart and cardiac arrhythmias. The ordinary differential equations contained within these mathematical models are challenging to solve. This challenge often means that the physiological accuracy of a model is limited by how efficient we can make the numerical solution process. In this thesis, we examine the efficiency of the numerical solution of four cardiac electrophysiological models using Implicit-Explicit (IMEX) methods. We find that a particular IMEX method, ARK5, can be up to 275 times faster than some methods most frequently used in practice.

ACKNOWLEDGEMENTS

I would like to thank Dr. Raymond Spiteri for all the time, encouragement, constructive criticism, financial support, and effort he has given to myself and this project. This project would not have been possible without him

I would also like to thank Mary MacLachlan and Sarah Healy for supplying Matlab code to help debug the `odeToJava` code, and for some helpful comments along the way.

CONTENTS

Abstract	i
Acknowledgements	ii
Contents	iii
List of Tables	iv
List of Figures	v
1 Introduction	1
2 Electrical Activity of the Heart	3
2.1 Relevant Background on the Heart	3
2.2 The Model of Luo and Rudy	4
2.3 The Model of Courtemanche et al.	6
2.4 The Model of Winslow et al.	6
2.5 The Model of Puglisi and Bers	7
3 Mathematical Background	8
3.1 Differential Equations	8
3.2 Numerical Methods	9
3.2.1 Basic Concepts	9
3.2.2 Error and Stability	11
3.2.3 Runge-Kutta Methods	12
3.2.4 Implicit-Explicit Methods	13
3.2.5 Implementation	15
4 Results	16
4.1 Constant Step Size Tests	17
4.1.1 Explicit Runge-Kutta Methods	17
4.1.2 NSFD Method	17
4.2 Variable Step Size Tests	17
4.3 Sparsity Patterns	21
4.4 Eigenvalues	22
5 Conclusion	30
A Mathematical Models	33
A.1 The Model of Luo-Rudy	33
A.2 The Model of Courtemanche et al.	36
A.3 The Model of Winslow et al.	37
A.4 The Model of Puglisi-Bers	40

LIST OF TABLES

4.1	Initial values for V	16
4.2	ERK with Constant Step Size	17
4.3	NSFD with Constant Step Size	18
4.4	Luo-Rudy Dormand Prince Results	18
4.5	Luo-Rudy ARK5 Results	18
4.6	Courtemanche et al. Dormand-Prince Results	19
4.7	Courtemanche et al. ARK5 Results	19
4.8	Winslow et al. Dormand-Prince Results	20
4.9	Winslow et al. ARK5 Results	20
4.10	Puglisi-Bers ARK5 Results	21
4.11	Puglisi-Bers Dormand-Prince Results	21
A.1	Parameters for the Luo-Rudy Phase I model	36

LIST OF FIGURES

2.1	Transmembrane potential over time in the Luo-Rudy model.	5
3.1	Forward Euler	10
4.1	Sparsity pattern for the Jacobian of the Luo-Rudy model.	22
4.2	Sparsity pattern for the Jacobian of the Courtemanche et al. model.	23
4.3	Sparsity pattern for the Jacobian of the Winslow et al. model.	24
4.4	Sparsity pattern for the Jacobian of the Puglisi-Bers model.	25
4.5	Eigenvalues for the Jacobian of the Luo-Rudy model.	26
4.6	Eigenvalues for the Jacobian of the Courtemanche et al. model.	27
4.7	Eigenvalues of the Jacobian of the Winslow et al. model.	28
4.8	Eigenvalues for the Jacobian of the Puglisi-Bers model.	29

CHAPTER 1

INTRODUCTION

Computer simulation is quickly becoming an important tool in cardiovascular research. Mathematical models of the heart can be used to simulate heart conditions and how certain drugs treat these problems. The development of a drug often costs hundreds of millions of dollars [8]; computer simulation aims to cut this cost by reducing the number of physical experiments needed in testing a given drug.

Electrophysiological models of the heart describe how electricity flows through the heart, controlling the contraction of the muscle. The models we consider consist of systems of ordinary differential equations (ODEs). Cardiac electrophysiological models are often based on the Nobel prize winning work of Hodgkin and Huxley [12] in the 1950s that modelled neural tissue mathematically as a circuit. Modern cardiac electrophysiological models adapt the work of Hodgkin and Huxley to describe electrical activity in the heart and include data gathered with from experiments to form increasingly physiologically accurate models.

One of the major barriers to getting useful results from these simulations is the challenge of performing the simulations efficiently. Sometimes the physiological accuracy of the mathematical model must be reduced in the name of simplicity before we may perform the simulation efficiently or even at all [11]. The ODEs found in these models are nonlinear and stiff. The consequence of the stiffness is that the speed with which we can solve these ODEs is limited by stability instead of accuracy. This limitation means that the solution process is much less efficient than it might otherwise be.

In this thesis, we propose to improve the efficiency of the simulation process by using an Implicit-Explicit (IMEX) numerical method [13]. An IMEX method approximates the solution of an ODE which is split into two parts: one part better suited for an implicit numerical method and one part better suited for an explicit numerical method. The IMEX solver uses both an implicit and an explicit method to approximate the solution to the respective parts of the ODE. Using these methods together, we are able to maximize the efficiency of the solution by using the better method for each part. Cardiac electrophysiological models contain linear and nonlinear terms as well as stiff and non-stiff terms, and so an IMEX method is a natural choice. Despite this, there is no published work on the use of IMEX schemes as described in this thesis to solve cardiac electrophysiological

models.

In this thesis, we consider four mathematical models of cardiac electrophysiology: the Luo-Rudy model of guinea pig ventricular tissue [17], the Courtemanche et al. model of human atrial tissue [7], the Winslow et al. model of canine ventricular tissue [25], and the Puglisi-Bers model of rabbit ventricular tissue [20]. In this thesis, we perform a rigorous comparison of an IMEX method, ARK5 [15], to standard methods to solving these four models, and we then consider optimizations to the IMEX process. The rest of the thesis is organized as follows. In Chapter 2 we give an introduction to the physiology and mathematical models of electrical activity in the heart. In Chapter 3 we give an introduction to ODEs and the numerical approximation to the solution of ODEs. In Chapter 4 we discuss the results of the numerical experiments. In Chapter 5 we give a summary of the results and discuss future work.

CHAPTER 2

ELECTRICAL ACTIVITY OF THE HEART

2.1 Relevant Background on the Heart

The heart is the muscle responsible for propelling blood throughout the body. A mammalian heart consists of four connected chambers, two *atria* and two *ventricles*, guarded by valves [21]. The atria receive and transfer blood to the ventricles. The ventricles then propel the blood outside of the heart. Inside the right atrium is the *sinoatrial node*, which acts as the heart's natural pacemaker. In the lower right atrium is the *atrioventricular node* (AV node), which is connected to the bundle of His. The heart contracts and relaxes on a regular cycle to pump blood into the lungs and body.

The heart muscle consists a large number of cells, every one of which has an electric potential. At its resting potential, a cell is negatively charged with respect to its surroundings. A reversal in membrane potential is called an *action potential*. This consists of several stages [21]. First there is a rapid depolarization caused by the opening of fast sodium channels and closing of potassium channels. This results in a short period of positive potential as the cell approaches equilibrium potential for sodium. Calcium then enters through L-type calcium channels, and the cell is in the *plateau phase*. During the plateau phase the cell remains depolarized so that the cell does not relax again until all the blood has been pumped. Next potassium channels are reopened and repolarization begins, causing the cell to return to resting potential. Finally, as the cell approaches its equilibrium potential for potassium, the cell is again at resting potential, ready for another cycle.

Electrical activity is responsible for the contraction and relaxation of the heart. An action potential is initiated spontaneously in specialized tissue inside the sinoatrial node [9]. Electrical activity spreads from one cell to another via areas of low resistance between them known as *gap junctions*. As the action potential reaches an individual cell, the result is a powerful force development and/or mechanical shortening. Action potentials first spread over the atria causing them to contract, transferring blood to the ventricles. Next the action potentials reach the AV node. The AV node spreads the action potential to the ventricles but delays it enough to allow the ventricles to fill completely.

Many heart problems are the result of irregularities in the flow of electricity in the heart. Abnormal electrical activity is called an *arrhythmia*, which in general is caused either by abnormal

impulse formation or *re-entry* [23]. Re-entry is the result of an electrical impulse that persists past the normal activation of the heart and re-excites tissue that has already contracted during the current heartbeat. This causes irregularities in the heartbeat that can lead to many serious problems, including death. To be able to study these conditions noninvasively is one common practical motivation for creating mathematical models of electrical activity in cardiac tissue.

The specific objective of these mathematical models is to model the heart and heart conditions in order to simulate treatments; i.e., we can use these mathematical models to perform computer simulations of the effects of new drugs to treat these heart conditions. The development of a new drug has an average cost of approximately \$900 million [8]; one objective of using computer simulation to simulate new drugs is to reduce the number of physical experiments needed to develop the drug and therefore reduce this cost. As the models become more physiologically accurate we are able to obtain more useful information from these simulations.

Obtaining physiologically accurate mathematical models is a difficult task. A major barrier to obtaining physiological accuracy is the challenging task of performing the simulation efficiently. One reason producing an efficient simulation is difficult is that the mathematical models are quite intricate. This means we must perform a large number of computations when evaluating the right hand side of the ODE¹. Another challenge to performing efficient simulations is that the mathematical models are numerically *stiff*, and so we must use sophisticated numerical methods to perform these simulations efficiently. The two challenges described above are amplified when we move from a model for one cell to a model of many cells in two or three dimensions. To move effectively beyond models for one cell, we need to include enough cells in our model to realistically approximate the geometry and physiology of the heart. Because the heart has more than two billion cells, any realistic simulation will have enough cells to magnify any inefficiencies in the numerical dramatically, possibly even hundreds of thousands of times. The difficulty posed by these challenges has caused some researchers to reduce the physiological accuracy of their model to allow the simulation to be performed within an acceptable amount of time [11]. If we are able to significantly improve the efficiency of the simulation process, then we can gain physiological accuracy in our model and therefore perform more realistic simulations.

2.2 The Model of Luo and Rudy

In 1991 Ching-Hsing Luo and Yoram Rudy developed a model² of guinea pig ventricular action potentials based on a previous model from Beeler and Rueter [4]. The Luo-Rudy model [17] extended the Beeler-Reuter model to include fast inward sodium and outward potassium currents

¹See Chapter 3 for a discussion of the mathematical concepts used in this paragraph.

²See Chapter 3 for a discussion of the mathematics contained within this model.

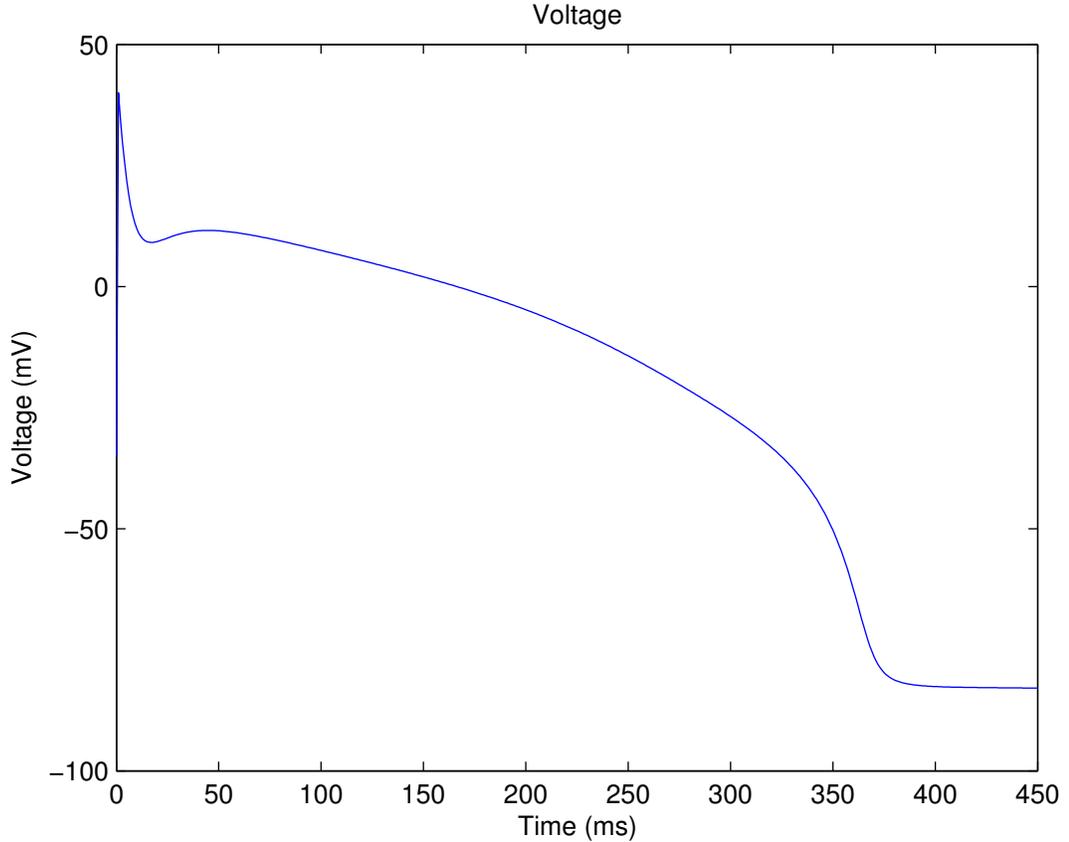


Figure 2.1: Transmembrane potential over time in the Luo-Rudy model. This is the solution of (A.1.22), which depends of the other seven ODEs [17] via the I_{ion} current.

to make the model more physiologically accurate. The general approach of these models is based on Hodgkin-Huxley-type formalism [12].

For an individual cardiac cell we have that the transmembrane potential is given by [17]:

$$\frac{dV_m}{dt} = -\frac{1}{C_m}(I_{ion} + I_{st}), \quad (2.2.1)$$

where C_m is the membrane capacitance, I_{ion} is the total transmembrane ionic current, and I_{st} is the stimulus current applied by the sinoatrial node. Figure 1 shows the graph of transmembrane potential over time. Note that the sign of I_{ion} determines the direction in which electrons are flowing [16]. A negative value for I_{ion} means that we have an inward flow of electrons, and so the potential has a positive derivative, and the potential is increasing. A positive value for I_{ion} means that electrons are flowing outward, and so the derivative is negative, and the potential is decreasing.

In the Luo-Rudy model we have six ionic currents that are determined, directly or indirectly, by gating variables [17]. The values of these gating variables are determined by the solution of six nonlinear ordinary differential equations (ODEs). These equations are determined by rate

parameters α_y and β_y , where y is any gating variable. These equations take the general form:

$$\frac{dy}{dt} = \frac{y_\infty - y}{\tau_y}, \quad (2.2.2)$$

where

$$y_\infty = \frac{\alpha_y}{\alpha_y + \beta_y}, \quad (2.2.3)$$

$$\tau_y = \frac{1}{\alpha_y + \beta_y}. \quad (2.2.4)$$

The remaining ODE in the Luo-Rudy model describes calcium concentration in the cell:

$$\frac{d([\text{Ca}]_i)}{dt} = -10^{-4}I_{\text{si}} + 0.07(10^{-4} - [\text{Ca}]_i), \quad (2.2.5)$$

where $[\text{Ca}]_i$ is the intracellular calcium concentration, and I_{si} is the slow inward calcium current [17]. The six gating equations of the form (A.1.22) are coupled with (2.2.2) and (2.2.5) to form the complete Luo-Rudy model.

See section A.1 for a complete listing of the Luo-Rudy model.

In 1994 Luo and Rudy published an improvement to this model, now known as the Luo-Rudy Phase II model [18, 19]. This model gives a more accurate physiological model by including the actions of ionic pumps and changes in ionic concentrations.

2.3 The Model of Courtemanche et al.

In 1998 Marc Courtemanche, Rafael Ramirez, and Stanley Nattel developed a model of human atrial action potentials [7]. This model was a response to new findings that show there are some important differences in human action potentials when compared to other mammals frequently used in models and tests. Courtemanche et al. developed this model with human data and supplemented it with animal data when needed. The Courtemanche et al. model is an extension of the Luo-Rudy Phase II model. It consists of 21 ODEs, which are listed in section A.2.

2.4 The Model of Winslow et al.

In 1999 Raimond Winslow, Jeremy Rice, Saleet Jafri, Eduardo Marbn, and Brian O'Rourke developed a model of canine ventricular tissue [25]. This model is based on a guinea pig model that was an extension of the Luo-Rudy Phase II model. The Winslow et al. model was developed using experimental data to modify the guinea pig model so that it would simulate canine ventricular tissue. The canine heart is more similar to a human heart than a guinea pig heart, and there are

more canine heart data than human heart data available. The Winslow et al. model is particularly detailed when describing the dynamics of Ca^{2+} , which is an important consideration in heart failure. The Winslow et al. model consists 32 ODEs, listed in section A.3, making it the most complex of the models in this study.

2.5 The Model of Puglisi and Bers

In 2001 Jose Puglisi and Donald Bers developed a model of rabbit ventricular tissue [20]. Although rabbit ventricular tissue is used frequently in physical experiments, no mathematical model had been previously developed for it. This model was adapted from the Luo-Rudy model to include data from the literature and from the joint laboratory of Puglisi and Bers. Much of the emphasis in this model was placed on the user interface as this was designed to be a learning aid for students. The convenient user interface was designed also as a tool for researchers to reproduce experimental and physical results on the computer. Thus, physiological accuracy is also of paramount importance. Puglisi and Bers give particular detail to calcium handling so that the model can accurately simulate heart failure. This model contains 17 ODEs. This model is also referred to as the *LabHeart* [20] model. See A.4 for a complete listing of equations.

CHAPTER 3

MATHEMATICAL BACKGROUND

3.1 Differential Equations

The material in this section is largely adapted from [6].

A *differential equation* is an equation that contains an unknown function and its derivatives. If the unknown function \mathbf{y} depends only on one independent variable t , then the equation

$$\mathbf{F}(t, \mathbf{y}, \mathbf{y}', \dots, \mathbf{y}^{(n)}) = \mathbf{0}. \quad (3.1.1)$$

is called an *ordinary differential equation* (ODE). In this study we are primarily concerned with ODEs of the form:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}).$$

That is, we are looking at equations that describe the rate of change of \mathbf{y} over time.

Sometimes we may need more than one independent variable to mathematically model a given process. For example, modelling heat flow through a metal rod requires independent variables in both time and space. If we have more than one independent variable, then we may take a derivative with respect to any one of the independent variables while treating the other independent variables as constants. This is called a *partial derivative*. If a differential equation contains a partial derivative of an unknown function of more than one independent variable, it is called a *partial differential equation*.

A *solution* of the ODE (3.1.1) is a function with n derivatives that satisfies (3.1.1). When we have known initial values for a function and $n - 1$ of its derivatives we have an *initial value problem* (IVP). This initial value problem usually gives us a *specific solution* to an ODE, as opposed to a *general solution* that will contain arbitrary constants of integration.

The theory of ODEs is both rich and verbose, but here we only list one result that is of importance in our study:

Theorem 3.1.1. (Existence and Uniqueness) *Consider the initial value problem:*

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)).$$

$$\mathbf{y}(0) = \mathbf{y}_0.$$

Let \mathbf{f} be continuous for all (t, \mathbf{y}) in D :

$$D = \{0 \leq t \leq b, -\infty < \|\mathbf{y}\| < \infty\}.$$

Let \mathbf{f} satisfy a Lipschitz condition for all (t, \mathbf{y}) in D :

$$\|\mathbf{f}(t, \mathbf{y}) - \mathbf{f}(t, \mathbf{y}^*)\| \leq L\|\mathbf{y} - \mathbf{y}^*\|$$

for some constant $0 < L < \infty$ and all $t, \mathbf{y}, \mathbf{y}^*$ in D . Then for any \mathbf{y}_0 there exists a unique and differentiable solution to the ODE in $[0, b]$.

3.2 Numerical Methods

3.2.1 Basic Concepts

The material in subsections 3.2.1, 3.2.2, and 3.2.3 has been mainly adapted from [22] and [14].

Although analytical methods exist to solve differential equations, we are often faced with a differential equation that cannot be solved analytically. Differential equations or systems of differential equations that model problems in all of the mathematical sciences are often large, complicated, and nonlinear. When analytical methods are unavailable, one may use numerical methods to approximate the solution to a differential equation.

The simplest numerical method for approximating the solution of an IVP is *Euler's method*, also known as *Forward Euler*. Euler's method approximates the solution of the initial value problem

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \mathbf{y}(0) = \mathbf{y}_0,$$

on the interval $[0, b]$. One step of Euler's method is given by

$$\begin{aligned} \mathbf{y}_n &= \mathbf{y}_{n-1} + \Delta t_n \mathbf{f}(t_{n-1}, \mathbf{y}_{n-1}), \\ t_n &= t_{n-1} + \Delta t_n. \end{aligned}$$

Each step approximates the solution at time t_n . Note that Δt_n need not be constant. Recalling that the slope of the tangent line at time t for $\mathbf{y}(t)$ is given by

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)),$$

we can see that Forward Euler is an approximation of the tangent line at each interval. See Figure

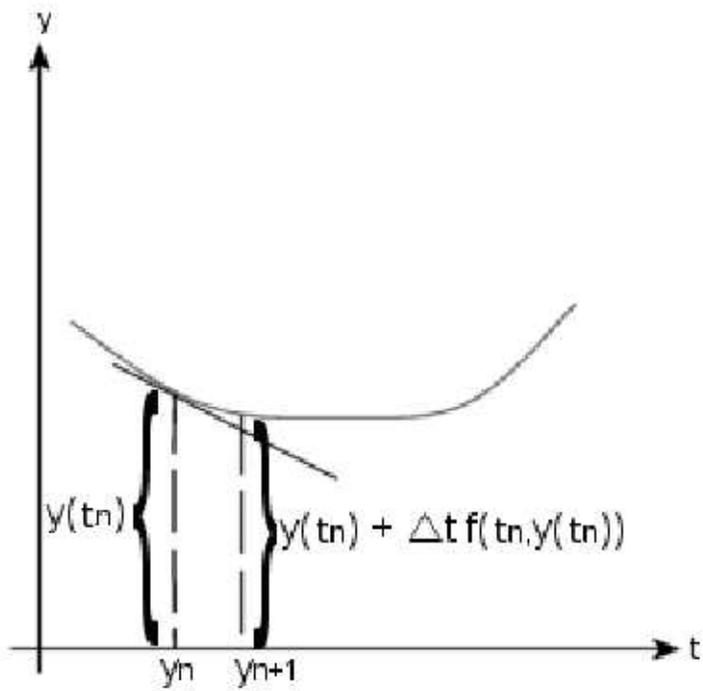


Figure 3.1: At each interval Forward Euler approximates the derivative using the tangent line at y_n . We perform this approximation once per interval and then take the set of all these approximations to be the approximate solution.

3.1. That is, geometrically, the approximate solution produced by Forward Euler is the union of each of these approximations over the entire solution interval. So, in theory, as Δt approaches zero, the approximation given by Forward Euler approaches the true solution. In other words, given infinite precision, Forward Euler's approximation converges to the true solution as Δt approaches zero, but in practice we are limited by the finite amount of precision available on a given computer.

This notion of convergence is very important when discussing numerical methods. Each numerical method has an *order of convergence*. We say a method is of order p if

$$\mathbf{y}_n - \mathbf{y}(t_n) = O(\Delta t^{p+1}). \quad (3.2.1)$$

It can be shown that Forward Euler is of order 1 [22], which is a lower order of convergence than we generally would like to have in practice.

3.2.2 Error and Stability

No matter how sound the numerical method used, the approximation process naturally produces some error. For example, we introduce error when we discretize our continuous equation, and further error is introduced when the solution is computed with finite precision. To obtain an acceptable approximation a numerical method must limit all sources of error. One method to control the error is to estimate the error at each time step and then, if necessary, adjust the size of the step. If the error is too large, the step is rejected, and the solver will try again using a smaller step size. If the error is much smaller than necessary the solver can increase the size of the next step and thus increase efficiency.

If the stepsize used is too large, then the approximation process generally becomes *unstable*. If the choice of step size of an approximation is determined by stability rather than accuracy, then the ODE is said to be *stiff*¹. Generally, the step size required for a stiff problem is much smaller than accuracy requirements would dictate; for efficiency purposes we want to be able to choose a method based only the accuracy requirements. We can understand this to some extent by considering *absolute stability theory*.

In absolute stability theory we consider the numerical approximation of the solution of

$$y' = \lambda y, \quad t \geq 0, \quad y(0) = y_0, \quad \lambda \in \mathbb{C},$$

with a given numerical method. The analytical solution of this equation is

$$y(t) = y_0 e^{\lambda t}, \quad (3.2.2)$$

¹Arguably, the definition of a stiff problem is not entirely universal, but this description of stiffness suffices for our study.

and so we have

$$\lim_{t \rightarrow \infty} |y_n| \rightarrow 0 \quad (3.2.3)$$

if and only if

$$Re(\lambda) \leq 0.$$

We say the *region of absolute stability* of the given numerical method is the set of all $\lambda\Delta t \in \mathbb{C}$ such that (3.2.3) holds and $\|y_{n+1}\| \leq \|y_n\|$. It is for these $\lambda\Delta t$ the method is absolutely stable. A special case is when the region of absolute stability contains the left hand side of the complex plane. Methods with this property are called *A-stable*. With A-stable methods we have no restrictions on Δt due to stability (at least for the model problem (3.2.2)), and hence they are a good choice for stiff methods.

If the numerical method is able to produce a stable approximation we are then interested in how close the approximation is to the true solution. When the true solution is known, this is simple: we can simply compare the two. Otherwise, we must make an estimate. In order to make an estimate on the error in the approximation, we generate a reference solution by lowering the error tolerance until we produce two approximations that are identical within a desired number of significant digits. We can then compare the approximation, \mathbf{Y} , to the reference solution, $\hat{\mathbf{Y}}$. A popular way to do this in the literature on heart simulation is the *Relative Root Mean Squared (RRMS) error*:

$$RRMS = \sqrt{\frac{\sum_{i=1}^N (\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^2}{\sum_{i=1}^N \hat{\mathbf{Y}}_i^2}},$$

where $\mathbf{Y}_i = \mathbf{Y}(t_i)$ and $\hat{\mathbf{Y}}_i = \hat{\mathbf{Y}}(t_i)$ are the numerical approximations at time $t_i \in [0, b], i = 1, 2, \dots, N$.

In biomedical research 5% RRMS error is considered acceptable; our goal is, therefore, to produce the most efficient solution we can with 5% or less RRMS error.

3.2.3 Runge-Kutta Methods

A general class of numerical methods for solving ODEs is called Runge-Kutta methods [14]. Euler's method is the simplest of this class of methods. We aim to improve on Euler's method by improving the order of convergence and hence the accuracy of our approximation. A general s -stage Runge-Kutta method has the form:

$$\mathbf{k}_i = \mathbf{f} \left(t_{n-1} + \Delta t c_i, \mathbf{y}_{n-1} + \Delta t \sum_{j=1}^s a_{ij} \mathbf{k}_j \right), \quad i = 1, 2, \dots, s,$$

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t \sum_{i=1}^s b_i \mathbf{k}_i.$$

These equations can be summarized by means of the *Butcher Tableau* [14]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

or

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

A Runge-Kutta method is explicit if \mathbf{A} is strictly lower triangular and implicit if any a_{ij} on or above the diagonal is non-zero. Implicit Runge-Kutta methods are useful for stiff problems [14].

The choice of parameters for a specific Runge-Kutta method is often made based on desired order requirements. In other words, we try to pick terms such that the truncation error is of a certain order. The idea is that repeated function evaluations are used to eliminate lower order truncation error terms. An example of a higher order method is the classical Runge-Kutta method, denoted by ERK4:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

This is a four stage, fourth order explicit Runge-Kutta method.

3.2.4 Implicit-Explicit Methods

When solving ODEs numerically we may find the ODE can be separated into two parts:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}) + \mathbf{g}(t, \mathbf{y}),$$

where $\mathbf{f}(t, \mathbf{y})$ is most efficiently approximated with one method and $\mathbf{g}(t, \mathbf{y})$ is most efficiently approximated with another method. In such cases, particularly when the efficient method for one part is inefficient for the other, a *splitting* technique may be used to obtain an approximation efficiently for the whole equation [13]. Here we break the approximation into two parts and use the appropriate method for each part of the equation.

In particular, if $\mathbf{f}(t, \mathbf{y})$ is such that it is best approximated with an explicit method and $\mathbf{g}(t, \mathbf{y})$ is such that it is best approximated with an implicit method, we may use an *Implicit-Explicit* (IMEX) method to approximate the solution to this equation efficiently [13]. An example of when an IMEX method would be useful is when $\mathbf{f}(t, \mathbf{y})$ consists of non-stiff and/or nonlinear terms and that $\mathbf{g}(t, \mathbf{y})$ consists of stiff and/or linear terms.

As an example [3] we can look at the combination of Forward and Backward Euler, given by the respective Butcher Tableaux:

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

which when padded, to simplify the form of the order conditions, become:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline & 0 & 1 \end{array} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}$$

This gives the following IMEX method:

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t (\mathbf{f}(\mathbf{y}_{n-1}) + \mathbf{g}(\mathbf{y}_n)).$$

In a more general sense we are considering an s -stage implicit method with coefficients $\mathbf{A}, \mathbf{c}, \mathbf{b}$ in the usual Butcher notation with an $(s+1)$ -stage explicit method with coefficients $\hat{\mathbf{A}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$. Here we have $\hat{\mathbf{c}} = (0 \ \mathbf{c})^T$. Let $\sigma = s+1$. One step of the IMEX method is given by [3]:

Set

$$\hat{\mathbf{K}}_1 = \mathbf{f}(\mathbf{y}_{n-1}).$$

Then for $i = 1, \dots, s$:

- Solve for \mathbf{K}_i

$$\mathbf{K}_i = \mathbf{g}(\mathbf{y}_i).$$

where

$$\mathbf{y}_i = \mathbf{y}_{n-1} + \Delta t \sum_{j=1}^i a_{ij} \mathbf{K}_j + \sum_{j=1}^i \hat{a}_{i+1,j} \hat{\mathbf{K}}_j.$$

- Evaluate

$$\hat{\mathbf{K}}_{i+1} = \mathbf{f}(\mathbf{y}_i).$$

Finally, evaluate

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t \sum_{j=1}^s b_j \mathbf{K}_j + \sum_{j=1}^{\sigma} \hat{b}_j \hat{\mathbf{K}}_j.$$

See [3] for further details.

This thesis considers the numerical method ARK5(3)8L[2]SA, also known as ARK5 [15], as an alternative to standard methods used to approximate the solution to cardiac electrophysiological models. ARK5 is a 2-additive Runge-Kutta method of order 5(4) with eight stages. In general, an n -additive embedded Runge-Kutta method consists of n Runge-Kutta methods to solve n different parts of the ODE. ARK5 uses two Runge-Kutta methods, one explicit and one implicit, and so it is an IMEX method that uses the procedure described in this section to take a step. The Butcher tableaux of ARK5 are listed in [15].

3.2.5 Implementation

All four models were implemented in `odeToJava` [2], a Java based initial value problem solving environment. Using `odeToJava` we performed numerical experiments using Forward Euler, ERK4, Dormand-Prince 5(4), and ARK5 and compared their performance.

Matlab implementations of the Luo-Rudy, Courtemanche et al., and Winslow et al. models were written by Mary MacLachlan, and a Matlab implementation of the Puglisi-Bers model was written by Sarah Healy. The Matlab code was used to verify that the `odeToJava` code had been implemented correctly and to verify results obtained using the `odeToJava` code. For example, the maximum stable step size of Forward Euler should be the same for both the Matlab and `odeToJava` code and any discrepancy is evidence of a bug. The Matlab code was also used to gain useful information using specialized Matlab routines. For example, Matlab's `spy` routine was used to produce the figures in Section 4.3, as there is no such feature in `odeToJava`.

The mathematical models were also partially implemented in Maple. This was also done to verify that the `odeToJava` code was implemented properly. Maple is a useful tool to perform such a task as it obtains results symbolically instead of numerically; one advantage of this is that we may avoid making the same mistake in both implementations because we are forced to use a very different process to obtain results.

CHAPTER 4

RESULTS

After the models were implemented, their solutions were approximated with a variety of numerical methods. This chapter lists the results of the comparisons of these numerical methods. We approximated the solution of the four cardiac electrophysiological models using the IVP solvers described in Chapter 3. As an initial value we use, with the exception of voltage, the values of all variables while the heart is in its resting state. In some models, initial value for voltage was altered to account for the current from the sinoatrial node. The discontinuous nature of the current from the sinoatrial node can cause numerical problems; we alter the alter the initial value of the voltage such that we remove this numerical problem yet maintain physiological accuracy. The initial values we use for voltage in each one of the models are listed in Table 4.1. See [17], [7], [25], [20] for complete listings of the initial values.

The models were solved over a time interval that represents one cardiac cycle. The Luo-Rudy model was considered on the interval [0,450] milliseconds, the Courtemanche et al. model was considered on the interval [0,500] milliseconds, the Winslow et al. model was considered on the interval [0,300] milliseconds, and the Puglisi-Bers model was considered on the interval [0,330] milliseconds. The different solution intervals are due to different physiological properties of the specific mammalian heart that the model represents.

Model	V_{rest}
Luo-Rudy	-35
Coutemanche et al.	-81.2
Winslow et al.	-35
Puglisi-Bers	-85.5

Table 4.1: Initial values for $V = V_{rest}$

4.1 Constant Step Size Tests

4.1.1 Explicit Runge-Kutta Methods

Here we have determined the maximum step size, to three significant figures, that Forward Euler and ERK4 produce an approximation with less than 5% RRMS error. For these methods this is also generally the same step size that produces a stable solution. The results are summarized in Table 4.2.

Model	Forward Euler		ERK4	
	Step Size	Time (seconds)	Step Size	Time (seconds)
Luo-Rudy	0.0134	0.224	0.0186	0.677
Coutemanche et al.	0.0194	0.805	0.0268	2.292
Winslow et al.	0.000107	40.369	0.00013	130.83
Puglisi-Bers	0.0108	0.454	0.0148	1.078

Table 4.2: Maximum step size with less than 5 % RRMS error using constant step size ERK methods. Execution time, in seconds, of each model with the listed step size is also given.

4.1.2 NSFD Method

An alternative to using Forward Euler or ERK4 is to use a *non-standard finite difference* (NSFD) method. A NSFD method developed in [24] solves exactly the gating equations in the form of (2.2.2). The exact solution of (2.2.2),

$$y_n = y_\infty + (y_{n-1} - y_\infty)e^{-\frac{\Delta t}{\tau y}},$$

is used to advance the solution of the gating equation while Forward Euler is used to advance the solution of the remaining equations. Using this method the Luo-Rudy model is no longer stiff [24].

Here we have determined the maximum step size, to three significant figures, that this NSFD method produces an approximation with less than 5% RRMS error. Note that one must exercise care when doing this for an NSFD method as error produced by an NSFD method does not necessarily increase as one increasing the step size; it is possible for it to decrease as well. The results are summarized in Table 4.3

4.2 Variable Step Size Tests

The next test performed compared two numerical methods with variable step sizes: Dormand-Prince 5(4) and ARK5. We list the results for each of the models below.

Model	Max Step Size	Time
Luo-Rudy	0.250	0.046
Coutemanche et al.	0.345	0.081
Winslow et al.	0.00028	57.32
Puglisi-Bers	0.43	0.066

Table 4.3: Maximum step size with less than 5 % error using constant step size using an NSFD method.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-12	5.198E-14	20.29	2.150E-2
-11	4.917E-14	1.316	3.157E-2
-10	4.154E-14	0.946	4.410E-2
-9	3.6164E-14	0.611	7.0104E-2
-8	5.9490E-13	0.551	7.8933E-2
-7	8.8183E-12	0.539	8.4065E-2
-6	1.4975E-10	0.514	8.6455E-2
-5	3.2922E-09	0.529	8.7327E-2
-4	9.6103E-08	0.516	8.7531E-2
-3	7.1231E-05	0.538	8.7599E-2
-2	0.0170	0.52	9.2573E-2

Table 4.4: Luo-Rudy Dormand-Prince variable step size results.

$rtol =$ $atol =$ 1×10^i	RRMS Error	Time (seconds)	Average Step Size
-13	4.2557E-7	6.048	1.6865E-3
-12	4.3221E-7	3.603	5.4534E-3
-11	4.6300E-7	2.801	1.8703E-2
-10	6.4571E-7	1.164	6.5607E-2
-9	1.2386E-6	0.583	1.9728E-1
-8	1.6158E-6	0.253	4.8859E-1
-7	2.4671E-6	0.118.	1.178
-6	5.0677E-6	0.069	2.486
-5	1.9600E-6	0.048	4.945
-4	2.6724E-5	0.038	8.333
-3	1.1127E-4	0.034	11.53
-2	1.000E-3	0.032	15.00

Table 4.5: Luo-Rudy ARK5 variable step size results.

The results for the Luo-Rudy model are listed in Tables 4.4 and 4.5. With Dormand-Prince the execution time is approximately constant for all numerical approximations with an acceptable amount of error; relaxing the error tolerance offers no significant improvement in execution time. This is a telltale sign of stiffness in the Luo-Rudy model. With ARK5 we can always produce a solution well within the maximum RRMS error, and we see a clear improvement in execution time when we relax the error tolerance. Using ARK5 as the numerical method for the Luo-Rudy model can produce an acceptable solution nearly 20 times faster than Dormand-Prince.

The results for the Courtemanche et al. model are listed in Tables 4.6 and 4.7. Dormand-Prince can produce an acceptable approximation as quickly as 0.6 s, but ARK5 can produce an acceptable approximation up to five times faster.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-12	8.874E-4	4.833	2.943E-2
-11	1.046E-3	3.220	1.964E-2
-10	1.200E-2	2.576	2.943E-2
-9	1.322E-3	2.358	4.136E-2
-8	1.575E-3	2.122	4.937E-2
-7	5.664E-3	2.027	5.3653E-2
-6	7.054E-3	2.009	5.7453E-2
-5	2.8643E-2	2.000	6.0125E-2
-4	2.8691E-2	1.981	6.0628E-2
-3	4.0326E-2	0.637	1.6556E-1
-2	1.3043E-1	0.6000	2.1204E-1

Table 4.6: Courtemanche et al. Dormand-Prince variable step size results.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-12	9.822E-3	28.12	0.0306
-11	1.037E-2	15.66	0.01668
-10	1.233E-2	11.81	0.02781
-9	1.636E-2	5.56	0.08474
-8	1.842E-2	3.23	0.1589
-7	2.088E-2	1.55	0.3160
-6	2.233E-2	0.766	0.6613
-5	2.9812E-2	0.404	2.2123
-4	2.9195E-2	0.262	2.8248
-3	6.7496E-2	0.537	2.3584
-2	1.1757E-2	0.137	11.904

Table 4.7: Courtemanche et al. ARK5 variable step size results.

The results for the Winslow et al. model are listed in Tables 4.8 and 4.9. Dormand-Prince produces an acceptable numerical approximation in approximately the same amount of time, regardless of the error tolerances. This is a sign of stiffness in the Winslow et al. model. ARK5 shows a dramatic improvement in time as we weaken the error tolerances. The error produced by

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-11	1.0384E-14	32.770	4.2768E-3
-10	1.8867E-14	32.467	4.3168E-3
-9	6.5554E-13	32.447	4.3308E-3
-8	5.1820E-12	32.501	4.33567E-3
-7	4.8173E-10	32.481	4.33780E-3
-6	5.5044E-8	32.463	4.339440E-3
-5	1.0285E-5	32.211	4.343587E-3
-4	5.0398E-3	32.436	4.333590E-3
-3	3.6473E-2	30.037	4.881488E-3

Table 4.8: Winslow et al. Dormand-Prince variable step size results.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-11	4.5222E-8	81.534	1.3366E-2
-10	2.7646E-4	26.237	4.3265E-2
-9	2.77525E-4	7.638	1.4925E-1
-8	2.25193E-6	2.749	1.2087E-1
-7	2.86524E-4	1.137	1.0714
-6	3.02370E-4	0.604	2.1126
-5	1.10832E-5	0.432	2.9801
-4	5.41536E-5	0.471	2.8125
-3	1.83147E-4	0.465	3.0612

Table 4.9: Winslow et al. ARK5 variable step size results.

ARK5 behaves erratically, but we can produce an approximation within the desired RRMS error in 0.465 seconds. For the model of Winslow et al. ARK5 is approximately up to 70 times faster than Dormand-Prince.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-13	8.46E-4	18.884	0.01644
-12	6.33E-4	10.9	0.02904
-11	4.22E-4	6.066	0.05182
-10	3.65E-4	3.185	0.09906
-9	3.18E-3	1.593	0.19400
-8	1.15E-3	0.0803	0.37078
-7	8.48E-3	0.0411	0.70663
-6	0.057	0.0261	1.1224
-5	9.96E-3	0.0201	0.7236
-4	9.21E-3	0.0180	1.2132
-3	1.578	0.0112	1.4798
-2	0.118	0.0163	0.0871

Table 4.10: The results of numerical solutions of the Puglisi-Bers using the IMEX method ARK5 with variable step sizes.

$rtol = atol = (1 \times 10^i)$	RRMS Error	Time(seconds)	Average Step Size
-13	1.20E-4	3.582	0.01503
-12	1.37E-4	2.821	0.01960
-11	1.69E-4	2.382	0.02365
-10	1.75E-4	2.106	0.02695
-9	1.45E-4	1.996	0.02928
-8	1.05E-4	1.885	0.03072
-7	1.08E-4	1.843	0.03157
-6	1.34E-4	1.823	0.03199
-5	2.177E-4	1.812	0.03219
-4	3.25E-4	1.809	0.03228
-3	1.112E-3	1.804	0.03231
-2	0.0303	1.525	0.04092

Table 4.11: The results of numerical solutions of the Puglisi-Bers using Dormand-Prince with variable step sizes.

The results for the Puglisi-Bers model are listed in Tables 4.10 and 4.11. We find that ARK5 can produce an acceptable result up to 84 times faster than Dormand-Prince.

4.3 Sparsity Patterns

A *sparsity pattern* can be thought of as a map of a matrix: it describes which entries of a matrix are always zero and which entries can be non-zero; in our case this is across the entire solution interval of an ODE. A blue dot in the sparsity pattern represents a point that can be non-zero and a blank

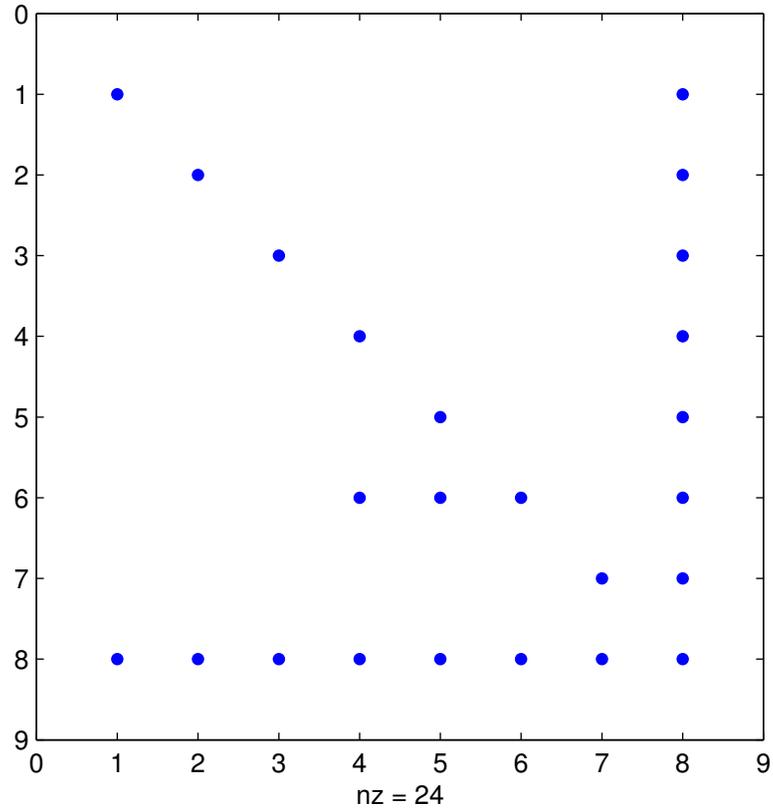


Figure 4.1: Sparsity pattern for the Jacobian of the Luo-Rudy model.

spot represents a point that is always zero. The sparsity pattern for the Jacobian of all four models was generated. If a point is always zero in the Jacobian, we should not to consider it for the sake of efficiency when we perform Gaussian Elimination [10], a process required when using an implicit solver. This means that we may further optimize the IMEX results from the previous section by altering our Gaussian Elimination code to take advantage of these sparsity patterns. See Figures 4.1, 4.2, 4.3, and 4.4.

4.4 Eigenvalues

The eigenvalues for the Jacobian matrix of each of the four models were computed at every 1 ms in the approximation to the solution. We then computed the maximum and minimum real parts of the set of all eigenvalues at each 1 ms. These eigenvalues can give us an indication of a problem's stiffness. The ratio of the maximum modulus of the largest and smallest eigenvalues is called the *stiffness ratio* [14]. Systems of ODEs with large stiffness ratios are likely to be stiff, although this is not a hard and fast rule [14]. These eigenvalues can give us information about how to optimize

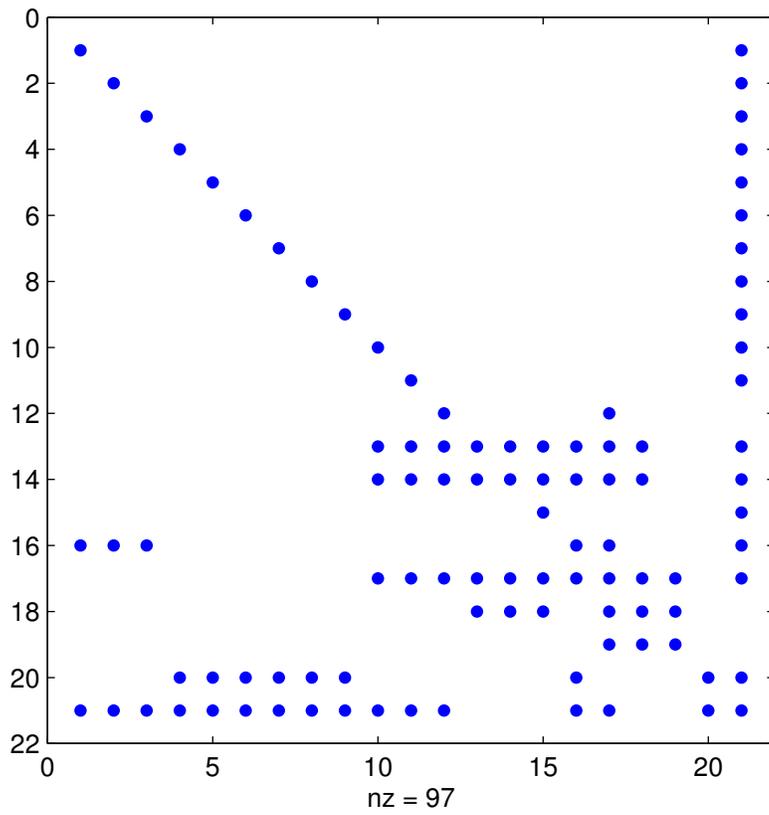


Figure 4.2: Sparsity pattern for the Jacobian of the Courtemanche et al. model.

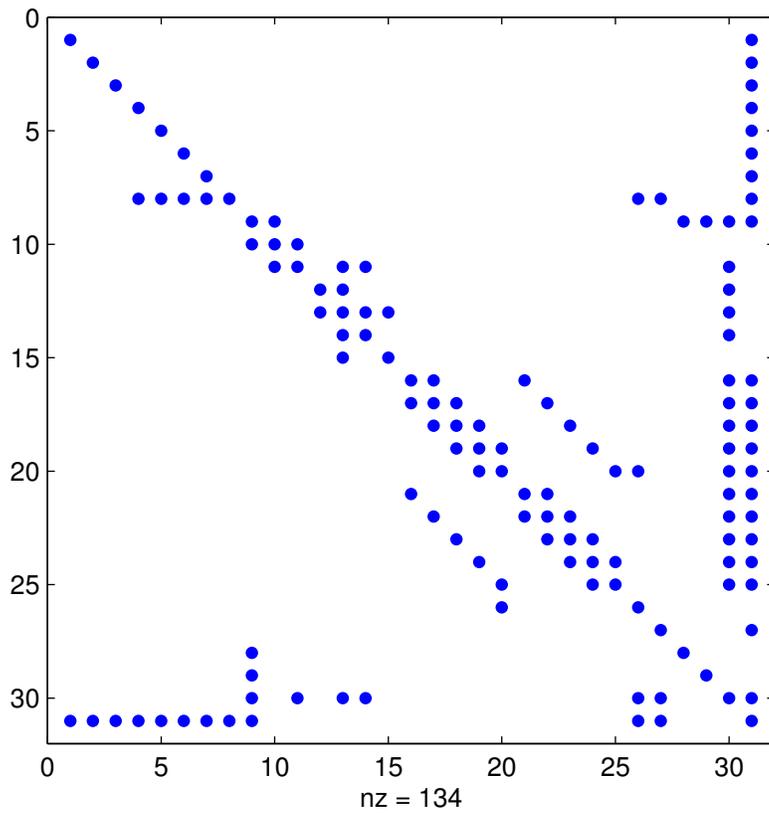


Figure 4.3: Sparsity pattern for the Jacobian of the Winslow et al. model.

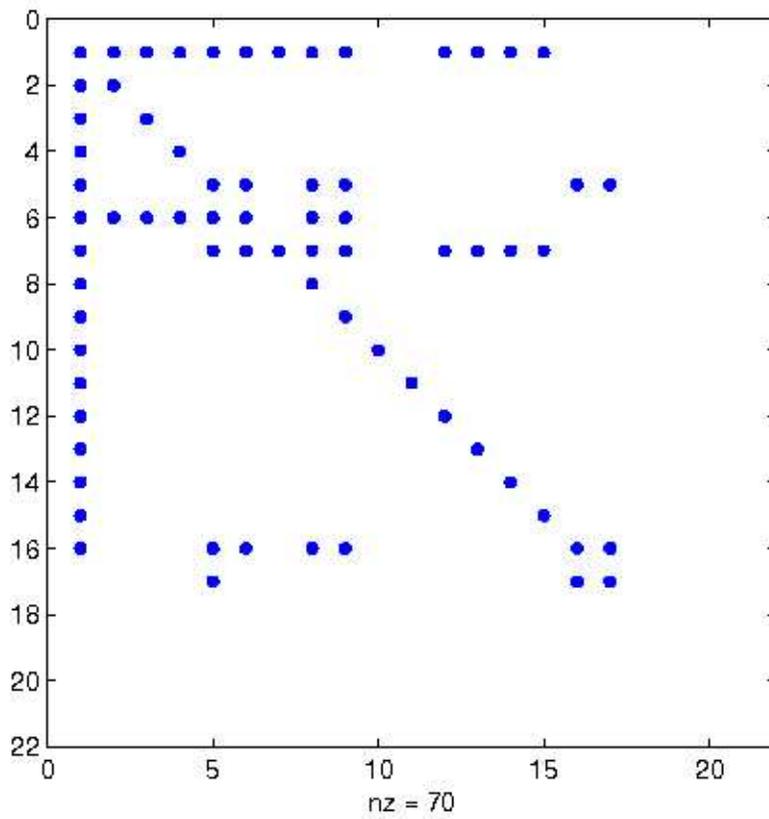


Figure 4.4: Sparsity pattern for the Jacobian of the Puglisi-Bers model.

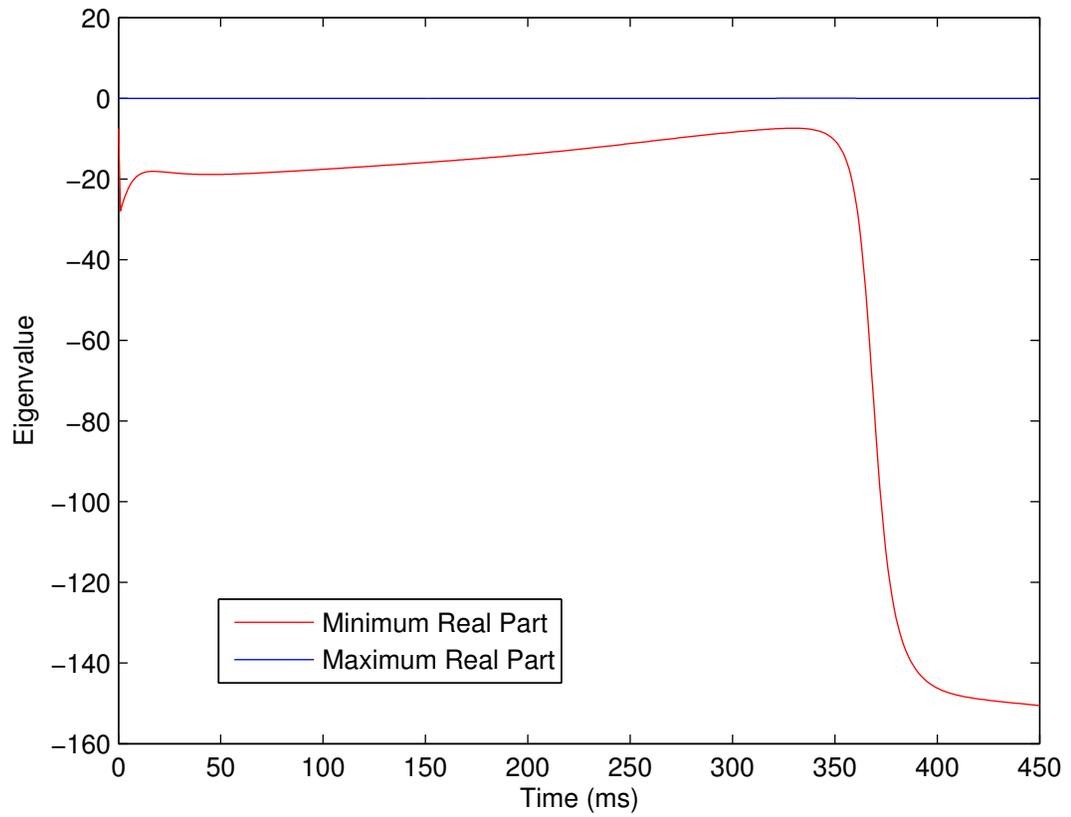


Figure 4.5: Eigenvalues for the Jacobian of the Luo-Rudy model.

the IMEX method. The results are displayed in Figures 4.5, 4.6, 4.7, and 4.8.

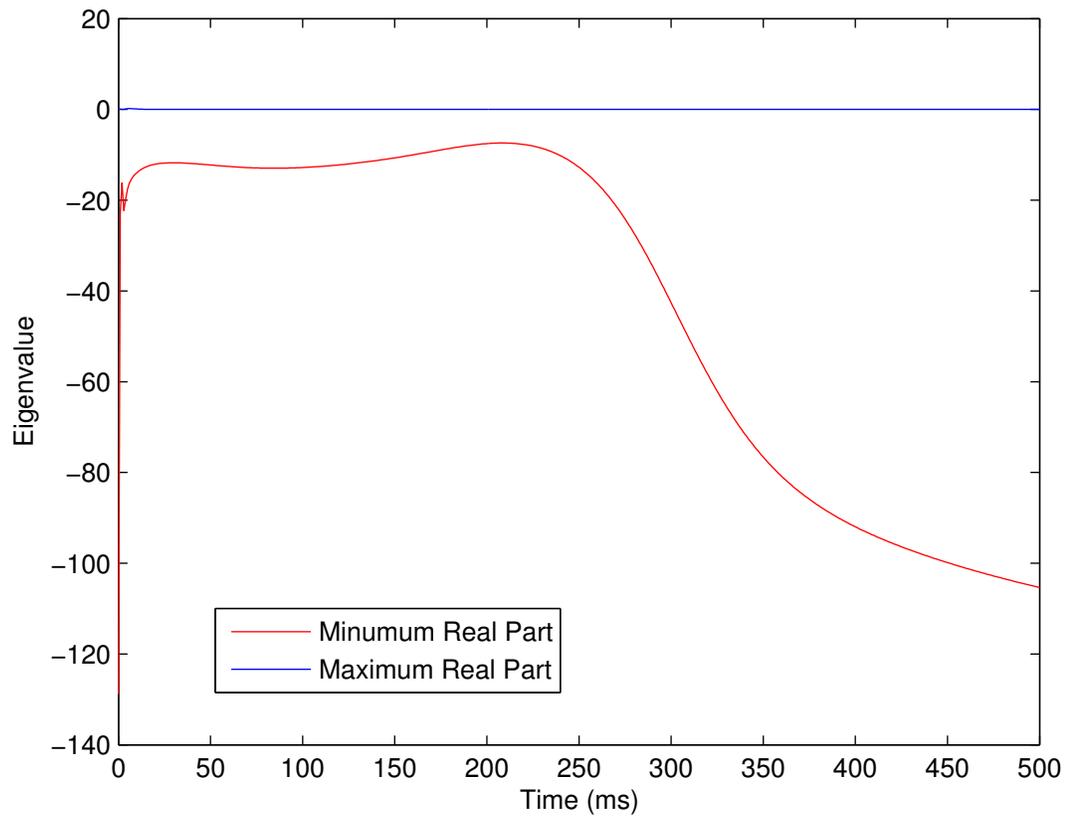


Figure 4.6: Eigenvalues for the Jacobian of the Courtemanche et al. model.

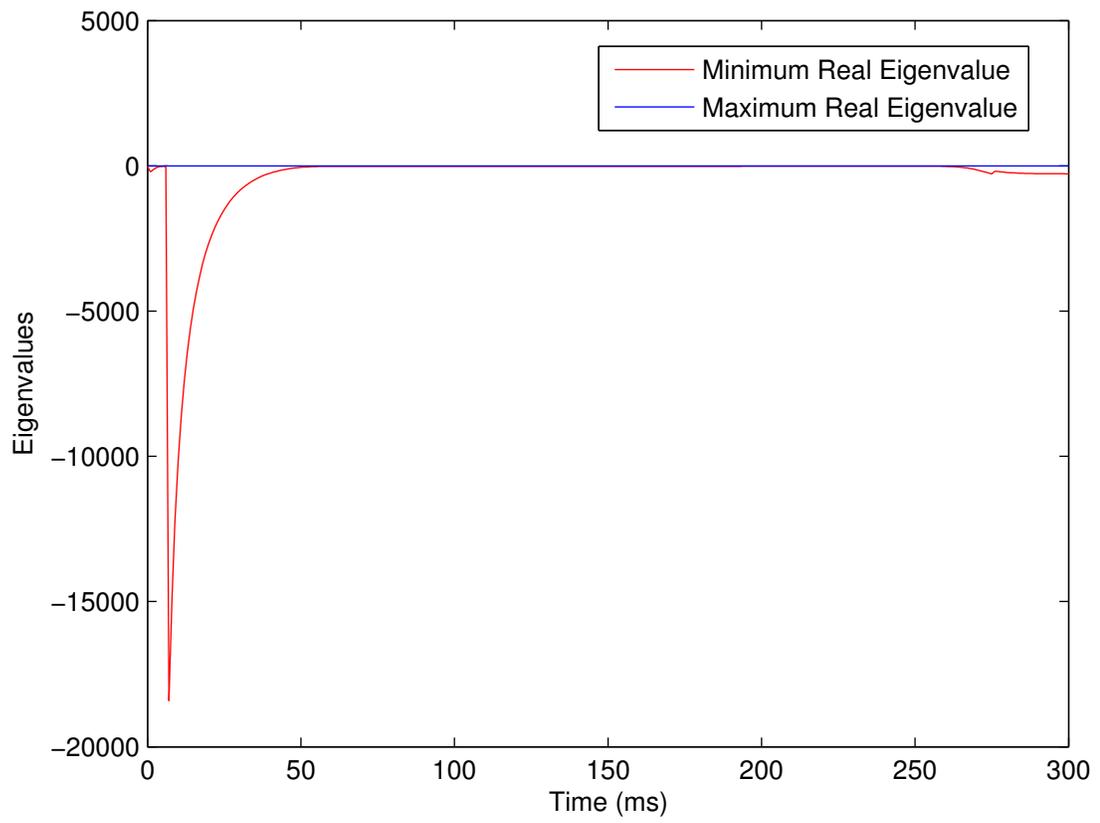


Figure 4.7: Eigenvalues of the Jacobian of the Winslow et al. model.

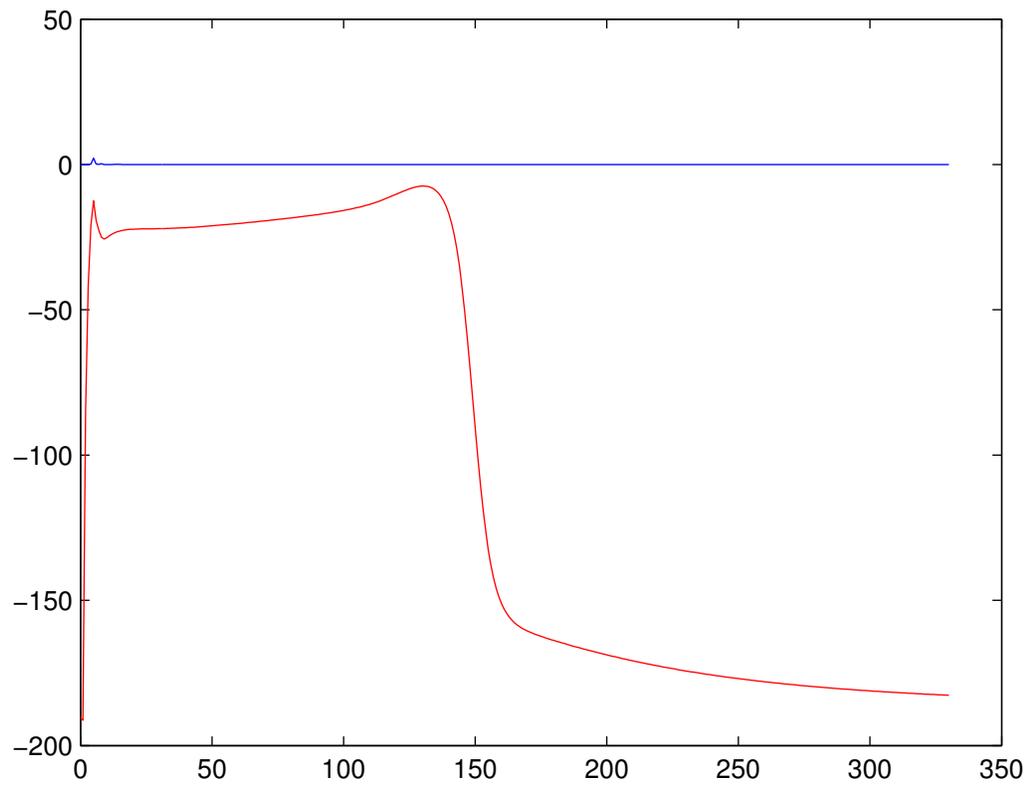


Figure 4.8: Eigenvalues for the Jacobian of the Puglisi-Bers model.

CHAPTER 5

CONCLUSION

In this thesis we considered the numerical approximation of solutions ODEs found in cardiac electrophysiological models. In particular, we compared the efficiency of IMEX methods for performing the numerical approximation to the solution of the ODEs found in four different electrophysiological models to standard numerical methods. When comparing the IMEX method ARK5 to Forward Euler, the results indicate that ARK5 is approximately 7 times faster for the Luo-Rudy model, approximately 6 times faster for the Courtemanche et al. model, approximately 87 times faster for the Winslow et al. model, and approximately 25 times faster for the Puglisi-Bers model. Overall, ARK5 is the most efficient numerical method for both the Winslow et al., Puglisi-Bers and Luo-Rudy models, and the NSFD method is the most efficient for the Courtemanche et al. model. In particular, the ARK5 IMEX method is 1.4, 64, and 4 times faster than any other numerical method we considered for the Luo-Rudy, Winslow et al., and Puglisi-Bers models, respectively. The NSFD method is 1.7 times faster than any other numerical method we considered for the Courtemanche et al. model. We also computed the eigenvalues and sparsity pattern of the Jacobian of each model so that we can further optimize the IMEX method in the future.

This work has several natural extensions. The sparsity patterns, in Section 4.3, could be used to optimize the linear algebra in the IMEX method. It may be possible to further increase efficiency of the IMEX method by optimizing the how we split the components of the ODEs between the implicit and explicit numerical methods. We may also gain efficiency by creating a specialized IMEX method or methods for these cardiac electrophysiological models. In this thesis we have only considered models of one cell so further study of models involving large numbers of cells coupled in two or three dimensions would also be useful.

BIBLIOGRAPHY

- [1] <http://www.cellml.org/examples/repository/PB.model.2001.doc.html>.
- [2] <http://www.netlib.org/ode/odeToJava.tgz>.
- [3] ASCHER, U. M., RUUTH, S. J., AND SPITERI, R. J. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* 25, 2-3 (1997), 151–167. Special issue on time integration (Amsterdam, 1996).
- [4] BEELER, G., AND REUTER, H. Reconstruction of the action potential of ventricular myocardial fibers. *J. Physiol. (Lond)* 268 (1977), 177–210.
- [5] CLEMENTS, C. J. Nonlinear wave propagation in an anisotropic medium. Master’s thesis, Dalhousie University, 1996.
- [6] CODDINGTON, E. A. *An introduction to ordinary differential equations*. Prentice-Hall Mathematics Series. Prentice-Hall Inc., Englewood Cliffs, N.J., 1961.
- [7] COURTEMANCHE, M., RAMIREZ, R. J., AND NATTEL, S. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *Am J Physiol.* 275 (1998), H301–H321.
- [8] DIMASI, J. A., HANSEN, R. W., AND GRABOWSKI, H. G. The price of innovation: new estimates of drug development costs. *Journal of Health Economics* 22 (2003), 151–185.
- [9] DOWNEY, J. M., AND HEUSCH, G. Sequence of cardiac activation and ventricular mechanics. In *Heart Physiology and Pathophysiology*, N. Sperelakis, Y. Kurachi, A. Terzic, and M. V. Cohen, Eds., fourth ed. Academic Press, 2001.
- [10] DUFF, I. S., ERISMAN, A. M., AND REID, J. K. *Direct methods for sparse matrices*, second ed. Monographs on Numerical Analysis. The Clarendon Press Oxford University Press, New York, 1989. Oxford Science Publications.
- [11] GREENSTEIN, J. L. *Local Control of Calcium Release and its Implications for Cardiac Myocyte Properties*. PhD thesis, The Johns Hopkins University, Baltimore, Maryland, USA, April 2002.
- [12] HODGKIN, A., AND HUXLEY, A. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. (Lond)* 117 (1952), 500–544.
- [13] HUNSDORFER, W., AND VERWER, J. *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2003.
- [14] ISERLES, A. *A first course in the numerical analysis of differential equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 1996.
- [15] KENNEDY, C. A., AND CARPENTER, M. H. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.* 44, 1-2 (2003), 139–181.
- [16] KLÉBER, A. G., AND RUDY, Y. Basic mechanisms of cardiac impulse propagation and associated arrhythmias. *Physiol. Rev.* 84 (2004), 431–488.

- [17] LUO, C., AND RUDY, Y. A model of ventricular cardiac action potential. *Circ. Res.* 68 (1991), 1501–1526.
- [18] LUO, C., AND RUDY, Y. A dynamic model of the cardiac ventricular action potential. i. simulations of ionic currents and concentration changes. *Circ. Res.* 74 (1994), 1071–1096.
- [19] LUO, C., AND RUDY, Y. A dynamic model of the cardiac ventricular action potential. ii. afterdepolarizations, triggered activity, and potentiation. *Circ. Res.* 74 (1994), 1097–1113.
- [20] PUGLISI, J. L., AND BERS, D. M. Labheart: an interactive computer model of rabbit ventricular myocyte ion channels and ca transport. *Am. J. Physiol. Cell Physiol.* 281 (2001), C2049–C2060.
- [21] RANDELL, D., BURGGREN, W., AND FRENCH, K. *Eckert Animal Physiology*, fourth ed. W.H. Freeman and Company, 1997.
- [22] SHAMPINE, L. F., GLADWELL, I., AND THOMPSON, S. *Solving ODEs with MATLAB*. Cambridge University Press, Cambridge, 2003.
- [23] SPERELAKIS, N., KURACHI, Y., AND TERZIC, A. Cardiac arrhythmias: Reentry and triggered activity. In *Heart Physiology and Pathophysiology*, N. Sperelakis, Y. Kurachi, A. Terzic, and M. V. Cohen, Eds., fourth ed. Academic Press, 2001.
- [24] SPITERI, R. J., AND MACLACHLAN, M. C. An efficient non-standard finite difference scheme for an ionic model of cardiac action potentials. *J. Difference Equ. Appl.* 9, 12 (2003), 1069–1081. Dedicated to Professor Ronald E. Mickens on the occasion of his 60th birthday.
- [25] WINSLOW, R. L., RICE, J., JAFRI, S., MARBÁN, E., AND ROURKE, B. O. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, ii model studies. *Circ Res.* 84 (1999), 571–586.

APPENDIX A

MATHEMATICAL MODELS

A.1 The Model of Luo-Rudy

Inward currents

Fast sodium current

$$I_{\text{Na}} = \bar{G}_{\text{Na}} \cdot m^3 \cdot h \cdot j \cdot (V_m - E_{\text{Na}}) \quad (\text{A.1.1})$$

Activation gate, m

$$\frac{dm}{dt} = \alpha_m(1 - m) - \beta_m m \quad (\text{A.1.2a})$$

$$\alpha_m = \frac{0.32(V_m + 47.13)}{1 - e^{-0.1(V_m + 47.13)}} \quad (\text{A.1.2b})$$

$$\beta_m = 0.08e^{-V_m/11} \quad (\text{A.1.2c})$$

Fast inactivation gate, h

$$\frac{dh}{dt} = \alpha_h(1 - h) - \beta_h h \quad (\text{A.1.3a})$$

$$\alpha_h = \begin{cases} 0.135e^{(V_m + 80)/-6.8} & V_m < -40\text{mV} \\ 0 & V_m \geq -40\text{mV} \end{cases} \quad (\text{A.1.3b})$$

$$\beta_h = \begin{cases} 3.56e^{0.079V_m} + 3.1 \cdot 10^5 e^{0.35V_m} & V_m < -40\text{mV} \\ \frac{1}{0.13(1 + e^{(V_m + 10.66)/-11.1})} & V_m \geq -40\text{mV} \end{cases} \quad (\text{A.1.3c})$$

Slow inactivation gate, j

$$\frac{dj}{dt} = \alpha_j(1 - j) - \beta_j j \quad (\text{A.1.4a})$$

$$\alpha_j = \begin{cases} \frac{-1.2714 \cdot 10^5 e^{0.2444V_m} - 3.474 \cdot 10^{-5} e^{-0.04391V_m} \cdot (V_m + 37.78)}{1 + e^{0.311(V_m + 79.23)}} & V_m < -40\text{mV} \\ 0 & V_m \geq -40\text{mV} \end{cases} \quad (\text{A.1.4b})$$

$$\beta_j = \begin{cases} \frac{0.1212e^{-0.01052V_m}}{1 + e^{-0.1378(V_m + 40.14)}} & V_m < -40\text{mV} \\ \frac{0.3e^{-2.535 \cdot 10^{-7}V_m}}{1 + e^{-0.1(V_m + 32)}} & V_m \geq -40\text{mV} \end{cases} \quad (\text{A.1.4c})$$

Slow inward current

$$I_{\text{si}} = \bar{G}_{\text{si}} \cdot d \cdot f \cdot (V_m - E_{\text{si}}) \quad (\text{A.1.5})$$

$$E_{\text{si}} = 7.7 - 13.0287 \cdot \ln([\text{Ca}]_i) \quad (\text{A.1.6})$$

Activation gate, d

$$\frac{dd}{dt} = \alpha_d(1 - d) - \beta_d d \quad (\text{A.1.7a})$$

$$\alpha_d = \frac{0.095e^{-0.01(V_m - 5)}}{1 + e^{-0.072(V_m - 5)}} \quad (\text{A.1.7b})$$

$$\beta_d = \frac{0.07e^{-0.017(V_m + 44)}}{1 + e^{0.05(V_m + 44)}} \quad (\text{A.1.7c})$$

Inactivation gate, f

$$\frac{df}{dt} = \alpha_f(1 - f) - \beta_f f \quad (\text{A.1.8a})$$

$$\alpha_f = \frac{0.012e^{-0.008(V_m + 28)}}{1 + e^{0.15(V_m + 28)}} \quad (\text{A.1.8b})$$

$$\beta_f = \frac{0.0065e^{-0.02(V_m + 30)}}{1 + e^{-0.2(V_m + 30)}} \quad (\text{A.1.8c})$$

Calcium uptake

$$\frac{d([\text{Ca}]_i)}{dt} = -10^{-4}I_{\text{si}} + 0.07(10^{-4} - [\text{Ca}]_i) \quad (\text{A.1.9})$$

Outward Currents

Time-dependent potassium current

$$I_K = \bar{G}_K \cdot X \cdot X_i \cdot (V_m - E_K) \quad (\text{A.1.10})$$

$$\bar{G}_K = 0.282 \cdot \sqrt{[\text{K}]_o / 5.4} \quad (\text{A.1.11})$$

Activation gate, X

$$\frac{dX}{dt} = \alpha_X(1 - X) - \beta_X X \quad (\text{A.1.12a})$$

$$\alpha_X = \frac{0.0005e^{0.083(V_m + 50)}}{1 + e^{0.057(V_m + 50)}} \quad (\text{A.1.12b})$$

$$\beta_X = \frac{0.0013e^{-0.06(V_m + 20)}}{1 + e^{-0.04(V_m + 20)}} \quad (\text{A.1.12c})$$

Inactivation gate, X_i

$$X_i = \begin{cases} \frac{2.837(e^{0.04(V_m+77)} - 1)}{(V_m + 77)e^{0.04(V_m+35)}} & V_m > -100\text{mV} \\ 1 & V_m \leq -100\text{mV} \end{cases} \quad (\text{A.1.13})$$

Time-independent potassium current

$$I_{K1} = \overline{G}_{K1} \cdot K1_\infty \cdot (V_m - E_{K1}) \quad (\text{A.1.14})$$

$$\overline{G}_{K1} = 0.6047 \cdot \sqrt{[K]_o}/5.4 \quad (\text{A.1.15})$$

Inactivation gate, $K1$

$$K1_\infty = \frac{\alpha_{K1}}{\alpha_{K1} + \beta_{K1}} \quad (\text{A.1.16a})$$

$$\alpha_{K1} = \frac{1.02}{1 + e^{0.2385(V_m - E_{K1} - 59.215)}} \quad (\text{A.1.16b})$$

$$\beta_{K1} = \frac{0.49124e^{0.08032(V_m - E_{K1} + 5.476)} + e^{0.06175(V_m - E_{K1} - 594.31)}}{1 + e^{-0.5143(V_m - E_{K1} + 4.753)}} \quad (\text{A.1.16c})$$

Plateau potassium current

$$I_{Kp} = \overline{G}_{Kp} \cdot Kp \cdot (V_m - E_{Kp}) \quad (\text{A.1.17})$$

$$E_{Kp} = E_{K1} \quad (\text{A.1.18})$$

$$Kp = \frac{1}{1 + e^{(7.488 - V_m)/5.98}} \quad (\text{A.1.19})$$

Background potassium current

$$I_b = \overline{G}_b \cdot (V_m - E_b) \quad (\text{A.1.20})$$

Total ionic current

$$\begin{aligned} I_{\text{ion}} &= I_{\text{Na}} + I_{\text{si}} + I_K + I_{K1} + I_{Kp} + I_b \\ &= \overline{G}_{\text{Na}} \cdot m^3 \cdot h \cdot j \cdot (V_m - E_{\text{Na}}) + \overline{G}_{\text{si}} \cdot d \cdot f \cdot (V_m - E_{\text{si}}) \\ &\quad + \overline{G}_K \cdot X \cdot X_i \cdot (V_m - E_K) + \overline{G}_{K1} \cdot K1_{\text{inf}} \cdot (V_m - E_{K1}) \\ &\quad + \overline{G}_{Kp} \cdot Kp \cdot (V_m - E_{Kp}) + \overline{G}_b \cdot (V_m - E_b) \end{aligned} \quad (\text{A.1.21})$$

For an individual cardiac cell we have that the transmembrane potential is given by [17]:

$$\frac{dV_m}{dt} = -\frac{1}{C_m}(I_{ion} + I_{st}), \quad (\text{A.1.22})$$

where C_m is the membrane capacitance and I_{st} is the stimulus current applied by the sinoatrial node.

The following table shows the values of the channel conductances, the reversal potentials for the ions, and other parameters.

Table A.1: Parameters for the Luo-Rudy Phase I model; the conductances are in mS/cm^2 and the reversal potentials in mV [5].

Channel Conductance	Reversal Potential	Other Parameters
$\bar{G}_{\text{Na}} = 23.0$	$E_{\text{Na}} = 54.4$	Resting Membrane Potential $V_{rest} = -84.0\text{mV}$
$\bar{G}_{\text{si}} = 0.09$	$E_{\text{si}} = 118.7$	Membrane Threshold Potential $V_{threshold} = -60\text{mV}$
$\bar{G}_{\text{K}} = 0.282$	$E_{\text{K}} = -77$	$[\text{K}]_o = 5.4\text{mM}$
$\bar{G}_{\text{K1}} = 0.6047$	$E_{\text{K1}} = -87.2$	Membrane Capacitance $C_m = 1 \mu\text{F}/\text{cm}^2$
$\bar{G}_{\text{Kp}} = 0.0183$	$E_{\text{Kp}} = -87.2$	
$\bar{G}_{\text{b}} = 0.03921$	$E_{\text{b}} = -59.87$	

A.2 The Model of Courtemanche et al.

The transmembrane potential, V , is given by

$$\frac{dV}{dt} = -\frac{1}{C_m}(I_{ion} + I_{st}),$$

where I_{ion} is defined as

$$\begin{aligned} I_{ion} &= I_{\text{Na}} + I_{\text{K1}} + I_{\text{to}} + I_{\text{Kur}} + I_{\text{Kr}} + I_{\text{Ks}} \\ &+ I_{\text{Ca,L}} + I_{\text{p,Ca}} + I_{\text{NaK}} + I_{\text{NaCa}} + I_{\text{b,Na}} + I_{\text{b,Ca}}, \end{aligned}$$

and I_{st} is the stimulus current. There are 15 gating equations in the form

$$\frac{dy}{dt} = \frac{y_{\infty} - y}{\tau_y}, \quad (\text{A.2.1})$$

where y is the gating variable and y_{inf} and τ_y are defined as

$$y_{\infty} = \frac{\alpha_y}{\alpha_y + \beta_y},$$

$$\tau_y = \frac{1}{\alpha_y + \beta_y},$$

with both α_y and β_y being functions of V . The remaining ODEs relate to ionic concentrations and are defined as

$$\begin{aligned} \frac{d[\text{Na}^+]_i}{dt} &= \frac{-3I_{\text{Na,K}} - 3I_{\text{NaCa}} - I_{\text{b,Na}} - I_{\text{Na}}}{FV_i}, \\ \frac{d[\text{K}^+]_i}{dt} &= \frac{2I_{\text{Na,K}} - I_{\text{K1}} - I_{\text{to}} - I_{\text{Kur}} - I_{\text{Kr}} - I_{\text{Ks}} - I_{\text{b,K}}}{FV_i}, \\ \frac{d[\text{Ca}^{2+}]_i}{dt} &= \frac{B1}{B2}, \\ B1 &= \frac{2I_{\text{NaCa}} - I_{\text{p,Ca}} - I_{\text{Ca,L}} - I_{\text{b,Ca}}}{2FV_i} \\ &+ \frac{V_{\text{up}}(I_{\text{up,leak}} - I_{\text{up}}) + I_{\text{rel}}V_{\text{rel}}}{V_i}, \\ B2 &= 1 + \frac{[\text{Trpn}]_{\text{max}}K_{\text{m,Trpn}}}{([\text{Ca}_i^{2+}] + K_{\text{m,Trpn}})^2} + \frac{[\text{Cmdn}]_{\text{max}}K_{\text{m,Cmdn}}}{([\text{Ca}^{2+}]_i + K_{\text{m,Cmdn}})^2}, \\ \frac{d[\text{Ca}^{2+}]_{\text{up}}}{dt} &= I_{\text{up}} - I_{\text{up,leak}} - I_{\text{tr}} \frac{V_{\text{rel}}}{V_{\text{up}}}, \\ \frac{d[\text{Ca}^{2+}]_{\text{rel}}}{dt} &= (I_{\text{tr}} - I_{\text{rel}}) \left\{ 1 + \frac{[\text{Csqn}]_{\text{max}}K_{\text{m,Csqn}}}{([\text{Ca}^{2+}]_{\text{rel}} + K_{\text{m,Csqn}})^2} \right\}^{-1}. \end{aligned}$$

For further details, see [7].

A.3 The Model of Winslow et al.

The transmembrane potential, V , is defined as

$$\begin{aligned} \frac{dV}{dt} &= -(I_{\text{Na}} + I_{\text{Ca}} + I_{\text{Ca,K}} + I_{\text{Kr}} + I_{\text{Ks}} + I_{\text{to1}} + I_{\text{K1}} + I_{\text{Kp}} \\ &+ I_{\text{NaCa}} + I_{\text{NaK}} + I_{\text{p(Ca)}} + I_{\text{Ca,b}} + I_{\text{Na,b}}). \end{aligned}$$

There are eight gating equations to describe sodium and potassium:

$$\begin{aligned}
\frac{dm}{dt} &= \alpha_m(1 - m) - \beta_m m, \\
\frac{dh}{dt} &= \alpha_h(1 - h) - \beta_h h, \\
\frac{dj}{dt} &= \alpha_j(1 - j) - \beta_j j, \\
\frac{dX_{Kr}}{dt} &= K_{12}(1 - X_{Kr}) - K_{21}X_{Kr}, \\
\frac{dX_{Ks}}{dt} &= \frac{(X_{Ks}^\infty - X_{Ks})}{\tau_{X_{Ks}}}, \\
\frac{dX_{to1}}{dt} &= \alpha_{X_{to1}}(1 - X_{to1}) - \beta_{X_{to1}}X_{to1}, \\
\frac{dY_{to1}}{dt} &= \alpha_{Y_{to1}}(1 - Y_{to1}) - \beta_{Y_{to1}}Y_{to1}, \\
\frac{dy}{dt} &= \frac{y_\infty - y}{\tau_y}.
\end{aligned}$$

There are a number of equations related to calcium concentration:

$$\begin{aligned}
\frac{dP_{C_1}}{dt} &= -k_a^+[Ca^{2+}]_{ss}^n P_{C_1} + k_a^- P_{O_1}, \\
\frac{dP_{O_1}}{dt} &= k_a^+[Ca^{2+}]_{ss}^n P_{C_1} - k_a^- P_{O_1} - k_b^+[Ca^{2+}]_{ss}^m P_{O_1} \\
&\quad + k_b^- P_{O_2} - k_c^+ P_{O_1} + k_c^- P_{C_2}, \\
\frac{dP_{O_2}}{dt} &= k_b^+[Ca^{2+}]_{ss}^m P_{O_1} - k_b^- P_{O_2}, \\
\frac{dP_{C_2}}{dt} &= k_c^+ P_{O_1} - k_c^- P_{C_2}.
\end{aligned}$$

The following system describes the membrane current of calcium through the so-called L-type

channels.

$$\begin{aligned}
\frac{dC_0}{dt} &= \beta C_1 + \omega C_{Ca0} - (4\alpha + \gamma)C_0, \\
\frac{dC_1}{dt} &= 4\alpha C_0 + 2\beta C_2 + \frac{\omega}{b} C_{Ca1} - (\beta + 3\alpha + \gamma a)C_1, \\
\frac{dC_2}{dt} &= 3\alpha C_1 + 3\beta C_3 + \frac{\omega}{b^2} C_{Ca2} - (2\beta + 2\alpha + \gamma a^2)C_2, \\
\frac{dC_3}{dt} &= 2\alpha C_2 + 4\beta C_4 + \frac{\omega}{b^3} C_{Ca3} - (3\beta + \alpha + \gamma a^3)C_3, \\
\frac{dC_4}{dt} &= \alpha C_3 + gO + \frac{\omega}{b^4} C_{Ca4} - (4\beta + f + \gamma a^4)C_4, \\
\frac{dO}{dt} &= fC_4 - gO, \\
\frac{dC_{Ca0}}{dt} &= \beta' C_{Ca1} + \gamma C_0 - (4\alpha' + \omega)C_{Ca0}, \\
\frac{dC_{Ca1}}{dt} &= 4\alpha' C_{Ca0} + 2\beta' C_{Ca2} + \gamma a C_1 - (\beta' + 3\alpha' + \frac{\omega}{b})C_{Ca1}, \\
\frac{dC_{Ca2}}{dt} &= 3\alpha' C_{Ca1} + 3\beta' C_{Ca3} + \gamma a^2 C_2 - (2\beta' + 2\alpha' + \frac{\omega}{b^2})C_{Ca2}, \\
\frac{dC_{Ca3}}{dt} &= 2\alpha' C_{Ca2} + 4\beta' C_{Ca4} + \gamma a^3 C_3 - (3\beta' + \alpha' + \frac{\omega}{b^3})C_{Ca3}, \\
\frac{dC_{Ca4}}{dt} &= \alpha' C_{Ca3} + \gamma a^4 C_4 - (4\beta' + f' + \frac{\omega}{b^4})C_{Ca4},
\end{aligned}$$

Intracellular calcium buffering is described by

$$\begin{aligned}
\frac{d[\text{HTRPNCa}]}{dt} &= k_{\text{htrpn}}^+ [\text{Ca}^{2+}]_i ([\text{HTRPN}]_{\text{tot}} - [\text{HTRPNCa}]) \\
&\quad - k_{\text{htrpn}}^- [\text{HTRPNCa}], \\
\frac{d[\text{LTRPNCa}]}{dt} &= k_{\text{ltrpn}}^+ [\text{Ca}^{2+}]_i ([\text{LTRPN}]_{\text{tot}} - [\text{LTRPNCa}]) \\
&\quad - k_{\text{ltrpn}}^- [\text{LTRPNCa}],
\end{aligned}$$

where the k -coefficients are constants.

Intracellular ionic concentrations are described by:

$$\begin{aligned}
\frac{d[\text{Na}^+]_i}{dt} &= -(I_{Na} + I_{Na,b} + 3I_{NaCa} + 3I_{NaK}) \frac{A_{cap} C_{sc}}{V_{myo} F}, \\
\frac{d[\text{K}^+]_i}{dt} &= -(I_{Kr} + I_{Ks} + I_{to1} + I_{K1}, \\
&\quad + I_{Kp} + I_{Ca,K} - 2I_{NaK}) \frac{A_{cap} C_{sc}}{V_{myo} F}, \\
\frac{d[\text{Ca}^{2+}]_i}{dt} &= \beta_i \left[J_{xfer} - J_{up} - J_{trpn} \right. \\
&\quad \left. - (I_{Ca,b} - 2I_{NaCa} + I_{p(Ca)}) \frac{A_{cap} C_{sc}}{2V_{myo} F} \right], \\
\frac{d[\text{Ca}^{2+}]_{ss}}{dt} &= \beta_{ss} \left(J_{rel} \frac{V_{JSR}}{V_{myo}} - J_{xfer} \frac{V_{myo}}{V_{ss}} - I_{Ca} \frac{A_{cap} C_{sc}}{2V_{myo} F} \right), \\
\frac{d[\text{Ca}^{2+}]_{JSR}}{dt} &= \beta_{JSR} (J_{tr} - J_{rel}), \\
\frac{d[\text{Ca}^{2+}]_{NSR}}{dt} &= J_{up} \frac{V_{myo}}{V_{NSR}} - J_{tr} \frac{V_{JSR}}{V_{NSR}}.
\end{aligned}$$

There are 33 ODEs in total. See [25] for details.

A.4 The Model of Puglisi-Bers

The transmembrane potential, V , is given by

$$\begin{aligned}
\frac{dV}{dt} &= \frac{I_{stim} - (I_{Na} + I_{CaL} + I_{CaT} + I_{Kr} + I_{Ks} + I_{NaCa} + I_{K1} + I_{Kp})}{C} \\
&+ \frac{I_{stim} - (I_{pCa} + I_{Na_b} + I_{Ca_b} + I_{NaK} + I_{to} + I_{Cl(Ca)})}{C}
\end{aligned}$$

There are nine gating equations:

$$\begin{aligned}
\frac{dm}{dt} &= \alpha_m(1 - m) - \beta_m m, \\
\frac{dh}{dt} &= \alpha_h(1 - h) - \beta_h h, \\
\frac{dj}{dt} &= \alpha_j(1 - j) - \beta_j j, \\
\frac{dd}{dt} &= \alpha_d(1 - d) - \beta_d d, \\
\frac{df}{dt} &= \alpha_f(1 - f) - \beta_f f, \\
\frac{db}{dt} &= \frac{b_\infty - b}{\tau_b}, \\
\frac{dg}{dt} &= \frac{g_\infty - g}{\tau_g}, \\
\frac{dX_r}{dt} &= \frac{X_{r_\infty} - X_r}{\tau_{X_r}}, \\
\frac{dX_s}{dt} &= \frac{X_{s_\infty} - X_s}{\tau_{X_s}}
\end{aligned}$$

There are seven equations to describe ionic concentrations:

$$\begin{aligned}
\frac{dNa_i}{dt} &= -(I_{Na} + I_{CaNa} + I_{Na_b} + 3I_{NaCa} + 3I_{NaK}) \frac{A_{cap}}{V_{myo}F}, \\
\frac{dCa_i}{dt} &= ((I_{CaCa} + I_{pCa} + I_{Ca_b} + I_{CaT}) - I_{NaCa}) \frac{A_{cap}}{2V_{myo}F} + I_{rel} \frac{V_{JSR}}{V_{myo}} + (I_{leak} - I_{up}) \frac{V_{NSR}}{V_{myo}}, \\
\frac{dK_i}{dt} &= -(I_{CaK} + I_{Kr} + I_{Ks} + I_{K1} + I_{Kp} + I_{to} - 2I_{NaK}) \frac{A_{cap}}{V_{myo}F}, \\
\frac{dK_o}{dt} &= (I_{CaK} + I_{Kr} + I_{Ks} + I_{K1} + I_{Kp} + I_{to} - 2I_{NaK}) \frac{A_{cap}}{V_{cleft}F}, \\
\frac{dCa_{JSR}}{dt} &= -(I_{rel} - I_{tr} \frac{V_{NSR}}{V_{JSR}}), \\
\frac{dCa_{NSR}}{dt} &= -(((I_{leak} + I_{tr}) - I_{up})), \\
\frac{dCa_{foot}}{dt} &= (I_{CaCa}) \frac{A_{cap}}{2V_{myo}F} R_{AV}
\end{aligned}$$

There are 17 ODEs in total. See [20] and the references within for more details. Also, see [1] for a concise description of the model.