



"El saber de mis hijos  
hará mi grandeza"

---

---

# UNIVERSIDAD DE SONORA

## DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

### Programa de Posgrado en Matemáticas

Numerical Solutions of the scroll wave type on  
Reaction-Diffusion systems: Applications to Cardiac  
Dynamics

## T E S I S

Que para obtener el grado académico de:

**Doctor en Ciencias**  
(Matemáticas)

Presenta:

M.C. Jesús Jairo Rodríguez Padilla

Directores de Tesis:  
Dr. Flavio H. Fenton  
Dr. Daniel Olmos Liceaga

Hermosillo, Sonora, México,

19 de diciembre de 2017



## SINODALES

Dr. Víctor Francisco Breña Medina  
Universidad Nacional Autónoma de México

Dr. Saúl Diaz Infante Velasco  
CONACyT-Universidad de Sonora

Dr. Flavio H. Fenton  
Georgia Institute of Technology

Dr. Roberto Núñez González  
Universidad de Sonora

Dr. Daniel Olmos Liceaga  
Universidad de Sonora



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The Hodgkin and Huxley Model . . . . .	4
1.2	Brief History of the Cable Equation . . . . .	6
1.3	Cardiac Dynamics and Reaction-Diffusion Systems . . . . .	9
1.4	Solutions of the Spiral and Scroll Wave Type in Cardiac Dynamics .	13
<b>2</b>	<b>Preliminaries</b>	<b>15</b>
2.1	Euler Finite Difference . . . . .	15
2.1.1	More Realistic Geometries . . . . .	17
2.2	Historical Background on Pseudospectral Methods . . . . .	18
2.2.1	Spatial Derivative by Chebyshev Pseudospectral Multidomain Method . . . . .	20
2.2.2	The Gibbs Phenomenon . . . . .	21
<b>3</b>	<b>Interaction of Solutions of the Scroll Ring Type with Boundaries for the Weak Meandering Regime</b>	<b>23</b>
3.1	Model and Parameters . . . . .	23
3.1.1	Numerical Considerations . . . . .	24
3.2	Numerical Studies . . . . .	25
3.2.1	Scroll ring dynamics away from the boundaries . . . . .	29
3.2.2	Scroll ring dynamics close to the boundaries . . . . .	31
3.3	Discussion and Conclusions . . . . .	35
<b>4</b>	<b>Numerical Solutions of Equations of Cardiac Wave Propagation based on Chebyshev Multidomain Pseudospectral Methods</b>	<b>37</b>
4.1	Model equations. . . . .	37
4.2	Numerical Methods . . . . .	38
4.2.1	Implementation of the CMD method. . . . .	39
4.2.2	Implementation of the SICMD method. . . . .	39
4.2.3	Implementation of the implicit method. . . . .	40
4.2.4	Implementation of the OS method. . . . .	41
4.3	One dimensional numerical convergence analysis. . . . .	41
4.3.1	The Karma model. . . . .	41

4.3.2	The MV model . . . . .	43
4.4	Numerical precision as a function of the size of the interval and the stiffness of the equation . . . . .	44
4.5	A comparison of the stability region for the CMD, SICMD and ICMD methods . . . . .	47
4.6	Two dimensional studies . . . . .	48
4.6.1	The Karma Model . . . . .	50
4.6.2	Some physiological quantities of interest . . . . .	52
4.6.3	The MV model . . . . .	55
4.7	Three dimensional studies . . . . .	57
4.7.1	Rotational anisotropy . . . . .	59
4.8	Discussion . . . . .	60
<b>5</b>	<b>Chebyshev Multidomain Pseudospectral Method to Solve Cardiac Wave Equations with Rotational Anisotropy</b>	<b>71</b>
5.1	Numerical Methods . . . . .	71
5.1.1	Implementation of the CMD method . . . . .	71
5.1.2	Implementation of rotational anisotropy . . . . .	72
5.2	Numerical studies and results . . . . .	74
5.2.1	Stable scroll wave numerical solutions . . . . .	75
5.2.2	Unstable scroll wave numerical solutions . . . . .	79
5.3	About the convergence of the method . . . . .	81
5.4	Discussion . . . . .	87
<b>6</b>	<b>Computational Studies</b>	<b>89</b>
6.1	OpenAcc: Comparing C and Fortran . . . . .	89
6.2	Three dimensional studies . . . . .	92

# Chapter 1

## Introduction

Mathematical models have played a very important role throughout the history of science. With the theory of differential equations developed by Newton, an infinite amount of possibilities arose to describe phenomena that appears in nature.

In molecular biology, the use of mathematical models had its great breakthrough with the work done in 1952 by professors Alan Lloyd Hodgkin and Andrew Huxley where they developed a mathematical model to describe and explain the ionic mechanisms that underlie in the initiation and the propagation of action potentials in nerve cells. In 1963 they were given the Nobel prize in physiology-medicine due to this remarkable achievement.

The work done by Hodgkin and Huxley not only has been used to study the nervous system. Together, Arturo Rosenblueth and Norbert Wiener, on their research paper “The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle”, in 1946, was the starting point of theoretical research in this field. Their paper seemingly deals with cardiac arrhythmia and its mathematical formulation.

The model of Wiener and Rosenblueth describes the propagation of an excitable wave. It considers the motion of curves with free ends representing the wave front. The attractive feature of this kinematic model is that it perfectly mimics biophysical reaction-diffusion equations of waves in excitable media in the parameter window of weak excitability ([Brazhnik et al. \(1988\)](#); [Mikhailov et al. \(1994\)](#)). The kinematic theory of wave propagation attempts to follow the spatial and temporal aspects based only on the fundamental underlying biophysical processes. It can predict differences between the spatio-temporal aura pattern caused by a neural phenomena and those caused by a vascular phenomena.

From here on, there has been a huge quantity of research devoted to understand the origins and onsets of cardiac arrhythmias.

The use of the models that arose with this ongoing research has allowed to understand properties such as: propagation velocity, action potential duration (APD), studies on the origin and break up of spirals, responsible for the appearance of dynamic anomalies in the propagation of action potentials.

There are a wide range of models in the study of action potential propagation. These depend on the type of cell that is being modelled. Some examples include the 4 variable Noble model (Purkinje cells), the Beeler-Reuter model, which is a generic ventricular model consisting of 8 variables, the ten Tusscher-Noble-Noble-Panfilov

model, which is a human ventricular model with 24 variables, the Courtemanche-Ramirez-Nattel model (1998), consisting of 21 variables, which models human atrial cells, to name a few.

All these models have been developed, in their majority, thanks to experimental work. Depending the different kind of questions that we want to address, we incline towards a certain model.

A disadvantage when using these models is that there is not an analytical solution, and, therefore, they are required to be solved using a computer.

The principal problem that arises when solving the equations in these models is that there are a wide variety of dynamics in them that are presented in very different time scales. As a consequence of these, very small spatial discretizations are required along with very small time steps in order to obtain numerical stable solutions.

The main purpose of this thesis is to develop and test numerical methods to solve partial differential equations of the reaction diffusion type. More precisely, PDE's that appear in cardiac dynamics.

The numerical solution of equations that model the propagation of action potentials in cardiac tissue is a very hard problem. Multiple temporal scales arising from the local dynamics joined to stiffness of the diffusion operator, results in a phenomenon that evolves in a multiple spatio-temporal scale. The solution of such equations in two and three dimensions can be very time and memory consuming. Such equations are part of a larger family known as the reaction-diffusion equations and one of the main characteristic of this type of equations is stiffness in both the diffusion and the reactive parts. Additionally to the stiffness of the diffusion part of the equation, the active currents that lead to change in the membrane voltage are highly nonlinear.

Scroll waves and scroll rings, special type of solutions to reaction-diffusion systems of equations, have been observed in excitable media, in particular in cardiac tissue, where they are associated with cardiac arrhythmias.

The present thesis is divided in the following main parts:

- (i) brief history of cardiac dynamics and preliminary theory (present chapter);
- (ii) the study of scroll rings, described by the Karma model, in the parameter region of weak meandering and positive filament tension. In here, the dynamics of the scroll rings for the long time and their interaction with the boundaries are numerically investigated for distinct initial conditions (Chapter 3);
- (iii) the study and development of alternative numerical methods to those that appear in the literature for solving equations of the reaction-diffusion type. In here, a pseudospectral approach based on Chebyshev polynomials is considered (Chapter 4). More over, in Chapter 5 we employ a numerical method developed in Chapter 4, namely the CMD method, to (numerically) solve a reaction diffusion system on an anisotropic heterogeneous media. To this end we also include a methodology known as the phase field method, to treat irregular geometries on a wide variety of domains.

(iv) as an appendix, in Chapter 6, we include a section of computational studies.

More precisely, we discuss and present a comparison when performing numerical simulations utilizing two of the most common known programming languages, C and Fortran, when parallel programming techniques are implemented using graphic processing units. In here we discuss the different hardware equipment used along with some advantages and disadvantages of them. We have tested our different implementations using three different cardiac models: a simple two variable model, known as the Karma model; a more complicated 8-variable model, known as the BR (Beeler-Reuter) model and a highly complex human atria model known as the ten Tusscher model.

In the vast majority of cases when studying excitable media Biktashev et al. (1994); Panfilov and Winfree (1985) and cardiac dynamics Clayton et al. (2011), the numerical methods used to solve the model equations are based on finite differences schemes. This due to the simplicity of the implementation and because finite difference schemes can be parallelized without too much effort Sato et al. (2009).

Other approaches used to solve reaction-diffusion systems in heterogeneous media include Finite Element Franzone et al. (1998) and Finite Volume Harrild et al. (2000), Wavelets Aziz et al. (2013); Hariharan and Kannan (2014), and the novel radial basis functions approach Reutskiy and Lin (2017), to name some of the most important.

Modelling cardiac wave activity can be done by means of the monodomain and bidomain approach. In the class that use the monodomain approach to solve these systems of equations there has been a variety of methods, ranging from the usual finite difference methods (Peñaranda et al. (2012), Ying et al. (2008)) to more complex ones like adaptive methods Heidenreich et al. (2008), whereas for the bidomain approach we can also encounter the usual finite difference methods and also the ones known as finite element and finite volume methods (Trayanova et al. (2011); Linge et al. (2009); Vigmond et al. (2008)).

Reaction diffusion equations that model cardiac dynamics do not solve or model in a very exact manner what really happens in nature. However, said models do replicate relevant physiological properties. When solving, numerically, these kind of systems, the aim is to do so as best as possible assuring that the phenomena encountered are consequence of the model and not of the numerical solution. In this sense, we do require reliable solutions obtained with the numerical methods in order to build knowledge that allow us, for instance, to construct techniques to control and monitor cardiac arrhythmias, develop studies on the impact of certain drugs, to name some of the challenges remaining still to date.

In this work, we present also explicit, implicit and semi-implicit schemes based on Chebyshev Multidomain Pseudospectral approach; some advantages and disadvantages for each of them and estimates on the computing time to obtain approximated solutions. Also, in this work a scaling law is discussed to find numerical equivalence between all the cardiac models based on the Hodgkin-Huxley formalism with a discussed example.

The main motivation to use these schemes based on Chebyshev polynomials is

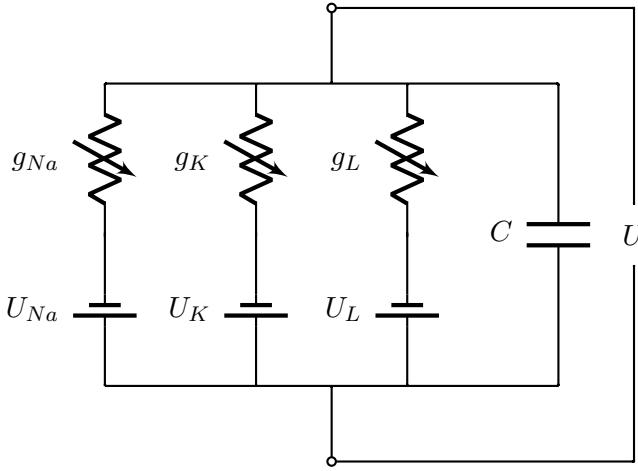


Figure 1.1: Electrical circuit to model the membrane of an excitable cell;  $g_{Na}$ ,  $g_K$  and  $g_L$  are the conductances for sodium, potassium and leakage current, respectively, and,  $U_{Na}$ ,  $U_K$  and  $U_L$  represent the Nernst potential for each respective ion.  $U$  and  $C$  are the potential and the capacitance of the membrane, respectively.

that we want to achieve better precision (if possible), with less discretization points and smaller time steps than the methods usually utilized. It is of great importance also to solve these models on more realistic geometries and not only rectangular slabs of tissue. Again, the main problem when trying this is, usually, that the more realistic the geometry in question, the more points for the discretization is needed and as a consequence, smaller time steps are needed for the numerical solution to converge. Then, the aim for utilizing Chebyshev pseudospectral methods is to be able to relax the number of discretization points and, therefore, reduce the time steps.

Also, even though we have tested these pseudospectral methods on cardiac models, they can be used on systems of the reaction diffusion type.

## 1.1 The Hodgkin and Huxley Model

An essential part of the work done by Hodgkin and Huxley was to establish that the sodium and potassium channels can be open or closed, and that this behavior depends on the action potential of the membrane on a given time.

The main purpose of the Hodgkin and Huxley model (HH) is to describe the changes of the potential  $U$  throughout the membrane. These changes are a direct consequence of ion exchange through the ionic channels of the membrane, and the conductance of said channels depends on the voltage and time. For a better understanding of the changes in  $U$  during an action potential it is useful to consider an equivalent electrical circuit. Due to the fact that the proportion of the area of the membrane that is covered by ionic channels is 100 times less than the part of the membrane that acts as an insulator, the membrane can be considered as a capac-

itor with a leaking current. Because of this property, it is possible to formulate a parallel RC circuit. The insulating membrane can be considered as a capacitor and each ionic channel as a resistor variable. In figure (1.1) we show the corresponding electrical circuit.

In the circuit, we have that for every ion a current,  $I_S$ , is obtained that satisfies the relation

$$I_S = g_S(U - U_S) \quad (1.1)$$

where  $S$  refers to the  $K$ ,  $Na$  or the leaking current given by the less important ions; in equation (1.1),  $g_S$  represents the conductance of the  $S$  ion,  $U$  is the membrane potential and  $U_S$  is the Nernst potential or equilibrium potential of the  $S$  ion.

The Nernst potential for any given ionic species is the membrane potential at which the ionic species is in equilibrium; i.e., there is no net movement of the ion across the membrane.

Utilizing Kirchoff's laws of conservation for the circuit (1.1), we obtain the equation

$$\begin{aligned} C_m \frac{dU}{dt} &= -I_{ion} = -(I_{Na} + I_K + I_L) \\ &= -[g_{Na}(U - U_{Na}) + g_K(U - U_K) + g_L(U - U_L)] \end{aligned} \quad (1.2)$$

where  $U$  is the membrane potential;  $I_{ion}$  is the current carried by the ions through the membrane;  $I_{Na}$  and  $I_K$  represent the sodium and potassium currents, respectively;  $I_L$  represents a leaking current given by less important ions for the electrical conductivity, such as  $Cl$  in the case of cardiac tissue and  $C_m$  is the membrane capacitance.

For equation (1.2), Hodgkin and Huxley found that the conductances  $g_{Na}$  and  $g_K$  depend on the voltage and time, developing a model in which the conductances  $g_{Na}$  and  $g_K$  are given by the following

$$g_{Na} = \bar{g}_{Na}n^4, \quad g_K = \bar{g}_Km^3h \quad (1.3)$$

where  $\bar{g}_{Na}$  and  $\bar{g}_K$  represent the maximum conductance and the variables  $m, n, h$  are functions that depend on the voltage and time. The exponents of each of the variables were determined experimentally, and said variables take real values between 0 and 1. Besides, each of these variables satisfies

$$\frac{dz}{dt} = \alpha_z(1 - z) - \beta_zz \quad (1.4)$$

where  $\alpha_z$  and  $\beta_z$  depend on the voltage and the variable  $z$  correspond to the variables  $m, n$  and  $h$ . The variables  $m, n, h$  represent probabilities that the case  $z$  is

in an open state and are known as gating variables. In equation (1.3), the terms  $n^4$  and  $m^3h$  can be interpreted as the fraction of open channels for a given time.

We refer to [Soto \(2011\)](#) for a more detailed explanation on the derivation of the model and the ionic currents.

## 1.2 Brief History of the Cable Equation

A simple core conductor can be described as a long thin tube of membrane that is filled with a core of electrically conducting medium and is bathed on the outside by another electrically conducting medium. This membrane tube is typically a cylinder whose length is very much greater than its diameter. For nerve axons or dendrites, the resistance to electric current flow across the membrane is much greater than the core resistance for short length(that is, small, compared with the length constant of the nerve axon or dendrite) increments along the cylinder. Because of this relative distances, it follows that electric current inside the core conductor tends to flow parallel to the cylinder axis for considerable distance before a significant fraction can leak out across the membrane. It is this simple physical concept that provides the basis for a cable theory treatment of steady-state distributions of current and potential in neuronal core conductors.

Both the concepts and the mathematical theory of core conductors have played a key role electrophysiology. They have provided a basis for the understanding of electrophysiological observations in terms of the underlying anatomic structures. The early mathematical theory arose from the need to interpret early experiments made on whole nerve trunks.

Cable theory predictions were elaborated mathematically, computed numerically, and displayed graphically, thus providing the basis for improved experimental designs. This led to remarkable success in the characterization of axonal membrane properties and cable properties. It is relevant to note that the most sophisticated studies of nonlinear membrane properties were made under experimental conditions(space clamp and voltage clamp) designed to eliminate cable properties. Although this was highly successful with excised giant axons, such space clamping was not applicable to cells with dendritic trees.

It is remarkable that not only electrophysiology, but also electrochemistry and much of electrophysics, can be traced back to a common origin in the observations and arguments of Galvani(Professor of Anatomy in Bologna) and Volta(Professor of Physics at Pavia). Systematic physical measurements of electric currents generated by nerve and muscle were begun in the 1840's by Mateucci(Professor of Physics at Pisa) and by du Bois-Reymund(in the Berlin physiological laboratory of Johannes Müller).

With the development of core conductor theory and the improvement of experimental techniques over the years, it became recognized that linear aspects of the observed phenomena(those explained by means of a core conductor having a passive membrane) should be distinguished from the nonlinear aspects. A classic example was provided by the asymmetry found beneath the cathode and anode when ex-

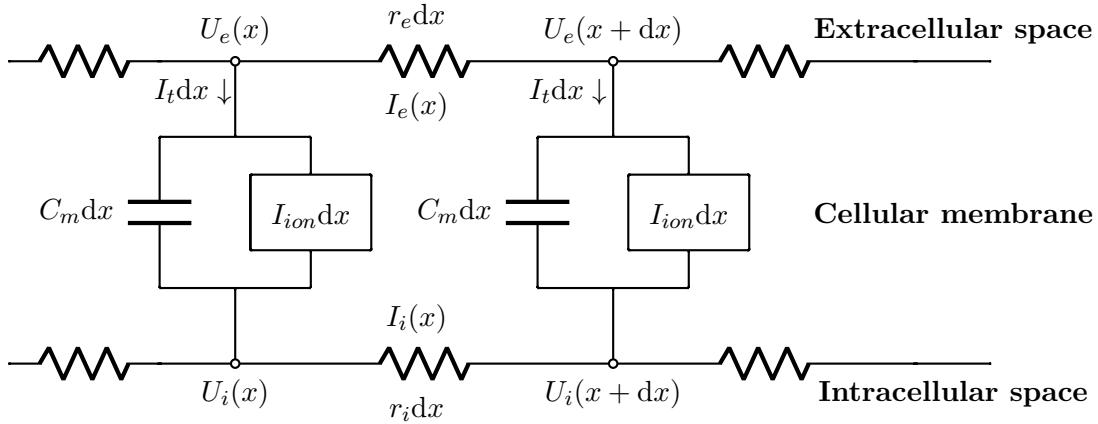


Figure 1.2: Circuit that models the cable equation

ternally applied current was just below the threshold for initiation of a propagated impulse. Such nonlinearity was referred to as the local response by Hodgkin, Katz and Rushton and was sometimes characterized as an active membrane property.

Cable theory dates back to 1855 when professor William Thompson (later known as lord Kelvin) presented to the Royal Society a series of his excerpts from his correspondence with professor Stokes. This provided a mathematical theory and practical applications for the submarine (transatlantic) telegraph cable then being planned. Thompson not only derived the cable equation, but also, being thoroughly familiar with the mathematical advances of his time, he presented both steady-state and transient solutions for particular cable boundary and initial conditions.

An important merit of cable theory is the simplifying assumption that reduces the problem to a single spatial dimension, namely, distance along the cable; this greatly facilitates the theoretical treatment of the problem. Since 1945 the two most important presentations of cable theory (with applications to electrophysiology) have been provided by the now classic papers of Hodgkin and Rushton and Davis & Lorente de Nó.

In what follows we will explain and derive the cable equation. Suppose that in any place along the cable the membrane potential only depends on the axial length and not on any radial or angular variables. Thus, we consider that we have a one dimensional cable. Now we split the cable in small pieces of length  $dx$ . In any part of the cable, we have two kinds of currents, transmembrane current and axial current. The latter one consisting of intracellular and extracellular currents, each of which we will assume are linear functions that depend on the voltage,  $U$ . Let us write down  $I_i$  and  $I_e$  to denote the intracellular and extracellular currents, respectively. Then, applying Ohm's law, we have

$$\begin{aligned} U_i(x+dx) - U_i(x) &= -I_i(x)r_i dx, \\ U_e(x+dx) - U_e(x) &= -I_e(x)r_e dx, \end{aligned} \quad (1.5)$$

where the minus sign on the right hand side of (1.5) is due to the convention that

a flow of current from left to right is negative. This translates into increasing  $x$ . Then, by rearranging (1.5) and taking the limit as  $dx$  tends to zero, we have

$$\begin{aligned} I_i &= -\frac{1}{r_i} \frac{\partial U_i}{\partial x}, \\ I_e &= -\frac{1}{r_e} \frac{\partial U_e}{\partial x}. \end{aligned} \quad (1.6)$$

The numbers  $r_i$  and  $r_e$  are the resistances of the inside and outside of the membrane. In general, we have that

$$r_i = \frac{R_c}{A_i},$$

where  $R_c$  is the axoplasmic resistance, measured in Ohms( $\Omega$ ), and  $A_i$  is the cross section area of the cable measured in  $\text{cm}^2$ . Similarly, for  $r_e$  we have

$$r_e = \frac{R_c}{A_e}.$$

By conservation of current(Kirchhoff's law), the following is true

$$I_i(x) - I_i(x + dx) = I_t dx = I_e(x + dx) - I_e(x), \quad (1.7)$$

where  $I_t$  is the total transmembrane current(positive when the direction of the flow is to the outside) per length unit of the membrane. Taking the limit when  $dx$  tends to zero in equation (1.7), we arrive at

$$I_t = -\frac{\partial I_i}{\partial x} = \frac{\partial I_e}{\partial x}. \quad (1.8)$$

In a cable without any additional external currents, the total axial current is given by  $I_t = I_i + I_e$ . Thus, considering  $U = U_i - U_e$ , we obtain

$$-I_t = \frac{r_i + r_e}{r_i r_e} \frac{\partial U_i}{\partial x} - \frac{1}{r_e} \frac{\partial U}{\partial x},$$

rearranging some terms, we arrive at

$$\frac{1}{r_i} \frac{\partial U_i}{\partial x} = \frac{1}{r_e + r_i} \frac{\partial U}{\partial x} - \frac{r_e}{r_e + r_i} I_t. \quad (1.9)$$

Substituting equation (1.9) into (1.8) yields

$$I_t = \frac{\partial}{\partial x} \left( \frac{r_e}{r_e + r_i} \frac{\partial U}{\partial x} \right). \quad (1.10)$$

Here we have used the fact that  $I_t$  is constant and equation (1.6). Finally, as the transmembrane current is the sum of ionic and capacitive currents, we have that

$$I_t = p \left( C_m \frac{\partial U}{\partial t} + I_{ion} \right) = \frac{\partial}{\partial x} \left( \frac{r_e}{r_e + r_i} \frac{\partial U}{\partial x} \right), \quad (1.11)$$

where  $p$  is the axon perimeter measured in cm,  $C_m$  is the capacitance per unit of area measured in  $\frac{\mu\text{F}}{\text{cm}^2}$  and  $I_{ion}$  is the current per unit of area whose units are A cm. Equation (1.11) is called the cable equation.

In practice, it is common to use an equivalent dimensionless equation to equation (1.11) to avoid confusion with the units of each variable. To achieve this, let us define the membrane resistance  $R_m$  as the resistance per unit of area of the membrane  $\Omega \text{cm}^2$ . For a given(fixed) value  $U_0$ ,  $R_m$  is determined by measuring the change of the membrane current with respect to the change of the voltage due to  $U_0$ , that is

$$\frac{1}{R_m} = \frac{dI_{ion}}{dU} \Big|_{U=U_0}. \quad (1.12)$$

Usually  $U_0$  is taken as the resting potential of the membrane to define  $R_m$ . Let us assume that both  $r_e$  and  $r_i$  are constant. Then, multiplying equation (1.11) by  $R_m$  yields

$$R_m C_m \frac{\partial U}{\partial t} + R_m I_{ion} = \frac{R_m}{p(r_e + r_i)} \frac{\partial^2 U}{\partial x^2}. \quad (1.13)$$

The term  $R_m C_m$  has units of time. That is due to the fact that

$$\Omega \text{cm}^2 \cdot \frac{\text{F}}{\text{cm}^2} = \frac{U \text{ As}}{\text{A} \text{ U}} = \text{s}.$$

For this reason,  $R_m C_m$  is called the membrane time constant  $\tau_m$ .

Now, the units of the term  $\sqrt{\frac{R_m}{p(r_e + r_i)}}$  are length units. That is because

$$\sqrt{\frac{\Omega \text{cm}^2}{\text{cm}(\frac{\Omega \text{cm}^2}{\text{cm}^2})}} = \text{cm}.$$

This is why this term is referred as the space constant of the cable  $\lambda_m$ .

By substituting  $\tau_m$  and  $\lambda_m$  in equation (1.13) we obtain

$$\tau_m \frac{\partial U}{\partial t} + R_m I_{ion} = \lambda^2 \frac{\partial^2 U}{\partial x^2}. \quad (1.14)$$

The extracellular resistance is negligible due to the fact that the extracellular area is significantly greater in comparison with the intracellular one. Because of this, the term  $\lambda_m$  is reduced to  $\lambda_m = \sqrt{\frac{R_m}{p r_i}} = \sqrt{\frac{R_m d}{4 R_c}}$ , where  $d$  is the axon's diameter. Finally, we rescale the ionic current. This can be written, in general, as  $I_{ion} = -\frac{f(U,t)}{R_m}$ , where  $f$  is a function that depends on time and voltage. Then, we take  $T = \frac{t}{\tau_m}$  and  $f = f(U,T)$ . Under these considerations and taking  $X = \frac{x}{\lambda_m}$ , equation (1.14) takes the form

$$\frac{\partial U}{\partial T} = \frac{\partial^2 U}{\partial X^2} + f(U,T), \quad (1.15)$$

which is known as the dimensionless cable equation.

### 1.3 Cardiac Dynamics and Reaction-Diffusion Systems

Reaction diffusion systems arise in the study of population dynamics Mogilner and Edelstein-Keshet (2002), epidemiology Demongeot et al. (2012), biology Cantrell and

Cosner (2004); Volpert and Petrovskii (2009) and physiology Jalife (2003). Classic examples are the modelling of animal coat patterns Murray (2002b); Murray (2002a), the Belousov-Zhabotinski(BZ) reaction Winfree (1984), the Hodgkin-Huxley model of the propagation of action potencials along the nerve cells Hodgkin and Huxley (1952), and models of the propagation of a disease in an ecosystem. The general form of a reaction diffusion system is given by

$$\frac{\partial \mathbf{V}}{\partial t} = \nabla \cdot (D \nabla \mathbf{V}) + \mathbf{F}(\mathbf{V}), \quad (1.16)$$

where  $\mathbf{V} = (V_1, V_2, \dots, V_N)$  with  $V_i = V_i(\mathbf{x}, t)$ ,  $i = 1, N$  might represent the concentrations of  $N$  chemical species in a chemical reaction, or the number of susceptible, infected and recovered individuals at position  $\mathbf{x}$  and time  $t$  for a model in epidemiology. The function  $\mathbf{F}$  which is called the reaction term, depends on the  $V_i$  variables and models the interaction between the  $V_i$  species at a single point  $x$ . The spatial variation  $V_i(\mathbf{x}, t)$  is modeled with the diffusion term  $\nabla \cdot (D \nabla \mathbf{V})$ , where  $\mathbf{D}$  is the diffusion coefficient matrix.

In this work we focus on the particular case of the bioelectric activity of the heart which is subject of a vast interdisciplinary literature in medicine, bioengineering, mathematical biology and physics to name a few Aliev and Panfilov (1996); Biktashev et al. (1999); Keener and Bogar (1998).

The propagation of electrical waves through cardiac tissue is a very important phenomenon to study since those waves activate the mechanisms for cardiac contraction, responsible to pump blood to the body. An electrical wave of excitation, called also an action potential wave, is initiated periodically at a place called the sinoatrial node, the natural pacemaker of the heart. This wave, propagates throughout the atria where it arrives at the atrioventricular node, where after some time delay, it propagates to the ventricles via the Purkinje fibers Setaro et al. (1992). In normal conditions, this process is repeated approximately 70 to 100 times each minute and is commonly referred to as a heartbeat. The condition at which abnormal generation or propagation of excitation waves during the process described above, is termed as arrhythmia.

One of the proposed mechanisms involved in the development of certain type of arrhythmias, are spiral waves, a particular form of functional reentry Fenton et al. (2002); Nanthakumar et al. (2004). Spiral waves, are self sustained waves of excitation that rotate freely or around an obstacle, reactivating the same area of tissue at a higher frequency than the normal SA node would do, increasing the normal heartbeat rate. Figure 1.3 (A) depicts a typical spiral wave solution in a cardiac model. In the worst scenario, a spiral wave might break up into smaller spiral waves giving uncoordinated contractions of the heart in a phenomenon known as fibrillation. An example of multiple spiral waves is shown in Figure 1.3 (B).

When this phenomenon occurs in the ventricles, the heart quivers and loses its strength to pump blood to the body leading to immediate cardiac arrest Fenton et al. (2002); Nanthakumar et al. (2004).

An important research area is the study of the interaction of spiral waves in cardiac tissue with obstacles. Obstacles in cardiac tissue can be partially excitable

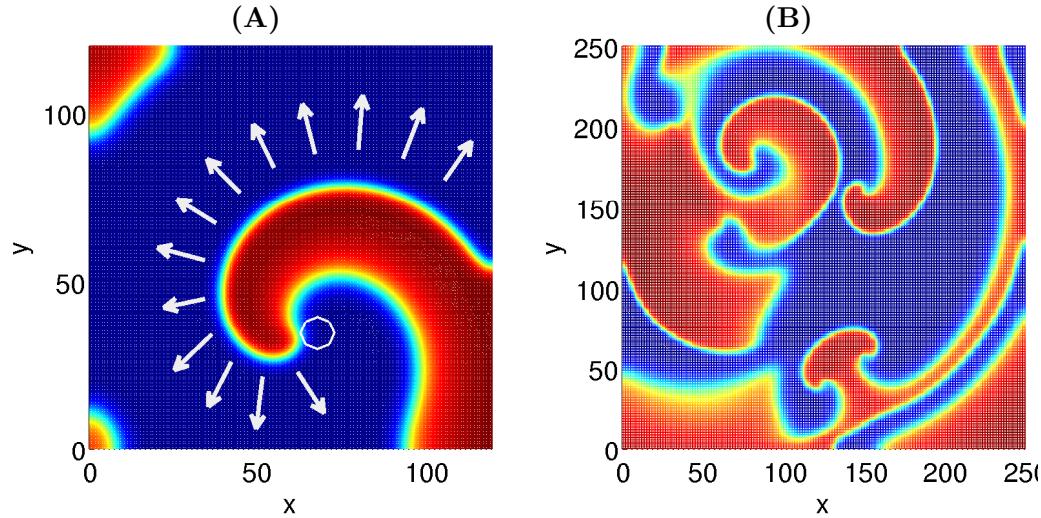


Figure 1.3: Typical solutions in a cardiac model. In frame (A): Spiral wave solution. Frame (B): Solution with multiple spiral waves.

or non excitable.

Spiral waves have been observed to occur in cardiac tissue [Ikeda et al. \(1997\)](#); [Isomura et al. \(2008\)](#); [Pertsov et al. \(1993\)](#) and in computer models [Isomura et al. \(2008\)](#); [Olmos \(2010\)](#); [Otani \(2000\)](#).

There are different ways in which a spiral wave might be generated. For example, spiral waves arise when an unexcitable obstacle is stimulated with high frequency of AP [Panfilov and Keener \(1993\)](#); they can also be generated by using the method of cross-field stimulation [Pertsov et al. \(1993\)](#); and they might arise due to the appearance of ectopic beats [Otani \(2000\)](#). Ectopic beats can arise due to abnormal calcium cycling [Benson and Holden \(2005\)](#) or by overload of calcium inside the cell [Luo and Rudy \(1994\)](#).

When a spiral wave evolves in excitable media in general, its dynamics are ruled by (*i*) the local conducting mechanisms, and; (*ii*) the heterogeneities of the medium. The former gives rise to a phenomenon called meandering, whereas the later to a phenomenon referred to as drift of a spiral wave.

One way to get a better understanding of meandering and drift of a spiral wave, is by studying the evolution of the position of its tip. The tip of a spiral wave can be defined in a variety of ways and a resume can be found in [Fenton et al. \(2002\)](#).

### On the Numerical Solution of Cardiac Wave Equations

Computational studies and numerical simulations have played an important role in electrocardiology. Due to the difficulty of direct measurements, many experimental studies have been coupled with numerical investigations. Particularly intense has

been the computational study of reentry phenomena and their relationships with cardiac arrhythmias Fenton et al. (2002).

The vast majority of cases when studying excitable media and cardiac dynamics, the numerical methods used to solve the model equations are based on finite differences schemes Bär and Or-Guil (1999); Panfilov and Rudenko (1987); Woltering and Markus (2002); Zaritski and Pertsov (2002); Keener (1980); Biktashev et al. (1994); Panfilov and Winfree (1985); Clayton et al. (2011); Alonso and Panfilov (2008); Bernus et al. (2002); Efimov et al. (1995); Garzón et al. (2009); Panfilov (1998); Ramirez et al. (2000); Virag et al. (2002); Watanabe et al. (2001).

Our main goal is to develop numerical methods that will provide a fast, reliable and accurate solutions of PDE equations of the reaction-diffusion type, with special emphasis in two and three dimensional systems that model cardiac wave propagation. A second objective is to provide estimates on the number of discretization points and the time step, in order to obtain a given desired accuracy for any cardiac models based on the Hodgkin-Huxley formulation Hodgkin and Huxley (1952) and not only the models presented here.

The time integration schemes considered in this manuscript are straightforward. Specifically, we use explicit forward finite difference, a semi-implicit scheme and a straightforward implicit scheme. Even though these methods are of low order convergence and we might lose precision, we focus our efforts on the spatial discretization where we use a pseudospectral approach that will yield good results as shown later in this work. The purpose of using such time integrators schemes, is to build algorithms that are as simple as possible for implementation and yet provide with good numerical accuracy and reduced computing time.

The spatial discretization of our methods is based on a pseudospectral approach which have proven to provide accurate solutions with a few number of discretization points. Particularly, schemes based on Chebyshev polynomials, which has been useful to solve equations of the reaction-diffusion type Olmos and Shizgal (2006, 2009); Olmos-Liceaga and Segundo-Caballero (2016).

Pseudospectral methods are generally considered useful for solving smooth problems and to provide exponential convergence of the solution with respect to the number of collocation points used Shizgal (2002). Also, it has been demonstrated that pseudospectral methods can provide a significant improvement over finite difference methods for non smooth problems that develop shock-like steep fronts Olmos and Shizgal (2006). Due to their applicability, they have been used to solve the Poisson and Laplace equations Tal-Ezer (1989), parabolic problems in general Sabetghadam et al. (2009), as well as pseudospectral methods to find the numerical solution of reaction diffusion equations (Jones and O'Brien (1996); Bar-Yoseph et al. (1995); Feng et al. (2006); Barillot and Boissonade (1993); Eilbeck (1983)). Particularly, there has been work on cardiac dynamics equations using Fourier in irregular domains Bueno-Orovio et al. (2006) and the solution of fractional reaction-diffusion equations (Bueno-Orovio et al. (2014); Li and Xu (2009); Zayernouri and Karniadakis (2014)) as well as solving systems on more complex domains (Orszag (1980); Korczak and Patera (1986); Mulholland et al. (1998); Lui (2009)). The discretized equations that result from a pseudospectral method are stiff and generally a very

small step is required for their integration.

Even though there has been previous work utilizing Fourier spectral methodology, we focus on an approach based on Chebyshev polynomials. The method takes the advantage of a blockwise diagonal matrix, which is very helpful for numerical computations and it helps to reduce the norm of the largest eigenvalue associated to the diffusion operator.

Although in this work a comparison is made between finite differences (FD) and pseudospectral methods, it is not the main purpose to claim that pseudospectral methods are better. We use FD to place the reader in a strong position to compare the results, as FD methods are widely studied and used.

## 1.4 Solutions of the Spiral and Scroll Wave Type in Cardiac Dynamics

Since the discovery of the Belousov-Zhabotinski reaction Zhabotinsky (1964); Winfree (1984), there has been a wide variety of studies that focus on wave patterns on reaction-diffusion equations. Similar phenomena have been to exist in biological excitable systems: these are rotating waves of spreading depression in the brain and rotating waves of excitation in myocardial tissue Allessie et al. (1973). Such waves are responsible for some diseases, in particular, cardiac arrhythmias Krinsky (1978).

In most biological systems and active media, such as cardiac tissue Jalife (2003), the retina Gorelova and Bureš (1983); Lansdell et al. (2014); Dahlem and Müller (2004) and the brain Skaggs et al. (1988), the rotation of spiral waves close to a boundary is of interest. This due to the fact that the dimensions of the region where the spiral wave evolves is comparable with the size of the region of its core.

Unlike Wiener and Rosenblueth's two dimensional theoretical abstraction Wiener (1946), real excitable and oscillatory media, including the BZ reaction and heart tissue, are three dimensional Verschelde et al. (2007); Clayton and Holden (2003). While spiral waves rotate around their cores, which can be considered point-like geometric objects, scroll waves rotate around filaments, line-like geometric objects. As spiral and scroll waves do not require any obstacle to rotate about, they can be located anywhere in the reactive medium. An immediate consequence is that their position can change in time, i.e., they can drift Keener and Tyson (1990, 1992); Fenton et al. (2002); Vinson et al. (1997). The scroll waves in three dimensions have more degrees of freedom: their filament can also change in space. The phase of rotation may vary along the filament, the feature known as twist of the scroll wave Biktashev et al. (1994). Twist of a scroll wave and curvature of its filament are specifically three dimensional factors for its dynamics.

Although there has been some studies on the interaction of spiral waves and scroll rings with boundaries Pertsov et al. (1984); Ermakova and Pertsov (1986); Jiménez and Steinbock (2012); Bánsági Jr et al. (2008), this phenomenon is far from being completely understood. One of the main goals of this work is to study scroll ring dynamics in the long time along with their interaction with impermeable boundaries. In particular, we present a study of scroll rings for the weak meandering and positive

filament tension regime. It is well known that scroll waves in the positive filament tension regime shrink until they disappear in the medium Panfilov and Winfree (1985). However, under some conditions, scroll waves shrink but do not collapse as they start interacting with the boundary. In this work, we analyse the different outcomes for the scroll rings in the weak meandering regime as they interact with the boundary in the positive filament tension parameter regime. Reports of studies of the interaction of the solutions with boundaries for this parameter regime has not been as extensive as the negative filament tension.

Also, based on the theory developed by Keener Keener (1986); Keener and Tyson (1990) and an analysis on the model parameters  $M$  and  $Re$ , we present some data fitting that will allow us to state predictions on the scroll ring's dynamics, which constitutes the other main purpose of this work.

# Chapter 2

## Preliminaries

As we have stated earlier, our main goal is to solve equations of the form

$$\frac{\partial u}{\partial t} = \nabla \cdot D \nabla u + f(u), \quad (2.1)$$

where  $f(u)$  is nonlinear function,  $\nabla$  is the gradient operator and  $D$  is a diffusion tensor. Let us assume, for now, that  $u = u(x, t)$  and that we make a partition of our domain consisting of  $n$  equally spaced points,  $x_i$ , with  $x_0$  and  $x_n$  being the initial point and the last point, respectively, of our domain. Thus  $x_i = x_0 + i\Delta x$ ,  $i = 0, 1, 2, \dots, n$ . We proceed to describe the first of the numerical methods we used throughout our studies.

### 2.1 Euler Finite Difference

Recall that for a given sufficiently smooth function  $f$ , we can approximate its first derivative at a point  $x_i$  by means of a forward finite difference

$$f'(x_i) = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} + \mathcal{O}(\Delta x) \quad (2.2)$$

and its second derivative at a point  $x_i$  using a centred finite difference

$$f''(x_i) = \frac{f(x_i + \Delta x) + f(x_i - \Delta x) - 2f(x_i)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \quad (2.3)$$

For our first numerical method, the explicit scheme, we use forward finite difference and centred finite difference to approximate time and space respectively. By assuming that  $D = 1$  in equation (2.1), meaning that the diffusion coefficient is constant, we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} + f(u_i^n), \quad (2.4)$$

where  $u_i^n$  denotes the function  $u$  evaluated at point  $x_i$  at time step  $n$ . Thus, by rearranging some terms in (2.4) we have

$$u_i^{n+1} = u_i^n + \Delta t \left( \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} + f(u_i^n) \right). \quad (2.5)$$

Let us suppose now that  $u = u(x, y, t)$ . For simplicity in writing, let us assume that the partition of points in each dimension,  $x_i$  and  $y_i$ , are equally spaced, that is,  $\Delta x = \Delta y$ . Then, by following the same reasoning as earlier, we obtain the following:

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \left( \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n}{\Delta x^2} + f(u_{i,j}^n) \right). \quad (2.6)$$

Finally, for the case when  $u = u(x, y, z, t)$  and making the assumption that we discretize our domain exactly the same in each dimension ( $\Delta x = \Delta y = \Delta z$ ), we have that:

$$u_{i,j,k}^{n+1} = u_{i,j,k}^n + \Delta t \left( \frac{u_{i+1,j,k}^n + u_{i-1,j,k}^n + u_{i,j+1,k}^n + u_{i,j-1,k}^n + u_{i,j,k+1}^n + u_{i,j,k-1}^n - 6u_{i,j,k}^n}{\Delta x^2} + f(u_{i,j,k}^n) \right). \quad (2.7)$$

### Special Case: Rotational Anisotropy.

We are interested in the case when we have distinct diffusion coefficients. Let us consider the case when our tensor matrix  $D$  is of the form

$$\tilde{\mathbf{D}} = \begin{pmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{\perp 2} \end{pmatrix}, \quad (2.8)$$

where

$$\begin{aligned} D_{11} &= D_{\parallel} \cos^2 \theta(z) + D_{\perp 1} \sin^2 \theta(z), \\ D_{22} &= D_{\parallel} \sin^2 \theta(z) + D_{\perp 1} \cos^2 \theta(z), \\ D_{12} &= D_{21} = (D_{\parallel} - D_{\perp 1}) \cos \theta(z) \sin \theta(z), \end{aligned} \quad (2.9)$$

with  $D_{\parallel}$ ,  $D_{\perp 1}$  and  $D_{\perp 2}$  representing propagation parallel to the fiber axis, perpendicular to this axis in each plane, and transmurally, respectively, and

$$\theta(z) = -\Delta\theta/2 + z(\Delta\theta/S) \quad 0 \leq z \leq S, \quad (2.10)$$

measures the angle between the fiber and the  $y$ -axis in each plane.

Under this circumstances our laplacian in (2.1) takes the form

$$\nabla \cdot [\tilde{\mathbf{D}} \nabla u] = D_{11} \frac{\partial^2 u}{\partial x^2} + D_{22} \frac{\partial^2 u}{\partial y^2} + D_{\perp 2} \frac{\partial^2 u}{\partial z^2} + 2D_{12} \frac{\partial^2 u}{\partial x \partial y}.$$

Now, we use again forward finite difference and centred finite difference to approximate time and space respectively. With this in mind, we obtain the numerical scheme

$$\begin{aligned} u_{i,j,k}^{n+1} &= u_{i,j,k}^n + \Delta t \left( D_{11} \frac{u_{i+1,j,k}^n + u_{i-1,j,k}^n - 2u_{i,j,k}^n}{\Delta x^2} + D_{22} \frac{u_{i,j+1,k}^n + u_{i,j-1,k}^n - 2u_{i,j,k}^n}{\Delta y^2} + \right. \\ &\quad \left. D_{\perp 2} \frac{u_{i,j,k+1}^n + u_{i,j,k-1}^n - 2u_{i,j,k}^n}{\Delta z^2} + 2D_{12} \frac{u_{i+1,j+1,k}^n - u_{i+1,j-1,k}^n - u_{i-1,j+1,k}^n + u_{i-1,j-1,k}^n}{\Delta x \Delta y} + f(u_{i,j,k}^n) \right) \end{aligned} \quad (2.11)$$

### 2.1.1 More Realistic Geometries

To treat irregular geometries using the phase field methodology (Fenton et al. (2005)), we introduce an auxiliary field  $\phi$  that takes on different values inside and outside cardiac tissue and varies smoothly across a thin diffusive interface connecting these two regions.

In the interior and the exterior of the heart,  $\phi$  is initially set to 1 and 0, respectively. The smooth values of  $\phi$ , which are to be used later in wave propagations simulations, are then determined by solving the equation

$$\frac{\partial \phi}{\partial t} = \xi^2 \nabla^2 \phi - \frac{\partial G(\phi)}{\partial \phi}, \quad (2.12)$$

where  $\xi$  is a parameter that controls the width of the interface and the function  $G(\phi)$  has the form of a double-well potential with minima at  $\phi = 0$  and  $\phi = 1$ . Therefore, it attempts to maintain  $\phi$  at the values 1 and 0 in the interior and exterior regions, respectively, whereas the diffusion operator tends to smooth out the spatial discontinuity of  $\phi$  at the boundary between these two regions. For our simulations,  $G$  has the form

$$G(\phi) = -\frac{(2\phi - 1)^2}{4} + \frac{(2\phi - 1)^4}{8}. \quad (2.13)$$

Once we have our function  $\phi$  which defines the structure, we have to solve the new system

$$\phi \frac{\partial V}{\partial t} = \nabla \cdot [\mathbf{D}\phi \nabla V] - \phi \frac{I_{ion}}{C_m}, \quad (2.14)$$

or, equivalently

$$\frac{\partial V}{\partial t} = \frac{1}{\phi} \nabla \cdot [\mathbf{D}\phi \nabla V] - \frac{I_{ion}}{C_m}. \quad (2.15)$$

First, let us assume that we have the same diffusion in each dimension, that is, the matrix  $\mathbf{D}$  is diagonal, and suppose also, to make the calculations easier, that our diffusion coefficients for each dimension is one. Then, our matrix  $\mathbf{D}$  has the form

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us focus on the laplacian term on equation (2.15). Expanding said term, we have

$$\nabla \cdot [\mathbf{D}\phi \nabla V] = \frac{\partial \phi}{\partial x} \frac{\partial V}{\partial x} + \phi \frac{\partial^2 V}{\partial x^2} + \frac{\partial \phi}{\partial y} \frac{\partial V}{\partial y} + \phi \frac{\partial^2 V}{\partial y^2} + \frac{\partial \phi}{\partial z} \frac{\partial V}{\partial z} + \phi \frac{\partial^2 V}{\partial z^2}. \quad (2.16)$$

Now we can get a numerical scheme to approximate (2.16),

$$\begin{aligned} \nabla \cdot [\mathbf{D}\phi \nabla V] &= \frac{1}{4\Delta x^2} [(\phi_{i+1,j,k} - \phi_{i-1,j,k})(V_{i+1,j,k} - V_{i-1,j,k})] + \frac{\phi_{i,j,k}}{\Delta x^2} [V_{i+1,j,k} + V_{i-1,j,k} - 2V_{i,j,k}] + \\ &\quad \frac{1}{4\Delta y^2} [(\phi_{i,j+1,k} - \phi_{i,j-1,k})(V_{i,j+1,k} - V_{i,j-1,k})] + \frac{\phi_{i,j,k}}{\Delta y^2} [V_{i,j+1,k} + V_{i,j-1,k} - 2V_{i,j,k}] + \\ &\quad \frac{1}{4\Delta z^2} [(\phi_{i,j,k+1} - \phi_{i,j,k-1})(V_{i,j,k+1} - V_{i,j,k-1})] + \frac{\phi_{i,j,k}}{\Delta z^2} [V_{i,j,k+1} + V_{i,j,k-1} - 2V_{i,j,k}]. \end{aligned} \quad (2.17)$$

We can also implement rotational anisotropy for this scenario. The laplacian term for the case when  $D$  is of the form (2.8) takes the form

$$\begin{aligned} \nabla \cdot [\tilde{\mathbf{D}}\phi \nabla V] &= \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \left[ \begin{pmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{\perp 2} \end{pmatrix} \begin{pmatrix} \phi \frac{\partial V}{\partial x} \\ \phi \frac{\partial V}{\partial y} \\ \phi \frac{\partial V}{\partial z} \end{pmatrix} \right] \\ &= \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} D_{11}\phi \frac{\partial V}{\partial x} + D_{12}\phi \frac{\partial V}{\partial y} \\ D_{21}\phi \frac{\partial V}{\partial x} + D_{22}\phi \frac{\partial V}{\partial y} \\ D_{\perp 2}\phi \frac{\partial V}{\partial z} \end{bmatrix} \\ &= D_{11}\frac{\partial \phi}{\partial x} \frac{\partial V}{\partial x} + D_{11}\phi \frac{\partial^2 V}{\partial x^2} + D_{22}\frac{\partial \phi}{\partial y} \frac{\partial V}{\partial y} + D_{22}\phi \frac{\partial^2 V}{\partial y^2} + D_{\perp 2}\frac{\partial \phi}{\partial z} \frac{\partial V}{\partial z} + \\ &\quad D_{\perp 2}\phi \frac{\partial^2 V}{\partial z^2} + 2D_{12}\phi \frac{\partial^2 V}{\partial x \partial y} + D_{12} \left( \frac{\partial \phi}{\partial y} \frac{\partial V}{\partial x} + \frac{\partial \phi}{\partial x} \frac{\partial V}{\partial y} \right). \end{aligned} \quad (2.18)$$

Under this considerations, our numerical scheme is given by:

$$\begin{aligned} \nabla \cdot [\tilde{\mathbf{D}}\phi \nabla V] &= \frac{D_{11}}{4\Delta x^2} [(\phi_{i+1,j,k} - \phi_{i-1,j,k})(V_{i+1,j,k} - V_{i-1,j,k})] + \frac{D_{11}\phi_{i,j,k}}{\Delta x^2} [V_{i+1,j,k} + V_{i-1,j,k} - 2V_{i,j,k}] + \\ &\quad \frac{D_{22}}{4\Delta y^2} [(\phi_{i,j+1,k} - \phi_{i,j-1,k})(V_{i,j+1,k} - V_{i,j-1,k})] + \frac{D_{22}\phi_{i,j,k}}{\Delta y^2} [V_{i,j+1,k} + V_{i,j-1,k} - 2V_{i,j,k}] + \\ &\quad \frac{D_{\perp 2}}{4\Delta z^2} [(\phi_{i,j,k+1} - \phi_{i,j,k-1})(V_{i,j,k+1} - V_{i,j,k-1})] + \frac{D_{\perp 2}\phi_{i,j,k}}{\Delta z^2} [V_{i,j,k+1} + V_{i,j,k-1} - 2V_{i,j,k}] + \\ &\quad \frac{2D_{12}\phi_{i,j,k}}{4\Delta x \Delta y} [V_{i+1,j+1,k} - V_{i+1,j-1,k} - V_{i-1,j+1,k} + V_{i-1,j-1,k}] + \\ &\quad \frac{D_{21}}{4\Delta x \Delta y} [(\phi_{i,j+1,k} - \phi_{i,j-1,k})(V_{i+1,j,k} - V_{i-1,j,k}) + (\phi_{i+1,j,k} - \phi_{i-1,j,k})(V_{i,j+1,k} - V_{i,j-1,k})] \end{aligned} \quad (2.19)$$

## 2.2 Historical Background on Pseudospectral Methods

Spectral methods have become increasingly popular in recent years, especially since the development of fast transform methods, with applications in numerical weather prediction, numerical simulations of turbulent flows, and other problems where high accuracy is desired for complicated solutions.

The origin of the terminology “spectral” is not entirely clear but probably arises from the original use of Fourier sines and cosines as basis functions ([Gottlieb and Orszag \(1977\)](#)) especially in connection with a time series analysis and the fundamental frequencies of a process, namely the “spectrum”.

Spectral methods may be viewed as an extreme development of the class of discretization schemes for differential equations known generically as the method of weighted residuals (MWR) ([Finlayson and Scriven \(1966\)](#)). The key elements of the MWR are the trial functions (also called the expansion or approximating functions) and the test functions (also known as weight functions). The trial functions are used as the basis functions for a truncated series expansion of the solution. The test functions are used to ensure that the differential equation is satisfied as closely as possible by the truncated series expansion. This is achieved by minimizing the residual, i.e., the error in the differential equation produced by using the truncated expansion instead of the exact solution, with respect to a suitable norm. An equivalent requirement is that the residual satisfy a suitable orthogonality condition with respect to each of the test functions.

The choice of trial functions is one of the features which distinguish spectral methods from finite-element and finite-difference methods. The trial functions for

spectral methods are infinitely differentiable global functions. In the case of finite-element methods, the domain is divided into small elements, and a trial function is specified in each element. The trial functions are thus local in character, and well suited for handling complex geometries. The finite-difference trial functions are likewise local.

The choice of test functions distinguishes between the three most commonly used spectral schemes, namely, the Galerkin, collocation, and tau versions. In the Galerkin approach, the test functions are the same as the trial functions. They are, therefore, infinitely smooth functions which individually satisfy the boundary conditions. The differential equation is enforced by requiring that the integral of the residual times each test function be zero. In the collocation approach the test functions are translated Dirac delta functions centered at special, so-called collocation points. This approach requires the differential equation to be satisfied exactly at the collocation points. Spectral tau methods are similar to Galerkin methods in the way that the differential equation is enforced. However, none of the test functions need satisfy the boundary conditions. Hence, a supplementary set of equations is used to apply the boundary conditions.

The earliest applications of the spectral collocation method to partial differential equations were made for spatially periodic problems by Kreiss and Oliger (1972) (who called it the Fourier method) and Orszag (1972) (who termed it pseudospectral). This approach is especially attractive because of the ease with which it can be applied to variable-coefficient and even non-linear problems.

Spectral methods are generally based on the representation of a real, continuous, “well-behaved” function,  $f(x)$ , on some interval not necessarily bounded as an expansion in an orthonormal set of functions,  $P_n(x)$ , that is,

$$f(x) = \sum_{n=0}^{\infty} a_n P_n(x), \quad x \in [a, b], \quad (2.20)$$

where the polynomials are orthonormal

$$\int_a^b w(x) P_n(x) P_m(x) dx = \delta_{nm}, \quad (2.21)$$

with respect to some appropriate weight function,  $w(x)$ , and the Kronecker delta is defined by

$$\delta_{nm} = \begin{cases} 1, & n = m, \\ 0, & n \neq m. \end{cases} \quad (2.22)$$

The term “pseudospectral” refers to the solution of the defining equations on a grid of discrete points,  $x_i$ , and the solution,  $f(x_i)$ , as determined at the grid points. This is often referred to as a collocation.

Spectral methods are distinguished not only by the type of the method (Galerkin, collocation, or tau), but also by the particular choice of the trial functions. The most

frequently used trial functions are trigonometric polynomials, Chebyshev polynomials, and Legendre polynomials. Throughout this work we will use Chebyshev polynomials.

### 2.2.1 Spatial Derivative by Chebyshev Pseudospectral Multidomain Method

In this section, we explain how the derivative in space will be implemented ([Shizgal \(2002\)](#), [Olmos and Shizgal \(2006\)](#)), as most of the numerical schemes are based on this approach. Also, we present the details of the pseudospectral method of solution of the Karma equation based on Chebyshev polynomials. The Chebyshev polynomials,  $T_k(z)$ , are orthogonal with respect to the weight function  $w(z) = (1 - z^2)^{-1/2}$  on the interval  $[-1, 1]$ , that is,

$$\frac{1}{c_k} \int_{-1}^1 w(z) T_k(z) T_l(z) dz = \frac{1}{2} \pi \delta_{k,l}, \quad (2.23)$$

where  $c_k = 1$  for all  $k$  except for  $c_0 = 2$ . The Lobatto quadrature points and weights associated with the Chebyshev-Gauss polynomials are given by  $x_i = -\cos(\pi i/N)$  and the weights are  $w_i = \pi/N$  for all  $i$  except  $w_0 = w_N = 2\pi N$  ([Peyret \(2002\)](#); [Boyd \(2001\)](#); [Canuto \(1988\)](#); [Canuto et al. \(2012\)](#)). These points and weights provide the approximate quadrature,

$$\int_{-1}^1 w(z) f(z) dz \simeq \sum_{i=0}^N w_i f(z_i), \quad (2.24)$$

where  $N$  is the number of points. Since any piecewise continuous function,  $f \in L_w^2[0, 1]$ , can be expanded in a Chebyshev polynomial series that is convergent in the mean of the  $L_w^2$  norm , we have that

$$f(z) \approx f_N(z) = \sum_{k=0}^N a_k T_k(z), \quad (2.25)$$

where

$$a_k = \frac{2}{c_k \pi} \int_{-1}^1 w(z) f(z) T_k(z) dz. \quad (2.26)$$

From relations (2.24) and (2.25) we can express (2.26) as a sum

$$a_k = \frac{2}{c_k \pi} \int_{-1}^1 w(z) f(z) T_k(z) dz \simeq \sum_{i=0}^N w_i(z_i) f(z_i) T_k(z_i). \quad (2.27)$$

With equations (2.24)-(2.26) we obtain the interpolation algorithm

$$f_N(z) \simeq \sum_{j=0}^N I_j(z) f(z_j), \quad (2.28)$$

where the interpolating polynomials,  $I_j(z)$ , are given by

$$I_j(z) = \frac{2\nu_j}{N} \sum_{k=0}^N \nu_k T_k(z_j) T_k(z), \quad (2.29)$$

where  $\nu_0 = \nu_N = 1/2$  with  $\nu_k = 1$  if  $k \neq 0, N$  and where  $I_j(z_i) = \delta_{ij}$  is the cardinal condition such that  $f_N(z_i) = f(z_i)$ . The second derivative of  $f(z)$  at the quadrature points is then given approximately by

$$f_N^{(2)}(z_k) \simeq \sum_{k=0}^N I_j^{(2)}(z_k) f(z_j). \quad (2.30)$$

If we denote by  $\mathbf{f}$ , the  $N$  dimensional vector of the function evaluated at the Chebyshev-Lobatto points, equation (2.30) can be rewritten as

$$f^{(2)} = D^{(2)} \cdot \mathbf{f}, \quad (2.31)$$

where the second derivative matrix is given explicitly by

$$D_{jk}^{(2)} = I_j^{(2)}(z_k) = \frac{d^2 I_j(z)}{dz^2} \Big|_{z=z_k}. \quad (2.32)$$

This is the basis for the Chebyshev pseudospectral method.

### 2.2.2 The Gibbs Phenomenon

One of the main problems when approximating a function by means of Chebyshev polynomials arises when we want to approximate a function that presents discontinuities. In this case, the Gibbs phenomenon occurs, which practically says that when we want to approximate a function,  $f$ , defined on the interval  $[-\pi, \pi]$  such that  $f(-\pi) \neq f(\pi)$ , by means of trigonometric polynomials (linear combinations of  $h(x) = \cos(kx)$  and  $g(x) = \sin(kx)$ ), will imply that the approximation will present oscillations in a neighborhood of  $x = -\pi$  and  $x = \pi$ .

Formally, the Gibbs phenomenon states that when approximating a discontinuous function by a finite Fourier series, nonphysical oscillations occur. No matter how many terms are used, there will always be a point next to the discontinuity that has an  $O(1)$  error. This type of oscillations will occur whenever spectral methods are used for PDE, if the solution has shocks or contact discontinuities. Because of the global character of polynomial methods, they will show the same type of behavior as Fourier methods do. This is a serious limitation of spectral methods, but David Gottlieb and others set out to do something about it.

The difficulties with discontinuities are not limited to numerical solutions obtained by spectral methods, they show up with all shock capturing methods. Many well working difference methods based on some form of upwinding have been developed during the last decades. However, there is one remaining difficulty. If one family of characteristics is propagating through the shock, the accuracy deteriorates

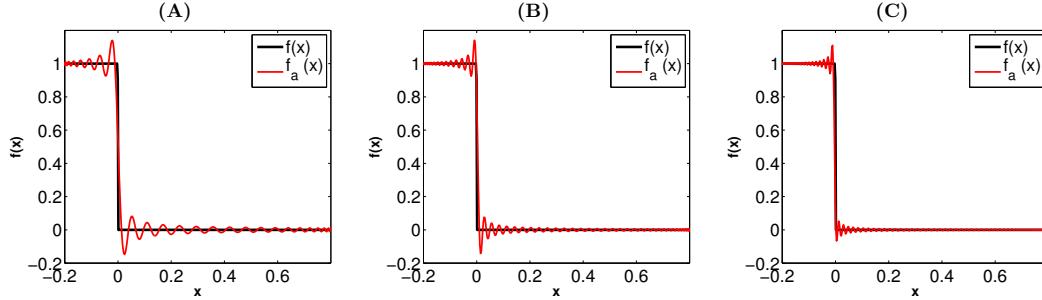


Figure 2.1: In frames (A), (B) and (C): Function  $f(x)$  (black) given by equation 2.33 along with its approximation,  $f_a(x)$  (red) using the first 50, 100 and 150 Chebyshev polynomials, respectively.

on the downstream side regardless of the formal high order accuracy of the basic method.

The work on Gibbs phenomenon was initiated by the difficulties with spectral methods. However, the results did not only influence the use of spectral methods, but also helped solving the well known difficulty with accuracy deterioration for shock capturing difference methods.

Let us observe what happens when we approximate a function that presents an abrupt change, in such a way that we can consider this change as a discontinuity, by means of the Chebyshev polynomials. Let us consider the function

$$f(x) = \frac{1}{(1 + e^{\sqrt{s/6}x})^2}, \quad (2.33)$$

where  $s = 10^8$ . This function has an abrupt change at  $x = 0$ . In figure 2.1 we show the function  $f(x)$  (black) and its approximation,  $f_a(x)$  (red) using the first 50, 100 and 150 Chebyshev polynomials in frames (A), (B) and (C), respectively. Notice that even when increasing the number of polynomials to approximate the function, still the oscillation around  $x = 0$  remains.

Clearly, we note that the approximation with Chebyshev polynomials in a neighborhood of  $x = 0$  is wrong. However, it is important to note that the function  $f$  does not exhibit a true discontinuity at  $x = 0$ , what really happens is that it has an exponential change in a very small spatial scale. The problem lies in that when calculating the polynomial approximation, it is necessary to consider only points of the function  $f(x)$ ; and for practical purposes, the computer does not know how to differentiate between a true discontinuity and an exponential change in a very small spatial scale.

## Chapter 3

# Interaction of Solutions of the Scroll Ring Type with Boundaries for the Weak Meandering Regime

The outline of the present chapter is as follows. In section 3.1 we present the model considered in our studies, the numerical methods and parameters chosen for our purposes. We move to Section 3.2 where we discuss the experiments that were performed, after which we give some conclusion based on the results obtained. Finally, on Section 3.3 we give some concluding remarks and some ideas on the future work that this area of study offers.

### 3.1 Model and Parameters

The study was based on the two variable Karma model [Karma \(1994\)](#). The basic equations can be written in the form,

$$\begin{aligned}\frac{\partial E}{\partial t} &= \gamma \nabla^2 E + \frac{f(E,n)}{\tau_E}, \\ \frac{\partial n}{\partial t} &= \frac{g(E,n)}{\tau_n},\end{aligned}\tag{3.1}$$

where,

$$\begin{aligned}f(E,n) &= -E + [E^* - D(n)]h(E), \\ g(E,n) &= R(n)H(E - E_n) - [1 - H(E - E_n)]n,\end{aligned}\tag{3.2}$$

and  $H$  denotes the Heaviside function. Also,  $\nabla^2$  denotes the Laplace operator. The variable  $E$  is dimensionless representation of the transmembrane voltage and the variable  $n$  plays the role of a slow gate variable. Here we have used, for the  $h(E)$  function,

$$h(E) = [1 - \tanh(E - E_h)] \frac{E^2}{2}.\tag{3.3}$$

$R(n)$  and  $D(n)$  are known as the restitution and dispersion functions respectively. Here we have used

$$R(n) = \frac{1 - [1 - e^{-Re}]n}{1 - e^{-Re}},\tag{3.4}$$

---

A version of this present chapter will be submitted for publication.

$\tau_E$	$\tau_n$	$\gamma$	$E_h$	$E_n$	$E^*$	$dt$	$dr$	$dz$
5.0	250	0.001	3	1	1.5415	0.0133	0.025	0.025

Table 3.1: Parameters used in the Karma model.

for the restitution function, and

$$D(n) = n^M \quad (3.5)$$

for the dispersion function. We refer to the work done by Karma [Karma \(1994\)](#) for a fully detailed explanation of the parameters and variables of the model.

It is well known that solving PDE's in three dimensional domains is a very demanding problem, computationally speaking [Cherry et al. \(2003\)](#). As we are studying scroll rings, simulations in cartesian coordinates results in a demanding computational work. For that reason we transform our problem from cartesian to cylindrical coordinates, thus taking advantage of the angular symmetry and reducing our three dimensional dynamics to a two dimensional problem. Under this consideration we have that the Laplace operator in our new coordinate system takes the form

$$\nabla_{cyl}^2 E = \frac{\partial^2 E}{\partial r^2} + \frac{1}{r} \frac{\partial E}{\partial r} + \frac{\partial^2 E}{\partial z^2}. \quad (3.6)$$

In order to discretize system (3.6), we use the usual forward and centered finite difference to approximate the first and second derivative, respectively. Special care needs to be taken into account when  $r = 0$ , where we use the following approximation [Ozisik \(1993\)](#)

$$\nabla^2 E = 4 \frac{E_{M,j} - E_{0,j}}{\Delta r^2} \quad (3.7)$$

The solutions of the spiral wave type obtained with system (3.1) in cylindrical coordinates, correspond to scroll rings for system (3.1) in cartesian coordinates.

To solve system (3.1) we implemented Euler finite difference in parallel using a graphic processing unit. The numerical parameters for our simulations were  $dt = 0.0133$ ,  $dr = dz = 0.025$  and we solved it on a domain of  $400 \times 500$ , for  $r$  and  $z$  respectively. No-flux boundary conditions were used. The parameters of the model that we used are shown table (3.1).

### 3.1.1 Numerical Considerations

Due to the fact that most of the solutions dynamics of interest for us in this present manuscript are for the case when very large integration times are considered, it is imperative that we use parallel programming techniques as well as graphic processing units to perform the simulations. Without the usage of these techniques and equipment, to generate a single solution in either two or three dimensional domains results in a very large computing time making it very impractical. The program was written in Fortran language with OpenAcc parallelization, compiled with PGI compiler (pgfortran). Simulations were performed in a single desktop PC under Ubuntu

Scenario	1	2	3	4	5
$M$	3	4	4	5	5
$Re$	0.9	0.9	1.0	0.8	0.9

Table 3.2: Description of the parameters on each scenario studied in this work.

14.04 Linux equipped with an Intel core i7-4790k Haswell processor, 12 GB of RAM and an Nvidia GeForce GTX-970.

To generate a single solution of the spiral wave kind, where the core of the tip trajectory reaches an oscillatory state and interacts with both  $r = 0$  and  $z = 0$  boundaries, takes around 20 hours of computing time without using parallelization, while the implementation using a graphic processing unit to generate the same solution takes around 30 minutes in the same desktop PC. Now, with the equipment at our disposal we can study a broad set of possibilities in two and three dimensions with relatively low computing times and memory costs without using a computer cluster.

## 3.2 Numerical Studies

In this section we proceed to discuss the different outcomes of the interaction of scroll rings with impermeable boundaries. To this end, we describe the experimental protocol, followed by the results obtained.

In order to generate solutions of the scroll ring type we proceed as follows. In cylindrical coordinates we initiate a travelling front by setting

$$E(r_i, z_j) = 3, \quad (3.8)$$

for  $i, j = 1, \dots, 30$  and zero otherwise, and  $n = 0.5$  in the whole domain. We let the front evolve for a time,  $t^* = 600$ . Following this we set

$$E(r_i, z_j) = 0, \quad (3.9)$$

for  $i = 1, \dots, Np_r$ ,  $j = h_z, \dots, Np_z$ . Here,  $h_z$  represents the height at which we want to generate the spiral wave. We let the solution evolve for the required time. In our experiments we take a total time of integration of  $T = 280000$ , which accounts for approximately 280 seconds.

In two dimensional simulations in cartesian coordinates, the tip trajectory that the solutions of the spiral wave kind for the model traces, while varying  $M$  and  $Re$ , is shown in Figure 3.1. We are interested on regimes where the spiral wave traces a circular core or the circular core is slightly perturbed (weak meandering). In particular, we study five cases (labelled 1-5 in Figure 3.1). Table 3.2 contains a summary of the parameters  $M$  and  $Re$  on the Karma model considered in order to generate the solutions of interest. In Figure 3.2 we present the evolution of the radius of the tip trajectory in each scenario, calculated as the radius that the last three points of the tip trajectory form.

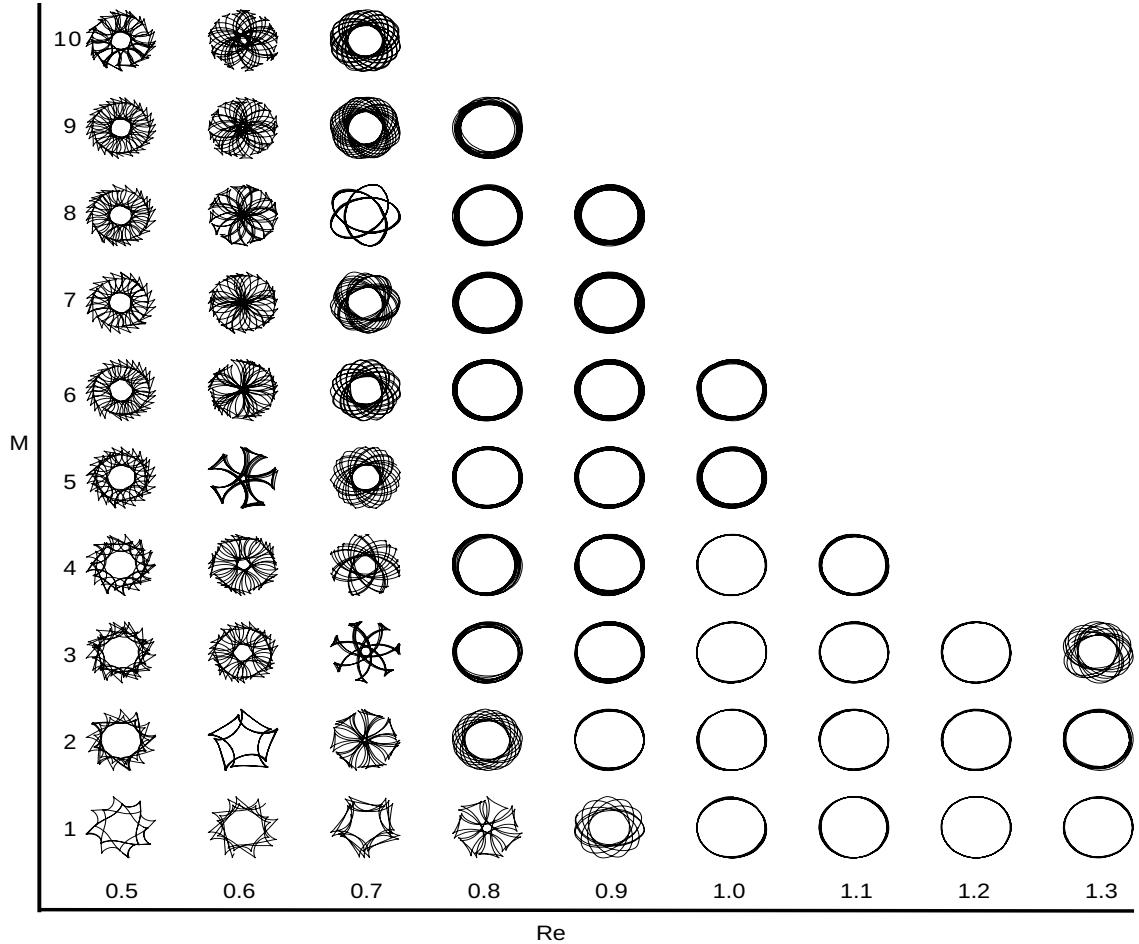


Figure 3.1: Tip trajectories in the Karma model (Eq. 3.1). From left to right we vary the  $Re$  parameter from 0.5 to 1.4 with increments of 0.1. From bottom to top, we vary the  $M$  parameter from 1 to 10 with increments of 1. The numbers 1-5 represent the cases that we study in this work. Parameters taken as discussed on table (3.1).

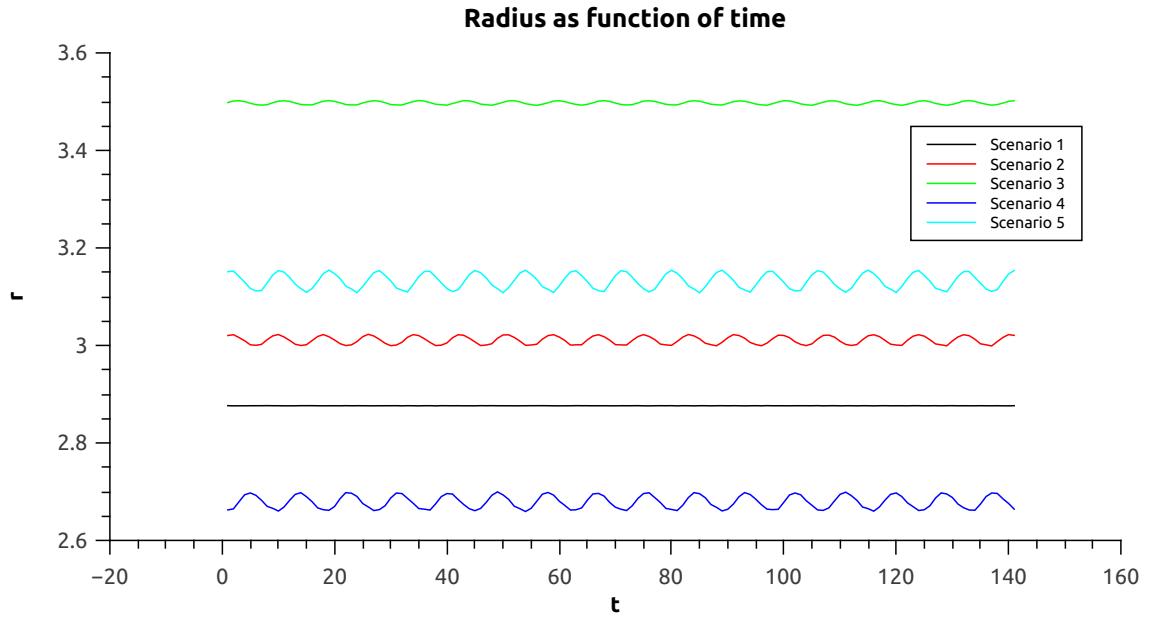


Figure 3.2: Top frame: Radius of the core as function of time on each scenario (1-5); Middle and Bottom frame have, respectively, the relative to average radius of the core as function of time for Scenarios 1-3 and 4-5.

From the five cases studied, scenario 1 represents a trajectory with variations in the radius of the order of  $10^{-4}$  which, compared to the other scenarios, is practically a circumference. Scenarios 2-5 are what we consider weak meandering behavior, due to the fact that they exhibit an oscillation of considerable amplitude in the radius of the tip trajectory.

For each of these 5 cases, we are interested in studying the corresponding three dimensional scroll ring as well as the dynamics that describe its evolution. To achieve this, we study the change in time of the variables  $r$  and  $z$  which represent a 2D slice of the scroll ring in three dimensions.

For every case, we generate spiral waves for a fixed value of  $r$  at different heights  $z$  and also generate spiral waves for a fixed  $z$  and varying  $r$ . In Figure 3.3 we show the evolution of the tip trajectory of the spiral wave in each case, whereas in figures 3.4 and 3.5 we show the evolution of the variables  $r$  and  $z$  as time evolves, corresponding to the tip trajectories shown in Figure 3.3. By starting the spiral waves at different  $z$ , we obtain some critical values that will split the  $(r, z)$  space in three different regions (Figure 3.3). In region I the spiral wave dies at the  $r = 0$  boundary, meaning that the corresponding three dimensional scroll ring will collapse. In region II, the spiral wave disappears at the  $z = 0$  boundary. In this case, the corresponding scroll ring in three dimensions also collapses but due to the interaction with two boundaries ( $r = 0$  and  $z = 0$ ). Finally, in region III, the spiral wave evolves until a time  $t = t_1^*$ , after which starts interacting with the  $z = 0$  boundary, affecting its dynamics, i.e., the spiral waves drifts along the boundary  $z = 0$ . Then, the spiral wave evolves until a time  $t = t_2^*$ , after which starts interacting with both  $r = 0$  and  $z = 0$  boundaries

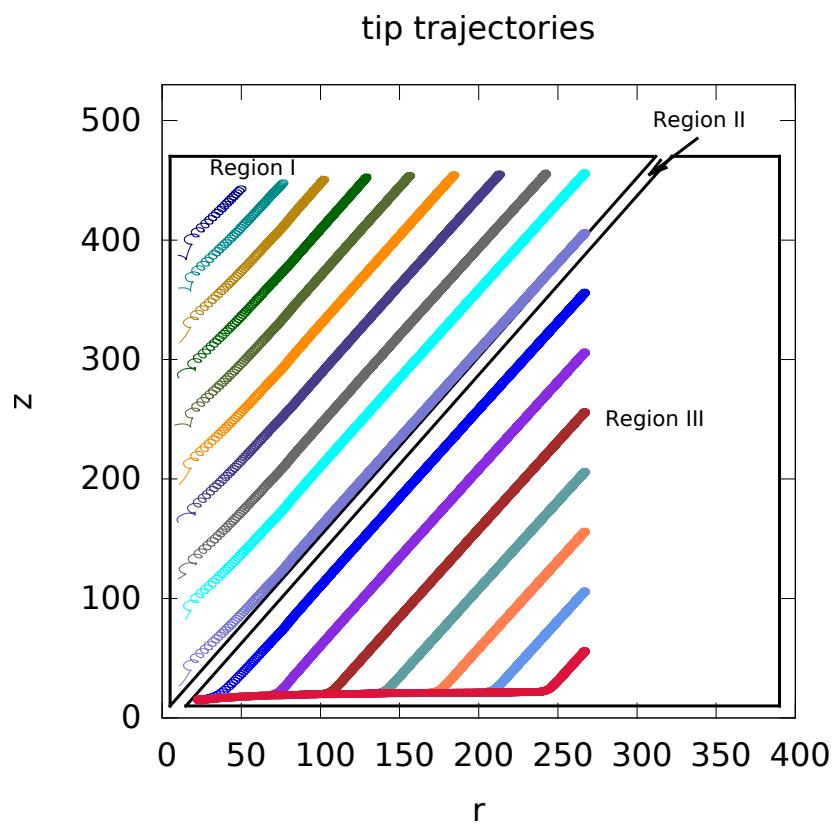


Figure 3.3: Tip trajectories in the Karma model when generating the spiral waves at different values of  $z$  and fixed  $r = 270$  as well as when generating the spiral waves at different values of  $r$  and fixed  $z = 450$ . Parameters from scenario 1.

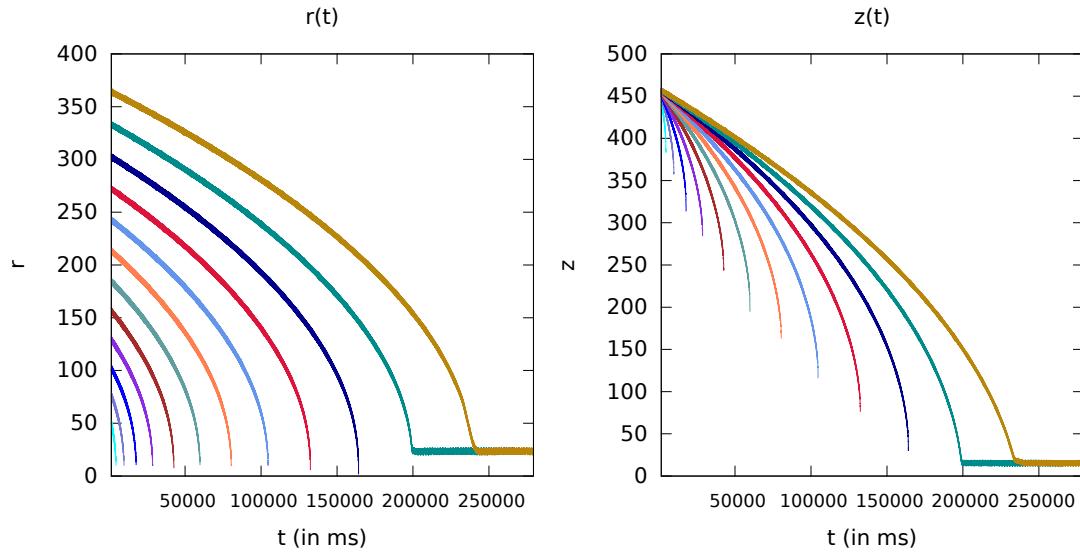


Figure 3.4: Evolution of  $r$  and  $z$  as time advances corresponding to solutions depicted in Figure 3.3. Tip trajectory initiated at fixed  $r = 270$  and varying  $z$ .

and the tip of the spiral wave gets trapped inside a region close to the boundaries  $z = 0$  and  $r = 0$ . The corresponding three dimensional scroll ring in this scenario, will be affected by both boundaries. As a consequence of this interaction, the spiral wave will reach a region where it gets trapped and remains there.

It is important to note that the behavior of all the solutions in every case we have studied is of a similar manner. That is, we have that for every case, there are three different regions (I,II and III) on which the first quadrant of the  $(r,z)$  plane can be split into.

### 3.2.1 Scroll ring dynamics away from the boundaries

In our studies, for a given spiral wave solution, the evolution of the tip of the spiral wave in the  $(r,z)$  plane traces a path as those depicted in Figure 3.3. As time advances, the behavior of  $r$  and  $z$  is shown in figures 3.4 and 3.5.

We are interested in finding some functions that determine the evolution of the tip trajectory. The goal of this is to predict and give information about the dynamics of the spiral wave solutions and, therefore, understand the dynamics of the respective three dimensional scroll ring solutions.

In order to reach our objectives we follow the ideas from Keener [Keener \(1988\)](#) and try to formulate our results as a function of the  $Re$  parameter and weak meandering. From the theory developed by Keener, for solutions of the scroll ring kind, we have

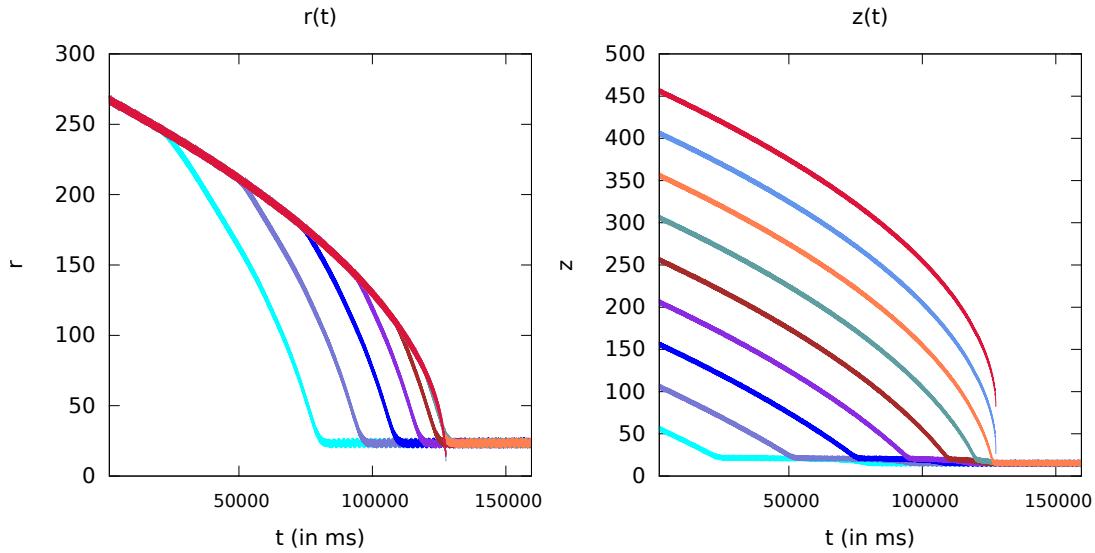


Figure 3.5: Evolution of  $r$  and  $z$  as time advances corresponding to solutions depicted in Figure 3.3. Tip trajectory initiated at  $z = 450$  and varying  $r$ .

that the change of both variables  $r$  and  $z$  is proportional to  $r$ , that is,

$$\begin{aligned}\frac{dr}{dt} &= \frac{-b_2}{r}, \\ \frac{dz}{dt} &= \frac{c_3}{r},\end{aligned}$$

or, equivalently

$$\begin{aligned}r(t) &= \sqrt{r_0^2 - 2b_2 t}, \\ z(t) &= z_0 - \frac{c_3 \sqrt{r_0^2 - 2b_2 t}}{b_2}.\end{aligned}\tag{3.10}$$

It is also worth noting that the  $z$  variable is a linear function of  $r$ ,

$$z(r) = z_0 - \frac{c_3 r}{b_2}.\tag{3.11}$$

For simplicity in notation, let's rename the constants in equation (3.10) as

$$\begin{aligned}c_1 &= 2b_2, \\ c_2 &= \frac{c_3}{b_2},\end{aligned}$$

so that  $r(t)$  in system (3.10) and equation (3.11) have the form

$$\begin{aligned}r(t) &= \sqrt{r_0^2 - c_1 t}, \\ z(r) &= z_0 - c_2 r.\end{aligned}\tag{3.12}$$

Under these considerations, once the starting point of the scroll wave solution is given  $(r_0, z_0)$ , we use system (3.12) to calculate the fit function for the  $r$  variable.

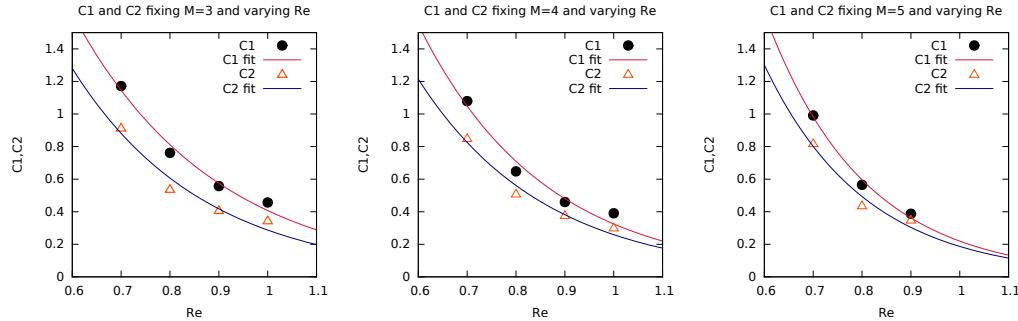


Figure 3.6: Constants  $C_1$  and  $C_2$  of the fitting functions for  $r(t)$  and  $z(r)$  for fixed  $M = 3$  and varying  $Re$ , fixed  $M = 4$  and varying  $Re$  and fixed  $M = 5$  and varying  $Re$  shown in the left, middle and right frame, respectively.

That is, we propose a function  $r(t)$  that has the form of a square root and look for constants  $c_1$  and  $c_2$  such that the system (3.12) holds. After that we calculate the fit function for  $z(r)$ .

To illustrate this procedure, for Scenario 1, we generate a spiral wave solution starting at  $(r_0, z_0) = (275, 420)$ . For this particular case, we obtained the following for  $r$  and  $z$ , respectively,

$$\begin{aligned} r(t) &= \sqrt{\frac{130079-t}{1.79475}}, \\ z(r) &= 83258.4 + 0.407768r. \end{aligned} \quad (3.13)$$

Table 3.3 contains a summary of the values for constants  $c_1$  and  $c_2$  for the respective scenarios. Figure 3.2.1 shows the behavior of  $c_1$  and  $c_2$  when we fix the value of  $M$  for three different values (3,4 and 5) and we vary the parameter  $Re$ . Notice the decay in the behavior of constants  $c_1$  and  $c_2$  as functions of parameters  $Re$  and  $M$ . We can approximate these decays by means of an exponential fitting functions. As a consequence, even for a small parameter region, it is possible to describe the behavior of constants  $c_1$  and  $c_2$ .

With this information, we can predict the dynamics of the spiral waves, and therefore, the scroll rings in three dimensions without the need to actually solve the system. This is, once we have chosen a value for parameters  $M$  and  $Re$ , and an initial condition for the spiral wave, we use constants  $c_1$  and  $c_2$  to say what the behavior of the respective solution will be.

### 3.2.2 Scroll ring dynamics close to the boundaries

It is important to remark that the procedure described in section 3.2.1 only works when our solutions are away from the boundaries. To have a better understanding of the solution's interaction with the  $r = 0$  and  $z = 0$  boundaries we study the evolution of  $r$  and  $z$  as functions of time. For each of the scenarios 1,4 and 5 we take a solution that belongs to region III. Both variables,  $r$  and  $z$ , for this particular solution will constitute of three parts. The first one corresponds to the case where

$M$		$Re$			
		0.7	0.8	0.9	1.0
3	$c_1$	1.17147	0.76124	0.55720	0.45665
	$c_2$	0.91432	0.53845	0.40775	0.34515
4	$c_1$	1.07933	0.64768	0.45944	0.39035
	$c_2$	0.85004	0.50912	0.37755	0.30116
5	$c_1$	0.99133	0.56501	0.38743	-
	$c_2$	0.81875	0.43730	0.34934	-

Table 3.3: Values of fitting constants  $c_1$  and  $c_2$  for every scenario fixing parameter  $M$  and varying parameter  $Re$ . The symbol – means that for  $M = 5$ , constants  $c_1$  and  $c_2$  are missing because for this scenario, break-up of the spiral wave occurs and this kind of solutions is not of interest in our studies.

the spiral wave is away from the boundaries which was discussed in Section 3.2.1. In the second one, the solution is affected by the  $z = 0$  boundary and the third one is when there is interaction of the solution with both boundaries  $r = 0$  and  $z = 0$ .

### Interaction with one boundary (small $z$ )

For each scenario we generate solutions that belong to region III and  $r$  is large. The purpose is to analyse the behavior of the solution when there is interaction with one boundary ( $z = 0$ ). To this end, we let the spiral wave evolve and eventually it will start interacting with the boundary  $z = 0$ . From that moment on, the spiral wave starts drifting in the direction parallel to the  $z = 0$  boundary, and is moving towards the  $r = 0$  boundary. Figure 3.7 depicts this behavior for the three distinct scenarios studied. By examining the variables  $r$  and  $z$  separately as functions of time, we notice that  $r$  still has the same behavior as that predicted by Keener, but the variable  $z$  no longer has it, that is, the variable  $z$  is not decaying proportional to  $r$ . Even further,  $z$  has reached part of the domain and no longer leaves it (Figure 3.8).

Observe that the behavior of the  $r$  variable can be fitted by Keener's formulation. However, the fitting constants are not in accordance with the ones for the situation when the spiral wave is away from the boundaries. In the case when there is interaction with the boundary, the decay is faster for the  $r$  variable than in the case when the solution is away from the boundaries. Additional to the intrinsic decay on the radius, when there is interaction of the spiral wave with the boundary  $z = 0$ , there is an increment on the velocity as shown by Yermakova and Pertsov [Ermakova and Pertsov \(1986\)](#), creating an additional drift making the evolution of the  $r$  variable faster, covering around 180 space units (almost half of the domain) in considerably less time (Figure 3.8).

### Interaction with both boundaries (small $r$ and $z$ )

We end this section by analyzing the interaction of the spiral wave solutions with both  $r = 0$  and  $z = 0$  boundaries. As presented in the earlier section, the result

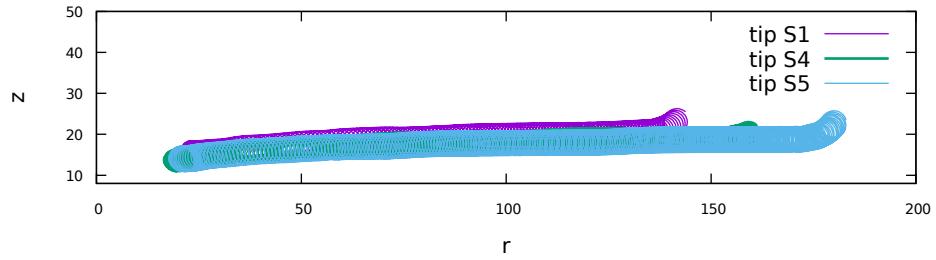


Figure 3.7: Tip trajectories for the distinct scenarios studied when the spiral wave interacts with the boundary  $z = 0$ .

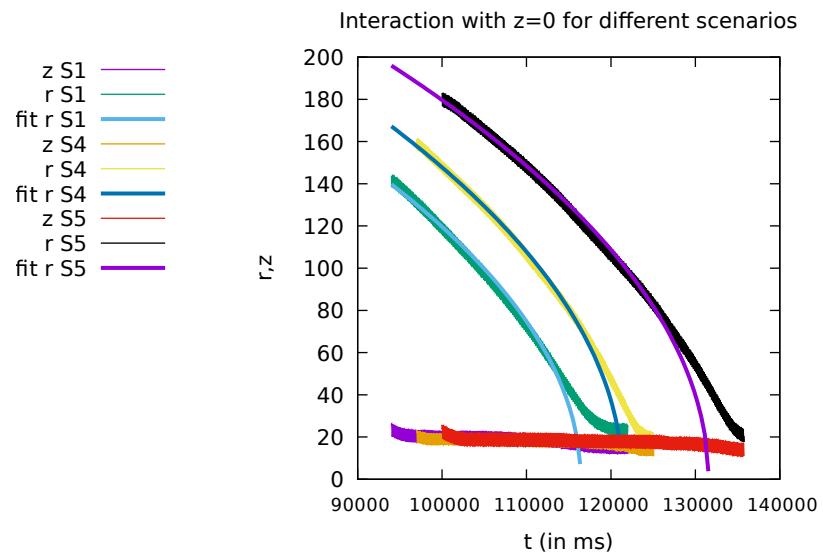


Figure 3.8: Evolution of  $r$  and  $z$  as time advances when there is interaction with the boundary  $z = 0$  for scenarios 1, 4 and 5.

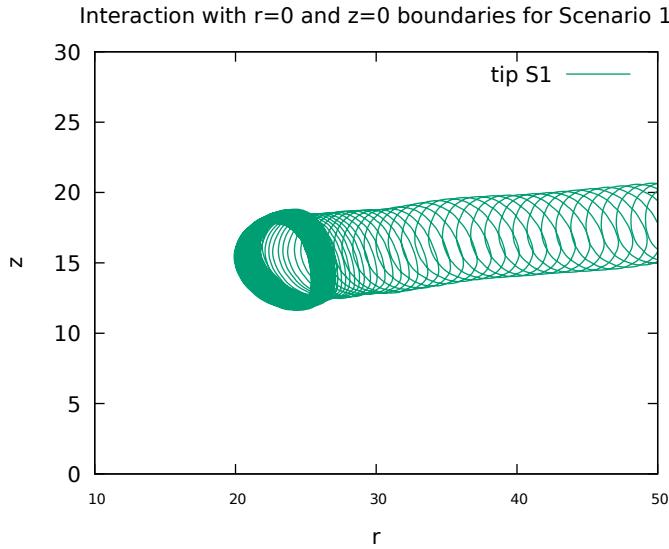


Figure 3.9: Tip trajectory for scenario 1 when interacting with both boundaries  $z = 0$  and  $r = 0$ .

of the interaction of the spiral wave solution with one boundary ( $z = 0$ ) results in drifting along the same  $z = 0$  boundary in the direction towards  $r = 0$ .

For scenario 1, when the solution has already reached a stabilized zone (see Figure 3.9), drifting along  $z = 0$  is no longer possible due to the presence of the  $r = 0$  boundary. In here, the effect of the  $r = 0$  boundary creates a gaining of curvature on the spiral wave solution, changing its path and getting it repelled from the same  $r = 0$  boundary. This phenomenon translates into drifting of the spiral wave along the  $z = 0$  but in the direction away from the  $r = 0$  boundary. Immediately after this, there is interaction with the  $z = 0$  boundary, thus, the spiral wave solution gains curvature again and starts drifting now towards the  $r = 0$  boundary. The left frame in Figure 3.10 shows the  $r$  and  $z$  variables as functions of time depicting this phenomenon. Observe that when  $r$  is maximum, that is, when the spiral wave solution is farthest from the  $r = 0$  boundary (left frame in Figure 3.10, label A), the variable  $r$  starts decreasing, meaning that the spiral wave is getting closer to the  $r = 0$  boundary until the solution gets repelled by the same  $r = 0$  boundary (left frame in Figure 3.10, label B). After some time,  $r$  attains the maximum again (left frame in Figure 3.10, label C).

At the same time, notice that the  $z$  variable is also having the same behavior than  $r$  but the different lies in the phase at which it does it. This is, the times for which  $r$  is maximum (or minimum) is not the same for  $z$ .

The phenomenon that occurs when there is interaction of the spiral wave solution with both boundaries exhibits two scales: a fast one and a slow one. The fast scale being the revolving of the spiral wave, whereas the slow scale is the movement of the core of the tip trajectory as a whole. This movement of the core is a drifting along the  $z = 0$  boundary towards the  $r = 0$  boundary and away from the  $r = 0$

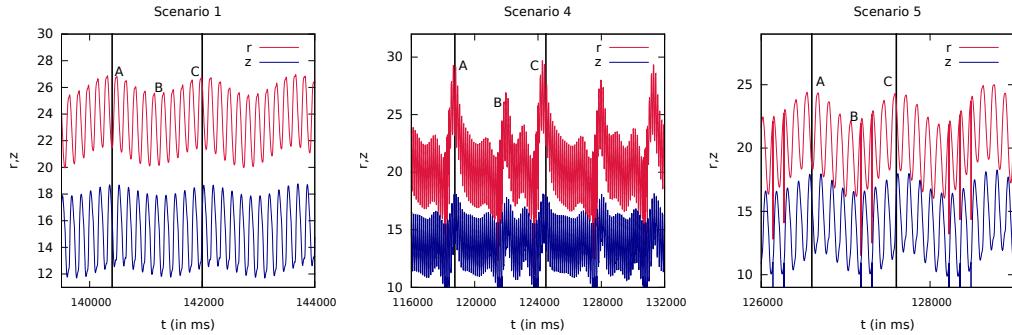


Figure 3.10: Evolution of  $r$  and  $z$  as time advances when there is interaction with both  $r = 0$  and  $z = 0$  boundaries for scenarios 1, 4 and 5 shown in the left, middle and right frame, respectively.

boundary.

We do not include scenarios 2 and 3 because they behave in a very similar manner to scenario 1.

For scenario 4 (see Figure 3.10, middle frame) we have also a similar behavior but the oscillations of the variables  $r$  and  $z$  are less regular than for the case of scenarios 1 and 5. This due to the fact that the spiral wave tip trace in two dimensional cartesian coordinates for this selection of parameters ( $M = 4$  and  $Re = 0.8$ ) exhibits a weak meandering. This meandering is responsible for the irregular oscillation of the respective scroll wave in cylindrical coordinates.

For scenario 5 (see Figure 3.10, right frame) we have, qualitatively, the same behavior as in that explained earlier for scenario 1. The difference lying in the phase and amplitude.

Notice that scenario 4 has the most irregular behavior in the oscillations for the variables  $r$  and  $z$ , followed by scenario 5 and finally scenario 1.

Recall that all our studies are for the case of positive filament tension parameter regime. It is well known in the literature that the solutions of the scroll ring kind eventually collapse. However, the dynamics that the solutions for scenarios 1,4 and 5 depicted in Figure 3.10 are due to the interaction with the  $r = 0$  and  $z = 0$  boundaries, neglecting completely the effect of the positive filament tension parameter regime. This kind of behavior has been reported also for negative filament tension [Azhand et al. \(2014\)](#), where the effect of the parameter regime gets neglected by the effect of the boundaries.

### 3.3 Discussion and Conclusions

We have investigated the dynamics of solutions of the spiral wave kind which corresponds to slices of three dimensional scroll rings for the Karma model (3.1) for a particular set of parameters (Scenarios 1-5). We were able to calculate some critical values of  $z$  along with some functions (Section 3.2.1) that have helped us to predict

the qualitative behaviour on the long time of the solutions in question. This is, once  $M$  and  $Re$  are chosen and an initial condition given we can predict what the behavior of the scroll ring will be, meaning that we are able to give information about the dynamics of the solutions, that is, if the corresponding scroll ring in three dimensions will shrink entirely (collapsing with the  $r = 0$  boundary); if it will disappear at the bottom ( $z = 0$ ) boundary, or if the scroll ring stabilizes for which case we can explicitly state this region.

As the solutions considered in this work are for very large integrating times they can be physiologically irrelevant. This because the heart cannot be either in an arrhythmic state nor in a fibrillatory state for long periods of time. However, understanding the mathematical behaviour of those solutions for the long times is of interest to have a complete knowledge of the solutions and their dynamics.

Although our implementation was very straightforward, we were able to obtain insightful information on the dynamics in three dimensions of scroll rings in the Karma model. Also, it is possible to modify the experiment to look for additional information without tuning up or modifying too much the programs. We are interested now in studying the interaction of the solutions with the boundaries with a different set of parameters where the solutions in two dimensions are of the spiral wave kind but the trajectory of the tip follows not only circular or meandering cores, but more complicated ones as depicted in Figure 3.1 for small  $Re$  and  $M$  values.

Of interest are also the solution's in three dimensions where we have negative filament tension. This due to the fact that some instabilities arise in three dimensions where the respective solutions in two dimensions are stable and well studied [Mikhailov et al. \(1985\)](#); [Pertsov et al. \(1993\)](#); [Alonso et al. \(2004\)](#); [Alonso and Panfilov \(2007\)](#); [Alonso et al. \(2013\)](#), making it impossible to generalize the behaviour from two to three dimensions. In order to study this type of solutions, the constant times  $\tau_E$  and  $\tau_n$  would have to be modified in the Karma model (Eq. (3.1)). We can repeat our experiments and study those solutions with these new parameters.

With the results obtained, we now have a better understanding of the solutions of the scroll ring type in the Karma model. The natural question now is to study solutions of a more general kind such as scroll waves, which appear in cardiac dynamics [Karma \(1993\)](#); [Winfree \(1994\)](#) and apply the knowledge that we have so far to predict the behaviour in this scenario. The aim is to be able to apply this results in even more complicated cases of three dimensional solutions where break up of scroll waves occur, which have been reported in the literature that they appear when patients suffer from cardiac arrhythmias and ventricular tachycardias and ventricular fibrillation.

With the experience gained by studying this simple model, we can move now to study more complicated ones and perform these same experiments or small variations of them and predict obtain good (qualitative) data from them.

## Chapter 4

# Numerical Solutions of Equations of Cardiac Wave Propagation based on Chebyshev Multidomain Pseudospectral Methods

The structure of this present chapter is developed as follows. First, we present the model equations that we will address (Section 4.1) in our study. In Section 4.2, we describe the methods we will be using for our comparison. Moving on, in section 4.3, we study the models equations for the one dimensional case. In section 4.4, we present a scaling study which will be helpful to compare different models of cardiac propagation in terms of the convergence of the numerical scheme. Section 4.5 deals with stability issues of the presented methods. Later on, we present advantages and disadvantages in the two dimensional case (Section 4.6) where we discuss solutions of the spiral wave kind. We continue with some estimates for three dimensions (Section 4.7) about computing time. Finally, we conclude with a section of discussion and conclusions (Section 4.8).

### 4.1 Model equations.

In order to get consistency in our results, we will perform experiments in two different models, namely the Karma model which is a two variable model that reproduces, qualitatively, generic restitution and dispersion properties of cardiac tissue [Karma \(1994\)](#), and the minimal ventricular (MV) model which contains the minimum number of variables necessary for an action potential (AP) that can reproduce arbitrary action potential duration and conduction velocity restitution curves as well as a range of realistic AP shapes [Bueno-Orovio et al. \(2008\)](#).

The Karma model studied is given by equations (3.1)-(3.5) described in Chapter 3. The differential equations for the MV model are as follows:

$$\begin{aligned}\partial_t u &= \nabla(\tilde{D}\nabla u) - (J_{fi} + J_{so} + J_{si}), \\ \partial_t v &= (1 - H(u - \theta_v))(v_\infty - v)/\tau_v^- - H(u - \theta_v)v/\tau_v^+, \\ \partial_t w &= (1 - H(u - \theta_w))(w_\infty - w)/\tau_w^- - H(u - \theta_w)w/\tau_w^+, \\ \partial_t s &= ((1 + \tanh(k_s(u - u_s))/2 - s)/\tau_s,\end{aligned}\tag{4.1}$$

---

CMD: Explicit Chebyshev Multidomain, SICMD: Semi-Implicit Chebyshev Multidomain, ICMD: Implicit Chebyshev Multidomain, OS: Operator Splitting.

where the variable  $u$  represents the transmembrane voltage and  $\tilde{D}$  is the diffusion matrix, which is equal the identity. The three currents are given by the following equations:

$$\begin{aligned} J_{fi} &= -vH(u - \theta_v)(u - \theta_v)(u_u - u)/\tau_{fi}, \\ J_{so} &= (u - u_0)(1 - H(u - \theta_w))/\tau_o + H(u - \theta_w)/\tau_{so}, \\ J_{si} &= -H(u - \theta_w)ws/\tau_{si}, \end{aligned} \quad (4.2)$$

and  $H(u)$  is the standard Heaviside function. Most of the time constants are functions of the voltage variable  $u$  and are defined as follows:

$$\begin{aligned} \tau_v^- &= (1 - H(u - \theta_v^-))\tau_{v1}^- + H(u - \theta_v^-)/\tau_{v2}^-, \\ \tau_w^- &= \tau_{w1}^- + (\tau_{w2}^- - \tau_{w1}^-)(1 + \tanh(k_w^-(u - u_w^-)))/2, \\ \tau_{so} &= \tau_{so1} + (\tau_{so2} - \tau_{so1})(1 + \tanh(k_{so}(u - u_{so})))/2, \\ \tau_s &= (1 - H(u - \theta_w))\tau_{s1} + H(u - \theta_w)\tau_{s2}, \\ \tau_o &= (1 - H(u - \theta_o))\tau_{o1} + H(u - \theta_o)\tau_{o2}, \end{aligned} \quad (4.3)$$

and the infinity values are defined as

$$\begin{aligned} v_\infty &= \begin{cases} 1, & u < \theta_v^- \\ 0, & u \geq \theta_v^- \end{cases}, \\ w_\infty &= (1 - H(u - \theta_o))(1 - u/\tau_{w_\infty}) + H(u - \theta_o)w_\infty^*. \end{aligned} \quad (4.4)$$

A more detailed description of this model can be found in Bueno-Orovio et al. (2008).

## 4.2 Numerical Methods

The numerical methods to be used in this work are as follows. The first method is the standard finite difference scheme (FD) where space is discretized with the usual three point centered finite difference and time is integrated with the Euler method. For the second method (CMD) space is discretized using multidomain pseudospectral derivative and explicit integration in time with Euler. For the third method (SICMD) space is discretized as in (CMD), and the non linear term is treated explicitly. The fourth method (ICMD) considers both, the reactive and diffusive parts as implicit. Finally, we use an Operator Splitting method (OS) Qu and Garfinkel (1999) where we use a pseudospectral approach to discretize space and for the time we use explicit integration.

All the implementations of the methods have been optimized by tabulating precomputed lookup tables for computationally expensive functions (such as the exponential and hyperbolic tangent) of one variable. For the  $LU$  implementations, we have used the CSR (Compressed Sparse Row) format Saad (2003) to store the matrices  $L$  and  $U$  as vectors, thus saving memory. The calculations were done on a computer with an AMD opteron (6282SE, 2.6 GHZ) processor.

---

A version of this present chapter has been submitted for publication in the journal Mathematics and Computers in Simulation.

### 4.2.1 Implementation of the CMD method.

Thus, with the application of the pseudospectral method based on equation (2.32) for the Karma model (Eq. 3.1), in two dimensions we obtain the following

$$\begin{aligned}\frac{dE_{ij}}{\Delta t} &= A \sum_{k=0}^N D_{ik}^{(2)} E_{kj} + B \sum_{k=0}^N u_{ik} D_{kj}^{(2)\top} + \frac{1}{\tau_E} f(E_{ij}, n_{ij}), \\ \frac{dn_{ij}}{\Delta t} &= \frac{1}{\tau_n} g(E_{ij}, n_{ij}),\end{aligned}\quad (4.5)$$

where  $E_{ij}(t) \approx E(x_i, y_j, t)$ ,  $n_{ij}(t) \approx n(x_i, y_j, t)$ . In equation (4.5),  $A = 4\gamma/(x_R - x_L)^2$  and  $B = 4\gamma/(y_R - y_L)^2$ , and they appear as a consequence of the linear transformations  $[x_L, x_R]$  and  $[y_L, y_R]$  to  $[-1, 1]$  and include the respective diffusion coefficients. No-flux boundary conditions are implemented by solving the system

$$\begin{aligned}\sum_{k=0}^N D_{0k}^{(1)} E_{kj} &= 0, \\ \sum_{k=0}^N D_{Nk}^{(1)} E_{kj} &= 0,\end{aligned}\quad (4.6)$$

for  $E_{0j}$  and  $E_{Nj}$ ,  $j = 1, N - 1$  and similar conditions hold for  $E_{i0}$  and  $E_{iN}$ ,  $i = 1, N - 1$ .

In order to apply the Chebyshev pseudospectral method, we employ a multidomain approach used previously in Olmos and Shizgal (2006). It consists of dividing the intervals  $[x_L, x_R]$  and  $[y_L, y_R]$  into  $N_i$  overlapping subintervals,  $I_\mu = [x_0^\mu, x_{N_c-1}^\mu]$  and  $I_\nu = [y_0^\nu, y_{N_c-1}^\nu]$ , respectively, and  $\mu$  and  $\nu = 1, N_i$ . In each dimension all the subintervals have the same length. For each subinterval, we apply the procedure described in equations (2.25)-(2.30) with the resulting system of ODE's given by equation (4.5) with  $A = \frac{4D_x}{(x_{N_c-1}^\mu - x_0^\mu)^2}$  and  $B = \frac{4D_y}{(y_{N_c-1}^\nu - y_0^\nu)^2}$  and the indices in equations (4.5) and (4.6) going from 0 to  $(N_c - 2)N_i + 1$ .  $D_x$  and  $D_y$  are the diffusion coefficients. The first and second derivative matrices  $D^{(1)}$  and  $D^{(2)}$  in equations (4.6) and (4.5), respectively, for the Chebyshev multidomain(CMD) method, are block diagonal matrices as shown in Olmos and Shizgal (2006).

The application of the Chebyshev multidomain in the solution of equation (4.1) requires a choice of two parameters, the number of subdomain  $N_i$  and the number of Chebyshev points per subdomain  $N_c$ , chosen sufficiently large so as to achieve numerical convergence. For Chebyshev multidomain, we can increase both  $N_c$  and  $N_i$  or fixing one while increasing the other.

### 4.2.2 Implementation of the SICMD method.

The spirit of the SICMD method to solve equation (1.16) with constant  $D$ , is to consider the diffusion term as implicit, whereas the reactive term as explicit. Following the formulation in Keener and Bogar (1998) we arrive to the formulation for the one dimensional case and scalar  $V$ ,

$$V^{n+1} - \Delta t \hat{D}^2 V^{n+1} = (I - \Delta t \hat{D}^2) V^{n+1} = V^n + \Delta t F(V^n) \quad (4.7)$$

where  $V^n = (V_2^n, V_2^n, \dots, V_{N_{px}-1}^n)^\top$  with  $V_j^n \approx V(x_j, t^n)$  and  $\hat{D}^{(2)}$  is the second derivative operator of size  $N_{px} - 2 \times N_{px} - 2$  as given in Olmos and Shizgal (2006)

with the boundary conditions given in equation (4.6).  $V_1^n$  and  $V_{N_{px}}^n$  are given by equation (4.6).

The implementation of the SICMD method reduces to solve a linear system of the form  $Ax = b$  at each time step. This system has a very particular form which arises from the structure of the matrices to approximate the second derivative operator by means of the Chebyshev pseudospectral method [Olmos and Shizgal \(2006\)](#). The resulting matrix  $A$  is a blockwise diagonal matrix. An  $LU$  decomposition is applied taking the advantage that both  $L$  and  $U$  “preserve” the structure of  $A$ , that is,  $L$  has the following form

$$\left( \begin{array}{cccccc} l_{2,2}^1 & & & & & \\ l_{3,2}^1 & l_{3,3}^1 & & & & \\ \vdots & \vdots & \ddots & & & \\ l_{N_{cx}-1,2}^1 & l_{N_{cx}-1,3}^1 & \dots & l_{N_{cx}-1,N_{cx}-1}^1 & & \\ & l_{2,1}^2 & & l_{2,2}^2 & & \\ & \vdots & & \vdots & \ddots & \\ l_{N_{cx}-1,1}^2 & l_{N_{cx}-1,2}^2 & \dots & l_{N_{cx}-1,N_{cx}-1}^2 & & \\ & & & & l_{2,1}^{N_{ix}} & \\ & & & & \vdots & \ddots \\ & & & & l_{N_{cx}-1,1}^{N_{ix}} & \dots & l_{N_{cx}-1,N_{cx}-1}^{N_{ix}} \end{array} \right)$$

The matrix  $U$  has a similar form but it is upper triangular. Under these considerations, solving a system  $Ax = b$  is equivalent to solve  $LUX = b$ , which is easily solved by means of a forward and backward substitution. Finally, the solution for the two and three dimensional cases requires the additional implementation of an ADI method [LeVeque \(2007\)](#).

#### 4.2.3 Implementation of the implicit method.

The implicit version used in this work, is based on the method developed in [Olmos-Liceaga and Segundo-Caballero \(2016\)](#). In order to solve equation (1.16), we consider both the reaction and the diffusion parts as implicit, i.e.

$$V^{n+1} - \Delta t \hat{D}^2 V^{n+1} = (I - \Delta t \hat{D}^2) V^{n+1} = V^n + \Delta t F(V^{n+1})$$

However, the reaction part is approximated by the first two terms of its Taylor approximation, around  $V^n$ , which gives

$$(I - \Delta t \hat{D}^2) V^{n+1} = V^n + \Delta t (F(V^n) + (V^{n+1} - V^n) F_V(V^n)) \quad (4.8)$$

or

$$(I - \Delta t \hat{D}^2 - \Delta t I \otimes F_V(V^n)) V^{n+1} = V^n + \Delta t (F(V^n) - V^n F_V(V^n)) \quad (4.9)$$

where  $I \otimes F_V(V^n)$  denotes the  $N_{px} - 2 \times N_{px} - 2$  diagonal matrix whose elements are given by  $F_V(V^n)$ .

#### 4.2.4 Implementation of the OS method.

The operator splitting method (OS), will be used for comparisons in two and three dimensions. We follow the same approach as in Qu and Garfinkel (1999). For our implementation, we use Euler finite difference scheme to solve the reaction part, whereas the diffusion part was solved implicitly with Euler.

### 4.3 One dimensional numerical convergence analysis.

In order to know the performance of each of the methods presented in this work, an error analysis is considered. The equations are solved for an integration time  $T = 250$ , and for each approximated solution  $U_a$ , we measure the error by the equation

$$e = \max_{\{i=2, N_{px}\} \{j=1, J\}} \|U_e(x_i, t_j) - U_a(x_i, t_j)\|_\infty, \quad (4.10)$$

where  $U_e$ , denotes the exact solution and  $\|\cdot\|_\infty$  denotes the usual  $l_\infty$  norm. However, in this case, the exact solution is not known. Thus, we take as our exact solution for the Karma model the one obtained with CMD with a configuration of  $N_{ix} = 667$  and  $N_{cx} = 8$ . This solution was taken because the error between two of the solutions obtained with CMD, one with  $N_{ix} = 500$  and  $N_{cx} = 8$  and the other with  $N_{ix} = 667$  and  $N_{cx} = 8$  gives an error of the order of  $1 \times 10^{-11}$ . For the MV model, we took as our converged solution the one obtained with CMD with 4000 points ( $N_{ix} = 1999$  and  $N_{cx} = 4$ ). Therefore, we assume the solution to be our exact solution and we will study how fast the solutions obtained with the different methods get closed to it.

In order to find the error given in Eq. (4.10), we use relations (2.25) and (2.27). For each subdomain of the converged solution, we compute the corresponding  $a_k$  values (using (2.27)). Then, we use equation (2.25) to calculate  $f(x_j)$ , where  $x_j$  is the domain where the approximated solution is defined.

#### 4.3.1 The Karma model.

In this section we study the Karma model (3.1) where the Heaviside function in equation (3.2) is replaced by the sigmoid function

$$f(E) = \frac{1}{2}(1 + \tanh(10(E - E_n))). \quad (4.11)$$

Therefore, (3.2) takes the form

$$\begin{aligned} f(E, n) &= -E + [E^* - D(n)]h(E), \\ g(E, n) &= R(n)\frac{1}{2}(1 + \tanh(10(E - E_n))) - [1 - \frac{1}{2}(1 + \tanh(10(E - E_n)))]n. \end{aligned}$$

The initial condition used to obtain our results is given by

$$\begin{aligned} E(x, 0) &= \frac{1}{[1+e^{4(x-5)}]^2} - \frac{1}{[1+e^{4(x+10)}]^2}, \\ N(x, 0) &= 0.5. \end{aligned} \quad (4.12)$$

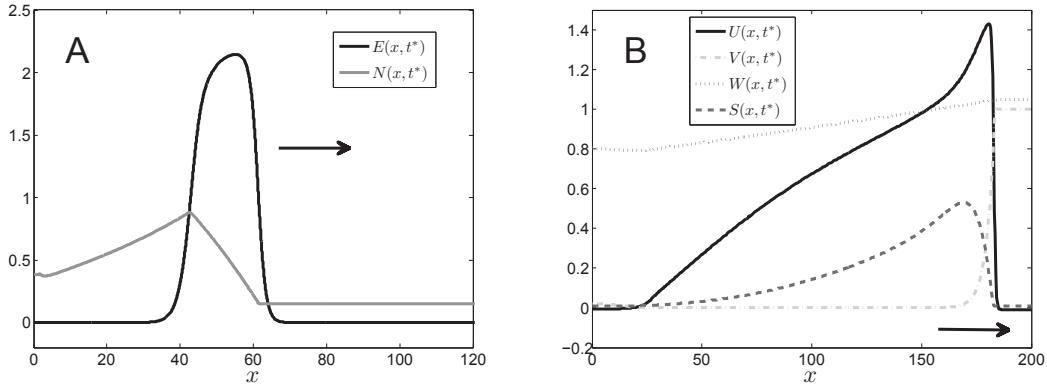


Figure 4.1: A) Numerical solution of the Karma model (Eq. 3.1). Parameters used were as in table (4.1). Solution shown at integration time  $t^* = 240$ . The front is located around  $x = 65$  space units. B) Numerical solution of equation (4.1) on a domain of  $x \in [0, 200]$  with initial condition given by relation (4.13). Model parameters taken as in table 1 in Bueno-Orovio et al. (2008) with  $u_{so} = 0.55$  and  $u_s = 1.15$ . Time for which the plot is shown is  $t^* = 69.3$ . The front is located around  $x = 185$  space units.

Under these considerations we obtain a propagating pulse which travels from left to right and a profile for  $T = 240$  time units is shown in Figure 4.1A.

In the present section we consider a convergence analysis for the pseudospectral approach and finite differences for equation (3.1). For the pseudospectral methods there are three parameters that can be used to optimize the performance of the method,  $N_{px}$ ,  $N_{cx}$  and  $N_{ix}$ . In order to achieve a better understanding of the performance of the pseudospectral methods, we present two convergence analysis, where in each scenario we fix the value of  $N_{cx} = 4$  and 8.

In tables 4.3 and 4.4, a convergence analysis for the CMD method increasing  $N_{px}$  with  $N_{cx} = 4$  and  $N_{cx} = 8$ , respectively, is shown. From the tables it is clear that as we increase the number of points, a more accurate solution is obtained, with the requirement that smaller time steps are taken into account. It follows that taking  $N_{cx} = 8$  provides a convergence to the exact solution faster than taking  $N_{cx} = 4$ .

Observe that when using CMD with  $N_{ix} = 74$  and  $N_{cx} = 4$  and  $N_{ix} = 25$  and  $N_{cx} = 8$ , for an approximate of  $N_{px} \approx 150$  points, and  $\Delta t = 1 \times 10^{-3}$ , the errors obtained are of the order of  $1 \times 10^{-2}$  (tables 4.3 and 4.4). The conclusion then is that even though the method with  $N_{cx} = 8$  provides a faster convergence, is for  $N_{cx} = 4$  when we obtain a better solution if our aim is to get solutions within an error of  $1 \times 10^{-2}$ . The solution with CMD and  $N_{cx} = 4$  is almost twice as fast as the solution with CMD and  $N_{cx} = 8$ . This is due to the amount of calculations required to obtain an approximated solution with this scheme.

Now we discuss the performance of FD (Table 4.5), SICMD (Table 4.6) and ICMD (Table 4.7). By gathering the information shown on Tables 4.5-4.6 we infer that SICMD method converges faster than FD. Simulations with SICMD were done with  $N_{cx} = 8, 16$  and 32 and the results were very similar than those obtained on Table

[4.6](#), where  $N_{cx} = 4$ . On the other hand, the results obtained with ICMD, were similar to those obtained with SICMD. So far, there is a disadvantage for ICMD, compared to SICMD as the linear system to solve, changes in each iteration. The possible advantage of the time step for ICMD is discussed in section [4.5](#).

It is important to note that for  $N_{cx} = 8$  and  $N_{ix} = 4000$  and  $\Delta t = 1 \times 10^{-5}$ , the error was of the order of  $1 \times 10^{-13}$ . This error was the minimum obtained even if we increased the value of  $N_{cx}$  to 16 or 32 and  $\Delta t$  was fixed. However, a larger choice of  $N_{cx}$  implies a smaller choice of  $\Delta t$ , and thus we stayed with  $N_{cx} = 4$ .

As a conclusion, SICMD provides a better solution than the other methods even though the CMD method performs better in terms of precision. Also, we consider worth it to implement the SICMD method as the computing time to generate approximated solutions is reduced by a large margin compared to CMD and FD and we obtain very precise solutions with SICMD.

### 4.3.2 The MV model.

We proceed now to present the results obtained when solving Eq. [\(4.1\)](#). We used the model parameters as in Table 1 in [Bueno-Orovio et al. \(2008\)](#) with the exception of  $u_{so}$  and  $u_s$ , whose values we choose to be 0.55 and 1.15, respectively. The initial condition used is

$$\begin{aligned} U(x, 0) &= \frac{1}{[1+e^{4(x-10)}]^2} - \frac{1}{[1+e^{4(x-5)}]^2}, \\ V(x, 0) &= 1, \\ W(x, 0) &= 1, \\ S(x, 0) &= 0, \end{aligned} \tag{4.13}$$

which gives solutions as the one depicted in Figure [4.1B](#).

As in the previous model, we present a convergence analysis for each of the numerical methods. We begin with an analysis of the CMD method. In this case we only took  $N_{cx} = 4$ . Slightly better results were obtained with  $N_{cx} = 8$ , where the minimum error was of the order of  $1 \times 10^{-4}$ . Based on the similarity, we decided to show only the case of  $N_{cx} = 4$  (Table [4.8](#)). Tables [4.9](#) and [4.10](#), show the analysis for the FD and the SICMD methods. By comparing these tables, we see that with SICMD is possible to obtain an approximated solution with half the number of points that attains the same precision as that of FD (2000 points and 4000 points, respectively). However, it is noted that convergence of the methods for the MV model is much slower than for the previous model.

This is due to the width of the narrow band where the excitation takes place. In other words, the solution of equation [\(4.1\)](#) decays much faster than the solution of equation [\(3.1\)](#). To show this, we calculated how fast is the decay in both models. In Figure [4.2](#) we show the exponential decay of the solutions by showing a semi-log graph for voltage versus position, near the location of the front. The slope of the line for the Karma model is around  $m_K = -84$  and for the MV model we have that  $m_{MV} = -417$  approximately. This means, that the front in the Karma model behaves as an exponential decay of approximately  $u_K \approx e^{-84x}$  whereas for the MV model the front behaves as  $u_{VM} \approx e^{-417x}$ . It is important to note that the domains

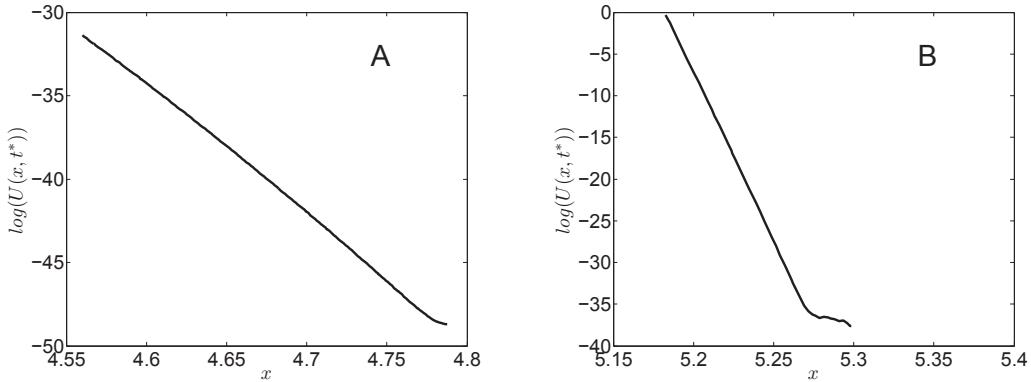


Figure 4.2: Exponential behavior of the solution of equations (3.1) and (4.1) at the location of the fronts. (a) Karma model. Domain  $x \in [0, 120]$ . Slope  $m_K = -84$  approximately; (b) MV model. Domain  $x \in [0, 200]$ . Slope  $m_{MV} = -417$  approximately.

of the models for which we solve the models are not the same (for the Karma model  $x \in [0, 120]$  and for the MV model we have  $x \in [0, 200]$ ). Then, to compare them, we need to rescale one of the domains into the other. We do this by following the procedure as in [Olmos and Shizgal \(2006\)](#) to map the domain  $[0, 200]$  into  $[0, 120]$ . After this, we obtain that the front of the MV model behaves as  $u_{MV} \approx e^{-541x}$ .

#### 4.4 Numerical precision as a function of the size of the interval and the stiffness of the equation

In the previous section, we presented a precision analysis as a function of the number of discretization points for two models in cardiac dynamics. Our goal is now to establish a criterium such that for a given model in cardiac electrophysiology based on the formulation of Hodgkin and Huxley [Hodgkin and Huxley \(1952\)](#) and a particular physical domain, it can be proposed a discretization of the domain, as well as in the time step such that we can obtain a desired accuracy.

This idea follows from the fact that equations for cardiac wave propagation have similar behavior. The major source of the different scales, is due to the time scale of the fast inward sodium current [Hodgkin and Huxley \(1952\)](#); [Beeler and Reuter \(1977\)](#). This process occurs in a much faster time scale than the rest of the processes involved in the formation of action potentials [Hodgkin and Huxley \(1952\)](#); [Beeler and Reuter \(1977\)](#). The fast entrance of the inward sodium current provokes a fast change in what is called the voltage variable (variables  $E$  and  $U$  for the Karma and the MV models, respectively). This time scale is the one that imposes the fast changes in time in the model. By considering the complete model, the scale given by the coefficient of the diffusive process, the fast time scale and the size of the physical domain, dictate the different scales in space.

In general, for a model of cardiac wave propagation, the formulation has the following

form

$$\frac{\partial u}{\partial t} = K \frac{\partial^2 u}{\partial x^2} + \rho f(u) \quad (4.14)$$

where  $u$  is the voltage variable and  $\rho$  is the inverse of the time constant for the sodium fast inward current times the maximum value of  $f$  over the corresponding domain. Based on this information, and the fact that simulations take place over a finite domain  $[a, b]$ , it is possible to scale equation (4.14) as done in [Olmos and Shizgal \(2006\)](#), such that the problem is equivalent to

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2} + \hat{f}(u) \quad (4.15)$$

over the domain  $[\sqrt{\frac{\rho}{K}}a, \sqrt{\frac{\rho}{K}}b]$  with  $z = \sqrt{\frac{\rho}{K}}x$  and  $\tau = \rho t$  and where  $\hat{f}$  is just  $f$  times a scaling factor such that  $\hat{f}$  has its maximum value equals 1. Such scaling becomes very important if we want to compare difficulty of solving different type of cardiac model equations. As an example, for equation (3.1)  $\frac{1}{\tau_E} = 0.4$ , and for equation (4.1), and according to [Bueno-Orovio et al. \(2008\)](#)  $\frac{1}{\tau_{fi}} = 9.09$ . In order to compute  $\rho$  for each case, we can find theoretically the maximum of  $f$  for each model or simply give an accurate estimate by solving the corresponding ODE system and finding the maximum in absolute value of the derivative for the variables that model voltage. It follows that  $\rho = 0.27$  and  $3.2$  for the Karma and the MV models, respectively.

At first glance, it would be easy to claim that because the value of  $\rho$  for model (4.1), is larger, then it would imply that such model needs more discretization points in order to get a given precision for both models. As it will be the case, nonetheless we still need to rescale space. For the Karma model in this work,  $K = 1.171$ ,  $a = 0$  and  $b = 120$ , giving a rescaled domain of  $z \in [0, 57.62]$ . In the case of the MV model,  $K = 1$ ,  $a = 0$  and  $b = 200$ , which implies a scaled domain of  $z \in [0, 357.77]$ .

From the changes of scale, we can see that the Karma model has the form of equation (4.15) over an interval  $z \in [0, 57.62]$ , whereas the MV model can be changed to the form of equation (4.15) over the interval  $z \in [0, 357.77]$ . In this case, the fast dynamics in space due to  $\rho$  and  $K$ , has been converted into a problem about the size of the numerical domain. Observe that  $f$  for the Karma and the MV model are not the same. Nonetheless, what really matters is the time constant  $\rho$ , because  $f$  only plays the role of the transition between the resting and the excited states [FitzHugh \(1961\)](#).

If we solve equation (4.15), over two domains of different length  $L_1 \ll L_2$  and the same initial condition, clearly, with the same number of discretization points, the solution with the interval with length  $L_1$  will be much more precise than the numerical solution obtained for the interval of length  $L_2$ . The reason is basically that for the domain with length  $L_2$  there are much less points than the domain with length  $L_1$ , that will capture the fast changes of the solution.

What follows now is to connect both problems. If particular numerical precision is needed for a complex model, can we use information of the simple model to predict the number of discretization points and time step? The answer is yes, and can be

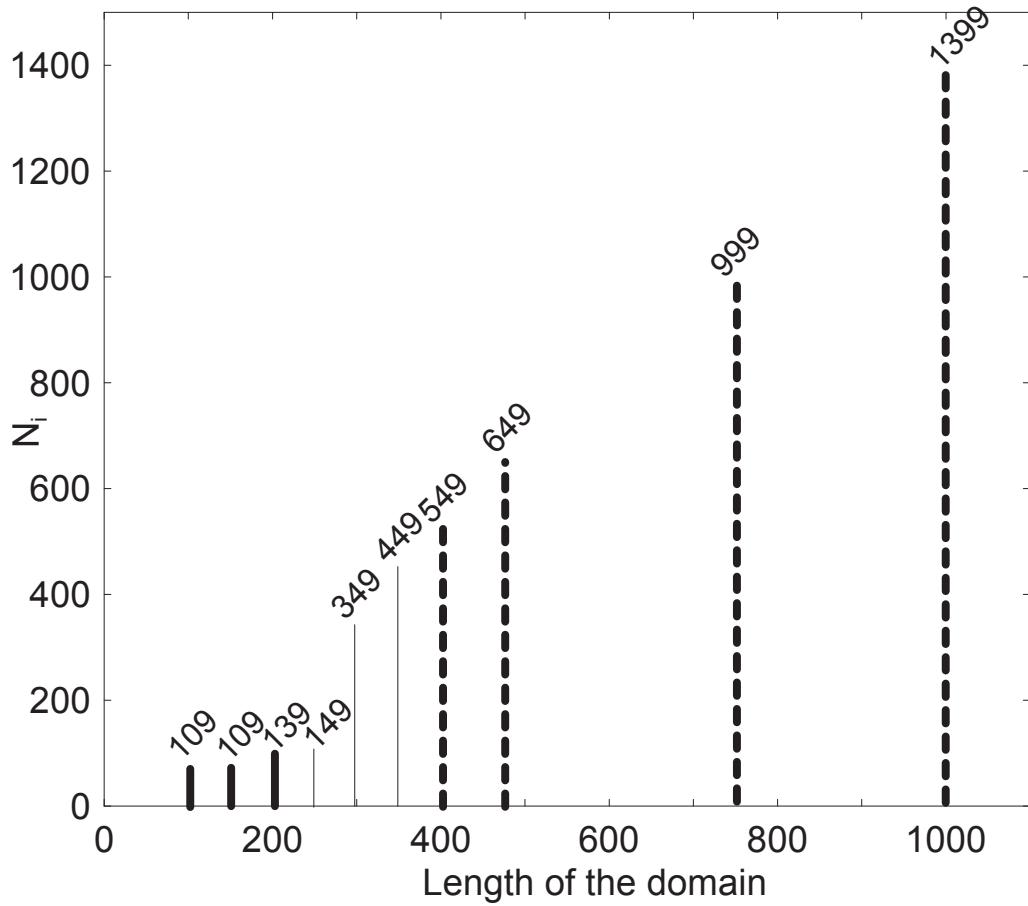


Figure 4.3: Number of subintervals  $N_i$  (with  $N_c = 4$ ) in order to get precision of  $O(10^{-2})$  as a function of the length of the domain, for the Karma model (Eq. 3.1). Solid Bold ( $\Delta t = 1 \times 10(-2)$ ), Solid ( $\Delta t = 1 \times 10(-3)$ ), Dashed Bold ( $\Delta t = 1 \times 10(-4)$ )

done easily by the proper change of variables and having information about the numerical precision (number of discretization points and time step) for the simple model.

We begin with a graph that shows the number of subintervals  $N_i$  (with  $N_c = 4$ ) needed to attain an error of  $10^{-2}$  for a given length of the domain, for the Karma model (Eq. 4.1). With this information, if we ask for the MV model (Eq. 4.1) over the domain  $[0, 200]$  and total integration time of  $T = 100$ , to have an error of  $10^{-2}$ , we transform our problem into the scaled model (Eq. 4.15) and then, change it to the parameter space provided by the Karma model (Eq. 3.1). This implies to solve the Karma model over the domain  $x \in [0, 745]$  with total integration time  $T = 1185$ .

From Figure 4.3, clearly, it is needed to take  $N_i = 999$ ,  $N_c = 4$  and  $\Delta t = 0.0001$  in order to obtain a solution with error of  $10^{-2}$ . This result is corroborated from Table 4.10.

To conclude, we have proposed an easy mechanism to assure certain level of confidence when solving a model of cardiac wave propagation based on the Hodgkin-Huxley mechanism. What is only requires is to do the analysis on a simple two variable model, like the Karma model (Eq. 3.1) or a simple Fitzhugh-Nagumo generic model [FitzHugh \(1961\); Keener and Sneyd \(2010\)](#), and then transform the complex model to the simple dynamics.

## 4.5 A comparison of the stability region for the CMD, SICMD and ICMD methods

We end the one dimensional case by analyzing the stability regions for the SICMD, ICMD and CMD methods. In order to analyze the stability regions, we consider the scalar linear problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \rho u \quad (4.16)$$

over a finite domain  $[a, b]$ , with Neumann boundary conditions equal to zero, i.e.

$$\left. \frac{\partial u}{\partial x} \right|_{x=a} = 0 \quad \text{and} \quad \left. \frac{\partial u}{\partial x} \right|_{x=b} = 0$$

When applying the SICMD, ICMD and CMD methods to solve equation (4.16) with boundary conditions, we arrive to the difference equation

$$U^{n+1} = (1 - \Delta t)[I - \Delta t \hat{D}^{(2)}]^{-1} U^n \quad \text{and} \quad n = 0, 1, 2, \dots \quad (4.17)$$

for the SICMD method. For the ICMD method, the corresponding equation is

$$U^{n+1} = [I - \Delta t \hat{D}^{(2)} + \rho \Delta t I]^{-1} U^n \quad \text{and} \quad n = 0, 1, 2, \dots \quad (4.18)$$

whereas for the CMD method we obtain

$$U^{n+1} = [(1 - \rho \Delta t)I + \Delta t \hat{D}^{(2)}]U^n \quad \text{and} \quad n = 0, 1, 2, \dots \quad (4.19)$$

where  $U^n = (U_2^n, U_2^n, \dots, U_{N_{px}-1}^n)^\top$  with  $U_j^n \approx u(x_j, t^n)$ .  $U_1^n$  and  $U_{N_{px}}^n$  are given by equation (4.6). Clearly, the stability of the schemes depends on that the largest eigenvalue of each of the matrices on the right hand side, lies inside the unit circle. For the SICMD method, consider the matrix

$$A = I - \Delta t \hat{D}^{(2)}. \quad (4.20)$$

Now, in order to find the minimum eigenvalue in norm  $\psi$  for  $A$ , we consider  $Av = \psi v$ . Then,

$$(I - \Delta t \hat{D}^{(2)})v = v - \Delta t \hat{D}^{(2)}v = \psi v$$

or

$$\hat{D}^{(2)}v = \left( \frac{1 - \psi}{\Delta t} \right) v. \quad (4.21)$$

As  $\hat{D}^{(2)}$  is a second derivative operator with no-flux boundary conditions, it has its minimum eigenvalue zero.

From equation (4.21) the corresponding norm of  $\psi$  due to  $\lambda$  is 1. Numerically, it was observed that the rest of the eigenvalues of  $\hat{D}^{(2)}$ , are real, negative and distinct, in concordance with Gottlieb and Lustman (1983). This would imply that the rest of the eigenvalues of matrix  $A$  are real, positive and with norm larger than 1.

Therefore, the smallest eigenvalue for  $A$  must be  $\psi = 1$ . It follows that the maximum eigenvalue in norm for  $A^{-1}$  is one. From equation (4.17), the stability of the method is practically given by the factor  $(1 - \rho\Delta t)$ . If  $\Delta t$  is less than  $\frac{2}{\rho}$ , the method is stable. Otherwise is unstable.

For the ICMD method, we follow the same steps than for the SICMD method and implies that the largest eigenvalue of  $[I - \Delta t\hat{D}^{(2)} + \rho\Delta tI]^{-1}$  is 1. Therefore, for any  $\Delta t$  value, we obtain a stable solution.

For the CMD method, there is no bound for the maximum eigenvalue of the matrix defined in equation (4.19). Clearly, if  $\xi$  is the largest eigenvalue of  $\hat{D}^{(2)}$  with eigenvector  $v$ , it follows that

$$[(1 - \rho\Delta t)I + \Delta t\hat{D}^{(2)}]v = v - \rho\Delta tv + \Delta t\hat{D}^{(2)}v = (1 - \rho\Delta t + \Delta t\xi)v$$

and if  $\xi = a + ib$ , then  $|\lambda| = \sqrt{(1 - \rho\Delta t + \Delta ta)^2 + (\Delta tb)^2}$ . This implies that the behavior of the largest eigenvalue of  $(1 - \rho\Delta t)I + \Delta t\hat{D}^{(2)}$ , follows the behavior of the largest eigenvalue of  $\hat{D}^{(2)}$ .

In Figure 4.4, it is shown the maximum value of  $\Delta t$  required to obtain a stable solution with CMD, for different values of  $N_i$  and  $N_c = 4$ .

From the information obtained in this section it is clear that ICMD performs better, followed by SICMD and CMD. Stability, however, does not imply a better solution. For the case of SICMD a stable solution is obtained with  $\Delta t < \frac{2}{0.27} \approx 7.4$ , whereas for ICMD any value of  $\Delta t$  is valid. However, with SICMD a meaningful physical solution was obtained with values up to  $\Delta t = 3$ , whereas for ICMD the maximum value for that purpose was  $\Delta t = 1.9$  (Larger values implied no pulse propagation). With this information and considering that SIMCD requires much less operations, from now on we will not consider ICMD for higher dimensions.

## 4.6 Two dimensional studies

Because the equations that we are solving are blunt approximations of what really happens, having extremely high precisions in our solutions is not that pressing or rigorously relevant. However, it is of key importance to know the reliability of the methods to replicate the phenomena that the model equations describe.

In this section we are interested in (i) reducing the number of operations that each method needs to solve the equations up to a time  $T$ ; (ii) find a suitable configuration for the pseudospectral based methods, such that the precision of the obtained solution with a maximized  $\Delta t$  and minimized  $N_p$  is comparable to that obtained

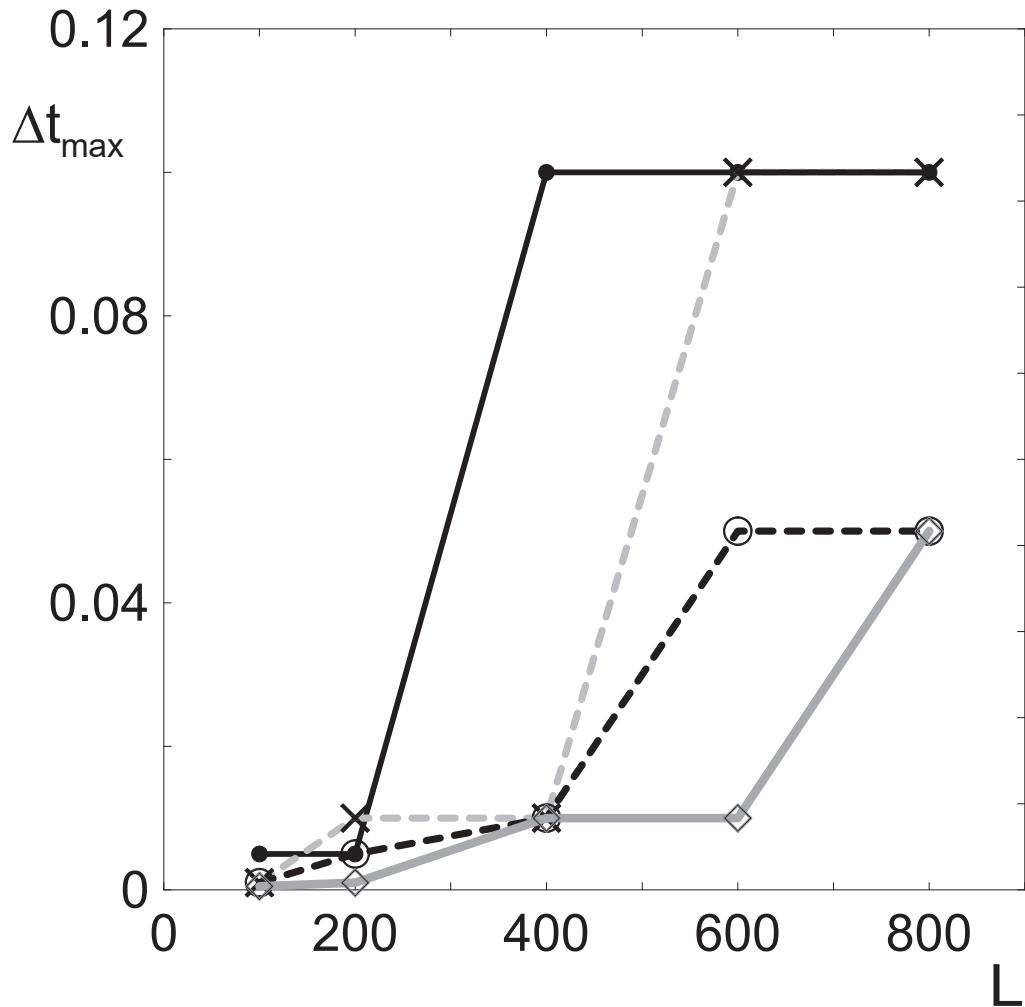


Figure 4.4: Maximum  $\Delta t$  in order to obtain a stable solution for the Karma model using the CMD method for different values of length of the domain  $L$ , fixed  $N_{cx} = 4$ ,  $N_{ix} = 299$  (Solid black),  $N_{ix} = 499$  (dashed gray),  $N_{ix} = 699$  (dashed black),  $N_{ix} = 899$  (Solid gray).

with FD, and (iii) compare the performance in the evaluation of solutions typical of cardiac dynamics.

To achieve this, we focus on four numerical methods: FD, CMD, SICMD and OS. OS has proven to be useful for solving equations of the Reaction-Diffusion type for cardiac wave equations ([Olmos and Shizgal \(2009\)](#), [Qu and Garfinkel \(1999\)](#)), with the pseudospectral approach to discretize space.

#### 4.6.1 The Karma Model

We begin our studies by solving equation (3.1) over a squared domain  $x \in [0, 120]$ ,  $y \in [0, 120]$ . When solving cardiac models it is of great importance to study solutions of auto sustainable waves that rotate in the media. These are known as spiral waves and we focus on solutions of this type. In order to generate such waves we take the initial condition  $E(x, y, 0) = E(x, 0)$  as in equation (4.12) and  $N(x, y, 0) = 0.5$ . We let evolve the generated pulse for a certain time,  $t = t^*$ , then we set

$$\begin{aligned} E(x_i, y_i, t^*) &= 0, \\ N(x_i, y_i, t^*) &= 0, \end{aligned} \tag{4.22}$$

for  $y_i$ , where  $i \geq \frac{N_{py}}{2}$ . Simulations were taken with 150, 300 and 600 points for each dimension for FD. In the case of CMD, SICMD and OS, we took 152 ( $N_{ix} = 25$ ,  $N_{cx} = 8$ ), 302 ( $N_{ix} = 50$ ,  $N_{cx} = 8$ ), 602 ( $N_{ix} = 100$ ,  $N_{cx} = 8$ ) per dimension.

In Figure 4.5 (Left) we show solutions of equation (3.1) ( $E(x, y, t^*) = 1$ ) for  $t^* = 1700$  using FD with the three different configurations along with the solution obtained with CMD with 602 points per dimension as, based on the one dimensional analysis, provides an accurate solution to approximate the exact one.

It is important to remark here that simulations with CMD method taking configurations of 152 ( $N_{ix} = 25$ ,  $N_{cx} = 8$ ), 302 ( $N_{ix} = 50$ ,  $N_{cx} = 8$ ), 602 ( $N_{ix} = 100$ ,  $N_{cx} = 8$ ) per dimension, respectively, are practically indistinguishable, that is, solutions with CMD converge faster than FD.

The solutions obtained with the SICMD method behave in a similar manner than the CMD method. This is, they are practically indistinguishable between themselves when configurations of more than 150 points per dimension are considered. For the OS method, although it is capable of producing numerical solutions with relatively large time steps, in terms of convergence, we obtained a similar behavior to that of FD.

From the results obtained, it can be concluded that the methods based on Chebyshev polynomials are better than the finite difference approach as clearly the obtained solutions with CMD and SICMD with 150 points ( $N_i = 25$ ,  $N_c = 8$ ) are closer to the converged solution compared to the solution obtained with FD and 600 points. However, the question we want to answer is which of the developed methods provides better results for the two dimensional solutions based on the number of points for the spatial discretization and the time step in time.

Moreover, to obtain the solution with FD with 600 points per dimension we have used a time step of  $\Delta t = 1 \times 10^{-3}$  whereas for the solution with CMD with 150

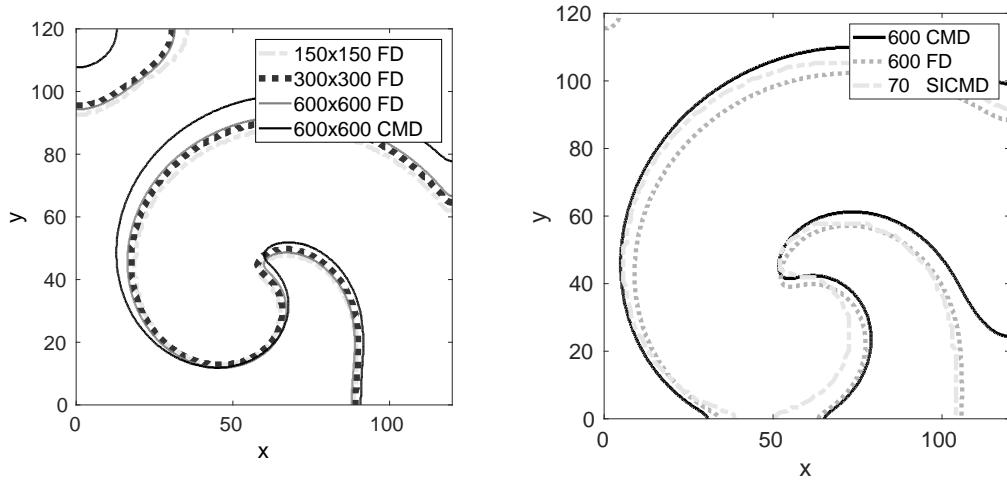


Figure 4.5: (Left) Solution of equation (3.1) in two dimensions using FD compared with CMD ( $N_{ix} = 667$ ,  $N_{cx} = 8$ ,  $N_{iy} = 667$  and  $N_{cy} = 8$  giving a total of 602 points per dimension) for time  $t^* = 1700$ . Numerical parameters:  $\tau_E = 2.5$ ,  $\tau_n = 200$ ,  $\gamma = 1.171$ ,  $E_h = 1.75$ ,  $E_n = 1.0$ ,  $E^* = 1.5414$ ,  $Re = 0.9$  and  $M = 4$ . The domain used is  $x \in [0, 120]$  and  $y \in [0, 120]$ . The surface level of the contour plots is  $E(x, y, t^*) = 1$ . (Right) Comparing solutions with FD (600 points per dimension), CMD ( $N_{ix} = 100$ ,  $N_{cx} = 8$ ,  $N_{iy} = 100$ ,  $N_{cy} = 8$  for a total of 602 points per dimension) and SICMD ( $N_{ix} = 35$ ,  $N_{cx} = 4$ ,  $N_{iy} = 35$ ,  $N_{cy} = 4$  for a total of 70 points per dimension). Time for which plots are shown is  $t = 1200$ . The surface level of the contour plots is  $E(x, y, t) = 1$ .

points per dimension we need to take a time step of  $\Delta t = 1 \times 10^{-2}$ . This is an order of magnitude larger, which helps also to speed up the the computations. For the SICMD method with 150 points per dimension, we took  $\Delta t = 0.1$ . This is three orders of magnitude larger than the one with FD with 600 points per dimension. This implies that computations in two dimensions are much faster with SICMD than with the rest of the methods.

Now we ask the converse question. If we consider a solution with FD with 600 points per dimension, how many points does it take to have a solution using the SICMD method with a precision comparable to that obtained with FD? In Figure 4.5 (Right), we present three approximated solutions, the one obtained with FD (600 points per dimension), the one obtained with CMD ( $N_{cx} = 8$  and  $N_{ix} = 667$ ) and the one obtained with SICMD ( $N_{ix} = 35$  and  $N_{cx} = 4$ ), and we observe that the one with SICMD is closer to the one with CMD than the FD one. Then, to obtain a solution with a precision like the one with FD with 600 points per dimension ( $\Delta t = 1 \times 10^{-3}$ ), SICMD only requires 70 points per dimension ( $\Delta t = 0.1$ ), which is considerably less points and time steps and therefore reduces the computation time dramatically.

It is worth to mention that stable and meaningful solutions could be obtained with  $\Delta t = 3.5$  and  $N_{ix} = N_{iy} = 74$  and  $N_{cx} = N_{cy} = 4$  for the SICMD method. For this  $\Delta t$  solutions of the spiral wave kind were obtained with acceptable physical meaning. Based on this comment, the solutions obtained with SICMD are four orders of magnitude faster than FD.

From the count of operations (See Appendix) it follows that FD takes less steps than any of the pseudospectral methods. Nonetheless, this is only the number of operations per time step. Clearly, from the results in the previous section and in Section 4.5, it is expected that at least the SICMD method will allow larger time steps and then, reverting the order of the total number of operations, giving a much smaller computing time. This issue is explored in detail for the three dimensional case in Section 4.7.

#### 4.6.2 Some physiological quantities of interest

In Figure 4.6 we show the tip trajectories for the Karma model using the parameter set described in Table 4.1 with the SICMD method for two different configurations of points and we compared them with the FD case. Using SICMD with 600 points per dimension is ten times faster than FD with 500 points per dimension, this again, as it is possible to take a time step for SICMD that is one order of magnitude larger than the FD case.

The methodology used to calculate the tip trajectory is the same as discussed in Fenton et al. (2002). This is, we identify spiral wave tips as points with zero normal velocity at an arbitrarily chosen isopotential line that defines the boundary between the depolarization and the repolarization wave back.

In Figure 4.7 we show some tip trajectories using the same set of parameters but varying  $Re$ . This is in concordance with what is reported in the literature.

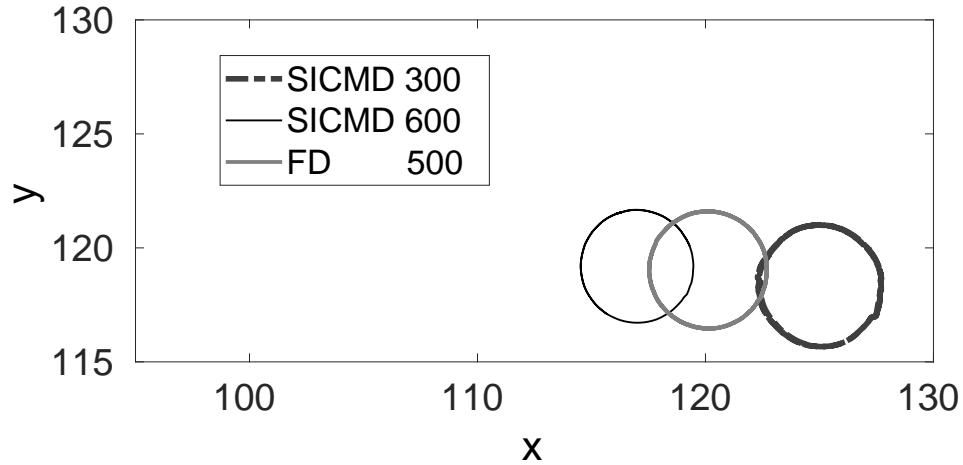


Figure 4.6: Tip trajectories in the Karma model for parameter set provided by Table 4.1. Comparison using the SICMD for different configurations of points and FD.

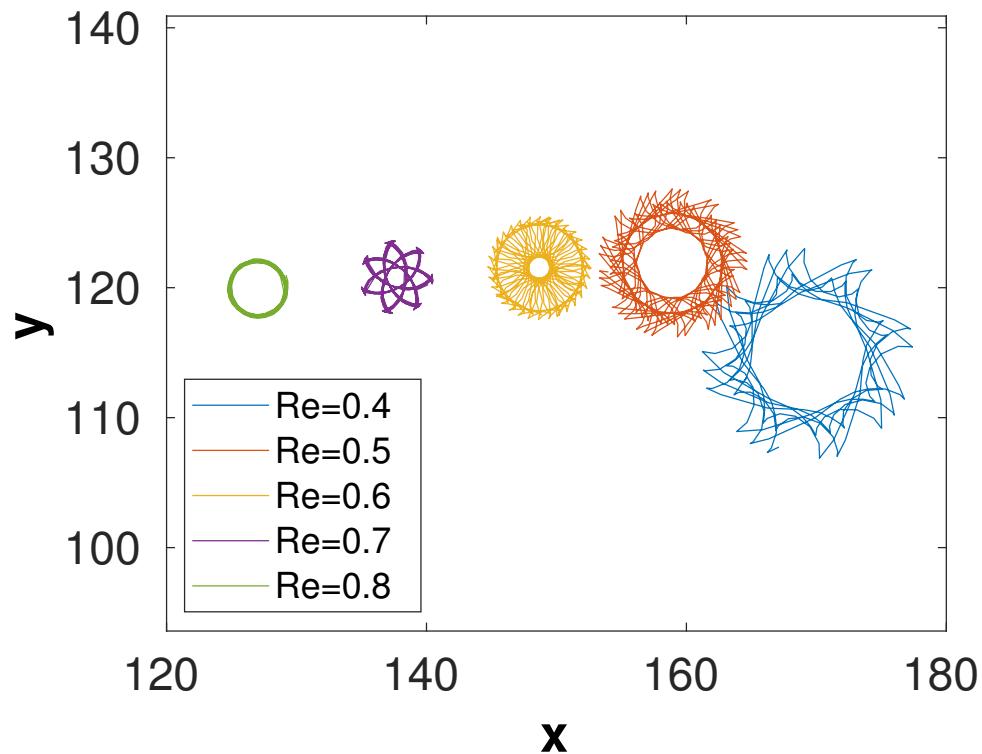


Figure 4.7: Tip trajectories in the Karma model for parameter set provided by Table 4.1 and varying  $Re$  using the SICMD method.

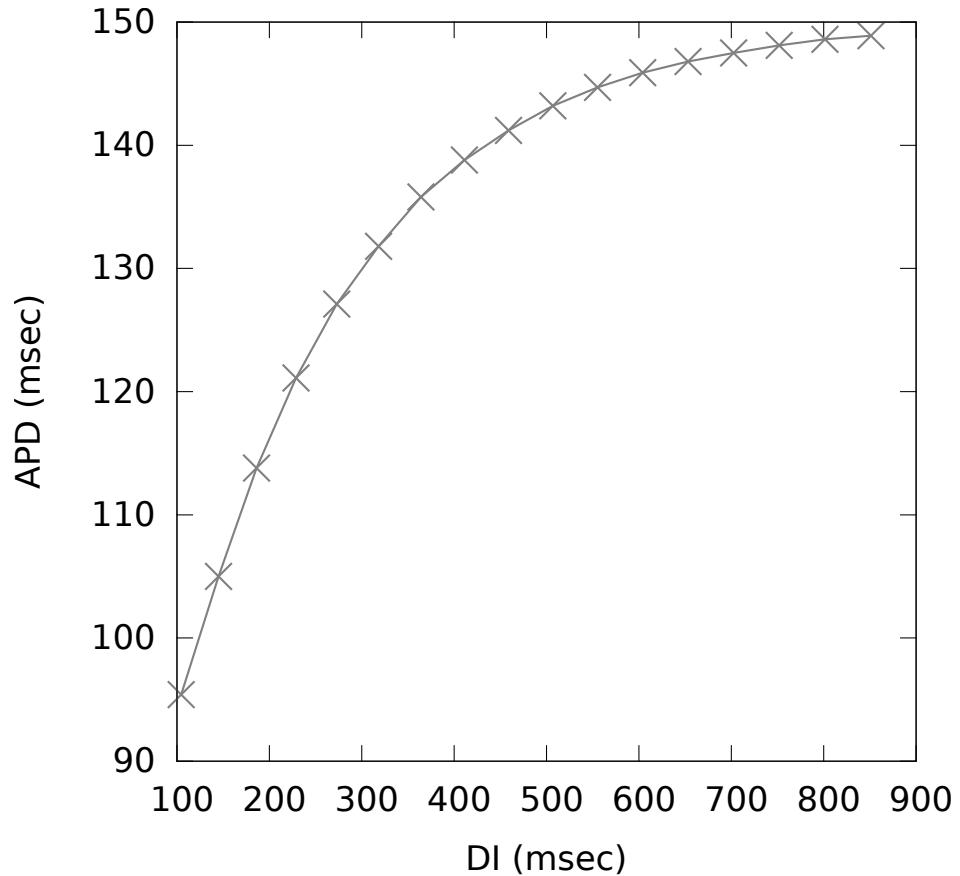


Figure 4.8: Action potential duration restitution curve in the Karma model, Eq. (3.1), for parameter set shown in Table 4.1 and initial conditions given as explained in Section 4.6.1.

Also, we used the SICMD method and calculated the APD restitution curve for the equation (3.1), using the methodology explained in Nolasco and Dahlen (1968). In Figure 4.8, we show this curve. We started by pacing the cable with an initial basic cycle length of 1000 ms, and once we reached a steady state, we calculated both action potential duration (APD) and diastolic interval (DI). Then repeated the experiment reducing the basic cycle length. Clearly, this curve is similar, qualitatively, to the work previously reported by Karma Karma (1994), and Fenton et. al Fenton et al. (2002).

Therefore, we conclude this present section by stating that the pseudospectral methods can be used as an improved alternative over the current used methods for studying the behavior of the tip of a spiral wave and to study restitution curves.

### 4.6.3 The MV model

Now we move on to present an analogous study but for the MV model, (Eq. 4.1). When using FD we utilized 150, 200, 300 and 500 points per dimension. For the CMD, SICMD and OS methods we used 152 ( $N_{ix} = 25$ ,  $N_{cx} = 8$ ), 200 ( $N_{ix} = 33$ ,  $N_{cx} = 8$ ), 302 ( $N_{ix} = 67$ ,  $N_{cx} = 8$ ), and 500 ( $N_{ix} = 83$ ,  $N_{cx} = 8$ ) points per dimension.

The domain of study is  $x \in [0, 200]$ ,  $y \in [0, 200]$  and we have integrated for a total time of  $T = 1000$  time units. We have used the model parameters as in Table 1 in ? with the exception of  $\tau_{fi}$ ,  $u_{so}$  and  $u_s$ , whose values we choose to be 0.15, 0.55 and 1.15, respectively.

The time step used in each of the methods were: 0.1 for FD with 150 and 200 points,  $5 \times 10^{-2}$  for FD with 300 points and  $1 \times 10^{-2}$  for FD with 500 points per dimension. For CMD with 152, 200 and 302 points we used a time step of  $1 \times 10^{-2}$ , whereas for the configuration with 500 points we used a time step of  $5 \times 10^{-3}$ . For the solutions generated with SICMD we used  $\Delta t = 0.1$  for the configurations with 152, 200 and 302 points. Finally for SICMD with 500 points we used  $\Delta t = 1 \times 10^{-2}$ .

The initial condition used is given by the functions in Eq. (4.13). To generate solutions of the spiral wave kind we proceeded as follows. After generating the initial condition, we let the front evolve  $t = 200$ , and after that we set  $u(x_i, y_j, t) = 1$  for  $i = 1, \dots, \frac{3}{4}N_{px}$  and  $j = 1, \dots, \frac{1}{2}N_{py}$ . We choose the approximated solution generated with CMD with 500 points per dimension as our best converged solution and we compare it to all the other solutions obtained with FD, SICMD and OS. A comparison of these methods with the converged solution is shown in Figure 4.9

Convergence for all the methods is shown to be much slower than for the Karma model (Eq. (3.1)) in which with a configuration of 150 points per dimension the solutions were close to the converged solutions whereas for the MV model (Eq. (4.1)) the configurations with 150 points per dimension still are far from the converged solutions. From Figure 4.9, it follows that if we consider simulations with 200 points per dimension, then the methods perform from best to worst in the following order: CMD, SICMD, FD and OS. Computationally speaking, the fastest of the schemes for the simulations with 200 points per dimension was FD, followed by OS which took 50% more time than FD; then SICMD which took twice the time than FD. Finally, CMD was 10 times slower than FD. It is important to remark that FD does a very good work regarding computing time. However, the solution is far from being close to the converged solution. From Figure 4.9 we can infer that with the pseudospectral methods CMD and SICMD we obtain more reliable solutions for this configuration of points (200 per dimension), as those generated with FD are still far from the converged solution that we are taking as our base, namely, the one generated with CMD with 500 points per dimension. If we increase the number of points to 300, we obtain that now the the methods perform from best to worst in the following order: CMD, SICMD, FD and OS. Another important observation here is that FD and SICMD are comparable in the number of operations that each scheme performs to obtain a solution (Table 4.2 with  $N_{ix} = 50$ ,  $N_{cx} = 8$ ,  $N_{px} = 302$ ,  $N_{iy} = 50$ ,  $N_{cy} = 8$  and  $N_{py} = 302$  for SICMD and  $N_{px} = 300$ ,  $N_{py} = 300$  for FD). It would mean that

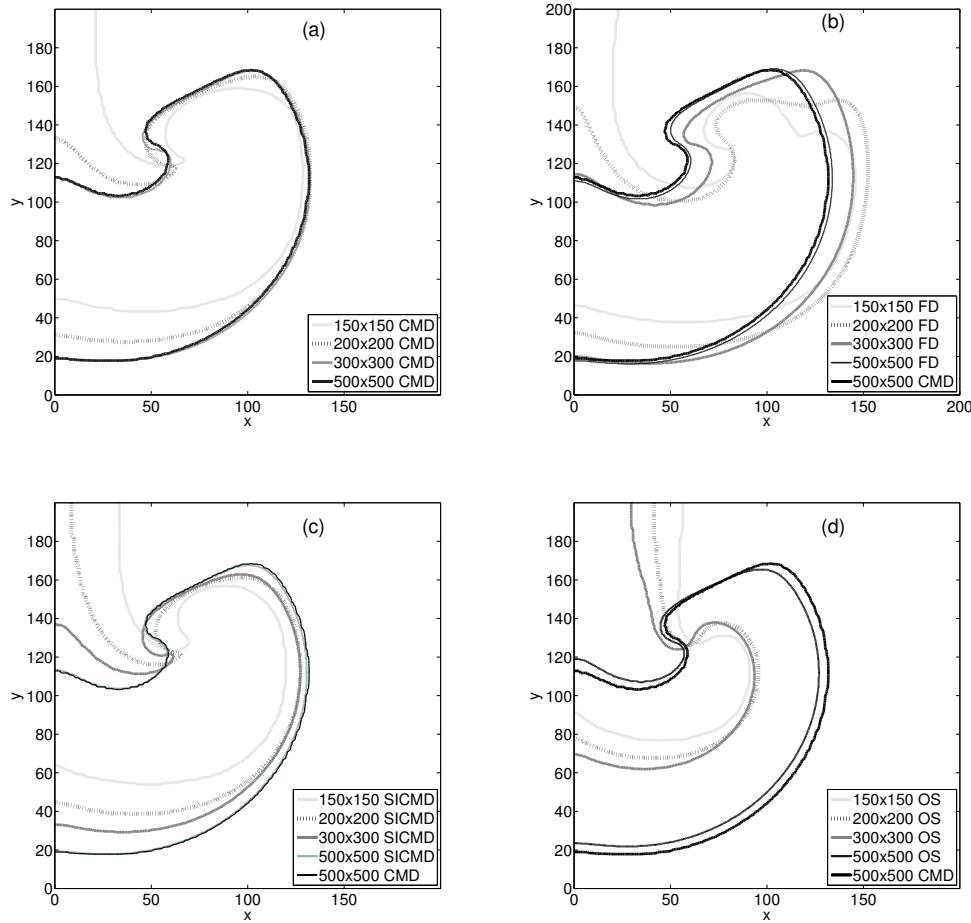


Figure 4.9: Solution of equation (4.1) in two dimensions using (a) FD for  $t^* = 900$ ; (b) CMD for  $t^* = 900$ ; (c) SICMD for  $t^* = 900$ ; (d) OS for  $t^* = 900$ . For every method, a comparison was made with CMD with 500 points per dimension ( $N_{ix} = 83, N_{cx} = 8, N_{iy} = 83$  and  $N_{cy} = 8$ ). The domain used is  $x \in [0, 200]$  and  $y \in [0, 200]$ . The surface level of the contour plots is  $U(x, y, t^*) = 1$ .

FD provides a good solution for this model. Nevertheless, before any conclusion, we still have to discuss the three dimensional case.

To summarize, the results were consistent with what we obtained for the Karma model. The SICMD method generates precise solutions and it is faster than CMD in terms of computing time. The OS method, while faster than both CMD and SICMD to obtain an approximated solution, it deforms the solutions as we tune up the parameters to achieve that speed in terms of the computing time.

## 4.7 Three dimensional studies.

Moving on to three dimensions, we have solved the Karma model (Eq. (3.1)) on a domain of  $[0, 120] \times [0, 120] \times [0, 48]$  for  $T = 1000$  using FD with a configuration of 600 points in the  $x$  and  $y$  dimensions and 240 points in  $z$  and SICMD with a configuration of  $N_{ix} = 35, N_{cx} = 4, N_{iy} = 35, N_{cy} = 4, N_{iz} = 11$  and  $N_{cz} = 4$  for a total of 72 points in the  $x$  and  $y$  dimensions and 24 points in  $z$ . We chose this configurations due to the obtained experience when studying the two dimensional case. We have seen that with the respective configurations in the  $x$  and  $y$  dimensions, the precision of both solutions was already comparable. The initial condition considered for our simulations was given by  $E(x, y, z, 0) = E(x, 0)$  given in equation (4.12) and  $n(x, y, z, 0) = 0.5$ . The generated front evolves, and for the time  $t^* = 120$ , we set

$$\begin{aligned} E(x_i, y_j, z_k, t^*) &= 0, \\ N(x_i, y_j, z_k, t^*) &= 0, \end{aligned} \tag{4.23}$$

where  $1 \leq i \leq N_{px}$ ,  $1 \leq j \leq \frac{N_{py}}{2}$  and  $1 \leq k \leq N_{pz}$ , in order to generate a scroll wave. The time steps considered were  $1 \times 10^{-3}$  for the FD scheme and 0.1 for SICMD. With these we already gain a speed up, computationally speaking, as the time step needed for the SICMD is three orders of magnitude than that of FD. Also, as we are considering considerably less points for the SICMD scheme, we have that this method requires less operations to generate a solution. This translates into generating an approximated solution faster. For our particular case, SICMD required only about 0.006% of the time that FD took to generate a solution. In Figure 4.10 (left column), we show solutions of equation 3.1 using FD (Top row) and SICMD (Bottom row) for  $t = 1000$ .

We have solved equation (4.1) on a three dimensional domain of  $[0, 200] \times [0, 200] \times [0, 100]$  for a total time  $T = 1000$ . We used FD with 300 points in  $x$  and  $y$  dimensions and 75 points in the  $z$  dimension and SICMD with a configuration of  $N_{ix} = 50, N_{cx} = 8, N_{iy} = 50, N_{cy} = 8, N_{iz} = 12$  and  $N_{cz} = 8$  for a total of 302 points in the  $x$  and  $y$  dimensions and 74 points in  $z$ . In figure 4.10 (right column) we present a numerical solution for  $t = 1000$  with FD (Top row) and SICMD (Bottom row), respectively. The initial condition used for our simulations is given by the functions (4.13).

Moving on to the MV model (Eq. 4.1), to generate solutions of the scroll wave kind, after we generated our initial condition, we let the front evolve and then when  $t = 200$  we set  $u(x_i, y_j, z_k, t) = 1$  for  $i = 1, \dots, \frac{3}{4}N_{px}$ ,  $j = 1, \dots, \frac{1}{2}N_{py}$  and  $k = 1, \dots, N_{pz}$ . The time step used on the FD scheme was  $\Delta t = 1 \times 10^{-3}$

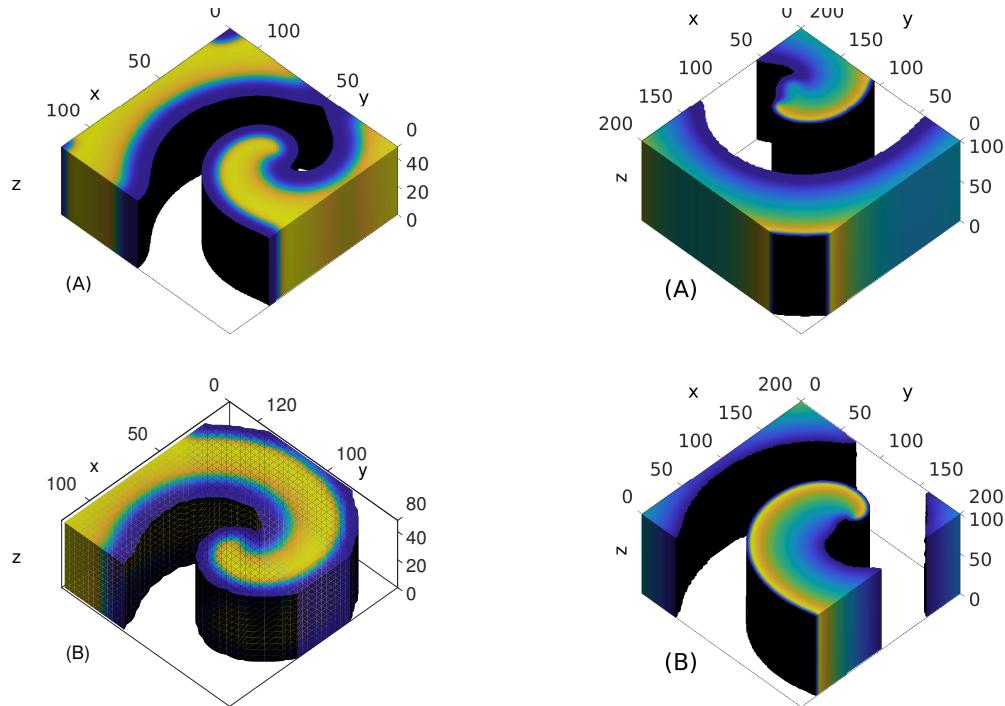


Figure 4.10: (Left column): Solution of equation (3.1) in three dimensions using FD (frame (A)) with a configuration of  $600 \times 600 \times 60$  points, and (frame (B)) SICMD with a configuration of  $N_{ix} = 35$ ,  $N_{cx} = 4$ ,  $N_{iy} = 35$ ,  $N_{cy} = 4$ ,  $N_{iz} = 3$  and  $N_{cz} = 4$  for a total of  $72 \times 72 \times 8$  points. The integration time is  $t = 1000$  time units. The domain used is  $x \in [0, 120]$ ,  $y \in [0, 120]$  and  $z \in [0, 48]$ . (Right column): Solution of equation (4.1) in three dimensions using FD (frame (A)) with a configuration of  $300 \times 300 \times 75$  points, and (frame (B)) SICMD with a configuration of  $N_{ix} = 50$ ,  $N_{cx} = 8$ ,  $N_{iy} = 50$ ,  $N_{cy} = 8$ ,  $N_{iz} = 12$  and  $N_{cz} = 8$  for a total of  $302 \times 302 \times 74$  points. The integration time is  $t = 1000$  time units. The domain used is  $x \in [0, 200]$ ,  $y \in [0, 200]$  and  $z \in [0, 100]$ .

whereas for the SICMD method it was possible to use  $\Delta t = 0.1$ . This is three orders of magnitude larger, thus, depending on the total integration time, it is computationally faster to generate a solution with SICMD. To further emphasize this, for our example, SICMD takes only about 3% of the time that FD does to generate a solution with similar precision.

We also ran simulations using the CMD method, but as it does more operations than FD and SICMD, even when it was possible to take a time step of  $\Delta t = 1 \times 10^{-3}$ , the time it took to generate the solution was about 70% more than in the case of FD. This is the reason why we only considered FD and SICMD for our three dimensional simulations.

OS was also considered, but the front of the wave was deformed considerably with the selection of numerical parameters to obtain a fast simulation. Thus we conclude that OS is not very suitable to obtain three dimensional simulations because the selection of numerical parameters of the scheme itself is model dependent.

For this three dimensional scenario, we also implemented the SICMD method calculating the reaction part once in each of the three intermediate steps (instead of three times) in the ADI scheme. When we compare this two approaches, calculating the reaction part once or thrice, we have that by calculating the reaction part once, we gain a speed up of approximately 30% in terms of computing time. This means that following this approach we would need only 1% of the time that FD does to generate a solution. The price of doing this is of course that we lose precision in our solutions.

Again, as in the two dimensional scenario, the more robust and reliable of the methods considered in this work is SICMD. It generates relatively fast simulations with a good enough precision.

#### 4.7.1 Rotational anisotropy

We have also tested our numerical schemes for the scenario when we have rotational anisotropy in a more realistic domain. To this end, we used the phase field methodology discussed in [Fenton et al. \(2005\)](#) in order to implement no flux boundary conditions on a more complex geometric domain. However, due to the fact that the laplacian operator for this case involves crossed derivatives of the auxiliary phase function  $\phi$ , it was not possible to implement the SICMD method. Nevertheless, it is possible to use the explicit CMD method along with the phase field methodology and obtain very good results. Also, we can apply the same methodology as discussed in [Fenton et al. \(2002\)](#) to find the filament of scroll waves in three dimensions with the CMD method and, again, having that our pseudospectral methods are a fine alternative to study the filament's behavior for different set of parameters. For a fully detailed explanation on the scheme and the treatment of boundary conditions we refer to [Rodríguez-Padilla and Olmos-Liceaga \(2017\)](#).

The bottom plane of our cylindrical slab has a  $-60$  degree rotation with respect to the horizontal. As we increase in height, we increase the rotation with respect to the horizontal so that the plane at the top of our cylinder has a difference of  $120$

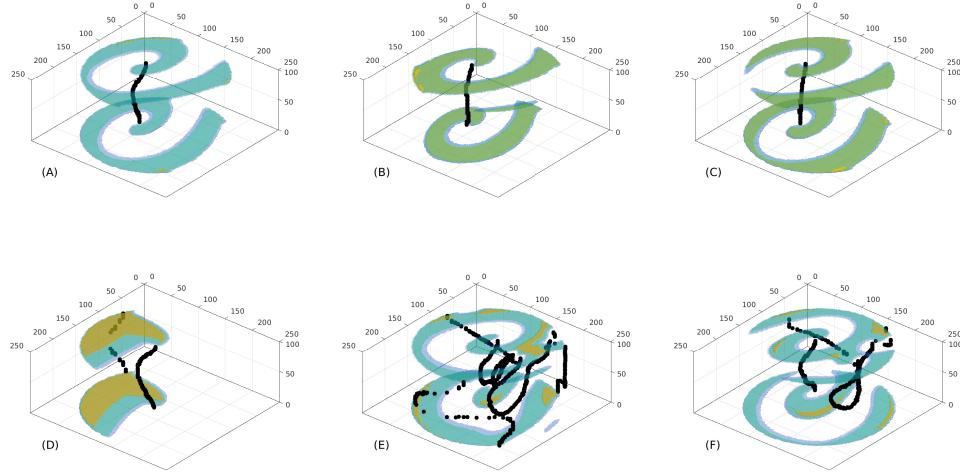


Figure 4.11: Solution of equation (3.1) in three dimensions using CMD cylindrical domain in a three dimensional parallelepipedal box to simulate a cylindrical slab of tissue implementing no flux boundary conditions. The dimensions of the box considered are  $[0, 250] \times [0, 250] \times [0, 50]$  spatial units. The cylinder radius is 100 and it is centered at the domain considered. We also show the filament of the scroll wave. (Top row) Parameter set taken as in Table 4.1. (Bottom row) Parameter set taken as in Table 4.1 with  $M = 7$  and  $Re = 0.9$ .

degrees with the bottom plane thus has a rotation of 60 degrees with respect to the horizontal.

The initial condition used in our simulations is given as follows. We take  $n = 0.5$  for all our domain and we set the variable  $E = 3.0$  on a strip of the domain. That is,  $E(x, y, z) = 3$  for all  $x \leq 50$ .

As previously studied in Karma (1994), by varying the parameters  $Re$  and  $M$  in the model it is possible to obtain solutions with a very different behavior.

In Figure 4.11 we show the evolution of the filament of a scroll wave in the Karma model in a cylindrical domain immersed in a three dimensional parallelepipedal box for two different set of parameters. In the top row of Figure 4.11, we show the evolution of a solution that is known to be stable. Notice that as time advances, the filament of the scroll wave starts to resemble a straight line. On the other hand, by taking a different choice of parameters ( $M = 7$  and  $Re = 0.9$ ) the solution becomes stable, and the appearance of multiple filaments arises.

## 4.8 Discussion

We have developed methods to solve reaction-diffusion equations based on Cheby-shev polynomials. In the end, we have found one that is reliable precision and robust, namely, the SICMD method. What we mean by robust is that we look for a method that only requires the minimum tuning up in the implementation.

With the SICMD method developed in this work we obtained solutions to systems (3.1) and (4.1) as precise as those obtained with FD and we achieved a speed up of about three orders of magnitude using SICMD compared with FD when solving the models in three dimensions. This is an important breakthrough, as now we can reduce by a large margin the computing time to obtain solutions that are precise enough with less amount of computational resources. It is important to point out that the solutions obtained with the SICMD method not only take much less computation time and resources for a desired accurate solution, but also generates reliable solutions to understand cardiac dynamics. For example, spiral wave tip trajectories, restitution curves and filament evolution are well obtained with the SICMD method. Therefore, the SICMD method can be used by cardiac modelers to gain better insights of their models, with low computational cost.

The developed methods were tested with cardiac wave equations but can be used in general to solve equations of the reaction-diffusion type with multiple spatio-temporal scales in general.

OS can be a very good option to solve reaction-diffusion systems but it requires a proper selection of numerical parameters in the method itself to be able to provide good approximated solutions. When we move from the two dimensional scenario to the three dimensional one we have seen that the OS method does not perform well. It is necessary to readjust the numerical parameters in order to obtain a good approximated solution, and also does not preserve the formation of reentrant waves that it was possible to generate in two dimensions. On the other hand, FD, CMD and SICMD are more robust when the extension to three dimensions is taken. It is important to remark that even when these three methods were utilized in three dimensions, for FD and CMD it was needed to reduce the time step in order to obtain numerically stable approximated solutions. This was not the case for SICMD in which was still possible to take the same time step as in the two dimensional case.

From the stability analysis shown in Section 4.5 for the CMD, SICMD and ICMD methods, it follows that larger steps can be taken with ICMD, but that does not imply that solutions are better. In this sense, SICMD proved to be more robust than the ICMD scheme.

A second important result in this work is the proposed scaling (Section 4.4). With the proposed scaling is possible to evaluate the computing time and accuracy that will be obtained when solving a complex model of cardiac wave propagation (independently of the number of variables or parameters) without actually solving the equations. The only information needed is the size of the domain, the maximum rate of change in time for the voltage variable and the coefficient of the second derivative operator. For our current discussion, it is assumed that the system of ODEs for voltage and gate variables of the Hodgkin-Huxley formalism, has its fastest change in the solution due to the inward sodium current. Even that such current has its major contribution during the upstroke, its associated component in the solution of the mathematical model, is always present, giving the major source of stiffness of the ODE. This analysis is very useful when there is a decision to take about sacrificing accuracy versus computing time.

Therefore, the proposed scaling is a generalization of the scaling proposed for Fisher's

equation [Olmos and Shizgal \(2006\)](#). In this case, the value of  $\rho$  in Fisher's equation is equivalent to the inverse of the time constant for the sodium current.

The implementations of each of the methods in this work on more realistic geometries can be achieved by following the phase field approach as in [Fenton et al. \(2005\)](#).

In this work we have used FD to have a reference frame and thus compare and say how good the SICMD method can be. An important observation is that even when the implementation of SICMD is not as straightforward than FD, is very amiable and the speed up that we gain is remarkable. Therefore we can study reaction-diffusion systems varying different parameters of the models, especially in three dimensions, and obtain good approximated solutions in a relatively small amount of time.

As a final conclusion of this work, we can state that the SICMD method has a high probability of performing much better (more accurate solution and less computing time) than FD methods, when solving any three dimensional cardiac model based in the Hodgkin-Huxley formalism. Also, from this work it is possible to obtain estimates of the number of points and time step needed for a pseudo-spectral scheme in order to have some desired accuracy for a model based on the HH formalism.

## Appendix

### Counting the number of operations: FD scheme

The second derivative operator in this case requires, for each node, 4 additions and 2 multiplications. Thus, we have a total of

$$N_{add} = 4N_{px}N_{py} \quad \text{and} \quad N_{mult} = 2N_{px}N_{py}$$

The reaction part of equation (3.1) takes, for each node, 14 additions and 15 multiplications, so that we have, in total,

$$N_{add} = 14N_{px}N_{py} \quad \text{and} \quad N_{mult} = 15N_{px}N_{py}$$

Finally, the Euler scheme does, for each node, 4 additions and 4 multiplications. Then

$$N_{add} = 4N_{px}N_{py}, \tag{4.24}$$

and

$$N_{mult} = 4N_{px}N_{py}. \tag{4.25}$$

### Counting the number of operations: CMD scheme

Recall that the second derivative operators  $\mathbf{D}_x^2$  and  $\mathbf{D}_y^2$  here are blockwise matrices whose sizes are determined by the selection of  $N_{ix}$ ,  $N_{cx}$ ,  $N_{iy}$  and  $N_{cy}$ .

The  $\mathbf{D}_x^2$  operator has 2 blocks of  $(N_{cx} - 1) \times (N_{cx} - 2)$ , each of which requiring  $(N_{cx} - 2)^2$  additions and  $(N_{cx} - 2) \times (N_{cx} - 1)$  multiplications when calculating  $\mathbf{D}_x^2\mathbf{u}$ , and  $N_{ix} - 2$  blocks of  $N_{cx} \times (N_{cx} - 2)$ , each of these requiring  $(N_{cx} - 2)(N_{cx} - 1)$  additions and  $N_{cx}(N_{cx} - 2)$  multiplications when calculating  $\mathbf{D}_x^2\mathbf{u}$ . Taking all these into account, we have that,  $\mathbf{D}_x^2\mathbf{u}$  does a total of

$$N_{add} = 2(N_{cx} - 2)^2 + (N_{ix} - 2)(N_{cx} - 1)(N_{cx} - 2) \tag{4.26}$$

additions and

$$N_{mult} = 2(N_{cx} - 1)(N_{cx} - 2) + (N_{ix} - 2)(N_{cx})(N_{cx} - 2) \tag{4.27}$$

multiplications. Analogously, the number of operations for  $\mathbf{D}_y^2\mathbf{u}$  is given by the previous formulas but with  $N_{iy}$  and  $N_{cy}$ . Finally, the reaction part in equation (3.1) and the Euler scheme requires 14 additions and 15 multiplications and 4 additions and 4 multiplications per node, respectively.

### Counting the number of operations: SICMD scheme

The SICMD method, requires the solution of a system  $L\mathbf{U}\mathbf{x} = \mathbf{b}$ . In two dimensions we use an ADI method to solve a linear system in the  $x$  direction and then another

one in the  $y$  direction LeVeque (2007). Each of the triangular systems per direction uses

$$N_{add} = \frac{(N_c - 3)(N_c - 2)}{2} + (N_i - 1) \left( \frac{(N_c - 2)(N_c - 1)}{2} \right) \quad (4.28)$$

additions, and

$$N_{mult} = \frac{(N_c - 2)(N_c - 1)}{2} + (N_i - 1) \left( \frac{(N_c - 1)(N_c)}{2} - 1 \right) \quad (4.29)$$

multiplications, and where  $N_i$  and  $N_c$  are the number of subintervals and number of points per subinterval in the  $x$ ,  $y$  or  $z$  directions, respectively. In one dimension, to solve the system  $Ax = b$ , the number of operations needed are  $2 * N_{add}$  additions and  $2 * N_{mult}$  multiplications.

Let us compare the number of operations when solving the diffusion equation with the parameters given by the Karma model (Eq. 3.1) in two dimensions with FD with  $N = 600$  points per dimension with CMD, SICMD and OS with 602 points ( $N_{ix} = N_{iy} = 100$  and  $N_{cx} = N_{cy} = 8$ ). According to Table 4.2, FD takes approximately  $1.4 \times 10^6$  and  $7.2 \times 10^5$  additions and multiplications, respectively. On the other hand CMD takes  $5.04 \times 10^6$  and  $5.76 \times 10^6$  additions and multiplications, respectively. Finally, SICMD and OS require  $5.04 \times 10^6$  and  $5.53 \times 10^6$  additions and multiplications, respectively. If instead of  $N_{cx} = N_{cy} = 8$  and  $N_{ix} = N_{iy} = 100$ , we take  $N_{cx} = N_{cy} = 4$  and  $N_{ix} = N_{iy} = 299$ , we halve the number of operations for the pseudospectral methods.

Table 4.1: Physical parameters used in the Karma model to obtain action potentials as in figure (4.1A).

Parameter	$\tau_E$	$\tau_n$	$\gamma$	$E_h$	$E_n$	$E^*$	$Re$	$M$
Value	2.5	200	1.171	1.75	1.0	1.5414	0.8	4

Table 4.2: Summary of the number of operations that each method calculates per time step to obtain an approximated solution when the spatial domain is two dimensional.

Method	Diffusion		Reaction	
	Additions	Multiplications	Additions	Multiplications
FD	$4N_{px}N_{py}$	$2N_{px}N_{py}$	$14N_{px}N_{py}$	$15N_{px}N_{py}$
CMD	$N_{py}(2(N_{cx}-2)^2 + (N_{ix}-2)(N_{cx}-1)(N_{cx}-2)) + N_{px}(2(N_{cy}-1)^2 + (N_{iy}-2)(N_{cy}-1)(N_{cy}-2))$	$N_{py}(2(N_{cx}-1)(N_{cx}-2) + (N_{ix}-2)(N_{cx})(N_{cx}-2)) + N_{px}(2(N_{cy}-1)(N_{cy}-2) + (N_{iy}-2)(N_{cy})(N_{cy}-2))$	$14N_{px}N_{py}$	$15N_{px}N_{py}$
SICMD	$N_{py}((N_{cx}-3)(N_{cx}-2) + (N_{ix}-1)(N_{cx}-2)(N_{cx}-1)) + N_{px}((N_{cy}-3)(N_{cy}-2) + (N_{iy}-1)(N_{cy}-2)(N_{cy}-1))$	$N_{py}((N_{cx}-2)(N_{cx}-1) + (N_{ix}-1)((N_{cx}-2)(N_{cx})-2)) + N_{px}((N_{cy}-2)(N_{cy}-1) + (N_{iy}-1)((N_{cy}-2)(N_{cy})-2))$	$28N_{px}N_{py}$	$30N_{px}N_{py}$
OS	$N_{py}((N_{cx}-3)(N_{cx}-2) + (N_{ix}-1)(N_{cx}-2)(N_{cx}-1)) + N_{px}((N_{cy}-3)(N_{cy}-2) + (N_{iy}-1)(N_{cy}-2)(N_{cy}-1))$	$N_{py}((N_{cx}-2)(N_{cx}-1) + (N_{ix}-1)((N_{cx}-2)(N_{cx})-2)) + N_{px}((N_{cy}-2)(N_{cy}-1) + (N_{iy}-1)((N_{cy}-2)(N_{cy})-2))$	$14N_{px}N_{py}$	$15N_{px}N_{py}$

Table 4.3: Error analysis performed on the Karma model (3.1). We have used the CMD scheme. Total time of integration  $T = 250$ . Data concerning the case when the choosing of points per subdomain is 4 and  $x \in [0, 120]$ . The  $-$  symbol means that we obtained an unstable for that selection of  $\Delta t$ ,  $N_{ix}$  and  $N_{cx}$ .

Configurations	Error				
	$N_{px}(N_{ix}, N_{cx})$	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
3000(1499,4)	-	-	$1.27381 \times 10^{-3}$	$1.26330 \times 10^{-4}$	
2000(999,4)	-	-	$1.27364 \times 10^{-3}$	$4.85629 \times 10^{-4}$	
1000(499,4)	-	$1.40262 \times 10^{-2}$	$1.29045 \times 10^{-3}$	$2.41190 \times 10^{-3}$	
500(249,4)	-	$1.33834 \times 10^{-2}$	$6.88233 \times 10^{-4}$	$9.87054 \times 10^{-3}$	
300(149,4)	0.14075	$1.38483 \times 10^{-2}$	$1.23881 \times 10^{-3}$	$1.48678 \times 10^{-2}$	
150(74,4)	0.12869	$1.12384 \times 10^{-2}$	$1.17882 \times 10^{-2}$	0.16034	

Table 4.4: Error analysis performed on the Karma model (3.1). We have used the CMD scheme. Total time of integration  $T = 250$ . Data concerning the case when the choosing of points per subdomain is 8 and  $x \in [0, 120]$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
3002(500,8)	-	-	$1.27381 \times 10^{-3}$	$4.45484 \times 10^{-11}$
2000(333,8)	-	-	$1.27364 \times 10^{-3}$	$1.79418 \times 10^{-8}$
1004(167,8)	-	$1.40262 \times 10^{-2}$	$1.29045 \times 10^{-3}$	$1.66787 \times 10^{-5}$
506(84,8)	-	$1.33834 \times 10^{-2}$	$6.88233 \times 10^{-4}$	$5.80410 \times 10^{-4}$
302(50,8)	0.14075	$1.38483 \times 10^{-2}$	$1.23881 \times 10^{-3}$	$2.67712 \times 10^{-5}$
152(25,8)	0.12869	$1.12384 \times 10^{-2}$	$1.17882 \times 10^{-2}$	$1.30801 \times 10^{-2}$

Table 4.5: Error analysis performed on the Karma model (3.1). Solutions obtained with FD. Time of integration  $T = 250$ . The numerical parameters used were as discussed on table (4.1) and  $x \in [0, 120]$ . The - and  $\circ$  symbols denote, respectively, that we obtained an unstable solution and that it was not possible to generate an action potential with that configuration.

<b>Configurations</b> $N_{px}$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
4000	-	-	$2.72437 \times 10^{-2}$	$2.59699 \times 10^{-2}$
3000	-	-	$3.65193 \times 10^{-2}$	$3.52455 \times 10^{-2}$
2000	-	$6.87875 \times 10^{-2}$	$5.60647 \times 10^{-2}$	$5.47919 \times 10^{-2}$
1000	-	0.13615	0.12346	0.12219
500	0.44570	0.32239	0.31005	0.31005
300	0.89439	0.78965	0.77906	0.77906
150	$\circ$	$\circ$	$\circ$	$\circ$

Table 4.6: Error analysis performed on the Karma model (3.1). Solutions obtained with SICMD choosing  $N_{cx} = 4$ . Time of integration  $T = 250$ . The numerical parameters used were as discussed in table (4.1) and  $x \in [0, 120]$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
4000(1999,4)	0.17934	$1.76546 \times 10^{-2}$	$1.49251 \times 10^{-3}$	$1.24315 \times 10^{-4}$
3000(1499,4)	0.17842	$1.75271 \times 10^{-2}$	$1.36653 \times 10^{-3}$	$2.50363 \times 10^{-4}$
2000(999,4)	0.17801	$1.71648 \times 10^{-2}$	$1.00764 \times 10^{-3}$	$6.09771 \times 10^{-4}$
1000(499,4)	0.17590	$1.52296 \times 10^{-2}$	$9.27802 \times 10^{-4}$	$2.53625 \times 10^{-3}$
500(249,4)	0.16794	$7.75907 \times 10^{-3}$	$8.38753 \times 10^{-3}$	$9.99542 \times 10^{-3}$
300(149,4)	0.16290	$5.77849 \times 10^{-3}$	$1.33856 \times 10^{-2}$	$1.49921 \times 10^{-2}$
150(74,4)	0.33010	0.17760	0.16180	0.16022

Table 4.7: Error analysis performed on the Karma model for the ICMD scheme with  $N_{cx} = 4$  and  $x \in [0, 120]$ . Total time of integration  $T = 250$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$dt = 1 \times 10^{-2}$	$dt = 1 \times 10^{-3}$	$dt = 1 \times 10^{-4}$	$dt = 1 \times 10^{-5}$
500(249,4)	$6.78394 \times 10^{-2}$	$6.24123 \times 10^{-3}$	$8.60515 \times 10^{-3}$	$4.44609 \times 10^{-3}$
300(149,4)	$6.05049 \times 10^{-2}$	$3.29637 \times 10^{-3}$	$7.40026 \times 10^{-3}$	$6.34348 \times 10^{-3}$
150(74,4)	$6.46634 \times 10^{-2}$	$5.31783 \times 10^{-3}$	$1.22143 \times 10^{-2}$	$1.31227 \times 10^{-2}$

Table 4.8: Error analysis performed on the MV model (4.1). We have used the CMD scheme. Total time of integration  $T = 100$ . Data concerning the case when the choosing of points per subdomain is 4 and  $x \in [0, 200]$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
3000(1999,4)	-	0.34697	$1.63408 \times 10^{-2}$	$1.76170 \times 10^{-2}$
2000(999,4)	-	0.30442	$2.99449 \times 10^{-2}$	$6.38185 \times 10^{-2}$
1000(499,4)	1.43991	0.16482	0.17184	0.20399
500(249,4)	1.43299	0.65232	0.42027	0.38865
300(149,4)	1.44134	1.44023	1.43850	1.43836
150(74,4)	1.43805	1.43805	1.43805	1.43805

Table 4.9: Error analysis performed on the MV model (4.1) for FD. Total time of integration  $T = 100$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
4000	-	0.41391	$8.74863 \times 10^{-2}$	$5.35818 \times 10^{-2}$
3000	-	0.45173	0.12916	$9.52909 \times 10^{-2}$
2000	-	0.55788	0.24904	0.21572
1000	1.43751	1.01556	0.83280	0.81280
500	1.44111	1.43750	1.43503	1.43467
300	1.44111	1.44111	1.44111	1.44111
150	1.43392	1.43392	1.43392	1.43392

Table 4.10: Error analysis performed on the MV model (4.1) for the SICMD. Total time of integration  $T = 100$ .  $N_c = 4$  and  $x \in [0, 200]$ .

<b>Configurations</b> $N_{px}(N_{ix}, N_{cx})$	<b>Error</b>			
	$\Delta t = 1 \times 10^{-2}$	$\Delta t = 1 \times 10^{-3}$	$\Delta t = 1 \times 10^{-4}$	$\Delta t = 1 \times 10^{-5}$
4000(1999,4)	1.43660	0.45169	$4.26413 \times 10^{-2}$	$5.68470 \times 10^{-5}$
3000(1499,4)	1.43653	0.43559	$2.51087 \times 10^{-2}$	$1.76163 \times 10^{-2}$
2000(999,4)	1.43611	0.39398	$2.11190 \times 10^{-2}$	$6.38151 \times 10^{-2}$
1000(499,4)	1.43486	0.26006	0.16339	0.20404
500(249,4)	1.43887	0.69291	0.42722	0.38858
300(149,4)	1.44134	1.44032	1.43859	1.43836
150(74,4)	1.43806	1.43806	1.43806	1.43806



# Chapter 5

## Chebyshev Multidomain Pseudospectral Method to Solve Cardiac Wave Equations with Rotational Anisotropy

In this chapter we will use the Karma model, Eq. (3.1), and we will apply and compare two distinct numerical methods, namely, the FD and the CMD methods described in Chapter 4, but now we will implement rotational anisotropy to approach more to reality.

The structure of this chapter is developed as follows. In Section 5.1, the treatment of irregular domains and the implementation of boundary conditions are discussed. Moving on, in Section 5.2, we show the results obtained. Then, in Section 5.3 we present a convergence study of our numerical method. We conclude this manuscript with a discussion and future work that this field of study has to offer (Section 5.4).

### 5.1 Numerical Methods

The numerical methods to be used in this work are as follows. The first method is the standard finite difference scheme (FD) where space is discretized with the usual three point centered finite difference and time is integrated with the Euler method. For the second method (CMD) space is discretized using multidomain pseudospectral derivative and explicit integration in time with Euler.

All the implementations of the methods have been optimized by tabulating precomputed lookup tables for computationally expensive functions (such as the exponential and hyperbolic tangent) of one variable.

#### 5.1.1 Implementation of the CMD method.

Thus, with the application of the pseudospectral method based on equation (2.32) for the Karma model (Eq. 3.1), in a three dimensional isotropic media we obtain

---

CMD: Explicit Chebyshev Multidomain, FD: Euler Finite Differences.

the following

$$\begin{aligned} E_{ijk}^{m+1} &= E_{ijk}^m + \Delta t A \sum_{p=0}^N D_{ip}^{(2x)} E_{pj}^m + \Delta t B \sum_{p=0}^N E_{il}^m D_{pj}^{(2y)} + \\ &\quad \Delta t C \sum_{p=0}^N E_{ijp}^m D_{pk}^{(2z)} + \frac{\Delta t}{\tau_E} f(E_{ijk}^m, n_{ijk}^m), \\ n_{ijk}^{m+1} &= n_{ijk}^m + \frac{\Delta t}{\tau_n} g(E_{ijk}^m, n_{ijk}^m), \end{aligned} \quad (5.1)$$

where  $E_{ijk} \approx E(x_i, y_j, z_k)$ ,  $n_{ijk} \approx n(x_i, y_j, z_k)$ , and the operators  $D^{(2x)}$ ,  $D^{(2y)}$  and  $D^{(2z)}$  denote, respectively, the derivatives with respect to  $x$ ,  $y$  and  $z$  each of them given by relation (2.32). In equation (5.1),  $A = 4\gamma/(x_R - x_L)^2$ ,  $B = 4\gamma/(y_R - y_L)^2$  and  $C = 4\gamma/(z_R - z_L)^2$  and they appear as a consequence of the linear transformations  $[x_L, x_R]$ ,  $[y_L, y_R]$  and  $[z_L, z_R]$  to  $[-1, 1]$  and include the respective diffusion coefficients.

In order to apply the Chebyshev pseudospectral method, we employ a multidomain approach used previously in [Olmos and Shizgal \(2006\)](#). It consists of dividing the intervals  $[x_L, x_R]$ ,  $[y_L, y_R]$  and  $[z_L, z_R]$  into  $N_i$  overlapping subintervals,  $I_{x,\mu} = [x_0^\mu, x_{N_c-1}^\mu]$ ,  $I_{y,\nu} = [y_0^\nu, y_{N_c-1}^\nu]$  and  $I_{z,\eta} = [z_0^\eta, z_{N_c-1}^\eta]$  respectively, and  $\mu$ ,  $\nu$  and  $\eta = 1, \dots, N_i$ . In each dimension all the subintervals have the same length. For each subinterval, we apply the procedure described in equations (2.25)-(2.30) with the resulting system of ODE's given by equation (5.1) with  $A = \frac{4D_x}{(x_{N_c-1}^\mu - x_0^\mu)^2}$ ,  $B = \frac{4D_y}{(y_{N_c-1}^\nu - y_0^\nu)^2}$  and  $C = \frac{4D_z}{(z_{N_c-1}^\eta - z_0^\eta)^2}$  and the indices in equations (4.5) going from 0 to  $(N_c - 2)N_i + 1$ .  $D_x$ ,  $D_y$  and  $D_z$  are the diffusion coefficients. The second derivative matrices  $D^{(2)}$  in equations (5.1) for the Chebyshev multidomain(CMD) method, are block diagonal matrices as shown in [citeOlmos2006](#).

The application of the Chebyshev multidomain in the solution of equation (3.1) requires a choice of two parameters, the number of subdomain  $N_i$  and the number of Chebyshev points per subdomain  $N_c$ , chosen sufficiently large so as to achieve numerical convergence. For Chebyshev multidomain, we can increase both  $N_c$  and  $N_i$  or fixing one while increasing the other.

### 5.1.2 Implementation of rotational anisotropy.

Throughout the years there has been a very intense study of the structure of ventricular tissue [Thomas \(1957\)](#); [Streeter \(1979\)](#); [Nielsen et al. \(1991\)](#). As discussed in [Fenton and Karma \(1998\)](#), for an ideal parallelepipedal slab of muscle, the fiber axis rotates continuously between the two bounding surfaces of the muscle. Some of the important features are: (i) The cells are shaped as flattened tubes. (ii) They are arranged in sheets roughly parallel to the surfaces (epicardium and endocardium) of the muscle. (iii) The fiber axis (long axis of the cells) rotates continuously between the top and bottom sheets by an angle  $\Delta\theta$ , and the rotation is counter-clockwise from epicardium to endocardium as viewed from the top of the epicardium.

---

A version of this present chapter has been submitted for publication in the International Journal of Modeling, Simulation, and Scientific Computing journal.

To account for these characteristics, the laplacian term in equation 3.1 takes the form  $\nabla \cdot [\mathbf{D} \nabla E]$ , where

$$\mathbf{D} = \begin{pmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{\perp 2} \end{pmatrix},$$

and the coefficients are given by

$$\begin{aligned} D_{11} &= D_{\parallel} \cos^2 \theta(z) + D_{\perp 1} \sin^2 \theta(z), \\ D_{22} &= D_{\parallel} \sin^2 \theta(z) + D_{\perp 1} \cos^2 \theta(z), \\ D_{12} &= D_{21} = (D_{\parallel} - D_{\perp 1}) \cos \theta(z) \sin \theta(z), \end{aligned} \quad (5.2)$$

with  $D_{\parallel}$ ,  $D_{\perp 1}$  y  $D_{\perp 2}$  representing propagation parallel, perpendicular and transmural to the fiber axis, respectively, and

$$\theta(z) = -\Delta\theta/2 + z(\Delta\theta/S) \quad 0 \leq z \leq S, \quad (5.3)$$

measures the angle between the fiber and the  $y$  axis in each plane.

Also as discussed in Fenton and Karma (1998), we focus our attention to an electrically insulated piece of tissue and therefore impose that there is no net current flow normal to the surfaces bounding the tissue. This yields the Neumann boundary conditions

$$n \cdot (\mathbf{D} \nabla E) = 0, \quad (5.4)$$

where  $n$  is the normal unit vector to each of the six bounding surfaces, which translates into solving the following set of equations

$$\begin{aligned} \frac{\partial E}{\partial z} &= 0, \\ D_{11} \frac{\partial E}{\partial x} + D_{12} \frac{\partial E}{\partial y} &= 0, \\ D_{21} \frac{\partial E}{\partial x} + D_{22} \frac{\partial E}{\partial y} &= 0. \end{aligned} \quad (5.5)$$

Unlike with the explicit Euler finite differences scheme, solving system (5.5) with our approach is not straightforward. In fact, the implementation is too cumbersome and difficult, as we need to solve system (5.5) in each of the smaller domains that we define for the method (parameters  $N_{ix}$ ,  $N_{iy}$  and  $N_{iz}$ ).

In order to circumvent this issue we employ a methodology known as phase field. It has been shown that this method is very useful to solve partial differential equations incorporating no flux boundary conditions on irregular domains Bueno-Orovio et al. (2006) for different kind of problems. Also, it has been used to treat irregular geometries in cardiac dynamics to implement Neumann boundary conditions Fenton et al. (2005).

As discussed in Fenton et al. (2005), to treat irregular geometries using the phase-field methodology, we introduce an auxiliary field  $\phi$  that takes on different values inside and outside cardiac tissue and varies smoothly across a thin diffusive interface connecting these two regions.

We refer to Fenton et al. (2005) for a fully detailed explanation of the phase field methodology and the construction of the field  $\phi$ .

Once we have generated our auxiliary field  $\phi$ , we solve the auxiliary equation

$$\phi \frac{\partial E}{\partial t} = \nabla \cdot [\mathbf{D}\phi \nabla E] - \phi f(E), \quad (5.6)$$

or, equivalently,

$$\frac{\partial E}{\partial t} = \frac{1}{\phi} \nabla \cdot [\mathbf{D}\phi \nabla E] - f(E). \quad (5.7)$$

Finally, to solve equation (5.7) we use forward finite differences for time and Chebyshev multidomain pseudospectral for space. With these considerations the numerical scheme results in the following

$$\begin{aligned} E^{n+1} = & E^n + \frac{\Delta t}{\phi} (D_{11}\phi_x E_x^n + D_{11}\phi E_{xx}^n + D_{12}\phi_x E_y^n + \\ & D_{21}\phi_y E_x^n + D_{22}\phi_y E_y^n + D_{22}\phi E_{yy}^n + \\ & 2D_{12}\phi E_{xy}^n + D_{\perp 2}\phi E_{zz}), \end{aligned} \quad (5.8)$$

and we solve for  $\phi \neq 0$  (computationally speaking, we solve the equations for  $\phi \geq .005$ ). The subscripts denote derivatives with respect to the respective variable. For instance, the term  $\phi_z E_z$  denotes the product of the spatial derivatives of  $\phi$  and  $E$  with respect to  $z$ .

Notice that in the absence of rotational anisotropy and setting  $\phi = 1$  on the whole domain, that is if we want to solve on a regular parallelepipedal isotropic domain, equation (5.8) is reduced to equation (5.1).

The implementations of both FD and CMD methods were written in Fortran language compiled with PGI compiler (pgfortran). For the case of the FD method, we used parallelization techniques by means of OpenAcc directives. Simulations were performed in a single desktop PC under Ubuntu 14.04 Linux equipped with an Intel core i7-4790k Haswell processor, 12 GB of RAM and an Nvidia GeForce GTX-970 graphic processing unit.

## 5.2 Numerical studies and results.

We use FD and CMD methods to solve system (3.1). By means of the auxiliary field  $\phi$  we immerse a cylindrical domain in a three dimensional parallelepipedal box to simulate a cylindrical slab of tissue implementing no flux boundary conditions. The dimensions of the box considered are  $[0, 250] \times [0, 250] \times [0, 50]$  spatial units. The cylinder radius is 100 and it is centered at the domain considered.

The bottom plane of our cylindrical slab has a  $-60$  degree rotation with respect to the horizontal. As we increase in height, we increase the rotation with respect to the horizontal so that the plane at the top of our cylinder has a difference of 120 degrees with the bottom plane thus has a rotation of 60 degrees with respect to the horizontal.

The initial condition used in our simulations is given as follows. We take  $n = 0.5$  for all our domain and we set the variable  $E = 3.0$  on a strip of the domain. That is,  $E(x, y, z) = 3$  for all  $x \leq 50$ .

As previously studied in [Karma \(1994\)](#), by varying the parameters  $Re$  and  $M$  in the model it is possible to obtain solutions with a very different behavior. Therefore, in the next subsections we study two different scenarios of these possible solutions.

### 5.2.1 Stable scroll wave numerical solutions.

**Table 5.1:** Parameters used in the Karma model in this work

Parameter	$\tau_E$	$\tau_n$	$\gamma$	$E_h$	$E_n$	$E^*$	$Re$	$M$
<b>Set 1</b>	5	250	1.171	3.0	1.0	1.5414	0.8	3
<b>Set 2</b>	5	250	1.171	3.0	1.0	1.5414	0.9	7

For this case, we use parameter set 1 in Table 5.1 which has been known to produce stable physiological solutions in two and three dimensions. This means that scroll wave solutions obtained with system (3.1) are stable under small perturbations.

We integrate system 3.1 for a total of  $T = 500$  time units using both methods, FD and CMD and for the initial condition and domain we use the ones mentioned in Section 5.2.

For the FD method, we have a configuration of  $180 \times 180 \times 90$  points in our domain. This configuration in the  $x$  and  $y$  dimensions was the minimum needed in order to have propagation of the pulse. The time step for the FD method is  $\Delta t = 0.001$ . Larger time steps taken lead to instabilities in the numerical solution.

On the other hand, for the case of the CMD method we have a configuration of  $N_{ix} = N_{iy} = 40$ ,  $N_{iz} = 14$  and  $N_{cx} = N_{cy} = N_{cz} = 4$  for a total of  $82 \times 82 \times 30$  points. The time step needed for the CMD method to obtain a stable numerical solution with the afore mentioned configuration of points is  $\Delta t = 0.01$ . Notice that this is an order of magnitude larger than the case for the FD method.

In Figure 5.1 we show the evolution of the numerical solutions of the Karma model generated with both methods, FD and CMD, for an initial condition as mentioned above. Frames A, B and C in Figure 1 correspond to the numerical solution generated with the FD method, whereas frames D, E and F correspond to the numerical solution obtained with the CMD for  $t = 100, 300$  and  $t = 500$  time units, respectively.

In Figure 5.1, the solutions obtained with both methods are very similar. A clearer comparison between the obtained solutions is shown in Figure 5.2 which shows level curves ( $E = 1.2$ ) of the solutions obtained with both methods (FD (gray) and CMD (black)) but shown only at different plane projections:  $z = 0$  (Bottom plane),  $z = 50$  (Middle plane) and  $z = 100$  (Top plane).

Initially, the solutions obtained with both methods are very similar but, as time advances, both solutions start to differ (see third column in Figure 5.2).

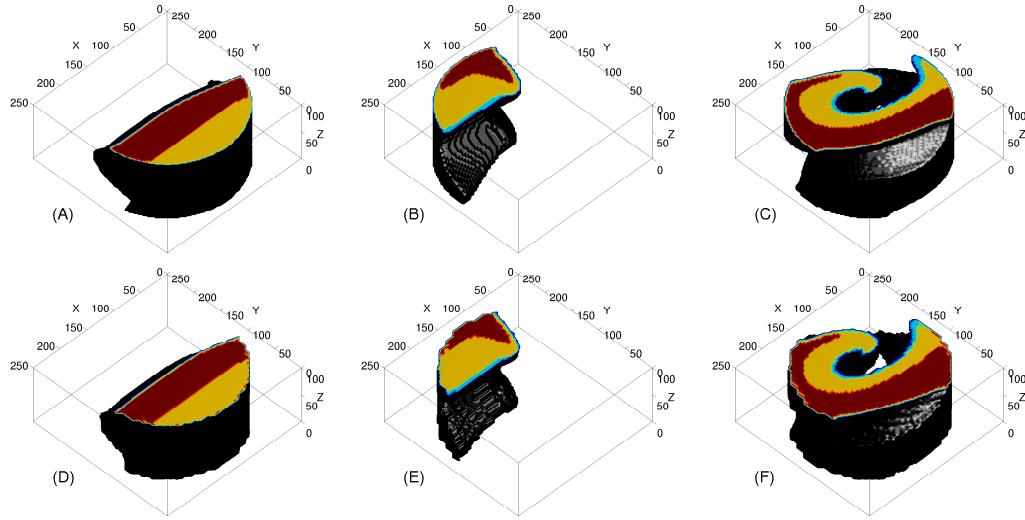


Figure 5.1: Frames (A)-(C): numerical solution generated with the FD method with a configuration of  $180 \times 180 \times 70$  points for times  $t = 100, 300$  and  $t = 500$ , respectively. Frames (D)-(F): numerical solution obtained with the CMD method with a configuration of  $N_{ix} = N_{iy} = 40$ ,  $N_{iz} = 14$  and  $N_{cx} = N_{cy} = N_{cz} = 4$  for a total of  $82 \times 82 \times 30$  points for  $t = 100, 300$  and  $t = 500$ , respectively. Parameter Set 1 in table 5.1 were used to generate both numerical solutions.

An important observation here is the difference in the number of points needed in each method to generate the numerical solution,  $180 \times 180 \times 90 = 2916000$  for the FD method, versus  $82 \times 82 \times 30 = 201720$  for the CMD method which is about 15 times less points than FD. This along with the time step needed for each case ( $\Delta t = 0.001$  for FD and  $\Delta t = 0.01$  for CMD) translates automatically into a dramatical reduction of computational cost when using the CMD method.

In order to see how far are the compared solutions to a converged one we have generated numerical solutions with more points. Especifically, for the FD method, we generated a numerical solution consisting of  $600 \times 600 \times 240$  points whereas for the CMD method we generated a numerical solution with a configuration of  $240 \times 240 \times 120$  points. We used the same initial condition and domain as mentioned in Section 5.2 and parameter set 1 from Table 5.1.

To have a clearer comparison between these two solutions, in Figure 4.3 we show level curves ( $E = 1.2$ ) of the solutions obtained with both methods (FD (gray) and CMD (black)) for three different times, namely  $t = 100, 300$  and  $t = 500$ , but shown only at different plane projections:  $z = 0$  (Bottom plane, bottom row frames),  $z = 50$  (Middle plane, middle row frames) and  $z = 100$  (Top plane, top row frames). In every frame we have plotted the 4 corresponding solutions, the one generated with FD with  $180 \times 180 \times 90$  points, FD with  $600 \times 600 \times 240$  points, the solution generated with CMD consisting of  $240 \times 240 \times 120$  points and the solution generated with CMD consisting of  $82 \times 82 \times 30$  points.

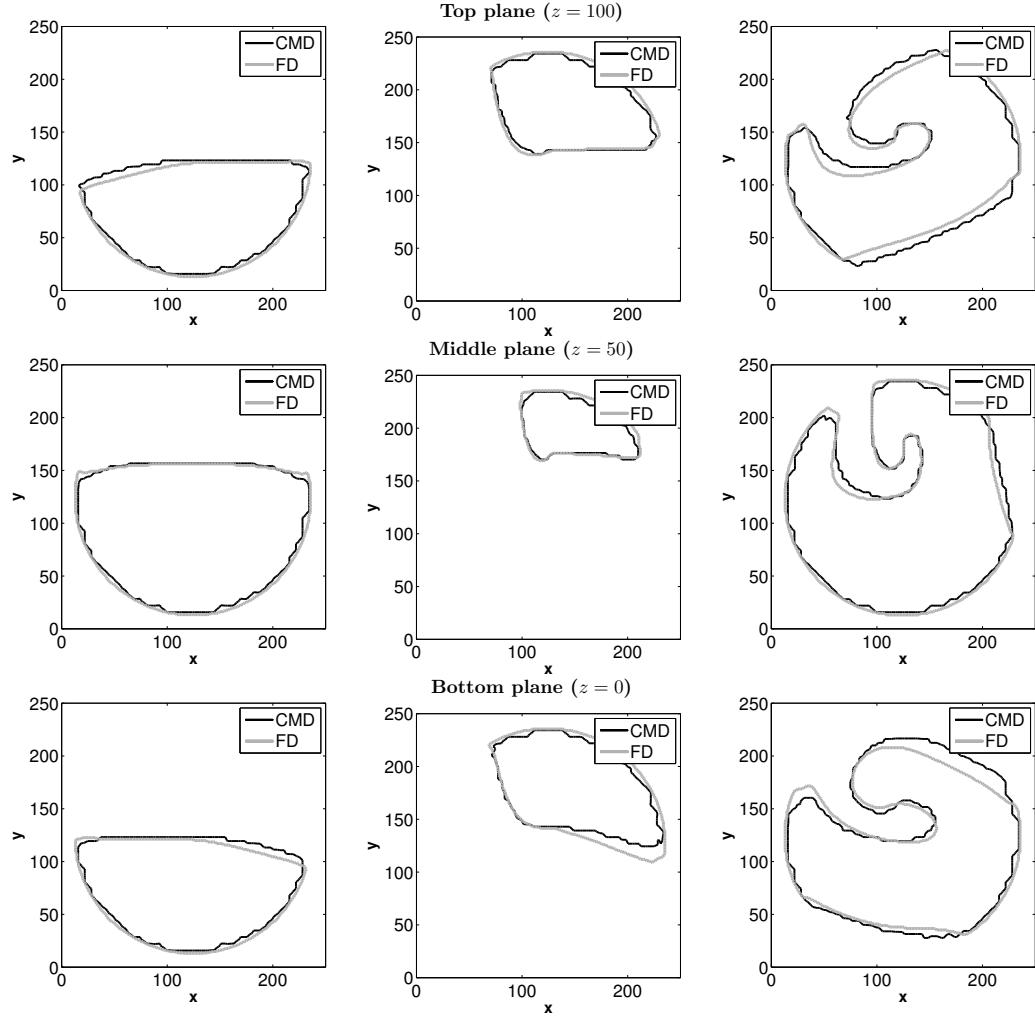


Figure 5.2: Numerical solutions generated with both methods, FD and CMD, corresponding to Figure 5.1. Each column represents a different integration time: Left ( $t = 100$ ), Middle ( $t = 300$ ) and Right ( $t = 500$ ). Each row shows the projection of the solution at plane  $z = 100$  (Top),  $z = 50$  (Middle) and  $z = 0$  (Bottom). Parameter Set 1 in table 5.1 were used to generate both numerical solutions. The contour plots correspond to  $E(x, y, z) = 1.2$ .

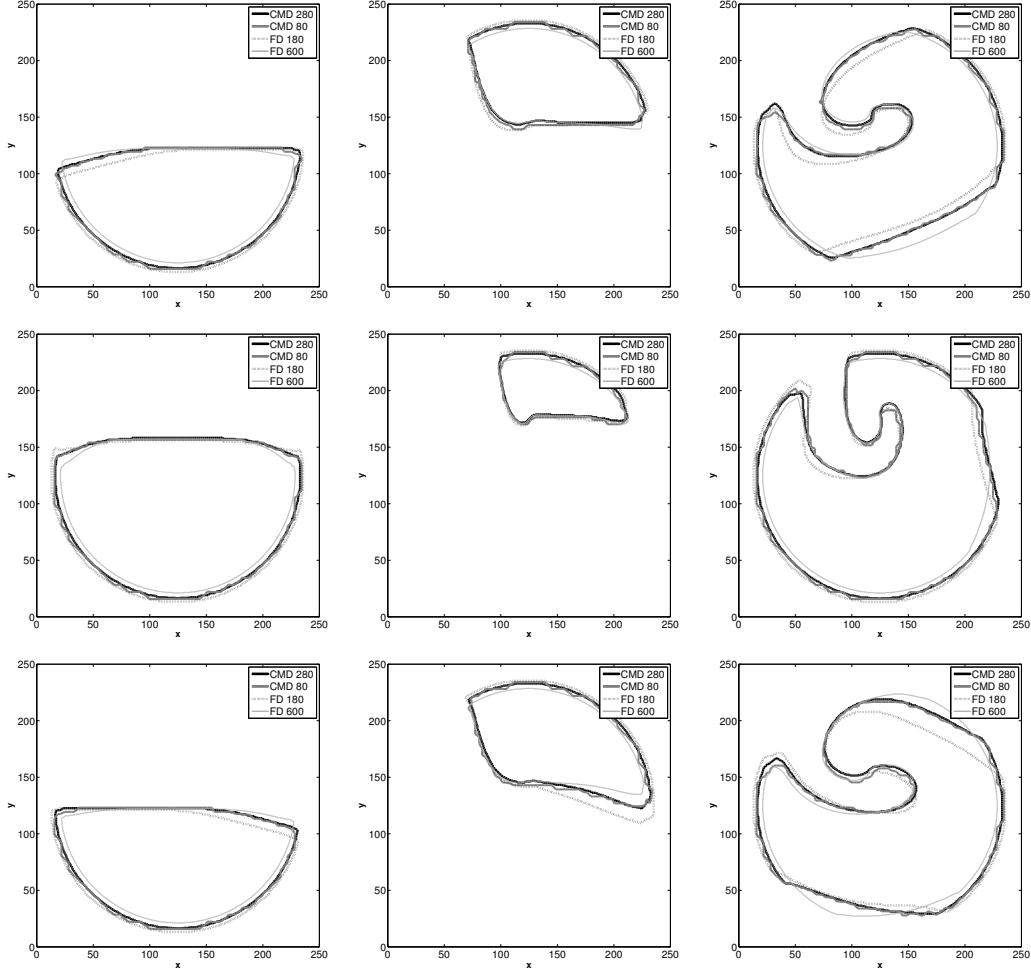


Figure 5.3: Level curves ( $E = 1.2$ ) of the solutions obtained with both methods (FD and CMD) for four different configurations of points. Namely, CMD consisting of  $240 \times 240 \times 120$  points (black), solution generated with CMD consisting of  $82 \times 82 \times 30$  points (dark gray), solution generated with FD with  $180 \times 180 \times 90$  (dotted line) points and FD with  $600 \times 600 \times 240$  points (light gray). Each column represents a different integration time: Left ( $t = 100$ ), Middle ( $t = 300$ ) and Right ( $t = 500$ ). Each row shows the projection of the solution at plane  $z = 100$  (Top),  $z = 50$  (Middle) and  $z = 0$  (Bottom). Parameter Set 1 in table 5.1 were used to generate all numerical solutions.

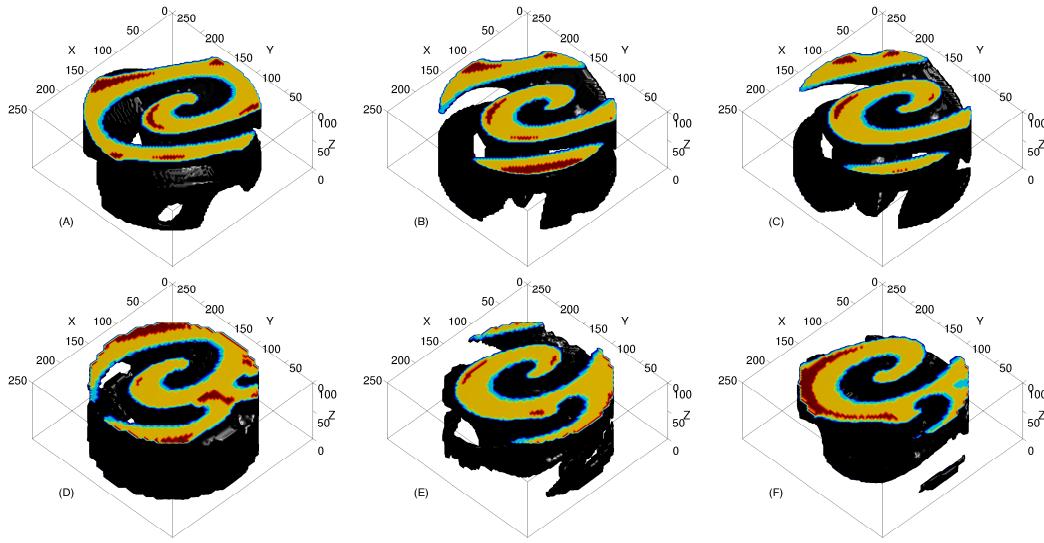


Figure 5.4: Frames (A)-(C): numerical solution generated with the FD method with a configuration of  $180 \times 180 \times 70$  points for times  $t = 500, 1000$  and  $t = 1500$ , respectively. Frames (D)-(F): numerical solution obtained with the CMD method with a configuration of  $N_{ix} = N_{iy} = 40$ ,  $N_{iz} = 14$  and  $N_{cx} = N_{cy} = N_{cz} = 4$  for a total of  $82 \times 82 \times 30$  points for  $t = 500, 1000$  and  $t = 1500$ , respectively. Parameter Set 2 in Table 5.1 were used to generate both numerical solutions.

Notice that the solution generated with CMD consisting of  $82 \times 82 \times 30$  points is closer to the one generated with FD with  $600 \times 600 \times 240$  points than the solution generated with FD with  $180 \times 180 \times 90$  points. With this we are in a position to claim that the CMD method is a very good alternative to solve system 3.1.

### 5.2.2 Unstable scroll wave numerical solutions.

We have explored also some other configurations of parameters which are known to produce chaotic patterns [Karma \(1994\)](#). This means that if we generate a numerical solution of the scroll wave kind, eventually presents break up of the wave, and multiple scroll waves appear. In Figure 5.4, we show the evolution of two numerical solutions in the Karma model, Eq. (3.1), generated with both methods, FD and CMD, utilizing parameter set 2 in Table 5.1. Frames A, B and C in Figure 5.4 correspond to the numerical solution obtained with FD, whereas the remaining three frames, D, E and F in Figure 5.4 show the numerical solution generated with the CMD for three different times  $t = 500, 1000$  and  $t = 1500$ , respectively. Both of the aforementioned numerical solutions were generated using the initial condition described at the beginning of the present section.

In Figure 5.5, the top row shows both solutions corresponding to Figure 5.4 but only looking at the top plane ( $z = 100$ ) for the respective three different times  $t = 500, 1000$  and  $t = 1500$ . In the middle row of Figure 5.5, we have the analogous scenario for the middle plane ( $z = 50$ ); whereas the bottom row in Figure 5.5

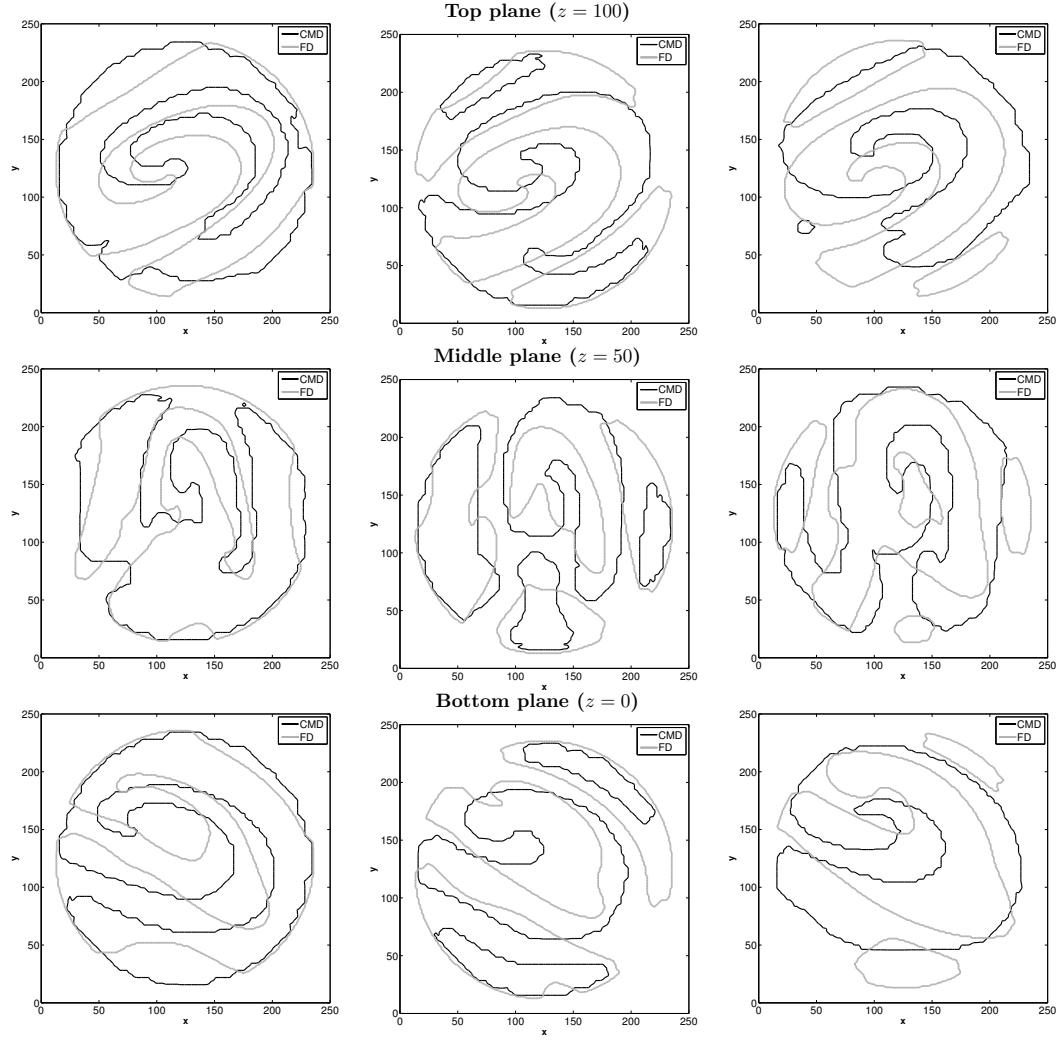


Figure 5.5: Numerical solutions generated with both methods, FD and CMD, corresponding to Figure 5.4. Each column represents a different integration time: Left ( $t = 500$ ), Middle ( $t = 1000$ ) and Right ( $t = 1500$ ). Each row shows the projection of the solution at plane  $z = 100$  (Top),  $z = 50$  (Middle) and  $z = 0$  (Bottom). Parameter Set 2 in table 4.1 were used to generate both numerical solutions. The contour plots correspond to  $E(x, y, z) = 1.2$

corresponds to the same solutions as depicted in Figure 5.4 for the  $z = 0$  plane.

With the experience gained in Chapter 4, we have learned that when using the CMD method to solve the Karma model, Eq. 3.1, in an isotropic medium, the precision of the solutions is comparable to the usual finite differences schemes. This past experience along with the results shown in figures 5.1-5.5 let us claim that using the CMD method with the configurations described in the present section gives acceptable solutions in comparison with the FD scheme.

As in Section 5.2.1, the configurations of points were  $180 \times 180 \times 90$  points for FD and  $82 \times 82 \times 30$  for CMD. Also, as in Section 5.2.1, the time steps were  $\Delta t = 0.001$  and  $\Delta t = 0.01$  for FD and CMD, respectively.

### Computation with GPU's

An important remark on the computational cost of generating the numerical solutions described in the present section is that even though parallel techniques were implemented for the FD method, the solutions obtained with the CMD method were faster. More specifically, the numerical solutions generated with the FD method took, for the solutions depicted in figures 4.1 and 4.2, on the equipment described in Section 4.2, around 6500 seconds of computing, which is roughly about 2 hours, compared to the numerical solution obtained with the CMD method, which took, with the same equipment, around 2400 seconds of computing time, which is roughly around 40 minutes. To put things in perspective and be more incisive, the implementation of the FD method without the parallel techniques takes, in the same equipment, around 32 hours of computing time to generate a single solution. Then, as we claim that the solutions generated with CMD and FD are comparable, we have that by implementing CMD alone we have a speed up of about 3 times, when we consider the parallel techniques for FD, and an astonishing 48x speed up when the parallel techniques are not implemented for FD.

## 5.3 About the convergence of the method

As is well known, Gottlieb (Gottlieb (1981)) has made a groundbreaking work by proving that Chebyshev pseudospectral methods are convergent for the heat equation. Also, Reddy and Trefethen (Reddy and Trefethen (1990)) have demonstrated that full discrete pseudospectral methods exhibit Lax-Stability. It is important to remark that our system of interest, Eq. (3.1), is basically the heat equation plus a nonlinear term.

In this section, we address the issue of convergence of the method. We will separate the study in consistency and stability. For each topic, we will analyse both cases: assuming first that we have  $N_i = 1$  and arbitrary  $N_c$ . Then we will proceed with the multidomain case.

Equations for cardiac wave propagation have similar behavior. The major source of the different scales, is due to the time scale of the fast inward sodium current Hodgkin and Huxley (1952); Beeler and Reuter (1977). This process occurs in a

much faster time scale than the rest of the processes involved in the formation of action potentials Hodgkin and Huxley (1952); Beeler and Reuter (1977). The entrance of the fast inward sodium current provokes a fast change in what is called the voltage variable. This time scale is the one that imposes the fast changes in time in the model. By considering the complete model, the scale given by the coefficient of the diffusive process, the fast time scale and the size of the physical domain, dictate the different scales in space.

In general, for a model of cardiac wave propagation, the formulation has the following form

$$\frac{\partial u}{\partial t} = K \frac{\partial^2 u}{\partial x^2} + \rho f(u) \quad (5.9)$$

where  $u$  is the voltage variable and  $\rho$  is the inverse of the time constant for the sodium fast inward current times the maximum value of  $f$  over the corresponding domain. Based on this information, and the fact that simulations take place over a finite domain  $[a, b]$ , it is possible to scale equation (5.9) such that the problem is equivalent to

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2} + f(u) \quad (5.10)$$

over the domain  $[\sqrt{\frac{\rho}{K}}a, \sqrt{\frac{\rho}{K}}b]$  with  $z = \sqrt{\frac{\rho}{K}}x$  and  $\tau = \rho t$ . In order to compute  $\rho$  for each case, we can find theoretically the maximum of  $f$  for each model or simply give an accurate estimate by solving the corresponding ODE system and finding the maximum in absolute value of the derivative for the variables that model voltage.

If we study equation (5.10) without the diffusive term, it is clear that in order to have absolute stability it is necessary to know the maximum rate of change of the field  $f$ . Let  $\lambda$  be the maximum rate of change of  $f(u)$  over the interval  $[a, b]$ . Thus, equation (25) can be studied by means of the test linear equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \lambda u. \quad (5.11)$$

Let us observe that in this type of equations, the domain size plays a very important role in order to find  $\lambda$ . When dealing with arbitrary domains, the role of the parameter  $\rho$  (Eq. (5.9)) is key in order to obtain the value  $\lambda$ .

Now we proceed to address the issue of consistency of our method. We will present our analysis for the case where the space is one dimensional  $u = u(x, t)$  for clarity of exposition. As we are taking an explicit method, increasing spatial dimensions requires additional analysis on the discretization size, but the treatment is of an analogous manner.

Also, as our auxiliary field function  $\phi$  is constructed in a way that is smooth and is always immersed on a closed and bounded interval (Fenton et al. (2005)), we have that the function  $\phi$  is bounded. Thus, we will do an analysis where we suppose that  $\phi = 1$ , so that our system of interest, initially a one dimensional problem of the form (5.8), can be reduced to a system of the form (5.1). To test consistency of our method, we follow the same spirit as in ODEs. To this end, we consider a linear test equation. Under these considerations we have the following:

**Lemma 5.3.1.** *The CMD method is consistent.*

*Proof.* Let us consider the linear equation (5.11) over a given interval  $[a, b]$ . By using forward Euler in time and spatial monodomain Chebyshev pseudospectral, this is, taking  $N_i = 1$  and  $N_c$  arbitrary, we have, for each element  $u(x, t)$ , the following

$$\frac{u(x, t+k) - u(x, t)}{k} = D_x^2 u(x, t) - \lambda u(x, t). \quad (5.12)$$

Thus, the local truncation error is given by

$$\tau(x, t) = \frac{u(x, t+k) - u(x, t)}{k} - [D_x^2 u(x, t) - \lambda u(x, t)]. \quad (5.13)$$

Using Taylor expansion in the first term in the right hand side of equation (5.13) we have

$$\tau(x, t) = u'(x, t) + \mathcal{O}(k) - [D_x^2 u(x, t) - \lambda u(x, t)], \quad (5.14)$$

where the prime superscript denotes differentiation with respect to time. Let us examining the term  $D_x^2 u(x, t)$ . Recall that the matrix operator  $D_x^2$  is constructed via equation (2.32) where the interpolator is given by relation (2.29). On the other hand, we know that if  $u$  is a sufficiently smooth function and  $P$  is an interpolator for  $u$  we have the following (Burden and Faires (2011))

$$u(x) = P(x) + R(x) \quad (5.15)$$

where  $R(x)$  is given by

$$R(x) = \frac{u^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i), \quad (5.16)$$

$n$  is the number of points with which we are interpolating, the  $x_i$ 's are the actual quadrature points and  $\xi(x)$  is a number contained in  $[a, b]$  not generally known. In practice, as we usually do not know  $u$  explicitly it is not possible prove smoothness analytically. However, what we do know, thanks to experimental data, is that the function  $u$ , although it can have abrupt changes over the interval of interest  $[a, b]$ , it is of an exponential behavior. Thus, we are in position to use equation (5.15).

Differentiating twice equation (5.15) and using relation (5.12), the local truncation error for our case takes the form

$$\tau(x, t) = \mathcal{O}(k) + R''(x). \quad (5.17)$$

Let us analyze now the term  $R''(x)$ . From equation (5.16), we arrive at the next expression

$$R''(x) = \frac{u^{(n-1)}(\xi(x))}{(n-1)!} Q_1(x) + \frac{u^{(n)}(\xi(x))}{n!} Q_2(x) + \frac{u^{(n)}(\xi(x))}{n!} Q_3(x) + \frac{u^{(n+1)}(\xi(x))}{(n+1)!} Q_4(x), \quad (5.18)$$

where  $Q_1(x), Q_2(x), Q_3(x)$  and  $Q_4(x)$  are polynomials of degree  $n, n - 1, n - 1$  and  $n - 2$ , respectively. As we are in a closed interval, all these 4 polynomials are bounded. Therefore, each of the expressions in the right hand side of equation (5.18) goes to zero as  $n \rightarrow \infty$ . Thus, the local truncation error goes to zero as we refine the time and space discretizations, assuring us that the numerical scheme is consistent.

Recall that our numerical scheme has two parameters with which to increase the number of discretization points,  $N_i$  and  $N_c$ . If we fix  $N_i$ , by the reasoning in the paragraphs above, we have that our numerical scheme is consistent in each of the  $N_i$  subintervals, and, therefore, the method is consistent on the whole domain.

We proceed to make an analysis when we fix  $N_c$ . By construction, the last point in  $N_{i-1}$  is the second point in  $N_i$ , thus, by tending  $N_i \rightarrow \infty$ , the distance between these said points tends to zero. Therefore, for  $N_i$  sufficiently large, the original interval of interest,  $[a, b]$ , gets partitioned into the  $N_i$  subintervals. Now, for every subinterval  $N_i$  we have  $N_c$  points, and, as a consequence, we have that the degree of the interpolating polynomials is  $N_c$ . When calculating the local truncation error for this case, the residual in Eq. (5.18) has a similar form. This is, each term in the right hand side of Eq. (5.18) is a product of a term that decays in a factorial manner and a polynomial term. It is important to note that as  $N_i \rightarrow \infty$ , the maximum distance of the quadrature points  $\max|x_i - x_{i-1}| \rightarrow 0$ , so that the term in the residual (Eq.(5.16))

$$\prod_{i=0}^n (x - x_i) \rightarrow 0.$$

Therefore, we have that our method is consistent for the multidomain case.  $\square$

We now proceed to study and give some insights on stability of the numerical method in the sense of Lax-Richtmyer ([Trefethen \(1996\)](#)). The domain in which we will work from now on will be  $x \in [-1, 1]$ . To address stability of our method we will use the next result.

**Lemma 5.3.2.** *The CMD method is stable in the Lax-Richtmyer sense for both monodomain and multidomain cases.*

*Proof.* By using forward Euler in time, and pseudospectral in space in Eq. (5.11) we arrive at the following scheme

$$u_{i,j}^{k+1} = (I - \Delta t \lambda I + \Delta t M D_x^2) u_{i,j}^k, \quad (5.19)$$

where  $u_{i,j} \approx u(x_i, t_j)$ ,  $M = \left(\frac{2}{X_R - X_L}\right)^2$  is a constant that rescales the domain of interest into  $[-1, 1]$  and  $D_x^2$  is the second derivative matrix operator with No-Flux boundary conditions. Clearly, the stability of our method depends on the eigenvalues of the operator  $A = (I - \Delta t \lambda I + \Delta t M D_x^2)$ . If the maximum eigenvalue is less or equal than 1, we have stability. However, notice that the behavior of the eigenvalues

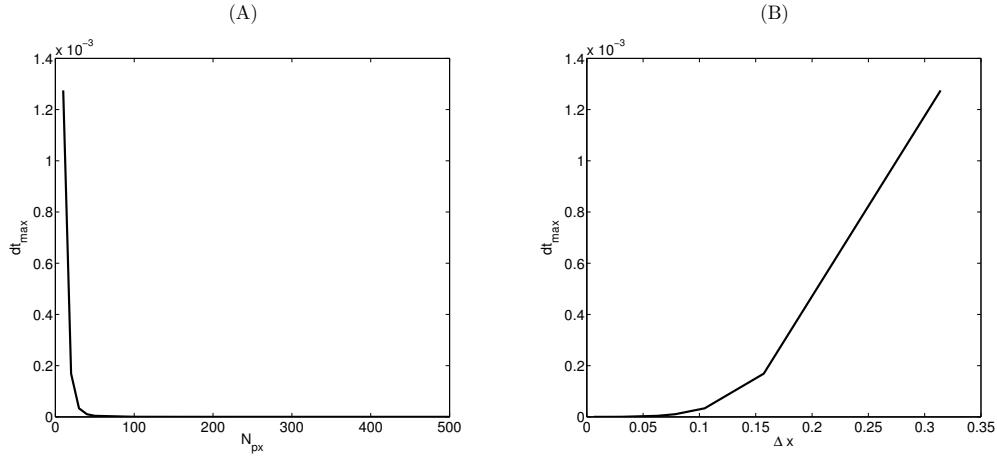


Figure 5.6: Behavior of  $dt_{\max}$  (maximum value of  $dt$ ) such that the maximum eigenvalue of the operator  $A$  is equal to 1: (A) as a function of  $N_{px}$  and (B): as a function of  $\Delta x$ , respectively.

of  $A$  can be modified by selecting different  $\Delta t$ ,  $\lambda$ ,  $N_i$  or  $N_c$ . The eigenvalues of the operator in the right hand side of equation (5.19) are not known analitically to our understanding. Therefore, we proceed by approximating such values numerically.

Let us define the function  $g(\Delta t)$  as follows

$$g(\Delta t) = |\max(eig(A))| - 1, \quad (5.20)$$

where  $eig(A)$  denotes the eigenvalues of the operator  $A$ . Thus, finding the roots of the function  $g$  gives us information for which our numerical scheme will be stable.

### Monodomain case

Let us explore first the monodomain case for the derivative operator,  $D_x^2$ , this is,  $N_i = 1$ . In the collocation method, it is known that the collocation points are not equally spaced. In this case, the maximum grid size is easily approximated by using a Taylor approximation and is given by  $\Delta x \approx \frac{\pi}{N_{px}}$ .

In Figure 5.6 we have the behavior of the maximum of  $\Delta t$  that is possible to take as a function  $N_{px}$  (frame (A)) and also as a function of the grid size  $\Delta x$  (frame (B)).

From Figure (5.6), frame (A), it is clear that the behavior of  $dt_{\max}$  is of the form  $dt_{\max} = c(N_{px})^a$ . In Figure (5.7), we show our data, after a log-log transformation, along with the respective fit. For our case, we have  $c = 1.8838$  and  $a = -3.8$ .

It is important to note here that for distinct values of  $\lambda$ , the behavior of  $dt_{\max}$  was of a similar form. It is for values of  $\lambda$  larger than  $1 \times 10^6$  that we start to notice the change on the behavior of  $dt_{\max}$ . Thus, the value of  $\lambda$  does not dictate a major role on the behavior of the maximum time step needed in order to assure stability of the method. What really dictates this behavior is the operator  $D_x^2$ .

Therefore, under all these considerations, if we take a configuration of points such that  $dt$  and  $N_{px}$  satisfies the relation

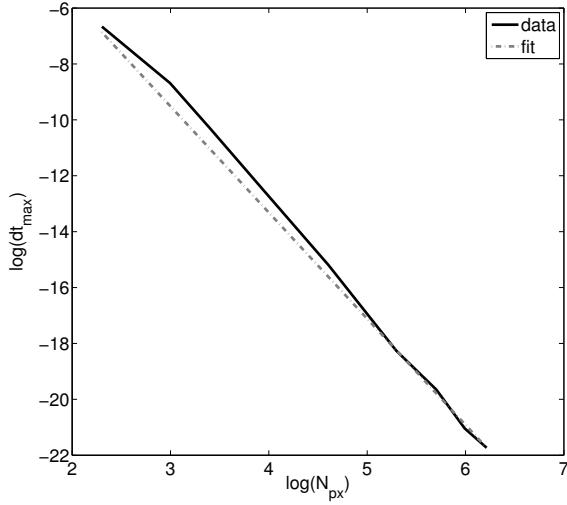


Figure 5.7: Behavior of  $dt_{\max}$  as a function of  $N_{px}$  in a log-log scale along with the respective fit  $dt_{\max} = c(N_{px})^a$ . For this case,  $c \approx 1.8838$  and  $a \approx -3.8$ .

$$\frac{\Delta t}{(N_{px})^{3.8}} < c, \quad (5.21)$$

we will have that our operator  $A$  will have a maximum eigenvalue less than one, and, thus, our numerical scheme will be stable.

An important remark is that by changing the domain in which we are interested in solving our equation, the behavior of  $dt_{\max}$  still obeys the same power law, but with different value of  $c$ .

### Multidomain case

We move now to study the multidomain case. For this scenario, we have now 4 distinct parameters that we can adjust, namely,  $N_i, N_c, \lambda$  and  $\Delta t$ . We proceed as the last section. We construct a function  $g$  given by (5.20) and we search for the roots. As we have a function of  $\Delta t$ , we are left with three parameters that we can choose for,  $N_i, N_c, \lambda$ . Notice that by increasing  $N_i$  or  $N_c$  (or both) we are increasing the number of points. This because the number of points is given by  $N_p = N_c + (N_i - 1)(N_c - 2)$ .

Then, for clarity in exposition, we will fix  $N_c = 4$ , and we will increase  $N_i$ . Unlike the monodomain case, in this scenario it is possible to take larger  $N_i$  and still obtain  $dt$  such that the operator  $A$  in equation (5.20) is less than one. However, again, unlike monodomain, the value of  $\lambda$  for this scenario has an enormous impact. Increasing  $\lambda$  is reflected immediately on the maximum  $dt$  that is possible to take in order to have the eigenvalues of  $A$  less than one. Increasing an order of magnitude of  $\lambda$  results in a decrease of an order of the maximum  $dt$  that is possible to take to assure that the maximum eigenvalue of  $A$  is less than one.

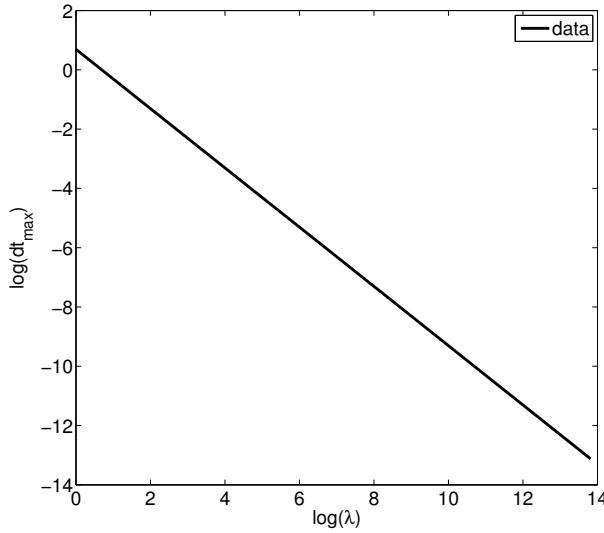


Figure 5.8: Behavior of  $dt_{max}$  as a function of  $\lambda$  in a log-log scale. For this case, we fixed  $N_c = 4$  and  $N_i = 500$  for a total of 1002 collocation points.

In Figure (5.8) we show the behavior of the maximum of  $dt$  as a function of  $\lambda$  in a log-log scale. Clearly, we have an exponential decay. As is possible to take any configuration of  $N_i$ , we have taken for this case  $N_i = 500$ , thus giving us a total of  $N_{px} = 1002$  collocation points.

Recall that the value of  $\lambda$  represents the decay in the reaction part. Then, for the multidomain case we have that more rapid decays in the model (larger  $\lambda$ 's) have a direct impact on the time step.  $\square$

By having both consistency and stability of our numerical method, we are in position to assert the following result:

**Proposition 5.3.1.** *The CMD method is convergent.*

*Proof.* Combining both lemmas (5.3.1) and (5.3.2) we have that the CMD method is both consistent and stable. Thus, we have that the CMD method is convergent.  $\square$

## 5.4 Discussion

We have implemented a method to solve reaction-diffusion equations based on Chebyshev polynomials. The main advantage of this method is the possibility of taking less discretization points to obtain a numerical solution compared to the usual explicit Euler Finite Difference method. A direct consequence of this is a decrease in computing times to generate the solutions. With this, we have the possibility to explore different configurations of the model parameters to study the behavior of the

numerical solutions in a relatively small amount of time and thus obtaining more knowledge about the dynamics of the solutions in the Karma model.

It is important to note that even when using considerably less points to obtain a numerical solution with the CMD method, the precision of such solution is comparable with those solutions usually obtained with Euler finite differences methods with more discretization points.

When taking into consideration the comments and conclusions in the earlier section, the question in which the authors are interested is, how much speed up can we gain by implementing parallel techniques on the CMD method? This is the present and future focus of our work. With the acceleration gained when implementing these techniques in the FD method, we are inclined to believe that the same can be achieved for the Chebyshev multidomain method. The impact of this is big because by decreasing computing times to generate solutions in three dimensional media we have the means to study a broader set of parameters of the models, in relatively small amount of times.

# Chapter 6

## Computational Studies

The present chapter is devoted to present a comparison study of the performance of three different graphic processing units. To this end, we implement a method to obtain numerical solutions to systems of equations of the reaction-diffusion type that arise in cardiac dynamics. Specifically, we use the Karma model 3.1, which is a simple two variable model. We also study the Beeler-Reuter model Beeler and Reuter (1977) which is an eight, first order, simultaneous, non-linear differential equations system. Finally, we move on to study the complex ten Tusscher model Ten Tusscher et al. (2004); Ten Tusscher and Panfilov (2006), which is a human ventricular tissue model consisting of 24 variables.

We will use the same numerical method to solve all three systems, namely, the well known Euler finite difference scheme, for both space and time. We will implement also look up tables, a methodology known to optimize computationally expensive functions, such as hyperbolic tangents, exponentials, to name some of the usual.

To have a more thorough study, we have carried out simulations in single and double precision in each language.

The experiments that we carried out are as follows. For each model, we have solved the system in question for six different two dimensional domains consisting of  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$ ,  $1024 \times 1024$  and  $2048 \times 2048$  points, respectively. Time step for the Karma model was taken as  $dt = 0.025 \text{ ms}$  and we solve the system for a total of 25 seconds. Space discretization for the Karma model was taken as  $dx = dy = 0.025 \mu\text{m}$ . For the Beeler-Reuter model, the time step considered was  $dt = 0.01 \text{ ms}$  and we solve the system for a total of 10 seconds. Space discretization for the Beeler-Reuter was  $dx = dy = 0.02 \mu\text{m}$ . Finally, for the ten Tusscher model the time step considered was  $dt = 0.02 \text{ ms}$  and we solve the system for a total of 20 seconds. Space discretization for the ten Tusscher was  $dx = dy = 0.02 \mu\text{m}$ .

### 6.1 OpenAcc: Comparing C and Fortran

We have used the PGI compiler along with C and Fortran languages using OpenAcc directives to use the GPU's. In this particular case, we have used three distinct graphic cards, GTX-770, GTX-Titan Black and K40, whose (main) specifications we present on table (6.1).

Table 6.1: Hardware Specifications of the GPUs used in this study.

Specifications	K40	TITAN	GTX 770
Compute Capability	3.5	3.5	3.0
Clock Speed (GHz)	0.875	0.889	1.046
Memory (GB)	12	6.14	2.048
Float (TFLOPS)	4.29	5.1	3.2
Double (TFLOPS)	1.43	1.3	
CUDA Cores	2880	2880	1536
Bandwidth (GB/s)	288	336	224.3
Power (W)	235	250	230
Price (USD)	\$3870	\$1000	\$150

OpenACC is a new programming standard developed by the Portland Group (PGI), Cray and NVIDIA. It allows programmers to easily access the parallel processing capabilities of a GPU without the need to use alternative programming languages such as CUDA or OpenCL. In this directive-based approach, simple code annotations known as pragmas are added to existing C/C++ and Fortran programs to alert the compiler about code blocks or loops that it should try to map onto the GPU for parallel execution. Many programmers have become familiar with these annotations through the use of the OpenMP—an API that enables codes to utilize multi-core, shared memory CPUs. In order to support GPU coprocessors with separate memory spaces, OpenACC pragmas can annotate data placement and transfer as well as loop and block parallelism.

One of the main advantages of using OpenACC is portability. Once a program is running on a PC, that same program will do so on another one without additional tuning or set up, even when there is another version of the compiler and toolkit that it uses.

Another good advantage of OpenACC is its compatibility with other GPU languages and libraries. This means that it is possible to call optimized functions(CUFFT, CUBLAS, CUSPARSE) from within our codes.

The basic idea when using OpenACC can be described as follows. After defining the variables that we are going to use for our program, we upload them to the graphic card, then perform the necessary calculations. Followed by that, we retrieve the data from the card. Each and every one of these steps are done by a single directive command.

We have run some simulations of the BR and Ten Tusscher models on a grid of

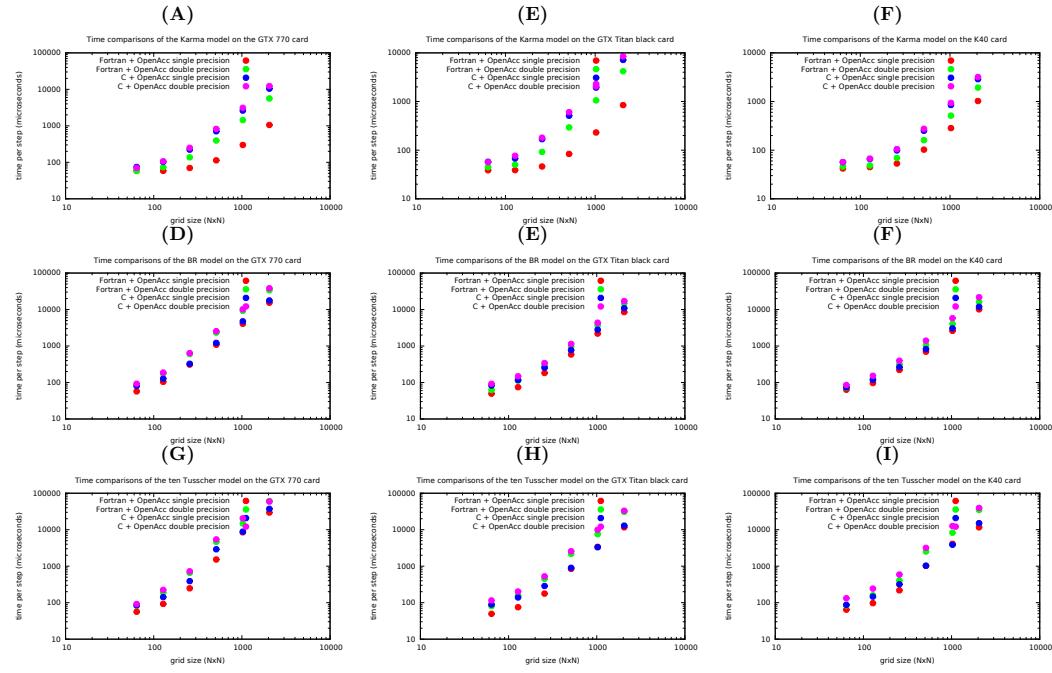


Figure 6.1: Frames (A), (B) and (C), (D), (E) and (F), (G), (H) and (I): Implementation of the Karma model, Beeler-Reuter model and ten Tusscher model on the GTX-770, GTX-Titan Black and K40 GPU's, respectively. No look-up tables were used to optimize computationally expensive functions.

$1024 \times 1024$  using only the CPU to give us a major scope of the benefits of using this model of programming. The BR model took around 34.5 hours to complete whereas the Ten Tusscher model it took around 101 hours.

Just to give us a rough idea of the time speed up that we can get by simply adding a couple of directives to our serial code, the simulations of the BR and Ten Tusscher model on a grid of  $1024 \times 1024$  now took, on the card with the lowest specifications(GTX-770), around 2.8 and 5.6 hours respectively, with the discretizations discussed at the beginning of the present chapter.

The results obtained using C and OpenAcc are summarized in tables 6.2-6.6. Tables 6.7-6.11 contain the results of the implementations using Fortran and OpenAcc.

For a clearer and better understanding of these results, we have separated the information. We show some plots with the corresponding results summarized on tables 6.2-6.11 for the comparisons between the C and Fortran implementations of the models in the three different graphic processing units. Figure 6.1 shows the results obtained for the case when the simulations were carried out without look up tables to optimize computationally expensive functions. The left, middle and right columns in Figure 6.1 show the results of the simulations implemented using a GTX-770, GTX-Titan Black and K40 graphic card, respectively. Each row in Figure 6.1 correspond to a different cardiac model. The top row correspond to the Karma

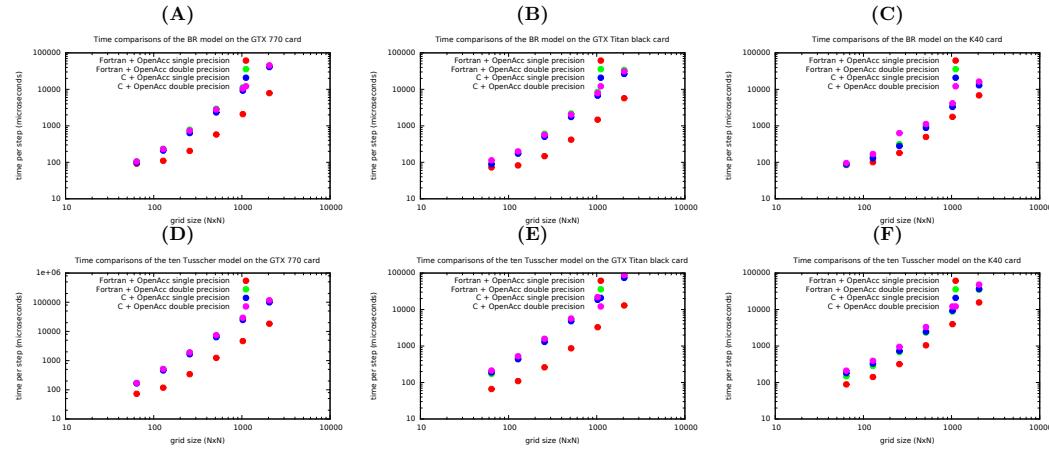


Figure 6.2: Frames (A), (B) and (C), (D), (E) and (F): Implementation of the Beeler-Reuter model and ten Tusscher model on the GTX-770, GTX-Titan Black and K40 GPU's, respectively. Look up tables were used to optimize computationally expensive functions.

model, whereas the middle and bottom rows correspond to the Beeler-Reuter and ten Tusscher models, respectively.

On the other hand, in Figure 6.2, we have plots of the results obtained for the case when look up tables were utilized to optimize functions. For this case, we omitted the Karma model. This due to the fact that using look up tables for this scenario does not show a representative improvement. The left, middle and right columns in Figure 6.2 show the corresponding implementations using a GTX-770, GTX-Titan Black and K40 graphic processing unit, respectively. The top and bottom rows in Figure 6.2 correspond to the Beeler-Reuter and ten Tusscher models, respectively.

## 6.2 Three dimensional studies

Now we present some results obtained when dealing with three dimensional domains. It is well known that the computational cost of performing simulations of cardiac models in three dimensional domains is very high. To show the impact and benefit of using graphic processing units to our advantage in this topic we present some of our simulation results when implementing phase field method along with Euler finite difference in parallel using Fortran and OpenAcc on a three dimensional heart structure. The size of the grid has dimensions  $250 \times 326 \times 298$  in  $x, y$  and  $z$ , respectively.

We have performed our studies for three distinct models, namely, Karma, BR and ten Tusscher in both single and double precision and on three different graphic cards, GTX 770, GTX Titan Black and K40. For the simulations in single precision, for each variable that we store we need  $4 \times 250 \times 326 \times 298$  bytes whereas for the simulations in double precision we need, for every stored variable,  $8 \times 250 \times 326 \times 298$  bytes. Because of this, we had not been able to run the simulation for the ten

Tusscher model in double precision on the GTX 770 graphic card. This is due to the fact that we need more than 20 variables to store and that translates into an amount of approximately 3.8 gb of memory ram, and we only have to our disposal around 3.5 gb out of the 4 gb that that graphic card has.

For the Karma model, we have run the codes for a  $100k$  iterations with  $dt = 0.025ms$  and  $dh = 0.025cm$ , which results is  $2.5s$  of real time simulation. In case of the BR model, we have done also  $100k$  iterations with  $dt = 0.01ms$  and  $dh = 0.02cm$ , then we have a simulation of  $1s$  of real time. Finally, for the ten Tusscher model, we have done  $100k$  iterations using  $dt = 0.02ms$  and  $dh = 0.02cm$  which results on  $2s$  of real time simulation. Table (6.12) contains the total time that it took for the whole simulation to finish, measured in seconds.

Table 6.2: Simulation results on the Karma model. The results are time per step measured in microseconds. The numerical parameters used in the simulations for this models were  $dt = 0.025$  and  $dx = .025$  for a total of  $3 \times 10^6$  iterations.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	74.73	70.09	57.81	56.52	56.77	56.92
$128 \times 128$	102.93	106.76	67.5	76.80	65.75	66.93
$256 \times 256$	222.86	250.23	168.2	180.7	98.75	105.4
$512 \times 512$	708.63	824.82	510.8	602.83	250.86	275.49
$1024 \times 1024$	2620.29	3131.03	1933.5	2285.62	848.57	937.89
$2048 \times 2048$	10471.72	12348.9	7155.615	8456.79	2885.15	3188.82

Table 6.3: Simulation results on the BR model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.01$ ,  $dx = 0.02$  using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	80.29	92.62	82.71	92.42	73.55	85.22
$128 \times 128$	127.37	187.37	116.43	147.87	118.3	152.99
$256 \times 256$	326.10	639.68	253.65	341.59	263.87	396.33
$512 \times 512$	1217.47	2552.28	771.86	1141.6	826.48	1405.67
$1024 \times 1024$	4755.12	10133.88	2806	4359.84	3051.69	5774.71
$2048 \times 2048$	17715.81	38279.65	10903.79	17063.05	11884.2	21730.25

Table 6.4: Simulation results on the BR model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.01$ ,  $dx = 0.02$  without using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	98.41	105.45	88.28	113.41	87.24	95.55
$128 \times 128$	209.8	230.99	172.97	199.07	128.63	169.3
$256 \times 256$	633.7	739.5	500.88	570.72	279.68	631.64
$512 \times 512$	2329.67	2798	1749.98	2042.06	879.74	1123.71
$1024 \times 1024$	9198.46	10539.3	6706.87	7890.58	3293.21	4158.1
$2048 \times 2048$	40761.67	45355.73	26453.76	31246.34	12917.14	16313.72

Table 6.5: Simulation results on the ten Tusscher model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.02$ ,  $dx = 0.02$  without using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	165.93	171.27	188.9	212.51	180.17	211.02
$128 \times 128$	460.6	517.33	435.39	523.82	330.39	394.71
$256 \times 256$	1650.7	1910.75	1296.73	1580.35	734.42	949.97
$512 \times 512$	6336.34	7462.14	4784.02	5658.22	2469.8	3318.83
$1024 \times 1024$	24716.45	29283.51	18450.88	21889.65	9281.38	12124.42
$2048 \times 2048$	99106.57	116715.6	73733.88	87616.46	36683.7	48139.78

Table 6.6: Simulation results on the ten Tusscher model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.02$ ,  $dx = 0.02$  using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	84.96	92.17	89.02	115.25	87.44	132.24
$128 \times 128$	142.08	223.87	139.24	202.22	146.4	243.2
$256 \times 256$	389.46	723.72	286.82	525.68	318.45	590.62
$512 \times 512$	2910.6	5418.22	901.89	2567.31	1031.55	3186.56
$1024 \times 1024$	8849.69	20494.99	3305.92	9959.81	3866.46	12621.52
$2048 \times 2048$	37290.7	59064.86	12926.2	33090.39	15111.09	39600.32

Table 6.7: Simulation results on the Karma model. Time measured in seconds. The numerical parameters used in the simulations for this model were  $dt = 0.025$  and  $dx = .025$  for a total of  $3 \times 10^6$  iterations.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	58.94	58.07	38.63	44.65	42.01	46.04
$128 \times 128$	58.41	76.36	38.87	50.42	44.83	48.43
$256 \times 256$	69.95	137.01	46.19	92.10	53.21	69.16
$512 \times 512$	113.35	396.94	83.65	293.89	102.60	160.31
$1024 \times 1024$	299.15	1140.34	231.11	1056.2	284.35	512.89
$2048 \times 2048$	1056.99	5635.71	843.57	4185.34	1021.9	1934.27

Table 6.8: Simulation results on the BR model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.01$ ,  $dx = 0.02$  without using look up tables. Time measured in seconds.

Grid Size	GTX 770		GTX Titan Black		K40	
	real	double	real	double	real	double
$64 \times 64$	91.79	105.78	72.45	92.52	84.95	85.09
$128 \times 128$	110.39	235.33	82.47	180.50	101.14	144.97
$256 \times 256$	205.15	785.85	147.34	606.76	180.56	314.90
$512 \times 512$	578.54	2944.81	417.45	2185.13	497.23	994.44
$1024 \times 1024$	2092.71	11292.59	1478.06	8553.45	1762.15	3723.76
$2048 \times 2048$	7894.85	44861.21	5732.70	34023.48	6834.39	14647.76

Table 6.9: Simulation results on the BR model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.01$ ,  $dx = 0.02$  using look up tables. Time measured in seconds.

Grid Size	GTX 770		GTX Titan Black		K40	
	real	double	real	double	real	double
$64 \times 64$	57.12	84.66	49.50	62.39	63.40	71.76
$128 \times 128$	104.3	174.96	74.28	113.08	96.54	124.28
$256 \times 256$	309.23	603.64	182.57	288.07	221.21	317.73
$512 \times 512$	1084.68	2331.72	586.42	973.88	699.11	1066.32
$1024 \times 1024$	4081.73	9268.73	2190.32	3715.56	2601.95	4056.51
$2048 \times 2048$	15419.43	33658.43	8514.27	14644.37	10116.82	16123.9

Table 6.10: Simulation results on the ten Tusscher model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.02$ ,  $dx = 0.02$  without using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	73.18	168.95	66.2	171.76	88.92	148.13
$128 \times 128$	117.95	509.73	109.39	464.65	141.55	283.44
$256 \times 256$	342.72	1868.03	260.79	1504.43	318.68	685.91
$512 \times 512$	1241.87	7127.14	864.33	5351.71	1043.58	2329.43
$1024 \times 1024$	4663.12	27947.23	3277.74	21181.64	3966.02	8853.77
$2048 \times 2048$	18403.42	110657.6	12949.82	84709.44	15647.84	34837.69

Table 6.11: Simulation results on the ten Tusscher model. We integrated the equations for a total of  $1 \times 10^6$  iterations with parameters  $dt = 0.02$ ,  $dx = 0.02$  using look up tables. Time measured in seconds.

	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
Grid Size	real	double	real	double	real	double
$64 \times 64$	56.43	82.09	49.52	80.84	64.34	84.39
$128 \times 128$	92.64	190.62	75.15	157.02	96.92	159.11
$256 \times 256$	247.34	654.88	177.66	448.54	218.06	405.85
$512 \times 512$	1525.66	4634.31	844.17	2169.01	1019.24	2525.70
$1024 \times 1024$	8479.92	14848.85	3393.84	7540.41	4102.21	8301.01
$2048 \times 2048$	29555.49	61301.08	11627.89	30724	11608.23	34917.61

Table 6.12: Simulation results on when implementing finite difference and phase field on a  $250 \times 326 \times 298$  grid on three different models and three distinct graphic cards. Time measured in seconds.

Model	<b>GTX 770</b>		<b>GTX Titan Black</b>		<b>K40</b>	
	real	double	real	double	real	double
Karma	1294.01	2549.7	641.24	1677.39	918.68	1282.12
BR	4089.46	7384.57	2019.86	3649.73	3212.38	4873.3
ten Tusscher	6279.25	out of memory	3706.72	8331.87	4986.25	9838.26



## Bibliography

- Aliev, R. R. and Panfilov, A. V. (1996). A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7(3):293–301.
- Allessie, M. A., Bonke, F. I., and Schopman, F. J. (1973). Circus movement in rabbit atrial muscle as a mechanism of tachycardia. *Circulation research*, 33(1):54–62.
- Alonso, S., Bär, M., and Panfilov, A. V. (2013). Negative tension of scroll wave filaments and turbulence in three-dimensional excitable media and application in cardiac dynamics. *Bulletin of mathematical biology*, 75(8):1351–1376.
- Alonso, S., Kähler, R., Mikhailov, A. S., and Sagués, F. (2004). Expanding scroll rings and negative tension turbulence in a model of excitable media. *Physical Review E*, 70(5):056201.
- Alonso, S. and Panfilov, A. V. (2007). Negative filament tension in the luo-rudy model of cardiac tissue. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 17(1):015102.
- Alonso, S. and Panfilov, A. V. (2008). Negative filament tension at high excitability in a model of cardiac tissue. *Physical review letters*, 100(21):218101.
- Azhand, A., Buchholz, R., Buchholz, F., Totz, J., and Engel, H. (2014). No-flux boundaries stabilize scroll rings in excitable media with negative filament tension. *arXiv preprint arXiv:1401.8119*.
- Aziz, I., Al-Fhaid, A., Shah, A., et al. (2013). A numerical assessment of parabolic partial differential equations using haar and legendre wavelets. *Applied Mathematical Modelling*, 37(23):9455–9481.
- Bánsági Jr, T., Meyer, K. J., and Steinbock, O. (2008). Wave-pinned filaments of scroll waves. *The Journal of chemical physics*, 128(9):094503.
- Bär, M. and Or-Guil, M. (1999). Alternative scenarios of spiral breakup in a reaction-diffusion model with excitable and oscillatory dynamics. *Physical review letters*, 82(6):1160.
- Bar-Yoseph, P., Moses, E., Zrahia, U., and Yarin, A. L. (1995). Space-time spectral element methods for one-dimensional nonlinear advection-diffusion problems. *Journal of Computational Physics*, 119(1):62–74.
- Barillot, E. and Boissonade, J. (1993). Asymptotic pseudospectral method for reaction-diffusion systems. *The Journal of Physical Chemistry*, 97(8):1566–1570.

- Beeler, G. W. and Reuter, H. (1977). Reconstruction of the action potential of ventricular myocardial fibres. *The Journal of physiology*, 268(1):177–210.
- Benson, A. P. and Holden, A. V. (2005). Calcium oscillations and ectopic beats in virtual ventricular myocytes and tissues: Bifurcations, autorhythmicity and propagation. In *FIMH*, pages 304–313. Springer.
- Bernus, O., Wilders, R., Zemlin, C. W., Verschelde, H., and Panfilov, A. V. (2002). A computationally efficient electrophysiological model of human ventricular cells. *American Journal of Physiology-Heart and Circulatory Physiology*, 282(6):H2296–H2308.
- Biktashev, V., Holden, A., Mironov, S., Pertsov, A., and Zaitsev, A. (1999). Three-dimensional aspects of re-entry in experimental and numerical models of ventricular fibrillation. *International Journal of Bifurcation and Chaos*, 9(04):695–704.
- Biktashev, V., Holden, A., and Zhang, H. (1994). Tension of organizing filaments of scroll waves. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 347(1685):611–630.
- Boyd, J. P. (2001). *Chebyshev and Fourier spectral methods*. Courier Corporation.
- Brazhnik, P. K., Davydov, V. A., and Mikhailov, A. S. (1988). Kinematic approach to the description of autowave processes in active media. *Theoretical and Mathematical Physics*, 74(3):300–306.
- Bueno-Orovio, A., Cherry, E. M., and Fenton, F. H. (2008). Minimal model for human ventricular action potentials in tissue. *Journal of theoretical biology*, 253(3):544–560.
- Bueno-Orovio, A., Kay, D., and Burrage, K. (2014). Fourier spectral methods for fractional-in-space reaction-diffusion equations. *BIT Numerical Mathematics*, 54(4):937–954.
- Bueno-Orovio, A., Pérez-García, V. M., and Fenton, F. H. (2006). Spectral methods for partial differential equations in irregular domains: the spectral smoothed boundary method. *SIAM Journal on Scientific Computing*, 28(3):886–900.
- Burden, R. and Faires, J. (2011). Numerical analysis 9th edn (boston, ma: Brooks/cole).
- Cantrell, R. S. and Cosner, C. (2004). *Spatial ecology via reaction-diffusion equations*. John Wiley & Sons.
- Canuto, C. (1988). Spectral methods and a maximum principle. *Mathematics of Computation*, 51(184):615–629.
- Canuto, C., Hussaini, M. Y., Quarteroni, A. M., Thomas Jr, A., et al. (2012). *Spectral methods in fluid dynamics*. Springer Science & Business Media.

- Cherry, E. M., Greenside, H. S., and Henriquez, C. S. (2003). Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 13(3):853–865.
- Clayton, R., Bernus, O., Cherry, E., Dierckx, H., Fenton, F., Mirabella, L., Panfilov, A., Sachse, F. B., Seemann, G., and Zhang, H. (2011). Models of cardiac tissue electrophysiology: progress, challenges and open questions. *Progress in biophysics and molecular biology*, 104(1):22–48.
- Clayton, R. H. and Holden, A. V. (2003). Dynamics and interaction of filaments during reentry and fibrillation in mammalian virtual ventricular tissue. *International Journal of Bifurcation and Chaos*, 13(12):3733–3745.
- Dahlem, M. A. and Müller, S. C. (2004). Reaction-diffusion waves in neuronal tissue and the window of cortical excitability. *Annalen der Physik*, 13(7-8):442–449.
- Demongeot, J., Gaudart, J., Lontos, A., Mintsa, J., Promayon, E., and Rachdi, M. (2012). Zero-diffusion domains in reaction-diffusion morphogenetic and epidemiologic processes. *International Journal of Bifurcation and Chaos*, 22(02):1250028.
- Efimov, I. R., Krinsky, V. I., and Jalife, J. (1995). Dynamics of rotating vortices in the beeler-reuter model of cardiac tissue. *Chaos, Solitons & Fractals*, 5(3):513–526.
- Eilbeck, J. (1983). A collocation approach to the numerical calculation of simple gradients in reaction-diffusion systems. *Journal of Mathematical Biology*, 16(3):233–249.
- Ermakova, E. and Pertsov, A. (1986). Interaction of rotating spiral waves with a boundary. *Biofizika*, 31(5):855–861.
- Feng, W., Yu, P., Hu, S., Liu, Z.-K., Du, Q., and Chen, L.-Q. (2006). Spectral implementation of an adaptive moving mesh method for phase-field equations. *Journal of Computational Physics*, 220(1):498–510.
- Fenton, F. and Karma, A. (1998). Vortex dynamics in three-dimensional continuous myocardium with fiber rotation: Filament instability and fibrillation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(1):20–47.
- Fenton, F. H., Cherry, E. M., Hastings, H. M., and Evans, S. J. (2002). Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 12(3):852–892.
- Fenton, F. H., Cherry, E. M., Karma, A., and Rappel, W.-J. (2005). Modeling wave propagation in realistic heart geometries using the phase-field method. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 15(1):013502.
- Finlayson, B. and Scriven, L. (1966). The method of weighted residualsa review. *Appl. Mech. Rev*, 19(9):735–748.

- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466.
- Franzone, P. C., Guerri, L., Pennacchio, M., and Taccardi, B. (1998). Spread of excitation in 3-d models of the anisotropic cardiac tissue. ii. effects of fiber architecture and ventricular geometry. *Mathematical biosciences*, 147(2):131–171.
- Garzón, A., Grigoriev, R. O., and Fenton, F. H. (2009). Model-based control of cardiac alternans on a ring. *Physical Review E*, 80(2):021932.
- Gorelova, N. and Bureš, J. (1983). Spiral waves of spreading depression in the isolated chicken retina. *Developmental Neurobiology*, 14(5):353–363.
- Gottlieb, D. (1981). The stability of pseudospectral-chebyshev methods. *Mathematics of Computation*, 36(153):107–118.
- Gottlieb, D. and Lustman, L. (1983). The spectrum of the chebyshev collocation operator for the heat equation. *SIAM journal on numerical analysis*, 20(5):909–921.
- Gottlieb, D. and Orszag, S. A. (1977). *Numerical analysis of spectral methods: theory and applications*. SIAM.
- Hariharan, G. and Kannan, K. (2014). Review of wavelet methods for the solution of reaction–diffusion problems in science and engineering. *Applied Mathematical Modelling*, 38(3):799–813.
- Harrild, D. M., Penland, R. C., and Henriquez, C. S. (2000). A flexible method for simulating cardiac conduction in three-dimensional complex geometries. *Journal of electrocardiology*, 33(3):241–251.
- Heidenreich, E., Rodriguez, J., Gaspar, F., and Doblaré, M. (2008). Fourth-order compact schemes with adaptive time step for monodomain reaction–diffusion equations. *Journal of Computational and Applied Mathematics*, 216(1):39–55.
- Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544.
- Ikeda, T., Yashima, M., Uchida, T., Hough, D., Fishbein, M. C., Mandel, W. J., Chen, P.-S., and Karagueuzian, H. S. (1997). Attachment of meandering reentrant wave fronts to anatomic obstacles in the atrium. *Circulation Research*, 81(5):753–764.
- Isomura, A., Hörning, M., Agladze, K., and Yoshikawa, K. (2008). Eliminating spiral waves pinned to an anatomical obstacle in cardiac myocytes by high-frequency stimuli. *Physical Review E*, 78(6):066216.
- Jalife, J. (2003). Rotors and spiral waves in atrial fibrillation. *Journal of cardiovascular electrophysiology*, 14(7):776–780.

- Jiménez, Z. A. and Steinbock, O. (2012). Stationary vortex loops induced by filament interaction and local pinning in a chemical reaction-diffusion system. *Physical review letters*, 109(9):098301.
- Jones, W. B. and O'Brien, J. J. (1996). Pseudo-spectral methods and linear instabilities in reaction-diffusion fronts. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 6(2):219–228.
- Karma, A. (1993). Spiral breakup in model equations of action potential propagation in cardiac tissue. *Physical review letters*, 71(7):1103.
- Karma, A. (1994). Electrical alternans and spiral wave breakup in cardiac tissue. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 4(3):461–472.
- Keener, J. and Bogar, K. (1998). A numerical method for the solution of the bidomain equations in cardiac tissue. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(1):234–241.
- Keener, J. and Sneyd, J. (2010). *Mathematical physiology: I: cellular physiology*. Springer Science & Business Media.
- Keener, J. P. (1980). Waves in excitable media. *SIAM Journal on Applied Mathematics*, 39(3):528–548.
- Keener, J. P. (1986). A geometrical theory for spiral waves in excitable media. *SIAM Journal on Applied Mathematics*, 46(6):1039–1056.
- Keener, J. P. (1988). The dynamics of three-dimensional scroll waves in excitable media. *Physica D: Nonlinear Phenomena*, 31(2):269–276.
- Keener, J. P. and Tyson, J. J. (1990). Helical and circular scroll wave filaments. *Physica D: Nonlinear Phenomena*, 44(1-2):191–202.
- Keener, J. P. and Tyson, J. J. (1992). The dynamics of scroll waves in excitable media. *SIAM review*, 34(1):1–39.
- Korczak, K. Z. and Patera, A. T. (1986). An isoparametric spectral element method for solution of the navier-stokes equations in complex geometry. *Journal of Computational Physics*, 62(2):361–382.
- Krinsky, V. (1978). Mathematical models of cardiac arrhythmias (spiral waves). *Pharmacology & Therapeutics. Part B: General and Systematic Pharmacology*, 3(4):539–555.
- Lansdell, B., Ford, K., and Kutz, J. N. (2014). A reaction-diffusion model of cholinergic retinal waves. *PLoS computational biology*, 10(12):e1003953.
- LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM.
- Li, X. and Xu, C. (2009). A space-time spectral method for the time fractional diffusion equation. *SIAM Journal on Numerical Analysis*, 47(3):2108–2131.

- Linge, S., Sundnes, J., Hanslien, M., Lines, G. T., and Tveito, A. (2009). Numerical solution of the bidomain equations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1895):1931–1950.
- Lui, S. (2009). Spectral domain embedding for elliptic pdes in complex domains. *Journal of Computational and Applied Mathematics*, 225(2):541–557.
- Luo, C.-h. and Rudy, Y. (1994). A dynamic model of the cardiac ventricular action potential. i. simulations of ionic currents and concentration changes. *Circulation research*, 74(6):1071–1096.
- Mikhailov, A., Davydov, V., and Zykov, V. (1994). Complex dynamics of spiral waves and motion of curves. *Physica D: Nonlinear Phenomena*, 70(1-2):1–39.
- Mikhailov, A., Panfilov, A., and Rudenko, A. (1985). Twisted scroll waves in active three-dimensional media. *Physics Letters A*, 109(5):246–250.
- Mogilner, A. and Edelstein-Keshet, L. (2002). Regulation of actin dynamics in rapidly moving cells: a quantitative analysis. *Biophysical journal*, 83(3):1237–1258.
- Mulholland, L. S., Huang, W., and Sloan, D. M. (1998). Pseudospectral solution of near-singular problems using numerical coordinate transformations based on adaptivity. *SIAM Journal on Scientific Computing*, 19(4):1261–1289.
- Murray, J. D. (2002a). Mathematical biology. i: An introduction. ii: Spatial models and biomedical applications. *Interdisciplinary Applied Mathematics*, 17.
- Murray, J. D. (2002b). Mathematical biology. i, volume 17 of *interdisciplinary applied mathematics*.
- Nanthakumar, K., Plumb, V. J., Epstein, A. E., Veenhuyzen, G. D., Link, D., and Kay, G. N. (2004). Resumption of electrical conduction in previously isolated pulmonary veins. *Circulation*, 109(10):1226–1229.
- Nielsen, P., Le Grice, I., Smaill, B., and Hunter, P. (1991). Mathematical model of geometry and fibrous structure of the heart. *American Journal of Physiology-Heart and Circulatory Physiology*, 260(4):H1365–H1378.
- Nolasco, J. and Dahlen, R. W. (1968). A graphic method for the study of alternation in cardiac action potentials. *Journal of applied physiology*, 25(2):191–196.
- Olmos, D. (2010). Reflection and attachment of spirals at obstacles for the fitzhugh-nagumo and beeler-reuter models. *Physical Review E*, 81(4):041924.
- Olmos, D. and Shizgal, B. D. (2006). A pseudospectral method of solution of fisher's equation. *Journal of Computational and Applied Mathematics*, 193(1):219–242.
- Olmos, D. and Shizgal, B. D. (2009). Pseudospectral method of solution of the fitzhugh–nagumo equation. *Mathematics and Computers in Simulation*, 79(7):2258–2278.

- Olmos-Liceaga, D. and Segundo-Caballero, I. (2016). An implicit pseudospectral scheme to solve propagating fronts in reaction-diffusion equations. *Numerical Methods for Partial Differential Equations*, 32(1):86–105.
- Orszag, S. A. (1980). Spectral methods for problems in complex geometries. *Journal of Computational Physics*, 37(1):70–92.
- Otani, N. F. (2000). Computer modeling in cardiac electrophysiology. *Journal of Computational Physics*, 161(1):21–34.
- Ozisik, M. (1993). Heat conduction, john willey & sons. Inc., New York.
- Panfilov, A. and Rudenko, A. (1987). Two regimes of the scroll ring drift in the three-dimensional active media. *Physica D: Nonlinear Phenomena*, 28(1-2):215–218.
- Panfilov, A. and Winfree, A. (1985). Dynamical simulations of twisted scroll rings in three-dimensional excitable media. *Physica D: Nonlinear Phenomena*, 17(3):323–330.
- Panfilov, A. V. (1998). Spiral breakup as a model of ventricular fibrillation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(1):57–64.
- Panfilov, A. V. and Keener, J. P. (1993). Effects of high frequency stimulation on cardiac tissue with an inexcitable obstacle. *Journal of theoretical biology*, 163(4):439–448.
- Peñaranda, A., Cantalapiedra, I. R., Bragard, J., and Echebarria, B. (2012). Cardiac dynamics: a simplified model for action potential propagation. *Theoretical Biology and Medical Modelling*, 9(1):50.
- Pertsov, A., Ermakova, E., and Panfilov, A. (1984). Rotating spiral waves in a modified fitz-hugh-nagumo model. *Physica D: Nonlinear Phenomena*, 14(1):117–124.
- Pertsov, A., Vinson, M., and Müller, S. C. (1993). Three-dimensional reconstruction of organizing centers in excitable chemical media. *Physica D: Nonlinear Phenomena*, 63(1-2):233–240.
- Peyret, R. (2002). Chebyshev method. In *Spectral Methods for Incompressible Viscous Flow*, pages 39–100. Springer.
- Qu, Z. and Garfinkel, A. (1999). An advanced algorithm for solving partial differential equation in cardiac conduction. *IEEE Transactions on Biomedical Engineering*, 46(9):1166–1168.
- Ramirez, R. J., Nattel, S., and Courtemanche, M. (2000). Mathematical analysis of canine atrial action potentials: rate, regional factors, and electrical remodeling. *American Journal of Physiology-Heart and Circulatory Physiology*, 279(4):H1767–H1785.

- Reddy, S. C. and Trefethen, L. N. (1990). Lax-stability of fully discrete spectral methods via stability regions and pseudo-eigenvalues. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):147–164.
- Reutskiy, S. and Lin, J. (2017). A meshless radial basis function method for steady-state advection-diffusion-reaction equation in arbitrary 2d domains. *Engineering Analysis with Boundary Elements*, 79:49–61.
- Rodríguez-Padilla, J. and Olmos-Liceaga, D. (2017). Chebyshev multidomain pseudospectral method to solve cardiac wave equations with rotational anisotropy. *International Journal of Modeling, Simulation, and Scientific Computing*, page 1850025.
- Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.
- Sabetghadam, F., Sharafatmandjoor, S., and Norouzi, F. (2009). Fourier spectral embedded boundary solution of the poisssons and laplace equations with dirichlet boundary conditions. *Journal of Computational Physics*, 228(1):55–74.
- Sato, D., Xie, Y., Weiss, J. N., Qu, Z., Garfinkel, A., and Sanderson, A. R. (2009). Acceleration of cardiac tissue simulation with graphic processing units. *Medical & biological engineering & computing*, 47(9):1011–1015.
- Setaro, J. F., Soufer, R., Remetz, M. S., Perlmutter, R. A., and Zaret, B. L. (1992). Long-term outcome in patients with congestive heart failure and intact systolic left ventricular performance. *The American journal of cardiology*, 69(14):1212–1216.
- Shizgal, B. D. (2002). Spectral methods based on nonclassical basis functions: the advection–diffusion equation. *Computers & fluids*, 31(4):825–843.
- Skaggs, W., Lugosi, E., and Winfree, E. (1988). Stable vortex rings of excitation in neuroelectric media. *IEEE transactions on circuits and systems*, 35(7):784–787.
- Soto, D. A. L. (2011). Interacción de ondas en espiral y obstáculos en medios excitables con la ecuación de fitzhugh-nagumo. Master’s thesis, UNIVERSIDAD DE SONORA.
- Streeter, D. D. (1979). Gross morphology and fiber geometry of the heart. *Handbook of physiology*, pages 61–112.
- Tal-Ezer, H. (1989). Spectral methods in time for parabolic problems. *SIAM Journal on Numerical Analysis*, 26(1):1–11.
- Ten Tusscher, K., Noble, D., Noble, P., and Panfilov, A. V. (2004). A model for human ventricular tissue. *American Journal of Physiology-Heart and Circulatory Physiology*, 286(4):H1573–H1589.
- Ten Tusscher, K. H. and Panfilov, A. V. (2006). Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):H1088–H1100.

- Thomas, C. E. (1957). The muscular architecture of the ventricles of hog and dog hearts. *Developmental Dynamics*, 101(1):17–57.
- Trayanova, N., Constantino, J., Ashihara, T., and Plank, G. (2011). Modeling defibrillation of the heart: approaches and insights. *IEEE reviews in biomedical engineering*, 4:89–102.
- Trefethen, L. N. (1996). Finite difference and spectral methods for ordinary and partial differential equations.
- Verschelde, H., Dierckx, H., and Bernus, O. (2007). Covariant stringlike dynamics of scroll wave filaments in anisotropic cardiac tissue. *Physical Review Letters*, 99(16):168104.
- Vigmond, E., Dos Santos, R. W., Prassl, A., Deo, M., and Plank, G. (2008). Solvers for the cardiac bidomain equations. *Progress in biophysics and molecular biology*, 96(1):3–18.
- Vinson, M., Mironov, S., Mulvey, S., and Pertsov, A. (1997). Control of spatial orientation and lifetime of scroll rings in excitable media. *Nature*, 386(6624):477.
- Virag, N., Jacquemet, V., Henriquez, C., Zozor, S., Blanc, O., Vesin, J.-M., Pruvot, E., and Kappenberger, L. (2002). Study of atrial arrhythmias in a computer model based on magnetic resonance images of human atria. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 12(3):754–763.
- Volpert, V. and Petrovskii, S. (2009). Reaction–diffusion waves in biology. *Physics of life reviews*, 6(4):267–310.
- Watanabe, M. A., Fenton, F. H., Evans, S. J., Hastings, H. M., and Karma, A. (2001). Mechanisms for discordant alternans. *Journal of cardiovascular electrophysiology*, 12(2):196–206.
- Wiener, N. (1946). The mathematical formulation of problem of conduction of impulses in a network of connected excitable elements, specially in cardiac muscle. *Arch Inst Cardiol Mex*, 16:205–265.
- Winfree, A. (1994). Electrical turbulence in three-dimensional heart muscle. *SCIENCE-NEW YORK THEN WASHINGTON-*, pages 1003–1003.
- Winfree, A. T. (1984). The prehistory of the belousov-zhabotinsky oscillator. *J. Chem. Educ*, 61(8):661.
- Woltering, M. and Markus, M. (2002). Turbulence control by wave splitting in excitable media. *Physics Letters A*, 297(5):363–369.
- Ying, W., Rose, D. J., and Henriquez, C. S. (2008). Efficient fully implicit time integration methods for modeling cardiac dynamics. *IEEE Transactions on Biomedical Engineering*, 55(12):2701–2711.

- Zaritski, R. M. and Pertsov, A. M. (2002). Stable spiral structures and their interaction in two-dimensional excitable media. *Physical Review E*, 66(6):066120.
- Zayernouri, M. and Karniadakis, G. E. (2014). Fractional spectral collocation method. *SIAM Journal on Scientific Computing*, 36(1):A40–A62.
- Zhabotinsky, A. M. (1964). Periodical oxidation of malonic acid in solution (a study of the belousov reaction kinetics). *Biofizika*, 9:306–311.