



HÔPITAL DU SACRÉ-COEUR
DE MONTRÉAL



Master degree in Mathematical Engineering (Advanced Scientific Computing)

<http://mathematiques.univ-lille1.fr/Formation/>

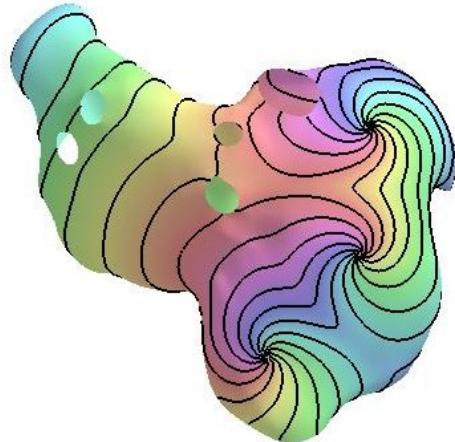
Internship report entitled:

An eikonal approach for the initiation of arrhythmias in biophysical models of the atria

Antoine Herlin

March 21st - September 9th 2011

Presentation June 20th 2011



Tutors:

Vincent Jacquemet

Université de Montréal

Hôpital du Sacré-Coeur de Montréal

Emmanuel Creusé

Laboratoire de Mathématiques Paul Painlevé UMR 8524

Université des Sciences et Technologies de Lille

Acknowledgments

I want to thank Vincent Jacquemet who is supervising my internship, for his explanations, comments and precious help from the beginning; and the Lausanne Heart Group (Ecole Polytechnique Fédérale de Lausanne) for sharing simulation tools and the atrial geometry. I also thank the time and energy devoted by Christophe Besse and Nouredine Melab (Université des Sciences et Technologies de Lille), professors in charge of the Master degree in Mathematical Engineering (Advanced Scientific Computing), and Emmanuel Creusé (Université des Sciences et Technologies de Lille) for his advice and comments, without whom I may not have applied to this internship. Finally, my family and friends who support and follow me even from across the Atlantic Ocean.

Contents

1 Internship context	6
1.1 Hôpital du Sacré-Coeur de Montréal (HSCM)	6
1.2 Research Center of HSCM	6
1.3 Heart Health	6
1.3.1 Patients	7
1.4 My supervisor, Vincent Jacquemet, Ph. D.	7
2 Background	8
2.1 Anatomy of the heart	8
2.2 Normal heart rhythm	9
2.3 Bioelectric function of cardiac cells	10
2.3.1 Cell membrane	10
2.3.2 Electric impulse and action potential	11
2.3.3 Ionic basis of the action potential	12
2.4 Atrial fibrillation	13
2.5 Project	15
3 Methods	17
3.1 Cardiac propagation model	17
3.2 Identification of phase singularities	19
3.3 Random distribution of phase singularities	21
3.3.1 Input and output variables	21
3.3.2 Calculation of the maximum empirical distance	22
3.3.3 Selection of nodes	23
3.3.4 Sign affectation	23
3.3.5 Circuits identification	24
3.4 Phase field reconstruction	26
3.4.1 Input and output variables	26
3.4.2 Reconstruction of the phase field	27
3.4.3 Wavelength	28
3.4.4 Regularization of the phase field	28
3.5 Initial condition for a monodomain model	30
3.6 Test cases	30
4 Results	32
4.1 Generation of phase maps	32
4.2 Monodomain simulations	37
5 Current work	40
6 Conclusion	42
A Matlab function <i>eik_distr_nodes.m</i>	45

B Matlab function *eik_spirals_initcond.m* **50**

C Matlab function *eik_spirals_regularize.m* **52**

Introduction

The heart is divided into two upper chambers called *atria* and two lower ones named *ventricles*. One of the most common cardiac arrhythmia (problem with the rate or rhythm of the heartbeat) is *atrial fibrillation* (AF), which can lead to severe complications such as heart failure and stroke.

Computer models of the atria have been developed to describe the propagation of electrical excitation waves in the heart muscle using nonlinear partial differential equations of the reaction-diffusion type. These models have contributed to the understanding of AF and arrhythmogenic processes. Reentry (single or multiple self-sustained electrical activation waves) was found to be a key element in these mechanisms. In order to test a therapeutic intervention, a diagnosis technique or a signal processing tool within an atrial model, many different, independent episodes of simulated AF have to be initiated. The project consists in generating initial conditions for simulated AF initiation using a new approach (based on the eikonal-diffusion equation) in order to allow the creation of a database of simulated AF episodes with varying degrees of complexity.

After a short description of the hospital and the research center, this report starts with an overview of heart anatomy and the workings of this vital organ, in which electric impulse plays a major role. Then it focuses on cardiac cells, cell membrane and ionic exchanges that occur with each heart beat. The essential notions of transmembrane voltage and action potential are introduced. The report goes on with the cardiac propagation model used for AF simulation and how to use the output data in order to identify the reentries. Then, the process to generate activation maps to initiate simulations is explained, from the eikonal-diffusion equation to implemented Matlab codes and results. Images of AF simulation and presentation of current work complete this report, written at the halfway mark of the internship duration.

1 Internship context

1.1 Hôpital du Sacré-Coeur de Montréal (HSCM)

The HSCM is one of the major hospitals affiliated with the Université de Montréal. Listed today among the largest hospitals in Quebec, HSCM remains the only hospital to provide all medical and surgical care. Its mission is to provide general care and highly specialized care, teaching, research and technology assessment and intervention methods in health. The HSCM also provides regional hyperbaric medicine and a tertiary trauma center.

1.2 Research Center of HSCM

Renowned for its excellence in health research activities and education, the research center of HSCM is a place of creativity and knowledge transfer, it focuses specifically on the following themes:

- Genetics and epidemiology of renal diseases
- Neuroscience, Mental Health
- Heart Health
- Lung Health
- Trauma-Orthopedics-ICU-Emergency Medicine

1.3 Heart Health

Tertiary cardiology center, the Hospital was the first facility to perform surgery for cardiac arrhythmias in Quebec. Recognized internationally as a leading center in surgery of the aorta and great vessels, the Hospital is also the first institution to have acquired a surgical robot.

Cardiology service has two major disciplines, cardiology and cardiac surgery.

Diagnostic and therapeutic services offered to the public are:

- Coronarography and angioplasty (coronary dilation in the heart)
- Electrophysiology, pacemaker and defibrillator cardiac, ablation
- Stress test
- transthoracic echocardiography, transesophageal echocardiography, contrast and stress

- ECG → electrocardiography
- Holter
- Clinical risk factors (prevention) / clinical heart failure
- Treatment of arrhythmias
- Outpatient Clinics

1.3.1 Patients

For general and specialized care, patients come mainly from the Montreal area. The cardiac clinic can receive and take care of more than 2400 users annually.

Concerning specialized care, it comes largely from the Laurentides (St-Eustache, Ste-Agathe, L'Annonciation, Mont-Laurier, St-Jérôme, Lachute, Val-d'Or).

Five major diagnoses identified:

- Coronary Atherosclerosis
- Myocardial infarction
- Unstable angina
- Heart failure
- Arrhythmia

1.4 My supervisor, Vincent Jacquemet, Ph. D.

Vincent Jacquemet received the M.S. degree in physics in 2000 from the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, and the Ph.D. degree in biomedical engineering in 2004 from the Signal Processing Institute of EPFL. The topic of his thesis was the development of biophysical models of atrial fibrillation. He then worked as a postdoc researcher in the Lausanne Heart Group at EPFL. Between 2007 and 2009, he was with the Department of Biomedical Engineering at Duke University with a “fellowship for advanced researcher” awarded by the Swiss National Science Foundation. Since 2009, he is research assistant professor at Université de Montréal and Hôpital du Sacré-Coeur de Montréal. His research interests include complex dynamical systems, biophysical modeling, numerical simulation and signal processing.

2 Background

Basic cardiac electrophysiology notions, required to understand the framework of the project, are introduced in the present section.

2.1 Anatomy of the heart

The heart is a specialized muscle that contracts regularly and continuously, pumping blood to the body and the lungs. The pumping action is caused by a flow of electricity through the heart that repeats itself in a cycle.

The human heart has four chambers, two at the top (the atria) and two at the bottom (the ventricles). The atria are the receiving chambers and the ventricles are the discharging chambers. The pathway of blood through the human heart consists of a pulmonary circuit and a systemic circuit. Deoxygenated blood flows through the heart in one direction, entering through the vena cava into the right atrium and is pumped through the tricuspid valve into the right ventricle being pumped out through the pulmonary valve to the pulmonary arteries into the lungs. It returns from the lungs through the pulmonary veins to the left atrium where it is pumped through the mitral valve into the left ventricle before leaving through the aortic valve to the aorta. See Fig. 1 to follow the circuit of blood into the heart.

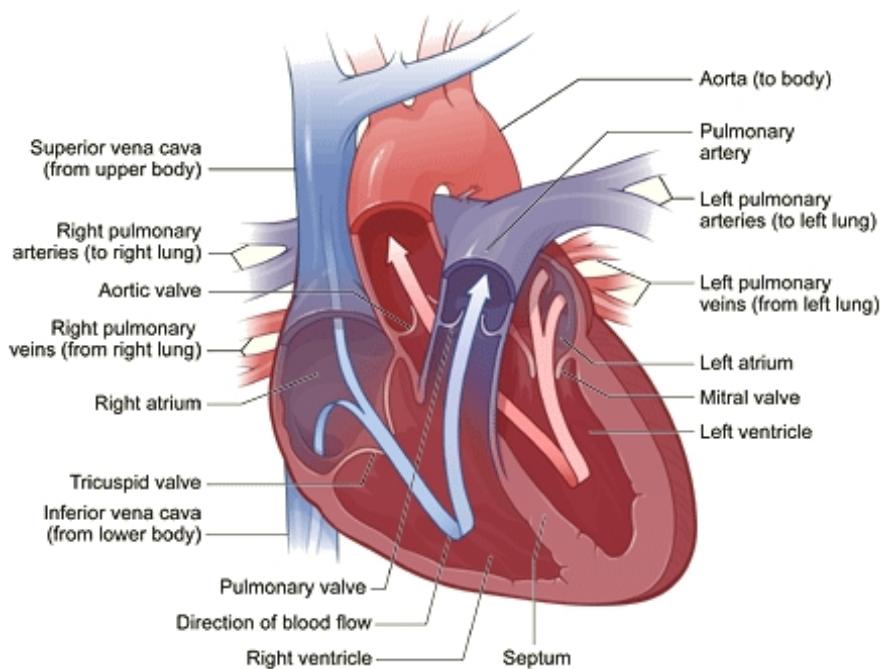
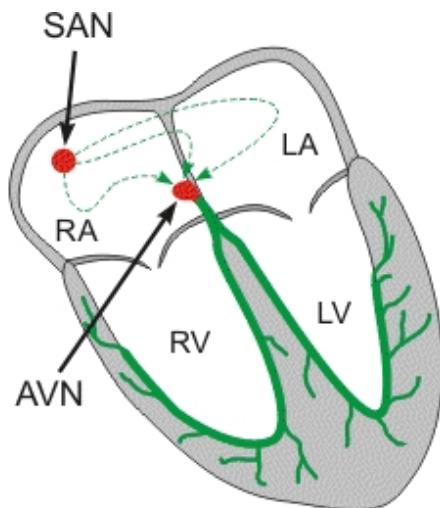


Figure 1: anatomy of the heart

2.2 Normal heart rhythm



SAN, sinoatrial node; AVN, atrio-ventricular node; RA, right atrium; LA, left atrium, RV, right ventricle; LV, left ventricle.

Figure 2: pathway of the electric impulse through the atria

The rhythm of the heart is normally determined by a pacemaker site called the sinoatrial (SA) node located in the posterior wall of the right atrium near the superior vena cava (see Fig. 2). The SA node consists of specialized cells that undergo spontaneous generation of action potentials (see 2.3.2) at a rate of 100-110 action potentials ("beats") per minute. This intrinsic rhythm is strongly influenced by autonomic nerves, with the vagus nerve being dominant over sympathetic influences at rest. This "vagal tone" brings the resting heart rate down to 60-80 beats/minute. The normal range for sinus rhythm is 60-100 beats/minute.

The sinus rhythm normally controls both atrial and ventricular rhythm. Action potentials generated by the SA node spread throughout the atria, depolarizing this tissue and causing atrial contraction. The impulse then travels into the ventricles via the atrioventricular node (AV node). Specialized conduction pathways (bundle branches and Purkinje fibers) within the ventricle rapidly conduct the wave of depolarization throughout the ventricles to elicit ventricular contraction. Therefore, normal cardiac rhythm is controlled by the pacemaker activity of the SA node.

2.3 Bioelectric function of cardiac cells

2.3.1 Cell membrane

Cardiac cells - as any other cells - are enclosed by a cell membrane whose thickness is about 7.5 - 10.0 nm. Its structure and composition resemble a soap-bubble film, since one of its major constituents, fatty acids, has that appearance. The fatty acids that constitute most of the cell membrane are called phosphoglycerides. A phosphoglyceride consists of phosphoric acid and fatty acids called glycerides. The head of this molecule, the phosphoglyceride, is hydrophilic (attracted to water). The fatty acids have tails consisting of hydrocarbon chains which are hydrophobic (repelled by water).

If fatty acid molecules are placed in water, they form little clumps, with the acid heads that are attracted to water on the outside, and the hydrocarbon tails that are repelled by water on the inside (follow on Fig. 3 the behavior of these molecules in water). If these molecules are very carefully placed on a water surface, they orient themselves so that all acid heads are in the water and all hydrocarbon tails protrude from it. If another layer of molecules were added and more water put on top, the hydrocarbon tails would line up with those from the first layer, to form a double (two molecules thick) layer. The acid heads would protrude into the water on each side and the hydrocarbons would fill the space between. This bilayer is the basic structure of the cell membrane.

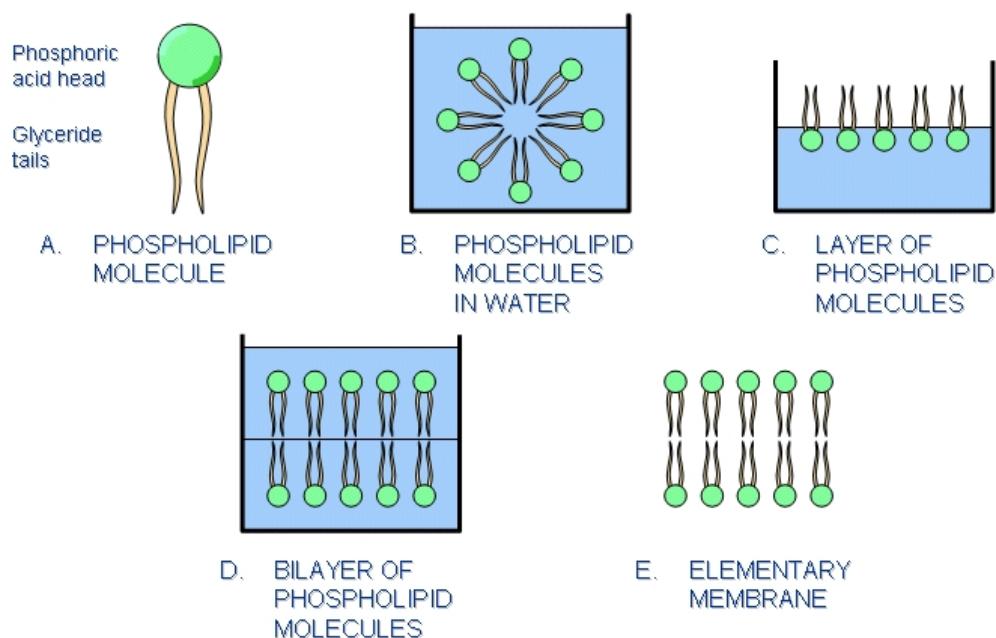


Figure 3: A sketch illustrating how the phosphoglyceride (or phospholipid) molecules behave in water [1].

From the bioelectric viewpoint, the ionic channels constitute an important part of

the cell membrane. These are macromolecular pores through which sodium, potassium, and chloride ions flow through the membrane. The flow of these ions forms the basis of bioelectric phenomena. Figure 4 illustrates the construction of a cell membrane.

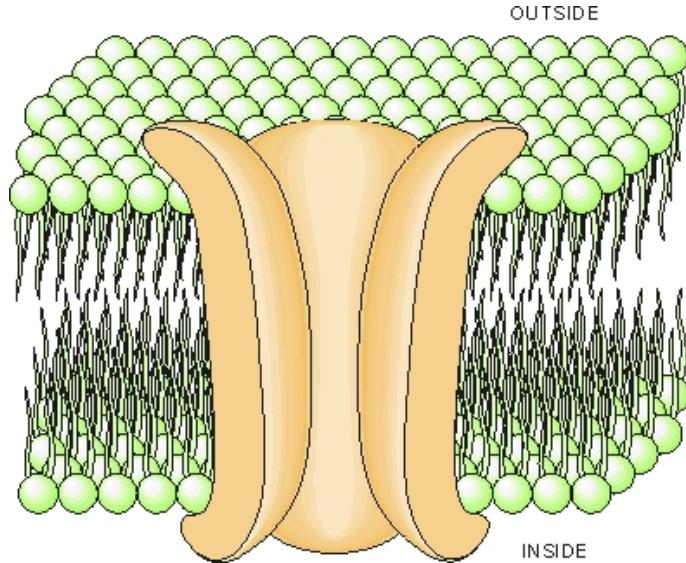


Figure 4: cell membrane with an ionic channel [1].

2.3.2 Electric impulse and action potential

Cardiac cells are excitable cells. The membrane voltage (transmembrane voltage) (V_m) of an excitable cell is defined as the potential at the inner surface (Φ_i) relative to that at the outer (Φ_o) surface of the membrane, i.e. $V_m = (\Phi_i) - (\Phi_o)$. This definition is independent of the cause of the potential, and whether the membrane voltage is constant, periodic, or non-periodic in behavior.

If a cardiac cell is stimulated, the transmembrane voltage necessarily changes. After stimulation the membrane voltage returns to its original resting value. If the membrane stimulus is insufficient to cause the transmembrane potential to reach the threshold, then the membrane will not activate. The response of the membrane to this kind of stimulus is essentially passive. If the excitatory stimulus is strong enough, the transmembrane potential reaches the threshold, and the membrane produces a characteristic electric impulse. This potential response follows a characteristic form regardless of the strength of the transthreshold stimulus. It is said that the action impulse of an activated membrane follows an all-or-nothing law. The electric recording of the electric impulse is called the *action potential*.

2.3.3 Ionic basis of the action potential

The concentration of sodium ions (Na^+) is about 10 times higher outside the membrane than inside, whereas the concentration of the potassium (K^+) ions is about 30 times higher inside as compared to outside. Follow on Fig. 5 the ionic exchanges through the cell membrane during an action potential.

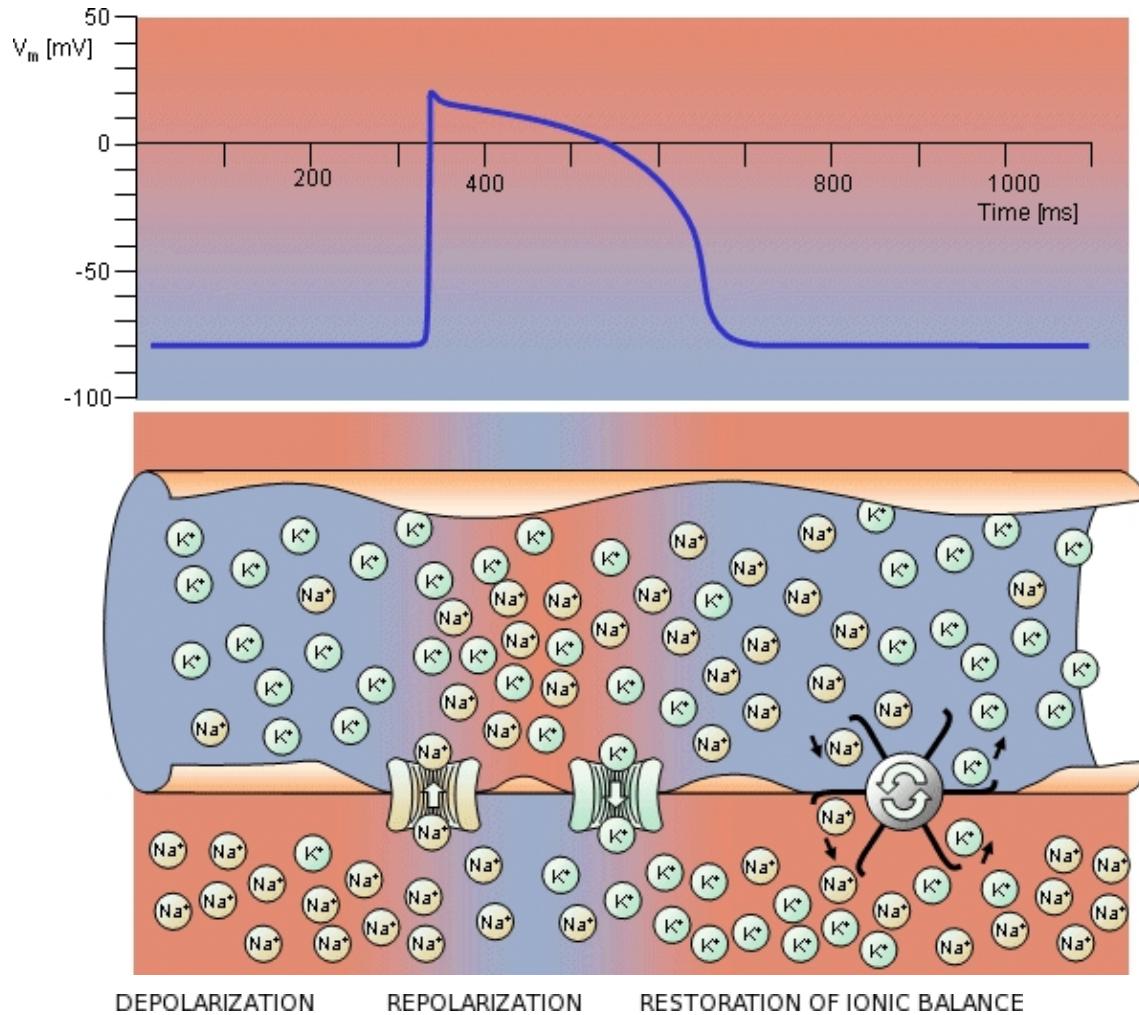


Figure 5: Electrophysiology of the cardiac muscle cell

When the membrane is stimulated so that the transmembrane potential rises about 20 mV and reaches the threshold - that is, when the membrane voltage changes from -70 mV to about -50 mV (these are illustrative and common numerical values) - the sodium and potassium ionic permeabilities of the membrane change. The sodium ion permeability increases very rapidly at first, allowing sodium ions to flow from outside to inside, increasing the positive charge on the inside. The inside reaches a potential of about +20 mV. After that, the more slowly increasing potassium ion permeability allows

potassium ions to flow from inside to outside, thus returning the intracellular potential to its resting value. These considerations apply to nerve and muscle cells in general.

The amplitude of the action potential is about 100 mV for both nerve and muscle. The duration of the cardiac muscle impulse is, however, two orders of magnitude longer than that in either nerve cell or skeletal muscle. A plateau phase follows cardiac depolarization, and thereafter repolarization takes place. As in the nerve cell, repolarization is a consequence of the outflow of potassium ions. The duration of the action impulse is about 300 ms.

Associated with the electric activation of a cardiac muscle cell is its mechanical contraction, which occurs a little later. An important distinction between cardiac muscle tissue and skeletal muscle is that in cardiac muscle, activation can propagate from one cell to another in any direction. As a result, the activation wavefronts are of rather complex shape. The only exception is the boundary between the atria and ventricles, which the activation wave normally cannot cross except along a special conduction system, since a nonconducting barrier of fibrous tissue is present.

2.4 Atrial fibrillation

Atrial fibrillation (AF) is the most common type of arrhythmia. An arrhythmia is a problem with the rate or rhythm of the heartbeat. During an arrhythmia, the heart can beat too quickly, too slowly, or with an irregular rhythm.

AF occurs when rapid, disorganized electrical signals cause the atria to fibrillate. The term "fibrillate" means to contract very rapidly and irregularly.

Often, people who have AF may not feel symptoms. However, even when not noticed, AF can increase the risk of stroke (up to 7 times that of the general population). In some people, AF can cause palpitations, fainting, chest pain, or congestive heart failure, particularly when the heart rhythm is very rapid. Stroke risk increases during AF because blood may pool and form clots in the poorly contracting atria and especially in the left atrial appendage.

AF may occur rarely or every now and then, or it may become a persistent or permanent heart rhythm lasting for years (Chronic AF). Risk increases with age, with 8% of people over 80 having AF.

In AF, the normal electrical impulses that are generated by the sinoatrial node are overwhelmed by *disorganized electrical impulses* that originate in the atria and pulmonary veins called *reentries* (see fig. 6, 7, 8), leading to conduction of irregular impulses to the ventricles that generate the heartbeat.

The abnormal electrical signals flood the atrio-ventricular node with electrical impulses. As a result, the ventricles also begin to beat very fast. However, the AV node cannot conduct the signals to the ventricles as fast as they arrive. So, even though the ventricles may be beating faster than normal, they are not beating as fast as the atria.

Thus, the atria and ventricles no longer beat in a coordinated way. This creates a fast and irregular heart rhythm. In AF, the ventricles may beat 100 to 175 times a minute, in contrast to the normal rate of 60 to 100 beats a minute.

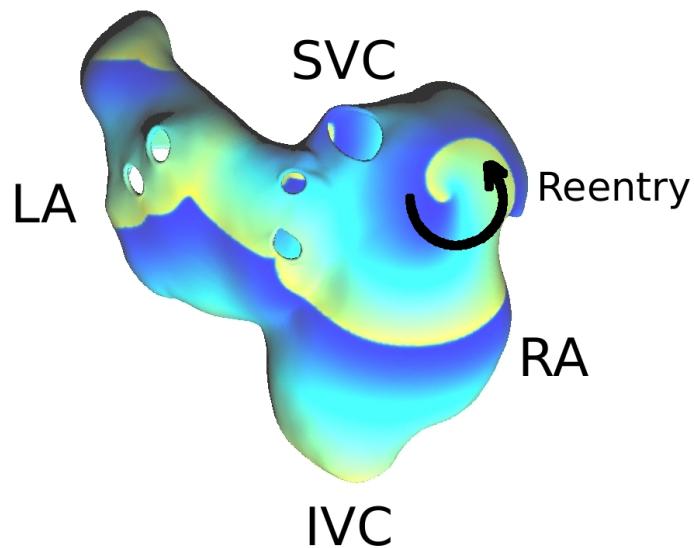


Figure 6: self-sustained activation wave (in yellow), here a *functional reentry* (spiral).
 LA: Left Atrium; RA: Right Atrium; SVC: Superior Vena Cava; IVC: Inferior Vena Cava.

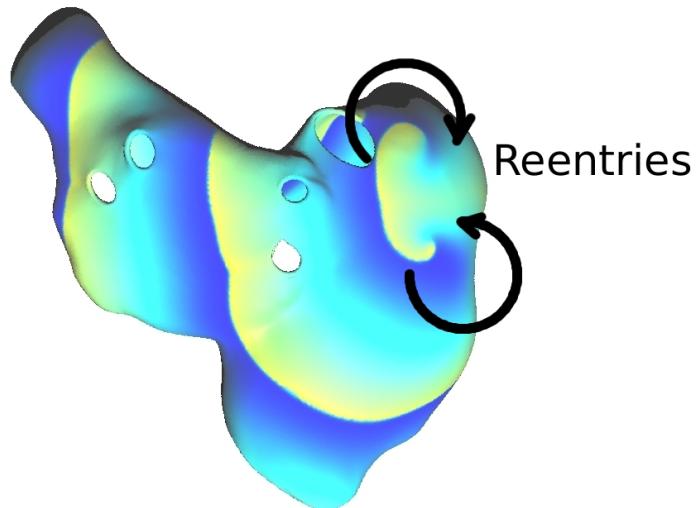


Figure 7: *functional reentries* located at the pivot point of a U-turn.

When this happens, blood is not pumped into the ventricles as well as it should be. Also, the amount of blood pumped out of the ventricles to the body is based on the randomness of the atrial beats.

The body may get rapid, small amounts of blood and occasional larger amounts of blood. The amount will depend on how much blood has flowed from the atria to the ventricles with each beat.

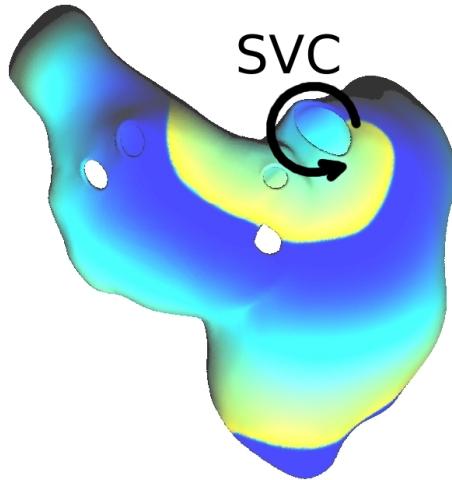


Figure 8: *anatomical reentry*, the activation wave rotates around the Superior Vena Cava.

Most of the symptoms of AF are related to how fast the heart is beating. Atrial fibrillation may be treated with medication which either slow the heart rate or revert the heart rhythm back to normal. People with AF are often given anticoagulants such as warfarin to protect them from stroke. Synchronized electrical cardioversion may also be used to convert AF to a normal heart rhythm. Surgical and catheter-based therapies may also be used to prevent recurrence of AF in certain individuals.

Currently, advanced surgical robotics can be used to create all of the lesions of the classic maze procedure; the maze procedure is the surgical ablation approach with the greatest long-term success in treating atrial fibrillation. The robotically-assisted maze procedure includes creation of lines of conduction block (scar tissue) that block the abnormal impulses that cause atrial fibrillation, enabling restoration of normal sinus rhythm. The lines of conduction block are created using cryotherapy (freezing) or radiofrequency energy. The success rate is approximately 80% to 90%, varying with patient characteristics. The research I am assisting with is conducted in the belief that computer models and numerical simulation can help improve our understanding of the underlying mechanisms [2].

2.5 Project

Computer models of the atria have been developed to describe the propagation of electrical excitation waves in the heart muscle using nonlinear partial differential equations of the reaction-diffusion type. These models have contributed to the understanding of AF by isolating and illustrating different mechanisms of AF and by investigating the arrhythmogenic processes in controlled setup types. Reentry (single or multiple self-sustained activation waves) was found to be a key element in these mechanisms.

In order to test a therapeutic intervention, a diagnosis technique or a signal processing tool within an atrial model, many different, independent episodes of simulated AF have to be initiated. Previous works enable to create an activation map to launch an AF simulation, using “obstacles” placed manually and then withdrawn on the computer model, but with a rather short range of possibilities [3]. The project consists in using a new approach for simulated AF initiation in which a random but plausible initial condition is generated with *a priori* knowledge about wavefront propagation. The creation of activation maps is decomposed in several tasks, fulfilled by Matlab functions implemented during the internship. First, determining the location of the reentries and their sense of rotation, then reconstructing the phase field solving a Laplace equation. The resulting phase field provides an inaccurate representation of a reentrant activation pattern, and iteratively solving the eikonal-diffusion equation enables to regularize it. Generation of initial conditions from phase fields for AF simulation was already implemented. Launching many AF simulations was naturally followed by data processing. The complexity of the dynamics, measured as the number of phase singularities, has to be controlled, in order to allow the creation of a database of simulated AF episodes with varying degrees of complexity. Such a database will eventually play a role in assisting the clinical diagnosis and treatment of atrial arrhythmias [2].

3 Methods

Atrial fibrillation simulations used during the internship rely on the monodomain theory. Construction of phase maps to initiate such simulations is based on the eikonal-diffusion equation.

3.1 Cardiac propagation model

The propagation of the cardiac impulse in the myocardium can be described by the evolution of the membrane potential field $V_m(\mathbf{x}, t)$. According to the monodomain theory, this evolution is governed by a reaction-diffusion equation [4]:

$$C_m \frac{\partial V_m}{\partial t} = \beta^{-1} \nabla \cdot \boldsymbol{\sigma} \nabla V_m - I_{\text{ion}}, \quad (1)$$

where C_m is the membrane capacitance per unit area of membrane, β is the area of membrane per unit volume, and $\boldsymbol{\sigma}$ is the (effective) conductivity tensor. The ionic current I_{ion} depends on V_m and on internal variables \mathbf{s} (intracellular ionic concentrations and channel gate states) that satisfy a system of ordinary differential equations $d\mathbf{s}/dt = F_{\mathbf{s}}(V_m, \mathbf{s})$. No-flux boundary condition is assumed, i.e. $\mathbf{n} \cdot \boldsymbol{\sigma} \nabla V_m = 0$ where \mathbf{n} is the unit vector normal to the boundary. An initial condition $V_m(\mathbf{x}, 0) = V_0(\mathbf{x})$ and $\mathbf{s}(\mathbf{x}, 0) = \mathbf{s}_0(\mathbf{x})$ has to be specified.

The ionic current I_{ion} is described by the Courtemanche *et al.* model [5] (Fig. 9):

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

with

- I_{Na} : Fast inward Na^+ current
- I_{K1} : Inward rectifier K^+ current
- I_{to} : Transient outward K^+ current
- I_{Kur} : Ultrarapid delayed rectifier K^+ current
- I_{Kr} : Rapid delayed rectifier K^+ current
- I_{Ks} : Slow delayed rectifier K^+ current
- $I_{Ca,L}$: L-type inward Ca^{2+} current
- $I_{p,Ca}$: Sarcoplasmic Ca^{2+} pump current
- I_{NaK} : $Na^+ - K^+$ pump current
- I_{NaCa} : Na^+ / Ca^{2+} exchanger current
- $I_{b,Na}$: Background Na^+ current

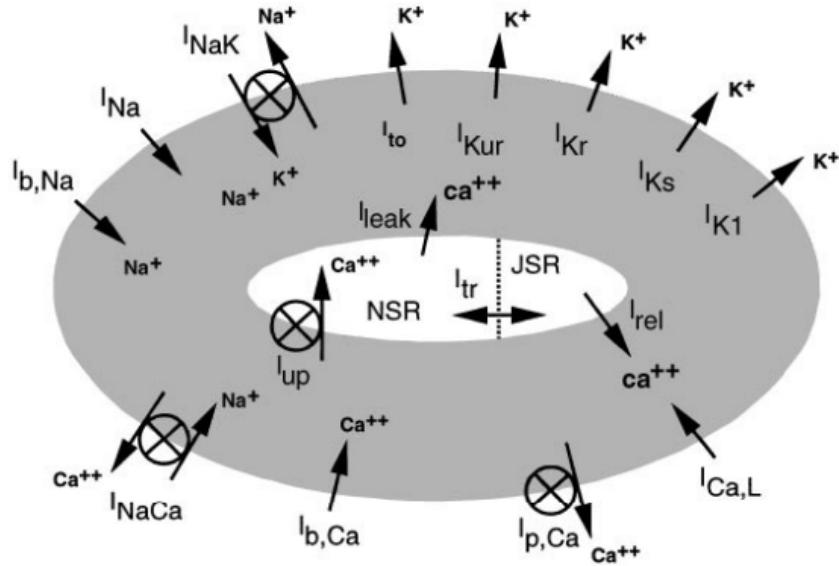


Figure 9: Schematic representation of currents, pumps, and exchangers included in model. Cell includes 3 intracellular compartments: myoplasm, sarcoplasmic reticulum (SR) release compartment [junctional SR (JSR)], and SR uptake compartment [network SR (NSR)].[5]

- $I_{b,Ca}$: Background Ca^{2+} current

To reproduce ion channel remodeling induced by chronic atrial fibrillation [6], channel maximum conductances are modified. Similarly to Kharche *et al.* [7], the L-type Ca^{2+} current (I_{CaL}) is reduced by 63% and the transient outward K^+ current (I_{to}) by 65% (data from human chronic AF atrial cells measured by whole cell patch clamp [8]). The inward rectifier K^+ current (I_{K1}) is increased by 73% (measurement near the resting potential in chronic AF patients [9]). As a result of these modifications, the action potential is triangular-shaped (see Fig. 10) and the effective refractory period at 60 bpm is about 160 ms. The *effective refractory period* is the period of time from the initiation of an action potential to the moment when it becomes possible to initiate a new action potential.

During the internship, in order to run AF simulations, simulation tools based on equation 1 were used. Numerical resolution relies on finite differences discretization. Spatial discretization is made of a 3D mesh with cubic elements ($dx = 0.033$ cm); time discretization is done with operator splitting reaction/diffusion ($dt_{split} = 0.1$ ms). For reaction the scheme is forward Euler with adaptive sub-time steps (for equations of the type $dy/dt = a(V_m)y + b(V_m)$, the analytical solution assuming $V_m = const$ is used instead). For diffusion a semi-implicit Crank-Nicholson scheme is used with $x/y/z$ operator splitting (only tridiagonal system solving) (for more information see [10]).

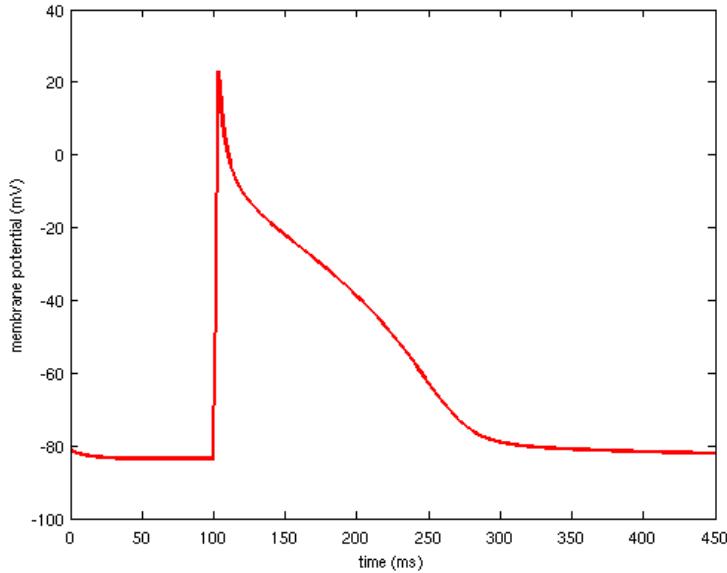


Figure 10: action potential after modification of channel maximum conductances

3.2 Identification of phase singularities

Phase singularity analysis provides a tool to quantify the complex spatio-temporal behavior observed in computer models or animal models of cardiac arrhythmia [11, 12]. It enables to count the number of simultaneous wavelets and identify the location of arrhythmogenic regions associated with reentries or wavebreaks. During a reentrant activation, a phase singularity is typically located at the center of the rotating wave or at the pivot point of a U-turn (different kinds of reentry, see fig. 6, 7 & 8 page 14). When the membrane potential field $V_m(\mathbf{x}, t)$ is known, the phase field θ is defined as:

$$\theta(\mathbf{x}, t) = \text{atan2}(V_m(\mathbf{x}, t + \tau_{\text{delay}}) - V^*, V_m(\mathbf{x}, t) - V^*) , \quad (2)$$

where atan2 is the four-quadrant inverse tangent; this two-argument function is a variation of the arctangent function. $\text{atan2}(y, x)$ computes the principal value of the argument function applied to the complex number $x + iy$: $\text{atan2}(y, x) = \text{Pr arg}(x + iy) = \text{Arg}(x + iy)$, that is $-\pi < \text{atan2}(y, x) \leq \pi$. τ_{delay} is the time delay for phase space reconstruction and (V^*, V^*) is the chosen origin of the phase space [12]. For the membrane model used here, we chose $\tau_{\text{delay}} = 5$ ms and $V^* = -55$ mV.

Since only thin-walled three-dimensional atrial models are considered here, the electrical activity will be analyzed through its manifestation on the epicardial surface (outer layer of heart tissue), like in most experimental optical mapping setups. In these two-dimensional cases, the topological charge q of a domain delimited by a closed curve Γ is

defined as the contour integral

$$q(\Gamma) = \frac{1}{2\pi} \oint_{\Gamma} \nabla \theta \cdot d\ell , \quad (3)$$

which gives an integer. The gradient takes into account the angular nature of θ (θ is defined modulo 2π). When Γ encircles a discontinuity point, a nonzero value for q may be obtained (typically ± 1). This situation corresponds to a phase singularity. The sign is associated with the sens of rotation of the reentrant activity (chirality).

Phase singularity location can be identified and tracked by successive application of this formula (for example on every triangle of a triangulated surface) [12]. The state at a given time can therefore be qualitatively described by the number n of phase singularities (see fig. 11), by their location \mathbf{x}_i and their topological charge q_i .

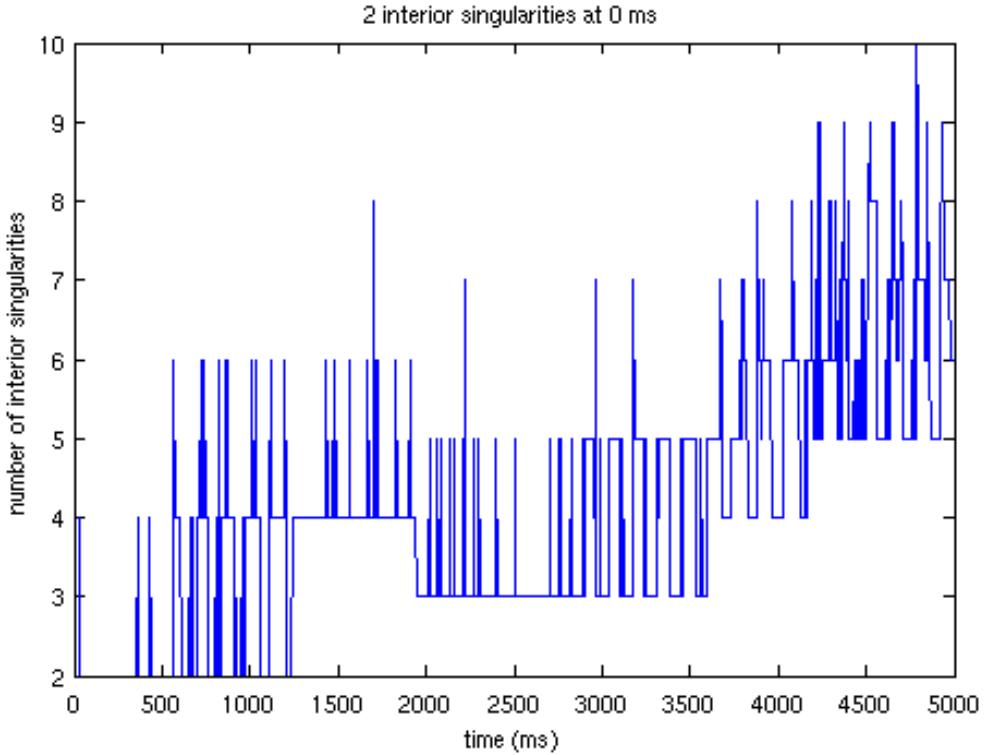


Figure 11: number of phase singularities during a 5000 ms simulation.

The question arises whether it is possible to construct a tissue state with n phase singularities with given location and topological charge. The next subsections address this problem and its application to the initiation of fibrillatory activity.

3.3 Random distribution of phase singularities

The objective is to distribute n points on a coarse triangulated surface ($N = 800\text{--}2,000$ nodes) and assign their topological charge ± 1 . The locations of these points will be the ones of phase singularities, and the signs will stand for the sens of rotation of the electric impulse. The full Matlab function implemented during the internship is in appendix A. Let us explain the approach and parts of Matlab codes.

For each configuration $\{\mathbf{x}_i, q_i\}$, the minimum distance d and the energy E are defined as:

$$d(\{\mathbf{x}_i\}) = \min_{i \neq j} \text{dist}(\mathbf{x}_i, \mathbf{x}_j) \quad (4)$$

$$E(\{\mathbf{x}_i, q_i\}) = \sum_{i \neq j} \frac{q_i q_j}{\text{dist}(\mathbf{x}_i, \mathbf{x}_j)} . \quad (5)$$

The idea is to maximize d while minimizing E , so that the points are equally dispersed on the geometry. The distance $\text{dist}(\mathbf{x}, \mathbf{y})$ is the length of the shortest path between \mathbf{x} and \mathbf{y} on the surface [13]. The distance matrix (all-pairs shortest path) is computed using Johnson's algorithm [14] as implemented by David Gleich in the toolbox MatlabBGL (available at Matlab Central file exchange).

3.3.1 Input and output variables

```

function [nodes, signs, circuits] = eik_distr_nodes(VER,TRI,D,n,prc,N,meth)
% EIK_DISTR_NODES
% distribute phase singularities on a triangulated surface
%
% [nodes, signs] = eik_distr_nodes(D,n,prc);
% [nodes, signs] = eik_distr_nodes(D,n,prc,N,meth);
% [nodes, signs, circuits] = eik_distr_nodes(D,n,prc);
% [nodes, signs, circuits] = eik_distr_nodes(D,n,prc,N,meth);
%
% inputs:
%   VER: position of vertices
%   ITRI: indices of triangles
%   D: distance matrix of the mesh (tri_distmatrix.m)
%   n: number of nodes to be selected
%   prc: distance allowed from optimum (0<prc<1; 0=optimum; default=0.2)
%   N: number of iterations to estimate the optimum (default=1000)
%   meth: method ('dist' or 'energy', 'dist' by default)
%
% outputs:
%   nodes: indices of selected nodes

```

```
% signs:      topological charges of the nodes (= +/- 1)
% circuits:  circuits{i} = liste of nodes around nodes(i)
```

3.3.2 Calculation of the maximum empirical distance

First, a point \mathbf{x}_1 is randomly selected, anywhere except on the anatomical boundaries (valves or veins boundaries). “boundary_nodes” is the list of nodes on the anatomical boundaries.

```
while 1
    first = ceil(size(D,1)*rand);
    if isempty(intersect(boundary_nodes,first))
        break
    end
end
```

The optimum minimum distance $d_{\text{opt}}(\mathbf{x}_1, n)$ is estimated by taking the maximum of $d(\{\mathbf{x}_i\})$ with \mathbf{x}_1 fixed over a number (nb_editions) of random selections of the $n - 1$ remaining points (1,000 random selections by default).

```
for k=1:nb_editions

    %generation of (n-1) other nodes
    nodes = [first ; ceil(total_nb_nodes.*rand(n-1,1))];

    % calculation of the shortest distance between 2 nodes and the "energy":
    shortest_distance=inf;
    E=0;
    for i = 1:n
        for j = i+1:n
            distance=D(nodes(i),nodes(j));
            E = E + 1/distance;
            if distance<shortest_distance
                shortest_distance=distance;
            end
        end
    end
    results(k,1)=shortest_distance;
    results(k,2)=E;
end

% greatest shortest_distance
[greatest_shortest_distance, index] = max(results(:,1));
```

```

E_min=min(results(:,2));
nodes=matrix_of_nodes(index,:);
dist = greatest_shortest_distance;
energy = E_min;

```

3.3.3 Selection of nodes

The previous step gives a criteria to keep a certain random selection of nodes or not. Sets of $n - 1$ points are randomly selected until $d(\{\mathbf{x}_i\}) > (1 - \epsilon_d) d_{\text{opt}}(\mathbf{x}_1, n)$ where ϵ_d lies in the range 0.1 to 0.2. This ensures that the points are well spread over the surface.

3.3.4 Sign affectation

The topological charges are obtained by minimizing the energy, the positions $\{\mathbf{x}_i\}$ being fixed. The energy is computed for all configurations of the n points satisfying the constraint $\sum_i q_i = 0$ if n is even and $|\sum_i q_i| = 1$ if n is odd.

```

if mod(n,2)==0    % if n is EVEN
    signs = choose_signs(0);
else % if n is ODD
    signs = choose_signs(1);
end

function signs = choose_signs(k)
q=zeros(2^n,n);

```

All possible configurations of signs are generated :

```

for i=1:2^n
    bla= i;
    for j=1:n
        q(i,j)=mod(bla,2);
        bla = floor(bla/2);
        if q(i,j)==0
            q(i,j)=-1;
        end
    end
end

```

then only the ones with the right repartition of positive and negative signs are kept :

```

if abs(sum(q(i,:)))~=k
    q(i,:)=0;
end
end

```

```
q = q(any(q,2),:); % removes 0 rows
```

The energy is computed for all those configurations :

```
ener = zeros(size(q,1),1);
for i = 1:size(q,1)
    E=0;
    for j=1:n
        for k=j+1:n
            E=E+q(i,j)*q(i,k)/D(nodes(j),nodes(k));
        end
    end
    ener(i)=E;
end
```

The configuration with minimal energy is selected.

```
[signed_energy , index] = min(ener);
signs=q(index,:);
end
```

3.3.5 Circuits identification

Circuits of edges surrounding the points are identified.

```
function circuits = identify_circuits

NEW_TRI = TRI;
for i = 1 : size(TRI,1)
    for j = 1 : size(nodes,1)

        if (NEW_TRI(i,1)==nodes(j)) || (NEW_TRI(i,2)==nodes(j)) || ...
           (NEW_TRI(i,3)==nodes(j))
            NEW_TRI(i,:)=0;
        end
    end
end
NEW_TRI = NEW_TRI(any(NEW_TRI,2),:); % removes 0 rows
```

The holes in the new geometry are identified.

```
B = tri_find_boundary(NEW_TRI);
nb_boundaries = size(B,2);
```

The coordinates of the points are calculated...

```
coor=zeros(3,n);
for j=1:n
    coor(:,j)=VER(nodes(j),:)';
end
```

... and also the coordinates of the isobarycentre of the boundaries of the new geometry...

```
link_bdy_nodes=zeros(n,1);
isobarycentre_boundaries=zeros(3,nb_boundaries);
for i=1:nb_boundaries
    mmean = [ 0, 0, 0];
    for j=1:size(B{i},2)-1 % -1 because the last vertex is the same as the
        % first one
        mmean = mmean + VER(B{i}(j),:);
    end
    mmean = mmean ./ (size(B{i},2)-1);
    isobarycentre_boundaries(:,i)=mmean';
end
```

... then a comparison can be made in order to identify the circuits.

```
for i=1:n
    liste_distances=zeros(1,nb_boundaries);
    for j=1:nb_boundaries
        liste_distances(j)=
            sqrt((coor(1,i)-isobarycentre_boundaries(1,j))^2+...
            (coor(2,i)-isobarycentre_boundaries(2,j))^2+...
            (coor(3,i)-isobarycentre_boundaries(3,j))^2);
    end
    [mini,link_bdy_nodes(i)]=min(liste_distances);
end

circuits = [];
for i=1:n
    circuits{i} = B{link_bdy_nodes(i)};
end
```

Typically, the two nearest phase singularities will have opposite topological charges (illustration Fig. 12).

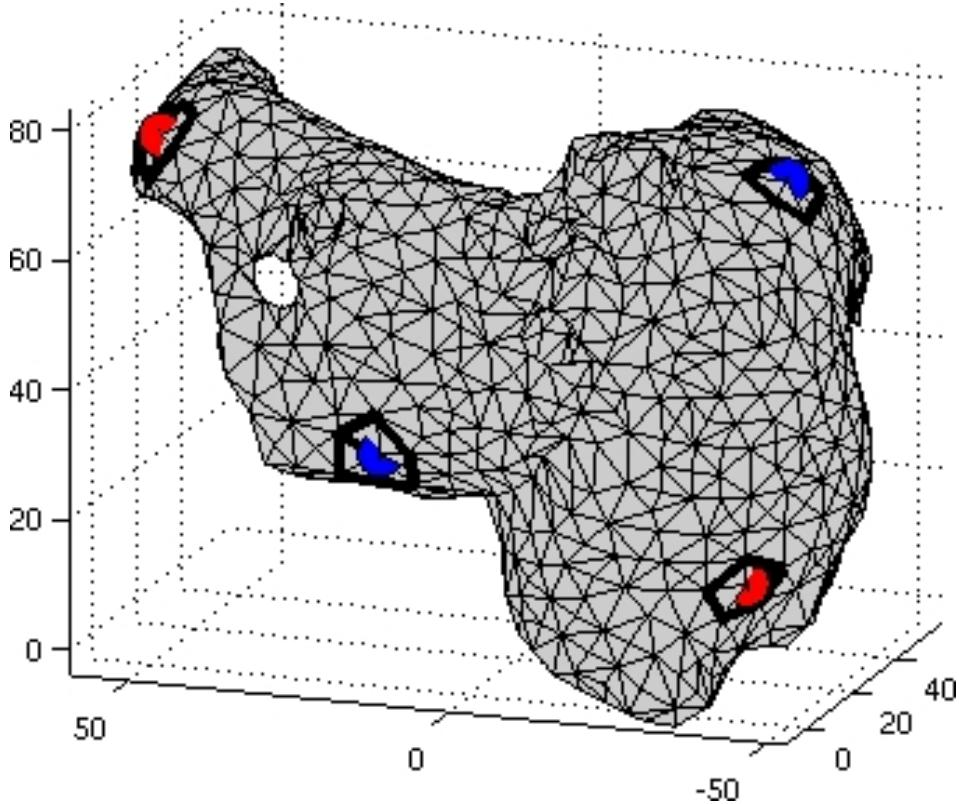


Figure 12: Random selection of 4 nodes on a coarse mesh, with positive (red) and negative (blue) signs, and the circuits surrounding them.

3.4 Phase field reconstruction

Interpolation of the phase field $\theta(\mathbf{x})$ from phase singularities configuration $\{\mathbf{x}_i, q_i\}$ is performed on an oriented triangular mesh of intermediate resolution ($N = 11,000$ to $17,000$ nodes; $\Delta x \approx 1$ mm). The small circuit formed by the triangles surrounding the point \mathbf{x}_i is denoted by Γ_i . Its orientation follows that of the surrounding triangles and its length L_i is of the order of 6 mm. To avoid a phase unwrapping problem, the complex phase field $\phi = \exp(i\theta)$ is used [15]. Its value is set to $\phi = \exp(iq_i\ell/L_i)$ on Γ_i , where ℓ is the curvilinear coordinate along the oriented curve Γ_i . Here are parts of the implemented code to make the phase field reconstruction. Full Matlab function is available in appendage B.

3.4.1 Input and output variables

```
function [phi0,path,th,cv] = eik_spirals_initcond(VER,TRI,nodes,signs)
```

```

% EIK_SPIRALS_INITCOND
% generates an initial condition for the eikonal equation including
% several spirals with given location and sens of rotation
%
% [phi0,path,theta,cv] = eik_spirals_initcond(VER,TRI, nodes, signs);
%
% inputs:
%   VER:    position of vertices
%   ITRI:   indices of triangles
%   nodes:  spiral positions (indices in VER)
%   signs:  sens of rotation (+/- 1)
%
% outputs:
%   phi0:   complex phase field
%           returns [] if a node is too close to a boundary
%   path:   indices of the circuits
%   theta:  phase (0-2pi) in the circuits
%   cv:     scaled conduction velocity for each triangle

```

3.4.2 Reconstruction of the phase field

Interpolation in the entire domain is obtained by solving the Laplace equation $\Delta\phi = 0$ with no-flux boundary condition where no value is assigned to ϕ (e.g. veins or valves) [16]. Then $\phi_0 = \phi/|\phi|$ provides an estimate of the desired complex phase field. Indeed, by construction, $\theta_0 = \arg \phi_0$ has a phase singularity at \mathbf{x}_i with topological charge q_i for each i .

First, identification of the circuits surrounding the reentries

```
circuits = identify_circuits;
```

An already implemented function is used to define the finite element matrices

```
c = 1;
D = 1;
P = eik_solver_setup( VER,TRI,c,D);
```

Values between 0 and 2π are given to the circuits,

```
for i=1:n
    path_temp = [circuits{i}];
    [th_temp,len_temp] = eik_circuit_phase(VER,path_temp);
```

according to the predefined sens of rotation.

```
if signs(i)==-1
    th_temp=2*pi-th_temp;
```

```

    end
    path =[ path, path_temp ];
    th =[th, th_temp];
end

```

Solver for the Laplace equation :

```
[ phi0 ] = eik_solver_initcond( P, path, th);
```

Indices of the circuits are stored.

```
% include the newly created phase singularities in path
tau = angle(phi0);
I = eik_find_phasesing(TRI,tau);
path = unique(reshape(TRI(I,:),[],1));
th = tau(path);
```

ϕ may have one or more zeros. This results in the creation of additional phase singularities at these points. The number of reentries obtained may thus be greater than the one initially asked.

3.4.3 Wavelength

The local wavelength of a phase field can be defined as:

$$\lambda(\mathbf{x}) = 2\pi \|\nabla\theta\|^{-1} = 2\pi \|\nabla\phi\|^{-1}. \quad (6)$$

For a reentry with period T , this local wavelength is related to the conduction velocity (CV in cm/s) by the formula $\lambda = CV \cdot T$ [15].

The local wavelength for ϕ_0 is typically very non-uniform, especially when n is small. As a result, the phase field provides an inaccurate representation of a reentrant activation pattern. In order to iteratively regularize the phase field using *a priori* information about wavefront propagation, the eikonal-diffusion equation will be applied as a filter.

3.4.4 Regularization of the phase field

The complex form of the eikonal-diffusion equation reads [15]:

$$c \|\nabla\phi\| = 1 + D \operatorname{Im} \nabla \cdot (\phi^* \nabla\phi), \quad (7)$$

where c is the scaled conduction velocity (in cm/rad) and D is a diffusion coefficient (in cm^2). When D tends to zero, the equation becomes $\lambda(\mathbf{x}) = 2\pi c = \text{const}$. A Newton-based iterative scheme can be derived to solve this equation [15, 17]. At step k , a phase

correction $\phi_{k+1} = \phi_k \exp(i\Psi)$ is applied (so $|\phi_k| = 1$ always holds), where Ψ is a solution to the steady-state convection-diffusion equation:

$$c \frac{\operatorname{Im} \phi_k \nabla \phi_k^*}{\|\nabla \phi_k\|} \nabla \Psi + D \Delta \Psi = c \|\nabla \phi_k\| - 1 - D \operatorname{Im} \nabla \cdot (\phi_k^* \nabla \phi_k) , \quad (8)$$

with Dirichlet boundary condition on each of the circuits Γ_i and no-flux boundary condition everywhere else. Spatial discretization was performed through a dedicated finite element method published previously by Vincent Jacquemet [17].

This equation can be interpreted the following way in the low-diffusion limit. With $\phi_k = \exp(i\theta_k)$ and $D = 0$, Eq. (8) can be written as:

$$\mathbf{n}_k \cdot \nabla \Psi = \frac{1}{c} - \|\nabla \theta_k\| , \quad (9)$$

where $\mathbf{n}_k = \nabla \theta_k / \|\nabla \theta_k\|$ is the forward-oriented unit vector normal to the isochrone $\theta_k = \text{const}$. If the local propagation velocity is too fast (*i.e.*, the right hand side is positive), a correction $\theta_{k+1} = \theta_k + \Psi$ is applied so that the normal component of the gradient $\mathbf{n}_k \cdot \nabla \theta_{k+1} = \|\nabla \theta_k\| + \mathbf{n}_k \cdot \nabla \Psi$ is increased, leading to a slower local propagation velocity and thus regularizing the propagation pattern. The diffusion is needed to obtain a unique, smooth solution and the use of complex numbers facilitates the evaluation of the gradient in the presence of 2π jumps.

The parameters c and D need to be specified. On Γ_i , the local wavelength $\lambda = 2\pi c$ should be equal to the length $L_i \approx 6$ mm of the circuit, so c is set to 1 mm/rad. The diffusion coefficient D enforces the stability of the equation. Its value should be of the order of the squared spatial resolution Δx^2 and is set to 1 mm². The results are not very sensitive to this choice. With such a small target conduction velocity ($CV = 2\pi c/T = 4$ cm/s for $T = 160$ ms), the local wavelength decreases at each iteration while becoming more uniform. Below, parts of Matlab function regularizing conduction velocity (full code in appendage C).

Input and output variables :

```
function [phi1,CV,COV] = eik_spirals_regularize(VER,ITRI,path,phi0,lambda)
% EIK_SPIRALS_REGULARIZE
% regularize a phase map (make conduction velocity more uniform)
%
% inputs:
%   VER:    position of vertices
%   ITRI:   indices of triangles
%   path:   nodes with fixed values
%   phi0:   initial phase map
%   lambda: target wavelength (= 2 pi * conduction velocity)
%
```

```
% outputs:
% phi1: final phase map
% CV: conduction velocity for each iteration
% COV: coefficient of variation of conduction velocity
```

Solver for eikonal-diffusion equation iteration:

```
[phi1,cv,err] = eik_solver_iter(P,phi0,path(:));
```

Output data for local wavelength :

```
COV(iter) = median(abs(cv-median(cv))/abs(median(cv)));
CV(iter) = median(cv);
```

Iterations are stopped when the median of the local wavelength λ reaches a target value obtained from physiological considerations (CV in the tissue \times period of reentry):

```
if iter > miniter && 2*pi*CV(iter) < lambda
    break;
end
```

3.5 Initial condition for a monodomain model

From the phase field $\theta = \arg \phi_k$, k being the last iteration, an initial condition for the monodomain propagation model is constructed [15]. First, the period of reentry T is estimated based on the effective refractory period of the membrane model used in the monodomain simulation ($T = 200$ ms). The phase field $\theta(\mathbf{x})$ is then converted into an activation map $t_{\text{act}}(\mathbf{x})$ by the formula

$$t_{\text{act}}(\mathbf{x}) = \frac{1 - \theta(\mathbf{x})}{2\pi} T.$$

The initial membrane potential is given by $V_0(\mathbf{x}) = V_{\text{paced}}(t_{\text{act}}(\mathbf{x}))$ and the initial membrane state by $\mathbf{s}_0(\mathbf{x}) = \mathbf{s}_{\text{paced}}(t_{\text{act}}(\mathbf{x}))$, where $V_{\text{paced}}(t)$ and $\mathbf{s}_{\text{paced}}(t)$ are the (steady-state) time courses of the membrane potential and state of the midcell in a strand of cells paced at cycle length T at one of its extremity. A monodomain simulation is finally run from this initial condition and the evolution of the number and location of phase singularities is analyzed.

3.6 Test cases

The approach was tested in a simplified geometry representing the atrial epicardium by a triangular surface mesh. Two versions, a coarse ($\approx 13,800$ nodes (Fig. 14); for the eikonal-diffusion solver) and a very coarse (≈ 900 nodes (Fig. 13); for phase singularity

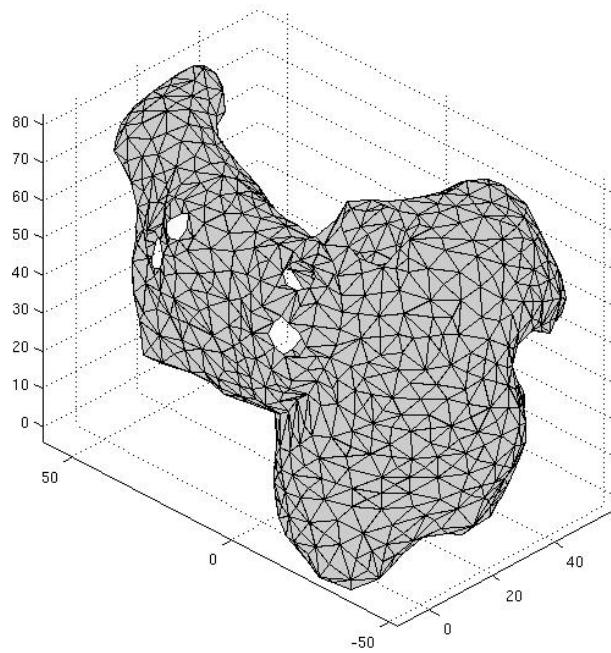


Figure 13: very coarse mesh \approx 900 nodes

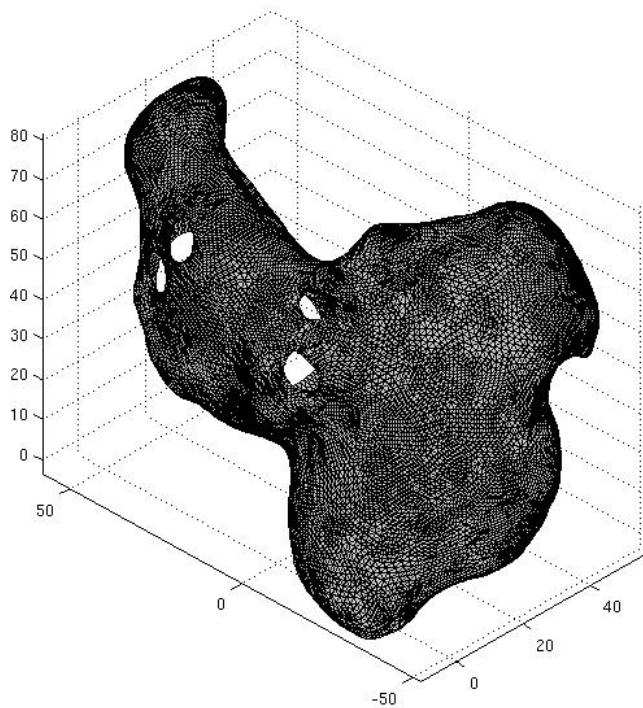


Figure 14: coarse mesh \approx 13,800 nodes

distribution), were designed. These surface meshes were generated and processed using Matlab (The MathWorks, Natick, MA) and VRMesh (VirtualGrid, Seattle City, WA).

In order to run monodomain simulations in a three-dimensional volumetric model, a cubic mesh ($\approx 750,000$ nodes, $\Delta x = 0.33$ mm, thickness ≈ 1.6 mm) was created from the surface mesh [18]. The coarse atrial surface model lied within the bulk of the 3D model to enable data extrapolation from the surface mesh to the full 3D mesh.

4 Results

4.1 Generation of phase maps

A thousand phase maps were generated on the simplified atrial epicardial surface based on $n = 1$ to 10 phase singularities randomly distributed over the surface (method of Subsect. 3.3; 100 realizations for each n). An example with $n = 4$ is displayed in Fig. 15. The initial estimate (Fig. 15A) is qualitatively correct, but the isochrones (level lines of the phase) are not regularly spaced. The eikonal-diffusion solver iteratively improves the spacing between isochrones, creates spiral-like curved wavefronts near phase singularities and handles wavefront collision patterns (Fig. 15F).

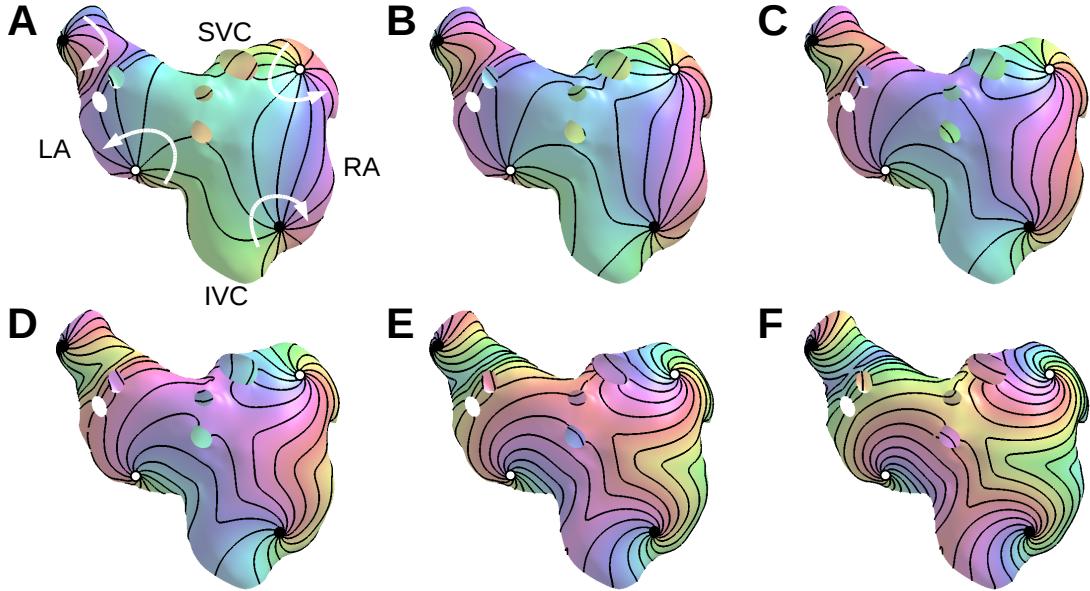


Figure 15: Phase maps generated from four phase singularities. Phase singularities with positive/negative topological charge are shown as a white/black circle. White arrows illustrate the direction of propagation. Solid lines represent ten isochrones regularly distributed between 0 and 2π . (A) initial phase map ϕ_0 obtained by Laplacian interpolation; (B)–(E) phase map after 10, 20, 30, 50 iterations of the eikonal-diffusion solver; (F) final phase map (after 68 iterations, the target wavelength of 6 cm is reached). LA: left atrium; RA: right atrium; SVC: superior vena cava; IVC: inferior vena cava.

The regularization effect of the eikonal-diffusion iterations is quantified in Fig. 16. The median $\bar{\lambda} = \text{median}(\lambda)$ and the coefficient of variation $\text{median}(|\lambda - \bar{\lambda}|/|\bar{\lambda}|)$ of the local wavelength λ (Eq. 6) is shown for all 1000 phase maps. In the course of the iterations, the median wavelength (and thus the propagation velocity) and the coefficient of variation decrease monotonically. When n is smaller, more iterations are needed to reach a target wavelength. The evolution of the coefficient of variation is not significantly affected by the number of phase singularities.

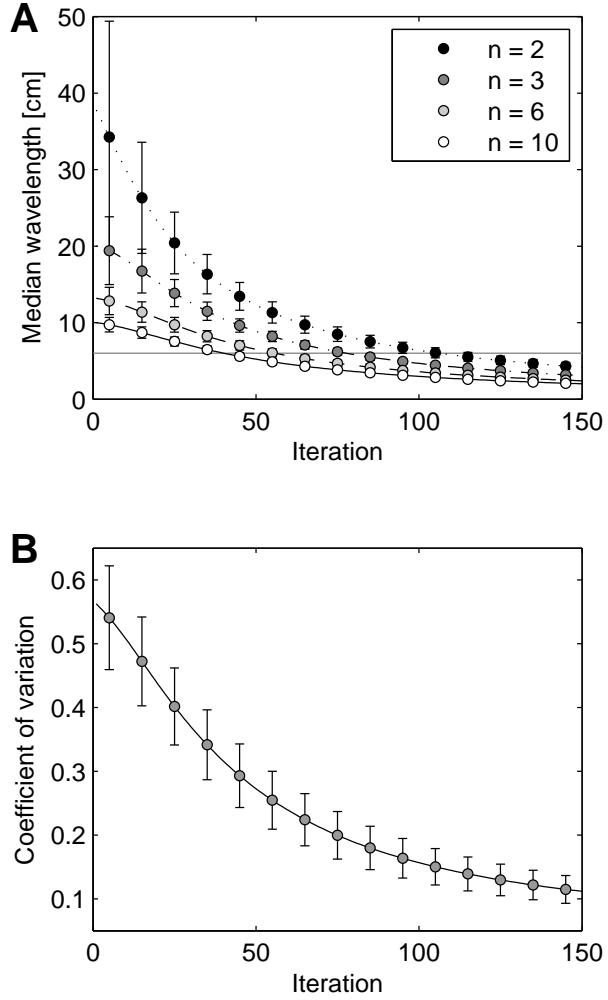


Figure 16: Regularization of phase maps using eikonal-diffusion iterations. (A) Median wavelength λ (spatial average of Eq. (6)) throughout the iterations for $n = 2, 3, 6$ and 10 initial phase singularities. Mean and standard deviation over 100 phase maps are displayed. The horizontal line indicates the target wavelength $\lambda = 6$ cm. (B) Coefficient of variation of the wavelength λ throughout the iterations (pooled data for $n = 1$ to 10). Mean and standard deviation over 1000 phase maps are displayed.

As mentioned above, the actual number of phase singularities in the interpolated phase map (ϕ_0) may be larger than the desired number (n) of phase singularities. An example of such case is shown in Fig. 17. These additional phase singularities result from the process of Laplacian interpolation. They may be required to guarantee the existence of a smooth solution with bounded gradient. Then, the eikonal-diffusion solver preserves the number of phase singularities.

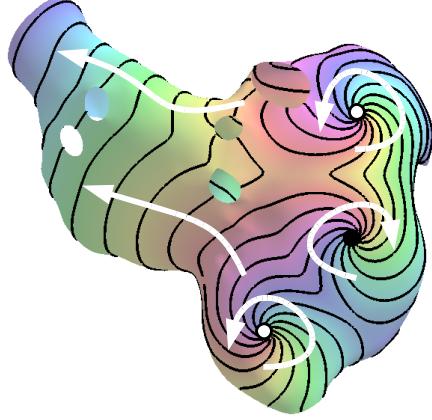


Figure 17: Example of phase map generated from two phase singularities with the same positive topological charge (white circle). An additional phase singularity with opposite topological charge (black circle) is created by the Laplacian interpolation procedure between the two original ones. White arrows illustrate the direction of propagation. Solid lines represent ten isochrones regularly distributed between 0 and 2π .

The average number of additional phase singularities is documented in Table 1 for different values of n . A proper choice of the topological charges (based on energy minimization) reduces the number of additional phase singularities as compared to a purely random choice (± 1 with probability $1/2$). Configurations exist (even with $n = 2$), however, for which no choice of topological charges prevents the creation of additional phase singularities.

Although the desired and actual number of phase singularities may differ, they are well correlated (correlation coefficient: 0.94) and the distribution of actual number of phase singularities is sufficiently uniform (see Fig. 18) to allow the generation of a large number of initial conditions (for a monodomain model) with various numbers of functional reentries.

Table 1: Number of additional phase singularities (PS) for different values of the desired number (n) of PS using the energy-based (minimizing Eq. (5)) or purely random choice of topological charges. Mean and standard deviation over 100 phase maps are reported.

n	#additional PS	
	energy-based	random sign
1	0.4 ± 0.5	0.4 ± 0.5
2	1.0 ± 0.7	1.0 ± 0.7
3	1.2 ± 0.7	1.4 ± 0.7
4	1.7 ± 0.9	1.8 ± 0.8
5	2.0 ± 1.0	2.4 ± 1.0
6	2.1 ± 1.1	2.7 ± 1.1
7	2.5 ± 1.2	3.3 ± 1.1
8	3.1 ± 1.2	4.0 ± 1.2
9	3.2 ± 1.4	4.7 ± 1.4
10	3.6 ± 1.4	5.1 ± 1.5

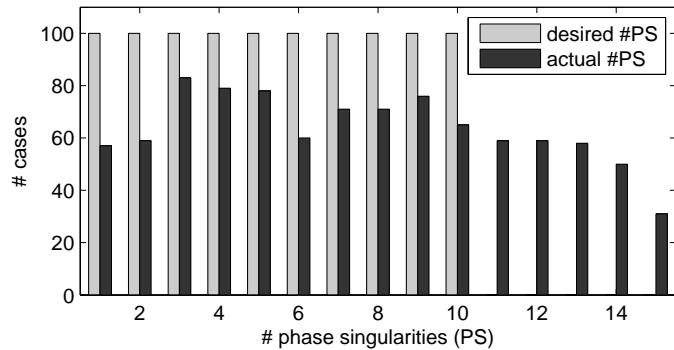


Figure 18: Histogram of desired (n) and actual number of phase singularities (as measured in the phase maps resulting from the eikonal-diffusion solver). The total number of phase maps generated is 1000.

4.2 Monodomain simulations

From the thousand phase maps generated, a hundred monodomain simulations are run over 5000 ms, ten for each initial number of phase singularities from 1 to 10, reproducing the complex spacio-temporal behavior of atrial fibrillation. Below is an example of parameter file enabling to run such a simulation.

Specification of the geometry :

```
geomdir = '/home/jacquem/heart/geom/wilcoepi330u/' ;
```

Membrane model and type of geometry :

```
membrane.model = 'CRN';
grid.domain = "Cables";
grid.cables = [geomdir,"atria.cables"];
```

Modification of the channel maximum conductances :

```
% Dobrev et al., Cardiovasc. Res. 2002
modify{1}.prm = 'PK1';
modify{1}.val = 1.73; % = 10.9/6.3
% Workman et al., Cardiovasc. Res. 2001
modify{2}.prm = 'PCaL';
modify{2}.val = 0.37;
modify{3}.prm = 'Pto';
modify{3}.val = 0.35;
```

Data for simulation (σ is the conductivity tensor) :

```
simtime = 5000;
dt = 0.1;
dx = 0.033;
sigma = 1.5; % in mS/cm
```

Initial condition :

```
initcond.filename = [datadir,'initstate.dat.gz'];
```

Output data :

```
cmap.filename = [datadir,'cmap'];
cmap.spacing = 200;

vmmmap.filename = 'out.bin';
vmmmap.format = 'bin';
vmmmap.spacing = 5;
vmmmap.nodefile = '/home/herlina/matlab/eikonal/wilcoepi13k_to_atria.nodes';
vmmmap.writetime = on;
vmmmap.writevm = on;
vmmmap.writeim = off;
```

Every 5 ms the membrane potential V_m is recorded at the 13,000 nodes corresponding to those of the triangular mesh in order to reconstruct the phase field as explained in subsection 3.2, followed by the identification process of phase singularities. During a simulation, the number of reentries can increase, decrease, stabilize after a while or sometimes the reentries disappear shortly after the beginning of the simulation. Figure 19 illustrates the variability of the number of phase singularities during the simulations, from 1000 ms to 5000 ms. Figure 20 shows phase maps, initial conditions and images of simulations after 2 seconds, for different initial number of reentries. The reentries can also move during a simulation, a functional reentry can meet a valve or a vein and then turn around this anatomical boundary (see fig. 20 B after 2 seconds). The database of simulated AF episodes with varying degrees of complexity is obtained.

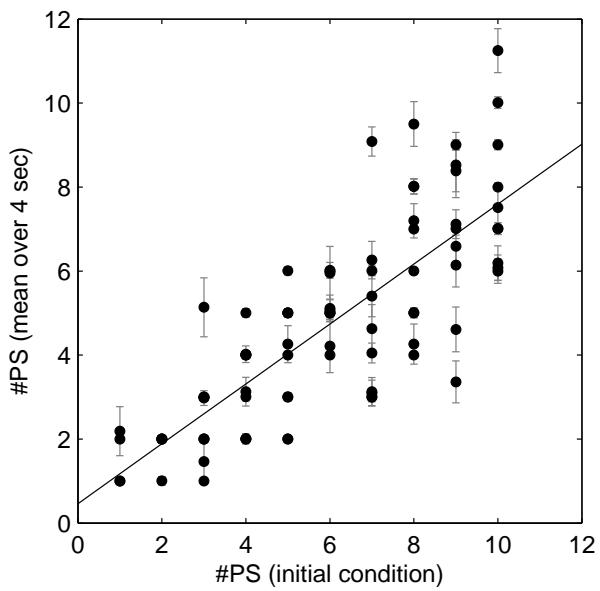


Figure 19: mean number of phase singularities and standard deviation over 4 seconds during 100 monodomain simulations, 10 for each number of phase singularities from $n = 1, \dots, 10$.

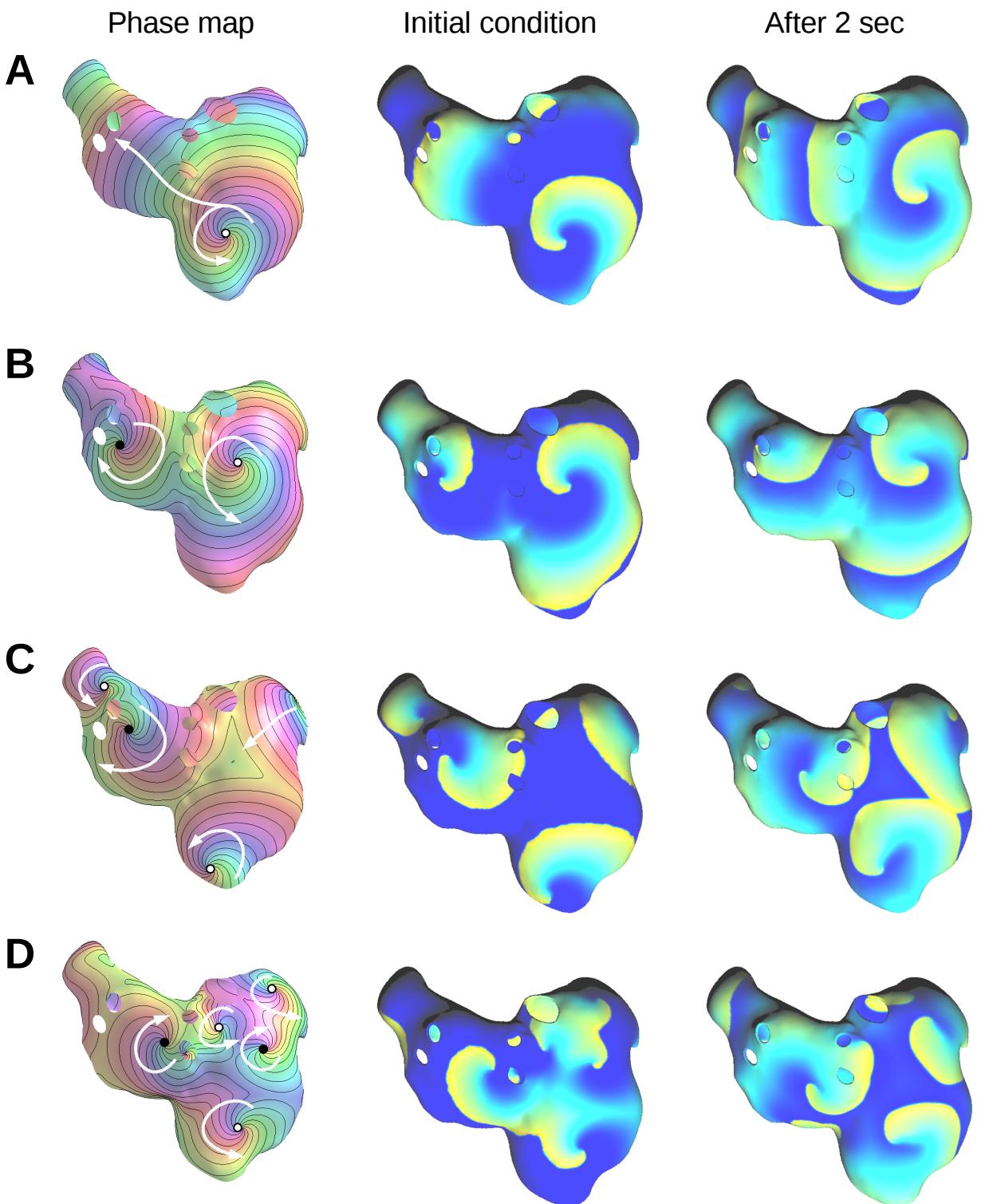


Figure 20: simulations for different initial number of phase singularities.

5 Current work

It is now possible to generate many different phase maps of reentrant activation on an epicardial surface. In order to be specific to each patient, the next step is to be able to transfer a phase map from one geometry of the atria to another one. Six simplified geometries representing the left atrial epicardium of six different patients have been generated. For each geometry, as before, two versions, a coarse ($\approx 10000\text{-}17000$ nodes) and a very coarse ($\approx 800\text{-}2000$ nodes), were designed. It is quite amazing to compare the atria of different human beings. Different sizes, areas, shapes, shapes of the left atrial appendages, positions of the pulmonary veins... The idea is to use a spherical mapping for 3D closed surfaces [19], transforming left atrial meshes into spherical meshes. Then, an interpolation of data on the surface of the sphere would enable the phase map transfer between two geometries.

An example of the current step is presented in Fig. 21. The holes are filled with additional triangles added in the mesh, before mapping each geometry to a spherical geometry.

Future developments will be made on a new geometry with more complex properties such as anisotropy (the electrical impulse not propagating at the same speed in all directions), and with fiber bundles, which is a specialized conduction system for the electric impulse located in the heart.

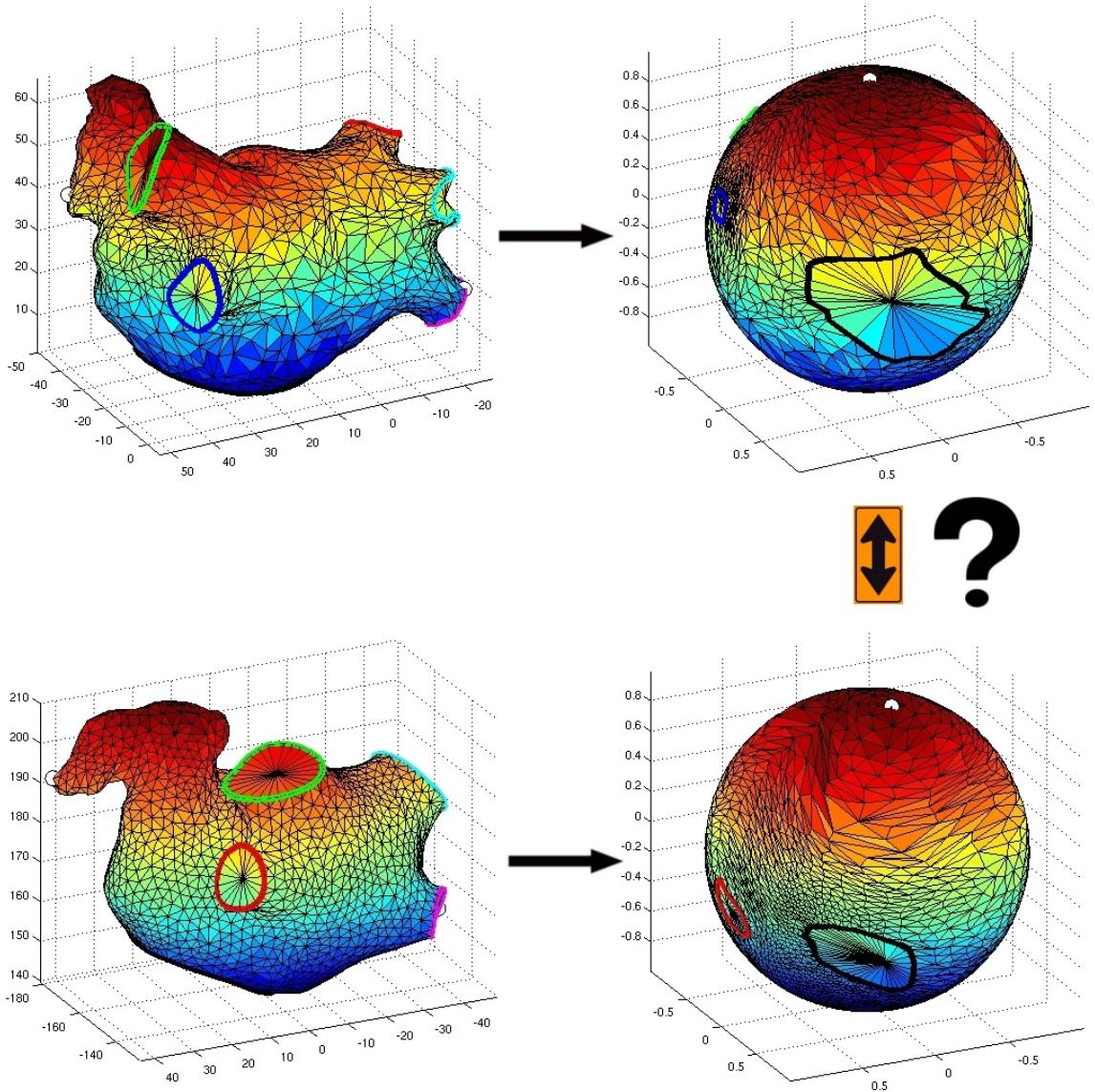


Figure 21: Two geometries of left atria and their spherical mapping. Veins and valves boundaries are identified on the original geometries and the spherical ones thanks to wide lines of different colors. On the original geometries, the green, red, cyan and magenta lines (and blue on the top geometry) circle the pulmonary veins. The interpolation of phase map between the spherical meshes is the next step to be fulfilled.

6 Conclusion

Atrial Fibrillation. Well, it must be a problem about the heart, right? That was pretty much all I could say about it a few months ago. I still have a lot to learn and understand about cardiac electrophysiology in general, and especially about numerical simulations of atrial fibrillation; yet I feel like I know a bit more about it everyday.

Everything started with an internship proposal. Three months later, I landed in Montreal, Canada, for a six months internship with researcher Vincent Jacquemet, Ph. D., in Hôpital du Sacré-Coeur de Montréal research center from March 21st to September 9th, 2011. Getting familiar with the environment, the research field, the Matlab codes implemented prior to my arrival, the simulation tools to run a monodomain simulation on single cells or 3D cardiac tissue, reading articles (e.g. [15], [17]) and chapters of books [1] were an interesting preliminary for the first couple of weeks, in order to understand the framework of the project. It is indeed very stimulating to discover a new field, broadening the interdisciplinarity of the master.

Then I focused my attention on the creation of activation maps in order to run AF simulations. Vincent Jacquemet guided me to proceed step by step, from random distribution of phase singularities to regularization of the phase maps, using already implemented Matlab functions and creating new ones. I was already used to Matlab implementation thanks to several master courses and it was very helpful. Something I did not expect but which is fully part of the work is data processing. Indeed, once we are able to generate activation maps, it is necessary to extract information about the results, whether it is statistics or relevant images. It also enables to make a decision about what to do next or when we need to pick up a short selection of activation maps among the whole lot for instance.

The launching of many simulations was performed with basic bash scripts, then phase field reconstruction and data processing was achieved, still with Matlab. Current work about transfer of a phase map from one geometry of the atria to another one, and other developments will go on in the next weeks.

This internship is introducing me to a new application field of mathematics and scientific computing. I now have a better idea about the work of a researcher and the functioning of a biomedical research center. I hope it will help me obtain a position in biomedical engineering. As I moved from Lille to Montreal, it is also widening my horizons, my vision of the world, and my vision of life. I met people from many different places, at the research center and elsewhere; and I know I will come back to this special place.

References

- [1] J. Malmivuo and R. Plonsey, *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 1995.
- [2] V. Jacquemet, L. Kappenberger, and C. S. Henriquez, “Modeling atrial arrhythmias: Impact on clinical diagnosis and therapies,” *IEEE Rev Biomed Eng*, vol. 1, pp. 94–114, 2008.
- [3] V. Jacquemet, “The inverse problem of phase singularity distribution: An eikonal approach,” *Computing in Cardiology*, vol. 37, pp. 863–866, 2010.
- [4] R. Plonsey and R. C. Barr, *Bioelectricity: A Quantitative Approach*. Kluwer Academic Plenum Publishers, 2000.
- [5] M. Courtemanche, R. J. Ramirez, and S. Nattel, “Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model,” *Am J Physiol*, vol. 275, no. 1 Pt 2, pp. H301–21, 1998.
- [6] S. Nattel, A. Maguy, S. Le Bouter, and Y.-H. Yeh, “Arrhythmogenic ion-channel remodeling in the heart: heart failure, myocardial infarction, and atrial fibrillation,” *Physiol Rev*, vol. 87, no. 2, pp. 425–56, 2007.
- [7] S. Kharche, G. Seemann, J. Leng, A. Holden, C. Garratt, and H. Zhang, “Scroll waves in 3D virtual human atria: A computational study,” in *Proceedings of the 4th international conference on Functional imaging and modeling of the heart*, pp. 129–138, Springer-Verlag, 2007.
- [8] A. J. Workman, K. A. Kane, and A. C. Rankin, “The contribution of ionic currents to changes in refractoriness of human atrial myocytes associated with chronic atrial fibrillation,” *Cardiovasc Res*, vol. 52, no. 2, pp. 226–35, 2001.
- [9] D. Dobrev, E. Wettwer, A. Kortner, M. Knaut, S. Schuler, and U. Ravens, “Human inward rectifier potassium channels in chronic and postoperative atrial fibrillation,” *Cardiovasc Res*, vol. 54, no. 2, pp. 397–404, 2002.
- [10] V. Jacquemet, *A biophysical model of atrial fibrillation and electrograms : formulation, validation and applications*. PhD thesis, EPFL, Lausanne, Switzerland, 2004. available at <http://library.epfl.ch/theses/?display=detail&nr=2996>.
- [11] R. H. Clayton, E. A. Zhuchkova, and A. V. Panfilov, “Phase singularities and filaments: simplifying complexity in computational models of ventricular fibrillation,” *Prog Biophys Mol Biol*, vol. 90, no. 1-3, pp. 378–98, 2006.
- [12] A. N. Iyer and R. A. Gray, “An experimentalist’s approach to accurate localization of phase singularities during reentry,” *Ann Biomed Eng*, vol. 29, no. 1, pp. 47–59, 2001.

- [13] P. M. van Dam and A. van Oosterom, “Atrial excitation assuming uniform propagation,” *J Cardiovasc Electrophysiol*, vol. 14, no. 10 Suppl, pp. S166–71, 2003.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.
- [15] V. Jacquemet, “An eikonal approach for the initiation of reentrant cardiac propagation in reaction-diffusion models,” *IEEE Trans Biomed Eng*, vol. 57, no. 9, pp. 2090–2098, 2010.
- [16] T. Oostendorp, A. van Oosterom, and G. Huiskamp, “Interpolation on a triangulated 3D surface,” *J Comput Phys*, vol. 80, no. 2, pp. 331–343, 1989.
- [17] V. Jacquemet, “An eikonal-diffusion solver and its application to the interpolation and the simulation of reentrant cardiac activations,” *Comput Methods Programs Biomed*, no. in press, 2011.
- [18] V. Jacquemet, A. van Oosterom, J. M. Vesin, and L. Kappenberger, “Analysis of electrocardiograms during atrial fibrillation. a biophysical model approach,” *IEEE Eng Med Biol Mag*, vol. 25, no. 6, pp. 79–88, 2006.
- [19] F. M. Li Shen, “Spherical mapping for processing of 3d closed surfaces,” *Image and Vision Computing*, vol. 24, pp. 743–761, 2006.

A Matlab function *eik_distr_nodes.m*

```

function [nodes, signs, circuits] = eik_distr_nodes(VER,TRI,D,n,prc,N,meth)
% EIK_DISTR_NODES
% distribute phase singularities on a triangulated surface
%
% [nodes, signs] = eik_distr_nodes(D,n,prc);
% [nodes, signs] = eik_distr_nodes(D,n,prc,N,meth);
% [nodes, signs, circuits] = eik_distr_nodes(D,n,prc);
% [nodes, signs, circuits] = eik_distr_nodes(D,n,prc,N,meth);
%
% inputs:
%   VER: position of vertices
%   ITRI: indices of triangles
%   D: distance matrix of the mesh (tri_distmatrix.m)
%   n: number of nodes to be selected
%   prc: distance allowed from optimum (0<prc<1; 0=optimum; default=0.2)
%   N: number of iterations to estimate the optimum (default=1000)
%   meth: method ('dist' or 'energy', 'dist' by default)
%
% outputs:
%   nodes: indices of selected nodes
%   signs: topological charges of the nodes (= +/- 1)
%   circuits: circuits{i} = liste of nodes around nodes(i)
%             if circuits is not output, the existence of these circuits
%             is not checked (faster)

% Antoine Herlin, April 2011

if nargin<7
    meth = 'dist';
end
if nargin<6
    N = [];
end
if nargin<5
    prc = 0.2;
end
if isempty(N)
    N = 1000;
end

if n>16
    error('number of nodes (n) should be <= 16');
end

B_old = tri_find_boundary(TRI);
boundary_nodes = unique(cell2mat(B_old));

```

```

cnt = 0;

while 1
    if meth(1)=='dist'
        [nodes, signs, dist, energy, signed_energy] = gimme_signed_nodes_3(n, ...
        D,1-prc,-2,N);
    else
        [nodes, signs, dist, energy, signed_energy] = gimme_signed_nodes_3(n, ...
        D,inf,prc,N);
    end

    nodes = nodes(:);
    signs = signs(:);

    if nargout>2
        circuits = identify_circuits;
        if ~isempty(circuits)
            break;
        end
        cnt = cnt+1;
    else
        break;
    end

end

%=====
function [nodes, signs, dist, energy, signed_energy] = ...
    gimme_signed_nodes_3(n,D,percentage_of_empirical_distance, ...
    percentage_of_empirical_energy,nb_editions)

    % n is the number of nodes to be chosen.
    % D is the distance matrix
    % percentage_of_empirical_distance is the percentage of the empirical
    % distance we want to have
    % nodes is a n-sized vector containing the indices in VER of the chosen
    % nodes
    % dist is the shortest distance between the nodes
    % energy is the 'energy' of the chosen nodes.

    % optional nb_editions
    if nargin < 4
        nb_editions = 1000;
    end

```

```

% random generation of a first node
while 1
    first = ceil(size(D,1)*rand);
    if isempty(intersect(boundary_nodes,first))
        break
    end
end

%-----
% Calculation of the maximum empirical distance over nb_editions editions
%-----

total_nb_nodes=size(D,1);
matrix_of_nodes=zeros(nb_editions,n);
results=zeros(nb_editions,2);

for k=1:nb_editions

    %generation of (n-1) other nodes
    nodes = [first ; ceil(total_nb_nodes.*rand(n-1,1))];
    matrix_of_nodes(k,1:n)=nodes';

    % calculation of the shortest distance between 2 nodes and the "energy":
    shortest_distance=inf;
    E=0;
    for i = 1:n
        for j = i+1:n
            distance=D(nodes(i),nodes(j));
            E = E + 1/distance;
            if distance<shortest_distance
                shortest_distance=distance;
            end
        end
    end
    results(k,1)=shortest_distance;
    results(k,2)=E;
end

% greatest shortest_distance
[greatest_shortest_distance, index] = max(results(:,1));
E_min=min(results(:,2));
nodes=matrix_of_nodes(index,:);
dist = greatest_shortest_distance;
energy = E_min;

%-----
% Calculation of the nodes, the first edition that matches the
% requirements is chosen

```

```

%-----

results=zeros(1,2);
results(1,1)=-inf;
results(1,2)=inf;

while results(1,1)<percentage_of_empirical_distance*dist ...
    && results(1,2) > energy + percentage_of_empirical_energy*energy

    %generation of (n-1) other nodes
    nodes = [first ; ceil(total_nb_nodes.*rand(n-1,1))];

    % calculation of the shortest distance between 2 nodes and the "energy":
    shortest_distance=inf;
    E=0;
    for i = 1:n
        for j = i+1:n
            distance=D(nodes(i),nodes(j));
            E = E + 1/distance;
            if distance<shortest_distance
                shortest_distance=distance;
            end
        end
    end
    if results(1,1)<shortest_distance
        results(1,1)=shortest_distance;
        results(1,2)=E;
    end
end

dist = results(1,1);
energy = results(1,2);

% Choice of the signs for each node
q=zeros(2^n,n);

if mod(n,2)==0    % if n is EVEN
    signs = choose_signs(0);
else % if n is ODD
    signs = choose_signs(1);
end
end
%=====
function signs = choose_signs(k)
    for i=1:2^n
        bla= i;
        for j=1:n
            q(i,j)=mod(bla,2);
            bla = floor(bla/2);
        end
    end
end

```

```

        if q(i,j)==0
            q(i,j)=-1;
        end
    end
    if abs(sum(q(i,:)))~=k
        q(i,:)=0;
    end
end

q = q(any(q,2),:); % removes 0 rows

ener = zeros(size(q,1),1);
for i = 1:size(q,1)
    E=0;
    for j=1:n
        for k=j+1:n
            E=E+q(i,j)*q(i,k)/D(nodes(j),nodes(k));
        end
    end
    ener(i)=E;
end

[signed_energy , index] = min(ener);
signs=q(index,:);
end
%=====
function circuits = identify_circuits

    NEW_TRI = TRI;
    for i = 1 : size(TRI,1)
        for j = 1 : size(nodes,1)
            if (NEW_TRI(i,1)==nodes(j)) || (NEW_TRI(i,2)==nodes(j)) || ...
                (NEW_TRI(i,3)==nodes(j))
                NEW_TRI(i,:)=0;
            end
        end
    end
    NEW_TRI = NEW_TRI(any(NEW_TRI,2),:); % removes 0 rows

%-----
%    detection of boundaries
B = tri_find_boundary(NEW_TRI);
nb_boundaries = size(B,2);
if nb_boundaries ~= length(B_old) + n
    circuits = [];
    return;
end

```

```

% coordinates of the nodes for functional reentry
coor=zeros(3,n);
for j=1:n
    coor(:,j)=VER(nodes(j),:)';
end

% identification of functional boundaries
link_bdy_nodes=zeros(n,1);
isobarycentre_boundaries=zeros(3,nb_boundaries);
for i=1:nb_boundaries
    mmean = [ 0, 0, 0];
    for j=1:size(B{i},2)-1 % -1 because the last vertex is the same as the ..
        % first one
        mmean = mmean + VER(B{i}(j),:);
    end
    mmean = mmean ./ (size(B{i},2)-1);
    isobarycentre_boundaries(:,i)=mmean';
end

% for each node, let us find the closer isobarycentre.
for i=1:n
    liste_distances=zeros(1,nb_boundaries);
    for j=1:nb_boundaries
        liste_distances(j)=
            sqrt((coor(1,i)-isobarycentre_boundaries(1,j))^2+...
            (coor(2,i)-isobarycentre_boundaries(2,j))^2+...
            (coor(3,i)-isobarycentre_boundaries(3,j))^2);
    end
    [mini,link_bdy_nodes(i)]=min(liste_distances);
end

circuits = [];
for i=1:n
    circuits{i} = B{link_bdy_nodes(i)};
end
end

```

B Matlab function *eik_spirals_initcond.m*

```

function [phi0,path,th,cv] = eik_spirals_initcond(VER,TRI,edges,signs)
% EIK_SPIRALS_INITCOND
% generates an initial condition for the eikonal equation including
% several spirals with given location and sens of rotation
%
% [phi0,path,theta,cv] = eik_spirals_initcond(VER,TRI,edges,signs);

```

```

%
% inputs:
%   VER:    position of vertices
%   ITRI:   indices of triangles
%   nodes:  spiral positions (indices in VER)
%   signs:  sens of rotation (+/- 1)
%
% outputs:
%   phi0:   complex phase field
%           returns [] if a node is too close to a boundary
%   path:    indices of the circuits
%   theta:   phase (0-2pi) in the circuits
%   cv:     scaled conduction velocity for each triangle

% Antoine Herlin, April 2011

n = length(nodes);

% This function gives the circuits surrounding the reentries
circuits = identify_circuits;

%-----
% create the initial phase map
%-----

c = 1;
D = 1;

%define the Finite Element matrices
P = eik_solver_setup( VER,TRI,c,D );

% path is a vector composed of the indexes of the chosen vertices (from the
% boundary) and theta is a same-sized vector with the values assigned to
% each vertex

path = [];
th = [];
for i=1:n
    path_temp = [circuits{i}];
    % give value between 0 and 2*pi to the circuits
    [th_temp,len_temp] = eik_circuit_phase(VER,path_temp);
    if signs(i)==-1
        th_temp=2*pi-th_temp;
    end
    path =[ path, path_temp ];
    th =[th, th_temp];
end

```

```

path = path(:);
th = th(:);

% solve the Laplace equation
[ phi0 ] = eik_solver_initcond( P, path, th);

% include the newly created phase singularities in path
tau = angle(phi0);
I = eik_find_phasesing(TRI,tau);
path = unique(reshape(TRI(I,:),[],1));
th = tau(path);

% scaled conduction velocity
if nargout>3
    if P.dim == 2
        Norm = sqrt( abs(P.Grad.x * phi0).^2 + abs(P.Grad.y * phi0).^2 );
    else
        Norm = sqrt( abs(P.Grad.x * phi0).^2 + abs(P.Grad.y * phi0).^2 + ...
                     abs(P.Grad.z * phi0).^2 );
    end
    cv = 1./Norm;
end

```

C Matlab function *eik_spirals_regularize.m*

```

function [phi1,CV,COV] = eik_spirals_regularize(VER,ITRI,path,phi0,lambda)
% EIK_SPIRALS_REGULARIZE
% regularize a phase map (make conduction velocity more uniform)
%
% [phi1,CV,COV] = eik_spirals_regularize(VER,ITRI,path,phi0,lambda)
%
% inputs:
%   VER: position of vertices
%   ITRI: indices of triangles
%   path: nodes with fixed values
%   phi0: initial phase map
%   lambda: target wavelength (= 2 pi * conduction velocity)
%
% outputs:
%   phi1: final phase map
%   CV: conduction velocity for each iteration
%   COV: coefficient of variation of conduction velocity

% Antoine Herlin, April 2011

```

```

miniter = 10;
maxiter = 200;

c = 1;
D = 1;
%define the finite element matrices
P = eik_solver_setup( VER,ITRI,c,D );

iter = 0;
err = 1;
while 1
    iter = iter + 1;
    % iteration of the eikonal-diffusion equation
    [phi1,cv,err] = eik_solver_iter(P,phi0,path(:));

    COV(iter) = median(abs(cv-median(cv))/abs(median(cv)));
    CV(iter) = median(cv);

    fprintf('%.6i: cv = %.3f; COV = %.3f; err = %.2e\n',...
        iter,CV(iter),COV(iter),err);

    phi0=phi1;

    if err<1e-8 || iter>maxiter
        break;
    end
    if iter > miniter && 2*pi*CV(iter) < lambda
        break;
    end
end

```