Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Programa de Pós-Graduação em Modelagem Computacional

Lucas Arantes Berg

Generation and Uncertainty Quantification of Patient-specific Purkinje network models

Juiz de Fora

2022

Lucas Arantes Berg

Generation and Uncertainty Quantification of Patient-specific Purkinje network models

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, como requisito para obtenção do título de Doutor em Modelagem Computacional.

Supervisor: Rodrigo Weber dos Santos

Co-supervisor: Rafael Alves Bonfim de Queiroz

Co-supervisor: Elizabeth Maura Cherry

Juiz de Fora

2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Berg, Lucas.
Generation and Uncertainty Quantification of Patient-specific Purkinje network models / Lucas Arantes Berg. – 2022.
140 f.

Supervisor: Rodrigo Weber dos Santos
Co-supervisor: Rafael Alves Bonfim de Queiroz
Co-supervisor: Elizabeth Maura Cherry
Tese (Doutorado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Modelagem Computacional, 2022.

1.Electrophysiology. 2.Patient-specific. 3.Purkinje Fibers. 4.Uncertainty Quantification. I.Lucas Arantes Berg, Rodrigo Weber dos Santos, Rafael Alves Bonfim de Queiroz, Elizabeth Maura Cherry. II. Generation and Uncertainty Quantification of Patient-specific Purkinje network models

Lucas Arantes Berg

Generation and Uncertainty Quantification of Patient-specific Purkinje network models

Tese apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Modelagem Computacional. Área de concentração: Modelagem Computacional.

Aprovada em 16 de agosto de 2022.

BANCA EXAMINADORA

Prof. Dr. Rodrigo Weber dos Santos - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Rafael Alves Bonfim de Queiroz - Coorientador
Universidade Federal de Ouro Preto

Profa. Dra. Elizabeth Maura Cherry - Coorientadora
Georgia Tech

Prof. Dr. Rafael Sachetto Oliveira - Membro titular externo
Universidade Federal de São João del-Rei

Prof. Dr. Joakim Sundnes - Membro titular externo

Simula Research Laboratory

Prof. Dr. Bernardo Martins Rocha - Membro titular interno
Universidade Federal de Juiz de Fora

Prof. Dr. Marcelo Lobosco - Membro titular interno
Universidade Federal de Juiz de Fora

Juiz de Fora, 16/08/2022.

Documento assinado eletronicamente por **Rodrigo Weber dos Santos**, **Professor(a)**, em 16/08/2022, às 16:45, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Marcelo Lobosco**, **Professor(a)**, em 16/08/2022, às 17:21, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Elizabeth Cherry**, **Usuário Externo**, em 16/08/2022, às 17:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Rafael Sachetto Oliveira**, **Usuário Externo**, em 16/08/2022, às 18:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Joakim Sundnes**, **Usuário Externo**, em 17/08/2022, às 05:18, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Rafael Alves Bonfim de Queiroz**, **Usuário Externo**, em 17/08/2022, às 12:09, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Bernardo Martins Rocha**, **Coordenador(a)**, em 18/08/2022, às 21:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **0910389** e o código CRC **292D9A6A**.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my parents, who continually and in all situations supported me and encouraged me to study, especially during the hard period of the pandemic, whose support was essential for building this thesis. I would also like to thank my brother, that despite not being with us anymore, had always helped and guided me on the path of knowledge.

Secondly, I also leave here my thanks to all the friends I made at UFJF since the time of my graduation, who always stayed together with me even after our graduation. Also, I could not forget the great friendships I made in Graduate Program in Computational Modeling and our research group, FISIOCOMP.

Furthermore, my acknowledgments to my supervisors Rodrigo, Rafael Bonfim and Elizabeth Cherry for their guidance, friendship and patience, without which this work would not have been accomplished. Also, I would also like to extend this recognition to all the Graduate Program in Computational Modeling professors for their teachings and to all its employees, who, during all these years in the program, contributed in some way to my professional and personal enrichment.

In addition, I would like to thank professors Rafael Sebastian and Blanca Rodriguez for their contribution to this thesis by providing the patient-specific data necessary for the validation step of our method. Also, my gratitude to Rafael Sebastian for participating in my qualifying exam and giving valuable suggestions for developing the final part of the thesis.

Finally, my thanks to CAPES, FAPEMIG, CNPq and UFJF for all the financial support.

”Dedico este trabalho aos meus pais Ricardo e Maria Imaculada, e a meu irmão Rafael.”

RESUMO

As redes de Purkinje são uma parte fundamental do sistema de condução cardíaco e são conhecidas por iniciar uma variedade de arritmias cardíacas. No entanto, a modelagem específica das redes de Purkinje de um paciente permanece desafiadora devido à alta complexidade morfológica e à falta de técnicas de imagem não invasivas para identificar estas estruturas. Este trabalho tem como objetivo apresentar um novo método chamado *Shocker* baseado em princípios de otimização para a geração e quantificação de incertezas de redes de Purkinje específicas de paciente que combinam precisão geométrica e elétrica no tamanho do ramo, ângulos de bifurcação e ativação das Junções-Músculo-Purkinje. Vários modelos de redes de Purkinje são gerados em quatro malhas biventriculares diferentes com complexidade crescente para atingir esse objetivo. Estas malhas são utilizadas para avaliar o desempenho do nosso modelo em uma variedade de cenários diferentes. Simulações adicionais de monodomínio acoplando as redes de Purkinje ao tecido biventricular são executadas para avaliar as redes geradas em um cenário mais realista usando os modelos celulares humanos Purkinje/ventricular mais recentes, valores fisiológicos para o atraso característico das Junções-Músculo-Purkinje e um resolvedor GPU de alto desempenho. Os resultados demonstram que o novo método é capaz de gerar redes de Purkinje específicas de paciente com métricas morfológicas controladas, tempos de ativação nas Junções-Músculo-Purkinje, nos pontos estimados dados pelo mapa eletroanatômico do paciente e por eletrocardiograma. Além disso, a geração de vários modelos de rede de Purkinje que podem reproduzir os mesmos dados específicos do paciente é uma importante ferramenta para quantificar as incertezas associadas à modelagem computacional desse importante sistema de condução do coração humano.

Palavras-chave: Eletrofisiologia. Paciente-específico. Fibras de Purkinje. Quantificação de incerteza.

ABSTRACT

Cardiac Purkinje networks are a fundamental part of the conduction system and are known to initiate a variety of cardiac arrhythmias. However, patient-specific modeling of Purkinje networks remains challenging due to the high morphological complexity and the lack of non-invasive imaging techniques to identify these structures. This work aims to present a novel method called *Shocker* based on optimization principles for the generation and uncertainty quantification of patient-specific Purkinje networks that combine geometric and electrical accuracy in branch size, bifurcation angles, and Purkinje-Ventricular-Junction activation. Several Purkinje network models are generated over four different biventricular meshes with increasing complexity to reach this goal. They are used to evaluate the performance of our model in a range of different scenarios. Additional Purkinje-tissue coupled monodomain simulations are executed to evaluate the generated networks in a more realistic scenario using the most recent Purkinje/ventricular human cellular models, physiological values for the Purkinje-Ventricular-Junction characteristic delay, and a high-performance GPU solver. The results demonstrate that the new method can generate patient-specific Purkinje networks with controlled morphological metrics, local activation times at the Purkinje-Ventricular-Junctions, estimated points given by the patient electro-anatomical map and electrocardiogram. In addition, the generation of multiple Purkinje network models that can reproduce the same patient-specific data is an important tool for quantifying uncertainties associated with the computational modeling of this important conduction system of the human heart.

Keywords: Electrophysiology. Patient-specific. Purkinje Fibers. Uncertainty Quantification.

LIST OF ILLUSTRATIONS

LIST OF ACRONYMS

AP	Action potential
APD	Action potential duration
AV	Atrioventricular node
BCL	Basic cycle length
CCO	Constrained Constructive Optimization
CDTs	Cardiac Digital Twins
CO	Constructive Optimization
CPU	Central processing unit
CRM	Cardiac magnetic resonance
CV	Conduction velocity
CVD	Cardiac Vascular Disease
DAD	Delayed-after-depolarization
EAD	Early-after-depolarization
EAM	Electroanatomical mapping system
ECG	Electrocardiogram
GHK	Goldman-Hodgkin-Katz
GPU	Graphics processing unit
HB	His-Bundle
LAT	Local Activation Time
LBB	Left-bundle branch
LDRB	Laplace–Dirichlet Rule-Base
LV	Left ventricle
MRI	Magnetic resonance imaging
ODE	Ordinary differential equation

PC	Purkinje cell
PN	Purkinje network
PVJ	Purkinje-Ventricular-Junction
PDE	Partial differential equation
RBB	Right-bundle branch
RL	Rush-Larsen
RMSE	Root mean square error
RRMSE	Relative root mean square error
RV	Right ventricle
SA	Sinoatrial node
SMP	*Sampson et al.* Purkinje model
STW	*Stewart et al.* Purkinje model
TT08	*Ten Tusscher et al.* Purkinje model
VC	Ventricular cell
VCS	Ventricular Conduction System
VF	Ventricular Fibrillation

LIST OF SYMBOLS

V_m	Transmembrane potential
I_{ion}	Ionic current
I_{stim}	Stimulus current
I_{PVJ}	Purkinje-Ventricular-Junction current
R_{PVJ}	Purkinje-Ventricular-Junction resistance
V^{purk}	Transmembrane potential of the terminal Purkinje cell
V^{tiss}	Transmembrane potential of the tissue cell
N_{PVJ}	Number of ventricular cells inside a PVJ site
N_{cells}	Number of Purkinje cells stimulated
$\partial\Omega$	Surface boundary
η	Gating variables
β	Surface-volume ratio
C_m	Membrane capacitance (monodomain equation)
t_{max}	Total simulation time
dt	Time discretization
σ_l	Tissue longitudinal conductivity
σ_t	Tissue transverse conductivity
σ_n	Tissue normal conductivity
σ_{pk}	Purkinje conductivity
G	Internal conductivity
C_f	Membrane capacitance (cable equation)
τ_f	Time constant
d	Diameter
CV	Conduction velocity

l	Segment length
L_{error}	Local activation time error limit
r_0	Initial region radius
r_{inc}	Region radius increment
S	Cloud of points with inactives and actives points
S_i	Cloud of points with only inactives points
S_a	Cloud of points with only actives points
N_p	Maximum number of passive neighboring segments
N_a	Maximum number of active neighboring segments
CF	Cost function
CF_p	Passive cost function
CF_a	Active cost function
x_{prox}	Proximal location
x_{dist}	Distal location
x_{term}	Terminal location
x_{PVJ}	Active PVJ location
Ω_s	Endocardial surface
k_{term}	Current number of terminal segments
d_{crit}	Critical distance
d_{thresh}	Adaptive distance threshold
L_{rate}	PVJ connection rate
ϵ	Percentage of active PVJs connected with an error below a specified threshold

SUMMARY

1 Introduction

1.1 Motivation

The ventricular conduction system (VCS) is an essential part of the heart since it is responsible for synchronous stimulation of the ventricular walls and crucial for the correct maintenance of the heart rhythm. The VCS comprises the His-bundle, left and right bundles branches, and the Purkinje fiber network. The system provides ventricular contraction from apex to basal, together with synchronization of the left and right ventricles, which allows the correct coordination of electrical activity in all mammalian species and is a fundamental process to maintain optimal stroke volume [54].

In the current clinical and modeling literature, several studies indicated the Purkinje network (PN) as both trigger and maintainer of deadly cardiac arrhythmias, like ventricular fibrillation (VF) and left or right bundle branch block [25, 26, 23, 32]. Furthermore, there is enough evidence which sustains that Purkinje cells (PCs) and ventricular cells (VCs) have different electrophysiological properties [74, 78]. For instance, PCs' action potential (AP) has a faster depolarization and a more negative plateau phase. This characteristic results in longer action potential duration (APD) when compared to VCs [10], which turns PCs more susceptible to developing pro-arrhythmic abnormalities, like early-after-depolarizations and delayed after-depolarization. This contrast in the APD form is likely apparent at the Purkinje-Ventricular-Junctions (PVJs) sites. Due to the distinct shape of the action potential (AP) of PCs and VCs, electrotonic effects and APD dispersion occur between the cells from the Purkinje fiber and the ones from the ventricular tissue [78]. In addition, at the PVJs sites, a characteristic delay in conduction, 3 to 5 ms, is observed from the Purkinje to the ventricular tissue. Mainly, this phenomenon arises when a small volume (i.e., the source, PCs) tries to depolarize a large volume (i.e., the sink, many connected VCs). This is also referred to as source-sink mismatch [47, 79, 17, 40]. Therefore, to properly study cardiac electrophysiology via computational modeling, reliable models of the PN are needed and must take into account both morphological and electrical properties.

The usage of personalized models of cardiac electrophysiology that match clinical observation with high fidelity, referred to as cardiac digital twins (CDTs), show promise as a tool for tailoring cardiac precision therapies [21]. An essential step towards building CDTs models relies on the ability of models to replicate the ventricular activation sequence under a broad range of conditions. However, even with all today's progress on cardiac functional and imaging techniques, there is not a valid technique to extract a PN from clinical data, which turns the development of a coupled Purkinje-Ventricular patient-specific computational model a challenge [77, 53]. Therefore, as a result of its complex geometry, manual generation of PNs can become a highly demanding and time-consuming task [38], making the development of methods that can automatically generate

PNs a requirement to advance the studies in this field.

The main motivation of the present thesis is to develop a novel method using optimization principles to generate patient-specific PNs with geometrical and electrical accuracy. The new method, named *Shocker*, is an extension of a previous method developed by our research group [76] and considers cost functions that rely on electrical and geometrical metrics and focus on generating patient-specific PN models that can be utilized in coupled Purkinje-tissue biventricular simulations.

1.2 Thesis goals

The aim of the thesis is to investigate the role of Purkinje networks play on ventricular activation. This is done by implementing a novel method for generating patient-specific Purkinje networks which relies on optimization principles in terms of geometric and electrical accuracy. In addition, realistic computational models of biventricular meshes are utilized to validate the proposed model alongside with the usage of the most recent human cellular models for Purkinje and ventricular cells. Furthermore, important physiological features of the phenomenon, such as the characteristic delay that occur at the PVJ sites, are also considered in the simulations that evaluate the generated Purkinje networks by the novel model. The method should also be expandable and reusable by providing the possibility to extend Purkinje networks generated by other methods available in the literature by including additional branches or by adding more PVJs on its structure.

The specific goals of the thesis are:

1. Develop a new method able to automatically generate models for patient-specific Purkinje networks in any endocardium surface with morphological and electrophysiological accuracy;
2. Validate the generated PN models with electroanatomical maps and electrocardiograms;
3. Provide the necessary methods to quantify the uncertainties in the generated patient-specific PN models.

1.3 Thesis outline

After the Introduction chapter, Chapter 2 includes a brief overview of the basics concepts related to cardiac electrophysiology. Within this context, the anatomy and function of the human heart is explained focusing on the morphology of the VCS. Fundamental aspects related to the mathematical modelling of the electrical stimulation of the organ are described in the following sections, explaining how to model from cellular to tissue level.

Chapter 3 outlines a description of the novel method to generate patient-specific Purkinje networks. Next, a description of all the biventricular meshes that will be utilized in the work is

presented. For each mesh all the materials and methods necessary to reproduce the simulations of the work are described. Furthermore, a description of how the Purkinje-Ventricular-Junctions are modeled is presented alongside a sensitivity analysis of the input parameters of the method.

Chapter 4 focuses on the first results of the method by presenting the geometric and electrical features of the generated Purkinje networks over three biventricular meshes that possess an increasing level of complexity. In this initial experiment the ability of the method to adjust the structure of network trying to minimize the associated cost functions are analyzed for each ventricular region and important insights are discussed.

Chapter 5 considers the evaluation of the best Purkinje networks generated in the previous experiments in a Purkinje-tissue coupled monodomain simulation using a high-performance GPU solver. In the following simulations the most recent cellular models for human Purkinje and ventricular cells are utilized, as well as important physiological features of the phenomenon, like the characteristic delay that happens at the PVJ sites, are correctly modeled. The final part of the chapter is dedicated to the analysis and discussion of the experiments and the validation of the best Purkinje network generated for the patient-specific mesh using the available electroanatomical map data.

Chapter 6 explores both the method ability to extend an already constructed Purkinje network over a patient-specific mesh with new branches at the same time that generate different PVJ density clouds using the new extra branched trees. Both the Purkinje networks and the PVJ clouds are utilized to investigate their roles in biventricular activation by comparing the results with the clinical electrocardiogram readings that are available for the mesh.

Chapter 7 provides a summary of the thesis, including main findings and future work.

1.4 List of publications

During the development of the thesis, the following works were published in national and international journals.

- *An optimization-based algorithm for the construction of cardiac Purkinje network models.* Authors: Ulysses, J. N., Berg, L. A., Cherry, E. M., Liu, B. R., Dos Santos, R. W., de Barros, B. G., and de Queiroz, R. A. IEEE Transactions on Biomedical Engineering, 65(12), 2760-2768. (2018).
- *A Study of the Electrical Propagation in Purkinje Fibers.* Authors: Berg, L. A., Santos, R. W. D., and Cherry, E. M. In International Conference on Computational Science (pp. 74-86). Springer, Cham. (2019).

- *Potential Roles of Purkinje Fibers in Ischemia-Induced Arrhythmias.* Authors: Walz, T. P., Azzolin, L., Chleilat, E., Berg, L., and Arevalo, H. In 2020 Computing in Cardiology (pp. 1-4). IEEE. (2020).

- *Variability in electrophysiological properties and conducting obstacles controls re-entry risk in heterogeneous ischaemic tissue.* Authors: Lawson, B. A., Oliveira, R. S., Berg, L. A., Silva, P. A., Burrage, K., and Dos Santos, R. W. Philosophical Transactions of the Royal Society A, 378(2173), 20190341. (2020).

- *A comparison between different Purkinje network generation methods.* Authors: Berg, L. A., and dos Santos, R. W. Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782), 5(2). (2020).

- *Improved accuracy of cardiac tissue-level simulations by considering membrane resistance as a cellular-level optimization objective.* Authors: Pouranbarani, E., Berg, L. A., Oliveira, R. S., dos Santos, R. W., and Nygren, A. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 2487-2490). IEEE. (2020).

- *Calibration of single-cell model parameters based on membrane resistance improves the accuracy of cardiac tissue simulations.* Authors: Pouranbarani, E., Berg, L. A., Oliveira, R. S., dos Santos, R. W., and Nygren, A. Journal of Computational Science, 53, 101375. (2021).

2 Background on Computational Cardiac Electrophysiology

In this chapter, general concepts about cardiac electrophysiology are presented. Starting with a description of the structures and functioning of the heart with a focus on the cardiac conduction system. Then it is explained how the electrical stimulus propagates through heart cells. Next, mathematical models capable of capturing the phenomena are presented. And finally, the basic concepts for the computational modeling of the electrical stimulation of the heart are depicted.

2.1 The heart: anatomy and function

The heart is a muscle that works as a pump capable of carrying blood to all body parts, supplying each cell with nutrients essential for life. Its structure comprises four cameras, two atriums in the superior part, separated by the interatrial septum, and two ventricles in the inferior part, separated through the interventricular septum through the interventricular septum, as depicted in Figure 1.

The atria have the function of serving as a reservoir and an entryway for blood to the ventricles. In addition, it functions as a weak pump, which helps to propel the blood. The ventricles, in turn, provide the main force to propel the blood through the pulmonary and peripheral circulations [37]. For the heart to function correctly, the entire process of pumping must be synchronized. This synchronization is controlled by an electrical stimulus that causes the heart muscle to contract [34].

As can be seen in Figure 1, the electrical activity of the heart originates in the sinoatrial node (SA), which is a group of pacemaker cells located in the atrium and capable of self-stimulation. Under normal conditions, cells generate an electrical stimulus that propagates through the right and left atria. In this way, the myocardial muscle of both atria is contracted.

The activation wave then reaches the atrioventricular (AV) node located at the base of the atria. AV cells have a relatively low propagation speed and are responsible for most of the delay in contraction between the atria and ventricles. These delays are coordinated to optimize pumping and protect the ventricles from prior stimulation.

After its activation, the AV cells conduct the stimulus through a septum, activating a set of specialized fibers of the bundle of His and the Purkinje system, which spread branches in a tree-like shape ending on the surface of the endocardium of the ventricles. Purkinje fibers are characterized by conducting the electrical stimulus in very high speed, ranging from $1m/s$ to $4m/s$ [19, 61, 56]. This feature is necessary for strong and rapid activation of the ventricle muscles so that its contraction releases blood to all parts of the body during peripheral circulation.

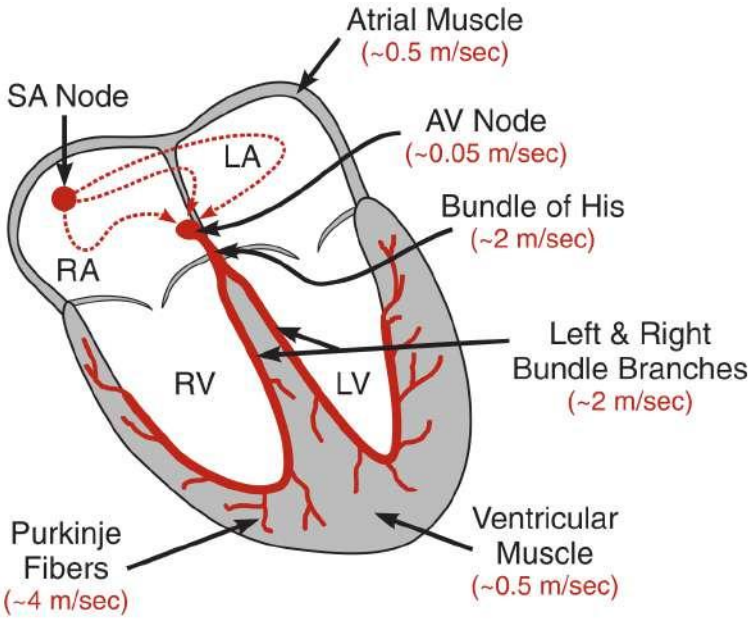


Figure 1 – Main structure of the human heart showing the main structures of the cardiac conduction system. Figure adapted from [37], where the sinoatrial (SA) and atrioventricular (AV) nodes are depicted. In addition, the approximate conduction velocity of the electrical stimulus in each part of the cardiac conduction system is illustrated in red between parentheses.

As shown in Figure 2 the Purkinje fibers begin to branch after the bundle of His and seek to cover the entire surface of the endocardium uniformly. From the image, it can be seen that the gauge of the fibers is different depending on their localization. In the region close to the bundle of His, its size is slightly larger, while in the terminal regions, close to the surface of the endocardium, the fibers are smaller.

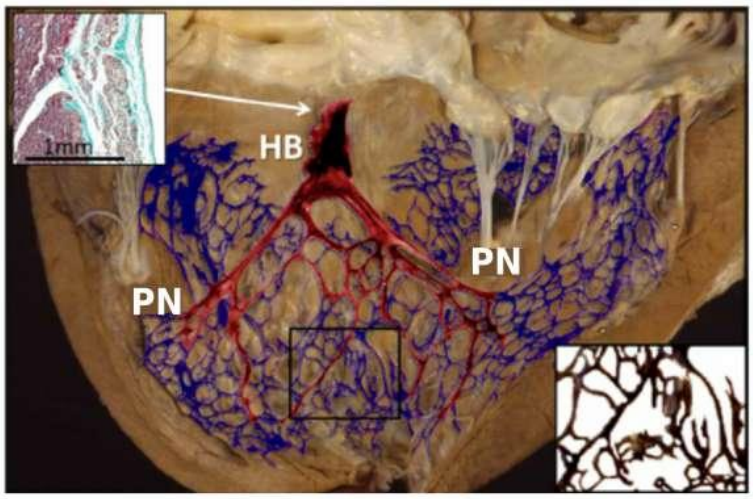


Figure 2 – Image highlighting the Purkinje fibers in a calf ventricle. The heart muscle was prepared with a solution in order to facilitate the visualization of the Purkinje fibers, which were artificially colored using a specilized ink. Figure adapted from [65], where PN describes the Purkinje network and HB the His-bundle.

At the terminal sites of the Purkinje network, there are structures that are called Purkinje-Ventricular-Junctions (PVJs). They are responsible for transmitting the electrical stimulus from PN to the ventricular walls and enabling ventricular stimulation to occur. Under normal propagation conditions, in the regions where the PVJs are located, there is a characteristic delay in the conduction of the stimulus, which may vary from 3 to 5 ms [3]. Mainly, this phenomenon arises when a small volume (i.e., the source, Purkinje cells) tries to depolarize a large volume (i.e., the sink, many connected ventricular cells). This is also referred to as source-sink mismatch [47, 79, 40], and is illustrated schematically in Figure 3.

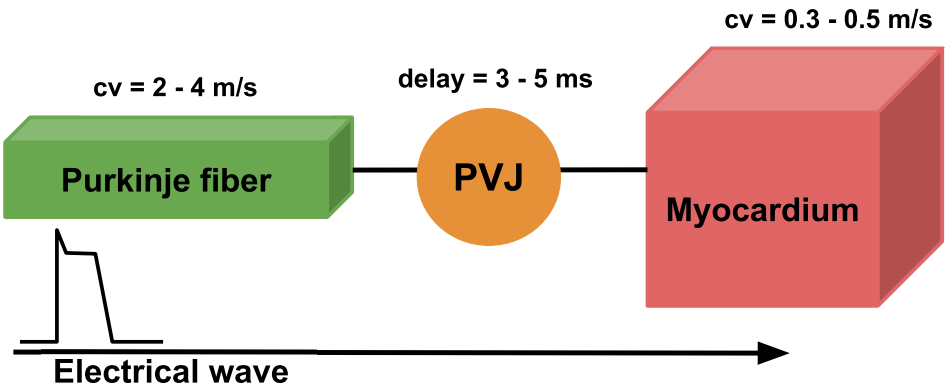


Figure 3 – Illustration of the PVJ coupling between the Purkinje fibers and the working myocardium. The electrical wave coming from the Purkinje encounters a characteristic delay of approximately 3 to 5 ms when reaches the working myocardium cells. The conduction velocity (CV) in the Purkinje fibers ranges between 2 to 4 m/s, while in the ventricular myocardium the CV is around 0.3 to 0.5 m/s [18].

2.2 Cardiac action potential

As in other muscle cells, contraction of cardiac cells is initiated by an electrical activation caused by an action potential (AP). An action potential is a depolarizing current capable of raising the transmembrane potential of a excitable cell from its resting value, typically between -90 to -80 mV, to slightly positive values. Then a repolarizing current returns the potential transmembrane to its resting value. The difference in transmembrane potential is caused by ionic currents that cross the ion channels of cells. Figure 4 represents all phases of an action potential in a cardiac cell, as well as the ionic flux that occurs in each phase.

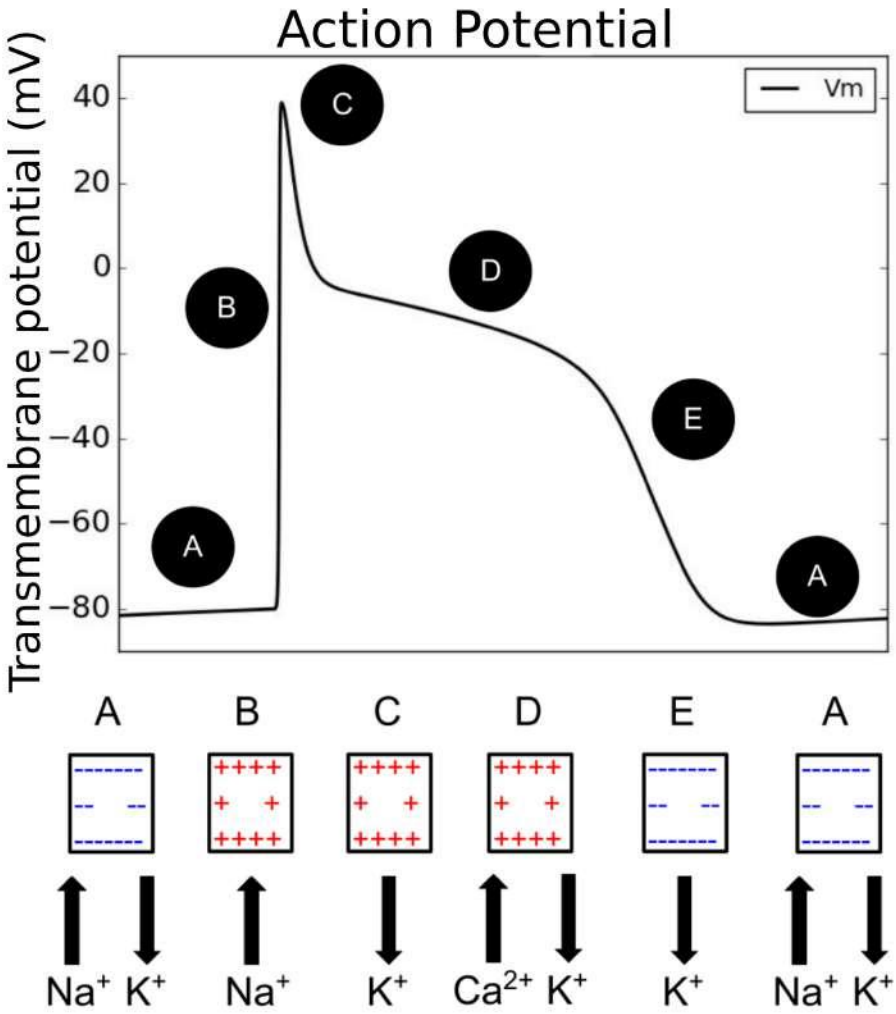


Figure 4 – Representation of an action potential in a cardiac cell and its phases. (A) Rest, (B) Depolarization, (C) Initial repolarization, (D) Plateau and (E) Repolarization. The ionic currents related to the potential change in each phase are shown at the bottom of the figure.

For the generation of an action potential to occur, the cell needs to be stimulated by an external current strong enough and capable of crossing the threshold necessary for the opening of sodium channels (Na^+). Otherwise, the action potential is not generated. Therefore, phase B (depolarization) begins shortly after the sodium channels opening, which enables rapid depolarization of the cell membrane. After this elevation, these channels are quickly closed, causing the characteristic peak behavior of the initial repolarization that occurs during phase C.

Then, the cell begins to open calcium channels (Ca^{2+}), stimulating the release of calcium stored in the cell's sarcoplasmic reticulum, causing an increase in the calcium level in the intracellular environment and promoting the constriction of the cardiomyocytes, responsible for generating cardiac muscle contraction. The plateau identifies this one behavior in the action potential, which occurs during the D phase. During this period, calcium channels remain open and calcium entry is

counterbalanced mainly by a repolarizing potassium current (K^+). As time passes, calcium channels become inactivated, while more calcium rectifier channels potassium open. A more significant number of open potassium channels results in repolarization of the cell, characteristic of the E phase. In this way, the transmembrane potential returns to its initial resting value, and the whole process is repeated for one more heart cycle.

The propagation of an action potential from one cell to another only occurs due to the presence of gap junctions, which are specialized proteins present in the cell membrane and allow the diffusion of ions and other particles between neighboring cells. Furthermore, as cardiac tissue is composed of a set of cells connected in series with each other when a cell is excited, ions are transported to the cell neighbors through gap junctions, thus altering the ionic concentration of the cells that are still at rest. An action potential will be generated if the ion flux in neighboring cells is large enough to exceed the current threshold. This process is schematized in Figure 5.

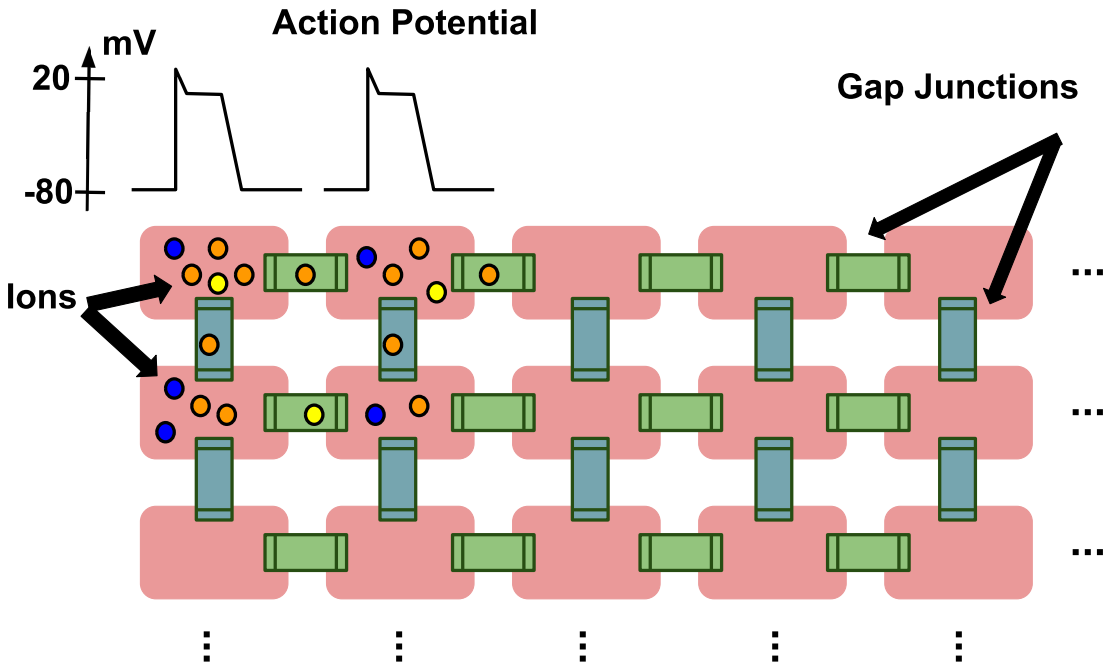


Figure 5 – Representation of the electrical propagation through a bidimensional cardiac tissue. The difference in ionic concentration of the cell membrane generates a difference in potential, which is responsible for triggering an AP, depolarizing the cell when it reaches a certain threshold. In addition, ions can pass from one cell to another by gap junctions, activating the adjacent cells in a wave-like form.

2.3 Electrical model of the cell membrane

The cell membrane plays an essential role in action potential propagation. Its structure is composed of a phospholipid bilayer, as can be seen in Figure 6, with each phospholipid, as

highlighted in Figure 7, formed by two hydrophobic ends connected by a hydrophilic one [35].

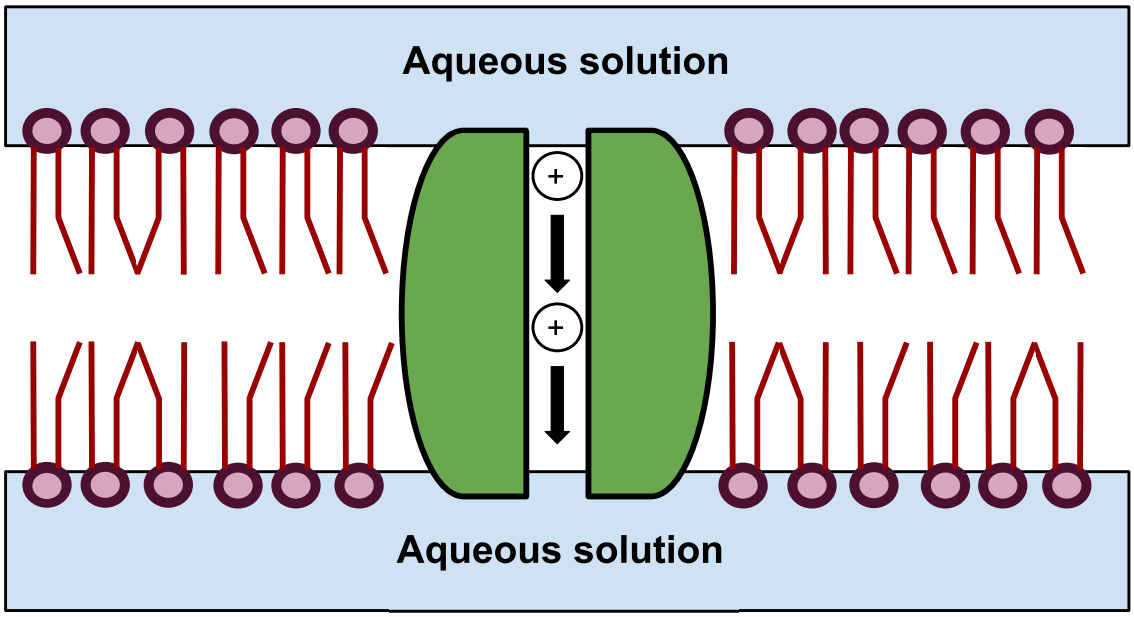


Figure 6 – Illustration of the cell membrane.

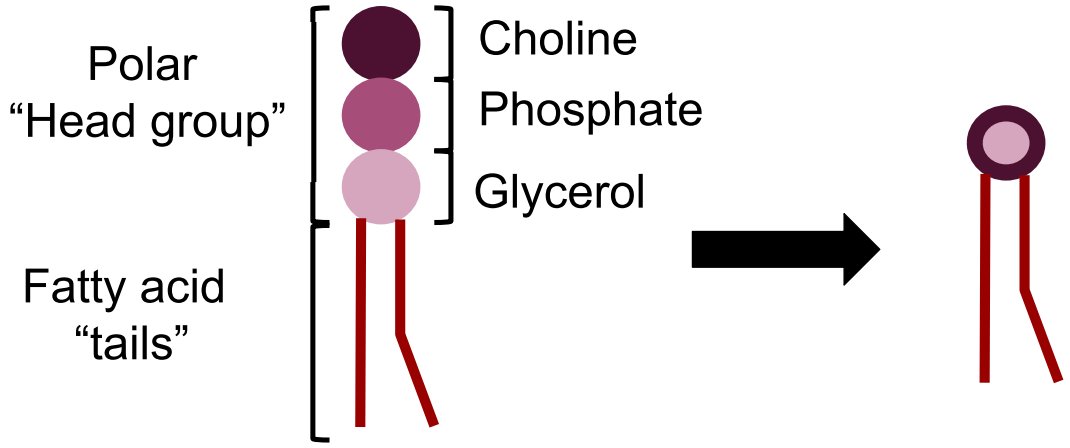


Figure 7 – Illustration of a macromolecular phospholipid.

Because of this characteristic of having a part of the membrane with a higher affinity for water and another with lower affinity, the hydrophobic ends of the phospholipid repelled by water face towards the inside of the bilayer while the hydrophilic end faces outward, thus forming a natural barrier that prevents the passage of charged molecules [35].

In addition to the phospholipid bilayer, the membrane is also formed by small fractions of sugars and proteins, which include ion channels, which are structures responsible for controlling the

flow of ions through the membrane. The intracellular and extracellular media are aqueous solutions of dissolved salts, mainly of $NaCl$ and KCl, which dissociate into K^+ , Na^+ and Cl^- ions. Due to this difference in ionic concentration, a potential difference forms across the membrane [35].

From an electrical point of view, the cell membrane can be compared to a capacitor, where its value is determined by the quotient between the amount of charge stored and the potential difference between the two plates that compose it:

$$C_m = \frac{Q}{V_m} \tag{2.1}$$

The ion channels present in the cell membrane can be modeled as resistances, which are nonlinear functions of the potential difference between the intracellular and extracellular environments, which is given by $V_m = V_i - V_e$. In this way, the model of the cell membrane can be represented by the electrical circuit of Figure 8, where I_c represents the capacitive current, I_{stim} is an external stimulus current and I_{ion} is the current connected to the flow of energy ions that occur in the cell membrane.

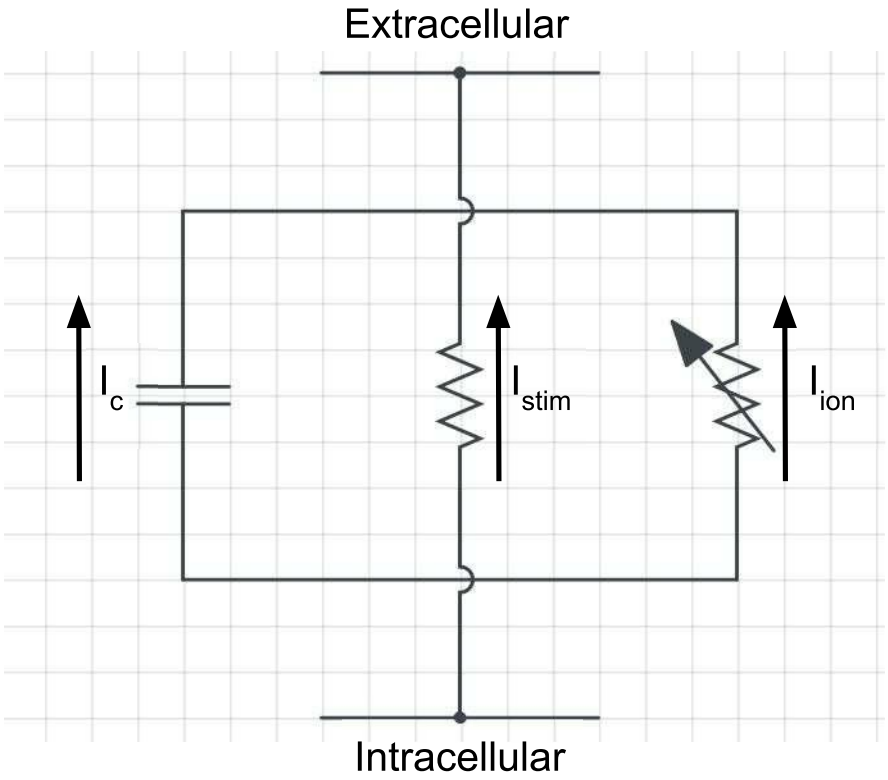


Figure 8 – Illustration of the electric model for the cell membrane.

Based on the electrical circuit in Figure 8, a differential equation for the transmembrane potential, V_m, can be extracted. This equation is obtained by applying Kirchhoff's law for electric

currents, which relates that the sum of the currents leaving a node in the circuit must be zero. Thus, considering that a given node has k currents, Kirchhoff's law is written as follows:

$$\sum_{i=1}^{k} I_i = 0. \tag{2.2}$$

In the circuit of Figure 8 we have 3 currents. The capacitive current I_c is obtained by deriving the expression (2.1):

$$Q = C_m V_m \quad \Rightarrow \quad \frac{dQ}{dt} = C_m \frac{dV_m}{dt} \quad \Rightarrow \quad I_c = C_m \frac{dV_m}{dt}. \tag{2.3}$$

Applying Kirchhoff's law (2.2) on the lower node of the circuit in Figure 8 we get:

$$\begin{aligned} I_c + I_{stim} + I_{ion} = 0 \quad &\Rightarrow \quad C_m \frac{dV_m}{dt} + I_{stim} + I_{ion} = 0 \quad \Rightarrow \\ \Rightarrow \quad &\boxed{\frac{dV_m}{dt} = -\frac{1}{C_m}(I_{ion} + I_{stim})}, \end{aligned} \tag{2.4}$$

where C_m is the capacitance of the cell membrane, I_{stim} is an external stimulus current and I_{ion} is the sum of the ionic currents that cross the ion channels and depends directly on the type of cellular model being used.

2.4 Ionic current modelling

Ion channels are pores formed by specialized proteins that allow the selective flow of ions across the cell's plasma membrane. The conductance of these channels may depend on the value of the transmembrane potential since changes in its value change the arrangement of proteins altering the permeability of the channel [35].

This potential difference occurs due to the movement of ions across the cell membrane. During this ion transport, there are two main flows. The first is a diffusive flux J_F, which is related to the difference in ion concentration between intracellular and extracellular domains. This difference causes particles from a region of higher concentration to be transported to regions of lower concentration.

When a system exhibits a concentration difference inside and outside the membrane, there is a concentration gradient proportional to the number of particles crossing the membrane per unit of time. It is possible to define an unit area perpendicular to the direction of the diffusion flow, where the proportionality constant is given by a diffusion coefficient of the molecules that cross the membrane [35]. That process is known as Fick's law and its mathematical representation is described by:

$$J_F = -D\nabla c, \tag{2.5}$$

where ∇c is a concentration gradient of ions c, D is the diffusion coefficient of the domain and J_F is the ion flux related to the diffusion.

The second flux J_P occurs due to an electrical force that is generated by the electric field related to the potential difference between the intracellular and extracellular media. This flow has the opposite direction to the diffusion and is given by:

$$J_P = -m\frac{z}{|z|}c\nabla V, \tag{2.6}$$

where m is the ionic mobility, z is the ion charge, $|z|$ is the ion valency, c is the ion concentration and ∇V is the electrical potential gradient [35].

When the two fluxes J_F and J_P are equal the cell membrane reaches a steady state scenario:

$$\begin{aligned} J = J_F + J_P = 0 \Rightarrow \\ \Rightarrow J = -D\nabla c - m\frac{z}{|z|}c\nabla V = 0. \end{aligned} \tag{2.7}$$

This process is illustrated in Figure 9. There is a concentration difference across the cell membrane and due to this concentration difference a diffusive flux J_F and an electric flux J_P are generated.

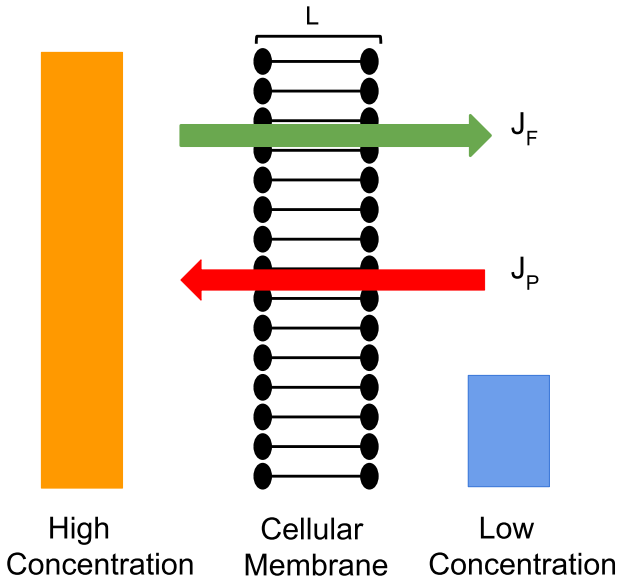


Figure 9 – Illustration of the equilibrium between the diffusive (J_P) and electric (J_F) fluxes across a cell membrane of length L.

The ionic mobility, m, is given by the following expression:

$$m = D\frac{|z|F}{RT}, \tag{2.8}$$

where F is the Faraday constant, R is the ideal gas constant and T is the absolute temperature [35]. Then, we substitute equation (2.8) in (2.7) and obtain,

$$\begin{aligned} J = -D\nabla c + \frac{-DzF}{RT}c\nabla V = 0 \Rightarrow \\ J = -D\left(\nabla c + \frac{zF}{RT}c\nabla V\right) = 0. \end{aligned} \tag{2.9}$$

Considering that the ionic flux occurs by only one ion, the previous expression can be simplified as follows:

$$\frac{dc}{dx} + \frac{zF}{RT}c\frac{dV}{dx} = 0. \tag{2.10}$$

Dividing both sides of the expression by c and calculating the integral from $x = 0$ to $x = L$:

$$\begin{aligned} \frac{1}{c}.\frac{dc}{dx} + \frac{1}{c}.\frac{zF}{RT}c\frac{dV}{dx} = 0 \Rightarrow \int_0^L \frac{1}{c}\frac{dc}{dx}dx + \int_0^L \frac{zF}{RT}\frac{dV}{dx}dx = 0 \Rightarrow \\ \Rightarrow \int_0^L \frac{dc}{c} + \int_0^L \frac{zF}{RT}dV = 0. \end{aligned} \tag{2.11}$$

Considering that the extracellular domain is in $x = L$ and the intracellular in $x = 0$ and $V_m = V_i - V_e$, we obtain:

$$\begin{aligned} \ln(c)\Big|_{c=0}^{c=L} = -\frac{zF}{RT}V\Big|_{V=0}^{V=L} \quad \Rightarrow \quad \ln(c_e) - \ln(c_i) = -\frac{zF}{RT}(V_e - V_i) \quad \Rightarrow \\ \Rightarrow \quad \ln\left(\frac{c_e}{c_i}\right) = \frac{zF}{RT}V_m \quad \Rightarrow \quad \boxed{V_m = \frac{RT}{zF}\ln\left(\frac{c_e}{c_i}\right),} \end{aligned} \tag{2.12}$$

where c_e and c_i are the extracellular and intracellular concentrations of ion c, respectively.

In the expression (2.12), V_m is also known as the Nerst equilibrium potential, and when the transmembrane potential is different from the Nerst potential, an ionic current passes through the channels of the cell membrane.

It is important to address that equation (2.12) is valid only if a single ion crosses the membrane. If the membrane allows the passage of more than one ion, another approach known as Goldman-Hodgkin-Katz (GHK) should be used. More information about deducing the equations from this formulation can be found in literature [35]. In order to exemplify the GHK equation, let's consider the flow of ions Na^+ and K^+:

$$V_m = -\frac{RT}{F} \ln \left(\frac{P_{Na}[Na^+]_i + P_K[K^+]_i}{P_{Na}[Na^+]_e + P_K[K^+]_e} \right), \tag{2.13}$$

where P_K e P_{Na} are the cell membrane permissiveness to the ions K^+ e Na^+, respectively.

It is worth mentioning that both in equation (2.12) and in equation (2.13) if the value of the transmembrane potential is equal to the Nerst equilibrium potential, there will be no flow of ions in the cell membrane, thus denominating a steady state scenario.

In addition, the relationship between ion channels and transmembrane potential can be modeled by two different models, one linear and the other non-linear.

In the linear model, the difference between the transmembrane potential (V) and the Nerst equilibrium potential (V_m) is multiplied by the conductance G of the ion channel, with the conductance being the inverse of the resistance [35].

$$I = G(V - V_m). \tag{2.14}$$

On the other hand, the nonlinear model considers that the relationship between potential and ionic current must follow the GHK equations, which are represented as follows:

$$I = P\frac{z^2F^2}{RT}V\frac{(c_i - c_e)\exp(-zFV/RT)}{1 - \exp(-zFV/RT)}, \tag{2.15}$$

where *exp* indicates the exponential function, P is the permeability of the membrane to the considered ion, and c_i and c_e are the internal and external ionic concentrations of this ion. To obtain the equation (2.15) it is assumed that the electric field is constant over the cell membrane [35].

Depending on the value of the transmembrane potential, some ion channels may be open while others are closed. This behavior can be modeled considering that an ion channel will be in only two possible states, open (O) or closed (C). In addition, there is a transition probability that depends on the value at which the transmembrane potential is, in order to allow a channel to change from one state to another, as represented by:

$$C \underset{\tau}{\overset{\mu}{\rightleftharpoons}} O. \tag{2.16}$$

Considering that X denotes the number of open channels, we can describe the temporal variation of X as follows:

$$\frac{dX}{dt} = \mu(V_m)(1 - X) - \tau(V_m)X, \tag{2.17}$$

where the term $\mu(V_m)$ determines the transition rate from the closed to the open state and $\tau(V_m)$ the transition rate from the open to the closed state.

In most cellular models of cardiac electrophysiology, the variables related to the opening or closing of channels are also known as variables of type *gate* and are described by differential equations as in equation (2.17).

However, some ion channels may have more than one unit, as shown in Figure 10. Thus, the channel will only be fully open if all sub-units are also open.

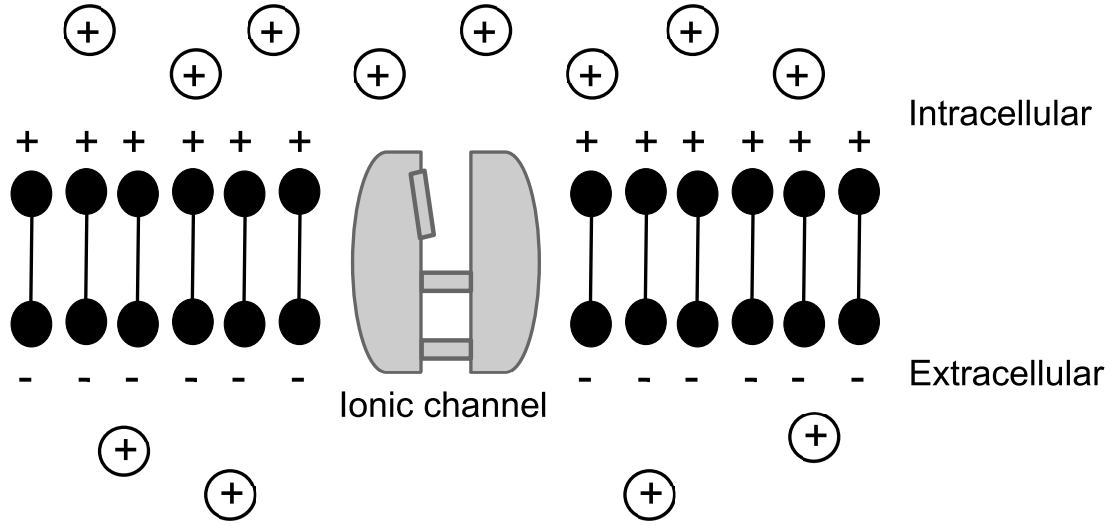


Figure 10 – Illustration of a cell membrane with an ionic channel with more than one sub-unit, where one is open and the other two are closed.

Within this context and considering a simple case in which an ion channel has a total of n independent and equal sub-units, the probability of this channel being in the open state is given by:

$$O = X^n, \tag{2.18}$$

where X is given by equation (2.17).

However, both the sub-units and transition rates may be different. In this case, we will consider that an ion channel is composed of m units of type X and with transition rates μ_X and τ_X, and n units of type Y and with transition rates μ_Y and τ_Y. Thus, the number of open channels for this ion channel is given by following expression:

$$O = X^m Y^n. \tag{2.19}$$

Similarly, the previous expression can be extended to any number of sub-units that a given ion channel has.

Substituting to the expression (2.14), we can represent the ionic channel equation as given by:

$$I = G_{max} O(V - V_m), \tag{2.20}$$

where the term G_{max} equals the maximum conductance obtained when all channels are open.

2.4.1 Noble model

The first mathematical model to express specific cardiac action potentials for Purkinje cells was the 1962 Noble cellular model [49]. Much of its development was based on the famous work of Hodgkin-Huxley which won the Nobel Prize in Physiology or Medicine in 1963 for describing the behavior of the action potential in squid giant axons [30]. In addition to replicating the shape of the action potential, as shown in Figure 11, the model could also predict the presence of other ionic currents that were later discovered experimentally. Another significant achievement of this model was to show how the generation and control of the cardiac cycle occur in mammals without an explicit oscillator.

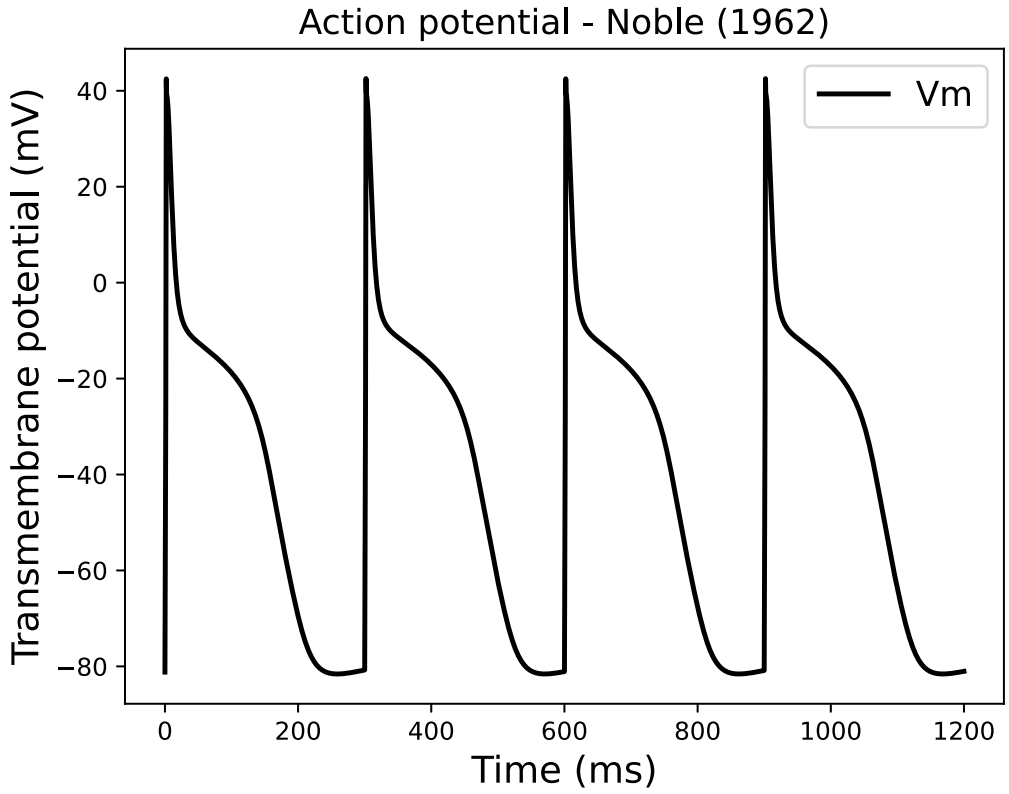


Figure 11 – Action potential described by the Noble cellular model with a basic cycle length of $300ms$ [49].

Most of the ionic currents formulations of the cellular model was developed using Hodgkin-Huxley equations and are similar to what is given by equation (2.20). For instance, three types of current over the cell membrane are considered, one for the sodium I_{Na}, one for the potassium I_K and a leak current I_{leak}, which is related to chloride ions Cl^-. Figure 12 illustrates each of these currents.

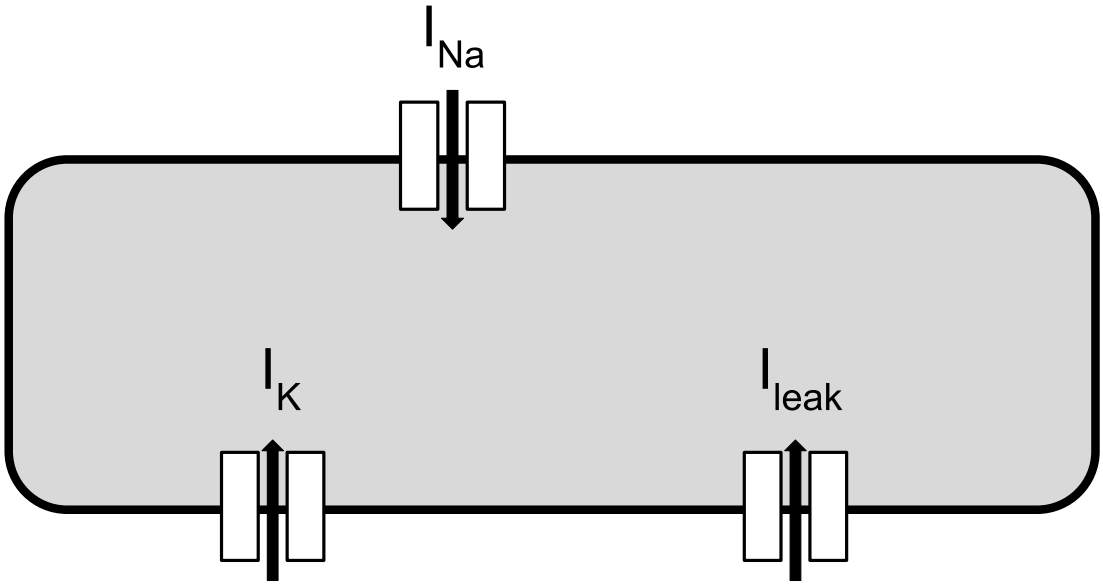


Figure 12 – Ionic currents present in the Noble cellular model [49].

The transmembrane potential across the cellular membrane V_m is given by:

$$\frac{dV_m}{dt} = -\frac{I_{Na} + I_K + I_{leak}}{C_m}, \tag{2.21}$$

where the sodium current I_{Na} is given by:

$$I_{Na} = (400m^3h + 0.14)(V_m - E_{Na}), \tag{2.22}$$

where m and h are calculated in equations (2.25) and (2.26). The potassium current I_K can be obtained from:

$$I_K = \left(1.2e^{\frac{-(V_m+90)}{50}} + 0.015e^{\frac{(V_m+90)}{60}} + 1.2n^4\right)(V_m - E_K), \tag{2.23}$$

where n is calculated in equation (2.27). The leak current I_{leak} is defined by:

$$I_{leak} = 0.075(V_m - E_{leak}). \tag{2.24}$$

In addition, the Nerst potential for the sodium is $E_{Na} = 40\ mV$, potassium $E_K = -100\ mV$ and leak $E_{leak} = -60\ mV$. The membrane capacitance is given by $C_m = 12\ \mu F$ and the gating variables m, h e n are described by:

$$\frac{dm}{dt} = \alpha_m(1-m) - \beta_m m, \tag{2.25}$$

$$\frac{dh}{dt} = \alpha_h(1-h) - \beta_h h, \tag{2.26}$$

$$\frac{dn}{dt} = \alpha_n(1-n) - \beta_n n, \tag{2.27}$$

where the transition coefficients of each gating variable are written as follows:

$$\alpha_m = \frac{-0.1(V_m+48)}{e^{\frac{-(V_m+48)}{15}} - 1}, \quad \beta_m = \frac{0.12(V_m+8)}{e^{\frac{(V_m+8)}{5}} - 1}, \tag{2.28}$$

$$\alpha_h = 0.17e^{\frac{-(V_m+90)}{20}}, \quad \beta_h = \frac{1}{1+e^{\frac{-(V_m+42)}{10}}}, \tag{2.29}$$

$$\alpha_n = \frac{-0.0001(V_m+50)}{e^{\frac{-(V_m+50)}{10}} - 1}, \quad \beta_n = 0.002e^{\frac{-(V_m+90)}{80}}. \tag{2.30}$$

Therefore, the Noble cellular model is described by equations (2.21), (2.25), (2.26) and (2.27), which together form a non-linear system of Ordinary Differential Equations (ODEs). The physical units for this model, following the equations above, are given in millivolts, microsiemens and milliseconds, respectively.

The cellular model was an important step in the cardiac electrophysiology field, since determinant features of an AP are present as seen in Figure 11. For instance, the sharp upstroke comes from a large, fast, inward Na^+ current, and the plateau is maintained by a continued inward Na^+ current, which nearly counterbalances the instantaneous outward K^+ current. Repolarization occurs by a slow outward K^+ current which is activated over time. In addition, a small inward Na^+ leak is responsible to increase the potential again leading to another AP.

Although the Noble model succeeds in reproducing the Purkinje fiber action potential with a model of Hodgkin–Huxley type, the underlying physiology is incorrect, primarily because the model was constructed before data on the ionic currents were available. This lack of data was

mostly because the voltage-clamp technique was not successfully applied to cardiac membrane until 1964.

The weakness of the physiology in the Noble model is exemplified by the fact that there is no calcium current, and the inward sodium current was given the dual role of generating the upstroke and maintaining the plateau [49].

2.4.2 Trovato for human Purkinje cells model

The state of human cardiac cellular models has evolved significantly over the last years. In the past, most cellular models relied on data from animal experiments, which in turn could not be accurate to study specific phenomena observed in humans, like for example arrhythmia, diseases and genetic mutations. However, over the last decades much more electrophysiological data have been collected from humans, leading to the development of more accurate cellular models that surpasses limitations of previous models.

Among these models is the recent human Purkinje cellular model *Trovato2020* [74] which had the objective of integrating the current knowledge on human Purkinje cells electrophysiology in order to investigate mechanisms of pro-arrhythmic abnormalities. The model presents Purkinje-specific ionic currents and a detailed Ca^{2+} subsystem which was not accounted in the previously published Purkinje models, like for example the Noble [49], STW [68], TT08 [71] and SMP [60].

The main contributions of the *Trovato2020* Purkinje cellular model were the reproduction of several experimental recordings in a wide range of protocols as well as electrophysiological alterations followed by Ca^{2+} channels blocks. Secondly, the model presents explanations to the ionic mechanisms underlying pro-arrhythmic abnormalities and automaticity. For this particular study the cellular model also accounts for biological variability by constructing and evaluating a population of models and demonstrate the mechanisms related to Early-After-Depolarization (EADs), Delayed-After-Depolarization (DADs) and triggered activity in human Purkinje cells.

Figure 13 illustrates the main structure and ionic currents presented in the *Trovato2020* cellular model. The model was built based on the ionic formulations of the O'Hara-Rudy human ventricular model, *ORd* [50], and the Purkinje-specific Ca^{2+} sub-system, cellular compartments and intracellular ionic fluxes of the canine Purkinje model, (PRd) [41]. Within this context the *Trovato2020* model includes the *ORd* mathematical formulation for each of the following currents: fast Na^+ current (I_{Na}), Na^+ late component (I_{NaL}), L-type Ca^{2+} current (I_{CaL}), rapid and slow delayed K^+ rectifiers (I_{Kr} and I_{Ks} , respectively), Na^+-Ca^{2+} exchanger (I_{NCX}) and Na^+-K^+ pump (I_{NaK}). I_{to} , I_{sus} and I_{K1} were formulated based on the data from ionic recordings [27]. In addition, two Purkinje-specific currents from the PRd model were included: T-type Ca^{2+} current (I_{CaT}) and funny current (I_f). An important aspect of the model is to account for the differences between ionic current in ventricular and Purkinje cells, and between human and canine Purkinje

cells.

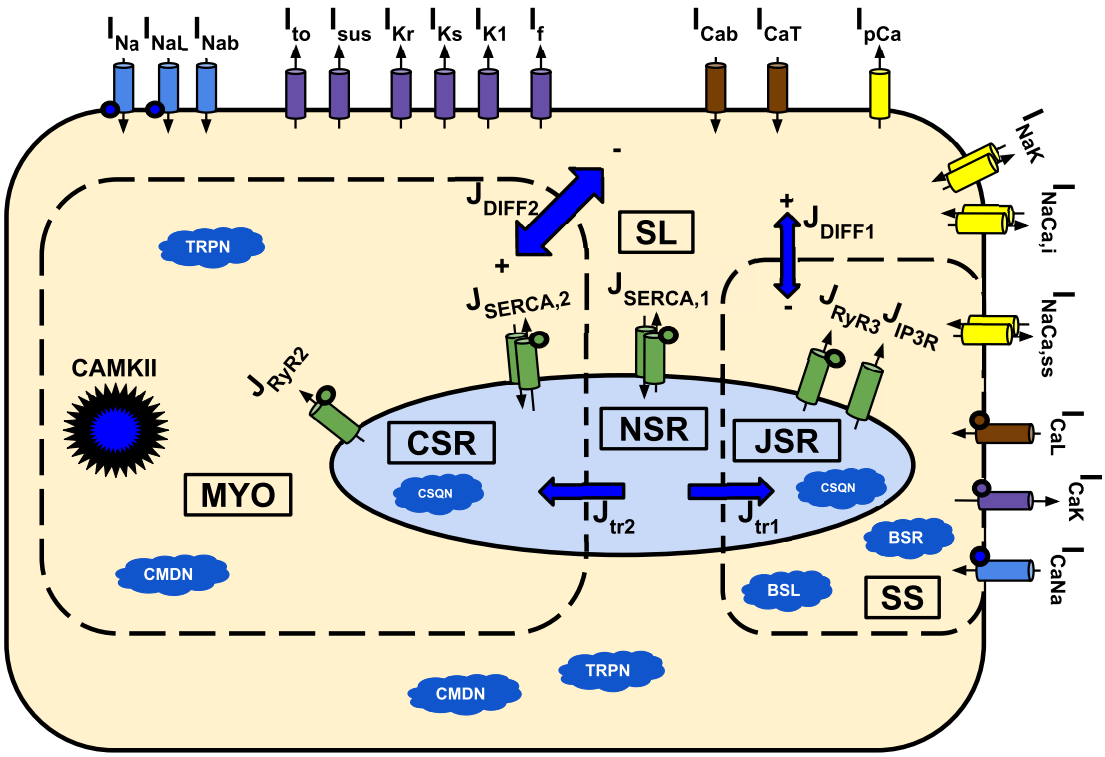


Figure 13 – Illustration showing the ionic currents present in the *Trovato2020* human Purkinje cellular model. The intracellular space is represented with 3 different compartments: peripheral coupling subspace (SS), sub-sarcolemma (SL), and bulk myoplasm (MYO). The sarcoplasmic reticulum (SR) also consists of 3 compartments: junctional (JSR), network (NSR), and corbular (CSR). 18 dynamic current models are included for Na^+ (blue), K^+ (purple) and Ca^{2+} (brown) channels, Na^+-K^+ pump, and Na^+-Ca^{2+} exchanger (yellow). Intracellular Ca^{2+} release and up-take fluxes (green) are distributed across the 3 SR compartments. Ca^{2+} buffers are shown as blue clouds. Global *CaMKII* phosphorylation is also included, and the affected currents are marked by a spiky circle. Figure adapted from [74].

Furthermore, the model is composed by a system of ODEs with a total of 46 state variables to be solved and present stiff properties, similarly to some other recent cellular models [33, 66, 45]. For that reason the performance of standard explicit solvers, such as the Euler method, are highly affected due to stability constraints. The baseline AP of the *Trovato2020* human Purkinje model with a Basic-Cycle-Length (BCL) of 1000ms is shown in Figure 14.

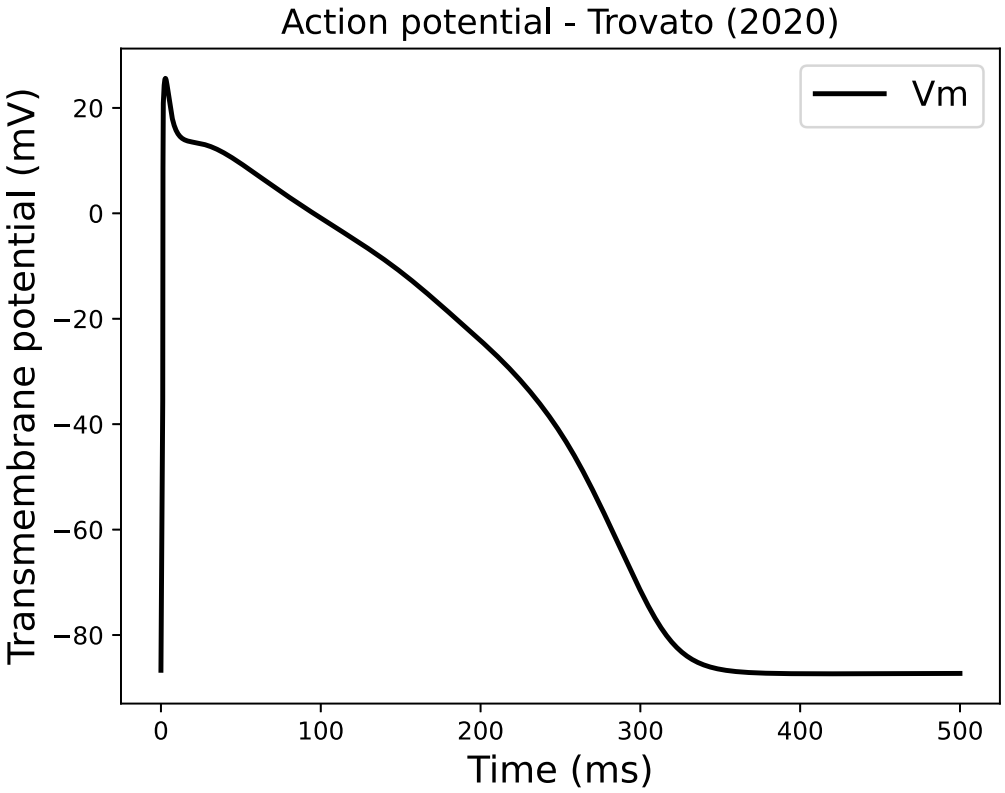


Figure 14 – Action potential described by the *Trovato2020* human Purkinje cellular model with a basic cycle length of $1000ms$ [74].

2.4.3 ToRORd human ventricular models

From the same research group (*Oxford Computational Cardiovascular Science team*) that developed the *Trovato2020* Purkinje human cellular model, the human ventricular cellular model *ToRORd-fkatp-2019* was implemented in 2019 [72]. Similarly to the *Trovato2020*, most of its development was based on the ionic formulations of the O'Hara-Rudy human ventricular model, *ORd* [50]. The main contributions of the *ToRORd-fkatp-2019* was to improve the current state-of-art *ORd* model in several aspects, especially when compared to human ventricular experimental data.

Using a development strategy based on strictly separated model calibration and validation, to design, develop, calibrate and validate the model of human ventricular electrophysiology and excitation contraction coupling, the *ToRORd-fkatp-2019* model aims for simulations that are able to reproduce all key depolarisation, repolarisation and calcium dynamics properties in healthy ventricular cardiomyocytes, under drug block, and in key diseased conditions such as hyperkalemia (central to acute myocardial ischemia), and hypertrophic cardiomyopathy [72].

The main contributions of the *ToRORd-fkatp-2019* human ventricular model was to improve ionic balances during the repolarization phase of the *ORd* by focusing its implementation on

a re-evaluation of the L-type calcium current (I_{CaL}) formulation, given its fundamental role in determining the AP, the calcium transient and sodium homeostasis through the $Na-Ca$ exchanger. The second main focus is the re-assessment of the rapid delayed rectifier current (I_{Kr}), the dominant repolarisation current in human ventricle, under conditions that reflect experimental data-driven plateau potentials by using an implementation of the Lu-Vandenberg [46] Markov model.

Regarding its structure, the *ToRORd-fkatp-2019* model follows the general *ORd* structure. The cardiomyocyte is subdivided into several compartments: main cytosolic space, junctional subspace, and the sarcoplasmic reticulum (SR, further subdivided into junctional and network SR). Within these compartments are placed ionic currents and fluxes described by Hodgkin-Huxley equations or Markov models. The main ionic current formulations altered compared to *ORd* are highlighted in orange in Figure 15A, while the structure of the Lu-Vandenberg Markov model is illustrated in Figure 15B.

Although the *ToRORd-fkatp-2019* is able to ensure the reproduction of key physiological cellular features, with independent multiscale validation demonstrating a correct response to channel blocking drugs and pathophysiological remodelling, the model display drifts in its behaviour in very long simulations, like for example simulations considering extremely long protocols, or for studies on model stability. It was found later in 2020 that the main reason to these changes are caused by modelling chloride concentrations as constant values. To remedy this issue, an updated version, termed *ToRORd-dynCl-2020*, with dynamic representation of intracellular chloride was introduced. This model behaves very similarly to the original *ToRORd-fkatp-2019*, but with stable properties over long simulations and only a small increase in model complexity as shown in Figure 16.

Therefore, in the *ToRORd-dynCl-2020*, intracellular chloride concentrations are not held constant, as in *ToRORd-fkatp-2019*, but updated dynamically according to its two chloride currents (calcium-sensitive Cl current $I_{(Ca)Cl}$ and background Cl current I_{Clb}). Simulations with *ToRORd-dynCl-2020* achieve a steady state, and its steady-state behaviour is very similar to *ToRORd-fkatp-2019*. The original *ToRORd-fkatp-2019* is nevertheless suitable for simulations of thousands of beats, whereas for long simulations, the stability of the *ToRORd-dynCl-2020* is a clear advantage [73].

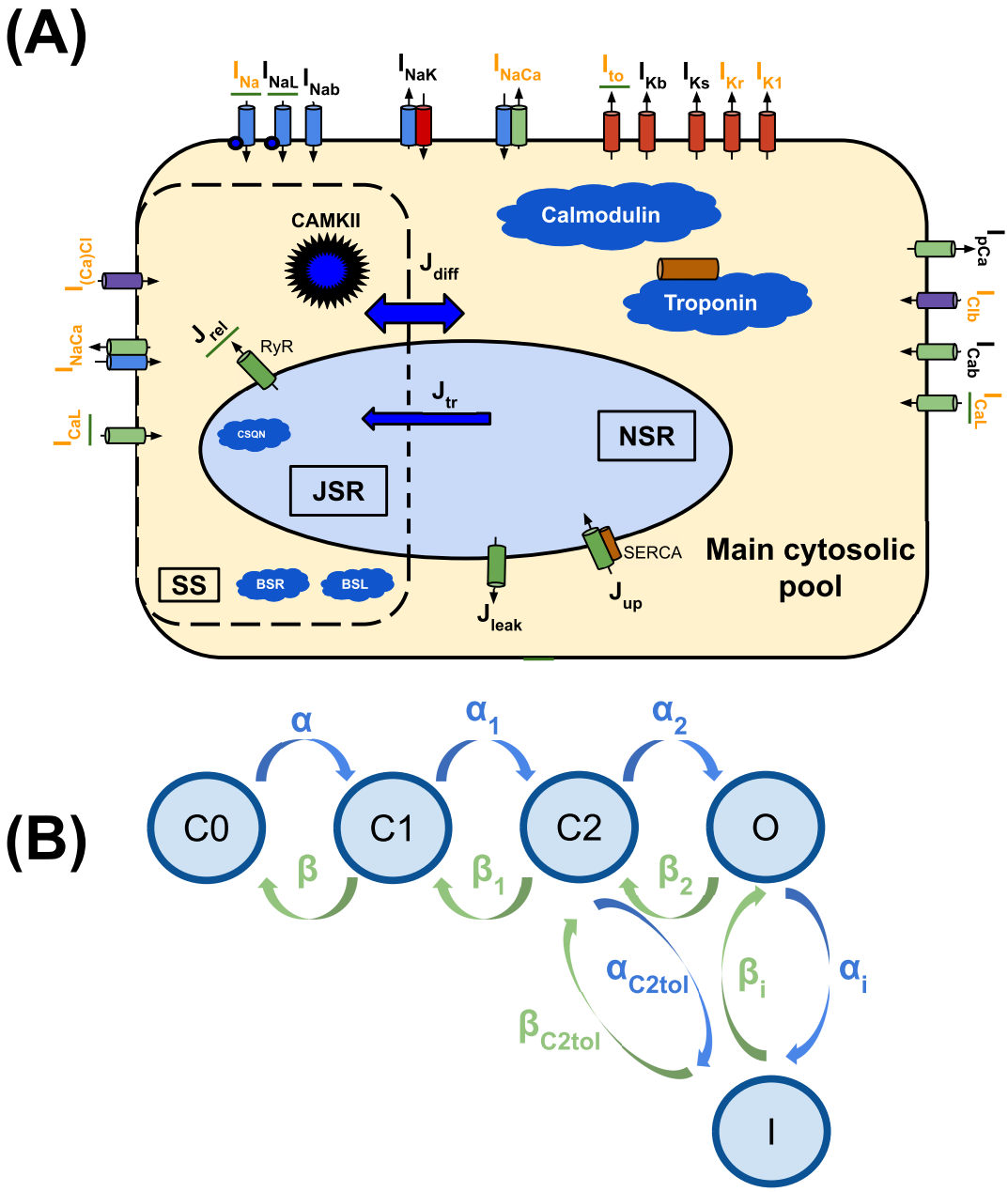


Figure 15 – Illustration showing the ionic currents and associate Markov model present in the *ToRORd-fkatp-2019* human ventricular cellular model. In panel (A), the ionic currents of the model are depicted, where in orange indicates components, substituted, or added, compared to the original ORd model. 'SS' indicates junctional subspace compartment, where calcium influx via L-type calcium current occurs and where calcium is released from the sarcoplasmic reticulum. 'JSR' and 'NSR' are junctional and network sarcoplasmic reticulum compartments, respectively. 'Main cytosolic pool' is the remaining intracellular space. Transmembrane currents are indicated with an 'I' in their name, with fluxes indicated as 'J'. Components with a green underscore are modulated by *CaMKII* signalling. In panel (B), the structure of the Lu-Vandenberg [46] Markov model used for the rapidly activating delayed rectifier repolarisation current (I_{Kr}). Figure adapted from [72].

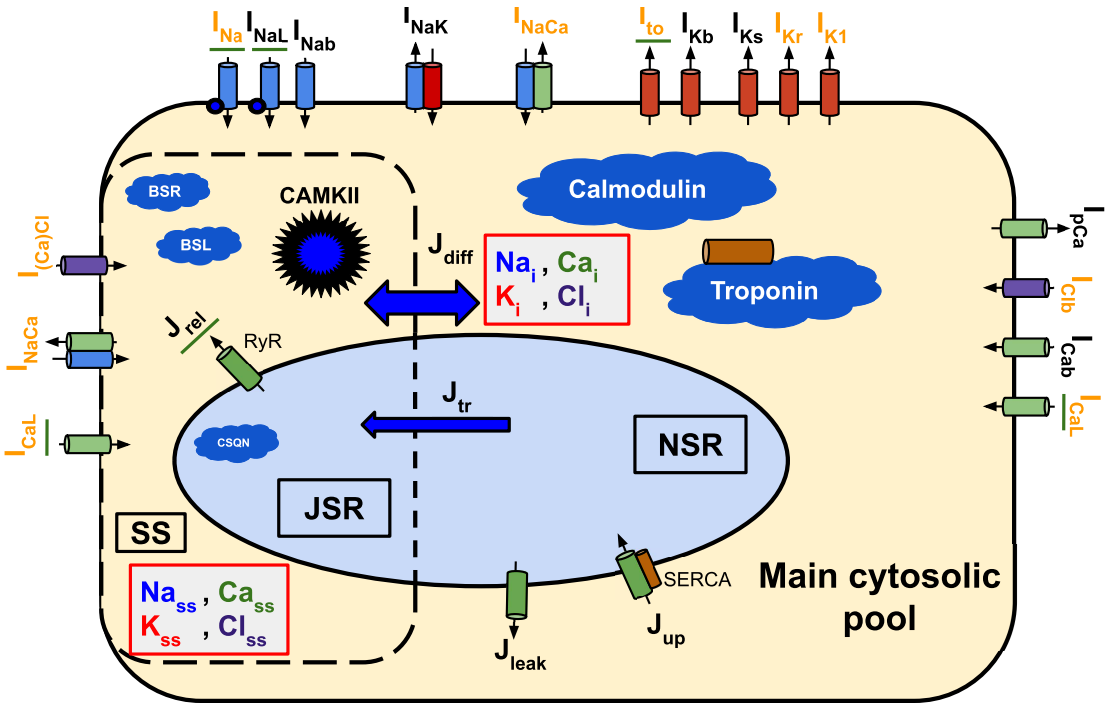


Figure 16 – Illustration with the main differences between the original *ToRORd-fkatp-2019* cellular model [72] and the new *ToRORd-dyncl-2020* model [73]. The main model compartments (main cytosolic pool, junctional subspace SS, and subcompartments of the sarcoplasmic reticulum), currents, fluxes, and buffers are depicted. Intracellular concentration of ions are listed in the main cytosolic pool and SS compartments using color labels within a grey box. Compared to the original model which dynamically updates concentrations of sodium, calcium, and potassium, *ToRORd-dyncl-2020* also dynamically updates chloride concentrations. Figure adapted from [73].

The updated *ToRORd-dyncl-2020* human ventricular model is composed by a system of ODEs with a total of 45 state variables, and just like the *ORd* and *ToRORd-fkatp-2019*, presents transmural heterogeneity in the cellular dynamics of endocardium, midmyocardial and epicardial ventricular cells. The baseline APs of the *ToRORd-dyncl-2020* model with a Basic-Cycle-Length (BCL) of 1000ms are shown in Figure 17 and illustrate the transmural cellular dynamics differences for each ventricular region.

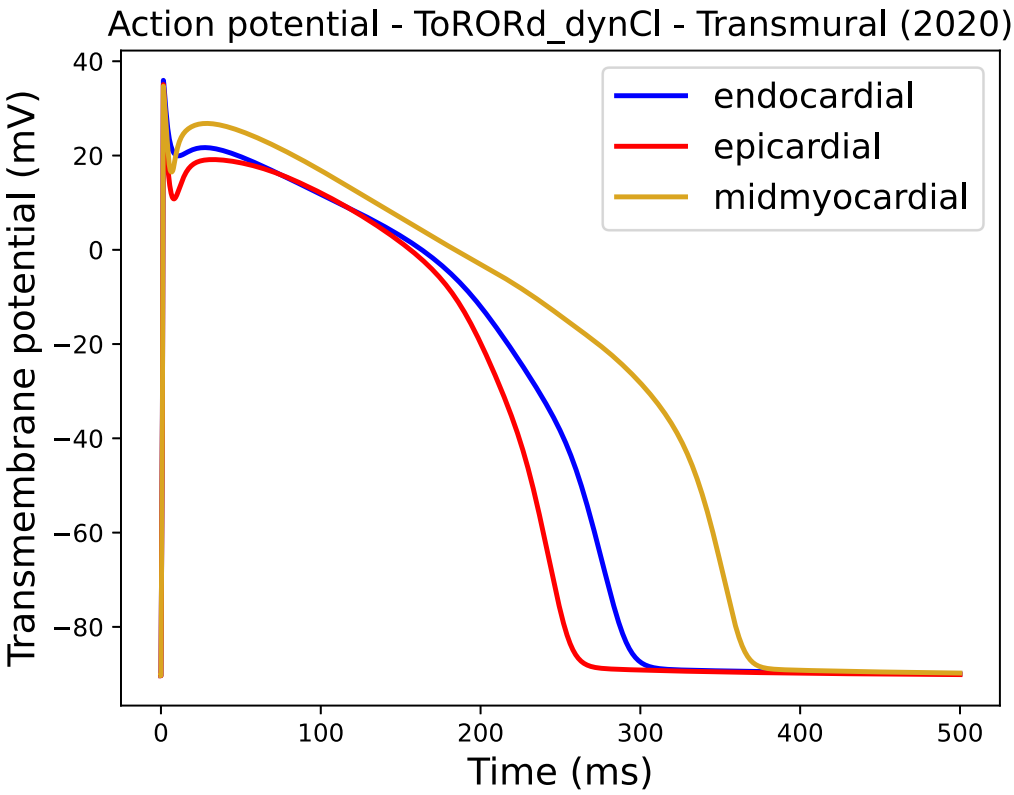


Figure 17 – Action potentials described by the *ToRORd-dynCl-2020* human ventricular cellular model with a basic cycle length of $1000ms$ [73], where the transmural heterogeneity of model is illustrated by their corresponding APs, blue (endocardium cell), red (epicardium cell) and orange (midmyocardial cell).

2.4.4 Explicit Euler and Rush-Larsen methods

To properly solve the non-linear system of ODEs given by a cardiac cellular model a numerical scheme must be applied. The explicit Euler method is a first order method to approximate solution of ODEs. Consider the ODE $\frac{dy}{dt} = f(t, y)$, where $f(t, y)$ is a function given and $y(t)$ is the function to be determined or approximated. By expanding $y(t)$ in a Taylor series around t

$$y(t+h) = y(t) + hf(y(t), t) + \frac{h^2}{2!}f^{(1)}(y(t), t) + \frac{h^3}{3!}f^{(2)}(y(t), t) + ..., \tag{2.31}$$

where h is the time step. Truncating the approximation $y(t+h)$ after the series terms which power of h are superior to 2 we achieve

$$\overline{y}(t+h) = y(t) + hf(y(t), t) = y(t+h) + O(h^2). \tag{2.32}$$

Given $y_0 = y(t_0)$ and $y_1 = \overline{y}(t_0 + h)$, for $n = 2, 3, ...$ is established by the Euler method that

$$y_{n+1} = y_n + hf(y_n, t_n), \tag{2.33}$$

where $t_n = t_0 + nh$. The error that arises from the approximation of $y(t_0 + nh)$ by y_n is order h. Furthermore, it can be verified that the explicit Euler method has stability conditioned to the step h used and the characteristics of the ODE to be approximated, being classified, therefore, as a conditionally stable method.

Such techniques, limit the size of the time steps taken in the simulation process, due to its stability conditions. Such a limitation is accentuated with the presence of *stiff* equations. In general a problem is called *stiff* when we attempt to compute a particular solution that is smooth and slowly varying (relative to the time interval of the computation), but in a context where the nearby solution curves are much more rapidly varying. In other words, if we perturb the solution slightly at any time, the resulting solution curve through the perturbed data has rapid variation [39].

The *Rush-Larsen* (RL) method [58] seeks to mitigate the problem of instability on certain types of equations present in electrophysiological models. When such equations constitute the *stiff* part of the system as a whole, as in the case of the recent models *Trovato2020* [74] and *ToRORd-dynCl-2020* [73], the method allows the use of considerably larger time steps, reducing the time of computing sharply [44, 22].

The RL method focus on the gating variables that are written using Hodgkin-Huxley's formulation [30]. This method considers the coefficients α_j and β_j of the gating variables equations, which have the form of equation (2.34), approximately constant in a small time interval.

$$\frac{dy}{dt} = \alpha_j(1 - y) - \beta_j y. \tag{2.34}$$

Therefore, the method consists of a local linearization of the gating variables equations (also called *quasi-linear* equations), which are numerically solved by

$$\begin{aligned} y_{n+1} = \left(y_n - \frac{\alpha_j}{\alpha_j + \beta_j}\right) e^{-(\alpha_j + \beta_j)h} + \frac{\alpha_j}{\alpha_j + \beta_j}, \\ k = |\alpha_j + \beta_j|, \end{aligned} \tag{2.35}$$

where h is the time step. The remaining equations of the model are evaluated by the Euler method. If the value of k, defined by equation (2.35), is close to zero, the Euler method is used instead of equation (2.35) for the corresponding gating variable at that point of the simulation.

For example, the Noble model defined in section 2.4.1 can be directly solved using the RL method by considering the gating variables m, h and n as Hodkin-Huxley type. Consequently, the

variable V defined by equation (2.21) is solved using an explicit Euler scheme, while variables m, h and n given by equations (2.28), (2.29) and (2.30), respectively, are solved using the RL method.

In equation (2.36) we show each state variable from the *ToRORd-dynCl-2020* model and the appropriate scheme used to solve the equations of the model, where θ_{euler} denotes the variables solved using an Explicit Euler scheme and θ_{RL} the ones being solved with the *Rush-Larsen* one.

$$\begin{aligned} \theta_{euler} = \{&v, CaMKt, nai, nass, ki, kss, cansr, cajsr, cai, \\ &nca_{ss}, nca_i, C1, C2, C3, I, O, cli, clss\} \\ \theta_{RL} = \{&m, h, j, hp, jp, mL, hL, hLp, a, iF, iS, ap, iFp, iSp, \\ &d, ff, fs, fcaf, fcas, jca, ffp, fcafp, xs1, xs2, Jrel_{np}, Jrel_p\}, \end{aligned} \tag{2.36}$$

Similarly, in equation (2.37) we show each state variable from the *Trovato2020* model and the appropriate scheme used to solve the equations of the model, where θ_{euler} denotes the variables solved using an Explicit Euler scheme and θ_{RL} the ones being solved with the *Rush-Larsen* one.

$$\begin{aligned} \theta_{euler} = \{&v, CaMKt, cass, nai, nasl, nass, ki, kss, ksl, \\ &cai, casl, cansr, cajsr, cacsr, Jrel1, Jrel2, nca, u\} \\ \theta_{RL} = \{&m, hf, hs, j, hsp, jp, mL, hL, hLp, a, i1, i2, d, ff, \\ &fs, fcaf, fcas, jca, ffp, fcafp, b, g, xrf, xrs, xs1, xs2, y, xk1\}, \end{aligned} \tag{2.37}$$

In addition to the first-order methods described by the Explicit Euler and RL schemes, it is available in the literature second-order and hybrid methods that can provide an increase in accuracy at the expense of an increase in computation time, like for example the second-order method provided by *Sundnes et al.* [69], also known as SAST2 [22]; and hybrid methods that consider not only the resolution using RL schemes but also applying uniformization techniques for the Markov chains associated to these models [22].

2.5 Cardiac tissue modelling

Cardiac tissue comprises a set of cells that connect to each other through gap junctions, which allows the passage of a cell's action potential to another. Among the models capable of describing this phenomenon, two famous models are available in the literature: the bidomain and monodomain.

2.5.1 Bidomain and monodomain models

The bidomain model considers that the cardiac tissue is composed of a set of cells, and the transmembrane potential of each cell depends on both intracellular and extracellular domains [35].

In this model, the intracellular cells are connected by gap junctions, while for the extracellular cells there is a current flowing in the spaces between each cell. In this way, for each domain, a potential and an electrical current are defined, where for the intracellular domain we call V_i and I_i for the potential and the intracellular current, respectively, and V_e and I_e for the potential and current of the extracellular domain, respectively. Figure 18 shows how the bidomain model is represented.

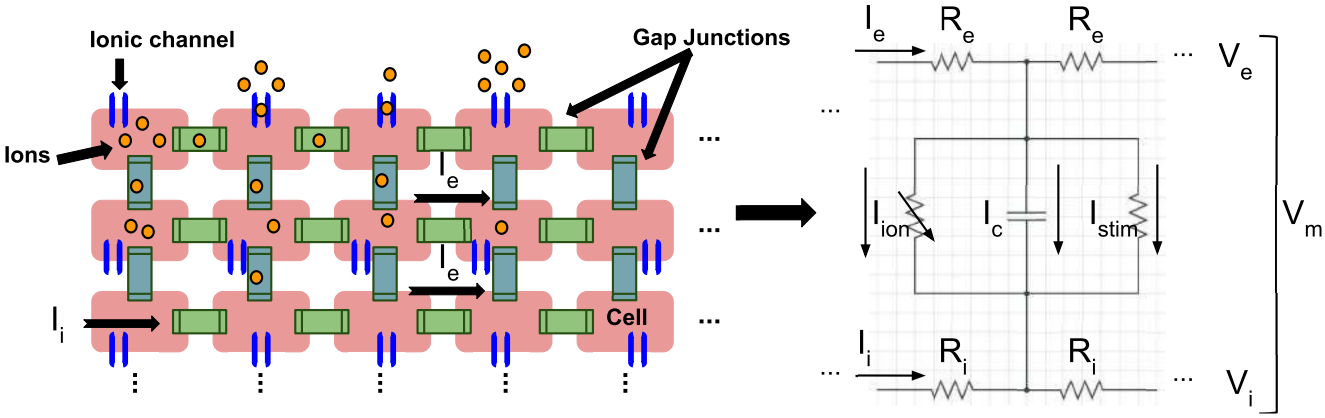


Figure 18 – Schematic of the bidomain model.

Using Ohm's law, the relationship between potential and current is obtained for each domain

$$\begin{aligned} I_i &= \boldsymbol{D_i} \nabla V_i, \\ I_e &= \boldsymbol{D_e} \nabla V_e, \end{aligned} \tag{2.38}$$

where $\boldsymbol{D_i}$ and $\boldsymbol{D_e}$ are intracellular and extracellular conductivity tensors, respectively. Furthermore, by the conservation of current principle, the total current at any point in the circuit will be conserved if no other external current is applied, so:

$$\nabla \cdot I_t = \nabla \cdot (\boldsymbol{D_i} \nabla V_i + \boldsymbol{D_e} \nabla V_e) = 0. \tag{2.39}$$

At each point in the domain, the transmembrane potential V_m and the transmembrane current I_m can be expressed by:

$$V_m = V_i - V_e, \tag{2.40}$$

$$I_m = \nabla \cdot (\boldsymbol{D_i} \nabla V_i) = -\nabla \cdot (\boldsymbol{D_e} \nabla V_e). \tag{2.41}$$

The transmembrane current I_m relates that all current leaving one domain must enter the other.

Now rearranging the equation (2.4) and converting the transmembrane current per unit of area to unit of volume through a factor β, which is the surface-to-volume ratio of the cell membrane, we arrive at the equation:

$$I_m = \beta \left(C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \right) = \nabla \cdot (\boldsymbol{D_i} \nabla V_i). \tag{2.42}$$

Using the expression $V_i = V_m + V_e$ in order to eliminate the intracellular potential V_i and include the extracellular potential V_e and the transmembrane potential V_m in the expressions (2.39) and (2.42)

$$\nabla \cdot (\boldsymbol{D_i} \nabla V_m) + \nabla \cdot (\boldsymbol{D_i} \nabla V_e) = \beta \left(C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \right), \tag{2.43}$$

$$\nabla \cdot (\boldsymbol{D_i} \nabla V_m + (\boldsymbol{D_i} + \boldsymbol{D_e}) \nabla V_e) = 0, \tag{2.44}$$

where the equations (2.43) and (2.44) are known as the bidomain model in its most used formulation for modeling electrical propagation in cardiac tissues [28].

The bidomain model is a system of partial differential equations (PDE), which requires a high computational cost. Thus, the model can be simplified considering that the extracellular environment does not affect the electrical activity, in such a way that the extracellular potential is equal to zero. As a result, the expression (2.40) becomes:

$$V_m = V_i. \tag{2.45}$$

Considering this simplification in the equation (2.43), the formulation of the monodomain model is expressed by:

$$\nabla \cdot (\boldsymbol{D_i} \nabla V_m) = \beta \left(C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \right). \tag{2.46}$$

If the propagation of the electrical stimulus occurs in a single direction, as is the case with Purkinje fibers, the above equation can be further simplified by

$$I_m = \sigma_x \frac{\partial^2 V_m}{\partial x^2} = \beta \left(C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \right), \tag{2.47}$$

where σ_x represents the conductivity of the Purkinje cells in the direction of the fiber propagation.

Although this model is not able to capture certain phenomena that the bidomain model can, the computational cost of the monodomain model is much lower. Therefore, it is a model widely used in several works [42, 55, 51].

Within this context, if we assume that the intracellular and extracellular domains are anisotropic, but to the same degree, the bidomain equations can be reduced to the monodomain equations. Consequently, the cardiac tissue can be simplified by a single conducting domain that relates $\boldsymbol{D_e} = \lambda \boldsymbol{D_i}$, where λ is a scalar. Using this expression we can rewrite the monodomain equation.

$$\nabla \cdot (\boldsymbol{D_m} \nabla V_m) = \beta \left(C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \right), \tag{2.48}$$

where $\boldsymbol{D_m}$ is the effective bulk conductivity. As a result of this expression it is possible to match conduction velocities between bidomain and monodomain formulations along the axial directions of the tissue as can be seen in more details in the works from *Bishop, M. J. and Plank, G* [7, 8] or in the work from *Sundnes, J. et al.* [70].

In both of these works it is shown that activation patterns including bath-loading effects observed only in bidomain models can also be reproduced with monodomain models in which conductivity tensors are further correctly chosen. In this case, the effective bulk conductivity tensor $\boldsymbol{D_m}$ in equation (2.48) is represented by $\boldsymbol{D_m^*}$, where $*$ means either b in the tissue bulk or s along the surfaces. If the propagation occurs along the individual principal axes, the eigenvalues of $\boldsymbol{D_m^*}$ are given as

$$\boldsymbol{D_m^{b\zeta}} = \frac{\boldsymbol{D_i^\zeta D_e^\zeta}}{\boldsymbol{D_i^\zeta + D_e^\zeta}}, \tag{2.49}$$

where $\zeta = l/t/n$ are the eigendirections of the tissue along the longitudinal cardiac fiber direction ($\zeta = l$), transverse ($\zeta = t$) to fibres within a sheet, and along a sheet normal direction ($\zeta = n$). A thin augmentation layer is defined by assuming that extracellular resistivity close to tissue surfaces is rather given by the bath conductivity $\boldsymbol{D_b}$ than the interstitial conductivity $\boldsymbol{D_e}$ which leads to a conductivity in the augmented layer given by:

$$\boldsymbol{D_m^{s\zeta}} = \frac{\boldsymbol{D_i^\zeta D_b}}{\boldsymbol{D_i^\zeta + D_b}}. \tag{2.50}$$

The above representation is also called *augmented monodomain* and is used to reproduce the bath-loading effects of a surrounding extracellular medium when wavefront morphologies and conduction velocities are considered [7, 8].

2.6 Pseudo-ECG

A major advantage of the *augmented monodomain* formulation, shown in the previous section, is that it can provide a similar activation sequence as the bidomain model with a much

cheaper computation cost and , most importantly, it can provide realistic ECG signals in a bounded medium surrounding the heart.

However, to calculate an approximation of the 12-leads-ECG we can simply use the monodomain formulation. Basically, the procedure consists of first compute the bioelectric activity of the heart using the monodomain model given by equation (2.46). Next, we can retrieve the ϕ_e surface potential if we assume that the tissue is immersed in an unbounded volume conductor during a standart monodomain simulation by solving the following equation:

$$\phi_e = \frac{1}{4\pi D_b} \int_\Omega \frac{\beta I_m}{||r||} d\Omega, \tag{2.51}$$

where r is the distance vector between source and field points, which are essentially the electrode positions of the 12-leads-ECG approximation. Using a standard monodomain simulation the source term βI_m is given by the solution of $\nabla.(\boldsymbol{D_i}\Delta V_m)$, which is available in every timestep.

To properly measure an 12-leads-ECG in a patient, a total of 10 physical electrodes, commonly known as leads, must be used. This 10 physical leads are placed over the body of the patient at certain locations as can be seen in Figure 19. In this figure, there are four limb leads, namely RA(right arm), RL(right left), LA(left arm), LL(left leg); and six chest leads, known as V1, V2, V3, V4, V5 and V6.

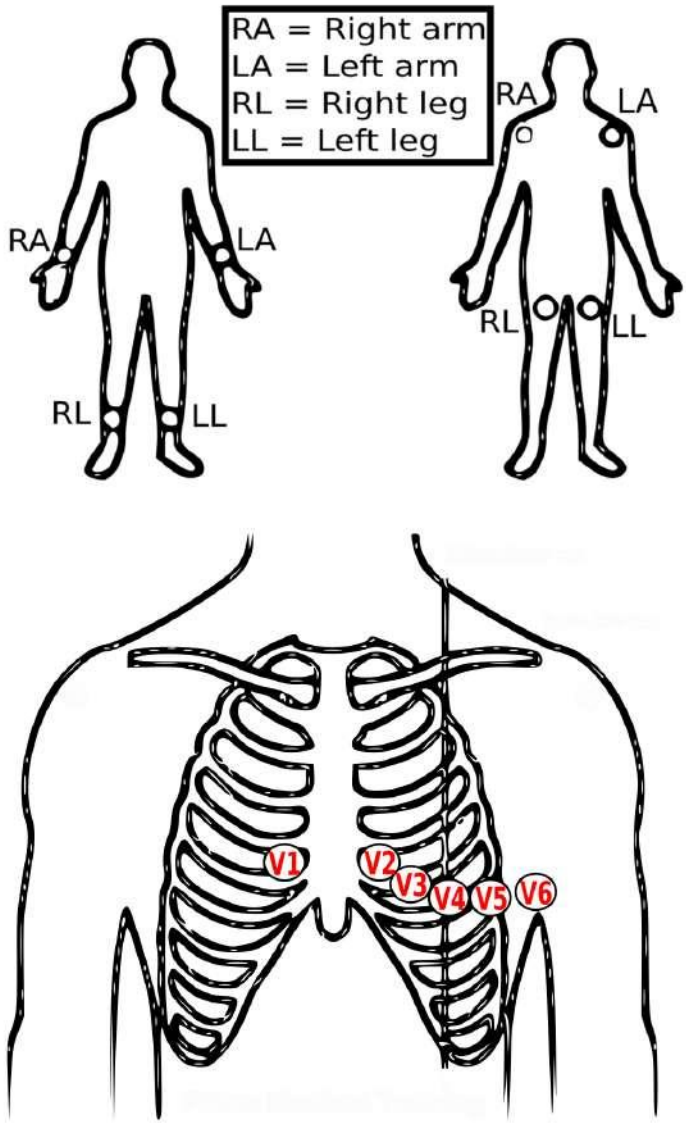


Figure 19 – Illustration of the 10 physical leads placement over a patient body for an ECG measurement.

When the ECG is calculated, we have at our disposal the signals coming from these 10 physical leads. However, the ECG readings are commonly represented to a medical person using 12 tracing leads. To convert the 10-leads to 12-leads ECG we must use the Einthoven Triangle shown in Figure 20, which retrieves the readings coming from the limb leads and generate six new leads, namely Lead-I, Lead-II, Lead-III, aVR, aVL and aVF. To calculate the new leads readings we use the following equations:

$$
\begin{aligned}
LeadI &= LA - RA \\
LeadII &= LL - RA \\
LeadIII &= LL - LA \\
aVL &= \frac{LeadI - LeadIII}{2} \\
aVR &= \frac{-LeadI - LeadII}{2} \\
aVF &= \frac{LeadII + LeadIII}{2}
\end{aligned}
\tag{2.52}
$$

After this procedure the 12-leads ECG is computed with the precordial leads given by V1, V2, V3, V4, V5, V6; and the frontal leads given by Lead-I, Lead-II, Lead-III, aVR, aVL and aVF.

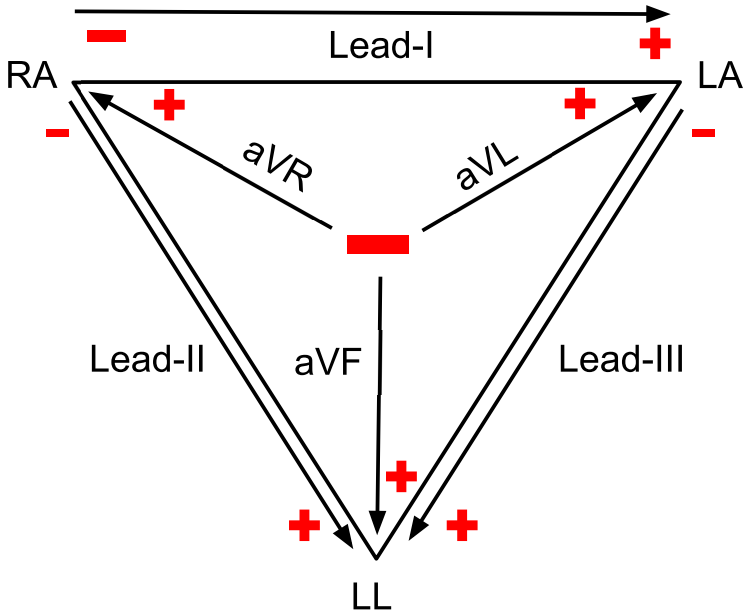


Figure 20 – Representation of the Einthoven Triangle utilized to generate the six additional leads for the 12-leads ECG. Normally, the RL electrode is consider the grounding lead and does not appear in this sketch.

Furthermore, the ECG readings are commonly normalized before presenting in a comparison study. In this work we normalize all the ECGs using the technique presented in the work from *Camps, J. et al.* [12]. In addition, only the readings from 8-leads (V1, V2, V3, V4, V5, V6, Lead-I, Lead-II) are utilized for comparison. The main idea of the normalization is to first average the values given by the leads LA, RA, LL and RL, then normalize the values of all leads to the maximum amplitude of each one. To compare two ECG readings we consider the computation of the Pearson correlation factor, which is given by the equation:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{2.53}$$

where $\rho_{X,Y}$ is referred to as the population correlation coefficient or the population Pearson correlation coefficient, cov is the covariance, σ_X is the standard deviation of the random variable X and σ_Y is the standard deviation of the random variable Y.

2.7 Geodesic paths

To generate Purkinje networks sustaining its structure over the endocardium surface, the usage of the geodesic path algorithm is commonly utilized [13, 48], especially when the endocardium surface is extremely irregular and have several particularities, like for examples tendons and papillary muscles. In addition, the shortest geodesic paths is a common operation in many algorithm that tackles problems in the field of computer science. In general this operation is commonly used in the area of computer graphics, where triangular meshes are used to represent a given object. Geodesic paths provide solution to several problems, like mesh parametrization [80], shape matching and classification [29] and shape retrieval [57].

The geodesic path problem can be stated as: find the shortest path between two points on the surface of a polyhedron. A shortest path $\pi(s,t)$ between s and t is defined as a path of minimum Euclidean length among all possible paths joining s to t, constrained to lie on the surface of the polyhedron. Moreover, the length of $\pi(s,t)$ is defined as the sum of the lengths of all segments on the faces which the path traverses [64]. In Figure 21, we show an example of computational geodesic path constructed over a simplified surface which represents the structure of the left ventricle.

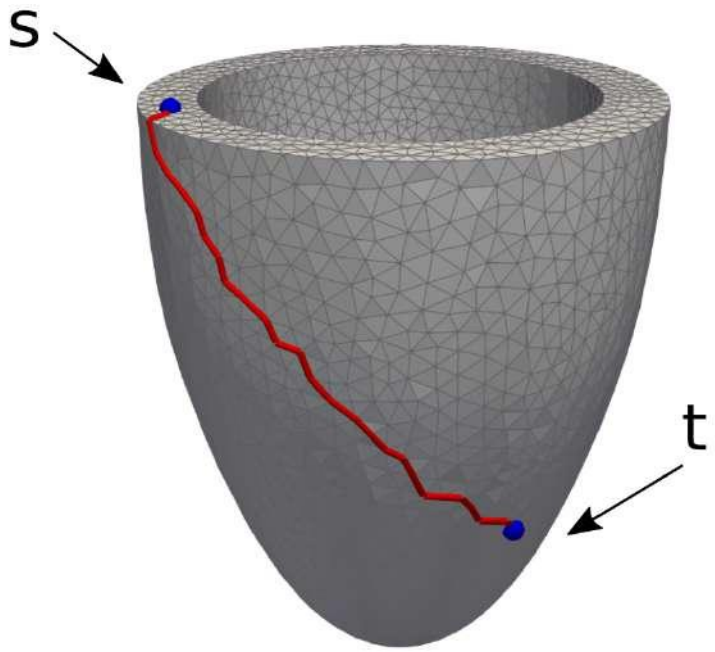


Figure 21 – Illustration of a geodesic path build over a simplified representation of the left ventricle. The given mesh is represented by triangles and the source s and target t point of the geodesic path $\pi(s,t)$ are highlighted in blue. The pathway $\pi(s,t)$ is colored in red and is the shortest path between s and t within the surface.

A wide range of algorithms for computing geodesic paths handle the single source variant of the problem, which seeks to determine shortest paths from a source vertex to all other vertices of the polyhedron. Basically, all algorithms that employ a graph to discretize the paths are divided in two stages: building such a graph and computing the shortest geodesic paths. A simple way of building the graph is to consider the input triangular mesh as the graph itself. For each triangular cell in the mesh, the connections between the points that represent the cell are parsed to nodes and edges in a graph data structure. Once the graph is built for the whole mesh, the second stage can be performed by executing a shortest path algorithm from any source vertex [64].

An optimization aspect of this variational problem is that when the triangular mesh does not change over time, the associated surface graph needs to be constructed only once. This feature becomes valuable in applications where the computation of paths from multiple sources in an unchanged surface is required, which is the scenario associated to the Purkinje network generation over an endocardium surface problem.

In this work to compute geodesic paths we use the VTK library [63] implementation written in *C++* and described in the *vtkDijkstraGraphGeodesicPath* class. The main idea of this implementation is to first load the associated surface, which must be given by triangles, and convert it to a *Polydata* object. Next, the associated surface graph is construct using the triangles of the mesh only once. After the main structure of the graph is built, any subsequent geodesic path is

computed by simply computing the single source shortest path calculus using the Dijkstra algorithm passing the location of the source and target points within the surface. The implementation is similar to the one described in *Cormen, T. H. et al.* [15]. Some minor enhancement are added though. All vertices are not pushed on the minimum heap at start, instead a front set is maintained. Furthermore, the heap is implemented as a binary heap. The output of the operation is a set of lines describing the shortest path from s to t.

In terms of computational cost, the time to build the associated surface graph is proportional to the number of triangles in the surface, which can highly affect the resultant execution time if the given endocardium mesh is extremely detailed. After the associated graph is constructed and supposing that it is given by a directed graph $G(V, E)$, with a total of V vertices and E edges, the total running time of querying a Dijkstra single shortest path search is $O(E.logV)$ if all vertices are reachable from the source and the graph is sufficiently sparse, in particular, $E = o(V^2/logV)$, which is exactly the case for G. For more details regarding the performance of the Dijkstra algorithm, please refer to *Cormen, T. H. et al.* [15].

2.8 Literature review

Purkinje networks are known to have a complex morphology with several branches and bifurcations, as can be seen in physiological images [42, 48, 67, 52]. A wide variety of methods have been used to address the generation of realistic PNs. In particular, PNs can be generated using image processing techniques by extracting the structure from images of dissected ventricles and them projecting these flat networks onto realistic endocardial surfaces [42]. Such models gather both the geometry and the Local Activation Time (LAT) from the processed network. On the other hand, the generated PN may not be acceptable for other subjects due to biological variability. Another set of widely utilized methods for PN generation is fractal trees. In such models, the L-System is commonly described as an alternative to automatically produce PNs using a pre-defined set of rules that can not only prevent collisions [31] but also enhance its geometry [65]. An alternative to the L-System is the fractal method proposed by *Costabal, F. S. et al.* [16] which allows the automatic generation of PNs using controllable curvature of the branches, enhancing the geometry of the tree, especially in irregular surfaces. Moreover, fractal trees and image processing techniques can be combined to construct the VCS. For instance, in the work of *Bordas, R. et al.* [9], from Magnetic Resonance Images (MRI) of a rabbit, it was possible to identify the Purkinje system partially. After manually constructing the His-Bundle, a fractal method was applied to extend the PN to the myocardium.

Another class of methods is based on optimization principles [76] and was inspired by the Constructive Constrained Optimization (CCO) method that can generate detailed and realistic vascular trees [62]. A major advantage of this method is the flexibility of using any cost function

in the method's optimization process and including as many topological and electrical metrics as needed (e.g., bifurcation angles, PVJ location, LAT).

In addition to these observations, important aspects of the Purkinje system alongside the proper activation of PVJ sites still need to be further investigated as can be seen in the work from *Behradfar, E. et al.* [6]. Mainly, it is not entirely clear how many PVJs are actually active in a Purkinje network. Within this context, evaluating different Purkinje network morphologies and distinct PVJ density clouds can be a valuable study to investigate their role in not only phenomena associated to their ventricular activation, like the resultant eletrocardiogram, but also help the understanding of reentry dynamics which can appear around these structures and can ultimately evolve to ventricular arrhythmias.

The main contribution of this thesis is the development of a novel method based on optimization principles to generate patient-specific PNs with geometrical and electrical accuracy. The new method, named *Shocker*, is an extension of the CCO algorithm and considers cost functions that rely on electrical and geometrical metrics and focus on generating patient-specific PN models. We have evaluated the generated PN models by coupling them to biventricular meshes and comparing the LATs obtained in the simulations to different references. The simulations are based on modern cellular electrophysiology models for human Purkinje and ventricular cells and a fast parallel cardiac simulator. Our results show that the generated patient-specific PNs can accurately reproduce important geometrical and electrical features. In addition, the new PN models also correctly reproduced the physiological delay at the Purkinje-Ventricular-Junctions. Therefore, we believe the results presented in this work are an essential step towards a better understanding of Purkinje fibers and provide a valuable tool to study the role of patient-specific models and their impact, for instance, in the simulations of cardiac arrhythmias.

3 Materials and methods

3.1 Generation of Purkinje Networks

The Constructive Optimization (CO), a method developed to generate Purkinje networks [76] in tri-dimensional domains following the minimization of a cost function that computes the total length of the network, is the foundation to develop the PN models of this work. The method uses as input a given cloud of points representing the surface to be covered, an initial root position, and a prescribed cost function that will be minimized. Then, the method tries to generate a tree that homogeneously places branches that satisfy a set of restrictions and, at the same time, minimizes the user-specified cost function.

The main structure of our novel method is shown in Algorithm 1. To generate the PN, the input data of the method is the following: set of points S with distal locations for the terminal branches, proximal location of the root branch x_{prox}, which must be within the endocardium surface Ω_s. In addition, set S can contain the locations of the active PVJs and an extra cloud of passive points that homogeneously cover Ω_s.

Algorithm 1: *Shocker* main program.

Data: S, x_{prox}, [*initial PN*].
Result: Purkinje network generated within the set of points S.

S_p, $S_a \leftarrow$ *PreProcessing*(S, x_{prox}) ;
$k_{term} \leftarrow$ *RootPlacement*(S_p, l_d, x_{prox}, *[initial PN]*) ;
while *(not pass one time through* S_p*)* **do**
    New passive branch $\leftarrow$ *GenerateTerminal*(S_p, N_p, CF_p, l_d, k_{term}) ;
    Advance to next passive point in S_p ;
    $k_{term} \leftarrow k_{term} + 1$;
    if *(*k_{term} % $L_{rate} == 0$*)* **then**
        New active branches $\leftarrow$ *AttemptPVJConnection*(S_a, N_a, CF_a, L_{error}, l_d, k_{term}) ;
    end
end
PostProcessing(S_a, N_a, CF_a, L_{error}, l_d, k_{term}) ;
Compute metrics and save network topology to a file ;

Another feature that the method provides to the user is passing a given PN configuration as the initial topology of the tree. It can be advantageous in two scenarios. Firstly, well-known structures of the Purkinje system, like the Left-Bundle-Branch (LBB) and Right-Bundle-Branch (RBB), can be constructed beforehand using different techniques and be supplied as the initial network in order to provide better guidance of the regions that the PN certainly will occupy based on physiological observations. The second advantage of this feature is extending an already

constructed PN by including different branches on its topology. For instance, these PNs could come as the output of another method already available in the literature. Thus, the user could want to increase its endocardium coverage or connect additional PVJs points, which is a feature that the available methods to generate PN in the literature currently do not have implemented.

Moreover, the PN is represented with a graph data structure, where the nodes represent points over the surface, and the edges, segments that link two points. Based on this type of data structure, each segment have access to its parent and left/right offsprings. Besides, a terminal segment is defined as a segment with no offspring, and a branch is a set of segments between bifurcations. For more details related to the geometrical concepts and data structures used by the *Shocker* method please refer to Appendix C.

Initially, before growing the PN, a *PreProcessing* step must be executed over the surface points that will be covered as highlighted in line 1. Given the endocardium surface Ω_s, we extract all the points and store them in a set S. Secondly, a procedure that reorders the indexes of the points in S is executed and is illustrated in Figure 23A. This operation considers the initial root position, x_{prox}, passed as an input parameter to reorder the points in S to such a degree that points that are close to the initial root position will be renumbered at the early indexes of the set. To reorder the points, a sphere centered at the given root position and with an initial radius $r_0 = 0.01mm$ grows in intervals of $r_{inc} = 0.05mm$ until all the points in the set S are covered. At each iteration, the points within the growing sphere have their indexes renumbered consecutively. In addition, the amount of passive points that the method will use is filtered by 1% of the total number of passive points inside S. We randomly select passive points inside S until the prescribed quantity is achieved. Ultimately, the result of this process is a new set S_p, which is the passive cloud of points that the method mainly utilizes. The main advantage of performing this reordering over the points in S is that the growth of the PN will start near the region of the root and slowly cover the other regions of the endocardium.

The points in S_p are considered passive PVJs, but the user can also provide an input set of active PVJs points with their corresponding Local Activation Time (LAT) inside the set S. This active points set, S_a, contains all the points that the PN must connect with a given LAT. If the S_a set is supplied, we pre-process the PVJs by sorting them concerning their LAT. This distinction between passive and active PVJs sets is essential since it will guide the execution of the main logic of the method. A detailed explanation regarding the *PreProcessing* and subsequently subroutines are given in the Appendix section A.1.

After this initial step, the method constructs the root branch in line 2 given the initial root position x_{prox} and the characteristic length l_d. A distal location x_{term} is selected from the set S_p and must attend some geometrical restrictions. After selecting a feasible position, a geodesic pathway, which connects x_{prox} to x_{term}, is constructed over Ω_s. Alternatively, the user can also

provide an initial root structure that can be used to initialize the method. For more information about the *RootPlacement* and associated subroutines, please refer to the Appendix section A.2.

To generate a new terminal branch in the network, a distal location x_{term} can be selected in two different scenarios. In case the new terminal branch is considered passive, we randomly select a point from the set S_p. On the other hand, if the new terminal branch is an active PVJ we retrieve x_{term} from the set S_a. The prospective location x_{term} is only connected to the PN if satisfies a distance criterion.

The distance criterion consists of checking whether the distance d_{crit} of a new candidate terminal location x_{term} exceeds an adaptive threshold (d_{thresh}). This threshold is dynamically decreased when the number of terminal branches increases during the process of PN generation. The same 3D formulation utilized in *Ulysses, J. N. et al.* [76] was applied in the present work. Before adding a new terminal branch, the threshold distance d_{thresh} is initially given by:

$$d_{thresh} = \sqrt{\frac{l_d^2}{k_{term}}}, \tag{3.1}$$

where k_{term} is the current number of terminals in the PN and l_d is the characteristic length of the domain given in micrometers. In Figure 23B it is illustrated how the threshold distance d_{thresh} varies as the number of terminals in the PN, k_{term}, increases considering $l_d = 10mm$. It is important to notice that as the network grows the threshold distance decreases as more terminals are connected to the tree. This behaviour leads to longer branches in the early iterations of the method, as the PN tries to cover as much as the domain. Consequently, shorter branches start to appear during the last iterations. All the steps for computing the distance criterion for a given x_{term} location are described in details in Appendix section A.3.

After the root is placed, the method enters in the main loop in line 3. At each iteration, a point x_{term} from S_p is selected and a new passive branch is generated to the PN by calling the *GenerateTerminal* subroutine in line 4. This procedure is repeated until the last passive point in S_p is reached. To generate a new passive branch, the point x_{term} must first attend the distance criterion, then it is temporarily connected to the nearest N_p segments in the current PN by a geodesic pathway. The calculus of the N_p nearest segments is done by calculating the distance between the middle point of all segments in the current network to x_{term}. Next, we sort the segments by their Euclidean distance and filter only the closest N_p segments to x_{term}. After this step, we decide how to connect x_{term} to one of the N_p neighboring segments. This choice is based on the minimization of the passive cost function CF_p that relates the total length of the network accordingly to equation (3.2):

$$CF_p = \sum_{k=0}^{N_{seg}} l_k, \tag{3.2}$$

where l_k is the length of segment k and N_{seg} is the total number of segments in the model. The subroutine associated to the evaluation of a given cost function is explained in Appendix section A.4.

Finally, the candidate connection with the lowest value of CF_p and whose connection does not generate a collision to other segments in the tree is adopted and made permanent, as illustrated in Figures 23C and 23D. In case all feasible segments cause collisions, we sort a different x_{term} from S_p and recheck the distance criterion. After a suitable passive branch is included to the tree we advance to the next passive point in S_p and increase the number of terminal branches k_{term} in the PN by one, as lines 5 and 6 show. A more detailed explanation regarding the *GenerateTerminal* and associated subroutines to generate a new passive branch to the tree are described in Appendix section A.5.

During the main loop we attempt to connect the active PVJs present in S_a when k_{term} is divisible by a connection rate parameter L_{rate}, as presented in line 7 with a calling to the *AttemptPVJConnection* subroutine. In this scenario, we try to sequentially connect all the unconnected PVJs inside the set S_a using the same technique previously described for the passive case. For each active PVJ, x_{PVJ}, inside S_a we use the nearest N_a segments sorted by an approximation of their LAT error alongside the active cost function CF_a which calculates the LAT error of a given segment using the cable equation. The active cost function is shown in Eq. (3.3).

$$CF_a = |LAT(s) - T(PVJ)|, \tag{3.3}$$

where s is a terminal segment linked to an active PVJ, $LAT(s)$ is a function that returns the LAT of a given segment s using the cable equation, and $T(PVJ)$ is the known value for the LAT of the PVJ. More details regarding the cable equation can be found in the Appendix B.

The cable equation is commonly used to evaluate the electrical flow in neurons and cardiac cells [35]. In this assumption, the potential depends only on the length variable so that the cable can be viewed as one-dimensional, just like a Purkinje fiber. The cell is considered a cylindrical piece of the membrane with a certain length, diameter, internal conductivity, and membrane capacitance. The conduction velocity (CV) across the cable is given by

$$CV = \sqrt{\frac{Gd}{4C_f\tau_f}}, \tag{3.4}$$

where G is the internal conductivity, C_f is the membrane capacitance, τ_f is a time constant, and d is the diameter of the cable. The cable properties of a Purkinje fiber can be measured and typical values are given in the work from *Schoenberg, M. et al.* [61], $G = 7.9m\Omega/cm$, $C_f = 3.4\mu F/cm^2$, $\tau_f = 0.1ms$. The diameter of a Purkinje fiber is reported to vary between $50\mu m$ to $300\mu m$ with a propagation velocity ranging from $1m/s$ to $4m/s$ [19, 61, 56].

In Figure 22 it is highlighted how the LAT computation of a segment in the tree is done using the cable equation. As can be seen in the figure, for each segment that is in the pathway between the target segment and the root node, which is illustrated by the red arrows, the cable equation is applied. To access the segments within the pathway, the parent pointer from the graph data structure of the PN is utilized.

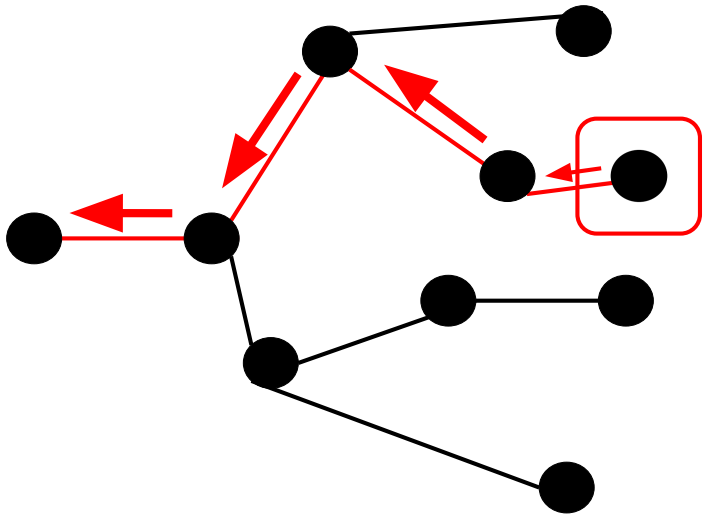


Figure 22 – Illustration of the LAT computation using the cable equation for a given segment in the tree. The cable equation is applied in each segment that is in the pathway linking the target segment until the root node using the parent pointer from the graph data structure that represents the PN, as depicted by the red arrows.

When the active cost function CF_a is calculated for a given segment, the connection is only accepted if the new branch has an absolute LAT error between the best candidate branch found during the evaluation phase and the reference LAT value of the target active PVJ is less or equal to L_{error}. Otherwise, the PVJ returns to set S_a and the entire branch is pruned. Moreover, whenever a PVJ is connected to the PN, the procedure is repeated until there are no updates to the tree.

Similarly to the passive case, the selection of the N_a best segments is done by passing through all the segments and computing an approximation of the LAT error to x_{PVJ} reference value. This approximation is calculated by the sum of the current LAT of a given segment s_i and the LAT given by the line that links the middle position x_M of segment s_i to x_{PVJ}. After this, the geodesic paths are computed for the N_a segments and the LAT error is recalculated. The whole procedure of connecting an active PVJ, x_{PVJ}, is illustrated in Figures 23E and 23F. A detailed explanation about the connection of the active PVJs is presented in Appendix A.6.

Finally, after we reach the end of S_p, a *PostProcessing* step is applied and illustrated in Figures 23G and 23H. Any remaining active PVJs that were not connected in the main section of the method are handled in this step. Firstly, we prune all passive segments of the tree. A segment

is considered passive if it is not directly part of a pathway that links an active PVJ to the root. Next, the LAT error tolerance constraint is dropped by setting $L_{error} = \infty$, and we attempt to connect all unconnected PVJs in S_a using the pruned tree with the N_a best segments considering the LAT error. However, after this procedure some PVJs still could not be connected due to the distance criterion. This scenario could happen if x_{PVJ} is already close to the current tree. In this particular case, we drop the distance criterion for these PVJs and attempt to connect the point with a feasible segment which returns the minimum LAT error using a geodesic pathway. If there are no geodesic pathway possible for x_{PVJ}, we consider the 5 closest segments by distance to x_{PVJ} and force its connection to the one that returns the minimum LAT error using a straight line, regardless of any restriction. The whole *PostProcessing* subroutine is available in the Appendix section A.7.

After this step, all the active PVJs that the user specified are connected and the geometric and electric metrics are computed in this topology, which is referred to as *minimum network*. More information and details regarding all the steps of the *Shocker* method are supplied in the Appendix A.

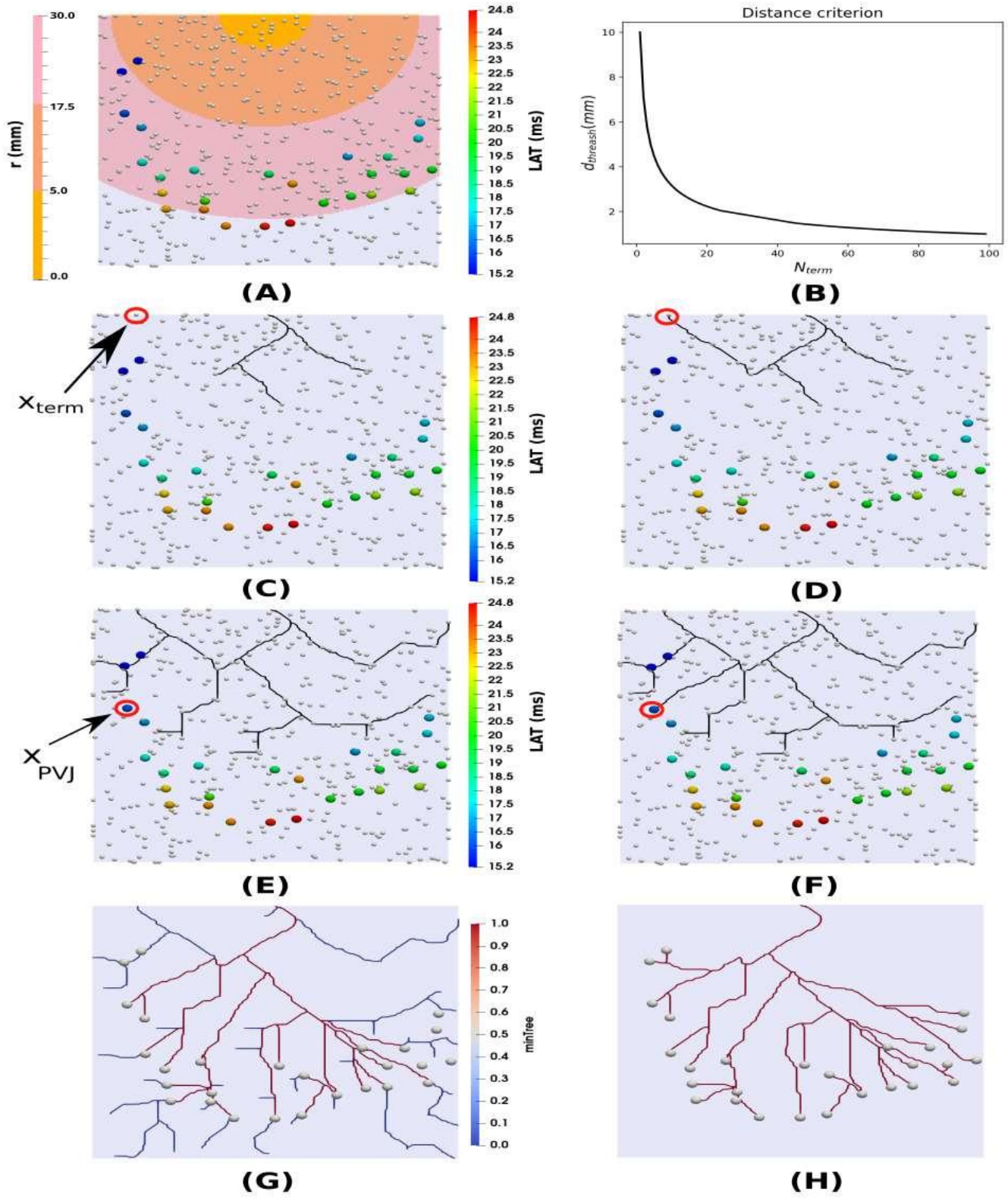


Figure 23 – Summary of the *Shocker* method. In panels (A) and (B), the *Pre-processing* step is highlighted showing how the passive points are renumbered and the effect of the distance criterion in the d_{thresh} evolution, as more terminals are added to the PN, respectively. Next in panels (C) and (D), an example connection for a passive terminal, x_{term}, is depicted, while in panels (E) and (F), a connection considering an active PVJ, x_{PVJ}, is illustrated. Finally in panels (G) and (H), the *Post-processing* step is shown illustrating the pruning of the passive branches and connection of the remaining active PVJs.

3.2 Biventricular meshes

In order to evaluate the robustness of our method for PN generation, four different meshes were selected. The first one considers a simplified mesh that captures the main structures of the biventricular system which is depicted in Figure 24A. Secondly, a canine mesh is the target of the study as can be seen in Figure 24B. Lastly, two human patient-specific meshes are used as the final experiment to evaluate our model as shown in Figures 24C and 24D.

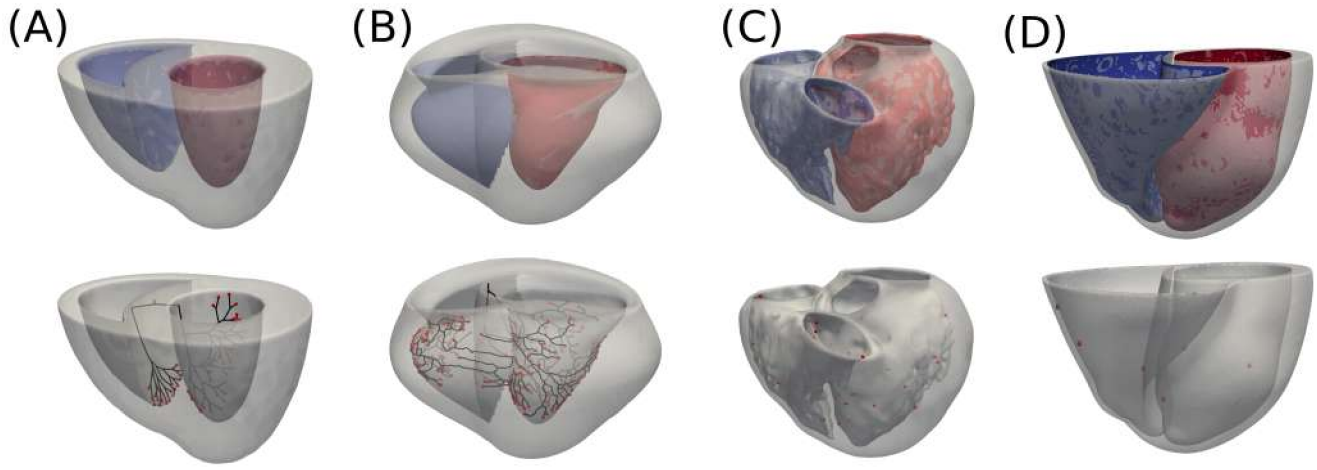


Figure 24 – Biventricular meshes used during the experiments of the present work. In panel (A), the simplified mesh alongside the corresponding reference PN (black) and their active PVJs points (red). In panel (B), the canine mesh from *Liu, B. R. et al.* [42] is highlighted with the gold standard PN (black) and their active PVJs points (red). In panel (C), the patient-specific mesh from *Lopez-Perez, A. et al.* [43] is depicted with their active PVJs points (red). In panel (D), the patient-specific mesh from *Camps, J. et al.* [12] is illustrated with their active PVJs points (red). The left ventricle is colored red in all the upper panels, while the right ventricle is blue.

The first mesh is a biventricular mesh constructed by taking the differences between three ellipsoids, one for the Left ventricle (LV) and Right ventricle (RV) endocardium and one for the epicardium. The main idea consists of subtracting and cutting their surfaces along the base plane, as can be visualized in Figure 24A. Fiber orientation and transmural distance of the mesh were calculated using the Laplace–Dirichlet Rule-Based (LDRB) algorithm [5] from the open-source library available at `https://github.com/finsberg/ldrb`. In addition, a simplified PN was generated using a fractal method [16], which is publicly available at `https://github.com/fsahli/fractal-tree`. This simplified PN was considered as our reference for comparisons. The branches of the reference PN activate the endocardium at the early region sites of the LV and RV as reported in the literature [16, 18]. For instance, these regions are close to the apex in the left and right ventricles and the posterior-basal region of the left ventricle. The

root locations from both ventricles and the His-bundle were manually selected and constructed to provide a septum activation pattern similar to what is observed physiologically [18]. Furthermore, all the terminals of the reference PN are considered active PVJs, representing a total of 43 and 20 PVJs in the LV and RV, respectively. More details regarding the construction and configuration of the Simplified mesh are available in Appendix D.

A biventricular canine mesh which had its histological PN reconstructed from two-dimensional photographs [42] and later processed into a graph representation [76] was considered as our reference for comparison as can be visualized in Figure 24B. The canine mesh was kindly provided by professor Elizabeth Cherry and it was first used on the study [42] and later, in collaboration with our research group, in the following study [76]. In the previous work proposed by *Ulysses, J. N. et al.*, [76] only the LV was considered in the experiments. In the present work, both ventricles are used to better evaluate the new proposed method. Similar to what was done in the simplified mesh, the fiber orientation and transmural distance of the mesh were calculated using the LDRB algorithm [5]. Furthermore, since only the reference PNs of the LV and RV were available, we manually built the His-bundle structure by linking the root points from the networks of the two ventricles. The provided data has a total of 130 and 98 PVJs over the LV and RV, respectively. In addition, these PVJs are the source of stimulation for the biventricular tissue.

Furthermore, two different human patient-specific biventricular meshes were utilized in this work to further evaluate our novel method in a more realistic scenario. The first human patient-specific mesh was kindly provided by professor Rafael Sebastian and the mesh has already been utilized in some works in the literature [43, 48] and is illustrated by Figure 24C. Alongside the mesh, the locations and LAT from the active PVJs are estimated using electroanatomical maps (EAM), and contact-mapping catheter systems (CARTO3 ^{TM}, Webster BioSense Inc.) [4, 20]. Counting 31 and 16 active PVJs points in the LV and RV surfaces, respectively. For this particular mesh, fiber orientation and transmurality are already available. By the term transmurality, we consider the different phenotypes of the cells across the ventricular wall, which are the epicardium, mid-myocardium and endocardium cells. Furthermore, the mesh originally contains a large ischemic and fibrotic region around the anterior LV and septum areas, which was not considered during the experiments in this study. Regarding the EAM points, a set of points located in healthy regions of the LV and RV were selected for validation.

The second human patient-specific mesh depicted in Figure 24D is utilized to investigate how different Purkinje networks can affect the ECG readings. The mesh was kindly provided by professor Blanca Rodriguez's team and it was first introduced in the work of *Camps, J. et al.* [12]. Alongside the mesh, the locations and LAT of 6 active PVJs, also referred to as root nodes, were estimated in order to approximate the available clinical ECG of the patient. There are 3 active PVJs in each of the ventricles. Together with the PVJs, a minimal Purkinje tree constructed using the method from *Miralles, F. B.* [4], which activates the PVJs approximately at the required LAT,

is used to evaluate the ability of our method to extend an already constructed Purkinje network by including extra branches on its structure. Similarly to the previous mesh, fiber orientation and transmurality are already available. The subject is considered to be healthy and no fibrotic or ischemic regions are present in the mesh.

An essential aspect of the experiments using the patient-specific meshes is that we do not have a reference PN for comparison; only the early activation sites, some EAM points in the endocardium and a clinical ECG are available.

3.3 Purkinje-Ventricular-Junctions

To model the Purkinje-Ventricular-Junctions (PVJs) sites, an additional current, I_{PVJ}^{tiss}, was included on the tissue cells that are coupled to the terminal cells from the PN as described in the following equation:

$$I_{PVJ}^{tiss} = \sum_{i=0}^{N_{PVJ}} \frac{(V^{purk} - V_i^{tiss})}{R_{PVJ}}, \tag{3.5}$$

where V^{purk} is the transmembrane potential of the terminal Purkinje cells, V_i^{tiss} is the transmembrane potential of the ventricular cells i attached to the PC, R_{PVJ} is a fixed-resistance and N_{PVJ} is the maximum number of VCs that are inside the PVJ site [10]. This additional current is included on the right hand side of the associate linear system of the ventricular domain only for the VCs related to the coupling.

Similarly, for the PCs a current, I_{PVJ}^{purk}, coming from the VCs that are coupled to the terminal PC, is included on the right hand side of the associate linear system of the Purkinje domain only for the PC related to the coupling and is given by the following equation:

$$I_{PVJ}^{purk} = \sum_{i=0}^{N_{PVJ}} \frac{(V_i^{tiss} - V^{purk})}{R_{PVJ}}, \tag{3.6}$$

In Figure 25, we illustrate how the Purkinje coupling was implemented.

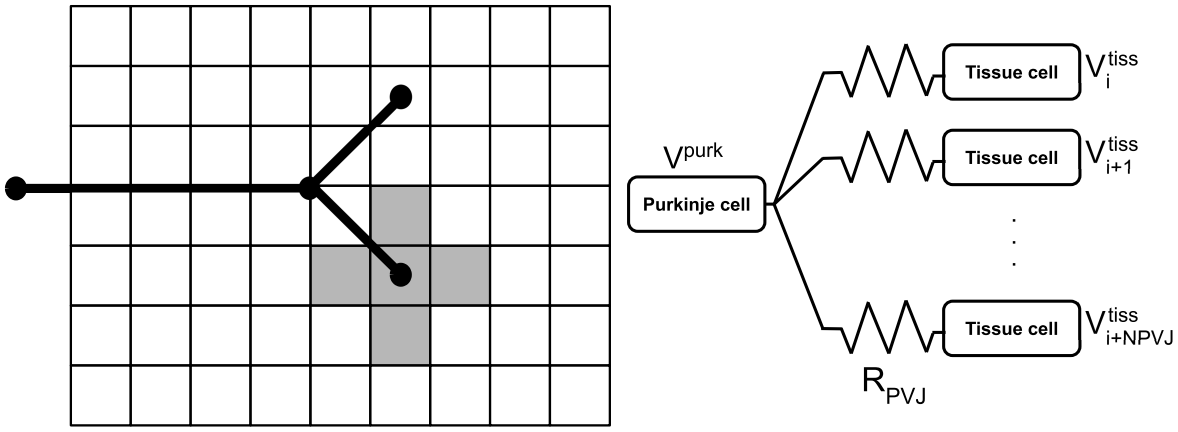


Figure 25 – Illustration of the Purkinje coupling model. The terminal PCs are linked to the nearest N_{PVJ} VCs by a fixed resistance R_{PVJ} and an additional current I_{PVJ} is computed following equations (3.5) and (3.6) for only the Purkinje and ventricular cells that are related to the coupling.

The parameters R_{PVJ} and N_{PVJ} must be properly calibrated in order to reproduce a physiological value for the anterograde characteristic delay that occur at the PVJ sites, which is between $3ms$ to $5ms$ [79]. Within this context, a tuning experiment was made using monodomain simulations to evaluated how the delay is affected by these two parameters and is illustrated in Figure 26. The idea of the experiment is to couple a single $10cm$ Purkinje cable to a small ventricular mass given by a cuboid volume, where the PCs and VCs are modeled by the *Trovato2020* and *ToRORd-dynCl-2020*, respectively.

Regarding the parameters of the monodomain model, the surface-to-volume ratio is set to $\beta = 0.14\mu m^{-1}$, tissue capacitance is equal to $C_m = 100pF/\mu^2$. A total simulation time of $t_{max} = 100ms$ was used and a time discretization $dt = 0.02ms$ to solve the associated PDE. For the ODEs system related to the Purkinje and ventricular cellular models, a RL scheme with a fixed timestep of $dt = 0.01$ was utilized. In addition, VCs are all considered to be endocardium and no fiber orientation is set in the cuboid mesh. A space discretization of $400\mu m$ was used for the ventricular cuboid, while for the Purkinje this value was set to $100\mu m$. The conductivities from the ventricular domain are anisotropic with $\sigma_l = 0.75S/m$, $\sigma_t = 0.225S/m$ and $\sigma_n = 0.1125S/m$, in the longitudinal, transverse and normal direction, respectively, resulting on CV approximately close to physiological values given by *Durrer et al.* [18]. For the Purkinje domain the conductivity was set to $\sigma_{purk} = 2.567S/m$ with a CV approximately equal to $2m/s$. The stimulus protocol is a single pulse coming from the opposite side of the PVJ coupling with the following parameters: $I_{amp} = 40pA/pF$, duration $= 1ms$, $N_{cells} = 25$. There are a total of 1000 Purkinje cells and 15625 ventricular cells which are solved in under half a minute using a high performance GPU monodomain solver [59].

Applying different combinations of R_{PVJ}, and N_{PVJ} we measured the PVJ delay by taking

the difference between the mean LAT of the tissue cells and the LAT of the PC associated to the PVJ coupling. In Table 1 we show the results of this experiment.

(R_{PVJ}, N_{PVJ})	**40**	**45**	**50**	**55**	**60**
300	2.43	2.15	1.7	1.69	1.35
400	5.39	2.97	2.31	1.86	1.75
500	block	block	2.84	2.38	2.17
600	block	block	3.91	2.85	2.51
700	block	block	block	3.6	**3.1**
800	block	block	block	4.61	3.5
900	block	block	block	block	4.21
$\{k\Omega\}$					

Table 1 – Results of the PVJ calibration experiments show how the anterograde PVJ delay (in milliseconds) varies for different values of R_{PVJ} and N_{PVJ}. When the Purkinje fiber could not stimulate the VCs, we define the PVJ delay to ∞ and use the tag "block" in the table.

Based on these results, the Purkinje coupling parameters were set to $R_{PVJ} = 700k\Omega$ and $N_{PVJ} = 60$, generating a PVJ delay of approximately 3 ms in the anterograde direction, which is within the acceptable range [79]. In Figure 26, we show the Purkinje cable stimulating the ventricular tissue cuboid and the resultant AP associated to the PVJ site highlighting the characteristic delay that occur when using $R_{PVJ} = 900k\Omega$ and $N_{PVJ} = 60$ as the configuration for the PVJ coupling.

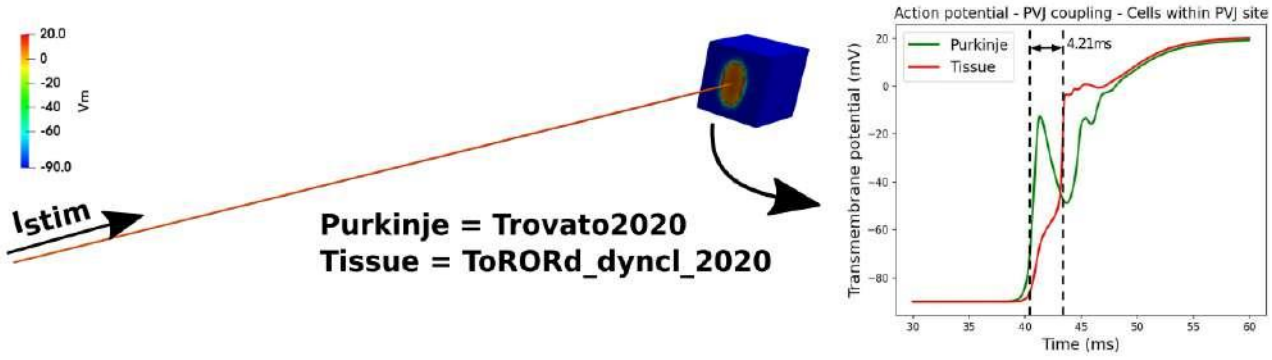


Figure 26 – Experiment to calibrate the Purkinje-Ventricular-Junction delay in the anterograde direction. A single $10cm$ Purkinje fiber stimulates one of the faces of a tissue block ($1cm^3$). The PVJ delay in the anterograde direction is measured by the difference between the LAT of the VCs and PC related to the coupling. The PCs are modeled using the *Trovato2020* human Purkinje model [74] and the VCs using the *ToRORd-dynCl-2020* human ventricular model [73]. The action potential traces from both the Purkinje and one of the ventricular cells of the coupling are depicted alongside the PVJ delay when $R_{PVJ} = 900k\Omega$ and $N_{PVJ} = 60$ in the right panel. A characteristic delay of approximately $4ms$ is observed when using this particular configuration.

3.4 Sensitivity analysis of the input parameters

To evaluate the robustness of our method we performed a sensitivity analysis over the input parameters of the model. Our baseline simulation considered the generation of 10 LV trees using the same seeds over the Simplified mesh with the following parameter set: $N_p = 80$, $N_a = 80$, $L_{rate} = 25$, $l_d = 10mm$, $L_{error} = 2ms$. The choice regarding the baseline parameters were made initially to achieve a reasonable approximation for the LAT at the active PVJ sites.

In addition, the Root Mean Square Error (RMSE) and Relative Root Mean Square Error (RRMSE) between the LAT from the active PVJ sites of the reference points and the generated networks are calculated as follows:

$$RMSE = \left(\frac{\sum_{i=1}^{N} (\bar{y_i} - y_i)^2}{N} \right)^{1/2}, \qquad RRMSE = \left(\frac{\sum_{i=1}^{N} (\bar{y_i} - y_i)^2}{\sum_{i=1}^{N} |\bar{y_i}|} \right)^{1/2}, \tag{3.7}$$

where $\bar{y_i}$ is the reference value, y_i is the approximated one, and N is the number of samples.

For the sensitivity analysis experiment, the RMSE error at the PVJ sites and the total execution time to generate the PNs are utilized to compare the different parameter combinations as illustrated by Table 2.

Parameters	**Values**	**RMSE (ms)**	**Total execution time (min)**
N_p	20	0.95 ± 0.31	1.81 ± 0.30
	400	0.95 ± 0.31	5.80 ± 0.54
N_a	20	1.20 ± 0.42	1.55 ± 0.12
	400	0.86 ± 0.22	6.68 ± 1.75
L_{rate}	10	0.95 ± 0.33	2.85 ± 0.55
	40	0.91 ± 0.32	2.36 ± 0.17
l_d	5mm	1.10 ± 0.66	3.39 ± 0.34
	15mm	1.18 ± 0.43	1.69 ± 0.15
L_{error}	1ms	0.90 ± 0.33	2.54 ± 0.32
	5ms	1.30 ± 1.06	2.56 ± 0.32
Baseline	**-**	**0.95 ± 0.31**	**2.50 ± 0.31**

Table 2 – Sensitivity analysis of the input parameters considering the baseline configuration as: $N_p = 80$, $N_a = 80$, $L_{rate} = 25$, $l_d = 10mm$, $L_{error} = 2ms$. For each configuration, 10 networks were generated using the same seed. The mean and standard deviation of the RMSE (milliseconds) at the PVJ sites and the total execution time (minutes) to generate the networks are depicted.

Based on the results of Table 2 we notice that the N_p parameter does not affect the accuracy of the trees at the expense of increasing the total time. The main reason behind this behaviour is because more segments are being evaluated by the passive cost function CF_p and more geodesic pathways are generated in this process, which increases significantly the total execution time

of the method. In addition, the results demonstrate that we could achieve the same result in less computation time by decreasing N_p. In the passive scenario, the branches are connected by minimizing the total length of the tree following cost function (3.2), and since we already sorted the segments in the tree by their Euclidean distance to x_{term}, the first N_p segments are already good candidates to build a geodesic pathway that will generate reasonable evaluations by the passive cost function.

Analyzing the other parameters, we could verify that the L_{rate} parameter does not affect substantially the solution either by electrical accuracy or computation time. Furthermore, for the L_{error} parameter the precision of the trees can be improved by decreasing its value without increasing the computation time. On the other hand, depending on the value of the l_d parameter we can decrease the accuracy of the PNs.

In terms of electrical accuracy the N_a parameter is the most sensible as can be verified in Table 2. Variations on its value can not only affect drastically the RMSE but also execution time. Differently to what happened with the N_p parameter, we need to evaluate more segments using the active cost function CF_a to improve the trees electrically. In this regard, the N_a parameter should be carefully adjusted in order to generate more precise PNs in a reasonable amount of time.

As a result of this initial analysis, we perform a new set of simulations varying the N_a parameter in order to find the optimal value regarding electrical accuracy and total computation time for this particular scenario. Our baseline configuration for this second analysis considers: $N_p = 20$, $N_a = 80$, $L_{rate} = 25$, $l_d = 10mm$, $L_{error} = 2ms$ and the results are shown in Table 3.

Parameters	Values	RMSE (ms)	Total execution time (min)
N_a	80	0.95 ± 0.31	1.78 ± 0.28
	120	0.88 ± 0.30	2.39 ± 0.41
	160	0.88 ± 0.27	2.89 ± 0.60
	200	0.86 ± 0.22	3.39 ± 0.79
	240	0.86 ± 0.22	3.79 ± 1.00
	280	0.86 ± 0.22	4.33 ± 1.17
	320	0.86 ± 0.22	4.86 ± 1.38
	360	0.86 ± 0.22	5.45 ± 1.55
	400	0.86 ± 0.22	5.99 ± 1.75
Baseline	**-**	**0.95 ± 0.31**	**1.78 ± 0.28**

Table 3 – Sensitivity analysis of the N_a input parameter considering the baseline configuration. For each configuration, 10 networks were generated using the same seed. The mean and standard deviation of the RMSE (milliseconds) at the PVJ sites and the total execution time (minutes) to generate the networks are depicted.

From the results of Table 3 we could verify that when N_a surpasses the value 120 the RMSE converges to a value of approximately $0.88ms$. Within this context, we can conclude that it is not

necessary to keep increasing this parameter to improve the electrical accuracy of the PNs. This reinforces the conclusion that, although not entirely accurate, sorting the feasible segments using an approximation of their LAT error, can in fact help the method to find feasible segments.

4 Results: Matching the Purkinje-Ventricular-Junctions Local Activation Time

The proposed method in this work was evaluated using initially three different biventricular meshes with an increasing level of complexity. In this section, different Purkinje networks are generated for the Simplified, Canine and Patient-specific meshes. For the Patient-specific scenario, the mesh from *Lopez-Perez, A. et al (2019)* [43] was utilized.

A total of 200 PNs were generated for each mesh, counting 100 PNs for the LV and the remaining ones for the RV. The RMSE error was calculated for each PN by comparing the LAT given at the reference active PVJ sites. Using the previous results, the PNs samples were sorted by their RMSE error, and the best 10 PNs for each ventricular region were filtered and considered in our comparison set. The parameters of the performed simulations were fixed to $N_p = 20$, $L_{rate} = 25$, $L_{error} = 2ms$ and $d = 69\mu m$ giving a CV of approximately $2m/s$ when using the cable equation (3.4) with Purkinje parameters. For the Simplified and Canine meshes, $l_d = 10mm$ and N_a was set to 120 for both ventricles. Finally for the Patient-specific mesh $l_d = 30mm$ and N_a was set to 200 for both ventricles. The choice regarding the l_d parameter was made considering that the Patient-specific mesh has a larger volume than the Simplified and Canine meshes, while for the N_a parameter the increase of value is justified to improve the accuracy of the generated PNs.

All the simulations were performed without any restriction regarding the bifurcation angle and segment length. During early tests, there was an increase in computation time and the generation of unfeasible solutions that could not connect all PVJs within the required geometrical constraints of angle and length. The usage of geodesic pathways to maintain the PNs within the endocardium surface is justified due to the high geometrical complexity of the patient-specific mesh from *Lopez-Perez, A. et al (2019)* [43], especially near the regions of the papillary muscles. If we use the strategy of our previous work [76] some branches will not be entirely within the endocardium surface, given the idea of false tendons, which was not our desired objective to the generated PNs by the model. For that reason, to increase the performance of the method due to the geodesic pathway computation, which was identified as the bottleneck of the model, we decreased the number of triangles of the endocardium surfaces, but sustaining its topology.

In addition, the geometrical and electrical features of all the PNs were computed considering only the minimum networks (after the *Post-processing* procedure). Consequently, all the terminals of the resultant PNs are connected to active PVJs.

To properly evaluate the geometrical features of our PN models, the mean and standard deviation of the branch size and angle of the bifurcations are computed together with the total number of branches and bifurcations. More information regarding the geometrical concepts of the method are described in the Appendix C.

To assess the electrical accuracy of our model, the minimum/maximum LAT from the active

PVJ sites are computed alongside the maximum absolute error observed in these points in the generated PN. Secondly, the RMSE and RRMSE values at active PVJ points are computed. Finally, the percentage of the active PVJs within a specific range of error (i.e., 2ms and 5ms) is another measurement considered, given by the values $\epsilon < 2$ms and $\epsilon < 5$ms.

The computational resources utilized to generated all the PNs in this section was a machine equipped with a processor *Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (12 cores)*, a total of 16 gigabytes (GB) of memory RAM and a GPU *NVIDIA GeForce GTX 1660 Ti* with 6 gigabytes (GB) of RAM. The generation process was done in parallel by dividing the 100 PNs in 10 groups of 10 PNs. For each network within these groups, a CPU core was allocated and the procedure was done synchronously one group at time.

4.1 Geometrical metrics

In the following sections the geometrical results from the 10 best PNs regarding the three biventricular meshes are presented. To evaluate the PNs geometrically the branch size (millimeters), bifurcation angle (^{o}) and the total number of branches and angles in the trees are calculated considering the mean and standard deviation of its values.

4.1.1 Simplified mesh

In Table 4 the geometrical results for the Simplified mesh are illustrated. As can be seen in this table, the values in terms of branch size demonstrate that the RV trees have a longer size when compared to the LV ones. When the average and standard deviation values are compared to the reference PN, we can noticed that the values of branch size for the 10 best networks can vary depending on the endocardium region where trees were generated. For the bifurcation angle measurement in this mesh, we could verify that the average values from both biventricular regions were close to the reference values and that the angles for the RV trees have more acute shape. Finally, for the total number of branches and bifurcations in the trees the values were exactly the same as the reference ones, which was already expected since all the generated PNs have the same number of terminals as the reference. This can also be verified by the fact that all the active PVJs are connected to the minimum network.

LV	Branch size (mm)	Angle(º)	#Branches	#Angles
Reference	**2.93 ± 1.71**	**52.02 ± 3.95**	**85**	**42**
PK0	4.16 ± 4.04	42.54 ± 28.67	85	42
PK1	4.71 ± 4.64	46.33 ± 32.45	85	42
PK2	4.00 ± 4.05	50.71 ± 32.72	85	42
PK3	4.49 ± 4.30	53.57 ± 31.39	85	42
PK4	4.15 ± 3.59	49.64 ± 34.94	85	42
PK5	4.38 ± 4.22	50.23 ± 34.02	85	42
PK6	4.80 ± 4.73	52.51 ± 30.30	85	42
PK7	4.47 ± 3.92	41.02 ± 32.46	85	42
PK8	4.57 ± 4.10	49.02 ± 32.55	85	42
PK9	4.27 ± 4.29	59.57 ± 41.34	85	42
RV	Branch size (mm)	Angle(º)	#Branches	#Angles
Reference	**3.41 ± 3.00**	**34.23 ± 0.28**	**39**	**19**
PK0	6.16 ± 6.27	35.32 ± 26.07	39	19
PK1	6.64 ± 7.34	40.41 ± 27.69	39	19
PK2	6.16 ± 6.26	35.29 ± 26.09	39	19
PK3	6.67 ± 6.51	39.04 ± 26.70	39	19
PK4	6.70 ± 7.20	33.86 ± 28.10	39	19
PK5	6.12 ± 5.96	45.90 ± 27.21	39	19
PK6	6.57 ± 7.28	36.47 ± 30.09	39	19
PK7	7.51 ± 8.61	35.54 ± 25.65	39	19
PK8	7.06 ± 7.92	34.47 ± 29.19	39	19
PK9	7.05 ± 7.90	35.82 ± 28.26	39	19

Table 4 – Results for the geometric features from the best 10 Purkinje networks of the Simplified mesh. For this particular mesh, the Reference Purkinje network was generated using the fractal method by *Costabal et al.* [16].

4.1.2 Canine mesh

Analyzing the geometrical results for the Canine mesh in Table 5 we noticed that the average values for the branch size were closer to the reference ones in both biventricular regions when compared to the previous results from the Simplified mesh. In terms of bifurcation angle, the values from the generated PNs presented a high variability indicating that the complexity of the endocardium surface can affect the shape of the angles in the trees. Similarly, to the Simplified mesh results the values for the total number of branches and bifurcations angles match exactly the reference values since the minimum networks generated by the *Shocker* method have the same number of terminals as the gold standard trees.

LV	Branch size (mm)	Angle(º)	#Branches	#Angles
Reference	**3.56 ± 3.09**	**89.48 ± 44.47**	**259**	**129**
PK0	4.07 ± 4.01	54.30 ± 39.14	259	129
PK1	4.24 ± 3.79	50.57 ± 39.29	259	129
PK2	4.11 ± 4.24	61.16 ± 45.47	259	129
PK3	4.19 ± 4.18	50.65 ± 38.88	259	129
PK4	4.22 ± 4.25	56.21 ± 36.68	259	129
PK5	4.31 ± 4.43	53.31 ± 41.84	259	129
PK6	3.97 ± 3.83	58.95 ± 39.94	259	129
PK7	4.26 ± 3.91	56.69 ± 39.25	259	129
PK8	4.18 ± 3.79	57.49 ± 41.04	259	129
PK9	4.28 ± 4.17	56.23 ± 37.51	259	129
RV	Branch size (mm)	Angle(º)	#Branches	#Angles
Reference	**3.43 ± 3.61**	**76.58 ± 39.57**	**195**	**97**
PK0	3.77 ± 4.20	60.00 ± 43.87	195	97
PK1	4.06 ± 4.75	55.61 ± 44.73	195	97
PK2	3.75 ± 4.74	51.21 ± 38.99	195	97
PK3	4.08 ± 4.86	48.54 ± 40.01	195	97
PK4	4.38 ± 6.14	53.63 ± 42.20	195	97
PK5	4.31 ± 4.64	58.97 ± 47.72	195	97
PK6	4.21 ± 5.49	45.15 ± 32.27	195	97
PK7	4.12 ± 4.81	51.21 ± 43.23	195	97
PK8	3.90 ± 4.39	53.86 ± 41.50	195	97
PK9	4.08 ± 4.82	49.08 ± 39.90	195	97

Table 5 – Results for the geometric features from the best 10 Purkinje networks of the Canine mesh. In the case of the Canine mesh, the Reference network was provided from the works [42, 76].

4.1.3 Patient-specific mesh

In Table 6 the results for the patient-specific could not be compared to a reference PN since only the location and LAT of the active PVJs sites were available. However, important topological observations can be made from the generated PNs. In terms of branch size, the LV and RV trees for this particular mesh present the highest values. This can be justified due to the fact that the Patient-specific mesh have the largest endocardium volume when it is compared to the Simplified and Canine meshes leading to prolonged branches in either biventricular regions.

LV	Branch size (mm)	Angle(º)	#Branches	#Angles
PK0	16.10 ± 17.05	54.35 ± 32.09	63	31
PK1	18.55 ± 19.90	45.82 ± 30.01	63	31
PK2	17.20 ± 13.53	51.62 ± 32.52	63	31
PK3	15.90 ± 15.54	59.75 ± 33.06	63	31
PK4	17.08 ± 19.74	58.29 ± 40.93	63	31
PK5	15.20 ± 17.38	59.11 ± 40.69	63	31
PK6	15.65 ± 16.87	53.76 ± 33.22	63	31
PK7	16.04 ± 16.50	60.20 ± 41.07	63	31
PK8	16.15 ± 16.38	49.70 ± 32.23	63	31
PK9	16.96 ± 19.91	59.88 ± 35.55	63	31
RV	Branch size (mm)	Angle(º)	#Branches	#Angles
PK0	18.53 ± 15.86	65.55 ± 48.63	33	16
PK1	17.92 ± 15.23	65.50 ± 47.24	33	16
PK2	19.57 ± 16.93	64.89 ± 43.16	33	16
PK3	18.30 ± 16.52	55.73 ± 45.12	33	16
PK4	19.25 ± 17.46	45.75 ± 33.48	33	16
PK5	17.38 ± 12.80	64.60 ± 48.96	33	16
PK6	19.85 ± 15.40	62.86 ± 42.40	33	16
PK7	19.65 ± 13.65	58.54 ± 41.07	33	16
PK8	17.52 ± 14.42	63.71 ± 43.60	33	16
PK9	19.00 ± 18.00	59.51 ± 45.24	33	16

Table 6 – Results for the geometric features from the best 10 Purkinje networks of the Patient-specific mesh.

4.2 Electrical metrics

In the next sections the electrical results from the 10 best PNs of the three biventricular meshes are depicted. To evaluate the PNs electrically the minimum and maximum LAT (milliseconds), maximum LAT error (milliseconds), RMSE (milliseconds), RRMSE (%), $\epsilon < 2ms$ (%) and $\epsilon < 5ms$ (%) are evaluated at the active PVJ sites in the trees.

4.2.1 Simplified mesh

In Table 7 the electrical results for the Simplified mesh are illustrated. As can be seen in the table, the generated PNs presented a minimum and maximum LAT close to the reference values in both biventricular regions. When the maximum LAT error is analyzed the trees presented a good accuracy with errors below $3ms$. This observation is also sustained by RMSE error results below $1ms$ and almost the majority of the active PVJs being connected within a LAT error tolerance below $2ms$.

LV	**minLAT (ms)**	**maxLAT (ms)**	**maxError (ms)**	**RMSE (ms)**	**RRMSE (%)**	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**11.83**	**18.25**	**-**	**-**	**-**	**-**	**-**
PK0	11.84	19.04	1.01	0.28	1.84	100.00	100.00
PK1	12.14	19.05	0.80	0.33	2.16	100.00	100.00
PK2	11.82	19.45	1.20	0.37	2.43	100.00	100.00
PK3	11.82	19.39	1.13	0.37	2.40	100.00	100.00
PK4	11.85	19.03	1.54	0.40	2.59	100.00	100.00
PK5	11.78	19.30	1.72	0.42	2.76	100.00	100.00
PK6	11.85	19.32	1.87	0.45	2.92	100.00	100.00
PK7	11.82	19.06	2.55	0.46	3.01	97.67	100.00
PK8	11.79	19.07	2.69	0.47	3.07	97.67	100.00
PK9	11.85	19.19	2.47	0.48	3.13	97.67	100.00
RV	**minLAT (ms)**	**maxLAT (ms)**	**maxError (ms)**	**RMSE (ms)**	**RRMSE (%)**	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**14.68**	**17.50**	**-**	**-**	**-**	**-**	**-**
PK0	14.27	17.75	1.08	0.37	2.28	100.00	100.00
PK1	14.19	17.73	1.10	0.37	2.27	100.00	100.00
PK2	14.27	17.75	1.08	0.37	2.28	100.00	100.00
PK3	14.21	17.77	1.10	0.38	2.34	100.00	100.00
PK4	14.23	17.79	1.10	0.38	2.33	100.00	100.00
PK5	14.25	17.75	1.11	0.38	2.36	100.00	100.00
PK6	14.20	17.80	1.11	0.39	2.42	100.00	100.00
PK7	14.28	17.80	1.18	0.41	2.52	100.00	100.00
PK8	14.28	17.81	1.18	0.41	2.54	100.00	100.00
PK9	14.28	17.80	1.18	0.41	2.53	100.00	100.00

Table 7 – Results for the electric features from the best 10 Purkinje networks of the Simplified mesh. For this particular mesh, the Reference Purkinje network was generated using the fractal method by *Costabal et al.*

4.2.2 Canine mesh

In Table 8 the electrical results for the Canine mesh are highlighted. Based on these results, for the minimum and maximum LAT, the generated PNs demonstrate a good accuracy with values close to the reference ones in both biventricular regions. In terms of maximum LAT error, we noticed that the LV trees have the highest values when compared to the RV trees which can be justified due to the fact that are more active PVJs in the LV region of this mesh than the RV. Analyzing the RMSE, RRMSE, $\epsilon < 2ms$ and $\epsilon < 5ms$ metrics the PNs from both regions presented good accuracy with errors equivalent to the results from the Simplified mesh.

LV	minLAT (ms)	maxLAT (ms)	maxError (ms)	RMSE (ms)	RRMSE (%)	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**9.86**	**40.16**	**-**	**-**	**-**	**-**	**-**
PK0	10.10	38.96	4.22	0.64	2.49	97.69	100.00
PK1	9.90	40.35	3.74	0.65	2.54	98.46	100.00
PK2	7.94	40.33	4.31	0.68	2.67	97.69	100.00
PK3	9.93	38.70	4.39	0.74	2.90	96.92	100.00
PK4	9.49	40.09	4.80	0.74	2.90	98.46	100.00
PK5	9.68	40.19	4.37	0.74	2.91	96.92	100.00
PK6	9.44	40.29	4.71	0.74	2.90	97.69	100.00
PK7	9.63	40.38	5.07	0.76	2.99	97.69	99.23
PK8	9.85	40.44	4.98	0.77	3.00	98.46	100.00
PK9	9.72	40.16	4.38	0.80	3.13	96.92	100.00
RV	**minLAT (ms)**	**maxLAT (ms)**	**maxError (ms)**	**RMSE (ms)**	**RRMSE (%)**	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**9.76**	**53.93**	**-**	**-**	**-**	**-**	**-**
PK0	10.44	53.88	1.59	0.51	1.41	100.00	100.00
PK1	10.04	53.92	2.31	0.54	1.48	98.98	100.00
PK2	10.08	53.84	1.94	0.59	1.63	100.00	100.00
PK3	9.79	53.92	2.10	0.60	1.64	98.98	100.00
PK4	9.81	54.11	2.26	0.63	1.71	95.92	100.00
PK5	10.09	53.93	1.93	0.64	1.75	100.00	100.00
PK6	9.79	53.89	2.01	0.64	1.76	97.96	100.00
PK7	9.79	53.92	1.94	0.67	1.84	100.00	100.00
PK8	10.54	53.89	2.27	0.67	1.83	97.96	100.00
PK9	12.06	53.99	2.33	0.68	1.85	96.94	100.00

Table 8 – Results for the electric features from the best 10 Purkinje networks of the Canine mesh. In the case of the Canine mesh, the Reference network was provided from the works [42, 76].

4.2.3 Patient-specific mesh

In Table 9 the electric results for the Patient-specific mesh are presented. Similarly to the previous meshes, the generated PNs performed well in terms of minimum and maximum LAT leading to values close to the reference values. Only in the RV region of the Patient-specific mesh we found that a particular PVJ could not be connected with a LAT error below $5ms$. When the other electrical measurements are analyzed, we noticed that the LV trees presented a better approximation to the reference LAT at the active PVJ sites. In addition, an interesting observation that can be made for the RV trees is that the 10 best networks for this region presented a very similar LAT even with a very distinctly topology.

LV	minLAT (ms)	maxLAT (ms)	maxError (ms)	RMSE (ms)	RRMSE (%)	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**36.13**	**123.77**	**-**	**-**	**-**	**-**	**-**
PK0	35.55	123.75	3.63	1.16	1.50	90.62	100.00
PK1	35.96	123.76	3.70	1.22	1.59	87.50	100.00
PK2	35.96	123.70	4.02	1.32	1.72	90.62	100.00
PK3	35.55	123.70	3.46	1.36	1.76	87.50	100.00
PK4	35.62	125.55	5.47	1.52	1.97	84.38	96.88
PK5	36.40	123.93	3.53	1.54	2.00	78.12	100.00
PK6	35.70	123.73	4.12	1.54	2.00	84.38	100.00
PK7	35.70	123.05	4.21	1.77	2.29	78.12	100.00
PK8	36.40	124.01	4.94	1.79	2.32	81.25	100.00
PK9	35.55	125.49	7.40	1.82	2.36	84.38	96.88
RV	**minLAT (ms)**	**maxLAT (ms)**	**maxError (ms)**	**RMSE (ms)**	**RRMSE (%)**	$\epsilon < 2ms$ (%)	$\epsilon < 5ms$ (%)
Reference	**35.71**	**60.87**	**-**	**-**	**-**	**-**	**-**
PK0	31.05	60.88	6.83	1.79	3.90	94.12	94.12
PK1	31.05	60.78	6.85	1.81	3.95	94.12	94.12
PK2	31.05	61.34	6.84	1.81	3.93	94.12	94.12
PK3	31.05	60.82	6.85	1.81	3.93	94.12	94.12
PK4	31.05	60.77	6.86	1.82	3.96	94.12	94.12
PK5	31.05	60.57	6.83	1.82	3.96	94.12	94.12
PK6	31.05	60.80	6.85	1.82	3.97	94.12	94.12
PK7	31.05	60.92	6.83	1.82	3.97	94.12	94.12
PK8	31.05	60.69	6.83	1.83	3.99	94.12	94.12
PK9	31.05	61.59	6.85	1.83	4.00	94.12	94.12

Table 9 – Results for the electric features from the best 10 Purkinje networks of the Patient-specific mesh. The LAT values of the Patient-specific mesh were estimated by the CARTO ^{TM} maps.

4.3 Comparison and discussion

The results of Tables 4, 5 and 6 are summarized in the boxplots presented in Figure 27. To build the boxplots, we consider all the bifurcation angles and branches from the ten PNs generated for each mesh. Based on the results of this figure, we can verify that for the Simplified mesh the generated PNs present values approximately close to the mean reference values in terms of bifurcation angle and branch size for both ventricles.

Analyzing the structure of best/worst PNs generated by the method for the Simplified mesh in Figure 32A and from the ten best PNs in Figure 33, we can sustain this conclusion. Although, the method was able to produce very distinct PNs with different pathway combinations leading to the active PVJs, the branch size of the two networks highlighted in Figure 32A have almost the same length in both ventricles. Moreover, when the generated PNs are compared to the reference tree presented in Figure 24A and colored in black, the method was able to correctly replicate some of the branch pathway patterns used to activate the PVJs.

For the Canine mesh, the geometrical results shown in Figure 27 presented a good approximation in terms of branch size for both ventricles. For the bifurcation angle our method was able to sustain the reference value for the RV within the first and third quartile of the boxplot, while for the LV networks, the majority of the angles had a value lesser than the baseline. In addition, a high variability is observed in the bifurcation angle feature when compared to the Simplified mesh results, which could indicate that with a more complex endocardium geometry the method needs to generate bifurcations in a more wide range of values. Based on this previous observations, we

can conclude that the method had a better performance for the RV surface of the Canine mesh than the LV one in terms of geometrical features.

One possible reason behind this difference is the existence of more active PVJs to connect in the LV, which could lead to longer segments and bifurcations with more acute angles in this region. An important result that was observed for the Canine mesh, was the ability of the method to correctly connect the active PVJs located at the posterolateral region of the RV. This is because for this particular mesh, the root location of the reference network is located in a region of the septum that requires that the PN network makes a turn around the posterior surface of the RV to properly connect the active PVJs that are on the opposite side of the septum. The geodesic pathway not only correctly connect the active PVJs but also keep the PN within the endocardium surface at this complex and sharp region of the Canine mesh.

Similarly to the Simplified mesh PNs morphology, the structure of the best/worst networks generated for the Canine mesh, presented in Figure 32B, and the overview picture of the ten best PNs for this mesh, depicted in Figure 34, demonstrates that the method was able to connect the active PVJs covering the endocardium surface with different pathway combinations, leading to the hypothesis that there are several PNs that could activate a set of PVJs with a similar LAT.

On the other hand, the results from the Patient-specific mesh from *Lopez-Perez, A. et al (2019)* [43] could not be compared to a reference PN, since only the estimated active PVJs sites were available. Nevertheless, important observations can be retrieved from the geometrical results shown in Figure 27. For instance, there is a high variability in the bifurcation angle and branch size features, which indicates that in order to connect some of the active PVJs, the method tries to construct longer branches at the same time that the bifurcations have a more wide characteristic. This observation sustains our previous observation from the Canine mesh, which demonstrate that the complexity of the mesh is directly related to the variablity of the bifurcation angle. Furthermore, the volume of the Patient-specific mesh is larger than the two previous meshes, and we increased the characteristic length l_d for this particular mesh. For that reason it is acceptable that the branch size feature values are increased for both ventricles in this scenario.

An important observation that can be made from Figures 32, 33, 34 and 35 is that the method produced very different PNs with a similar LAT, illustrating the high uncertainty related to the problem. At the same time, the results for the Patient-specific mesh shows that for LV surface, in particular, some active PVJs could only be connected properly by leading the PN to the apex region first and later growing branches from the apex to the basal area, making the generation of PNs for this mesh the most challenging one for the method.

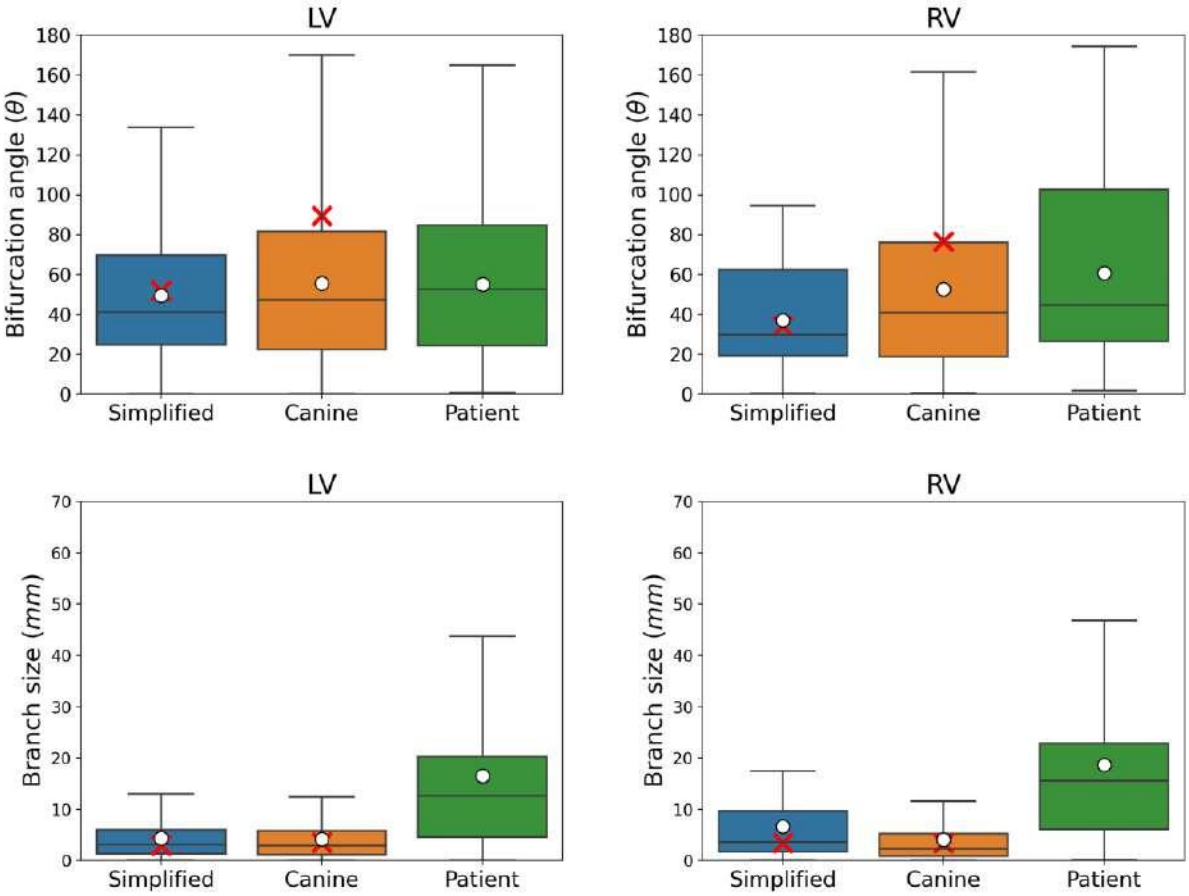


Figure 27 – Geometrical results from both ventricles of the ten best PNs generated for each biventricular mesh. The boxplot for the bifurcation angle is illustrated in the top panel and the branch size is in the bottom panel. The red cross denotes the mean reference values for the gold standard PNs of the Simplified and Canine meshes.

For the electrical results regarding the minimum and maximum LAT developed by the PNs depicted in Figure 28, we can verify that in general the method was able to match the reference values in almost all the PNs. The results in Tables 4 and 5 from the Simplified and Canine mesh, respectively, confirm this conclusion, in which there is not much variability between these features for each PN. Regarding the results of the Patient-specific mesh shown in Figure 28, we notice a difference of approximately $5ms$ in the minimum LAT of the RV. This phenomenon indicates that the PVJ with the lowest LAT was not connected during the main loop of the method; i.e. this point was only linked after pruning the inactive branches and dropping the LAT error tolerance in the *Post-processing* step. On the other hand, this phenomenon was not observed in the maximum LAT feature, where almost all PNs were very close to the baseline values in both ventricles. Within this context, we can conclude by these results that a majority of the PNs generated by the method could correctly connect the PVJs with the minimum and maximum LAT observed for each mesh.

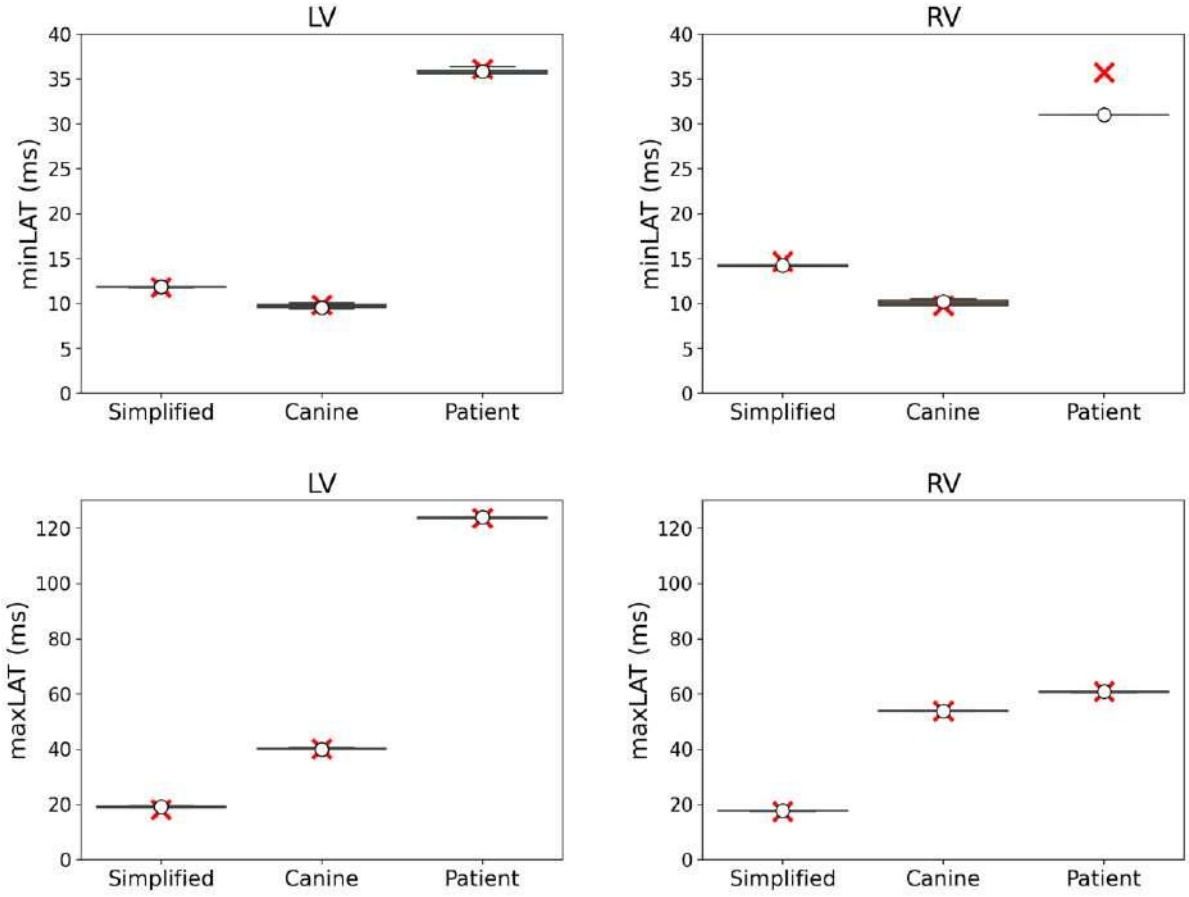


Figure 28 – Electrical results from both ventricles of the ten best PNs generated for each biventricular mesh. The minimum/maximum LAT values were recorded at the active PVJ sites. The boxplot for the minimum LAT is presented in the top panel, while the maximum LAT is on the bottom panel. The red cross denotes the mean reference values for all the meshes.

Examining the RMSE and RRMSE results from Figure 29 we observe an increase in the RMSE error as the mesh complexity elevates, which was already expected. For instance, the Simplified mesh presented an RMSE error below $0.5ms$ for both ventricles, the Canine mesh an error below $1ms$ for both ventricles and for the Patient-specific mesh an error below $2ms$ for both ventricles. In the case of the Patient-specific mesh, the LV networks had a better performance than the RV networks, with an RMSE error of approximately $1.5ms$ in the LV compared to $1.9ms$ in the RV. In the case of the RRMSE measurement all the meshes presented a good performance with values below 4% in both ventricles.

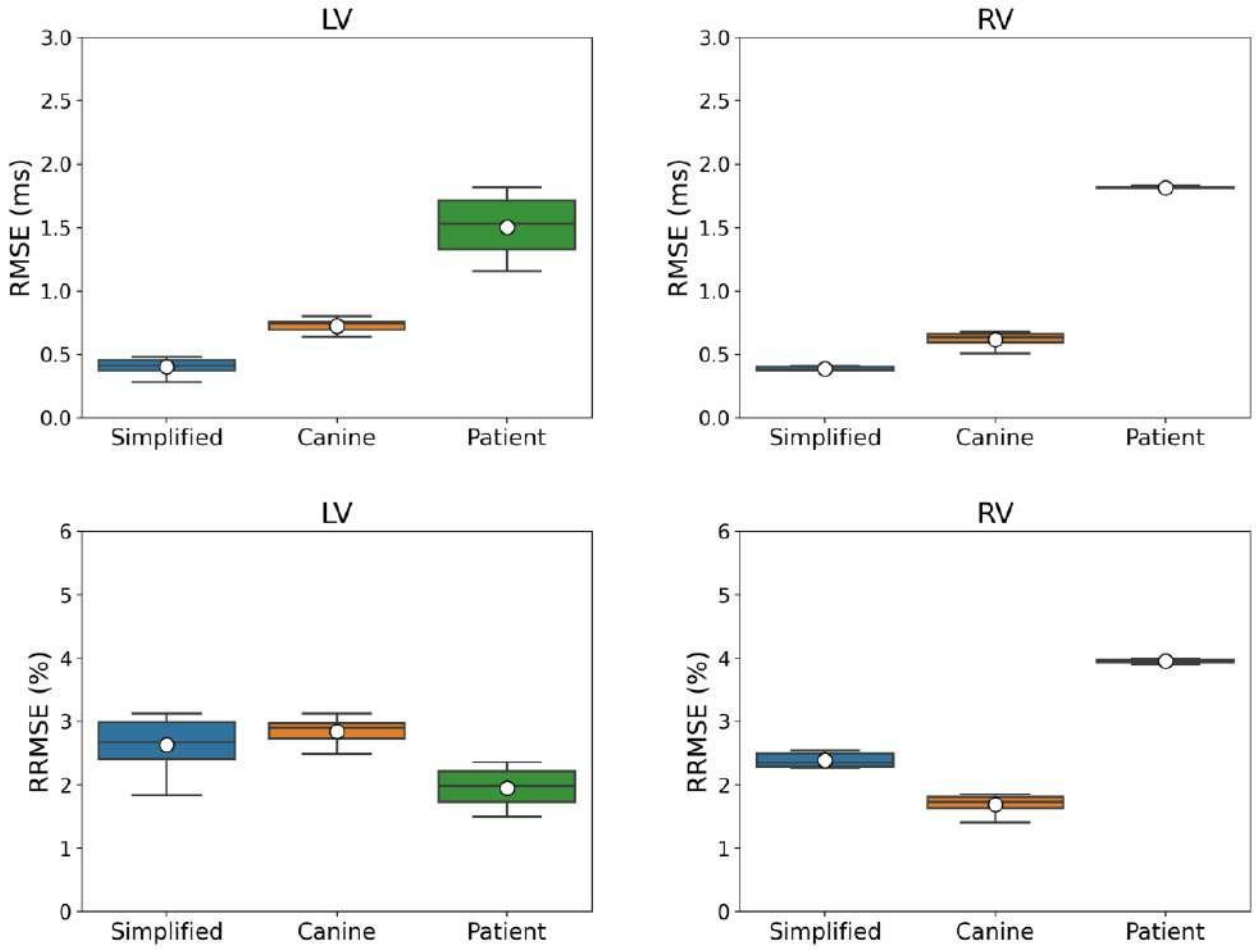


Figure 29 – Electrical results from both ventricles of the ten best PNs generated for each biventricular mesh. The RMSE and RRMSE error values were recorded at the active PVJ sites. The boxplot for the RMSE is presented in the top panel, while the RRMSE is on the bottom panel.

Analyzing the $\epsilon < 2ms$ and $\epsilon < 5ms$ metrics in Figure 30, the amount of PVJs that were connected within the LAT error tolerance of $5ms$ are almost 100% for the LV region in all the meshes. Now, for the $2ms$ scenario, we noticed that this value is above 90% in both ventricles for the Simplified and Canine meshes, while for the Patient-specific mesh this value is approximately equal to 85% for the LV networks and 95% for the RV networks. Based on these results, we can conclude that the method was able to generate PNs with electrical accuracy in all the biventricular domains.

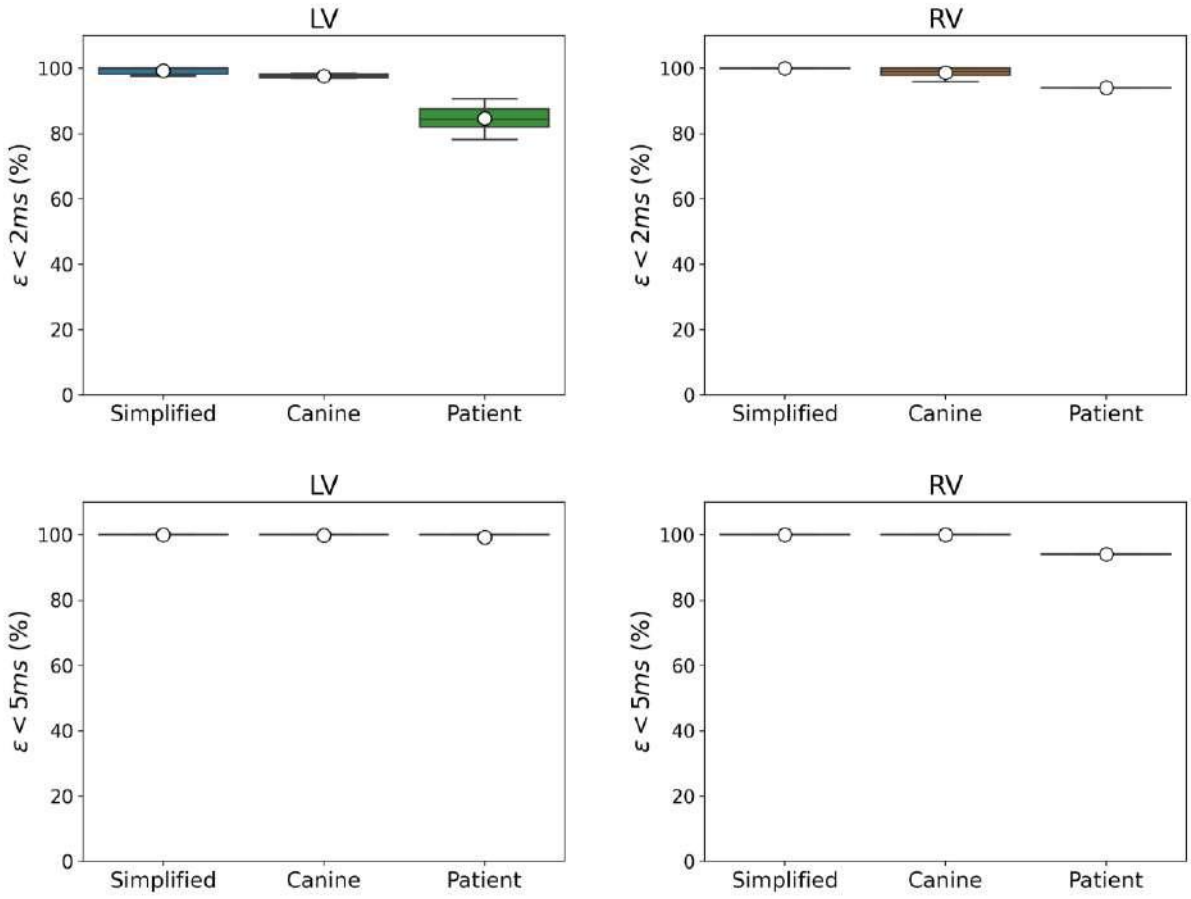


Figure 30 – Electrical results from both ventricles of the ten best PNs generated for each biventricular mesh. The $\epsilon < 2ms$ and $\epsilon < 5ms$ values were recorded at the active PVJ sites. The boxplot for the $\epsilon < 2ms$ is presented in the top panel, while the $\epsilon < 5ms$ is on the bottom panel.

In addition to the geometrical and electrical results, we measured important features related to the performance of the *Shocker* method in each mesh. The execution time spent on each section of the program, the percentage of time required for the geodesic pathway calculus and the percentage of PVJs connected in each step were calculated and presented in Tables 10, 11 and 12. The *Shocker* method was divided in four phases; the main loop and three *pos-processing* procedures, which correspond to the LAT error tolerance and distance criterion removal (geodesic and straight line connections).

Analyzing the results from Table 10 we can verify that the main loop is responsible for the majority of the execution time with a percentage above 90% in the meshes in almost all scenarios. In particular for the PNs generated in the LV of the Simplified mesh, the percentage is around 79%. This behaviour can be explained due to fact that a certain number of PVJs could not be connected due to branch collisions to the inactive segments that are still present in tree before the pruning that occur at the beginning of the *pos-processing* step. A further reason can be related

to PVJs being already too close to the tree, which means that only by removing the distance criterion constrain we could connect these points. These two observations can be confirmed both in Table 10, by the increase in computation time in the second and the sum between the third and fourth phases of the method with a percentage close to 7% and 13%, respectively, and in Table 12 where we can verify that almost half of the PVJs were connected during these three phases of the *pos-processing* step.

Another observation that can be made from Table 10 is that computation time increases as the mesh complexity elevates. The mean execution time to generate a PN for the Simplified mesh was around 2 minutes in both ventricles, for the Canine mesh this time was approximately 3 minutes for the LV and 5 minutes for the RV, while for the Patient-specific mesh is close to 19 minutes for the LV and 7 minutes for the RV. This increase in computation time in the LV trees of the Patient-specific mesh can be explained by two main factors. Firstly, the Patient-specific mesh has several physiological features that do not appear in the other two meshes, like the papillary muscles and a very irregular endocardium surface with tendons which increased the total number of triangular elements necessary to represent this particularities. The complexity and computation time of calling the geodesic pathway procedure increases proportionally to the number of elements of the covered surface. This first observation can be verified in Table 11, where the geodesic pathway calculus consumes more than half of the total execution time in all the meshes and is clearly the bottleneck of our method. Secondly, to increase the accuracy of the generated PNs for the Patient-specific mesh we increase the N_a parameter, which as presented in the Sensitivity analysis section, is responsible for the major increase in computation time by the method.

Mesh	**Region**	**Phase I (min / %)**	**Phase II (min / %)**	**Phase III+IV (min / %)**	**Total (min / %)**
Simplified	LV	1.65 / 79.33	0.16 / 7.57	0.27 / 13.06	2.08 / 100
	RV	2.25 / 95.79	0.00 / 0.01	0.10 / 4.16	2.35 / 100
Canine	LV	2.87 / 94.47	0.03 / 0.95	0.14 / 4.56	3.04 / 100
	RV	4.23 / 90.94	0.10 / 2.16	0.32 / 6.88	4.65 / 100
Patient-specific	LV	17.77 / 92.58	0.35 / 1.81	1.08 / 5.61	19.20 / 100
	RV	6.42 / 92.87	0.17 / 2.42	0.33 / 4.70	6.92 / 100

Table 10 – Summary of the mean execution time on each phase of the *Shocker* method for all the biventricular meshes considering the time from the 10 best PNs. Phase I is the main loop, Phase II is the first *Post-processing* subroutine which drops the LAT error tolerance and Phase III+IV are the subsequent procedures that eliminate the distance criterion to connect the remaining PVJs.

Region	Simplified (%)	Canine (%)	Patient (%)
LV	81.88	68.78	75.20
RV	76.62	77.38	72.49

Table 11 – Summary of the percentage of the total execution time spent on the geodesic pathway subroutine for all the biventricular meshes considering the mean execution time from the 10 best PNs.

Mesh	Region	Phase I (%)	Phase II (%)	Phase III (%)	Phase IV (%)
Simplified	LV	54.19	16.28	23.49	6.05
	RV	81.00	0.00	5.50	13.50
Canine	LV	88.00	1.31	4.23	6.46
	RV	67.14	6.53	10.92	15.41
Patient-specific	LV	66.13	7.74	17.42	8.71
	RV	75.62	7.50	16.88	0.00

Table 12 – Summary of the percentage of active PVJs connected on each phase of the *Shocker* method for all the biventricular meshes considering its mean value from the 10 best PNs. Phase I is the main loop, Phase II is the first *pos-processing* subroutine which drops the LAT error tolerance, Phase III is the subsequent procedure that eliminates the distance criterion to connect the remaining PVJs but which utilizes a geodesic path to link the points and Phase IV is when the connection of the remaining PVJs after droping the distance criterion occurs using a straight line.

In Table 12 we analyze the percentage of PVJs connected during all the phases of the method. When the Phase III and IV are examined we can verified how many PVJs were either connected by a geodesic pathway or straight lines after the distance criterion removal. From these results, when considering the Simplified mesh, from the 29.53% PVJs of the LV trees connected during Phase III and IV, 23.49% are connected using a geodesic pathways, while 6.05% of the PVJs are linked using a straight lines in Phase IV. For the RV trees, from the total of 19% PVJs connected during the third and fourth phases; 5.50% of the PVJs are connected via geodesic and 13.50% using straight lines in the fourth phase. For the Canine mesh, a total 10.69% of the PVJs located in the LV region are connected after eliminating the distance criterion and from this percentage 4.23% are linked using geodesic and 6.46% with a straight line. In the RV region the same percentages from the total of 26.33% PVJs connected after distance criterion is dropped, 10.92% are linked using geodesic and 15.41% with straight lines. Finally, for the Patient-specific mesh, from the total of 26.13% PVJs connected during Phase III+IV in LV trees; 17.42% are connected by geodesic pathways and 8.71% using straight lines. For the RV trees all the 16.88% of PVJs connected in the third phase of the method are linked by geodesic pathways. Within this context, we can conclude that only a small fraction of the PVJs are forced to connect using straight lines during the last step of the method.

Regarding total execution time to generate the PNs, the method demonstrated to be very dependant of the mesh complexity, as can be seen in Figure 31. For instance, in the Simplified and Canine mesh which had the simplest endocardium surfaces in this study, we could generate a PN in almost 4 minutes, with the RV trees of the Canine mesh being the most demanding ones in terms of computation time. Considering the Patient-specific mesh, the values are increased for the LV networks, with values close to 18 minutes, while for the RV networks is around 7 minutes. As mentioned before, the differences between execution time over the biventricular surfaces are closely related to the mesh complexity and to the number of times the program calls the geodesic pathway function, which is by far the most demanding procedure of the method.

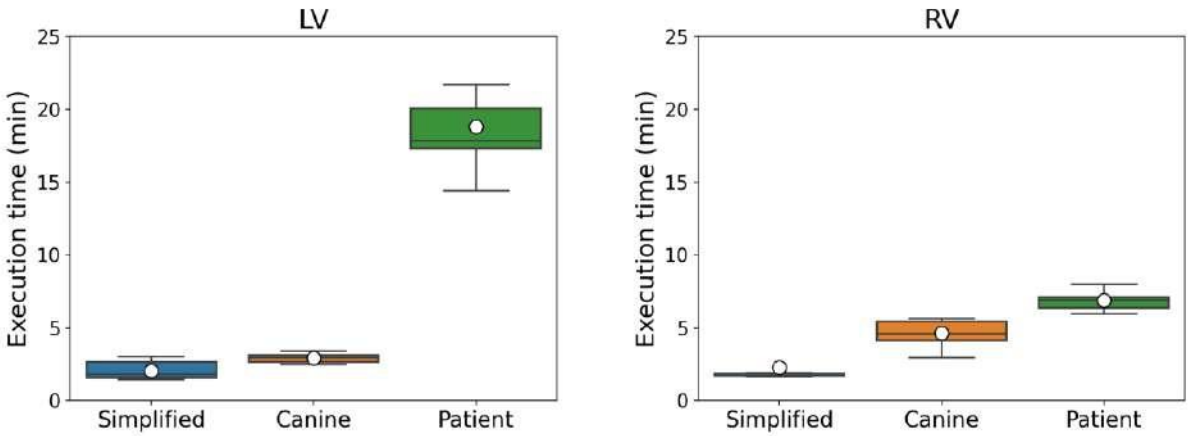


Figure 31 – Total execution time results from both ventricles of the ten best PNs generated for each biventricular mesh.

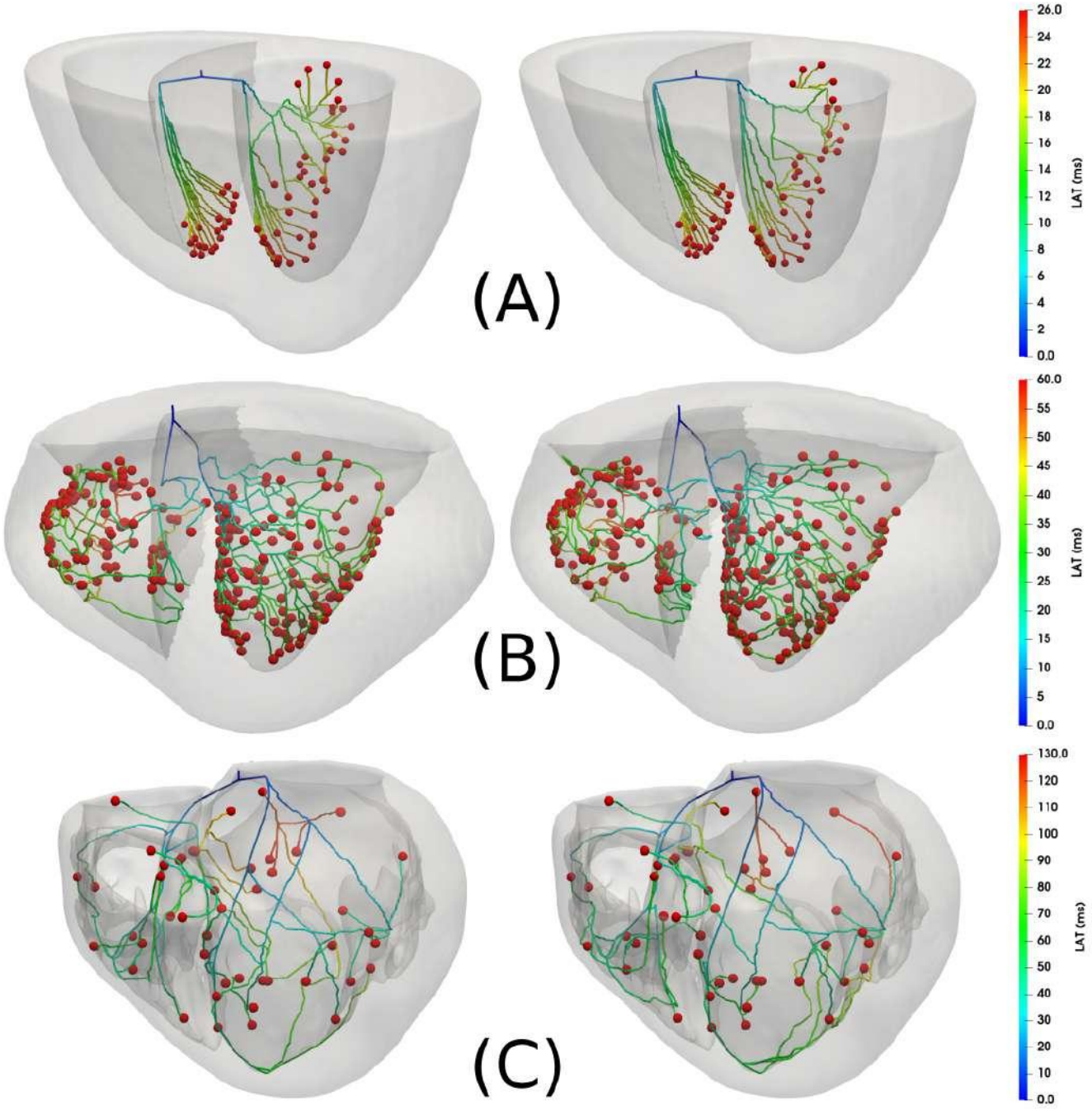


Figure 32 – Comparison of the PNs morphology generated by the method for each biventricular mesh, where the Simplified, Canine and Patient-specific meshes are represented in panels (A), (B) and (C), respectively. The figure depicts the difference between the best (left) and worst (right) networks regarding their LAT at the PVJ sites for ten PNs of the comparison set.

In general, the method was able to generate PNs with reasonable accuracy in terms of LAT, in the case that only LAT from the PNs terminals located at the PVJs sites is considered. In

addition the method was able to generate different PNs morphology with a similar LAT at the active PVJ sites, as can be seen in Figure 32. For that reason, even for the same set of PVJs it is possible to have multiple PNs that share a LAT close to the given reference values. When the best and worst PNs are compared in Figure 32, we could notice that the same PVJ site can be activated within the LAT error tolerance through different pathway combinations. This opens the possibility to study if a particular set of pathways would be more feasible for a specific mesh geometry leading to a better solution. Furthermore, based on this observation raises the idea that an uncertainty quantification study over PNs generation might provide insights on improving such models.

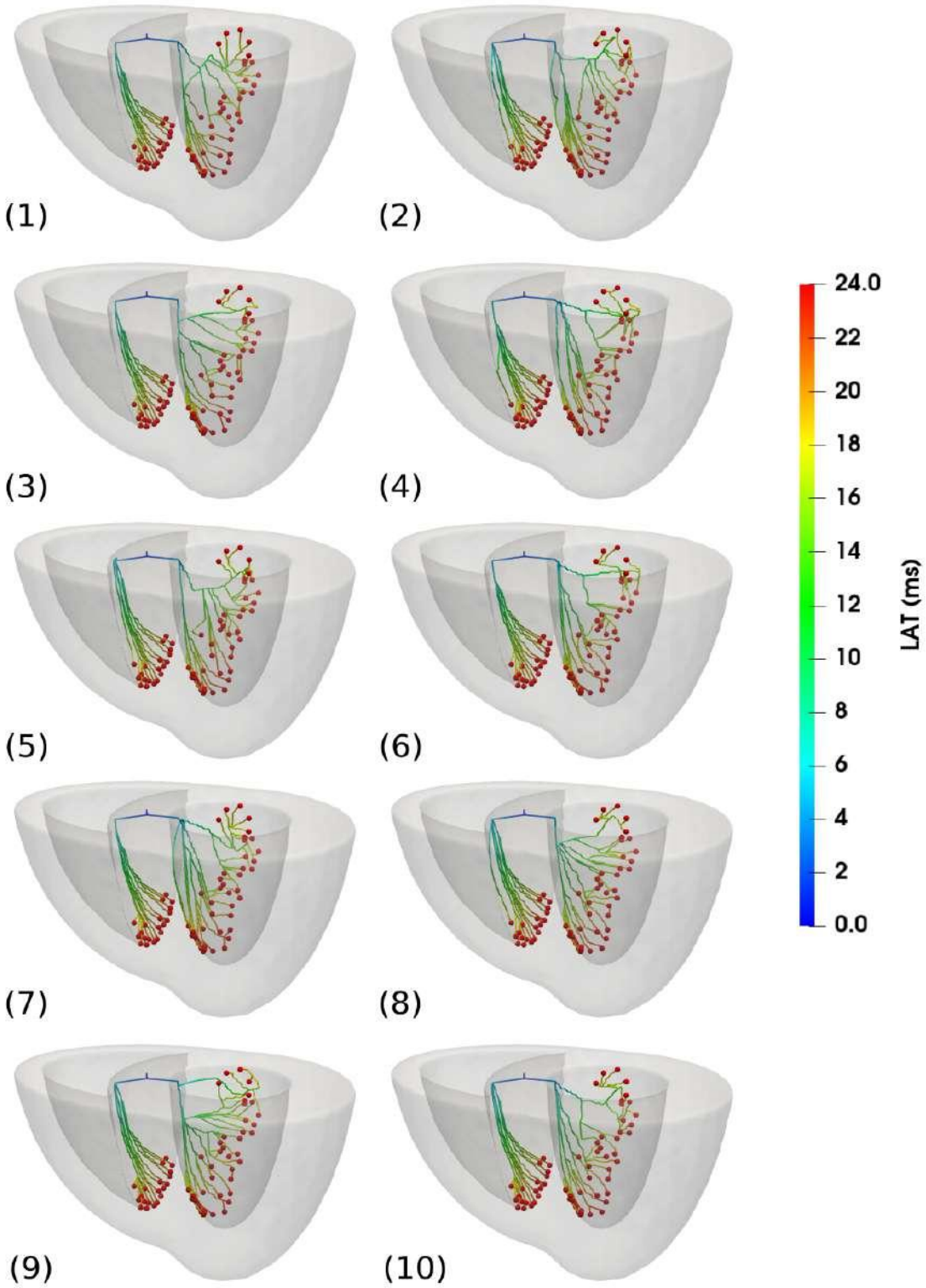


Figure 33 – Difference between the morphology of the ten best networks generated by the *Shocker* method for the Simplified mesh.

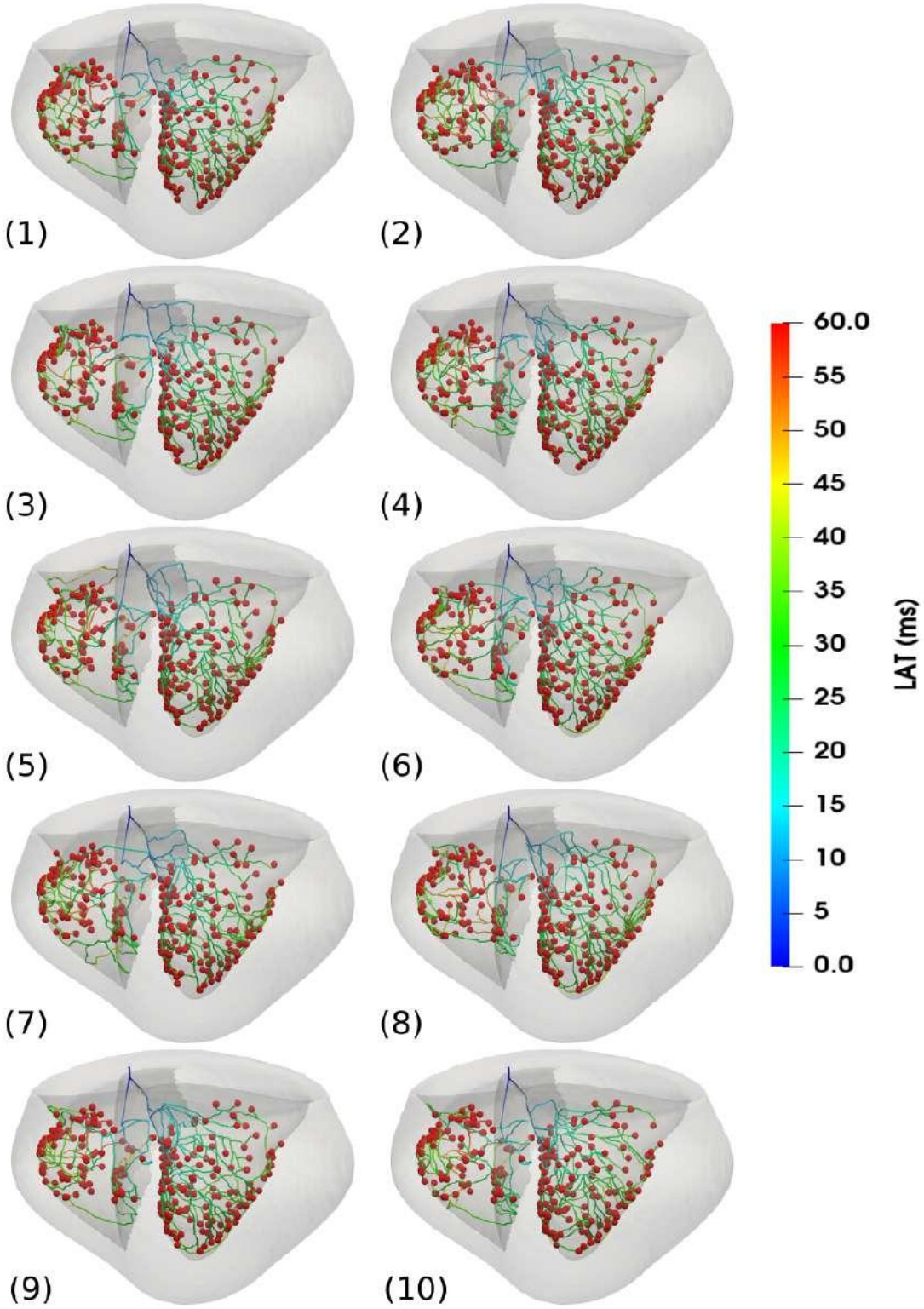


Figure 34 – Difference between the morphology of the ten best networks generated by the *Shocker* method for the Canine mesh.

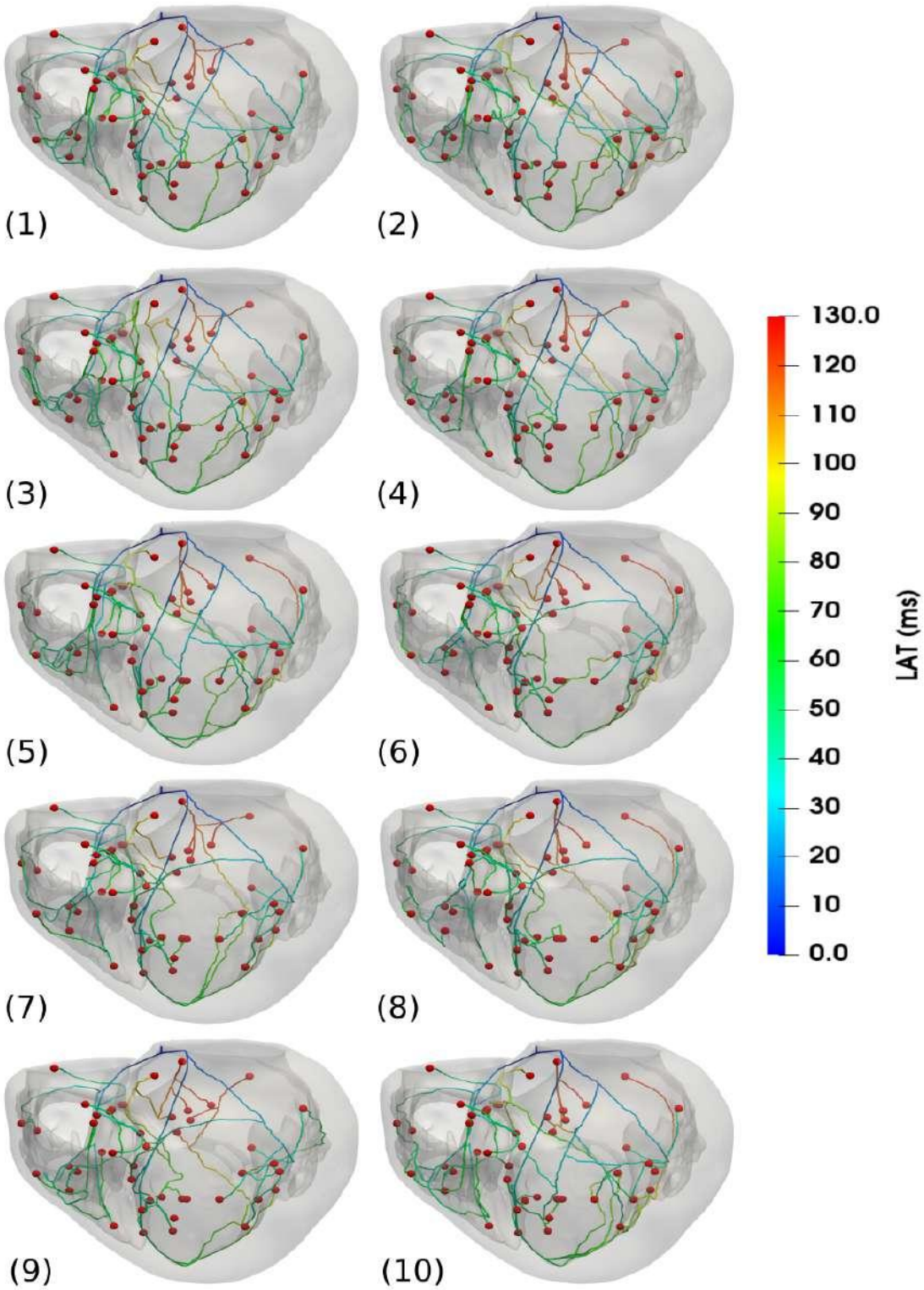


Figure 35 – Difference between the morphology of the ten best networks generated by the *Shocker* method for the Patient-specific mesh.

5 Results: Analysis of Purkinje Network Activation with Monodomain Coupled Simulations

To assess the PNs performance generated by the proposed method in 3D cardiac simulations, we simulate the activation of the 10 best PNs presented in Figures 33, 34 and 35 using a high-performance GPU monodomain solver [59]. The computational resource used for these simulations was a machine equipped with a processor *Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz (4 cores)*, a total of 32 gigabytes (GB) of memory RAM and a GPU *Nvidia TITAN Xp* with 12 gigabytes (GB) of RAM.

Regarding the parameters to enable the solution of the monodomain model, the surface-to-volume ratio is set to $\beta = 0.14\mu m^{-1}$, tissue capacitance is equal to $C_m = 100pF/\mu^2$. A total simulation time of $t_{max} = 200ms$ and a time discretization of $dt = 0.02ms$ was used to solve the associated PDE. For the ODEs system related to the Purkinje and ventricular cellular models, a RL scheme with a fixed timestep of $dt = 0.01$ was utilized. A space discretization of $400\mu m$ was used for all the ventricular domains, while for the Purkinje this value was set to $100\mu m$. The conductivities from the ventricular domain are anisotropic with $\sigma_l = 0.75S/m$, $\sigma_t = 0.225S/m$ and $\sigma_n = 0.1125S/m$, in the longitudinal, transverse and normal direction, respectively, resulting on CV approximately close to physiological values given by *Durrer et al.* [18]. For the Purkinje domain the conductivity was set to $\sigma_{purk} = 2.567S/m$ with a CV approximately equals to $2m/s$. The stimulus protocol is a single pulse coming from the His-bundle with the following parameters: $I_{amp} = 40pA/pF$, duration $= 2ms$, $N_{cells} = 25$.

The Purkinje coupling parameters were set to $R_{PVJ} = 700k\Omega$ and $N_{PVJ} = 60$, generating a PVJ delay of approximately $3ms$ in the anterograde direction, which is within the acceptable range [79]. When there are overlap between two or more PVJ sites, we consider N_{PVJ} equal to the minimum number of tissue cells that are not already coupled to a Purkinje cell. The values for the PVJ delay were calibrated using the experiment and methods described in section 3.3.

In addition, regarding the cellular dynamics, the usage of the recent *ToRORd-dynCl-2020* model for human VCs [72, 73] and the *Trovato2020* model for PCs [74] are justified mainly because these two cellular models are now considered the latest models in terms of human cellular model available in the literature, presenting important advances that surpass old models like the classical *TT3* model [75] and the *ORd* model [50] in numerous aspects. Similarly, the *Trovato2020* Purkinje model provides several features that the old *STW* model [68] was not able to capture. Moreover, there was no published work that couples both models in a Purkinje-biventricular simulation.

To properly evaluate the activation time generated by the PNs, the monodomain equation was solved both for the Purkinje and ventricular domain, instead of the coupled Eikonal model seen in the previous work from *Ulysses, J. N. et al.* [76]. This was performed to capture essential features of propagation from PCs and VCs accurately and at the PVJs sites, such as the characteristic delay

that happens at these locations [79], giving rise to a more realistic simulation in terms of cardiac electrophysiology.

5.1 Simplified mesh

In Figure 36 the results of the monodomain simulations for the Simplified mesh are depicted. Firstly, in Figure 36A, it is illustrated the baseline LAT calculated when the reference PN is coupled to the biventricular tissue. Secondly, in Figures 36B and 36C, it is shown the LAT obtained when the best and worst PN generated by the *Shocker* method are coupled to the biventricular tissue, respectively. Finally, in Figure 36D and 36E, it is presented the absolute error between the baseline and approximate LAT given by the PNs, where the results for best PN is on panel D and the worst PN on panel E.

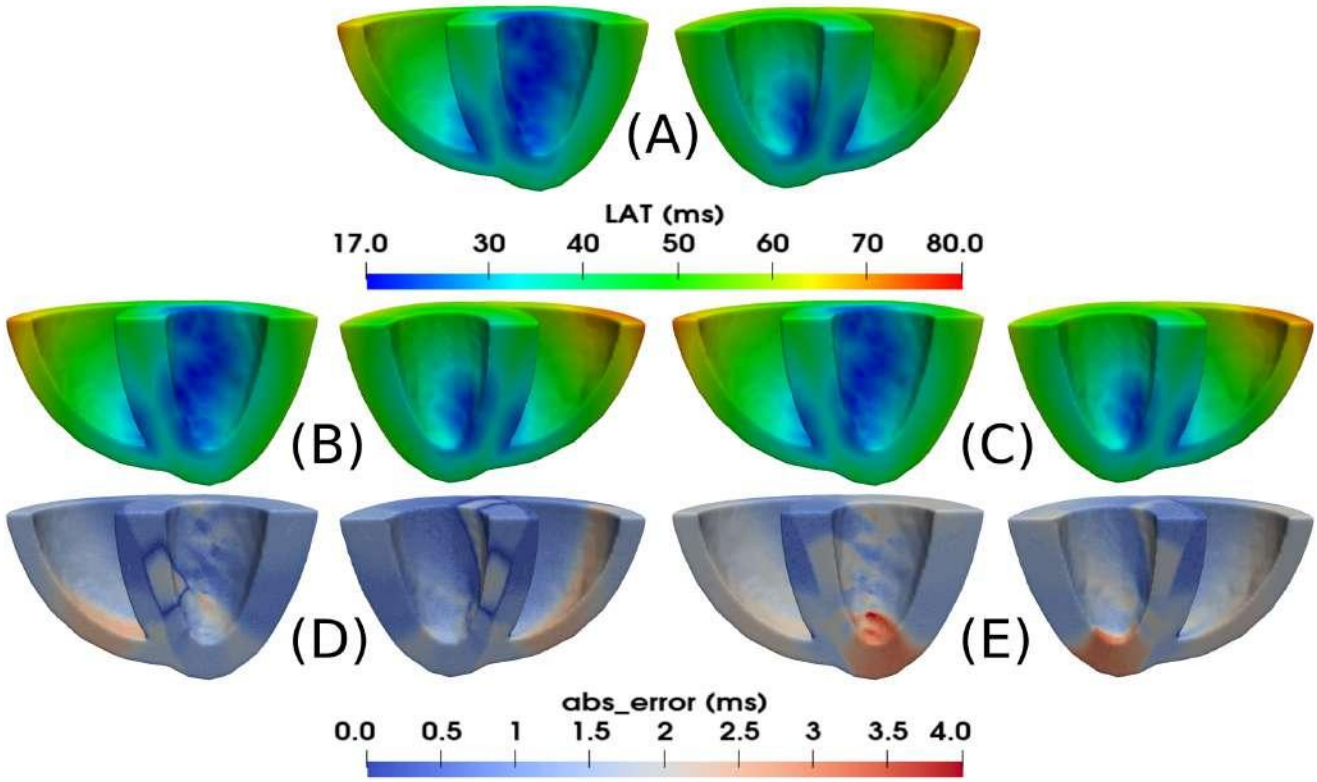


Figure 36 – Results for the LAT maps when a coupled monodomain simulation is executed over the Simplified mesh. First, in the top panel (A), the active PVJs are activated following the LAT provided by the reference PN. Next, in the middle panels (B) and (C), the LAT maps of the best and worst PNs generated by our method are depicted, respectively. Finally, in the bottom panels (D) and (E), the absolute error between the LAT maps from the best and worst PNs is calculated, respectively.

5.2 Canine mesh

In Figure 37 the results of the monodomain simulations for the Canine mesh are illustrated. Initially, in Figure 37A, it is depicted the baseline LAT calculated when the reference PN is coupled to the biventricular tissue. Next, in Figures 37B and 37C, it is shown the LAT obtained when the best and worst PNs generated by our method are coupled to the biventricular tissue, respectively. Lastly, in Figures 37D and 37E, it is presented the absolute error between the baseline and approximate LAT given by the PNs, where the results for best PN is on the left and the worst PN on the right side of the panel.

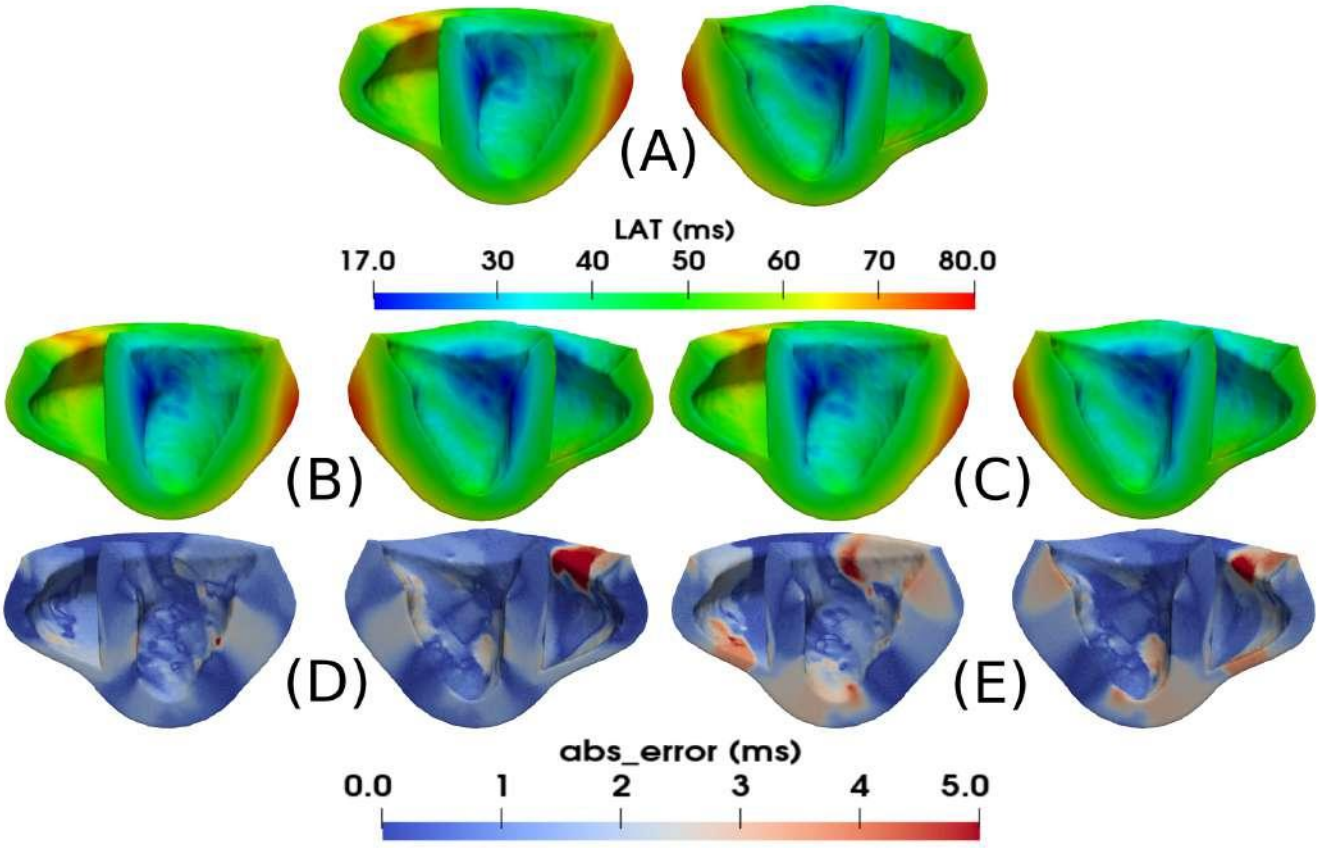


Figure 37 – Results for the LAT maps when a coupled monodomain simulation is executed over the Canine mesh. First, in the top panel (A), the active PVJs are activated following the LAT provided by the reference PN. Next, in the middle panels (B) and (C), the LAT maps of the best and worst PNs generated by our method are depicted, respectively. Finally, in the bottom panels (D) and (E), the absolute error between the LAT maps from the best and worst PNs is calculated, respectively.

5.3 Patient-specific mesh

In Figure 38 the results of the monodomain simulations for the Patient-specific mesh from *Lopez-Perez, A. et al (2019)* [43] are shown. First of all, in Figure 38A, it is illustrated the baseline LAT calculated when the active PVJ sites estimated using CARTO ™ maps are activated at the specified times. Subsequently, in Figures 38B and 38C, it is shown the LAT obtained when the best and worst PNs generated by our method are coupled to the biventricular tissue, respectively. Finally, in Figures 38D and 38E, it is presented the absolute error between the baseline and approximate LAT given by the PNs, where the results for best PN is on panel D and the worst PN on panel E.

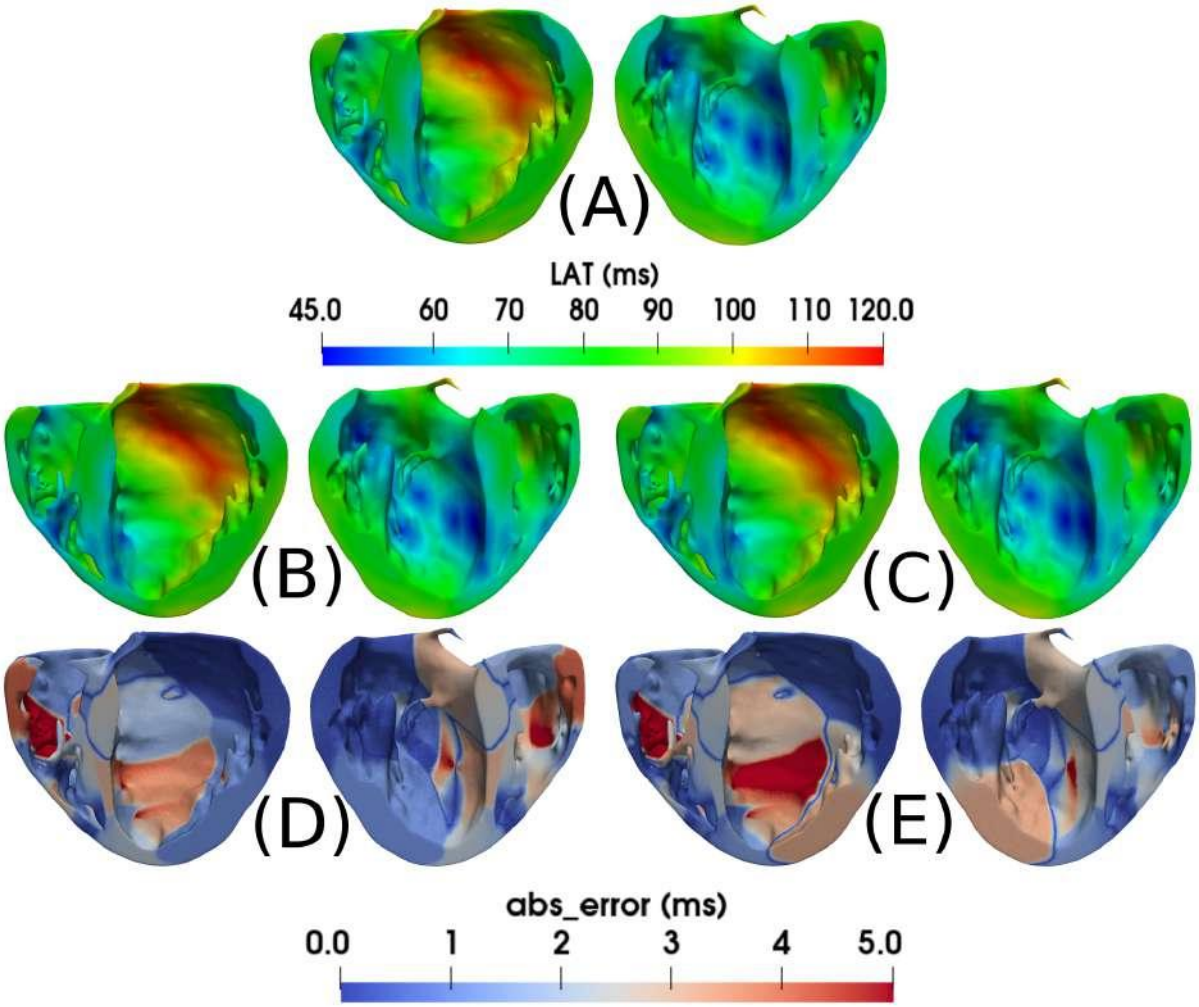


Figure 38 – Results for the LAT maps when a coupled monodomain simulation is executed over the Patient mesh. First, in the top panel (A), the active PVJs are activated following the LAT provided by the CARTO3 ™ points. Next, in the middle panels (B) and (C), the LAT maps of the best and worst PNs generated by our method are depicted, respectively. Finally, in the bottom panels (D) and (E), the absolute error between the LAT maps from the best and worst PNs is calculated, respectively.

5.4 Comparison and discussion

From the results of Figure 36, we noticed that both the best and worst PNs of our comparison set show a feasible LAT map for the Simplified mesh. As can be seen in panel (C) of Figure 36, the best and worst PNs present a maximum absolute LAT error around $3ms$, which is acceptable in terms of LAT matching. To conclude, in Figure 39, the RMSE and RRMSE values were around $1.4ms$ and 3.2%, respectively, considering the ten best PNs generated for the Simplified mesh which indicates that the method is able to generate PNs with electrical accuracy.

Regarding the Canine mesh results presented in Figure 39, we notice an improvement in the RMSE and RRMSE errors when compared to the Simplified mesh. Based on Figure 39, the RMSE and RRMSE values were around $1.3ms$ and 2.5%, respectively, considering the ten best PNs generated for the Canine mesh. From the LAT and absolute error maps in Figure 37, we can conclude that the best PN was able to activate the ventricular tissue similar to the reference LAT in almost all the biventricular tissue, except in a specific region of the RV. On the other hand, the worst PN could not activate several PVJs at the correct LAT even with a good approximation when the cable equation was used. This behaviour could be related to the PVJ coupling parameters used in the monodomain simulation. Mainly, from these results, we can conclude that certain PVJs must activated with a LAT very close to each other to stimulate these regions. In addition, as can be seen in these results, the PVJ delay plays an important role in ventricular activation and depending on certain conditions might be responsible for different LAT.

For the case of the Patient-Specific mesh depicted in Figure 38, we noticed how complex and variable ventricular activation can be depending on the PN that is coupled to the tissue. From Figure 39, the RMSE and RRMSE value were around $2.5ms$ and 3.1%, respectively, considering the ten best PNs generated for the mesh. Furthermore, the results from the Patient-Specific simulations show that in specific PVJ sites, the PN deviates from the reference value, especially near the regions around the papillary muscles of the RV. This behavior indicates that even with RMSE error below $3ms$, there are still regions with a lower LAT accuracy due to branches that were forced to connect during the final steps of the method.

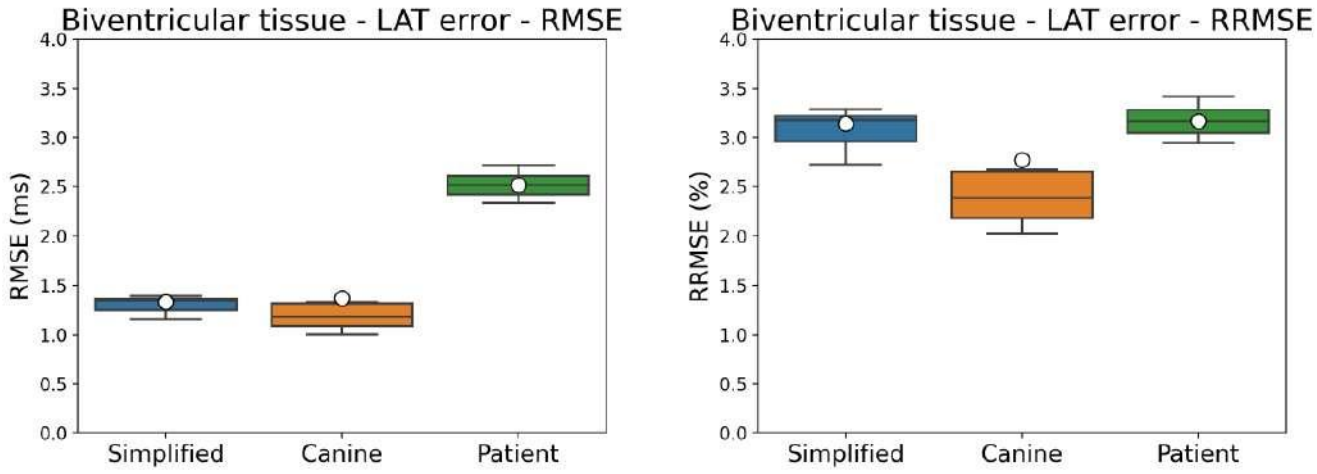


Figure 39 – Results for the RMSE and RRMSE errors from the LAT maps of the biventricular tissues. The RMSE and RRMSE errors were calculated for all the tissue cells by taking the difference between the LAT given by the reference times at the PVJ sites and the LAT generated by the 10 best PNs generated for each mesh.

An important observation that can be made from the previous experiments is that several factors affect ventricular activation when a PN is coupled to the tissue.

First of all, the relation between the cable equation and the monodomain model presented in Appendix B was validated by the results, leading to the conclusion that using the cable equation to approximate the LAT of a given PN is fast and generate good results when coupled to the ventricular tissue in a monodomain simulation. Secondly, in some simulations of the Canine and Patient-specific meshes we noticed that some PVJs could not be proper activated at the right time even by matching the LAT with the cable equation. One of the main reasons for this behaviour is related to the source-sink mismatch problem that happens at the PVJ locations, where certain PVJ sites could only be activated in the monodomain simulation by a combination of terminal branches. This means that for an accurate activation certain groups of active PVJs must be activated close to their reference time, otherwise delays or blocks in tissue stimulation will appear. In addition, another possible reason for this issue can be retropropagation coming from the tissue to the Purkinje.

Another observation that can be made from these results is that certain parameters related to the monodomain simulation, like tissue conductivity and fiber orientation of the mesh, can activate regions of the tissue first than the Purkinje terminal. As a result of, reentry points can appear at these PVJ sites, leading to different LAT values than the ones previous considered in the generation procedure of the PN.

Within this context we noticed that in regions with more active PVJs the characteristic

delay of these points decreased, since more terminal branches are depolarizing a particular area of the tissue, diminishing the source-sink mismatch at these sites. Moreover, the different morphology of the PNs plays an important role in activation. This is because depending on the region in the ventricle tissue that Purkinje terminal is coupled it is not possible to achieve the exactly amount of N_{PVJ} cells. This was observed especially near the irregular regions of the Patient-specific mesh where some active PVJs were very close to each other, which lead to overlaps between the coupled tissue cells associated to these PVJs sites.

Based on this previous observation, we could conclude that a more robust experiment for the PVJ coupling presented in section 3.3 can be done to attenuate these differences in the characteristic delay. The main limitations of the PVJ coupling experiment is that we first consider a simple cuboid without any type of fiber orientation, which based on the previous results can affect the delay. Secondly, as the number of coupled cells N_{PVJ} vary because of overlapped PVJ sites, the resistance R_{PVJ} parameter should be chosen accordingly to the number of N_{PVJ} connected to every particular Purkinje terminal, instead of considering it fixed for all the coupled cells. This way the characteristic delay values observed at certain PVJ sites can be controlled properly and decrease also the risk of conduction blocks.

In addition, a further improvement that can be made to decrease the difference in LAT is to adjust the diameter of the Purkinje fiber accordingly to the cable equation as the tree grows. Most of the necessary tools to implement this new feature are already available in the method. The relation between the cable equation and monodomain equation can be used to adjust the conductivity of the Purkinje branches and the resultant LAT at the PVJ sites could be more precisely calibrated. However, to enable the correct measurements for monodomain simulations it still be necessary to improve the PVJ coupling experiment by considering different conductivity values in the Purkinje cable, since this parameter affect the characteristic delay at the PVJ sites.

Regarding the execution time necessary to solve the monodomain model for all the meshes using the high-performance GPU solver [59] the results of each section are shown in Table 13, ODE_{pk} and ODE_{tiss} are the times required to solve the non-linear ODE systems for the Purkinje and tissue cells, respectively, CG_{pk} and CG_{tiss} are times related to the Conjugate gradient (CG) resolution of the associated linear system of the diffusion part for the Purkinje and tissue cells, respectively. Also, the Write section of the table is the time to compute and write the LAT of the Purkinje network and the biventricular tissue. Finally, the Total part is the global execution time of the entire simulation. In addition, the times presented in Table 13 are averaged, since they consider the resolution of the monodomain model for each of the 10 best generated PNs by our method.

Execution times (min / %)	Simplified	Canine	Patient
ODE_{pk}	0.68 / 1.90	1.96 / 2.84	1.83 / 1.88
CG_{pk}	0.29 / 0.81	0.42 / 0.61	0.38 / 0.39
ODE_{tiss}	14.53 / 40.66	27.87 / 40.28	41.17 / 42.31
CG_{tiss}	6.82 / 19.08	13.07 / 18.89	19.28 / 19.81
Write	10.71 / 29.97	20.67 / 29.88	27.03 / 27.77
Total	35.72 / 100.00	69.18 / 100.00	97.32 / 100.00

Table 13 – Averaged times from the resolution of the monodomain model for the 10 best PNs generated by our method considering the three biventricular meshes.

As can be verified in Table 13 the most demanding section of the monodomain model resolution in all the meshes is associated to the non-linear ODE system related to the human VCs [72, 73] model, which takes approximately 40% of the total execution time. This result can be justified due to the fact that the discretization of the ventricular domain has much more degrees of freedom than the Purkinje network. More specifically, for the Simplified, Canine and Patient-specific meshes utilized there are a total of 1128203, 2155121 and 3156659 ventricular cells, respectively, to be solved when a space discretization of $400\mu m$ is utilized. For the Purkinje networks, the number of cells does not surpass 25000 considering all the generated trees.

5.5 Eletroanatomical map validation

As a final validation experiment, a subset of the original electroanatomical map points used to estimate the active PVJs in the patient-specific mesh is selected to compare the errors of the best and worst generated networks. Only EAM points within healthy regions of the endocardial surface in the LV and RV were considered. The points inside or near the damaged tissue that could be affected by the fibrosis are eliminated, counting 100 and 47 EAM points selected in the LV and RV, respectively.

A 3D monodomain simulation is performed considering our method's best and worst generated Purkinje networks. Tissue conductivity was decreased by half in order to result in a tissue LAT close to the ones provided by the EAM points. From Figure 40, the RMSE values were around $6.3ms$ and $7.2ms$, for the best and worst PN in the EAM points located in the LV, respectively. For the points within the RV, the RMSE is $4.8ms$ in both PNs. Regarding the RRMSE results, the values for the LV points were around 11.2% and 12.7% for the best and worst PNs, respectively, while for the RV the values are between 11.1% in both PNs. Based on these results, we noticed that the LV networks have the highest errors, indicating the complexity of matching activation around this region. In addition, another issue that might have caused an increase in the errors in both biventricular areas is that we do not consider fibrosis and ischemia in the tissue, which are known mechanisms that can alter the LAT depending on its geometry and characteristics.

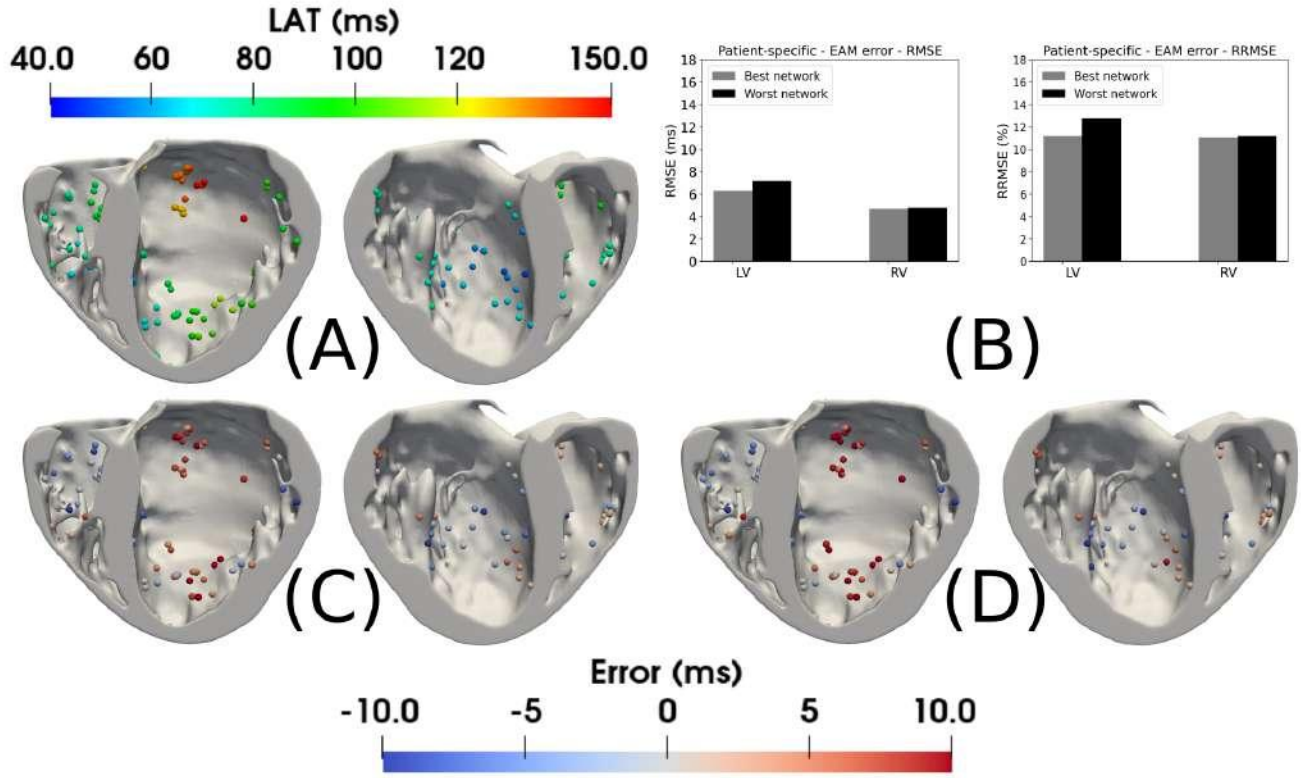


Figure 40 – Overview of the EAM points selected for LAT validation alongside results for the RMSE and RRMSE errors from the EAM points selected in the patient-specific LV and RV endocardial surface are shown in panel (A). In panel (B), the RMSE and RRMSE errors were calculated for all the EAM points by taking the difference between the LAT given by the reference times and the LAT value from the closest tissue cell generated by a monodomain simulation with the best and worst PNs. In panels (C) and (D), the LAT error at the EAM points for the best/worst PNs are illustrated.

6 Results: Effects of Purkinje Network Variability with Pseudo-ECG Simulations

Based on the previous results that demonstrate that several factors can affect ventricular activation when a PN is coupled to the tissue, we propose a final experiment to study the effects of different PVJ density on electrical activation of a patient ventricles and in matching their corresponding clinical ECG. For this particular experiment, the fourth Patient-specific mesh from *Camps, J. et al (2020)* [12] is used. Moreover, this mesh is also referred to as *DTI003* in the work [12].

In addition, this study utilizes an already built minimum geodesic PN that connects the 6 estimated active PVJs (root nodes) of the mesh, which are able to approximate the available clinical ECG of the patient with a certain accuracy. The position and LAT of the active PVJs are estimated using an inference technique that uses an efficient sequential Monte Carlo approximate Bayesian computation-based method, integrated with Eikonal simulations and torso-biventricular models constructed based on clinical cardiac magnetic resonance (CMR) imaging [12]. Regarding the minimum geodesic PN, it activates the 6 estimated PVJs simultaneously at $40ms$ and it was constructed using the method provided by *Barber, F. et al.* [4].

6.1 Extra branched Purkinje networks

In the experiment of this chapter we aim to explore the ability of our method to generate additional branches to an already constructed PN. Two different extra branched PNs are generated in order to cover the endocardium surface. Thus, these PNs are used to produce new active PVJs density clouds that activate the ventricular tissue in a manner that tries to sustain the baseline LAT that is responsible for reproducing the clinical ECG of this subject. The extra branching procedure considers the inclusion of new passive branches to the initial root passed as an input parameter to the method, as can be seen in Figure 42.

To further investigate the effects of the PVJ density, a fast endocardium layer is introduced over the cells that are within a certain endocardium region in the LV and RV, as illustrated by the transmurality colormap of the mesh depicted in Figure 41. The inclusion of the fast endocardium layer is justified to reproduce an activation pattern similar to the one given by the clinical ECG. To achieve a reasonable approximation for the ECG, the VCs that are located within the fast endocardium layer have their conductivity increased by a factor of 4. To select this particular value, several monodomain simulations were executed and the Pseudo-ECG was compared to the clinical data. Moreover, in this experiment, we want to check variability in the patient ECG by using different PNs and different densities of PVJ clouds.

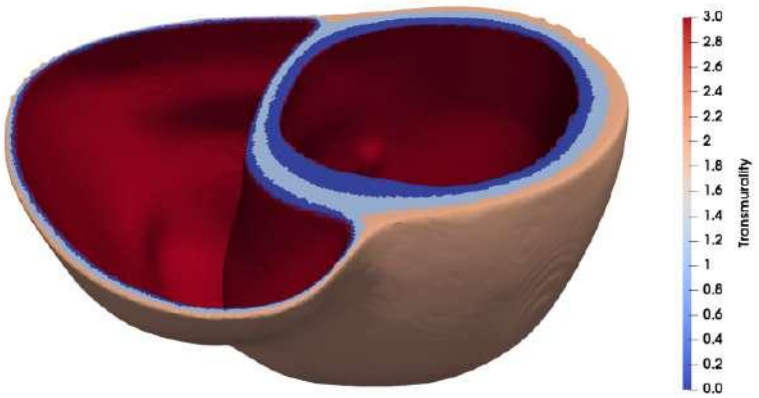


Figure 41 – Illustration of the transmurality of the Patient-specific mesh from *Camps, J. et al (2020)* [12], where in red the fast endocardium layer is highlighted. In addition, the endocardium cells are colored in dark blue, the mid-myocardium cells in light blue and the epicardium cells are in pink.

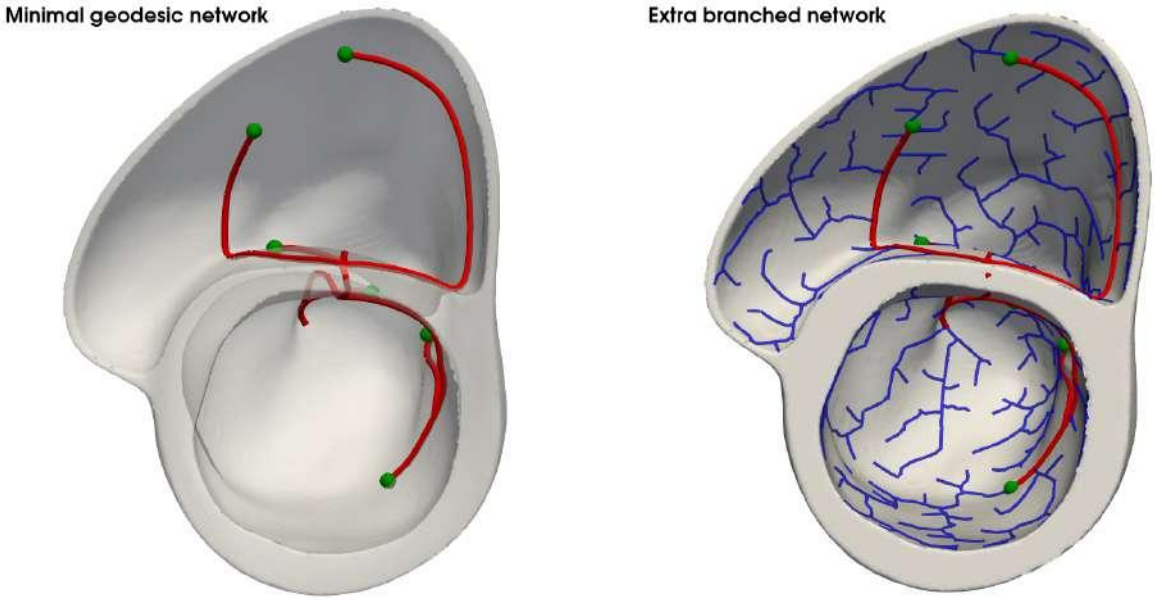


Figure 42 – Illustration of the extra branching procedure utilized to grow new passive branches over the structure of the minimal geodesic PN given by *Barber, F. et al.* [4]. The minimum geodesic PN is colored in red, the extra branches generated by our method are in blue and the 6 estimated PVJs are represented as green spheres.

6.2 Pseudo-ECG simulations

Monodomain simulations are executed and the Pseudo-ECG is calculated following the guidelines presented in section 2.6. Furthermore, in this study, whenever the ECG is calculated we consider the term D_b in equation (2.51) constant and equal to 20. Regarding the baseline parameters to enable the solution of the monodomain model, the surface-to-volume ratio is set

to $\beta = 0.14\mu m^{-1}$, tissue capacitance is equal to $C_m = 100pF/\mu^2$. A total simulation time of $t_{max} = 200ms$ and a time discretization of $dt = 0.01ms$ was used to solve the associated PDE. For the ODEs system related to the Purkinje and ventricular cellular models, a Rush-Larsen scheme was applied with a fixed timestep of $dt = 0.01ms$ for both the *Trovato* and *ToRORd* models. A space discretization of $400\mu m$ was used for all the ventricular domains, while for the Purkinje this value was set to $100\mu m$. The conductivities from the ventricular domain are anisotropic with $\sigma_l = 0.375S/m$, $\sigma_t = 0.1125S/m$ and $\sigma_n = 0.05625S/m$, in the longitudinal, transverse and normal direction, respectively, resulting on CV that was able to reproduce the given clinical ECG with reasonable accuracy. For the Purkinje domain the conductivity was set to $\sigma_{purk} = 2.75S/m$ with a CV approximately equal to $2m/s$. Purkinje coupling parameters were set to $R_{PVJ} = 500k\Omega$ and $N_{PVJ} = 60$, resulting on a characteristic delay in the range of $2ms$. The stimulus protocol is a single pulse coming from the His-bundle with the following parameters: $I_{amp} = 40pA/pF$, duration $= 2ms$, $N_{cells} = 25$.

For the Pseudo-ECG computation, the 10 electrodes positions used to compute the 8-leads ECG are shown alongside the Patient-specific mesh from *Camps, J. et al (2020)* [12] in Figure 43. In addition, the shape of the 8-lead clinical ECG is shown in Figure 44 after been normalized following the guidelines presented in section 2.6.

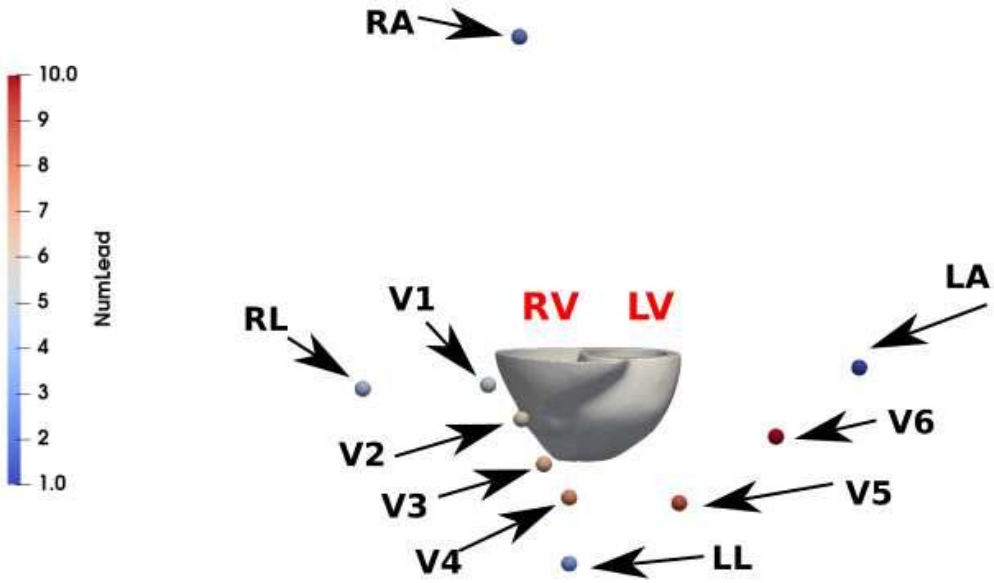


Figure 43 – Electrode positions used for the 8-lead ECG computation in the Patient-specific mesh from *Camps, J. et al (2020)* [12].

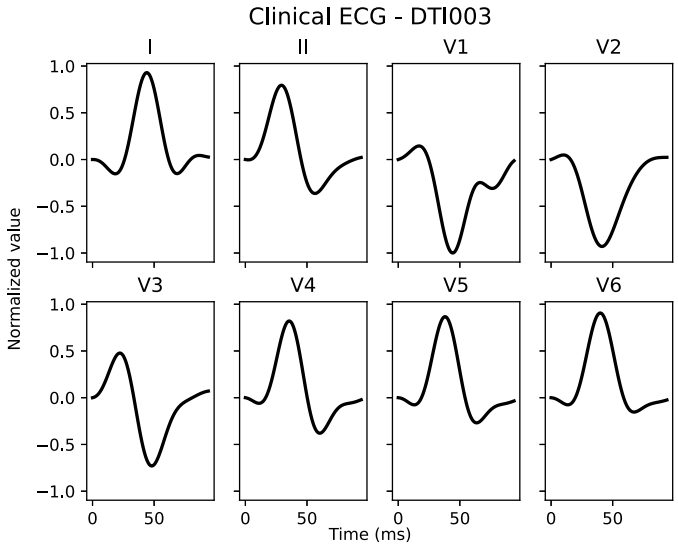


Figure 44 – Clinical 8-lead ECG for the Patient-specific mesh from *Camps, J. et al (2020)* [12].

For the generation of additional active PVJs, we developed a technique to activate only a certain percentage of the inactive PVJs (terminals) present in the extra branched PNs in way that the activation of this new active PVJs do not alter the original clinical ECG. The main idea is to first compute the LAT both for the PN and the biventricular tissue considering only activation coming from the 6 estimated root nodes. From this initial result, we analyze the LAT of each Purkinje terminal cell and their closest tissue cell. If the difference between the tissue LAT, t_{tiss}, and the Purkinje terminal LAT, t_{pk}, is below a certain time threshold, t_{thresh}, then we consider this Purkinje terminal a new active PVJ. In summary, this procedure only activates the Purkinje terminals that activated almost at the same time as the tissue in the 6 root nodes simulation. Consequently, if we choose wisely the time threshold t_{thresh}, we can increase the number of active PVJs in the extra branched PN, minimizing the loss in the accuracy of the baseline LAT.

In the following experiment, we define a low PVJ density cloud over the extra branched PNs when the time threshold parameter is set to $t_{thresh} = 5ms$, while a high PVJ density cloud is defined when $t_{thresh} = 10ms$. For the first extra branched PN shown in Figure 45 we have a total of 35 and 62 active PVJs for the low and high PVJ density cloud, respectively. For the second extra branched PN, these numbers are 58 and 81, respectively. An important observation regarding these numbers is that the 6 original root nodes are also included within the PVJ clouds, as can be seen in Figure 45.

In Figure 45 the results for the LAT of the two extra branched PNs are shown. As can be seen from these results, the PVJ location and density, alongside the proper PN morphology affect the LAT of the tissue. For instance, in the first extra branched PN we noticed that for the low PVJ density shown in Figure 45B, the resultant LAT is close to the baseline, given by Figure 45A. On

the other hand, for the second extra branched PN the inclusion of the new active PVJs results in a different LAT, with regions of the biventricular tissue being activated sooner than expected. When the high PVJ density clouds, illustrated in Figure 45, are analyzed, we notice a similar behaviour with more regions being depolarized early, which is understandable since we increased the time threshold parameter t_{thresh}.

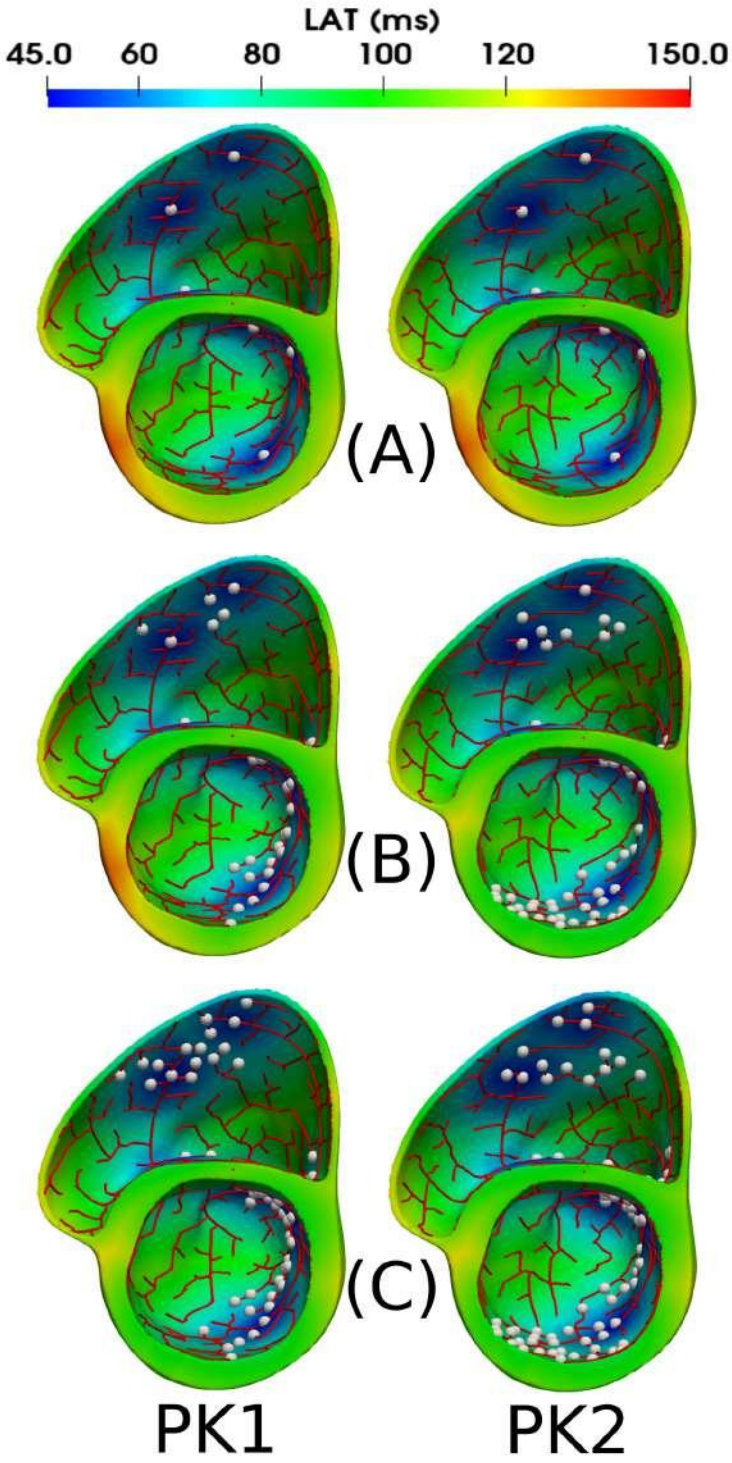


Figure 45 – Results for the LAT of the two extra branched PNs with different active PVJ density clouds. In panel (A), both PNs only activate the 6 inferred root nodes, while in panels (B) and (C) the low and high active PVJ clouds generated for the two PNs using $t_{thresh} = 5ms$ and $t_{thresh} = 10ms$, respectively, are shown.

Analyzing the computed Pseudo-ECG of the above trees, we noticed from Figure 46 that

when only the 6 root nodes are activated, both PNs have the same approximation for the ECG with an averaged Pearson correlation of 0.77. From these results, the worst approximation is observed for Lead-I with a correlation of 0.18 in both PNs, while the best result is for Lead-II with 0.94 in both trees. Other leads also presented good approximations, as can be seen in the traces of leads V4, V5 and V6, with correlation values around 0.92 and 0.93.

In the case of low PVJ density clouds are activated by the PNs, which are shown in Figure 47, we observed an increase in the Pearson correlation of both PNs when compared to the previous result. This increase is more pronounced in the results from the second PN, with a value of 0.87, while for the first PN this value is 0.78.

Finally, when the high PVJ density cloud results depicted in Figure 48 are analyzed, we verified an increase in the Pearson correlation value to 0.81 for the first PN and a decrease of this metric for the second PN to 0.85. Based on these results, we noticed that maybe activating the original 6 estimated root nodes plus certain regions of the ventricle at the right time can improve the clinical ECG matching. In particular, these regions can be located around the anterior wall of the LV, since the activation of the PVJs near this region increased the Pearson correlation for Lead-I from 0.18 to 0.44 in the first PN. When the second PN is analyzed, the increase in the Pearson correlation for Lead-I is even more pronounced from 0.18 to 0.79.

In summary, these results also demonstrate that the problem is a source of uncertainties. First of all, it is possible to generate Purkinje networks with a very distinct morphology but that share a common LAT. Alongside with that, the proper Purkinje network can have different combinations of active PVJs depending on the PVJ density cloud that was used. This observation can be seen on the ECG results where the Pearson correlation varied accordingly to these two factors.

In addition, PVJ density may change due to heart size and aging [14]. Another factor that can alter the total number of active PVJs is ischemia [36]. An increase in the number of PVJs can lead to more retrograde conduction and allow more escape paths for arrhythmias but can also decrease tissue heterogeneity. Thus, its actions are both pro-arrhythmic and anti-arrhythmic [11]. Cardiac arrhythmias associated to reentry dynamics coming from the Purkinje can also be impacted by the PVJ density of the network. This is because a reentry circuit occur due to two factors: 1) a block in conduction, which can happen depending on the number of active connections between the Purkinje network and ventricular tissue; 2) a long pathway within the Purkinje network that can reactive a portion of the tissue [1]. Within this context, both of these factors can be further explored using the available tools that were developed and used during this work.

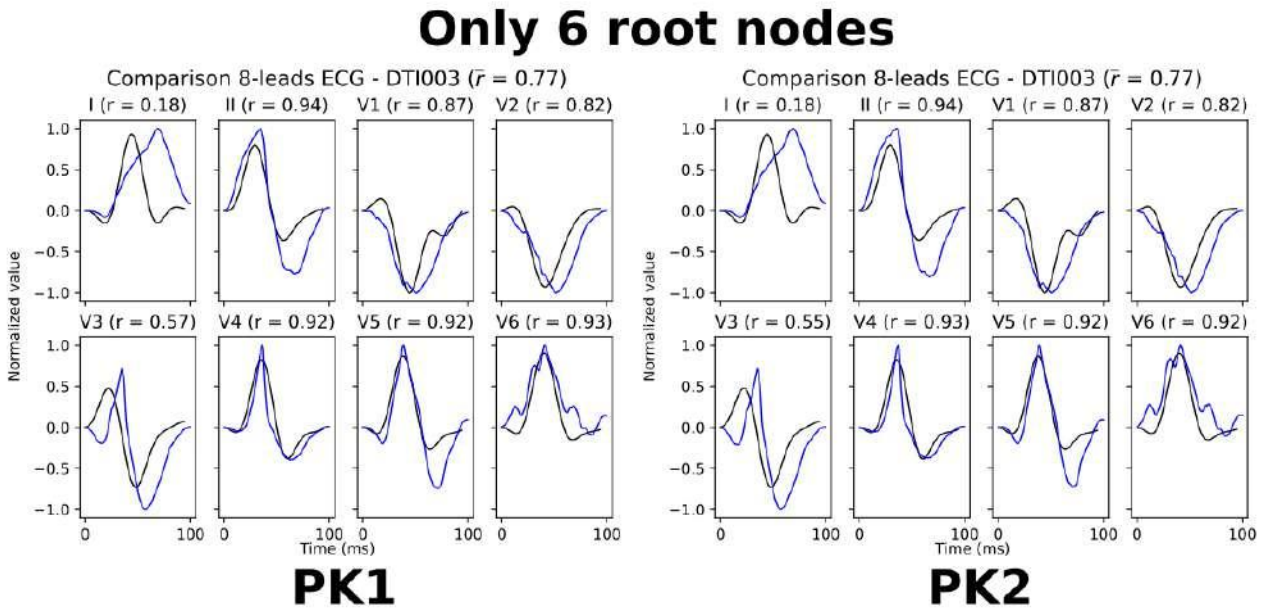


Figure 46 – Results for the 8-lead Pseudo-ECG from both extra branched PNs when only the 6 estimated root nodes from *Camps, J. et al.* [12] are activated by the two PNs. The clinical ECG is highlighted in black, while the Pseudo-ECG approximation is colored in blue.

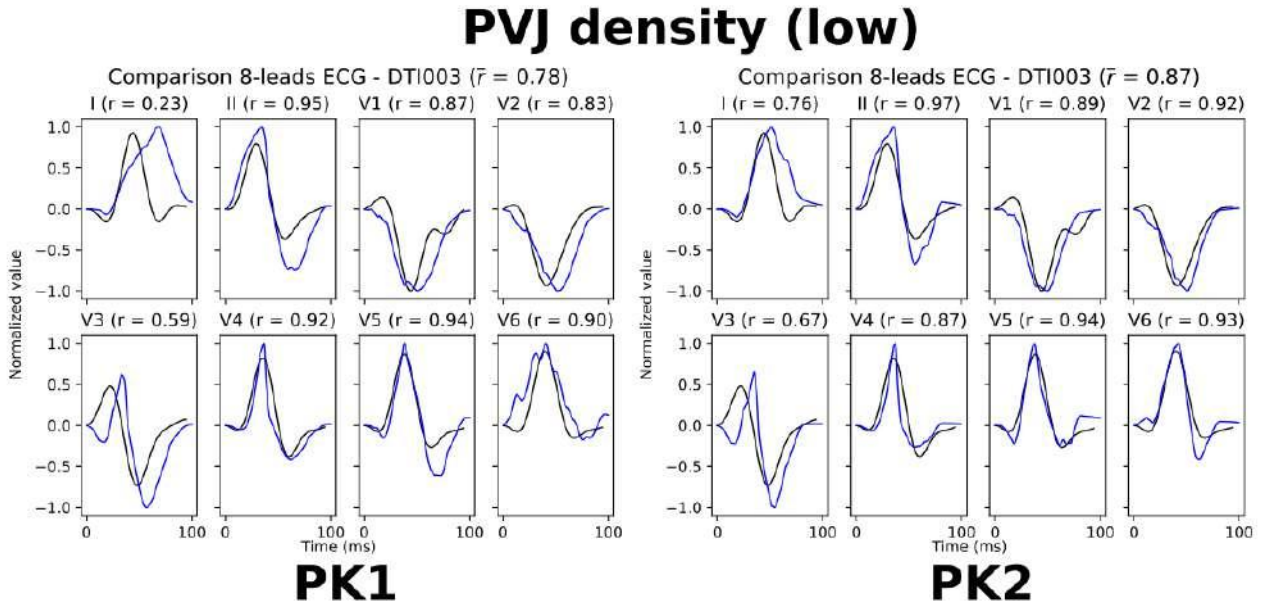


Figure 47 – Results for the 8-lead Pseudo-ECG from both extra branched PNs when the low PVJ density cloud are activated by the two PNs. The clinical ECG is highlighted in black, while the Pseudo-ECG approximation is colored in blue.

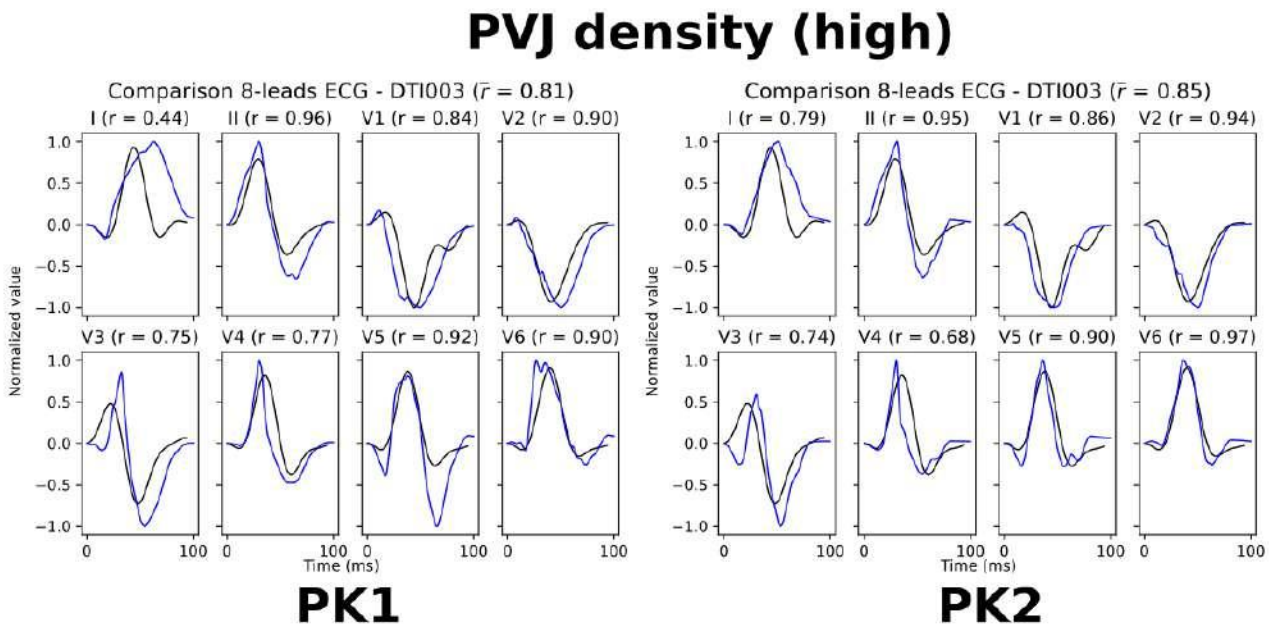


Figure 48 – Results for the 8-lead Pseudo-ECG from both extra branched PNs when the high PVJ density cloud are activated by the two PNs. The clinical ECG is highlighted in black, while the Pseudo-ECG approximation is colored in blue.

In addition, in Figure 49 we can verify the differences that appear in the ECG as not only the topology of the PN changes but also its PVJ density. Based on this figure and on the previous ones, we can conclude that both the geometry of the PN and PVJ density cloud that activates the tissue have an effect in the ECG. These two parameters combined can either made the ECG approximation more closer to the aspect of the clinical ECG, but also diverge from the reference depending on the combination of these parameters. For that reason, we can also conclude that there is a high uncertainty associated to the generation and activation of PNs.

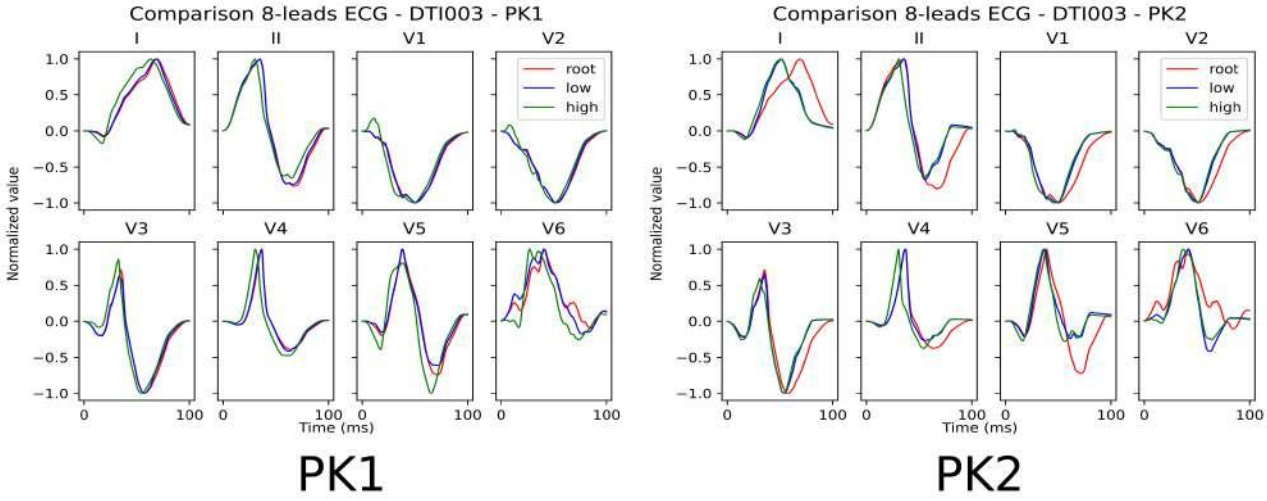


Figure 49 – Comparison between the 8-lead Pseudo-ECG from both extra branched PNs with different PVJ density clouds, where in red the ECG calculated using only the 6 root nodes are depicted, colored in blue is the resultant ECG when a low density PVJ cloud is used, and illustrated by a green line, is the ECG computed when a high density PVJ cloud is applied to activate the tissue.

7 Conclusions and Future works

Prior to this thesis, no models for patient-specific PNs combine geometric and electrical precision, which are essential for more accurate development of cardiac electrical simulations. In addition, the recent increase in the development of Cardiac Digital Twins (CDTs) turns the usage of Purkinje generation methods an essential step towards more realistic simulations. Furthermore, to study cardiac pathologies, like, for example, arrhythmias, models that consider PNs have substantial importance since this structure is known to be a source of ventricular fibrillation [24].

In this work, we presented a new method called *Shocker* for the automatic construction of the PN to be used in cardiac electrical simulations. The method is an extension of the work from *Ulysses, J. N. et al.* [76], which is based on optimization principles and will be available for usage in a public repository. Our primary contribution to the model is a new cost function that relies on electrical principles using the classic cable equation and approximates the LAT of a given set of active PVJs. We demonstrated the capabilities of the proposed method with the generation of different PNs over four biventricular meshes. In addition, the electrical activity of the best PNs generated by our method is evaluated on a biventricular Purkinje-coupled monodomain simulation using the latest human ventricular/Purkinje cellular models and a high-performance GPU solver.

Our results, both in terms of activation times and geometric features, indicate that the *Shocker* method can capture key aspects of the PNs (bifurcation angles, branch sizes, and LAT at the active PVJs sites) while preserving the structure within the endocardial surfaces, especially at the regions near the papillary muscles. Secondly, the method is flexible and scalable by allowing PNs generated by other methods to be used as initial roots. As a result, these networks could extend their branches to attend to the desired objective, such as increasing the PN endocardial coverage or linking additional PVJs.

The main findings of this study are that very different Purkinje networks can give rise to similar LAT. From the results of the experiments it is demonstrated that different pathways can be used to reach an active PVJ site at the specified LAT. Also the usage of the cable equation to approximate the LAT of a growing Purkinje network is a low-cost and feasible solution to be utilized instead of the Eikonal model, which is applied in other similar works [48]. Finally, the source-sink mismatch related to the PVJ coupling and characteristics of the mesh, like fiber orientation and irregular surfaces, are found to affect the activation of the PVJ sites.

Regarding the main contributions of this work we present a novel method to generate Patient-specific Purkinje networks over an endocardium surface with geometric and electrical accuracy. Purkinje-coupled monodomain simulations were performed using the latest models of human Purkinje and ventricular cells, and GPU-based monodomain solver [59]. Furthermore, several improvements to the respective solver: 1) Implemention of the Purkinje module; 2) Implementation

of new cellular models using RL schemes; 3) Help with the development of the Pseudo-ECG module; 4) Several functions and scripts associated to a simulation (AP and ECG traces, APD and error computation, CV tuning, PVJ generation, mesh parsing, video generation and more). In addition, it was developed a methodology which permits the generation of several PNs that reproduce the electroanatomical map and electrocardiogram. The generated PNs differ in their morphology and quantity of active PVJs. From this tool it is possible to realize uncertainty quantification studies for different PNs of a patient.

The main limitations of the proposed method are its high computational cost for very detailed mesh, which based on the results are mainly caused by the geodesic pathway function. To attenuated this problem more optimized implementations of the geodesic pathway algorithm can be used in order to reduce the bottleneck of the method. Secondly, the PVJ coupling experiment was only used to find a reasonable value for the PVJ coupling parameters when the two human cellular models are coupled. Therefore, a more robust scheme is necessary for a more accurate choice of the PVJ coupling parameters. Also from the experiments it is observed that the PVJ characteristic delay plays an important role in the activation and should be accounted in the active cost function of the method.

For future works, the applicability and performance of the model can be further improved. For instance, the usage of additional electrical information, such as ECG readings can, be an alternative to generate the Purkinje networks. Furthermore, the computational time of the cost function evaluation step can be executed faster if the calculus is made using parallel programming. A further improvement to the method is to include an adaptive diameter adjustment in the segments of the growing tree. Also, we can improve the robustness of the PVJ coupling experiment considering fiber orientation, different conductivities for the Purkinje cable and more evaluations for the R_{PVJ} and N_{PVJ} parameters. The accuracy of the current RL scheme utilized to solve the *Trovato2020* and *ToRORd-dynCl-2020* models can be improved by implementing the high-order and adaptive schemes [22, 69] in the GPU-based monodomain solver [59]. Finally, the presented novel methods can be used to study how different PN topology and features are related to the initiation and maintenance of cardiac arrhythmias.

APPENDIX A – *Shocker method description*

A.1 Pre-processing steps

The *PreProcessing* subroutine, described in **Algorithm 2** and illustrated in Figure 50A, is responsible for filtering a percentage of the points within the input set S and sorting the active PVJs, returning the passive set of points S_p based on the proximal location of the root branch x_{prox} and the active points set S_a.

Algorithm 2: *PreProcessing* subroutine.

Data: S, x_{prox}, **global** $r_0 = 0.01mm$, **global** $r_{inc} = 0.05mm$.
Result: Passive and active cloud of points S_p and S_a.

```
r ← r_0 ;
S_p ← ∅ ;
while (|S_p| < |S|) do
    for i ∈ {0, ..., |S|} do
        x ← S[i] ;
        dist ← ||x − x_prox|| ;
        if (dist < r) then
            Append x to S_p ;
            Erase x from S ;
        end
    end
    r ← r + r_inc ;
end
S_p ← Filter 1% percentage of the passive points inside set S_p ;
S_a ← ExtractActivePVJ(S); # Algorithm 3 ;
return S_p, S_a ;
```

The *ExtractActivePVJ* subroutine, presented in **Algorithm 3**, manages the set S by extracting the points that are considered active PVJs. In particular, each active PVJ must have an associated LAT value as a reference for comparison. With this value, we sort the active PVJ points using the *QuickSort* algorithm [15].

Algorithm 3: *ExtractActivePVJ* subroutine.

Data: S.

Result: Active cloud of points S_a sorted by their LAT.

```
S_a ← ∅ ;
for i ∈ {0, ..., |S|} do
    if (S[i] is active) then
        x ← S[i] ;
        lat ← S[i].LAT ;
        Append the pair (x, lat) to S_a ;
    end
end
Apply QuickSort algorithm on S_a using lat as comparison parameter ;
return S_a ;
```

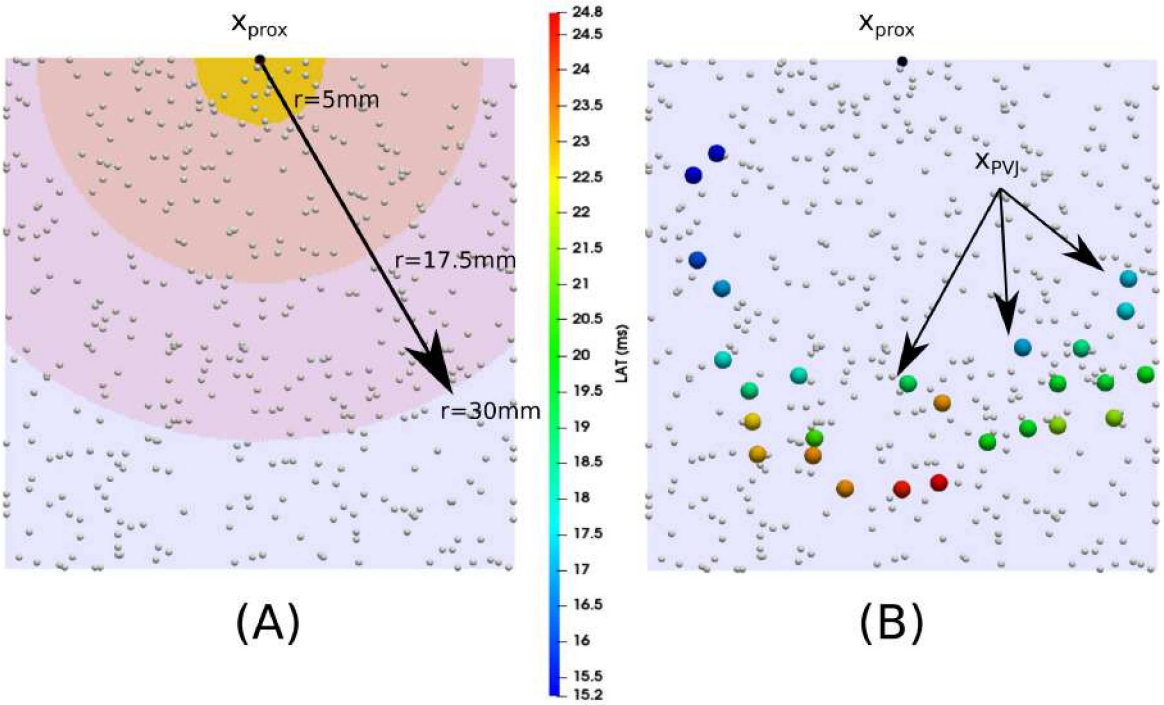


Figure 50 – Illustration of the *PreProcessing* and *ExtractActivePVJ* subroutines, where the *PreProcessing* function is shown in panel (A) and *ExtractActivePVJ* is depicted in panel (B). In panel (A), the remapped and filtered passive points S_p are highlighted by white spheres. In addition, the evolution of the growing sphere used to remap the points in set S is illustrated by the distinct values of its radius r. In panel (B), the passive points set S_p is highlighted by white spheres, while the active points set S_a is highlighted by colored spheres using the LAT colormap.

A.2 Root placement

The *RootPlacement* subroutine, detailed in **Algorithm 4**, is responsible for generating or loading the initial root of the PN. The method starts by constructing the root branch, in which, given the proximal location x_{prox}, a distal location x_{dist} is randomly selected from the passive set S_p and must attend the distance criterion. After selecting a feasible position, a geodesic pathway which connects x_{prox} to x_{dist} is constructed. Alternatively, the user can also provide an initial root structure that can be used to initialize the root of the method.

Algorithm 4: *RootPlacement* subroutine.

Data: S_p, l_d, x_{prox}, [*initial PN*])
Result: Initial state of the Purkinje network.

if *(initial PN is given)* **then**
  $k_{term} \leftarrow$ Load initial PN topology from file ;
else
  $is\_root\_ok \leftarrow False$;
  $counter \leftarrow 0$;
  $d_{thresh} \leftarrow l_d$;
  while $(is\_root\_ok == False)$ **do**
    $x_{dist} \leftarrow SortPoint(S_p)$; # **Algorithm 5** ;
    $d \leftarrow ||x_{dist} - x_{prox}||$;
    if $(d >= d_{thresh})$ **then**
      $is\_root\_ok = True$;
    else
      $counter \leftarrow counter + 1$;
    end
    if $(counter > 8)$ **then**
      $d_{thresh} \leftarrow d_{thresh} * 0.95$;
    end
  end
  Geodesic pathway: Build a geodesic path from x_{prox} to x_{dist} ;
  $k_{term} \leftarrow 1$;
end
return k_{term} ;

The subroutine *SortPoint* presented in **Algorithm 4** is responsible for selecting a point from the passive set S_p and is detailed in **Algorithm 5**. The points from the passive set are selected sequentially, and to allow variability in the PNs, this selection is made using different gap values. This gap value is defined as the difference between the previous and current index selected

in S_p. This value is calculated by taking the modulo between a random integer number, computed using the default *rand* function from the *C++* standard library, and a fixed offset parameter, which was defined as 4 in this work. To avoid an index out of bounds error, we always apply a modulo operation between the next index and the total number of passive points in set S_p. When the *next_id* returns to the start of the S_p array, the stop criterion flag of the method's main loop is activated.

Algorithm 5: *SortPoint* subroutine.

Data: S_p, **global** *prev_id*

Result: Index of the next point to be selected in the set S_p.

```
1 k ← rand()%4 + 1 ;
2 next_id ← (prev_id + k)%|S_p| ;
3 while ( S_p[next_id] is taken ) do
4 |   prev_id ← next_id ;
5 |   k ← rand()%4 + 1 ;
6 |   next_id ← (prev_id + k)%|S_p| ;
7 end
8 return S_p[next_id] ;
```

A.3 Distance criterion

The subroutine *DistanceCriterion* in **Algorithm 6** controls if all segments in set P_s attend the distance criterion for a given x_{term} position.

Algorithm 6: *DistanceCriterion* subroutine.

Data: x_{term}, d_{thresh}, P_s

Result: Return *True* if all segments in set P_s attend the criterion or *False* otherwise.

for $i \in \{0, \ldots, |P_s|\}$ **do**
  $u \leftarrow x_p(i) - x_d(i)$;
  $v \leftarrow x_{term} - x_d(i)$;
  $w \leftarrow x_{term} - x_p(i)$;
  $d_i \leftarrow u.v/||u||^2$; # **scalar product** ;
  if *(* $d_i >= 0$ ***and*** $d_i <= 1$ *)* **then**
    $d_{crit} \leftarrow ||v \times w||/||u||$ # **cross product** ;
  else
    $d_{crit} \leftarrow \min\{||v||, ||w||\}$;
  end
  if *(* $d_{crit} < d_{thresh}$ *)* **then**
    return *False* ;
  end
end
return *True* ;

A.4 Cost function evaluation

The *EvaluateCostFunction* subroutine detailed in **Algorithm 7** evaluates a given cost function CF for every element in the feasible segment set F_s.

Algorithm 7: *EvaluateCostFunction* subroutine.

Data: x_{term}, F_s, CF

Result: A list of evaluations E using cost function CF.

$E \leftarrow \emptyset$;
for $i \in \{0, \ldots, |F_s|\}$ **do**
  Geodesic pathway: Build geodesic pathway linking x_{dist} from segment s_i to x_{term} ;
  $e \leftarrow$ Evaluate CF under this PN topology ;
  if *(No collision is detected)* **then**
    Append e the evaluation to E ;
  end
end
return E ;

A.5 Generation of passive terminals

The subroutine *GenerateTerminal* depicted in **Algorithm 8** and illustrated in Figure 51, generates a new passive terminal branch to the PN. The generation proceeds by sorting a point x_{term} from set S_p using **Algorithm 5**. The prospective location x_{term} is accepted if x_{term} satisfies the distance criterion, which is defined by **Algorithm 6**, otherwise we select a new point from S_p. In case the above procedure fails 10 times, i.e., $d_{crit} < d_{thresh}$, the threshold distance d_{thresh} is decreased by a factor 0.95. This is repeated until the acceptance of x_{term}.

To generate a new passive terminal branch, after x_{term} has been accepted as a distal end of a new terminal branch, it is temporarily connected to the nearest N_p segments in the model by a geodesic pathway. Moreover, N_p is an input parameter of the method that represents the maximum number of temporary connections for a new passive terminal location x_{term}. The calculus of the N_p nearest segments is done by calculating the distance between the middle point of all segments in the current network to x_{term}. Next, we sort the segments by their Euclidean distance and filter only the closest N_p segments to x_{term}. This step is defined by the *FillFeasibleSegmentsPassive* subroutine in **Algorithm 9**. After this step, we decide how to connect x_{term} to one of the N_p neighboring segments. This choice is based on the minimization of the passive cost function CF_p.

Algorithm 8: *GenerateTerminal* subroutine.

Data: S_p, N_p, CF_p, l_d, k_{term}

Result: New terminal branch added to the Purkinje network.

$d_{thresh} \leftarrow \sqrt{l_d^2/k_{term}}$;
$tosses \leftarrow 0$;
$point\_is\_ok \leftarrow False$;
$P_s \leftarrow$ Get all segments from the PN ;
while *(point_is_ok == False)* **do**
    $F_s \leftarrow \emptyset$;
    $x_{term} \leftarrow SortPoint(S_p)$; # **Algorithm 5** ;
    # **Algorithms 6 and 9** ;
    if *(DistanceCriterion(x_{term}, d_{thresh}, P_s)* ***and*** *FillFeasibleSegmentsPassive(x_{term}, N_p, F_s, P_s))* **then**
        $E_p \leftarrow EvaluateCostFunction(x_{term}, F_s, CF_p)$; # **Algorithm 7** ;
        if *($|E_p| > 0$)* **then**
            $point\_is\_ok = True$;
        end
    else
        $tosses \leftarrow tosses + 1$;
        if *($tosses > 10$)* **then**
            $tosses \leftarrow 0$;
            $d_{thresh} \leftarrow d_{thresh} * 0.95$;
        end
    end
end
Geodesic pathway: Build a branch linking x_{dist} from the segment with the best CF_p evaluation in E_p to x_{term} ;
return x_{term} ;

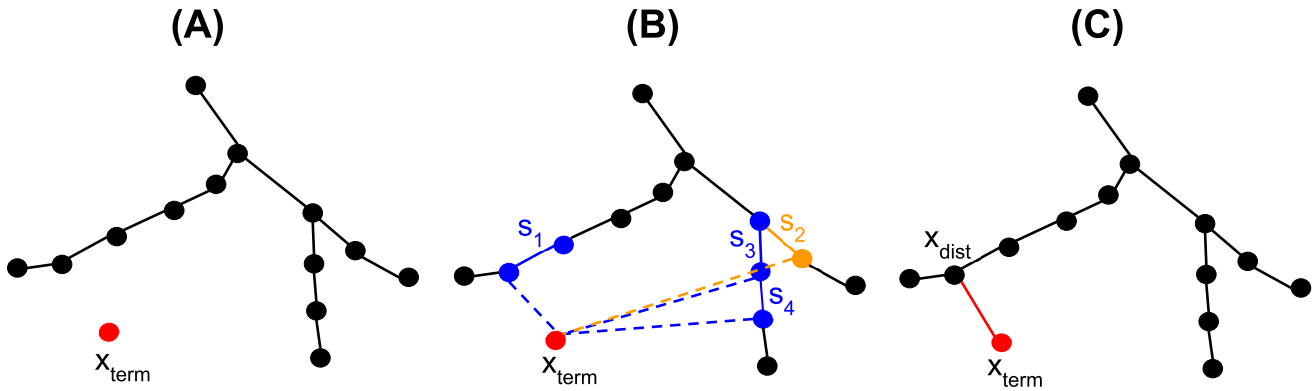
The *FillFeasibleSegmentsPassive* subroutine as presented in **Algorithm 9** is responsible for populating the F_s set for the passive cost function with the nearest N_p segments to the x_{term} position. This is done by passing through all the segments in the current PN and computing the Euclidean distance between the middle position of each segment, x_M, and x_{term}.

Algorithm 9: *FillFeasibleSegmentsPassive* subroutine.

Data: x_{term}, N_p, F_s, P_s

Result: Fill the feasible segments set F_s to connect x_{term}.

```
for i ∈ {0, ..., |P_s|} do
    dist ← ||x_term − x_M(s_i)|| ;
    Append (i, dist) to F_s ;
end
Apply QuickSort algorithm on F_s using dist as comparison parameter ;
return F_s[0 : N_p] ;
```



Feasible segment	Cost function evaluation
s1	**2**
s2	Collision!
s3	4
s4	5

→ Best evaluation

Figure 51 – Illustration of the *GenerateTerminal* subroutine. In panel (A), a prospective location for a new terminal branch, x_{term}, is selected after attending the distance criterion. Next, in panel (B), N_p feasible segments are evaluated by the cost function CF_p and ranked in a table. After the evaluation step, segment 2(s2) is unfeasible for the connection since it generates a collision in the PN. Segment 1 returns the minimum value for the cost function and is considered the best evaluation. Finally, in panel (C), a new terminal branch is constructed by linking the distal position of segment 1, x_{dist}, to the location x_{term}.

A.6 Generation of active terminals

For the case where k_{term} is divisible by L_{rate}, the terminal point is considered to be an active PVJ. In this scenario, we try to sequentially connect all the unconnected PVJs inside the set S_a using the same technique previously described for the passive case. Although, we use the nearest N_a segments alongside the cost function given by Eq. (3.3) and an additional constraint is

imposed to accept the connection. The new branch is only made permanent if the absolute LAT error between the best candidate branch and the reference LAT value of the target active PVJ is less or equal to L_{error}. Otherwise, the PVJ returns to set S_a and the entire branch is pruned. Moreover, if at least one PVJ is connected to the PN, the procedure is repeated until there are no updates to the tree. The active PVJs connection step is illustrated in Figure 52 and described in **Algorithms 10** and **11**.

The *AttemptPVJConnection* subroutine, described in **Algorithm 10**, attempt to connect all the remaining active PVJ points from set S_a to the current PN. This procedure is repeated until no new active PVJ can be connected to the structure.

Algorithm 10: *AttemptPVJConnection* subroutine.

Data: S_a, N_a, CF_a, L_{error}, l_d, k_{term}

Result: Try to connect remaining PVJs inside S_a with new terminal branches.

```
repeat
    for i ∈ {0, ..., |S_a|} do
        x_PVJ ← S_a(i) ;
        if (x_PVJ is not connected) then
            AttemptGeneratePVJ(x_PVJ, N_a, CF_a, L_error, l_d, k_term); # Algorithm 11 ;
        end
    end
until (No new PVJ is connected);
```

The *AttemptGeneratePVJ* subroutine described in **Algorithm 11** and illustrated by Figure 52, respectively, attempt to connect an active PVJ, x_{PVJ}, by generating a new terminal branch. The branch is only made permanent if the absolute LAT error between the best candidate branch and the reference LAT value of the target active PVJ is less or equal to L_{error}. Otherwise, the PVJ returns to set S_a and the entire branch is pruned.

Algorithm 11: *AttemptGeneratePVJ* subroutine.

Data: x_{PVJ}, N_a, CF_a, L_{error}, l_d, k_{term}

Result: Try to connect the x_{PVJ} within the LAT error tolerance with a new branch.

```
1  d_thresh ← sqrt(l_d^2 / k_term) ;
2  F_s ← ∅ ;
3  P_s ← Get all segments from the PN ;
4  # Algorithms 6 and 12 ;
5  if ( DistanceCriterion(x_PVJ, d_thresh, P_s) and FillFeasibleSegmentsActive(x_PVJ, N_a, F_s, P_s)
     ) then
6      E_a ← EvaluateCostFunction(x_PVJ, F_s, CF_a); # Algorithm 7 ;
7      if ( |E_a| > 0 ) then
8          Geodesic pathway: Build a branch linking x_dist from the segment with the best CF_a
            evaluation in E_a to x_PVJ ;
9          if (x_PVJ is not connected with an error less than L_error) then
10             Prune the previous branch with x_PVJ ;
11         else
12             Tag x_PVJ as connected ;
13         end
14     end
15 end
```

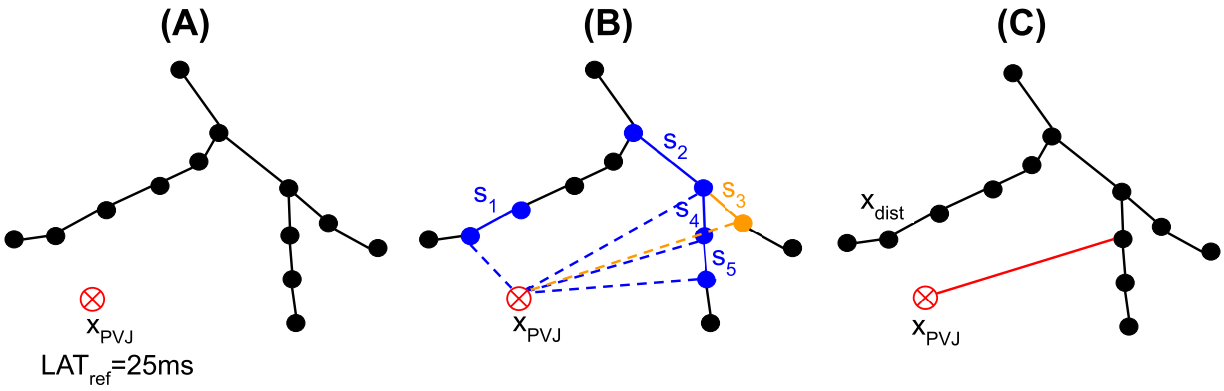
The *FillFeasibleSegmentsActive* subroutine as presented in **Algorithm 12** is responsible for populating the F_s set for the active cost function with the nearest N_a segments to the x_{PVJ} position. This is done by passing through all the segments in the current PN and computing an approximation of the LAT error to the x_{PVJ} reference value. This approximation is calculated by the sum of the current LAT of a given segment s_i and the LAT given by the line that links the middle position x_M of segment s_i to x_{PVJ}.

Algorithm 12: *FillFeasibleSegmentsActive* subroutine.

Data: x_{PVJ}, N_a, F_s, P_s

Result: Fill the feasible segments set F_s to connect x_{term}.

for $i \in \{0, \ldots, |P_s|\}$ **do**
  $dist \leftarrow ||x_{PVJ} - x_M(s_i)||$;
  $lat \leftarrow s_i.LAT + (dist/s_i.cv)$;
  $error \leftarrow |x_{PVJ}.LAT - lat|$;
  Append $(i, error)$ to F_s ;
end
Apply *QuickSort* algorithm on F_s using *error* as comparison parameter ;
return $F_s[0 : N_a]$;



Feasible segment	Cost function evaluation
s1	|25-30|=5ms
s2	|25-21|=4ms
s3	Collision!
s4	**|25-24|=1ms**
s5	|25-27|=2ms

→ Best evaluation

Figure 52 – Illustration of the *AttemptGeneratePVJ* subroutine. In panel (A), a prospective location for a new PVJ branch, x_{PVJ}, is selected after attending the distance criterion. Next, in panel (B), N_a feasible segments are evaluated by the cost function CF_a and ranked in a table. After the evaluation step, segment 3(s3) is unfeasible for the connection since it generates a collision in the PN and segment 4(s4) returns the minimum value for the cost function and is within the LAT error tolerance of $L_{error} = 2ms$, for that reason is considered the best evaluation. Finally, in panel (C), a new terminal PVJ branch is constructed by linking the distal position of segment 4, x_{dist}, to the location x_{PVJ}.

A.7 Post-processing steps

In the case there are remaining active PVJs to be connected after the main loop, we apply a post-processing step described in **Algorithm 13**. The first step in this procedure is to prune all passive segments of the tree. A segment is considered to be passive if it is not directly part of a pathway that links an active PVJ to the root. Next, the LAT error tolerance constraint is dropped by setting $L_{error} = \infty$, and we attempt to connect all unconnected PVJs in S_a using the pruned tree. The first N_a feasible segments sorted by the LAT error are evaluated using cost function CF_a and the segment with the best evaluation is connected using a geodesic pathway to x_{PVJ}.

However, after this procedure some PVJs still could not be connected due to the distance criterion. This scenario could happen if x_{PVJ} is already too close to the current tree. In this particular case, we drop the distance criterion for these PVJs and attempt to connect the point with a feasible segment which returns the minimum LAT error using a geodesic pathway. If there are no geodesic pathway possible for x_{PVJ}, we consider the 5 closest segments by distance to x_{PVJ} and force its connection to the one that returns the minimum LAT error using a straight line, regardless of any restriction. After this step, all the active PVJs that the user-specified are connected and the geometric and electric metrics are computed in this topology, which is referred to as *minimum network*.

Algorithm 13: *PostProcessing* subroutine.

Data: S_a, N_a, CF_a, L_{error}, l_d, k_{term}

Result: Minimum Purkinje network with all active PVJs connected.

```
Prune passive segments ;
L_error ← ∞ ;
for i ∈ {0, ..., |S_a|} do
    x_PVJ ← S_a[i] ;
    if (x_PVJ is not connected) then
        AttemptGeneratePVJ(x_PVJ, N_a, CF_a, L_error, l_d, k_term); # Algorithm 11 ;
        if (x_PVJ is not connected) then
            Remove the distance criterion for x_PVJ ;
            F_s ← ∅ ;
            P_s ← Get all segments from the PN ;
            FillFeasibleSegmentsActive(x_PVJ, N_a, F_s, P_s); # Algorithm 12 ;
            E_a ← EvaluateCostFunction(x_PVJ, F_s, CF_a); # Algorithm 7 ;
            if ( |E_a| > 0 ) then
                Geodesic pathway: Build a branch linking x_dist from the segment with the
                 best CF_a evaluation in E_a to x_PVJ ;
            else
                F_s ← Get the 5 closest segment by distance to x_PVJ ;
                Straight line: Build a branch linking x_dist from the segment with the best
                 CF_a evaluation in F_s to x_PVJ ;
            end
        end
    end
end
```

APPENDIX B – Cable equation

An important feature in the electrical conduction is the conduction velocity (CV) of the stimuli. The conduction velocity in a neuron can be calculated using the cable equation, given by expression (B.1) and considering that the cell membrane can be modeled as an one-dimensional cable as schematized in Figure 53.

$$I_m = p\left(C_m \frac{\partial V}{\partial t} + I_{ion}\right) = \frac{\partial}{\partial x}\left(\frac{1}{r_i + r_e}\frac{\partial V}{\partial x}\right), \tag{B.1}$$

where r_i and r_e define the intracellular and extracellular resistances per unit length, respectively; p is the cell perimeter, C_m is the membrane capacitance and I_{ion} is the ionic current and has unit of current per unit area.

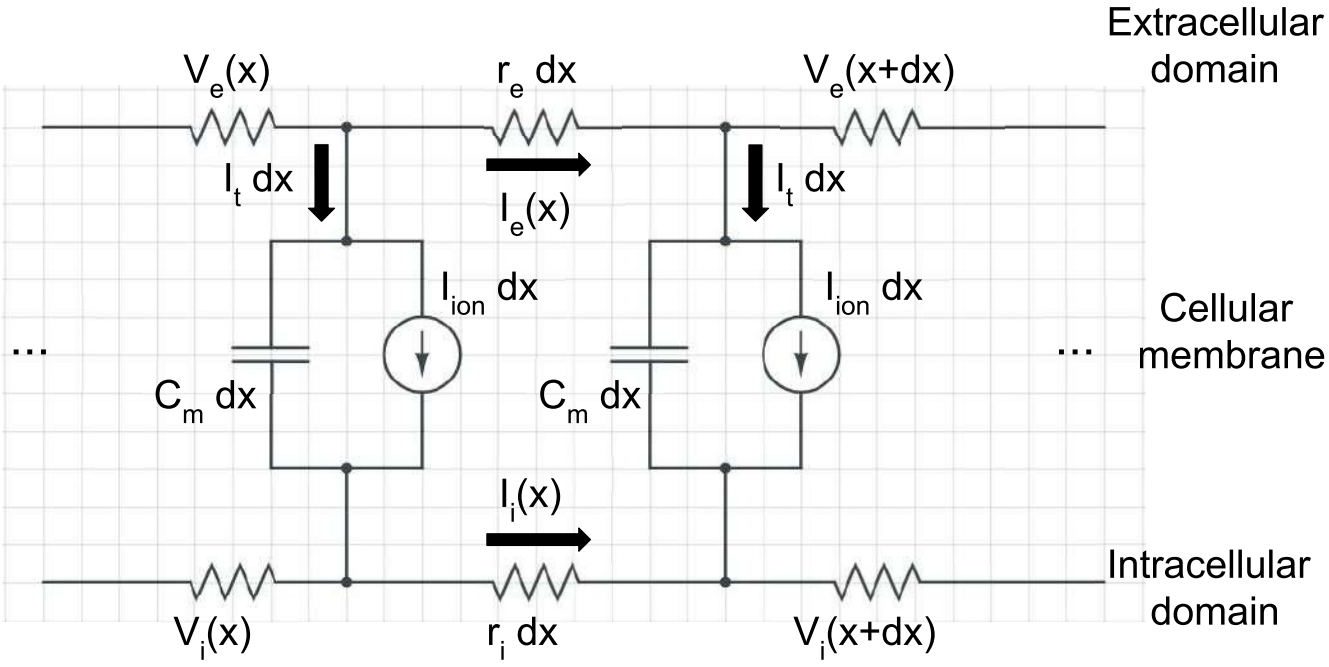


Figure 53 – Schematic of the cable model considered with isopotential circuit elements of size dx.

The cable equation is commonly used to evaluate the electrical flow in neurons and cardiac cells [35]. In this assumption, the potential depends only on the length variable so that the cable can be viewed as one-dimensional, just like a Purkinje fiber. The cell is considered a cylindrical piece of the membrane with a certain length, diameter, internal conductivity and membrane capacitance. The conduction velocity across the cable is given by equation (3.4).

Furthermore, the monodomain model is directly correlated to the cable equation. Consequently, it is possible to construct a correspondence between the parameters d, σ and the resultant conduction velocity from both models as Figure 54 shows. To calculate this relation, we ran several monodomain simulations over a $10cm$ linear cable composed of PCs that were modeled with *Trovato2020* Purkinje cellular model. The LATs of all the PCs were computed across the cable for different conductivity values. The conduction velocity of the cable was calculated by taking the difference between the LAT of the PCs that are $0.25cm$ distant from the middle PC of the cable. Next, we apply equation (3.4) to correlate each CV previously calculated to a diameter and finally adjust the line that best fits the sample points using the least-squares method, as shown in Figure 54.

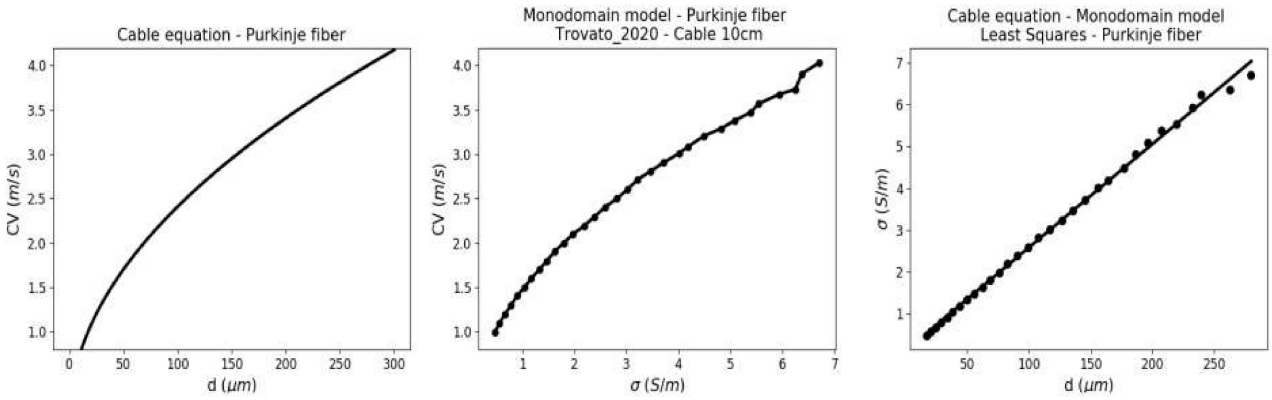


Figure 54 – Correlation between the cable equation and the monodomain model. The values for the conduction velocity in both models are shown alongside the fitted line, obtained with the least-squares method, in which the slope $m = 0.0248$ and the constant $c = 0.0956$ of the linear model.

APPENDIX C – Geometrical concepts and data structures

To properly evaluate the geometrical features of our PN models, the mean and standard deviation of the segment length, branch size, and angle of the bifurcations are computed together with the total number of segments, branches, and bifurcations. A segment in the tree is equivalent to an edge in their corresponding graph representing the PN, while a branch is a set of segments between bifurcations. Moreover, the PN is represented with a graph data structure, allowing each segment node to access its parent and left/right off-springs. In Figure 55, the main geometrical concepts of the *Shocker* algorithm are illustrated together with the data structure utilized to store the nodes and segments of the PN.

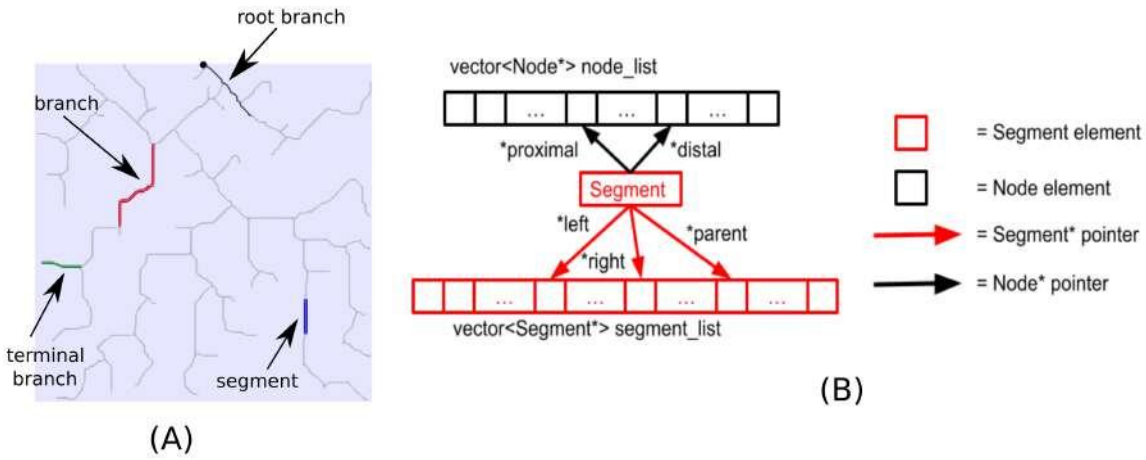


Figure 55 – Illustration showing the different geometrical concepts related to the *Shocker* method alongside the data structure used to store the nodes and segments of the PN. In panel (A), the concept of a segment and branch is highlighted in blue and red, respectively. In addition, the initial root branch of the PN is colored in black, while an example of a terminal branch is presented in green. In panel (B), the data structures used to store a PN are illustrated. There are two arrays, one of the nodes and another of segments which keep track of the current state of the PN. Each segment has access to its parent and left/right off-springs, which are all pointers to the segment structure. Moreover, each segment has two pointers to the two nodes that define the segment. Using these pointers is possible to retrieve the proximal and distal coordinates of any segment.

APPENDIX D – Simplified mesh generation

As mentioned before, the simplified biventricular mesh was constructed by taking the differences between three ellipsoids, one for the LV and RV endocardium and one for the epicardium.

The following equation gives an ellipsoid that is not centered on the domain origin:

$$\frac{(x-c_x)^2}{a^2}+\frac{(y-c_y)^2}{b^2}+\frac{(z-c_z)^2}{c^2}=1, \tag{D.1}$$

where c_x, c_y and c_z are the center coordinates.

The LV surface is built by taking the difference between two ellipsoids. The LV epicardium given by equation (D.3) is subtracted from the LV endocardium given by equation (D.2). Similarly, the RV surface is built by taking the difference between three ellipsoids. The RV epicardium given by equation (D.5) is subtracted from the RV endocardium equation (D.4) and the LV surface previously calculated. Next, the biventricular surface is assembled by joining the LV and RV surfaces and cutting the base plane defined by $x = 0$. Finally, the mesh is translated to the origin and scaled to micrometers. All described operations were done using the *FEniCS* library [2].

$$\frac{x^2}{1.5^2}+\frac{y^2}{0.5^2}+\frac{z^2}{0.5^2}=1, \tag{D.2}$$

$$\frac{x^2}{2^2}+\frac{y^2}{1^2}+\frac{z^2}{1^2}=1, \tag{D.3}$$

$$\frac{x^2}{1.45^2}+\frac{(y-0.5)^2}{1.25^2}+\frac{z^2}{0.75^2}=1, \tag{D.4}$$

$$\frac{x^2}{1.75^2}+\frac{(y-0.5)^2}{1.5^2}+\frac{z^2}{1^2}=1, \tag{D.5}$$

To build the reference PN for the simplified mesh, we use the fractal method from [16] with the parameters given by Table 14. Regarding the parameter description from the fractal method by *Costabal et al.* [16], the Root and Second node coordinates are utilized to give the initial growth direction of the first segment. Fascicle angle and length are related to the generation of the first segment by providing information if there will be a bifurcation after the first segment using a specified angle. If the Fascicle angle is zero, a linear fascicle segment grows with a designated length from the distal position of the first segment. The Repulsive parameter regulates the branch curvature: the larger the repulsion parameter, the more the branches repel each other. The branch angle controls the mean branch bifurcation angles. The number of iterations configures the number of growth iterations. After the networks were generated, the structures were translated to the origin and scaled to micrometers.

Parameter name	**LV**	**RV**
Length of the first branch (mm)	10	15
Mean length of the branches (mm)	4	4
Number of iterations	5	5
Branch angle (degrees)	26	26
Repulsive parameter	0.1	0.1
Root coordinates (mm)	[2.40, 37.15, 26.35]	[2.40, 49.19, 20.02]
Second node coordinates (mm)	[4.84, 36.95, 26.43]	[11.20, 48.81, 21.99]
Fascicles angle (degrees)	74.48	0
Fascicles length (mm)	10	10

Table 14 – Parameters used for the generation of the reference PN of the Simplified mesh.

REFERENCES

[1] Akhtar, M., Gilbert, C., Wolf, F., Schmidt, D.: Reentry within the His-Purkinje system. elucidation of reentrant circuit using right bundle branch and his bundle recordings. Circulation **58**(2), 295–304 (1978)

[2] Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The FEniCS project version 1.5. Archive of Numerical Software **3**(100) (2015)

[3] Aslanidi, O.V., Stewart, P., Boyett, M.R., Zhang, H.: Optimal velocity and safety of discontinuous conduction through the heterogeneous Purkinje-ventricular junction. Biophysical journal **97**(1), 20–39 (2009)

[4] Barber, F., García-Fernández, I., Lozano, M., Sebastian, R.: Automatic estimation of Purkinje-Myocardial junction hot-spots from noisy endocardial samples: A simulation study. International journal for numerical methods in biomedical engineering **34**(7), e2988 (2018)

[5] Bayer, J.D., Blake, R.C., Plank, G., Trayanova, N.A.: A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. Annals of biomedical engineering **40**(10), 2243–2254 (2012)

[6] Behradfar, E., Nygren, A., Vigmond, E.J.: The role of Purkinje-myocardial coupling during ventricular arrhythmia: a modeling study. PLoS One **9**(2), e88,000 (2014)

[7] Bishop, M.J., Plank, G.: Bidomain ECG simulations using an augmented monodomain model for the cardiac source. IEEE transactions on biomedical engineering **58**(8), 2297–2307 (2011)

[8] Bishop, M.J., Plank, G.: Representing cardiac bidomain bath-loading effects by an augmented monodomain approach: application to complex ventricular models. IEEE transactions on biomedical engineering **58**(4), 1066–1075 (2011)

[9] Bordas, R., Gillow, K., Lou, Q., Efimov, I., Gavaghan, D., Kohl, P., Grau, V., Rodriguez, B.: Rabbit-specific ventricular model of cardiac electrophysiological function including specialized conduction system. Progress in biophysics and molecular biology **107**(1), 90–100 (2011)

[10] Boyden, P.A., Hirose, M., Dun, W.: Cardiac Purkinje cells. Heart Rhythm **7**(1), 127–135 (2010)

[11] Boyle, P.M., Massé, S., Nanthakumar, K., Vigmond, E.J.: Transmural IK (ATP) heterogeneity as a determinant of activation rate gradient during early ventricular fibrillation: mechanistic insights from rabbit ventricular models. Heart Rhythm **10**(11), 1710–1717 (2013)

[12] Camps, J., Lawson, B., Drovandi, C., Minchole, A., Wang, Z.J., Grau, V., Burrage, K., Rodriguez, B.: Inference of ventricular activation properties from non-invasive electrocardiography. arXiv preprint arXiv:2010.15214 (2020)

[13] Cárdenes, R., Sebastian, R., Soto-Iglesias, D., Berruezo, A., Camara, O.: Estimation of Purkinje trees from electro-anatomical mapping of the left ventricle using minimal cost geodesics. Medical image analysis **24**(1), 52–62 (2015)

[14] Cooper, L.L., Odening, K.E., Hwang, M.S., Chaves, L., Schofield, L., Taylor, C.A., Gemignani, A.S., Mitchell, G.F., Forder, J.R., Choi, B.R., et al.: Electromechanical and structural alterations in the aging rabbit heart and aorta. American Journal of Physiology-Heart and Circulatory Physiology **302**(8), H1625–H1635 (2012)

[15] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press (2022)

[16] Costabal, F.S., Hurtado, D.E., Kuhl, E.: Generating Purkinje networks in the human heart. Journal of biomechanics **49**(12), 2455–2465 (2016)

[17] Dos Santos, R.W., Otaviano Campos, F., Neumann Ciuffo, L., Nygren, A., Giles, W., Koch, H.: ATX-II effects on the apparent location of M cells in a computational model of a human left ventricular wedge. Journal of cardiovascular electrophysiology **17**, S86–S95 (2006)

[18] Durrer, D., Van Dam, R.T., Freud, G., Janse, M., Meijler, F., Arzbaecher, R.: Total excitation of the isolated human heart. Circulation **41**(6), 899–912 (1970)

[19] Fozzard, H.A.: Membrane capacity of the cardiac Purkinje fibre. The Journal of physiology **182**(2), 255 (1966)

[20] Gepstein, L., Hayam, G., Ben-Haim, S.A.: A novel method for nonfluoroscopic catheter-based electroanatomical mapping of the heart: in vitro and in vivo accuracy results. Circulation **95**(6), 1611–1622 (1997)

[21] Gillette, K., Gsell, M.A., Bouyssier, J., Prassl, A.J., Neic, A., Vigmond, E.J., Plank, G.: Automated framework for the inclusion of a His–Purkinje system in cardiac digital twins of ventricular electrophysiology. Ann Biomed Eng pp. 1–11 (2021)

[22] Gomes, J.M., Alvarenga, A., Campos, R.S., Rocha, B.M., da Silva, A.P.C., dos Santos, R.W.: Uniformization method for solving cardiac electrophysiology models based on the Markov-chain formulation. IEEE Transactions on Biomedical Engineering **62**(2), 600–608 (2014)

[23] Haissaguerre, M., Cheniti, G., Escande, W., Zhao, A., Hocini, M., Bernus, O.: Idiopathic ventricular fibrillation with repetitive activity inducible within the distal Purkinje system. Heart rhythm **16**(8), 1268–1272 (2019)

[24] Haissaguerre, M., Cheniti, G., Hocini, M., Sacher, F., Ramirez, F.D., Cochet, H., Bear, L., Tixier, R., Duchateau, J., Walton, R., et al.: Purkinje network and myocardial substrate at the onset of human ventricular fibrillation: implications for catheter ablation. European Heart Journal (2022)

[25] Haïssaguerre, M., Shoda, M., Jaïs, P., Nogami, A., Shah, D.C., Kautzner, J., Arentz, T., Kalushe, D., Lamaison, D., Griffith, M., et al.: Mapping and ablation of idiopathic ventricular fibrillation. Circulation **106**(8), 962–967 (2002)

[26] Haissaguerre, M., Vigmond, E., Stuyvers, B., Hocini, M., Bernus, O.: Ventricular arrhythmias and the His-Purkinje system. Nature Reviews Cardiology **13**(3), 155 (2016)

[27] Han, W., Zhang, L., Schram, G., Nattel, S.: Properties of potassium currents in Purkinje cells of failing human hearts. American Journal of Physiology-Heart and Circulatory Physiology **283**(6), H2495–H2503 (2002)

[28] Henriquez, C.S.: Simulating the electrical behavior of cardiac tissue using the bidomain model. Critical Reviews in Biomedical Engineering **21**(1), 1–77 (1993)

[29] Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3d shapes. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 203–212 (2001)

[30] Hodgkin, A.L., Huxley, A.F.: Action potentials recorded from inside a nerve fibre. Nature **144**(3651), 710–711 (1939)

[31] Ijiri, T., Ashihara, T., Yamaguchi, T., Takayama, K., Igarashi, T., Shimada, T., Namba, T., Haraguchi, R., Nakazawa, K.: A procedural method for modeling the Purkinje fibers of the heart. The journal of physiological sciences pp. 0810170,079–0810170,079 (2008)

[32] Imanishi, R., Seto, S., Ichimaru, S., Nakashima, E., Yano, K., Akahoshi, M.: Prognostic significance of incident complete left bundle branch block observed over a 40-year period. The American journal of cardiology **98**(5), 644–648 (2006)

[33] Iyer, V., Mazhari, R., Winslow, R.L.: A computational model of the human left-ventricular epicardial myocyte. Biophysical journal **87**(3), 1507–1525 (2004)

[34] Katz, A.M.: Physiology of the Heart. Lippincott Williams & Wilkins (2010)

[35] Keener, J.P., Sneyd, J.: Mathematical physiology, vol. 1. Springer (1998)

[36] Kienzle, M.G., Tan, R., Ramza, B., Young, M., Joyner, R.: Alterations in endocardial activation of the canine papillary muscle early and late after myocardial infarction. Circulation **76**(4), 860–874 (1987)

[37] Klabunde, R.: Cardiovascular physiology concepts. Lippincott Williams & Wilkins (2011)

[38] Krishnamoorthi, S., Perotti, L.E., Borgstrom, N.P., Ajijola, O.A., Frid, A., Ponnaluri, A.V., Weiss, J.N., Qu, Z., Klug, W.S., Ennis, D.B., et al.: Simulation methods and validation criteria for modeling cardiac ventricular electrophysiology. PloS one **9**(12), e114,494 (2014)

[39] LeVeque, R.J.: Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM (2007)

[40] Li, J., Zhang, H., Boyett, M.: Numerical analysis of conduction of the action potential across the Purkinje fibre-ventricular muscle junction. In: 2016 Computing in Cardiology Conference (CinC), pp. 265–268. IEEE (2016)

[41] Li, P., Rudy, Y.: A model of canine purkinje cell electrophysiology and Ca2+ cycling: rate dependence, triggered activity, and comparison to ventricular myocytes. Circulation research **109**(1), 71–79 (2011)

[42] Liu, B.R., Cherry, E.M.: Image-based structural modeling of the cardiac Purkinje network. BioMed research international **2015** (2015)

[43] Lopez-Perez, A., Sebastian, R., Izquierdo, M., Ruiz, R., Bishop, M., Ferrero, J.M.: Personalized cardiac computational models: from clinical data to simulation of infarct-related ventricular tachycardia. Frontiers in physiology **10**, 580 (2019)

[44] MacLachlan, M.C., Sundnes, J., Spiteri, R.J.: A comparison of non-standard solvers for odes describing cellular reactions in the heart. Computer Methods in Biomechanics and Biomedical Engineering **10**(5), 317–326 (2007)

[45] Marsh, M.E., Ziaratgahi, S.T., Spiteri, R.J.: The secrets to the success of the Rush-Larsen method and its generalizations. IEEE transactions on biomedical engineering **59**(9), 2506–2515 (2012)

[46] Martyn, Y.L., Mahaut-Smith, P., Varghese, A., Huang, C.L.H., Kemp, P.R., Vandenberg, J.I.: Effects of premature stimulation on HERG K+ channels. The Journal of Physiology **537**(3), 843–851 (2001)

[47] Mendez, C., Mueller, W.J., Merideth, J., Moe, G.K.: Interaction of transmembrane potentials in canine Purkinje fibers and at Purkinje fiber-muscle junctions. Circulation research **24**(3), 361–372 (1969)

[48] Miralles, F.B.: Inverse estimation of the cardiac Purkinje system from electroanatomical maps. Ph.D. thesis, Universitat de València (2020)

[49] Noble, D.: A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pacemaker potentials. The Journal of Physiology **160**(2), 317–352 (1962)

[50] O'Hara, T., Virág, L., Varró, A., Rudy, Y.: Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation. PLoS Comput Biol **7**(5), e1002,061 (2011)

[51] Oliveira, R.S., Rocha, B.M., Amorim, R.M., Campos, F.O., Meira, W., Toledo, E.M., dos Santos, R.W.: Comparing CUDA, OpenCL and OpenGL implementations of the cardiac monodomain equations. In: International Conference on Parallel Processing and Applied Mathematics, pp. 111–120. Springer (2011)

[52] Ono, N., Yamaguchi, T., Ishikawa, H., Arakawa, M., Takahashi, N., Saikawa, T., Shimada, T.: Morphological varieties of the Purkinje fiber network in mammalian hearts, as revealed by light and electron microscopy. Archives of histology and cytology **72**(3), 139–149 (2009)

[53] Palamara, S., Vergara, C., Catanzariti, D., Faggiano, E., Pangrazzi, C., Centonze, M., Nobile, F., Maines, M., Quarteroni, A.: Computational generation of the Purkinje network driven by clinical measurements: the case of pathological propagations. International journal for numerical methods in biomedical engineering **30**(12), 1558–1577 (2014)

[54] Park, D.S., Fishman, G.I.: Development and function of the cardiac conduction system in health and disease. Journal of cardiovascular development and disease **4**(2), 7 (2017)

[55] Potse, M., Dubé, B., Richer, J., Vinet, A., Gulrajani, R.M.: A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart. IEEE Transactions on Biomedical Engineering **53**(12), 2425–2435 (2006)

[56] Pressler, M.L.: Cable analysis in quiescent and active sheep Purkinje fibres. The Journal of Physiology **352**(1), 739–757 (1984)

[57] Rabin, J., Peyré, G., Cohen, L.D.: Geodesic shape retrieval via optimal mass transport. In: European Conference on Computer Vision, pp. 771–784. Springer (2010)

[58] Rush, S., Larsen, H.: A practical algorithm for solving dynamic membrane equations. IEEE Transactions on Biomedical Engineering (4), 389–392 (1978)

[59] Sachetto Oliveira, R., Martins Rocha, B., Burgarelli, D., Meira Jr, W., Constantinides, C., Weber dos Santos, R.: Performance evaluation of GPU parallelization, space-time adaptive algorithms, and their combination for simulating cardiac electrophysiology. International journal for numerical methods in biomedical engineering **34**(2), e2913 (2018)

[60] Sampson, K., Iyer, V., Marks, A., Kass, R.: A computational model of Purkinje fibre single cell electrophysiology: implications for the long QT syndrome. The Journal of physiology **588**(14), 2643–2655 (2010)

[61] Schoenberg, M., Dominguez, G., Fozzard, H.A.: Effect of diameter on membrane capacity and conductance of sheep cardiac Purkinje fibers. The Journal of general physiology **65**(4), 441–458 (1975)

[62] Schreiner, W., Buxbaum, P.F.: Computer-optimization of vascular trees. IEEE Transactions on Biomedical Engineering **40**(5), 482–491 (1993)

[63] Schroeder, W., Martin, K.M., Lorensen, W.E.: The visualization toolkit an object-oriented approach to 3D graphics. Prentice-Hall, Inc. (1998)

[64] Schwartz, W.R., de Rezende, P.J., Pedrini, H.: Faster approximations of shortest geodesic paths on polyhedra through adaptive priority queue. In: VISAPP (1), pp. 371–378 (2015)

[65] Sebastian, R., Zimmerman, V., Romero, D., Sanchez-Quintana, D., Frangi, A.F.: Characterization and modeling of the peripheral cardiac conduction system. IEEE transactions on medical imaging **32**(1), 45–55 (2012)

[66] Spiteri, R.J., Dean, R.C.: Stiffness analysis of cardiac electrophysiological models. Annals of biomedical engineering **38**(12), 3592–3604 (2010)

[67] Stephenson, R.S., Atkinson, A., Kottas, P., Perde, F., Jafarzadeh, F., Bateman, M., Iaizzo, P.A., Zhao, J., Zhang, H., Anderson, R.H., et al.: High resolution 3-dimensional imaging of the human cardiac conduction system from microanatomy to mathematical modeling. Scientific reports **7**(1), 1–13 (2017)

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

[69] Sundnes, J., Artebrant, R., Skavhaug, O., Tveito, A.: A second-order algorithm for solving dynamic cell membrane equations. IEEE Transactions on Biomedical Engineering **56**(10), 2546–2548 (2009)

[70] Sundnes, J., Nielsen, B.F., Mardal, K.A., Cai, X., Lines, G.T., Tveito, A.: On the computational complexity of the bidomain and the monodomain models of electrophysiology. Annals of biomedical engineering **34**(7), 1088–1097 (2006)

[71] Ten Tusscher, K., Panfilov, A.V.: Modelling of the ventricular conduction system. Progress in biophysics and molecular biology **96**(1-3), 152–170 (2008)

[72] Tomek, J., Bueno-Orovio, A., Passini, E., Zhou, X., Minchole, A., Britton, O., Bartolucci, C., Severi, S., Shrier, A., Virag, L., et al.: Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block. Elife **8**, e48,890 (2019)

[73] Tomek, J., Bueno-Orovio, A., Rodriguez, B.: ToR-ORd-dynCl: an update of the ToR-ORd model of human ventricular cardiomyocyte with dynamic intracellular chloride. BioRxiv (2020)

[74] Trovato, C., Passini, E., Nagy, N., Varró, A., Abi-Gerges, N., Severi, S., Rodriguez, B.: Human Purkinje in silico model enables mechanistic investigations into automaticity and pro-arrhythmic abnormalities. Journal of Molecular and Cellular Cardiology (2020)

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

[76] Ulysses, J.N., Berg, L.A., Cherry, E.M., Liu, B.R., dos Santos, R.W., de Barros, B.G., Rocha, B.M., de Queiroz, R.A.: An optimization-based algorithm for the construction of cardiac Purkinje network models. IEEE Transactions on Biomedical Engineering **65**(12), 2760–2768 (2018)

[77] Vergara, C., Palamara, S., Catanzariti, D., Nobile, F., Faggiano, E., Pangrazzi, C., Centonze, M., Maines, M., Quarteroni, A., Vergara, G.: Patient-specific generation of the Purkinje network driven by clinical measurements of a normal propagation. Medical & biological engineering & computing **52**(10), 813–826 (2014)

[78] Walton, R.D., Martinez, M.E., Bishop, M.J., Hocini, M., Haïssaguerre, M., Plank, G., Bernus, O., Vigmond, E.J.: Influence of the Purkinje-muscle junction on transmural repolarization heterogeneity. Cardiovascular research **103**(4), 629–640 (2014)

[79] Wiedmann, R.T., Tan, R.C., Joyner, R.W.: Discontinuous conduction at Purkinje-ventricular muscle junction. American Journal of Physiology-Heart and Circulatory Physiology **271**(4), H1507–H1516 (1996)

[80] Zigelman, G., Kimmel, R., Kiryati, N.: Texture mapping using surface flattening via multidimensional scaling. IEEE Transactions on Visualization and Computer Graphics **8**(2), 198–207 (2002)