

Cardiac Electrophysiology using LS-DYNA®

Pierre L'Eplattenier¹, Sarah Bateau-Meyer¹, Dave Benson¹, Vikas Kaul², Carl Schu²
Mark Palmer², Darrell Swenson², Joshua Blauer²

¹Livermore Software Technology Corporation, Livermore, California 94551

²Medtronic plc, Minneapolis, Minnesota 55432

Abstract

Heart disease is among the leading causes of death in the Western world; hence, a deeper understanding of cardiac functioning will provide important insights for engineers and clinicians in treating cardiac pathologies. However, the heart also offers a significant set of unique challenges due to its extraordinary complexity. In this respect, some recent efforts have been made to be able to model the multiphysics of the heart using LS-DYNA.

The model starts with electrophysiology (EP) which simulates the propagation of the cell transmembrane potential in the heart. This electrical potential triggers the onset of cardiac muscle contraction, which then results in the pumping of the blood to the various organs in the body. The EP/mechanical model can be coupled with a Fluid and Structure Interaction (FSI) model to not only study the clinically relevant blood flow parameters as well as valves or cardiac devices. This paper concentrates on the EP part of the model. Other papers in this conference will present the mechanical and FSI parts.

Different propagation models, called "mono-domain" or "bi-domain", which couple the diffusion of the potential along the walls of the heart with ionic equations describing the exchanges between the inner and the outer parts of the cells have been implemented. These models were first benchmarked against published results obtained from other EP research codes on a simple cuboid heart tissue model. Other simulations were then performed on more realistic geometries. Since the potential diffusion is highly orthotropic, with much larger diffusion coefficients along the fibers of the tissue than transversally, it is important to correctly model these fibers, which creates models with very large numbers of elements (several tens to hundreds of millions of elements). We thus implemented capabilities to be able to handle such large-scale models. Some EP models will be presented and first results will be shown.

1-Introduction

A cardiac computational model can give biomedical researchers an additional source of information to understand how the heart works. Simulation can be the base of theoretical studies into the mechanisms of cardiac pathologies, provide diagnostic value or can be used to assist in therapy planning. The goal of LSTC is to be able to simulate the pumping heart, with a coupling scheme between the EP equations describing the propagation of the transmembrane potential, the mechanical deformations triggered by this electrical potential, and the blood flow in this pumping heart.

Section 2 will present the EP models introduced in LS-DYNA, section 3 will show some benchmarks between LS-DYNA and other EP codes, and in section 4, we will show an example of spiral wave development in a ventricle.

2-Presentation of the model

The wall of the heart has three layers: epicardium, myocardium and endocardium. The endocardium and epicardium are thin layers consisting primarily of collagen and elastic tissue. In the middle layer, the myocardium, the cells that constitute the muscle show electrical excitability. These specialized cells, called myocytes, are organized into parallel cardiac fibers giving the muscle the striated appearance. The fibers form sheets which are connected by a collagenous network [1].

A cardiac cell (myocyte) is typically 10 to 20 μm in diameter and 80 to 125 μm in length. The cell membrane acts as an electrical insulator and contains ion channels which transport electrical current by diffusion. The potential difference across the membrane is called the transmembrane potential. Initially, a cardiac cell is at rest, with a potential difference across the membrane. The potential inside the cell is negative compared to the external, with a potential difference around 80mV. If the membrane potential rises to a certain threshold value (close to 40 mV) a rapid process occurs, during which different ions, mainly Na^+ , K^+ , and Ca^{2+} , are exchanged between the inner and the outer part of the cell, creating a fast depolarization, an early repolarization, a plateau and a final repolarization. The complete cycle of depolarization and repolarization lasts around 300 ms and is called “action potential”. It is shown in Figure (1). This action potential diffuses from cell to cell through a network of gap junctions, creating a wave of depolarization and repolarization through the myocardium [1].

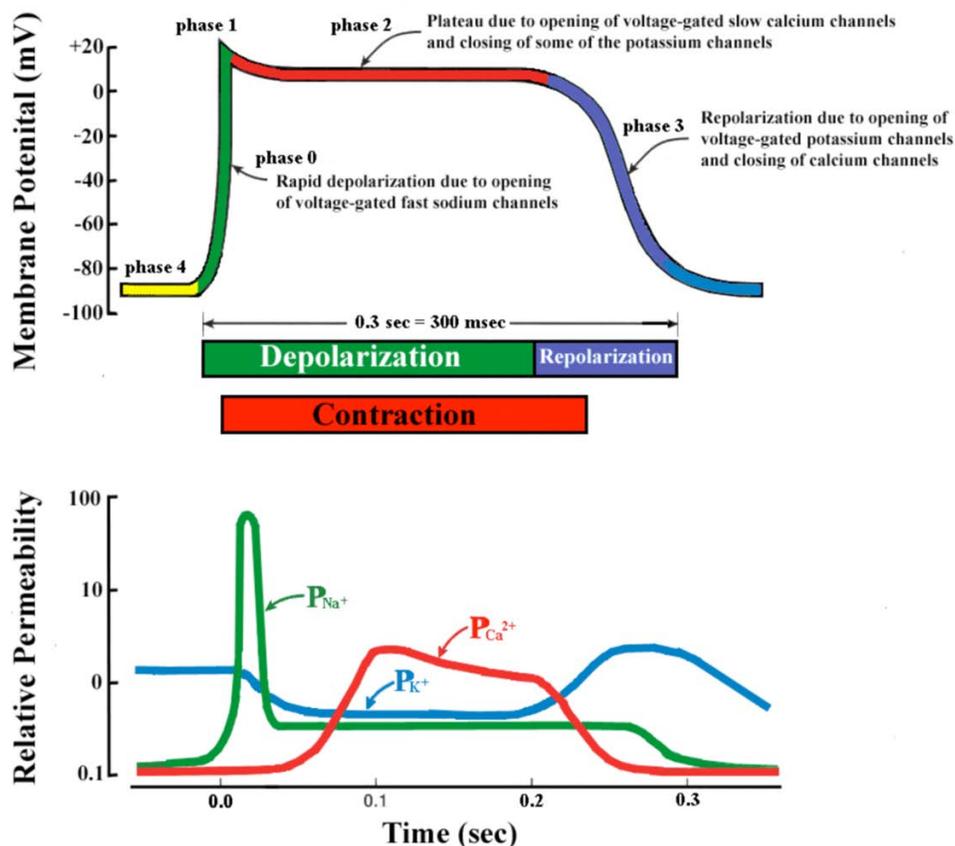


Figure 1: A typical action potential of a ventricular myocyte and the underlying ion currents. The resting membrane potential is approximately ~ 80 mV (phase 4). The rapid depolarization is primarily due to the voltage gated Na^+ current (phase 0), which results in a relatively sharp peak (phase 1) and transitions into the plateau (phase 2) until repolarization (phase 3). Also indicated are the refractory period and timing of the ventricular contraction. Modified from Tortora GJ, Grabowski SR. *Principles of Anatomy and Physiology*, ninth edition. New York: John Wiley & Sons, Inc., 2000

2-1 The bidomain model

Since describing the whole heart, or even part of it like a ventricle, at the cell level would be computationally too expensive, continuous approximations are made, where the inner part of the cells is treated as one continuum “domain” with an inner potential $\phi_i(\vec{x}, t)$, and the outer part as another domain with an external potential $\phi_e(\vec{x}, t)$. Each domain is characterized by a conductivity tensor, called respectively σ_i and σ_e . These tensors are usually highly non-isotropic, with factors that can be as high as 5 to 10 between the conductivity along the fibers and the one across the fibers. Therefore, it is very important to correctly model the fiber orientation, which can be consumed from an imaging technique called the diffusion tensor MRI. A transmembrane current with surface density I_m flows between the two domains hence the so called “bi-domain” equations [2]:

$$\nabla \cdot (\sigma_i \nabla \phi_i) = \beta I_m \quad (1)$$

$$\nabla \cdot (\sigma_e \nabla \phi_e) = -\beta I_m \quad (2)$$

where β is the membrane surface to volume ratio.

This transmembrane current density I_m consists of a capacitive part, an ionic part generated by the cell membrane I_{ion} , and an imposed stimulation current density I_{stim} :

$$I_m = C_m \frac{\partial V_m}{\partial t} + I_{ion} + I_{stim} \quad (3)$$

where C_m is the membrane capacity per unit area, and we introduced the transmembrane potential:

$$V_m = \phi_i - \phi_e \quad (4)$$

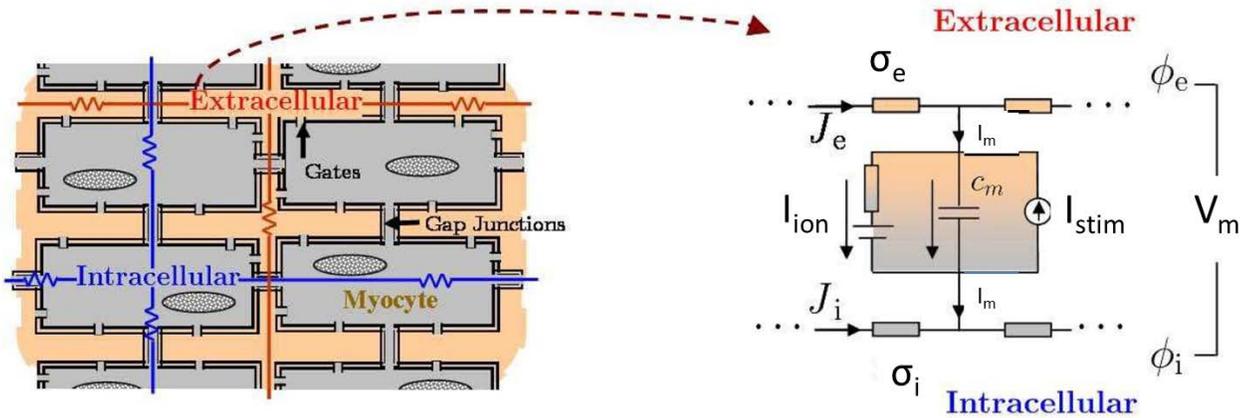


Figure 2: Illustration of the bidomain method (adapted from “Multiscale forward electromagnetic model of uterine contractions during pregnancy”, La Rosa et al. BMC Medical Physics 2012, 12:4.).

Using (3) and (4), we can rewrite equations (1) and (2) in terms of V_m and ϕ_e as:

$$\beta C_m \frac{\partial V_m}{\partial t} + \beta I_{ion}(V_m, u) - \nabla \cdot (\sigma_i \nabla V_m) - \nabla \cdot (\sigma_i \nabla \phi_e) = \beta I_{stim}(\vec{x}, t) \quad (5)$$

$$\nabla \cdot (\sigma_i \nabla V_m) + \nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e) = 0 \quad (6)$$

In equation (5), we wrote $I_{ion}(V_m, u)$, to indicate that the ionic current density depends not only on the transmembrane potential V_m , but also on an extra set of variables that we represent by u . The number of such variables and their time evolution depend on the cell model chosen, which we write, in a general way:

$$\frac{\partial u}{\partial t} = f(u, V_m) \quad (7)$$

These cell models locally describe the exchange of ions through the cell membrane, as schematically shown in Figure 1. Depending upon the question of interest, one can select from a wide class of ionic models, ranging from the FitzHugh-Nagumo model [3][4] with two variables or the Fenton-Karma model with 3 variables [5] to the one discussed in this paper, the ten Tusscher and Panfilov model [6] with 19 variables.

Projecting equations (5) and (6) onto the FEM basis functions, we get:

$$\beta C_m M \frac{\partial V_m}{\partial t} + \beta I_{ion} - S_i V_m - S_i \phi_e = \beta I_{stim} \quad (8)$$

$$S_i V_m - S_{ie} \Phi_e = 0 \quad (9)$$

where

$$M(i, j) = \int_{\Omega} \varphi_i \varphi_j d\Omega \quad (10)$$

Is the mass matrix, and

$$S_i(i, j) = \int_{\Omega} \sigma_i \nabla \varphi_i \cdot \nabla \varphi_j d\Omega \quad (11)$$

and

$$S_{ie}(i, j) = \int_{\Omega} (\sigma_i + \sigma_e) \nabla \varphi_i \cdot \nabla \varphi_j d\Omega \quad (12)$$

are diffusion stiffness matrices corresponding to different conductivities.

In order to solve the coupled diffusion equations (8)-(9) with the ionic one (7), we use a so called ‘‘Spiteri-Ziaratgahi’’ operator splitting [7] where the advance from time t to time $t+1$ reads:

$$u(t+1) = u(t) + dt f(u(t), V_m(t), t) \quad (13)$$

$$\begin{bmatrix} \frac{\beta C_m}{dt} M + S_i & S_i \\ S_i & S_{ie} \end{bmatrix} \cdot \begin{bmatrix} V_m(t+1) \\ \phi_e(t+1) \end{bmatrix} = \begin{bmatrix} \frac{\beta C_m}{dt} M V_m(t) - \beta M I_{ion}(u(t+1), V_m(t), t) + \beta M I_{stim} \\ 0 \end{bmatrix} \quad (14)$$

2-2 The monodomain model

The monodomain model makes the extra hypothesis that the inner and outer conductivity tensors are proportional: $\sigma_e = \lambda \sigma_i$. We introduce a mean conductivity [2]:

$$\sigma = \frac{\sigma_i \sigma_e}{\sigma_i + \sigma_e} \quad (15)$$

or

$$\sigma_i = (1 + \lambda) \sigma \quad (16)$$

$$\sigma_e = \frac{1 + \lambda}{\lambda} \sigma \quad (17)$$

Equation (6) gives:

$$\nabla \cdot (\sigma_i \nabla \phi_e) = - \frac{\lambda}{1 + \lambda} \nabla \cdot (\sigma_i \nabla V_m) \quad (18)$$

which gives, when using it in (5), an equation on V_m only:

$$\beta C_m \frac{\partial V_m}{\partial t} + \beta I_{ion}(V_m, u) - \nabla \cdot (\sigma \nabla V_m) = \beta I_{stim}(\vec{x}, t) \quad (19)$$

This is the monodomain equation.

When projecting equation (19) onto the FEM basis functions, we get:

$$\beta C_m M \frac{\partial V}{\partial t} + \beta I_{ion} - S V = \beta I_{stim} \quad (20)$$

with

$$S(i, j) = \int_{\Omega} \sigma \nabla \varphi_i \cdot \nabla \varphi_j d\Omega \quad (21)$$

And M is defined by (10).

3-Benchmark against other EP codes

3-1 Presentation of the benchmark

The benchmark we decided to perform is described in detail in [8], where 11 codes were compared on the same diffusion test case. Their main characteristics are presented in table 1.

Index	Code/developer	Home institute	Numerical method	Element type
A	CHASTE	University of Oxford	FEM	hexahedra
B	CARP	University of Graz	FEM	tetrahedra
C	Sander Land	University of Oxford	FEM	hexahedra
D	Richard Clayton	University of Sheffield	FDM	regular grid
E	EMOS	University of Zaragoza	FEM	hexahedra
F	OpenCMISS	University of Auckland	FEM	hexadreja
G	Alan Garny	University of Oxford	FDM	regular grid
H	FEniCS/PyCC	Simula	FEM	tetrahedra
I	acCELLerate	Karlsruhe Inst. Tech.	FDM	regular grid
J	Alan Benson	University of Leeds	FDM	regular grid
K	E.M Cherry	Rochester Inst. Tech.	FDM	regular grid

Table 1. Code index, name and developers. Abbreviations: FDM, finite difference method; FEM, finite element method. See more details in [8].

The test case consists in a 20mm x 7mm x 3mm cuboid part of tissue which is stimulated by injecting a current at one corner. The diffusion of the transmembrane potential through the tissue is studied.

Figure (3) represents the simulation domain. The cardiac electrical activation or stimulus, is applied within a 1.5 mm cube placed at the bottom left corner of the domain around point P₁, at time t=0. As mentioned, the ten Tusscher & Panfilov ionic ventricular cell model [6] is used. The conductivity tensor is anisotropic with a higher conductivity in the y-direction where the cuboid is the longest (20mm), in units of S.m⁻¹:

$$\sigma = \begin{bmatrix} \sigma_T = 0.017606 & 0 & 0 \\ 0 & \sigma_L = 0.133418 & 0 \\ 0 & 0 & \sigma_T = 0.017606 \end{bmatrix} \quad (22)$$

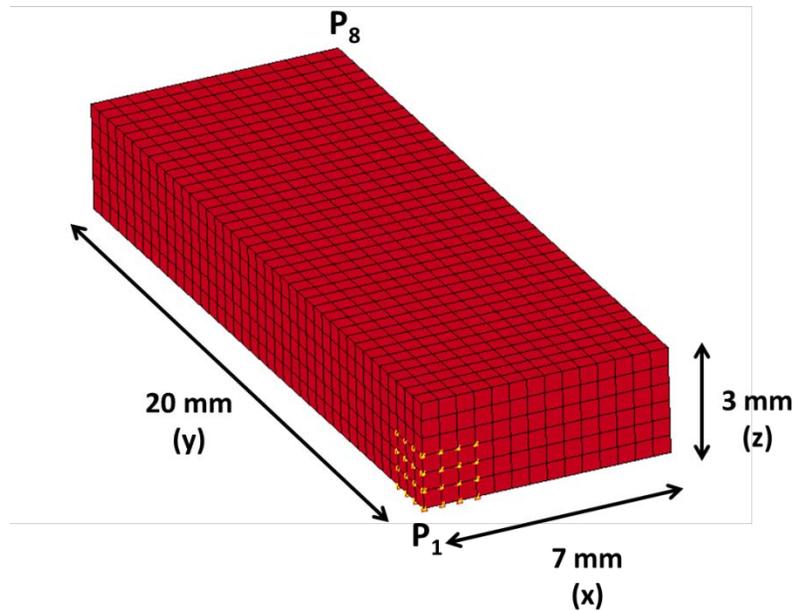


Figure 3: Simulation domain for the benchmark: the stimulation is specified at point P_1 (see yellow node set), and propagates towards the opposite corner at point P_8

The benchmark proposes to solve the problem at three spatial resolutions ($dx = 0.5, 0.2$ and 0.1 mm) and with three different time-steps ($dt = 0.05, 0.01$ and 0.005 ms) for a total of 9 simulations. The following presents the simulation results obtained first using the monodomain model and then with the bidomain model.

dt1 (ms)	dt2 (ms)	dt3 (ms)
0.05	0.01	0.005

dx1 (mm)	dx2 (mm)	dx3 (mm)
0.5	0.2	0.1

Table 2: Time-steps and mesh sizes used in the benchmark model

3-2 Monodomain results

In the first set of simulations, we used the explicit Qu-Garfinkel Operator Split integration scheme ([10], [11] and Appendix A). Eight simulations succeeded but the 9th one, with the smallest discretization $dx = 0.1$ mm and the largest time-step $dt = 0.05$ ms has a time-step over the limit of the Courant-Friedrichs-Lewy (CFL) condition, and did not converge. The CFL condition is given by [9] as:

$$dt \leq \frac{\beta C_m dx^2}{2(\sigma_L \sigma_T)} \quad (23)$$

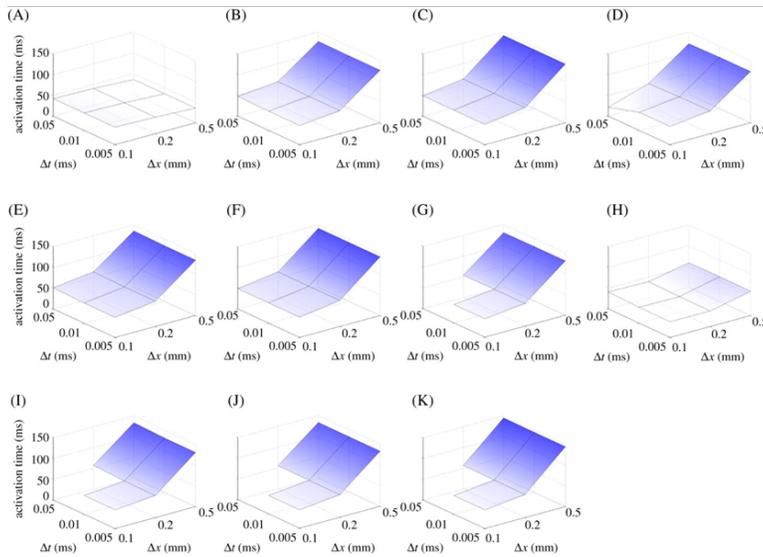
With the smaller discretization $dx = 0.1$ mm the equation gives $dt_{max} = 0.046$ ms, which is inferior to the largest benchmark time-step.

For the second set of simulations, we developed three others integration schemes (Appendices B, C, and D) with the goal of solving the issue of the CFL condition and reducing the computation time. The second integration scheme, explicit first order operator split (Appendix B), did not solve the CFL issue of the case 9, but the two implicit integration schemes (Appendices C and D) succeeded and gave an accurate result.

Figure 4 shows the activation time at node P_8 for all 9 simulations. It is compared to the activation time given by the 11 codes used for benchmark in [8]. According to [8] the actual solution lies between 42.5 and 43 ms as

the eleven codes compared in [8] mostly converge towards such values. One can see that LS-DYNA gives results in this range.

11 benchmark codes



LS-DYNA

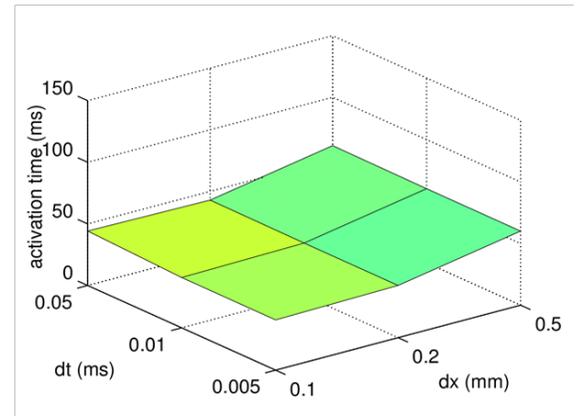
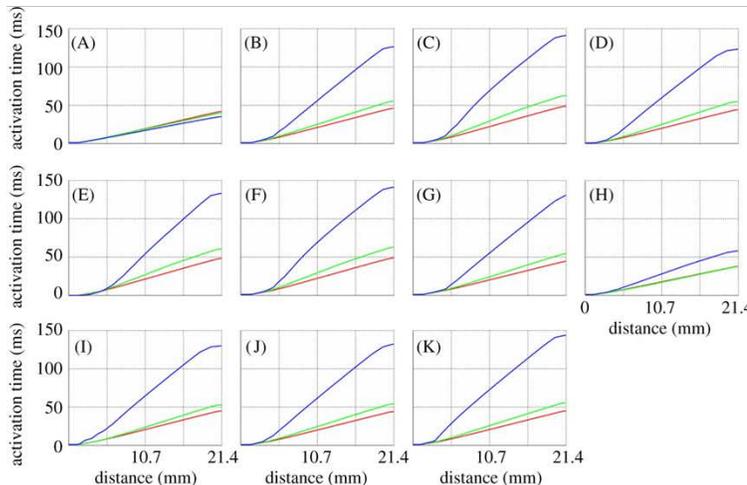


Figure 4: Activation time at P_8 for every combination of spatial and temporal refinement in monodomain (11 benchmark codes on the left, LS-DYNA on the right)

Figure 5 shows the activation time along the line from P_1 to P_8 , compared with the results from the other 11 benchmark codes. One can see that as soon as we have $dx \leq 0.2 \text{ mm}$, LS-DYNA gives very accurate results. Only code (A) in the 11 benchmarks code manages to get an accurate result even for $dx \leq 0.5 \text{ mm}$.

11 benchmark codes



LS-DYNA

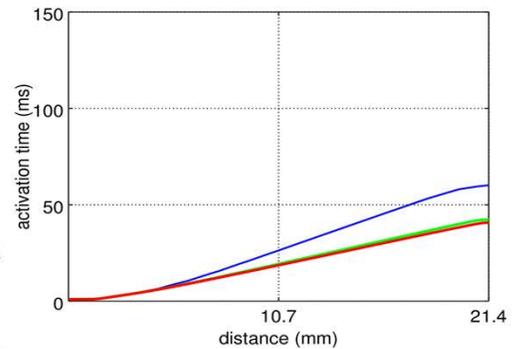


Figure 5: Activation times along the line P_1 - P_8 for solutions with $dt = 0.005 \text{ ms}$ and $dx = 0.5 \text{ mm}$ (blue line), 0.2 mm (green line) and 0.1 mm (red line) in monodomain. The results of the 11 benchmark codes are on the left, and the LS-DYNA results on the right.

Figure 6 shows the activation time propagation on a plane going through P_1 and P_8 . The figure may be compared to [8]-Figure 4.

from dark blue (0 ms) to red (130 ms) with contour bands at 10 ms intervals.

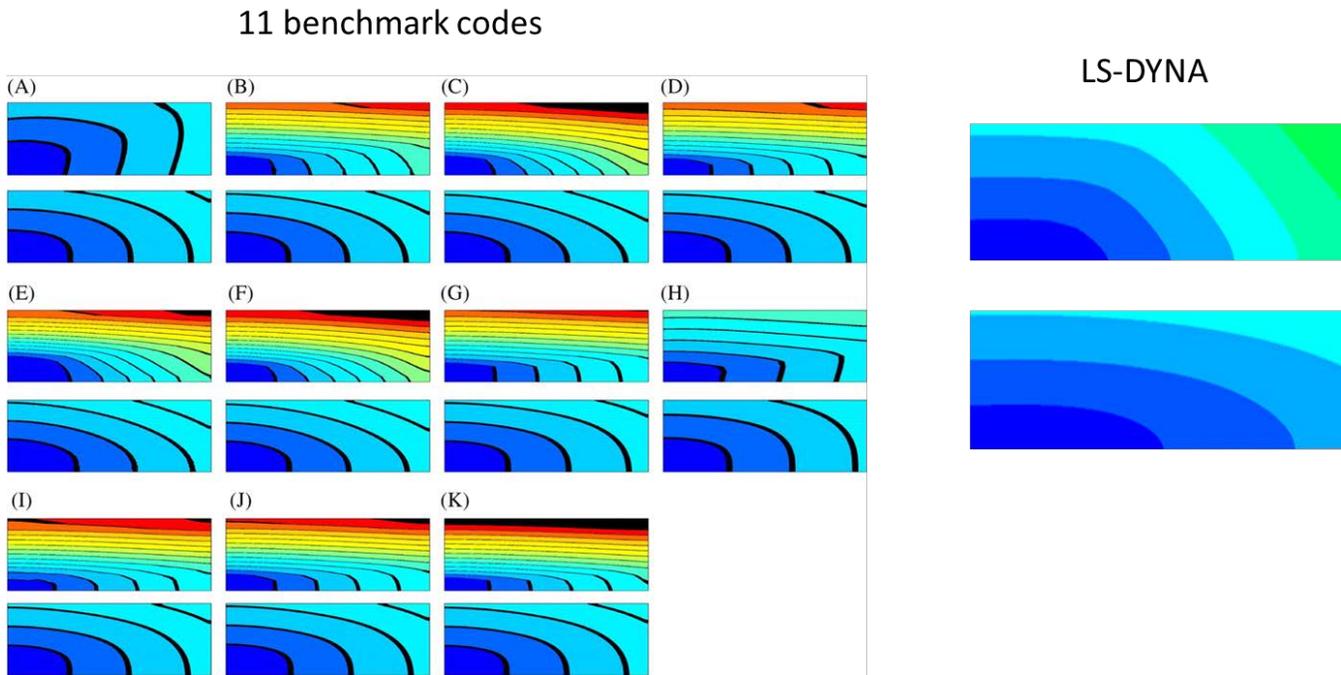


Figure 6. Activation times on a plane containing P_1 and P_8 , for the benchmark codes (left) and LS-DYNA (right). For each code, the upper and lower planes correspond to the solutions with $dx = 0.5\text{mm}$, $dt = 0.005\text{ ms}$ and $dx = 0.1\text{mm}$, $dt = 0.005\text{ ms}$, respectively. The activation times are represented by the color map from dark blue (0 ms) to red (130 ms) with contour bands at 10 ms intervals.

3-3 Bidomain results

The Spiteri-Ziaratgahi Operator Split [7] (Appendix E) is used in this case. We repeated all 9 simulations which were all successful. For internal and external conductivities, we used:

$$\sigma_i = \begin{bmatrix} 0.019 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.019 \end{bmatrix} \quad (24)$$

And

$$\sigma_e = \begin{bmatrix} 0.24 & 0 & 0 \\ 0 & 0.62 & 0 \\ 0 & 0 & 0.24 \end{bmatrix} \quad (25)$$

Note that the monodomain electrical conductivity (22) was taken as an average of the internal and external bidomain conductivities (24) and (25) in the sense of equation (15).

Figures 7 and 8 show the results, which again show activation times very similar to the ones in the benchmark codes.

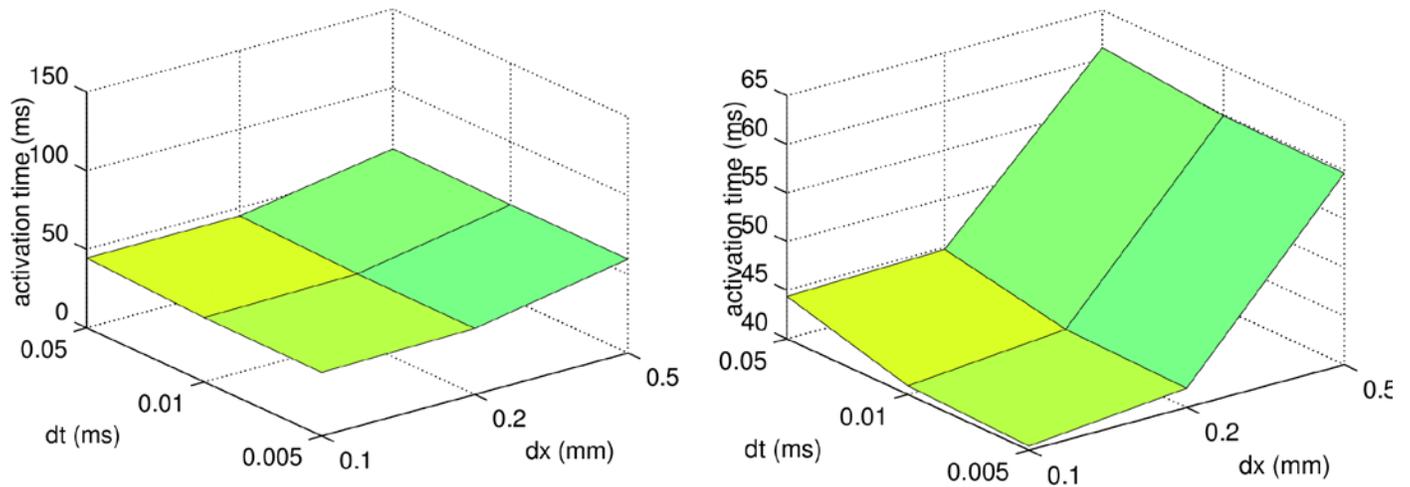


Figure 7: Activation time at P9 for every combination of spatial and temporal refinement in bidomain (same scale as benchmark paper on the left, zoomed scale in the z direction on the right)

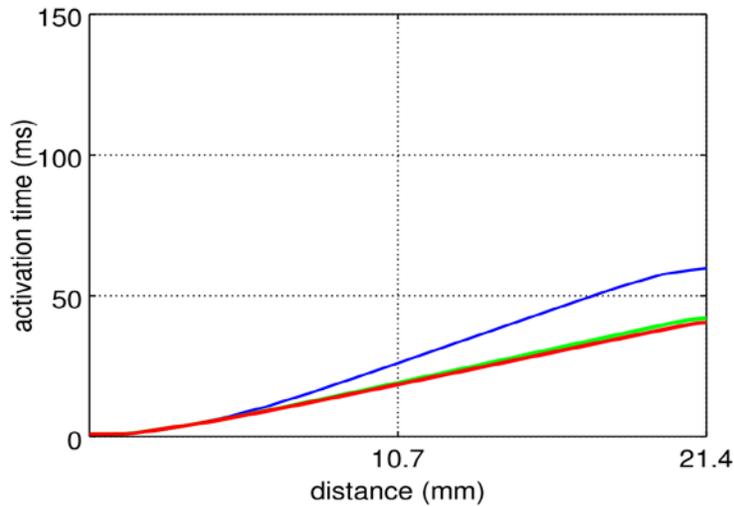


Figure 8: Activation times along line P₁-P₈ for solutions with dt = 0.005 ms and dx = 0.5 mm (blue line), 0.2 mm (green line) and 0.1 mm (red line) in bidomain

3-4 Results summary

The following table gives the activation times at P₈ obtained for the different simulations performed.

		dt1-dx1	dt1-dx2	dt1-dx3	dt2-dx1	dt2-dx2	dt2-dx3	dt3-dx1	dt3-dx2	dt3-dx3
monodomain	A	60.79	42.95	X	60.21	42.42	40.82	60.12	42.36	40.75
	B	60.54		X				60.09		
	C	61.49		44.21				60.19		
	D	61.94		46.09				60.22		
bidomain	E	61.16	44.8	44.36	59.89	42.3	40.89	59.7	41.97	40.42

Table 3: Activation times at P₈ (in ms) for each simulation. The X indicates that the simulation did not converge, and an empty block that the simulation was not done.

3-4 Large models

The geometry presented in [8] was also used to generate much larger meshes than the one corresponding to $dx=dx_3$, which has 420,000 elements. As already mentioned, the electrical conductivity is much larger along the fibers than across them, so it will be important to have the fiber orientation represented accurately. Also, it is important to capture the first rapid depolarization of the action potential (see Figure 1) in the wave propagation, which necessitates extremely fine meshes. We thus increased the mesh density step by step, and after some changes in the code to better handle the initial MPP decomposition of the model in terms of memory and computation time, a case with 100 million elements on 8 nodes of one of our machine that has 96 GB and 12 cores per node was successfully run to completion. The run was done using the monodomain method with an implicit first order operator-split (Appendix B) and a Pre-Conditioned Gradient (PCG) method, with a time-step $dt=0.01ms$. It gave the same results as the case $dt=3-dx_3$ above.

4-Spiral waves and arrhythmia in a ventricle

The propagation of electrical waves through cardiac tissue is a very important phenomenon to study since those waves activate the mechanisms for cardiac contraction, responsible to pump blood to the body. In a healthy heart, an electrical wave of action potential propagates in a regular way through the various regions of the heart. Figure 9, which corresponds to a monodomain simulation on a 3D mesh with about 160,000 elements with the size and the shape of a ventricle, and where a stimulus was applied at the bottom, shows such a regular propagation.

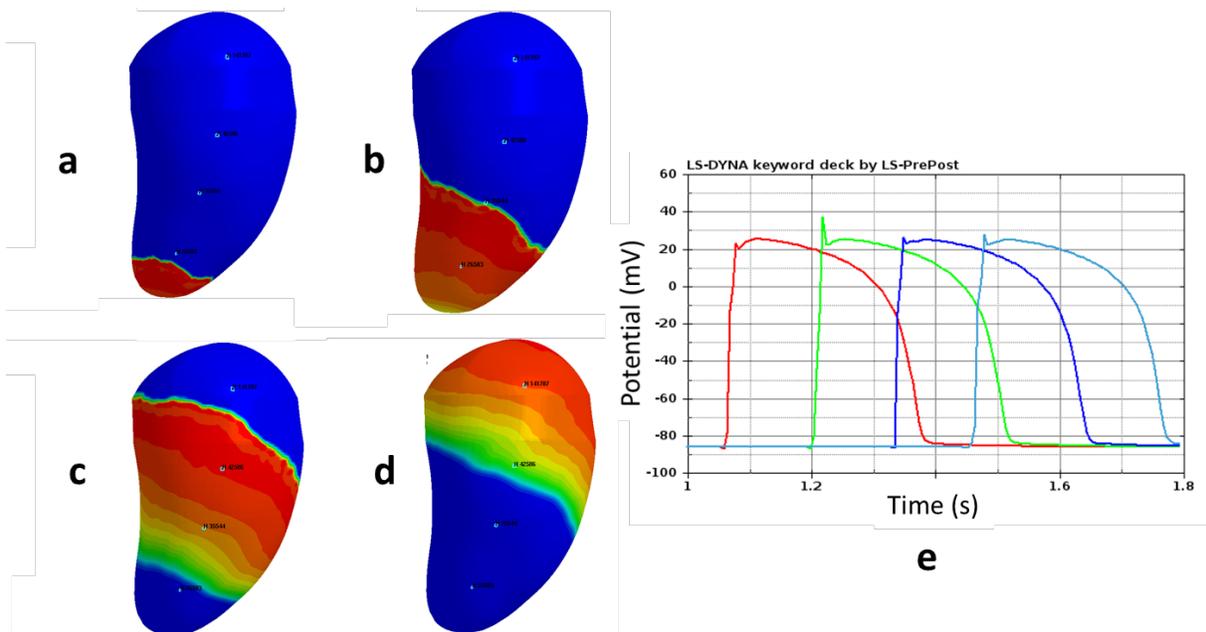


Figure 9: Propagation of the transmembrane potential in a healthy ventricle. The plots on the left show the fringe components of the potential at time 1050 ms (a), 1200 ms (b), 1400 ms (c), and 1600 ms (d). e shows time evolution of the potential at the 4 elements marked on the plots.

One of the proposed mechanisms involved in the development of certain type of arrhythmias are spiral waves, which are symptomatic of functional reentry [6]. Spiral waves are self-sustained waves of excitation that rotate freely or around an obstacle, reactivating the same area of tissue at a higher frequency than normal, increasing the normal heartbeat rate (tachycardia). In the worst-case scenario, a spiral wave might break up into smaller

spiral waves giving uncoordinated contractions of the heart in a phenomenon known as fibrillation. When this phenomenon occurs in the ventricles, the heart quivers and loses capacity to pump blood to the body leading to immediate cardiac arrest. Figure 10 shows the development of such a spiral wave on the same model as the one used for Figure 9. In this case, the spiral was triggered by adding a second stimulus at a certain location of the ventricle just after the normal wave had passed through this location. One can clearly see the spiral on the fringe components, and the reactivations of the pulse at higher frequencies on the potential vs time plot, where the chaotic dynamics can also be seen.

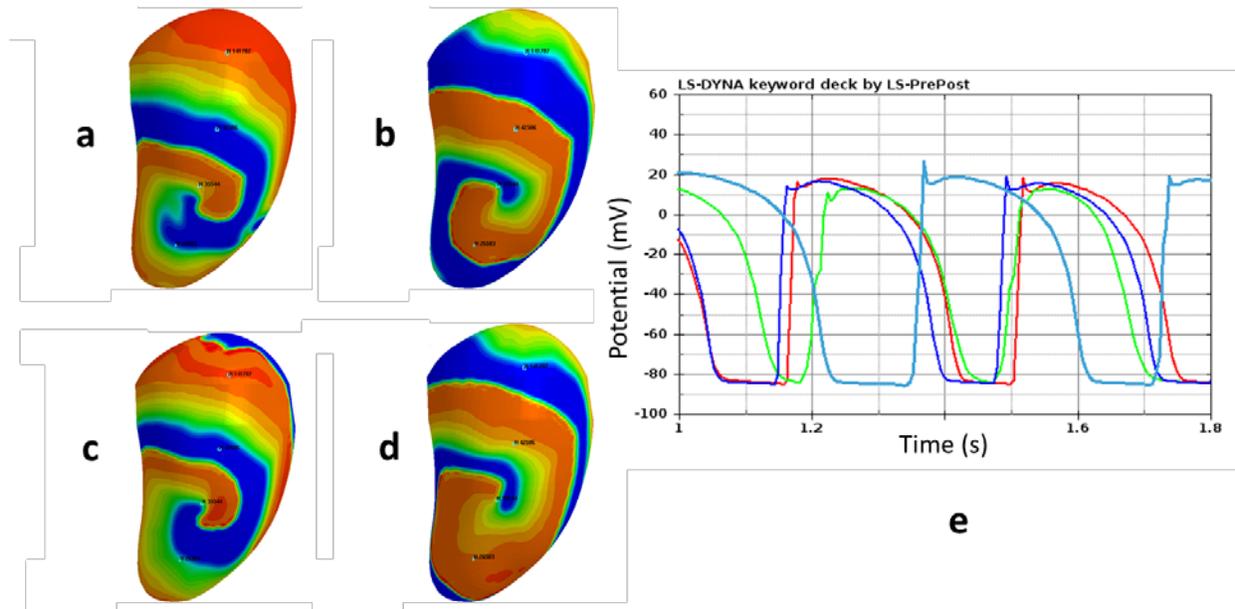


Figure 10: propagation of the transmembrane potential in a ventricle where a spiral wave has developed. The plots on the left show the fringe components of the potential at time 1050 ms (a), 1200 ms (b), 1400 ms (c), and 1600 ms (d). (e) shows time evolution of the potential at the 4 elements marked on the plots.

Conclusion

An EP solver was introduced in LS-DYNA. Both monodomain and bidomain methods have been developed, with different algorithms for each of them. At this point, these 2 models are coupled to a ten Tusscher and Panfilov cell model. The model was compared to a benchmark case on a cuboid tissue and shows good agreement with the other codes. The model was then used on a ventricular geometry, where the formation of spiral waves was observed.

In terms of EP, future work will consist of adding more cell models, and the capability to handle bath loading to account for the electrical conductive properties of the blood surrounding the heart. In terms of heart simulation in general, future work will consist of coupling the EP solver with the mechanical one, along with realistic tissue models, to simulate a heart pulse, and the ICFD solver to solve the blood flow.

References

- [1] Ruth Aris Sanchez, *Electromechanical Large Scale Computational Models of the Ventricular Myocardium*, PhD Thesis, Universitat Politecnica de Catalunya, Sept. 2014.
- [2] Mark Potse, Bruno Dubé, Jacques Richer, Alain Vinet, and Ramesh M. Gulrajani, *A Comparison of Monodomain and Bidomain Reaction-Diffusion Models for Action Potential Propagation in the Human Heart*, IEEE Trans. Biomed. Eng., vol. 53, no. 12, 2006.
- [3] FitzHugh R. *Mathematical models of threshold phenomena in the nerve membrane*. Bull. Math. Biophysics, 17:257—278, 1955
- [4] Nagumo J., Arimoto S., and Yoshizawa S. *An active pulse transmission line simulating nerve axon*. Proc IRE. 50:2061–2070, 1962

-
- [5] Fenton FH, Karma A., *Vortex dynamics in three-dimensional continuous myocardium. Filament instability and fibrillation*. Chaos 8: 20-47, 1998
- [6] K. H., W. J., ten Tusscher, A.V. Panfilov, *Alternans and spiral breakup in a human ventricular tissue model*, Am. J. Physiol. Heart Circ., Physiol 291: H1088-H1100, 2006.
- [7] M.E. Marsh, S.T. Ziaratgahi, and R.J. Spiteri, *The secrets to the success of the Rush-Larsen method and its generalizations*, IEEE Trans. Biomed. Engrg., 59(9):2506-2515., 2015
- [8] Steven A. Niederer, Eric Kerfoot, Alan P. Benson, Miguel O. Bernabeu, Olivier Bernus, Chris Bradley, Elizabeth M. Cherry, Richard Clayton, Flavio H. Fenton, Alan Garny, Elvio Heidenreich, Sander Land, Mary Maleckar, Pras Pathmanathan, Gernot Plank, Jos   F. Rodr  guez, Ishani Roy, Frank B. Sachse, Gunnar Seemann, Ola Skavhaug, Nic P. Smith, *Verification of cardiac tissue electrophysiology simulators using a N-version benchmark*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 369, issue 1954, pp. 4331-4351, November 2011.
- [9] Gernot Plank, Lufang Zhou, Joseph L. Greenstein, Sonia Cortassa, Raimond L. Winslow, Brian O'Rourke, Natalia A. Trayanova, *From mitochondrial ion channels to arrhythmias in the heart: computational techniques to bridge the spatio-temporal scales*, Philos Trans A Math Phys Eng Sci, 366(1879):3381-3409, September 2008.
- [10] Qu Z. abd Garfinkel A., *An advanced algorithm for solving partial differential equation in cardiac conduction*, IEEE Trans. Biomed. Eng. 46:1166-1168, 1999
- [11] Hankarjee Krishnamoorthi, Mainak Sarkar, William S. Klug, *Numerical Quadrature and Operator Splitting in Finit Element Methods to Cardiac ELectionophysiology*, Int J Numer Method Biomed Eng., 29(11):1243-1266, November 2013.

Appendices

Appendix A – Explicit Qu-Garfinkel Operator Split

The Qu-Garfinkel Operator Split [10], [11] consists in a 1/2 step for the diffusion equation followed by a full step for the ionic equation followed by a 1/2 step for the diffusion equation. The two 1/2 steps are for the diffusion explicit.

- Initialize $V = V_t$. Integrate diffusion operator for half time-step:

$$M.V_{t+1/2} = M.V_t - \frac{dt}{2\beta C_m} S.V_t$$

- Initialize $V = V_{t+1/2}$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_{t+1/2}^*$$

With f described in [6]

- Initialize $V = V_{t+1/2}^*$. Integrate diffusion PDE for half time-step:

$$M.V_{t+1} = M.V_{t+1/2}^* - \frac{dt}{2\beta C_m} S.V_{t+1/2}^*$$

Appendix B – Explicit first order Operator Split

The explicit first order Operator Split consists in a 1 step for the diffusion equation followed by a 1 step for the ionic equation at even time-step; and 1 step for the ionic equation followed by 1 step for the diffusion equation at odd time steps. The diffusion time steps are explicit.

B-1 At even time step

- Initialize $V = V_t$. Integrate diffusion operator for one time-step:

$$M.V_{t+1} = M.V_t - \frac{dt}{2\beta C_m} S.V_t$$

- Initialize $V = V_{t+1}$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_{t+1}^*$$

With f described in [6]

- Set $V_{t+1} = V_{t+1}^*$

B-2 At odd time step

- Initialize $V = V_t$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_t^*$$

With f described in [6]

- Set $V_t = V_t^*$
- Initialize $V = V_t$. Integrate diffusion operator for one time-step:

$$M.V_{t+1} = M.V_t - \frac{dt}{2\beta C_m} S.V_t$$

Appendix C – Implicit Qu-Garfinkel Operator Split

The Qu-Garfinkel Operator Split consists in a 1/2 step for the diffusion equation followed by a full step for the ionic equation followed by a 1/2 step for the diffusion equation. The two 1/2 steps for the diffusion are implicit.

- Initialize $V = V_t$. Integrate diffusion operator for half time-step:

$$\left(M + \frac{dt}{2\beta C_m} S \right) . V_{t+1/2} = M.V_t$$

- Initialize $V = V_{t+1/2}$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_{t+1/2}^*$$

With f described in [6]

Initialize $V = V_{t+1/2}^*$ Integrate diffusion PDE for half time-step:

$$\left(M + \frac{dt}{2\beta C_m} S \right) . V_{t+1} = M.V_{t+1/2}^*$$

Appendix D – Implicit first order Operator Split

The implicit first order Operator Split consists in a 1 step for the diffusion equation followed by a 1 step for the ionic equation at even time step; and 1 step for the ionic equation followed by 1 step for the diffusion equation at odd time steps. The diffusion time steps are implicit.

D-1 At even time step

- Initialize $V = V_t$. Integrate diffusion operator for one time-step:

$$\left(M + \frac{dt}{2\beta C_m} S \right) . V_{t+1} = M.V_t$$

- Initialize $V = V_{t+1}$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_{t+1}^*$$

With f described in [6]

- Set $V_{t+1} = V_{t+1}^*$

D-2 At odd time step

- Initialize $V = V_t$. Integrate ionic operator for full time-step:

$$\left. \begin{array}{l} C_m \dot{V} = I(V, u) \\ \frac{du}{dt} = f(V, u) \end{array} \right\} V_t^*$$

With f described in [6]

- Set $V_t = V_t^*$
- Initialize $V = V_t$. Integrate diffusion operator for one time-step:

$$\left(M + \frac{dt}{2\beta C_m} S\right) \cdot V_{t+1} = M \cdot V_t$$

Appendix E – Spiteri-Ziaratgahi Operator Split

$$u_{t+1} = u_t + dt f(u_t, V_t, t)$$

$$\begin{bmatrix} \frac{\beta C_m}{dt} M + S_i & S_i \\ S_i & S_{ie} \end{bmatrix} \cdot \begin{bmatrix} V_{t+1} \\ U_{et+1} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\beta C_m}{dt} M \cdot V_t - \beta M \cdot I_{ion}(u_{t+1}, V_t, t) \\ 0 \\ 0 \end{bmatrix}$$

The block system is solved using a PCG method, where the preconditioner is the diagonal line of the matrix, or with the hybrid-parallel, multifrontal, sparse direct solver, MF2.