Chapter 4

Operator Splitting and Finite Difference Schemes for Solving the EMI Model

Karoline Horgmo Jæger[1], Kristian Gregorius Hustad[1,2], Xing Cai[1,2] and Aslak Tveito[1,2]

Abstract We want to be able to perform accurate simulations of a large number of cardiac cells based on mathematical models where each individual cell is represented in the model. This implies that the computational mesh has to have a typical resolution of a few μm leading to huge computational challenges. In this paper we use a certain operator splitting of the coupled equations and show that this leads to systems that can be solved in parallel. This opens up for the possibility of simulating large numbers of coupled cardiac cells.

4.1 Introduction

In recent publications (31; 30; 13) we have shown that a cell-based model is useful for accurately representing the electrophysiology of excitable cells. Traditionally, excitable tissue is simulated based on homogenized models where the cells are not explicitly resolved, see e.g., (26; 7). In the cell-based model, we explicitly represent both the extracellular space (E), the cell membrane (M) and the intracellular space (I), and it is therefore referred to as the *EMI model*. Similar approaches to modeling excitable tissue have been used by several authors; see e.g., (2; 18; 25; 22; 24; 23; 11; 16; 34).

The EMI model is solved, numerically, using an operator splitting scheme which results in two steps; a non-linear system of ordinary differential equations (ODEs) to be solved in each computational node (i.e, degree of freedom) placed on the cell membrane, and a linear system of algebraic equations coupling the discrete Laplace equations of E and I with continuity requirements of the current over M. The spatial

[1]Simula Research Laboratory, Norway
[2]Department of Informatics, University of Oslo, Norway

A. Tveito et al. (eds.), *Modeling Excitable Tissue*, Simula SpringerBriefs on Computing 7, https://doi.org/10.1007/978-3-030-61157-6_4

resolution used in the discretization of the model is usually between 1 μm and 4 μm, thus only 1 mm^3 of tissue leads to more than 10^7 computational nodes. For an adult human cardiac cell, with a resolution of 2 μm, the number of computational nodes per cell (including the associated extracellular space) is about 6000 (see (30), Table 7). Thus, for a limited number of cells, the linear system coupling all the discrete Laplace equations is manageable. In fact, the system was solved using Matlab for up to 16,384 cells, with about 9.8×10^7 computational nodes, see (30).

However, not only the sheer size of the linear system is a challenge, also the properties of the linear system are unusual. In scientific computing, one of the most well-studied problems is solution of linear systems arising from the discretization of elliptic boundary value problems; see e.g., (5; 21; 8). Unfortunately, the EMI system does not naturally fall into the category of elliptic boundary value problems that can be solved using well-developed numerical machinery. It is therefore of importance to develop a splitting strategy for the EMI model that leads to sub-problems of the elliptic type. In (14), we showed that such a splitting can indeed be achieved. Here, we will review this convenient way of splitting the EMI model and show how to solve the system numerically using a finite difference method. Moreover, we will use the numerical scheme to assess the conduction properties in a small collection of cells where a sub-group of the cells are ischemic. Furthermore, we will present a parallel implementation of the splitting strategy, based on using open-source numerical libraries. This code is considerably faster than the existing Matlab code, and well suited for shared-memory parallel computers.

4.2 The EMI Model

We model the electrical properties of collections of cardiac cells using the EMI model introduced in (24; 1; 2; 31; 30). In Figure 4.1 we show the computational domains in the case of two coupled cells. Here, Ω_i^1 and Ω_i^2 denote the intracellular domains, and Ω_e denotes the extracellular space. The cell membranes are denoted by Γ_1 and Γ_2, respectively. The intercalated disc at the intersection between Ω_i^1 and Ω_i^2, allowing for currents between the cells, is denoted by $\Gamma_{1,2}$. With this notation at hand, the EMI model takes the following form:

$$
\begin{aligned}
\nabla\cdot\sigma_i\nabla u_i^k &= 0 \quad \text{in } \Omega_i^k, & n_e\cdot\sigma_e\nabla u_e &= -n_i^k\cdot\sigma_i\nabla u_i^k \equiv I_m^k \quad \text{at } \Gamma_k,\\
\nabla\cdot\sigma_e\nabla u_e &= 0 \quad \text{in } \Omega_e, & v_t^k &= \tfrac{1}{C_m}(I_m^k - I_{\text{ion}}^k) \quad \text{at } \Gamma_k,\\
u_e &= 0 \quad \text{at } \partial\Omega_e^D, & u_i^k - u_i^{\tilde{k}} &= w^k \quad \text{at } \Gamma_{k,\tilde{k}},\\
n_e\cdot\sigma_e\nabla u_e &= 0 \quad \text{at } \partial\Omega_e^N, & n_i^{\tilde{k}}\cdot\sigma_i\nabla u_i^{\tilde{k}} &= -n_i^k\cdot\sigma_i\nabla u_i^k \equiv I_{k,\tilde{k}} \quad \text{at } \Gamma_{k,\tilde{k}},\\
u_i^k - u_e &= v^k \quad \text{at } \Gamma_k, & w_t^k &= \tfrac{1}{C_g}(I_{k,\tilde{k}} - I_{\text{gap}}^k) \quad \text{at } \Gamma_{k,\tilde{k}},\\
s_t^k &= F^k \quad \text{at } \Gamma_k.
\end{aligned}
$$

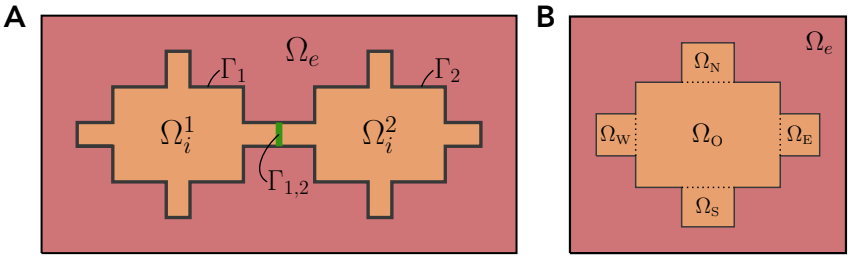

Fig. 4.1: **A**: Two-dimensional version of the EMI model domain in the case of two connected cells. Here, the cells Ω_i^1 and Ω_i^2, with cell membranes denoted by Γ_1 and Γ_2, respectively, are connected to each other by the intercalated disc, $\Gamma_{1,2}$, and surrounded by an extracellular space, denoted by Ω_e. **B**: Two-dimensional illustration of the geometry used for a single cell. The intracellular domain of each cell is composed of five subdomains Ω_O, Ω_W, Ω_E, Ω_S, and Ω_N. The sizes of the subdomains are specified in Table 4.1.

The model is stated for cell number k, and $\tilde{k}$ denotes one of the six neighboring cells (in 3D: north, west, south, east, above, below). In the model, u_e, u_i^k, and $v^k = u_i^k - u_e$ denote the extracellular, intracellular, and transmembrane potentials, respectively. Also, w^k is the potential difference across the intercalated disc[1], $\Gamma_{k,\tilde{k}}$, and σ_i and σ_e denote intracellular and extracellular conductivities, whereas C_m and C_g represent the specific capacitance of the membrane and the intercalated disc, respectively. Furthermore, n_e, n_i^k, and $n_i^{\tilde{k}}$ represent the outward pointing unit normal vectors of Ω_e, Ω_i^k and $\Omega_i^{\tilde{k}}$, respectively. A homogeneous Dirichlet boundary condition is applied at the outer extracellular boundary in the x-direction ($\partial\Omega_e^D$), and a homogeneous Neumann boundary condition is applied at the boundary in the y- and z-directions ($\partial\Omega_e^N$). The parameters used in the computations below are summarized in Table 4.1. The properties of the cell membrane and the gap junctions are represented by F, I_{ion} and I_{gap}. In the computations reported below, we use the Grandi et al. model(9), to model the dynamics of the membrane (F and I_{ion}), and for the gap junctions we use the simple passive model $I_{\text{gap}}^k = w^k / R_g$.

4.2.1 Operator Splitting Applied to the EMI Model

As mentioned above, a key step in solving the EMI model is to split the equations into parts that can be solved using standard tools. In (14), we derived a splitting scheme that leads to two key numerical challenges: Non-linear systems of ODEs to be solved

[1] Note that w^k is defined specifically for each cell.

Parameter	Value	Parameter	Value
Size Ω_O	$100\ \mu m \times 18\ \mu m \times 18\ \mu m$	C_m	$1\ \mu F/cm^2$
Size Ω_W, Ω_E	$4\ \mu m \times 10\ \mu m \times 10\ \mu m$	C_g	$0.5\ \mu F/cm^2$
Size Ω_N, Ω_S	$10\ \mu m \times 4\ \mu m \times 10\ \mu m$	σ_i	4 mS/cm
$\Delta x, \Delta y, \Delta z$	$2\ \mu m$	σ_e	20 mS/cm
Δt	0.02 ms	R_g	$0.0045\ k\Omega cm^2$
Δt_{ODE}	0.001 ms	M_{it}, N_{it}	2

Table 4.1: Parameter values used in the simulations, based on (13). For parameters of the Grandi model, see (9).

Algorithm 1: Summary of the splitting algorithm for the EMI model for connected cells.

Initial conditions: $v^{k,0}$, $s^{k,0}$, $w^{k,0}$, u_e^0 for all k.
for $n = 1, \dots, N_t$ **:**

Step 1: For all k, find $s^{k,n}$ and $\bar{v}^k$ at the nodes of the membrane Γ_k of cell k by solving a time step Δt from $(s^{k,n-1}, v^{k,n-1})$ of

$$\begin{aligned} v_t^k &= -\tfrac{1}{C_m} I_{ion}(v^k, s^k), \\ s_t^k &= F(v^k, s^k). \end{aligned}$$

Define $\bar{u}_e = u_e^{n-1}$, $\bar{w}^k = w^{k,n-1}$.

for $j = 1, \dots, N_{it}$ **:**

Step 2:

for $m = 1, \dots, M_{it}$ **:**

For every k, find $\bar{u}_i^k$ by solving

$$\begin{aligned} \nabla \cdot \sigma_i \nabla \bar{u}_i^k &= 0 && \text{in } \Omega_i^k, \\ \bar{u}_i^k + \tfrac{\Delta t}{C_m} n_i^k \cdot \sigma_i \nabla \bar{u}_i^k &= \bar{v}^k + \bar{u}_e && \text{at } \Gamma_k, \\ -n_i^k \cdot \sigma_i \nabla \bar{u}_i^k &= \tfrac{1}{R_g} \bar{w}^k + C_g \tfrac{\bar{w}^k - w^{k,n-1}}{\Delta t} && \text{at } \Gamma_{k,\tilde{k}}, \end{aligned}$$

where $\tilde{k}$ denotes each of the neighboring cells of cell k.

Update $\bar{w}^k = \bar{u}_i^k - \bar{u}_i^{\tilde{k}}$ at $\Gamma_{k,\tilde{k}}$ for all k and $\tilde{k}$.

end

Step 3: Find $\bar{u}_e$ by solving

$$\begin{aligned} \nabla \cdot \sigma_e \nabla \bar{u}_e &= 0 && \text{in } \Omega_e, \\ \bar{u}_e &= 0 && \text{at } \partial\Omega_e^D, \\ n_e \cdot \sigma_e \nabla \bar{u}_e &= 0 && \text{at } \partial\Omega_e^N, \\ n_e \cdot \sigma_e \nabla \bar{u}_e &= -n_i^k \cdot \sigma_i \nabla \bar{u}_i^k && \text{at } \Gamma_k \text{ for all } k. \end{aligned}$$

end

Define $u_e^n = \bar{u}_e$, $u_i^{k,n} = \bar{u}_i^k$, $w^k = \bar{w}^k$ for all k.

Step 4: Define $v^{k,n} = u_i^{k,n} - u_e^n$ at Γ_k for all k.

end

at each computational node located at the cell membranes, and a series of elliptic equations; see Algorithm 1. All the differential equations involved in Algorithm 1 are of classical type and can be solved using well-established numerical methods. In our present implementation, we apply a straightforward finite difference scheme (see e.g., (29) for an elementary introduction to finite differences) for the elliptic equations and the Forward Euler method with a substepping time step Δt_{ODE} for solving the non-linear ODEs modeling the membrane dynamics (see (30)). However, it is worth observing that elliptic equations can as well be solved using the finite element method, or a finite volume method, thus allowing for more flexible and adaptive meshes.

4.3 Simulating the Effect of a Region of Ischemic Cells

In order to demonstrate an application of Algorithm 1 above, we consider a collection of cells where a fraction of the cells are ischemic. This is known to perturb the electrical conduction and may lead to arrhythmias; see e.g., (33; 28; 19; 27). This problem has been carefully studied using homogenized models (mostly the monodomain model), but here we will show that the ischemic regions also have local effects when only very few cells are considered. In Figure 4.2, we consider a collection of cells organized in a two-dimensional mesh of 22×12 cells. The cells are modeled using the Grandi model with parameters as stated in (9). Within the domain, 8×6 of the center cells are ischemic in the sense that the extracellular potassium concentration surrounding these cells is increased from 5.4 mM to 10 mM. For the ischemic cells, we use the steady-state values of the state variables for the increased extracellular potassium concentration as initial conditions, and for the remaining cells, we use the steady-state values of the default Grandi model. In addition, we run the simulation for 5 ms before stimulation.

In the simulation results we observe that the ischemic region slows down conductions and thus perturbs the wave in the intracellular potential moving from left to right. This is consistent with the result obtained in (6) (cf. Figure 5), where the monodomain model was used. Such perturbations are known to be arrhythmogenic and have been observed several times in numerical experiments; see e.g., (33; 19; 15; 3; 6). Here, we observe that such perturbations can be initiated locally when only a few cells are subject to surroundings with elevated potassium concentration.

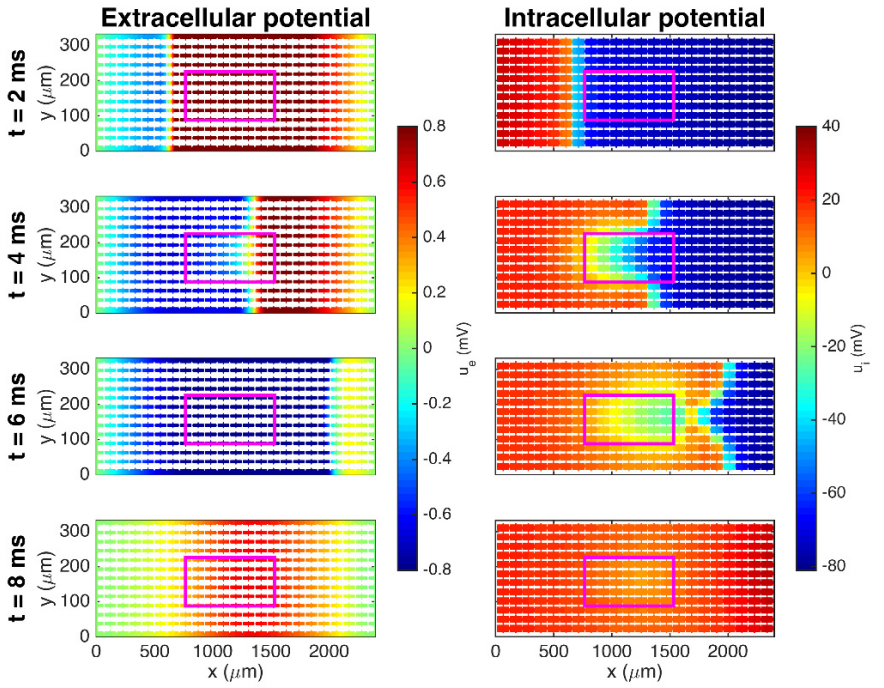

Fig. 4.2: Extracellular potential (left) and intracellular potential (right) at four different points in time in an EMI model simulation with an ischemic region in the center of the domain, marked by the purple rectangle. The parameter values used in the simulation are given in Table 4.1.

4.4 A Scalable Implementation of the Splitting Scheme

In expectation of future simulations of excitable issues that may involve a huge number of cells, we see the need of a scalable implementation of the new splitting scheme, so that it can run efficiently on parallel computers. One specific criterion is that the computation time should grow linearly with the number of cells involved. Additionally, the design goals of this new code should also include independence of proprietary software (such as Matlab) and plug-and-play of the different numerical components. This section will present a preliminary version of such a scalable implementation.

4.4.1 The Linear System for the Intracellular Potential

The main benefit of the new the splitting scheme is that the intracellular Laplace equations (one per cell) are decoupled from the extracellular Laplace equation, as stated in Algorithm 1. If we assume a constant intracellular conductivity σ_i and that each cell is of the same shape and size, as shown in Figure 4.1, the matrices arising from a standard finite difference discretization of the intracellular Laplace equations for the individual cells will be mostly identical. There are only a small number of unique intracellular matrices, depending on whether there is a neighboring cell connected to each of the intercalated discs. It is thus unnecessary to compute an intracellular matrix for each cell. Instead, the cells that have the same neighbor connectivity situation can share the same intracellular matrix. This not only reduces the memory usage of an implementation, but also improves data reuse in the caches of a computer. Moreover, since the number of computational nodes per intracellular domain is relatively small (each intracellular domain has about 5300 degrees of freedom for the simulations used in this chapter), it is very efficient to use a direct solver each time an intracellular Laplace problem needs to be solved. Specifically, the LU factorization of each unique intracellular matrix A_I can be pre-calculated, which renders the solution of $A_I \bar{u}_i^k = b_i^k$ per cell to be merely invoking the forward-backward substitution procedure. Parallelism of the computation mainly arises from the fact that the intracellular Laplace equations can be solved independently of each other, while limited parallelism also exists within each forward-backward substitution.

4.4.2 The Linear System for the Extracellular Potential

For the overall extracellular Laplace problem, which can be huge depending on the spatial resolution and the total number of cells, an iterative solver is more appropriate. Take for instance the case of 128×128 cells. The corresponding discrete extracellular Laplace equation has $107,202,214$ degrees of freedom. Independent of the spatial resolution and the number of cells involved, the extracellular matrix A_E arising from a standard finite difference discretization is symmetric and positive-definite (some care is needed to discretize the boundary conditions on the membranes). The resulting linear system $A_E \bar{u}_e = b_e$ is thus a perfect candidate for the conjugate gradients (CG) method with an algebraic multigrid (AMG) preconditioner. Under optimal conditions, an AMG preconditioner requires a constant number of iterations to reach convergence independent of the linear system size, although the number of grid levels inside the AMG preconditioner may increase with the system size. Parallelism readily exists in iterative solvers, with several software libraries providing parallel implementations of CG and AMG.

4.4.3 The Non-Linear ODE System for the Membrane Potential

For solving the non-linear ODE system per computational node on the membranes, a straightforward and often very efficient numerical strategy is the Forward Euler method with a substepping time step Δt_{ODE}. Since the non-linear ODE system on each membrane node is independent of the others, the ODE computation possesses the most ample parallelism.

4.4.4 The Implementation

The Python programming language has been chosen for the implementation, mostly because of its flexibility for interfacing with numerical software libraries written in performance-friendly languages such as C and C++. We have used the `ctypes` module from the standard Python library for this purpose. The choice of Python also simplified a partial translation from the existing MATLAB code developed in (14).

We have chosen the SuperLU library (17) for performing the LU factorization of the intracellular matrices and the subsequent forward-backward substitution, via the bindings that are provided by SciPy (32). For the extracellular Laplace equation, we have used the ViennaCL library (20) for its implementation of CG and AMG. The CG iterations are by default configured to terminate when a tolerance of 10^{-5} is reached. The AMG preconditioner has been configured to use the maximum independent set (MIS), see (4), as the coarsening algorithm and smoothed aggregation as the interpolation algorithm. For the ODE part, the Gotran automated code generator (10) has been used to translate the Grandi cell model into C code, callable from the Python side.

4.4.5 Parallelization

The Python implementation currently relies on the adopted numerical libraries (SuperLU and ViennaCL) for an implicit parallelization of the PDE computation through multi-threading. This form of parallelization suits for shared-memory parallel computers, such as laptops or servers that use multicore CPUs. Multi-threading of the ODE computation is also enabled by inserting OpenMP compiler directives into the C code that is generated automatically by Gotran. The advantage of this implicit parallelization is that the user does not have to care about parallelization-specific coding. The downside is that all the computations have to run on a shared-memory system. It is possible to achieve the more general parallelization that targets distributed-memory parallel computers, which will be a task for future work.

4.4.6 Performance Results

The simulations in this section were run on a dual-socket server with two 32-core AMD EPYC 7601 CPUs, each with 8-channel memory operating at 2666 MT/s. The number of OpenMP threads was set by default to the number of logical cores, equaling 128. Moreover, the environment variable `OMP_PROC_BIND=TRUE` was set to prevent the threads from migrating between the cores (which typically leads to unnecessary performance loss).

Table 4.2 shows the average solution time per time step for the 10 first time steps, where all the parameters are as prescribed in Table 4.1. The number of cardiac cells is doubled in the x and y directions for each row, and we observe that the time per cell remains fairly constant, indicating that the time to solution is a linear function of the number of cardiac cells simulated.

Cells	time usage for all cells (s)				time per cell (ms)			
	E	M	I	total	E	M	I	total
4×4	0.38	0.03	0.09	0.50	24.0	2.2	5.3	31.5
8×8	1.54	0.11	0.34	1.99	24.1	1.8	5.2	31.2
16×16	2.27	0.45	1.20	3.92	8.8	1.7	4.7	15.3
32×32	8.91	1.72	4.98	15.61	8.7	1.7	4.9	15.2
64×64	30.46	6.73	19.15	56.33	7.4	1.6	4.7	13.8
128×128	123.73	30.60	72.83	227.16	7.6	1.9	4.4	13.9

Table 4.2: Average solution time per time step for the E, M and I domains.

4.5 Software

The Matlab code used to compute the solutions shown in Figure 4.2 and the Python code discussed in Section 4.4 can be found at `https://github.com/KGHustad/emi-book-2020-splitting-code`. An archived version (12) is also available.

4.6 Conclusion

In this chapter we have presented a numerical scheme for solving the EMI equations using operator splitting. The scheme allows for parallel solution of individual cells combined with a global solution of the equation modeling the extracellular potential.

The latter is well suited for using optimal linear solvers such as AMG. The overall code scales linearly with the number cells and thus allows for simulation of a large number of cells. It remains to be seen how well this will work for very large numbers of cells; this is subject for further work.

Acknowledgements The research presented in this chapter has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

References

1. Agudelo-Toro A (2012) Numerical simulations on the biophysical foundations of the neuronal extracellular space. PhD thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen
2. Agudelo-Toro A, Neef A (2013) Computationally efficient simulation of electrical activity at cell membranes interacting with self-generated and externally imposed electric fields. Journal of Neural Engineering 10(2):026019
3. Alonso S, Bär M, Echebarria B (2016) Nonlinear physics of electrical wave propagation in the heart: a review. Reports on Progress in Physics 79(9):096601
4. Bell N, Dalton S, Olson LN (2012) Exposing fine-grained parallelism in algebraic multigrid methods. SIAM Journal on Scientific Computing 34(4):C123–C152, DOI 10.1137/110838844
5. Benzi M (2002) Preconditioning techniques for large linear systems: a survey. Journal of computational Physics 182(2):418–477
6. Dutta S, Mincholé A, Quinn TA, Rodriguez B (2017) Electrophysiological properties of computational human ventricular cell action potential models under acute ischemic conditions. Progress in biophysics and molecular biology 129:40–52
7. Franzone PC, Pavarino LF, Scacchi S (2014) Mathematical Cardiac Electrophysiology. Springer International Publishing
8. Gergelits T, Mardal KA, Nielsen BF, Strakos Z (2019) Laplacian preconditioning of elliptic pdes: Localization of the eigenvalues of the discretized operator. SIAM Journal on Numerical Analysis 57(3):1369–1394
9. Grandi E, Pasqualini FS, Bers DM (2010) A novel computational model of the human ventricular action potential and Ca transient. Journal of Molecular and Cellular Cardiology 48:112–121
10. Hake J, Finsberg H, Hustad KG, Bahij G (2020) Gotran – General ODE TRANslator. `https://github.com/ComputationalPhysiology/gotran`
11. Hogues H, Leon LJ, Roberge FA (1992) A model study of electric field interactions between cardiac myocytes. IEEE Transactions on Biomedical Engineering 39(12):1232–1243
12. Jæger KH, Hustad KG (2020) Supplementary material (code) for the chapter "Operator splitting and finite difference schemes for solving the EMI model" appearing in "EMI: Cell-based Mathematical Model of Excitable Cells". DOI 10.5281/zenodo.3707472, URL `https://doi.org/10.5281/zenodo.3707472`
13. Jæger KH, Edwards AG, McCulloch A, Tveito A (2019) Properties of cardiac conduction in a cell-based computational model. PLoS computational biology 15(5):e1007042
14. Jæger KH, Hustad KG, Cai X, Tveito A (2020) Efficient numerical solution of the EMI model representing the extracellular space (E), cell membrane (M) and intracellular space (I) of a collection of cardiac cells. Preprint
15. Kazbanov IV, Clayton RH, Nash MP, Bradley CP, Paterson DJ, Hayward MP, Taggart P, Panfilov AV (2014) Effect of global cardiac ischemia on human ventricular fibrillation: in-

sights from a multi-scale mechanistic model of the human heart. PLoS computational biology 10(11):e1003891
16. Krassowska W, Neu JC (1994) Response of a single cell to an external electric field. Biophysical Journal 66(6):1768–1776
17. Li XS (2005) An overview of SuperLU: Algorithms, implementation, and user interface. ACM Trans Math Softw 31(3):302–325
18. Roberts SF, Stinstra JG, Henriquez CS (2008) Effect of nonuniform interstitial space properties on impulse propagation: a discrete multidomain model. Biophysical Journal 95(8):3724–3737
19. Romero L, Trénor B, Alonso JM, Tobón C, Saiz J, Ferrero JM (2009) The relative role of refractoriness and source–sink relationship in reentry generation during simulated acute ischemia. Annals of Biomedical Engineering 37(8):1560–1571
20. Rupp K, Tillet P, Rudolf F, Weinbub J, Morhammer A, Grasser T, JÃŒngel A, Selberherr S (2016) ViennaCL—linear algebra library for multi- and many-core architectures. SIAM Journal on Scientific Computing 38(5):S412–S439, DOI 10.1137/15M1026419
21. Smith B, Bjorstad P, Gropp W (2004) Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge university press
22. Stinstra J, MacLeod R, Henriquez C (2010) Incorporating histology into a 3D microscopic computer model of myocardium to study propagation at a cellular level. Annals of Biomedical Engineering 38(4):1399–1414
23. Stinstra JG, Hopenfeld B, MacLeod RS (2005) On the passive cardiac conductivity. Annals of Biomedical Engineering 33(12):1743–1751
24. Stinstra JG, Roberts SF, Pormann JB, MacLeod RS, Henriquez CS (2006) A model of 3D propagation in discrete cardiac tissue. In: Computers in Cardiology, 2006, IEEE, pp 41–44
25. Stinstra JG, Henriquez CS, MacLeod RS (2009) Comparison of microscopic and bidomain models of anisotropic conduction. In: Computers in Cardiology, IEEE, pp 657–660
26. Sundnes J, Lines G, Cai X, Nielsen B, Mardal KA, Tveito A (2006) Computing the Electrical Activity of the Heart. Springer
27. Tveito A, Lines G (2008) A condition for setting off ectopic waves in computational models of excitable cells. Math Biosci 213:141–150
28. Tveito A, Lines GT (2009) A note on a method for determining advantageous properties of an anti-arrhythmic drug based on a mathematical model of cardiac cells. Mathematical Biosciences 217(2):167–173, DOI DOI:10.1016/j.mbs.2008.12.001, URL http://www.sciencedirect.com/science/article/B6VHX-4V70RG1-1/2/8b8ae8d1fbf9e2c74235b7e7a97c6f6e
29. Tveito A, Langtangen HP, Nielsen BF, Cai X (2010) Elements of scientific computing, vol 7. Springer Science & Business Media
30. Tveito A, Jæger KH, Kuchta M, Mardal KA, Rognes ME (2017) A cell-based framework for numerical modeling of electrical conduction in cardiac tissue. Frontiers in Physics 5:48
31. Tveito A, Jæger KH, Lines GT, Paszkowski Ł, Sundnes J, Edwards AG, Mäki-Marttunen T, Halnes G, Einevoll GT (2017) An evaluation of the accuracy of classical models for computing the membrane potential and extracellular potential for neurons. Frontiers in Computational Neuroscience 11:27
32. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Jarrod Millman K, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey C, Polat i, Feng Y, Moore EW, Vand erPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, Contributors S (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17:261–272, DOI https://doi.org/10.1038/s41592-019-0686-2
33. Xie F, Qu Z, Garfinkel A, Weiss JN (2001) Effects of simulated ischemia on spiral wave stability. American Journal of Physiology-Heart and Circulatory Physiology 280(4):H1667–H1673
34. Ying W, Henriquez CS (2007) Hybrid finite element method for describing the electrical response of biological cells to applied fields. IEEE Transactions on Biomedical Engineering 54(4):611–620