COMPUTATIONALLY EFFICIENT METHODS FOR SOLVING THE BIDOMAIN EQUATIONS IN 3D

Edward J. Vigmond[1], Felipe Aguel[2] and Natalia A. Trayanova[2]
[1]Department of Electrical and Computer Engineering, University of Calgary, Calgary, Canada
[2]Department of Biomedical Engineering, Tulane University, New Orleans, USA

Abstract— **The bidomain equations represent the most complete description of cardiac electrical activity. However, the equations prove computationally burdensome as the resulting system of equations has two entries per spatial node. This paper examines the computational performance obtained by decoupling the bidomain equations into two separate systems of equations, an elliptic equation for the extracellular potential, and a parabolic equation for the transmembrane voltage. Each set of equations was solved on different grids with different time steps. For the elliptic problem, the performances of direct and iterative solvers were compared. For the parabolic equation, the interconnected cable method (ICCM) was compared to the finite element method (FEM). Results were obtained by simulating activity in a 3D slab of cardiac tissue whose ionic currents were described by modified Beeler-Reuter equations. It was found that the elliptic equation solution dominated the calculation. Reducing the frequency of solution and/or halving the spatial resolution resulted in considerable speed up while maintaining a reasonable error. Direct solvers were faster by a factor of 2–3 and the ICCM was about twice as fast in solving the parabolic equation as compared to the FEM. Both the elliptic and parabolic equations scaled linearly with the number of nodes.**

Keywords— **computer modeling, bidomain, cardiac, finite element method**

I. Introduction

ELECTRICAL shocks are the only known therapy for hearts in fibrillation. Without intervention, death will quickly result. The mechanisms underlying electrical defibrillation, however, still remain elusive. Electrical mapping of the cardiac electrical activity during the delivery of defibrillatory shocks is hindered by the high shock strength, while optical mapping techniques are limited to surface measurements. Modeling is therefore needed to help resolve the events during defibrillation in the 3D volume of the heart.

The bidomain representation of cardiac tissue is the most complete description of cardiac electrical activity[1]. It describes both the intracellular and extracellular potential fields, linking them through membrane behavior. It has predicted the appearance of shock-induced virtual electrodes[2] which were later confirmed experimentally[3]. The bidomain equations are computationally expensive as they require two unknowns for each spatial node, resulting in large matrices which consume much memory and require long solution times.

Besides this inherent computational burden in solving the bidomain equations, simulating defibrillation in the heart is also an intrinsically large problem. A piece of tissue large enough to support fibrillation must be modeled, on the order of centimeters, and because of the length constants involved, it must be discretized on the order of hundreds of micrometers. Furthermore, the kinetics of the sodium gate impose a time step on the order of microseconds while the window of observation to determine the outcome of a shock is on the order of hundreds of milliseconds. Thus, both spatial and temporal considerations contribute to the size of the problem.

Many techniques are available to solve the reaction-diffusion equations describing cardiac electrical activity. The InterConnected Cable Method (ICCM)[4], [5] is a computationally efficient method that has been used in monodomain simulations of three dimensional cardiac tissue with fiber rotation[6]. It is based on decomposing the tissue into a set of cables which may follow arbitrary trajectories, but, by itself, is not suitable for solving the bidomain equations. The finite element method (FEM) allows modeling of complex geometry and has been used to solve bidomain problems on a whole rabbit heart[7], but is more computationally demanding than the ICCM.

This study examines several techniques to increase the computational efficiency of solving the bidomain equations. Benefits to be gained from recasting the bidomain equations into decoupled elliptic and hyperbolic problems are examined. With the problems isolated, each is solved with different time steps and on different meshes in order to reduce computational demand. By comparing results with the fully coupled bidomain solution, the simulation parameters which maximize computational speed while maintaining sufficient accuracy are determined.

II. Methods

A. Governing Equations

The basic bidomain equations[1] relate the intracellular potential, ϕ_i, to the extracellular potential, ϕ_e, through the transmembrane current density, I_m:

$$\nabla \cdot \bar{\sigma}_i \nabla \phi_i = \beta I_m - I_{trans} \tag{1}$$

$$\nabla \cdot \bar{\sigma}_e \nabla \phi_e = -\beta I_m + I_{trans} - I_e \tag{2}$$

$$I_m = C_m \frac{\partial V_m}{\partial t} + I_{ion} \tag{3}$$

where $\bar{\sigma}_i$ and $\bar{\sigma}_e$ are respectively the intracellular and extracellular conductivity tensors, β is the surface to volume ratio of the cardiac cells, I_{trans} is the transmembrane current stimulus, I_e is an extracellular current stimulus, C_m is the capacitance per unit area, V_m is the transmembrane voltage which is defined as $\phi_i - \phi_e$, and I_{ion} is the current density flowing through the ionic channels. The Beeler-Reuter Drouhard-Roberge model modified to handle large voltages was used as the ionic model[8] in this study. This formulation will be referred to as the coupled set of equations since

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25 Oct 2001	N/A	-

Title and Subtitle Computationally Efficient Methods for Solving the Bidomain Equations in 3D	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Department of Electrical and Computer Engineering University of Calgary Calgary Canada	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
Papers from 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom.

Abstract

Subject Terms

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU

Number of Pages
4

the intracellular and extracellular potentials are solved simultaneously.

By adding Eqn. 1 and Eqn. 2 and using the definition of V_m, the equations can be cast in a slightly different form with V_m and ϕ_e as the independent variables.

$$\nabla \cdot (\bar{\sigma_i} + \bar{\sigma_e}) \nabla \phi_e = -\nabla \cdot \bar{\sigma_i} \nabla V_m - I_e \quad (4)$$

$$\nabla \cdot \bar{\sigma_i} \nabla V_m = -\nabla \cdot \bar{\sigma_i} \nabla \phi_e + \beta I_m - I_{trans} \quad (5)$$

Equations 4 and 5 are decoupled and solved sequentially as an elliptic problem (Eqn. 4) and a parabolic problem (Eqn. 5). Since the two systems are now independent of each other, they can be solved with different time steps and at different spatial resolutions. This formulation will be referred to as the decoupled system.

If either the extracellular electric field can be ignored, or, the ratios of the longitudinal to transverse conductivities in the intracellular and extracellular domains are equal, the bidomain equations can be replaced with the monodomain equation. This is Eqn. 5 with the intracellular conductivity tensor replaced by the the monodomain conductivity tensor, $\bar{\sigma}_m$, which is a function of the bidomain conductivity tensors[9], $\bar{\sigma}_m = \bar{\sigma_i}(\bar{\sigma_i} + \bar{\sigma_e})^{-1}\bar{\sigma_e}$.

B. Solution Methods

To solve the fully coupled system (Eqn.'s 1 and 2) the FEM was used based on a Galerkin formulation. In matrix notation with a time step of Δt, the resultant discretized system is given by

$$\begin{bmatrix} \mathbf{K_e} - \kappa\mathbf{M} & \kappa\mathbf{M} \\ \kappa\mathbf{M} & \mathbf{K_i} - \kappa\mathbf{M} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_e^{t+1} \\ \boldsymbol{\phi}_i^{t+1} \end{bmatrix} = \mathbf{M}\left(\kappa \begin{bmatrix} \mathbf{v_m^t} \\ -\mathbf{v_m^t} \end{bmatrix} + \beta \begin{bmatrix} -\mathbf{i_{ion}^t} \\ \mathbf{i_{ion}^t} \end{bmatrix} - \begin{bmatrix} \mathbf{i_e^t} - \mathbf{i_{trans}^t} \\ \mathbf{i_{trans}^t} \end{bmatrix} \right) \quad (6)$$

where $\kappa = \beta C_m / \Delta t$, $\mathbf{M}$ is the FEM lumped mass matrix and $\mathbf{K}$ is the FEM stiffness matrix. Both matrices were computed using linear tetrahedral elements with the subscript on $\mathbf{K}$ denoting whether the matrix was created using $\bar{\sigma_i}(i)$ or $\bar{\sigma_e}(e)$. Superscripts refer to the time step.

To solve the elliptic equation of the decoupled system (Eqn. 4), an FEM approach was also used:

$$\mathbf{K_{i+e}} \boldsymbol{\phi_e^t} = \mathbf{K_i} \mathbf{v_m^t} - \mathbf{M} \mathbf{i_e^t} \quad (7)$$

where the subscript of the stiffness matrix denotes that the sum of the conductivity tensors was used.

To solve the parabolic problem of the decoupled system (Eqn. 5), two different schemes were used, ICCM and FEM. The ICCM solution utilized a semi-implicit time integration on a one-dimensional linear grid to solve the particular solution[4]. The FEM solution utilized a forward Euler time integration:

$$\mathbf{v_m^{t+1}} = \mathbf{v_m^t} + \frac{\mathbf{1}}{\kappa}\Big(\mathbf{M^{-1}} \mathbf{K_i} \big(\mathbf{v_m^t} + \boldsymbol{\phi_e^t} \big) - \beta \mathbf{i_{ion}^t} + \mathbf{i_{trans}^t} \Big) \quad (8)$$

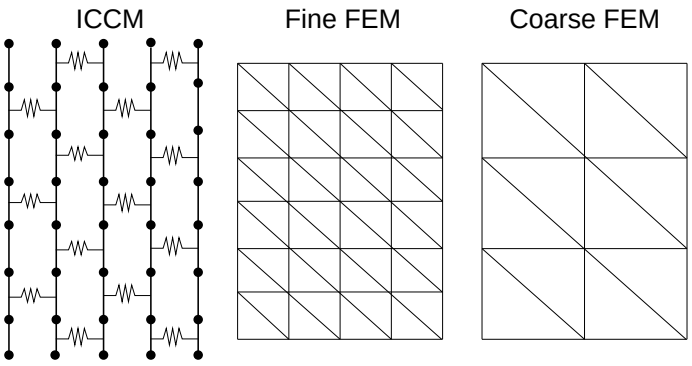

Fig. 1. Grids used for different solution methods. All grids represent the same space. They are shown in 2D for clarity.

C. Grid Generation

Grid generation began by constructing an ICCM grid as has been described previously[5] (see Figure 1). Cables were laid in sheets spaced 100 μm apart. Within each sheet, cables were parallel and all ran in the y direction. The domain spanned by the cables was a right-angled hexahedron. Cables were then discretized into 100 μm long segments with each segment connected to a segment in a neighboring cable through a gap junction that was represented by a fixed resistance.

Two different FEM meshes were constructed from the cables, a fine mesh and a coarse mesh. To construct the fine three-dimensional FEM mesh, the centers of the cable segments were used as nodes from which to construct first-order tetrahedrons. To construct the coarse FEM mesh, every other point in every other cable of every other layer was used resulting in a mesh that was approximately one-eighth the size of the finer mesh. The conductivity tensors in the tetrahedral elements were defined by the cable directions in the ICCM model.

The effect of using a coarse mesh for the elliptic equation and a fine mesh for the parabolic equation was tested. Once a solution was obtained for $\boldsymbol{\phi}_e$ on the coarse mesh, interpolation was used to assign potential values at the fine mesh nodes which were used in the solution of the parabolic equation.

D. Matrix Solvers

To solve the coupled set of equations and the elliptic equation, both iterative and direct methods were used. While direct methods are generally faster when repeatedly solving the same system of equations, they require much more memory since performing a decomposition on a sparse matrix preserves matrix bandwidth but fills in the zero entries between bands[10]. One must therefore use iterative methods on large problems where performing a matrix decomposition would exceed computer memory. The direct method used here was an SGI (Mountain View, CA) supplied LDL^T decomposition, where L is a lower triangular matrix and D is a diagonal matrix. After the decomposition, the system was solved by forward and backward substitutions. A cus-

tom coded conjugate gradient method with an Incomplete Cholesky decomposition preconditioner was used as the iterative solver[11].

All simulations were performed on an SGI Origin 2100 computer which had 350 MHz MIPS R12000 processors and 4 gigabytes of memory. Times given for simulations are CPU times for a single processor.

III. Results

A. Solution Methods

The CPU time taken to simulate 25 ms of activity in a rectangular 3D block of cardiac tissue with zero flux boundary conditions was measured. The block measured $1.6 \times 0.6 \times 0.11$ ($x \times y \times z$) cm and was composed of 108,031 intracellular nodes. Activity was initiated in one corner of the block by applying a 2 ms suprathreshold transmembrane stimulus. The activity propagated out with an ellipsoidal wavefront which reached the x and y edges at approximately 25 ms. The CPU time to perform the simulation is given in Table I for the various methods. $\Delta t^E/\Delta t^P$ refers to the ratio of the time step used for the elliptic equation solve over the time step used for the parabolic equation solve. Parabolic solves were always performed with a time step of 10 μs. Thus, $\boldsymbol{\phi}_e$ was not necessarily updated as frequently as $\mathbf{v_m}$. The value of ∞ corresponds to solving the monodomain equation. Fine and Coarse refer to the discretization of the FEM grid on which the elliptic equation was solved. Iterative refers to solving the elliptic equation on a fine grid using the conjugate gradient method.

TABLE I
CPU time to simulate 25 ms for various methods.

Method	$\Delta t^E/\Delta t^P$	Fine	Coarse	Iterative
ICCM	1	3989.67	941.20	8517.56
	2	2270.54	796.63	5909.88
	4	1396.46	629.02	4115.56
	10	878.25	602.41	2761.32
	∞	398.42	-	-
FEM	1	4868.44	1171.68	10071.34
	2	2515.26	979.92	7111.40
	4	1643.83	884.71	5018.22
	10	1123.82	819.91	3404.27
	∞	757.88	-	-
Coupled	-	13489.57	-	65867.97

Solving the coupled equation was much slower than solving the decoupled system. The ICCM method was approximately twice as fast as the FEM method in solving the parabolic problem. An iterative solver was slightly more than twice as slow to solve the elliptic problem at a $\Delta t^E/\Delta t^P$ of one and became three times as slow when $\Delta t^E/\Delta t^P$ was ten. This was due to the larger change in $\boldsymbol{\phi}_e$ when computed less often, thereby requiring more iterations for convergence. Increasing $\Delta t^E/\Delta t^P$ from one to ten increased the average number of iterations per elliptic solve from 52 to 94.

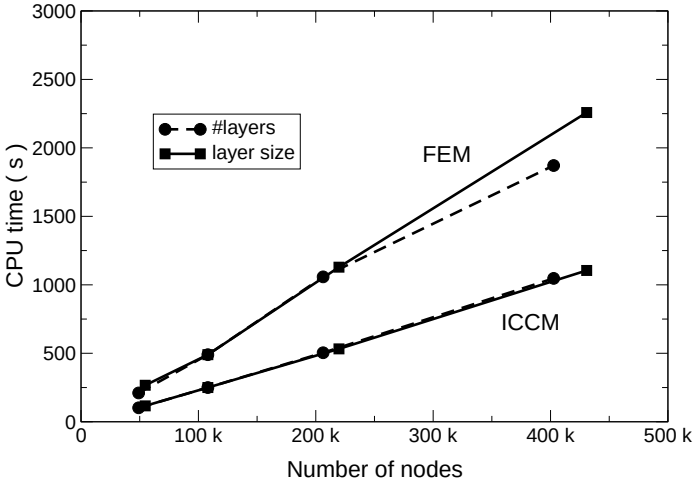

Fig. 2. Time spent in the parabolic solution for a 25 ms long simulation as a function of problem size. The size of the problem was increased by increasing the number of layers (#layers) or by increasing the size of each layer (layer size).

The elliptic solve was the most costly part of the solution. Reducing the number of times that the elliptic equation is solved (increasing $\Delta t^E/\Delta t^P$) or reducing the size of the elliptic problem (using a coarse grid) greatly decreased the simulation time. Monodomain solutions were obviously the quickest.

B. Problem Size

The dependence of simulation time on problem size was next ascertained. The problem size was varied in two different manners. The first manner was to increase the z dimension which simply increased the number of layers in the block. This resulted in an increase in the number of nodes while preserving the bandwidth of any matrices. The second method involved scaling the size of each layer in the x and y directions by the same factor while keeping the number of layers constant. This method resulted in matrices whose bandwidth increased with the scaling factor. The effect of problem size was broken down into the effect on the time of the parabolic solution (Fig. 2) and the effect on the time of the elliptical solution for a coarse gird (Fig. 3).

The ICCM method was about twice as fast as the FEM method in solving the parabolic equation. The ICCM method scaled linearly with problem size, regardless of how the problem size was increased. The curves for the ICCM method were on top of each other since bandwidth is not a consideration for this method.

The FEM showed a linear increase in CPU time with problem size. At the smaller sizes, the manner in which the problem was increased in size did not affect the computation time. However, for very large problems, more than 200,000 nodes, increasing the layer size caused a larger increase in computation time compared to an increase in the number of layers.

The effect of increasing the problem size for the elliptical problem was similar to the one obtained for the FEM parabolic solve. The manner in which the size was increased

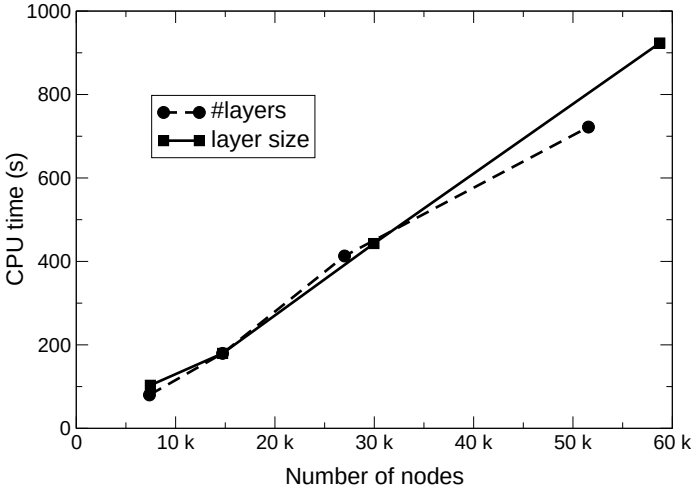

Fig. 3. Time spent solving the elliptic equation in a simulation of 25 as a function of problem size. The number of nodes in the model was increased by either increasing the number of layers in the model or by increasing the size of each layer.

was only significant for very large problems.

IV. Discussion

This paper analyzed the time to compute a simulation using the bidomain model for various methods and techniques. Solving the decoupled system of the bidomain equations is computationally advantageous. By splitting the problem into two, each part can be solved independently. This splitting leads to two smaller problems whose total work is less than that required to solve the coupled system. Furthermore, there is a large savings in memory since the matrices to be constructed are much smaller.

In splitting the bidomain equations into two parts, it was found that the elliptic equation was much more costly to solve than the parabolic equation. Thus, computational speed was increased by a combination of solving the elliptic problem on a grid with coarser discretization or by solving this equation less often than the parabolic equation. Using a coarse grid had a similar speedup to decreasing the frequency at which the elliptic equation was solved to one tenth. Performing the elliptic solve at less than this frequency lead to large errors in computation. Combining a coarse grid with periodic elliptical solves only increased the performance marginally beyond a $\Delta t^E/\Delta t^P$ value of 4.

The ICCM was found to be about twice as fast as the FEM for solving the parabolic portion of the bidomain problem. However, when the elliptic portion and ionic computation are factored into the problem, the savings in computation time between using the FEM and ICCM reduces to about 30%. Also, this savings of 30% will diminish if a more detailed ionic model is used. The ionic model used here was very simple, only having seven state variables, while recent models have over 20[12]. The added complexity in setting up the problem to utilize both an ICCM grid and FEM grid may not be justified for an increase in performance of only 10%.

The direct solver for the elliptic elliptic was 2–3 times faster than the iterative solver. It is therefore only reasonable to use the iterative solver if memory is an issue and the decomposed matrix does not fit into memory. A decomposed sparse matrix can have 10–20 times as many nonzero entries as the original matrix due to fill in. This an additional reason to solve the elliptic equation on a coarse FEM grid since memory for an LDL^T decomposition will be reduced by a factor of eight.

V. Conclusions

Decoupling the bidomain equations into an elliptic and parabolic equation offer computational advantages which can be exploited to solve much larger problems in much less time. The elliptic problem was computationally more expensive as well as requiring more memory. CPU times were greatly reduced by solving it on a coarser spatial grid and at fewer instances in time while keeping errors within reasonable bounds. For solving the parabolic problem, the ICCM was approximately twice as fast as the FEM. Due to the sparsity of the problem, both the elliptic and parabolic problems scaled linearly with problem size. Finally, for problems which fit into memory, direct methods are two to three times faster than iterative methods.

VI. Acknowledgment

This work was supported by NSF Grants BES-9809132 and DMF-9709754, and NIH Grant HL63195.

References

[1] R. Plonsey, "Bioelectric sources arising in excitable fibers (ALZA lecture)," *Ann Biomed Eng.*, vol. 16, no. 5, pp. 519–546, 1988.

[2] N. G. Sepulveda, B. J. Roth, and J. P. Wikswo Jr, "Current injection into a two-dimensional anistropic bidomain," *Biophysical Journal*, vol. 55, pp. 987–999, 1989.

[3] J. P. Wikswo Jr, S.-F. Lin, and R. A. Abbas, "Virtual electrodes in cardiac tissue: a common mechanism for anodal and cathodal stimulation," *Biophysical Journal*, vol. 69, pp. 2195–2210, 1995.

[4] L. J. Leon and F. A. Roberge, "Structural complexity effects on transverse propagation in a two-dimensional model of myocardium," *IEEE Trans. Biomed. Eng.*, vol. 38, no. 10, pp. 997–1009, 1991.

[5] E. J. Vigmond and L. J. Leon, "A computationally efficient model for simulating electrical activity in cardiac tissue with fiber rotation," *Ann. Biomed. Eng.*, vol. 27, no. 2, pp. 160–170, 1999.

[6] E. J. Vigmond and L. J. Leon, "Effect of fibre rotation on the initiation of reentry," *Med. Biol. Eng. Comp.*, 2001. In press.

[7] I. R. Efimov, F. Aguel, Y. N. Cheng, B. Wollenzier, and N. A. Trayanova, "Virtual electrode polarization in the far field: implications for external defibrillation," *American Journal of Physiology*, vol. 279, pp. H1055–H1070, 2000.

[8] K. Skouibine, N. A. Trayanova, and P. K. Moore, "Anode/cathode make and break phenomena in a model of defibrillation," *IEEE Transactions on Biomedical Engineering*, vol. 46, pp. 769–777, July 1999.

[9] B. J. Roth, "Electrical conductivity values with the bidomain model of cardiac tissue," *IEEE Transactions on Biomedical Engineering*, vol. 44, pp. 326–328, 1997.

[10] A. George and J. Liu, *Computer Solutions of Large Sparse Positive Definite Systems*. Englewood Cliffs, NJ: Prentice Hall, 1981.

[11] G. Golub and C. Van Loan, *Matrix Computations*. London: Johns Hopkins, 3 ed., 1996.

[12] A. Nygren, C. Fiset, L. Firek, J. W. Clark, D. S. Lindblad, R. B. Clark, and W. R. Giles, "Mathematical model of an adult human atrial cell: the role of K+ currents in repolarization," *Circ. Res.*, vol. 82, pp. 63–81, 1998.