Large-scale Modeling of Cardiac Electrophysiology

JB Pormann[1], JA Board[1], DJ Rose[2], CS Henriquez[3]

[1]Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
[2]Department of Computer Science, Duke University, Durham, NC, USA
[3]Department of Biomedical Engineering, Duke University, Durham, NC, USA

Abstract

Simulation of wavefront propagation in the whole heart requires significant computational resources. The growth of cluster computing has made it possible to simulate very large scale problems in a lab environment. In this work, we present computational results of simulating a reaction diffusion system of equations of various sizes on a Beowulf cluster. To facilitate comparisons at different spatial resolutions, an idealized ventricular geometry was used. The model incorporates anisotropy, fiber rotation, and realistic membrane dynamics to determine the computational constraints for the most detailed situations of interest. Three meshes with mesh spacings of $378\mu m$, $238\mu m$, *and* $150\mu m$, *corresponding to roughly* $1M$, $4M$, *and* $16M$ *nodes in the computational domain, were considered. The results show that good parallel performance is possible on a cluster up to 32 processors.*

1. Introduction

Sophisticated computer simulations are being used to investigate the factors that generate and sustain life-threatening heart rhythms such as ventricular fibrillation. For models with domain sizes that approach the size of the human heart, the computational resources required to perform the simulation can exceed that found on a typical workstation. A domain comprising only 16M nodes with a membrane model of 5-8 state variables can use over 8GB of memory. To overcome these computational constraints, investigators have made use of commercial-class supercomputers such as the Cray YMP/T90 series and the IBM SP. While these supercomputers still provide outstanding performance, they are expensive and not always available to the average investigator.

An alternative to using commercial supercomputers is a Beowulf cluster that involves several closely networked workstations. Such clusters have become increasingly popular, but have been viewed as being too experimental to handle the algorithmic and communication demands involved in solving the reaction-diffusion equations used to model electrical dynamics in cardiac muscle. Recent advances in hardware and software, however, have made cluster computing attractive for large scale simulations. In this paper, we describe the computational performance of a model of wavefront propagation in domain size approximating the human heart. To investigate the computational needs and test the cluster environment, an idealized ventricular geometry, with non-uniform, rotational ansiotropy and various models of cardiac membrane ionic fluxes was used. The idealized geometry permitted different grids with the same shape but different elements sizes to be studied. The simulations were performed using CardioWave, a modular simulation system for the Bidomain Equations [1], developed in our laboratory. The results show that when using a distributed memory parallel approach, the computational and memory resources of multiple but otherwise independent workstations can be used efficiently up to 32 processors for a domain size of 16 million computational nodes.

2. Methods

CardioWave was developed to solve the system of bidomain equations on a parallel computer. The specific have been described previously [1]. Unlike other simulators of wavefront dynamics in the heart, Cardiowave is not a single, monolithic program, but rather a set of program modules related to time integration, output, membrane kinetics, matrix solvers etc, from which the user selects to create a custom executable for the problem of interest. In the typical simulation described in this work, the explicit-monodomain time-integrator module, the LR-1 membrane dynamics module, the simple stimulus and output modules were selected to create simulator. An additional module was selected to instruct the simulator that the execution would be performed in parallel. As such, the same set of modules can be selected and compiled on any distributed parallel computer without modification.

To test the parallel performance, an idealized geometry with anisotropic properties was used such that diffferent grid spacings could be considered while minimizing the

Table 1. Mesh Parameters

spacing (μm)	# Nodes (Tissue)	Memory (MB)
378	1035005	320
238	4138452	1279
150	16511310	5102

distortion to the domain shape. The ellipsoidal geometry is based on that given by Colli-Franzone [2] et al. In this model, the fiber angles are described analytically. The model includes both fiber rotation through the myocardial wall as well as the spiralling of the fibers from apex to base. Repeating the coordinate description from their paper:

$$\begin{aligned} x &= a(r)\cos\theta\cos\phi \\ y &= a(r)\cos\theta\sin\phi \\ z &= c(r)\sin\theta \end{aligned}$$

where $a(r) = 1.5 + 1.2r$ and $c(r) = 4.4 + 0.6r$. Coordinates, as well as a and c, are in cm; r goes from 0 to 1, endocardium to epicardium, respectively. For their model, they allowed θ to vary from $-3\pi/8$ to $\pi/8$. In our model, we based the cut-off on the z coordinate directly; z goes from 0 to 5, base to apex, respectively.

The computational domain is generated by laying a regular grid over a box surrounding the Colli-Franzone geometry. Each vertex in this regular-grid domain is then tested to see if it is inside or outside the surface of the ventricle. Those points that lie within the tissue are tagged for possible inclusion in the final output mesh. Only elements that are associated with 8 tagged nodes are included in the final output mesh. The resulting surface is a stair-step mesh approximating the ellipsoidal geometry.

Any discretization scheme [3] can have problems at edge and corner nodes when there is anisotropy and fiber curvature present. This can cause small current sources to appear at those points. Given that a stair-step grid has many such corners, we assigned isotropic conducitivies at the surface nodes only in order to attenuate this effect.

Three stairstep meshes were constructed that were each $4\times$ larger than its predecessor. The final mesh specifications are shown in Table 1. The mesh spacings 378 to 150 microns, and from 1 million to over 16 million nodes in size. The ”Memory” column in the table indicates the size of the output matrix-file that was stored on disk. The largest mesh occupied over 5 gigabytes of disk space. Although large, the mesh file only needs to be created once and can be used for other simulations (eg. different membrane models, different stimulus protocols, rotor formation, studies of re-entry, etc.).

Three membrance models were used: the Luo-Rudy (LR-1) model [**?**], the Pandit model [4] and the Rogers and McCulloch [5] model.

All simulations were performed on a custom ”Beowulf” cluster in our laboratory, developed by Linux Networx. This cluster consisted of 16 dual-CPU machines with AMD $1.53GHz$ processors and $1GB$ of memory each. All of the machines were connected by a gigabit-ethernet network. CardioWave uses the industry-standard MPI communications library; the free, open-source MPICH [6, 7] implementation from Argonne National Labs was installed on the cluster.

3. Results

A series of simulations were performed on the cluster using the LR-1 model on domains of various sizes. 1000 time steps were simulated to obtain the run-time values. Table 2 gives the run-time and memory usage results for the three meshes (each being $4\times$ larger than the next) on 1, 4, and 16 CPUs of our cluster. Given that a given processor had only $0.5GB$ of memory, all meshes would not fit on 1 or 4 CPUs (see Table 1 for a rough memory estimate of the mesh alone, without allocating the nodal voltages and state variables).

Analysis of Table 2 provides a measure of the overall parallel performance of the code. Going down the columns (problem size is fixed but the number of processors is increasing), we see a steady decline in run-time as expected. Going from 1 to 4 CPUs in the 1M mesh, for example, the code runs the same problem almost 3.4 times faster. This decrease in run-time is called the ”speed-up” of the parallel program. From 1 to 16 CPUs on the 1M mesh, we see a speed-up of nearly 13.6. From 4 to 16 CPUs on the 4M mesh, we see a speed-up of 3.8. The near linear speedup with the number of processors indicates that we are efficiently utilizing the CPU power of the extra machines. Additional simulations of the $150\mu m$ (16M) mesh with 32 CPUs ran in $3062.25sec$, nearly $1.84\times$ faster than the 16 CPU runs.

If we compare the 1 CPU-1M mesh time to the 4 CPU-4M mesh time, we see that they are nearly identical (less than 1% difference). This is to be expected since we have quadrupled the amount of work (1M to 4M nodes), but we have also quadrupled the amount of computational power. We can also compare these times to the 16 CPU-16M mesh case, where again we have increased the amount of work but exactly increased the computational power. All three run times are within 5% of each other indicating that we are efficiently using the additional memory of the extra machines.

3.1. Alternate membrane models

Since CardioWave uses a modular approach, simulations were also performed using different membrane models

Table 2. Parallel Performance Metrics

#	$378\mu m$ 1M Nodes		$238\mu m$ 4M Nodes		$150\mu m$ 16M Nodes	
Num. CPUs	Memory (MB/cpu)	Run-Time (sec)	Memory (MB/cpu)	Run-Time (sec)	Memory (MB/cpu)	Run-Time (sec)
1	489	5407.83	n/a	n/a	n/a	n/a
4	136	1595.39	489	5463.50	n/a	n/a
16	48	398.38	136	1430.21	488	5638.56

(Pandit [4], Rogers-McCulloch [5] (RM) model, and LR-1 with table look-up for the computationally expensive α and β terms in the state equations (referred to here as LR-look-up). Compiling a new simulator from scratch, with a new membrane model, required a negligible amount of time.

These three models represent several different conceptual approaches for describing the membrane ion fluxes. The RM model is a mathematical abstraction that uses only 1 state variable in addition to the voltage. It reproduces the qualitative behavior, and does not produce a very realistic transmembrane potential waveshape as output. The Pandit model is an experimentally derived model that utilizes 25 state variables to describe ion fluxes through the membrane, based on experimental patch clamp data. Clearly, solving the 25 state variable equations require significatly more computational work than with RM. The LR-look-up model is analagous to the LR model described previously, but precomputes the α and β terms for each ionic current. By creating a table of values and using lookup, the cost of performing the floating-point operations to compute the functions is reduced since only a few integer computations and memory references are required. As before, 1000 time steps are simulated. Only the largest mesh (16M nodes) was used on 16 processors of the cluster.

Due to its computational simplicity, the RM model required only $1336sec$ of run-time and $375MB$ of computer memory per processor. Thus, it is nearly $4.2\times$ faster to compute using RM than using the regular LR model, and uses about 30% less memory. This model is useful for quickly gaining insight into how geometry and tissue properties that might affect re-entry.

The Pandit model required approximately $10969sec$ of run-time and $791MB$ of memory per processor, or nearly $2\times$ slower than the LR model, and needing 62% more memory. Similar detailed membrane models give access to a number of ionic currents that could be used to investigate membrane biology, such as drug interactions on wavefront propagation.

Finally, the LR-look-up model required $2832sec$ of run-time, a nearly $2\times$ speed-up over regular LR, and yet required almost no extra memory (the tabulated data took approximately $200KB$, compared to $488MB$, per processor). While each model is somewhat unique, many current membrane models, including the Pandit model, could potentially use a similar look-up table approach at least for some of the internal functions.

4. Conclusions

The results demonstrate that cluster-based parallel computing can provide the computational resources to tackle large, realistic problems in cardiac electrophysiology. The simulation using the most detailed membrane model (the Pandit model) involved a grid with $150\mu m$ spatial resolution, full anisotropy and fiber rotation, This simulation needed approximately $14GB$ of computer memory, much more than is available on a commodity-class workstation. The use of parallel computational techniques allows us to efficiently spread the memory and computational load over multiple machines. For a fixed-size problem, we see speed-ups of 13.6 on 16 CPUs (85% of optimal). When we run larger problems, we see even better parallel efficiency (99% of optimal for the 16 CPU-16M run). It is important to emphasize that even with parallel processing, the simulation times are tractable but can still be very long. Assuming a fixed time step of $10\mu s$, a simulation of one second of activity for the most detailed model considered here (Pandit, 16M nodes and no lookup) would require about 7 days on 32 processors.

While constructed from a regular block. the stair-step mesh is stored internally as an irregular grid. Similarly, since the model incorporated fiber curvature and rotation, the conductivity tensor was computed and stored separately for each location in the grid, just as one would need for a truly anatomically accurate model. We adopted this approach since it represented a future direction for models where both the geometry and fiber descriptions are obtained from MRI data. The results show that with proper segmentation schemes, a very large-scale model with realistic geometry, realistic membrane dynamics, and full fiber rotation can be rapidly constructed at MRI resolution and simulated on 16-32 networked computers, a resource that is generally available in many laboratories.

Acknowledgements

This work was supported in part by the Duke-North Carolina NSF Grant DBI-99-74533, and National Institutes of Health Grant R29-HL57473.

References

[1] Pormann J. A Modular Simulation System for the Bidomain Equations. Ph.D. thesis, Duke University, 1999.

[2] Colli-Franzone P, Guerri L, Pennacchio M, Taccardi B. Spread of excitation in 3-d models of the anisotropic cardiac tissue, ii, effects of fiber architecture and ventricular geometry. Mathematical Biosciences 1998;147:131.

[3] Penland R, Harrild D, Henriquez C. Modeling impulse propagation and extracellular potential distributions in anisotropic cardiac tissue using a finite volume discretization. Computing and Visualization in Science in press 2002;.

[4] Pandit S, Clark R, Giles W, Demir S. A mathematical model of action potential heterogeneity in adult rat left ventricular myocytes. Biophysical J 2001;81:3029.

[5] Rogers J, McCulloch A. A collocation-galerkin finite element model of cardiac action potential propagation. IEEE Transactions on Biomedical Engineering 1994;41:743.

[6] Gropp W, Lusk E, Doss N, Skjellum A. A high-performance, portable implementation of the MPI message passing interface standard. Parallel Computing September 1996;22(6):789–828.

[7] Gropp W, Lusk E. User's Guide for `mpich`, a Portable Implementation of MPI. Mathematics and Computer Science Division, Argonne National Laboratory, 1996. ANL-96/6.

[8] Harrild D. Why Anatomy Matters: Normal Conduction and Flutter in a Computer Model of the Human Atria. Ph.D. thesis, Duke University, June 2000.

[9] Luo C, Rudy Y. A model of the ventricular action potential: depolarization, repolarization and their interaction. Circulation Research 1991;68:1501.

[10] Mpi: a message-passing interface standard. Tech Report Version 1.1, Message-Passing Interface Forum, June 1995.

[11] Pollard A, Burgess M, Spitzer K. Computer simulations of three-dimensional propagation in ventricular myocardium. Circulation Research 1993;72(4):744.

[12] Ng K, Saleheen H. Three-dimensional bidomain simulation of electrical propagation in cardiac tissue. In 17th IEEE Engineering in Medicine and Biology Conference. 1995; 1.1.5.3.

[13] Winslow R, Kimball A, Noble D, Denyer J, Varghese A. Simulation of very large sinus node and atrial cell networks on the connection machine cm-2 massively parallel computer. Journal of Physiology 1991;438:180P.

[14] Winslow R, Varghese A, Denyer J, Kimball A. Modeling large sa node-atrial cell networks on a massively parallel computer. Journal of Phiosiology 1992;446:242P.

[15] Winslow R, Kimball A, Varghese A, Noble D. Simulating cardiac sinus and atrial network dynamics on the connection machine. Physica D 1993;64:281.

[16] Stockbridge N. Solution of the hodgkin-huxley and cable equations on an array processor. Annals of Biomedical Engineering 1989;17:253.

[17] Veronese S, Othmer H. A computational study of wave propagation in a model for anisotropic cardiac ventricular tissue. In Proceedings of the 1995 International Conference on High Performance Computing and Networking. 1995; .

[18] Quan W, Evans S, Hastings H. Using domain decomposition and priority queue integration scheme to solve two-dimensional model of myocardium. In 17th IEEE Engineering in Medicine and Biology Conference. 1995; 1.1.5.2.

[19] Fishler M, Thakor N. A massively parallel computer model of propagation through a two-dimensional cardiac syncytium. PACE 1991;14:1694.

[20] Sepulveda N, Barach J, Wikswo J. A three-dimensional finite element bidomain model for cardiac tissue. In Proceedings of the 13th Annual IEEE/EMBS Conference. 1987; 311.

[21] Leon L, Horacek B. Computer model of excitation and recovery in the anisotropic myocardium. Journal Electrocardiology 1991;24:1.

[22] Keener J, Panfilov A. Models of Cardiac Cells and Tissues. W.B. Saunders Company, 1995; 335–348.

[23] Hodgkin A, Huxley A. A quantitative description of membrane current and its application to conductance and excitation in nerve. Journal of Physiology 1952;117:500.

[24] FitzHugh R. Mathematical models of threshold phenomena in the nerve membrane. Bulletin of Mathematical Biophysics 1955;17:257.

[25] Plonsey R, Barr R. Bioelectricity: A Quantitative Approach. New York: Plenum Press, 1988.

[26] Nygren A, Fiset C, Firek L, Clark J, Lindblad D, Clark R, Giles W. Mathematical model of an adult human atrial cell the role of k+ currents in repolarization. Circulation Research 1998;82:63.

[27] Pilkington T, Loftis B, JF Thompson ea. High Performance Computing in Biomedical Engineering. Boca Raton, FL: CRC Press, 1992.

[28] Saleheen H, Ng K. A new three-dimensional finite difference bidomain formulation for inhomogeneous anisotropic cardiac tissues. IEEE Trans BME 1998; 45(1):15.

[29] Pormann J, Board J, Rose D, Henriquez C. Automated membrane model creation. In Proceedings of 2000 Computers in Cardiology. 2000; 235.

Address for correspondence:

John B. Pormann
Deptartment of Electrical and Computer Engineering
Duke University
Box 90291
Durham, NC, USA 27708-0291
jpormann@ee.duke.edu