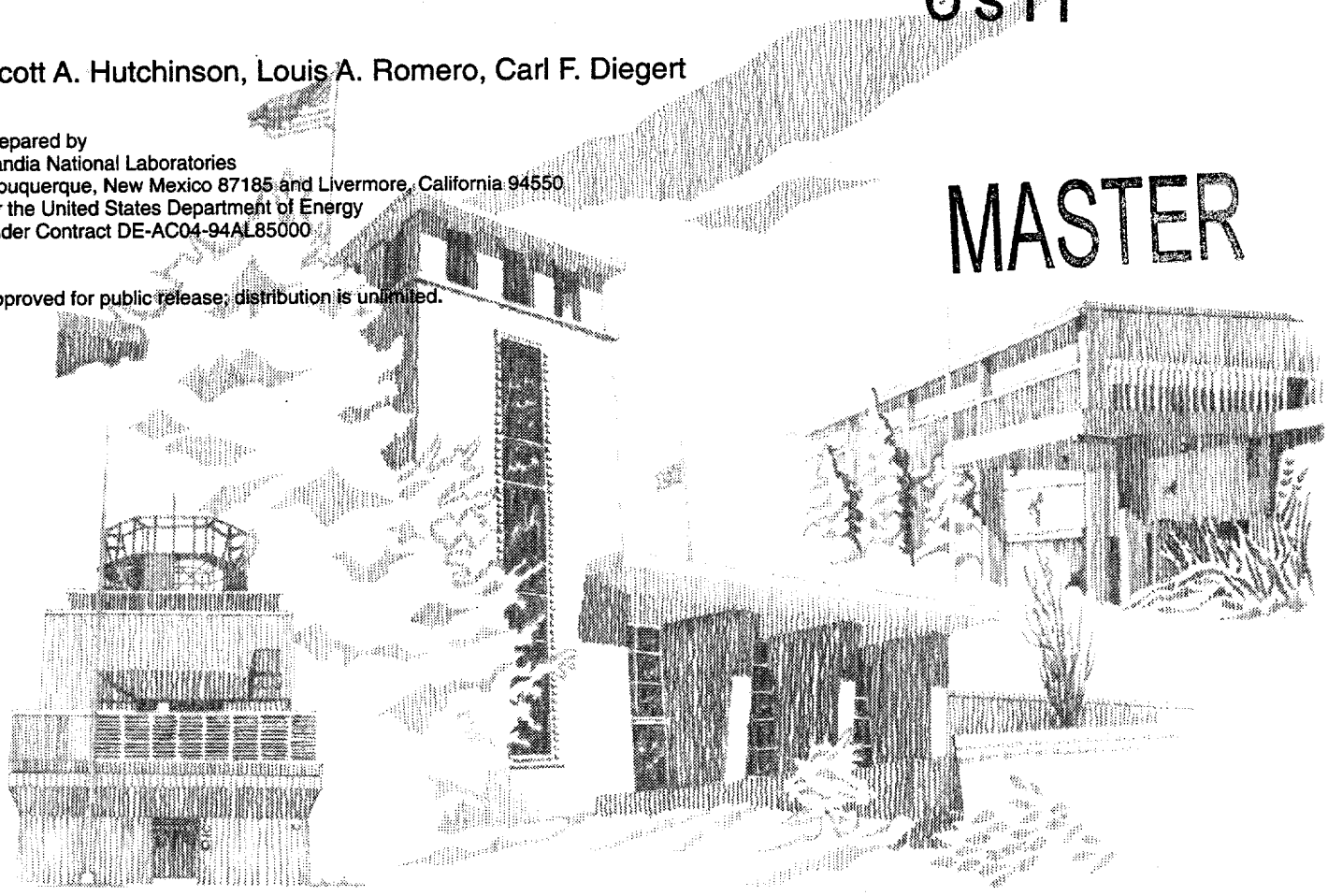
SANDIA REPORT

SAND96–1976 • UC–408
Unlimited Release
Printed August 1996

Advanced Computer Techniques for Inverse Modeling of Electric Current in Cardiac Tissue

Scott A. Hutchinson, Louis A. Romero, Carl F. Diegert

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000



SF2900Q(8-81)

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND96–1976
Unlimited Release

Printed August 1996

Distribution
Category UC–408

Advanced Computer Techniques for Inverse Modeling of Electric Current in Cardiac Tissue

Scott A. Hutchinson† Louis A. Romero§ Carl F. Diegert‡
Massively Parallel Computing Research Laboratory
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

For many years, ECGs and vectorcardiograms have been the tools of choice for non-invasive diagnosis of cardiac conduction problems, such as found in reentrant tachycardia or Wolff-Parkinson-White (WPW) syndrome. Through skillful analysis of these skin-surface measurements of cardiac generated electric currents, a physician can deduce the general location of heart conduction irregularities. Using a combination of high-fidelity geometry modeling, advanced mathematical algorithms and massively parallel computing, Sandia's approach would provide much more accurate information and thus allow the physician to pinpoint the source of an arrhythmia or abnormal conduction pathway.

* This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract No. DE-AC04-94AL85000.

† Parallel Computational Sciences Department; sahutch@cs.sandia.gov; (505) 845-7996
§ Applied & Numerical Mathematics Department; laromer@cs.sandia.gov; (505) 845-7512
‡ Computer Architectures Department; diegert@cs.sandia.gov; (505) 845-7193

Contents

1 Introduction

For many years, ECGs (electrocardiograms) and vectorcardiograms have been the tools of choice for non-invasive diagnosis of cardiac conduction problems. Through skillful analysis of these skin-surface measurements of cardiac generated electric currents, a physician can deduce the nature and general location of heart conduction irregularities. Once located, the offending tissue is often ablated in order to reestablish normal electrical function. However, there is a need for accuracy improvement since often more tissue than necessary is ablated in order to ensure that all the offending tissue is removed.

Using advanced computational mathematics and massively parallel (MP) computers, we planned to studied the correlation of the electric current measurements on the patients' body to electrical activity within the heart with greatly improved accuracy. Our approach would provide more accurate location information and allow the physician to pinpoint the source of an arrhythmia or an abnormal conduction pathway, such as in reentrant tachycardia or Wolff-Parkinson-White (WPW) syndrome. This would, in turn, reduce the invasiveness of the ablation procedure.

Our improved approach proceeded on three fronts. First, through the use of advanced imaging techniques, image processing and data segmentation algorithms, highly accurate, three-dimensional models of the conductive anatomy of the patients thorax were generated. This model was then used to generate a finite-element mesh for analysis. This lead to the second innovation – a novel algorithm for solving the inverse problem involved in determining electro-static potential on the surface of the myocardium given a finite number of measurements on the surface of the thoracic cavity. This algorithm was implemented in the MP finite-element code **MPSalsa** [1] for solution of the resulting fourth order PDE. The use of the MP code allowed for the solution of a much larger number of unknowns in the discretized system and thus promoted the use of high fidelity geometry models and improved overall solution accuracy. In the following description of the work we discuss the three areas on which significant advancements were made or were planned for the improved ECG inverse problem. We conclude with some general remarks on the project.

2 Geometry Acquisition, Segmentation and Meshing

To improve outcome and to reduce cost, a surgeon would like to know the location of a cardiac conduction defect before starting an intervention to correct the problem. Solving the inverse electrocardiographic problem from a patient's surface ECG based on an idealized model of human anatomy is not accurate enough to make a clinically-relevant estimation. By substituting information on the particular anatomy of the patient, we can set up a patient-specific inverse problem. Numerical solution of the specific problem may allow an estimate of conduction defect location that is accurate enough to be effective in planning the surgical intervention.

We have worked with the Department of Veterans Affairs in Albuquerque to obtain patient-specific anatomy information by magnetic-resonance imaging acquisitions and have developed methodology and software to reduce these data to the 3D anatomy geometries that define the boundaries of regions of the patient's anatomy with disparate electrical properties. Subsequent meshing of these regions and numerical solution of the inverse electrocardiographic problem then form a basis for estimating the location of any conduction defects. Within the scope of this project, we completed four acquisitions on the thorax of three different healthy volunteers using the Siemens "GBS III Magnetom" magnetic-resonance imaging equipment at the Department of Veterans Affairs in Albuquerque.

All of these acquisitions were gated to the cardiac cycle. As we gained experience, we

Date	Patient
13-APR-1994	Diegert, Carl
6-JUL-1994	Diegert, Carl
15-MAR-1994	Hutchinson, Scott
9-JUN-1994	Stearley, Jon

Figure 1: Magnetic-Resonance thoracic cavity image acquisitions

tuned the pulse sequence to yield better observations. The first two acquisitions observe the heart only at its rest (diastolic) phase. In the last two acquisitions, we were able to time the repeated measurements to observe the heart at twelve points in its cycle. All of these data are archived at Sandia National Laboratories, and are maintained by Carl Diegert.

Our methodology for finding the boundaries that divide the patient's anatomy into regions of electrically-similar material is described in two phases. First is a model building phase. The input to the model building phase is the collection of files that constitute the acquisition data from a series of magnetic-resonance observations. These data include multiple, spatially-sampled arrays of imaging information (12-bit reconstructed voxel values), as well as information on the geometry of these acquisitions (in-plane resolution, slice thickness, patient orientation, etc.). The output from the model building phase are lists of triangular facets that define the boundary of the patient's skin, the boundary of the lungs, the extent of the heart muscle, the extent of the ribs, etc. Our software writes these lists in any one of three data formats: (1) our own "dmp" format, (2) Advanced Visual Systems "geom" format, and/or (3) Microsoft SoftImage "hrc" format.

Figure 2 shows the boundary surfaces for Carl Diegert's skin, lungs, and heart tissue, as recovered from the magnetic-resonance acquisition performed on July 6, 1994.

The work flow and software tools that accomplish the model-building phase are:

- After the magnetic-resonance acquisition is complete, the digital data that comprise the acquisition are resident at the hospital on the same computer that controls the MR machine. During this project, we arranged for an Internet connection to the hospital's imaging center. The first step in the model-building phase is to use this connection to transfer the imaging data to the computers at Sandia National Laboratories.

- At Sandia, we then use our software to convert the digital data format used at the hospital to Sun Microsystem's "visualization file format" (program siemens2vff).

  The model-building phase continues with a less structured, highly interactive analysis of the dataset. During this analysis we rely heavily on collaboration with the clinically-trained staff at the hospital for guidance. The Internet connection speeds this interaction by enabling our tools for video conferencing, tools for shared whiteboard and shared tools for image annotation and editing. The tools we use to coax the needed boundaries from the observations fall into four categories:

  1. Our software for estimating iso-surface boundaries, with surface estimates given as lists of triangular facets (program vengine).
  2. Our software for filtering, simplifying (decimating) and organizing (connected component analysis) faceted geometries (program tmunch).
  3. A high-end "paint" program (program Matador from Parallax Software). We use this tool to create the mat images that guide the automatic segmentation

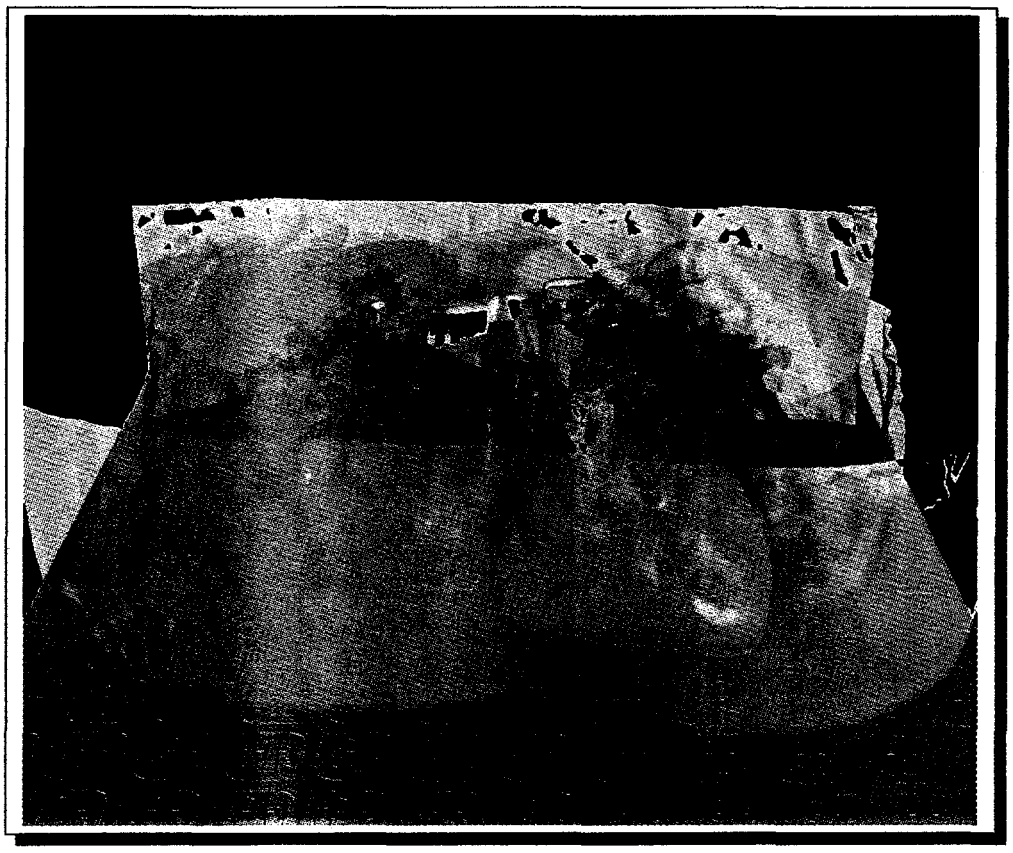


Figure 2: Example of Tissue Segmentation for Thoracic Cavity Showing heart in red, lungs in blue and skin in peach.

processing done by programs vengine and tmunch.

4. Our "plug-in" modules and the supporting high-end software tool for 3D content creation (the Softimage environment from Microsoft). We apply tools in this category to understand and to manipulate 3D boundary geometries.

After model-building is complete, we start a mesh-generation phase. Early in this project, we concluded that the free-form regions defined by patient anatomy like heart muscle and ribs, could not be automatically divided into hexahedral (brick) elements by any currently-available software package. We therefore decided to accept the extra difficulties that arise when the subsequent inverse problem is solved on a computational mesh formed from tetrahedral, rather than from hexahedral, elements.

Our experiments with a leading commercial CAD system showed that we could apply this software to automatically generate quality computational mesh datasets from the sorts of free-form boundary geometries we anticipated. That is, the meshing phase was to be:

- Apply our "plug-in" modules, together with a supporting, high-end CAD software tool to generate a computational mesh from the boundary geometries (the I-DEAS Master Series system from SDRC).

Unfortunately, we were unable to set up the large scale computational meshes that capture the detail inherent in our boundary geometries. The bottleneck proved to be the in the commercial CAD system: the commercial system was unable to accept the long lists of facet geometry that comprise our boundary geometries. We did write the plug-in modules that insert this information directly into a executing copy of the CAD system, thus using the most efficient mechanism for communicating with this software. However, when we used this mechanism to feed a few thousand facets of boundary to the CAD process, the CAD software accepted the geometry, then died. Whatever processing the CAD software did to verify the (already correct) topology of our facet list simply took too long and tripped timers that other parts of the CAD system use to automatically recover from errors. We could use our plug-in module to load and mesh regions defined by short lists of facets, as this protective time-out did not kick in. When feeding what we consider geometries of modest size – a few thousand facets – to the CAD software, the software decided that its own processing was taking so long that there must be something wrong. In this case, the CAD package neatly closed up shop, without producing any computational mesh. Support from the vendor under their third-party developers program confirmed the problem, but did not offer any workable means of overcoming it.

By developing a technology for giving our estimate of boundary geometry as a collection of NURB patches, we believe that the mesh generation phase using the same commercial CAD software tool would become workable. In fact, it was by experimenting with using the commercial tool to mesh this sort of geometry that we settled on the whole approach. At that time we did not understand the high overhead that lies deep within the commercial software and is associated with each facet it accepts. To use this software effectively on complicated geometries, one must make each patch represent much more than a tiny, flat triangular facet. That is, the commercial package offers no special-case processing to economically process a great many simple flat facets. Instead, the commercial package substitutes a high-overhead, degenerate NURB patch for each and every flat facet we feed it. Our early experiments with the commercial package were correct in that the package can effectively mesh complex geometries. What we learned is that these geometries must be given as a short list of NURB patches, rather that long lists of simple, flat facets.

Giving estimates of boundaries as a collection of NURB patches is a difficult problem that we could not solve in the context of this project. One author (Carl Diegert) has received subsequent funding under DOE's Math and Information Computer Sciences (MICS) program to pursue the estimation problem. If he can develop a workable method for this sort of estimation, we would like to return to the inverse electrocardiography application and construct some clinically-useful estimates of conduction defect location.

3 Inverse Problems in Electro-statics Using Fourth Order Partial Differential Equations

Suppose we are interested in solving Laplace's equation in a region V that has an exterior boundary Ω_2, and an interior boundary Ω_1. Suppose also that we know both the function ϕ and its normal derivative ϕ_n on the exterior boundary but have no information on the interior boundary:

$$\nabla^2\phi = 0 \ \ for \ \ \mathbf{x} \in V$$
$$\phi = f \ \ on \ \ \Omega_2$$
$$\phi_n = g \ \ on \ \ \Omega_2$$

This is an ill-posed problem. The ill-posedness arises from the fact that although in theory one could determine the function ϕ from this data, we could perturb the data f and g by an arbitrarily small amount and change our answer by a finite amount. This is caused by the unbounded growth of the high frequencies.

Although this is a mathematically ill-posed problem, one can still expect to be able to find approximate solutions to this problem if one does not demand too much information from the high frequencies. People have solved problems like this by discretizing the system of equations to get a linear system of equations, and then performing a singular value decomposition or regularization of the resulting matrix [2, 3, 4, 5]. In the former case, the linear system is approximated numerically, but the information associated with the smallest singular values is given very little weight. In the latter method, the governing PDE is recast into a well-posed one through the addition of terms.

These methods are not easily implemented using standard finite element codes because of the difficulty of obtaining a singular value decomposition of very large sparse matrices as well as difficulties associated with discrete regularization approaches. We propose a technique that avoids this difficulty by discretizing a well-posed partial differential equation that approximates the solution to the ill-posed problem. This differential equation is motivated by the following variational problem,

$$\min I(\phi) = \frac{1}{2}\int_V \mid \nabla^2\phi \mid^2 + \lambda\phi^2 \, dv$$

subject to the constraints

$$\phi = f \ \text{ on } \ \Omega_2$$
$$\phi_n = g \ \text{ on } \ \Omega_2$$

As the parameter λ approaches zero, the solution to this variational problem will very nearly satisfy Laplace's equation and will continue to satisfy the boundary conditions. Suppose that ϕ is a solution to this variational problem and ψ is an arbitrary perturbation to this solution. To first order we have

$$\delta I = I(\phi + \psi) - I(\phi) = \int_V \nabla^2\phi\nabla^2\psi + \lambda\phi\psi \, dv + \ldots$$

We now use the identity

$$\nabla^2\phi\nabla^2\psi = \nabla\cdot\left(\nabla\psi\nabla^2\phi\right) - \nabla\cdot\left(\psi\nabla\nabla^2\phi\right) + \psi\Delta^2\phi$$

along with Green's identity to show that

$$\Delta I = \int_V \psi\left(\delta^2\phi + \lambda\phi\right)\, dx + \int_{\Omega_1} \nabla^2\phi\psi_n - \psi\frac{\partial\nabla^2\phi}{\partial n}\, ds$$

Here we have assumed that $\psi = \psi_n = 0$ on Ω_2 since ϕ and ϕ_n are specified on that surface.

In order to have δI vanish for all ψ we must have

$$\Delta^2\phi + \lambda\phi = 0 \tag{1a}$$

$$\phi = f \text{ on } \Omega_2 \tag{1b}$$

$$\phi_n = g \text{ on } \Omega_2 \tag{1c}$$

$$\frac{\partial\nabla^2\phi}{\partial n} = 0 \text{ on } \Omega_1 \tag{1d}$$

$$\nabla^2\phi = 0 \text{ on } \Omega_1 \tag{1e}$$

We now argue that this is in fact a well-posed system. To do this we need to show that the solutions exist, are unique, and that they depend continuously on the data. The existence and uniqueness are related to each other through Fredholms alternative. For a finite dimensional space if we have an $n \times n$ system of equations $\mathbf{Ax} = \mathbf{b}$, we can solve this for all $\mathbf{b}$ if and only if there is no nontrivial solution to $\mathbf{Ax} = 0$. For a finite dimensional system showing uniqueness is equivalent to showing the existence of the solutions. We are dealing with an infinite dimensional space, so this argument requires some delicate functional analysis. We will not supply the functional analysis, but will note that the functional analysis works for all of the commonly used elliptic equations, so we doubt that it will fail here. With all of this in mind we now show that this system of equations has no non-trivial solutions when $f = g = 0$.

To do this we multiply the partial differential equation by ϕ and integrate over V to get

$$\int_V \left(\Delta^2\phi + \lambda\phi\right)\phi\, dv = 0$$

Applying the same integration by parts that we used in our variational problem we find that

$$\int_V \mid \nabla^2\phi \mid^2 + \lambda\phi^2\, dv = 0$$

If $\lambda > 0$ we conclude that $\phi \equiv 0$.

In order to show that this problem depends continuously on the initial data, it is sufficient to show that it is not ultra-sensitive to the high frequencies. In order to do this it is sufficient to show that we can actually solve our system of equations with our boundary conditions specified on a plane. For example, we need to show that we can solve the system of equations

$$\Delta^2\phi + \lambda\phi = 0 \text{ for } z > 0$$

$$\phi(x,y,0) = f(x,y)$$

$$\frac{\partial\phi(x,y,0)}{\partial z} = g(x,y)$$

$$\phi(x, y, z) \to 0 \text{ as } z \to \infty$$

It is a straightforward matter to solve this system of equations using Fourier transforms. We can similarly solve the system of equations with the boundary conditions specified at the surface Ω_1. The conclusion is that out system of equations with $\lambda > 0$ is a well-posed system.

4 Massively Parallel Finite-Element Solution

Here the overall parallel implementation framework for an unstructured finite-element discretization and solution approach to the equations developed above is described. This implementation is capable of approximating solutions to the partial differential equations (PDE's) in complex 2D and 3D geometries. The focus in the development of this code has been primarily directed towards both the FEM implementation of (1) and the parallel performance of important kernel operations in such MP unstructured applications. These include the distributed sparse matrix creation and the kernels of the iterative solution methods based on preconditioned Krylov techniques.

4.1 Finite-Element Approximation

Equation (1) is a fourth order system and FEM solutions of this system require C^1 continuous elements – elements which are not efficient in two-dimensions and nonexistent in three. Thus, we reduce the order by introducing an auxiliary variable ξ such that

$$\nabla^2 \phi = \xi$$

which allows us to express (1) as

$$\nabla^2 \phi - \xi = 0 \tag{2a}$$

$$\nabla^2 \xi + \lambda \phi = 0 \tag{2b}$$

$$\phi = f \text{ on } \Omega_2 \tag{2c}$$

$$\phi_n = g \text{ on } \Omega_2 \tag{2d}$$

$$\xi_n = 0 \text{ on } \Omega_1 \tag{2e}$$

$$\xi = 0 \text{ on } \Omega_1 \tag{2f}$$

where we have gone from a single fourth order PDE to two second order PDE's, resulting in an additional unknown. Notice also that the boundary conditions have changed.

Using a standard Galerkin FEM approximation to (2), we approximate a general unknown ϕ with

$$\phi = \sum_{J=1}^{n} \phi_J \Psi_J(x, y, z)$$

where $\Psi_J(x, y, z)$ is a global representation of the FEM basis function at the node J. This is a fairly standard procedure, which, when combined with Green's theorem reduces the order of an individual PDE by one. For further information on how this is implemented, see [1]. Ultimately, this results in a sparse system of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3}$$

where the components of the vector $\mathbf{x} \in \mathbf{R}^n$ are the global values of the unknowns ϕ and ξ at the nodal points (or, at least, approximations thereof).

Another significant issue with this implementation is the difficulty in implementing the boundary conditions. Notice in (2) that we have both a Dirichlet and a Neumann boundary condition applied over both boundaries. While this is perfectly acceptable mathematically, it is difficult to implement in a FEM approximation using the standard techniques.

4.2 Finite-Element Mesh Partitioning for MP Computing

Considering a typical complex unstructured grid as in Figure 3 the following comments about the general structure of a parallel FEM implementation can be made. First, it is evident from Figure 3 that a general, automated method for subdividing and assigning an unstructured computational mesh to the MIMD computer is necessary. An ad-hoc method would prove unusable for a large class of complicated meshes and the resulting parallel communication efficiency would be difficult to predict, access and control. For the implementation described here, a general graph-partitioning utility for unstructured meshes [6] developed at Sandia National Laboratories is used. With this utility the basic parallel MIMD implementation on p processors is as follows. Consistent with the choice of the nodal based FEM scheme, it is necessary to *load balance* the work associated with the matrix setup and solution by partitioning the n nodes among the p processors[1]. The load balance algorithm partitions the n nodes into p sets of $\lfloor n/p \rfloor$ or $\lfloor n/p + 1 \rfloor$ FEM nodes[2] and then maps these sets to the p processors of the parallel computer such that the overall interprocessor communication cost is minimized. After the required load balance and mapping, the FEM code can then set up the distributed interaction (coefficient) matrix. This distributed sparse matrix roughly corresponds to a rectangular submatrix of the global interaction matrix. Thus, each processor, in parallel, performs the necessary element integrations and constructs a local set of equations for each of the nodes for which it is responsible. In this implementation, these equations are fully summed and actually correspond to a complete row of the corresponding global interaction matrix. The union of these distributed rectangular submatrices is therefore equivalent to a serial global interaction matrix.

After construction of the distributed sparse matrix, a library of parallel Krylov solvers (PCG in this case) is called. On each processor, the solvers operate on the local sparse matrix and local solution vector using a combination of global structured and unstructured communication to achieve the parallel solver kernels.

4.3 Parallel CG Solver and Distributed Sparse Matrix Data Structures

Here, we discuss MP performance issues associated with the Krylov-subspace methods used for iterative solution of the sparse, linear equations given in (3). The main kernels are the matrix-vector product, DAXPY operations and vector inner products. The key to performance in these solvers is typically the efficiency of the matrix-vector multiply kernel and, within this, the communication it requires. For these parallel kernels to operate as efficiently as possible, the interprocessor communication times during the required matrix-vector multiplies must be minimized. In turn, key to this minimization is the data structures in which both the distributed sparse matrix and the distributed vectors are stored.

Logically, each processor is given a set of nodes for which it is responsible. Thus, in the formation of the sparse matrix and vectors, each processor will have a set of rows in both the sparse matrix and any associated vectors, each corresponding to unknowns located at "its"

[1] Other choices such as element based schemes are possible. The relative performance of such schemes should not vary greatly for reasonable implementations of either the nodal or the elemental schemes.

[2] $\lfloor a \rfloor$ is the floor function which returns the largest integer, $m \in \mathbf{I}$ such that $m < a$.

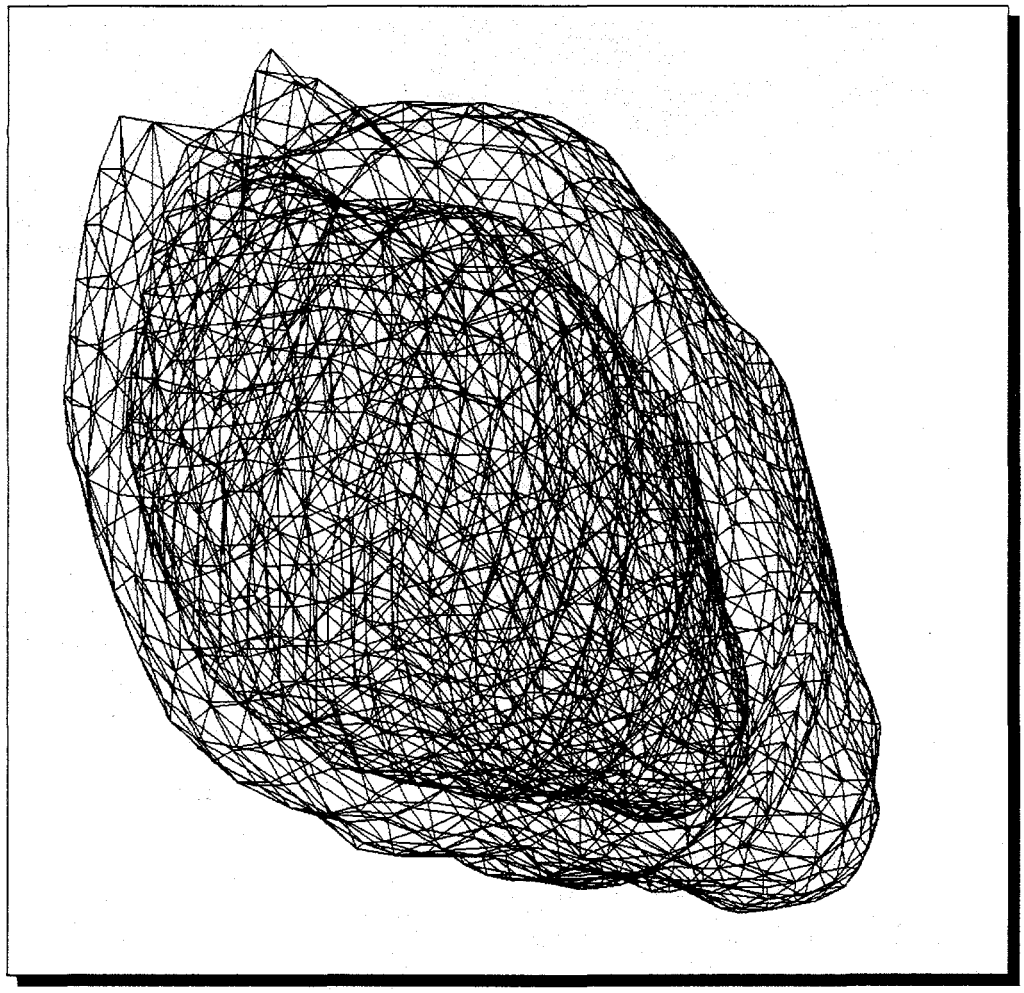


Figure 3: Canine heart unstructured mesh.

nodes. We let $\mathbf{x}_l \in \mathbf{R}^{n_l}$ be the vector of unknowns for which processor l is responsible and n_l is the number of these unknowns. In addition, assume that these unknowns exist at nodes which make up a continuous partition of the graph of the mesh. Then, in order to complete the matrix-vector product, processor l will need additional values of $\mathbf{x}$ which reside on other neighboring processors. These values are required to complete the interactions between processor l's "border" unknowns and its "external" unknowns. That is, if $\mathbf{x}_{lb} \subset \mathbf{x}_l$ are the border unknowns of processor l, these unknowns interact with border unknowns on neighboring processors via the connectivity of the FEM mesh. These border unknowns on neighboring processors are referred to as processor l's external unknowns. Figure 4 gives an illustration of this partitioning.

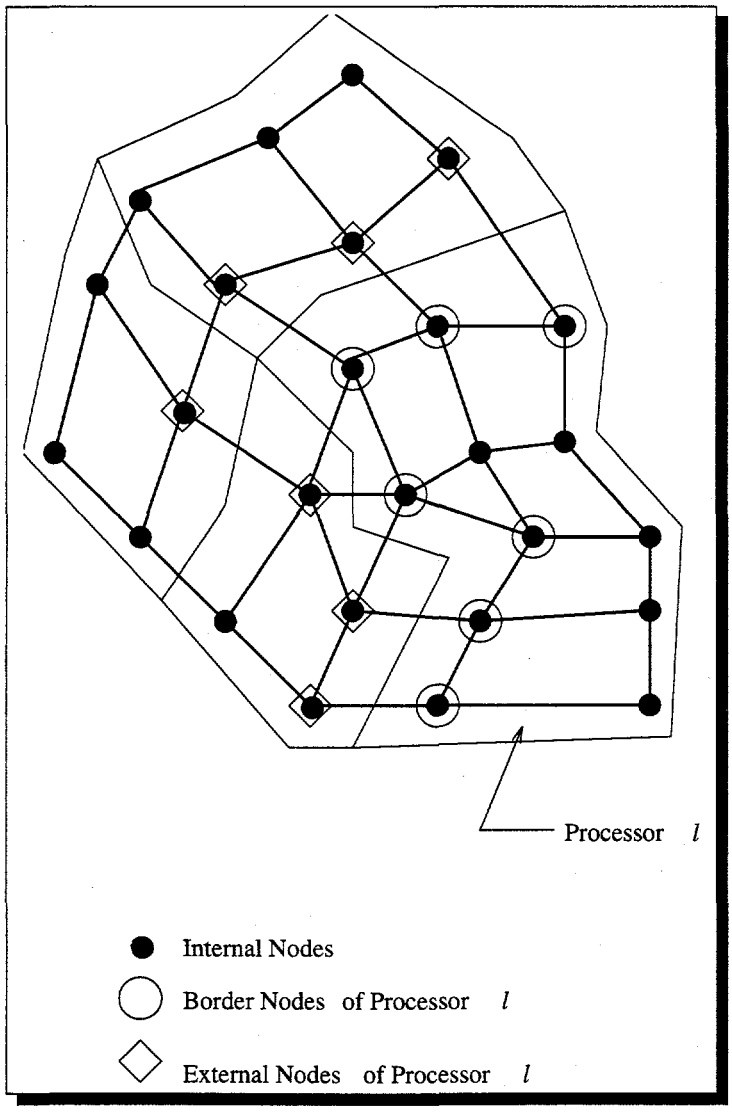


Figure 4: Unstructured mesh partitioning illustrating the internal, border and external nodes

Given the partitionings and unknown-distinctions (internal, border and external) described above, a distributed sparse matrix storage scheme [7] is used. On each processor, the node numbers are reordered such that the first n_{in} nodes are the internal nodes of the processor, the next n_b are the border nodes and the last n_e are the external nodes. Also, the aforementioned n_l is given as $n_l = n_{in} + n_b$. Thus, the last n_e members of the vector must be obtained by communicating with their "owning" processors. Once this portion of the vector is filled, the matrix-vector product may be computed.

Note that the number of messages is equal to the number of neighboring processors and that the size of each message is directly proportional to the number of external nodes associated with the specific processor. Further, if the processor is not a physical neighbor, the message may have to traverse several processors in order to arrive at its destination. These three factors, number of messages, size of each message and the distance each message must travel, are critical in determining the speed of the matrix-vector products. All of these factors are influenced to some degree by the load balance of the problem. For a more complete description of the load balance methods available through the code used in this study, see [6].

5 Remarks

This work was overtaken from a project whose PI had left Sandia and was refocused approximately one year into the grant. While the new method has shown significant promise in simple, test problems, difficulties arose on two fronts which prevented the timely completion of the project. First, obtaining quality solid models for meshing was never completed and thus accurate geometry information for a true ECG solution could not have been performed. Second, as mentioned above, implementation of the boundary conditions was difficult and while the work was progressing, it was never completed.

Further funding for the project was sought but, given Sandia's change in focus away from biomedical research, none was forthcoming. The authors still feel strongly that this work has the capability of improving standard ECG results and should be pursued.

References

[1] John N. Shadid, Harry K. Moffat, Scott A. Hutchinson, Gary L. Hennigan, Karen D. Devine, and Andrew G. Salinger. **MPSalsa**, A finite element computer program for reacting flow problems. Part 1 – Theoretical Development. Technical Report SAND95-2572, Sandia National Laboratories, Albuquerque, NM, 1996.

[2] P.C. Stanley and T.C. Pilkington. The combination method: A numerical technique for electrocardiographic calculations. *IEEE Trans. Biomed. Eng.*, 36(4):456–461, April 1989.

[3] A. Vahid Shahidi and Pierre Savard. Forward and inverse problems of electrocariography: Modeling and recovery of epicardial potential in humans. *IEEE Trans. Biomed. Eng.*, 41(3):249–256, March 1994.

[4] Christopher R. Johnson, Robert S. MacLeod, and Philip R. Ershler. A computer model for the study of electrical current flow in the human thorax. *Comput. Biol. Med.*, 22(5):305–323, 1992.

[5] Robert D. Throne and Lorraine G. Olson. A generalized eigensystem approach to the inverse problem of electrocardiography. *IEEE Trans. Biomed. Eng.*, 41(6):592–600, June 1994.

[6] B. Hendrickson and R. Leland. The Chaco user's guide, Version 1.0. Technical Report SAND93-2339, Sandia National Laboratories, Albuquerque, NM, 1993.

[7] J.N. Shadid and R.S. Tuminaro. A comparison of preconditioned nonsymmetric krylov methods on a large–scale MIMD machine. Technical Report SAND91-0333, Sandia National Laboratories, Albuquerque, NM, 1991.

INTERNAL DISTRIBUTION:

1	MS 1111	Sudip Dosanjh, 9221
5	MS 1111	Scott Hutchinson, 9221
2	MS 1110	Louis Romero, 9222
2	MS 1109	Carl Diegert, 9215
1	MS 1436	LDRD Office, 4523
1	MS 9018	Central Technical Files, 8523-2
5	MS 0899	Technical Library, 4414
2	MS 0619	Review and Approval Desk, 12630 For DOE/OSTI
3	MS 0161	Patent and Licensing Office, 11500