Effects of adaptive refinement on the inverse EEG solution

David M. Weinstein, Christopher R. Johnson and John A. Schmidt

Department of Computer Science,
University of Utah, Salt Lake City, UT 84112,
dweinste@cs.utah.edu, crj@cs.utah.edu, jas@cs.utah,edu.

ABSTRACT

One of the fundamental problems in electroencephalography can be characterized by an inverse problem. Given a subset of electrostatic potentials measured on the surface of the scalp and the geometry and conductivity properties within the head, calculate the current vectors and potential fields within the cerebrum. Mathematically the generalized EEG problem can be stated as solving Poisson's equation of electrical conduction for the primary current sources. The resulting problem is mathematically ill-posed i.e., the solution does not depend continuously on the data, such that small errors in the measurement of the voltages on the scalp can yield unbounded errors in the solution, and, for the general treatment of a solution of Poisson's equation, the solution is non-unique. However, if accurate solutions to such problems could be obtained, neurologists would gain non-invasive access to patient-specific cortical activity. Access to such data would ultimately increase the number of patients who could be effectively treated for pathological cortical conditions such as temporal lobe epilepsy.

In this paper, we present the effects of spatial adaptive refinement on the inverse EEG problem and show that the use of adaptive methods allow for significantly better estimates of electric and potential fields within the brain through an inverse procedure. To test these methods, we have constructed several finite element head models from magnetic resonance images (MRI) of a patient. The finite element meshes ranged in size from 2724 nodes and 12,812 elements to 5224 nodes and 29,135 tetrahedral elements, depending on the level of discretization. We show that an adaptive meshing algorithm minimizes the error in the forward problem due to spatial discretization and thus increases the accuracy of the inverse solution.

Keywords: inverse EEG, regularization, finite element method, adaptive refinement

1. INTRODUCTION

The electrical activity on and within the cerebral cortex gives rise to electric, and thus potential, fields within the volume of the head and upon the scalp surface. The voltages on the scalp are related to voltages on the cerebral cortex via the resistive properties of the intermediary tissues (volume conductor). The general inverse problem in electroencephalography can be stated as follows: given a subset of electrostatic potentials measured on the scalp and the geometry and conductivity properties of the head, calculate the potential fields on, and source currents, within, the cerebrum. Mathematically this can be posed as an inverse source problem in terms of the primary current sources within the brain. Poisson's equation of electrical conduction is the governing equation:

$$\nabla \cdot \sigma \nabla \Phi = -I_v \quad \text{in } \Omega \tag{1}$$

with the boundary condition:

$$\sigma \nabla \Phi \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_S \tag{2}$$

where Φ are the electrostatic potentials, σ are the electrical conductivity tensors, I_v are the cerebral current sources per unit volume, and Γ_S and Ω represent the scalp and head volume respectively. In general, (1) does not have a unique solution. In order to solve the general source problem, one usually breaks up the cerebral volume into subregions, within which simplifying assumptions are made regarding the form of the source term (such as dipoles or other more complex sources models). One then attempts to recover magnitude and directional information of the simplified model sources. Alternatively, if instead of solving (1) in terms of the current sources within the brain, one solves for the electrostatic potentials on a surface bounding the brain, this form of the problem does have a unique solution. Thus instead of solving Poisson's equation we solve a generalized Laplace's equation with Cauchy boundary conditions:

$$\nabla \cdot \sigma \nabla \Phi = 0 \quad \text{in } \Omega \tag{3}$$

with boundary conditions:

$$\Phi = \Phi_0 \quad \text{on } \Sigma \subseteq \Gamma_S \quad \text{and} \quad \sigma \nabla \Phi \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_S \tag{4}$$

While this version of the inverse problem has a unique solution,[1] the problem is still mathematically ill-posed in the Hadamard sense; i.e., the solution does not depend continuously on the data, such that small errors in the measurement of the voltages on the scalp can yield unbounded errors in the solution.

An accurate solution to the inverse problem in electroencephalography would provide a non-invasive procedure for functionally mapping the cortex. Traditional efforts of non-invasive cortical mapping have used simplified geometries such as concentric spheres to approximate the complex physical domains of the scalp, skull, cerebrospinal fluid, grey matter and white matter. While these methods yield a good first approximation to the true cortical potentials for simple cases involving a single dipole source, they are not accurate enough for many clinical applications.[2] An accurate inverse solution would also provide a non-invasive means for localizing and analyzing the activity of ectopic foci in epileptic patients,[3] thereby providing neurologists reliable data on which to diagnose and treat patients.

2. MATHEMATICAL THEORY

2.1 Finite element approximation

In order to solve (1) or (3) in terms of the cerebrocortical potentials, we need to pose the problem in a computationally tractable way. This involves approximating the boundary value problem on a finite dimensional subspace and reformulating it in terms of a linear matrix equation. For this study we utilize the finite element method (FEM) to approximate the field equation and construct the set of matrix equations. Briefly, one starts with the Galerkin formulation of (1); note that this includes the Dirichlet and Neumann boundary conditions,

$$(\sigma \nabla \Phi, \nabla \overline{\Phi}) = -(I_v, \overline{\Phi}) \tag{5}$$

where $\overline{\Phi}$ is an arbitrary test function, which can be thought of physically as a virtual potential field, and the notation, $(\phi_1, \phi_2) \equiv \int_\Omega \phi_1 \phi_2 \, d\Omega$, denotes the inner product in $L_2(\Omega)$. We now use the finite element method to turn the continuous problem into a discrete formulation. First we discretize the solution domain, $\Omega = \bigcup_{c=1}^{N} \Omega_c$, and define a finite dimensional subspace, $V_h \subset V = \{\overline{\Phi} : \overline{\Phi} \text{ is continuous on } \Omega, \nabla \overline{\Phi} \text{ is piecewise continuous on } \Omega\}$. We define parameters of the function $\overline{\Phi} \in V_h$ at node points, $\alpha_i = \overline{\Phi}(x_i), i = 1, 2, \ldots, N$ and define the basis functions, $\Psi_i \in V_h$ as linear piecewise continuous functions that take the value 1 at node points and 0 elsewhere. We can then represent the function, $\overline{\Phi} \in V_h$ as:

$$\overline{\Phi} = \sum_{i=1}^{N} \alpha_i \Psi(x_i) \tag{6}$$

such that $\overline{\Phi} \in V_h$ can be written in a unique way as a linear combination of the basis functions, $\Psi_i \in V_h$. The finite element approximation of the original boundary value problem, (1) can be stated as:

$$\text{Find } \Phi_h \in V_h \text{ such that } (\sigma \nabla \Phi_h, \nabla \overline{\Phi}) = -(I_v, \overline{\Phi}) \tag{7}$$

Furthermore, since $\Phi_h \in V_h$ satisfies (7), then we have $(\sigma \nabla \Phi_h, \nabla \Psi) = -(I_v, \Psi_i)$. Finally, since Φ_h itself can be expressed as the linear combination:

$$\Phi_h = \sum_{i=1}^{N} \xi_i \Psi_i(x) \quad \xi_i = \Phi_h(x_i) \tag{8}$$

we can then write (7) as:

$$\sum_{i=1}^{N} \xi_i (\sigma_{ij} \nabla \Psi_i, \nabla \Psi_j) = -(I_v, \Psi_j) \quad j = 1, \ldots, N \tag{9}$$

The finite element approximation of (1) can equivalently be expressed as a system of N equations with N unknowns, $\xi_1, \ldots, \xi_N$ (the electrostatic potentials). In matrix form, the above system is expressed as, $A\xi = b$, where $A = (a_{ij})$ is the global stiffness matrix and has elements $(a_{ij} = (\sigma_{ij}\nabla\Psi_i, \nabla\Psi_j))$, while $b_i = -(I_v, \Psi_i)$ is the vector of source contributions.

The matrix A contains all the geometry and conductivity information and is symmetric and positive definite. Because the basis function differs from zero for only a few intervals, A is very sparse. These features of A significantly affect the strategy for solving the system of equations and will be exploited later in the section on computations.

2.2 Adaptive methods

Discrete approaches to bioelectric inverse and imaging problems have usually centered around classical numerical techniques for solving partial differential equations: finite difference, finite element, and boundary element methods. To date, instead of developing and utilizing specific, quantitative methods, most scientists have assigned spatial discretization levels by relying largely on previous experience and their common sense, utilizing what has apparently worked before and perhaps reducing the discretization level in regions *a priorily* known to contain high gradients. This situation has arisen, at least in part, because of the computational load represented by the large, complex, inhomogeneous, even anisotropic geometries that characterize many problems in bioelectric field imaging.

We have found, however, that by using *a posteriori* estimates from the finite element approximation of the governing equations, we can locally refine the mesh discretization and reduce the errors in the direct solution. It has been assumed — and our findings support this notion — that improving the accuracy of the direct solution also improves the subsequent inverse solution. The novel aspect of our approach is that it uses local approximations of the error in the numerical solutions to drive an automatic adaptive mesh refinement. The basis of the error estimations comes from recent research on the finite element method.[4]

The essential feature of the finite element method is that the approximate forms of the scalar field satisfies the governing equation in each element in some weighted sense. If we reduce the size of the elements, *h-refinement*, or increase the order of the basis function, *p-refinement*, we increase the accuracy of the approximation.[5] While either approach can be applied globally, computational limits make it more efficient to apply refinement locally, to regions where it is deemed most beneficial. One way to monitor the overall effect of refinement is to compute the total energy, which must converge monotonically if refinement is progressing effectively.

For this study, we improve our mesh via an adaptive h-refinement technique based on an estimate of the element energy error, derived from methods suggested by Lewis.[6]

The error in the potential, e_Φ, is defined as the difference between the exact potential, Φ, and the calculated potential, $\hat{\Phi}$.

$$e_\Phi = \Phi - \hat{\Phi} \tag{10}$$

Similarly, the error in the gradient of the potential (electric field), q, is

$$e_q = q - \hat{q} \tag{11}$$

where $q = \nabla\Phi$ and $\hat{q} = \nabla\hat{\Phi}$. The error norm is defined as:

$$||e_q|| = (\int_\Omega (\nabla\Phi - \nabla\hat{\Phi})^T \sigma(\nabla\Phi - \nabla\hat{\Phi}) d\Omega)^{\frac{1}{2}}. \tag{12}$$

Zienkiewicz[7,8] has shown that

$$\int_\Omega (\nabla\Phi)^T \sigma(\nabla\hat{\Phi}) d\Omega = \int_\Omega (\nabla\hat{\Phi})^T \sigma(\nabla\Phi) d\Omega = \int_\Omega (\nabla\hat{\Phi})^T \sigma(\nabla\hat{\Phi}) d\Omega. \tag{13}$$

Using this result, (12) becomes

$$||e_q||^2 = \int_\Omega (\nabla\Phi)^T \sigma(\nabla\Phi) d\Omega - \int_\Omega (\nabla\hat{\Phi})^T \sigma(\nabla\hat{\Phi}) d\Omega \tag{14}$$

or equivalently,

$$||e_q||^2 = ||q||^2 - ||\hat{q}||^2 \tag{15}$$

Here, $||q||^2$ is a measure of the total energy in the domain. Using this, the percentage error, η, can be defined to be

$$\eta = \frac{||e_q||}{||q||} \cdot 100\% \tag{16}$$

This error measure gives a ratio of the energy difference and the total energy which is in effect a percentage error for each element.

In this study, we use linear basis functions to describe the variation of the potential in each tetrahedral element. At every point in the mesh, the potential is uniquely specified. However, the gradient is not specified at every point in space but is defined to be constant over each element and discontinuous across element boundaries. Zienkiewicz showed that when the current densities arising from the gradients are globally smoothed, they provide a more accurate estimate of the energy than the energy using the constant current densities.[7,8] Hence, (16) can be calculated using the energy due to the smoothed currents and the energy due to the constant currents. The smoothed current densities are now defined at the same nodal locations as the potentials and are continuous across element boundaries. The smoothing process uses the Galerkin technique[6] and involves minimizing the difference between the two current densities,

$$\sigma(\nabla\tilde{\Phi} - \nabla\hat{\Phi}) \tag{17}$$

where $\nabla\tilde{\Phi}$ is the smoothed gradient representing the exact gradient and $\nabla\hat{\Phi}$ is the constant gradient from the finite element solution. This method produces a system of equations in the following form:

$$(\int_{\Omega_e} \Psi_i \Psi_j d\Omega_e)(\sigma\nabla\tilde{\Phi}_i) = (\int_{\Omega_e} \sigma\nabla\hat{\Phi}_i \Psi_i d\Omega_e). \tag{18}$$

where Ψ_i is a linear basis function defined at node i, $\nabla\tilde{\Phi}_i$ is the value of the smoothed gradient at that point, σ is the conductivity, and $\nabla\hat{\Phi}_i$ is the constant gradient resulting from the finite element solution. In equation (18), the current density, $\sigma\nabla\tilde{\Phi}_i$ is the unknown that is solved for at each node, i of the grid. Thus we calculate a new estimate, $\tilde{q}$, which can be used in place of the exact gradient, q. The current density now varies linearly in each tetrahedron and is continuous everywhere.

Once the energy error has been computed, the mesh refinement can begin. The error must be related to some parameter which guides the mesh refinement. We utilized a spacing function, h_e, which controls the size and number of elements.[6] This spacing function, h_e is defined to be the linear interpolation of the spacing values at the four nodes of the tetrahedron. A variable β is defined as the ratio of the element error to the average of the element errors:

$$\beta_e = \frac{||e_q||_e}{||\hat{e}_q||_e}. \tag{19}$$

If $\beta_e > 1$, then a new spacing function can be defined as

$$h^* = \frac{h}{\beta_e}. \tag{20}$$

Thus, the spacing values at each node of the element are reduced. This new spacing function will then increase the number of points in the regions of large errors. This whole process is repeated until the global error estimate falls below the specified level.

The algorithm does not depend on how the modification was made such that a different error estimate other than the one just described can be used by only modifying the spacing values in some fashion. This allows any number of error estimators to be used to guide the adaptive mesh refinement process.

2.3 Regularization

Traditional schemes for solving the inverse problem, (3), have involved reformulating the linear equation, $A\xi = b$, into $\xi_S = K\xi_C$, where ξ_S and ξ_C are the voltages on the scalp and cerebral cortex respectively, and K is the $S \times C$ transfer matrix of coefficients relating the measured scalp voltages to the voltages measured on the cortex. The inverse problem can then be stated as, $\xi_C = K^{-1}\xi_S$.

Unfortunately, K is ill-conditioned, and regularization techniques are necessary to restore continuity of the solution back onto the data. Regularization schemes work by finding an approximate vector, $\xi_{C\varepsilon}$, such that $\xi_{C\varepsilon} \to \xi_C$ as $\varepsilon \to 0$, where ε is the error between the approximate and true values. As introduced in our 1993 SPIE paper,[9] we found that it is possible to reformulate the K matrix into an expression involving sub-matrices of K such that regularization can then be applied at a *local* level.

Researchers have long noticed that the discontinuities in the inverse solution appear irregularly distributed throughout the data. Tikhonov and other regularization schemes are, in effect, filters, which restore continuity by attenuating the high (spatial) frequency components of the solution. Since regularization is usually applied globally, the results can leave some regions overly damped or smoothed while others remain poorly constrained. Our idea was then, to apply regularization only to sub-matrices that require it and to apply different amounts of regularization to different sub-matrices.

Briefly, we begin by expressing $A\xi = b$ for the Cauchy problem in (3) as

$$\begin{pmatrix} A_{SS} & A_{SV} & A_{SC} \\ A_{VS} & A_{VV} & A_{VC} \\ A_{CS} & A_{CV} & A_{CC} \end{pmatrix} \begin{pmatrix} \xi_S \\ \xi_V \\ \xi_C \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{21}$$

We can then rearrange the matrix to solve directly for the cortical voltages in terms of the measured scalp voltages,

$$\xi_C = (A_{SV} A_{VV}^{-1} A_{VC} - A_{SC})^{-1} (A_{SS} - A_{SV} A_{VV}^{-1} A_{VS}) \xi_S, \tag{22}$$

where the subscripts, S, V, and C stand for the nodes in the regions of the scalp, volume, and cortex, respectively.

Setting A_S equal to $A_{SS} - A_{SV} A_{VV}^{-1} A_{VS}$ and A_C equal to $A_{SV} A_{VV}^{-1} A_{VC} - A_{SC}$, yields

$$\xi_C = A_C^{-1} A_S \xi_S. \tag{23}$$

Note that we now have (possibly pseudo) inverses of two matrices, A_{VV} and A_C, as well as several other matrix operations to perform. If one estimates the condition number, κ, of the two matrices using the ratio of the maximum to minimum singular values from a singular value decomposition (SVD), one finds that the condition number varies significantly (from $\kappa \approx 400$ for A_{VV} to $\kappa \approx 3.0 \times 10^{22}$ for A_C in the three-dimensional finite element model described below). Thus, one can regularize matrices differently, leaving some of the sub-matrices untouched from filtering. The overall effect of this *local regularization* is more control over the regularization process.

2.4 Computation

The software that has been developed for the solution of direct and inverse problems includes 1) programs to construct, manipulate, and display large scale, three-dimensional geometric models, 2) a three-dimensional finite element program for elliptic partial differential equations with general boundary conditions and source terms, 3) programs which apply the local regularization and solve the system of equations, and 4) visualization tools for displaying the results. Due to the large size of the problem, much of the software development has been designed for high performance computing architectures. For implementation and software details, see [10–12].

As is generally the case, there are trade-offs in obtaining more accurate solutions. Here, the principle of *conservation of complexity* applies. Achieving greater accuracy in the inverse solution using the local regularization method increases computational costs. In the global method, the regularization procedure and the estimation of the regularization parameter, α, is accomplished using the $S \times C$ transfer matrix, K, while the local scheme must "invert" two (or more, depending on the partition) sub-matrices which are $V \times V$ and $S \times C$ in size. It is almost always the case that $\dim\{V\} >> \dim\{S, C\}$. Thus, one needs to make compromises concerning accuracy versus CPU time.

3. METHODS

To study the effects of refinement on the direct and inverse EEG problems, we solved both problems on a series of six increasingly h-refined finite element head models constructed entirely from human MRI scans. For the direct solution, we chose two cortical nodes to represent a tangentially oriented dipole source, and assigned them Dirchilet boundary conditions. The system was then solved, and the surface potentials were recorded as our forward solution.

The cortical potentials were then computed using the forward solution scalp potentials in the inverse procedure. We compared these cortical potentials to the potentials computed in the forward solution to determine the accuracy of our inverse results. Finally, the mesh was refined using Schmidt's adaptive refinement algorithm,[13] and the new FE system was solved using the direct and inverse methods. This process was repeated five times, yielding a set of six forward and inverse solutions.

3.1 Model construction

The finite element models consisted of two surfaces – scalp and cerebral cortex, as well volume nodes between these two surfaces, and nodal connectivities. To construct these models, the scalp and cortex were extracted from the MRI data. The cortical surface was obtained by segmenting the individual MRI slices using a semi-automatic contouring algorithm,[14] and then combining the contours into a closed surface using a post-processing lacing algorithm.[15] The triangulated cortical surface consisted of 772 nodes and 1540 triangles.

While the scalp could have been extracted using the same segmenting technique, we chose to use a volume extraction method to avoid the necessary manual interaction of contour extraction. Instead, we merged the slices into a 3-dimensional volume data set and executed a marching cubes isosurface thresholding algorithm[16] to locate the scalp. The extraction of the scalp was fully automatic and required less than 1 second after the appropriate thresholding value was determined (determining this value was interactive – requiring less than a minute in the SCIRun software system.[10]) The extracted scalp surface consisted of 752 nodes and 1500 triangles.

To create the volume, we added 1000 evenly distributed volume nodes to the space between the cortex and the scalp. These nodes provided the coarse discretization from which we refined on subsequent adaptive iterations.

3.2 Direct solution

We selected two nodes on the cortical surface to represent our dipole source. The nodes were chosen to be typical of an ectopic focus recorded from the temporal lobe of an epileptic patient. These nodes were assigned Dirchilet boundary values of ± 10.0 mV, and were the only non-Neumann boundary nodes in our simulation.

The initial mesh consisted of 2724 nodes and 12,812 elements. This mesh was then fed into a conjugate gradient finite element solver, and the solutions were recorded to disk. For a complete discussion of this direct solving method, see [11].

3.3 Inverse solution

Solving the inverse problem required calculating ξ_C in (22). As discussed above, this process involves inverting the A_C and A_{VV} matrices, as well as several matrix multiplications.

To save space, A_{VV} was stored in a column major, sparse format. The specific format we chose was made to be compatible with the sparse storage structure utilized by the *Numerical Recipes in C*[17] library. This allowed us to use the *Numerical Recipes* LU decomposition and back-substitution code. After decomposing A_{VV}, we back-substituted A_{VS} and A_{VC} one column at a time to build the $A_{VV}^{-1}A_{VS}$ and $A_{VV}^{-1}A_{VC}$ products.

The last step in generating the inverse solution required an inversion of the A_C matrix. This matrix had a condition number on the order of 10^{22} and was thus highly ill-conditioned, requiring some degree of regularization in order to be inverted. Singular Value Decomposition (SVD) was used to invert A_C, and the smallest values were truncated. To find the optimum truncation parameter, α, we implemented an iterative algorithm in which each singular value was tagged with a normalized weighting defined as the current singular value divided by the maximum singular value of the system. Singular values below α were then truncated. Our iterative algorithm began with $\alpha = 0.1$ and divided this parameter by 10 at each iteration. The algorithm stopped when the RMS error between the inverse and true solutions stopped improving. (We are using the most refined forward solution as our "true" solution.) In practice we truncated between 35% and 90% of the values between the first and last iterations.

3.4 Mesh refinement

After solving the forward and inverse problems on the initial mesh, the mesh was refined based on the error analysis of the forward solution, as discussed in the Adaptive methods section above. The newly refined mesh was then solved via the forward and inverse methods. This process was repeated several times in order to analyze the effects of h-adaption on our inverse solutions.

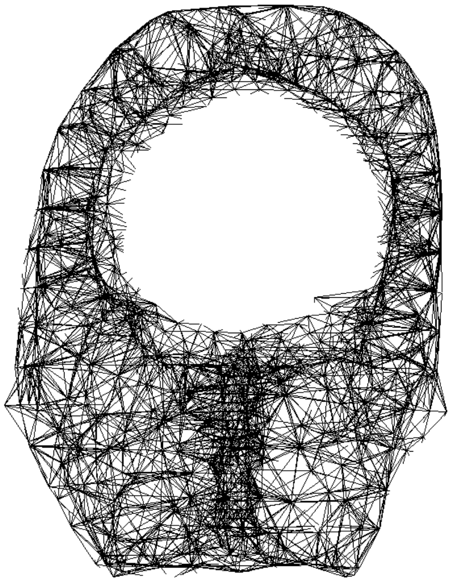

Figure 1: Coronal cross-section of the initial mesh. The full initial mesh contains 2724 nodes and 12,812 elements.

4. RESULTS AND DISCUSSION

4.1 Results

Applying the mesh adaptation algorithm described in section 2.2, we created a sequence of six increasingly refined models of a patient's head with a simulated ectopic focus from high-resolution MRI scans. Each model was refined based upon the forward simulation error estimates from the proceeding simulation. The models increased in complexity from 2724 nodes and 12,812 elements in the first simulation, to 5224 nodes and 29,135 elements in the last simulation. The element distributions for the initial mesh and the most refined mesh can be seen in Figure 1 and Figure 2, respectively. We note that most of the refinement occurred near the simulated dipole located in the patient's right temporal lobe.

Each simulation took between 1.5 and 3.0 hours to run on an SGI Indigo 2 with a 200 MHz R4400 processor and 128 Mb of RAM. Partial results were stored to disk (specifically A_S and A_C) allowing the algorithm to quickly iterate through the truncation parameters. The vast majority of the computational time (over 80%) was spent in the LU decomposition of the A_{VV} sub-matrix.

The RMS errors contained in Table 1 indicate that the forward solution is converging to the true solution with increased mesh refinement. Furthermore, we observe in Table 2 that the inverse solution at each level of refinement varies from the forward solution by a constant of approximately 45%. From these results it might be inferred that the inverse solution will improve proportionately with the forward solution. This inference is confirmed by our results in Table 3, showing the RMS error between the inverse and true solutions at each level of refinement. This data supports our hypothesis that the inverse solution will improve with refinement, as we note the RMS error decreases from 88% to 46% between the first and sixth levels of adaptation.

4.2 Future work

Previous research[9] indicates that increased refinement will further improve our results. Consequently, it is a primary focus of our future research efforts to increase the number of nodes we are able to solve for in our inverse simulations.

The 29,000 elements used for these simulations was, in all likelihood, too coarse of a discretization to adequately capture the physics of the inverse EEG problem. Fundamentally, this limitation was imposed by the spatial and

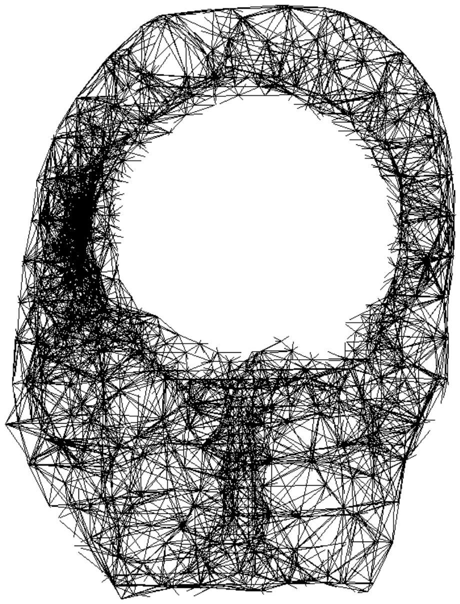

Figure 2: Coronal cross-section of the final mesh. The elements in the right frontal lobe have been the most thoroughly refined due to the steep gradients and large errors in that region. The full final mesh contains 5224 nodes and 29,135 elements.

Number of Nodes					
2724	3224	3724	4224	4724	5224
73.76%	106.45%	74.88%	87.61%	21.67%	0.0%

Table 1: RMS error percentages between the true solution and previous forward solutions for six levels of mesh refinement.

temporal demands of inverting and storing large sub-matrices on a single CPU workstation. As a result, we are currently pursuing several techniques to improve our sparse matrix storage and manipulation routines.

We employed a sparse storage structure consistent with that used in *Numerical Recipes in C* [17] in order to use their LU decomposition and back-substitution code, as well as their SVD code. This storage technique required an $N \times K$ matrix where K was equal to the full bandwidth of the matrix. While we were able to significantly reduce this number via a Reverse Cuthill-McKee[18] node reordering, K was generally $> 10\%$ of N. We plan to rewrite this code using a more sparse structure (e.g. the Yale sparse storage format), thereby further decreasing our storage costs.

The number of volume nodes was also limited temporally by the time required to decompose A_{VV} into its L and U components. An improved sparse matrix structure will significantly reduce this expense, as will employing a parallel block LU decomposition algorithm. Our other matrix operations, though a lesser percentage of our total execution time, were significant as well. We intend to improve this performance through distributed matrix multiplication and singular value decomposition algorithms.

ACKNOWLEDGMENTS

This work was supported in part by the NSF and NIH. The authors would like to thank Steve Parker for his

	Number of Nodes					
Truncation Parameter	2724	3224	3724	4224	4724	5224
0.1	98.62%	80.23%	85.98%	99.27%	98.40%	89.70%
0.01	93.67%	60.85%	68.28%	68.00%	68.44%	63.5%
0.001	65.03%	55.05%	57.17%	50.41%	55.96%	54.11%
0.0001	58.77%	52.93%	59.06%	41.70%	46.69%	50.83%
0.00001	53.69%	–	51.34%	–	–	–
0.000001	52.69%	–	–	–	–	–

Table 2: RMS error percentages between the forward and inverse solutions for six levels of mesh refinement. Multiple levels of SVD truncation are shown.

	Number of Nodes					
Truncation Parameter	2724	3224	3724	4224	4724	5224
0.1	99.19%	117.75%	95.98%	99.80%	98.40%	89.70%
0.01	94.81%	113.77%	90.60%	95.01%	64.86%	63.56%
0.001	90.81%	113.91%	87.09%	91.86%	54.13%	54.11%
0.0001	90.28%	112.51%	87.19%	89.71%	46.59%	50.83%
0.00001	88.76%	–	82.95%	–	–	–
0.000001	88.27%	–	–	–	–	–

Table 3: RMS error percentages between the true and inverse solutions for six levels of mesh refinement and multiple levels of SVD truncation are shown. Note: we are using the most refined forward simulation solution as the "true" solution.

suggestions and comments. Furthermore, we appreciate access to facilities which are part of the NSF/ARPA STC for Computer Graphics and Scientific Visualization.

References

[1] Y. Yamashita. Theoretical studies on the inverse problem in electrocardiography and the uniqueness of the solution. *IEEE Transactions on Biomedical Engineering*, 29:719–725, 1982.

[2] Y. Yan, P.L. Nunez, and R.T. Hart. Finite-element model of the human head: scalp potentials due to dipole sources. *Medical & Biological Engineering & Computing*, 29:475–481, 1991.

[3] F.H. Lopes da Silva. A critical review of clinical applications of topographic mapping of brain potentials. *Journal of Clinical Neurophysiology*, 7(4):535–551, 1990.

[4] P.G. Ciarlet and J.L Lions. *Handbook of Numerical Analysis: Finite Element Methods*, volume 1. North-Holland, Amsterdam, 1991.

[5] D. Burnett. *Finite Element Analysis- From Concepts to Applications.* Addison-Wesley, Reading, MA, 1987.

[6] R. Lewis, H. Huang, A. Usmani, and J. Cross. Finite element analysis of heat transfer and flow problems using adaptive remeshing including application to solidification problems. *International Journal of Numerical Methods in Engineering*, 32:767–781, 1991.

[7] O. Zienkiewicz and J. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal of Numerical Methods in Engineering*, 24:337–357, 1987.

[8] O. Zienkiewicz and J. Zhu. Adaptivity and mesh generation. *International Journal of Numerical Methods in Engineering*, 32:783–810, 1991.

[9] C.R. Johnson and R.S. MacLeod. Inverse solutions for electric and potential field imaging. In R.L. Barbour and M.J. Carvlin, editors, *Physiological Imaging, Spectroscopy, and Early Dectection Diagnostic Methods*, volume 1887, pages 130–139. SPIE, 1993.

[10] C.R. Johnson and S.G. Parker. A computational steering model applied to problems in medicine. In *Supercomputing '94*, pages 540–549. IEEE Press, 1994.

[11] J.A. Schmidt, C.R. Johnson, J.C. Eason, and Macleod R.S. *Modeling, Mesh Generation, and Adaptive Methods for Partial Differential Equations*, chapter Applications of Automatic Mesh Generation and Adaptive Methods in Computational Medicine. Springer-Verlag, 1994.

[12] R.S. MacLeod, C.R. Johnson, and M.A. Matheson. Visualization tools for computational electrocardiography. In *Visualization in Biomedical Computing*, pages 433–444, 1992.

[13] J. Schmidt. *Mesh generation with applications in computational electrophysiology.* PhD thesis, Duke University, Durham, NC, 1993.

[14] H.W. Shen and C.R. Johnson. Semi-automatic image segmentation: A bimodel thresholding approach. Technical report, UUCS-94-019, Dept. of CS, Univ. of Utah, 1994.

[15] R.S. MacLeod, C.R. Johnson, and M.A. Matheson. Visualization tools for computational electrocardiography. In *Visualization in Biomedical Computing*, pages 433–444, 1992.

[16] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[17] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1992.

[18] A. Jennings and J.J. McKeown. *Matrix Computation.* Wiley, New York, 1992.