COMPUTERS AND BIOMEDICAL RESEARCH **31,** 348–362 (1998)
ARTICLE NO. CO981486

Multigrid Solution of the Potential Field in Modeling Electrical Nerve Stimulation

Rudi Hoekema,*[,1] Kees Venner†, Johannes J. Struijk,* and Jan Holsheimer*

**Institute for Biomedical Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands; and †Faculty of Mechanical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

Received March 3, 1997

In this paper, multilevel techniques are introduced as a fast numerical method to compute 3-D potential field in nerve stimulation configurations. It is shown that with these techniques the computing time is reduced significantly compared to conventional methods. Consequently, these techniques greatly enhance the possibilities for parameter studies and electrode design. Following a general description of the model of nerve stimulation configurations, the basic principles of multilevel solvers for the numerical solution of partial differential equations are briefly summarized. Subsequently, some essential elements for successful application are discussed. Finally, results are presented for the potential field in a nerve bundle induced by tripolar stimulation with a cuff electrode surrounding part of the nerve.

Key Words: multilevel techniques; nerve stimulation; numerical solution methods.

INTRODUCTION

Modeling field potentials imposed on neural tissue by electrical stimulation, as a tool to predict the effects of the stimulation, is being used for several clinical applications, such as spinal cord stimulation (*1*, *2*, *3*, *4*), peripheral nerve stimulation (*5*, *6*, *7*), and sacral root stimulation (*8*). The aim of these studies is to selectively stimulate either a specific part of the nervous tissue or nerve fibers of a specific diameter range or spatial orientation. Physico-mathematical models of the conducting biological media are often applied in combination with a model representing the electrical properties of the neural elements, usually myelinated nerve fibers (*9*). These models can be used to improve the understanding of the exact mechanisms involved in the stimulation process or to validate models used in the design of electrodes for clinical applications.

Due to differences in the electrical properties and the complex shapes of the

[1] To whom correspondence should be addressed at the Department of Clinical Neurophysiology, University Hospital Utrecht, P.O. Box 85500, 3508 GA Utrecht, The Netherlands.

0010-4809/98 $25.00

various tissues involved, the potential field cannot be obtained analytically and a numerical approach is needed. This involves discretization of the region of interest and deriving a system of equations (by means of a finite element method or finite difference method) from which the potential must be solved. Unfortunately this numerical solution is often hampered by computing time problems. To describe the potential field accurately for a 3-D configuration requires a grid with a large number of nodes. For example, taking 50 grid points in each direction results in a discrete system of $O(10^5)$ equations and associated unknown variables to be solved. Since most common numerical algorithms available for solving this system of equations require a computing time proportional to n^2–n^3 (with n, the number of unknowns), excessive computing times may result. This limitation partly accounts for the use of restricted models such as 2-D representations (*1*) or 3-D models assuming rotational symmetry (*5*, *8*). Nevertheless, studies using a full 3-D geometry have been performed as well (*2*, *3*).

The objective of the present study is to develop a fast solver for the potential field in nerve stimulation using multilevel techniques (*10*, *11*). These techniques have the prospect of solving a problem in a computing time proportional to n. This textbook efficiency was first obtained for elliptic partial differential problems with constant coefficients. Nowadays they are widely used in computational fluid dynamics, find increasing application in engineering, and have resulted in major reductions in computing time.

In this paper the specific aspects of the application of these techniques to the numerical solution of the potential field in electrical nerve stimulation are described. Subsequently, results are presented and discussed for a nerve and cuff configuration. To demonstrate the efficiency of the developed algorithm the computing time needed to solve the problem to a certain accuracy is compared with the time needed to obtain this same accuracy with a standard iterative method, i.e., Gauss–Seidel relaxation.

VOLUME CONDUCTOR MODEL

The problem can be described as the solution of the unknown potential field u in a 3-D domain Ω from the 3-D Poisson's equation:

$$\nabla(\sigma\nabla u) = f \qquad \mathbf{x} = (x_1, x_2, x_3) \in \Omega \qquad [1]$$

with $u = g$ on the boundary $\partial\Omega$. In this equation σ denotes the conductivity tensor. Characteristic for a nerve stimulation problem are anisotropy and a strong inhomogeneity, in accordance with the different electrical properties of the various tissues.

Typically Ω will be taken a 3-D rectangular domain including part of the nerve and the stimulating electrode. Assuming that the boundaries are sufficiently far from the electrodes and the location of the neural structure(s) to be activated, $g = 0$ is an acceptable choice. Finally, f is the so-called forcing function which contains the current sources, i.e., the electrode(s)

$$f = A(\mathbf{x_e})\delta(\mathbf{x_e}), \qquad [2]$$

where δ is the (3-D) Dirac function and $\mathbf{x_e}$ is the location of the electrode.

For the numerical solution the domain is divided into cube shaped cells with each of these cells having a specific size and an (an)isotropic conductivity. For the present work the model consists of 56 × 56 × 56 cells. The conductivities were chosen to model the nerve and cuff configuration. Assuming a Cartesian coordinate system and a diagonal conductivity tensor equation [1] can be written as

$$\sum_{k=1}^{d} \left(\frac{\partial}{\delta x_k} \left(\sigma_k \frac{\partial u}{\partial x_k} \right) \right) = f, \qquad [3]$$

where d is the dimension of the problem, i.e., $d = 3$. Subsequently, using Taylor series, in each point $\mathbf{i} \equiv (i_1, i_2, i_3)$, Eq. [3] can be approximated by

$$\sum_{k=1}^{d} (\overline{\sigma}_k, u_{\mathbf{i}-1} - (\overline{\sigma}_{k,\mathbf{i}} + \overline{\sigma}_{k,\mathbf{i}+1})u_{\mathbf{i}} + \overline{\sigma}_{k,\mathbf{i}+1}u_{\mathbf{i}+1}) = f_{\mathbf{i}}, \qquad [4]$$

where $\mathbf{i} \pm 1$ denotes the nearest neighbors of the point $\mathbf{i}$ in the kth dimension, i.e., if $k = 1$, $\mathbf{i} \pm 1 = (i_1 \pm 1, i_2, i_3)$, etc. The coefficient $\overline{\sigma}_k$ is defined as

$$\overline{\sigma}_{k,\mathbf{i}} = \frac{2\sigma_{k,\mathbf{i}-1/2}}{h_{\mathbf{i}}(h_{\mathbf{i}+1} + h_{\mathbf{i}})},$$

where $h_{\mathbf{i}}$ is defined as the distance between the points with index $\mathbf{i}$ and $\mathbf{i} - 1$. Furthermore, $\sigma_{k,\mathbf{i}-1/2}$ denotes the value of σ_k at the center of the grid line connecting the points $\mathbf{i}$ and $\mathbf{i} - 1$, i.e., between (i_1, i_2, i_3) and $(i_1 - 1, i_2, i_3)$ if $k = 1$. This value can be computed from the value of σ_k in the neighboring cells. Function $f_{\mathbf{i}}$ in Eq. [4] is defined as

$$f_{\mathbf{i}} = \frac{A_{\mathbf{i}}}{\prod_{k=1}^{d} h_{\mathbf{i}}} \qquad [5]$$

for all points $\mathbf{i}$ within the region covered by the electrode, and $f_{\mathbf{i}} = 0$ elsewhere. The coefficients $A_{\mathbf{i}}$ are not known in advance. Their values are obtained by using *current scaling*, i.e., they are solved iteratively using the total electrode current and the physical condition that the potential is uniform over the electrode. Finally, the (Dirichlet) boundary condition is $g_{\mathbf{i}} = 0$ for the points at the boundary $\partial\Omega$.

MULTIGRID TECHNIQUES

Introduction

For a uniform grid with mesh size h, Eq. [4] can be written as

$$L^h \boldsymbol{u}^h = \boldsymbol{f}^h \qquad \text{on} \qquad \Omega^h, \qquad [6]$$

where L^h is a matrix, $\boldsymbol{u}^h$ is a vector containing the potential field to be solved, and $\boldsymbol{f}^h$ is a vector representing f in the grid points.

For the solution of $\boldsymbol{u}^h$ various (direct or iterative) methods can be used. In general, these methods require a computing time of $O(n^p)$, if n is the total number of grid points, with the value of p depending on the chosen algorithm and the dimension of the problem. For most algorithms $p \approx 2$. As a result these methods are poorly suited for configurations in which n is large (in our case $n = 55^3$). Since in parameter or design studies large numbers of solutions are required, an alternative method is needed.

This alternative was found in the application of multilevel techniques (*10*, *11*). These techniques have the prospect of attaining $p = 1$, i.e., solving the problem in a computing time linearly proportional to the number of grid points.

Application of multilevel techniques in problems similar to the problem considered here is described in (*12*, *13*).

Relaxation

Characteristic for iterative processes is that a given approximation to the solution is improved by visiting all grid points in some prescribed order, at each site changing the value of $u_{\mathbf{i}}$ according to a rule specific for the process.

To solve Eq. [4] using, e.g., Gauss–Seidel relaxation, the current approximation $\tilde{u}_{\mathbf{i}}$ at each grid point is replaced by $\bar{u}_{\mathbf{i}}$

$$\bar{u}_{\mathbf{i}} = \tilde{u}_{\mathbf{i}} + \omega r_{\mathbf{i}} \left(\frac{\partial L}{\partial u_{\mathbf{i}}} \right)^{-1}_{\mathbf{i}}, \qquad [7]$$

where $r_{\mathbf{i}}$ denotes the dynamic residual

$$r_{\mathbf{i}} = f_{\mathbf{i}} - \sum_{k=1}^{d} (\bar{\sigma}_{k,\mathbf{i}} \bar{u}_{\mathbf{i}-1} - (\bar{\sigma}_{k,\mathbf{i}} + \bar{\sigma}_{k,\mathbf{i}+1}) \tilde{u}_{\mathbf{i}} + \bar{\sigma}_{k,\mathbf{i}+1} \tilde{u}_{\mathbf{i}+1}) \qquad [8]$$

and

$$\left(\frac{\partial L}{\partial u_{\mathbf{i}}} \right)_{\mathbf{i}} = - \sum_{k=1}^{d} (\bar{\sigma}_{k,\mathbf{i}} + \bar{\sigma}_{k,\mathbf{i}+1}); \qquad [9]$$

ω has a value $1 \leq \omega \leq 2$. When applied repeatedly this iterative scheme converges to the solution of Eq. [4].

However, when monitoring some norm of the residual or the error, it will show that in the first few iterations large reductions occur, but then the speed of convergence slows down and the iterative process becomes inefficient. This is characteristic for ''local'' iterative processes, like Gauss–Seidel relaxation. The iterative process is well capable of reducing errors with a wavelength of the order of the mesh size h, but smooth errors (with a larger wave length) are hardly affected, which causes the slow asymptotic convergence.

The consequence is that after a few iterations the remaining error in the solution will be smooth relative to the mesh size. In fact, it can be accurately represented and solved on a coarser grid. This concept forms the basis of a multigrid solver: instead of continuing the iterative process when convergence

has slowed down, one switches over to a coarser grid for the solution of the (smooth) remaining error. Solving the error on the coarse grid can be done more efficiently because the number of nodes is smaller and (due to the larger mesh size) the iterative process converges faster. Once an accurate approximation to the error is obtained on the coarser grid it is used to correct the solution on the fine grid.

Coarse Grid Correction

Consider the solution of Eq. [6] and let $\tilde{\boldsymbol{u}}^h$ denote the approximation to the solution obtained after some relaxations. The error in this approximation is defined as

$$\boldsymbol{v}^h \equiv \boldsymbol{u}^h - \tilde{\boldsymbol{u}}^h. \quad [10]$$

Using $\tilde{\boldsymbol{u}}^h$, residuals can be calculated according to

$$\boldsymbol{r}^h = \boldsymbol{f}^h - L^h\tilde{\boldsymbol{u}}^h. \quad [11]$$

By definition $L^h\boldsymbol{u}^h = f^h$. Hence, this equation can be written as

$$\boldsymbol{r}^h = L^h\boldsymbol{u}^h - L^h\tilde{\boldsymbol{u}}^h. \quad [12]$$

If L^h is a linear operator. Equation [12] can be written as an equation for the error

$$L^h\boldsymbol{v}^h = \boldsymbol{r}^h. \quad [13]$$

Since, after some relaxations, $\boldsymbol{v}^h$ will be smooth compared to the mesh size, it can be represented and solved (by the same iterative procedure) on a coarser grid. A coarse grid approximation $\boldsymbol{v}^H$ to $\boldsymbol{v}^h$ is solved from Eq. [13] on grid H

$$L^H\boldsymbol{v}^H = I_h^H\boldsymbol{r}^h. \quad [14]$$

L^H is the representation of the matrix operator L^h on the coarse grid and I_h^H in Eq. [14] is a restriction operator from the fine to the coarse grid.

After an approximation $\tilde{\boldsymbol{v}}^H$–$\boldsymbol{v}^H$ has been calculated, it is used to correct the approximation $\tilde{\boldsymbol{u}}^h$ on the fine grid according to

$$\bar{\boldsymbol{u}}^h = \tilde{\boldsymbol{u}}^h + I_H^h\tilde{\boldsymbol{v}}^H, \quad [15]$$

where I_H^h is an interpolation operator. Because the corrections calculated on the coarse grid are smooth, a linear interpolation generally provides sufficient accuracy.

Coarse Grid Correction Cycle

If the number of nodes on the fine grid is large, the number of nodes on a coarse grid H may still be relatively large and after a few relaxations convergence on the coarse grid will also slow down. At that point the same reasoning applies as described before. On a coarser grid, the remaining error can be accurately described and solved. This process of coarsening can be repeated recursively

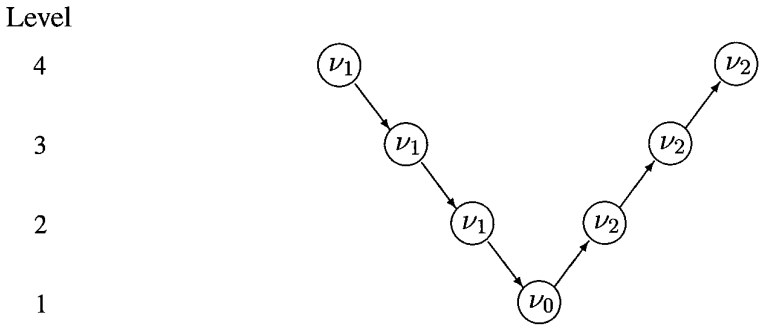

FIG. 1. Flow chart of a multigrid coarse grid correction cycle $V(\nu_1, \nu_2)$.

until a grid is reached, where the number of nodes is so small compared to the original fine grid that the equations can be solved exactly or almost exactly in only a few operations.

This leads to the so-called coarse grid correction cycle. For a situation of four levels (grids) such a cycle is displayed in Fig. 1. ν_1, ν_2, and ν_0 denote the number of relaxation sweeps on the respective grids. Generally ν_1 and ν_2 are small, i.e., 1 or 2, whereas ν_0 is in general larger, e.g., $\nu_0 = 10$. The cycle displayed in Fig. 1 is referred to as a $V(\nu_1, \nu_2)$-cycle.

A coarse grid correction cycle as explained here in principle reduces the error by a factor of $\overline{\mu}^{(\nu_1+\nu_2)}$, where $\overline{\mu}$ is the worst of the factors by which a single relaxation reduces the ''high frequency'' components in the error, i.e., those components that have to be resolved on the fine grid, and cannot be seen by the coarse grid. With the standard choice of coarsening $H = 2h$, these are the components that have an oscillating character in at least one of the spatial dimensions. For elliptic partial differential problems with coefficients that vary on a large scale compared to the mesh size, a simple scheme such as Gauss–Seidel relaxation, already has a $\overline{\mu}$ independent of the mesh size. For example, for the present problem (3-D) with constant coefficients and discretized on a uniform grid, it has $\overline{\mu} = 0.56$. Consequently a cycle with $\nu_1 = 2$ and $\nu_1 = 1$ already reduces the error by almost a factor of 6. The computational cost of the cycle is just a little larger than the work invested in the $(\nu_1 + \nu_2)$ relaxations that are actually performed on the finest grid (7/8 times this work for a three-dimensional problem). This clearly illustrates the potential gain in efficiency that can result. For practical problems, large jumps in coefficients, strong anisotropies, more elaborate relaxation schemes may be needed to obtain such $\overline{\mu}$. Alternatively, if certain components cannot be reduced by the relaxation, one can modify the coarsening strategy, i.e., only coarsen the grid in one direction (at the time). This would then be at the expense of extra computational effort on the coarse grids.

Summarizing, a coarse grid cycle is not a given strictly prescribed device. Various choices can be made in its design. The principle guideline however is

always to obtain a cycle with a convergence factor that is independant of the grid size. This means that in the course of developing a coarse grid correction cycle for a problem, a number of such choices have to be made, depending on the character of the problem. For the problem considered here some specific aspects are discussed in the next section. It may be obvious that aside from these algorithmic compromises, often additional compromises have to be made in practice, i.e., textbook efficiency is not the goal, but simply solving the problem as it is given sufficiently fast is.

IMPLEMENTATION

Implementation of the method described to the problem at hand requires a number of modifications. First, the grid used was not uniform and thus, a linear interpolation to describe the (smoothed) error on a coarser grid is not suitable. Second, in the nerve and cuff model large jumps of the conductivity in the domain exist. These large jumps cause large differences between the coefficients of the discrete system of equations for grid points in different parts of the domain.

Grid Structure

The first objective of this study was to improve upon the Gauss–Seidel-like solver used by Struijk *et al.* (*4*). Therefore, their nonuniform grid structure of $56 \times 56 \times 56$ cells was used as the finest level. Subsequently the coarser grids were constructed by each time discarding every second grid line in all three directions.

Coarse Grid Conductivity Tensor

To solve the error on the coarse grid, a coarse grid representation L^H of the fine grid operator L^h is needed. This requires the definition of the conductivities at the center of the coarse grid links, i.e., ($\sigma_{k,\mathbf{i}-1/2}$ in Eq. [4]). Various alternative definitions are presented by Alcouffe *et al.* (*12*) for a 2-D case. From the one yielding the best efficiency for the coarse grid correction cycle a 3-D version was constructed as follows (see Fig. 2).

An efficient coarse grid correction requires that the coarse grid coefficient on the link between A and B accurately reflects the conductivity relation between the fine grid points coinciding with A and B. This can be ensured if the coarse grid coefficient is taken as the weighted average of a number of fine grid conductivity coefficients between A and B. However, between every pair of adjacent coarse grid points, on the fine grid one can distinguish many pathways. In the present study only the pathways consisting of at most four fine grid links are considered. In that case five paths remain. In Fig. 2, these paths are indicated with the solid lines. Thus, the coarse grid coefficient was defined as

$$\overline{\sigma}_{k,I_1,I_2,I_3} = \tfrac{1}{6}(3\overline{\sigma}^h_{AB} + \overline{\sigma}^h_{ACDB} + \overline{\sigma}^h_{AEFB} + \overline{\sigma}^h_{AGHB} + \overline{\sigma}^h_{AIJB}), \qquad [16]$$

where (I_1, I_2, I_3) is a coarse grid index, and

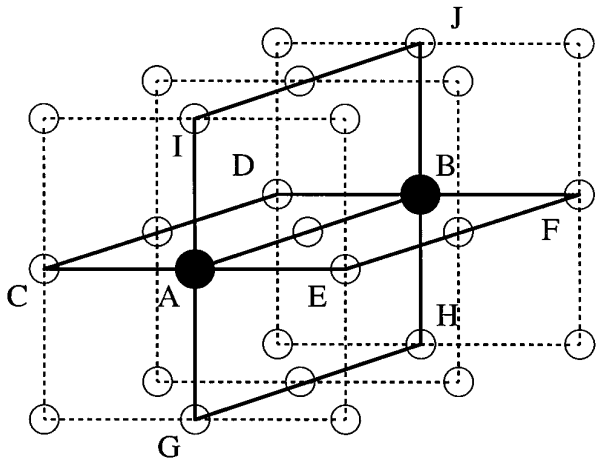

FIG. 2. Coarse grid points (black circles) and fine grid points (white circles) involved in the definition of the coarse grid conductivity at the center of the link (AB).

$$\frac{1}{\overline{\sigma}_{AB}} = \left(\frac{1}{\overline{\sigma}_{2,i_1,i_2,i_3}} + \frac{1}{\overline{\sigma}_{2,i_1,i_2-1,i_3}} \right) \qquad [17]$$

$$\frac{1}{\overline{\sigma}_{ACDB}} = \left(\frac{1}{\overline{\sigma}_{1,i_1+1,i_2-2,i_3}} + \frac{1}{\overline{\sigma}_{2,i_1+1,i_2-1,i_3}} + \frac{1}{\overline{\sigma}_{2,i_1+1,i_2,i_3}} + \frac{1}{\overline{\sigma}_{1,i_1+1,i_2,i_3}} \right). \qquad [18]$$

Intergrid Transfers

For the simple case of constant conductivity, multilinear interpolation of the potential itself provides a sufficiently accurate basis for the intergrid operators. The interpolation itself will serve well in the process of correcting the fine grid approximation (I_H^h in Eq. [15]), and its transpose (full weighting) can be used for the restriction of the residuals to the coarse grid (I_{h}^H in Eq. [14]). Also, for changes of $\overline{\sigma}_k$ values over the domain of up to one order of magnitude, a coarse grid correction cycle in which these operators are used will provide sufficiently large error reductions. However, to attain fully efficient coarse grid correction cycles in the case of conductivity jumps of two orders of magnitude or more, the intergrid transfers should be based upon the continuity of $\sigma \nabla u$ (*12*). To satisfy this continuity, the conductivity values and the mesh size are incorporated in the interpolation. The resulting interpolation for a 3-D configuration is given in the Appendix. The restriction operator was subsequently obtained by taking its normalized transpose.

Coarse Grid Operator

For the operator L^h, the fine grid conductivities between the grid points are used. L^H can be chosen as the same operator but defined on the coarse grid and using the definition of the coarse grid conductivity. The resulting 7-point operator

will very well describe the characteristic behavior of L^h for smooth components if the conductivities vary moderately. However, in the case of conductivity jumps of magnitude of two orders or more, it is better to use variational coarsening (*11*)

$$L^H \equiv I_h^H L^h I_H^h. \qquad [19]$$

This yields a 27-point coarse grid operator (instead of a 7-point operator). In the nerve and cuff model, the use of Eq. [19] became essential in obtaining an efficient coarse grid correction cycle. Unfortunately, when using this operator the computing time for each relaxation sweep on the coarse grid increases by a factor $27/7 \approx 4$. However, as the number of coarse grid points is 8 times smaller than the number of fine grid points (3-D problem), the relaxation on the coarse grid is still faster. Another disadvantage of the 27-point operator could be its significant storage requirement, i.e., all 27 coefficients of the operator for all coarse grid points.

Relaxation

A relaxation scheme that efficiently reduces high frequency error components forms the core of a multigrid solver. As mentioned before, it should reduce all components that cannot be solved on a coarser grid. With standard coarsening $H = 2h$ in all directions for a 3-D problem this implies that it should in fact reduce 7/8 of the harmonic components, i.e., all components that are ''high frequency'' in at least one of the grid directions. This puts quite a strain on the relaxation process. As long as the jumps in the coefficients are moderate the required smoothing is sufficiently well ensured by pointwise Gauss–Seidel relaxation. However, for the ''pathological'' cases where large jumps in coefficient values occur, such a pointwise scheme is no longer suited. Instead, in such cases the relaxation should solve the discrete equation in ''blocks'' of strongly coupled points simultaneously (*12*). Various alternatives can be chosen, depending on the specifics of the problem. In the present study we have used alternating line relaxation.

In the case that an electrode is placed in biological tissue, the electrode potential will be uniform. This is a constraint that should be applied in the model as well: the potential at all grid points describing the electrode should be equal. In the results described in the next section, the current flowing to each of the electrode points is scaled such that the total current is constant and the constraint is satisfied. This is done iteratively: after each V-cycle, the electrode potential is evaluated and the current to each point is adjusted if necessary using

$$A_{\mathbf{i}} = \omega * A_{\text{new},\mathbf{i}} + (1 - \omega) * A_{\text{old},\mathbf{i}}, \qquad [20]$$

where ω is an under relaxation factor ($0 < \omega < 1$) and $A_{\text{new},\mathbf{i}}$ is calculated in two steps: the electrode potential is calculated using $L^h u^h = f^h$, treating all electrode points of the electrode as one point. Secondly, $f_{\mathbf{i}}$ to obtain this potential is calculated for each electrode point, which yields $A_{\text{new},\mathbf{i}}$ when divided by $\prod_{k=1}^{d} 1/h_{\mathbf{i}}$.

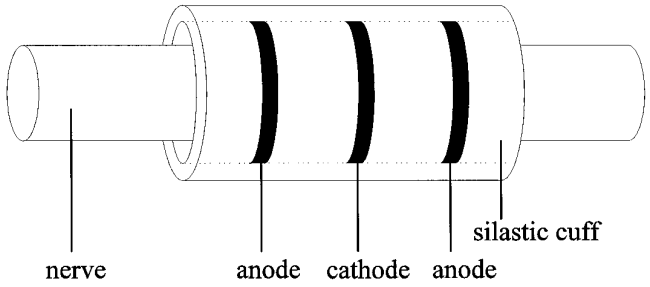

FIG. 3. Schematic representation of a nerve with a tripolar cuff electrode.

RESULTS

The multilevel solver was applied to compute the potential field in a spinal cord model and in a nerve and cuff model. Of these two problems the latter is the most demanding and results for this problem are presented here. Figure 3 shows the nerve and cuff model. Three 1-mm wide ring-shaped electrodes are embedded at the inner surface of the cuff: a central cathode and an anode at 3-mm distance on each side. The conductivities used for the various compartments in the model are given in Table 1. The indices in the first column of this table correspond to the numbers in the Figs. 4(a)–4(b).

From Table 1 it can be seen that in this model, only the nerve bundle (fascicle) has an anisotropic conductivity. The largest jumps in conductivity occur from the silastic cuff to the saline (both inside and outside the cuff), and from the (inner) saline to the epineurium. Both jumps are about a factor 2500. Because the largest current density will occur at the inner surface of the cuff where the cathode is positioned, the grid spacing was minimal in this area (0.05 mm). The stimulation amplitude was taken 1 mA for the cathode and 0.5 mA for each anode.

In Fig. 4(a), the isopotential lines of the computed field in a transverse plane through the center of cathode are shown. Figure 4(b) shows the isopotential

TABLE 1

Conductivity Values Used in the Nerve and Cuff Model

Index	Compartment	Conductivity $[\Omega\ m]^{-1}$
1	saline	2.0
2	fascicle (transverse)	0.08
	fascicle (longitudinal)	0.50
3	surrounding layer	0.02
4	silastic cuff	0.0008
5	perineurium	0.003
6	epineurium	0.008

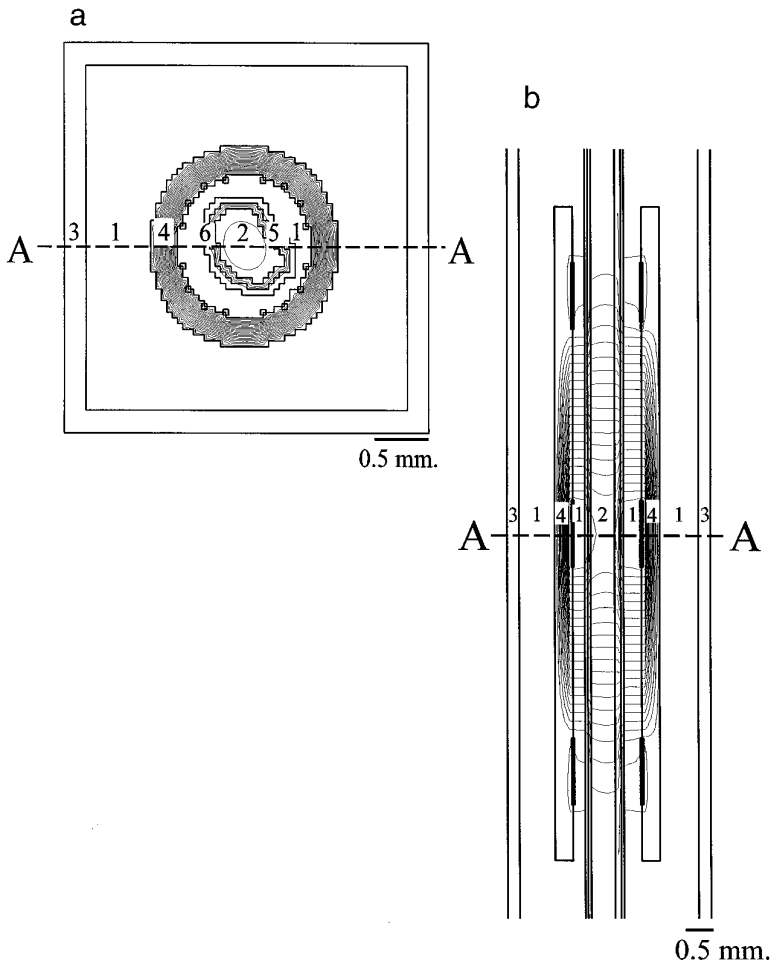

FIG. 4. Calculated potential field in the nerve and cuff model (tripolar stimulation); (a) transverse cross-section and (b) longitudinal cross-section.

lines in a longitudinal plane. The displayed isopotential lines are 20 mV apart. Due to the symmetrical electrode configuration, in which the two anodes had the same potential, current flowing outside the cuff is almost zero. Almost no current is crossing the surrounding layer.

Notice that the transverse potential gradient is rather small in the fascicle itself. This is due to the shunt effect of the saline between the cuff and the fascicle.

To demonstrate the efficiency of the multigrid solver, the convergence behavior has been studied and compared with the convergence behavior of two single-grid solution methods applied to the same model.

For this purpose the average residual norm (the average over all grid points

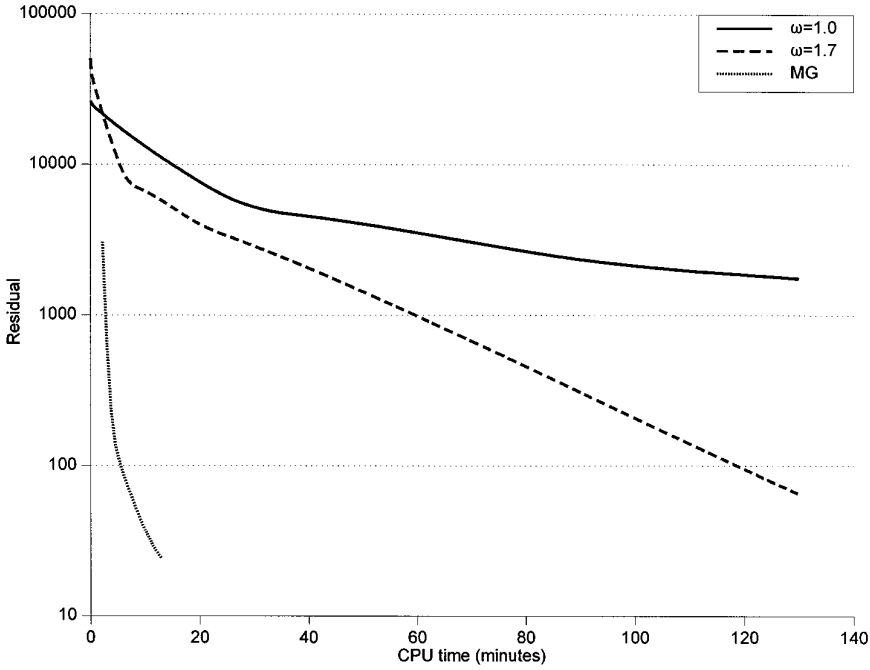

FIG. 5. Average residual norm as a function of invested CPU time for the developed Multigrid Algorithm, classical Gauss–Seidel ($\omega = 1.0$) (single grid), and Gauss–Seidel SOR ($\omega = 1.7$) (single grid).

of the absolute value of the residual of the discrete equation, see Eq. [8]) is monitored as a function of the CPU time (on an HP 9000/730 workstation). This is indeed the appropriate quantity to monitor as it is directly related to the error, and a given reduction of the residual norm implies the same reduction of the error in the solution.

Three cases are considered: Gauss–Seidel relaxation on a single grid ($\omega = 1.0$), Gauss–Seidel relaxation successive over relaxation (SOR) on a single grid ($\omega = 1.7$), and the multigrid solver. Note that by multigrid solver we mean only repeated coarse grid correction cycles, i.e., techniques such as full multigrid are not used. The initial approximation on the finest grid was the same for each case.

Figure 5 presents the results for the nerve and cuff model. Both single grid algorithms show a fast initial convergence after which the convergence speed rapidly levels off to its low asymptotic value. This is exactly the behavior described in the multigrid section. Using the multigrid coarse grid correction cycles, however, the initial high speed of convergence is maintained. In that case each error component is reduced on a grid at which it is nonsmooth. The eventual slowdown shown by the multigrid results is due to the limitation of machine accuracy. The presented calculations were performed using 7-digit accuracy. Eventually the

curves for the one-grid algorithms will stop declining completely as well, but due to the slow convergence they simply do not reach this stage until far beyond the time scale of Fig. 5. It was found that the computing time needed for a reduction of the error by a factor of 10 when considering the characteristic part of the curves was 15 times as long for Gauss–Seidel relaxation.

DISCUSSION

Multilevel techniques were shown to be a fast and efficient alternative method for the solution of the potential field in a 3-D model for nerve stimulation. For the nerve and cuff model it was demonstrated that, compared with previously used methods, the required computing time is reduced by more than one order of magnitude. The algorithm described in this paper has also been applied successfully in the computation of the potential field in spinal cord stimulation. This potential field was generated by an epidurally placed tripolar electrode. The finite difference model included the white and grey matter, cerebrospinal fluid and vertebral bone and the electrode. Compared to Gauss–Seidel SOR., the multigrid solver reduced the computing time from 20 min to only 1 min CPU time. These computing time gains obviously greatly enhance the possibilities for further practical research, e.g., parameter studies and electrode design. Further research can also be directed to other problems, i.e., the developed algorithm can be applied to surface stimulation for wound healing or activation of muscles, intramuscular stimulation, and pacing of various organs such as the heart (*12*).

At some point further improvement to the algorithm as presented merits investigation. First we will consider modeling. In the present work electrodes were modeled as current sources. This is indeed justified for an electrode that consists of only a single grid point. However, in reality the electrodes cover a group of grid points. In such cases the solution should satisfy the additional condition that all such points have the same potential. This was ensured iteratively by using a procedure of current scaling, which unfortunately has a slight but noticeable adverse effect on the asymptotic convergence speed. Therefore, the next step will be to really model the electrodes as voltage sources, i.e., as an internal Dirichlet boundary condition, or in fact as a group (small island) of points of fixed potential.

Second, future research may address some numerical/mathematical aspects of the coarse grid correction cycle that was designed. At present the first approximation is simply given on the target (finest) grid, and coarse grid correction cycles are used to converge. This implies that the total work needed to solve the problem to the level of the discretization error still depends logarithmically on the number of points, i.e., a finer target grid requires a larger error reduction starting from the same first approximation. This logarithmic factor could be removed using full multigrid techniques, i.e., using the coarser grids not only to accelerate convergence on the target grid, but also to generate an accurate first approximation. Also, attention could be directed to finding a computationally ''cheaper'' alternative to the full variational coarsening which was needed in relation to

large conductivity jumps as present in the nerve and cuff model. However crudely said, this coarsening puts a significant effort into a device that given the choice of coarsening used here only corrects 1/8 of the harmonic components in the error. It may be more profitable to consider alternative coarsening strategies, where more grid points are maintained on the coarser grids to improve the effect of the coarse grid correction. These matters are however of little importance for the present problem, where the goal was to obtain a fast solver for the given target grid problem. However, they may become more pressing if very dense grids are required.

APPENDIX

The interpolation used in the multigrid implementation is a modification of multilinear interpolation, i.e., the interpolation is performed such that it accounts for the continuity of $\sigma\nabla u$. This interpolation can be performed in the following way with each step using results of the previous step. Let the quantity to be interpolated, i.e., potential or coarse grid correction be denoted by u.

(1) First the values of the u in coarse grid points $\mathbf{I}$ are injected to the coinciding point $\mathbf{i}$ on the fine grid:

$$u_{\mathbf{i}} = u_{\mathbf{I}}. \qquad [21]$$

Next, in three steps values are obtained for all fine grid points that do not coincide with coarse grid locations.

(2) First, using the values given to points that coincide with coarse grid locations, values for u in the points $\mathbf{i}$ which are on the center of a link between two coarse grid locations are computed according to

$$u_{\mathbf{i}} = \frac{\overline{\sigma}_{k,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k,\mathbf{i}+1} u_{\mathbf{i}+1}}{\overline{\sigma}_{k,\mathbf{i}} + \overline{\sigma}_{k,\mathbf{i}+1}}, \qquad [22]$$

where $k = 1$, 2, or 3, if the link is in the x_1, x_2, or x_3 direction, and $\mathbf{i} \pm 1$ stands for $(i_1 \pm 1, i_2, i_3)$ if $k = 1$ and for $(i_1, i_2 \pm 1, i_3)$ if $k = 2$, etc.

(3) Then values for all fine grid points $\mathbf{i}$ located in the center of the rectangle formed by four coarse links, i.e., the center of a side of the ''cube'' formed by eight coarse grid points are computed according to

$$u_{\mathbf{i}} = \frac{\overline{\sigma}_{k_1,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k_2,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k_1,\mathbf{i}+1} u_{\mathbf{i}+1} + \overline{\sigma}_{k_2,\mathbf{i}+1} u_{\mathbf{i}+1}}{\overline{\sigma}_{k_1,\mathbf{i}} + \overline{\sigma}_{k_2,\mathbf{i}} + \overline{\sigma}_{k_1,\mathbf{i}+1} + \overline{\sigma}_{k_2,\mathbf{i}+1}}, \qquad [23]$$

where k_1 and k_2 stand for the two directions which define the plane in which the rectangle lies.

(4) The last step consists of giving a value to the remaining fine grid points. These points are located in the center of a coarse grid cube:

$$u_{\mathbf{i}} = \frac{\overline{\sigma}_{k_1,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k_2,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k_3,\mathbf{i}} u_{\mathbf{i}-1} + \overline{\sigma}_{k_1,\mathbf{i}+1} u_{\mathbf{i}+1} + \overline{\sigma}_{k_2,\mathbf{i}+1} u_{\mathbf{i}+1} \overline{\sigma}_{k_3,\mathbf{i}+1} u_{\mathbf{i}+1}}{\overline{\sigma}_{k_1,\mathbf{i}} + \overline{\sigma}_{k_2,\mathbf{i}} + \overline{\sigma}_{k_3,\mathbf{i}} + \overline{\sigma}_{k_1,\mathbf{i}+1} + \overline{\sigma}_{k_2,\mathbf{i}+1} + \overline{\sigma}_{k_3,\mathbf{i}+1}}. \qquad [24]$$

Note that this step uses the result of the steps 1, 2, and 3.

Together the steps outlined above describe the interpolation, i.e., they form the recipe to obtain values for all fine grid points from a function given on the coarse grid points. An alternative to carrying out these steps each time is to realize that they together define an interpolation matrix I_H^h, which can be computed at the start of the algorithm and stored. Its transpose can then be used as restriction operator I_h^H.

REFERENCES

1. Coburn, B. Electrical stimulation of the spinal cord: Two-dimensional finite element analysis with particular reference to epidural electrodes. *Med. Biol. Eng. Comp.* **18,** 573–584 (1980).
2. Coburn, B., and Sin, W. K. A theoretical study of epidural electrical stimulation of the spinal cord—Part 1: Finite element analysis of stimulus fields. *Trans. Biom. Eng.* **32,** 971–977 (1985).
3. Struijk, J. J., Holsheimer, J., van Veen, B. K., and Boom, H. B. K. Epidural spinal cord stimulation: Calculation of field potentials with special reference to dorsal column nerve fibers. *IEEE Trans. Biomed. Eng.* **38,** 104–110, (1991).
4. Struijk, J. J., Holsheimer, J., Barolat, G., He, J., and Boom, H. B. K. Paresthesia thresholds in spinal cord stimulation: A comparison of theoretical results with clinical data. *IEEE Trans. on Rehab. Eng.* **1,** 101–108 (1993).
5. Fang, Z. P., and Mortimer, J. T. Selective activation of small motor axons by quasitrapezoidal current pulses. *IEEE Trans. Biomed. Eng.* **38,** 168–174 (1991).
6. Goodall, E. V., Kosterman, L. M., Holsheimer, J., and Struijk, J. J. Modelling study of activation and propagation delays during stimulation of peripheral nerve fibers with a tripolar cuff electrode. *IEEE Trans. on Rehab. Eng.* **3,** 272–282 (1995).
7. Sweeney, J. D., Ksienski, D. A., and Mortimer, J. T. A nerve cuff technique for selective excitation of peripheral nerve trunk regions. *IEEE Trans. Biomed. Eng.* **37,** 706–715 (1990).
8. Rijkhoff, N. J. M., Holsheimer, J., Koldewijn, E. L., Struijk, J. J., Van Kerrebroeck, P. E. V., Debruyne, F. M. J., and Wijkstra, H. Selective stimulation of sacral nerve roots for bladder control; a study by computer modeling. *IEEE Trans. Biomed. Eng.* **41,** 413–424, (1994).
9. McNeal, D. R. Analysis of a model for excitation of myelinated nerve. *IEEE Trans. Biomed. Eng.* **23,** 329–337 (1976).
10. Brandt, A. Multi-level adaptive solutions to boundary value problems. *Math. of Comp.* **31,** 333–390 (1977).
11. Brandt, A. "1984 Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," GMD Studien No. 85, Gesellschaft für Mathematik und Datenverarbeitung MBH, Bon, Germany, 1984.
12. Alcouffe, R. E., Dendy, J. R., and Brandt, A. The multigrid solution for problems with strongly discontinuous and anisotropic coefficients. *SIAM J. Sci. Comput.* **2,** 430–454 (1981).
13. Kettler, R., and Wesseling, P. Aspects of multigrid for problems in three dimensions. *Applied Math. in Computation* **19,** 159–168 (1986).
14. Pollard, A. E., and Henriquez, C. S. "A Comparison of Iterative Methods for the Determination of the Interstitial Potential Distribution with the Bidomain Model," Proc. of the 14th Annual Conf. IEEE EMBS, pp. 602-603, Paris, 1992.