Fast Anisotropic High Resolution Finite Element Head Modeling in EEG/MEG-source localization

C. H. Wolters[1,2], M. Kuhn[3,4], A. Anwander[2] and S. Reitzinger[4]

[1]*MPI for Mathematics in the Sciences, Leipzig, Germany,* [2]*MPI of Cognitive Neuroscience, Leipzig, Germany,* [3]*Bruker Saxonia Analytik GmbH, Leipzig, Germany,* [4]*Institute for Comp. Math., J. Kepler Univ., Linz, Austria*

Abstract

Accuracy and time play an important role in EEG/MEG-source localization. High resolution finite element head modeling allows the inclusion of tissue conductivity inhomogeneities and anisotropies. The finite element approach as a forward method within the inverse problem leads to a sparse, large scale, linear equation system with many different right hand sides to be solved. The presented solution process is based on a parallel algebraic multigrid method. It is shown that very short computation times can be achieved through the combination of the multigrid technique and the parallelization on distributed memory computers. The iterative solver approach is shown to be stable towards realistic high resolution modeling of skull and white matter anisotropy. A solver time comparison to a classical parallel Jacobi preconditioned conjugate gradient method is given.

1 Introduction

A bottleneck for a broader application of high resolution realistically shaped anisotropic finite element (FE) volume conductor modeling to EEG/MEG inverse source reconstructions is the time for solving the linear equation system arising from the FE discretization. Algebraic MultiGrid (AMG) methods use only single grid information while mostly preserving the properties of the geometric MG, i.e., being of optimal order with respect to arithmetic costs and memory requirement and being an efficient preconditioner for the CG method (AMG-CG). Even if AMG-CG was shown to be very fast in comparison to standard methods [1], additional speedup is required. In [2], we described for realistically shaped isotropic high resolution tetrahedra and cube head models how the latter can be achieved by using a parallel computer with a moderate number of processors. Within this article, we show that the approach is stable towards realistic head tissue anisotropy. We make use of the high resolution realistically shaped tetrahedral finite element head model with 1:10 anisotropic layers skull and white matter, whose generation process is described in [3].

2 Methods

2.1 Algebraic Multigrid Method

In the following, we will solve the FE equation system $\mathsf{K}_h\underline{\Phi} = \underline{\mathsf{J}}$, where K_h denotes the FE stiffness matrix, $\underline{\mathsf{J}}$ a right-hand-side vector (current source density vector) and $\underline{\Phi}$ the unknown potential distribution. We denote the FE space by $\mathbb{V}_h$. h is related to the fine grid and H to a "coarser grid", which is generated by means of an exclusive use of the algebraic information in K_h. Each AMG algorithm consists of the following components:

(a) Coarsening: define the splitting $\omega_h = \omega_C \cup \omega_F$ of ω_h (the index set of nodes) into sets of coarse and fine grid nodes ω_C and ω_F, respectively.

(b) Transfer operators: prolongation $\mathfrak{P}_h : V_H \mapsto V_h$ and restriction $\mathfrak{R}_h := \mathfrak{P}_h^T$.

(c) Definition of the coarse matrix by Galerkin's method, i.e., $\mathsf{K}_H := \mathfrak{R}_h\mathsf{K}_h\mathfrak{P}_h$.

(d) Appropriate smoother for the considered problem class.

Algorithm 1 (Parallel) $V(\nu_F, \nu_B)$-cycle MG$(\mathsf{K}_h, \underline{\Phi}, \underline{\mathsf{J}})$

if COARSEGRID **then**
  $\underline{\Phi} \Leftarrow$ DIRECTSOLVE $(\mathsf{K} \cdot \underline{\Phi} = \underline{\mathsf{J}})$
else
  $\underline{\widetilde{\Phi}} \Leftarrow \nu_F$ TIMES SMOOTH$(\mathsf{K}_h, \underline{\Phi}, \underline{\mathsf{J}})$
  $\underline{\mathsf{d}} \leftarrow \underline{\mathsf{J}} - \mathsf{K}_h \cdot \underline{\widetilde{\Phi}}$
  $\underline{\mathsf{d}}^H \leftarrow \mathfrak{P}^T \cdot \underline{\mathsf{d}}$
  $\underline{\mathfrak{w}}^H \leftarrow 0$
  $\underline{\mathfrak{w}}^H \Leftarrow$ MG$(\mathsf{K}_H, \underline{\mathfrak{w}}^H, \underline{\mathsf{d}}^H)$
  $\underline{\mathfrak{w}} \leftarrow \mathfrak{P} \cdot \underline{\mathfrak{w}}^H$
  $\underline{\widehat{\Phi}} \leftarrow \underline{\widetilde{\Phi}} + \underline{\mathfrak{w}}$
  $\underline{\Phi} \Leftarrow \nu_B$ TIMES SMOOTH$^T(\mathsf{K}_h, \underline{\widehat{\Phi}}, \underline{\mathsf{J}})$
end if

After the proper definition of the prolongation and coarse grid operators for our application (see [1; 2]), a matrix hierarchy can be setup in a recursive way and a multigrid cycle can be assembled, which is shown in Algorithm 1. Therein the variable COARSEGRID denotes the level where a direct solver is applied. For our application we use AMG-CG, i.e., AMG is applied as a preconditioner for the CG

Algorithm 2 (Par.) PCG algorithm $\text{PCG}(\mathsf{K}_h, \underline{\Phi}, \underline{\mathsf{J}}, \mathsf{C}_K)$

$\underline{\mathsf{r}} \leftarrow \underline{\mathsf{J}} - \mathsf{K}_h \underline{\Phi}$
$\underline{\mathfrak{w}} \Leftarrow \mathsf{C}_K^{-1} \cdot \underline{\mathsf{r}}$
$\underline{\mathfrak{s}} \leftarrow \underline{\mathfrak{w}}$
$\gamma \Leftarrow \langle \underline{\mathfrak{w}}, \underline{\mathsf{r}} \rangle$
repeat
  $\underline{\mathsf{v}} \leftarrow \mathsf{K}_h \cdot \underline{\mathfrak{s}}$
  $\alpha \Leftarrow \gamma / \langle \underline{\mathfrak{s}}, \underline{\mathsf{v}} \rangle$
  $\underline{\Phi} \leftarrow \underline{\Phi} + \alpha \underline{\mathfrak{s}}$
  $\underline{\mathsf{r}} \leftarrow \underline{\mathsf{r}} - \alpha \underline{\mathsf{v}}$
  $\underline{\mathfrak{w}} \Leftarrow \mathsf{C}_K^{-1} \cdot \underline{\mathsf{r}}$
  $\gamma \Leftarrow \langle \underline{\mathfrak{w}}, \underline{\mathsf{r}} \rangle$
  $\beta \leftarrow \gamma / \gamma_{\text{OLD}}$, $\gamma_{\text{OLD}} \leftarrow \gamma$
  $\underline{\mathfrak{s}} \leftarrow \underline{\mathfrak{w}} + \beta \underline{\mathfrak{s}}$
until TERMINATION

method [2]. For the m-$V(\nu_F, \nu_B)$-cycle AMG preconditioned CG method, the operation $\underline{\mathfrak{w}} = \mathsf{C}_K^{-1} \underline{\mathsf{r}}$ is realized by m calls of $\text{MG}(\mathsf{K}_h, \underline{\mathfrak{w}}, \underline{\mathsf{r}})$. For the Jacobi-preconditioner, it is $\mathsf{C}_K = \mathsf{D}$ with D the diagonal of K_h. The Preconditioned CG (PCG) method is shown in Algorithm 2.

2.2 Data Partitioning

The aim of parallelization is to split both data and operations to the P processors available. The consistency of the algorithms is preserved by message passing. In our case, the parallelization is based on a non-overlapping domain decomposition, i.e., we decompose the finite element head model $\overline{\Omega}$ into P subdomains $\overline{\Omega}_s$ such that $\overline{\Omega} = \bigcup_{s=1}^{P} \overline{\Omega}_s$ with

$$\Omega_s \cap \Omega_q = \emptyset \quad \forall q \neq s, \quad s, q = 1, \ldots, P$$

holds. Each subdomain Ω_s is discretized by a mesh $\tau_{h,s}$ such that the whole triangulation

$$\tau_h = \bigcup_{s=1}^{P} \tau_{h,s}$$

of Ω forms a conforming mesh. A global FE space $\mathbb{V}_h$ is defined with respect to τ_h and the local spaces $\mathbb{V}_{h,s}$ are restrictions of $\mathbb{V}_h$ onto $\tau_{h,s}$. Since we are interested in an "element-wise-" in contrast to a "node-wise-" distribution, the dual graph of the FE mesh is partitioned. For mesh-partitioning, we used the software package METIS [4]. The results were achieved in a few seconds on a single processor SGI workstation. Figure 1 shows a visualization of the partitioned FE head model for 4 and for 12 processors, using PMVIS [5].

2.3 Parallel AMG

The mapping of a vector $\underline{\Phi}_h \in \mathbb{R}^{N_h}$ (N_h denotes the number of vertices of the FE model) in global numbering onto a local vector $\underline{\Phi}_s \in \mathbb{R}^{N_s}$ in subdomain $\overline{\Omega}_s$ ($s = 1, \ldots, P$) is

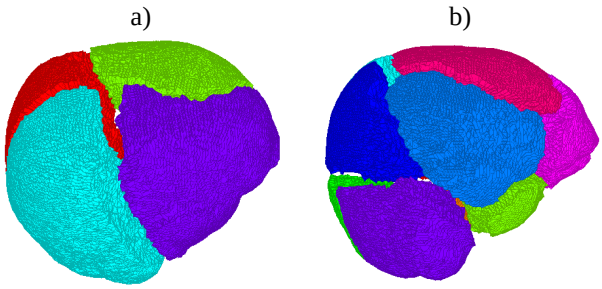

Figure 1: *Realistic tetrahedral FE head model, 892115 elements, partitioned with METIS and visualized with PMVIS: a) for 4 processors b) for 12 processors.*

represented symbolically by subdomain connectivity matrices $\mathcal{A}_s : \mathbb{R}^{N_h} \mapsto \mathbb{R}^{N_s}$ with entries

$$\mathcal{A}_s^{[i,j]} := \begin{cases} 1 & \text{if} \quad j = \text{LOC2GLOB}(i) \\ 0 & \text{else} \end{cases} \qquad \forall i \in \omega_s \,,\ \forall j \in \omega_h$$

where $\text{LOC2GLOB}(\cdot)$ maps a local index to the global index. The transpose $\mathcal{A}_s^T$ of these binary matrices $\mathcal{A}_s$ maps a local vector back onto the global one. The index set of all those subdomains to which an unknown $\Phi_h^{[j]}$, $j \in \omega_h$ belongs, is denoted by

$$\sigma^{[j]} := \{ s \mid \Phi_h^{[j]} \in \overline{\Omega}_s \} \,. \tag{1}$$

We store the data related to the i^{th} node in the subdomain Ω_s if $s \in \sigma^{[i]}$. This approach results in local data denoted by index s of two types [6]: *accumulated data* (vector $\underline{\Phi}$, matrix $\mathfrak{P}$) represented by

$$\underline{\Phi}_s := \mathcal{A}_s \cdot \underline{\Phi} \,, \qquad \mathfrak{P}_s := \mathcal{A}_s \cdot \mathfrak{P} \cdot \mathcal{A}_s^T$$

and *distributed data* (vector $\underline{\mathsf{d}}$, matrix K_h) represented by

$$\underline{\mathsf{d}} = \sum_{s=1}^{P} \mathcal{A}_s^T \cdot \underline{\mathsf{d}}_s, \qquad \mathsf{K}_h := \sum_{s=1}^{P} \mathcal{A}_s^T \cdot \mathsf{K}_s \cdot \mathcal{A}_s \,.$$

It turns out, that in Algorithms 1 and 2, the functionals are represented as distributed data ($\underline{\mathsf{J}}$, $\underline{\mathsf{v}}$, $\underline{\mathsf{r}}$, $\underline{\mathsf{d}}$, K_h), whereas functions are represented as accumulated data ($\underline{\Phi}$, $\underline{\mathfrak{s}}$, $\underline{\mathfrak{w}}$, $\mathfrak{P}$). The local FE accumulation with respect to $\mathbb{V}_{h,s}$ produces automatically distributed matrices K_s. If an accumulated matrix $\mathfrak{M}$ fulfills the condition

$$\forall i \in \omega_h \,, \forall j \in \omega_C \,:\, \sigma^{[i]} \not\subseteq \sigma^{[j]} \implies \mathfrak{M}^{[i,j]} = 0, \tag{2}$$

then the operations $\underline{\mathfrak{w}} = \mathfrak{M} \cdot \underline{\mathfrak{w}}^H$, $\underline{\mathsf{d}}^H = \mathfrak{M}^T \cdot \underline{\mathsf{d}}$ and $\mathsf{K}_H = \mathfrak{M}^T \mathsf{K}_h \mathfrak{M}$ can be performed locally without any communication [6]. In AMG the coarsening and prolongation operators are components which can be chosen. The main idea in the design of parallel AMG is to choose these components such that the resulting prolongation operators $\mathfrak{P}$ are of accumulated type satisfying the pattern condition (2). For a detailed discussion we refer to [7]. Now we

observe that Algorithm 1 and Algorithm 2 are also the appropriate parallel formulations, where double-line arrows "$\Leftarrow$" indicate that communication is required for the corresponding operation. In Algorithm 1, the coarse grid system is accumulated globally once in the setup phase. During the iteration only a vector has to be assembled for computing the coarse grid solution. Furthermore, the smoother requires communication and has to be adapted appropriately. We use a Gauss-Seidel smoother for the inner nodes and a Jacobi smoother for the interface nodes [6]. The Jacobi-smoother involves a vector conversion from distributed to accumulated type, i.e., one next neighbor communication across interfaces is required per smoothing step.

3 Results

We run simulations on an SGI ORIGIN 2000 with R10000, 195 MHz processors and overall 6GB of main memory. The FE space $\mathbb{V}_h$ consisted of piecewise linear Ansatz-functions. For the AMG-CG, we used the 1-$V(1,1)$-cycle AMG-preconditioner. The factorization in Algorithm 1 was carried out, if the size of the coarsest grid (COARSEGRID) in the preconditioner-setup was below 800. The coarse system is solved by means of a Cholesky-factorization. The process of determining the index set (1) for each node and scattering the data (vertices, elements, tensor-valued conductivities) to the processors, both of which are carried out by the root processor, and the local arrangement of nodes to groups according to their index-set and the allocation of corresponding communicator groups takes about half a minute. The setup of the AMG on 1 processor took 39 seconds and parallelized on 12 processors 7 seconds. The above operations only have to be performed once per head model with regard to the inverse problem. The zero starting vector $\Phi_0 = \vec{0}$ was chosen for the iterative solution process. The solver process was stopped after the i^{th} iteration if the relative error in the controllable $\mathsf{K}_h\mathsf{C}_K^{-1}\mathsf{K}_h$-energy norm was below $\epsilon = 10^{-8}$. Figure 2 shows the wall-clock time of the presented solvers for the chosen anisotropic FE head model.

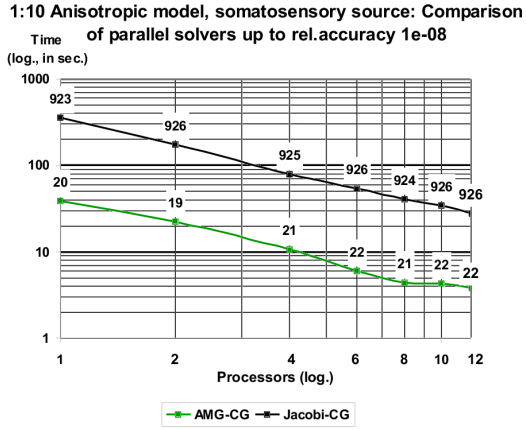

Figure 2: *SGI ORIGIN: Wall-clock time from 1 to 12 processors for the solver part of the parallel AMG-CG compared to the parallel Jacobi-CG up to an accuracy of* 10^{-8}. *The numbers of iterations are shown over the curves.*

4 Conclusions

An efficient and memory-economical way was presented to compute realistically shaped high resolution anisotropic FE forward solutions within the EEG/MEG inverse problem. Very short calculation times were achieved through the combination of AMG preconditioning techniques and its parallelization on distributed memory platforms. We compared the presented AMG-CG with the Jacobi-CG, the latter being a well-known solver method in FE-based source localization. If the Jacobi-CG on a single processor is taken as a reference, we achieved a speedup of 95 for a realistically shaped high resolution 1:10 anisotropic tetrahedra head model with 147287 nodes when comparing to the parallel AMG-CG on 12 processors, 9.3 through multigrid preconditioning and 10.2 through parallelization on 12 processors. The solver process was thus shown to be stable with respect to realistic tissue anisotropy. The partitioning of the dual graph of a convex head geometry generally leads to a relatively large percentage of interface nodes. Nevertheless, for the examined moderate processor numbers between 1 and 12, the AMG-preconditioner was found to be stable, i.e., a sensible increase of the number of subdomains did not result in a deterioration of the AMG-preconditioner and thus an increasing need for iterations. A first performance test on a Linux PC-cluster with 100 MBit ethernet showed very good parallelization speedups and, because of the 1 GB processors, a strongly reduced computation time compared to the presented results on the SGI ORIGIN 2000. The presented algorithms can thus be used on a simple PC-cluster.

Acknowledgements

This work was supported by Prof.E.Zeidler, by the Leibniz-Prize of the German Research Foundation awarded to Prof.A.D.Friederici, by the IST-program of the European Community, project SIMBIO (http://www.simbio.de), and by the Austrian Science Foundation (FWF).

References

[1] Wolters et al., http://biomag2000.hut.fi/, 2000.
[2] Wolters et al.,*Comp.Vis.Sci.*, in press, 2002.
[3] Anwander et al.,*Biomag2002, Proceedings*.
[4] http://www-users.cs.umn.edu/ ∼karypis, 1998.
[5] http://www-users.cs.umn.edu/∼oztekin, 1998.
[6] G. Haase, Teubner, 1999.
[7] Haase et al.,*SIAM J. Sci.Comp.*, in press, 2002.