Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Three-Dimensional Finite-Difference EEG forward problem solution on High Performance Computers

S. Mininel, F. Vatta, A. Collaone, P. Bruno and P. Inchingolo, *Member, IEEE*

Abstract— **EEG forward problem solution using numerical head models with the same resolution and geometry as that available from MRI is desirable. This implies dealing with realistic head models of over 2 million elements, for which problem solution has so far been impractical due to issues of computation time and memory. This paper investigates the possibilities given by High Performance Computing (HPC) to obtain efficient EEG forward problem solution with high resolution head models of over 2 million elements at reduced computation time. In this paper, a Finite Difference forward problem solution based on HPC is proposed and tested with parallel implementations of different complexity. Solution feasibility with different HPC schemes is analyzed to individuate, by cost-effectiveness, the most appropriate system configurations allowing the best performances. Results indicate that a feasible solution based on a cluster of 8 processors is convenient, obtaining computation times not higher than 2 minutes. Increasing the cluster size above 32 processors gives no significant improvement in the computation times.**

I. Introduction

NEURAL current sources in the brain produce scalp surface potentials measurable using electro-encephalography (EEG). Scalp-recorded potentials can be used to non-invasively estimate the localization of underlying neural activity in the specific patient's head [1]. This means solving the EEG inverse problem, i.e., seeking optimum neural source parameters responsible for a given potential distribution measured at the scalp electrodes. The inverse problem can be solved by an iterative process of EEG forward problems, i.e., the calculation of potential distribution on the scalp due to known sources. Given the aim of spatial and temporal high resolution in EEG source reconstruction, the investigation of efficient methods for computing the EEG forward problem is therefore of great significance.

In EEG forward problem solution we use the concept of "equivalent source" (e.g., a single current dipole), in conjunction with a specified volume conductor model of the head, which mathematically expresses the relationship between the equivalent source inside the volume and the corresponding surface potentials. The head model should be a precise and realistic 3-D representation of the electrical properties of subject's head in terms of shape and electric conductivities [2, 3], and characterized also by a high spatial resolution [1]. In the past decades, numerical computational methods such as the Boundary Element Method (BEM), the Finite Element Method (FEM), and the Finite Difference Method (FDM) have been used for EEG forward problem solution with realistic head models. Each method provides certain computational or practical benefits and restraints [4].

A numerical model of the head with the same resolution and geometry as that available from MRI is a desiderable objective, since numerical solution accuracy requires high spatial resolution [5]. Given the spatial resolution presently available with volumetric MRI scans, a normal sized adult head model would consist of over 2 million elements [5]. EEG forward problem solution using models of this magnitude has so far been impractical due to issues of computation time and memory. The FDM, which has been shown to yield accurate solutions to the EEG forward problem in head models with over 500,000 elements [6], shows useful advantages over the other methods due to the easy construction of the model from image-derived geometric information, easy allowance for electric anisotropic conduction and suitability for parallel processing which is increasingly available and affordable [4]. The last FDM property is extremely useful when dealing with models of high spatial resolution, e.g., with 2 million elements or more. A pre-conditioning method has been proposed enabling the forward problem to be solved using realistic head models of 2 million elements, leading to solutions in about 30 min on a 1 GHz Pentium III [5], although times would probably be consistently shorter with state of the art workstations. The aim of this paper is to investigate the possibilities given by High Performance Computing (HPC) in EEG forward problem solution with the purpose of obtaining an efficient problem solution with high-spatial-resolution realistic head models with over 2 million elements, with reduced computational time. In this paper, a Finite Difference EEG forward problem solution based on HPC is proposed and tested with parallel implementations of different complexity, to analyze the feasibility of problem solution by means of different HPC resources schemes and to individuate, by means of cost-effectiveness analysis, the most appropriate HPC configurations allowing the best solution performances for EEG forward problem solution.

Manuscript received April 24, 2006. This work was supported in part by MIUR, Italy, National project PRIN 2004 grant n. 2004090530, by University of Trieste and by the Interuniversity Consortium CINECA, Casalecchio di Reno (BO), Italy.

P. Bruno, A. Collaone, P. Inchingolo, and S. Mininel are with DEEI and with the BRAIN Center, University of Trieste, Trieste, 34100 Italy (e-mail: {bruno, collaone, paolo.inchingolo, mininel} @bioing.units.it).

F. Vatta is with DEEI and with the Interdepartmental Center BRAIN of the University of Trieste, Trieste, 34100 Italy (phone: +39 040 5587120; fax: +39 040 5583460; e-mail: federica.vatta@bioing.units.it).

1-4244-0033-3/06/$20.00 ©2006 IEEE.

II. METHODS

A. Finite Difference Method

The forward problem in EEG source analysis is governed by Poisson's differential equation

$$\nabla \cdot (\sigma(\mathbf{r}) \nabla V(\mathbf{r})) = I(\mathbf{r}) = -I\delta(\mathbf{r} - \mathbf{r}_2) + I\delta(\mathbf{r} - \mathbf{r}_1) \quad (1)$$

where V is the electric potential function, σ is the electric conductivity, I is the applied current source density at a point $\mathbf{r}$ (x, y, z). When I is generated by a current dipole the right hand side of eq. 1 is expressed as the point current source and sink of the dipole source, providing I and $-I$ at positions $\mathbf{r}_1$ and $\mathbf{r}_2$, where $\delta(\mathbf{r}\text{-}\mathbf{r}_*)$ denotes the 3-D delta function centered at $\mathbf{r}_*$. The FD head model used in this work is based on cubic volume elements (voxels) centered on nodes placed on a regular Cartesian grid. Discretization of eq. 1 on this grid leads to the following equation for each voxel (i,j,k):

$$\begin{aligned} &\sigma_{mi+1}\left(V_{i+1,j,k} - V_{i,j,k}\right) + \sigma_{mi-1}\left(V_{i-1,j,k} - V_{i,j,k}\right) + \\ &+ \sigma_{mj+1}\left(V_{i,j+1,k} - V_{i,j,k}\right) + \sigma_{mj-1}\left(V_{i,j-1,k} - V_{i,j,k}\right) + \\ &+ \sigma_{mk+1}\left(V_{i,j,k+1} - V_{i,j,k}\right) + \sigma_{mk-1}\left(V_{i,j,k-1} - V_{i,j,k}\right) = \frac{I_{i,j,k}}{h} \end{aligned} \quad (2)$$

where $V_{i,j,k}$ is the potential at the voxel node (i,j,k), $V_{i+1,j,k}$, $V_{i-1,j,k}$, $V_{i,j+1,k}$, $V_{i,j-1,k}$, $V_{i,j,k+1}$, $V_{i,j,k-1}$, are the potentials at adjacent voxel nodes, $I_{i,j,k}$ is the total current flow into the central node, and h is the edge length of each voxel. $I_{i,j,k}$ is zero for every node except those where a current source or sink is located. σ_m are the conductivities between the node (i,j,k) and each adjacent node and are given by the series combination of the conductivities of two adjacent half-voxels. For each of the N nodes of the discretized volume head model we obtain a linear equation given by eq. 2; thus, the potential value at each node (i,j,k) is written as a linear combination of the potentials of six adjacent nodes. As result, the FDM generates a large system of linear equations:

$$Ax = b \quad (3)$$

where A is the $N \times N$ system matrix containing model's conductivities, x is a column vector of the potentials at the N nodes, and b is a column vector of current sources at the nodes. Matrix A is sparse with each row consisting of nonzero diagonal elements and, at most, six nonzero off-diagonal locations. Iterative solvers for large sparse systems of linear equations are needed to solve this linear system and to obtain the potentials for a given source.

B. HPC implementation

The HPC implementation has been designed to build and solve efficiently the linear equations system of eq. 3, giving a high flexibility in the solvers choice and being able to run either on mono-processor personal computers or, in parallel, upon large HPC systems. Parallel computers are an adequate instrument for a consistent reduction in solution time for large scale problems solution, as the computational load is subdivided using more CPUs. Specific software libraries have been used at a high level of abstraction to leave "transparent" the low level calls and message exchanges between processors (CPUs), letting focusing on optimization and search for stable and accurate solution methods. The proposed solution application makes use of the PETSc open source libraries [7] and upon the MPI (Message Passing Interface) routines to manage the inter-processors communication.

In the proposed solution a typical parallelization strategy, named "divide and conquer", has been adopted [8]. Each CPU solves the problem in its sub-domain and MPI is used to exchange values necessary to each CPU for contour values (see Fig. 1). The large linear system Ax=b is thus fractionated in a number of smaller sub-problems, solved iteratively. To this end, a number of "ghost points" must be created to store and exchange data between "neighbor" CPUs. MPI routines take care of multiprocessor synchronizations so that each CPU, at the end of iteration, waits for his neighbors to pass the newly computed borders values. Store and pass values in the "ghost points" takes time, so, the use of N CPUs does not lead to computation time reduction of a factor N. In addition, there is a fraction of the code (thus of CPU time) that is inherently sequential and cannot be speeded up by parallelization. Optimal CPU number is not therefore – a priori – the maximum number available. Optimal CPU number issue is better exploited in Section III. At the end of the iterative solution, when required error tolerance is reached, the sub-domains arrays are "merged" to give the solution to the full problem. Merging is again performed by opportune MPI functions, called by higher level PETSc functions. To minimize the "dead times", an optimal domain partitioning has been chosen, to have a minimum of ghost cells (and thus of inter-processors messages) and a balanced computational load for all the CPUs. The number of employed CPUs depends upon the problem size, as excessively simple tasks for the CPUs mean wasting more time in message passing than in computing. Sub-domain partitioning has been designed to be easily (again by PETSc routines) configurable according to problem complexity.

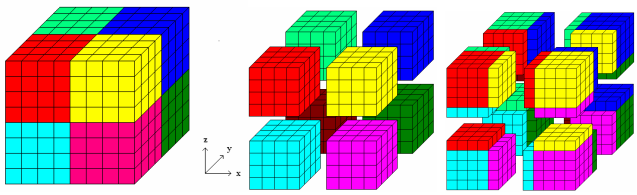

Fig. 1. Parallelization: the problem domain is divided in a suitable number of sub-domains (left). Each CPU is assigned a sub-domain to be solved locally (center). A number of "ghost points" must be created in each sub-domain, where values computed by other CPUs will be stored (exchanged by MPI) to give correct border values to each sub-domain (right).

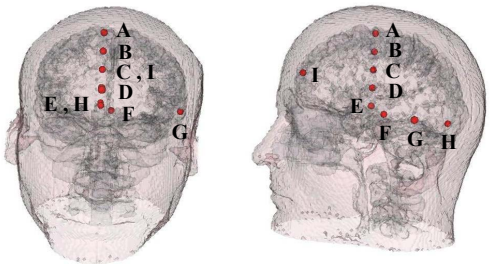

Fig. 2. Frontal (left) and lateral (right) view of the adopted 9 source positions (A, B, C, D, E, F, G, H, I).

The SP5 HPC system has been used in this work with the following characteristics: 512 processors on 64 nodes; each processor is an IBM Power5 dual-core 64 bit superscalar microprocessor with 120 floating point registers; clock frequency of 1.9 GHz, large Level 1, Level 2 and Level 3 cache memory; total RAM is 1.2 TB and peak performance 3,89 Tflops/s.

C. *Performed tests*

We used a patient-specific realistic head model constructed out of 115 sagittal MRI scans (256×256 pixels), resampling each scan to obtain the same spatial resolution along any direction. A semiautomatic tissue classifier was used to identify the following head tissue types (model compartments): scalp, skull, cerebrospinal fluid (CSF), grey matter, white matter and internal air. Conductivity values were assigned as follows: 0.35 S/m for the scalp, 1.0 S/m for CSF, 0.33 S/m for gray matter, 0.2 S/m for white matter and 6×10^{-5} S/m for internal air [8]. The head volume was discretized with a resolution of 2×2×2 mm, leading to a 3-dimensional uniform grid of 128×128×115 elements, corresponding to about 2 million elements.

The EEG forward problem solution has been computed considering different HPC schemes with 1, 2, 4, 8, 16, 32 and 64 processors. The following iterative solution methods have been analyzed: SOR; Symmetric SOR (SSOR); Conjugated Gradients (CG); Bi-Conjugated Gradients (BiCG); Squared Bi-Conjugated Gradients (BCGS). BCGS method has finally been chosen since it resulted to converge in a larger iterations number but with less memory needs and in shorter time. 9 dipole sources with different locations in the brain, have been considered, shown in Fig. 2, with two different orientations, either parallel (e.g., C_*x*) or orthogonal (e.g., C_*y*) to model base, thus totaling 18 test conditions. Although the rate of convergence of the CG method depends on the system matrix eigenvalues distribution, the considered source positions are aimed to give some insight on HPC system performance variability for the different domain partitioning with the various processors configurations. The adopted tolerance value for relative error norm was set at 10^{-7}. The HPC system performance in EEG forward problem computation has been evaluated by means of number of iterations and computation time.

III. Results and Discussion

The FDM EEG forward problem solution was validated by means of a 3-concentric-shells head model for which analytical solutions were available [8], adopting the Successive Over-Relaxation (SOR) method. Then, the consistency of the parallel solution has also been verified.

The HPC system performance has first been evaluated in terms of iterations number and solution time for all the 18 test sources for the 7 processors configurations considered. Then, as the total solution time is strictly dependant on the error tolerance set and thus on total iterations number, to evaluate system performance we considered the time needed per iteration (ratio between total solution time and iterations number) and the iterations per second (inverse of above ratio). Results of iterations number per second in the test situations are reported in Table I, showing a remarkable stability, demonstrated by a relative mean deviation lower than 1%.

Fig. 3 shows results of the time needed per iteration as a function of processors number, with interpolation of the results average. Interpolation shows very good conformance to a hyperbolic curve. This was an expected result since in these tests the inherently sequential code was very limited. The Amdahl law [9] states that the speed-up obtainable in parallelization is limited by the amount of parallelism inherent in the algorithm. The non parallel code is typically mostly due to input/output routines. In the performed tests, building of the A matrix is parallelized and no hard-disk saving of final computation results has been considered to simulate a situation analogous to an EEG inverse problem solution, with a large number of forward solutions using the same system matrix A with only final disk saving of results at the end of the inverse solution process. It can also be observed in Fig. 3 that the intercept between the hyperbolic

TABLE I
ITERATIONS NUMBER / SOLUTION TIME [1/S]

Source	Processors number						
	1	2	4	8	16	32	64
A_*x*	1.49	3.05	5.98	11.82	25.79	62.70	92.53
A_*y*	1.49	3.05	5.97	11.84	25.78	62.68	92.31
B_*x*	1.49	3.05	5.98	11.87	25.76	62.81	92.21
B_*y*	1.48	3.05	5.98	11.82	25.77	62.70	92.12
C_*x*	1.49	3.04	5.97	11.80	25.79	62.84	92.55
C_*y*	1.48	3.01	5.99	11.77	25.81	62.59	92.23
D_*x*	1.49	3.01	5.99	11.83	25.82	62.77	92.38
D_*y*	1.49	3.02	5.97	11.70	25.80	62.62	92.44
E_*x*	1.49	3.01	5.99	11.73	25.83	62.22	92.16
E_*y*	1.48	3.02	5.98	11.67	25.83	62.67	92.66
F_*x*	1.49	3.02	5.96	11.72	25.81	62.92	92.12
F_*y*	1.48	3.00	5.96	11.65	25.80	62.65	92.66
G_*x*	1.48	3.01	5.98	11.62	25.79	62.73	92.12
G_*y*	1.48	3.01	5.97	11.67	25.83	62.71	91.88
H_*x*	1.48	3.02	5.99	11.56	25.79	62.92	92.32
H_*y*	1.48	3.02	5.99	11.65	25.76	62.60	92.40
I_x	1.49	3.04	5.96	12.12	25.72	62.14	90.74
I_y	1.49	3.05	5.98	12.01	25.78	62.87	92.33
Mean	1.49	3.03	5.98	11.77	25.79	62.68	92.23
Std. dev.	0.00	0.02	0.01	0.11	0.02	0.14	0.25

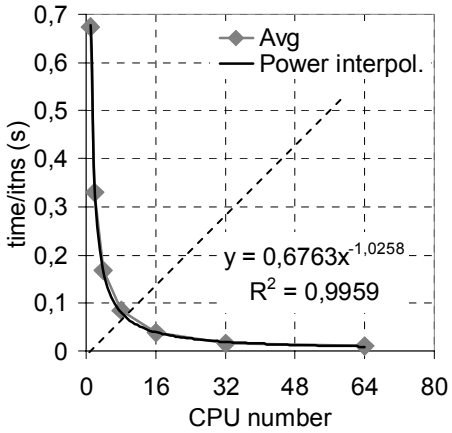

Fig. 3. Average time-per-iteration curve as function of CPUs number.

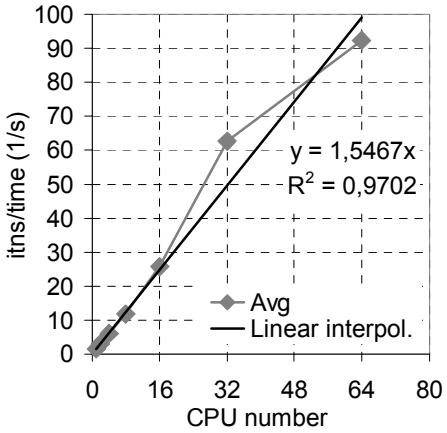

Fig. 4. Average iterations-per-second curve as function of CPUs number.

curve of time per iteration as function of CPUs number with the first bisector line holds for a number of 8 CPUs. This shows up as the most cost-effective processors configuration, since a greater number of parallel processors leads to less drastic reduction in solution times.

Fig. 4 shows the relationship between processors number and average iterations per second (iterations number/total computational time). The use of the average of the 18 test situations is justified by small dispersion of the reported values. It can be observed that the relationship is almost perfectly linear with up to 16 CPUs. For 64 CPU (as expected from Amdahl's law) speedup lowers. Yet, there is a curious aspect regarding the performance with 32 processors: with the doubling of CPUs number from 16 to 32, the time needed per iteration is consistently more than halved (from 38 to 16 ms).

We are currently investigating the reasons for this unexpected performance, testing with other models of higher spatial resolution (256x256x115, i.e., of 7.5 million elements), a larger processors number (128 or more) and different topological partitions of the sub-domains.

The most likely explanation is that the larger sub-domains in the 16 processors configuration have local linear systems (Ax=b) with a system matrix which is poorly conditioned (large spectral radius). The splitting into smaller sub-domains, as with 32 processing elements, may lead to better conditioned sub-systems. We are testing this hypothesis with use of different domain partitioning methods from the default operated by PETSc libraries.

It must be noted that, beyond cost-effectiveness, while using a shared HPC resource (as the CINECA SP5) optimal CPUs number used also depends on system availability. On a medium/large sized system with 512-1024 CPUs, asking the contemporary use of 64, 128 or more CPUs to the queue system becomes a heavy request that most likely cannot be received immediately, unless the account on the system has special priority privileges (usually too much expensive). Furthermore, the request for using up to 32 CPUs can usually be fulfilled in short time on such HPC systems.

IV. Conclusion

Results of this study show that a parallelized EEG forward problem solution with 8 processors is an optimal solution in terms of cost-effectiveness, with obtained average solution times of about 1.5 minutes for a realistic head model of about 2 million elements. If larger HPC systems are available, results obtained show that using up to 32 processors can lead to a substantial reduction of computing times, with results reduced to 16 seconds on average with the same models that can be surprisingly good in terms of computational speed-up. Increasing the cluster size above 32 processors gives no significant improvement in the computation times. The findings of the present paper could be generalized to other similar problems requiring parallel computing solutions, at least in method used for evaluating HPC resources needed.

References

[1] S. Baillet, J.C. Mosher, R.M. Leahy, "Electromagnetic brain mapping," *IEEE Signal Processing Magazine*, vol. 18(6), 2001, pp. 14-30.

[2] B. N. Cuffin, "EEG dipole source localization," *IEEE Trans. Biomed. Eng. Mag.*, vol. 17, pp. 118–122, 1998.

[3] F. Vatta, P. Bruno, P. Inchingolo, "Multiregion bicentric-spheres models of the head for the simulation of bioelectric phenomena," *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 384–389, 2005.

[4] L. Jing, S. Zhu, B. He, "A Finite Difference Method for Solving the Three-Dimensional EEG Forward Problem," *Proc. IEEE EMBS Conf. 2005*, pp. 1540-1543.

[5] L.A. Neilson, M. Kovalyov, Z.J. Koles, "A computationally efficient method for accurately solving the EEG forward problem in a finely discretized head model," *Clin. Neurophysiol.*, vol. 116, pp. 2302–2314, 2005.

[6] B. Vanrumste, G. Van Hoey, R. Van de Walle, M.R. D'Have´, I.A. Lemahieu, and P.A. Boon, "The validation of the finite difference method and reciprocity for solving the inverse problem in EEG dipole source analysis," *Brain Topogr.*, vol. 14, pp. 83-92, 2001.

[7] S. Balay, *PETSc users manual*, Technical Report ANL-95/11 Revision 2.1.5, Argonne National Laboratory, 2002.

[8] S. Mininel, "*True Electrical Brain Activity Mapping: new models and computational methods*", PhD Dissertation, University of Trieste, Italy, 2006.

[9] G. Amdahl, "Validity of the single processor approach to achieving large-scale computing capabilities", *AFIPS Conf. Proc.*, vol. 30, pp. 483-485, 1967.