A 3D Vector-Additive Iterative Solver for the Anisotropic Inhomogeneous Poisson Equation in the Forward EEG problem

Vasily Volkov[1], Aleksei Zherdetsky[1], Sergei Turovets[2], and Allen Malony[2]

[1] Department of Mathematics and Mechanics, Belarusian State University, 4 Independence Ave., Minsk 220050, Republic of Belarus
volkovvm@bsu.by
[2] NeuroInformatics Center, 5294 University of Oregon, Eugene, OR 97403, USA
(sergei,malony)@cs.uoregon.edu

Abstract. We describe a novel 3D finite difference method for solving the anisotropic inhomogeneous Poisson equation based on a multi-component additive implicit method with a 13-point stencil. The serial performance is found to be comparable to the most efficient solvers from the family of preconditioned conjugate gradient (PCG) algorithms. The proposed multi-component additive algorithm is unconditionally stable in 3D and amenable for transparent domain decomposition parallelization up to one eighth of the total grid points in the initial computational domain. Some validation and numerical examples are given.

1 Introduction

The challenge in most tomographic techniques is to determine unknown complex coefficients or driving sources in the partial differential equations (PDEs) governing the physics of the particular experimental modality. Problems in neuroscience such as electroencephalography (EEG) and magnetoencephalograpy (MEG) source localization, electrical impedance tomography (EIT) or diffuse optical tomography (DOT) are inherently non-linear, underdetermined and ill-posed, requiring high accuracy in measurements and PDE inverse modeling [1]. The first step in solving such inverse problems is to find a numerical method to solve the direct (forward) problem. When the physical model is three-dimensional and geometrically complex, like the human brain, the high- resolution forward solution can be difficult to construct and compute.

Until recently, most practical research in this field has opted for simplistic analytical or semi-analytical models of a human head in the forward calculations [2]. With geometric information becoming more readily available from MRI or CT scans, finite element (FE) and finite difference (FD) approaches can now incorporate realistic 3D head geometry for human head model construction. However, most of the published models, with a few exceptions, treat the human head tissues as isotropic, while it is well known that brain white matter, skull and facial/scalp muscles are highly anisotropic, with the anisotropic ratio estimated to be between 1:3 and 1:10 [3 and references therein].

G. Allen et al. (Eds.): ICCS 2009, Part I, LNCS 5544, pp. 511–520, 2009.

In the present study we propose an algorithm for solving the anisotropic diffusion equation based on multi-component (vector-additive) implicit FD methods. Not only are these methods unconditionally stable in 3D, but they offer the potential for high domain decomposition parallelization, and are promising candidates for computational acceleration with GPGPUs (general purpose graphics processing units) [4]. We introduce the algorithm and assess the serial performance of the proposed method in comparison with the most efficient solvers from the family of the preconditioned conjugate gradient (PCG) algorithms.

2 Statement of the Problem

The relevant frequency spectrum in EEG, MEG and EIT of the human head is typically below 1 kHz, and most studies deal with frequencies between 0.1 and 100 Hz. Therefore, the physics of EEG/MEG can be well described by the quasi-static approximation of the Maxwell equations, the Poisson equation. The electrical forward problem can be stated as follows: given the positions, orientations and magnitudes of dipole current sources, $\varphi(x, y, z)$, as well as geometry and electrical conductivity of the head volume (Ω), calculate the distribution of the electrical potential on the surface of the head (scalp) (Γ_Ω). Mathematically, it means solving the inhomogeneous anisotropic Poisson equation [2]:

$$\nabla \bullet (\sigma(\nabla u) = \varphi(x, y, z), \text{ in } \Omega \tag{1}$$

with no-flux Neumann boundary conditions on the scalp:

$$\sigma(\nabla u) \bullet n = 0, \text{ on } \Gamma_\Omega. \tag{2}$$

Here $\sigma = \sigma_{ij}(x,y,z)$ is an inhomogeneous symmetric tensor of the head tissues conductivity. Having computed potentials $u(x,y,z)$ and current densities $J=- \sigma(\nabla u)$, the magnetic field B can be found through the Biot-Savart law. The similar non-stationary anisotropic diffusion equation is relevant also in the DOT forward problem modeling [1] and the white matter tractography studies using diffusion tensor MRI imaging [5].

Previously, we built an iterative finite difference forward problem solver for an isotropic version of (1) and (2) based on the multi-component alternating directions implicit (ADI) algorithm [6]. It is a generalization of the classic ADI algorithm, but with improved stability in 3D (the multi-component FD ADI scheme is unconditionally stable in 3D for any value of the time step [7,8]). To describe the electrical conductivity in the heterogeneous biological media within arbitrary geometry, the method of the embedded boundaries has been used. Here an object of interest is embedded into a cubic computational domain with extremely low conductivity values in the external complimentary regions modeling the surrounding air. This effectively guarantees there are no current flows out of the physical area (the Neumann boundary conditions, (2), is naturally satisfied). The idea of the iterative implicit method is to find the solution of (1) and (2) as a steady state of the appropriate evolution (diffusion) problem. At every iteration step, the spatial operator is split into the sum of three 1D operators, which are evaluated alternatively at each sub-step. Such a scheme is accurate to $O[\tau + (\Delta x)^2) + (\Delta y)^2 + (\Delta z)^2]$. In contrast with the classic ADI method, the multi-component ADI uses the regularization (averaging) for evaluation of the variable at the previous instant of time.

Parallelization of the vector-additive ADI algorithm in a shared memory multiprocessor environment (OpenMP) is straightforward, as it consists of nests of independent loops over "bars" of voxels for solving the effective 1D problem in every iteration. However, it is less suitable for implementation in an environment with a distributed memory. In the next section we present a vector-additive algorithm of the domain decomposition type which is potentially amenable for implementation at greater parallel degree.

3 Numerical Scheme

In the Cartesian coordinate system, (1) is expressed as

$$\frac{\partial}{\partial x}\left(\sigma_{xx}\frac{\partial u}{\partial x} + \sigma_{xy}\frac{\partial u}{\partial y} + \sigma_{xz}\frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial y}\left(\sigma_{yy}\frac{\partial u}{\partial y} + \sigma_{yx}\frac{\partial u}{\partial x} + \sigma_{yz}\frac{\partial u}{\partial z} \right) +$$
$$+\frac{\partial}{\partial z}\left(\sigma_{zz}\frac{\partial u}{\partial z} + \sigma_{zx}\frac{\partial u}{\partial x} + \sigma_{zy}\frac{\partial u}{\partial y} \right) = \varphi(x,y,z).$$

To discretize this equation we will use finite difference approximation of the spatial derivatives on the reactangular grid (x_i, y_j, z_k), $i = \overline{1, N_x}$, $j = \overline{1, N_y}$, $k = \overline{1, N_z}$, where N_x, N_y, N_z are the numbers of grid points in *x,y, z* spatial directions. The finite difference approximation of the second order accuracy for the Poisson equation with mixed derivatives can be made with a minimal stencil of 7 points in 2D [9]. Generalization to 3D leads to a 13-point stencil, as shown in Fig. 1. It consists of two diagonal compartments (cells) with one common corner. The whole problem computational domain is represented by a 3D checkerboard lending itself for domain decomposition (partitioning). One can take into account only even (or only odd) mesh cells, each of them having eight neighboring computational cells. Every internal node of this checkerboard grid belongs simultaneously to two neighboring cells. Therefore, it is natural to introduce two components of an approximate numerical solution, (u_m, $\bar{u}_{9-m}$), where $m = \overline{1,8}$ (see Fig. 1). The first component of such pair, u_m, is considered as an internal component of the given mesh cell while the second one is a complimentary component belonging to the corresponding neighboring mesh cell. In these notations, the finite difference approximation, *L*, of the differential operator in (1) in an arbitrary node of the grid, (x_i, y_j, z_k), can be represented as

$$Lu = A_m u + \bar{A}_m \bar{u} \ , \tag{3}$$

where $u = (u_1, u_2, \ldots, u_8)^T$ and $\overline{u} = (\overline{u_8}, \overline{u_7}, \ldots, \overline{u_1})^T$ are the vectors of two components of the approximate numerical solution in two neighboring cells on the grid with a common node at (x_i, y_j, z_k).

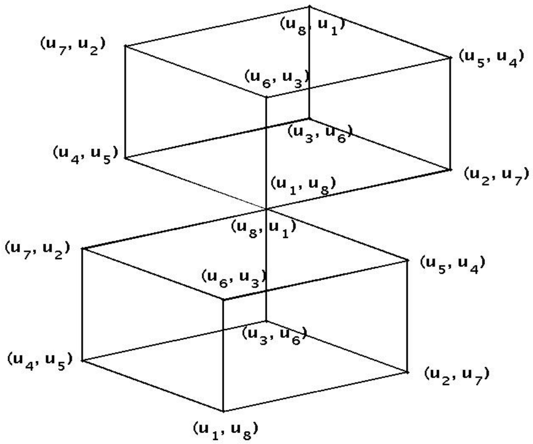

Fig. 1. Schematic view of the finite difference stencil for (1)

In (3) factors A_m and $\overline{A}_m$ are vectors with components given by coefficients of the finite difference approximation for (1), which is obtained by the standard finite volume method [10]. As a result, the derivatives in (1) are given by the following finite differences:

$$
\begin{aligned}
&\frac{\partial}{\partial x}\left(\sigma_{xx}\frac{\partial u}{\partial x}\right) \cong \sigma_{xx}^{12}\frac{u_2 - u_1}{h_x^2} - \sigma_{xx}^{78}\frac{\overline{u_8} - \overline{u_7}}{h_x^2}, \frac{\partial}{\partial y}\left(\sigma_{yy}\frac{\partial u}{\partial y}\right) \cong \sigma_{yy}^{14}\frac{u_4 - u_1}{h_y^2} - \sigma_{yy}^{58}\frac{\overline{u_8} - \overline{u_5}}{h_y^2},\\
&\frac{\partial}{\partial z}\left(\sigma_{zz}\frac{\partial u}{\partial z}\right) \cong \sigma_{zz}^{16}\frac{u_6 - u_1}{h_z^2} - \sigma_{zz}^{38}\frac{\overline{u_8} - \overline{u_3}}{h_z^2},\\
&\frac{\partial}{\partial x}\left(\sigma_{xy}\frac{\partial u}{\partial y}\right) \cong \sigma_{xy}^{34}\frac{u_3 - u_4}{2h_x h_y} - \sigma_{xy}^{12}\frac{u_2 - u_1}{2h_x h_y} + \sigma_{xy}^{78}\frac{\overline{u_8} - \overline{u_7}}{2h_x h_y} - \sigma_{xy}^{56}\frac{\overline{u_5} - \overline{u_6}}{2h_x h_y},\\
&\frac{\partial}{\partial y}\left(\sigma_{yx}\frac{\partial u}{\partial x}\right) \cong \sigma_{yx}^{23}\frac{u_3 - u_2}{2h_x h_y} - \sigma_{yx}^{14}\frac{u_4 - u_1}{2h_x h_y} + \sigma_{yx}^{58}\frac{\overline{u_8} - \overline{u_5}}{2h_x h_y} - \sigma_{yx}^{76}\frac{\overline{u_7} - \overline{u_6}}{2h_x h_y},\\
&\frac{\partial}{\partial x}\left(\sigma_{xz}\frac{\partial u}{\partial z}\right) \cong \sigma_{xz}^{25}\frac{u_3 - u_4}{2h_x h_z} - \sigma_{xz}^{16}\frac{u_6 - u_1}{2h_x h_z} + \sigma_{xz}^{38}\frac{\overline{u_8} - \overline{u_3}}{2h_x h_z} - \sigma_{xz}^{47}\frac{\overline{u_7} - \overline{u_4}}{2h_x h_z},\\
&\frac{\partial}{\partial z}\left(\sigma_{zx}\frac{\partial u}{\partial x}\right) \cong \sigma_{zx}^{56}\frac{u_5 - u_6}{2h_x h_z} - \sigma_{zx}^{12}\frac{u_2 - u_1}{2h_x h_z} + \sigma_{zx}^{78}\frac{\overline{u_8} - \overline{u_7}}{2h_x h_z} - \sigma_{xz}^{34}\frac{\overline{u_3} - \overline{u_4}}{2h_x h_z},
\end{aligned}
\tag{4}
$$

$$\frac{\partial}{\partial y}\left(\sigma_{yz}\frac{\partial u}{\partial z}\right) \cong \sigma_{yz}^{47}\frac{u_7 - u_4}{2h_y h_z} - \sigma_{yz}^{16}\frac{u_6 - u_1}{2h_y h_z} + \sigma_{yz}^{38}\frac{\overline{u}_8 - \overline{u}_3}{2h_y h_z} - \sigma_{yz}^{25}\frac{\overline{u}_5 - \overline{u}_2}{2h_y h_z},$$

$$\frac{\partial}{\partial z}\left(\sigma_{zy}\frac{\partial u}{\partial y}\right) \cong \sigma_{zy}^{67}\frac{u_7 - u_6}{2h_y h_z} - \sigma_{zy}^{14}\frac{u_4 - u_1}{2h_y h_z} + \sigma_{zy}^{58}\frac{\overline{u}_8 - \overline{u}_5}{2h_y h_z} - \sigma_{zy}^{23}\frac{\overline{u}_3 - \overline{u}_2}{2h_y h_z}.$$

Here $\sigma^{mk} = 2\sigma^m\sigma^k /(\sigma^m + \sigma^k)$, where σ^k, σ^m are values of the conductivity tensor components in nodes k and m, and h_x, h_y, h_z are grid steps along the Cartesian axis. As it is seen from (4), variables u_1 and $\overline{u}_8$, which correspond to the most distant nodes in the two cell arrangement in Fig. 1, are absent. This means these nodes are not involved into the stencil. By grouping the terms belonging to one of two cells in the stencil in expressions for finite difference derivatives in (4) one can obtain an additive representation of operator L in (3), which allows us to express the components of vectors A_m and $\overline{A}_m$. For instance, for A_1 and $\overline{A}_8$ we have:

$$A_1 u = -\left(\frac{\sigma_{xx}^{12}}{h_x^2} + \frac{\sigma_{yy}^{14}}{h_y^2} + \frac{\sigma_{zz}^{16}}{h_z^2} - \frac{\sigma_{xy}^{14} + \sigma_{yx}^{12}}{2h_x h_y} - \frac{\sigma_{xz}^{16} + \sigma_{zx}^{12}}{2h_x h_z} - \frac{\sigma_{yz}^{16} + \sigma_{zy}^{14}}{2h_y h_z}\right)u_1 +$$
$$+\left(\frac{\sigma_{xx}^{12}}{h_x^2} - \frac{\sigma_{xy}^{32} + \sigma_{yx}^{12}}{2h_x h_y} - \frac{\sigma_{xz}^{25} + \sigma_{zx}^{12}}{2h_x h_z}\right)u_2 + \frac{\sigma_{xy}^{32} + \sigma_{yx}^{34}}{2h_x h_y}u_3 + \left(\frac{\sigma_{yy}^{14}}{h_y^2} - \frac{\sigma_{xy}^{14} + \sigma_{yx}^{34}}{2h_x h_y} - \frac{\sigma_{yz}^{47} + \sigma_{zy}^{14}}{2h_y h_z}\right)u_4 +$$
$$+\frac{\sigma_{xz}^{25} + \sigma_{zx}^{56}}{2h_x h_z}u_5 - \frac{\sigma_{xz}^{16} + \sigma_{zx}^{56}}{2h_x h_z}u_6 + \frac{\sigma_{yz}^{47} + \sigma_{zy}^{67}}{2h_y h_z}u_7; A_8 u = \frac{\sigma_{yz}^{25} + \sigma_{zy}^{23}}{2h_y h_z}u_2 + \left(\frac{\sigma_{zz}^{38}}{h_z^2} - \frac{\sigma_{xz}^{38} + \sigma_{zx}^{34}}{2h_x h_z}\right)u_3$$
$$+\frac{\sigma_{xz}^{47} + \sigma_{zx}^{34}}{2h_x h_z}u_4 + \left(\frac{\sigma_{yy}^{58}}{h_y^2} - \frac{\sigma_{xy}^{58} + \sigma_{yx}^{65}}{2h_x h_y}\right)u_5 + +\frac{\sigma_{xy}^{67} + \sigma_{yx}^{65}}{2h_x h_y}u_6 + \left(\frac{\sigma_{xx}^{78}}{h_x^2} - \frac{\sigma_{xy}^{67} + \sigma_{yx}^{87}}{2h_x h_y} - \frac{\sigma_{xz}^{47} + \sigma_{zx}^{78}}{2h_x h_z}\right)u_7 -$$
$$-\left(\frac{\sigma_{xx}^{78}}{h_x^2} + \frac{\sigma_{yy}^{58}}{h_y^2} + \frac{\sigma_{zz}^{38}}{h_z^2} - \frac{\sigma_{xy}^{58} + \sigma_{yx}^{87}}{2h_x h_y} - \frac{\sigma_{xz}^{38} + \sigma_{zx}^{78}}{2h_x h_z} - \frac{\sigma_{yz}^{38} + \sigma_{zy}^{58}}{2h_y h_z}\right)u_8.$$

The similar expressions are obtained for three remaining pairs of operators A_2 and A_7, A_3 and A_6, A_4 and A_5. In the boundary voxels of the computational domain the finite difference approximation is constructed taking into account the boundary conditions.

In the particular case of identity between two complimentary components $u_m \equiv \overline{u}_{m'}$, the numerical scheme presented above is equivalent to a system of finite difference equations with a 13 diagonal matrix and dimension $N = N_x \times N_y \times N_z$, where N is a total number of nodes in the grid. The high dimensionality of a finite difference model is a major obstacle in the computational complexity of this numerical problem. The introduction of additional (complimentary) solution components opens an opportunity for use of the vector-additive iterative methods [7-9], which are unconditionally stable and potentially amenable for multi-threading limited only by a total number of nodes in a grid.

An application of the vector-additive iterative scheme of the domain decomposition type to our problem leads to an algorithm with the following key features. Iterative approximations for the internal components in every cell of the grid are computed implicitly as solutions of the system of eight linear algebraic equations in respect of these unknown internal components. External components (belonging to eight neighboring cells) in such an implicit solution are taken from the previous iteration step. As a result, an elementary per-voxel step of the iterative process consists of solving a system of linear algebraic equations of the following type:

$$\frac{\overset{k+1}{u}_m - \tilde{u}}{\tau} = \lambda A_m(\overset{k+1}{u} - \overset{k}{u}) + A_m \overset{k}{u} + \overline{A}_m \overset{k}{\overline{u}} + \varphi, \quad m = \overline{1,8}, \quad \tilde{u} = (u_m + \overline{u}_{9-m})/2 . \tag{5}$$

Here, iteration parameters $\tau > 0$ and $\lambda \geq 1$, where k is an iteration number. Apparently, the calculation of the next iterative approximation requires solving a system of 8 equations of type (5). Thus, the computational complexity per iteration is $Q=NQ_0/8$, where Q_0 is the computational cost for solving the linear system in (5) with a matrix 8×8 , and $N/8$ is a number of computational cells in the checkerboard discretization. Assuming the Gaussian elimination algorithm for solving (5), we have approximately $Q_0 \sim (2/3)8^3 \approx 341$ floating operations per–cell at one iteration. Thus, the computational complexity per iteration is comparable with the standard PCG algorithms. The most important point is that an iterative solution in every computational cell can be updated concurrently as it is dependent from the neighboring cells input only from the previous iteration. Therefore, the structure of this algorithm allows natural partitioning up to N/8 parallel threads of execution.

Theoretical estimates of convergence for this class of the vector-additive numerical schemes and optimal choice of iteration parameters have been developed by Abrashin et. al. [7,8]. An example of using the similar iterative scheme in a 2D case for the convection-diffusion equation was given in one of our work [9].

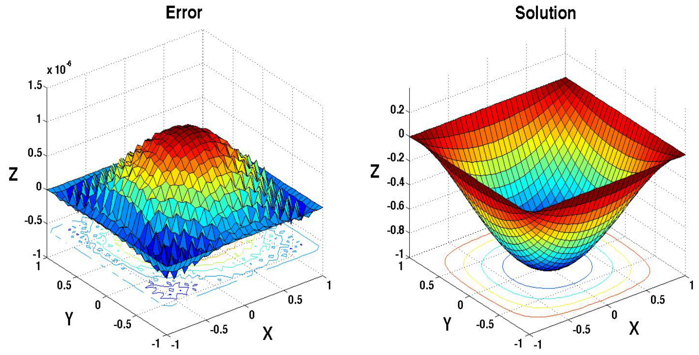

Fig. 2. Local error (left) and numerical solution (right) for a test analytical case (see text for details)

4 Validation and Numerical Examples

A serial version of the proposed forward anisotropic solver was prototyped in Matlab. It was validated against an analytical solution and tested on a cubic phantom with anisotropic inclusions.

A simple analytic test was constructed assuming that in a cubic computational domain with edge length 2a the solution has the form:

$$u(x,y,z) = (x-a)(x+a)(y-a)(y+a)(z-a)(z+a).$$

Apparently, such a solution satisfies the Dirichlet boundary conditions at the computational domain boundaries. The right-hand term, $\varphi(x, y, z)$, has been found by direct analytical differentiation of $u(x, y, z)$ according to (1) and a set of analytical conductivity tensor components. In Fig. 2 one can see the good agreement between the analytical and numerical solutions. The error between analytical and numerical solutions was computed in terms of the local norm. The algorithm converged at 54 iterations with accuracy 1.e-6 for the problem size 32x32x32 voxels. In addition, we

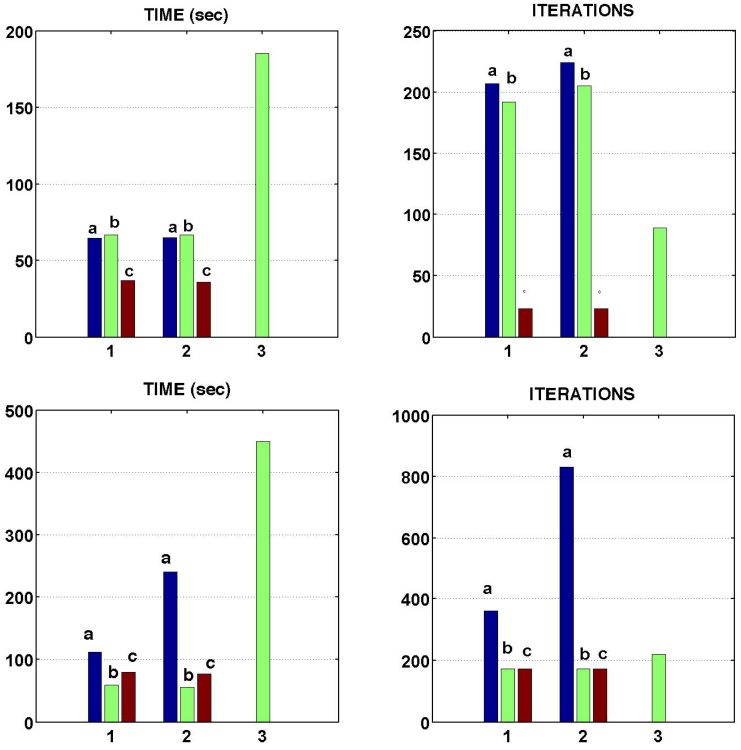

Fig. 3. Histograms of computational time (left) and number of iterations (right) to convergence for QMR (1), BiCG (2) and vector-additive method (3). Preconditioning: without (a), Jacobi (b) and IChF (c). Coefficients and accuracy: smooth, 1.e-6 (top) and heterogeneous, 1.e-4 (bottom).

compared performance of the vector-additive algorithm against the Quasi-Minimal Residual method (QMR) and BiConjugate Gradients method (BiCG) constructed with the same 13-point stencil, as in Fig. 1, and with different preconditioners (Jacobi and incomplete Cholesky factorization (IChF)) for smooth and highly heterogeneous anisotropic phantoms. The code for QMR and BiCG was prototyped in Matlab using the classic schemes [11-13].

As seen in Fig. 3, the QMR and BiCG algorithms perform about 4-5 times better than the vector-additive algorithm in terms of computational time for the heterogeneous and smooth problems of size 64x64x64. This is not surprising, as the serial vector-additive algorithm in the present Matlab implementation is not optimized in terms of matrix operations, while the QMR and BiCG implementations are completely vectorized by using the standard Matlab functions. Yet, the convergence of the vector-additive algorithm was found to be comparable (Fig. 3, right) in terms of a number of iterations needed to reach the prescribed accuracy.

For simulation of the more realistic case of the human head geometry we have employed a cubic phantom with the 20 centimeters edge. The phantom has several shells representing air, scalp, skull, Cerebro-Spinal Fluid (CSF) and different anisotropic inclusions modeling brain. The isotropic conductivity values of scalp (0.45 S/m), skull (0.018 S/m), CSF (1.9 S/m) have been chosen to be equal to the median values reported in the published literature [6]. The air conductivity has been set to 0.001 S/m. The anisotropic ratio of conductivity in the brain inclusion has been set to 1:10 in the orthotropic directions. The results of the current streamline calculations generated by a source and a sink placed in different anisotropic parts of brain and convergence of the vector-additive method and the BiCG method versus the number of discretization points along one direction are shown in Fig.4. Again, in terms of the number of iterations, K_ε, the vector-additive algorithm performance is comparable with the BiCG method. One can see that both methods are converging at the rate of about 300 iterations for the problem size 100x100x100. It is worth noting, that the Jacobi preconditioner performed much better in the case of heterogeneous anisotropic inclusions (comparable with performance of the IChF preconditioner), while in the case of the homogeneous anisotropic cube (Fig. 3, the top-right corner)

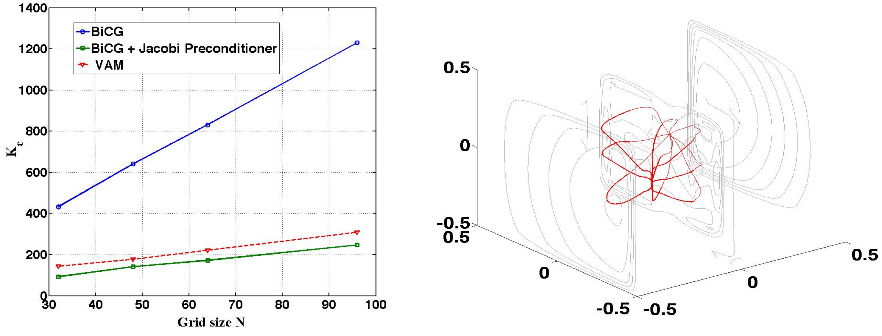

Fig. 4. Anisotropic phantom simulation. Left: convergence of the BiCG(with and without the Jacobi preconditioner), and vector-additive method (the red line) versus the grid size. Right: the current streamlines inside the brain phantom.

the BiCG method with the IChF preconditioner converged essentially faster. The current streamlines shown in Fig.4, right by the thick red color lines behave as expected in accordance with the anisotropic ratio model chosen for the brain inclusions: preferably vertically for the source, horizontally for the sink and equidistant in the surrounding isotropic CSF.

5 Conclusion

We have described a novel 3D finite volume algorithm for solving the anisotropic heterogeneous Poisson equation based on the vector-additive implicit methods with a 13-point stencil. The proposed multi-component additive algorithm is unconditionally stable in 3D and amenable for domain decomposition parallelization with a high number of threads, limited only by the number of grid points in the initial computational domain. We have introduced two major modifications to the classic multi-component vector-additive method suggested in [7-9]. First, we have reduced the number of components from four to two in 3D by using the checkerboard discretization which relaxes the requirements for the operational memory. In the original version of this method [7,8] the minimal number of components was estimated to be $2^{(D-1)}$, where D is the dimension of a computational problem. Secondly, we have introduced variable iterative parameters to improve the convergence rate in the case of essentially heterogeneous coefficients. Finally, to the best of our knowledge, this is the first attempt to use the multi-component numerical scheme for solving 3D anisotropic problems.

The estimated computational complexity per iteration and the method serial performance are found to be comparable to the most efficient iterative solvers from the family of the preconditioned conjugate gradient (PCG) algorithms, in particular the BiCG method with the Jacobi and IChF preconditioners. In the present Matlab implementation the serial version takes more time per iteration and to converge than the standard methods due to the specifics of Matlab, where the PCG algorithms are completely vectorized, while our method can not avoid some necessary cycles. We expect the serial performance to be significantly better in the case of C/C++ implementation. We believe the 3D vector additive method has better parallelism potential than PCG methods due to its cell-level data decomposition. We expect to see performance improvements that overcome the sequential deficiencies as the resolution of the head model scales. Our next step will be a parallel implementation of this algorithm on a computational cluster and a GPGPU accelerator for large size problems based on the high-resolution (256x256x256 voxels) human MRI/CT data.

References

1. Arridge, S.R.: Optical Tomography in Medical Imaging. Inverse Problems 15, R41–R93 (1999)
2. Gulrajani, R.M.: Bioelectricity and Biomagnetism. John Wiley & Sons, New York (1998)
3. Hallez, H., et al.: Review on Solving the Forward Problem in EEG Source Analysis. Journal of Neuroengineering and Rehabilitation 4, 46 (2007)
4. General-Purpose Computation Using Graphics Hardware, http://www.gpgpu.org

5. Qin, C., Kang, N., Cao, N.: Performance evaluation of anisotropic diffusion simulation based tractography on phantom images. In: 45th Annual Southeast Regional Conference ACMSE 2007, pp. 521–522. ACM, New York (2007)
6. Salman, A., Turovets, S., Malony, A., Volkov, V.: Multi-Cluster, Mix-Mode Computational Modeling of Human Head Conductivity. In: Mueller, M.S., et al. (eds.) IWOMP 2005 and IWOMP 2006. LNCS, vol. 4315, pp. 119–130. Springer, Heidelberg (2008)
7. Abrashin, V.N., Egorov, A.A., Zhadaeva, N.G.: On the Convergence Rate of Additive Iterative Methods. Differential Equations 37, 867–879 (2001)
8. Abrashin, V.N., Egorov, A.A., Zhadaeva, N.G.: On a Class of Additive Iterative Methods. Differential Equations 37, 1751–1760 (2001)
9. Volkov, V.M., Lechtikov, S.N.: Multicomponent Iterative Methods of Decomposition Type for Two-Dimensional Stationary Problems of Dissipative Transfer. Differential Equations 33, 927–933 (1997)
10. LeVeque, R.: Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge (2002)
11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: The Numerical Recipes in C: The Art of Scientific Computing, 2nd edn. Cambridge University Press, New York (1992)
12. Barrett, R., Berry, M., Chan, T.F., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, Philadelphia (1994)
13. Freund, R., Nachtigal, N.: QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems. Numer. Math. 60, 315–339 (1991)