Fast Multipole Acceleration of the MEG/EEG Boundary Element Method.

Jan Kybic*, Maureen Clerc, Olivier Faugeras, Renaud Keriven, Théo Papadopoulo

Odyssée Laboratory – ENPC/ENS/INRIA. Address: INRIA, 2004 Route des Lucioles, BP93, 06902 Sophia-Antipolis, France. email: Maureen.Clerc@cermics.enpc.fr, phone: +33-492 38 77 35. fax: +33-492 38 78 45.

*Center for Applied Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. email: kybic@fel.cvut.cz, tel: +420 2 2435 7264.

E-mail: kybic@fel.cvut.cz, Maureen.Clerc@sophia.inria.fr

Abstract. The accurate solution of the forward electrostatic problem is an essential first step before solving the inverse problem of magneto- and electro-encephalography (MEG/EEG). The symmetric Galerkin boundary element method is accurate but cannot be used for very large problems because of its computational complexity and memory requirements. We describe a fast multipole-based acceleration for the symmetric boundary element method (BEM). It creates a hierarchical structure of the elements and approximates far interactions using spherical harmonics expansions. The accelerated method is shown to be as accurate as the direct method, yet for large problems it is both faster and more economical in terms of memory consumption.

PACS numbers: 41.20.Cv

Submitted to: *Physics in Medicine and Biology*

1. Introduction

The Boundary Element Method (BEM) is widely used for solving the forward and inverse problems of Magneto-/Electroencephalography (MEG/EEG) on realistic geometries [1, 2]. It unfortunately leads to huge and dense linear systems which can be hard to handle.

Using fine models is essential for accurately modeling the electromagnetic behavior of the head. The geometry of the brain, especially the cortex containing the sources, is so complex and convoluted, that very small elements (of the order of 1 mm) are needed to model it accurately. Fine models are also needed to accurately represent the fine details of the spatially varying fields. Finally, most BEM have a precision that severely drops when the sources are close to an interface [3–6]. Therefore, as the brain sources are supposed to lie in a very thin layer of the cortex (several mm at most) and thus very close to the surface of the brain, the layers involved must be discretized finely.

The Fast Multipole Method (FMM) [7] acceleration significantly decreases the asymptotic time and memory complexity of solving the forward MEG/EEG problem. We described earlier a preliminary single-level FMM for the double-layer approach [6] that we extend here to multi-level FMM and adapt for the symmetric BEM [8]. To the best of our knowledge, this article describes the only implementation capable of accurately solving the MEG/EEG forward problem for realistic head models described by meshes with over 30,000 points (70,000 unknowns) on a single personal computer.

There is an extensive literature dealing with FMM [7, 9–12] for gravitational or electromagnetic scattering calculations. Some authors have considered the electrostatic Maxwell problem and the symmetric BEM approach, but have treated problems with only one interface [13].

2. Fast Multipole Method Expansions

The Fast Multipole Method (FMM) [6, 7, 9, 11] is a hierarchical approximation algorithm which significantly reduces the time and memory complexity required for the resolution of the linear system of equations $\mathsf{A}\mathbf{u} = \mathbf{c}$, produced by the BEM [8]. It takes advantage of the fact that interaction between surface elements decreases quickly with distance. We use an iterative method (MINRES) that accesses the matrix A only through matrix-vector multiplications $\mathsf{A}\mathbf{u}$. With the FMM, the matrix does not need to be formed explicitly and the overall complexity of calculating the product $\mathsf{A}\mathbf{u}$ representing the pairwise interactions between N elements is decreased from $O(N^2)$ to $O(N)$.

We first briefly recall the operators appearing in the symmetric BEM formulation (Section 2.1) and introduce the spherical harmonic expansion. Then the general FMM framework is presented (Section 3) and applied. We refer the reader to our report [14] for additional details about the symmetric BEM method and our FMM algorithm, that had to be omitted here for space reasons. We compare our approach to the precorrected-FFT acceleration [15] is compared to our approach in Section 4.1. It is also possible to hierarchically simplify the system matrix once it has been computed [16].

2.1. Symmetric Boundary Element Method Operators

To accelerate the symmetric BEM, we shall need to quickly evaluate the matrix vector products ($\mathsf{N}\mathbf{x}$, $\mathsf{D}^*\mathbf{y}$, $\mathsf{D}\mathbf{x}$, $\mathsf{S}\mathbf{y}$) that appear in the symmetric BEM system [8, 14, 17]

$$\underbrace{\begin{pmatrix} \mathsf{N} & \mathsf{D}^* \\ \mathsf{D} & \mathsf{S} \end{pmatrix}}_{\mathsf{A}} \underbrace{\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}}_{\mathbf{u}} = \underbrace{\begin{pmatrix} \mathbf{w} \\ \mathbf{z} \end{pmatrix}}_{\mathbf{c}} . \tag{1}$$

where the unknowns $\mathbf{x}$ and $\mathbf{y}$ represent a discretized version of the electric potential V and flow $p = \sigma\partial_{\mathbf{n}}V$, respectively, and the right hand side terms $\mathbf{w}$ and $\mathbf{z}$ represent the known free-space field (potential and flow) corresponding to the sources. The matrices $\mathsf{N}, \mathsf{D}, \mathsf{D}^*$ and S are the discretized versions of the following continuous operators (all mapping a scalar function f on an interface $\partial\Omega$ to another scalar function). For $\mathbf{r}$ on $\partial\Omega$,

$$\big(\mathcal{N}f\big)(\mathbf{r}) = \int_{\partial\Omega} \partial^2_{\mathbf{n},\mathbf{n}'} G(\mathbf{r}-\mathbf{r}') f(\mathbf{r}')\,\mathrm{d}s(\mathbf{r}') \qquad \big(\mathcal{D}f\big)(\mathbf{r}) = \int_{\partial\Omega} \partial_{\mathbf{n}'} G(\mathbf{r}-\mathbf{r}') f(\mathbf{r}')\,\mathrm{d}s(\mathbf{r}')$$

$$\big(\mathcal{D}^* f\big)(\mathbf{r}) = \int_{\partial\Omega} \partial_{\mathbf{n}} G(\mathbf{r}-\mathbf{r}') f(\mathbf{r}')\,\mathrm{d}s(\mathbf{r}') \qquad \big(\mathcal{S} f\big)(\mathbf{r}) = \int_{\partial\Omega} G(\mathbf{r}-\mathbf{r}') f(\mathbf{r}')\,\mathrm{d}s(\mathbf{r}')$$

with a Green function $G(\mathbf{r}) = 1/\big(4\pi\|\mathbf{r}\|\big)$ and where for example $\partial_{\mathbf{n}}$, $\partial_{\mathbf{n}'}$ stands for a partial derivative with respect to the surface normal at $\mathbf{r}$, $\mathbf{r}'$. We obtain matrices

$$\big(\mathsf{N}\big)_{ik} = \big\langle \mathcal{N}\varphi_k, \varphi_i \big\rangle, \qquad \big(\mathsf{S}\big)_{jl} = \big\langle \mathcal{S}\psi_l, \psi_j \big\rangle \tag{2}$$

$$\big(\mathsf{D}\big)_{jk} = \big(\mathsf{D}^*\big)_{kj} = \big\langle \mathcal{D}\varphi_k, \psi_j \big\rangle = \big\langle \mathcal{D}^*\psi_j, \varphi_k \big\rangle, \tag{3}$$

where ψ resp. φ are P0 (piecewise constant on each triangle) resp. P1 (piecewise linear on each triangle) boundary elements. As an example, the discretized system for a nested three layer model is

$$\underbrace{\begin{pmatrix} (\sigma_1+\sigma_2)\mathsf{N}_{11} & -\sigma_2\mathsf{N}_{12} & 0 & -2\mathsf{D}^*_{11} & \mathsf{D}^*_{12} \\ -\sigma_2\mathsf{N}_{21} & (\sigma_2+\sigma_3)\mathsf{N}_{22} & -\sigma_3\mathsf{N}_{23} & \mathsf{D}^*_{21} & -2\mathsf{D}^*_{22} \\ 0 & -\sigma_3\mathsf{N}_{32} & \sigma_3\mathsf{N}_{33} & 0 & \mathsf{D}^*_{32} \\ -2\mathsf{D}_{11} & \mathsf{D}_{12} & 0 & (\sigma_1^{-1}+\sigma_2^{-1})\mathsf{S}_{11} & -\sigma_2^{-1}\mathsf{S}_{12} \\ \mathsf{D}_{21} & -2\mathsf{D}_{22} & \mathsf{D}_{23} & -\sigma_2^{-1}\mathsf{S}_{21} & (\sigma_1^{-1}+\sigma_2^{-1})\mathsf{S}_{22} \end{pmatrix}}_{\mathsf{A}} \cdot \underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}}_{\mathbf{u}} = \underbrace{\begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}}_{\mathbf{c}}$$

where we denote the surfaces $1, 2, 3$, the operator subscripts denote targets and sources (e.g. N_{23} relates potential on surface 3 with free-space potential on surface 2), and σ_α are the conductivities of enclosed volumes, numbered from inside.

2.2. Spherical harmonics

The Green function $G(\mathbf{r}-\mathbf{r}') \sim 1/\|\mathbf{r}-\mathbf{r}'\|$ can be decomposed as [10]:

$$\underbrace{\|\mathbf{r}'-\mathbf{C}\| > \lambda\|\mathbf{r}-\mathbf{C}\|}_{\text{well-separateness}} \implies \frac{1}{\|\mathbf{r}-\mathbf{r}'\|} = \sum_{n=0}^{L}\sum_{m=-n}^{n} I_n^{-m}(\mathbf{C}-\mathbf{r})\, O_n^m(\mathbf{r}'-\mathbf{C}) + \text{error} \tag{4}$$

where $\mathbf{C}$ is the center of expansion and I_n^m resp. O_n^m are the inner resp. outer spherical harmonics (Appendix 1). To obtain a practical expression, the series is truncated to order L. Acceptable accuracy is guaranteed for $\mathbf{r}$, $\mathbf{r}'$ sufficiently far apart. The approximation error can be bounded by choosing a suitable parameter $\lambda > 1$.

2.2.1. Operator $\mathcal{S}$ To approximate the discretized operator $\mathcal{S}$, we integrate (4) with the P0 basis functions ψ_i. For each element (triangle) i we define the outer-field resp. inner-field expansion coefficients§

$$^{i}\mathsf{a}_n^m(\mathbf{C}) = \int I_n^m(\mathbf{C}-\mathbf{r})\psi_i(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{5}$$

$$^{i}\tilde{\mathsf{a}}_n^m(\mathbf{C}) = \int O_n^m(\mathbf{C}-\mathbf{r})\psi_i(\mathbf{r})\,\mathrm{d}\mathbf{r}\ . \tag{6}$$

The operator $\mathcal{S}$ is then approximated (if elements i and j are well-separated) by

$$4\pi\big\langle \mathcal{S}\psi_i, \psi_j \big\rangle \approx {}^{i}\mathsf{a}(\mathbf{C}) \odot {}^{j}\tilde{\mathsf{a}}(\mathbf{C}) \overset{\text{def}}{=} \sum_{\substack{n=0\ldots L \\ m=-n\ldots n}} (-1)^n\, {}^{i}\mathsf{a}_n^{-m}(\mathbf{C})\, {}^{j}\tilde{\mathsf{a}}_n^{m}(\mathbf{C}) \tag{7}$$

where we have used the symmetry relation $O_n^m(-\mathbf{r}) = (-1)^n O_n^m(\mathbf{r})$.

§ with a deliberate though unfortunate conflict in notation: the outer-field concerns the information propagating out, from elements $\mathbf{r}$ close to a center of expansion $\mathbf{C}$, and the inner-field concerns the information coming into an element $\mathbf{r}'$ from a remote center of expansion $\mathbf{C}$.

2.2.2. Operator $\mathcal{D}$ We apply a gradient with respect to $\mathbf{r}$ to (4) and integrate the result with the basis functions ψ_i (P0) and $\varphi'_{j'}$ (partial P1). The partial P1 function $\varphi'_{j'}$ is identical to some φ_j on one triangle and zero elsewhere. We define coefficients

$$ {}^{i'}\mathtt{b}_n^m(\mathbf{C}) = \int \nabla I_n^m(\mathbf{C}-\mathbf{r}) \cdot \mathbf{n}_{i'} \varphi'_{i'}(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{8} $$

$$ {}^{i'}\widetilde{\mathtt{b}}_n^m(\mathbf{C}) = \int \nabla O_n^m(\mathbf{C}-\mathbf{r}) \cdot \mathbf{n}_{i'} \varphi'_{i'}(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{9} $$

with $\mathtt{b}$,$\widetilde{\mathtt{b}}$ again representable by $(L+1)^2$ dimensional complex vectors. Then we have

$$ \begin{aligned} -4\pi \left\langle \mathcal{D}\varphi'_{i'}, \psi_j \right\rangle = -4\pi \left\langle \varphi'_{i'}, \mathcal{D}^*\psi_j \right\rangle \approx {}^{i'}\mathtt{b} \odot {}^{j}\widetilde{\mathtt{a}} &= \sum_{n,m} (-1)^n\, {}^{i'}\mathtt{b}_n^{-m}(\mathbf{C})\, {}^{j}\widetilde{\mathtt{a}}_n^m(\mathbf{C}) \\ = {}^{j}\mathtt{a} \odot {}^{i'}\widetilde{\mathtt{b}} &= \sum_{n,m} (-1)^n\, {}^{j}\mathtt{a}_n^m(\mathbf{C})\, {}^{i'}\widetilde{\mathtt{b}}_n^{-m}(\mathbf{C}) \end{aligned} \tag{10} $$

Coefficients ${}^{i}\mathtt{b}$ corresponding to a complete P1 basis function φ_i at vertex i are calculated by aggregating the coefficients ${}^{i'}\mathtt{b}$ on all triangles $T_{i'}$ sharing vertex i.

2.2.3. Operator $\mathcal{N}$ To calculate $\left(\mathtt{N}\right)_{i'j'} = \left\langle \mathcal{N}\varphi'_{i'}, \varphi'_{j'} \right\rangle$ we define coefficients

$$ {}^{i'}\mathtt{c}_n^m = (\mathbf{q}_{i'} \times \mathbf{n}_{i'})\, {}^{i}\mathtt{a}_n^m \qquad \text{and} \qquad {}^{j'}\widetilde{\mathtt{c}}_n^m = (\mathbf{q}_{j'} \times \mathbf{n}_{j'})\, {}^{j}\widetilde{\mathtt{a}}_n^m \tag{11} $$

represented by $(L+1)^2$ 3D complex vectors. We obtain

$$ 4\pi \left\langle \mathcal{N}\varphi'_{i'}, \varphi'_{j'} \right\rangle \approx {}^{i'}\mathtt{c}(\mathbf{C}) \odot {}^{j'}\widetilde{\mathtt{c}}(\mathbf{C}) = \sum_{\substack{n=0\ldots L \\ m=-n\ldots n}} (-1)^n\, {}^{i'}\mathtt{c}^{-m,n}(\mathbf{C}) \;\cdot\; {}^{j'}\widetilde{\mathtt{c}}^{m,n}(\mathbf{C})\ . \tag{12} $$

with a complex scalar product '$\cdot$'. The coefficients $\mathtt{c}$ for complete P1 elements are again aggregated from constituting triangles.

2.3. Translating multipolar representations

The formulas for the outer-outer, inner-inner and outer-inner operators R, S, resp. T (13) are generalized from [10] and are identical for all three types of expansion coefficients ($\mathtt{a}, \mathtt{b}, \mathtt{c}$, here represented by $\mathtt{x}$) provided that elementwise complex multiplication and addition are used for coefficients $\mathtt{c}$.

$$ \begin{aligned} \mathtt{x}_{n'}^{-m'}(\mathbf{M}) &= \left(\mathrm{R}_{\mathbf{NM}}\,\mathtt{x}\right)_{n'}^{-m'} = \sum_{n=0\ldots n',m=-n\ldots n} I_{n'-n}^{m-m'}(\mathbf{M}-\mathbf{N})\,\mathtt{x}_n^{-m}(\mathbf{N}) \\ \widetilde{\mathtt{x}}_{n'}^{-m'}(\mathbf{M}) &= \left(\mathrm{S}_{\mathbf{NM}}\,\widetilde{\mathtt{x}}\right)_{n'}^{-m'} = \sum_{n=n'\ldots L,m=-n\ldots n} I_{n-n'}^{m'-m'}(\mathbf{M}-\mathbf{N})\,\widetilde{\mathtt{x}}_n^{-m}(\mathbf{N}) \\ \widetilde{\mathtt{x}}_{n'}^{m'}(\mathbf{M}) &= \left(\mathrm{T}_{\mathbf{NM}}\,\mathtt{x}\right)_{n'}^{m'} = \sum_{n=0\ldots L,m=-n\ldots n} O_{n+n'}^{m+m'}(\mathbf{M}-\mathbf{N})\,\mathtt{x}_n^{-m}(\mathbf{N}) \end{aligned} \tag{13} $$

It is advantageous to precompute the values $I_n^m(\mathbf{M}-\mathbf{N})$ (usable for both R, S) and $O_n^m(\mathbf{M}-\mathbf{N})$.

The formulas above all require $O(L^4)$ operations, which makes them a bottleneck of the FMM, especially the operator T. Several acceleration techniques were proposed, reducing the complexity to $O(L^3)$ or even $O(L^2 \log L)$: using FFT [10], rotation to $z-$axis where translation is simpler [9], or a plane wave representation [7]. However, they are significantly more complex than (13) and thus faster only for high values of L; e.g. for $L \geq 20$ in the case of the FFT approach according to our tests.

3. Fast Multipole Method Algorithms

Given the expansion and translation formulas for operators $\mathcal{S}$, $\mathcal{D}$, $\mathcal{D}^*$, and $\mathcal{N}$, derived above, we can now use them to formulate the FMM. The particularity here is that the FMM needs to be applied to all four operators, so we first formulate it in generic terms.

3.1. FMM basics

We are to calculate the interaction between two groups of elements, $\mathcal{A}$ and $\mathcal{B}$,

$$y_j = \sum_{i \in \mathcal{A}} f_{ij} x_i, \quad \text{for all } j \in \mathcal{B} \ . \tag{14}$$

The values f_{ij} correspond to the elements of the matrices S, D, D^*, and N (2,3). The elements from $\mathcal{A}$ and $\mathcal{B}$ correspond to the support of the basis functions, i.e. to either triangles (P0 elements ψ), or to sets of triangles with a common vertex (P1 elements φ).

For each sufficiently *well-separated* pair of elements i in $\mathcal{A}$, resp. j in $\mathcal{B}$, the term f_{ij} can be approximated with an a priori given precision ε:

$$\underbrace{\Big(d(i,\mathbf{C}) > \lambda d(j,\mathbf{C}) \iff \mathscr{D}_{\mathbf{C}}(i,j)\Big)}_{\text{well-separated}} \stackrel{\text{def}}{\Longrightarrow} \mid f_{ij} - \phi_j(\mathbf{C}) \odot \widetilde{\phi}_i(\mathbf{C}) \mid \leq \varepsilon, \tag{15}$$

where $\phi_j(\mathbf{C})$, and $\widetilde{\phi}_i(\mathbf{C})$ are called an outer (far, or multipole) resp. inner (near, or local) expansion, $\mathbf{C}$ is the center of expansion, and $d(i,\mathbf{C})$ is a distance from element i to $\mathbf{C}$. The well-separateness condition (with $\lambda > 1$) can be identified in (4), the expansion formula corresponds to (7,10,12) and the expansions ϕ, $\widetilde{\phi}$ to coefficients $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, resp. $\widetilde{\mathtt{a}}$, $\widetilde{\mathtt{b}}$, $\widetilde{\mathtt{c}}$,

The expansions are functions $\phi_j : \mathbb{R}^3 \to \mathcal{Q}$, $\widetilde{\phi}_i : \mathbb{R}^3 \to \widetilde{\mathcal{Q}}$, with suitable domains $\mathcal{Q}, \widetilde{\mathcal{Q}}$, in our case 2D tables of complex numbers (coefficients $\mathtt{a}$, $\mathtt{b}$), or complex vectors (coefficients $\mathtt{c}$). The operator $\odot : \mathcal{Q} \times \widetilde{\mathcal{Q}} \to \mathbb{R}$ is a bilinear and not necessarily commutative. We also define addition operators $\oplus : \mathcal{Q} \times \mathcal{Q} \to \mathcal{Q}$ and $\widetilde{\oplus} : \widetilde{\mathcal{Q}} \times \widetilde{\mathcal{Q}} \to \widetilde{\mathcal{Q}}$, distributive with respect to $\odot$, and shortcuts for summation $\sum^{\oplus}$ and $\sum^{\widetilde{\oplus}}$. Finally, there is a multiplication (scaling) operation $\mathbb{R} \times \mathcal{Q} \to \mathcal{Q}$ with the natural semantics.

Our four interpretations of $\oplus$, $\odot$ and scaling should be clear from (7,10,12,13), i.e. they are standard complex vector operators, with an extra structure. The only subtlety is that while the approximation formulas (7,10,12) yield complex numbers in general, in our case the results are real thanks to the properties of spherical harmonics (4).

Since the error ε in (15) can be bounded by choosing a suitable minimum relative distance λ, the error of the FMM algorithms described later can be also bounded (see also Section 3.7).

We shall say that groups $\mathcal{A}_k$ and $\mathcal{B}_l$ are *well-separated*, denoted $\mathscr{D}_{\mathbf{C}}(\mathcal{A}_k, \mathcal{B}_l)$, iff $d(\mathcal{A}_k,\mathbf{C}) > \lambda \varrho_{\mathcal{B}_l}(\mathbf{C})$ with distance $d(\mathcal{A}_k,\mathbf{C}) = \min_{i \in \mathcal{A}_k} d(i,\mathbf{C})$ and radius $\varrho_{\mathcal{B}_l}(\mathbf{C}) = \max_{j \in \mathcal{B}_l} d(j,\mathbf{C})$. It implies that all elements $i \in \mathcal{A}_k$, $j \in \mathcal{B}_l$ are also well-separated, $\mathscr{D}_{\mathbf{C}}(i,j)$.

3.2. Single-level FMM algorithms

The simplest of the FMM algorithms is the 'grouping' or 'middle-man' algorithm. It is based on dividing the elements in $\mathcal{A}$ and $\mathcal{B}$ into spatially constrained cells, typically

by partitioning the space into rectangular cells of identical size [9]. The interaction between far (well-separated) cells $\mathcal{A}_k$, $\mathcal{B}_l$ is then carried out using an approximation

$$\sum_{i\in\mathcal{A}_k} x_i f_{ij} \approx \phi_j(\mathbf{C}_l) \odot \widetilde{\Phi}(\mathcal{A}_k, \mathbf{C}_l) = \phi_j(\mathbf{C}_l) \odot \underbrace{\Big(\sum_{i\in\mathcal{A}_k}^{\widetilde{\oplus}} x_i \widetilde{\phi}_i(\mathbf{C}_l)\Big)}_{\widetilde{\Phi}(\mathcal{A}_k,\mathbf{C}_l)}, \quad (16)$$

derived from (15). This algorithm has complexity of $O(N^{3/2})$, where N is the number of elements $\|\mathcal{A}\| \approx \|\mathcal{B}\|$. We shall use a translation operator T (13) that can convert an outer expansion at point $\mathbf{C}_k$ into an inner one around point $\mathbf{C}_l$

$$\widetilde{\phi}_i(\mathbf{C}_l) = \mathrm{T}_{\mathbf{C}_k\mathbf{C}_l}\ \phi_i(\mathbf{C}_k)\ . \quad (17)$$

In this way, we can improve the middle-man algorithm so that the inner fields $\widetilde{\Phi}(\mathcal{A}_k, \mathbf{C}_l)$ can be calculated more efficiently. Instead of computing $\widetilde{\Phi}(\mathcal{A}_k, \mathbf{C}_l)$ for each $\mathbf{C}_l$, the $\widetilde{\Phi}(\mathcal{A}_k, \mathbf{C}_k)$ have to be calculated only once, and are translated to the centers of all other cells $\mathcal{B}_l$ we want to interact with. The improved algorithm is called a single-level FMM algorithm and has an asymptotic complexity of $O(N^{4/3})$.

3.3. Multi-level FMM algorithm

In order to improve the single-level FMM, we build a hierarchy of cells of different sizes, so that an optimal cell-size can be chosen depending on the interaction distance. This leads to a multi-level FMM algorithm, often called simply FMM [9]. We create trees $\mathcal{A}$ resp. $\mathcal{B}$ from the input set $\mathcal{A}$ resp. output set $\mathcal{B}$. Children of each non-leaf cell (tree node) X are themselves cells contained in X. We shall further use an outer-to-outer translation operator R and an inner-to-inner translation operator S (13):

$$\phi_j(\mathbf{C}_l) = \mathrm{R}_{\mathbf{C}_k\mathbf{C}_l}\ \phi_j(\mathbf{C}_k) \quad (18)$$

$$\widetilde{\phi}_j(\mathbf{C}_l) = \mathrm{S}_{\mathbf{C}_k\mathbf{C}_l}\ \widetilde{\phi}_j(\mathbf{C}_k) \quad (19)$$

The operator R is used to calculate the outer field for each non-leaf cell X in the tree $\mathcal{A}$ by summing the outer fields of all its children Y — an *up-sweep*. Similarly, during the *down-sweep*, operator S translates the inner-field from non-leaf cells in tree $\mathcal{B}$ to their children (Figure 1).

3.3.1. Interaction plan We define a plan $\mathcal{P}$ composed of local and far interactions $\mathcal{P}_L$ and $\mathcal{P}_F$; $\mathcal{P}_L, \mathcal{P}_F \subseteq \mathcal{A} \times \mathcal{B}$. All pairs $(X, Y) \in \mathcal{P}_L$ are to be treated locally, by explicit summation (14). A pair $(X, Y) \in \mathcal{P}_F$ (for X, Y well-separated) indicates that outer field $\Phi(X, \mathbf{C}_X)$, corresponding to all elements in X, must be translated to $\mathbf{C}_Y$ and applied to calculate the contributions of X on all elements in Y.

A plan $\mathcal{P}$ is *well-formed* if each interaction between each pair of leaves of trees $\mathcal{A}$ and $\mathcal{B}$ is handled exactly once. In other terms, for each pair of leaf nodes $U \in \mathcal{A}$ and $V \in \mathcal{B}$, there must be exactly one path going from U up the tree $\mathcal{A}$ to a node X, then to a node $Y \in \mathcal{B}$ such that $(X, Y) \in \mathcal{P}_L \cup \mathcal{P}_F$, and finally down tree $\mathcal{B}$ to V.

3.3.2. Optimal interaction plan. To minimize the number of local interactions $\|\mathcal{P}_L\|$, we use local interactions exclusively on pairs of leaf cells which are not well-separated. We then want to minimize the number of far interactions $\|\mathcal{P}_F\|$. Also, for implementation reasons that will be explained later (Section 3.6) we must limit the

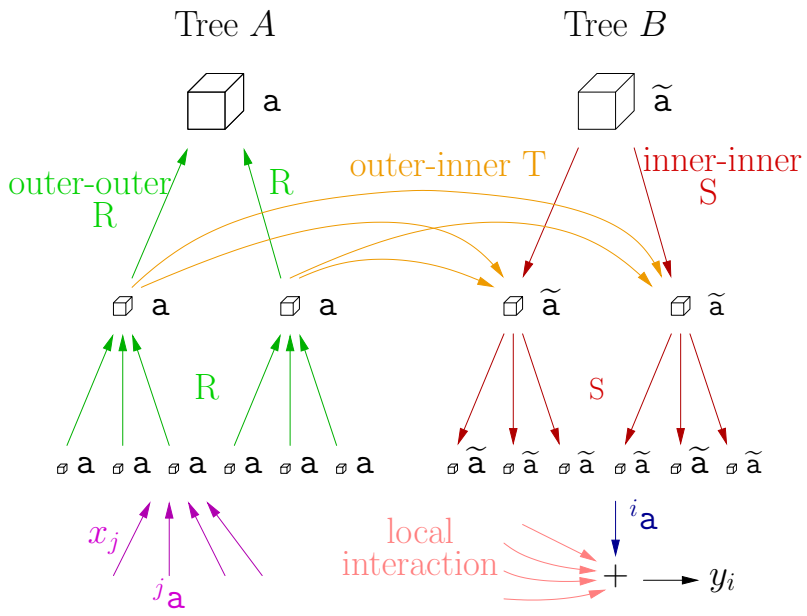

Figure 1. FMM interactions (for operator $\mathcal{S}$). The outer fields are propagated up the tree $\mathcal{A}$ during the up-sweep phase using operator R. They are transfered to tree $\mathcal{B}$ and converted to inner fields using operator T. The inner fields are propagated down tree $\mathcal{B}$ using operator S. At leaf cells, the far interactions calculated from the inner-field coefficients are summed with local interactions.

number of different $\mathrm{T}_{\mathbf{C}_k\mathbf{C}_l}$ operators as measured by the number of different $\mathbf{C}_k - \mathbf{C}_l$ vectors.

A classical approach is to descend simultaneously from the root to the leaves in both trees $\mathcal{A}$, $\mathcal{B}$ (which must be identical), and, at each level, to include all valid interactions that have not been already treated at higher levels, allowing only interactions between cells at the same level [7, 9].

Our planner (Algorithm 1) is more general, not requiring the trees $\mathcal{A}$, $\mathcal{B}$ to be identical and allowing interactions between cells at different levels (different sizes). It is based on the additional constraint that for cells $(X, Y) \in \mathcal{P}_F$, the cell X is never smaller than cell Y. As a consequence, X and Y will be almost the same size for X close to Y (but well-separated), while cells X further away from Y may be bigger if the separability condition allows it. If the trees share the same top-level bounding box, then 'bigger' (line 12 in Algorithm 1) is equivalent to 'at higher-level'.

3.3.3. Executing the interaction plan Once the interaction plan is created, it is executed by Algorithm 2 whenever the Au product is needed, i.e. at each iteration. Note that the local interaction coefficients f_{ij}, the outer expansions ϕ_i, and the operators R, S, T can be precalculated. The remaining cost of executing the plan consists of applying the operators R, S, and T, and the coefficients f_{ij}.

3.3.4. Asymptotic complexity of the described FMM is $O(N)$ under the following hypotheses which are simple and easy to fulfill.

Algorithm 1: Create an interaction plan for a Multi-Level FMM algorithm

Input: Trees $\mathcal{A}$ and $\mathcal{B}$, with cells $\mathcal{A}_k$ and $\mathcal{B}_l$ as leaves.
Output: An interaction plan $\mathcal{P} = \left(\mathcal{P}_L, \mathcal{P}_F\right)$

```
1  P_L ← ∅ ; P_F ← ∅
2  return CreatePlan({ root of A}, root of B)
   // CreatePlan adds to P_F,P_L interactions between a set X (from A) and
      a cell Y (from B)
3  procedure CreatePlan(X, Y):
4  Z ← ∅  // Cells from X to be processed later
5  while X ≠ ∅ do
6  |  Take X from X
7  |  if  d(X, C_Y) > λ ϱ_Y(C_Y)  // are X and Y well-separated?
8  |  then
9  |  |  add (X, Y) into P_F
10 |
12 |  else if (X not a leaf) ∧ ( ϱ_X(C_X) ≥ ϱ_Y(C_Y)  ∨  Y is leaf ) then
13 |  |  put children of X into X
14 |  else add X to Z
15 if Y is leaf then
16 |  add {(Z, Y); Z ∈ Z} to P_L
17
18 else call CreatePlan(Z, Y') for all Y' children of Y.
```

$\mathscr{H}_1$: There are at most K_c elements in any leaf cell.
$\mathscr{H}_2$: The number of leaf cells from $\mathcal{A}$ near to a given leaf cell from $\mathcal{B}$ is at most K_n.
$\mathscr{H}_3$: In any sphere of radius R, there are at most $K_s(R/\varrho)^{K_d}$ cells of radius $\varrho' > \varrho$.
$\mathscr{H}_4$: For any cell with radius ϱ, the radii of all its children are at least ϱ/K_r, $K_r > 1$.

where K_c, K_n, K_s, K_d, K_r are constants, independent of the total number of elements N. Here is a sketch of a proof: local interactions can be evaluated in $O(N)$ time, since each element can interact with at most K_nK_c others. Calculating outer fields and propagating the inner fields needs $O(N)$ operations as well, because there are $O(N)$ elements and nodes. Finally, we show that the number of far interactions added to $\mathcal{P}_F$ for each cell in $\mathcal{B}$ by the Algorithm 1 is at most $K_s(\lambda K_r)^{K_d}$, hence the total number of outer-to-inner translations (operator T), proportional to the number $\|\mathcal{P}_F\|$, is also $O(N)$. □ Unfortunately, due to the large constants involved, the superiority of the FMM over the brute-force approach only appears for large values of N.

3.4. Memory complexity

In order to execute Algorithm 2 efficiently, we need to precalculate and store the local interactions f_{ij} involved in the local plan $\mathcal{P}_L$ and the outer expansions ϕ_i at all elements, corresponding to storing at most $K_cK_nN/2$, resp. $2(4N_v + N_t)(L+1)^2 = 4N(L+1)^2$ real numbers, where N_v and N_t is the total number of vertices, resp. triangles of the mesh. Furthermore, we need to store the precalculated spherical harmonics for operators R,S, T, the biggest one being operator T needing $2(2L+1)^2$ real values per distinct $\mathbf{M} - \mathbf{N}$ vector.

Algorithm 2: Execute an interaction plan for a Multi-Level FMM algorithm

Input: Interaction plan $\mathcal{P} = (\mathcal{P}_L, \mathcal{P}_F)$; trees $\mathcal{A}$, $\mathcal{B}$; vector x_i. Precalculated values f_{ij}, ϕ_i, R, S, T.

Output: A vector y_j so that approximately $y_j = \sum_{i \in \mathcal{A}} x_i f_{ij}$ for all $j \in \mathcal{B}$.

$y_j \leftarrow 0$ for all j

// Treat local interactions

foreach $(X, Y) \in \mathcal{P}_L$ **do**
  foreach element $i \in X$, element $j \in Y$ **do**
    $y_j \leftarrow y_j + x_i f_{ij}$

// Up-sweep. Calculate outer-field recursively

foreach cell X from tree $\mathcal{A}$ **do**

$$\Phi(X, \mathbf{C}_X) = \begin{cases} \overset{\oplus}{\sum\limits_{\text{element } i \in X}} x_i \phi_i(\mathbf{C}_X) & \text{if } X \text{ is a leaf} \\ \overset{\oplus}{\sum\limits_{X' \text{ child of } X}} \mathrm{R}_{\mathbf{C}_{X'}\mathbf{C}_X}\, \Phi(X', \mathbf{C}_{X'}) & \text{otherwise} \end{cases}$$

// Down-sweep.

foreach cell Y from tree $\mathcal{B}$ **do**

$$\widetilde{\Phi}(Y, \mathbf{C}_Y) = \underbrace{\mathrm{S}_{\mathbf{C}_Z\mathbf{C}_Y} \widetilde{\Phi}(Z, \mathbf{C}_Z)}_{Z \text{ parent of } Y} \quad \widetilde{\oplus} \overset{\widetilde{\oplus}}{\sum_{(X,Y) \in \mathcal{P}_F}} \mathrm{T}_{\mathbf{C}_X\mathbf{C}_Y} \Phi(X, \mathbf{C}_X)$$

  if Y is leaf **then**
    foreach element $j \in Y$ **do**
      $y_j \leftarrow y_j + \phi_j(\mathbf{C}_Y) \odot \widetilde{\Phi}(Y, \mathbf{C}_Y)$

3.5. Timing of multipolar representation

The most expensive operations (Table 1) are calculating the spherical harmonic expansions and applying the translation operator T. The critical problem size above which it is advantageous to apply FMM is very big. In a very simple case of two well-separated groups of elements interacting 100 times, we need more than 200 elements in each group — this observation can guide our choice of K_c. For real geometries, not all cells are well separated; we can estimate that FMM only starts to be competitive for problems with more than 10^4 elements.

3.6. Tree structure

A *median tree* is an ideally balanced binary tree, based on splitting node elements using axis-parallel planes into equal halves. However, it requires calculating (and storing) the translation operators for all pairs of interacting cells, which soon becomes prohibitive.

For this reason, we have adopted a classical adaptive *octtree* structure.

Table 1. Timings§ of some operations involved in the FMM for $L = 10$, sorted by the elapsed time. By $\mathtt{a}$ we indicate the scalar expansion coefficients, while $\mathtt{c}$ stands for the vector expansion coefficients needed to approximate operator $\mathcal{N}$.

Operation to calculate	Time [μs]
$\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ for 1 triangle	4218.75
apply T for $\mathtt{c}$	3183.59
apply T for $\mathtt{a}$	1054.69
apply R for $\mathtt{c}$	933.10
apply S for $\mathtt{c}$	788.92
apply R for $\mathtt{a}$	310.06
apply S for $\mathtt{a}$	261.23
O_n^m for T	83.62
$\mathsf{D}_{ij'}$ directly	47.00
...	
...	
S_{ij} directly	43.34
I_n^m for R or S	22.13
$\odot$ for $\mathtt{c}$	12.28
$\odot$ for $\mathtt{a}$	3.66
$y \leftarrow y + x$ for $y \in \mathbb{R}$	< 0.10
$y \leftarrow y + x$ for $\mathtt{a}$	< 0.10
$y \leftarrow y + x$ for $\mathtt{c}$	< 0.10
$\alpha\mathtt{a}$, $\alpha \in \mathbb{R}$	< 0.10

A bounding box of all elements becomes a cell-box of the root element. At each level, a parent cell-box is divided into eight identical subboxes. Each of the eight children then receives elements whose center of gravity falls into its box. The subdivision is stopped for nodes with less then K_c elements. Empty branches are pruned. Expansion centers $\mathbf{C}_X$ are put into geometrical centers of each cell-box. Apart from the cell-boxes, each cell also has a tight bounding box, used to determine separateness.

An octtree may not be balanced. It is usually shallow, for example for our spherical head model with $N = 71686$ elements only 5 levels are needed with $K_c = 100$. Its major advantage is that the expansion centers are guaranteed to lie on a Cartesian grid with known spacing. Therefore only a limited number of distinct translation operators need to be precomputed. For our model with $N = 71686$ only 3776 operators T are needed.

3.7. Choice of parameters

The choice of λ and L is guided by time and accuracy considerations. The truncation error of (4) is proportional to $(\|\mathbf{r} - \mathbf{C}\|/\|\mathbf{r}' - \mathbf{C}\|)^{L+1}$ which is bounded by $\lambda^{-(L+1)}$. In Figure 2 (left) we show the relative accuracy of the approximation of $1/\|\mathbf{r}' - \mathbf{r}\|$ using (4) for various values of L as a function of the relative distance $\|\mathbf{r}' - \mathbf{C}\|/\|\mathbf{r} - \mathbf{C}\|$ for 10^4 random points.

For octtree-type cells the minimum useful value of λ is $\lambda_{\min} = \sqrt{3} \approx 1.73$ with a local-interaction neighborhood of $3^3 = 27$ cells. We have tested various values of λ and L for our three-layer sphere models. The optimal value of λ was always between $\lambda_{\min}$ and 3. In order to limit the amount of memory needed we therefore decided to set $\lambda = 2$, in agreement with [11].

There is no consensus about what accuracy is fundamentally required to calculate the MEG/EEG BEM interactions for the inverse (source identification) problem, due to the measurement and modeling errors. We have decided to require that the difference between the FMM and non-FMM implementations be less than 1% of the BEM discretization error (known for our spherical models). This corresponds to calculating the BEM interactions with relative accuracy about 10^{-4} which in turn required us to set $L = 10$.

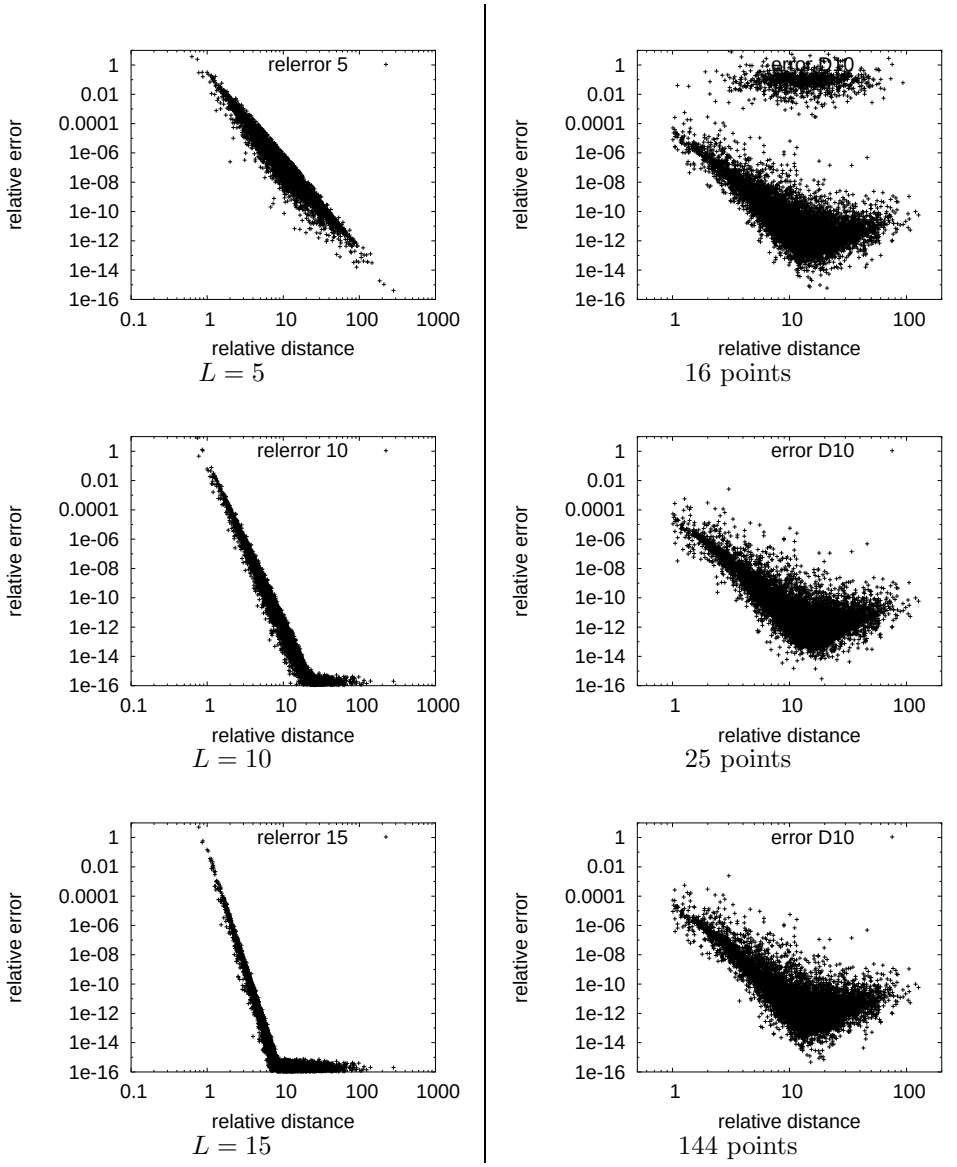

Figure 2. Left, from top to bottom: The relative error to approximate $1/\|\mathbf{r}' - \mathbf{r}\|$ using (4), as a function of the relative distance, for 10^4 random points, and for $L = 5, 10, 15$. Right, from top to bottom: The relative error to approximate elements of $\mathsf{D}_{i'j}$, for 10^4 random triangles and $L = 10$, as a function of the relative distance using three quadrature rules (QR): 16 point symmetric triangle QR, 25 point product Gauss QR , 144-point product Gauss QR.

3.8. Accuracy of the operator approximation

In order to choose the appropriate numerical quadrature procedure to implement (5,8), we evaluated the relative error of approximating S_{ij}, $\mathsf{D}_{i'j}$, $\mathsf{N}_{i'j'}$ as a function of relative distance between the elements (triangles) and the quadrature rule (QR) used, Figure 2, right. A 16 point QR is used for the direct computation (Section 2.1) [8, 14]. Integration of spherical harmonics at desired level of accuracy requires a $5 \times 5 = 25$ point tensor product Gauss quadrature. Increasing the order further does not bring any improvement.

3.9. Interaction between several trees

While standard FMM only considers one tree, for the symmetric BEM we need to treat multiple interacting trees because each surface is handled separately and we have two types of variables: potential V ($P1$) at vertices and flow p ($P0$) at faces. For external surfaces only the $P1$ tree is built, since the flow p is known to be zero there [8, 14]. First, an up-sweep phase is performed separately for each tree and outer fields are stored. Then a down-sweep phase is performed for each pair of trees that corresponds to surfaces delimiting a common volume, i.e. for all non-zero blocks in the system matrix A [8, 14]. All elements (vertices and faces) have a globally unique identification number that becomes an index of the corresponding variable.

To treat the multiple tree interactions efficiently, the direct interactions S,D, N are shared accross all trees, as well as the R, S and T operators – this requires sharing a common grid. Evaluating expansion coefficients a, b, c also shares many intermediate results.

4. Experiments

The superior accuracy of the symmetric BEM was already demonstrated in [8]. The purpose of this section is to demonstrate that this accuracy is not compromised by the FMM acceleration proposed, and that it allows to treat on a single computer, in a reasonable amount of time, far larger problems then the direct (non-accelerated) implementation. We have repeated the experiments from [8] and verified that the relative ℓ_2 error of the new FMM implementation with $\lambda = 2$, $L = 10$ is better than 10^{-3} with respect to the non-accelerated implementation, which is better than the error of the BEM method itself.

We have used spherical head models, as in [8], since analytical solutions are available for them. They consist of 3 concentric spheres with radii 0.87, 0.92, and 1.0, delimiting volumes with conductivities 1.0, 0.0125, 1.0 and 0.0, from inside towards outside. Different resolution meshes were used with $642, 2562$ and 10242 vertices per surface, corresponding to a total number of unknowns for the symmetric BEM equal to $4486, 17926$ and 71686, respectively. A dipolar source was placed at distance 0.425 from the center.

The experiments were performed on a computer with a 1.6 GHz 64 bit AMD Opteron processor with 5 GB of physical memory.

4.1. Single-sphere head models

The first series of experiments (Tables 2 and 3), was performed on a simplified head model containing one surface only, for a meaningful comparison with timings reported

Table 2. The elapsed CPU time and number of iterations needed to solve the forward problem for the single-sphere models as a function of the number of unknowns for the direct (no FMM) and FMM-accelerated symmetric BEM methods. Time needed to solve the problem using the double layer direct and precorrected-FFT method reported by Tissari and Rahola [15] for parameters $p = 3/p = 4$ are marked with *.

	Time [s]				Iterations	
Unknowns	direct sym.	direct dl.*	FMM	prec.-FFT*	FMM	prec.-FFT*
362		15		32/165		≤ 7
642	63		69		12	
1002		123		98/539		≤ 7
2252		714		270/1336		≤ 7
2562	1309		812		17	
5762		4760		724/4012		≤ 7
9002		12715		1223/6706		≤ 7
10242	26060		6325		26	
12962		27685		1934/9860		≤ 7

Table 3. The memory requirements (in MB) to solve the forward problem for the single-sphere head models as a function of the number of unknowns for the direct (no FMM) and FMM-accelerated symmetric BEM methods (the actual memory usage fluctuates due to the garbage collector). Memory requirements for the double layer direct and precorrected-FFT method reported by Tissari and Rahola [15] for parameters $p = 3/p = 4$ are marked with *.

Unknowns	direct sym.	direct dl.*	FMM	prec.-FFT*
362		20		22/25
642	65		21	
1002		34		32/40
2252		98		66/71
2562	288		95	
5762		539		126/169
9002		1288		202/268
10242	2546		881	
12962		2649		296/389

by Tissari and Rahola [15] using precorrected-FFT method. The programs were run with maximum number of elements per cell $K_c = 200$, expansion order $L = 10$, minimum relative distance $\lambda = 1.7$, and MINRES relative stopping threshold $\varepsilon = 10^{-6}$. The results of Tissari and Rahola are reported with parameters $p = 3, 4$ since this seems to correspond to the precision required. Non-accelerated direct method results are also shown.

According to the timings for the direct problem, their computer and implementation seem to be comparable to ours for the same number of unknowns, even though they use the double-layer formulation. It apparently has the advantage of involving a well-conditioned matrix with easy preconditioning, never requiring more than 6 or 7 iterations of the optimizer [15]. This emphasizes the assembly time with respect to the matrix-vector product evaluation time. For the largest problem, the assembly time is over 95 % of the total time, each iteration takes only about 10 s. This means that once the preprocessing is done, subsequent calculations for different sources can be relatively fast (< 5 min). Our FMM algorithm is always faster than the precorrected-FMM for $p = 4$ and they are comparable even for $p = 3$. On the other hand, the FMM algorithm seems to need more memory. Note that in this case (low

Table 4. The elapsed CPU time and number of iterations needed to solve the forward problem for the three-sphere models as a function of the number of unknowns for the direct (no FMM) and FMM-accelerated symmetric BEM methods. Relative error with respect to the analytical solution is also reported. We put between parentheses the extrapolated value for the largest problem, which could not be solved by the direct method due to lack of memory.

	Time [s]		Iterations		Rel. error [%]	
Unknowns	direct sym.	FMM	direct sym.	FMM	direct sym.	FMM
4486	2628	3030	238	238	0.989	0.989
17926	33928	70378	384	625	0.252	0.245
71686	*(542575)*	453600	N/A	900	N/A	0.090

Table 5. The memory requirements (in MB) to solve the forward problem for the three-sphere head models as a function of the number of unknowns for the direct (no FMM) and FMM-accelerated symmetric BEM methods in MB. We put between parentheses the extrapolated value for the largest problem, that could not be solved by the direct method due to lack of memory.

Unknowns	direct sym. [MB]	FMM
4486	91	123
17926	1390	1106
71686	*(22229)*	4400

number of iterations), the FMM version is faster and uses less memory than the direct version even for moderately sized problems and the subquadratic time complexity shows nicely.

The only other published FMM implementation we have found for symmetric BEM is by Of et al. [13]. After correcting for their use of only 7-point integration rules (the degree of their spherical harmonic expansion was not reported), their timings are comparable to ours as well.

4.2. Three-sphere head models

The results shown in Tables 4 and 5 correspond to $K_c = 300$, $L = 10$, $\lambda = 1.7$ and $\varepsilon = 10^{-6}$. We observe that no FMM acceleration takes place for the smallest mesh ($N = 4486$). For the middle one ($N = 17926$), FMM brings a memory saving but is still slower. For the largest mesh, FMM enables us to produce a valid result, which could not be calculated by the direct method for lack of memory. The assembly time is about 80 % of the total time, each iteration takes about 1.5 min. Calculation for another source would therefore take about 22 h.

5. Discussion

The Finite Element Method (FEM) implementation [18] is capable of solving a problem of this size but requires a well-formed and topologically correct 3D mesh which is difficult to automatically generate from the head MRI scans or the surface meshes.§

FMM performs much better for the single-sphere model compared to the three-sphere model, mainly because the conditioning of the system for three-sphere model

§ Most 3D meshing software is commercial (not freely available) and does not support adaptive meshing, making the resulting models extremely large.

system is one order of magnitude worse, due to the low conductivity of the middle layer corresponding to the skull. Moreover, because the three surfaces are close to one another, there are many more elements to be treated locally. Consequently, the subquadratic time complexity only manifests itself for very large problems. Nevertheless, the FMM enables us to solve middle-sized problems, which barely fit in the memory of current computers. Indeed, the most critical point in implementing the FMM turns out to be memory management. Most of the memory is used to store the precalculated local interactions.

Our FMM is tailored for the symmetric BEM and uses several mutually interacting trees of two different types (P0/P1). Caching and a predetermined interaction plan are established to eliminate overhead.

From the timings (Section 3.5) and by analysing the required number of most costly basic operations, we conclude that there is unfortunately little hope of significantly algorithmically accelerating the FMM for small or medium BEM problems, unless computationally more efficient expansions and translations can be established. Replacing spherical harmonics by pseudoparticles [19] or Cartesian polynomials [20] might be an alternative to consider.

6. Conclusion

We have developed a fast multipole method to accelerate the symmetric BEM, with application to the forward MEG/EEG problem. The FMM is as accurate as the symmetric BEM with direct assembly, and with increasing problem size it gets faster and requires less memory than the direct method.

1. Appendix: Spherical harmonics

There are several definitions of spherical harmonics and Legendre polynomials, differing mainly in normalization and sign conventions, and we follow the conventions of Epton and Dembart [10]. The spherical harmonic $Y_n^m(\theta,\phi)$ is

$$Y_n^m(\theta,\phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}}(-1)^m P_n^{|m|}(\cos\theta)\, e^{im\phi}$$

where $P_n^m(x)$ are the associated Legendre polynomials

$$P_n^m(x) = \frac{1}{2^n n!}\frac{(n+m)!}{(n-m)!}(1-x^2)^{-m/2}\frac{d^{n-m}}{dx^{n-m}}(x^2-1)^n$$

Given a vector $\mathbf{x} = (x,y,z) = (r\,\cos\phi\,\sin\theta, r\,\sin\phi\,\sin\theta, r\,\cos\theta)$, the inner and outer spherical harmonics are defined by

$$O_n^m(\mathbf{x}) = \frac{(-1)^n i^{|m|}}{A_n^m}\frac{Y_n^m(\theta,\phi)}{r^{n+1}}$$

$$I_n^m(\mathbf{x}) \;= i^{-|m|} A_n^m r^n Y_n^m(\theta,\phi) \text{ with } \quad A_n^m = \frac{(-1)^n}{\sqrt{(n-m)!\,(n+m)!}}.$$

Acknowledgments

Sponsored by the Czech Ministry of Education under Project MSM6840770012.

[1] J. Phillips, R. Leahy, J. Mosher, and B. Timsari, "Imaging neural activity using MEG and EEG," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 3, pp. 34–42, 1997.

[2] M. Hämäläinen, R. Hari, R. J. IImoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography— theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, no. 2, pp. 413–497, Apr. 1993.

[3] J. Rahola and S. Tissari, "Iterative solution of dense linear systems arising from the electrostatic integral equation," *Phys. Med. Biol.*, no. 47, pp. 961–975, 2002.

[4] ——, "Iterative solution of dense linear systems arising from boundary element formulations of the biomagnetic inverse problem," CERFACS, Tech. Rep. TR/PA/98/40, 1998, toulouse, France.

[5] A. S. Ferguson and G. Stroink, "Factors affecting the accuracy of the boundary element method in the forward problem — I: Calculating surface potentials," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 11, pp. 1139–1155, Nov. 1997.

[6] M. Clerc, R. Keriven, O. Faugeras, J. Kybic, and T. Papadopoulo, "The fast multipole method for the direct E/MEG problem," in *Proceedings of ISBI*. Washington, D.C.: IEEE, NIH, July 2002. [Online]. Available: http://www.biomedicalimaging.org/

[7] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," *J. Comput. Phys.*, no. 155, pp. 468–498, 1999. [Online]. Available: http://amath.colorado.edu/courses/7400/2003fall/005/papers/ChengGreenga%rdRokhlin.pdf

[8] J. Kybic, M. Clerc, T. Abboud, O. Faugeras, R. Keriven, and T. Papadopoulo, "A common formalism for the integral formulations of the forward EEG problem," *IEEE Transactions on Medical Imaging*, vol. 24, no. 1, pp. 12–28, Jan. 2005.

[9] R. K. Beatson and L. Greengard, "A short course on fast multipole methods," in *Wavelets, Multilevel Methods and Elliptic PDEs*, M. Ainsworth, J. Levesley, W. Light, and M. Marletta, Eds. Oxford University Press, 1997, pp. 1–37. [Online]. Available: http://www.math.canterbury.ac.nz/~mathrkb/pdfs/beatson+greengard/beatso%ngreengard.pdf

[10] M. A. Epton and B. Dembart, "Multipole translation theory for the three-dimensional Laplace and Helmholtz equations," *SIAM J. Sci. Comput.*, vol. 16, no. 4, pp. 865–897, July 1995.

[11] B. Dembart and E. Yip, "The accuracy of fast multipole methods for Maxwell's equations," *IEEE Comput. Sci. Eng.*, vol. 5, no. 3, pp. 48–56, 1998. [Online]. Available: http://dx.doi.org/10.1109/99.714593

[12] J. Rahola, "Experiments on iterative methods and the fast multipole method in electromagnetic scattering calculations," CERFACS, Tech. Rep. TR/PA/98/49, 98.

[13] G. Of, O. Steinbach, and W. L. Wendland, "A fast multipole boundary element method for the symmetric boundary integral formulation," in *Proceedings of IABEM*, Austin, TX, USA, 2002. [Online]. Available: http://cavity.ce.utexas.edu/iabem2002/fullpapers/of.pdf

[14] J. Kybic, M. Clerc, O. Faugeras, R. Keriven, and T. Papadopoulo, "Fast multipole method for the symmetric boundary element method in meg/eeg," INRIA, Tech. Rep. 5415, Dec. 2004.

[15] S. Tissari and J. Rahola, "A precorrected-FFT method to accelerate the solution of the forward problem in magnetoencephalography," *Phys. Med. Biol.*, no. 48, pp. 523–541, 2003.

[16] S. Börm and L. G. und Wolfgang Hackbusch, "Introduction to hierarchical matrices with applications," *Engineering Analysis with Boundary Elements*, no. 27, pp. 405–422, 2003.

[17] J. Kybic and M. Clerc, "Symmetric BEM and multiscale fast multipole method for the E/MEG problem," in *NFSI 2003: Proceedings of the 4th International Symposium on Noninvasive Functional Source Imaging Within the Human Heart and Brain*, V. Pizzella and G. L. Romani, Eds., Berlin, Germany, Sept. 2003, pp. 122–124.

[18] M. Clerc, A. Dervieux, O. Faugeras, R. Keriven, J. Kybic, and T. Papadopoulo, "Comparison of BEM and FEM methods for the E/MEG problem," in *Proceedings of BIOMAG 2002*, Aug. 2002.

[19] J. Makino, "Yet another fast multipole method without multipoles — pseudoparticle multipole method," *J. Comput. Phys.*, vol. 151, no. 2, pp. 910–920, 1999, academic Press Professional, Inc.

[20] D. Apalkov and P. Visscher, "Fast multipole method for micromagnetic simulation of periodic systems," *IEEE Transactions on Magnetics*, vol. 39, no. 6, pp. 3478–3480, 2003.