A FAST SOLUTION METHOD FOR THREE-DIMENSIONAL MANY-PARTICLE PROBLEMS OF LINEAR ELASTICITY

Yuhong Fu, Kenneth J. Klimkowski, Gregory J. Rodin†, Emery Berger, James C. Browne, Jürgen K. Singer, Robert A. van de Geijn, and Kumar S. Vemaganti

Texas Institute for Computational and Applied Mathematics

The University of Texas at Austin, Austin, Texas 78712, USA

ABSTRACT

A boundary element method for solving three-dimensional linear elasticity problems that involve a large number of particles embedded in a binder is introduced. The proposed method relies on an iterative solution strategy in which matrix-vector multiplication is performed with the fast multipole method. As a result the method is capable of solving problems with N unknowns using only $\mathcal{O}(N)$ memory and $\mathcal{O}(N)$ operations. Results are given for problems with hundreds of particles in which $N = \mathcal{O}(10^5)$.

KEY WORDS: boundary element method; fast multipole method; many-particle problem; linear elasticity; iterative solution strategy

†to whom correspondence should be addressed, e-mail gjr@ticam.utexas.edu

1 Introduction

In this paper, we introduce a fast boundary element method (BEM) for solving three-dimensional linear elasticity problems that involve a large number of particles embedded in a binder. We refer to those problems as many-particle problems and to the new method as FLEMS: Fast Linear Elastic Many-particle problem Solver.

For many-particle problems, FLEMS combines the robustness of conventional BEMs with superior performance. To date, FLEMS has been applied to analysis of problems that involve tens of particles on desktop computers and hundreds of particles on small parallel computers. Solution of problems with thousands of particles is feasible with large parallel computers. The ability to solve such problems may dramatically change the field of micromechanics. In this regard, we believe that in the near future various issues concerned with the overall response of composite materials (Hashin, 1983) will be settled and the focus of research will be shifted to problems concerned with edge effects, interlaminar stress distribution, size effects, etc.

In order to compare FLEMS with conventional BEMs, let us recall that conventional BEMs give rise to dense algebraic problems and, in the BEM community, it is customary to solve those problems using direct methods such as LU decomposition or others derived from Gaussian elimination. In this case, a problem with N unknowns requires $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ operations. These requirements are practically unacceptable for many-particle problems for which $N > 10^4$.

In the last decade, iterative solution strategies have become increasingly recognized by the BEM community. In particular, effective iterative solution strategies for many-particle problems have been developed by Phan-Tien and Kim (1994). FLEMS also relies on an iterative solution strategy that, for many-particle problems, requires only an $\mathcal{O}(N)$ memory and $\mathcal{O}(N)$ operations; in summary, FLEMS is an $\mathcal{O}(N)$ method. This dramatic improvement in performance, in comparison to

conventional BEMs, comes from two sources. First, as a rule, many-particle problems give rise to well-conditioned algebraic problems for which effective preconditioners can be constructed. As a result the number of required iterations is usually small and virtually independent of N and therefore the operation count is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$. Second, at each iteration, matrix-vector multiplication is performed using the fast multipole method (Rokhlin, 1985; Greengard and Rokhlin, 1987) which computes the product using only $\mathcal{O}(N)$ memory and $\mathcal{O}(N)$ operations. The reader unfamiliar with the fast multipole method (FMM) is referred to Rokhlin (1985), Greengard and Rokhlin (1987), Greengard (1988), and Greengard and Rokhlin (1996). Nevertheless, in order to understand this paper, the reader needs to know very little about the FMM, and this information is provided in Section 2. For now, it is sufficient to know that the FMM is a method for solving the many-body electrostatics problem.

Although the idea of applying the FMM to iterative solution of integral equations dates back to the seminal paper of Rokhlin (1985), there are good reasons why the FMM has not been frequently applied to three-dimensional elliptic boundary-value problems, especially those arising in linear elasticity and low-Reynolds-numbers hydrodynamics. For three-dimensional boundary-value problems, the FMM is used infrequently because it is meaningful only if the problem size exceeds $N = \mathcal{O}(10^3 - 10^4)$. In particular, we are aware of only one research group that uses the FMM to solve boundary integral equations corresponding to three-dimensional Laplace's equation (Nabors *et al.*, 1994). In contrast, the break-even point in two dimensions is $N = \mathcal{O}(10^2)$, and, for current desktop computers, this difference is significant. Fortunately, this situation will not last because recently Greengard and Rokhlin (1996) improved the three-dimensional FMM and reduced its break-even point to $N = \mathcal{O}(10^2)$; of course further progress in desktop computers will make the three-dimensional FMM even more attractive. The reason why the FMM is not frequently applied to solution of boundary-value problems other than those governed by Laplace's equation is due

to the fact that Laplace's equation is directly related to the many-body electrostatics problem while other partial differential equations are not. Nevertheless several researchers have been able to overcome this obstacle. Among those we cite the studies of Greenbaum et al. (1992) concerned with the two-dimensional biharmonic equation, of Peirce and Napier (1995) concerned with 2-D Navier's equations of linear elasticity, and of Sangani and Mo (1996) concerned with three-dimensional equations of low-Reynolds-numbers hydrodynamics (Stokes' equations). Those researchers modified either the integral equations to accommodate the FMM (Greenbaum et al., 1992) or the FMM to accommodate the integral equations (Peirce and Napier, 1995; Sangani and Mo, 1996). Such modifications are non-trivial and often problem specific. In particular, the approach of Greenbaum *et al.* (1992) cannot be extended to three dimensions, the version of the FMM developed by Sangani and Mo (1996) cannot be easily extended to three-dimensional linear elasticity because it is limited to rigid particles, and the version of the FMM developed by Peirce and Napier (1995) is somewhat inefficient.

The principal results reported in this paper are as follows. First, we established a relationship between boundary integral equations of three-dimensional linear elasticity and the FMM. This relationship allows one to combine any BEM formulation with any FMM formulation. Second, based on the new relationship, we developed FLEMS – a parallel C++/C code that can be easily modified to accommodate better BEM or FMM formulations.

The remainder of this paper is organized as follows. In Section 2, we describe how to adopt the FMM for solution of integral equations. First, for those unfamiliar with the FMM, we summarize its basic ideas. Then we explain how to extend the FMM to generalized many-body electrostatics problems and demonstrate that a combination of those problems can be used to represent the biharmonic kernel. In Section 3, we formulate BEM equations. Most of the material presented in Section 3

is standard, except for new expressions for the fundamental solutions that allow us to relate discretized boundary integral equations of three-dimensional linear elasticity to a combination of sixteen generalized many-body electrostatics problems. In Section 4, we describe the implementation of the iterative solution strategy. In Section 5, we explain the basic ideas behind the parallel version of FLEMS. In Section 6, we present test problems in which the number of particles varies from 8 to 343. In Section 7, we summarize results and briefly discuss future work.

2 Fast Multipole Method

In the many-body electrostatics problem, one is given a unit cube containing n charges q_i at positions $\boldsymbol{x}_i$ and asked to determine the forces $\boldsymbol{f}_i$ acting on those charges. The forces are calculated in terms of the potentials as

$$\boldsymbol{f}_i = -q_i \, \nabla \Phi(\boldsymbol{x}_i) \tag{1}$$

and

$$\Phi(\boldsymbol{x}_i) = \sum_{j=1, j \neq i}^{n} \frac{q_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|} \, ; \qquad i = 1, ..., n. \tag{2}$$

While ordinarily the solution of this problem requires $\mathcal{O}(n^2)$ operations, for large n, one can compute the forces with the FMM in $\mathcal{O}(n)$ operations only. We outline the FMM in the next paragraph. For details, the reader is referred to Rokhlin (1985), Greengard and Rokhlin (1987), Greengard (1988), and Greengard and Rokhlin (1996).

The FMM exploits the well-known idea that the potential field induced by a cluster of charges can be approximated by the potential field of their total charge or higher order moments, as long as the target charges are sufficiently far away from the source charges. In the FMM, the clusters are formed via hierarchical partitioning of the unit cube into 2^3, 4^3, 8^3, etc. identical cubic cells which interact among themselves based on the machinery of the multipole and solid spherical harmonic (local) expansions.

The computation proceeds in three stages. First, the multipole coefficients (charge, dipole, etc.) are computed for all cells. Second, the cell-cell interactions are computed so that the local expansion coefficients of the smallest cells are related to the multipole coefficients. Third, the potentials are computed by adding the potentials due to the local expansions and nearby charges. As a result the solution is expressed as

$$\Phi(\boldsymbol{x}_i) = \sum_{k=1}^{(p+1)^2} l_k h_k(\boldsymbol{x}_i - \boldsymbol{c}) + \widetilde{\sum_{j \neq i}} \frac{q_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|} + \text{error} \tag{3}$$

and

$$\boldsymbol{f}_i = \sum_{k=1}^{(p+1)^2} l_k[-q_i \nabla h_k(\boldsymbol{x}_i - \boldsymbol{c})] + \text{remaining terms}\,. \tag{4}$$

In these equations, l's are the local expansion coefficients, h's are the solid spherical harmonics, $\boldsymbol{c}$ is the center of the smallest cell containing q_i, and the tilde sign denotes that the second sum includes nearby charges only. The upper limit of the local expansion is $(p+1)^2$ because this is the number of the solid spherical harmonics of order less or equal to p. The error term, which can be bounded *a priori*, is controlled by choosing a sufficiently large p. Note that most of the FMM machinery is needed for computing l's – evaluation of (3) and (4) for given l's is a simple computation performed pointwise, one charge at a time.

In order to extend the FMM to many-body problems associated with discretized integral equations of three-dimensional linear elasticity we introduce two generalizations of the many-body electrostatics problem. The first generalization is related to the fact that collocation BEMs are endowed with two point sets. One point set is formed by the nodal points which are also used as the collocation points. The other point set is formed by the integration points. To distinguish between these two sets we regard the collocation points as the target points and the integration points as the source points. In what follows, we denote the target points by $\boldsymbol{x}$ and the source points by $\boldsymbol{y}$. The second generalization is related to the fact that the fundamental solutions of linear elasticity involve the biharmonic $|\boldsymbol{x} - \boldsymbol{y}|$ kernel to which, without

modifications, the FMM cannot be applied. In this regard, we observe that only minor modifications are required in order to extend the FMM to problems in which (i) the charges at the source points are prescribed in terms of a function $q = \mathcal{Q}(\boldsymbol{y})$ and (ii) the forces at the target points are prescribed in terms of a linear map $\mathcal{F}$ on $\Phi(\boldsymbol{x})$ so that

$$\boldsymbol{f}_i = \mathcal{F}[\Phi(\boldsymbol{x}_i), \boldsymbol{x}_i] = \sum_{k=1}^{(p+1)^2} l_k \mathcal{F}[h_k(\boldsymbol{x}_i - \boldsymbol{c}), \boldsymbol{x}_i] + \text{remaining terms}\,. \tag{5}$$

It is obvious that (5) is a pointwise computation that can be implemented once the FMM is supplemented with a 'pre-processing' procedure for evaluation of $q = \mathcal{Q}(\boldsymbol{y})$ and a 'post-processing' procedure for evaluation of $\mathcal{F}[h_k(\boldsymbol{x}), \boldsymbol{x}]$. Note that with the introduction of the generalized many-body electrostatics problem we amplify an observation of Greengard (1994) that the FMM is a separation of variables technique for the harmonic kernel $|\boldsymbol{x} - \boldsymbol{y}|^{-1}$. From this perspective, we extend the FMM to the kernel $X(\boldsymbol{x})\, Y(\boldsymbol{y})\, |\boldsymbol{x} - \boldsymbol{y}|^{-1}$.

As an example, let us consider how the generalization we have introduced can be applied to the biharmonic kernel. In the following expression, we use the subscripts to denote Cartesian components of $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\begin{aligned}
|\boldsymbol{x} - \boldsymbol{y}| = {} & \underbrace{\boldsymbol{x} \cdot \boldsymbol{x}}_{\mathcal{F}_1(\boldsymbol{x})} \times \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \times \underbrace{1}_{\mathcal{Q}_1(\boldsymbol{y})} \\
& + \underbrace{(-2x_1)}_{\mathcal{F}_2(\boldsymbol{x})} \times \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \times \underbrace{y_1}_{\mathcal{Q}_2(\boldsymbol{y})} \\
& + \underbrace{(-2x_2)}_{\mathcal{F}_3(\boldsymbol{x})} \times \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \times \underbrace{y_2}_{\mathcal{Q}_3(\boldsymbol{y})} \\
& + \underbrace{(-2x_3)}_{\mathcal{F}_4(\boldsymbol{x})} \times \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \times \underbrace{y_3}_{\mathcal{Q}_4(\boldsymbol{y})} \\
& + \underbrace{1}_{\mathcal{F}_5(\boldsymbol{x})} \times \frac{1}{|\boldsymbol{x} - \boldsymbol{y}|} \times \underbrace{\boldsymbol{y} \cdot \boldsymbol{y}}_{\mathcal{Q}_5(\boldsymbol{y})}\,.
\end{aligned}$$

In this representation the harmonic kernel is distilled from the biharmonic one and

as a result the many-body problem associated with the three-dimensional biharmonic equation is reduced to five generalized many-body electrostatics problems.

In conclusion of this section, we observe that, strictly speaking, the FMM is not an $\mathcal{O}(N)$ method because in order to optimize the performance the partitioning depth should be a logarithmic function of N. We avoid this issue in this paper and keep referring to the FMM as an $\mathcal{O}(N)$ method; for details see Aluru (1996).

3 Governing Equations

The principal objective in this section is to formulate the boundary integral and corresponding algebraic equations of many-particle problems in a form suitable for the FMM application. We accomplish this objective by combining well-established BEM procedures (Brebbia and Dominguez, 1984) with new representations for the fundamental solutions of three-dimensional linear elasticity.

3.1 Integral Equations

In this section, we formulate the boundary integral equations for a simple many-particle problem that involves an infinite binder with M identical particles perfectly bonded to the binder. This problem is selected for demonstration purposes. There are no conceptual obstacles to extending our approach to problems that involve finite binders, different particles, and debonding.

The problem of interest is formulated in three steps. First, we formulate an auxiliary boundary-value problem for a finite body with M embedded particles. Second, we formulate the boundary integral equations for the finite body and introduce new representations for the fundamental solutions. Finally, we derive the boundary integral equations for the infinite body. From now on, we use the subscripts to denote Cartesian components and superscripts to denote attributes. For position vectors, we use interchangeably the indicial and direct conventions; for example, the notations $\boldsymbol{x}$

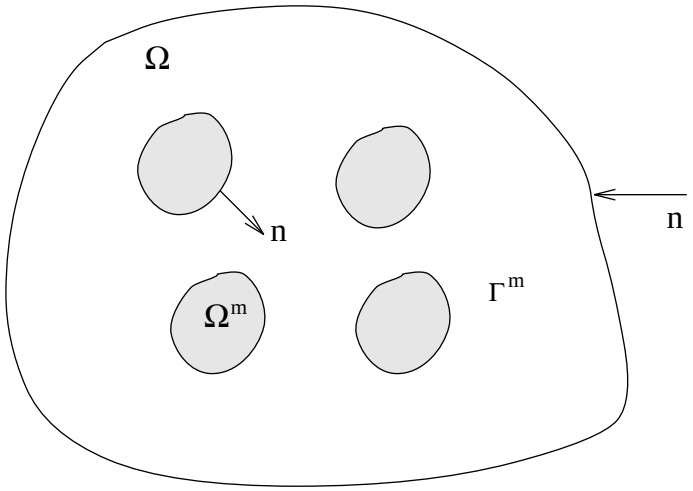

Figure 1: A binder containing identical particles.

and x_i are equivalent.

Consider a finite three-dimensional linear elastic body formed by a binder Ω and M embedded identical particles Ω^m perfectly bonded to the binder (Fig. 1). We denote the surface of this body by Γ^0, the interface boundaries by Γ^m, and their union, $\cup_{m=1}^{M}\Gamma^m$, by Γ. Lame's constants of the binder are denoted by λ and μ and the corresponding Poisson's ratio by ν. For the particles, the corresponding constants are marked with the asterisk superscript. The body is loaded by a displacement field $u_i^0(\boldsymbol{x})$ prescribed on Γ^0.

The displacement field in the body, $\tilde{u}_i(\boldsymbol{x})$, is the solution of the following boundary-value problem:

$$(\lambda+\mu)\tilde{u}_{j,ij}(\boldsymbol{x})+\mu\tilde{u}_{i,jj}(\boldsymbol{x})=0 \qquad \forall\,\boldsymbol{x}\in\Omega\,, \tag{6}$$

$$(\lambda^*+\mu^*)\tilde{u}_{j,ij}(\boldsymbol{x})+\mu^*\tilde{u}_{i,jj}(\boldsymbol{x})=0 \qquad \forall\,\boldsymbol{x}\in\cup_{m=1}^{M}\Omega^m\,, \tag{7}$$

$$\tilde{u}_i(\boldsymbol{x})=u_i^0(\boldsymbol{x}) \qquad \forall\,\boldsymbol{x}\in\Gamma^0\,, \tag{8}$$

$$[\![\tilde{u}_i(\boldsymbol{x})]\!]=0 \quad \text{and} \quad [\![\tilde{t}_i(\boldsymbol{x})]\!]=0, \qquad \forall\,\boldsymbol{x}\in\Gamma\,. \tag{9}$$

In (9) $[\![\ldots]\!]$ denotes the jump along the normal vector, $n_i(\boldsymbol{x})$, as defined in Figure 1.

Equations (6-9) are equivalent to a system of $M+1$ integral equations. The first equation in this system is defined on the binder boundary, $\Gamma^0 \cup \Gamma$, and each of the remaining equations is defined on the boundary of a particle:

$$\begin{aligned} \frac{1}{2}\tilde{u}_i(\boldsymbol{x}) \;&-\; \int_{\Gamma^0} T_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{u}_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;-\; \int_{\Gamma} T_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{u}_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;= \\ &-\; \int_{\Gamma^0} U_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{t}_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;-\; \int_{\Gamma} U_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{t}_j(\boldsymbol{y}) \end{aligned} \tag{10}$$

and

$$\frac{1}{2}\tilde{u}_i(\boldsymbol{x}) + \int_{\Gamma^m} T^*_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{u}_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \int_{\Gamma^m} U^*_{ij}(\boldsymbol{x},\boldsymbol{y})\tilde{t}_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}\,, \quad m = 1,2,...,M\,. \tag{11}$$

In (10) and (11) and in the rest of the paper it is implied that $\boldsymbol{x}$ is an arbitrary point on the surface on which the integral equation is defined.

The kernels $U_{ij}(\boldsymbol{x},\boldsymbol{y})$ and $T_{ij}(\boldsymbol{x},\boldsymbol{y})$ introduced in (10) are the fundamental solutions of three-dimensional linear elasticity. Usually they are expressed as

$$U_{ij}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{16\pi\mu(1-\nu)|\boldsymbol{x}-\boldsymbol{y}|}\left[(3-4\nu)\delta_{ij} + \frac{(x_i-y_i)(x_j-y_j)}{|\boldsymbol{x}-\boldsymbol{y}|^2}\right] \tag{12}$$

and

$$\begin{aligned} T_{ij}(\boldsymbol{x},\boldsymbol{y}) \;&=\; \frac{1-2\nu}{8\pi(1-\nu)|\boldsymbol{x}-\boldsymbol{y}|^3}\left[-(x_i-y_i)n_j(\boldsymbol{y}) + (x_j-y_j)n_i(\boldsymbol{y}) + \delta_{ij}(x_s-y_s)n_s(\boldsymbol{y})\right] \\ &+\; \frac{3}{8\pi(1-\nu)|\boldsymbol{x}-\boldsymbol{y}|^5}(x_i-y_i)(x_j-y_j)(x_s-y_s)n_s(\boldsymbol{y})\,, \end{aligned} \tag{13}$$

where δ_{ij} is the second rank identity tensor. The kernels $U^*_{ij}(\boldsymbol{x},\boldsymbol{y})$ and $T^*_{ij}(\boldsymbol{x},\boldsymbol{y})$ introduced in (11) can be obtained from (12) and (13) once μ and ν are replaced by μ^* and ν^*, respectively.

The kernels $U_{ij}(\boldsymbol{x},\boldsymbol{y})$ and $T_{ij}(\boldsymbol{x},\boldsymbol{y})$ can be also expressed as

$$U_{ij}(\boldsymbol{x},\boldsymbol{y}) = P_{ij}(\boldsymbol{x})\left(\frac{1}{|\boldsymbol{x}-\boldsymbol{y}|}\right) + Q_i(\boldsymbol{x})\left(\frac{1}{|\boldsymbol{x}-\boldsymbol{y}|}y_j\right) \tag{14}$$

and

$$T_{ij}(\boldsymbol{x},\boldsymbol{y}) = R_{ijp}(\boldsymbol{x})\left[\frac{1}{|\boldsymbol{x}-\boldsymbol{y}|}n_p(\boldsymbol{y})\right] + S_{ip}(\boldsymbol{x})\left[\frac{1}{|\boldsymbol{x}-\boldsymbol{y}|}n_p(\boldsymbol{y})y_j\right]\,, \tag{15}$$

where

$$P_{ij}(\boldsymbol{x}) = \frac{1}{16\pi\mu(1-\nu)}\left[(3-4\nu)\delta_{ij} - x_j\partial_i\right] , \tag{16}$$

$$Q_i(\boldsymbol{x}) = \frac{1}{16\pi\mu(1-\nu)}\partial_i , \tag{17}$$

$$R_{ijp}(\boldsymbol{x}) = \frac{1}{8\pi(1-\nu)}[(1-2\nu)(\delta_{jp}\partial_i - \delta_{ij}\partial_p) - 2(1-\nu)\delta_{ip}\partial_j + x_j\partial_i\partial_p] , \tag{18}$$

and

$$S_{ip}(\boldsymbol{x}) = -\frac{1}{8\pi(1-\nu)}\partial_i\partial_p . \tag{19}$$

In (16-19), the symbol ∂_i denotes the partial derivative with respect to x_i. Note that in (14) and (15) the terms dependent on $\boldsymbol{x}$ are separated from those dependent on $\boldsymbol{y}$ except for the harmonic kernel $|\boldsymbol{x}-\boldsymbol{y}|^{-1}$.

Now let us evaluate (10) and (11) under the conditions that Γ^0 tends to infinity and $u_i^0(\boldsymbol{x}) = \epsilon_{ij}^0 x_j$, where ϵ_{ij}^0 is a constant tensor. To do this we first consider an auxiliary problem for the finite binder from which the particles are removed and in which the linear displacement field $u_i^0(\boldsymbol{x}) = \epsilon_{ij}^0 x_j$ is realized by prescribing $u_i^0(\boldsymbol{x}) = \epsilon_{ij}^0 x_j$ on the entire boundary $\Gamma^0 \cup \Gamma$. The corresponding stress and traction fields are denoted by σ_{ij}^0 and t_i^0, respectively; $t_i^0 = \sigma_{ij}^0 n_j$. The integral equation for the auxiliary problem parallels (10):

$$\begin{aligned} \frac{1}{2}u_i^0(\boldsymbol{x}) \;-\; & \int_{\Gamma^0} T_{ij}(\boldsymbol{x},\boldsymbol{y})u_j^0(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;-\; \int_{\Gamma} T_{ij}(\boldsymbol{x},\boldsymbol{y})u_j^0(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;= \\ -\; & \int_{\Gamma^0} U_{ij}(\boldsymbol{x},\boldsymbol{y})t_j^0(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;-\; \int_{\Gamma} U_{ij}(\boldsymbol{x},\boldsymbol{y})t_j^0(\boldsymbol{y}) . \end{aligned} \tag{20}$$

Once (20) is subtracted from (10), the integrals over Γ^0 disappear as Γ^0 tends to infinity and we obtain

$$\frac{1}{2}u_i(\boldsymbol{x}) - \int_{\Gamma} T_{ij}(\boldsymbol{x},\boldsymbol{y})u_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} + \int_{\Gamma} U_{ij}(\boldsymbol{x},\boldsymbol{y})t_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = 0 , \tag{21}$$

where

$$u_i(\boldsymbol{x}) = \tilde{u}_i(\boldsymbol{x}) - u_i^0(\boldsymbol{x}) \quad \text{and} \quad t_i(\boldsymbol{x}) = \tilde{t}_i(\boldsymbol{x}) - t_i^0(\boldsymbol{x}). \tag{22}$$

In terms of the perturbation fields, (11) is rewritten in the form

$$\frac{1}{2}u_i(\boldsymbol{x}) + \int_{\Gamma^m} T^*_{ij}(\boldsymbol{x},\boldsymbol{y})u_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - \int_{\Gamma^m} U^*_{ij}(\boldsymbol{x},\boldsymbol{y})t_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;= g_i(\boldsymbol{x}) \tag{23}$$

with

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\epsilon^0_{ij}x_j - \epsilon^0_{js}\int_{\Gamma^m} T^*_{ij}(\boldsymbol{x},\boldsymbol{y})y_s\,\mathrm{d}\boldsymbol{y} + \sigma^0_{js}\int_{\Gamma^m} U^*_{ij}(\boldsymbol{x},\boldsymbol{y})n_s(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}\,. \tag{24}$$

In (23) and (24) it is implied that m runs from 1 to M.

In the remainder of the paper, we are concerned with solving integral equations (21) and (23). The most challenging part of this task is to solve (21) which can be characterized as a coupled problem for M particles. For this problem the use of the FMM is critical. In contrast, (23) represents M relatively small uncoupled problems for which a conventional BEM is adequate.

3.2 Algebraic Equations

In this section, (21) and (23) are converted into algebraic equations following standard finite element discretization techniques.

The finite element representation of Γ is based on the map

$$x_i(\boldsymbol{\xi}) = \sum_{k,l} x^{kl}_i \phi^l(\boldsymbol{\xi}). \tag{25}$$

In (25) $\boldsymbol{\xi}$ is the position vector in the master element space, $\boldsymbol{x}^{kl}$ is the position vector of the node l of the element k, and $\phi^l(\boldsymbol{\xi})$ is the shape function of the node l. For isoparametric finite elements, the maps $u_i[\boldsymbol{x}(\boldsymbol{\xi})]$ and $t_i[\boldsymbol{x}(\boldsymbol{\xi})]$ mimic (25),

$$u_i[\boldsymbol{x}(\boldsymbol{\xi})] = \sum_{k,l} u^{kl}_i \phi^l(\boldsymbol{\xi}) \quad \text{and} \quad t_i[\boldsymbol{x}(\boldsymbol{\xi})] = \sum_{k,l} t^{kl}_i \phi^l(\boldsymbol{\xi})\,. \tag{26}$$

When (25) and (26) are substituted in (21) we obtain

$$\begin{aligned} \frac{1}{2}u_i(\boldsymbol{x}) \;&-\; \sum_{k,l}\int_\Theta T_{ij}[\boldsymbol{x},\boldsymbol{y}(\boldsymbol{\xi})]u^{kl}_j\phi^l(\boldsymbol{\xi})J[\boldsymbol{y}(\boldsymbol{\xi})]\,\mathrm{d}\boldsymbol{\xi} \\ &+\; \sum_{k,l}\int_\Theta U_{ij}[\boldsymbol{x},\boldsymbol{y}(\boldsymbol{\xi})]t^{kl}_j\phi^l(\boldsymbol{\xi})J[\boldsymbol{y}(\boldsymbol{\xi})]\,\mathrm{d}\boldsymbol{\xi} = 0\,. \end{aligned} \tag{27}$$

In (27) the integrals are over the master element space Θ and $J[\boldsymbol{y}(\boldsymbol{\xi})]$ is the Jacobian corresponding to (25). Numerical integration of (27) followed by collocation at the nodal points $\boldsymbol{x}^q$ yields the algebraic equation

$$\begin{aligned}\frac{1}{2}u_i(\boldsymbol{x}^q) &- \sum_{k,l,s} T_{ij}[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]u_j^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s \\ &+ \sum_{k,l,s} U_{ij}[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]t_j^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s = 0. \end{aligned} \tag{28}$$

In (28) s runs over the integration points of the master element and ω^s is the integration weight at $\boldsymbol{\xi}^s$. Upon substitution of (14-19), (28) can be rewritten as

$$\begin{aligned}\frac{1}{2}u_i(\boldsymbol{x}^q) &- R_{ijp}(\boldsymbol{x}^q)\sum_{k,l,s}\frac{1}{|\boldsymbol{x}^q-\boldsymbol{y}(\boldsymbol{\xi}^s)|}\left\{u_j^{kl}\phi^l(\boldsymbol{\xi}^s)n_p[\boldsymbol{y}(\boldsymbol{\xi}^s)]J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s\right\} \\ &- S_{ip}(\boldsymbol{x}^q)\sum_{k,l,s}\frac{1}{|\boldsymbol{x}^q-\boldsymbol{y}(\boldsymbol{\xi}^s)|}\left\{u_j^{kl}\phi^l(\boldsymbol{\xi}^s)n_p[\boldsymbol{y}(\boldsymbol{\xi}^s)]y_j(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s\right\} \\ &+ P_{ij}(\boldsymbol{x}^q)\sum_{k,l,s}\frac{1}{|\boldsymbol{x}^q-\boldsymbol{y}(\boldsymbol{\xi}^s)|}\left\{t_j^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s\right\} \\ &+ Q_i(\boldsymbol{x}^q)\sum_{k,l,s}\frac{1}{|\boldsymbol{x}^q-\boldsymbol{y}(\boldsymbol{\xi}^s)|}\left\{t_j^{kl}\phi^l(\boldsymbol{\xi}^s)y_j(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s\right\} = 0. \end{aligned} \tag{29}$$

Again, note that in (29) the terms dependent on $\boldsymbol{x}$ are separated from those dependent on $\boldsymbol{y}$ except for the harmonic kernel $|\boldsymbol{x}-\boldsymbol{y}|^{-1}$.

The discretized forms of (23) and (24) are similar to (28):

$$\begin{aligned}\frac{1}{2}u_i(\boldsymbol{x}^q) &+ \widehat{\sum_{k,l,s}} T_{ij}^*[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]u_j^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s \\ &- \widehat{\sum_{k,l,s}} U_{ij}^*[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]t_j^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s = g_i(\boldsymbol{x}^q)\,, \end{aligned} \tag{30}$$

and

$$\begin{aligned} g_i(\boldsymbol{x}^q) = -\frac{1}{2}\epsilon_{ij}^0 x_j^q &- \epsilon_{jp}^0\widehat{\sum_{k,l,s}} T_{ij}^*[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]y_p^{kl}\phi^l(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s \\ &+\sigma_{jp}^0\widehat{\sum_{k,l,s}} U_{ij}^*[\boldsymbol{x}^q,\boldsymbol{y}(\boldsymbol{\xi}^s)]n_p^{kl}(\boldsymbol{\xi}^s)J[\boldsymbol{y}(\boldsymbol{\xi}^s)]\omega^s\,. \end{aligned} \tag{31}$$

The wide hat implies that the sums involve the nodes and integration points that belong to a single particle.

At this stage, it is useful to rewrite (29) through (31) in a symbolic matrix form. To this end we form N-dimensional nodal vectors $\mathbf{u}$, $\mathbf{t}$, and $\mathbf{g}$ organized on a particle-by-particle basis, so that each vector is formed by M blocks. Consequently the operators acting on $\mathbf{u}$ and $\mathbf{t}$ are represented as $N \times N$ matrices formed by $M \times M$ blocks. Thus we rewrite (29) as

$$\mathcal{A}\mathbf{u} + \mathcal{B}\mathbf{t} = \mathbf{0} \tag{32}$$

and (30) and (31) as

$$\mathcal{A}^*\mathbf{u} + \mathcal{B}^*\mathbf{t} = \mathbf{g}\,. \tag{33}$$

Equations (32) and (33) can be combined into

$$\left(\mathcal{A} - \mathcal{B}\mathcal{B}^{*-1}\mathcal{A}^*\right)\mathbf{u} = -\mathcal{B}\mathcal{B}^{*-1}\mathbf{g} \tag{34}$$

or simply

$$\mathcal{C}\mathbf{u} = \mathbf{b}\,. \tag{35}$$

Also it is useful to rewrite (35) in the residual form

$$\mathbf{r}(\mathbf{u}) = \mathcal{C}\mathbf{u} - \mathbf{b} = 0. \tag{36}$$

The structure of the matrices in (32) and (33) reflects the physics of the problem. The matrices $\mathcal{A}$ and $\mathcal{B}$ are dense and their off-diagonal blocks represent pair-wise particle interactions while the diagonal blocks represent particle self-interactions. The off-diagonal blocks become negligible for well-separated particles, and their decay rates follow directly from (12) and (13). The matrices $\mathcal{A}^*$ and $\mathcal{B}^*$ are block-diagonal because they represent M uncoupled problems. For identical particles, the diagonal blocks in each matrix are the same.

4 Iterative Solution Strategy

In this section, we outline a procedure for solving (35) using $\mathcal{O}(N)$ memory and $\mathcal{O}(N)$ operations. This procedure consists of three components: (i) decomposition of $\mathbf{u}$ along

the residual Krylov's vectors $\mathbf{r}$, $\mathcal{C}\mathbf{r}$, $\mathcal{C}^2\mathbf{r}$, $\mathcal{C}^3\mathbf{r}$, etc., (ii) matrix-vector multiplication with the FMM, and (iii) preconditioning.

The solution vector $\mathbf{u}$ is obtained following the generalized minimum residual algorithm (GMRES) of Saad and Shultz (1986). This algorithm requires construction of the residual Krylov's vectors. The k-th Krylov vector $\mathbf{v}^{(k)}$ is constructed by induction:

$$\mathbf{v}^{(k)} = \mathcal{C}\mathbf{v}^{(k-1)} \quad \text{with} \quad \mathbf{v}^{(0)} = \mathbf{r}.$$

Following (34), this multiplication is performed as

$$\mathbf{v}^{(k)} = \mathcal{A}\mathbf{v}^{(k-1)} - \mathcal{B}\mathbf{w}^{(k-1)}, \tag{37}$$

with

$$\mathbf{w}^{(k-1)} = \mathcal{B}^{*-1}\mathcal{A}^*\mathbf{v}^{(k-1)}.$$

Observe that the term $\mathbf{w}^{(k-1)}$ can be computed in $\mathcal{O}(N)$ since both $\mathcal{A}^*$ and $\mathcal{B}^*$ are block-diagonal. Now comes the crucial step – application of the FMM to (37). To explain this step let us go back to (29) and replace there the nodal displacements with the nodal values of $\mathbf{v}^{(k-1)}$ and the nodal tractions with with the nodal values of $\mathbf{w}^{(k-1)}$. The four sums in (29) have the same structure and therefore it is sufficient to explain how to apply the FMM to the first one. To connect this sum with the generalized many-body electrostatics problem defined in Section 2, we identify the collocation points $\boldsymbol{x}^q$ with the target points, the integration points $\boldsymbol{y}(\boldsymbol{\xi}^s)$ with the source points, the expression in the curly brackets with the generalized charge $\mathcal{Q}[\boldsymbol{y}(\boldsymbol{\xi}^s)]$, the sum with the potential $\Phi(\boldsymbol{x}^q)$, and the operator $R(\boldsymbol{x}^q)$ with the generalized force operator $\mathcal{F}[\Phi(\boldsymbol{x}^q), \boldsymbol{x}^q]$. Observe that the generalized charge is a second rank tensor and therefore it has to be represented by nine ordinary charges. Thus the first sum is computed by invoking the FMM nine times. For the remaining sums, the FMM is invoked three times for the second and third sums, and once for the last sum, so that the total number of times the FMM is invoked is sixteen.

For large algebraic problems, like those corresponding to many-particle problems, preconditioning is important because it allows one to reduce the number of required iterations. Preconditioning involves multiplication of both sides of (35) by a matrix $\mathcal{P}$, the preconditioner, which is as close to $\mathcal{C}^{-1}$ as possible. We choose $\mathcal{P}$ to be block-diagonal, with each block being equal to the inverse of the corresponding diagonal block of $\mathcal{C}$. Formally, this preconditioner is known as block-Jacobi. For many-particle problems, the block-Jacobi preconditioner has a clear physical meaning – it exactly represents non-interacting particles. Therefore the matrix $\mathcal{PC}$ represents the inter-particle interactions only. Details of incorporating the preconditioner into the GMRES can be found in Saad and Shultz (1986).

5 Parallel Implementation

The goals for our parallel implementation of the FMM algorithm were efficiency and scalability and provision for mesh adaptivity in a second phase of algorithm and code refinement. These goals were attained by building the parallel code on the Scalable Dynamic Distributed Array (SDDA) data management infrastructure (Edwards and Browne, 1996; Edwards, 1997). The SDDA provides normal array semantics for any array of objects (where each may have a different structure) which is distributed across multiple processors and which may expand and contract with preservation of locality and without copying to follow adaptive mesh refinement. The SDDA was originally developed to support hp-adaptive finite element methods (Edwards and Browne, 1996; Edwards, 1997) for which it provides data management for the geometry data of the adapting finite element model. This requires support for efficient handling of dynamic arrays of small objects which is the fundamental data management problem for the parallel FMM implementation.

The implementation of the SDDA is based upon use of space-filling curves and sep-

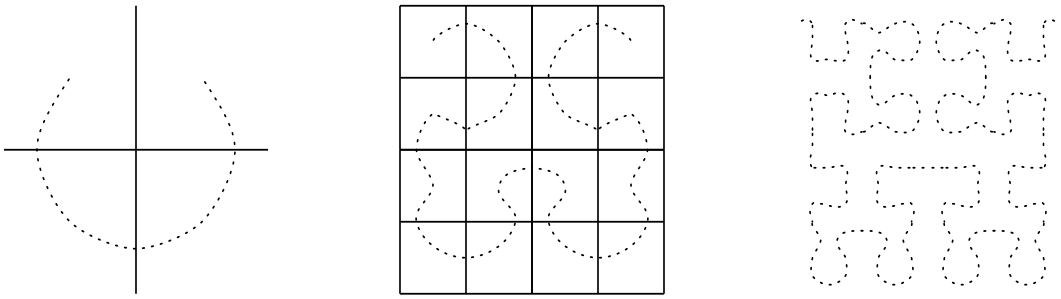

Figure 2: Partioning of the computational domain based on a space-filling curve.

aration of address management (indexing) from storage management (Parashar and Browne, 1997). Space-filling curves (Sagan, 1994) map the three-dimensional physical space to a one-dimensional space. Several authors have previously used space-filling curves for implementation of parallel tree codes (Salmon and Warren, 1992; Salmon and Warren, 1993; Singer, 1995). A simple Hilbert's space-filling curve is shown in Figure 2. It resembles a necklace with the curve as the carrier and the cells as the beads. Assignment of weights to the cells enables simple and efficient partitioning of the three-dimensional space with preservation of its geometrical localities in the one-dimensional space. The mapping to storage distributed across a parallel computer architecture and accessing of the data is accomplished through use of a hierarchical index space where the keys are derived from the positions of the cells on the space-filling curve. All partitioning, distribution, and communication are provided by the SDDA transparently to the user.

The current code does not implement mesh adaptivity but the SDDA provides the data management infrastructure for this refinement in a later phase of code development.

6 Test Problem

In this section, we describe application of FLEMS to a family of problems that involve identical spherical voids embedded in an infinite binder. In each problem, the spheres

form a simple cubic array in which the number of spheres is between $2^3 = 8$ and $7^3 = 343$ and the spacing between two nearest spheres is equal to 2.5 radii (Fig. 3). The principal objective in this section is to demonstrate that we can solve these problems quickly and accurately.

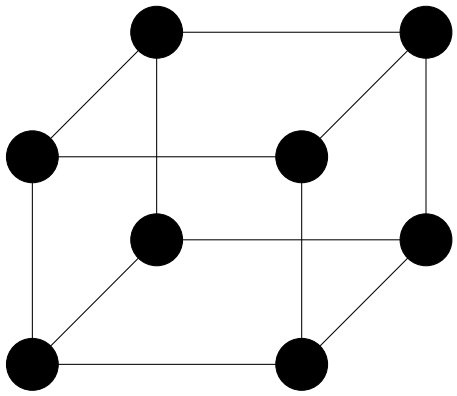

Figure 3: A simple cubic array of 8 spheres.

In our computations, we used the boundary element mesh shown in Figure 4. It allowed us to compute stresses near the surface of the voids to within 1% accuracy. We arrived at this conclusion by comparing the BEM solutions for two voids with detailed finite element solutions. The mesh is formed by 9-node isoparametric elements and the number of degrees of freedom per sphere is 1158. Within each element, integration is performed using 4×4 Gaussian and singular integration schemes (Le *et al.*, 1985).

Our version of the FMM code was developed by Singer (1995) who, to improve the performance of the method, represented the spherical harmonics with normalizations different from Greengard (1988). We used the FMM expansion depth $p = 10$ and the partitioning depth was chosen such that the number of charges in the smallest cells was close to is close to one hundred. This number was determined experimentally in order to balance the amount of computations involved in the sums in (3) (Singer, 1995; Greengard and Rokhlin, 1996).

First, we analyzed the geometries that can be solved on a desktop computer

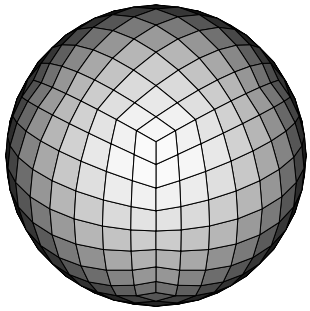

Figure 4: The surface discretization of a sphere.

(one 133 MHz processor of SP2 equipped with 128 Mb RAM). None of the problems reported here can be solved on this computer with solvers that require $\mathcal{O}(N^2)$ memory – the problem that involves two spheres only is very close to the memory capacity limit. In contrast, we were able to solve problems with up to 64 spheres and 74,112 unknowns using $\mathcal{O}(N)$ memory iterative solution strategies.

Computational results are summarized in Figure 5 which contains a log-log plot for the CPU time as a function of N. The cross symbols correspond to the $\mathcal{O}(N)$ iterative solution strategy that takes advantage of the FMM and the circular symbols to an $\mathcal{O}(N^2)$ strategy. The latter strategy was implemented by suppressing hierarchical partitioning and consequently eliminating the cell-cell interactions within the FMM. Thus the $\mathcal{O}(N^2)$ strategy is not the most efficient one in its class since it involves some of the FMM overhead. Nevertheless we estimate that, in the present computations, the performance of the $\mathcal{O}(N^2)$ strategy was within 20% from being optimal. The relative *a posteriori* l_2 error of the FMM matrix-vector multiplications was $\mathcal{O}(10^{-4})$. The number of performed matrix-vector multiplications was equal to 9 for all cases.

Each number of matrix-vector multiplications corresponds to a residual vector whose l_2 norm was 10^{-6} of the l_2 norm of the right-hand side vector. Results of these computations suggest that the break-even point is probably close to five thousand and that for $M = 64$ the $\mathcal{O}(N^2)$ strategy becomes impractical since it takes about ten days to complete the job. It appears that the $\mathcal{O}(N)$ strategy performs optimally between $M = 27$ and $M = 64$ and the case of $M = 8$ appears to be not large enough. In this regard, one should realize that in order to attain the $\mathcal{O}(N)$ performance, N should be large but not too large in order to avoid using the hard-disk memory (or paging), which may significantly deteriorate the performance. This issue becomes obvious in the next set of computations.

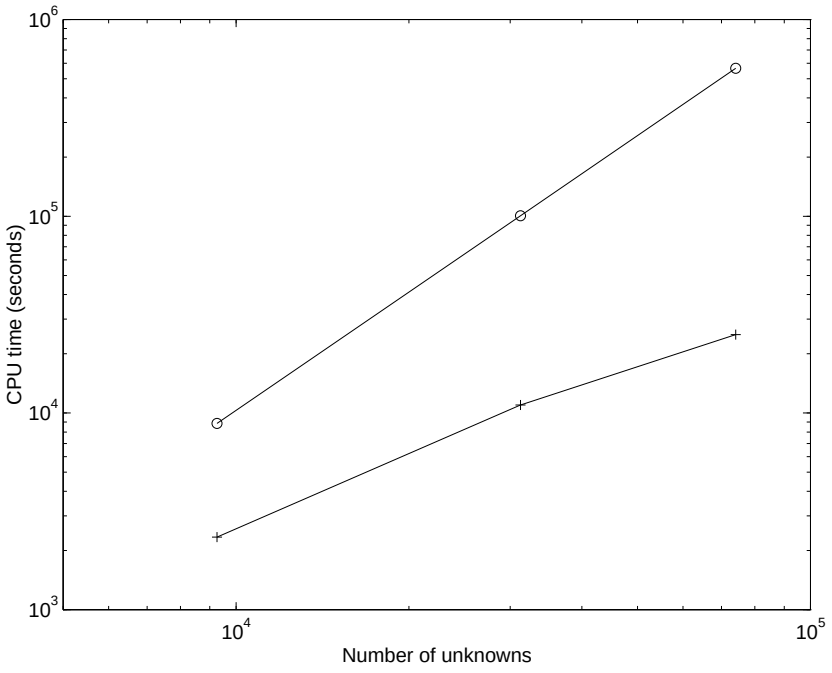

Figure 5: Computational results for 8, 27 and 64 spheres obtained with one processor of SP2. Crosses correspond to the $O(N)$ strategy and circles correspond to the $O(N^2)$ strategy.

The parallel version of the code was tested on an IBM SP2 computer with 16 133 MHz processors each equipped with 128 Mb RAM. The problems tested on this

computer involved the cases for 64, 125, 216 and 343 spheres and all computations were performed using the $\mathcal{O}(N)$ strategy only. Computational results are summarized in Figure 6 which contains a log-log plot for the CPU time per iteration as a function of N. Note that the performance is much better for the medium-size cases, $M = 125$ and $M = 216$, than for $M = 64$ and $M = 343$. For $M = 64$, the poor performance is due to the fact that the communication time constitutes a significant portion of the total time, i.e. the problem is not large enough. For $M = 343$, the poor performance is due to the fact that the processor that distributes the data had to resort to the hard-disk memory, i.e. the problem is too large for the 16 processors. We did not compute complete solutions for the considered problems only due to limited access to all 16 processors of the computer.

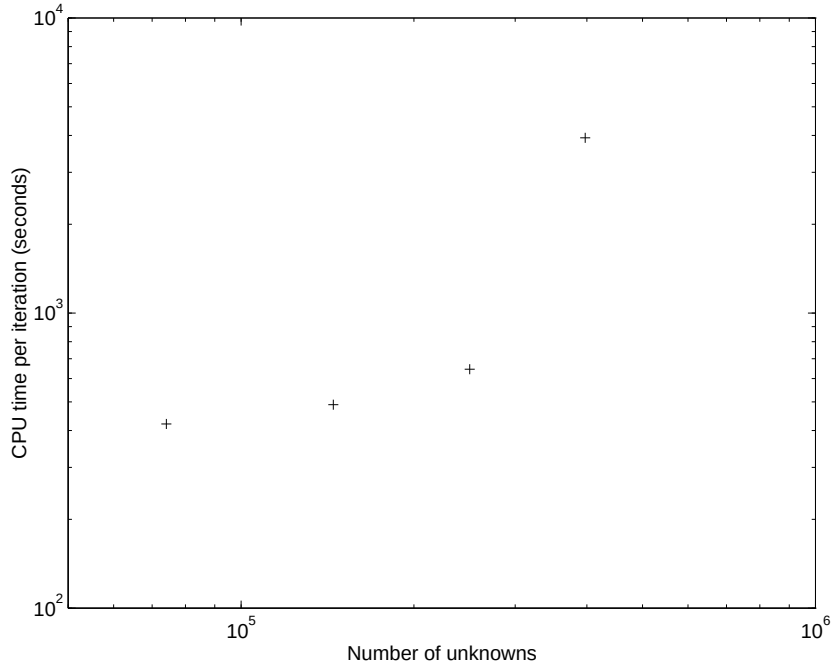

Figure 6: Computational results for the $O(N)$ strategy for 64, 125, 216 and 343 spheres obtained with 16 processors of SP2.

7 Concluding Remarks

In this paper, we introduced FLEMS, an $\mathcal{O}(N)$ method for analysis of large-scale three-dimensional micromechanical analysis of composite materials. This method is suitable for analyzing problems that involve hundreds and thousands of particles without imposing any essential restrictions on their geometry. The principal idea of FLEMS is to combine a conventional BEM with an iterative solution strategy that includes the FMM. We implemented this idea in a way that allows us to combine any BEM with any FMM and developed a parallel code that can be upgraded once better BEM or FMM formulations become available.

To a certain extent, this paper can be viewed as a first report – the work is in progress and several improvements to the code are currently under way. Those include incorporation of the improved FMM proposed by Greengard and Rokhlin (1996), modeling of more complex geometries, further parallelization of the code, development of local mesh refinement techniques, and preconditioning. Of course, the method needs to be tested on more complex geometries than those considered in this paper. Nevertheless, we believe that the fundamentals of FLEMS are sound and it will become a unique and useful tool of micromechanical analysis of composite materials.

Finally, we note that the generalized many-body electrostatics problem can be extended to the kernels that characterize long-range dislocation interactions (Rodin, 1997), so that the interactions of N dislocation segments can be computed in $\mathcal{O}(N)$ rather ordinarily $\mathcal{O}(N^2)$ operations. Originally, the FMM was applied to this class of problems by Wang and LeSar (1995).

Acknowledgment

This work was generously supported by the National Science Foundation through the

National Grand Challenges program grant ECS-9422707.

References

[1] S. Aluru, 'Greengard's N-body algorithm is not order N', SIAM J. Sci. Comput. **17**, 773-776 (1996).

[2] C. A. Brebbia and J. Dominguez, *Boundary Elements: An Introductory Course.* 2nd edn. Computational Mechanics Publications, McGraw-Hill Book Company, New York, 1989.

[3] H. C. Edwards, 'A parallel infrastructure for scalable adaptive finite element methods and its application to least squares C^∞ collocation', PhD Dissertation, Computational and Applied Mathematics, The University of Texas at Austin (1997).

[4] H. C. Edwards and J. C. Browne, 'The scalable distributed dynamic array and its application to a parallel hp-adaptive finite element code', *Proceedings of the parallel object-oriented methods and applications workshop*, Santa Fe, NM, 1996.

[5] A. Greenbaum, L. Greengard and A. Mayo, 'On the numerical solution of the biharmonic equation in the plane', *Physica D* **60**, 216-225 (1992).

[6] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, MA., 1988.

[7] L. Greengard, 'Fast algorithms for classical physics', *Science* **265**, 909-914 (1994).

[8] L. Greengard and V. Rokhlin, 'A fast algorithms for particle simulations', *J. Comput. Phys.* **73**, 325-348 (1987).

[9] L. Greengard and V. Rokhlin, 'A new version of the fast multipole method for the Laplace equation in three dimensions', Department of Computer Science, Yale University, Research Report YALEU/DCS/RR-1115 (1996).

[10] Z. Hashin, 'Analysis of composite materials', *J. Appl. Mech.* **50**, 481-505 (1983).

[11] H. B. Li, G. M. Han and H. A. Mang, 'A new method for evaluating singular integrals in stress analysis of solids by the direct boundary element method', *Int. J. Numer. Methods Engng.* **21**, 2071-2098 (1985).

[12] K. Nabors and F. T. Korsmeyer and F. T. Leighton and J. White, 'Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory', *SIAM, J. Sci. Stat. Comput.* **15**, 714-731 (1994).

[13] M. Parashar and J. C. Browne, 'System engineering for high performance computing software: the HDDA/DAGH infrastructure for implementation of parallel structured adaptive mesh refinement', to appear in *Structured Adaptive Mesh Refinement Grid Methods*, IMA volumes in Mathematics and its Applications, Springer Verlag, New York, 1997.

[14] A. P. Peirce and J. A. L. Napier, 'A spectral multipole method for efficient solution of large-scale boundary element models in elastostatics', *Int. J. Numer. Methods Engng.* **38**, 4009-4034 (1995).

[15] N. Phan-Thien and S. Kim, *Microstructures in Elastic Media: Principles and Computational Methods.* Oxford University Press, New York, 1994.

[16] G. J. Rodin, 'Toward rapid evaluation of the elastic interactions among three-dimensional dislocations', submitted to *Phil. Mag. Letters A* (1997).

[17] V. Rokhlin, 'Rapid solution of integral equations of classical potential theory', *J. Comput. Phys.* **60**, 187-207 (1985).

[18] Y. Saad and M. H. Schultz, 'GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM, J. Sci. Stat. Comput.* **7**, 856-869 (1986).

[19] H. Sagan, *Space Filling Curves.* Springer Verlag, New York, 1994.

[20] A. S. Sangani and G. Mo, 'An $O(N)$ algorithm for Stokes and Laplace interactions of particles' *Phys. Fluids* **8**, 1990-2010 (1996).

[21] M. S. Warren and J. K. Salmon, 'Astrophysical N-body simulations using hierarchical tree data structures', In: *Supercomputing*'92, Minneapolis, MN, IEEE Comput. Soc. Press, 570-576 (1992).

[22] M. S. Warren and J. K. Salmon, 'A parallel hashed oct-tree N-body algorithm', In: *Supercomputing*'93, Washington, DC, IEEE Comput. Soc. Press, 12-21 (1993).

[23] J. K. Singer, 'Parallel implementation of the fast multipole method with periodic boundary conditions', *East-West J. Numer. Math.* **3**, 199-216 (1995).

[24] Y. H. Wang and R. LeSar, '$O(N)$ algorithm for dislocation dynamics', *Phil. Mag. A* **71**, 149-163 (1995).

Figure Captions

Figure 1. A binder containing identical particles.

Figure 2. Partioning of the computational domain based on a space-filling curve.

Figure 3. A simple cubic array of 8 spheres.

Figure 4. The finite element discretization of the boundary of a sphere.

Figure 5. Computational results for 8, 27, and 64 spheres obtained with one processor of SP2. Circles correspond to the $\mathcal{O}(N)$ strategy and squares correspond to the $\mathcal{O}(N^2)$ strategy.

Figure 6. Computational results for the $\mathcal{O}(N)$ strategy for 64, 125, 216, and 343 spheres obtained with 16 processors of SP2.