# Multipole Fast Algorithm for the Least-Squares Approach of the Method of Fundamental Solutions for Three-dimensional Harmonic Problems

**I. Saavedra,**[1] **H. Power**[2]
[1]*Instituto de Mecánica de Fluidos, Universidad Central de Venezuela, Caracas 1041A, Venezuela*

[2]*Department of Mechanical Engineering, The University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom*

*Received 13 August 2002; accepted 15 March 2003*

*DOI 10.1002/num.10071*

In this article we describe an improvement in the speed of computation for the least-squares method of fundamental solutions (MFS) by means of Greengard and Rokhlin's FMA. Iterative solution of the linear system of equations is performed for the equations given by the least-squares formulation of the MFS. The results of applying the method to test problems from potential theory with a number of boundary points in the order of 80,000 show that the method can achieve fast solutions for the potential and its directional derivatives. The results show little loss of accuracy and a major reduction in the memory requirements compared to the direct solution method of the least squares problem with storage of the full MFS matrix. The method can be extended to the solution of overdetermined systems of equations arising from boundary integral methods with a large number of boundary integration points. © 2003 Wiley Periodicals, Inc. Numer Methods Partial Differential Eq 19: 828–845, 2003

*Keywords: multipole fast algorithm; 3D harmonic problem; method of fundamental solution*

## I. INTRODUCTION

The method of fundamental solution (MFS) is one of the most intuitive approaches to solve numerically linear elliptic partial differential equations. In this type of approach an approximate solution is represented in the form of a linear superposition of source functions (fundamental solutions) located outside the problem domain, $\Omega$. As the fundamental solution satisfies the differential equation at any point except at the source point, it follows that this representation

*Correspondence to:* Henry Power, School of Mechanical, Materials, Manufacturing Engineering, and Management, University of Nottingham, University Park, Nottingham NG7 2RD, UK (e-mail: henry.power@nottingham.ac.uk)

exactly satisfies the governing equation and one seeks to satisfy the imposed boundary conditions approximately at a set of boundary points (collocation points). In [1] a comprehensive review of the MFS is given, in which this approach is related to a regular indirect single layer boundary element approach (indirect-BEM). Some of the most relevant features discussed in [1] will be addressed in this section.

In the regular indirect single-layer BEM approach, the solution of the problem is represented in terms of a single layer potential, in which the unknown density is distributed over a auxiliary surface $\hat{S}$ which encloses the domain $\bar{\Omega}$, i.e., including the contour of $\Omega$ (the surface $S$). Since for such representation the kernel of the single-layer potential (the fundamental solution) is completely regular at points on the surface $S$, its follows that standard quadrature rules can be used to approximate the surface potential when it is evaluated at any boundary point on $S$, whatever interpolation function is used. Therefore, the numerical approximation of such single layer representation reduces to a linear superposition of fundamental solutions located outside the problem domain, i.e., the MFS.

One of the pioneer works on the MFS was the work by Kupradze and Aleksidze [2] where they prove the completeness of the potentials in $H(\Omega)$, the set of harmonics functions in $\Omega$. This work was later expanded upon by Bogomolny [3].

Although the MFS is useful for a large class of problems, the discussion in this article is restricted to the solution of the three-dimensional Laplace equation. In this type of approach, the solution of the Laplace interior problem is approximated by a linear combination of sources located outside the problem domain [4]. The coefficients of the fundamental solutions for given coordinates of the sources are determined in general by solving a system of equations that is obtained imposing the corresponding boundary conditions of the problem. This can be done by means of *direct collocation,* for which a system with a number of linear equations with an equal number unknowns is solved (see [5]). Alternatively, the boundary conditions can be satisfied *in the least-squares sense* with a number of sources smaller than the number of points where the boundary conditions are specified. This was the approach taken by Bogomolny in [3] and there is some numerical indication that this approach may be the most efficient [6]. In the present work the *least-squares* approach is used, focussing our attention to the solution of very large problems (in numerical analysis large problems means large number of degree of freedom).

A major problem encountered with the MFS is that for some problems as the number of sources increases the system of equations resulting form the scheme can become highly ill-conditioned [7–9]. However, it has been observed that despite the ill-conditioning, the accuracy of the numerical solution is largely unaffected [3, 7, 10]. Golberg and Chen [1] reported that in these cases direct solvers, as Gaussian elimination, can be used effectively, even if the resulting condition number of the matrix is very large, by ignoring the message of singular system of the numerical routine.

Traditionally in numerical analysis the stability of the scheme, i.e., the amplification of data errors, is measured by the condition number of the matrix (the ratio between the largest and the smallest singular values of the matrix). Typically ill-conditioning is indicated by very small values of the smallest singular value. It is known that in some cases only the consideration of the condition number will result in an under estimation of the stability. In [11] it is shown that a full understanding of the numerical stability requires knowledge of the complete singular values (SV) of the matrix and not only its largest and smaller values. In [8] and [9] Kitagawa showed that this is the case when using the MFS.

Convergence analysis of the MFS for the solution of the Laplace equation in two dimensions were considered by Cheng [12], Katsurada and Okamoto [13], and Katsurada [14]. In those articles it is shown that when $S$ and the boundary conditions are smooth functions, the MFS

converges exponentially to the solution of the partial differential problem. Although equivalent theoretical analysis are not available for the corresponding three-dimensional analogous problems, Golberg and Chen [7] presented an heuristic analysis, in which they suggest that similar convergent behavior could be expect for three-dimensional problems.

In the implementation of the MFS, the approximation of the geometry of the problem and the boundary conditions can always be improved by increasing the number of surface collocations points or/and the number of sources representing the solution. This leads to a major drawbacks of the methods due to the increasing in size of the resulting fully populated matrices. In order to overcome the difficulties in the handling of the huge dense matrices arising from the particle interactions, fast summation methods must be used. These methods use the fact that a collection of such particles viewed from a sufficiently large distance can be grouped in a single evaluation, the loss of accuracy in this approximation decays with the distance. Therefore the farther influences can be clustered in order to reduce the computational time of the so-called $N$-particles problem. $N$-particle problems pervade many different branches of numerical simulation. The naive approach involves an all-pairs computation and is $O(N^2)$ complexity. Although the exact computation of the pairwise interactions between all $N$ components of such a system is $O(N^2)$ in complexity, approximate solutions often may be computed with $O(N \log N)$ or $O(N)$ complexity. Those fast methods for the solution of the $N$-particles problem, have immediate implications on the boundary integral methods for the numerical solution of boundary value problems, as their discrete representation is given by summation of pairwise interaction between sources as well as in the direct collocation approach of the MFS. This is not necessarily the case when computing the least-squares approach of the MFS, as we will discuss in this article.

By using an expansion and grouping technique, it is possible to perform a fast evaluation of the matrix-vector products. In this approach it is not possible to have an explicit form of the matrix, and therefore the use of iterative solvers is required. A description of the application of this approach to obtain fast solution of the direct BEM collocation approach for the Laplace equation is given by Rokhlin [15], where the use of the generalized conjugate residual algorithm is suggested. By using the full implementation of Rokhlin's fast approach it is possible to reduces the computational cost of the BEM solution from $O(N^3)$ to $O(KN)$, $N$ being the number of degree of freedom and $K$ the number of iterations. Different formulations will lead to different preferred iterative methods of solution, and the use of GMRES [16] has been explored in recent applications of multipole schemes for a variety of problems using iterative solution of the linear systems of equations obtained from the BEM [17–19].

In the early 1980s several $O(N)$ and $O(N \log N)$ numerical algorithms were reported for computing the potential and force fields resulting from the gravitational or electrostatic interactions of $N$ particles ($N$-particle problem), for which the standard solution leads to a computational complexity of $O(N^2)$. These algorithms were based on the expansion of the potential field generated by $N$ sources in multipolar or Taylor series and grouping far-field influences. Among these works, Barnes and Hut's tree-codes scheme [20] and Greengard and Rokhlin's fast multipolar method [21] are worth special attention. Barnes and Hut did much of the early work with truncated multipole expansions. Their tree-code algorithm uses an oct-tree data structure to hierarchically subdivide the simulation domain into well-separated areas that can interact via the truncated expansions. Their method reduces the computational complexity of the problem from $O(N^2)$ to $O(N \log N)$. From another standpoint, Greengard and Rokhlin introduced the concept of local expansions to translate and sum the effects of multiple remote multipole expansions into a single local expansion. These local expansions combined with the oct-tree data structure provide a further reduction in the complexity of the evaluation to $O(N)$. This high-performance algorithm was achieved by expanding the kernel in terms of spherical harmonic series and using

the duality principle between the inner and outer expansions of harmonic functions. The group of Board and coworkers present parallel implementations of multipole-based $N$-particle algorithms for particle simulation, with application to molecular dynamics simulations [22–25]. These algorithms use multipole expansions and hierarchical decomposition to produce results with a known error boundary for a variety of parallel computing platforms.

Successfully implementation of efficient applications of the multipole algorithm to complex and computationally demanding practical problems has been recently reported in the literature, as is the case of the numerical simulation of concentrated emulsions of deformable drops (see Zinchenko and Davis [26]).

Although the fast multipole algorithm (FMA) is applicable to integration in the BEM, its application to the direct collocation approach of the MFS is more evident, because in such an approach the solution of the problem is approximated by a linear combination of $N$ sources located outside the problem domain [4], with an equal number $N$ of surface collocation points. Therefore, as in the $N$-particles problem, in this approach the solution reduces to the multiplication of a $N \times N$ matrix by a vector in each iteration.

Although the number of source points is smaller for the least-squares MFS, the operation count of iterative solvers is still $O(K \times M \times N)$, where $K$ is the number of iterations, $M$ is the number of boundary points, and $N$ is the number of sources, being usually $M \ll N$. Also, the storage capacity required is $O(M \times N)$. However, for very large problems both the operation count and storage capacity can becomes massive. Is for this reason that in this work we investigate how to accelerate the least-squares MFS and reduce the memory requirements to solve large three-dimensional problems by means of a multipole expansion method.

## II. SOLUTION OF HARMONIC PROBLEMS BY THE METHOD OF FUNDAMENTAL SOLUTIONS

In this section the MFS is described for interior boundary value problems of the Laplace equation. A domain $\Omega \in \Re^3$ is given, whose boundary $S$ is a simple closed surface. An unknown harmonic function $\Phi(P)$ must be found with prescribed boundary values on $S$, i.e., the following equations need to be satisfied:

$$\nabla^2 \Phi(P) = 0, \qquad P \in \Omega \tag{1}$$

$$\Phi(P) = f(P), \qquad P \in S_1 \tag{2}$$

$$\frac{\partial \Phi}{\partial n}(P) = g(P), \qquad P \in S_2, \tag{3}$$

where $S = S_1 \cup S_2$. Let $\Phi^*(x, \xi)$ be the fundamental solution of the Laplace equation

$$\Phi^*(P, Q) = \frac{1}{4\pi} \frac{1}{|P - Q|}. \tag{4}$$

A direct BEM gives an integral equation for $\partial\Phi/\partial n$, the unknown normal derivative on $S_1$, and for the unknown potential $\Phi$ on $S_2$. The integral equation is solved numerically and $\Phi$ can be evaluated in the interior of $\Omega$. The principal advantage of the boundary integral approach is that

there is no need to construct a mesh on $\Omega$. A main disadvantage is the computational expense to evaluate the required boundary integrals on $S$.

Alternative to the direct BEM, the indirect regular BEM approach represents the solution of an interior problem in terms of the single layer potential with unknown density $\sigma$:

$$\Phi(P) = \int_{\hat{S}} \Phi^*(P, Q)\sigma(Q)dS, \tag{5}$$

where $\hat{S}$ is the contour of a domain containing $\bar{\Omega}$.

Representing the unknown density $\sigma$, on $\hat{S}$, in terms of a set of interpolation functions $\varphi_j(Q)$, $j = 1, 2, \ldots, n$, i.e.,

$$\sigma(Q) = \sum_{j=1}^{n} c_j\varphi_j(Q); \qquad Q \in \hat{S}, \tag{6}$$

the above single layer can be approximated by

$$\Phi(P) = \sum_{j=1}^{n} c_j \int_{\hat{S}} \Phi^*(P, Q)\varphi_j(Q)dS. \tag{7}$$

In general, the integrals

$$\int_{\hat{S}} \Phi^*(P, Q)\varphi_j(Q)dS \qquad 1 \le j \le n \tag{8}$$

need to be evaluated numerically. Because in this approach the collocation points, $P$, are always different to the sources points, $Q$, where $P \in \bar{\Omega}$ and $Q \in \hat{S}$, the above integrals are nonsingular and therefore can be evaluated numerical by any standard quadrature rules with a weighting function $\varpi_l$, i.e.,

$$\int_{\hat{S}} \Phi^*(P, Q)\varphi_j(Q)dS \approx \sum_{l=1}^{N} \varpi_l\Phi^*(P, Q_l)\varphi_j(Q_l). \tag{9}$$

And therefore, the above single layer potential can be represented by

$$\hat{\Phi}(P) = \sum_{j=1}^{N} b_j\Phi^*(P, Q_j), \tag{10}$$

where

$$b_l = \varpi_l \sum_{j=1}^{n} c_j \varphi_j(Q_l). \tag{11}$$

The above superposition of fundamental solutions outside the problem domain, Eq. (10), is known as the MFS. This equation follows naturally from Eq. (5); however, usually it is necessary to add a constant term in particular in two dimensions, where it is required for completeness purposes:

$$\hat{\Phi}(P) = b_0 + \sum_{j=1}^{N} b_j \Phi^*(P, Q_j). \tag{12}$$

As can be observed a constant value is always a solution of the Laplace's equation.

From the above analysis, we can say that the MFS is a technique pertaining to the class of boundary elements methods, which is applicable when a fundamental solution is known for the partial differential equation for which a solution is desired (for more details see [1]). Therefore, the MFS is in a way similar to the method of boundary elements, with additional advantages with respect to the simplicity of its formulation and the fact that the distribution of points is practically mesh-free.

As the fundamental solutions satisfy the differential equation, $\hat{\Phi}(P)$ will automatically be harmonic. A good approximation to $\Phi(P)$ will then be obtained by imposing the boundary conditions at $M$ points $P_i$ on $S$. The expression of $\hat{\Phi}$ or its corresponding normal derivative $\partial\hat{\Phi}/\partial n$ are then equated to the values given by the boundary conditions. In this way, a system of $M$ linear equations with $N$ unknowns $b_j$ is obtained. Once the system of equations is solved, the potential and its directional derivatives at any point in $\bar{\Omega}$ can be computed by replacing the intensities $b_j$ in the expression for the potential $\hat{\Phi}(P)$ evaluated at the desired point.

After imposing the corresponding boundary conditions, the resulting system of linear equations can be expressed in matrix form as

$$\mathbf{Xb} = \mathbf{y}, \tag{13}$$

where $\mathbf{X}$ is the matrix of coefficients of the unknown source intensities $\mathbf{b}$, i.e., values of $\Phi^*(P_i, Q_j)$ and $\partial\Phi^*(P_i, Q_j)/\partial n$, and $\mathbf{y}$ is the right-hand-side known vector of potentials and normal derivatives at the boundary points. To obtain the estimate $b_0$ in the case of Dirichlet conditions a column vector $\mathbf{x}_0$ with its elements equal to 1 is added to the matrix $\mathbf{X}$. For the case of Neumann conditions, the corresponding elements in $\mathbf{x}_0$ are equal to 0. The direct collocation MFS corresponds to the case where $M = N$, whereas the overdetermined system with $M > N$, gives rise to a least-squares MFS.

## III. FORMULATION OF THE LEAST-SQUARES PROBLEM

Minimization of the euclidean error norm yields $N$ linear equations:

$$\sum_{i=1}^{M} \left[ y_i - \sum_{j=1}^{N} b_j x_{ij} \right] x_{ik} = 0; \qquad k = 1, \ldots, N, \tag{14}$$

which can be written as the matrix equation

$$\mathbf{Ab} = \mathbf{X}^T\mathbf{Xb} = \mathbf{X}^T\mathbf{y}, \tag{15}$$

where $\mathbf{X}$ is a matrix whose $M \times N$ components are constructed from $N$ basis functions evaluated at $M$ boundary points, $\mathbf{b}$ is a vector of length $N$ whose components are the source intensities, and $\mathbf{y}$ is a vector of length $M$ given by the values at the boundary points. These equations are called the *normal equations* of the least-squares problem with solution

$$\mathbf{b} = \mathbf{Cg}, \tag{16}$$

where $\mathbf{C} = \mathbf{A}^{-1}$ and $\mathbf{g} = \mathbf{X}^T\mathbf{y}$, with $N \leq M$, since there must be more boundary than source points to be solved.

One of the major problems on the application of the method of normal equations to obtain a least-squares approximation of an over-determined linear system of algebraic equations, as the one resulting from Eq. (15), is that its accuracy depends on the square of the condition number, which could be critical for the type of problem considered in this article (in particular when considering the position and number of source points as variables, since for such a case the resulting MFS matrix can be poorly conditioned). Alternatively, it is possible to use some of the scheme involving factorization or reflection of the matrix $\mathbf{X}$ to avoid the severe loss of accuracy due to the ill-conditioning of $\mathbf{X}^T\mathbf{X}$, at the expense of requiring about twice the number of arithmetic operations and much more computer storage. Of those alternative approaches, Golub [27] is likely to be one of the most accurate. Golub recognized that by application of Householder transformation the successive reduced matrices are computed with sufficient accuracy.

Substantial progress has been achieved in the development of numerical methods for least-squares problems over recent years. There has been a great increase in the capacity for automatic data capturing and computing. Least-squares problems of large size are now routinely solved. The solution for the vector of parameters can be obtained with or without explicitly computing the inverse matrix $\mathbf{C}$, by using a wide variety of routines [28] for direct solution methods such as LU decomposition and back-substitution or Gauss-Jordan elimination. Methods specifically designed for symmetric positive definite matrices such as Cholesky decomposition are usually preferred to solve the normal equations. However, the solution of a least-squares problem directly from the normal equations is rather susceptible to roundoff error. Alternative techniques such as QR decomposition of the design matrix $\mathbf{X}$ reduces the error since it does not involve explicit computation of the product $\mathbf{X}^T\mathbf{X}$.

As larger problems are considered and the matrix solution cost begins to dominate, some advantages can be obtained from the use of indirect solvers: the first is a possible improvement in solving time against direct methods provided that a good convergence is achieved, and the second resides in the huge storage requirements that can be avoided if the cost of computing the functions is less than the cost of storing the matrices, i.e. if explicit manipulation of the matrix can be avoided. Björck [29] examines several direct and iterative methods for least-squares problems. Among these, the conjugate gradient method (CG) has been widely applied to least-squares problems in FEM and BEM applications for some time, but its convergence depends on the expert selection of a good preconditioner. For ill-conditioned matrices LSQR [30] should be more reliable and stable than least squares adaptation of symmetric CG, at the expense of more storage and work per iterations.

Recently, the attractiveness of the GMRES (generalized minimal RESidual) algorithm [16] has been increasingly recognized in BEM, where the matrices that arise from surface discretization are fully populated and in general wholly unstructured, unlike those from finite element methods. With suitable modifications the GMRES can indeed be applied to large three-dimensional problems. The standard nonrestarted GMRES algorithm, also referred to as $GMRES_\infty$, calculate a new orthonormal basis vectors of the Krylov subspace for each iteration by using all previous orthonormal basis vectors and is guaranteed not to break down. However, a restarted version [GMRES(k)] is more convenient to reduce storage requirements. GMRES is particularly useful if the system matrix cannot be built explicitly, since it only involves its use through a product of the matrix by a vector.

In the fast algorithm developed in this work, we use the GMRES restarted indirect solver to find the solution of the system of algebraic Eqs. (15), and instead of storing a precomputed square matrix for the normal equations the product needed for each iteration is performed as two matrix-vector products, in which we used the appropriate version of the FMA to evaluate each of the products. Indeed this is the only way to obtain the product $\mathbf{X}^T\mathbf{X}\mathbf{b}$ when the matrix $\mathbf{X}$ is not stored explicitly, as it is required in the multipole-accelerated solution detailed in the following sections. Besides by performing the above two matrix-vector products instead of evaluating of the normal equation, even in the cases when using direct solvers without the FMA, it was possible to reduce the effect of roundoff error in the least-squares solution.

## IV. THE FAST MULTIPOLE ALGORITHM

The FMA of Greengard and Rokhlin [21] has been successfully used for efficiently computing the $N$-body problem (particle interactions) for large numbers of particles. The main feature of the FMA is that it reduces the operation count of the traditional $O(N^2)$ $N$-body problem to $O(N)$, while maintaining known accuracy bounded by rigorously derived error bounds. Many discrete-point applications have capitalized on the FMA's performance, especially in molecular dynamics and celestial mechanics. When this method is applied to continuum-mechanics as in the BEM numerical technique, the discretization procedure uses a representation where a set of points is chosen from the boundary and the integrals of the kernels in the Green's representation formula are expressed in terms of a sum of weighted singularities that are expanded by means of fast summations with the FMA. This operation is immediately interpreted in the MFS as the product of a vector of source intensities premultiplied by a matrix composed by the fundamental solutions as in the system of Eq. (13). In the present article we introduce the application of FMA to premultiplication by the transpose nonsquare matrix as in Eq. (15) for the solution of the least-squares MFS.

### A. FMA Basic Equations

The discussion and notation in this section follows closely the one given in [25]. The basic idea behind the FMA is simple. The potential at a certain point due to all interactions between sources can be divided into two components: one due to nearby sources that can be computed directly and one due to distant sources, given in the form of spherical harmonic expansions. The representations of the potential field exist in two series forms in the FMA. One form is known as the *multipole expansion,* obtained by application of the addition theorem for spherical harmonics, which converges for distant evaluation points:

$$\Phi(\mathbf{r}) = \sum_{n,m} \mathbf{M}_{n,m} \frac{\mathbf{Y}_n^m(\theta, \phi)}{r^{n+1}} \tag{17}$$

The other form is known as the *local expansion,* which converges within a cell:

$$\Phi(\mathbf{r}) = \sum_{n,m} \mathbf{L}_{n,m} r^n \mathbf{Y}_n^m(\theta, \phi), \tag{18}$$

where $\mathbf{M}_{n,m}$ and $\mathbf{L}_{n,m}$ are the corresponding expansion coefficients.

For each equation, multipole expansion coefficients ($M_{n,m}$), which are equivalent to multipole moments, or local expansion coefficients ($L_{n,m}$), determine the potential at an evaluation point ($r$, $\theta$, $\phi$). $Y_n^m(\theta, \phi)$ are spherical harmonics, based on Legendre polynomials. The multipole expansion series is summed over the defined range of the order $n$ ($0 \leq n \leq \infty$) and degree $m$ ($-n \leq m \leq n$) of the spherical harmonic function $Y_n^m(\theta, \phi)$. Although the mathematical equations employed in the FMA are exact for infinite series, its practical application uses truncated series, and therefore the overall accuracy of the algorithm must be controlled by setting parameters that include the level of spatial decomposition, the number of terms carried in the multipole expansions and the minimum separation between interacting cells.

### B. FMA Translations

In the FMA, there are several translations used to facilitate computing the potential field. By using these translations the coefficients of a cell within a larger cell (Children cells of the Parent cell) are shifted to the larger cell (Parent of the previous cells), the far-field potential at the center of a cell due to sources of another well-separated cell is taken from the multipole expansion of the latter and shifted to the local expansion, and small cells within a cell shift to their centers the local expansion of the larger cell. The derivation of the three series translations based mainly on spherical harmonic expansions is presented by Greengard [31]. In the equations below we follow the notation used in [25] to describe the translations, called the multipole-to-multipole (M2M) translation, multipole-to-local (M2L) translation, and local-to-local (L2L) translation. The result of each translation is an output coefficient matrix that is computed by weighted summation of the input matrix with a translation matrix $\mathbf{T}$ as described below.

#### *1. M2M.*

$$\mathbf{M}_{n'm'} = \sum_{n,m} \mathbf{T}^{M2M}_{n,m,n',m'}(\vec{\rho}) \mathbf{M}_{n,m} \tag{19}$$

$$\mathbf{T}^{M2M}_{n,m,n',m'}(\vec{\rho}) = \begin{cases} \dfrac{(-1)^{n'-n} A_n^m A_{n'-n}^{m'-m}}{A_{n'}^{m'}} Y_{n'-n}^{m'-m}(\alpha, \beta) \rho^{n'-n}, & \text{if } n \geq n' \text{ and} \\ & \text{if } |m - m'| \leq n - n' \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

#### *2. M2L.*

$$\mathbf{L}_{n',m'} = \sum_{n,m} \mathbf{T}^{M2L}_{n,m,n',m'}(\vec{\rho}) \mathbf{M}_{n,m} \tag{21}$$

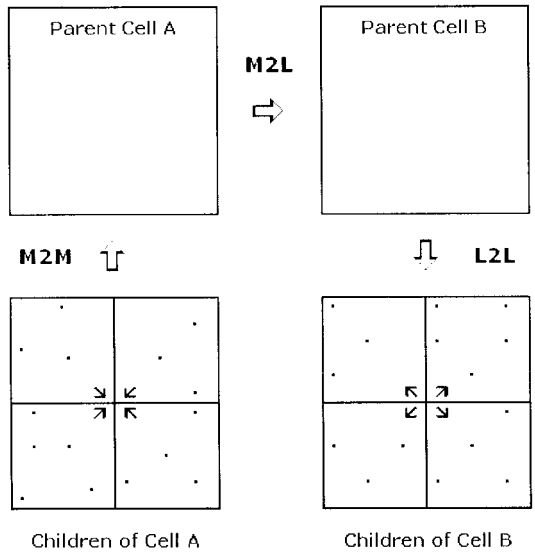


FIG. 1. Schematic representation of the FMA multipole expansion translations.

$$\mathbf{T}^{M2L}_{n,m,n',m'}(\vec{\rho}) = \begin{cases} \dfrac{(-1)^{n'+m} A_n^m A_{n'}^{m'} Y_{n'+n}^{m'-m}(\alpha, \beta)}{A_{n'+n}^{m'-m} \rho^{n'+n+1}}, & \text{if } n \geq n' \text{ and} \\ 0, & \text{if } |m - m'| \leq n - n' \\ & \text{otherwise.} \end{cases} \tag{22}$$

### 3. L2L.

$$\mathbf{L}_{n',m'} = \sum_{n,m} \mathbf{T}^{L2L}_{n,m,n',m'}(\vec{\rho}) \mathbf{L}_{n,m} \tag{23}$$

$$\mathbf{T}^{L2L}_{n,m,n',m'}(\vec{\rho}) = \begin{cases} \dfrac{A_{n'}^{m'} A_{n-n'}^{m-m'}}{A_n^m} \mathbf{Y}_{n-n'}^{m-m'}(\alpha, \beta) \rho^{n-n'}, & \text{if } n \geq n' \text{ and} \\ 0, & \text{if } |m - m'| \leq n - n' \\ & \text{otherwise,} \end{cases} \tag{24}$$

where

$$A_n^m = \frac{1}{\sqrt{(n+m)!(n-m)!}}.$$

The translation from a distant cell A to a local cell B simultaneously translates the origin $O$ of a truncated multipole expansion describing the potential field due to the sources in A to a new origin at point $(\rho, \alpha, \beta)$ at the center of B. In this way, the FMA replaces most of the particle-to-particle interactions with cell-to-cell interactions to improve the efficiency of calculations. The above three cell-to-cell translations (M2M, M2L, and L2L) are schematically represented in Fig. 1.

These input ($M_{n,m}$, $L_{n,m}$), output ($M_{n',m'}$, $L_{n',m'}$), and translation ($\mathbf{T}$) coefficient matrices are approximately half-filled, since the coefficients are zero outside the allowed range of $n$ and $m$. In [25], a weighting term is eliminated in addition to the standard normalization factor in order to make the computations with the multipole expansion equations more efficient. In addition, the translations are presented in a more concise form by separating each term in the series expansion into two factors: $\mathbf{F}_{n,m}(\vec{r})$, which includes powers of $r$, and $\mathbf{G}_{n,m}(\vec{\rho})$, which includes all partial derivatives of $1/\rho$.

### C. The FMA Expansion and Grouping Procedure

The grouping of the terms is accomplished by defining a tree of clusters. In a three-dimensional system of sources, the FMA starts by using an oct-tree for hierarchical spatial decomposition to divide the domain into cells. The grouping starts from the lowest clusters, usually called leafs, where sources are grouped, and then continued by forming the higher clusters where lower clusters are grouped. A truncated multipole expansion, which expresses the effect on distant points of all sources in a cell, is then calculated by using Eq. (17) for each cell at the finest level of spatial decomposition.

After spatial decomposition, starting with the multipole expansion coefficients at the leaf cells, the expansions are combined in a hierarchical fashion to represent the effects of larger and larger groups of sources in what is known as the *upward pass.* The child cells use the M2M translation in Eq. (19) to shift their multipole expansion to the center of the parent cell. In this way, the FMA computes the multipole expansion coefficients of all cells higher in the oct-tree, until similar multipole expansions are carried out for all cells at all levels.

As distant cells interact, each cell accumulates the far-field effects into its local expansion. In the *downward pass,* well-separated cells interact by creating a local expansion at the center of each cell using the M2L translation in Eq. (21) and the local expansions of each parent cell are used to transfer down its accumulative far-field interaction to each of its children cells using the L2L translation in Eq. (23). The procedure is made nonredundant by keeping an *interaction list* of cells, which is passed from parent to children and updated at each level.

When the grouping is finished, the influences of the whole far field that were transferred in the downward pass are distributed to the collocation points in each cluster by summation of the multiplication of the terms of the series and their adequate grouped moments. Finally, far-field effects are combined with direct short-field evaluations to yield the potential at each particle. For a small number of particles this is indeed more costly than computing the direct interactions. However, the computational cost increases linearly with $N$, and when the number of particles is large, this cost becomes much smaller than the $O(N^2)$ computational cost of the direct interaction.

### D. Well-Separated Cell Criterion

A solution to the $N$-particles problem must be configured to seek maximum efficiency with respect to runtime/accuracy trade-offs. Efficiency is improved when particles are grouped in such a way as to minimize runtime while remaining within some specified accuracy measurement. The well-separated cell interaction is carried out with the M2L translation, creating local expansions. To use the information in the far field, a *well-separatedness* criterion must be specified to enable decisions about whether the cells are far enough to interact as whole cells rather than its components. This criterion is derived from the case of worst relative error bounds and is a simple ratio that includes the geometries of the cells involved in the well-separated

interactions. In this way the following expression based only on the size and separation of the cells is obtained:

$$\frac{(r_A + r_B)}{R} \leq \mathbf{MAP}, \tag{25}$$

where $R$ is the distance between two cells involved in an M2L interaction, measured between cell centers, $r_A$ and $r_B$ are the radii of circumscribed spheres corresponding to the the two cells (for a cubic cell the radius is $\sqrt{3}/2$ times the length of the cube sides), and the **MAP** (multipole acceptance parameter) is a user-selected value between 0 and 1, which is used with other tunable parameters to optimize runtime/accuracy performance. By using a smaller **MAP** value, the accuracy is improved and for a fixed desired accuracy, the order of the expansions can be reduced at the cost of larger interaction lists. For more detail about the different criterion of well-separated cells, see [22].

### E. Application of the FMA to the Least-Squares MFS

The matrix-vector products $\mathbf{v} = \mathbf{Xb}$ and $\mathbf{u} = \mathbf{X}^T\mathbf{v}$ are needed at each iteration of the GMRES iterative solution in order to obtain the product in Eq. (15). Hereafter we will refer to these products as the *direct product* and the *transpose product,* respectively. The application of the FMA to obtain the direct product is straightforward and this is the only one needed for the direct collocation approach of the MFS. In this work we introduce the solution of the least-squares system, which requires the additional product by the matrix $\mathbf{X}^T$.

In the direct collocation approach, i.e., $N = M$, when applying the expansion and grouping procedure with the boundary and source points in two layers very close to each other in a fixed distribution, the construction of the oct-tree with all points together proceeds in a straightforward fashion. Although the *source points* are different from the *evaluation points,* this application differs little from the all-pairs computation, since most cells contain both kinds of points, and the speed of computation benefits from the multipole and local expansions at all cells. In this method the potentials at the boundary are forced to be satisfied exactly within multipole accuracy, but larger errors could arise in the estimation of the normal derivatives, especially when the distance between the source and evaluation layers is very small. From practical experience and some analytical works, in particular in two dimensions (see [12–14]), it has been observed that the accuracy of the approximation improves as the sources are moved farther away from $S$, but it is necessary to impose restrictions on the size of the surface containing the source points in order to prevent the system of MFS equations from becoming highly ill-conditioned.

Numerical experiments have shown that a least-squares formulation can allows more flexibility in the error balance over the boundary. In addition, it requires a smaller number of source points (see [6]). However, as stated in the introduction, these advantages can be outweighed by ill-conditioning in the normal system of equations. Therefore the least-squares formulation cannot be implemented in a simple way. In this work we introduce the *transpose product* required by the solution of the least-squares system and explain how we deal with the balance between speed and accuracy under the special conditions of the least-squares problem when the distance between the source points and boundary points is not small.

In the computation of the direct product for a Dirichlet interior problem, the boundary and source points are distributed in two layers. The external layer corresponds to the sources, and instead of computing the elements of the matrix as $x_{ij} = \Phi^*(P_i, Q_j)$, the potential due to the sources at the points $Q_j$ is computed by applying the upward and downward pass of the FMA

procedure. At the final step the influences of the far field from the sources that were transferred in the downward pass are distributed to the points in the leafs. These far-field effects are combined with direct near-field evaluations to yield the potential at each boundary point $P_i$. This procedure has in general little difference with that usually applied in the FMA fast evaluation of the direct BEM collocation approach. However, for the present least-squares application where the separation between the two layers is in most cases larger than the size of a leaf cell, the construction of the oct-tree with all points together needs an additional indicator for each cell in order to distinguish *source cells,* which possess nonzero multipole expansions, and *evaluation cells,* which need to accumulate local expansions. A cell in a level near to the root can be of both types, but leaf cells will normally be of one type or the other exclusively.

For the computation of the transpose product, the operation can be described in a very intuitive way when only the potential is given at all boundary points; this is a consequence of the symmetric property of the fundamental solution. Since the direct product $\mathbf{v} = \mathbf{Xb}$ is performed by computing the potentials $\mathbf{v}$ at the boundary points (rows of $\mathbf{X}$) due to the given intensities $\mathbf{b}$ at the specified source points (columns of $\mathbf{X}$), i.e., the evaluation of the potential at $M$ boundary points due to the distribution of $N$ sources of intensities $b_j$, $j = 1, 2, 3, \ldots, N$. The immediate consequence of reversing the direction of the influence of the sources is to exchange the columns by rows. In other word, the product $\mathbf{u} = \mathbf{X}^T\mathbf{v}$ is obtained for the Dirichlet problem by computing the potentials $\mathbf{u}$ at the source points (columns of $\mathbf{X}$) due to the given intensities $\mathbf{v}$ at the specified boundary points (rows of $\mathbf{X}$), i.e., the evaluation of the potential at $N$ points due to the distribution of $M$ sources with intensities $v_j$, $j = 1, 2, \ldots, M$. These matrix-vector products are thus implemented in a straightforward fashion, in which we have $M$ sources evaluated at $N$ points for $\mathbf{v} = \mathbf{Xb}$ and $N$ sources evaluated at $M$ point for $\mathbf{u} = \mathbf{X}^T\mathbf{v}$.

The computation of the transpose product for the case of the Neumann problem is somewhat more involved. The elements of the matrix $\mathbf{X}$ for this case are $x_{ij} = (\partial\Phi^*/\partial n)(P_i, Q_j)$, were $(\partial/\partial n) = (\partial/\partial P_k)n_k(P_i)$. Therefore, the product $\mathbf{v} = \mathbf{Xb}$ is obtained by evaluating the normal derivative at each of the $M$ boundary points due to the distribution of $N$ sources with intensities $b_j$, $j = 1, 2, \ldots, N$, requiring the computation of the three directional derivatives $\partial/\partial P_k$, $k = 1, 2, 3$, and the corresponding normal product. On the other hand, the product $\mathbf{u} = \mathbf{X}^T\mathbf{v}$ represents the evaluation of the potential at $N$ points due to the distribution of $M$ dipoles oriented in the direction of the normal vector at the $M$ boundary points, with intensities $v_j$, $j = 1, 2, \ldots, M$. As can be observed our FMA for a Neumann problem is more computational demanding than the Dirichlet one. Similar difficulties appears in the FMA of a BEM formulation when dealing with the normal derivative of a single layer potential and the evaluation of the potential due to a double layer, respectively.

## V. NUMERICAL RESULTS

In this section we explore the effects of problems size on the performance of our proposed multipole solution. In our analysis, we use as tunable parameters the order of the truncated series in the multipole expansion $p$ and the multipole acceptance parameter **MAP**. In the present implementation, rather than assuming an uniform cell size for the neighborhood and performing parental conversion to reduce the size of the interaction list, the interaction list for a cell is initially taken from that of its parent, and then the cells are divided only if they do not match the imposed criterion of well separatedness. In this way the accuracy can be improved by using a smaller **MAP** value.

A fixed desired accuracy can be achieved by increasing the order $p$ and reducing the *MAP* value simultaneously, at the cost of more direct interaction computations. Therefore, there is a minimum time depending on these two parameters for a given problem and desired accuracy. We performed a preliminary experiment with $N = M$ (direct collocation approach) and sources distributed on a sphere with a radius twice of that of the domain boundary. In general it was observed that for this configuration and within the usual ranges for the parameters, the mean square error is approximately proportional to $2.4^{-p}\mathbf{MAP}^{8.8}$. This relationship, together with estimates of the computational cost depending on both parameters, can be used as a rough guide to obtain a balance between accuracy and speed. However, since the accuracy of the approximation in the MFS improves as the sources are moved farther away, a lower **MAP** value may be preferred. In the results presented below we use $\mathbf{MAP} = 0.45$.

In order to test the performance of the proposed method of multipole summation for the least-squares approach of the MFA, we consider the harmonic problem inside a circular cylinder with a potential given in cylindrical coordinates as

$$\Phi = I_1(2r)\sin\theta\cos 2z, \tag{26}$$

where $I$ is the modified Bessel function of the first kind. This problem is solved in a cylinder with radius 1, and we vary the height $H$ for different problem sizes. Dirichlet conditions are imposed on all the boundaries.

In this section we compare the computational cost of our FMA solution, using the GMRES(k) solver with a fixed number of $k = 20$, with the solution of the direct normal equation, also using the same GMRE(k) solver, in which the previous two matrix-vector products were directly carried out. It is important to point out that for the problem sizes considered in the examples, it was not possible to find a direct solution of the normal equation using Cholesky decomposition or Gaussian elimination.

In principle we need only to define a distribution of the boundary points sufficient to describe the geometry and boundary conditions of the problem. We assume that the distribution of the boundary points is adequate, and we adjust the source distribution. For all the cases studied the total number of boundary points was given in terms of the height $H$ as

$$M = 768 + 900H.$$

The overall actual running times and memory usage for the numerical solutions obtained with the direct and multipole schemes are compared. Memory usage is counted for the stored matrix and the multipole coefficients, respectively, and for the iterative solver in both schemes. We report the actual running times, not including input-output operations, measured on a dedicated computer with a Pentium III processor at 1000 MHz. For the iterative solution in the multipole scheme we use $p = 8$, and a little additional error reduction is obtained adjusting the source intensities by a single coefficient given by a linear regression performed with a single multipole pass using $p = 12$.

An approximately uniform distribution of $N = 64H + 128$ sources on a curved surface which encloses the cylinder at a distance from the cylinder surface varying between 2 at the sides and $(H/4) + 3$ at the ends of the cylinder was used. For the used multipole parameters, the accuracy of the results are of the same order of magnitude of those of the direct scheme. The maximum relative error for both the potential and normal derivative was always smaller than $10^{-4}$; however, the number of iterations was in general smaller with the direct scheme.

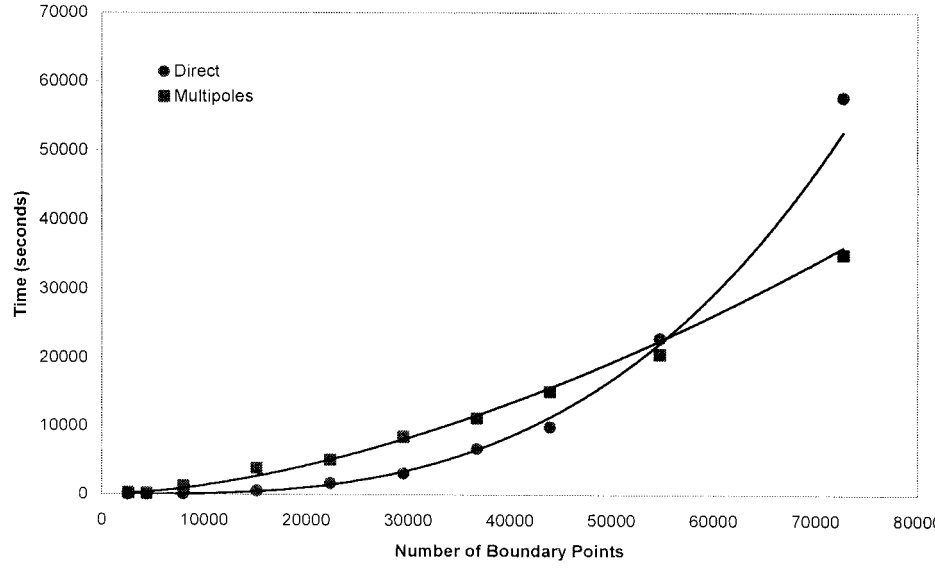


FIG. 2. Comparison of the execution times for direct normal and multipole methods.

Figure 2 shows the execution times of runs with values of $H$ ranging from 2 to 80, corresponding to a number of boundary points between $M = 2{,}568$ and $M = 72{,}768$, and sources points between $N = 256$ and $N = 5{,}248$. The computational time for the direct scheme was found to be smaller than the one corresponding to the FMA for cylinders smaller than $H = 60$, i.e., in the present case the computational advantage of using multipole expansion technique to carry out the matrix-vector products becomes evident for cylinders lagers than $H = 60$. However, since in the present case both $M$ and $N$ are linear in $H$ and because of the ill-conditioning of the normal matrix, the number of iterations was approximately proportional to $N$, it was found that the computational time for the direct scheme grows approximately as $N^{8/3}$, whereas the computational time for the multipole scheme was found to be in this case approximately proportional to $N^{5/3}$. The advantage in computational cost using the multipole scheme becomes evident above the point corresponding to a matrix size of $54{,}768 \times 3{,}968$. It is important to point out that no preconditioning was used in any of the two cases, since we never build the corresponding square matrix of the normal equation in order to avoid roundoff error due to the conditioned number of the normal equation. Therefore, a simple diagonal preconditioning and no computational costly was not easy to define.

Figure 3 shows that the memory used by the multipole scheme was always smaller than the one required by the direct approach.

When comparing the multipole algorithm in the least-squares approach for the direct product $\mathbf{v} = \mathbf{Xb}$ and the transpose product $\mathbf{u} = \mathbf{X}^T\mathbf{v}$, it was observed that the computation of the transpose product is more efficient since $N$ is much larger than $M$ and the number of or source cells in the interaction regions is very small compared to that of the direct product, since the boundary points are concentrated in a inner layer of smaller dimensions. For both products, the number of indexed stores for the resulting vector is less in the transpose product, so that the overhead in computing the transpose product, relative to that of the direct product only, is rather small, especially for processors that take significantly longer times in the operation of indexed stores.

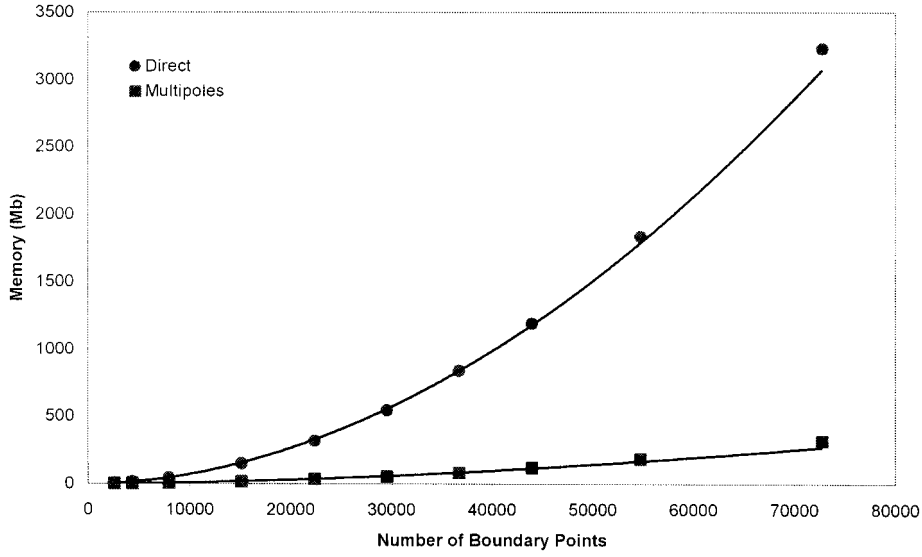


FIG. 3. Comparison of the memory usage by the direct normal and multipole methods.

For Neumann boundary conditions, the computational time required for the evaluation of the transpose product in the multipole scheme increases appreciably. Results of runs where potentials are imposed on the circular faces and normal derivatives on the curved cylinder walls show an increase in time with respect to the previous problems, which is about 10% for the direct normal scheme and 40% for the multipole scheme. Therefore, the multipole scheme can be less efficient for other problems rather than pure Dirichlet problems but it is still preferable in time for very lager number of sources. In the above mixed boundary value problem the FMA becomes more efficient than the direct approach for sources in numbers above 4000. In this article emphasis is made on the strategy as the specific selection of the multipole order, the well-separatedness threshold and the iteration termination criterion can vary between problems and specific machine and compiler implementations. The aspect of defining optimal combinations of these multipole and iterative solver parameters may deserve further study for particular applications.

## VI. CONCLUSIONS

The computation of the approximate solution to potential problems through the least-squares approach provides a appreciable reduction in the number of source points in relation to those required by the direct collocation scheme. In addition, a more reasonable distribution of the global error in the solution can be achieved. The solution with the iterative GMRES solver and the Fast Multipole Method makes this scheme a powerful tool for large three-dimensional potential problems.

The numerical experiments carried out in this work show that for the cases considered, the multipole scheme was in general more efficient, i.e., less computationally costly, for a number of sources above 4000 and number of boundary points about 10 times larger. It was observed that for smaller problems, although the computational time was longer for the multipoles scheme, the savings in memory were always significant.

Although this work shows the application of the FMA to the least-squares approach of the MFS, the extension of the proposed procedures to the least-squares solution of a BEM numerical formulation of boundary value problems is straightforward. Better performance than the one found is this work is expected for well posed second kind integral equations formulations, for which it is known that the number of iterations of an iterative solution to fixed precision is bounded independent of the number of surfaces nodes (for more details see Greengard et al. [32]).

The authors thank the reviewers of the journal for their helpful suggestions.

## References

1. M. A. Golberg and C. S. Chen, The method of fundamental solutions for potential, Helmholtz and diffusion problems, M. Goldberg, editor, Boundary integral methods numerical and mathematical aspects, WIT Press Computational Mechanics Publications, Southampton, 1999.
2. V. D. Kupradze and M. A. Aleksidze, The method of functional equations for the approximate solution of certain boundary value problems, U.S.S.R. Computat Math Math Phys 4 (1964), 82–126.
3. A. Bogomolny, Fundamental solutions method for elliptic boundary value problems, SIAM J Numer Anal 22 (1985), 644–669.
4. R. L. Johnston and G. Fairweather, The method of fundamental solutions for problems in potential flow, Appl Math Modelling 8 (1984), 265–270.
5. M. Katsurada and H. Okamoto, The collocation points of the fundamental solution method for the potential problem, Comput Math Appl 31(1) (1996), 123–137.
6. P. S. Han, M. D. Olsen, and R. L. Johnson, An adaptive boundary element method, Intl J Numer Methods Eng 24 (1987), 1187–1202.
7. M. A. Golberg and C. S. Chen, Discrete projection methods for integral equations, Computational Mechanics Publications, Southampton, 1996.
8. T. Kitagawa, On the numerical stability of the method of fundamental solutions applied to Dirichlet problem, Jpn J Appl Math 35 (1988), 507–518.
9. T. Kitagawa, Asymptotic stability of the fundamental solution method, J Comput Appl Math 38 (1991), 263–269.
10. M. A. Golberg, C. S. Chen, and S. R. Karur, Improved multiquadric approximation for partial differential equations, Eng Anal Boundary Elements 18 (1996), 9–17.
11. T. F. Chan and D. Foulser, Effectively well-conditioned linear systems, SIAM J Sci Stat Comput 9 (1988), 963–969.
12. R. S. C. Cheng, Delta-trigonometric and spline methods using the single layer potential representation, Ph.D. dissertation, University of Maryland, 1987.
13. M. Katsurada and H. Okamoto, A mathematical study of the charge simulation method, Journal of the Faculty of Science, University of Tokyo, Section 1A 35 (1988), 507–518.
14. M. Katsurada, Asymptotic error analysis of the charge simulation method in a Jordan region with an analytic boundary, Journal of the Faculty of Science, University of Tokyo, Section 1A 37 (1990), 635–657.
15. V. Rokhlin, Rapid solution of integral equations of classical potential theory, J Comput Phys 60 (1985), 187–207.
16. Y. Saad and M. H. Shultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J Sci Stat Comput 7(3) (1986), 856–869.
17. J. E. Gomez and H. Power, A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number, Eng Anal Boundary Elements 19 (1997), 17–31.

18. A. A. Mammoli and M. S. Ingber, Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods', Intl J Numer Methods Eng 44(7) (1999), 897–917.

19. V. Popov and H. Power, An O(N) Taylor series multipole boundary element method for three-dimensional elasticity problems, Eng Anal Boundary Elements 25 (2001), 7–18.

20. J. E. Barnes and P. Hut, A hierarchical O(N Log N) force calculation algorithm, Nature 324(4) (1986), 446–449.

21. L. Greengard and V. Rokhlin, A fast algorithm for particle simulation, J Comput Phys 73 (1987), 325–348.

22. J. F. Leathrum and J. A. Board, The parallel fast multipole algorithm in three dimensions, Technical Report 92-001, Duke University Department of Electrical Engineering, 1992.

23. J. A. Board, Z. S. Hakura, W. S. Elliot, and W. T. Rankin, Scalable variants of multipole-accelerated algorithms for molecular dynamics, Technical Report 94-006, Duke University Dept of Electrical Engineering, 1994.

24. J. A. Board and W. S. Elliot, Fast fourier transform accelerated fast multipole algorithm, Technical Report 94-001, Duke University Dept of Electrical Engineering, 1994.

25. W. D. Elliott, Multipole algorithms for molecular dynamics simulation on high performance computers, Technical Report 95-003, Duke University Department of Electrical Engineering, 1995.

26. A. Z. Zinchenko and R. H. Davis, Shear flow of highly concentrated emulsions of deformable drops by numerical simulations, J Fluid Mech 455 (2002), 21–62.

27. G. H. Golub and C. F. Van Loan, Matrix computations, 3rd ed., Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, 1996.

28. W. H. Press, S. A. Teukolsky, and W. T. Vetterling, Numerical recipes in Fortran, Cambridge University Press, Cambridge, Chap. 2, 15, 1992.

29. Å. Björck, Numerical methods for least squares problems, SIAM Society for Industrial & Applied Mathematics, 1996.

30. C. Paige and M. Saunders, LSQR, an algorithm for sparse linear equations and sparse least squares, ACM Transactions on Mathematical Software 8(1) (1982), 43–71.

31. L. Greengard, The rapid evaluation of potential fields in particle systems, MIT Press, Cambridge, MA, 1988.

32. L. Greengard, M. C. Kropiniski, and A. Mayo, Integral methods for Stokes flow and isotropic elasticity in the plane, J Comput Phys 125 (1996), 403–414.