

Research Article

The FMM-BEM Method for the 3D Particulate Stokes Flow

Hassib Selmi,^{1,2} Mohamed Abdelwahed,³ Hatem Hamda,² and Lassaad El Asmi²

¹ Technical and Vocational Training Corporation, Riyadh 11523, Saudi Arabia

² Laboratory of Engineering Mathematics, Polytechnic School of Tunisia, 2078 La Marsa, Tunisia

³ Department of Mathematics, College of Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Correspondence should be addressed to Mohamed Abdelwahed; mabdelwahed@ksu.edu.sa

Received 8 April 2014; Accepted 13 May 2014; Published 3 June 2014

Academic Editor: Hassan Eltayeb

Copyright © 2014 Hassib Selmi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work introduces new functions based on the spherical harmonics and the solid harmonics which have been used to construct a multipole development for the 3D Stokes problem in order to reduce the operations costs in the BEM method. We show that the major properties of those functions are inherited from the solid harmonics. The contribution of this paper is the introduction of new formulas that serve to calculate the multipole moments and the transfer functions that are necessary for the schemes of order $O(N \log N)$. Moreover, new translation formulas are introduced to obtain an $O(N)$ scheme. The error truncation of the resulting scheme is discussed. In comparison to the BEM that attains a limit storage at $O(10^4)$, we present here a method based on FMM-BEM that attains a storage at a limit of $O(10^6)$. The implementation of the method achieves a high accuracy level at a reasonable cost.

1. Introduction

The studies of complex fluids, such as particulate suspensions, emulsions, and sedimentation problem, remain a great challenge in spite of their omnipresence in many physical, chemical, or biomedical processes and industrial applications. This problem represents physically the creeping flow around several solid particles moving in a viscous fluid. Due to the size of the particles, this kind of problems is modelled by the linear Stokes equation. This linearity allows the application of the surface integral equation method or the boundary element method (BEM) which may be more efficient than the boundary value methods such as the finite element method (FEM) or the finite difference method (FDM). Indeed, the advantage of the BEM is that only the surface is discretized, generating fewer elements than a volume discretization and making the system matrix much smaller. However, the integral equation method is a global approach; every point is affected by the points of the entire system, giving a dense matrix. Since there are efficient algorithms, that is, GMRES, for solving the sparse matrix generated by FEM and FDM, it was believed that BEM was more expensive to use than

FEM or FDM. The development of the fast multipole method (FMM) with the aim of accelerating the product matrix vector and reducing the complexity of such operations from $O(N^2)$ to $O(N \log N)$ or even to $O(N)$, and the memory requirements to $O(N \log N)$ or $O(N)$ [1–3], brought new life to the classical BEM. Indeed such method approximates the effect of the far field points of the entire system. Like in FEM, only the terms in a certain band of the matrix are stored in an exact way and all the others are approximated and compressed.

We introduce in this paper new multipole moments and transfer functions for $O(N \log N)$ scheme and new formulas for the translations from multipole to multipole, multipole to local, and local to local that are necessary to construct the $O(N)$ scheme for solving the 3D particulate Stokes problem. The multipole and local expansions can be useful to develop an efficient method based on the FMM to determine a real time solution of the dynamics of rigid bodies in a 3D infinite or semi-infinite viscous fluid. This solution is obtained by determining the stresses exerted by the fluid on each body and then the velocities of the bodies. These stresses are represented by Stokeslets distributed over the surfaces of

the bodies. We begin in the next section by presenting a brief overview of the literature. In Section 3, we present the principle of the fast multipole method. Section 4 presents the two representations used in FMM formulas for the Laplace problem: the first is based on spherical harmonic [2] and the second on the solid harmonic [4]. We present also in this section the new functions that will be used to represent differently the Green function of the Laplace equation. For the sake of conciseness, clarity, and complexity of the algorithm, we choose to work with this new representation with the aim of developing the FMM formulas that permit solving the Stokes problem in an $O(N \log N)$ scheme in the first step and in $O(N)$ scheme in the final step. In Section 5, we present a detailed description of the FMM formulas founded to be applied to the Green function of the Stokes equation. We also present a discrete algorithm as well as the study of the error bound. Finally, in Section 6, we present, discuss, and interpret the numerical results obtained from a cluster.

2. Overview

FMM reduces the cost of the product matrix vector from $O(N^2)$ to $O(N \log N)$ or even $O(N)$ by approximating the calculation by handling numerical series. It uses a hierarchical subdivision of space into panels or clusters of sources calculating the multipolar moment that are used in the evaluation of the far field expansion, and it reduces the space of storage from $O(N^2)$ for the dense matrix of size N to $O(N \log N)$ or even to $O(N)$. From the algorithmic point of view, the FMM works down and up the TREE constructed by the recursive subdivision of the initial domain. FMM was initially introduced by Rokhlin [5] as a fast solution method for integral equations for two-dimensional Laplace equation. Barnes and Hut [6] developed an $O(N \log N)$ algorithm. The FMM was then developed by Greengard and Rokhlin [1–3], as a fast evaluation method for the pairwise force calculation in multibody problems with Coulombic potential. FMM has been applied to problems in various fields such as boundary integral equation method (BIEM) and molecular dynamics (MD). Rokhlin [7] and Coifman et al. [8] used the FMM to solve the Helmholtz equation in 2D. Greengard et al. [9] and Mammoli and Ingber [10, 11] applied the FMM to the Stokes equations in fluid mechanic and Greengard and Helsing [12] for the problem of elasticity. Nishimura et al. [13, 14] studied this topic in order to apply fast multipole boundary integral equation method (FM-BIEM) to practical problems in fracture mechanics and earthquake engineering. Gumerov and Duraiswami used the FMM to solve the biharmonic equation [15]. In this paper, we discuss the applications of the FMM-BIEM to the fundamental boundary value problems in three dimensions given by Stokes equations and show its efficiencies.

3. The FMM Principles for the 3D Stokes Problem

The FMM method replaces the classical product matrix vector by an operation called the multipole product that allows

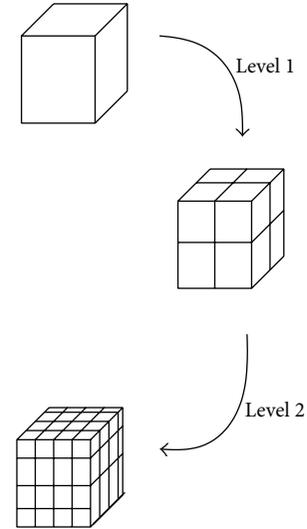


FIGURE 1: Recursive subdivision of the initial cube.

obtaining, by a very fast algorithm, a good approximation of the solution. The complexity is of $O(N \log N)$ or $O(N)$ in place of $O(N^2)$ for the direct calculus. The basic idea of the multipole method is the separation of the variables in the Green function. This Green function is rewritten differently in a new expression that determines the form of the moments and the transfer functions. To develop the FMM method to the kind of problems of our interest, we will consider the integral equation of the problem to be solved for the unknown values of the intensities of Stokeslets (see Pozrikidis [16]) located on the surfaces of the particles and will consider that this equation has been already discretized by the boundary element method. The problem is then supposed to be reduced to a linear system, of order N , of the unknown values of stress at the collocation points of the particles surfaces. N is related to the number N_p of the particles by $N = 3N_p N_T$, where N_T is the number of the triangles per particle. The linear system is solved by an iterative method. At each iteration, we start from a great imaginary cube containing the whole collocation points. This cube constitutes level zero of the subdivision. The n th subdivision is obtained from the $(n-1)$ th one by subdividing each cube into 8 small cubes (cf. Figure 1).

The level of the constructed tree is chosen after fixing a desired accuracy of computation ε . The set of collocation points is then distributed into several subdomains. For each one a moment centered at the middle of the subdomain is calculated as well as the transfer function for the far field interaction. To evaluate the integral equation at a given point x using the $O(N \log N)$ scheme, we have to work in a tree given by the recursive subdivision of the cubic initial domain and evaluate the far field interaction when necessary. The near interactions are evaluated at the finest level. If we have to use the $O(N)$, we must translate the moment of each box at each level to the far field box and convert the information of the far field subdomain at a local moment centered at the middle of the box. Beginning at the coarsest level, these local moments are shifted to the children level (cf. Figure 2).

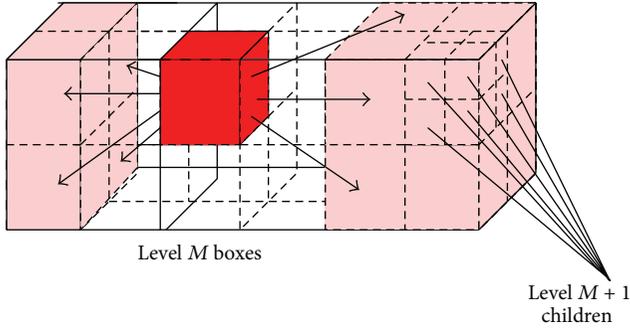


FIGURE 2: Transmission of the far field expansion to the boxes in its interaction list and translation to the children.

To evaluate the integral at a given point x , we go directly to the box containing this point at the finest level and calculate the far field expansion due to all points outside the neighbors boxes. The near interactions are evaluated directly in a similar manner as in $O(N \log N)$ scheme.

4. The Harmonic Functions and Laplace Equation

It is well known that the fundamental solution of the Laplace equation referred to as the Green function of the Laplacian operator can be written using the spherical harmonic Y_n^m [2]. The potential field due to a source point q located at $y(\rho, \alpha, \beta)$ and calculated at a point $x(r, \theta, \phi)$ is given by

$$\Phi(x, y) = \frac{1}{|x - y|} = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} Y_n^m(\theta, \phi), \quad (1)$$

where

$$M_n^m = q \rho^n Y_n^{-m}(\alpha, \beta), \quad (2)$$

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n - |m|)!}{(n + |m|)!}} P_n^m(\cos(\theta)) e^{im\phi}. \quad (3)$$

This potential can be written differently using the solid harmonics R_n^m and S_n^m [4]:

$$\begin{aligned} \Phi(x, y) &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{R_n^m(\vec{\sigma}y)} S_n^m(\vec{\sigma}x) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n R_n^m(\vec{\sigma}y) \overline{S_n^m(\vec{\sigma}x)}. \end{aligned} \quad (4)$$

In this paper, we will use new functions that we define from spherical harmonic as follows:

$$S_y_n^m = \alpha_n^m \frac{1}{r^{n+1}} Y_n^m(\theta, \phi), \quad R_y_n^m = \frac{1}{\alpha_n^m} \rho^n Y_n^m(\alpha, \beta), \quad (5)$$

where

$$\alpha_n^m = \sqrt{(n + |m|)! (n - |m|)!}. \quad (6)$$

From (3) and (5), we obtain the following expression for the functions introduced above:

$$\begin{aligned} S_y_n^m(\vec{\sigma}x) &= (n - |m|)! P_n^{|m|}(\cos \theta) e^{im\theta} \frac{1}{r^{n+1}}, \\ R_y_n^m(\vec{\sigma}y) &= \frac{1}{(n + |m|)!} P_n^{|m|}(\cos \alpha) e^{im\beta} \rho^n. \end{aligned} \quad (7)$$

Here, the special functions P_n^m are the associated Legendre functions, defined in [17] as

$$P_n^m(x) = (-1)^m (1 - x^2)^{m/2} \frac{d^m}{dx^m} P_n(x), \quad (8)$$

where $P_n(x)$ denotes the Legendre polynomial of degree n , defined by Rodrigues formula [17].

The first function coincides with solid harmonic S_n^m for $m \geq 0$, and the second one coincides with solid harmonic S_n^m for $m \geq 0$.

For the negative values of m , we have the following propriety:

$$\begin{aligned} S_n^{-m}(\vec{\sigma}x) &= (-1)^m \overline{S_n^m(\vec{\sigma}x)}; \\ R_n^{-m}(\vec{\sigma}x) &= (-1)^m \overline{R_n^m(\vec{\sigma}x)}; \\ S_y_n^{-m}(\vec{\sigma}x) &= \overline{S_y_n^m(\vec{\sigma}x)}; \\ R_y_n^{-m}(\vec{\sigma}x) &= \overline{R_y_n^m(\vec{\sigma}x)} \\ S_y_n^{-m}(\vec{\sigma}x) &= (-1)^m S_n^{-m}(\vec{\sigma}x); \\ R_y_n^{-m}(\vec{\sigma}y) &= (-1)^m R_n^{-m}(\vec{\sigma}y); \\ & \quad m > 0. \end{aligned} \quad (9)$$

4.1. Representation of the Laplacian Kernel. Like in (1) we have the following new representation of the Green function of the Laplace equation:

$$\begin{aligned} \Phi(x, y) &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{R_y_n^m(\vec{\sigma}y)} S_y_n^m(\vec{\sigma}x) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n R_y_n^m(\vec{\sigma}y) \overline{S_y_n^m(\vec{\sigma}x)}. \end{aligned} \quad (10)$$

Let us suppose that the point y is located inside a sphere D of radius a with center o and x is outside the sphere D . If we approach Φ by a finite sum obtained from (10) at the order p , we obtain an error bound which is the same error obtained by truncating the sums in (1) and (4):

$$E_p = \frac{1}{r - a} \left(\frac{a}{r} \right)^{p+1}. \quad (11)$$

The major advantage of this new representation is that we only have to work with positive index m , which decreases the complexity of the algorithms in a remarkable way.

5. The Schemes for 3D Stokes Problem

We will develop here the two numerical schemes of the $N \log(N)$ and the N order multipole expansion associated with the problem of several solid arbitrarily shaped particles moving in a creeping flow, which are based on a suitable translation formulas of Sy_n^m and Ry_n^m .

The fundamental solution or the Green function of the free-space Stokes problem [18] is

$$G_k^j(x, y) = -\frac{1}{8\pi\nu} \left[\frac{\delta_k^j}{|x-y|} + \frac{(x_j - y_j)(x_k - y_k)}{|x-y|^3} \right]. \quad (12)$$

Except for the multiplicative constant, the Green function may be written as follows:

$$G_k^j(x, y) = \left[\delta_k^j \frac{1}{|x-y|} - (\vec{o}\vec{x})_k \frac{\partial}{\partial x_j} \left(\frac{1}{|x-y|} \right) + (\vec{o}\vec{y})_k \frac{\partial}{\partial x_j} \left(\frac{1}{|x-y|} \right) \right], \quad (13)$$

and the integral equation for the velocity on a surface S_l of a particle l is

$$v_k(x) = \int_{S_l} G_k^j(x, y) q_j(y) ds; \quad x \in S_l, \quad (14)$$

where $S = \cup_{i=1}^{N_p} S_i$, N_p is the number of particles, S_i is the surface of the i th particle, and $q_j(y)$ is j th component of the surface stress.

The discrete problem associated with (14) is presented as follows:

$$v_k(x_G^{l,s'}) = \sum_{l=1}^{N_p} \sum_{s=1}^{N_t} \left(\int_{T_s^l} G_k^j(x_G^{l,s'}, y) ds(y) \right) q_j^{s,l}, \quad (15)$$

where T_s^l is the surface element, $x_G^{l,s'}$ is the gravity center of a given surface, N_p and N_t denote, respectively, the number of particle and surface element per particle, and $q_j^{s,l}$ is the stress applied on the surface element. Since we are interested here in the far field calculus, the treatment of the singularity problem will not be considered.

5.1. $O(N \log N)$ Scheme. Let us suppose that we have uniform distribution points on a certain surface S located inside a sphere D of radius a with center o . From (10) and (13), the velocity at a point x outside the sphere D distant of r from o is given by the following equation:

$$\begin{aligned} v_k(x) &= \int_S G_k^j(x, y) q_j(y) ds \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \left[\overline{Fy_{nm}^{jk}(\vec{o}\vec{x})} M_{n,j}^{m(1)}(o) + \overline{Hy_{nm}^j(\vec{o}\vec{x})} M_{n,j}^{m,k(2)}(o) \right], \end{aligned} \quad (16)$$

where

$$\begin{aligned} M_{n,j}^{m(1)}(o) &= \int_S Ry_n^m(\vec{o}\vec{y}) q_j(y) ds, \\ M_{n,j}^{m,k(2)}(o) &= \int_S (\vec{o}\vec{y})_k Ry_n^m(\vec{o}\vec{y}) q_j(y) ds, \\ Fy_{nm}^{jk}(\vec{o}\vec{x}) &= \delta_k^j Sy_n^m(\vec{o}\vec{x}) - (\vec{o}\vec{x})_k \frac{\partial}{\partial x_j} (Sy_n^m(\vec{o}\vec{x})), \\ Hy_{nm}^j(\vec{o}\vec{x}) &= \frac{\partial}{\partial x_j} (Sy_n^m(\vec{o}\vec{x})). \end{aligned} \quad (17)$$

One can notice that, for a given pair of n and m , $M_{n,j}^{m,k(2)}(o)$ has 9 components and $M_{n,j}^{m(1)}(o)$ has 3 components, and therefore the number of multipole moments in this formulation is 12. The derivatives that appear in the relations above are calculated in an exact way using the following formulas.

5.1.1. Derivatives Formulas. According to the relation between Cartesian coordinates and polar coordinates and some chain rules [19], the derivatives of Sy_n^m can be obtained as follows:

$$\begin{aligned} \frac{\partial}{\partial x_1} (Sy_n^m(\vec{o}\vec{x})) &= \begin{cases} \frac{1}{2} (Sy_{n+1}^{m-1}(\vec{o}\vec{x}) - Sy_{n+1}^{m+1}(\vec{o}\vec{x})), & \text{if } m > 0; \\ -\text{Real}(Sy_{n+1}^1(\vec{o}\vec{x})), & \text{if } m = 0; \\ \frac{\partial}{\partial x_1} (Sy_n^{-m}(\vec{o}\vec{x})), & \text{if } m < 0, \end{cases} \\ \frac{\partial}{\partial x_2} (Sy_n^m(\vec{o}\vec{x})) &= \begin{cases} \frac{i}{2} (Sy_{n+1}^{m-1}(\vec{o}\vec{x}) + Sy_{n+1}^{m+1}(\vec{o}\vec{x})), & \text{if } m > 0; \\ i \text{Imag}(Sy_{n+1}^1(\vec{o}\vec{x})), & \text{if } m = 0; \\ \frac{\partial}{\partial x_2} (Sy_n^{-m}(\vec{o}\vec{x})), & \text{if } m < 0, \end{cases} \\ \frac{\partial}{\partial x_3} (Sy_n^m(\vec{o}\vec{x})) &= \begin{cases} -Sy_{n+1}^m(\vec{o}\vec{x}), & \text{if } m > 0; \\ \frac{\partial}{\partial x_3} (Sy_n^{-m}(\vec{o}\vec{x})), & \text{else,} \end{cases} \end{aligned} \quad (18)$$

In a similar manner, we can obtain the derivatives formulas of Ry_n^m for $n \geq 1$ as follows:

$$\begin{aligned} \frac{\partial}{\partial x_1} (Ry_n^m(\vec{o}\vec{x})) &= \begin{cases} \frac{1}{2} (Ry_{n+1}^{m-1}(\vec{o}\vec{x}) - Ry_{n+1}^{m+1}(\vec{o}\vec{x})), & \text{if } m > 0; \\ -\text{Real}(Ry_{n+1}^1(\vec{o}\vec{x})), & \text{if } m = 0; \\ \frac{\partial}{\partial x_1} (Ry_n^{-m}(\vec{o}\vec{x})), & \text{if } m < 0, \end{cases} \end{aligned}$$

$$\begin{aligned} & \frac{\partial}{\partial x_2} (Ry_n^m(\vec{o}\vec{x})) \\ &= \begin{cases} \frac{i}{2} (Ry_{n+1}^{m-1}(\vec{o}\vec{x}) + Ry_{n+1}^{m+1}(\vec{o}\vec{x})), & \text{if } m > 0; \\ i \operatorname{Imag} (Ry_{n+1}^1(\vec{o}\vec{x})), & \text{if } m = 0; \\ \frac{\partial}{\partial x_2} (Ry_n^{-m}(\vec{o}\vec{x})), & \text{if } m < 0, \end{cases} \\ & \frac{\partial}{\partial x_3} (Ry_n^m(\vec{o}\vec{x})) = \begin{cases} -Ry_{n+1}^m(\vec{o}\vec{x}), & \text{if } m > 0; \\ \frac{\partial}{\partial x_3} (Ry_n^{-m}(\vec{o}\vec{x})), & \text{else,} \end{cases} \end{aligned} \tag{19}$$

where $i^2 = -1$. It should be noted that $(\partial/\partial x_j)(Ry_0^0(\vec{o}\vec{x})) = 0$ for all $j \in \{1, 2, 3\}$.

5.1.2. *Error Bound.* By truncating (16) at the order p , and basing on the error given by (11), the error bound is given by

$$E'_p = C \left(\frac{\delta_j^k}{r-a} + \frac{1}{r-a} + \frac{a}{r^2-ra} \right) \left(\frac{a}{r} \right)^{p+1}, \tag{20}$$

where $C = \int_S |q(y)| ds$.

5.2. *O(N) Scheme.* To develop a scheme of order N , we introduce three basic translations adequate to the functions Sy_n^m and Ry_n^m which will be useful to build the formulas of translations multipole to multipole (M2M), multipole to local (M2L), and local to local (L2L). For the rest of the paper, we will take $k = \inf(m, n')$.

5.2.1. *Relations Formulas between Sy_n^m and Ry_n^m .* Under the condition $|\vec{o}\vec{y}| < |\vec{o}\vec{o}'|$, the function Ry_n^m satisfies a relationship of the following form:

$$\begin{aligned} & Ry_n^m(\vec{o}'\vec{y}) \\ &= \sum_{n'=0}^n \left\{ \sum_{m'=0}^k Ry_{n'}^{m'}(\vec{o}'\vec{o}) Ry_{n-n'}^{m-m'}(\vec{o}\vec{y}) \right. \\ & \quad + \sum_{m'=k+1}^{n'} (-1)^{m'-m} Ry_{n'}^{m'}(\vec{o}'\vec{o}) \overline{Ry_{n-n'}^{m'-m}(\vec{o}\vec{y})} \\ & \quad \left. + \sum_{m'=0}^{n'} (-1)^{m'} \overline{Ry_{n'}^{m'}(\vec{o}'\vec{o})} Ry_{n-n'}^{m+m'}(\vec{o}\vec{y}) \right\} \end{aligned} \tag{21}$$

which allows a pole shift that will permit us to construct the multipolar to multipolar translation.

Moreover, and under the condition $|\vec{x}_0\vec{x}'| < |\vec{o}\vec{x}_0|$, the function Sy_n^m can be further expressed in the following form

that will be used later to construct the conversion multipolar to local moment:

$$\begin{aligned} & Sy_n^m(\vec{o}\vec{x}) \\ &= \sum_{n'=0}^{\infty} \left\{ \sum_{m'=0}^{n'} (-1)^{n'} Ry_{n'}^{m'}(\vec{x}_0\vec{x}') Sy_{n+n'}^{m+m'}(\vec{o}\vec{x}_0') \right. \\ & \quad + \sum_{m'=0}^k (-1)^{n'+m'} \overline{Ry_{n'}^{m'}(\vec{x}_0\vec{x}')} Sy_{n+n'}^{m-m'}(\vec{o}\vec{x}_0') \\ & \quad \left. + \sum_{m'=k+1}^{n'} (-1)^{n+2m'-m} \overline{Ry_{n'}^{m'}(\vec{x}_0\vec{x}')} Sy_{n+n'}^{m'-m}(\vec{o}\vec{x}_0') \right\}. \end{aligned} \tag{22}$$

Just like the first relation, under the condition $|\vec{x}_0\vec{x}'| < |\vec{x}_0\vec{x}_1|$, we have a third equation that satisfies a pole relationship that will be used later to construct the local to local translation:

$$\begin{aligned} & Ry_n^m(\vec{x}_0\vec{x}') \\ &= \sum_{n'=n}^{\infty} \left\{ \sum_{m'=0}^k Ry_{n'}^{m'}(\vec{x}_1\vec{x}') Ry_{n-n'}^{m-m'}(\vec{x}_0\vec{x}_1') \right. \\ & \quad + \sum_{m'=k+1}^{n'} (-1)^{m'-m} Ry_{n'}^{m'}(\vec{x}_1\vec{x}') \overline{Ry_{n-n'}^{m'-m}(\vec{x}_0\vec{x}_1')} \\ & \quad \left. + \sum_{m'=0}^{n'} (-1)^{m'} \overline{Ry_{n'}^{m'}(\vec{x}_1\vec{x}')} Ry_{n-n'}^{m+m'}(\vec{x}_0\vec{x}_1') \right\}. \end{aligned} \tag{23}$$

5.2.2. *Translation of a Multipole Expansion.* Let us suppose that the surface S contains uniform distribution points included inside a sphere D of radius a centered at a point o at a distance ρ from the origin o' with $\rho > (c+1)a$ and $c > 1$. The integral equation is given by the multipole expansion (16):

$$\begin{aligned} v_k(x) = \sum_{n=0}^{\infty} \sum_{m=-n}^n & \left[\overline{Fy_{nm}^{jk}(\vec{o}\vec{x})} M_{n,j}^{m(1)}(o) \right. \\ & \left. + \overline{Hy_{nm}^j(\vec{o}\vec{x})} M_{j,m}^{m,k(2)}(o) \right]. \end{aligned} \tag{24}$$

If the point x is outside the sphere D_0 of radius $a + \rho$, then the integral equation takes the following form:

$$\begin{aligned} v_k(x) = \sum_{n=0}^{\infty} \sum_{m=-n}^n & \left[\overline{Fy_{nm}^{jk}(\vec{o}'\vec{x})} M_{n,j}^{m(1)}(o') \right. \\ & \left. + \overline{Hy_{nm}^j(\vec{o}'\vec{x})} M_{j,m}^{m,k(2)}(o') \right], \end{aligned} \tag{25}$$

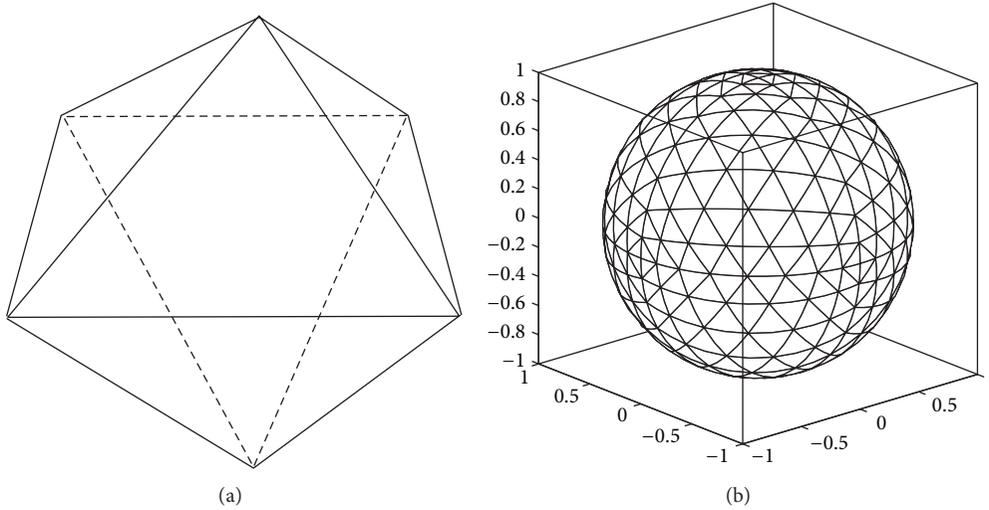


FIGURE 3: Initial configuration.

where

$$\begin{aligned}
 &M_{n,j}^{m(1)}(o') \\
 &= \sum_{n'=0}^n \left\{ \sum_{m'=0}^k R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) M_{n-n',j}^{m-m'(1)}(o) \right. \\
 &\quad + \sum_{m'=k+1}^{n'} (-1)^{m'-m} R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) \overline{M_{n-n',j}^{m-m'(1)}(o)} \\
 &\quad \left. + \sum_{m'=1}^{n'} (-1)^{m'} \overline{R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) M_{n-n',j}^{m+m'(1)}(o)} \right\}, \\
 &M_{n,j}^{m,k(2)}(o') \\
 &= \left(\overrightarrow{o'o} \right)_k M_{n,j}^{m(1)}(o') \quad \text{displaybreak} \\
 &\quad + \sum_{n'=0}^n \left\{ \sum_{m'=0}^k R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) M_{n-n',j}^{m-m',k(2)}(o) \right. \\
 &\quad + \sum_{m'=k+1}^{n'} (-1)^{m'-m} R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) \overline{M_{n-n',j}^{m'-m,k(2)}(o)} \\
 &\quad \left. + \sum_{m'=1}^{n'} (-1)^{m'} \overline{R y_{n'}^{m'} \left(\overrightarrow{o'o} \right) M_{n-n',j}^{m+m',k(2)}(o)} \right\}. \tag{26}
 \end{aligned}$$

5.2.3. Conversion of a Multipole Expansion into a Local Expansion. Let a surface S containing uniform distribution points included inside a sphere D of radius a centered at a point o at a distance ρ from a given point x_0 with $\rho > (c + 1)a$ and $c > 1$. The corresponding multipole expansion (16)

converges for any points x inside the sphere D_0 of radius a centered at x_0 . The velocity is given by

$$\begin{aligned}
 v_k(x) &= \int_S G_k^j(x, y) q_j(y) ds \\
 &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \left[\overline{F y R_{nm}^{jk}(\overrightarrow{x_0 \hat{x}})} L_{n,j}^{m(1)}(x_0) \right. \\
 &\quad \left. + \overline{H y R_{nm}^j(\overrightarrow{x_0 \hat{x}})} L_{n,j}^{m,k(2)}(x_0) \right], \tag{27}
 \end{aligned}$$

where

$$\begin{aligned}
 &L_{n,j}^{m(1)}(x_0) \\
 &= \sum_{n'=0}^{\infty} \left\{ \sum_{m'=0}^{n'} (-1)^n M_{n',j}^{m'(1)}(o) \overline{S y_{n+n'}^{m+m'}(\overrightarrow{o \hat{x}_0})} \right. \\
 &\quad + \sum_{m'=1}^k (-1)^{n+m'} \overline{M_{n',j}^{m'(1)}(o)} \overline{S y_{n+n'}^{m-m'}(\overrightarrow{o \hat{x}_0})} \\
 &\quad \left. + \sum_{m'=k+1}^{n'} (-1)^{n-m} \overline{M_{n',j}^{m'(1)}(o)} S y_{n+n'}^{m'-m}(\overrightarrow{o \hat{x}_0}) \right\}, \\
 &L_{n,j}^{m,k(2)}(x_0) \\
 &= -(\overrightarrow{o \hat{x}_0})_k L_{n,j}^{m(1)}(x_0) \\
 &\quad + \sum_{n'=0}^{\infty} \left\{ \sum_{m'=0}^{n'} (-1)^n M_{n',j}^{m',k(2)}(o) \overline{S y_{n+n'}^{m+m'}(\overrightarrow{o \hat{x}_0})} \right. \\
 &\quad + \sum_{m'=1}^k (-1)^{n+m'} \overline{M_{n',j}^{m',k(2)}(o)} \overline{S y_{n+n'}^{m-m'}(\overrightarrow{o \hat{x}_0})} \\
 &\quad \left. + \sum_{m'=k+1}^{n'} (-1)^{n-m} \overline{M_{n',j}^{m',k(2)}(o)} S y_{n+n'}^{m'-m}(\overrightarrow{o \hat{x}_0}) \right\},
 \end{aligned}$$

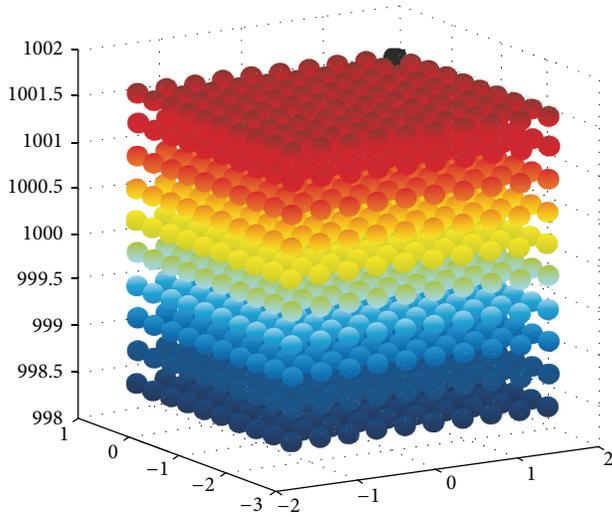


FIGURE 4: Sphere triangulation.

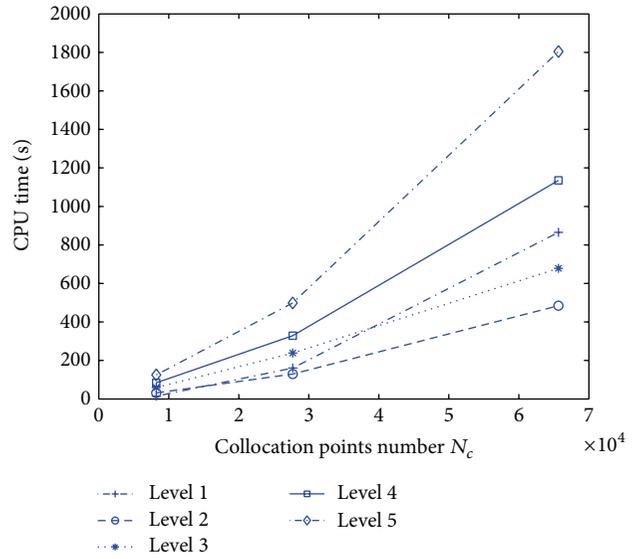


FIGURE 6: Zoom 1 of Figure 5.

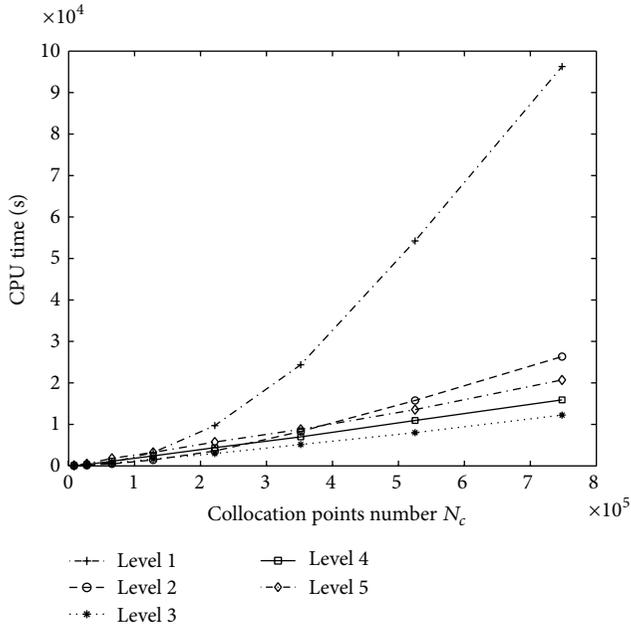


FIGURE 5: CPU time versus the number N_c of collocation points at different levels.

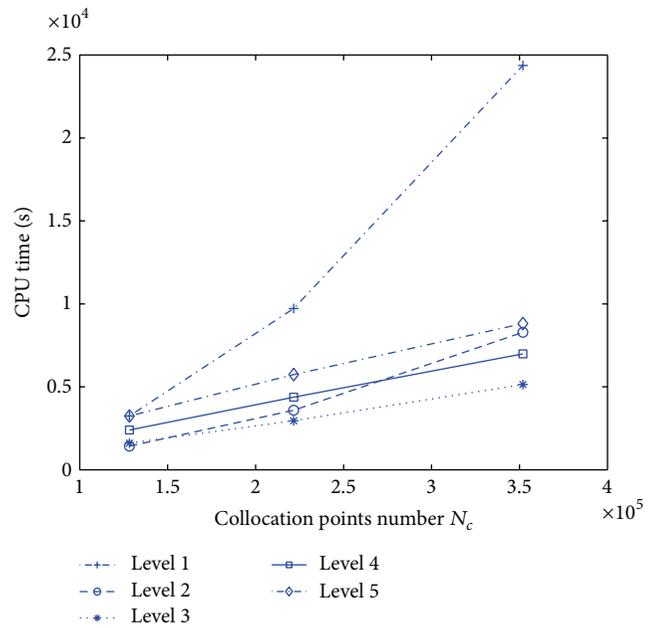


FIGURE 7: Zoom 2 of Figure 5.

$$F_y R_{nm}^{jk}(\vec{x}_0 \vec{x}) = \delta_k^j R_y^m(\vec{x}_0 \vec{x}) - (\vec{x}_0 \vec{x}) \frac{\partial}{\partial x_j} (R_y^m(\vec{x}_0 \vec{x})),$$

$$H_y R_{nm}^j(\vec{x}_0 \vec{x}) = \frac{\partial}{\partial x_j} (R_y^m(\vec{x}_0 \vec{x})).$$

(28)

5.2.4. Translation of a Local Expansion. Suppose that $v(x)$ can be written as in (29), and suppose that $x_1 \in D_0$; then for any point $x \in D_0$, $v(x)$ can be written according to the local moment centered at x_1 as follows:

$$v_k(x) = \int_S G_k^j(x, y) q_j(y) ds$$

$$= \sum_{n=0}^{\infty} \sum_{m=-n}^n \left[\overline{F_y R_{nm}^{jk}(\vec{x}_1 \vec{x})} L_{n,j}^{m(1)}(x_1) \right. \quad (29)$$

$$\left. + \overline{H_y R_{nm}^j(\vec{x}_1 \vec{x})} L_{n,j}^{m,k(2)}(x_1) \right],$$

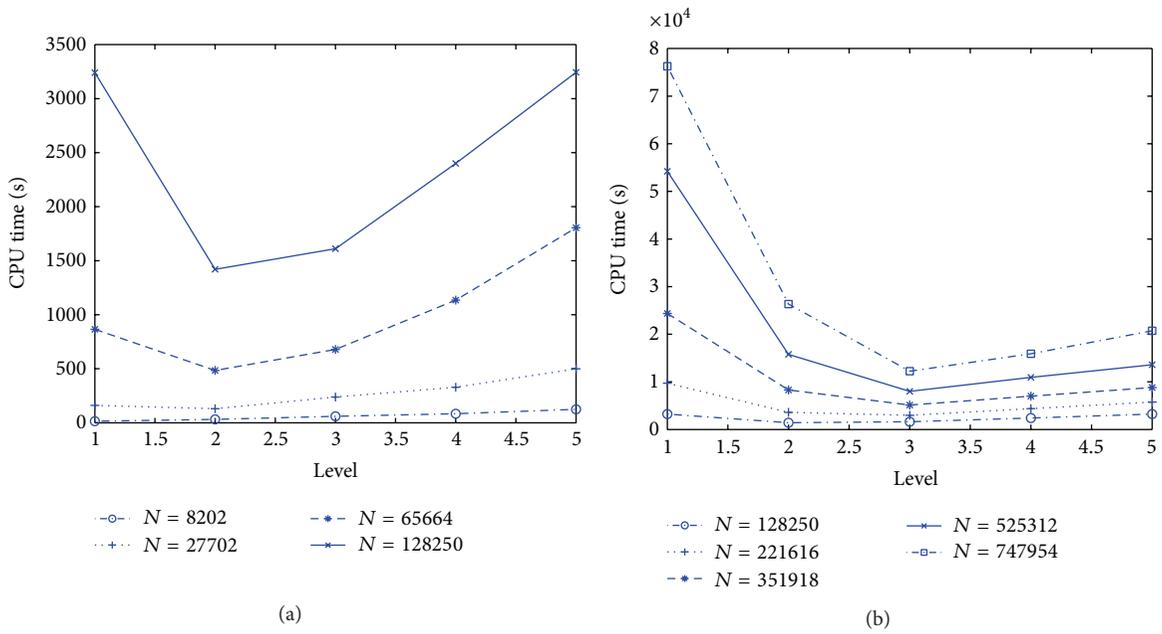


FIGURE 8: CPU time versus the level number for different collocation points N_c .

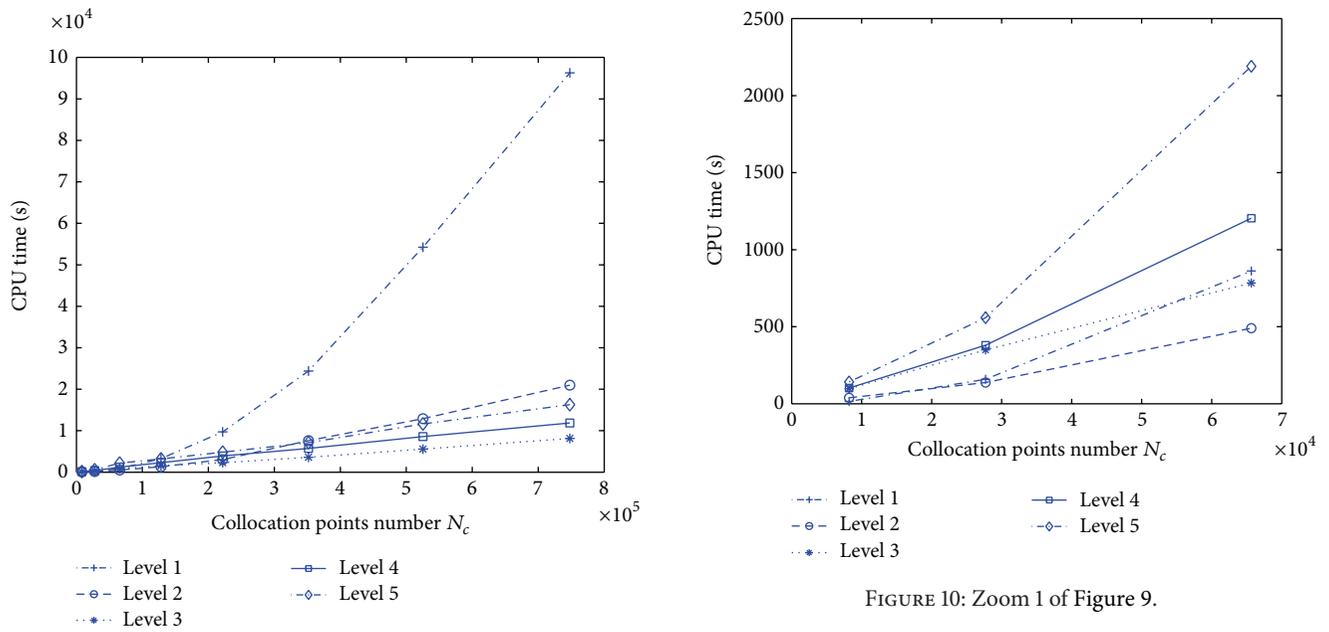


FIGURE 9: CPU time versus particles number for different levels.

FIGURE 10: Zoom 1 of Figure 9.

where

$$L_{n,j}^{m,k(1)}(x_1) = \sum_{n'=0}^{\infty} \left\{ \sum_{m'=0}^k (-1)^{m-m'} L_{n',j}^{m'(1)}(x_0) \overline{Ry_{n-n'}^{m-m'}(\overrightarrow{x_0x_1})} \right.$$

$$\left. + \sum_{m'=k+1}^{n'} L_{n',j}^{m'(1)}(x_0) Ry_{n-n'}^{m'-m}(\overrightarrow{x_0x_1}) + \sum_{m'=1}^{n'} (-1)^m \overline{L_{n',j}^{m'(1)}(x_0) Ry_{n-n'}^{m+m'}(\overrightarrow{x_0x_1})} \right\},$$

$$L_{n,j}^{m,k(2)}(x_1) = -(\overrightarrow{x_0x_1})_k L_{n,j}^{m(1)}(x_1)$$

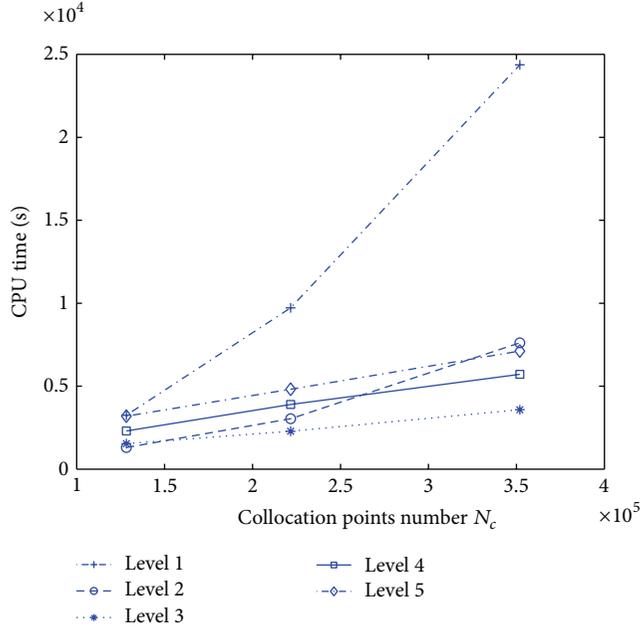


FIGURE 11: Zoom 2 of Figure 9.

$$\begin{aligned}
 & + \sum_{n'=0}^{\infty} \left\{ \sum_{m'=0}^k (-1)^{m-m'} L_{n',j}^{m',k(2)}(x_0) \overline{Ry_{n-n'}^{m-m'}(x_0 x_1)} \right. \\
 & + \sum_{m'=k+1}^{n'} L_{n',j}^{m',k(2)}(x_0) Ry_{n-n'}^{m'-m}(x_0 x_1) \\
 & \left. + \sum_{m'=1}^{n'} (-1)^m \overline{L_{n',j}^{m',k(2)}(x_0) Ry_{n-n'}^{m+m'}(x_0 x_1)} \right\}. \tag{30}
 \end{aligned}$$

5.2.5. *Error Bound.* With a similar reasoning as in the error bound of the $O(N \log N)$ scheme and under the same hypothesis of the representation (29), the evaluation of the velocity $v(x)$ at an order p generates an error bound expressed as follows:

$$E_p'' = C \left(\frac{\delta_j^k}{\rho - r} \right) + \left(\frac{1}{\rho - r} \right) + \left(\frac{r}{\rho^2 - \rho r} \right) \left(\frac{r}{\rho} \right)^{p+1}, \tag{31}$$

where $C = \int_C |q(y)| ds$.

6. Numerical Results

We present here some numerical results given by applying the FMM-BEM to the Stokes problem. The method uses several parameters on which depend the speed and the precision of the calculation, that is, the size of the cells, the level number of the grids, the order of truncation of the multipole series expansion, and so forth. The optimal values to obtain a good equilibrium between effectiveness and precision are given for the mono- and multilevel approaches. Theoretical

complexities are checked numerically. Problems of size $N = O(10^6)$ are solved in about two hours by iterations on a PC Pentium 4, 3.2GHZ with 4Go-RAM.

6.1. Configuration Test

6.1.1. *Sphere Triangulation.* The starting point is to generate a grid of many spheres that we put in a cube that represents the initial domain at the initial level of the tree (cf. Figure 3). To generate the collocation points on the spheres, we used a simple technique, presented by Pozrikidis [16] for the discretization of a sphere. Beginning from an octahedron, we developed a triangulation by subdividing recursively each triangle into four subtriangles until obtaining the desired level of discretisation. At each step, the triangle is projected on the surface of the sphere (cf. Figure 4).

6.1.2. *Numerical Parameters.* For the numerical consideration, we considered that two boxes are far away if they are separated by a third box. In addition, we took $r > 3a$ for the $O(N \log N)$ scheme and $\rho > 3r$ for the $O(N)$ scheme. The error given by (20) is thus controlled by $(3C/2L)2^l(1/3)^{p+1}$, where l is the level of the tree on which one approximates the far field interactions, and L represents the dispatcher of the initial cube. In the results of the tests presented below, we took $p = 12$ and we worked on a tree of depth equal to 5 which give a plug of error equal to $O(10^6)$. We obtained the same thing for the error given by (31).

6.2. *Presentation and Interpretation of the Results.* It is worth to be pointed out that the maximum value of the error obtained in the different tests is about 10^{-6} . These tests permit us to validate the method and show its precision. We remark from the calculation that the error introduced by the FMM compared to the traditional BEM does not have a practical incidence on the quality of the result.

6.2.1. *$O(N \log N)$.* We present here the results obtained with the $O(N \log N)$ scheme. Figure 5 that represents the CPU time versus the number N_c of collocation point at different levels ($N_c = 2(Nt + 1)$) shows that the method FMM in the case of the scheme $O(N \log N)$ is useful not only to minimize the memory storage of the matrix due to the compression carrying out but also to enable us to gain in terms of computing time. This method appears to be much better than the direct calculation for the large values of N where $N = 3Np(N2c1)$. Indeed we can see that the effectiveness of this method appears from a certain number N_c evaluated by the results between 10^4 and 3×10^4 . For $N_c \leq 9202$ the method FMM does not have any influence from the computing times point of view and from $N_c = 27702$ the efficiency of the FMM method becomes to be visible; see Figures 5 and 8. To be more precise, we consider two boxes, shown, respectively, in Figures 6 and 7, that are obtained from Figure 5 by zooming on the regions that seem to be interesting for the interpretation. For $N_c = 27702$ the efficiency of the method starts from level 2 (see Figure 6) and then for $N_c = 351918$ the efficiency becomes to be visible

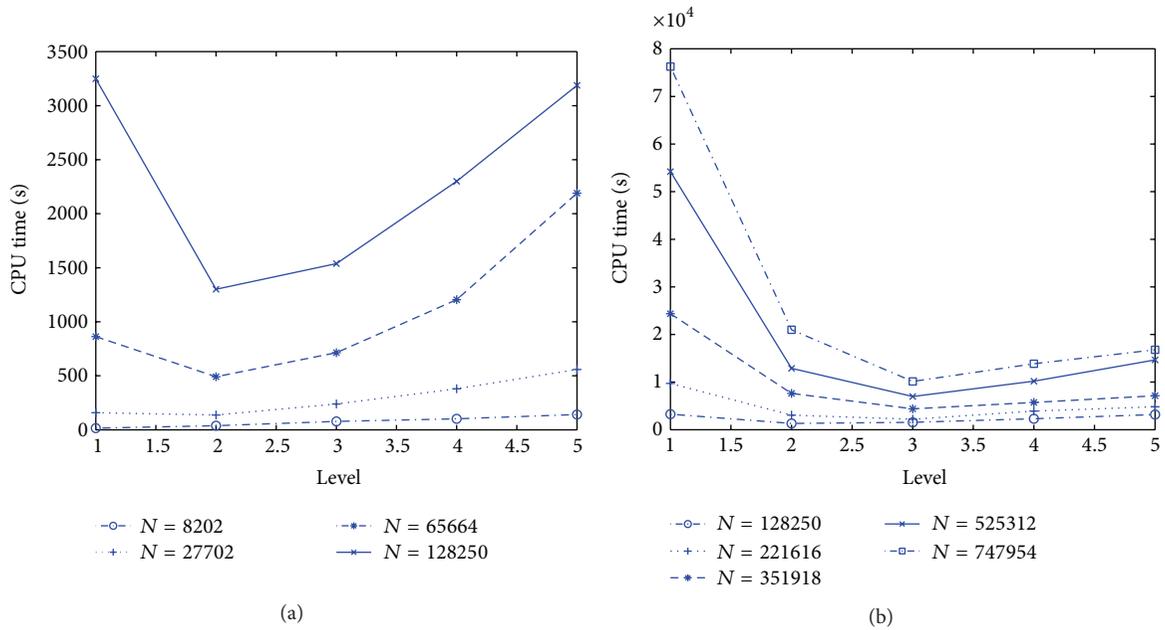


FIGURE 12: CPU time versus the level for several values of N .

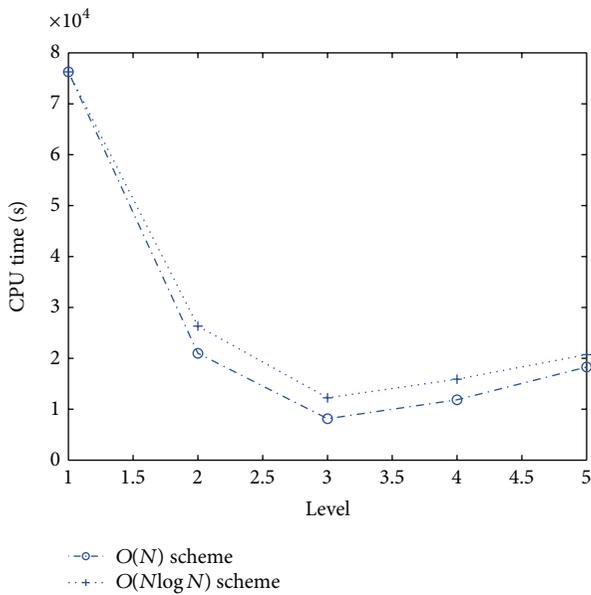


FIGURE 13: CPU time of the two schemes versus the level for $N_c = 351918$.

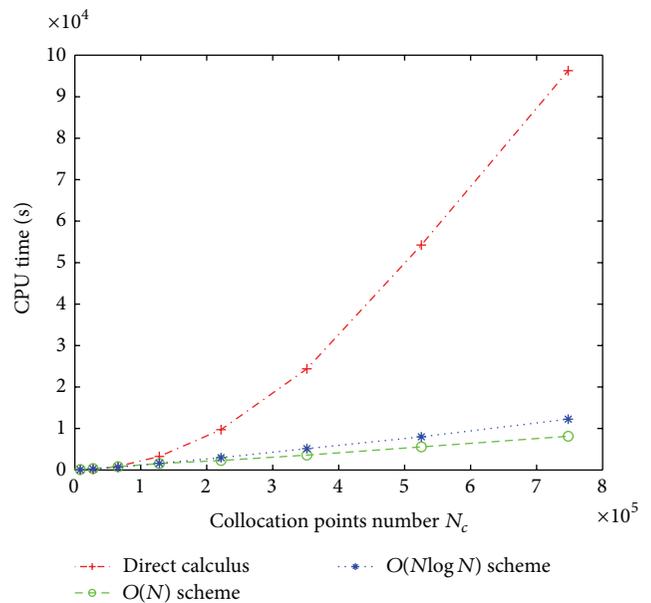


FIGURE 14: CPU time of the direct calculus and the two schemes $O(N \log N)$ and $O(N)$ versus the collocation points N_c for level = 3.

from level 3 (see Figure 7). We can also see from Figure 7 that level 4 becomes better than level 2 from a certain number N evaluated to be between 2.75×10^5 and 3×10^5 . Thus, as a general rule about the potential of the FMM method, we can conclude that the more the size of the problem N increases, the more we have to work down in the hierarchical tree of the code.

6.2.2. $O(N)$. The interpretation of the results found for the $O(N)$ scheme is similar to that obtained for the $O(N \log N)$ one. The same remarks given from all figures for the $O(N \log N)$ scheme may be reproduced here from Figures 9, 10, 11, and 12 which correspond to the $O(N)$. The comparison of the results of the two schemes $O(N \log N)$ and $O(N)$ is

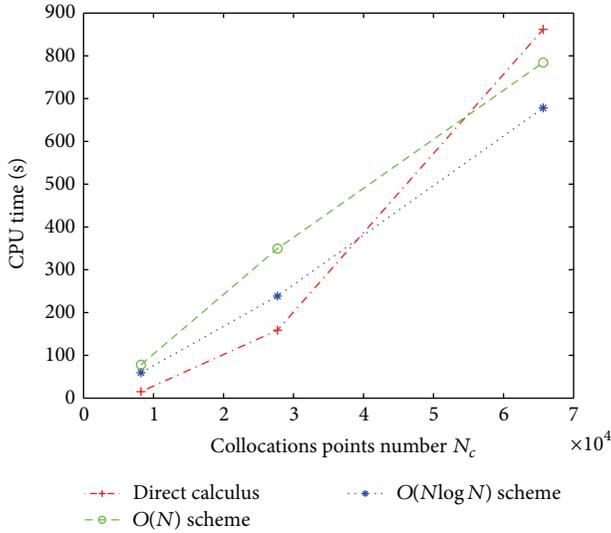


FIGURE 15: CPU time of the direct calculus and the two schemes $O(N \log N)$ and $O(N)$ versus the level for several values of N_c .

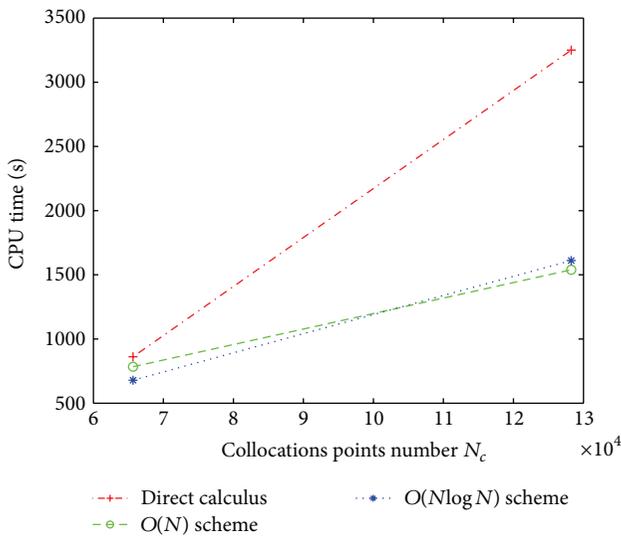


FIGURE 16: CPU time of the direct calculus and the two schemes $O(N \log N)$ and $O(N)$ versus the level for several values of N_c .

illustrated in Figures 13, 14, 15, and 16. It should be noted that the $O(N)$ scheme becomes more efficient than the $O(N \log N)$ scheme after a certain number N ; see Figure 16. However, one has to work under a certain depth of the tree; otherwise, the $O(N \log N)$ becomes more efficient from the CPU time point of view; see Figure 15.

7. Concluding Remarks

We presented in this work the fast multipole method applied to the Stokes problem in a 3D infinite viscous fluid in the presence of several rigid bodies. The formulas obtained for both schemes $O(N \log N)$ and $O(N)$, where N is the size of the matrix obtained from the discretization of the problem,

are implemented numerically. The numerical results obtained are very satisfactory and incite us to use the method in the future in a traditional IBEM code that will allow for an efficient resolution of the linear system by iterative methods such as GMRES. This work makes it possible to investigate in future works more complex and realistic configurations such as the sedimentation of dilute suspensions composed by a great number of particles or the movements of deformable particles. Such configurations could enable us to compare between the numerical results and the experimental ones. For the case of deformable bodies, the integral equation must take into account the double potential layer. The development of an adequate FMM-BEM based on formulae given previously constitutes an interesting perspective for a future area of research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This project was supported by King Saud University, Deanship of Scientific Research, College of Sciences Research Center.

References

- [1] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, 1987.
- [2] L. Greengard and V. Rokhlin, "On the efficient implementation of the fast multipole algorithm," Tech. Rep. 602, Department of Computer Science, Yale University, 1988.
- [3] L. Greengard and V. Rokhlin, "On the evaluation of electrostatic interactions in molecular modeling," *Chemica Scripta*, vol. 29, pp. 139–144, 1989.
- [4] H. Y. Wang and R. LeSar, "An efficient fast-multipole algorithm based on an expansion in the solid harmonics," *Journal of Chemical Physics*, vol. 104, no. 11, pp. 4173–4179, 1996.
- [5] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *Journal of Computational Physics*, vol. 60, no. 2, pp. 187–212, 1985.
- [6] J. Barnes and P. Hut, "A hierarchical $O(N \log N)$ force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, 1986.
- [7] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *Journal of Computational Physics*, vol. 86, no. 2, pp. 414–439, 1990.
- [8] R. Coifman, V. Rokhlin, and S. Wandzura, "Fast multiple method for the wave equation: a pedestrian prescription," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 3, pp. 7–12, 1993.
- [9] L. Greengard, M. C. Kropinski, and A. Mayo, "Integral equation methods for Stokes flow and isotropic elasticity in the plane," *Journal of Computational Physics*, vol. 125, no. 2, pp. 403–414, 1996.

- [10] A. A. Mammoli and M. S. Ingber, "Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods," *International Journal for Numerical Methods in Engineering*, vol. 44, no. 7, pp. 897–917, 1999.
- [11] A. A. Mammoli and M. S. Ingber, "Parallel multipole BEM simulation of two-dimensional suspension flows," *Engineering Analysis with Boundary Elements*, vol. 24, no. 1, pp. 65–73, 2000.
- [12] L. Greengard and J. Helsing, "On the numerical evaluation of elastostatic fields in locally isotropic two-dimensional composites," *Journal of the Mechanics and Physics of Solids*, vol. 46, no. 8, pp. 1441–1462, 1998.
- [13] N. Nishimura, K. I. Yoshida, and S. Kobayashi, "A fast multipole boundary integral equation method for crack problems in 3D," *Engineering Analysis with Boundary Elements*, vol. 23, no. 1, pp. 97–105, 1999.
- [14] N. Nishimura, "Fast multipole accelerated boundary integral equation methods," *Applied Mechanics Reviews*, vol. 55, no. 4, pp. 299–324, 2002.
- [15] N. A. Gumerov and R. Duraiswami, "Fast multipole method for the biharmonic equation in three dimensions," *Journal of Computational Physics*, vol. 215, no. 1, pp. 363–383, 2006.
- [16] C. Pozrikidis, *Numerical Computation in Science and Engineering*, Oxford University Press, New York, NY, USA, 1998.
- [17] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, NY, USA, 1965.
- [18] O. A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, Science Publishers, New York, NY, USA, 1969.
- [19] K. Yoshida, *Applications of fast multipole method to boundary integral method [Ph.D. thesis]*, Department of Global Environmental Engineering, Kyoto University, Kyoto, Japan, 2001.