Computational Mechanics (1992) 9, 121–135

Computational Mechanics

Completed double layer in half-space: a boundary element method

N. Phan-Thien[1], D. Tullock[1] and S. Kim[2]

[1] Department of Mechanical Engineering, The University of Sydney, NSW 2006, Australia
[2] Department of Chemical Engineering, University of Wisconsin-Madison, WI 53706, USA

Abstract. The paper reports a numerical method for the solution of Stokes flows past a system of particles of arbitrary shape near a planar surface on which the velocity vector is zero. The method is an application of the Completed Double Layer Boundary Integral Method (CDL-BIEM) by Kim and Karrila [1]. It uses an iterative solver and therefore can handle a large number of particles with complex geometries. Particles' trajectories for a few typical problems are presented to illustrate the feasibility of the method.

1 Introduction

The Boundary Element Method (BEM) has become an efficient numerical method for solving boundary-value problems in engineering science, especially for exterior and linear problems (for example, the unbounded flow past a particle). Since its introduction to continuum mechanics by Fichera [2], Rizzo [3], Hess and Smith [4], the method has become popular and there are a number of excellent texts dealing with the subject, e.g., Banerjee and Butterfield [5] and Brebbia et al. [6]. The technique has been applied to Stokes problems by Youngren and Acrivos [7], Bush and Tanner [8], Onishi et al. [9]; a recent review of its applications in Stokes flow can be found in Bush and Tanner [10]. Extension of the method to deal with half-space problems has been considered by Telles and Brebbia [11], in the case of a traction-free half-space boundary, and in the case of a velocity-free boundary by Tran-Cong and Phan-Thien [12] and more recently by Ascoli et al. [13]. In these half-space problems, an appropriate singularity solution is used which satisfies the boundary conditions on the boundary on the half-space exactly (Mindlin's kernel [14], or the image system of the Kelvin state [35]). These problems can also be dealt with by the standard BEM; however, to be efficient one must then use the so-called infinite elements, which are elements over large but finite surfaces—these elements are needed to model the boundary of the half-space.

The BEM is particularly well suited to a large class of Stokes flows known as the resistance and mobility problems of a system of particles. In the resistance problem, one knows the velocities (translational and rotational) of the particles and the task is to calculate the drag forces and torques acting on the particles. In the mobility problem, the forces and torques acting on each particle are known, and one wishes to find the rigid body motions of the particles that result from the applications of these forces and torques. Both of these types of problems are collectively known as the sedimentation problem. Successful treatments of the sedimentation problem using the standard BEM have been reported, [16]–[20]. However, the sedimentation problem results in a set of Fredholm integral equations of the first kind and may be ill-conditioned. As the geometry is modelled more and more accurately, by using more and more elements, the condition number of the system matrix increases unboundedly. In the limit of an infinite dimensional space, the integral operator, being compact, has no bounded inverse. An alternative integral formulation for the Stokes equations has been reported by Power and Miranda [21], who used a distribution of a double layer with a surface density that satisfies a Freedholm integral equation of the second kind. The double layer operator is compact, and that fact has been exploited by Karrila et al. [23]–[24] to arrive at a fixed point iteration scheme for solving the resulting integral equations.

Furthermore, iterative schemes of this type are amenable to parallel computing architectures, and applications of the method in a parallel computing environment have been successfully carried out [24]–[25]. The method, being essentially a range completion of the double layer operator, is referred to as the Completed Double Layer Boundary Integral Equation Method (CDL-BIEM).

Other methods dealing with Stokes flows are also available. For example, the methods developed by Mazur and van Saarloos [26] and Mazur [27] include hydrodynamic interactions among the particles by calculating approximations to the resistance matrix or the mobility tensors (see, Happel and Brenner [30]); but these methods require a large number of terms to be included if the near-field hydrodynamic interaction is to be modelled correctly. A recent development by Durlofsky et al. [28] and Brady and Bossis [29] includes correctly both the near-field and the far-field interactions. This method uses a combination of the multipole expansion and the two-body solutions to generate the resistance matrix to the problem. In principle the method can be used with compact shapes but the exact solution for the appropriate two-body problem is required.

In this paper we report an application of the CDL-BIEM for Stokes flow problems in a half-space. The problems have been considered by a standard BEM previously [12], but the standard BEM is not well suited to deal with a complex problem that requires a large number of boundary elements, either due to a lack of computing resources or the ill-conditioned problems with the standard boundary element formulation. The method is tested against the exact solution for the motion of a sphere in the presence of a plane interface by Lee and Leal [31]. The trajectories of three and eight spheres at the vertices of an equilateral triangle and a cube are reported. The periodicity of these systems (Caflisch et al. [32]; Durlofsky et al. [28]) is no longer present in a half-space.

2 Governing equations

2.1 The standard boundary element method

We consider the Stokes flow problem in a semi-infinite domain given by $\mathscr{D} = \{(x_1, x_2, x_3): x_3 \geqq 0\}$. The field (Stokes) equations are

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}. \quad \nabla \cdot \boldsymbol{u} = 0 \text{ in } \mathscr{D}, \tag{1}$$

$$\boldsymbol{\sigma} = -P\mathbf{1} + \eta(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T), \tag{2}$$

where $\boldsymbol{u}$ is the velocity vector, $\boldsymbol{\sigma}$ is the stress tensor, $\mathbf{1}$ is the unit tensor, P is the hydrostatic pressure, η is the fluid viscosity and the superscript T denotes the transpose operation. The bounding surface of $\mathscr{D}$ consists of the infinite plane S_∞ where $x_3 = 0$, and the bounding surfaces of M particles in $\mathscr{D}$, collectively called S, $S = \cup S_j$, where S_j is the bounding surface of particle $j = 1, \ldots, M$. We only deal with rigid particles in this paper. On S_∞, the velocity vector is nil, and on S, only rigid body motions exist. We will be concerned with the mobility problem, where, given the external forces and torques acting on the particles, it is desired to find the rigid body motions of every particle in $\mathscr{D}$, and subsequently the velocity field everywhere in $\mathscr{D}$.

By using the reciprocal theorem (see, for examples, Banerjee and Butterfield [5]; Brebbia et al. [6]; Kim and Karrila [1]), the integral representation of the velocity field is given by [12],

$$u_j(X) + \int_S H_{ij}(x, X) u_i(x) dS(x) = \int_S G_{ij}(x, X) t_i(x) dS(x), \quad X \in \mathscr{D}, \tag{3}$$

where $G_{ij}(x, X)$ is the singularity solution (the i component of the velocity) due to a point force acting at $X \in \mathscr{D}$ in the j direction, $t_i(x) = \sigma_{ki} n_k$ is the traction vector at x, $\boldsymbol{n}$ is the normal *outward* unit vector on S, and $H_{ij}(x, X)$ is the associated traction vector of $G_{ij}(x, X)$. Note that

$$H_{ij}(x, X) = n_k(x) \Sigma_{ijk}(x, X),$$

where $\Sigma_{ijk}(x, X)$ is the associated stress obtained from the singularity solution. This singularity

solution was derived by Blake [33], and Hasimoto and Sano [34] for Stokes flows, and Phan-Thien [35] for the elasticity problems. The expressions for G_{ij} and H_{ij} are given in the Appendix.

By allowing X to go to the boundary S, and taking the limiting process appropriately one obtains a set of boundary integral equations to solve for the unknown boundary values (velocity and traction vectors). The integral on the right hand side of (3) is called the single-layer potential, and the integral on the left hand side of (3) is called the double-layer potential, in analogy with potential theory. For rigid particles, it can be shown that only a single-layer potential is required in the integral velocity representation [12]. Solving the mobility requires the inversion of the compact single-layer integral operator. Since its inverse is unbounded in an infinite dimensional space, the more accurate the discretisation of the problem geometry, the more one experiences the ill-conditioning in the solution. Although this situation does not occur in simple problems (eg., spheres not near touching), it would be pleasing to consider a BEM that does not suffer from this ill-conditioning problem by a re-formulation of the integral representation of the Stokes equations. Furthermore, since the system matrix that results from the discretisation of the standard BEM is dense and non-symmetric, its solution is computationally expensive (of $O(N^3)$ operations, where $N \times N$ is the size of the matrix; storing a large matrix is also taxing on computer memory, usually a disk-based solver is employed, but this creates large overhead due to input–output operations).

2.2 The completed double layer boundary integral equation method

Power and Miranda [21] show that it is possible to formulate an integral representation for Stokes flow which leads to a set of Fredholm integral equations of the second kind. This is done by dropping the single-layer term in (3), and considering an integral representation of the velocity field using just the double layer:

$$u_j(\boldsymbol{X}) = \int_S K_{ij}(\boldsymbol{x}, \boldsymbol{X})\varphi_i(\boldsymbol{x})dS(\boldsymbol{x}), \quad \boldsymbol{X} \in \mathscr{D}, \tag{4}$$

where the double-layer kernel is given by

$$K_{ij}(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{x}) = 2\hat{n}_k(\boldsymbol{x})\Sigma_{ijk}(\boldsymbol{x}, \boldsymbol{X})$$

in which $\hat{\boldsymbol{n}} = -\boldsymbol{n}$ is the normal unit vector pointing into the fluid domain and $\boldsymbol{\varphi}$ is the unknown surface density of the double-layer distribution. The factor 2 is included so that the eigenvalues of the double-layer operator lie between ± 1 (refer to Kim and Karrila [1]).

On the surface of the domain, the jump in the double layer can be calculated exactly for Liapunov-smooth surface (by noting that the image system of the singularity solution contains no singularity), and we have,

$$u_j(\boldsymbol{\zeta}) = \varphi_j(\boldsymbol{\zeta}) + \int_S K_{ij}(\boldsymbol{x}, \boldsymbol{\zeta})\varphi_i(\boldsymbol{x})dS(\boldsymbol{x}), \quad \boldsymbol{\zeta} \in S. \tag{5}$$

The double layer cannot impart any force nor any torque on the fluid (Odqvist [36]) and therefore excludes most problems of interest. In fact, for M embedded particles, there are $6M$ null solutions to the double-layer kernels, each corresponds to rigid body motion of each particle. Therefore, solutions when they exist, are non-unique. The $6M$ missing pieces (the deficiency of the range) prevent (5) from having a solution at all, almost always.

Power and Miranda [21] recognize this and seek to complete the double layer representation by using the velocity fields of point forces and point torques. This leads to a velocity representation of the form

$$u_j(\boldsymbol{X}) - u_j^{\infty} = -\sum_{p=1}^{M} (F_i^{(p)} - \tfrac{1}{2}(\boldsymbol{T}^{(p)} \times \nabla)_i) G_{ji}(\boldsymbol{X}, \boldsymbol{x}_c^{(p)}) + \int_S K_{ij}(\boldsymbol{x}, \boldsymbol{X})\varphi_i(\boldsymbol{x})dS(\boldsymbol{x}), \quad \boldsymbol{X} \in \mathscr{D}, \tag{6}$$

where $\boldsymbol{u}^{\infty}$ is the ambient fluid velocity (any admissible Stokes solution in half-space), $\boldsymbol{F}^{(p)}$ and $\boldsymbol{T}^{(p)}$ are the force and torque acting on particle p, respectively, and $\mathbf{G}(\boldsymbol{X}, \boldsymbol{x}_c^{(p)})$ is the singularity solution that corresponds to a point force being placed at $\boldsymbol{x}_c^{(p)}$, any point inside the particle p. In this paper,

$x_c^{(p)}$ is chosen to be the center of mass of the particle. The solution $\frac{1}{2}T^{(p)} \times \nabla \cdot \mathbf{G}(X, x_c^{(p)})$ generates a torque $T^{(p)}$ acting on particle p. It is not necessary to use these point-force and point-torque solutions; any other solutions that generate a force and a torque on the particle p would be equally acceptable; indeed Pakdel and Kim [25] have considered a distribution of these point-force and point-torque solutions within particle p to model particles with complex geometries (e.g., spiked spheres).

On the surface S_n, $n = 1, \ldots, M$, the velocity is a rigid body motion and the resulting boundary integral equation is

$$U_j^{(n)} + (\omega^{(n)} \times (\zeta - x_c^{(n)}))_j - u_j^{\infty} = -\sum_{p=1}^{M} (F_i^{(p)} - \tfrac{1}{2}(T^{(p)} \times \nabla)_i) G_{ji}(\zeta, x_c^{(p)}) + \varphi_j(\zeta) + (\mathscr{K}\varphi)_j(\zeta), \quad \zeta \in S_n, \tag{7}$$

where $\mathscr{K}$ denotes the (surface integral) double-layer operator, and $U^{(n)}$, $\omega^{(n)}$ are the translation and rotational motion of particle n, respectively.

Since the null space of $1 + \mathscr{K}$ is non-trivial and has dimension $6M$, we need to impose $6M$ linearly independent extra equations to obtain a unique solution. Power and Miranda [21] choose to associate the force and the torque on particle n with the null solutions on its surface (which represent the rigid body motions of particle n); that is,

$$F_i^{(n)} = \langle \varphi^{(n,i)}, \varphi \rangle, \qquad i = 1, \ldots, 3,$$
$$T_i^{(n)} = \langle \varphi^{(n,i+3)}, \varphi \rangle, \quad i = 1, \ldots, 3,$$

where $\varphi^{(n,i)}$ represent the translational motion ($i = 1, \ldots, 3$) and the rotational motion ($i = 4, \ldots, 6$) of particle n, and the angular brackets denote the natural inner product

$$\langle a, b \rangle = \int_S a(x) \cdot b(x) dS(x).$$

It is also assumed that the null solutions have been normalized with respect to this inner product, i.e.,

$$\langle \varphi^{(n,i)}, \varphi^{(m,j)} \rangle = \delta_{mn}\delta_{ij}.$$

For arbitrarily shaped particles, a Gram–Schmidt orthonormalization process needs to be carried out. This is a straightforward task since $\varphi^{(n,i)}$ takes on non-zero values only on particle n. A numerical implementation of Power and Miranda's completion scheme was reported by Fan and Yeow [37].

Power and Miranda's [21] approach will certainly render a unique solution to the boundary integral equation (7). Different completion schemes for multiples particles, with or without a container surface are discussed in Kim and Karrila [1], and naturally, the class of boundary integral equations that arises from these completion schemes is collectively called the Completed Double Layer Boundary Integral Equation Methods.

Since we are interested in the mobility problem (given $F^{(n)}$ and $T^{(n)}$, find the rigid body motions of each and every particle), we will adopt a completion scheme discussed in Karrila et al. [24] (a more in-depth discussion of the method can be found in Kim and Karrila [1]).

In essence, we solve the following boundary integral equations

$$\varphi_j(\zeta) + (\mathscr{K}\varphi)_j(\zeta) + \varphi_j^{(n,l)}(\zeta)\langle \varphi^{(n,l)}, \varphi \rangle = -u_j^{\infty} + \sum_{p=1}^{M} (F_i^{(p)} - \tfrac{1}{2}(T^{(p)} \times V)_i) G_{ji}(\zeta, x_c^{(p)}), \quad \zeta \in S \tag{8}$$

for φ. Note that the usual summation convention with regard to the subscripts, and superscripts is observed here ($n = 1, \ldots, M, l = 1, \ldots, 6$).

After the boundary solution φ is obtained, the surface velocity field is, from (6),

$$u_j(\zeta) = -\varphi_j^{(n,l)}(\zeta)\langle \varphi^{(n,l)}, \varphi \rangle, \quad \zeta \in S. \tag{9}$$

The rigid body motions of particles p can be extracted from (9) by taking the inner product of (9) and $\varphi^{(p,m)}$ and using the orthonormalization conditions of these null solutions.

Note that the integral operator in (8) is just the (surface) integral operator in (6) plus a sum of the first-rank operators $\varphi^{(n,l)}\langle \varphi^{(n,l)}, \cdot \rangle$ that involves the null solution of $1 + \mathscr{K}$. This effectively

moves the eigenvalue -1 of the (surface) double-layer operator to zero. This technique of modification of a particular eigenvalue without affecting the rests is called Wielandt's deflation [38]; Kim and Karrila [1] refer to this as the physical deflation, because the motivation behind the technique can be purely physical.

As it stands, (8) is not yet amenable to simple iterative solution, because of the eigenvalue $+1$ of $\mathscr{K}$. Although the corresponding eigenvectors of $\mathscr{K}$ are not known, the eigenvectors for the adjoint operator $\mathscr{K}^\dagger$ are simply the unit normals on the M particles. These M eigenvectors, $\boldsymbol{\psi}^{(n)}$, can be orthonormalized so that

$$\boldsymbol{\psi}^{(n)}(\zeta) = \begin{cases} \alpha_n \hat{\boldsymbol{n}}, & \zeta \in S_n, \\ \boldsymbol{0}, & \text{otherwise}, \end{cases}$$

where $\alpha_n = 1/\sqrt{S_n}$ is the normalization factor.

To deflate the eigenvalue $+1$ we subtract from the operator in (8) a sum of the first-rank operators $\boldsymbol{\psi}^{(n)}\langle \boldsymbol{\psi}^{(n)}, \cdot \rangle$. This process also modifies the right hand side of (8) (for this reason, Kim and Karrila [1] refer to this deflation as the mathematical deflation). The final form of the CDL-BIEM is

$$\varphi_j(\zeta) + (\mathscr{K}\boldsymbol{\varphi})_j(\zeta) + \varphi_j^{(n,l)}(\zeta)\langle \boldsymbol{\varphi}^{(n,l)}, \boldsymbol{\varphi} \rangle - \psi_j^{(n)}(\zeta)\langle \boldsymbol{\psi}^{(n)}, \boldsymbol{\varphi} \rangle = b_j(\zeta) - \tfrac{1}{2}\psi_j^{(n)}\langle \boldsymbol{\psi}^{(n)}, \boldsymbol{b} \rangle, \quad \zeta \in S, \tag{10}$$

where

$$b_j(\zeta) = -u_j^\infty + \sum_{p=1}^{M} (F_i^{(p)} - \tfrac{1}{2}(\boldsymbol{T}^{(p)} \times \boldsymbol{\nabla})_i) G_{ji}(\zeta, \boldsymbol{x}_c^{(p)}), \quad \zeta \in S.$$

Equation (10) is basic of the CDL-BIEM as described by Karrila et al. [24] and can be solved by a successive iteration for $\boldsymbol{\varphi}$. The boundary solution can be used to extract the rigid body motions of the particles, and finally, the velocity at any point in $\mathscr{D}$ can be calculated using (6).

3 Numerical implementation

The numerical implementation of the CDL-BIEM proceeds in several stages. First, from the geometries of the particles the normalized null functions $\boldsymbol{\varphi}^{(n,l)}$, $l = 1, \ldots, 6$ for particle n are constructed. The first three are simply the translational motions:

$$\boldsymbol{\varphi}^{(n,l)} = \frac{\boldsymbol{e}_l}{\sqrt{S_n}}, \quad l = 1, 2, 3,$$

where $\{\boldsymbol{e}_l\}$ are the unit vectors along the x_l directions. The next three null functions are simply the rotational motions of particle n:

$$\boldsymbol{\varphi}^{(n,l+3)}(\zeta) = \frac{1}{\sqrt{I_l}} \boldsymbol{e}_l \times \boldsymbol{\rho}, \quad l = 1, 2, 3, \quad \zeta \in S_n,$$

where $\boldsymbol{\rho} = \zeta - \boldsymbol{x}_c^{(n)}$ is the vector from the centre of mass of S_n to ζ, i.e.,

$$\int_{S_n} \boldsymbol{\rho} dS = \boldsymbol{0},$$

and I_l are the normalized constants:

$$I_1 = \int_{S_n} (\rho_2^2 + \rho_3^2) dS, \quad I_2 = \int_{S_n} (\rho_3^2 + \rho_1^2) dS, \quad I_3 = \int_{S_n} (\rho_1^2 + \rho_2^2) dS.$$

If $\boldsymbol{x}_c^{(n)}$ is not the centre of mass of the boundary then these rotational null functions must be normalized by a Gram–Schmidt process.

Next, following the discretisation of S, the main iterative equation to solve for $\boldsymbol{\varphi}$ is

$$\boldsymbol{\varphi}' = \boldsymbol{b}' - \mathscr{K}\boldsymbol{\varphi} - \boldsymbol{\varphi}^{(n,l)}(\zeta)\langle \boldsymbol{\varphi}^{(n,l)}, \boldsymbol{\varphi} \rangle + \boldsymbol{\psi}^{(n)}(\zeta)\langle \boldsymbol{\psi}^{(n)}, \boldsymbol{\varphi} \rangle, \tag{11}$$

where $\boldsymbol{b}^i$ is the right hand side of (10) and $\boldsymbol{\varphi}^i$ is the next corrected solution. This step is repeated until the norm

$$\max_{\zeta \in S} \| \boldsymbol{\varphi}^i(\zeta) - \boldsymbol{\varphi}(\zeta) \| < \varepsilon, \tag{12}$$

where ε is a set tolerance. Our experience indicates that a tolerance of $O(10^{-3})$ is adequate.

Since the method is intended for large scale simulations, storage of the entire system matrix in core is not feasible. For example, a simultation involving 20000 elements resulting in a fully-populated system of size 60000 × 60000 will require storage of 14.4 G Bytes in single precision arithmetic. For this reason, the system matrix is not stored, but rather, individual rows are computed when needed in the iterative strategy. Consequently, a large problem such as this can be attempted on a mini-computer (Titan workstation) in a reasonable amount of time (about 10 hours CPU time).

3.1 Boundary elements

In the numerical scheme, the boundary S is partitioned into a number of elements over which φ is considered to be constant. The geometry is modelled by linear (TRIA3: 3 nodes per triangle, QUAD4: 4 nodes per quadrilateral), quadratic (TRIA6: 6 nodes per triangle, QUAD8: 8 nodes per quadrilateral) or quadratic Lagrangian (TRIA7: 7 nodes triangle, QUAD9: 9 nodes per quadrilateral) elements. Integrals on the right hand side of (11) can be calculated by Gaussian quadrature. For flat elements, a 1 point integration rule is sufficient for those integrals that involve the null functions $\boldsymbol{\varphi}^{(n,l)}$ and $\boldsymbol{\psi}^{(n)}$, since these functions vary at most linearly with the coordinates. However, a 2×2 rule is necessary for curvilinear elements. The only terms that require special consideration are those that involve the double-layer kernel, since the kernel has a singularity at $\boldsymbol{\zeta} = \boldsymbol{x}$. For a flat element, the kernel is evaluated to zero exactly at $\boldsymbol{\zeta} = \boldsymbol{x}$, and there is no contribution to the system matrix from the element containing both $\boldsymbol{\zeta}$ and $\boldsymbol{x}$ (the so-called on-diagonal terms).

For curvilinear elements the on-diagonal terms are no longer zero, and we need to evaluate the integral

$$\int_{\Delta} K_{ij}(\boldsymbol{x}, \boldsymbol{\zeta}) \varphi_i(\boldsymbol{x}) dS(\boldsymbol{x})$$

over the element Δ where $\boldsymbol{\zeta} \in \Delta$. Although the kernel takes the form $\hat{\boldsymbol{n}} \cdot \boldsymbol{rrr}/r^5$, where $\boldsymbol{r} = \boldsymbol{\zeta} - \boldsymbol{x}$, it is weakly-singular and is integrable. This is because on a Liapunov-smooth surface

$$\lim_{\zeta \to x} \hat{\boldsymbol{n}} \cdot (\boldsymbol{\zeta} - \boldsymbol{x}) = 0.$$

Its numerical evaluation by a standard Gaussian quadrature would require a high-order integration rule to give accurate results and would prove computationally expensive. Fortunately, there is a simple and much more accurate method of obtaining these terms. Realizing that (5) must apply to any set of boundary conditions, we are free to choose any trivial solution for which we know $\boldsymbol{\varphi}$ exactly and then obtain the on-diagonal terms indirectly. This well known technique is the preferred method for many boundary value problems and is documented in standard BEM texts ([5], [6]). The simplest solution to choose in our case is that due the null motion of the particles (with no forces and torques acting on them), that is the eigenfunctions for the double-layer operator. Since $\{\boldsymbol{e}_l\}$ are the null solutions of $1 + \mathscr{K}$, one has

$$\int_{S_n} K_{ij}(\boldsymbol{x}, \boldsymbol{\zeta}) dS(\boldsymbol{x}) + \delta_{ij} = 0,$$

where S_n contains Δ. Thus, the contribution from the element Δ is

$$-\delta_{ij} - \int_{S_n - \Delta} K_{ij}(\boldsymbol{x}, \boldsymbol{\zeta}) dS(\boldsymbol{x}).$$

Integrals involving the double-layer kernel when $\boldsymbol{\zeta} \notin \Delta$, may be evaluated using standard Gaussian quadratic rules. We use first-order, 2×2 and 3×3 quadrature, as well as an adaptive

integration scheme (up to 64-point quadrature) on the double-layer operator. The order of the quadrature for the adaptive scheme depends on the relative distance from the collocation point to the centroid of the element containing the field point to the maximum side of the element. It is surprising to find out that the improvement with the higher-order quadrature is only marginal. We will quantify this in a later section.

After a boundary solution $\boldsymbol{\varphi}$ is obtained, the rigid-body motions of the particles are extracted by using (9).

Finally, the boundary solution can also be used in (6) to find the velocity field (and stress field, if need be, by taking appropriate gradients) at an integral point to $\mathscr{D}$. The numerical implementation is done in Fortran, in single precision arithmetics.

4 Numerical results

4.1 Flow past a sphere

The unbounded flow past a sphere is included here as a means for testing the code. The sedimentation velocity of the sphere is simply given by $\boldsymbol{F}/6\pi\eta a$, where $\boldsymbol{F}$ is the force acting on the sphere and a is its radius. To model the sphere using TR1A3 elements we start with a 20-faced icosahedron. To obtain finer discretisations, the faces of the icosahedron may be further subdivided into equilateral triangular elements, the vertices of which are projected radially onto the sphere's surface. For curvilinear elements, we use Fan and Yeow's method [37] albeit slightly modified. Firstly, the faces of cube are discretized into QUAD9 parent elements. These parent elements are then mapped to the surface of a sphere by projecting radially their nodes onto the sphere's surface. Having specified the number of divisions along each side of the cube, the nodes on each face may be obtained from interpolation over the parent element. This method ensures that the elements on the sphere's surface will be approximately equal in size. Figure 1 shows some typical meshes used in the calculations.

In Fig. 2 the percentage error in the sedimentation velocity and the CPU time are plotted against the number of elements for a unit sphere discretized with QUAD9 elements. Two different quadrature schemes are compared, a 1-point and a 3×3 scheme. It is clear that the differences between quadrature schemes are slight, with the 1-point scheme surprisingly giving the best results and at considerable savings in CPU time. In all cases, we find that the convergence rate for the sedimentation velocity is at least quadratic, i.e., the error decreases as $O(1/N^2)$, where N is the number of elements (note that the elements are about equal in size). In fact, a regression analysis gives a slope of -2.01 and -1.71 for the error curves of the 1-point and 3×3 schemes respectively. Also, a regression analysis for the last 6 points (the linear portion) on the CPU curves shows that the computational cost is approximately $O(N^2)$ with slopes of 1.91 and 1.98 for the 1-point and 3×3 schemes respectively.

The total CPU time of our program can be divided amongst a number of solutions stages primarily requiring of $O(N^2)$ or of $O(N)$ operations. There are two $O(N^2)$ operations which require

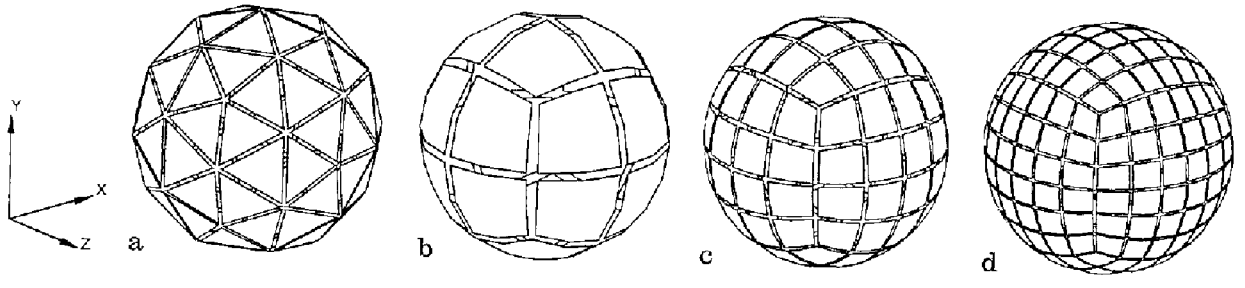

Fig. 1. **a** 80 TRIA3 element sphere; **b** 24 QUAD9 element sphere; **c** 96 QUAD9 element sphere; **d** 864 QUAD9 element sphere. These elements are shrunk for visual clarity

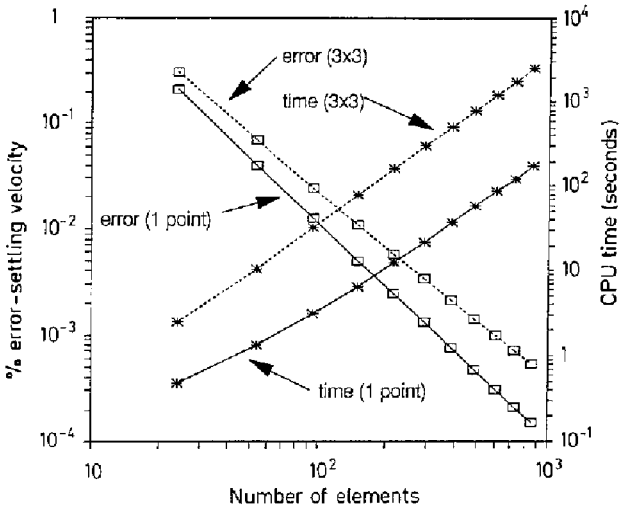

Fig. 2. Unbounded flow past a sphere, errors and CPU time (for the Titan workstation), for different order quadratures and Lagrangian elements. Note that the differences between the first-order quadrature scheme and higher-order schemes are negligible

careful coding to achieve good program performance. The first is the evaluation of the integrals needed to form the system of equations and the second is the iteration stage whereby we solve the matrix equations. The remaining operations such as input/output, computation of particle and element geometric properties and various book keeping tasks are of $O(N)$ and are therefore not significant contributors to the total CPU time. To determine the proportion of CPU time spent in the various solution stages of our program, we considered again the flow past a sphere discretized with a modest number of elements. In all, 1014 QUAD9 elements were used on the sphere and various schemes for integrating the double-layer kernel were compared. Table 1 shows the breakup of the CPU time for the different integration schemes. Clearly, the time spent in performing the integrations accounts for the majority of the program's CPU time, and a comparison of the different integration schemes show the adaptive and 1-point schemes to be the most efficient. These timings would suggest that a 1-point scheme should be used in preference to the other schemes, but, as will be shown later, achievement of accurate results in more complicated flow situations demands higher-order schemes. In all cases the number of iterations required was 4, owing to the efficient deflation of the extreme eigenvalues.

The rate of convergence of the sedimentation velocity for spheres discretized with TRIA3 elements is similar, but the errors are considerably larger for the same number of elements; for an 80 TRIA3 element sphere, the error is about 1% as compared to an error of 0.2% for a 24 QUAD9 element sphere. A problem as large as 20000 constant elements has been done on the Titan without difficulty. By extrapolation using standard bench-marks (e.g., LINPACK), even larger systems of the order 200000 elements can be solved on present day supercomputers.

Table 1. Breakup of the CPU time for the various integration schemes for the computation of the sedimentation velocity of a sphere. The timings are for an IBM RS6000 model 540 workstation

Integration Order	CPU time (seconds)		
	Integration	Iteration	Other
Adaptive	195.8	6.6	6.3
1×1	115.4	6.8	6.3
2×2	1071.1	6.6	6.3
3×3	2281.8	6.9	6.4

It is a pleasant surprise that the QUAD9 elements improve on the accuracy of the method significantly, while retaining the simplicity of the method (constant functional elements). This can be explained by the fact that the null functions play a central role in the method, and it is the geometries that determine these null functions—modelling the geometries more accurately would then help improving the overall accuracy of the solution.

The exact solution to the flow past a sphere near a plane interface has been provided by Lee and Leal [31], using a bipolar coordinate system. Here, the object is to compare the numerically predicted sedimentation velocity with the exact solution for a sphere translating normal and parallel to the plane boundary. Figs. 3a–b show the drag coefficients for a sphere moving normal and parallel to the plane interface. The drag coefficient is defined by

$$C_D = \frac{F}{6\pi\eta a U} = \frac{U_S}{U},$$

where $U_S = F/(6\pi\eta a)$ is the Stokes velocity in an unbounded flow, and U is the sedimentation of the sphere due to a force F applied to it. The drag coefficient is a function of z/a, where z is the distance from the interface to the sphere's centre. Unless the sphere is very near to the plane interface, both the TR1A3 and QUAD9 discretisations elements give good results for the drag coefficients. At $z/a = 1.4$, the 80 TR1A3 element sphere with 1-point quadrature gives an error of 4.5%, whereas the 24 QUAD9 element sphere yields an error of 0.41%, with 1-point integration, and 0.37%, with a 3 × 3 quadrature scheme. Very close to the interface, higher order integration schemes improve the results. At $z/a = 1.1$, the 96 QUAD9 element sphere with a 3 × 3 quadrature scheme yields an error of 16.5%; with 1-point quadrature, an error of 20.5% was obtained. Tables 2 and 3 show summarize our results for a range of heights above the plane.

It is clear from these Tables that the sphere discretized with TRIA3 elements and 1-point quadrature would be able to predict the particles' motions well, provided that these particles are not too close to the plane interface (about two radii away from the plane interface). The convergence rate for the sedimentation velocity of the QUAD9 element discretisation is now approximately linear, as Fig. 4 shows. Again, the 1-point integration scheme gives the best results, if the number of elements is large. The computational cost is still of $O(N^2)$.

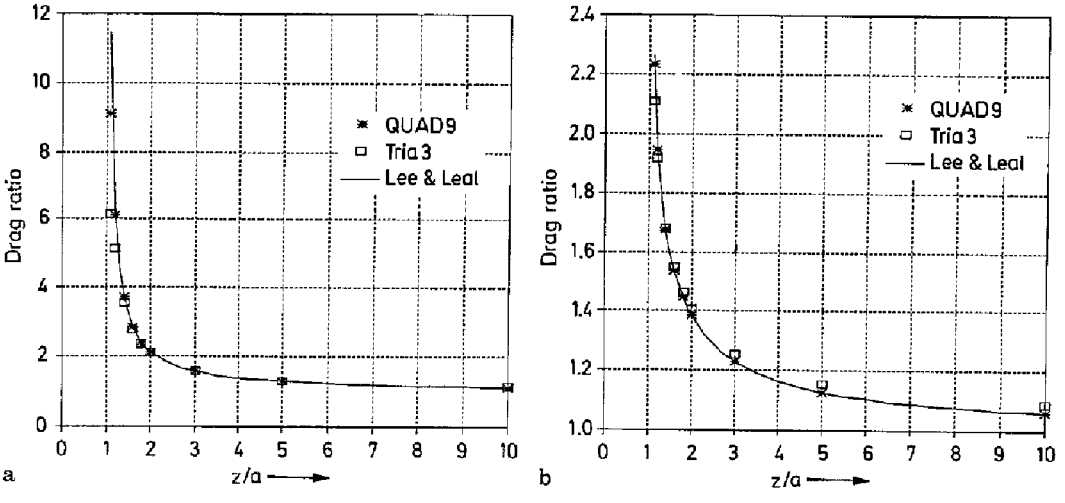

Fig. 3 a and b. Drag coefficients **a** for a sphere translated normal to the plane interface; **b** for a sphere translated parallel to the plane interface.

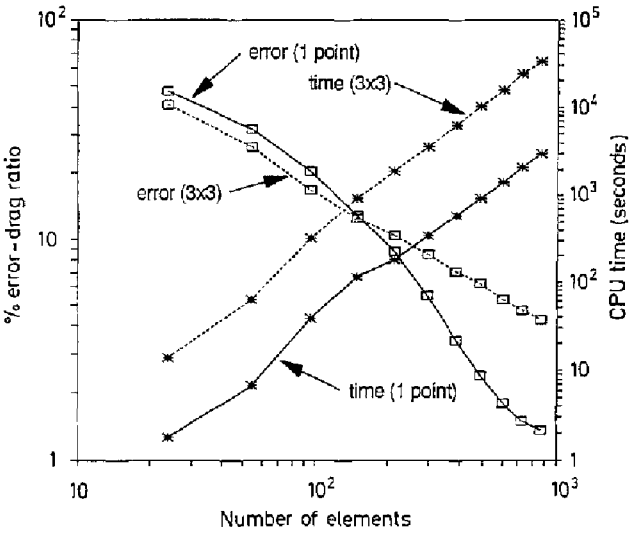

Fig. 4. Convergence rate of the sedimentation velocity and CPU time (for the Titan workstation) for QUAD9 elements at $z/a = 1.1$

Table 2. Drag coefficient for a sphere translating normal to the plane interface for an 80 TRIA3 element sphere with 1-point quadrature and 96 QUAD9 element sphere with 1-point and 3 × 3 quadrature; C_D^* is the exact solution from Lee and Leal [30]

z/a	C_D^*	80 TRIA3 1-point	96 QUAD9 1-point	96 QUAD9 3 × 3
1.1	11.4592	6.1533	9.1082	9.5707
1.2	6.3409	5.1362	6.1226	6.1026
1.4	3.7356	3.5728	3.7203	3.7216
1.6	2.8489	2.7828	2.8428	2.8495
1.8	2.3988	2.3643	2.3954	2.3998
2.0	2.1255	2.1083	2.1244	2.1273
3.0	1.5692	1.5827	1.5692	1.5700
5.0	1.2851	1.3098	1.2853	1.2855
10.0	1.1262	1.1548	1.1263	1.1265

Table 3. Drag coefficient for a sphere translating parallel to the plane interface for an 80 TRIA3 element sphere with 1-point quadrature and 96 QUAD9 element sphere with 1-point and 3 × 3 quadrature; C_D^* is the exact solution from Lee and Leal [30]

z/a	C_D^*	80 TRIA3 1-point	96 QUAD9 1-point	96 QUAD9 3 × 3
1.1	2.2643	2.1113	2.2327	2.2174
1.2	1.9527	1.9177	1.9480	1.9403
1.4	1.6755	1.6800	1.6748	1.6733
1.6	1.5344	1.5479	1.5340	1.5339
1.8	1.4452	1.4632	1.4449	1.4451
2.0	1.3828	1.4035	1.3827	1.3830
3.0	1.2272	1.2533	1.2272	1.2275
5.0	1.1259	1.1544	1.1260	1.1261
10.0	1.0595	1.0892	1.0596	1.0598

4.2 *Trajectories calculations*

The calculations of the particles' trajectories with the standard BEM have been done by some of us ([16]–[18]), Ingber [19]–[20], and Dingman et al. [40]. In these calculations, the particles' inertia is neglected and their positions are updated in time using the numerically predicted values of the translational and rotational velocities of each particle. In effect, each particle is moved through a small time step in the direction of the translational velocity and given a small, finite rigid rotation about the angular velocity axis. This results in a new configuration at which the translational and angular velocities of each particle can be re-calculated. Ingber [20] and Dingman et al. [40] use a high-order Runge–Kutta scheme to update the positions of the particles, while we find that the simpler Euler's scheme is sufficient (halving time step produces little changes in the position, provided that the particles move about $0.1a$ in each time step, [18]). This may be due to the fact that the particles' velocities are smooth functions of time, except when they are near-touching. The current time step is halved, if necessary, to prevent particles from overlapping. The two trajectory problems reported below have also been studied by Dingman et al. [40] using a standard BEM.

Three spheres. Caflisch et al. [32] find that periodic solutions are possible in the sedimentation of three spheres, if the three spheres are initially located at the vertices of an isosceles triangle. This periodicity is certainly lost in the presence of the interface.

Figures 5a–b show the trajectories of three spheres (of same radius a), initially locating at the vertices of an isosceles triangle of side $2\sqrt{3}a$ at a distance of $50a$ from the plane interface and sedimenting toward the interface (gravity acceleration is in the negative z axis); initially, the coordinates of the centers of these spheres are $(2, 0, 50)$ (sphere 1), $(-1, \sqrt{3}, 50)$ (sphere 2), and $(-1, -\sqrt{3}, 50)$ (sphere 3). As they move nearer to the interface, these spheres spread out, but still retain the configuration of an isosceles triangle. The projections of movement onto the x-direction of sphere 1 is about $0.7a$, and that of spheres 2 and 3 is about $0.3a$.

With the initial locations of the three spheres changed to destroy the symmetry with respect to the plane interface, the trajectories of the three spheres are no longer trivial. The three spheres are still initially at the vertices of a isosceles triangle of side $2\sqrt{3}a$ ($a = 1$), but their centers are

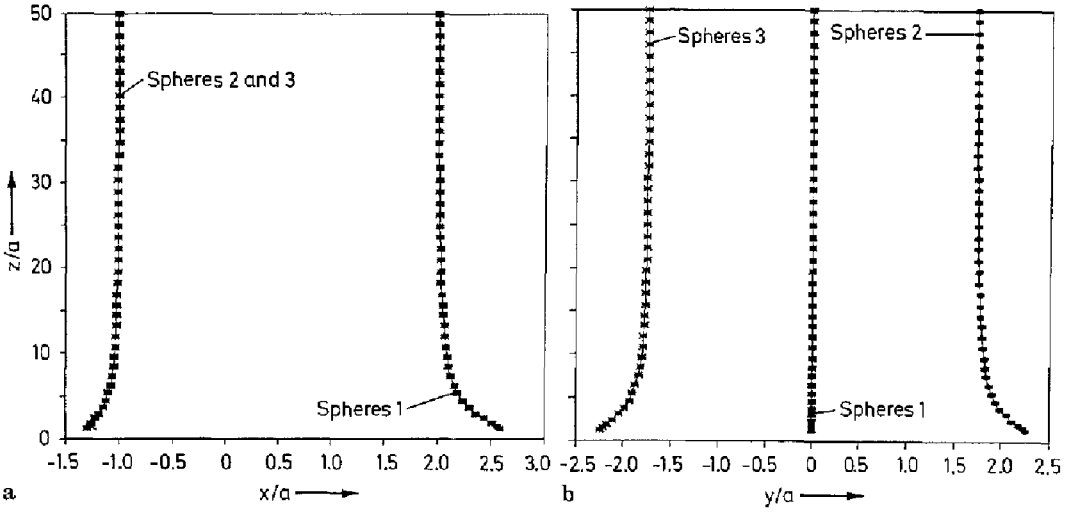

Fig. 5a and b. The trajectories of three 80 TRIA3 element spheres; **a** in the xz plane, and **b** in the yz plane. The three spheres are initially located at (2, 0, 50) (sphere 1), $(-1, \sqrt{3}, 50)$ (sphere 2) and $(-1, -\sqrt{3}, 50)$ (sphere 3). The spheres have the same radius ($a = 1$); initially the centre-to-centre distance is $2a\sqrt{3}$. Results from constant element scheme with time step 0.025. Halving the time step produces results identical to three significant figures

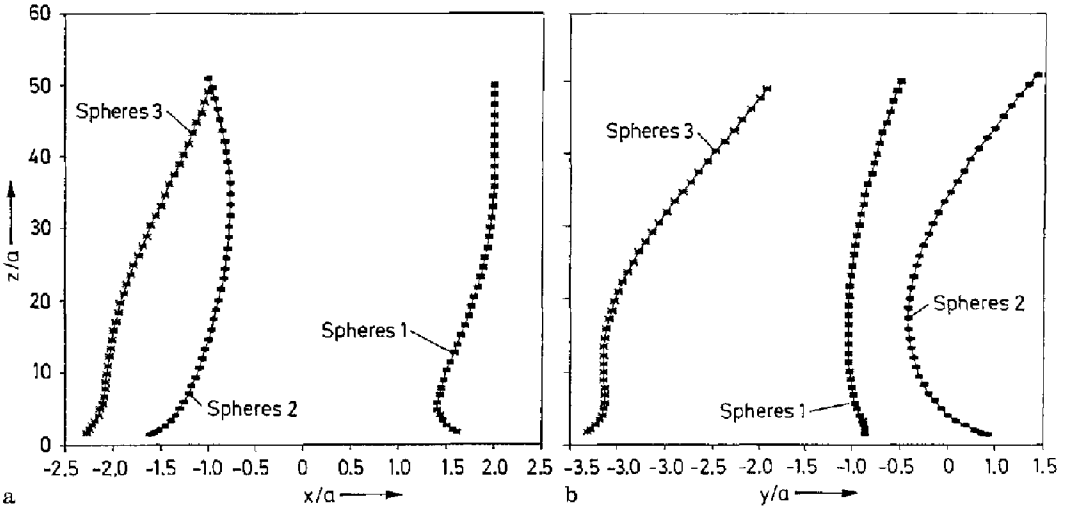

Fig. 6 a and b. The trajectories of three 80 TRIA3 element spheres; **a** in the xz plane, and **b** in the yz plane. The three spheres are initially located at (2, 0, 50) (sphere 1), $(-1, \sqrt{2}, 51)$ (sphere 2) and $(-1, -\sqrt{2}, 49)$ (sphere 3). The spheres have the same radius ($a = 1$); initially the centre-to-centre distance is $2a\sqrt{3}$. Results from constant element scheme with time step 0.025. Halving the time step produces results identical to three significant figures

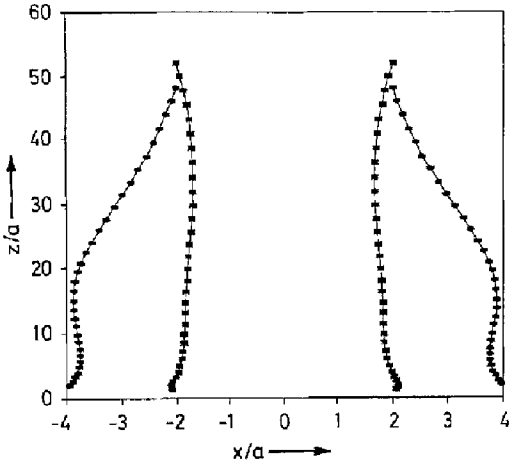

Fig. 7. The trajectories of eight 80 TRIA3 element spheres in the xz plane. The yz view cannot be visually distinguished from the xz plane (due to symmetry). Initially the spheres (of radius $a = 1$, centre-to-centre distance is $4a$) are located at the vertices of a cube, one face of which is parallel to the interface. They sediment toward the interface. Results from constant element scheme with time step 0.05

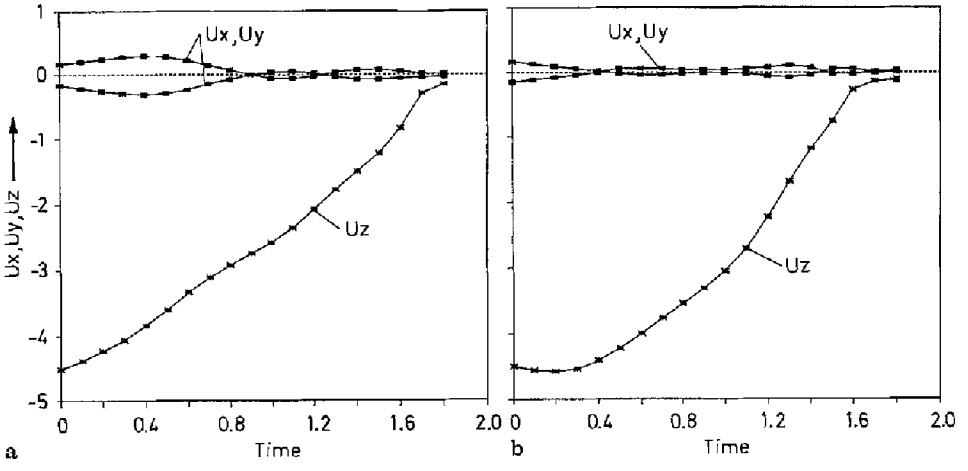

Fig. 8 a and b. The translational velocity components of **a** the top four spheres and **b** the bottom four

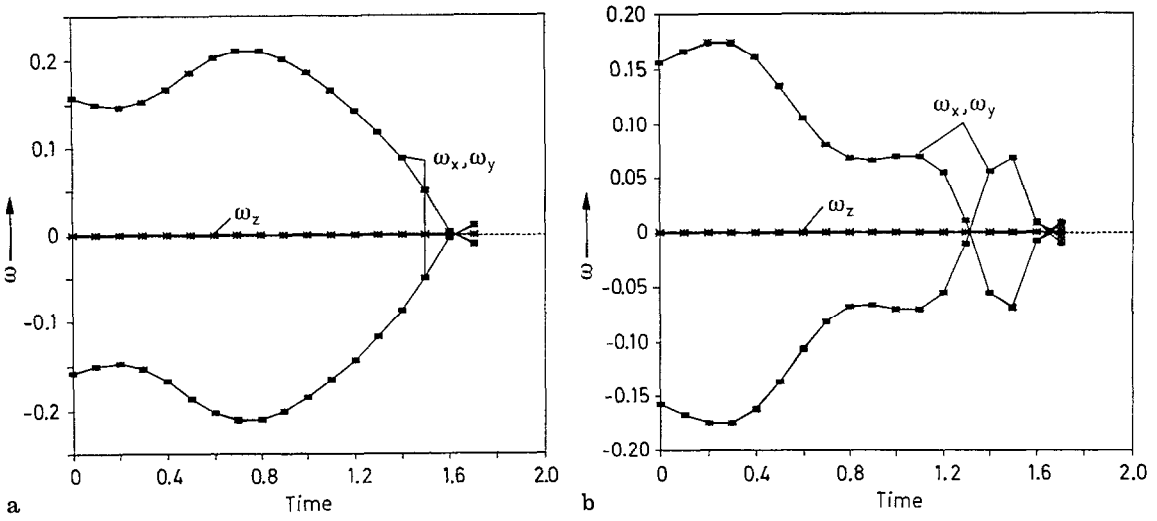

Fig. 9 a and b. The angular velocity components of **a** the top four spheres and **b** the bottom four

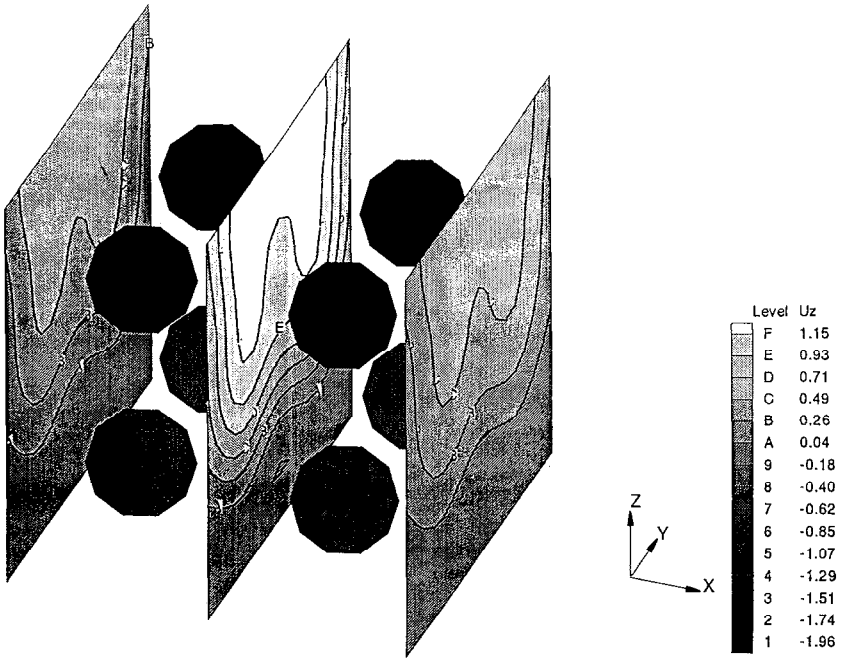

Fig. 10. The velocity contours in three selected planes. The spheres are at $(\pm 2, \pm 2, 6)$, $(\pm 2, \pm 2, 2)$

located at $(2, 0, 50)$ (sphere 1), $(-1, \sqrt{2}, 51)$ (sphere 2) and $(-1, -\sqrt{2}, 49)$ (sphere 3). The projections of movement onto the y-direction of sphere 3 is largest, about $2a$ (Fig. 6b).

Eight spheres. Durlofsky et al [28], Vincent et al. [18], and Dingman et al. [40] described the trajectories of eight spheres located initially at the vertices of a cube in an unbounded body fluid. Initially the top four spheres move inward and speed up while the bottom four spheres move outward and slow down. The top four eventually pass the bottom four and this results in an inversion of the cube.

In the presence of the interface, the periodicity of the motion is again lost. Initially the spheres are located at $(\pm 2, \pm 2, 52)$ (top four spheres), and $(\pm 2, \pm 2, 48)$ (bottom four spheres). The center-to-center distance is initially $4a$ ($a = 1$), and gravity is in the negative z direction. The top four

spheres do catch up with the bottom four, after sedimenting a vertical distance of about $30a$, but they then slow down as the interface is approached (Fig. 7). The lateral movement of the top four spheres is negligible compared to that of the bottom four spheres (about $2a$).

Figures 8a–b, and 9a–b display the translational and angular velocities components as functions of time. Note that the numerical scheme preserves the symmetry of the motion well, despite the fact that the element meshes are not created symmetrically (we simply create an 80-element sphere and then translated this to eight locations in space).

Finally, to illustrate that the method is also capable of yielding good results for the internal points to $\mathscr{D}$, we plot the surface contours of a velocity component of the eight sphere sedimentation problem in Fig. 10 at three selected planes normal to the interface. The symmetry of these contours is evident, as demanded by the locations of the spheres.

5 Conclusion

The CDL-BIEM is a powerful method for solving the mobility problem of a large number of particles. Although we only report results for modest number of particles, simulations of a swarm of spheres (up to 126 spheres, ≈ 10000 elements) have been performed without difficulty. Spatial methods such as the Finite Element or Finite Difference Methods cannot handle such a large problem due to a heavy demand on computing resources. The CDL-BIEM method can be adapted to different kernels, such as the half-space kernel as reported here. With a container surface, the null functions are somewhat more complicated, but the theoretical framework for the method is still very much the same [1]. Particles with complex shapes can also be accommodated, with a slight modification of the method [25]. With the emergence of parallel computing architectures (for which iterative schemes are well suited), the method should proved more powerful [41], and perhaps provides an analogue to the molecular simulation.

Acknowledgements

This research was funded by an Australian Research Council Grant (to NPT), and NFS Grant CTS-9015377 and an ACS Petroleum Research Fund Grant (to SK).

Appendix: Singularity solution in half-space

The singularity solution (i-component velocity) at point $\boldsymbol{x}$ due to a unit point force in the j-direction acting at $\boldsymbol{X}\in\mathscr{D}$ is given by, for the case where the Poisson's ratio is 0.5, [33], [35],

$$G_{ij}(\boldsymbol{x},\boldsymbol{X}) = \frac{1}{8\pi}\left\{\left(\frac{1}{r}-\frac{1}{R}\right)\delta_{ij} + \frac{r_i r_j}{r^3} - \frac{R_i R_j}{R^3} + \frac{2X_3}{R^3} \cdot\left[\delta_{j3}R_i + \delta_{i3}R_j - 2\delta_{i3}\delta_{j3}R_3 + x_3\left(2\delta_{i3}\delta_{j3} - \delta_{ij} + \frac{3R_i}{R^2}(R_j - 2\delta_{j3}R_3)\right)\right]\right\}, \tag{13}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{X}$, and $\boldsymbol{R} = \boldsymbol{x} - \boldsymbol{X}^*$ in which $\boldsymbol{X}^*$ is the reflected image of point $\boldsymbol{X}$ through the plane $x_3 = 0$. We note that this singularity solution consists of two Stokeslets, one at $\boldsymbol{X}$, and the other at $\boldsymbol{X}^*$, plus some extra terms (inside the square brackets).

The associated traction is given by

$$H_{ik}(\boldsymbol{x},\boldsymbol{X}) = H^S_{ik} - H^{S^*}_{ik} + H^E_{ik},$$

where H^S_{ik} and $H^{S^*}_{ik}$ are the tractions due to the two Stokeslets, and H^E_{ik} is the extra part. They are given by

$$T^S_{ik} = -\frac{3}{4\pi}\frac{n_m r_m r_i r_k}{r^5}; \qquad T^{S^*}_{ik} = -\frac{3}{4\pi}\frac{n_m R_m R_i R_k}{R^5}, \tag{14, 15}$$

and

$$T^E_{ik} = \frac{3X_3}{2\pi R^5}\Bigg\{ -\delta_{k3}R_iR_mn_m + R_3n_i(2\delta_{k3}R_3 - R_k) + x_3\Bigg[\delta_{ki}R_mn_m + R_in_k - 2\delta_{k3}(\delta_{i3}R_mn_m + R_in_3) + \left(5\frac{R_mn_mR_i}{R^2} - n_i\right)(2\delta_{k3}R_3 - R_k)\Bigg]\Bigg\}. \quad (16)$$

Note that the only singular terms are those associated with the Stokeslet solution at $\boldsymbol{X}$.

References

1. Kim, S.; Karrila, S. J. (1991): Microhydrodynamics: Principles and Selected Applications. Boston: Butterworth-Heinemann
2. Fichera, G. (1961): Proc. Conf. P.D.E. and Continuum Mechanics. 55–80
3. Rizzo, F. J. (1967): J. Appl. Math., 25, 83–95
4. Hess, J. L.; Smith, A. M. O. (1967): Prog. in Aeronautical Sciences. D. Kucheman (Ed.), London: Pergamon 1–138
5. Banerjee, P. K.; Butterfield, R.: Boundary Element Methods in Engineering Science, London: McGraw Hill
6. Brebbia, C. A.; Telles, J. C. F.; Wrobel, L. C. (1984): Boundary Element Techniques: Theory and Applications in Engineering. Berlin: Springer-Verlag
7. Youngren, G. K.; Acrivos, A. (1975): J. Fluid Mech. 69, 377–403
8. Bush, M. B.; Tanner, R. I. (1983): Int. J. Numer. Meth. Fluids 3, 71–92
9. Onishi, K.; Kuroki, T.; Tanaka, M. (1984): Eng. Anal. 1, 122–127
10. Bush, M. B.; Tanner, R. I. (1990): Boundary Element Methods in Nonlinear Fluid Dynamics, Vol. 6, 285–317. Eds. P. K. Banerjee and J. Morino, New York: Elsevier Applied Science
11. Telles, J. C. F.; Brebbia, C. A. (1981): Int. J. Solids Struct. 17, 1149–1158
12. Tran-Cong, T.; Phan-Thien, N. (1986): Computational Mech. 1, 259–268
13. Ascoli, E. P.; Dandy, D. S.; Leal, L. G. (1989): Int. J. Numer. Meths. Fluids 9, 651–688
14. Mindlin, R. D. (1936): Physics 7, 195–202
15. Phan-Thien, N. (1983): J. Elasticity 13, 231–235
16. Tran-Cong, T.; Phan-Thien, N. (1989): Phys. Fluids A 1, 453–461
17. Tran-Cong, T.; Phan-Thien, N.; Graham, A. L. (1990): Phys. Fluids A 2, 666–673
18. Vincent, J.; Phan-Thien, N.; Tran-Cong, T. (1991): J. Rheology 35, 1–27
19. Ingber, M. S. (1989): Int. J. Num. Method. Fluids 9, 263–273
20. Ingber, M. S. (1990): Int. J. Num. Method. Fluids 10, 791–809
21. Power, H.; Miranda, G. (1987): SIAM J. Appl. Math. 47, 689–698
22. Karrila, S. J.; Fuentes, Y. O.; Kim, S. (1989): J. Rheology 33, 913–947
23. Karrila, S. J.; Kim, S. (1989): Chem. Eng. Comm. 82, 123–161
24. Karrila, S. J.; Fuentes, Y. O.; Kim, S. (1989): J. Rheology 33, 913–947
25. Pakdel, P.; Kim, S. (1991): J. Rheology 35, 797–823
26. Mazur, P.; van Saarloos, W. (1982): Physica A 115, 21–57
27. Mazur, P. (1985): Can J. Phys. 63, 24–29
28. Durlofsky, L.; Brady, J. F.; Bossis, G. (1987): J. Fluid Mech. 180, 21–49
29. Brady, J. F.; Bossis, G. (1988): Ann. Rev. Fluid Mech. 20, 111–157
30. Happel, J.; Brenner, H. (1983): Low Reynolds Number Hydrodynamics, Dordrecht, The Netherlands: Martinus Nijhoff Publishers
31. Lee, S. H.; Leal, L. G. (1980): J. Fluid Mech. 98, 193–224
32. Caflisch, R. E.; Lim, C.; Luke, J. H. C.; Sangani, A. S. (1988): Phys. Fluids 31, 3175–3179
33. Blake, J. R. (1971): Proc. Camb. Phil. Soc. 70, 303–310
34. Hasimoto, H.; Sano, O. (1980): Ann. Rev. Fluid Mech. 12, 335–363
35. Phan-Thien, N. (1983): J. Elasticity 13, 231–235
36. Odqvist, F. K. G. (1930): Math. Z. 32, 329–375
37. Xijun, F.; Yeow, Y. L. (1991): preprint
38. Ramkrishna, D.; Amundson, N. R. (1985): Linear Operator Methods in Chemical Engineering, Englewood Cliffs, New Jersey: Prentice-Hall
39. Bathe, K.-J. (1982): Finite Element Procedures in Engineering Analysis, Englewood Cliffs, New Jersey: Prentice-Hall
40. Dingman, S.; Ingber, M. S.; Mondy, L.; Abbott, J. (1991): preprint
41. Fuentes, Y. O.; Kim, S. (1991): Rheology Research Report RRC 127, University of Wisconsin-Madison

Communicated by R. I. Tanner, August 5, 1991