Viscoelastic Mobility Problem Using A Boundary Element Method

Nhan Phan-Thien and Xi-Jun Fan
1Division of BioEngineering
The National University of Singapore
Singapore 119260
2Institute of High Performance Computing
Singapore, 118261

September 2001

Abstract

In this paper, the complete double layer boundary integral equation formulation for Stokes flows is extended to viscoelastic fluids to solve the mobility problem for a system of particles, where the non-linearity is handled by particular solutions of the Stokes inhomogeneous equation. Some techniques of the meshless method are employed and a point-wise solver is used to solve the viscoelastic constitutive equation. Hence volume meshing is avoided. The method is tested against the numerical solution for a sphere settling in the Oldroyd-B fluid and some results on a prolate motion in shear flow of the Oldroyd-B fluid are reported and compared with some theoretical and experimental results.

1 Introduction

In the mobility problem, one wishes to find the rigid body motions of a group of particles in a fluid, given the external forces/torques acting on them, and/or the ambient flow they are subjected to. Mobility problems are central in understanding particle interactions with the surrounding and consequently the evolution of microstructure in a complex fluid. The mobility problem in a viscous fluid is well understood and well documented, e.g., Goldman et al. [2], Kim and Karrila [1].

In a viscoelastic fluid, qualitatively different behaviour in the particle motion has been predicted and indeed observed, for example, Joseph et al. [3]. In the absence of inertia, a sphere settling in a viscous fluid parallel to a plane wall will not deviate from its vertical path. Inertia and weak elasticity (using the second-order fluid model) are predicted to push the particle away from the wall, in contrast to the observation of sphere moving toward the wall at a Deborah number of $O(1)$, Becker et al. [4]. There are indications that three-dimensionality is an important factor, in as much as some qualitatively different behaviour only occurs in three-dimensional flows. In a recent numerical simulation Singh and Joseph [5], using a fictitious domain method and the Oldroyd-B model, were able to show that the particle/wall interaction is indeed a three-dimensional effect, and that the sphere moves toward the wall to a preferred position (which does not occur for two-dimensional flows involving cylinders). This is due to normal (Joseph and Feng [6]), or shear stress contributions (Feng et al. [7]) from viscoelastic effects.

When the particle is slender, it tends to fall with its broad side parallel to gravity when viscoelastic effects dominate, and perpendicular to gravity when inertia dominates (Joseph and Liu [8], Huang et al. [9]). In a simple shear flow and if the fluid is viscous, a slender particle undergoes a periodic motion known as Jefferies’s orbit (Jeffery [10]), and this information has been used in constructing useful constitutive equations for fibre suspension (Hinch and Leal [11], Dinh and Armstrong [12], Folgar and Tucker [13], Phan-Thien and Graham [14]). It would be useful to understand how much if indeed viscoelasticity modifies Jeffery’s orbit. In order to investigate this, one needs a reliable code to solve the complex three-dimensional mobility problem in a viscoelastic fluid (given the driving force and ambient flow), calculate the rigid-body motion of the particles in a viscoelastic fluid). There are several robust numerical methods based on the finite element techniques that can, in principle, be applied to this problem. However, the massive computation requirement coupled with the need to re-mesh the flow domain at every time step make these methods unattractive at the present time. Joseph and his co-workers have developed a very efficient technique based on a distributed Lagrange multiplier (DLM) and fictitious domain method [15], which...
they applied successfully to a variety of problems. We found the method indeed efficient, and produced qualitative good results, but may lack the precision required for a quantitative assessment. Concurrent with further developing the DLM method, we also investigate an alternative technique based on an indirect boundary element method (BEM), the Completed Double Layer Boundary Element Method (CDLBEM, Kim and Karrila [1], Phan-Thien and Kim [16]). This is suitably modified for viscoelastic flow calculation. The main attraction of the boundary element method is a reduction in the dimensionality of the problem (only a three-dimensional surface mesh needs be generated), and translating and rotating mesh to a new position can be accomplished without re-meshing. Non-linearities associated with viscoelasticity, modelled by the Oldroyd-B fluid, are handled by the particular solution method [17] using a number of moving points, in the same spirit as the meshless method [18] [19]. In this paper, we report a general method for solving the mobility problem in an Oldroyd-B fluid, but other models could have been chosen. We start with a description of the method, followed by a detailed implementation. Code validation is done with the flow past a sphere, where comparison with axisymmetric results of Tiefenbruck and Leal [20] is made. The results for shear flow past an ellipsoid are then presented.

2 Formulation

The motion of an incompressible viscoelastic fluid is governed by the following equations

\[ r \; \frac{\partial u}{\partial t} + \nabla p = \nu \; \nabla^2 u \; x \; 2 \; V; \]

where \( V \) is the flow domain, \( \nu \) is the fluid density, \( u \) is the velocity field, and \( D = \frac{\partial u}{\partial t} \) is the material time derivative. To these equations, some relevant boundary/initial conditions are imposed. For the mobility problem, one has a finite number \( M \) of rigid inclusions in the viscoelastic fluid, labelled \( n = 1; \ldots; M \); and the external force \( F^n \) and torque \( T^n \) on a particle \( n \) are given,

\[ F^n = \int_{S_n} t(y) ds(y) \; \; \; T^n = \int_{S_n} (x - x^{(n)}_c) \times t(y) ds(y); \]

where \( S_n \) is its bounding surface, \( t = \frac{\partial u}{\partial n} \) is the surface traction, \( n \) is the outward unit vector on \( S_n \); and \( x^{(n)}_c \) is the mass centre of particle \( n \).

It is required to find the velocity field everywhere in \( V \); and in particular, the rigid body motion of particle \( n \)

\[ u(x) = U^n + \omega^n \times x \; \; \; x \; 2 \; S_n; \]

where \( U^n \) is the translational and \( \omega^n \) is the angular velocity of \( n \).

The total stress tensor in a viscoelastic fluid may be arbitrarily decomposed as

\[ \frac{\partial \sigma}{\partial t} = \frac{\partial \sigma^N}{\partial t} + \frac{\partial \sigma^v}{\partial t}; \]

where \( \frac{\partial \sigma^N}{\partial t} \) is a Newtonian stress tensor (which is usually, but not necessarily, the “solvent contribution”),

\[ \frac{\partial \sigma^N}{\partial t} = \frac{\partial p}{\partial t} + \nabla \cdot \; \left( \frac{\partial \sigma^N}{\partial t} \right) \; \; \; \frac{\partial \sigma^v}{\partial t} = \frac{\partial \sigma^N}{\partial t}; \]

with \( p \) the hydrostatic pressure, \( \nu \) a conveniently chosen viscosity value, \( \nabla \cdot \) the velocity gradient, and \( \frac{\partial \sigma^v}{\partial t} \) is the “polymer contribution” (the remaining part of the total stress tensor), and is given by a suitable constitutive equations. With this stress splitting, the balance of momentum becomes

\[ r \; \frac{\partial \sigma^N}{\partial t} + \frac{\partial \sigma^v}{\partial t}; \]

2.1 CDLBEM Formulation for Viscoelastic Flow

When inertial and viscoelastic terms are negligible compared to the Newtonian contribution \( r \; \frac{\partial \sigma^N}{\partial t} \), Eq. (7) is linear and can be solved by the conventional boundary element method, given some suitable boundary conditions. The advantage of the BEM is that it reduces the dimensionality of the problem by one. For
a three-dimensional problem it reduces to a three-dimensional surface computational domain, rather than a full three-dimensional domain, and therefore avoids volume meshing. Re-meshing after every time step is also avoided; the surface mesh can simply be translated and rotated with the particles. When inertial and viscoelastic terms are moderate in size, the terms containing the velocity gradients and stress fields on the right hand side of the momentum equation are usually regarded as known pseudo-body forces in an iteration process, and (7) becomes an inhomogeneous differential equation. The pseudo-body force terms are accounted in the boundary element formulation as volume integrals, which are evaluated based on the velocity and stress fields obtained at the previous iteration (Tran-Cong and Phan-Thien [25]). A volume mesh is usually required for this purpose and therefore negates the gain in the reduction of dimensionality.

An alternative is to use a particular solution of the Navier equations to replace the volume integration in the boundary integral formulation (Coleman et al. [17], Zheng et al. [26], Nguyen-Thien [27]). This particular solution can be expressed analytically and volume integration can be avoided. However, we have to be careful that the particular solution used is equal to the volume integral terms, otherwise convergence to the correct solution is not guaranteed. The main principle is that the general solution of the inhomogeneous equation can be treated as the superposition of a general solution of the homogeneous equation and a particular solution of the inhomogeneous equation. The solution of the flow problem is obtained, when the prescribed boundary conditions are satisfied. Hence the solution of Eq. (7) can be decomposed as

\[ \frac{\partial}{\partial t} \mathbf{u}^{H} = \mathbf{u}^{P} + \mathbf{u}^{V}; \]  

(8)

where \( \mathbf{u}^{H} \) is the solution of the homogeneous equation:

\[ r \cdot \frac{\partial}{\partial t} \mathbf{u}^{H} = 0; \]  

(9)

and \( \mathbf{u}^{P} \) is a particular solution of the inhomogeneous equation:

\[ r \cdot \frac{\partial}{\partial t} \mathbf{u}^{P} = r \cdot \mathbf{u}^{V} + 3 \frac{Du}{Dt}; \]  

(10)

where the right hand side is regarded as a known function. The total stress is

\[ \mathbf{\tau} = \mathbf{\tau}^{H} + \mathbf{\tau}^{P} + \mathbf{\tau}^{V}; \]  

(11)

and the traction at a point on the surface is

\[ t = t^{H} + t^{P} + t^{V}; \]  

(12)

where \( t^{H} = \mathbf{\tau}^{H} \cdot \mathbf{n} \) and \( \mathbf{u}^{H} \) are traction and velocity fields of the homogeneous solution, \( t^{P} = \mathbf{\tau}^{P} \cdot \mathbf{n} \) and \( \mathbf{u}^{P} \) are those of the particular solution, and \( t^{V} = \mathbf{\tau}^{V} \cdot \mathbf{n} \) is the contribution to the traction from the viscoelastic stress. The velocity field is

\[ \mathbf{u} = \mathbf{u}^{H} + \mathbf{u}^{V}; \]  

(13)

Boundary conditions can either be velocity and/or traction boundary conditions; these can be translated into the boundary conditions for the homogeneous solution

\[ \mathbf{u}^{H} \big|_{\text{S}} = \mathbf{u}^{P} \big|_{\text{S}}; \]

\[ \mathbf{t}^{H} \big|_{\text{S}} = \mathbf{t}^{P} \big|_{\text{S}}; \]

(14)

where \( \mathbf{u}^{P} \big|_{\text{S}} \) and \( \mathbf{t}^{P} \big|_{\text{S}} \) are the prescribed boundary conditions of the flow problems.

Since \( \mathbf{u}^{H} \) satisfies the homogeneous equation (9), the velocity field of the homogeneous solution can be expressed in terms of the double layer density \( \mathbf{v} \) (Kim and Karrila [1]),

\[ \mathbf{u}^{H}(x) = u^{1}(x) + \int_{S} K(x;y) \mathbf{v}(y) dS(y); \]  

(15)

where \( u^{1}(x) \) is the ambient deformation (i.e., the deformation in the absence of particles), and \( K(x;y) \) is the double layer kernel. When \( x \) is located on the surface \( S \); the double layer surfs a jump and the boundary integral equation for the homogeneous solution can be written as

\[ \mathbf{u}^{H}(x) \mid_{\text{S}} = u^{1}(x) + \int_{S} K(x;y) \mathbf{v}(y) dS(y); \]  

(16)
Using (13) and the no-slip boundary conditions on $S^*_n$, leads to
\[
U^n + \int E (x_i \ x_c^{(n)}) \ i \ u^p (x) \ i \ u^t (x) = \frac{Z}{s} (y) + K(x; y) \ \zeta' (y) dS(y); \quad \times 2 \ S_n: \quad (17)
\]

Using the completion process and de\’flation technique (Kim and Karrila [1]), we obtain the ..nal integral equation that is suitable to numerical implementation for multiparticle system without a container,
\[
(1 + H) = b;
\]
where
\[
H(\phi) = K(\phi + \sum_{i<k} D \ (k; i), \ E \ \zeta ;)
\]
where the sum is taken over $i = 1; \cdots ; 6; k = 1; \cdots ; M$; the angular brackets denote the natural product
\[
h_a b_i = \int a b dS;
\]
\[
\phantom{1} \ (k;i) \text{ is the normalized (with respect to the natural product) eigenvector of } K, \text{ representing the six rigid body motion modes (} i = 1; 2; 3 \text{ : translational, } i = 4; 5; 6 \text{ : rotational) of particle } k = 1; \cdots ; M; K \text{ is the double layer (integral) operator,}
\]
\[
K(\phi) = \sum_{i<k} K(x; y) \ \zeta' (y) dS(y); \quad (21)
\]
and $b$ is the known vector
\[
b = \int u^p \ i \ u^t \ i \ (k) = \frac{1}{2} \ (T(k) + T^p(k) + T^v(k) \ E r i \ G(x; x^{(k)}) \ )
\]
In (22), $F(k)$ and $T(k)$ are external force and torque acting on particle $k$; $F^p(k)$, $F^v(k)$, $T^p(k)$ and $T^v(k)$ are the contributions from the particular solution and the viscoelastic stress to the forces and torques on particle $k$:
\[
F^p(k) = \int Z^p(y) dS(y); \quad T^p(k) = \int Z^{(k)}(x_i \ x_c^{(k)}) \ E t^p(y) dS(y);
\]
\[
F^v(k) = \int Z^p(y) dS(y); \quad T^v(k) = \int Z^{(k)}(x_i \ x_c^{(k)}) \ E t^v(y) dS(y);
\]
and $G(x; x_c^{(k)})$ is the single layer kernel (Stokeslet):
\[
G(x; x_c^{(k)}) = \frac{1 + rr}{8\pi} \quad r = x_i \ x_c^{(k)};
\]
Here we assume the inertial forces and torques on all particles are negligible. The total forces and torques acting on particles consist of the components contributed from the external ..eld, particular solution and viscoelastic stress
\[
F^r(n) + F^p(n) + F^v(n) + F(n) = 0; \quad (25)
\]
\[
T^r(n) + T^p(n) + T^v(n) + T(n) = 0; \quad (26)
\]
In the completion process of the solution space, we need to use the force and torque on particle $n$, $F^H(n) = i \ F^H(n)$ and $T^H(n) = i \ T^H(n)$, for the homogeneous solution on the right hand of Eqs. (22) and (28).

The rigid body motion of particles can be extracted from the solutions of the double layer densities:
\[
U_i^{(k)} = i \ D \ i \ (k;i) ; \quad \iota((k)) = i \ i \ i \ D \ i \ (k;i+3) ; \quad i = 1; 2; 3;
\]
\[
(27)
\]
where \( S^{(k)} \) is the surface area of particle \( k \); and \( I^{(k)} \) represents the surface moment of area of \( k \).

When the double layer densities are known, the velocity at a field point \( x \) can then be expressed as

\[
\mathbf{u}(x) = \mathbf{u}^p(x) + \mathbf{u}^1(x) + \sum_k \mathbf{k}_{ij}^{(k)} \mathbf{j}(y) \mathbf{dS}(y) + \sum_k \frac{1}{2} \left( \mathbf{F}_{(k)}^p + \mathbf{F}_{(k)}^y \right) \mathbf{i} \mathbf{h} \mathbf{r} \mathbf{r} \mathbf{G}(\mathbf{x}; \mathbf{x}^{(k)}) \epsilon \left( \frac{\mathbf{r}}{8 \pi} \right):
\]

(28)

### 2.2 Particular Solutions

Now the problem reduces to how to obtain the particular solution of Eq. (10). If we denote the right side of this equation to be \( f(x) \), we obtain

\[
\sum_j \mathbf{f}_j = 0;
\]

(29)

The problem can be conveniently extended to linear elasticity the particular solution is sought from the Galerkin vector, \( \mathbf{G} \), where

\[
\sum_i \mathbf{G}_i + \mathbf{f}_i = 0;
\]

(30)

It is possible to obtain an analytical solution of Eq. (30) when \( f_i \) is a radial basis function (Coleman et al. [17]). Thus we assume that \( f_i \) can be approximated by a sum of radial basis functions \( \varphi \);

\[
\mathbf{f}_i = \sum_{n=1}^{N} \mathbf{G}^i \cdot \varphi_n \cdot \mathbf{A}(\mathbf{x}_i; \mathbf{x}_n);
\]

(31)

where \( \varphi_n \) are constants (determined by fitting), and seek a particular solution \( \mathbf{G}_i \) of the following form

\[
\mathbf{G}_i = \sum_{n=1}^{N} \mathbf{A}^i \cdot \varphi_n \cdot \mathbf{A}(\mathbf{x}_i; \mathbf{x}_n);
\]

(32)

With \( r = \| \mathbf{x}_i - \mathbf{x}_n \| \), where \( \mathbf{x}_n \) is a suitably chosen constant for point \( \mathbf{x}_n \) and usually equal to or larger than the distance to the closest neighbouring point. Introducing \( \mathbf{A}(r) = \mathbf{r} \mathbf{A}(r) \), a simple equation for \( \mathbf{A}(r) \) is obtained (Zheng et al. [26])

\[
\frac{d^i}{dr^i} \mathbf{A}(r) = r \mathbf{A}(r);
\]

(33)

For a given \( \mathbf{A}(r) \); it is not difficult to integrate the above equation and obtain a particular solution, \( \hat{\mathbf{A}}(r) \). Then \( \mathbf{G}_i \) is obtained from Eq. (32); \( \mathbf{u}^p \) can be determined from the Galerkin vector. The coefficients, \( \varphi_n \), are determined by solving a system of linear algebraic equations (31) based on the values of \( f = \mathbf{A} \) at \( N \) points. The functional form of \( \mathbf{A}(r) \) depends on the radial basis function chosen for \( \mathbf{A}(r) \). Several kinds of basis function have been investigated. We adopt the exponential basis function

\[
\mathbf{A}(r) = \exp(- \mathbf{r} \cdot \mathbf{r}^2)
\]

(34)

because it decays rapidly with \( r \), leading to a quick convergence when solving Eq. (31) for \( \varphi_n \) iteratively. With this choice of \( \mathbf{A}(r) \), a particular solution of Eq. (33) is

\[
\hat{\mathbf{A}}(r) = \mathbf{i} \frac{1}{8} \mathbf{r} + \frac{1}{2} \mathbf{p} \mathbf{r} \mathbf{r} \mathbf{p} \mathbf{r} \mathbf{r} + \exp(- \mathbf{r} \cdot \mathbf{r}^2) \cdot 2^i;
\]

(35)

where \( \mathbf{erf} \) is the error function.

### 2.3 Oldroyd-B Fluid

One of the popular constitutive equations for dilute polymer solutions is the Oldroyd-B fluid, which is a reasonable model for the Boger fluid in moderate shear rate regime. In this model, the stress tensor can
be split into two parts: a Newtonian plus a viscoelastic stress, as we did in Eq. (5). The latter can be expressed in the familiar Upper Convected Maxwell (UCM) equation:

$$
\dot{\varepsilon} + \frac{\varphi}{\eta} \dot{\varepsilon} \cdot \left( \frac{\partial}{\partial t} \dot{\varepsilon} + \mathbf{u} \cdot \nabla \dot{\varepsilon} \right) = (\mu - 1) \mathbf{I};
$$

(36)

where $\dot{\varepsilon}$ is the relative viscosity of polymer solutions, $\eta$ is the relaxation time of the fluid and $\mathbf{I} = \mathbf{u} \otimes \mathbf{u}$ is twice the strain rate tensor. Introducing the configuration tensor $C$

$$
\dot{C} = (\mu - 1) \mathbf{I}^{-1} \frac{\partial}{\partial t} (C \otimes C);
$$

(37)

and substituting it into the constitutive equation for the Oldroyd-B fluid, we have

$$
C + \frac{\mu}{\eta} \frac{\partial}{\partial t} \left( \mathbf{u} \otimes C \right) = \mathbf{I}:
$$

(38)

In the present simulation, we solve the Eq. (38) for the configuration tensor and then obtain viscoelastic stress through Eq. (37).

3 Numerical Methods

3.1 Field points

One of the important features of the present method is to avoid volume meshing in solving the constitutive equation and obtaining the particular solution in the flow domain. This makes the method flexible in dealing with problems with complex moving boundaries. Similar to the meshless methods (Belytschko et al. [18]; Duarte and Oden [19]; Oñate et al. [28]), we use distributed points in the computational domain instead of volume meshing to do numerical interpolation, differentiation and to solve differential equations. It is obvious that if the number of points is large, and if they are distributed evenly in the field, the solution would be more accurate and stable. However, the computation time would increase with the number of points. We are dealing with infinite flow domain problems, and it would require an infinite number of field points to be distributed in whole domain. Fortunately, for the boundary integral equations, what we need to know is just the values of the velocities and tractions on the boundary. We would like to reduce the number of points as many as possible, provided the solution is stable and accurate enough for the problems.
in hand. Hence we may distribute more points in the area relatively close to the boundaries to capture potentially large velocity and stress gradients there.

The particles will move along some trajectories during the simulation. We thus use a coordinate system which is located at and moves with the mass centre of the particle system and classify field points into two categories: the points xed relatively to the coordinate system and the points moving and rotating with each particle. We call the former the xed points and the latter the moving points. The xed points in a 2D case for an elliptical particle are sketched in Fig. 1. The moving points are represented by circles, which are distributed in a thin layer around the surface of particle. The xed points are represented by the plus signs and distributed in a larger area including inside the particle. The xed points can be either regularly or randomly distributed. From the above, it is seen that some points are located inside the particle. Hence we have to detect which points are inside the particle and disable them after moving the particle to a new position at each time step. To detect if a point is inside of an ellipsoid, we employed Perram’s contact function [29]. Some xed points, either coincident with, or too close to moving points, are all disabled in this process.

3.2 Fixed Least Square Method

The moving least square method (Lancaster and Salkauskas [30]) has been widely used in meshless methods, such as the element free Galerkin method (Belytschko et al. [18]), the reproducing kernel particle method (Swegle et al. [31]), and the finite point method (Oñate et al. [28]), to name a few. This is a locally fitted technique based on randomly distributed points. A similar technique called the xed least square is employed in the present method. The advantage of the xed least square method is its simplicity in calculating derivatives but its results are more sensitive to the support size of the support. We usually use monomials as the basis functions. In three-dimensional space (x1; x2; x3), they can be expressed in the local coordinates of y as

\[
\begin{align*}
\alpha^k = \begin{cases}
\alpha^1; x_1; x_2; x_3; a & k = 0; \\
\alpha^2; x_1; x_2; x_3; a & k = 1; \\
\alpha^3; x_1; x_2; x_3; a & k = 2; \\
\vdots & \vdots;
\end{cases}
\end{align*}
\]

where k is the order of the monomials. If the \( P_i(\|)_{i=1}^K \) points, x1, are linearly independent over the given N (N < K) points, x1, the coefficients, \( a_i(y) \), are determined by the least square method, i.e., to find \( a_i(y) \) such that

\[
\tilde{\mathbf{A}} = \mathbf{J} (a^\mu) = \sum_{i=1}^{N} \alpha^{\mu(y)} P_i(\|_1) \cdot \mathbf{u}(x_i) \cdot \mathbf{J} (a^\mu),
\]

where \( \|_1 \) is the position vector of point \( x_i \) in the local coordinates, and \( \phi \), \( \phi_y \) is a weighted inner product in the support of \( y, \|_y \):

\[
(u; v)_y = \sum_{i=1}^{N} \mathbf{u}(\|_1) W_i(y)v(\|_1);
\]

where \( W(y) \) is a weighted function, \( W(y) > 0 \) when \( x > 2 - y \), otherwise \( W(y) = 0 \); and \( W_i(y) \) is the value of \( W(y) \) at \( x_i \). Solving the inequality (42) requires the derivatives of \( \mathbf{J} (\mu) \) with respect to \( a_i(y) \) to be zero.
The method has its disadvantage as well. It defines the local approximation of a function in each support. When over the moving least square method and multi-fixed least square method. However, the fixed least square method avoids calculating the derivative of the weighting function and\( \mathbf{A}^{T} \mathbf{A}^{-1} \); when calculating the derivatives of local approximation of a function. This is an advantage of the fixed least square method. It can be seen that the weighting function for the fixed least square method we used here is constant and hence it does not require to calculate the derivative of the weighting function and \( \mathbf{A}^{T} \mathbf{A}^{-1} \); when calculating the derivatives of local approximation of a function. This is an advantage of the fixed least square method over the moving least square method and multi-fixed least square method. However, the fixed least square method has its disadvantage as well. It defines the local approximation of a function in each support. When
the supports belong to different points overlap and one point may belong to more than one supports, the interpolation and differentiation are multivalued according to the choice of the support. The decision has to be made to limit the choice of the support. In the present method, we only used the fixed least square method to interpolate the function to points which are very near to the data points, and to calculate the derivatives just right at the data points. It is easy to choose a right support which gives the best ...ting. The situation here is quite different from that of the element-free Galerkin method and other meshless method to solve differential equations.

During the simulation, it is frequently required to determine a support and create a list of all points within the support for each point. A support should contain enough points, at least larger than K, in order to guarantee the invertibility of the matrix A in Eq. (46). In practice, the number of points within a support is usually much larger than K. When the matrix A is found singular, the support should be enlarged to contain more points. Hence, a efficient search algorithm requiring minimum computation efforts is important. The algorithm reported by Swegle et al. [31] has been employed in the present method. This algorithm consists of three steps: sort, search and compare. The execution time of the algorithm is of O(N log2 N) for N points and N search regions.

3.3 Point-wise Solver for Constitutive Equations

The stress tensor for the Oldroyd-B fluid has been expressed by Eqs. (37) and (38) in a previous section. The velocity field is known after the solution of boundary integral equations in every time step. If Eq. (38) can be solved based the known kinematics, the stress tensor would be obtained from the configuration tensor C. The key problem in solving this equation is how to deal with the time derivative of C.

If the time derivative of C is treated in the Eulerian sense, Eq. (38) is not always solvable. For example, if a particle point x; is inside a particle at time t(n+1) but is in the fluid at time t(n) due to the particle motion, to solve for C(x,t(n)) we need to know C(x,t(n+1)) and its gradient but they cannot be determined since x was disabled at time t(n+1). Alternatively, if the time derivative of C in Eq. (38) is treated in the Lagrangian sense, an implicit pointwise difference form of this equation can be written as

$$\mu \frac{1}{1 + 4 t} \left[ C(X; t^{(n)}) \frac{h}{4} \phi C(X; t^{(n)}) + C(X; t^{(n)}) \phi \right] - \frac{1}{4 t} \phi X; t^{(n)} = C(X; t^{(n+1)}) + 4 t I; \quad (53)$$

where X = X(x; t^{(n)}) denotes a fluid particle which occupies the fluid point x at time t^{(n)}, and C(X; t^{(n+1)}) is the value of the configuration tensor of this fluid particle at time t^{(n+1)}. To determine C(X; t^{(n+1)}), one usually has to trace the fluid particle backward to find its position at time t^{(n+1)}, x = x(x; t^{(n+1)}); and get the value of the configuration tensor at this position. Here we suppose that the configuration tensor at all fluid points has been solved at time t^{(n)} and the fluid particles occupied these fluid points will move to new positions at time t^{(n+1)} carrying the values of the configuration tensor determined at time t^{(n+1)}. Though these new positions do not coincide with the fluid points at time t^{(n)}, the convection on the configuration tensor of t^{(n+1)} has been known and the values at the fluid points, C(X; t^{(n+1)}), can be interpolated from those at the new positions.

As mentioned in section 3.1, the coordinate system moves with the mass centre of the particle system. Hence, the translation of the coordinate system has to be taken into account. Assume that the position of a fluid point i in the coordinate system at time t^{(n+1)} is y^{(i)} and the velocity of the mass centre of the particle system at t^{(n+1)} is u^{0}. At time t^{(n)}, the displacement of the coordinate system is u^{0}t \text{ where } \dot{\xi} t = t^{(n)} \text{ and t^{(n)}}. The position of this fluid point relative to the coordinate system at t^{(n)} should be y^{0} + u^{0} t. If the fluid particle that occupied position y^{(i)}at t^{(n+1)} has velocity u^{0}, it would move to y = y^{0} + u^{0} t \text{ at time t^{(n+1)}} and carry the value of the configuration tensor at t^{(n+1)} to y. Hence we know the convection on the configuration tensor of time t^{(n+1)}, i.e., C^{(i)} y^{(i)}; and from which we can obtain C^{(i)} X; t^{(n+1)} \text{ in terms of the interpolation of the fixed least square method, where x is the position of a fluid point at which the fluid particle X occupies at time t^{(n+1)}}.

When C(X; t^{(n+1)}) is obtained, C(X; t^{(n+1)}) can be determined by solving Eq. (53). Since C is a symmetrical tensor of the second order, it has six independent components. We only need to solve a linear equation system with 6 unknowns point by point based on the kinematics obtained at previous time step. The initial condition for the configuration tensor can be set optionally, for example C(X; 0) = I.

3.4 Numerical procedures

Initially, the configuration tensor is set to be the unit tensor, i.e., zero viscoelastic stress tensor. Hence, the velocity of particular solutions, u^{0}, and the force and torque on particles due to the particular solution
and viscoelastic stress, $F_{p_i}^0$, $F_{p_i}^n$, $T_{p_i}^0$, $T_{p_i}^n$, in Eq. (18) are all equal to zero. The ambient fluid, external forces and torques acting on particles are given. The initial positions of particles and moving points are known. The origin of coordinates are located at the mass centre of particles and the positions of the xed points are xed with the coordinates. All xed points which are covered by any particle and too close to any moving points or element nodes are disabled. At each time step, the following procedures are conducted:

1. Solve the boundary integral equation, Eq. (18) for the double layer densities using the boundary element method;
2. Extract the rigid body motion of particles from the double layer densities using Eqs. (27);
3. Move the particles and moving points to the new positions according to their velocities determined in step 2.
4. Calculate uid velocities at each active xed points and moving points using Eq. (28);
5. Calculate the convection uid of $C(X; t^{(n)})$ based on the configuration tensor uid at the previous time step, using interpolation of the xed least square method;
6. Check all xed points to xed points which are covered by particles or too close to the element nodes or moving points and disable them;
7. Create a support for each active xed points, moving points and element nodes (for simplicity, these points are called active points thereafter) and a list of points and nodes in each support;
8. Calculate the velocity gradient at each active points using the xed least square method;
9. Solve the evolution equation for the configuration tensor, $C(X; t^{(n)})$, Eq. (53) and obtain the viscoelastic stress tensor, Eq. (37), and its divergence using the xed least square method at each active points;
10. Calculate the pseudo-body force at each active points, i.e., $f_i$ in the left hand side of Eq. (31) and solve this equation for $\dot{\omega}_i$, using GMRES algorithm, and calculate the velocity and stress due to the particular solution, $u^p$ and $\sigma^p$ according to Eqs. (??) and (??);
11. Calculate the force and torque due to the particular solution and viscoelastic stress acting on the particles and go to step 1. until the nal time step is reached.

All simulations were carried out in the cluster of Compaq Alpha workstations using Parallel Virtual Machine (PVM) library software.

4 Numerical Examples

4.1 Sphere Falling in an Oldroyd-B Fluid

The indirect CDL-BEM formulation is then applied to the simulation of a sphere sedimenting under gravity in the Oldroyd-B uid for veriication. The radius of sphere is normalized to unity, and 294,384 and 486 surface elements are used. The uid is quiescent and ils an inite space. To minimize computation cost we use as small number of uid points as possible and nally 9052 and 25785 uid points are used. The uid points are not uniformly generated. To generate the uid conuration, we use as small number of uid points as possible and nally the uid is generated in such a way that

- 4 layers of moving points are distributed in a thin layer on the sphere surface.
- 2 layers of moving points are distributed in a thin layer on the sphere surface.
- 1 layer of moving points are distributed in a thin layer on the sphere surface.

They are located on the line through the centre of the sphere and the node of a surface element with the distance of 1:015; 1:05; 1:10 and 1:20 of sphere’s radius from the centre. Hence the total number of moving points is equal to 4E6 number of surface elements.

The present method assumes negligible particle inertia. The initial condition for the conuration tensor was $C(X; 0) = I$, i.e., the initial viscoelastic stress was zero. Hence the sphere was settling as it
Figure 2: Dimensionless settling velocity of a sphere for \( \Omega = 1.94932 \), \( \chi_r = 1.3 \); \( \Delta t = 0.01 \) with various surface meshes and 9052 .eld points.

were in Newtonian .uid initially, and its sedimentation velocity would slow down gradually as viscoelastic stresses were building up and .nally, its settling velocity would reach a steady value.

Tiefenbruck and Leal [20] reported an axisymmetric numerical method for streaming .ow past a rigid sphere and a spherical bubble in an Oldroyd-type .uid. Their results for the Oldroyd-B .uid can be directly compared with those of the present method. They reported that the dimensionless drag force on the sphere were \( 2.999; 2.997; 2.985 \) and \( 2.98 \) at Weissenberg number of \( 0.1; 1=3; 2=8 \) and \( 1.0 \), respectively.

The dimensionless drag force is defined as

\[
\frac{f_{\text{drag}}}{24 \pi r U a} = \frac{\text{drag}}{24 \pi r U a}; \quad (54)
\]

where \( U \) corresponds to the steady settling velocity and \( a \) is the radius of the sphere. The Weissenberg number is defined as \( U/a \). In the present simulation, we set \( \chi_r = 1.3; \chi = 1.0 \) and gravity force on the sphere was \( 2.97 \) and balanced with the drag force. The steady settling velocity should be

\[
U = \frac{2}{3.97} \pi; \quad (55)
\]

The dimensionless settling velocities are plotted in Figs. 2, 3 and 4 for \( \chi_r = 1.3 \) and \( \chi = 1.94932; 3.89064 \) and \( 5.84795 \), where the dimensionless sedimentation velocity is defined as the ratio of the sedimentation velocity of the sphere in the Oldroyd-B .uid to that in Newtonian .uid, which is known from the solution of a Stokes .ow past a sphere and is equal to 2=9 under the condition mentioned above. In these simulations, 9052 .eld points and 294, 384 and 486 surface elements were used. Tiefenbruck and Leal's results are plotted in the .gures as well, for comparison. From these .gures, we can see that the settling velocity calculated by the present method approaches to steady-state solutions of Tiefenbruck and Leal as the viscoelastic force develops and that the di'erence between two set of results decreases as the number of surface elements increases for \( \chi = 1.94932 \) and 3.89064; i.e., the Weissenburg number to be about 1=3 and 2=8. However, Fig. 4 shows that for \( \chi = 5.84795 \), i.e., \( Wi = 1.0 \), the solutions of the present method are divergent due to the accumulation of numerical errors. The numerical error is mainly due to insufficient number of .eld points. In the above mentioned simulations, the .eld points were only distributed within a domain of about 6 times the size of the sphere. The minimum separation between points is about 0:164 radius of the sphere. This domain is too small to cover the disturbed .ow region by the sphere, especially

11
Figure 3: Dimensionless settling velocity of a sphere for $\phi = 3.89864$, $\gamma_r = 1.3$, $\phi t = 0.01$ with various surface meshes and 9052 field points.

Figure 4: Dimensionless settling velocity of a sphere for $\phi = 5.84795$, $\gamma_r = 1.3$, $\phi t = 0.01$ with various surface meshes and 9052 field points.
as large Weissenberg number. We next distributed the .el points into a larger domain of which the size was about 10 times of the radius and generated the second con.guration with 25785 .el points and the minimum separation between points to be 0.158 radius of the sphere. The dimensionless settling velocities of the sphere for \( Wi = 1.0 \) using 25785 .el points are shown in Fig. 5. Comparing with Fig. 4, we can see the results are improving significantly: the solution is convergent and the error relative to Tiefenbruck and Leal’s is small, 0.35% for 486 surface elements, 1.16% for 384 surface elements and 0.63% for 296 surface elements. However, the computation time increases drastically with the number of .el points. This places a constraint on the practical number of .el points.

In all above simulations, the time step is chosen to be 0.01. The explicit time dependence is contained in Eq. (53), from which we can see that the solution depends only on the parameter \( 4t = \epsilon \), i.e. \( 4t \) can be increasing with \( \epsilon \). However, the error in determining particle’s con.guration would increase with \( 4t \) and the accuracy of the solution would be doubtful if \( 4t \) is too large. Hence, \( 4t \) is still required to be small enough even for large \( \epsilon \). We found that \( 4t = 0.01 \) is suitable for \( \epsilon = O(1) \): For smaller \( 4t \), such as 0.005; or larger one, such as 0.02; the nal settling velocity changes by about 1–2%. This is due to the accumulation of numerical errors using 9052 .el points. This percentage “error” increases with \( \epsilon \), but decreases with the number of .el points, at a given \( 4t \).

4.2 Prolate spheroid in shear flow

In a Newtonian uid, a force- and torque-free prolate in shear ow rotates along a J e˘cry’s orbit, which is an analytical Stokes solution. However, if the uid is non-Newtonian, the orbit would deviate from J e˘cry’s due to the viscoelasticity effect. Though there are no complete analytical solution available for a prolate spheroid in a viscoelastic shear ow so far, some of analyses on bra’s motion in shear ow have been reported in the literature. Leal [21] obtained a asymptotic solution of a rod-like particle moving in shear ow of a second-order uid at low Weissenberg number. He showed that the second-normal-stress difference of the uid causes a drift across J e˘cry orbits towards the vorticity axis. Harlen and Koch [23] analyzed bra’s in shear ow of dilute Hookean dumbbell solutions at high Weissenburg number and found the similar spiral motion of bra but independent of the second-normal-stress difference. Both analyses were based on the assumption that the elastic stress is much smaller than the viscous stress and suggested that at low Weissenberg number the motion of bra spiral towards vorticity axis depends on a small parameter, which is proportional to the second-normal stress difference. At high Weissenberg number it
can be characterized by one parameter, \( \tilde{\omega} \), which is independent of the second-normal-stress difference [24]:

\[
\tilde{\omega} = \frac{\hat{\tau} \cdot \mathbf{i}}{W_i};
\]

(56)

For suit client small \( \tilde{\omega} \); the .bre follows approximately Je\( \text{\'}{\ \text{e}}\)ry's orbits but slowly crosses to orbits with progressively lower Je\( \text{\'}{\ \text{e}}\)ry orbital constants and thereby spiral toward the vorticity axis. The period of the motion, \( T \), increases with \( \tilde{\omega} \) in such a way that

\[
T = \frac{2\sqrt{4a_0 + a_1^3}}{\bar{c}(1 - \bar{c})a_2} \frac{1}{h}.
\]

(57)

At the critical value, \( \bar{\omega}_c = 2a_0 \), the .bre remains in the \( \langle \rangle \)ow-vorticity plane and no longer rotates in Je\( \text{\'}{\ \text{e}}\)ry orbits. Then the .bre will slowly rotate within the \( \langle \rangle \)ow-vorticity plane until it aligns with the vorticity axis. When \( \tilde{\omega} = 0 \), Eq. (57) predicts the period of the Je\( \text{\'}{\ \text{e}}\)ry orbits, \( T_0 \), for Newtonian \( \langle \rangle \)ow. We can further express the ratio of the non-Newtonian period to the Newtonian one as a function of \( \tilde{\omega} = \bar{\omega}_c \):

\[
\frac{T}{T_0} = \frac{h}{1} (\tilde{\omega} = \bar{\omega}_c)^{2}.
\]

(58)

The increase of the period was cons..rmed by observations of Bartram et al [22], who found that the period of rotation of a particle in non-Newtonian \( \langle \rangle \)uid is considerable longer than that in Newtonian \( \langle \rangle \)uid. Iso et al [24] reported detailed observations on the motion of .bres in shear \( \langle \rangle \)ow of polyacrylamide in corn syrup-water (PAAm) and polyisobutylene in polybutene (PIB-PB) solutions. Their .ndings were in qualitative agreement with the high Deborah number Oldroyd-B theory of Harlen and Koch, that is, the deviation from Je\( \text{\'}{\ \text{e}}\)ry's orbits can be found in the Boger \( \langle \rangle \)uid, i.e., independent of the second normal stress difference. They also found that the period of .bre in viscoelastic \( \langle \rangle \)uid is longer than that in Newtonian \( \langle \rangle \)uid. However, their observed periods were larger than those predicted from Eq. (57): for PAAm solutions the predicted periods were only \( 1-3% \) longer than those for Newtonian \( \langle \rangle \)uid but observed to be up to \( 10% \) longer; and for PIB-PB solutions, the predicted ones were only \( 6-25% \) longer but observed to be \( 55-84% \) longer, also see Fig. 6 below. Another observed deviation from the theory is the .nal orientation of the .bre, which is about \( 15^\circ \) away from instead of aligning with the vorticity axis in PAAm solutions, and about \( 10^\circ-50^\circ \) in PIB-PB solutions. As shear rate or aspect ratio increases, the long-time behaviour of .bres was signi..cantly deviated from the prediction, even no initial spiralling motion of .bres was observed.

A suitable numerical simulation is an ideal way to provide detailed information on .bres motion in viscoelastic \( \langle \rangle \)ow. We use the present numerical method to simulate a prolate spheroid rotating in shear \( \langle \rangle \)ow of the Oldroyd-B \( \langle \rangle \)uid. The prolate spheroid is neutrally buoyant in the shear \( \langle \rangle \)ow with unity shear rate. The aspect ratio of the prolate is 2 with the length of its major axis set at 1; and 384 surface elements with 1536 moving points are used. 25785 .xed.eled points of are generated in same way to the second .eled point con..guration mentioned previously. This .eled point con..guration was used in simulating the sphere sedimentation and resulted in good results. Though this cannot guarantee its performance in simulating a prolate in shear \( \langle \rangle \)ow, it is not practical to use more .eled points with computing facilities currently available to us. The relaxation time of the \( \langle \rangle \)uid is set to be \( 0.7 \) and the shear rate was \( 1 \); i.e., \( \text{Weissenberg number} = 0.7 \). The initial con..guration of the con..guration tensor was \( C(X;0) = \mathbf{I} \) and time step was \( 0.01 \).

The initial orientation vector of the prolate was \( (0.50;0.0;0.866) \); i.e., the major axis of the prolate is in the shear and vortex plane \( (xz) \) and is inclined at an angle of \( 60^\circ \) to the shear direction \( (x) \). The relative viscosity is chosen to be from \( 1.001 \) to \( 1.30 \), i.e., \( \tilde{\omega} = \bar{\omega} \) from \( 1=700 \) to \( 3=7 \). The simulations were conducted in a cluster of Compaq Alpha workstation and 4 machines were used for each run. The typical CPU time to simulate one time step is about 33 minutes. The Newtonian period, \( T_0 \), is \( 5\frac{4}{7} \) when \( a_0 = 2 \) and \( \omega = 1.0 \). It requires 1571 time steps, i.e., about 864 CPU hours to simulate one period of the prolate's motion when the time increment is \( 0.01 \): A saving on the computation time was done by simulating only \( 1=4 \) of the period for most of jobs to determine \( T = T_0 \). The results would contain some errors due to not fully-established viscoelastic stress. However, the errors may be not too serious for a qualitative comparison with the theory. We checked one typical case of \( \hat{\tau} = 1.3 \) and \( \tilde{\omega} = 3=7 \) and found that it took 480 time steps to simulate the \( 1=4 \) of the period and 1933 time steps to simulate whole period, i.e., \( T = 19.33 \). If we calculate the period based on the time spent in simulating the \( 1=4 \) period, a .igure of \( T = 19.52 \) is obtained, with an error of less than \( 1% \). The simulated relative delay of particle's rotating due to viscoelastic stress is shown in Fig. 6, with the theory of Harlen and Koch [23] (solid line) and the experimental observation of Iso et al [24] (solid circles and triangles). From the .igure, we can see that our numerical results qualitatively agree with the experimental observations and theoretical
predictions. The viscoelastic stress slowed down the rotation of particles in shear flow. The deviation of the numerical results from the theoretical predictions may be attributed to the basic assumption in the theory, i.e., the Deborah number is much larger than \((\ln(a))^1\) and \((\ln(a))^2 \ll 1:0\) [23]. In the simulation, Deborah number is less than \((\ln(a))^1\) and \((\ln(a))^2 > 1:0\): In addition, the theory and experiments deal with fibre suspensions, not a single fibre in isolation. The numerical accumulated errors in a multi-step calculation may be a factor in the accuracy of the numerical solution. As demonstrated in the previous section, insufficient number of field points is a key factor contributing to the numerical errors. However, a larger number of field points requires considerable computation efforts, and is not attempted with current computation resources available to us. More accurate comparison would be a subject for further research.

However the present simulation do provide some information on the particles motion in viscoelastic shear flow. The next few figures, we show how the particle rotates and leaves J e\'ery’s orbit due to viscoelastic stress. The orbits for \(\gamma_r = 1:001\) and \(1:3\) (solid lines) are shown in Fig. 7 with J e\'ery’s orbit (dotted line). When \(\gamma_r = 1:001\), the viscoelasticity effect is very weak and the fluid is nearly Newtonian. We can see its orbit almost coincides with J e\'ery’s. When \(\gamma_r\) increases, the orbit deviates from J e\'ery’s under the driving action of viscoelastic torque. As the viscoelastic torque develops, the particle is driven from J e\’ery’s to another unclosed orbit. Fig. 8 shows the simulated orbit for Wi = 0:7 and \(\gamma_r = 1:3\) with the J e\’ery’s (dotted line). Due to the limitation in computational resources, we only simulate a total time of 30:25, i.e., 3025 time steps. The prolate spheroid is clearly moving along an unclosed orbit. In Fig. 9 the view along the vorticity axis is displayed. The above two figures show that the prolate spheroid is gradually deviating from J e\’ery’s orbit while the viscoelastic stress is build up and the viscoelastic torque drives the spheroid rotating toward the vortex axis, the z-axis in Figs. 8 and 9.

5 Final Remarks

In this paper, we report the formulation and the implementation of an indirect boundary integral equation method, suitable for solving the mobility problem of a particle system in a viscoelastic fluid. We also present some results of a sphere settling in the Oldroyd-B fluid as a test case and a prolate spheroid in shear flow of an Oldroyd-B fluid. The simulated results are compared with numerical results of Tiefenbruck and Leal [20], the theory of Harlen and K och [23], and experimental observations of Iso et al [24]. It is demonstrated that the present method is suitable at least for simulating the motion of one particle viscoelastic flow, with currently available computing resources. It is straight forward to use this method to multi-particle systems.
Figure 7: Jeffrey’s orbit (dotted line) compared to the numerical orbits.

Figure 8: Jeffrey’s orbit (dotted line) compared to the numerical orbits.
Figure 9: Jeffery's orbit (dotted line) compared to the numerical orbits, as viewed along the vorticity axis.

if more powerful computing resources becoming available.

The features of the present method can be outlined as follows:

1. The boundary elements are used to represent the surfaces of the particles, and the boundary conditions on the surfaces can be satisfied more accurately. The boundary element mesh can be easily updated with the particles motion. Computation efficiency is gained from the reduction in dimensionality.

2. The radial basis function and the particular solution method are used to avoid volume integration in the boundary integral equation formulation. Hence the volume mesh is not needed in the numerical discretization of the boundary integral equations.

3. The complete double layer density formulation of the boundary integral equation with completing and deactivation schemes is well-posed for the mobility problems of multi-particle systems. This guarantees the stability of the solution procedure.

4. The fixed least square methods are employed in numerical fitting and differentiation without the need of volume meshing.

5. A point-wise solver is further developed to solve the constitutive equation for viscoelastic fluid at discrete points in the flow field. All field points move with the mass centre of the particle. Hence, the number of field points can be reduced but the accuracy is maintained. This method requires much less computation and memory than other solvers with mesh or meshless.

6. A master/slave programming paradigm using Parallel Virtual Machines (PVM) library software is employed to raise the computation efficiency further.

The present method can be ideally used to flow problems in which inertial and viscoelastic forces do not dominate the flow, i.e., the flow has low Reynolds number and weak elasticity. This is inherent in the iteration process employed in the boundary element formulation.

The boundary integral equation is formulated for the infinite domain but we can only distribute field points within a finite domain in the simulation. The problem is how large the finite domain ought to...
be and how many field points should be distributed in order to obtain accurate numerical solutions for
the infinite domain problem. Tiefenbruck and Leal (1982) used a domain of 10 sphere radii in their axisymetric steady simulation of a viscoelastic flow pass a sphere. We are dealing with a 3D time-dependent
viscoelastic flow problem with moving boundaries. In addition, the particle has to be tracked for a long
time to obtain its trajectory. But we only distribute field points within a cubic box of 10 times the particle
size. The replacement of infinite domain by a finite domain results in numerical errors, especially at large
Weissenberg number. However, the simulated results agree qualitatively with some theoretical, numerical
and experimental results. Hence, this method would find more complicated applications as computer power
increases.

Though the present method may be more efficient than some other numerical methods, the computation
requirement is still too much for typical present-day computing resources. How to reduce the computation
requirements is the key problem of this method. Roughly speaking, the period is increase linearly with
the aspect ratio but time increment should decrease with the aspect ratio. The number of field points and
boundary elements should increase with the aspect ratio. Even for an aspect ratio of 2, the simulation
requires 1600 to 2200 time steps with the time increment of 0.01 to cover one orbital period. A typical
CPU time per time step is about 33 minutes for each slave and 19 minutes for the master to simulate a
prolate rotating in shear flow with 384 surface element and 25785 field points when 4 machines are used
in the Compaq-workstation cluster. The simulations were also conducted in a 3 CPU batch queue of
Compaq GS320. The total CPU time per time step is about 68:4 minutes. Hence we could not simulate
more complicated problems. But it may be possible to speed up the calculation further. We found that
the most CPU time was spent in solving Eq.(31) for the radial basis function using GMRES method. If
a suitable preconditioner and more efficient parallel iterative solver can be used, the CPU time can be
further reduced.

Acknowledgement: The authors appreciate Dr. R. Zheng, Moldflow Corp, for his helpful advices on the
particular solution method.

References


