Preconditioned iterative solution of the corrected mobility problem

A. A. Mammoli

Department of Mechanical Engineering, University of New Mexico, USA

Abstract

Boundary integral equation methods are, on the surface, ideally suited for the description of the motion of rigid particles in flowing concentrated suspensions. However, several difficulties exist, the principal one being the treatment of the flow in regions of fluid bounded by surfaces in close proximity to one another. In such regions, the flow is not represented accurately, resulting in poor relative velocity and macroscopic rheological behavior predictions.

Recently, analytic lubrication solutions describing the forces arising from the relative motion between two particles have been used to correct the boundary integral equation, with very promising results. The main difficulty found in that approach lies in the evaluation of the lubrication forces. These are a linear function of the velocity, and as a consequence they can be incorporated into the linear system that results from the discretization of the boundary integral equation.

The modification to the linear system disrupts the diagonal dominance inherent in second-kind integral formulations, negating the gain in efficiency that results from the utilization of iterative solvers. The application of suitable preconditioners can restore the diagonally dominant nature of the systems. The relative merits of two different preconditioners are discussed, and examples showing their effectiveness are shown. In both cases, the number of iterations required to obtain a solution to within a given tolerance is reduced drastically. The more effective preconditioner is more expensive to compute than the other, but both are very inexpensive compared to the overall solution time.
1 Introduction

The mobility problem, in which the velocity of rigid particles suspended in a fluid is unknown, while the applied forces are known, is the object of intense study in the field of suspension rheology. For Stokes flows, the completed double layer boundary integral equation (CDL-BIE) appears to have many qualities which make it ideally suitable for the treatment of this problem [3, 7]. As with all boundary element methods, particles and containers of any shape can be discretized readily. In the mobility problem, the CDL-BIE is a Fredholm equation of the second kind, so that the linear systems that result from its discretization are characterized by low condition numbers and can be solved using Krylov space iterative methods. Finally, the operation count for the solution of the linear system can be reduced from $O(N^2)$ to $O(N \log N)$ with the application of multipole acceleration techniques [5, 6].

Sadly, large scale simulations performed to predict the rheology of dense suspensions resulted in severe underestimation of the apparent viscosity of a suspension of identical spheres at solids volume fractions ($\phi$) above 0.3 when compared to widely used empirical correlations [4]. One of the sources of error is likely to be the inability of the method to represent the lubrication flow between particles in near-contact. In a recent study performed using a large-scale Stokesian Dynamics simulation [9], it was shown that the peak in the pair distribution function is located between 2.00125$a$ at $\phi = 0.1$ and 2.00011$a$ at $\phi = 0.5$, where $a$ is the particle radius. In other words, the average separation between particles is very small, and lubrication flows must exist at moderate and dense concentrations. The hypothesis that the incorrect treatment of lubrication forces is responsible for the underprediction of the apparent viscosity is supported by recent work by Qi et al. [8]. In this, the authors substitute the boundary integral on the surface of neighboring particles with the lubrication forces exerted by such particles on the particle where collocation is performed. The viscosity around the particle is assumed to be that of an equivalent homogenized fluid representing the suspension, as predicted by an empirical model. The resulting apparent viscosity is in agreement with empirical correlations.

While the approach suggested by Qi et al. appears to provide satisfactory results, it suffers from two drawbacks. First, a constitutive model for the viscosity of a suspension is required. Second, some of the information in the intermediate field, which may be critical in determining particle trajectories, is removed when the neighboring particles are substituted with equivalent forces. A similar method, which makes use of lubrication solutions, but does not require the use of a homogenized equivalent fluid and does not remove the intermediate field information, was recently proposed [4]. The methodology is an adaptation of the approach used by Zinchenko [10] for the analysis of heat flow in loaded granular media. The method was used successfully to simulate the relative motion of two particles in a sheared suspension, and to produce rheological data in the form of a self-diffusion coefficient [1]. In that implementation, the particle velocities were first calculated without lubrication forces. These velocities were subsequently used to obtain the
lubrication forces, which were then used to obtain a further velocity estimate, and so on until convergence. Further work showed this approach to be unsuited to the case of many particles.

In the following sections, the incorporation of the lubrication force evaluation into the CDL-BIE coefficient matrix is described. The effect of this operation on the condition number of the matrix, and hence on the overall operation count to obtain a solution, is discussed. The application of preconditioners based on Schur complement blocks is illustrated, in the context of parallel algorithms. Finally, the effect of preconditioning is illustrated through various examples.

2 The corrected boundary integral equation

In the CDL-BIE, the velocity \( \mathbf{u} \) at a point \( \mathbf{x} \in \Gamma \), where \( \Gamma = \Gamma_1 \cup \Gamma_2 \cup \ldots \Gamma_n \) is the union of all individual particle surfaces, can be written as:

\[
\mathbf{u}(\mathbf{x}) = \mathbf{u}_\infty(\mathbf{x}) + \frac{1}{2} \mathbf{\phi}(\mathbf{x}) + \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \cdot \mathbf{\phi}(\mathbf{y}) d\Gamma_y \\
+ \sum_{p=1}^{n} \left[ \mathbf{F}^p - \frac{1}{2} (\mathbf{T}^p \times \nabla) \right] \cdot \frac{\mathbf{G}(\mathbf{x} - \mathbf{y}^p)}{8\pi},
\]

(1)

where \( \mathbf{F}^p \) and \( \mathbf{T}^p \) are the known force and torque applied to particle \( p \); \( \mathbf{u}_\infty(\mathbf{x}) \) is the velocity that would exist at location \( \mathbf{x} \) in the absence of particles due to an underlying flow field; the kernel function \( K \) and the Oseen tensor \( \mathbf{G} \) are well-known and can be found in the literature \([3, 7]\). On the surface of particle \( p \), the velocity is given by:

\[
\mathbf{u}(\mathbf{x}) = \mathbf{u}^p + \mathbf{\omega}^p \times \mathbf{r}^p,
\]

(2)

where \( \mathbf{u}^p \) and \( \mathbf{\omega}^p \) represent the translation and rotation of particle \( p \). The rank deficiency of the system is removed by setting:

\[
\mathbf{F}^p = \int_{\Gamma_p} \mathbf{\phi}(\mathbf{x}) d\Gamma, \quad \mathbf{T}^p = \int_{\Gamma_p} \mathbf{\phi}(\mathbf{x}) \times \mathbf{r}^p(\mathbf{x}) d\Gamma,
\]

(3)

which also has the effect of constraining out the eigensolutions of the system. The resulting linear system has the form:

\[
\begin{bmatrix}
\frac{1}{2} \mathbf{I} + \mathbf{K} & \mathbf{U} \\
\mathbf{F} & 0
\end{bmatrix}
\begin{Bmatrix}
\mathbf{\phi} \\
\mathbf{v}
\end{Bmatrix}
= 
\begin{Bmatrix}
\mathbf{s} \\
\mathbf{f}
\end{Bmatrix}
\]

(4)

For small interparticle gaps, on the order of \( 5 \times 10^{-2} \alpha \), the flow between the surfaces is not described adequately by the integral equation. This is because the discretization of the neighboring surfaces cannot be made fine enough, in practice, to capture the variation in the hydrodynamic potential. It is possible to utilize analytical solutions between two isolated particles to correct the lack of resolution in the lubrication region.
The incorporation of the lubrication solution is difficult because it is already captured in part by the solution of the BIE. Ideally, the contribution to the lubrication flow provided by the BIE solution would be substituted with the analytical solution. This is accomplished by adapting a technique suggested by Zinchenko [10] for the simulation of heat transfer in a packed bed. With reference to Fig. 1, given a collocation point on particle A, the surface of particle B is modified so that a minimum critical gap between the two exists. The critical gap is set so that an acceptably accurate solution is achieved with a specified mesh refinement level. The lubrication force for the modified geometry, which is now correctly predicted by the BIE solution, is removed using the analytical lubrication force for the modified geometry and replaced with the analytical lubrication force for the original geometry. The discretized version of the 'corrected' BIE for an unbounded flow and a collocation node on particle k then becomes:

\[
\frac{1}{2} \phi(x) + \int_{\Gamma_k} K(x,y) \phi(y) \, d\Gamma_y - u^k - \omega^k \times r^k = -u_{\infty}(x) - \sum_{p=1}^{n} \left\{ \left[ F^p - \frac{1}{2}(T^p \times \nabla) \right] \right. \\
+ \left. \sum_{q=N_p(1)}^{N_p(N(p))} \left[ (F_{pq}^{orig} - F_{pq}^{mod}) - \frac{1}{2} (T_{pq}^{orig} - T_{pq}^{mod}) \times \nabla \right] \right\} \cdot \frac{G(x - y^k)}{8\pi},
\]

where \( \Gamma_k \) is a modified boundary, similar to the original boundary except that the neighbors of particle k, which owns the collocation node, are shrunk; \( N(p) \) is the number of neighboring particles to \( p \), and \( N_p(i) \) is the \( i \)th neighbor of particle \( p \). The force \( F_{pq}^{orig} \) and \( F_{pq}^{mod} \) represent the analytical lubrication forces for the original and modified geometry applied to particle \( p \) by its neighbor \( q \), and similarly for the torques.

Analytical solutions for the interactions between two particles have been obtained for the set of relative particle motions shown in Fig. 2 [1]. Any relative
motion between two particles can be decomposed into a linear combination of these basic motions.

![Image of four basic modes of motion.](image)

Figure 2: Four basic modes of motion. From left to right: mode I – rotation about an axis normal to the line of centers; mode II – translation along an axis normal to the line of centers; mode III – rotation about the line of centers; mode IV – translation along the line of centers.

In the calculation of the lubrication forces, it is convenient to operate in a local coordinate system defined by the relative position of the particles, as shown in Fig. 3. In this coordinate system, it is simple to show that the lubrication force and torque to which particle \( p \) is subjected due to the presence of neighboring particle \( q \) can be written in terms of a linear mapping of the respective velocities:

\[
\begin{align*}
\mathbf{F}_{\text{orig}}^{pq} - \mathbf{F}_{\text{mod}}^{pq} &= \mathbf{F}_{v}^{pq,p} \mathbf{v}_p + \mathbf{F}_{v}^{pq,q} \mathbf{v}_q + \mathbf{F}_{\omega}^{pq,p} \mathbf{\omega}_p + \mathbf{F}_{\omega}^{pq,q} \mathbf{\omega}_q \quad (6) \\
\mathbf{T}_{\text{orig}}^{pq} - \mathbf{T}_{\text{mod}}^{pq} &= \mathbf{T}_{v}^{pq,p} \mathbf{v}_p + \mathbf{T}_{v}^{pq,q} \mathbf{v}_q + \mathbf{T}_{\omega}^{pq,p} \mathbf{\omega}_p + \mathbf{T}_{\omega}^{pq,q} \mathbf{\omega}_q, \quad (7)
\end{align*}
\]
where the coefficient matrices $F$ and $T$ can be deduced easily from the work of Hampton [1]. Now, the local rate of translation and rotation can be expressed in terms of its global value by the coordinate rotation $R_{pq}$ for the particle pair $pq$:

$$
\bar{v} = R_{pq}v
$$

$$
\bar{\omega} = R_{pq}\omega.
$$

(8) (9)

Forces in the global coordinate system can be obtained by using the inverse rotation, so that finally one has:

$$
F_{pq}^{\text{orig}} - F_{pq}^{\text{mod}} = F_{pq}^{v,P}v_p + F_{pq}^{v,q}v_q + F_{pq}^{\omega,P}\omega_p + F_{pq}^{\omega,q}\omega_q
$$

$$
T_{pq}^{\text{orig}} - T_{pq}^{\text{mod}} = T_{pq}^{v,P}v_p + T_{pq}^{v,q}v_q + T_{pq}^{\omega,P}\omega_p + T_{pq}^{\omega,q}\omega_q,
$$

(10) (11)

where, for example, $F_{pq}^{v,p} = (R_{pq})^{-1}F_{v}^{p}R_{pq}$.

The attractiveness of this approach lies in the fact that the matrix structure for the solution of the mobility problem, Eq. 4, remains unaltered, at least on the large scale. However, upon closer scrutiny, it will be noted that the sub-matrix $U$ no longer consists of a diagonal array of $6 \times 6$ matrices, but is in fact fully populated. This change in sub-structure has implications in the formulation of a preconditioner.

3 Preconditioning

As noted previously, the principal reason for using CDL--BIE is the low condition number of the linear system that results from its discretization, which makes the system suitable for the application of very efficient iterative solution techniques, such as the combination of multipole acceleration with a GMRES solver. The presence of the $U$ and $F$ matrices disturbs the diagonal dominance of the system, and the number of iterations required to obtain a solution may increase substantially. Previous work [5] has shown that a left preconditioner of the form:

$$
P = \begin{bmatrix}
\frac{1}{2} & U \\
F & 0
\end{bmatrix}^{-1}
$$

is a good approximation to the inverse of the mobility matrix and is very effective at restoring the diagonal dominance of the CDL--BIE. The application of the preconditioner on a vector is described by:

$$
\begin{bmatrix}
\frac{1}{2} & U \\
F & 0
\end{bmatrix}^{-1}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -U(FU)^{-1}[2Fx_1 - x_2] + 2x_1 \\ \frac{1}{2}(FU)^{-1}[2Fx_1 - x_2] \end{bmatrix}.
$$

(12) (13)

It is readily apparent that the formation and inversion of the matrix $FU$ is the most computationally expensive operation in the preconditioning process. As such, its parallel implementation is desirable. To understand the process, it is instructive to inspect the structure of the $F$ and $U$ matrices. The $F$ matrix results from
the discretization of Eqns. 3. In both the original and corrected CDL–BIE, it takes the form:

\[ F = \begin{bmatrix} F^1 & \cdots & F^n \end{bmatrix}, \]

where the submatrices \( F^i \) have dimension \( 6 \times M^i \), \( M^i \) being the number of degrees of freedom on particle \( i \). The structure of the matrix \( U \), on the other hand, is affected by the correction procedure. The form of the uncorrected and corrected matrices, \( U \) and \( U_C \) respectively, is:

\[ U = \begin{bmatrix} U^1 \\ U^2 \\ \vdots \\ U^n \end{bmatrix}, \quad U_C = \begin{bmatrix} U^{11} & U^{12} & \cdots & U^{1n} \\ U^{21} & U^{22} & \cdots & U^{2n} \\ \vdots & \vdots & \ddots & \vdots \\ U^{n1} & U^{n2} & \cdots & U^{nn} \end{bmatrix}, \]

where each submatrix \( U^i \) or \( U^{ij} \) has dimension \( M^i \times 6 \). The product \( FU \) is a diagonal array of \( n \) \( 6 \times 6 \) matrices. Its inverse is simply the array of inverses of the original matrices. It is ideally suited for parallel evaluation and inversion, with minimal operation count and communication overhead, because the preconditioner is calculated and applied locally.

The calculation becomes substantially more complex in the case of the preconditioner for the ‘corrected’ CDL–BIE. The matrix \( U_C \) is potentially fully populated because all particles (except for isolated ones) are subject to lubrication correction forces from their neighbors, which must be accounted for in the range completer, as can be inferred from Eq. 5. The product \( FU_C \) is also a fully populated matrix, of dimension \( 6n \times 6n \). The parallel evaluation of the matrix coefficients and the inversion of the matrix are now both more desirable and more difficult. The evaluation of the matrix coefficients is parallelized by splitting the number of particles as evenly as possible between processes. On each process, the submatrices \( F^i \) are evaluated for the particles allocated locally. Likewise, only the row blocks of the matrix \( U_C \) which correspond to the collocation nodes on the locally allocated particles are evaluated on each process. The row block of the product \( FU_C \) can be evaluated locally by multiplying the partial rows of the matrix \( F \) with the partial columns of \( U_C \).

The matrix \( FU_C \) has dimensions \( 6n \times 6n \), substantially smaller than the overall dimensions of the corrected CDL–BIE system. However, if the number of particles \( n \) is large, say on the order of 1000, the evaluation of the inverse can still be computationally expensive, and should be done in parallel by using a suitable direct solver. For small to moderate values of \( n \), say up to 100, it is simpler to gather the entire matrix to a single process, perform the inversion on this process, and scatter the results back to the individual processes. This latter option is used for the purposes of the work described here.
4 Results

The system of equations generated by the corrected CDL–BIE was solved using a public domain parallel iterative solver (PIM [2]). The iterative method chosen was a restarted GMRES, with a vector space of 30, a reasonable compromise between performance and memory usage. The only user defined operation required by PIM is the forward multiplication of the coefficient matrix with a vector supplied by the solver. The iteration is terminated when the ratio of the norms of the current residual and the initial residual is below a specified tolerance, in this case $10^{-8}$.

The results for the case of two and four closely interacting particles, arranged as shown in Fig. 4, are plotted in Fig. 5. For the case of two particles, the unpreconditioned iteration converges in 117 iterations, versus 50 for the iteration with the preconditioner for the conventional CDL–BIE and 28 for the iteration with the preconditioner for the corrected CDL–BIE. In the case of four particles, the number of iterations for the unpreconditioned system increases to 1136, approximately a tenfold increase. In the case of the preconditioned systems, convergence was reached at 89 and 52 iterations.

Figure 4: Test particle arrangements for the simulation of the motion of two and four closely interacting particles in a shear flow. Shearing is in the plane containing the particle centers. The minimum distance between particle surfaces is 0.001a.

Figure 5: Residual, normalized by its initial value, as a function of the number of iterations, for a system of two and four closely interacting particles, left and right respectively. Preconditioner 1 is built for the uncorrected CDL–BIE and preconditioner 2 for the corrected CDL–BIE.
The case for the application of a preconditioner seems evident. As the geometric complexity of the system increases, the diagonal dominance of the discretized CDL–BIE system is increasingly disrupted, severely affecting the iterative solution of the system. It must be noted that iterative solution is not only desirable, but unavoidable if multipole acceleration methods are used. This is because although the multiplication of a vector with a matrix can be calculated to a high degree of precision, individual matrix coefficients are not known explicitly, rendering the solution of the system by Gaussian elimination impossible.

5 Conclusions

For closely interacting particles suspended in a flow at low Reynolds numbers, it has been shown that conventional boundary integral calculations provide very poor estimations of the relative velocities between particles, and inaccurate prediction of the rheological behavior of dense suspensions. The formulation of a corrected CDL–BIE, which accounts for the lubrication flow in the regions of fluid bounded by nearly touching surfaces, has been shown. In this formulation, the lubrication forces are added to the range completer in the CDL–BIE by expressing them as a function of the particle velocities. The macroscopic structure of the linear system produced is unaltered, but the substructure of one of the matrices is substantially modified. Thus, the effectiveness of preconditioners previously used with the CDL–BIE is reduced. A new preconditioner is implemented which fully accounts for the altered submatrix structure. The coefficients for this preconditioner are evaluated locally, but the inversion of one matrix is performed in series on one processor using Gauss–Jordan elimination. For systems with moderate numbers of particles, this is acceptable since the matrix inversion represents a very small fraction of the solution time of the system. For systems with large numbers of particles, parallel inversion of the preconditioner matrix should be considered.

The preconditioner formulated for the corrected CDL–BIE provides a factor of 4 and a factor of 20 reduction in the number of iterations for systems of two and four particles respectively. For larger systems, the improvement is expected to grow accordingly. It can be stated with confidence that only through the use of the preconditioner does the solution of such problems become feasible.

Surprisingly, the preconditioner that was used in the regular CDL–BIE formulation also provides a substantial improvement in the efficiency of the iterative solution. Systems preconditioned with this preconditioner converge approximately two times slower than with the optimal preconditioner. When considering the added complexity involved in the calculation of the optimal preconditioner, and the added difficulty in its parallel implementation, the use of the conventional preconditioner is an option which should be considered. However, if maximum efficiency is imperative, the full parallel implementation of the optimal preconditioner is recommended.
Acknowledgement

This work was supported in part by U.S. Department of Energy grants DE-FG03-97ER14778 and DE-FG03-97ER25332. This financial support does not constitute an endorsement by the DOE of the views expressed in this paper. The support of the High Performance Computing Education and Resource Center is also gratefully acknowledged.

References


