SAND20XX-XXXXR

LDRD PROJECT NUMBER: 158805

LDRD PROJECT TITLE: Next Generation Suspension Dynamics Algorithms

PROJECT TEAM MEMBERS: P. R. Schunk (SNL), Prof. Jonathon Higdon and Steven Chen (University of Illinois Urbana Champagne)

ABSTRACT

This research project has the objective to extend the range of application, improve the efficiency and conduct simulations with the Fast Lubrication Dynamics (FLD) algorithm for concentrated particle suspensions in a Newtonian fluid solvent. The research involves a combination of mathematical development, new computational algorithms, and application to processing flows of relevance in materials processing. The mathematical developments clarify the underlying theory, facilitate verification against classic monographs in the field and provide the framework for a novel parallel implementation optimized for an OpenMP shared memory environment. The project considered application to consolidation flows of major interest in high throughput materials processing and identified hitherto unforeseen challenges in the use of FLD in these applications. Extensions to the algorithm have been developed to improve its accuracy in these applications.

INTRODUCTION

Fast Lubrication Dynamics (FLD)

1. Governing Equation

The movement of colloidal particles suspended in a Newtonian fluid is described by the Langevin equation

$$m\frac{\boldsymbol{dU}}{dt} = \boldsymbol{F}^H + \boldsymbol{F}^B + \boldsymbol{F}^P \tag{1.1}$$

where m is the particle mass, U is particle velocity, and F^H, F^B, F^P are the generalized force vectors due to hydrodynamic interactions, Brownian motions, and interparticle interactions respectively. Each of the velocity and force vectors comprises of a liner and a angular component resulting a length of $6N_p$, where N_p is the number particles in the system. The inertia of the particles in the above equation is usually negligible and is usually taken to be zero in our simulations and therefore we may simplify our governing equation to

$$\boldsymbol{F}^H + \boldsymbol{F}^B + \boldsymbol{F}^P = 0 \tag{1.2}$$

In the following sections we will describe in detail the calculations of each force,i.e.
F^H, F^B, F^P, and the method of solution in solving our governing equation.

2. Hydrodynamic Interactions

The hydrodynamic force F^H, torque T^H, and stresslet S^H exerted by the fluid on the colloidal particles can be described by a linear function expressed in the form of a resistance tensor R.

$$\begin{pmatrix} \mathbf{F}^H \\ \mathbf{T}^H \\ \mathbf{S}^H \end{pmatrix} = \mathbf{R} \begin{pmatrix} \boldsymbol{U}^\infty(\boldsymbol{x}) - \mathbf{u} \\ \boldsymbol{\Omega}^\infty - \boldsymbol{\omega} \\ \mathbf{E}^\infty \end{pmatrix} \tag{2.1}$$

where **u** and **ω** are the velocity/angular velocity of the particles respectively and $\mathbf{E}^\infty$ is the rate of strain. The resistance tensor is comprised of second rank tensors A, B, B˜ and C, third rank tensors $G, \tilde{G}, H, \tilde{H}$ and a fourth rank tensor M.

$$\mathbf{R} = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} & \tilde{\mathbf{G}} \\ \mathbf{B} & \mathbf{C} & \tilde{\mathbf{H}} \\ \mathbf{G} & \mathbf{H} & \mathbf{M} \end{pmatrix} \tag{2.2}$$

In the FLD method, the resistance tensor **R** is expressed as a sum of an isotropic resistance tensor and a pairwise lubrication resistance tensor as seen in equation

$$\mathbf{R}_{FLD} = \mathbf{R}_{ISO} + \mathbf{R}_{LUB} \tag{2.3}$$

The formulation of $\mathbf{R}_{ISO}$ and $\mathbf{R}_{LUB}$ as well as the sub-block resistance tensors will be described below.

2.1 Isotropic Resistance Tensor

The isotropic resistance tensor is described by

$$\mathbf{R}_{ISO} = \begin{pmatrix} R_{0_{FU}}\boldsymbol{I} & 0 & 0 \\ 0 & R_{0_{T\Omega}}\boldsymbol{I} & 0 \\ 0 & 0 & R_{0_{SE}}\boldsymbol{I} \end{pmatrix} \tag{2.4}$$

The terms R_{0FU}, $R_{0T\Omega}$, R_{0SE} used in FLD are chosen to match the short-time self-diffusivity D_s of a hard sphere calculated using Stokesian dynamics (SD). In our method, these terms depend solely on volume fraction and are given by the following equations

$$\frac{R_{0_{FU}}}{6\pi\mu a} = 1 + 2.725\phi - 6.583\phi^2 \tag{2.5}$$

$$\frac{R_{0_{T\Omega}}}{8\pi\mu a^3} = 1 + 0.749\phi - 2.469\phi^2 \tag{2.6}$$

$$\frac{R_{0_{SE}}}{\frac{20}{3}\pi\mu a^3} = 1 + 3.643\phi - 6.951\phi^2 \tag{2.7}$$

2.2 Lubrication Resistance Tensor

2.2.1 Full Resistance Tensor

In this section, we will formulate the lubrication resistance tensor using notations given by Kim. We begin by noting that the lubrication matrix has the same form as equation (1.2) and to avoid confusion between the overall resistance tensor and lubrication resistance tensor we will apply the "LUB" subscript when referring to the lubrication contributions

$$\mathbf{R}_{LUB} = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} & \tilde{\mathbf{G}} \\ \mathbf{B} & \mathbf{C} & \tilde{\mathbf{H}} \\ \mathbf{G} & \mathbf{H} & \mathbf{M} \end{pmatrix} \tag{2.8}$$

Combining equation (1.1) and (1.8) we can describe the resistance matrix and lubrication $\mathbf{F}^H$, $\mathbf{T}^H$, and $\mathbf{S}^H$ as

$$\begin{pmatrix} \mathbf{F}^H \\ \mathbf{T}^H \\ \mathbf{S}^H \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} & \tilde{\mathbf{G}} \\ \mathbf{B} & \mathbf{C} & \tilde{\mathbf{H}} \\ \mathbf{G} & \mathbf{H} & \mathbf{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}^\infty(\boldsymbol{x}) - \mathbf{u} \\ \boldsymbol{\Omega}^\infty - \boldsymbol{\omega} \\ \mathbf{E}^\infty \end{pmatrix} \tag{2.9}$$

By considering only the terms of $O(a/\delta+ln(a/\delta))$ in the lubrication interactions we acquire the following relations (see Appendix A for proofs)

$$\boldsymbol{A} \cdot \boldsymbol{U}^\infty(\boldsymbol{x}) + \tilde{\boldsymbol{B}} \cdot \boldsymbol{\Omega}^\infty + \tilde{\boldsymbol{G}} : \boldsymbol{E}^\infty = 0 \tag{2.10}$$

$$\boldsymbol{B} \cdot \boldsymbol{U}^\infty(\boldsymbol{x}) + \boldsymbol{C} \cdot \boldsymbol{\Omega}^\infty + \tilde{\boldsymbol{H}} : \boldsymbol{E}^\infty = 0 \tag{2.11}$$

$$\boldsymbol{G} \cdot \boldsymbol{U}^\infty(\boldsymbol{x}) + \boldsymbol{H} \cdot \boldsymbol{\Omega}^\infty + \tilde{\boldsymbol{M}} : \boldsymbol{E}^\infty = 0 \tag{2.12}$$

where

$$\boldsymbol{U}^\infty(x) = \boldsymbol{U}^\infty + \Omega^\infty \times \boldsymbol{x} + \boldsymbol{E}^\infty \cdot \boldsymbol{x} \tag{2.13}$$

The above relations imply that $U^\infty(x)$, $\boldsymbol{\Omega}^\infty$, and E^∞ provide no contribution to the lubrication forces or torque and we may simply equation (2.9) to

$$\begin{pmatrix} \mathbf{F}^H \\ \mathbf{T}^H \\ \mathbf{S}^H \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} & 0 \\ \mathbf{B} & \mathbf{C} & 0 \\ \mathbf{G} & \mathbf{H} & 0 \end{pmatrix} \begin{pmatrix} -\mathbf{u} \\ -\boldsymbol{\omega} \\ 0 \end{pmatrix} \tag{2.14}$$

This equation may be further simplified calculating the stresslet in a post processing step . Equation (1.10) becomes

$$\begin{pmatrix} \mathbf{F}^H \\ \mathbf{T}^H \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \begin{pmatrix} -\mathbf{u} \\ -\boldsymbol{\omega} \end{pmatrix} \tag{2.15}$$

$$\mathbf{F}^H = -\mathbf{G} \cdot \mathbf{u} - \mathbf{H} \cdot \boldsymbol{\omega} \tag{2.16}$$

The lubrication submatrices **A**, **B**, $\tilde{\mathbf{B}}$, and **C** can be defined in terms of smaller submatrices $\mathbf{A}_{\alpha\beta}$, $\mathbf{B}_{\alpha\beta}$, $\tilde{\mathbf{B}}_{\alpha\beta}$, and $\mathbf{C}_{\alpha\beta}$ describing the interactions between a pair of particles α and β. For $\alpha \neq \beta$ the submatrices describe the effect of particle α on particles β. For $\alpha = \beta$ (i.e. the diagonal terms) the submatrices describe the effect of particle α on itself due to its interaction with all other particles. In other words, the diagonal terms represent a sum of all pair interactions on particle α as described by equation (1.20) - (1.23). The following example illustrates how the submatrices fit into the larger lubrication matrices.

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1N} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{N1} & \mathbf{A}_{N2} & \cdots & \mathbf{A}_{NN} \end{pmatrix} \tag{2.17}$$

We will now describe in detail how the submatrices are formulated by first introducing the following notations:

$$r = \|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\| \tag{2.18}$$

$$\boldsymbol{d} = \frac{\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha}{r} \tag{2.19}$$

$$\delta = r - 2a \tag{2.20}$$

Next, we define the scalar resistance functions using the lubrication approximations given by Kim (pg. 279, pg. 283). We evaluate Kim's general expressions using β=1 in Kim's notation for equal sized particles and find

$$X^A = 6\pi\mu a\left(\frac{a}{4\delta} + \frac{9}{40}\ln\frac{a}{\delta}\right) \tag{2.21}$$

$$Y^A = 6\pi\mu a\left(\frac{1}{6}\ln\frac{a}{\delta}\right) \tag{2.22}$$

$$Y^B = -4\pi\mu a^2\left(\frac{1}{4}\ln\frac{a}{\delta}\right) \tag{2.23}$$

$$Y^C = \begin{cases} 8\pi\mu a^3\left(\frac{1}{5}\ln\frac{a}{\delta}\right), & \alpha = \beta \\ 8\pi\mu a^3\left(\frac{1}{20}\ln\frac{a}{\delta}\right), & \alpha \neq \beta \end{cases} \tag{2.24}$$

It is important to note that the above lubrication approximations X^A, Y^A, Y^B, and Y^C become 0 when $r < r_{LUB}$, where r_{LUB} is the cutoff range for the lubrication interaction between two particles. With this definition, the submatrices **A**,**B**, **B**$^-$, **C** become sparse matrices and the importance of sparsity will be discussed in the next section. We define three additional notations **D**, **I**, and **E** which will be used in conjunction with the scalar resistance functions to form the lubrication submatrices.

$$\mathbf{D} = d_i d_j \tag{2.25}$$

$$\mathbf{I} = \delta_{ij} \tag{2.26}$$

$$\mathbf{E} = \epsilon_{ijk} d_k \tag{2.27}$$

With the definitions above, the lubrication submatrices are formulated as follows

$$\mathbf{A}_{\alpha\beta} = \begin{cases} \sum_{\eta=1,\eta\neq\alpha}^{N_p} \mathcal{A}_{\alpha\eta}, & \alpha = \beta \\ -\mathcal{A}_{\alpha\beta}, & \alpha \neq \beta \end{cases} \tag{2.28}$$

$$\mathbf{B}_{\alpha\beta} = \begin{cases} \sum_{\eta=1,\eta\neq\alpha}^{N_p} \mathcal{B}_{\alpha\eta}, & \alpha = \beta \\ -\mathcal{B}_{\alpha\beta}, & \alpha \neq \beta \end{cases} \tag{2.29}$$

$$\tilde{\mathbf{B}}_{\alpha\beta} = \begin{cases} \sum_{\eta=1,\eta\neq\alpha}^{N_p} \tilde{\mathcal{B}}_{\alpha\eta}, & \alpha = \beta \\ -\tilde{\mathcal{B}}_{\alpha\beta}, & \alpha \neq \beta \end{cases} \tag{2.30}$$

$$\mathbf{C}_{\alpha\beta} = \begin{cases} \sum_{\eta=1,\eta\neq\alpha}^{N_p} \mathcal{C}_{\alpha\eta}, & \alpha = \beta \\ \mathcal{C}_{\alpha\beta}, & \alpha \neq \beta \end{cases} \tag{2.31}$$

where N_p is the number of particles.

The bold upper case **A**, **B**, **B˜**, and **C** on the LHS represent the full lubrication resistance submatrices. The scripted upper case A, B, B˜, C on the RHS represent a single pairwaise lubrication contribution which can be written in terms of scalars X^A, Y^A, and Y^B and matrices **D**, **I**, and **E** defined by equations (2.16 - 2.22).

$$\mathcal{A}_{\alpha\beta} = X^A\mathbf{D} + Y^A(\mathbf{I} - \mathbf{D}) \tag{2.32}$$

$$\mathcal{B}_{\alpha\beta} = Y^B\mathbf{E} \tag{2.33}$$

$$\tilde{\mathcal{B}}_{\alpha\beta} = \mathcal{B}_{\beta\alpha} \tag{2.34}$$

$$\mathcal{C}_{\alpha\beta} = Y^C(\mathbf{I} - \mathbf{D})) \tag{2.35}$$

Note that due to our definition of **E** in equation (1.22), the sign of the matrix **E** will change depending on the direction of the unit vector **d** such that $\mathbf{E}_{\alpha\beta} = -\mathbf{E}_{\alpha\beta}$

We will now redefine velocity and angular velocity as absolute velocity, u^0, and angular velocity, ω^0. We do so by expanding equation (2.21) and applying periodic
shift

$$r^2 = (x_{diff} + m_1L + m_2\dot{\gamma})^2 + (y_{diff} + m_2L)^2 + (z_{diff} + m_3L)^2 \tag{2.36}$$

where

$$\boldsymbol{x}_{diff} = \boldsymbol{x}_{\alpha} - \boldsymbol{x}_{\beta} \tag{2.37}$$

$$m_{(1,2,3)} = \begin{cases} 1 & \text{if } \boldsymbol{x}_{diff} > L \\ -1 & \text{if } \boldsymbol{x}_{diff} < -L \\ 0 & \end{cases} \tag{2.38}$$

where L is the system size. The absolute velocity and angular velocity are defined as

$$\boldsymbol{u}_{\alpha} = \boldsymbol{u}_{\alpha} \quad , \quad \boldsymbol{u}_{\beta} = \boldsymbol{u}_{\beta} + m_2 L \dot{\gamma}_i \tag{2.39}$$

$$\boldsymbol{\omega}_{\alpha} = \boldsymbol{\omega}_{\alpha} \quad , \quad \boldsymbol{\omega}_{\beta} = \boldsymbol{\omega}_{\beta} \tag{2.40}$$

DETAILED DESCRIPTION OF METHOD

.2.2 Pairwise Resistance Tensor

In the Introduction above, we denoted r_{LUB} as the cutoff range for the lubrication interactions between two particles. That is, the lubrication force applies to a pair of particles only if they have a interparticle separation of $r < r_{LUB}$. With this definition, $\mathbf{R}_{LUB}$ becomes a sparse matrix. Here we will devise a new storing scheme that takes advantage of the pairwise interactions and the sparsity of $\mathbf{R}_{LUB}$.

We begin by introducing a set of pairwise parameters that are analogous to the scalar resistance functions in equations (1.16) - (1.19) and the resistance submatrices in equations (1.28) - (1.31).

$$X^A_{PW} = 6\pi\mu a\left(\frac{a}{4\delta} + \frac{9}{40}\ln\frac{a}{\delta}\right) \tag{2.41}$$

$$Y^A_{PW} = 6\pi\mu a\left(\frac{1}{6}\ln\frac{a}{\delta}\right) \tag{2.42}$$

$$Y^C_{PW} = 8\pi\mu a^3\left(\frac{3}{40}\ln\frac{a}{\delta}\right) \tag{2.43}$$

$$\mathcal{A}_{PW} = X^A_{PW}\mathbf{D} + Y^A_{PW}(\mathbf{I}-\mathbf{D}) \tag{2.44}$$

$$\mathcal{B}_{PW} = -aY^A_{PW}\mathbf{E} \tag{2.45}$$

$$\mathcal{C}_{PW} = Y^C_{PW}(\mathbf{I}-\mathbf{D}) \tag{2.46}$$

Using these equations, we may write $\mathbf{R}_{LUB}$ in a pairwise fashion and store them in a newly defined pairwise resistance tensor $\mathbf{R}_{PW}$ of size $6N_{pair} \times 6$, where N_{pair} is the total number of near neighbor pairs with $r < r_{LUB}$. The tensor $\mathbf{R}_{PW}$ will aid us in calculating the hydrodynamic lubrication forces and torques in the following section.

$$\mathbf{R}_{PW} = \begin{pmatrix} \mathcal{A}_{PW} & 0 \\ \mathcal{B}_{PW} & \mathcal{C}_{PW} \end{pmatrix} \tag{2.47}$$

2.2.3 Force Calculation

Here, we will provide two different methods in calculating the hydrodynamic lubrication forces and torques: one using the full resistance tensor constructed with Kim's notation and one using the pairwise resistance tensor described in section 1.2.2. By the end of the section, we will demonstrate that our pairwise technique returns the same result as Kim's method while providing a more efficient approach in dealing with large scale systems. For simplicity, we will

analyze both methods using a case where only a pair of particles α and β exist in the system.

With Kim's method, we formulate the resistance tensor following the steps detailed in section 1.2.1 and substitute it into equation (1.11) to form the following

$$\begin{pmatrix} \mathbf{F}_\alpha \\ \mathbf{T}_\alpha \\ \mathbf{F}_\beta \\ \mathbf{T}_\beta \end{pmatrix}_{LUB} = \begin{pmatrix} \mathbf{A}_{\alpha\alpha} & \tilde{\mathbf{B}}_{\alpha\alpha} & \mathbf{A}_{\alpha\beta} & \tilde{\mathbf{B}}_{\alpha\beta} \\ \mathbf{B}_{\alpha\alpha} & \mathbf{C}_{\alpha\alpha} & \mathbf{B}_{\alpha\beta} & \mathbf{C}_{\alpha\beta} \\ \mathbf{A}_{\beta\alpha} & \tilde{\mathbf{B}}_{\beta\alpha} & \mathbf{A}_{\beta\beta} & \tilde{\mathbf{B}}_{\beta\beta} \\ \mathbf{B}_{\beta\alpha} & \mathbf{C}_{\beta\alpha} & \mathbf{B}_{\beta\beta} & \mathbf{C}_{\beta\beta} \end{pmatrix}_{LUB} \begin{pmatrix} -\mathbf{u}_\alpha \\ -\boldsymbol{\omega}_\alpha \\ -\mathbf{u}_\beta \\ -\boldsymbol{\omega}_\beta \end{pmatrix} \tag{2.48}$$

where $\mathbf{u}_\alpha$, $\mathbf{u}_\beta$, $\boldsymbol{\omega}_\alpha$, $\boldsymbol{\omega}_\beta$ are the velocities and the angular velocities of particles α and β.

With the definitions in equations (1.16) - (1.19) and equations (1.23) (1.26), we rewrite the above expression as

$$\begin{pmatrix} \mathbf{F}_\alpha \\ \mathbf{T}_\alpha \\ \mathbf{F}_\beta \\ \mathbf{T}_\beta \end{pmatrix}_{LUB} = \begin{pmatrix} X^A_{\alpha\alpha}\mathbf{D} + Y^A_{\alpha\alpha}(\mathbf{I}-\mathbf{D}) & -Y^B_{\alpha\alpha}\mathbf{E} & X^A_{\alpha\beta}\mathbf{D} + Y^A_{\alpha\beta}(\mathbf{I}-\mathbf{D}) & Y^B_{\alpha\beta}\mathbf{E} \\ Y^B_{\alpha\alpha}\mathbf{E} & Y^C_{\alpha\alpha}(\mathbf{I}-\mathbf{D}) & Y^B_{\alpha\beta}\mathbf{E} & Y^C_{\alpha\beta}(\mathbf{I}-\mathbf{D}) \\ X^A_{\alpha\beta}\mathbf{D} + Y^A_{\alpha\beta}(\mathbf{I}-\mathbf{D}) & -Y^B_{\alpha\beta}\mathbf{E} & X^A_{\alpha\alpha}\mathbf{D} + Y^A_{\alpha\alpha}(\mathbf{I}-\mathbf{D}) & Y^B_{\alpha\alpha}\mathbf{E} \\ -Y^B_{\alpha\beta}\mathbf{E} & Y^C_{\alpha\beta}(\mathbf{I}-\mathbf{D}) & -Y^B_{\alpha\alpha}\mathbf{E} & Y^C_{\alpha\alpha}(\mathbf{I}-\mathbf{D}) \end{pmatrix} \begin{pmatrix} -\mathbf{u}_\alpha \\ -\boldsymbol{\omega}_\alpha \\ -\mathbf{u}_\beta \\ -\boldsymbol{\omega}_\beta \end{pmatrix} \tag{2.49}$$

To solve for the lubrication forces and torques, matrix-vector multiplication is performed using packages such as LAPACK. For a large scale system, the formulation of the full $6N_P \times 6N_P$ resistance tensor is required with this method. This proves to be inefficient since the resistance tensor is mostly sparse.
For the pairwise method, two new parameters $\mathbf{u}_{diff}$ and ω_{diff} are defined

$$\mathbf{u}_{diff} = (\mathbf{u}_\beta - \mathbf{u}_\alpha) - (\boldsymbol{\omega}_\beta + \boldsymbol{\omega}_\alpha) \times a\mathbf{d} \tag{2.50}$$

$$\boldsymbol{\omega}_{diff} = \boldsymbol{\omega}_\beta - \boldsymbol{\omega}_\alpha \tag{2.51}$$

The vectors $\mathbf{u}_{diff}$ and $\boldsymbol{\omega}_{diff}$ are the velocity difference in shearing and rolling at the surface of the particles. We then substitute $\mathbf{u}_{diff}$, $\boldsymbol{\omega}_{diff}$ and the pairwise resistance tensor $\mathbf{R}_{PW}$ into equation (1.11) to derive our expressions for calculating the lubrication forces and torques in a pairwise fashion.

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{T} \end{pmatrix}_\alpha = \begin{pmatrix} \mathcal{A}_{PW} & 0 \\ \mathcal{B}_{PW} & \mathcal{C}_{PW} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{diff} \\ \boldsymbol{\omega}_{diff} \end{pmatrix} \tag{2.52}$$

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{T} \end{pmatrix}_\beta = \begin{pmatrix} \mathcal{A}_{PW} & 0 \\ -\mathcal{B}_{PW} & \mathcal{C}_{PW} \end{pmatrix} \begin{pmatrix} -\mathbf{u}_{diff} \\ -\boldsymbol{\omega}_{diff} \end{pmatrix} \tag{2.53}$$

Next, expand equations (1.42) and (1.43) and collect terms to obtain the following

$$\begin{pmatrix} \mathbf{F}_\alpha \\ \mathbf{T}_\alpha \\ \mathbf{F}_\beta \\ \mathbf{T}_\beta \end{pmatrix} = \left(\begin{array}{cccc} X^A_{PW}\mathbf{D} + Y^A_{PW}(\mathbf{I}-\mathbf{D}) & aY^A_{PW}\mathbf{E} & -X^A_{PW}\mathbf{D} - Y^A_{PW}(\mathbf{I}-\mathbf{D}) & aY^A_{PW}\mathbf{E} \\ -aY^A_{PW}\mathbf{E} & \left[a^2Y^A_{PW} + Y^C_{PW}\right](\mathbf{I}-\mathbf{D}) & aY^A_{PW}\mathbf{E} & \left[a^2Y^A_{PW} - Y^C_{PW}\right](\mathbf{I}-\mathbf{D} \\ -X^A_{PW}\mathbf{D} - Y^A_{PW}(\mathbf{I}-\mathbf{D}) & -aY^A_{PW}\mathbf{E} & X^A_{PW}\mathbf{D} + Y^A_{PW}(\mathbf{I}-\mathbf{D}) & -aY^A_{PW}\mathbf{E} \\ -aY^A_{PW}\mathbf{E} & \left[a^2Y^A_{PW} - Y^C_{PW}\right](\mathbf{I}-\mathbf{D}) & aY^A_{PW}\mathbf{E} & \left[a^2Y^A_{PW} + Y^C_{PW}\right](\mathbf{I}-\mathbf{D} \\ & & (2.54) & \end{array} \right) \tag{2.54}$$

By comparing equation (1.44) with equation (1.39), we can demonstrate that our pairwise scalar resistance functions are analogous to the scalar resistance used by Kim as seen in equations (1.45)-(1.49).

$$X^A_{\alpha\alpha} = X^A_{PW} \quad , \quad X^A_{\alpha\beta} = -X^A_{PW} \tag{2.55}$$

$$Y^A_{\alpha\alpha} = Y^A_{PW} \quad , \quad Y^A_{\alpha\beta} = -Y^A_{PW} \tag{2.56}$$

$$Y^B_{\alpha\alpha} = -aY^A_{PW} \quad , \quad Y^B_{\alpha\beta} = aY^A_{PW} \tag{2.57}$$

$$Y^C_{\alpha\alpha} = a^2Y^A_{PW} + Y^C_{PW} \quad , \quad Y^C_{\alpha\beta} = a^2Y^A_{PW} - Y^C_{PW} \tag{2.58}$$

Using our pairwise method, we may calculate the lubrication forces and torques using only the pairwise tensors for each pair instead of the full resistance tensor and the total lubrication forces and torques may be obtained by summing up each pairwise contributions.

3. Brownian Force

The Brownian forces and torques are obtained from the following expressions which satisfy the fluctuation dissipation theorem

$$\left\langle \begin{pmatrix} \mathbf{F}^B \\ \mathbf{T}^B \end{pmatrix} \right\rangle = 0 \tag{3.1}$$

$$\tag{3.2}$$

$$\left\langle \begin{pmatrix} \mathbf{F}^B \\ \mathbf{T}^B \end{pmatrix}_\alpha \begin{pmatrix} \mathbf{F}^B \\ \mathbf{T}^B \end{pmatrix}_\beta \right\rangle = \frac{2kT\mathbf{R}_{FLD}}{\Delta t} \tag{3.3}$$

Since $\mathbf{R}_{FLD} = \mathbf{R}_{ISO} + \mathbf{R}_{LUB}$, we may express the brownian forces and torques as a sum of isotropic and lubrication contributions

$$\begin{pmatrix} \mathbf{F}^B \\ \mathbf{T}^B \end{pmatrix} = \begin{pmatrix} \mathbf{F}^B_{ISO} \\ \mathbf{T}^B_{ISO} \end{pmatrix} + \begin{pmatrix} \mathbf{F}^B_{LUB} \\ \mathbf{T}^B_{LUB} \end{pmatrix} \tag{3.4}$$

The Brownian isotropic contributions can be written as

$$\begin{pmatrix} \mathbf{F}^B_{ISO} \\ \mathbf{T}^B_{ISO} \end{pmatrix} = \sqrt{\frac{2kT}{\Delta t}} \mathbf{J} \cdot \mathbf{\Phi} \tag{3.5}$$

$$\mathbf{J} \cdot \mathbf{J}^T = \begin{pmatrix} R_{0_{FU}}\mathbf{I} & 0 \\ 0 & R_{0_{T\Omega}}\mathbf{I} \end{pmatrix} \tag{3.6}$$

The Brownian lubrication contributions are calculated in a pairwise fashion similar to the method described in section 1.2.3.

$$\begin{pmatrix} \mathbf{F}^B_{LUB} \\ \mathbf{T}^B_{LUB} \end{pmatrix}_\alpha = \sqrt{\frac{2kT}{\Delta t}} \mathbf{K}_{\alpha\beta} \cdot \mathbf{\Psi} \tag{3.7}$$

$$\begin{pmatrix} \mathbf{F}^B_{LUB} \\ \mathbf{T}^B_{LUB} \end{pmatrix}_\beta = \sqrt{\frac{2kT}{\Delta t}} \mathbf{K}_{\beta\alpha} \cdot \mathbf{\Psi} \tag{3.8}$$

$$\mathbf{K} \cdot \mathbf{K}^T = \begin{pmatrix} \mathbf{A} & \tilde{\mathbf{B}} \\ \mathbf{B} & \mathbf{C} \end{pmatrix}_{LUB} \tag{3.9}$$

Note that the **Ψ** in equations (3.7) and (3.8) are the same vector. The vectors **Φ** and **Ψ** are uncorrelated random numbers with zero mean and unit variance such that

$$\langle \Phi_i \rangle = 0 \quad , \quad \langle \Psi_i \rangle = 0 \tag{3.10}$$

$$\langle \Phi_i \Phi_j \rangle = \delta_{ij} \quad , \quad \langle \Psi_i \Psi_j \rangle = \delta_{ij} \tag{3.11}$$

$$\langle \Phi_i \Psi_j \rangle = 0 \tag{3.12}$$

With expressions for Brownian forces and torques defined, we proceed to set up the **J** and **K** matrices. The matrix **J** is trivially written as

$$\mathbf{J} = \begin{pmatrix} \sqrt{R_{0_{FU}}}\mathbf{I} & 0 \\ 0 & \sqrt{R_{0_{T\Omega}}}\mathbf{I} \end{pmatrix} \tag{3.13}$$

The matrix **K** is constructed using the pairwise parameters defined by equations (1.31) - (1.36). First, let N_{Pair} be the number of pairs of particles with interparticle separation of $r < r_{LUB}$. We then designate vectors γ^1 and γ^2 to store the indices of the particles in each pair such that pair p consists of particles γ^1_p and γ^2_p. Next, the matrix **K** is defined in terms of smaller 6 × 6 submatrices.

$$\mathbf{K}_{\alpha p} = \begin{pmatrix} \mathbf{K}^A_{\alpha p} & 0 \\ \mathbf{K}^B_{\alpha p} & \mathbf{K}^C_{\alpha p} \end{pmatrix} \tag{3.14}$$

The submatrices $\mathbf{K}^A_{\alpha p}$, $\mathbf{K}^B_{\alpha p}$ and $\mathbf{K}^C{}_{\alpha p}$ describe the lubrication contribution of pair p to the Brownian force and torque exerted on particle α and are written as

$$\mathbf{K}^A_{\alpha p} = \begin{cases} \boldsymbol{\mathcal{K}}^A & , \quad \alpha = \gamma^1_p \\ -\boldsymbol{\mathcal{K}}^A & , \quad \alpha = \gamma^2_p \\ \mathbf{0} & , \quad otherwise \end{cases} \tag{3.15}$$

$$\mathbf{K}^B_{\alpha p} = \begin{cases} \boldsymbol{\mathcal{K}}^B & , \quad \alpha = \gamma^1_p \\ -\boldsymbol{\mathcal{K}}^B & , \quad \alpha = \gamma^2_p \\ \mathbf{0} & , \quad otherwise \end{cases} \tag{3.16}$$

$$\mathbf{K}^C_{\alpha p} = \begin{cases} \boldsymbol{\mathcal{K}}^C & , \quad \alpha = \gamma^1_p \\ -\boldsymbol{\mathcal{K}}^C & , \quad \alpha = \gamma^2_p \\ \mathbf{0} & , \quad otherwise \end{cases} \tag{3.17}$$

where K^A, K^B, and K^C are the block Cholesky of the pairwise submatrices in equations (1.34)-(1.36) written in terms of the pairwise scalar resistance functions.

$$\boldsymbol{\mathcal{K}}^A = \sqrt{X^A_{PW}}\mathbf{D} + \sqrt{Y^A_{PW}}(\mathbf{I} - \mathbf{D}) \tag{3.18}$$

$$\boldsymbol{\mathcal{K}}^B = -\frac{aY^A_{PW}}{\sqrt{Y^A_{PW}}}\mathbf{E} \tag{3.19}$$

$$\boldsymbol{\mathcal{K}}^C = \sqrt{Y^C_{PW}}(\mathbf{I} - \mathbf{D}) \tag{3.20}$$

As an important reminder, we note that due to our definition of E in equation (1.22), $E = \epsilon_{ijk} d_k$ the sign of E changes with respect to the indices of the pair such that $\mathbf{E}_{\gamma_p^1 \gamma_p^2} = -\mathbf{E}_{\gamma_p^2 \gamma_p^1}$ .With the submatrices defined, the full K matrix is written as

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1N_{Pair}} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{K}_{2N_{Pair}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{N1} & \mathbf{K}_{N2} & \cdots & \mathbf{K}_{NN_{Pair}} \end{pmatrix} \tag{3.21}$$

Similar to how we computed the hydrodynamic lubrication forces and torques, we may sum up each pairwise contributions to acquire the total Brownian lubrication forces and torques instead of using the full K matrix.
Npair

$$\mathbf{F}^B_{\gamma_p^1} = \sum_{q=1}^{N_{pair}} \mathbf{F}^B_p \quad , \quad \gamma_q^1 = \gamma_p^1 \tag{3.22}$$

$$\mathbf{F}^B_{\gamma_p^2} = \sum_{q=1}^{N_{pair}} \mathbf{F}^B_p \quad , \quad \gamma_q^2 = \gamma_p^2 \tag{3.23}$$

$$\mathbf{T}^B_{\gamma_p^1} = \sum_{q=1}^{N_{pair}} \mathbf{T}^B_p \quad , \quad \gamma_q^1 = \gamma_p^1 \tag{3.24}$$

$$\mathbf{T}^B_{\gamma_p^2} = \sum_{q=1}^{N_{pair}} \mathbf{T}^B_p \quad , \quad \gamma_q^2 = \gamma_p^2 \tag{3.25}$$

4. Overlap Correaction

The overlap correction algorithm removes the overlaps between particles after each Runge Kutta time step without jeopardizing the physics that the main algorithm is trying to capture. The algorithm is described below.

1) Determine the minimum interparticle separation

$$r_{min} = min(\|\boldsymbol{x}_\beta^* - \boldsymbol{x}_\alpha^*\|) \tag{4.1}$$

where x^* is the position of the particles at the end of RungeKutta

2) If $r_{min} \geq 2a$, go to step 10.

3) Calculate the largest gap, δ_{OC}, for which hydrodynamic interactions will be applied

$$\delta_{OC} = 5(2 - r_{min}) \tag{4.2}$$

4) Calculate the minimum numerical gap, $\delta_{NUM,OC}$, for evaluating hydrodynamics interactions

$$\delta_{NUM,OC} = \frac{1}{10}\delta_{OC} \tag{4.3}$$

5) Calculate forces on particles with overlaps using arbitrary time step Δt_{OC}

$$\boldsymbol{F}_\alpha^{OC} = \frac{1.01}{4}\frac{6\pi\mu a^2}{\Delta t_{OC}} \sum_{\beta=1,\beta\neq\alpha}^{N} \begin{cases} \frac{\delta}{\delta_{NUM,OC}}\boldsymbol{d} & \delta < 0 \\ 0 & \delta \geq 0 \end{cases} \tag{4.4}$$

The force moves the overlapped particles apart to a distance that is equal to the size of the overlap. The 1.01 factor moves the particles an additional distance that is 1% of the size of the overlap.

6) Calculate $R_{0,OC}$ and $X^A_{11,OC}$

$$R_{0,OC} = \frac{1}{1000}\frac{3\pi a^2}{2\delta_{NUM,OC}} \tag{4.5}$$

$$\boldsymbol{X}^A_{11,OC} = \begin{cases} \frac{3\pi a^2}{2\delta_{NUM,OC}} & \delta < \delta_{NUM,OC} \\ \frac{3\pi a^2}{2\delta} & \delta_{NUM,OC} \leq \delta < \delta_{OC} \\ 0 & \delta \geq \delta_{OC} \end{cases} \tag{4.6}$$

7) Calculate velocity

$$\boldsymbol{U}^{OC} = \mu(R_{0,OC}\boldsymbol{I} + \boldsymbol{A}^{OC})^{-1} \cdot \boldsymbol{F}^{OC} \tag{4.7}$$

8) Calculate new positions

$$\boldsymbol{x}^*(t+\Delta t) = \boldsymbol{x}^*(t+\Delta t) + \boldsymbol{U}^{OC}\Delta t_{OC} \tag{4.8}$$

9) Go to Step 1

10) Calculate final position

$$\boldsymbol{x}(t+\Delta t) = \boldsymbol{x}^*(t+\Delta t) \tag{4.9}$$

4.0.4 Additional Notes

The algorithm for overlap correction assembles a resistance tensor using the same algorithm from the main code with following modification in the variables used:

1) use $R_{o,OC}$ instead of R_o
2) use $\delta NUM,OC$ instead of δNUM
3) use δ_{OC} for hydrodynamic cutoff instead of r_{LUB}

The isotropic resistance, R_{OC}, calculated in equation (4.5) is intended to serve as safety net in the construction of the resistance tensor to prevent dividing by zero for the $\delta \geq \delta_{OC}$ case and does not have any significant contribution to the resistance tensor itself. Using the resistance tensor and the force calculated from equation (4.4), we can solve for the velocity to remove the overlaps in the system simultaneously.

In the case where only a pair of particles overlap without any near neighbor contribution, the overlap should be resolved in a single step as demonstrated below:

$$\boldsymbol{F}_{OC} = 1.01\frac{3\pi\mu a^2}{2}\frac{1}{\Delta t_{OC}}\left(\frac{\delta}{\delta_{NUM}}\right)\boldsymbol{d} \tag{4.10}$$

$$X^{A}_{11,OC} = \frac{3\pi a^2}{2\delta_{NUM,OC}} \tag{4.11}$$

$$\boldsymbol{U}_{OC} = \left(\mu R_{0,OC}\boldsymbol{I} + \mu \boldsymbol{A}_{OC}\right)^{-1} \cdot \boldsymbol{F}_{OC} \tag{4.12}$$

becomes

$$\boldsymbol{U}_{OC} = \mu\left(\frac{1}{1000}\frac{3\pi a^2}{2\delta_{NUM,OC}} + \frac{3\pi a^2}{2\delta_{NUM,OC}}\right)^{-1} \cdot 1.01\frac{3\pi\mu a^2}{2}\frac{1}{\Delta t_{OC}}\left(\frac{\delta}{\delta_{NUM}}\right)\boldsymbol{d} \tag{4.13}$$

Since the isotropic term has essentially no contribution to the resistance tensor we can effectively reduce the equation to

$$\boldsymbol{U}_{OC} = 1.01\left(\frac{-\delta}{\Delta t}\right) \tag{4.14}$$

RESULTS

The main thrusts of this project have been

1. the refinement of the theoretical underpinnings of the Fast Lubrication Dynamics algorithm,
2. tests for different strategies for fast preconditioners for improved convergence of the GMRES linear solves in the implicit form of the FLD algorithm
3. algorithm reformulation in terms of pairwise interaction loops in place of direct particle summation loops to optimize the algorithm for OpenMP on shared memory multicore/multiprocessor workstations

In thrust (1), we introduced changes in the analytical development to facilitate simpler cross-verification among the formulations of Kim and Karrila, Ball and Melrose and existing Sandia LAMMPS (Large-Scale Atomic Molecular Massively Parallel Simulator) library routines provided by Amit Kumar in our group at UIUC. The major results are presented in the Detailed Description section of the report above. In particular, there are three areas where the FLD approach has been refined and documented in these pages. First, we have for the first time given concise expressions for the general resistance tensor submatrices in a pairwise interaction format (eqs 2.41 through 2.58 above) which facilitate the rewriting of the algorithm in a pairwise loop formulation. We have derived the equivalence relations between these submatrices and the global particle formulation used in Kim's monograph allowing easy verification with standard published sources. Second, we have reformulated the Brownian force computation (eqs 3.1 through 3.25) above utilizing a consistent tensor formulation. Previous versions of the Brownian force algorithm both in our group (Kumar and LAMMP library routine) and in the early Ball and Melrose paper used a somewhat ad hoc formulation which required the use of a random initial vector to generate a local orthogonal triad in computing Brownian forces and torques. Third we introduced an improved overlap correction algorithm (eqs 4.1 through 4.14) which yields fewer repeat overlaps and larger time steps in Brownian simulations at low Peclet number and high volume fraction.

In thrust (2), we tried numerous different approaches including right handed, left handed and symmetric preconditioners (as discussed in the Saad monograph) with different choices for the preconditioner matrix. Basically we explored different combinations of sparse pairwise incomplete LU preconditioners as well as approaches including the next nearest neighbor contributions. Ultimately, despite exhaustive testing, this proved to be an unsuccessful research thrust. We concluded that the block diagonal, incomplete LU based on pairwise contributions which we had previously developed proved the best choice. The block diagonal preconditioners showed best overall efficiency, and a new higher order block form was utilized to maintain better

consistency with the system matrix. Simulations continued to affirm the superior performance of the implicit solver vs explicit solver.

In thrust (3), we restructured the main computational loops in the code to improve efficiency and to better exploit the sparse matrix structure. This facilitates smaller storage and data transfer for distributed processing and also paves the way for possible future tests on GPU assisted computations.

Improved Parallel Structure for OpenMP on Shared Memory Workstations. The previous algorithm was based on loop structures over the global particle number, suitable for serial computation or for massively parallel systems where memory bandwidth and latency were of prime concern. Some redundancy in computation was acceptable to minimize these more serious bottlenecks. For OpenMP, computations for all lubrication terms were performed in loops summing on the global *near neighbor count*. The precomputed data structures were then accessed in loops summing over *global particle number*. This hybrid computation allowed efficient use of OpenMP data typing and prevents data races and collisions and optimizes gather/scatter computations. This approach has been implemented and verified. Performance profiling was conducted both on older chipsets (Intel Q6600 quad, Xeon E5410 dual quad) and newer chipsets (I7-3770 quad, Xeon E5-1650 hex). We found that code revision was required to fully exploit the increased memory bandwidth and hyper-threading of the latest chipsets. The algorithm restructuring and refinement also included introduction of the higher order (log δ) hydrodynamic lubrication interactions.

Overall, we found the new hybrid loop structure performed at the highest efficiency with up to 80% utilization when running on 8 processors. Additional OpenMP testing on new 12 and 16 core workstation architectures is desirable. We found that the improved performance of the hybrid loops structure depended on the improved memory bandwidth of modern processors. Older processors such as the Q6600 quad chip lacked sufficient bandwidth to show any improvement. Fortunately all modern Core2 and later architectures showed significant performance improvements for the new hybrid loops, and Xeon processors commonly used in workstations always have sufficient memory bandwidth.

DISCUSSION

Overall, thrusts (1) and (3) above proved quite successful, while thrust (2) was unable to improve on the already efficient approach of our existing preconditioners. We were pleased with the results of these efforts. In the original scope of this project, after completing the theoretical work and algorithm development, we had hoped to complete a series of simulations on consolidation flows to examine how the particle distribution, microstructure and heterogeneities in the stress distribution were affected by processing conditions. The desired parametric considerations include rate of compression, Peclet number, interparticle forces (both attraction and compression) and degree of compression indicated by phi_initial/phi_final. This remains an important area for continued research. In our research group, Bybee conducted an extensive system of simulations on gelation in dense colloidal systems with both short attractive and long range repulsive interparticle forces. The gelation occurred under quiescent conditions. Bybee showed that the relative strengths of the attractive and repulsive forces had a profound effect on the properties of the final gel in terms of number of bonds, elastic modulus, brittleness of gel and ability to self heal – i.e. to reform interparticle bonds on time scales smaller than rheometric time scales associated with bond breakage. All of these issues are of profound importance in consolidation flows, where the same physiochemical processes are relevant but with the added complication of strong macroscopic hydrodynamic flows as opposed to the quiescent conditions in gels.

As we planned the simulations for the consolidation flows, the first step was verification against Bybee's results for the quiescent systems. The second step was verification against Amit Kumar's results for rheological properties in simple shear flows. For this latter test, we also had access to Qingjun Meng's results using the more expensive Particle Mesh Ewald Stokesian Dynamics algorithm. As we were planning the extension to the consolidation simulations, we began to considered possible limitations of the FLD approach when it is utilized for the consolidation flows where a linear flow field with compression along 1, 2 or 3 axes is considered. In these flows, as the particles are forced to higher volume fractions by moving permeable boundaries, the solvent must pass through the particle suspension similar to the action of a fluid passing through a fixed permeable material. It is well known that these flows present the greatest challenge for FLD (as well as the more expensive Stokesian dynamics approach). The great strength of the FLD (and Stokesian dynamics) algorithm is that it accurately calculates the strong lubrication forces which dominate in simple shear flows or in computation of particle diffusivity. In each case, there is significant relative particle motion and strong lubrication forces. At the other extreme, for dilute systems, there are negligible lubrication interactions, but FLD is calibrated against accurate results (theoretical, experimental or computational) for particle mobility (or equivalently self-diffusivity). The far field contributions to particle mobility are the result of a long range average over

many particles and thus are insensitive to particle microstructure. It is for this reason that we are able to use Monte Carlo configurations for calibration and yet still achieve excellent accuracy from dilute through concentrated suspensions in both diffusivity and rheology simulations.

Now consider flow through a fixed porous medium composed of immobilized particles in a suspension. There is zero relative motion of the particles and thus zero lubrication force. There is however significant fluid motion relative to the fixed particles, and the average particle resistance is characterized by the permeability of the medium. The permeability is a distinct material property which is not directly correlated with the particle mobility or self-diffusivity. Qualitatively permeability shows the same increasing or decreasing trend as particle mobility, but with different scaling and dependence on suspension volume fraction and microstructure. Most notably, particle mobility goes to zero at a finite volume fraction where the permeability retains a significant finite value. Results from Stokesian dynamics (and also FLD) give a factor of 2 error in estimating the permeability for a suspension of fixed spheres at volume fraction 50% whereas both algorithms do a good job for predicting the particle mobility (diffusivity) and excellent job predicting the shear viscosity.

Next, consider a consolidation flow. Here there is *some* relative motion of the particles but at higher degrees of compression, the flow resistance is dominated more and more by the flow through the porous particle network, and the FLD/Stokesian dynamics algorithms begin to lose their accuracy – and appeal. The problem is exacerbated if one is interested in predicting particle distributions for bidisperse suspensions often used for in high throughput processing for producing novel materials with different particle types with distinct physiochemical or electronic properties. If one particle species is distinctly smaller, it may have a tendency to be "washed" out of the medium by the departing solvent. The degree of this behavior will be poorly predicted if the permeability is poorly computed by the simulation. This is a significant concern if unmodified FLD or Stokesian dynamics algorithms are used for these simulations. Fortunately, there is a straightforward and efficient generalization of FLD which addresses these concerns. This is discussed in the section below.

ANTICIPATED IMPACT

Based on the successful algorithmic changes introduced in this project, we now have an efficient verified algorithm for conducting FLD simulations on multiprocessor workstations - a desirable platform for making this approach accessible to a broad range of researchers in industry and in academia. FLD simulations have proved their worth in studies of gel formation, diffusivity, microrheology and rheological flows. In the next phase of our research, we will continue our efforts to adapt these algorithms for application to processing flows and high throughput manufacturing.

The primary new features to be addressed in the next version of the algorithm are

- improved accuracy for compressional flows in consolidation processing
- inclusion of permeability solid boundaries
- extension to bidisperse suspensions
- inclusion of arbitrary boundary shapes and flow of suspensions in arbitrary macroscopic flow fields

For compressional flows, we need to modify the FLD algorithm to make it accurate for flow in porous media, while retaining the accurate calculation of lubrication forces and the calibration for particle mobility. We have derived expressions showing that this may be accomplished by inclusion of a third term in the FLD resistance tensor. Briefly, the standard FLD approach includes an analytic asymptotic lubrication term and a block diagonal tensor for the isotropic resistance calibrated to the mean mobility tensor of a single particle. To these terms we add a third contribution which captures the mean resistance of all particles and allows calibration to experimental or computed permeability data. The inclusion of this third resistance component introduces negligible computation cost and preserves the accuracy of the lubrication and mobility terms. While the analysis is complete and implementation is straightforward, we have not yet completed implementation and verification for the new approach. This is the next step in our efforts.

The next two items – solid boundaries and bidisperse suspension involve no new concepts or theoretical formulations and versions have been previously implemented by members of our research group. These generalizations need to be included in our updates algorithms. The final generalization – extension to include arbitrary macroscopic flows and boundaries is of high impact for simulations of processing flows and industrial applications. While several pathways look promising, that generalization is further down the line.

With the near term inclusion of the first three extensions – compressional flows, solid boundaries and bidisperse suspensions, we will move our focus to conducting computational studies of the microstructure, particle distribution and stress distribution in materials formed through consolidation processing of a suspension - e.g. from approximately 30% volume fraction to perhaps 55% to 60% volume fraction. Much as we found in our studies of quiescent gels, we expect that suspension microstructure, stress chains and stress distributions will be highly dependent on processing conditions such as compression rate or compressive stress applied to suspension, interparticle forces and bidisperse particle size ratios and relative volume fractions. Brownian motion for submicron particles or sedimentation for larger particles may also play a role. The particle microstructure or heterogeneous stress distribution in the consolidated material is of great significance because the next step in processing many of these materials is a drying operation. Non-uniformities or anomalies in microstructure may develop into cracks when subject to large stresses arising from interfacial tension at the moving drying front.

Steven Chen, the Sandia Fellow working on this research project was funded by Sandia through June 2014. After a period of reduced effort owing to personal circumstances, Steven will be returning to 100% effort on this project in December 2014. We currently have funding for the next 18 months (through unrestricted funding with Higdon's endowed chair) which should see Steven through to the conclusion of his PhD. We hope to demonstrate a successful effort in applying the new generalized FLD algorithms and in analyzing important features of consolidation flows and the resulting microstructure and material properties. At that time, we will approach Sandia National Labs concerning possible follow-on funding for this project.

Based on discussions with colleagues at Sandia, we believe that the FLD computational algorithms have a bright future for continued evolution and application to important processing flows of interest to the labs. Particular areas of application include the processing and optimization of microstructure for power source materials (i.e. batteries), energetic materials, and ceramic piece parts. Manufacturing processes for these applications involve the flow, rheology and drying of colloidal suspensions. This computational approach looks particularly useful in analyzing the microstructure and microscale particle distribution arising in processes critical for high throughput manufacturing of novel materials.

Conclusion

This project has lead to significant advances in the versatile Fast Lubrication Dynamics (FLD) algorithm for simulation of concentrated suspensions in fluid flow fields. The underlying mathematical formulation has been clarified and adapted to facilitate alternative parallel implementations. It has yielded significant improvements in parallel efficiency in an OpenMP shared memory environment. Hitherto unforeseen challenges were identified in applying this approach or other competitive approaches (Stokesian dynamics) to certain realistic processing flows involving one or more compressional flow axes. Novel modifications to the computational approach have been developed to address these challenges and are currently being implemented in the ongoing pursuit of this project.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.