Outline for a Lubrication Corrected Euler Maruyama Type Scheme

Brennan Sprinkle, Courant Institute

I. GENERATING THE MODIFIED MOBILITY

The lubrication corrected body mobility $\overline{\boldsymbol{\mathcal{N}}}$ can be stated as

$$\begin{aligned}\overline{\boldsymbol{\mathcal{N}}} &= \boldsymbol{\mathcal{N}} \cdot [\boldsymbol{I} + \boldsymbol{\Delta}_{MB} \cdot \boldsymbol{\mathcal{N}}]^{-1} \\ &= \left[\boldsymbol{\mathcal{N}}^{-1} + \boldsymbol{\Delta}_{MB}\right]^{-1} \end{aligned} \tag{I.1}$$

where in (I.1), $\boldsymbol{\mathcal{N}}$ is the uncorrected body mobility computed by the standard multiblob method [1]. $\Delta_{MB} = \boldsymbol{\zeta}^{sup} - \boldsymbol{\zeta}^{sup}_{MB}$ is the lubrication correction for the resistance matrix where $\boldsymbol{\zeta}^{sup}$ is given analytically as a pairwise approximation to the resistance of nearly touching spheres and $\boldsymbol{\zeta}^{sup}_{MB}$ is it's counterpart, computed using the multiblob method. The idea is to subtract off whatever the multiblob method computes for nearly touching spheres and replace it with an exact analytic formula wherever appropriate.

A. Formulation as a Big Linear System

In the absence of fluctuations, computing the lubrication corrected dynamics of a rigid multiblob, $\boldsymbol{U}$, simply amounts to the application of $\overline{\boldsymbol{\mathcal{N}}}$ to some prescribed forces and torques $\boldsymbol{F}$. That is,

$$\boldsymbol{U} = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} = \left[\boldsymbol{\mathcal{N}}^{-1} + \boldsymbol{\Delta}_{MB}\right]^{-1} \boldsymbol{F}.$$

We wish to reformulate the computation of $\boldsymbol{U}$ as the solution of a linear system so that a preconditioned Krylov method can be used to approximately yet efficiently apply $\overline{\boldsymbol{\mathcal{N}}}$ to $\boldsymbol{F}$. While there are certainly a few possible ways to do this, one way I've liked is analgous to the un-lubrication corrected linear system used to compute the action of $\boldsymbol{\mathcal{N}}$. That is, solving

$$\begin{pmatrix} \boldsymbol{\mathcal{M}} & -\boldsymbol{\mathcal{K}} \\ \boldsymbol{\mathcal{K}}^T & \boldsymbol{\Delta}_{MB} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{F} \end{pmatrix}, \tag{I.2}$$

gives

$$\boldsymbol{U} = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} = \left[\boldsymbol{\mathcal{N}}^{-1} + \boldsymbol{\Delta}_{MB}\right]^{-1} \boldsymbol{F}, \tag{I.3}$$

$$\boldsymbol{\lambda} = \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} = \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}} \left[\boldsymbol{\mathcal{N}}^{-1} + \boldsymbol{\Delta}_{MB}\right]^{-1} \boldsymbol{F}. \tag{I.4}$$

Equation (I.2) has a nonzero $(2, 2)$ block and therefore will require some more thought about how to efficiently solve. It will also be useful fro us to note that the solution of

$$\begin{pmatrix} \boldsymbol{\mathcal{M}} & -\boldsymbol{\mathcal{K}} \\ \boldsymbol{\mathcal{K}}^T & \boldsymbol{\Delta}_{MB} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_s \\ \boldsymbol{U}_s \end{pmatrix} = \begin{pmatrix} -\breve{\boldsymbol{u}} \\ \boldsymbol{F}_s \end{pmatrix}, \tag{I.5}$$

can be written as

$$\boldsymbol{U}_s = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F}_s + \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\breve{\boldsymbol{u}} \tag{I.6}$$

B. Preconditioning (I.2)

In principle, the nonzero $(2, 2)$ block should stabilize the saddle point problem and increase efficiency although this is just from a brief anecdote in talking to Boyce and Charles, I'm not sure of how this comes out. I've seen something called Hermitian and skew-Hermitian splitting (HSS) [2], where (I.2) is solved by splitting the linear system according to

$$\begin{pmatrix} \boldsymbol{\mathcal{M}} & -\boldsymbol{\mathcal{K}} \\ \boldsymbol{\mathcal{K}}^T & \boldsymbol{\Delta}_{MB} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Delta}_{MB} \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} & -\boldsymbol{\mathcal{K}} \\ \boldsymbol{\mathcal{K}}^T & \boldsymbol{0} \end{pmatrix} = \boldsymbol{H} + \boldsymbol{S},$$

and iterating the solution as

$$\begin{aligned}(\alpha \boldsymbol{I} + \boldsymbol{H})\, \boldsymbol{x}^{k+1/2} &= (\alpha \boldsymbol{I} - \boldsymbol{S})\, \boldsymbol{x}^{k} + \boldsymbol{b} \\ (\alpha \boldsymbol{I} + \boldsymbol{S})\, \boldsymbol{x}^{k+1} &= (\alpha \boldsymbol{I} - \boldsymbol{H})\, \boldsymbol{x}^{k+1/2} + \boldsymbol{b}.\end{aligned}$$

Some problems here are that the block diagonal system, $\boldsymbol{H}$, would require the solution of a resistance problem. Since this in an inner solve as a part of an iterative method, I suspect that it won't hurt that bad and a loose tolerance with the block diagonal preconditioner will be well tolerated. However, the block diagonal solve also requires the inversion (or approximate inversion) of $\boldsymbol{\Delta}_{MB}$ which does not have much in the way of general structure to my eye and will not be vary small for many bodies. Regardless of what method is being used, the approximate inversion of $\boldsymbol{\Delta}_{MB}$ seems to me to be the big issue in designing a good preconditioner.

II. GENERATING THE BROWNIAN INCREMENT

This isn't so different from how we were doing it without the lubrication but there is one little trick that I think is worth mentioning. Using equation (I.6), we can see that taking $\breve{\boldsymbol{u}} = \boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}^1$ and $\boldsymbol{F}_s = \boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{W}^2$, where $\boldsymbol{W}^1$ and $\boldsymbol{W}^2$are uncorrelated i.i.d Gaussian processes, gives

$$\boldsymbol{U}_s = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{W}^2 + \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}^1$$

so that

$$\begin{aligned}\langle \boldsymbol{U}_s\boldsymbol{U}_s\rangle &= \left\langle \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{\Delta}_{MB}^{1/2}\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{W}^2\boldsymbol{W}^2\right\rangle + \left\langle \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{W}^1\boldsymbol{W}^1\right\rangle + \left\langle \cdots \boldsymbol{W}^2\boldsymbol{W}^1\right\rangle + \left\langle \cdots \boldsymbol{W}^1\boldsymbol{W}^2\right\rangle \\ &= \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\Delta}_{MB}\overline{\boldsymbol{\mathcal{N}}} + \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{N}}^{-1}\overline{\boldsymbol{\mathcal{N}}} = \overline{\boldsymbol{\mathcal{N}}}\left(\boldsymbol{\Delta}_{MB} + \boldsymbol{\mathcal{N}}^{-1}\right)\overline{\boldsymbol{\mathcal{N}}} = \overline{\boldsymbol{\mathcal{N}}}.\end{aligned}$$

Using equation (I.5) we can state this as the solution to the linear system

$$\begin{pmatrix} \boldsymbol{\mathcal{M}} & -\boldsymbol{\mathcal{K}} \\ \boldsymbol{\mathcal{K}}^T & \boldsymbol{\Delta}_{MB} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_s \\ \boldsymbol{U}_s \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}^1 \\ \boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{W}^2 \end{pmatrix}, \tag{II.1}$$

so that $\langle \boldsymbol{U}_s\boldsymbol{U}_s\rangle = \overline{\boldsymbol{\mathcal{N}}}$. The main issue here (as with the preconditioning discussed in section I B) is how we will generate the quantity $\boldsymbol{F}_s = \boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{W}^2$. While we will likely still be able to use the Lanczos method, a good idea for a preconditioner for this eludes me.

III. A SPLIT EM SCHEME

A modified version of thee EM-T scheme from [1]is presented here to account for the lubrication correction.

Algorithm 1 Euler-Maruyama Lubrication (EM-L) Scheme

1. Compute relevant quantities for capturing drift:

   (a) Form $\boldsymbol{W}^{FT} = \left[\boldsymbol{W}_p^{FT}\right]$, where

   $$\boldsymbol{W}_p^{FT} = k_B T \begin{bmatrix} L_p^{-1}\boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix}$$

   and $\boldsymbol{W}_p^f$, $\boldsymbol{W}_p^\tau$ are standard Gaussian random vectors.

   (b) Solve RFD mobility problem:

   $$\begin{bmatrix} \boldsymbol{\mathcal{M}}^n & -\boldsymbol{\mathcal{K}}^n \\ \left(\boldsymbol{\mathcal{K}}^T\right)^n & \boldsymbol{\Delta}_{MB}^n \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{\mathrm{RFD}} \\ \boldsymbol{U}^{\mathrm{RFD}} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{W}^{FT} \end{bmatrix}.$$

   i.
   $$\boldsymbol{U}^{\mathrm{RFD}} = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{W}^{FT}$$
   $$\boldsymbol{\lambda}^{\mathrm{RFD}} = \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{W}^{FT}$$

   (c) Randomly displace particles to:

   $$\boldsymbol{Q}_p^{\pm} = \boldsymbol{Q}_p^n + \frac{\delta}{2}\begin{bmatrix} L_p\boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix}.$$

   (d) Compute the force-drift, $\boldsymbol{D}^F$, and the slip-drift, $\boldsymbol{D}^S$:

   $$\boldsymbol{D}^F = -\frac{1}{\delta}\left\{\boldsymbol{\mathcal{K}}^T\left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{K}}^T\left(\boldsymbol{Q}^-\right)\right\}\boldsymbol{\lambda}^{\mathrm{RFD}} - \frac{1}{\delta}\left\{\boldsymbol{\Delta}_{MB}\left(\boldsymbol{Q}^+\right) - \boldsymbol{\Delta}_{MB}\left(\boldsymbol{Q}^-\right)\right\}\boldsymbol{U}^{\mathrm{RFD}}$$
   $$\boldsymbol{D}^S = \frac{1}{\delta}\left\{\boldsymbol{\mathcal{M}}\left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{M}}\left(\boldsymbol{Q}^-\right)\right\}\boldsymbol{\lambda}^{\mathrm{RFD}} - \frac{1}{\delta}\left\{\boldsymbol{\mathcal{K}}\left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{K}}\left(\boldsymbol{Q}^-\right)\right\}\boldsymbol{U}^{\mathrm{RFD}}.$$

   Note that different δ may be used for the RFDs on $\boldsymbol{\mathcal{K}}$,$\boldsymbol{\mathcal{M}}$, and $\boldsymbol{\Delta}_{MB}$ depending on the relative accuracy with which their action is evaluated.

   i.
   $$\left\langle \boldsymbol{D}^F \right\rangle = -k_B T\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{K}}^T\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}} - k_B T\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\Delta}_{MB}\right\} : \overline{\boldsymbol{\mathcal{N}}} + \boldsymbol{\mathcal{O}}\left(\delta^2\right)$$
   $$\left\langle \boldsymbol{D}^S \right\rangle = k_B T\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{M}}\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}} - k_B T\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{K}}\right\} : \overline{\boldsymbol{\mathcal{N}}} + \boldsymbol{\mathcal{O}}\left(\delta^2\right)$$

2. Compute $\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n$ using a preconditioned Lancoz method or PSE and FIND a nice way to compute $\boldsymbol{\Delta}_{MB}^{1/2}\boldsymbol{W}^2$

3. Evaluate forces and torques at $\boldsymbol{F}^n = \boldsymbol{F}\left(\boldsymbol{Q}^n, t\right)$ and solve the mobility problem:

   $$\begin{bmatrix} \boldsymbol{\mathcal{M}}^n & -\boldsymbol{\mathcal{K}}^n \\ \left(\boldsymbol{\mathcal{K}}^T\right)^n & \boldsymbol{\Delta}_{MB}^n \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \boldsymbol{U}^n \end{bmatrix} = \begin{bmatrix} -\boldsymbol{D}^S - \sqrt{2k_BT/\Delta t}\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^1 \\ \boldsymbol{F}^n + \boldsymbol{D}^F + \sqrt{2k_BT/\Delta t}\left(\boldsymbol{\Delta}_{MB}^n\right)^{1/2}\boldsymbol{W}^2 \end{bmatrix}.$$

   (a)

   $$\begin{aligned} \boldsymbol{U}^n &= \overline{\boldsymbol{\mathcal{N}}}^n\boldsymbol{F}^n + \overline{\boldsymbol{\mathcal{N}}}^n\boldsymbol{D}^F + \left(\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\right)^n\boldsymbol{D}^S + \sqrt{2k_BT/\Delta t}\left[\left(\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\right)^n\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n\boldsymbol{W}^1 + \overline{\boldsymbol{\mathcal{N}}}^n\left(\boldsymbol{\Delta}_{MB}^n\right)^{1/2}\boldsymbol{W}^2\right] \\ &\sim \overline{\boldsymbol{\mathcal{N}}}^n\boldsymbol{F}^n + \overline{\boldsymbol{\mathcal{N}}}^n\boldsymbol{D}^F + \left(\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\right)^n\boldsymbol{D}^S + \sqrt{2k_BT/\Delta t}\overline{\boldsymbol{\mathcal{N}}}^{1/2}\boldsymbol{W} \end{aligned}$$

   (b)

   $$\begin{aligned} \left\langle \boldsymbol{U}^n \right\rangle &= \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} + \left\langle \overline{\boldsymbol{\mathcal{N}}} \cdot \boldsymbol{D}^F \right\rangle + \left\langle \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1} \cdot \boldsymbol{D}^S \right\rangle \\ &\approx \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} - k_BT\overline{\boldsymbol{\mathcal{N}}} \cdot \left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{K}}^T\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}} - k_BT\overline{\boldsymbol{\mathcal{N}}} \cdot \left\{\partial_{\boldsymbol{Q}}\boldsymbol{\Delta}_{MB}\right\} : \overline{\boldsymbol{\mathcal{N}}} \\ &+ k_BT\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1} \cdot \left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{M}}\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\overline{\boldsymbol{\mathcal{N}}} - k_BT\overline{\boldsymbol{\mathcal{N}}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1} \cdot \left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{K}}\right\} : \overline{\boldsymbol{\mathcal{N}}} \\ &= \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} - k_BT\overline{\boldsymbol{\mathcal{N}}} \cdot \left\{\partial_{\boldsymbol{Q}}\overline{\boldsymbol{\mathcal{N}}}^{-1} + \partial_{\boldsymbol{Q}}\boldsymbol{\Delta}_{MB}\right\} : \overline{\boldsymbol{\mathcal{N}}} = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} - k_BT\overline{\boldsymbol{\mathcal{N}}} \cdot \left\{\partial_{\boldsymbol{Q}}\overline{\boldsymbol{\mathcal{N}}}^{-1}\right\} : \overline{\boldsymbol{\mathcal{N}}} = \overline{\boldsymbol{\mathcal{N}}}\boldsymbol{F} - k_BT\left\{\partial_{\boldsymbol{Q}} \cdot \overline{\boldsymbol{\mathcal{N}}}\right\} \end{aligned}$$

4. Update configurations to time $t + \Delta t$:

   $$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n + \Delta t\boldsymbol{U}^n.$$

IV. BIG QUESTIONS

1. How do we efficiently apply $\mathbf{\Delta}_{MB}$?
2. How do we efficiently invert (or approximately invert) $\mathbf{\Delta}_{MB}$?
3. How do we efficiently root $\mathbf{\Delta}_{MB}$?

[1] Brennan Sprinkle, Florencio Balboa Usabiaga, Neelesh A. Patankar, and Aleksandar Donev. Large scale brownian dynamics of confined suspensions of rigid particles. *The Journal of Chemical Physics*, 147(24):244103, 2017. Software available at `https://github.com/stochasticHydroTools/RigidMultiblobsWall`.

[2] M. Benzi and G. Golub. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 2004.