# A fluctuating boundary integral method for Brownian suspensions

**Yuanxun (Bill) Bao**
Courant Institute, NYU

Collaborators:
Aleksandar Donev (Courant)
Leslie Greengard (Courant)
Eric Keaveny (Imperial)
Manas Rachh (Yale)

SIAM Computational Science and Engineering
March 2, 2017

# Brownian Dynamics with Hydrodynamic Interactions

- Consider a suspension of $N_b$ rigid bodies with configuration $\mathbf{Q} = \{\mathbf{q}_\beta, \boldsymbol{\theta}_\beta\}_{\beta=1}^{N_b}$ consisting of **positions** and **orientations** immersed in a Stokes fluid.

# Brownian Dynamics with Hydrodynamic Interactions

- Consider a suspension of $N_b$ rigid bodies with configuration $\mathbf{Q} = \{\mathbf{q}_\beta, \boldsymbol{\theta}_\beta\}_{\beta=1}^{N_b}$ consisting of **positions** and **orientations** immersed in a Stokes fluid.

- The Ito stochastic equation of **Brownian Dynamics** (BD) is

$$\frac{d\mathbf{Q}}{dt} = -\mathcal{N}\partial_{\mathbf{Q}} U + (2k_B T\mathcal{N})^{\frac{1}{2}}\mathcal{W}(t) + (k_B T)\,\partial_{\mathbf{Q}}\cdot\mathcal{N},$$

where $\mathcal{N}(\mathbf{Q})$ is the **body mobility matrix**, $U(\mathbf{Q})$ is the potential energy, $k_B T$ is the temperature, and $\mathcal{W}(t)$ is a vector of independent white noise processes.

# Brownian Dynamics with Hydrodynamic Interactions

- Consider a suspension of $N_b$ rigid bodies with configuration $\mathbf{Q} = \{\mathbf{q}_\beta, \boldsymbol{\theta}_\beta\}_{\beta=1}^{N_b}$ consisting of **positions** and **orientations** immersed in a Stokes fluid.

- The Ito stochastic equation of **Brownian Dynamics** (BD) is

$$\frac{d\mathbf{Q}}{dt} = -\mathcal{N}\partial_{\mathbf{Q}} U + \left(2k_B T\mathcal{N}\right)^{\frac{1}{2}}\mathcal{W}(t) + \left(k_B T\right)\,\partial_{\mathbf{Q}}\cdot\mathcal{N},$$

where $\mathcal{N}(\mathbf{Q})$ is the **body mobility matrix**, $U(\mathbf{Q})$ is the potential energy, $k_B T$ is the temperature, and $\mathcal{W}(t)$ is a vector of independent white noise processes.

- Here the stochastic noise amplitude is determined from the **fluctuation-dissipation balance**: $\mathcal{N}^{\frac{1}{2}}\left(\mathcal{N}^{\frac{1}{2}}\right)^{*} = \mathcal{N}$.

# Brownian Dynamics with Hydrodynamic Interactions

- Consider a suspension of $N_b$ rigid bodies with configuration $\mathbf{Q} = \{\mathbf{q}_\beta, \boldsymbol{\theta}_\beta\}_{\beta=1}^{N_b}$ consisting of **positions** and **orientations** immersed in a Stokes fluid.

- The Ito stochastic equation of **Brownian Dynamics** (BD) is

$$\frac{d\mathbf{Q}}{dt} = -\mathcal{N}\partial_{\mathbf{Q}} U + (2k_B T\mathcal{N})^{\frac{1}{2}}\mathcal{W}(t) + (k_B T)\,\partial_{\mathbf{Q}} \cdot \mathcal{N},$$

  where $\mathcal{N}(\mathbf{Q})$ is the **body mobility matrix**, $U(\mathbf{Q})$ is the potential energy, $k_B T$ is the temperature, and $\mathcal{W}(t)$ is a vector of independent white noise processes.

- Here the stochastic noise amplitude is determined from the **fluctuation-dissipation balance**: $\mathcal{N}^{\frac{1}{2}}\left(\mathcal{N}^{\frac{1}{2}}\right)^* = \mathcal{N}$.

- The **stochastic drift** term $\partial_{\mathbf{Q}} \cdot \mathcal{N} = \sum_j \partial_j \mathcal{N}_{ij}$ is related to the Ito interpretation of the noise.

# Hydrodynamic Body Mobility Matrix

- The **body mobility matrix** $\mathcal{N}(\mathbf{Q}) \succeq \mathbf{0}$ is a symmetric positive semidefinite (SPD) and it includes **hydrodynamic interactions** and (periodic) **boundary conditions**.

# Hydrodynamic Body Mobility Matrix

- The **body mobility matrix** $\mathcal{N}(\mathbf{Q}) \succeq \mathbf{0}$ is a symmetric positive semidefinite (SPD) and it includes **hydrodynamic interactions** and (periodic) **boundary conditions**.
- For viscous-dominated flows ($Re \to 0$), we can assume **steady Stokes flow** and solve the **Stokes mobility problem**,

$$\mathbf{U} = \mathcal{N}\mathbf{F},$$

where $\mathbf{U} = \{\mathbf{u}_\beta, \boldsymbol{\omega}_\beta\}_{\beta=1}^{N_b}$ collects the linear and angular velocities, $\mathbf{F} = \{\mathbf{f}_\beta, \boldsymbol{\tau}_\beta\}_{\beta=1}^{N_b}$ collects the applied forces and torques.

# Hydrodynamic Body Mobility Matrix

- The **body mobility matrix** $\mathcal{N}(\mathbf{Q}) \succeq \mathbf{0}$ is a symmetric positive semidefinite (SPD) and it includes **hydrodynamic interactions** and (periodic) **boundary conditions**.
- For viscous-dominated flows ($Re \to 0$), we can assume **steady Stokes flow** and solve the **Stokes mobility problem**,

$$\mathbf{U} = \mathcal{N}\mathbf{F},$$

where $\mathbf{U} = \{\mathbf{u}_\beta, \boldsymbol{\omega}_\beta\}_{\beta=1}^{N_b}$ collects the linear and angular velocities, $\mathbf{F} = \{\mathbf{f}_\beta, \boldsymbol{\tau}_\beta\}_{\beta=1}^{N_b}$ collects the applied forces and torques.
- At every time step of BD simulation, we need to generate particle velocity in the form of (dropping $k_B T$),

$$\widetilde{\mathbf{U}} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}}\mathbf{W}.$$

# Hydrodynamic Body Mobility Matrix

- The **body mobility matrix** $\mathcal{N}(\mathbf{Q}) \succeq \mathbf{0}$ is a symmetric positive semidefinite (SPD) and it includes **hydrodynamic interactions** and (periodic) **boundary conditions**.
- For viscous-dominated flows ($Re \to 0$), we can assume **steady Stokes flow** and solve the **Stokes mobility problem**,

$$\mathbf{U} = \mathcal{N}\mathbf{F},$$

where $\mathbf{U} = \{\mathbf{u}_\beta, \boldsymbol{\omega}_\beta\}_{\beta=1}^{N_b}$ collects the linear and angular velocities, $\mathbf{F} = \{\mathbf{f}_\beta, \boldsymbol{\tau}_\beta\}_{\beta=1}^{N_b}$ collects the applied forces and torques.

- At every time step of BD simulation, we need to generate particle velocity in the form of (dropping $k_B T$),

$$\widetilde{\mathbf{U}} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}}\mathbf{W}.$$

- This talk: How to accurately and efficiently compute **the action of** $\mathcal{N}$ **and** $\mathcal{N}^{\frac{1}{2}}$?

# First-Kind Boundary Integral Formulation

- Let us first ignore Brownian terms and solve a **mobility problem** to compute $\mathcal{N}\mathbf{F}$.

# First-Kind Boundary Integral Formulation

- Let us first ignore Brownian terms and solve a **mobility problem** to compute $\mathcal{N}\mathbf{F}$.
- For simplicity, consider only a single body $\Omega$. The **first-kind boundary integral equation** for the mobility problem,

# First-Kind Boundary Integral Formulation

- Let us first ignore Brownian terms and solve a **mobility problem** to compute $\mathcal{N}\mathbf{F}$.
- For simplicity, consider only a single body $\Omega$. The **first-kind boundary integral equation** for the mobility problem,

$$\mathbf{v}(\mathbf{q}) = \mathbf{u} + \boldsymbol{\omega} \times \mathbf{q} = \int_{\partial\Omega} \mathbb{G}(\mathbf{q} - \mathbf{q}')\, \boldsymbol{\mu}(\mathbf{q}')\, d\mathbf{q}' \quad \text{for all } \mathbf{q} \in \partial\Omega, \quad (1)$$

along with force and torque balance conditions

$$\int_{\partial\Omega} \boldsymbol{\mu}(\mathbf{q})\, d\mathbf{q} = \mathbf{f} \quad \text{and} \quad \int_{\partial\Omega} \mathbf{q} \times \boldsymbol{\mu}(\mathbf{q})\, d\mathbf{q} = \boldsymbol{\tau}, \qquad (2)$$

where $\boldsymbol{\mu}(\mathbf{q} \in \partial\Omega)$ is the surface traction (single-layer density) and $\mathbb{G}$ is the (periodic) Stokeslet.

# First-Kind Boundary Integral Formulation

- Let us first ignore Brownian terms and solve a **mobility problem** to compute $\mathcal{N}\mathbf{F}$.
- For simplicity, consider only a single body $\Omega$. The **first-kind boundary integral equation** for the mobility problem,

$$\mathbf{v}(\mathbf{q}) = \mathbf{u} + \boldsymbol{\omega} \times \mathbf{q} = \int_{\partial\Omega} \mathbb{G}(\mathbf{q} - \mathbf{q}')\, \boldsymbol{\mu}(\mathbf{q}')\, d\mathbf{q}' \quad \text{for all } \mathbf{q} \in \partial\Omega, \tag{1}$$

  along with force and torque balance conditions

$$\int_{\partial\Omega} \boldsymbol{\mu}(\mathbf{q})\, d\mathbf{q} = \mathbf{f} \quad \text{and} \quad \int_{\partial\Omega} \mathbf{q} \times \boldsymbol{\mu}(\mathbf{q})\, d\mathbf{q} = \boldsymbol{\tau}, \tag{2}$$

  where $\boldsymbol{\mu}(\mathbf{q} \in \partial\Omega)$ is the surface traction (single-layer density) and $\mathbb{G}$ is the (periodic) Stokeslet.
- Note that one can alternatively use a **completed second-kind** or a mixed first-second kind formulation for improved conditioning. We only know how to generate Brownian displacements efficiently in the first-kind formulation.

# First-Kind Boundary Integral Formulation

- Assume that the surface of the body is discretized in some manner, and the **single-layer operator** $\boldsymbol{\mathcal{M}}$ is approximated by some quadrature,

$$\int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \boldsymbol{\mu}(\mathbf{q}')\ d\mathbf{q}' \equiv \boldsymbol{\mathcal{M}}\boldsymbol{\mu} \to \mathbf{M}\boldsymbol{\lambda},$$

where $\boldsymbol{\mathcal{M}}$ is a SPD operator with kernel $\mathbb{G}$ with $r^{-1}$ singularity in 3D ($\log r$ in 2D), discretized as a SPD mobility matrix $\mathbf{M}$.

# First-Kind Boundary Integral Formulation

- Assume that the surface of the body is discretized in some manner, and the **single-layer operator** $\boldsymbol{\mathcal{M}}$ is approximated by some quadrature,

$$\int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \mu(\mathbf{q}')\ d\mathbf{q}' \equiv \boldsymbol{\mathcal{M}}\mu \rightarrow \mathbf{M}\boldsymbol{\lambda},$$

  where $\boldsymbol{\mathcal{M}}$ is a SPD operator with kernel $\mathbb{G}$ with $r^{-1}$ singularity in 3D ($\log r$ in 2D), discretized as a SPD mobility matrix $\mathbf{M}$.
- In matrix notation the **mobility problem** can be written as a **saddle-point** linear system for the surface forces $\boldsymbol{\lambda}$ and rigid-body motion $\mathbf{U}$,

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^{\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \end{bmatrix} = -\begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix}, \tag{3}$$

  where $\mathbf{K}$ is a simple geometric matrix.

# First-Kind Boundary Integral Formulation

- Assume that the surface of the body is discretized in some manner, and the **single-layer operator** $\boldsymbol{\mathcal{M}}$ is approximated by some quadrature,

$$\int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \mu(\mathbf{q}')\ d\mathbf{q}' \equiv \boldsymbol{\mathcal{M}}\mu \to \mathbf{M}\boldsymbol{\lambda},$$

  where $\boldsymbol{\mathcal{M}}$ is a SPD operator with kernel $\mathbb{G}$ with $r^{-1}$ singularity in 3D ($\log r$ in 2D), discretized as a SPD mobility matrix $\mathbf{M}$.
- In matrix notation the **mobility problem** can be written as a **saddle-point** linear system for the surface forces $\boldsymbol{\lambda}$ and rigid-body motion $\mathbf{U}$,

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^{\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \end{bmatrix} = -\begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix}, \tag{3}$$

  where $\mathbf{K}$ is a simple geometric matrix.
- Using Schur complement to eliminate $\boldsymbol{\lambda}$, we get

$$\mathbf{U} = \boldsymbol{\mathcal{N}}\mathbf{F} = (\mathbf{K}^{\top}\mathbf{M}^{-1}\mathbf{K})^{-1}\mathbf{F}.$$

# Brownian Displacements

- How do we compute the action of $\mathcal{N}^{\frac{1}{2}}$?
  More precisely, how to generate a Gaussian random vector with covariance $\mathcal{N}$?

# Brownian Displacements

- How do we compute the action of $\mathcal{N}^{\frac{1}{2}}$?
  More precisely, how to generate a Gaussian random vector with covariance $\mathcal{N}$?
- Assume for now we knew the action of $\mathbf{M}^{\frac{1}{2}}$ on a vector of Gaussian random variables $\mathbf{W}$, *i.e.*, that we knew how to generate a random **"slip"** velocity with covariance given by $\mathbf{M}$ (periodic Stokeslet).

# Brownian Displacements

- How do we compute the action of $\mathcal{N}^{\frac{1}{2}}$?
  More precisely, how to generate a Gaussian random vector with covariance $\mathcal{N}$?
- Assume for now we knew the action of $\mathbf{M}^{\frac{1}{2}}$ on a vector of Gaussian random variables $\mathbf{W}$, *i.e.*, that we knew how to generate a random **"slip"** velocity with covariance given by $\mathbf{M}$ (periodic Stokeslet).
- *Key idea 1*: solve the **mobility problem** with such a random slip

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \end{bmatrix} = - \begin{bmatrix} \mathbf{M}^{\frac{1}{2}}\mathbf{W} \\ \mathbf{F} \end{bmatrix}, \tag{4}$$

$$\Longrightarrow \mathbf{U} = \mathcal{N}\mathbf{F} + \mathcal{N}\mathbf{K}^\top\mathbf{M}^{-1}\mathbf{M}^{\frac{1}{2}}\mathbf{W} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}}\mathbf{W},$$

# Brownian Displacements

- How do we compute the action of $\mathcal{N}^{\frac{1}{2}}$?
  More precisely, how to generate a Gaussian random vector with covariance $\mathcal{N}$?
- Assume for now we knew the action of $\mathbf{M}^{\frac{1}{2}}$ on a vector of Gaussian random variables $\mathbf{W}$, *i.e.*, that we knew how to generate a random **"slip"** velocity with covariance given by $\mathbf{M}$ (periodic Stokeslet).
- *Key idea 1*: solve the **mobility problem** with such a random slip

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^{\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \end{bmatrix} = -\begin{bmatrix} \mathbf{M}^{\frac{1}{2}}\mathbf{W} \\ \mathbf{F} \end{bmatrix}, \tag{4}$$

$$\Longrightarrow \mathbf{U} = \mathcal{N}\mathbf{F} + \mathcal{N}\mathbf{K}^{\top}\mathbf{M}^{-1}\mathbf{M}^{\frac{1}{2}}\mathbf{W} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}}\mathbf{W},$$

which defines a $\mathcal{N}^{\frac{1}{2}}$ with the correct covariance:

$$\begin{aligned} \mathcal{N}^{\frac{1}{2}}\left(\mathcal{N}^{\frac{1}{2}}\right)^{*} &= \mathcal{N}\ \mathbf{K}^{\top}\mathbf{M}^{-1}\mathbf{M}^{\frac{1}{2}}\left(\mathbf{M}^{\frac{1}{2}}\right)^{*}\mathbf{M}^{-1}\mathbf{K}\ \mathcal{N} \\ &= \mathcal{N}(\mathbf{K}^{\top}\mathbf{M}^{-1}\mathbf{K})\mathcal{N} = \mathcal{N}(\mathcal{N})^{-1}\mathcal{N} = \mathcal{N}. \end{aligned} \tag{5}$$

# The Single-Layer Mobility Matrix

- We need accurate and efficient routines to compute the action of $\mathbf{M}$ and $\mathbf{M}^{\frac{1}{2}}$ for many bodies.

# The Single-Layer Mobility Matrix

- We need accurate and efficient routines to compute the action of $\mathbf{M}$ and $\mathbf{M}^{\frac{1}{2}}$ for many bodies.
- Recall that $\boldsymbol{\mathcal{M}}\boldsymbol{\mu} \equiv \int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \boldsymbol{\mu}(\mathbf{q}')\ d\mathbf{q}'$, where $\mathbb{G}$ is a **weakly singular** kernel that includes **periodic BC** effects.

# The Single-Layer Mobility Matrix

- We need accurate and efficient routines to compute the action of $\mathbf{M}$ and $\mathbf{M}^{\frac{1}{2}}$ for many bodies.
- Recall that $\boldsymbol{\mathcal{M}\mu} \equiv \int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \boldsymbol{\mu}(\mathbf{q}')\ d\mathbf{q}'$, where $\mathbb{G}$ is a **weakly singular** kernel that includes **periodic BC** effects.
- *Key idea 2*: Singular quadrature (Alpert in 2D) + Spectral Ewald method to split the Stokeslet into **near-field** and **far-field** pieces:

$$\mathbb{G} = \mathbb{G}_{\xi}^{(r)} + \mathbb{G}_{\xi}^{(w)}$$

.

# The Single-Layer Mobility Matrix

- We need accurate and efficient routines to compute the action of $\mathbf{M}$ and $\mathbf{M}^{\frac{1}{2}}$ for many bodies.
- Recall that $\boldsymbol{\mathcal{M}}\boldsymbol{\mu} \equiv \int_{\partial\Omega} \mathbb{G}(\mathbf{q}-\mathbf{q}')\ \boldsymbol{\mu}(\mathbf{q}')\ d\mathbf{q}'$, where $\mathbb{G}$ is a **weakly singular** kernel that includes **periodic BC** effects.
- *Key idea 2*: Singular quadrature (Alpert in 2D) + Spectral Ewald method to split the Stokeslet into **near-field** and **far-field** pieces:

$$\mathbb{G} = \mathbb{G}_\xi^{(r)} + \mathbb{G}_\xi^{(w)}$$

.

- This idea comes from the recent work "Rapid Sampling of Stochastic Displacements in Brownian Dynamics Simulations" by A. M. Fiore, F. Balboa Usabiaga, A. Donev and J. W. Swan, to appear in J. Chem. Phys., 2017 [**?**].

# Ewald Splitting of $\mathbf{M}$

- The splitting of $\mathbb{G}$ induces a corresponding splitting of $\mathbf{M}$ into **near-field** and **far-field** pieces:

$$\begin{aligned}\mathbf{M} &= \mathbf{M}^{(r)} + \mathbf{M}^{(w)} \\ &= \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)},\end{aligned}$$

# Ewald Splitting of $\mathbf{M}$

- The splitting of $\mathbb{G}$ induces a corresponding splitting of $\mathbf{M}$ into **near-field** and **far-field** pieces:

$$\begin{aligned}\mathbf{M} &= \mathbf{M}^{(r)} + \mathbf{M}^{(w)} \\ &= \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)},\end{aligned}$$

where $\mathbf{M}^{(r)}_{\text{Alpert}}$ is a **block-diagonal band-limited** matrix whose elements are (local) Alpert corrections to the trapezoidal rule and

$$\left(\mathbf{M}^{(r)}_{\text{trap}}\right)_{ij} = \mathbb{G}^{(r)}_{\xi}(\mathbf{q}_i - \mathbf{q}_j) \quad \text{and} \quad \left(\mathbf{M}^{(w)}\right)_{ij} = \mathbb{G}^{(w)}_{\xi}(\mathbf{q}_i - \mathbf{q}_j),\; i \neq j.$$

# Near-Field Piece of $\mathbf{M}$

$$\mathbf{M}\boldsymbol{\lambda} = \left(\mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)}\right)\boldsymbol{\lambda}$$

- $\mathbf{M}^{(r)}_{\text{Alpert}}$: Alpert correction matrix is **precomputed** for a single body in some reference configuration, and apply to each body via rotation.

# Near-Field Piece of $\mathbf{M}$

$$\mathbf{M}\boldsymbol{\lambda} = \left(\mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)}\right)\boldsymbol{\lambda}$$

- $\mathbf{M}^{(r)}_{\text{Alpert}}$: Alpert correction matrix is **precomputed** for a single body in some reference configuration, and apply to each body via rotation.
- $\mathbf{M}^{(r)}_{\text{trap}}$: **sparse matrix-vector multiplication** because the real-space kernel $\mathbb{G}^{(r)}_{\xi}$ decays like $e^{-\xi^2 d^2}$, where $d = |\mathbf{q}_i - \mathbf{q}_j|$.
  The action of $\mathbf{M}^{(r)}_{\text{trap}}$ can be efficiently computed by **cell linked-lists** as used in the classical MD method.

# Far-Field Piece of $\mathbf{M}$

$$\mathbf{M}\boldsymbol{\lambda} = \left(\mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)}\right)\boldsymbol{\lambda}$$

- $\mathbf{M}^{(w)}$: the wave-space kernel $\mathbb{G}^{(w)}_{\xi}$ is smooth and regular,

$$\mathbb{G}^{(w)}(\mathbf{r}) = \frac{1}{V}\sum_{\mathbf{k}\neq\mathbf{0}} e^{i\mathbf{k}\cdot(\mathbf{r})}\frac{H(k,\xi)}{k^2}(\mathbf{I} - \hat{\mathbf{k}}\hat{\mathbf{k}}), \tag{6}$$

where the Hasimoto splitting function $H(k,\xi) = \left(1 + \frac{k^2}{4\xi^2}\right)e^{-k^2/4\xi^2}$.

# Far-Field Piece of $\mathbf{M}$

$$\mathbf{M}\boldsymbol{\lambda} = \left( \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}} + \mathbf{M}^{(w)} \right) \boldsymbol{\lambda}$$

- $\mathbf{M}^{(w)}$: the wave-space kernel $\mathbb{G}^{(w)}_{\xi}$ is smooth and regular,

$$\mathbb{G}^{(w)}(\mathbf{r}) = \frac{1}{V} \sum_{\mathbf{k} \neq \mathbf{0}} e^{i\mathbf{k}\cdot(\mathbf{r})} \frac{H(k,\xi)}{k^2} (\mathbf{I} - \hat{\mathbf{k}}\hat{\mathbf{k}}), \tag{6}$$

  where the Hasimoto splitting function $H(k,\xi) = \left(1 + \frac{k^2}{4\xi^2}\right) e^{-k^2/4\xi^2}$.

- We can efficiently compute the action of $\mathbf{M}^{(w)}$ in Fourier space by using the Spectral Ewald method of Lindbo/Tornberg [?],

$$\mathbf{M}^{(w)} = \mathbf{S}^{\dagger}\mathbf{B}\mathbf{S}, \tag{7}$$

  where $\mathbf{S}$ is the non-uniform FFT (NUFFT) of Greengard/Lee [?], and $\mathbf{B}$ is a SPD block-diagonal matrix (in Fourier space),

$$\mathbf{B}(k,\xi) = \frac{H(k,\xi)}{k^2} (\mathbf{I} - \hat{\mathbf{k}}\hat{\mathbf{k}}).$$

# Far-Field Random Slip Velocity

- *Key Idea 3*: random slip velocity with covariance $\mathbf{M}$ is generated by

$$\mathbf{M}^{\frac{1}{2}}\mathbf{W} \overset{\text{d.}}{=} \left(\mathbf{M}^{(w)}\right)^{\frac{1}{2}}\mathbf{W}^{(w)} + \left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}}\mathbf{W}^{(r)}, \tag{8}$$

if both $\mathbf{M}^{(w)}$ and $\mathbf{M}^{(r)}$ are SPD and $\langle \mathbf{W}^{(w)}\mathbf{W}^{(r)}\rangle = \mathbf{0}$.
Also taken from recent work of Fiore et al. [**?**]

# Far-Field Random Slip Velocity

- *Key Idea 3*: random slip velocity with covariance $\mathbf{M}$ is generated by

$$\mathbf{M}^{\frac{1}{2}}\mathbf{W} \overset{\mathrm{d.}}{=} \left(\mathbf{M}^{(w)}\right)^{\frac{1}{2}} \mathbf{W}^{(w)} + \left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}} \mathbf{W}^{(r)}, \tag{8}$$

  if both $\mathbf{M}^{(w)}$ and $\mathbf{M}^{(r)}$ are SPD and $\langle \mathbf{W}^{(w)}\mathbf{W}^{(r)} \rangle = \mathbf{0}$.
  Also taken from recent work of Fiore et al. [**?**]
- The far-field piece $\mathbf{M}^{(w)}$ is SPD by construction and we can write

$$\mathbf{M}^{(w)} = \mathbf{S}^{\dagger}\mathbf{B}\mathbf{S} = \left(\mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\right)\left(\mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\right)^{\dagger}, \tag{9}$$

  so that the wave-space random slip velocity can be generated with a single call to the NUFFT,

$$\left(\mathbf{M}^{(w)}\right)^{\frac{1}{2}} \mathbf{W}^{(w)} = \mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\mathbf{W}^{(w)}, \tag{10}$$

# Far-Field Random Slip Velocity

- *Key Idea 3*: random slip velocity with covariance $\mathbf{M}$ is generated by

$$\mathbf{M}^{\frac{1}{2}}\mathbf{W} \overset{\text{d.}}{=} \left(\mathbf{M}^{(w)}\right)^{\frac{1}{2}}\mathbf{W}^{(w)} + \left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}}\mathbf{W}^{(r)}, \tag{8}$$

  if both $\mathbf{M}^{(w)}$ and $\mathbf{M}^{(r)}$ are SPD and $\langle \mathbf{W}^{(w)}\mathbf{W}^{(r)}\rangle = \mathbf{0}$.
  Also taken from recent work of Fiore et al. [?]
- The far-field piece $\mathbf{M}^{(w)}$ is SPD by construction and we can write

$$\mathbf{M}^{(w)} = \mathbf{S}^{\dagger}\mathbf{B}\mathbf{S} = \left(\mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\right)\left(\mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\right)^{\dagger}, \tag{9}$$

  so that the wave-space random slip velocity can be generated with a single call to the NUFFT,

$$\left(\mathbf{M}^{(w)}\right)^{\frac{1}{2}}\mathbf{W}^{(w)} = \mathbf{S}^{\dagger}\mathbf{B}^{\frac{1}{2}}\mathbf{W}^{(w)}, \tag{10}$$

- This is equivalent to how Brownian displacements are generated in methods like the Fluctuating Immersed Boundary [?] and the fluctuating Force Coupling Method [?] by using fluctuating hydrodynamics (putting stochastic forcing on fluid rather than on particles).

# Near-Field Random Slip Velocity

- For the near-field random slip velocity, we need to generate the action of the square root of a **sparse matrix**,

$$\mathbf{M}^{(r)} = \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}}$$

# Near-Field Random Slip Velocity

- For the near-field random slip velocity, we need to generate the action of the square root of a **sparse matrix**,

$$\mathbf{M}^{(r)} = \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}}$$

- For sparse matrices, the principal square root can be efficiently computed by a **Krylov Lanczos method** of Chow/Saad [**?**].

# Near-Field Random Slip Velocity

- For the near-field random slip velocity, we need to generate the action of the square root of a **sparse matrix**,

$$\mathbf{M}^{(r)} = \mathbf{M}^{(r)}_{\text{Alpert}} + \mathbf{M}^{(r)}_{\text{trap}}$$

- For sparse matrices, the principal square root can be efficiently computed by a **Krylov Lanczos method** of Chow/Saad [**?**].

- In general, $\mathbf{M}^{(r)}_{\text{Alpert}}$ is not symmetric, so $\mathbf{M}^{(r)}$ is not SPD strictly speaking. Nevertheless, we find that symmetrizing $\mathbf{M}^{(r)}_{\text{Alpert}}$ preserves the order of accuracy of Alpert quadrature, and the Krylov Lanczos iteration is rather **insensitive** to any small negative eigenvalues.

# Block-Diagonal Preconditioners

- To mitigate the inherent ill-conditioning of $\mathbf{M}$ due to the use of a first-kind boundary integral formulation, we apply a **block-diagonal preconditioner**, *i.e.*, we simply neglect all hydrodynamic interactions between distinct bodies in the preconditioner, both when solving the saddle-point mobility problem using GMRES, and in the Lanczos iteration for generating $\left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}}\mathbf{W}^{(r)}$.

# Block-Diagonal Preconditioners

- To mitigate the inherent ill-conditioning of $\mathbf{M}$ due to the use of a first-kind boundary integral formulation, we apply a **block-diagonal preconditioner**, *i.e.*, we simply neglect all hydrodynamic interactions between distinct bodies in the preconditioner, both when solving the saddle-point mobility problem using GMRES, and in the Lanczos iteration for generating $\left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}} \mathbf{W}^{(r)}$.

- Both preconditioners can be **precomputed** using LAPACK for a single body, and then applied to many bodies via two fast vector **rotations** per body.

# Block-Diagonal Preconditioners

- To mitigate the inherent ill-conditioning of $\mathbf{M}$ due to the use of a first-kind boundary integral formulation, we apply a **block-diagonal preconditioner**, *i.e.*, we simply neglect all hydrodynamic interactions between distinct bodies in the preconditioner, both when solving the saddle-point mobility problem using GMRES, and in the Lanczos iteration for generating $\left(\mathbf{M}^{(r)}\right)^{\frac{1}{2}} \mathbf{W}^{(r)}$.

- Both preconditioners can be **precomputed** using LAPACK for a single body, and then applied to many bodies via two fast vector **rotations** per body.

- GMRES and Lanczos converge in a constant number of iterations, growing only weakly with packing density.

# Numerical Results

- This proof-of-concept algorithm/implementation is in **2D only**, but the main ideas can be carried over to 3D in principle (but with some technical difficulties that need to be overcome!).

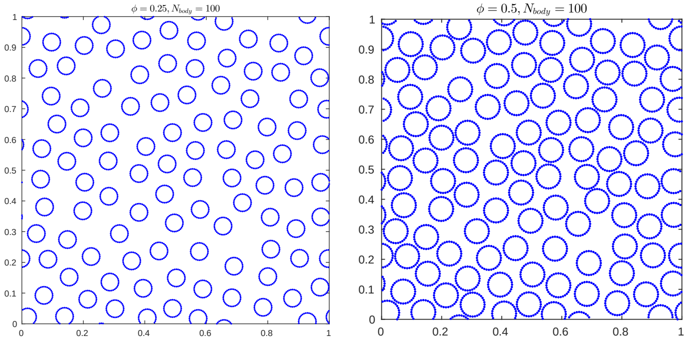


Figure: Random configurations of 100 disks with packing ratio $\phi = 0.25$ (low density) and $\phi = 0.5$ (moderately high density)

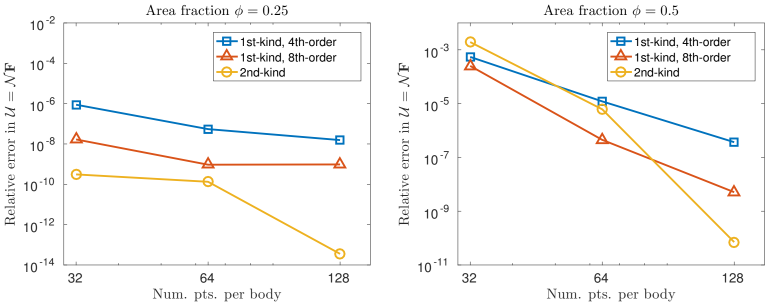


Figure: Accuracy of 1st- and 2nd-kind (spectral in 2D!) mobility solvers for dilute and dense hard-disk suspensions. While the 2nd kind gives spectral accuracy and converges faster with number of DOFs, the first kind is more accurate for low resolutions especially at higher densities (but what about 3D?).

# Convergence and robustness (2D specific!)

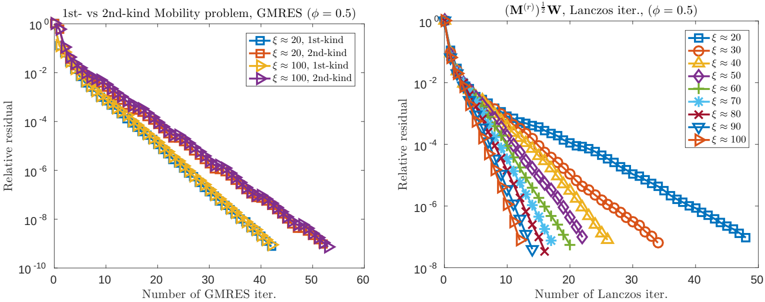


Figure: We expect much better scaling in 3D due to faster decay of Stokeslet.

# Efficiency and Scaling

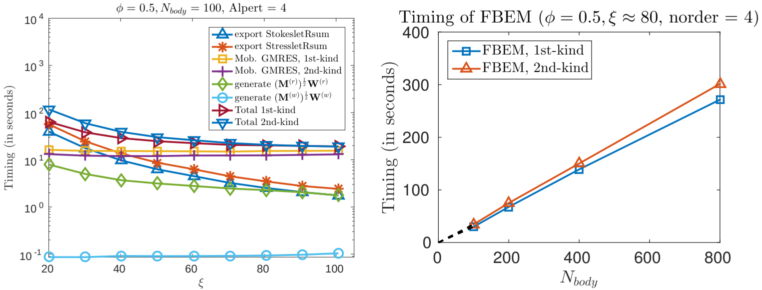


Figure: Left: Optimal Ewald splitting parameter, specific to our 2D Matlab implementation. In 3D one expects to see an optimal splitting parameter (see Fiore et al. [?]). Right: Linear scaling of the algorithm with the number of bodies.

# Conclusion

- **Ewald (Hasimoto) splitting** can be used to accelerate both deterministic and stochastic simulations in periodic domains.

- Key is to ensure both **far-field** and **near-field are (essentially) SPD** so one piece is generated using FFTs and the other using iterative methods.

- Using these principles we have constructed a **linear-scaling** fluctuating boundary element method for Brownian suspensions.

- **Can a similar idea be used with boundary integral methods based on grid-free fast multipole methods (e.g., unbounded domains)?**

# References


A. M. Fiore, F. Balboa Usabiaga, A. Donev, and J. W. Swan.
Rapid Sampling of Stochastic Displacements in Brownian Dynamics Simulations.
nov 2016.

Dag Lindbo and Anna-Karin Tornberg.
Spectrally accurate fast summation for periodic stokes potentials.
*Journal of Computational Physics*, 229(23):8994–9010, 2010.

L. Greengard and J. Lee.
Accelerating the nonuniform fast fourier transform.
*SIAM Review*, 46(3):443–454, 2004.

S. Delong, F. Balboa Usabiaga, R. Delgado-Buscalioni, B. E. Griffith, and A. Donev.
Brownian Dynamics without Green's Functions.
*J. Chem. Phys.*, 140(13):134110, 2014.
Software available at `https://github.com/stochasticHydroTools/FIB`.

Eric E. Keaveny.
Fluctuating force-coupling method for simulations of colloidal suspensions.
*J. Comp. Phys.*, 269(0):61 – 79, 2014.

Edmond Chow and Yousef Saad.
Preconditioned krylov subspace methods for sampling multivariate gaussian distributions.
*SIAM Journal on Scientific Computing*, 36(2):A588–A608, 2014.