

A fluctuating boundary integral method for Brownian suspensions



Yuanxun Bao ^{a,*}, Manas Rachh ^{b,d}, Eric E. Keaveny ^c, Leslie Greengard ^{a,d}, Aleksandar Donev ^a

^a Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA

^b Applied Mathematics Program, Yale University, New Haven, CT 06511, USA

^c Department of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom

^d Center for Computational Biology, Flatiron Institute, New York, NY, USA

ARTICLE INFO

Article history:

Received 6 September 2017

Received in revised form 14 April 2018

Accepted 13 August 2018

Available online 16 August 2018

Keywords:

Brownian dynamics

Stokes flow

Boundary integral equation

Fluctuating hydrodynamics

Fast algorithm

Colloidal suspension

ABSTRACT

We present a fluctuating boundary integral method (FBIM) for overdamped Brownian Dynamics (BD) of two-dimensional periodic suspensions of rigid particles of complex shape immersed in a Stokes fluid. We develop a novel approach for generating Brownian displacements that arise in response to the thermal fluctuations in the fluid. Our approach relies on a first-kind boundary integral formulation of a mobility problem in which a random surface velocity is prescribed on the particle surface, with zero mean and covariance proportional to the Green's function for Stokes flow (Stokeslet). This approach yields an algorithm that scales linearly in the number of particles for both deterministic and stochastic dynamics, handles particles of complex shape, achieves high order of accuracy, and can be generalized to three dimensions and other boundary conditions. We show that Brownian displacements generated by our method obey the discrete fluctuation–dissipation balance relation (DFDB). Based on a recently-developed Positively Split Ewald method Fiore et al. (2017) [24], near-field contributions to the Brownian displacements are efficiently approximated by iterative methods in real space, while far-field contributions are rapidly generated by fast Fourier-space methods based on fluctuating hydrodynamics. FBIM provides the key ingredient for time integration of the overdamped Langevin equations for Brownian suspensions of rigid particles. We demonstrate that FBIM obeys DFDB by performing equilibrium BD simulations of suspensions of starfish-shaped bodies using a random finite difference temporal integrator.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Complex fluids containing colloidal particles are ubiquitous in science and industrial applications. Colloidal particles span length scales from several nanometers, such as magnetic nano-propellers [1] and molecular motors [2,3], to a few microns, such as self-phoretic Janus particles [4] and motile microorganisms [5]. In the last decade, increasing attention has been given to the emerging field of *active* colloidal suspensions [6–10], in which particles move autonomously or in response

* Corresponding author.

E-mail addresses: billbao@cims.nyu.edu (Y. Bao), manas.rachh@yale.edu (M. Rachh), e.keaveny@imperial.ac.uk (E.E. Keaveny), greengard@cims.nyu.edu (L. Greengard), donev@courant.nyu.edu (A. Donev).

to external forces. Despite the advances in the theory and experimental design of passive and active colloids, developing accurate and efficient computational methods that are capable of simulating tens or hundreds of thousands of particles, as well as handling particles of complex shape, still remains a formidable challenge. Here we develop a novel algorithm for generating the Brownian (stochastic) displacements required to perform overdamped Brownian dynamics of a suspension of rigid particles immersed in a Stokes fluid. Our method is based on boundary integral techniques, scales linearly in the number of particles for both deterministic and stochastic dynamics, handles particles of complex shape, and achieves high order accuracy. Because of its close connection to fluctuating hydrodynamics, we refer to our method as the *Fluctuating Boundary Integral Method* (FBIM). We restrict our attention to two-dimensional periodic domains. However, our approach can be extended to three dimensions and confined suspensions.

The two key ingredients that need to be included in a computational method for colloidal suspensions are the long-ranged hydrodynamic interactions (HIs) and the correlated Brownian motion of the particles. In the absence of active and Brownian motion, describing the hydrodynamics of Stokesian suspensions requires the accurate solution of mobility problems [11,12], i.e., computing the linear and angular velocities of the particles in response to applied (external) forces and torques. This process defines the action of a *body mobility matrix*, which converts the applied forces to the resulting particle motions. The mobility matrix encodes the many-body hydrodynamic interactions among the particles, and is therefore dense with long-ranged interactions between *all* bodies. In the presence of Brownian motion, fluctuation–dissipation balance requires the Brownian displacements to have zero mean and covariance proportional to the hydrodynamic mobility matrix. This necessitates an algorithm for generating Gaussian random variables with covariance equal to the body mobility matrix. In this work, we present a new linear-scaling boundary integral method to generate, together, both the deterministic and Brownian displacements.

For passive suspensions of spherical particles in zero Reynolds number flows (infinite Schmidt number), the methods of Brownian [13–17] and Stokesian Dynamics (SD) [18–20] have dominated the chemical engineering community. These methods are tailored to sphere suspensions and utilize a multipole hierarchy truncated at either the monopole (BD) or dipole (SD) level in order to capture the far-field behavior of the hydrodynamic interactions. Modern fast algorithms can apply the action of the truncated mobility matrix with linear-scaling by using the Fast Multipole Method (FMM) for an unbounded domain [17], and using Ewald-like methods for periodic [20,21] and confined domains [15]. The Brownian (stochastic) displacements are typically generated iteratively by a Chebyshev polynomial approximation method [22], or by the Lanczos algorithm for application of the matrix square root [23]. However, since the hydrodynamic interactions among particles decay slowly like the inverse of distance in three dimensions and diverge logarithmically in two dimensions, the condition number of the mobility matrix grows as the number of particles increases (keeping the packing fraction fixed, see [24, Fig. 1]). Therefore, the overall computational scaling for generating Brownian displacements using iterative methods is only superlinear in general.

The fluctuating Lattice Boltzmann (FLB) method has been used for Brownian suspensions for some time [25,26]. This is an explicit solvent method which includes fluid inertia and thus operates at finite Schmidt number instead of in the overdamped limit we are interested in; furthermore, FLB relies on artificial fluid compressibility to avoid solving Poisson problems for the pressure. While the cost of each time step is linear in the number of particles (more precisely, the number of fluid cells) N , in three dimensions $\mathcal{O}(N^{2/3})$ time steps are required for vorticity to diffuse throughout the system volume, leading to superlinear $\mathcal{O}(N^{5/3})$ overall complexity [24].

As an alternative, methods such as the Fluctuating Immersed Boundary method (FIB) [27] and the fluctuating Force Coupling Method (FCM) [28,29] utilize fluctuating hydrodynamics to generate the Brownian increments in linear time by solving the fluctuating *steady* Stokes equations on a grid. This ensures that the computational cost of Brownian simulation is only marginally larger than the cost of deterministic simulations, in stark contrast to traditional BD approaches. The FIB/FCM approach to generating the Brownian displacements is further improved in the recently-developed Positively Split Ewald (PSE) method [24]. In PSE, the Rotne–Prager–Yamakawa (RPY) tensor [30] is used to capture the long-ranged hydrodynamic interactions, and its action is computed with spectral accuracy by extending the Spectral Ewald [21] method for the RPY tensor. The key idea in PSE is to use the Hasimoto splitting [31] to decompose the RPY tensor into near-field (short-ranged) and far-field (long-ranged) contributions, in a way that ensures that *both* contributions are independently *symmetric positive definite* (SPD). This makes it possible to apply a Lanczos algorithm [23] to generate the near-field contribution with only a small ($\mathcal{O}(1)$) number of iterations, while the far-field contribution is computed by fast Fourier-space methods based on fluctuating hydrodynamics using only a few FFTs. Later in Sec. 3.3, we will apply the same SPD splitting idea to the Green's function of steady Stokes flow to achieve linear scaling in the FBIM.

Essentially all commonly-used methods for overdamped Brownian suspension flows are limited to spherical particles (with some extensions to spheroids [32]), and generalization to include particles of complex shape is generally difficult. Further, these methods employ an *uncontrolled* truncation of a multipole expansion hierarchy and therefore become inaccurate when particles get close to one another as in dense suspensions.

For deterministic Stokes problems, the Boundary Integral Method (BIM) [12] is very well-developed [33–35] and allows one to handle complex particle shapes and achieve *controlled accuracy* even for dense suspensions [36,37]. In the boundary integral framework, the steady Stokes equations are reformulated as an integral equation of unknown densities that are defined on the boundary, using a first-kind (single-layer densities) or second-kind (double-layer densities) formulation, or a mixture of both. Suspended particles of complex geometry can be directly discretized by a surface mesh, and, by a suitable choice of surface quadrature, higher-order (or even spectral) accuracy can be achieved. The key difficulty is handling

the singularity of the Green's functions appearing in the boundary integral formulation. One possibility is to regularize the singularity, as done in the method of regularized Stokeslets [38] and the recently-developed linear-scaling rigid multiblob method [39], both of which rely on a regularized first-kind formulation. Regularization, however, comes at a drastic loss of accuracy, and to resolve near-field hydrodynamic interactions accurately one must make use of high-order accurate *singular quadratures* for the singular or near-singular kernel in the first- and second-kind integral operators, such as quadrature-by-expansion (QBX) [40,41].

Discretizing the boundary integral equation typically leads to a dense linear system, and fast algorithms for performing the dense matrix-vector product are required to achieve linear-scaling, such as the Fast Multipole Method (FMM) [33, 35], and Spectral Ewald methods [36,37]. Much of the state-of-the-art BIM work on Stokes mobility problems [35–37,42] uses (completed) *second-kind* (double-layer) formulations, and we will not review it in detail here since we will rely on a first-kind formulation. The current development of BIM for Stokesian suspensions is limited to the deterministic case only. In this work, we combine singular quadrature (namely, Alpert quadrature in two dimensions) and the idea of SPD splitting from the PSE method to develop a linear-scaling method for generating the Brownian displacements based on a *first-kind* boundary integral formulation. The key idea is that, instead of adding a stochastic stress to the fluid equations as done in fluctuating hydrodynamics, we prescribe a random surface velocity (distribution) on the particle surface that has zero mean and covariance proportional to the (singular) periodic Green's function of Stokes flow (the periodic *Stokeslet*). Reformulating the resulting stochastic boundary value problem as a first-kind boundary integral equation reveals that the random surface velocity has covariance proportional to the single-layer integral operator, which suggests that one ought to handle the singularity of the covariance using the same machinery used to handle the singularity of the Green's function in the first-kind BIM. This allows us to develop a numerical method that satisfies discrete fluctuation–dissipation balance (DFDB) to within solver and roundoff errors.

The outline of this paper is as follows. In the continuum formulation (Sec. 2), we show that computing the action of the hydrodynamic mobility matrix and its square root can be formulated equivalently as the solution of a Stokes boundary value problem, herein referred to as the *Stochastic Stokes Boundary Value Problem* (SSBVP). Reformulating the SSBVP as a first-kind boundary integral equation allows us to develop a numerical method (Sec. 3) that satisfies discrete fluctuation–dissipation balance (DFDB) to within solver and roundoff errors. We apply the FBIM to several benchmark problems in two dimensions (Sec. 4), and assess the effectiveness of the method by its accuracy and convergence, robustness, and scalability for suspensions of many rigid disks. We also couple the FBIM with stochastic temporal integrators based on the idea of random finite differences [27,43,44] to perform BD simulations with starfish-shaped particles, and confirm that DFDB is obtained for sufficiently small time step sizes by comparing the numerical equilibrium distribution to the correct Gibbs–Boltzmann distribution.

2. Continuum formulation

This section presents the continuum formulation of the equations of motion for Brownian suspensions. We consider a suspension of N rigid Brownian particles of complex shape immersed in a viscous incompressible fluid with constant density ρ , viscosity η , and temperature T in a domain \mathcal{V} with periodic boundary conditions. Each particle or body, indexed by β , is described by the position of a chosen “tracking point”, denoted by \mathbf{q}_β , and its rotation relative to a chosen reference configuration, denoted by θ_β . In this paper, we restrict the formulation and implementation to two spatial dimensions only, and $\theta_\beta \in \mathbb{R}$ is simply the angle of rotation. The generalization of the continuum formulation to three dimensional systems is conceptually straightforward but technically complicated by the fact that orientation in three dimensions cannot be represented by a vector in \mathbb{R}^3 ; instead, one can use normalized quaternions [43]. We denote the particle surface by Γ_β which encloses an interior domain denoted by D_β . The fluid domain exterior to the particles is defined by $E = \mathcal{V} \setminus \left\{ \bigcup_{\beta=1}^N \overline{D}_\beta \right\}$, where $\overline{D}_\beta = D_\beta \cup \Gamma_\beta$.

A number of prior works [45–48] have arrived at a fluctuating hydrodynamics formulation of the equations of motion when fluid and particle inertia (and perhaps compressibility [49]) is accounted for.¹ The fluid velocity $\mathbf{v}(\mathbf{r}, t)$ and the pressure π follows the time-dependent fluctuating Stokes equations for all $\mathbf{r} \in E$,

$$\rho \partial_t \mathbf{v} + \nabla \pi = \eta \nabla^2 \mathbf{v} + \sqrt{2\eta k_B T} \nabla \cdot \mathcal{Z}, \quad (2.1a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2.1b)$$

where k_B is Boltzmann's constant and $\mathcal{Z}(\mathbf{r}, t)$ is a random Gaussian tensor field whose components are white in space and time with mean zero and covariance

$$\langle \mathcal{Z}_{ij}(\mathbf{r}, t) \mathcal{Z}_{kl}(\mathbf{r}', t') \rangle = (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \delta(\mathbf{r} - \mathbf{r}') \delta(t - t'). \quad (2.2)$$

¹ Note that Hauge and Martin-Lof [48] explain that there is some ambiguity in whether the stochastic traction is taken to be zero or nonzero on the particle surface; this choice does not, however, affect the resulting equations of motion for the bodies. We consider the formulation given here to be the more physically meaningful and follow Hinch [46], see in particular Section 3 of the comprehensive work of Roux [47].

On the surface of Γ_β , we assume that *no-slip* boundary condition (BC) holds,

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{u}_\beta + \boldsymbol{\omega}_\beta \times (\mathbf{x} - \mathbf{q}_\beta), \quad \forall \mathbf{x} \in \Gamma_\beta, \tag{2.3}$$

where \mathbf{u}_β and $\boldsymbol{\omega}_\beta$ are the translational and rotational velocities of the particle. Each particle is also subject to applied force \mathbf{f}_β and torque $\boldsymbol{\tau}_\beta$, which are related to the fluid stress tensor $\boldsymbol{\sigma}$ and stochastic stress tensor $\boldsymbol{\sigma}^{(s)}$ by

$$\begin{aligned} m_\beta \frac{d\mathbf{u}_\beta}{dt} &= \mathbf{f}_\beta - \int_{\Gamma_\beta} (\boldsymbol{\lambda}_\beta + \boldsymbol{\lambda}_\beta^{(s)})(\mathbf{x}) dS_{\mathbf{x}}, \\ \mathbf{I}_\beta \cdot \frac{d\boldsymbol{\omega}_\beta}{dt} &= \boldsymbol{\tau}_\beta - \int_{\Gamma_\beta} (\mathbf{x} - \mathbf{x}_\beta) \times (\boldsymbol{\lambda}_\beta + \boldsymbol{\lambda}_\beta^{(s)})(\mathbf{x}) dS_{\mathbf{x}}, \end{aligned} \tag{2.4}$$

where m_β and \mathbf{I}_β are mass and moment of inertia tensor of body β . Here, $\boldsymbol{\lambda}_\beta(\mathbf{x}) = (\boldsymbol{\sigma} \cdot \mathbf{n}_\beta)(\mathbf{x})$ and $\boldsymbol{\lambda}_\beta^{(s)}(\mathbf{x}) = (\boldsymbol{\sigma}^{(s)} \cdot \mathbf{n}_\beta)(\mathbf{x})$ are the normal components of the stress tensors on the outside of the surface of the body,

$$\begin{aligned} \boldsymbol{\sigma} &= -\pi \mathbf{I} + \eta(\nabla \mathbf{v} + \nabla^\top \mathbf{v}), \\ \boldsymbol{\sigma}^{(s)} &= \sqrt{2\eta k_B T} \boldsymbol{\mathcal{Z}}, \end{aligned} \tag{2.5}$$

and \mathbf{n}_β is the unit outward normal vector of Γ_β pointing into the fluid domain.

In the overdamped (large Schmidt number) limit where the fluid velocity is eliminated as a fast variable through an adiabatic mode elimination procedure [50], the diffusive motion of the rigid bodies can be described by the stochastic differential equation (SDE) of *Brownian Dynamics* (BD)² [47]:

$$\frac{d\mathbf{Q}}{dt} = \mathcal{N}\mathbf{F} + \sqrt{2k_B T} \mathcal{N}^{\frac{1}{2}} \boldsymbol{\mathcal{W}} + (k_B T)(\partial_{\mathbf{Q}} \cdot \mathcal{N}), \tag{2.6}$$

where $\mathbf{Q}_\beta = \{\mathbf{q}_\beta, \boldsymbol{\theta}_\beta\}$ and $\mathbf{Q} = \{\mathbf{Q}_\beta\}_{\beta=1}^N$ is a composite vector that collects the positions and orientations of particles (in two dimensions, $\mathbf{Q}_\beta \in \mathbb{R}^3$). The first term on the right-hand-side of Eq. (2.6) is the deterministic motion of rigid bodies, where $\mathcal{N}(\mathbf{Q}) \geq \mathbf{0}$ is the symmetric positive semi-definite *body mobility* matrix that converts applied forces and torques $\mathbf{F}(\mathbf{Q}) = \{\mathbf{f}_\beta(\mathbf{Q}), \boldsymbol{\tau}_\beta(\mathbf{Q})\}_{\beta=1}^N$ to rigid body motion, and can be obtained by solving the standard mobility problem for rigid body motion,

$$\left\{ \begin{aligned} -\nabla \cdot \boldsymbol{\sigma} &= \nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{0}, \\ \nabla \cdot \mathbf{v} &= 0, \\ \mathbf{v}(\mathbf{x}) &= \mathbf{u}_\beta + \boldsymbol{\omega}_\beta \times (\mathbf{x} - \mathbf{q}_\beta), \quad \forall \mathbf{x} \in \Gamma_\beta, \\ \int_{\Gamma_\beta} \boldsymbol{\lambda}_\beta(\mathbf{x}) dS_{\mathbf{x}} &= \mathbf{f}_\beta \quad \text{and} \quad \int_{\Gamma_\beta} (\mathbf{x} - \mathbf{x}_c) \times \boldsymbol{\lambda}_\beta(\mathbf{x}) dS_{\mathbf{x}} = \boldsymbol{\tau}_\beta. \end{aligned} \right. \tag{2.7}$$

The random Brownian motion of the particles involves computing the “square root” of the body mobility matrix, denoted by $\mathcal{N}^{\frac{1}{2}}(\mathbf{Q})$ acting on a vector of independent white noise processes $\boldsymbol{\mathcal{W}}(t)$. In order for fluctuation–dissipation balance to hold, the matrix $\mathcal{N}^{\frac{1}{2}}$ must satisfy $\mathcal{N}^{\frac{1}{2}} (\mathcal{N}^{\frac{1}{2}})^\top = \mathcal{N}$, and does not necessarily have to be square. The last term on the right-hand-side of Eq. (2.6) is the stochastic drift term due to the Ito interpretation of the SDEs, where the divergence operator $(\partial_{\mathbf{x}} \cdot)$ for a matrix-valued function $\mathbf{A}(\mathbf{x})$ is defined by $(\partial_{\mathbf{x}} \cdot \mathbf{A})_i = \sum_j \partial A_{ij} / \partial x_j$.

Developing numerical schemes to integrate Eq. (2.6) has two main challenges. The first challenge is that, at every time step, one needs to generate the random velocity

$$\mathbf{U} = \{\mathbf{u}_\beta, \boldsymbol{\omega}_\beta\}_{\beta=1}^N = \bar{\mathbf{U}} + \tilde{\mathbf{U}} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}} \boldsymbol{\mathcal{W}}, \tag{2.8}$$

where $\bar{\mathbf{U}}$ is the deterministic particle velocity due to applied forces and torques, $\tilde{\mathbf{U}}$ is the random velocity due to the stochastic stress tensor, and $\boldsymbol{\mathcal{W}}$ is a vector of independent and identically distributed (i.i.d.) Gaussian random variables with mean zero and covariance

$$\langle W_i W_j \rangle = \frac{2k_B T}{\Delta t} \delta_{ij}, \tag{2.9}$$

² We use the differential notation of SDEs common in the physics literature.

where Δt is the time step size. More precisely, our task is to efficiently and accurately apply the action of \mathcal{N} and $\mathcal{N}^{\frac{1}{2}}$. The second challenge is to compute or approximate the stochastic drift term $(k_B T)(\partial_{\mathbf{Q}} \cdot \mathcal{N})$, which is conventionally handled by developing specialized stochastic temporal integrators [27,29,51–53]. The main focus of this paper is to tackle the first challenge by developing schemes that generate the action of \mathcal{N} and $\mathcal{N}^{\frac{1}{2}}$ based on a boundary integral formulation.

2.1. Boundary value problem formulation

For simplicity, we now consider only a single particle Γ described by $\{\mathbf{q}, \boldsymbol{\theta}\}$ immersed in a periodic domain $\mathcal{V} = [0, L]^2$, and we drop the subscript β . The generalization to account for many-body interaction is straightforward. In this section, we show that the random velocity given by Eq. (2.8) can be obtained by solving the *Stochastic Stokes Boundary Value Problem* (SSBVP):

$$\left\{ \begin{array}{l} -\nabla \cdot \boldsymbol{\sigma} = \nabla \pi - \eta \nabla^2 \mathbf{v} = 0, \quad \mathbf{r} \in \mathcal{V} \setminus \bar{D}, \\ \nabla \cdot \mathbf{v} = 0, \\ \mathbf{v}(\mathbf{x}) = \mathbf{u} + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{q}) - \check{\mathbf{v}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \\ \int_{\Gamma} \boldsymbol{\lambda}(\mathbf{x}) dS_{\mathbf{x}} = \mathbf{f} \quad \text{and} \quad \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \boldsymbol{\lambda}(\mathbf{x}) dS_{\mathbf{x}} = \boldsymbol{\tau}, \end{array} \right. \tag{2.10}$$

where $\check{\mathbf{v}}(\mathbf{x})$ is a random surface velocity prescribed on the particle, that has zero mean and covariance

$$\langle \check{\mathbf{v}}(\mathbf{x}) \check{\mathbf{v}}(\mathbf{y}) \rangle = \frac{2k_B T}{\Delta t} \mathbb{G}(\mathbf{x} - \mathbf{y}), \quad \text{for all } (\mathbf{x} \neq \mathbf{y}) \in \Gamma. \tag{2.11}$$

Here $\mathbb{G}(\mathbf{r})$ is the Green’s function for steady Stokes flow with viscosity η , and includes the specified boundary conditions (periodic BCs in our case). For $\mathbf{x} = \mathbf{y}$, Eq. (2.11) is not well-defined since \mathbb{G} is singular, which implies that $\check{\mathbf{v}}$ is a distribution and not a function; a more precise definition is given later in Eq. (2.28).

By linearity of Stokes flow, the solution of Eq. (2.10) is the superposition of

$$\mathbf{v} = \bar{\mathbf{v}} + \check{\mathbf{v}}, \quad \boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} + \check{\boldsymbol{\sigma}}, \quad \mathbf{U} = \bar{\mathbf{U}} + \check{\mathbf{U}}, \tag{2.12}$$

where $\bar{\mathbf{U}} = \{\bar{\mathbf{u}}, \bar{\boldsymbol{\omega}}\}$ and $\check{\mathbf{U}} = \{\check{\mathbf{u}}, \check{\boldsymbol{\omega}}\}$ with $\{\bar{\mathbf{v}}, \bar{\boldsymbol{\sigma}}, \bar{\mathbf{u}}, \bar{\boldsymbol{\omega}}\}$ satisfying a Stokes BVP *without* random surface velocity,

$$\left\{ \begin{array}{l} -\nabla \cdot \bar{\boldsymbol{\sigma}} = \nabla \bar{\pi} - \eta \nabla^2 \bar{\mathbf{v}} = \mathbf{0}, \\ \nabla \cdot \bar{\mathbf{v}} = 0, \\ \bar{\mathbf{v}}(\mathbf{x}) = \bar{\mathbf{u}} + \bar{\boldsymbol{\omega}} \times (\mathbf{x} - \mathbf{q}), \quad \mathbf{x} \in \Gamma \\ \int_{\Gamma} \bar{\boldsymbol{\lambda}}(\mathbf{x}) dS_{\mathbf{x}} = \mathbf{f} \quad \text{and} \quad \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \bar{\boldsymbol{\lambda}}(\mathbf{x}) dS_{\mathbf{x}} = \boldsymbol{\tau}, \end{array} \right. \tag{2.13}$$

and $\{\check{\mathbf{v}}, \check{\boldsymbol{\sigma}}, \check{\mathbf{u}}, \check{\boldsymbol{\omega}}\}$ satisfying a force- and torque-free Stokes BVP *with* a random surface velocity $\check{\mathbf{v}}$ with zero mean and covariance (2.11),

$$\left\{ \begin{array}{l} -\nabla \cdot \check{\boldsymbol{\sigma}} = \nabla \check{\pi} - \eta \nabla^2 \check{\mathbf{v}} = \mathbf{0}, \\ \nabla \cdot \check{\mathbf{v}} = 0, \\ \check{\mathbf{v}}(\mathbf{x}) = \check{\mathbf{u}} + \check{\boldsymbol{\omega}} \times (\mathbf{x} - \mathbf{q}) - \check{\mathbf{v}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \\ \int_{\Gamma} \check{\boldsymbol{\lambda}}(\mathbf{x}) dS_{\mathbf{x}} = \mathbf{0} \quad \text{and} \quad \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \check{\boldsymbol{\lambda}}(\mathbf{x}) dS_{\mathbf{x}} = \mathbf{0}. \end{array} \right. \tag{2.14}$$

The BVP given by Eq. (2.13) is the standard mobility problem for rigid body motion that solves for the deterministic part of the particle velocity $\bar{\mathbf{U}} = \{\bar{\mathbf{u}}, \bar{\boldsymbol{\omega}}\} = \mathcal{N}\mathbf{F}$. The BVP given by Eq. (2.14) generates its stochastic part $\check{\mathbf{U}} = \{\check{\mathbf{u}}, \check{\boldsymbol{\omega}}\}$.

To show that the random particle velocity $\check{\mathbf{U}}$ determined by the mobility problem (2.14) indeed obeys the fluctuation–dissipation balance,

$$\langle \check{\mathbf{U}} \check{\mathbf{U}}^{\top} \rangle = \frac{2k_B T}{\Delta t} \mathcal{N}, \tag{2.15}$$

we will invoke the *Lorentz Reciprocal Theorem* (LRT) [12, see Eq. (1.4.5)],

$$\int_{\Gamma} \mathbf{u} \cdot \boldsymbol{\lambda}' dS = \int_{\Gamma} \mathbf{u}' \cdot \boldsymbol{\lambda} dS, \tag{2.16}$$

where $\{\mathbf{u}, \boldsymbol{\lambda}\}$ and $\{\mathbf{u}', \boldsymbol{\lambda}'\}$ are two arbitrary velocity-traction pairs corresponding to solutions of the homogeneous Stokes equations. To apply the LRT, we substitute $\{\mathbf{u}, \boldsymbol{\lambda}\}$ by $\{\tilde{\mathbf{v}}, \tilde{\boldsymbol{\lambda}}\}$ from Eq. (2.14), and $\{\mathbf{u}', \boldsymbol{\lambda}'\}$ by $\{\mathbf{v}^{(i)}, \boldsymbol{\lambda}^{(i)}\}$, where $\mathbf{v}^{(i)}$ and $\boldsymbol{\lambda}^{(i)}$ are the velocity and traction of the standard mobility problem (2.13) with applied force and torque $\mathbf{F} = \{\mathbf{f}, \boldsymbol{\tau}\} = \mathbf{e}^{(i)}$, where $\mathbf{e}^{(i)} \in \mathbb{R}^3$ is the i th column of the identity matrix. By using the associated BCs in Eq. (2.13) and Eq. (2.14) to express $\tilde{\mathbf{v}}$ and $\tilde{\boldsymbol{\lambda}}$ on Γ , and then making use of the force and torque balance conditions for $\tilde{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}^{(i)}$, we can rewrite Eq. (2.16) as

$$\tilde{\mathbf{U}} \cdot \mathbf{e}^{(i)} = \tilde{U}_i = \int_{\Gamma} \tilde{\mathbf{v}}(\mathbf{x}) \cdot \boldsymbol{\lambda}^{(i)}(\mathbf{x}) dS_{\mathbf{x}}. \tag{2.17}$$

This allows to express the covariance of $\tilde{\mathbf{U}}$ as:

$$\begin{aligned} \langle \tilde{U}_i \tilde{U}_j \rangle &= \int_{\Gamma} \int_{\Gamma} \boldsymbol{\lambda}^{(j)}(\mathbf{x}) \cdot \langle \tilde{\mathbf{v}}(\mathbf{x}) \tilde{\mathbf{v}}(\mathbf{y}) \rangle \cdot \boldsymbol{\lambda}^{(i)}(\mathbf{y}) dS_{\mathbf{y}} dS_{\mathbf{x}} \\ &= \frac{2k_B T}{\Delta t} \int_{\Gamma} \int_{\Gamma} \boldsymbol{\lambda}^{(j)}(\mathbf{x}) \cdot \mathbb{G}(\mathbf{x} - \mathbf{y}) \cdot \boldsymbol{\lambda}^{(i)}(\mathbf{y}) dS_{\mathbf{y}} dS_{\mathbf{x}}. \end{aligned} \tag{2.18}$$

It can be shown (see Appendix A) that the last integral in Eq. (2.18) is equal to the (i, j) th element of the body mobility matrix \mathcal{N} , giving the desired result,

$$\langle \tilde{U}_i \tilde{U}_j \rangle = \frac{2k_B T}{\Delta t} \mathcal{N}_{ij}. \tag{2.19}$$

This shows that solving the SSBVP (2.10) gives the desired deterministic and stochastic particle velocity

$$\mathbf{U} = \tilde{\mathbf{U}} + \tilde{\mathbf{U}} = \mathcal{N}\mathbf{F} + \mathcal{N}^{\frac{1}{2}}\mathbf{W}. \tag{2.20}$$

While the LRT has been used in the past to analyze the nonhomogeneous Stokes BVP involving the fluctuating stress [45], here we employ it to establish that the homogeneous Stokes BVP given by Eq. (2.14) and Eq. (2.11) yields the correct statistics for the rigid body motion of the immersed particles. The removal of the fluctuating stress driving the surrounding fluid allows for the eventual application of boundary integral techniques to solve the SSBVP.

2.2. First-kind integral formulation

For rigid particles moving in Stokes flow, we observe that the details of what happens inside the particle do not actually matter for its motion and its hydrodynamic interactions with other particles or boundaries. Therefore, it is possible to extend the fluid to the entire domain so that the fluid inside the body moves with a velocity that is continuous across the boundary of the body. Once we extend the fluid to the interior of bodies, we may write down an alternative formulation of the SSBVP (2.10) as a *first-kind* boundary integral equation [12],

$$\mathbf{v}(\mathbf{x} \in \Gamma) = \mathbf{u} + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{q}) - \check{\mathbf{v}}(\mathbf{x}) = \int_{\Gamma} \mathbb{G}(\mathbf{x} - \mathbf{y}) \boldsymbol{\psi}(\mathbf{y}) dS_{\mathbf{y}}, \tag{2.21}$$

along with the force and torque balance conditions

$$\int_{\Gamma} \boldsymbol{\psi}(\mathbf{x}) dS_{\mathbf{x}} = \mathbf{f} \quad \text{and} \quad \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \boldsymbol{\psi}(\mathbf{x}) dS_{\mathbf{x}} = \boldsymbol{\tau}. \tag{2.22}$$

Equations (2.21) and (2.22) together define a linear system of equations to be solved for the *single-layer density* $\boldsymbol{\psi}(\mathbf{x} \in \Gamma)$ and particle velocity $\mathbf{U} = \{\mathbf{u}, \boldsymbol{\omega}\}$. We remark that $\boldsymbol{\psi}$ is the jump in the normal component of the stress when going across the body surface from the “interior” flow to the “exterior” flow. If $\check{\mathbf{v}} = 0$, then $\boldsymbol{\psi} = \boldsymbol{\lambda}$ is the traction.

In operator notation, we can write the system formed by Eqs. (2.21) and (2.22) as a *saddle-point* problem,

$$\begin{bmatrix} \mathcal{M} & -\mathcal{K} \\ -\mathcal{K}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi} \\ \mathbf{U} \end{bmatrix} = - \begin{bmatrix} \check{\mathbf{v}} \\ \mathbf{F} \end{bmatrix}, \tag{2.23}$$

where \mathcal{M} denotes the *single-layer* integral operator defined by

$$(\mathcal{M}\boldsymbol{\psi})(\mathbf{x} \in \Gamma) = \int_{\Gamma} \mathbb{G}(\mathbf{x} - \mathbf{y}) \boldsymbol{\psi}(\mathbf{y}) dS_{\mathbf{y}}, \tag{2.24}$$

and \mathcal{K} is a geometric operator that relates particle velocity to surface velocities,

$$(\mathcal{K}\mathbf{U})(\mathbf{x} \in \Gamma) = \mathbf{u} + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{q}), \tag{2.25}$$

and its adjoint \mathcal{K}^* is an integral operator that converts the single-layer density ψ to a force and torque,

$$\mathcal{K}^*\psi = \left(\int_{\Gamma} \psi(\mathbf{x}) \, dS_{\mathbf{x}}, \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \psi(\mathbf{x}) \, dS_{\mathbf{x}} \right) = (\mathbf{f}, \boldsymbol{\tau}). \tag{2.26}$$

The covariance of the random surface velocity $\check{\mathbf{v}}$ can be written as

$$\langle \check{\mathbf{v}} \check{\mathbf{v}} \rangle = \frac{2k_B T}{\Delta t} \mathcal{M}, \tag{2.27}$$

by which we mean that $\mathcal{M}\psi' = \langle (\check{\mathbf{v}}, \psi') \check{\mathbf{v}} \rangle$, for all ψ' in L^2 -space, and (\cdot, \cdot) denotes the L^2 -inner product defined by $(\mathbf{f}, \mathbf{g}) = \int_{\Gamma} \mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) \, dS_{\mathbf{x}}$. If the random surface velocity $\check{\mathbf{v}}$ were a function and could therefore be evaluated pointwise, Eq. (2.27) would simply be a formal rewriting of Eq. (2.11). However, we remind the reader again that Eq. (2.11) is also formal (in the same way that Eq. (2.2) is) and $\check{\mathbf{v}}$ is a distribution and therefore *cannot* be evaluated pointwise. We can define $\check{\mathbf{v}}$ more precisely as follows. Since \mathcal{M} is a compact, self-adjoint and positive-semidefinite operator in the L_2 sense, it has countably infinitely many eigenvalues $\lambda_i \geq 0$ and orthonormal eigenfunctions \mathbf{w}_i , so we may write $\check{\mathbf{v}}$ in the Karhunen–Loève expansion

$$\check{\mathbf{v}} \stackrel{\text{d.}}{=} \sum_{i=1}^{\infty} \sqrt{\lambda_i} W_i \mathbf{w}_i, \tag{2.28}$$

where as before W_i are independent Gaussian random variables with mean zero and variance $2k_B T / \Delta t$. In formal operator notation, we will write Eq. (2.28) as $\check{\mathbf{v}} = \mathcal{M}^{\frac{1}{2}} \mathbf{W}$, where $(\mathcal{M}^{\frac{1}{2}})^{\top} (\mathcal{M}^{\frac{1}{2}}) = \mathcal{M}$. We have found Eq. (2.27) (equivalently, Eq. (2.28)), rather than the deceptively simple Eq. (2.11), to be a suitable starting point for a finite-dimensional discretization of $\check{\mathbf{v}}$, as we explain shortly.

In this formulation, we require that $\check{\mathbf{v}}(\mathbf{x})$ is consistent with a divergence-free velocity field in the extended domain, i.e.,

$$\int_{\Gamma} \check{\mathbf{v}}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \, dS_{\mathbf{x}} = 0, \tag{2.29}$$

which is required for Eq. (2.23) to be solvable since the single-layer operator \mathcal{M} has a nontrivial null space consisting of single-layer densities that are normal to the boundary,

$$(\mathcal{M}\mathbf{n})(\mathbf{x} \in \Gamma) = \int_{\Gamma} \mathbb{G}(\mathbf{x} - \mathbf{y}) \mathbf{n}(\mathbf{y}) \, dS_{\mathbf{y}} = \mathbf{0}. \tag{2.30}$$

From Eq. (2.28), we note that $\check{\mathbf{v}} = \mathcal{M}^{\frac{1}{2}} \mathbf{W}$ is perpendicular to the null space of \mathcal{M} , and hence, the solvability condition Eq. (2.29) is fulfilled.

Formally,³ taking the Schur complement of Eq. (2.23) to eliminate ψ and solving for the body motion \mathbf{U} , we obtain

$$\begin{aligned} \mathbf{U} = \bar{\mathbf{U}} + \tilde{\mathbf{U}} &= (\mathcal{K}^* \mathcal{M}^{-1} \mathcal{K})^{-1} \mathbf{F} + (\mathcal{N} \mathcal{K}^* \mathcal{M}^{-1}) \check{\mathbf{v}} \\ &= (\mathcal{K}^* \mathcal{M}^{-1} \mathcal{K})^{-1} \mathbf{F} + (\mathcal{N} \mathcal{K}^* \mathcal{M}^{-1} \mathcal{M}^{\frac{1}{2}}) \mathbf{W} \\ &= \mathcal{N} \mathbf{F} + \mathcal{N}^{\frac{1}{2}} \mathbf{W}, \end{aligned} \tag{2.31}$$

which allows us to formally define \mathcal{N} and $\mathcal{N}^{\frac{1}{2}}$ explicitly as,

$$\begin{aligned} \mathcal{N} &= (\mathcal{K}^* \mathcal{M}^{-1} \mathcal{K})^{-1}, \\ \mathcal{N}^{\frac{1}{2}} &= \mathcal{N} \mathcal{K}^* \mathcal{M}^{-1} \mathcal{M}^{\frac{1}{2}}. \end{aligned} \tag{2.32}$$

Note that the random surface velocity $\check{\mathbf{v}}$ is in the range of \mathcal{M} by the construction of Eq. (2.28), so that $\mathcal{M}^{-1} \check{\mathbf{v}}$ is well-defined. We can therefore formally show that fluctuation–dissipation balance holds in the continuum operator sense,

³ For the operator \mathcal{M}^{-1} to be well-defined, we need to resort to its precise definition in either the space of band-limited functions or the fractional Sobolev space, but here, we will simply use this formal computation to inform our discretization and show later in Sec. 3.1 that our finite-dimensional discretization converges numerically.

$$\begin{aligned}
 \mathcal{N}^{\frac{1}{2}} \left(\mathcal{N}^{\frac{1}{2}} \right)^{\top} &= \mathcal{N} \mathcal{K}^* \mathcal{M}^{-1} \mathcal{M}^{\frac{1}{2}} \left(\mathcal{M}^{\frac{1}{2}} \right)^{\top} \mathcal{M}^{-1} \mathcal{K} \mathcal{N} \\
 &= \mathcal{N} \left(\mathcal{K}^* \mathcal{M}^{-1} \mathcal{K} \right) \mathcal{N} \\
 &= \mathcal{N} \left(\mathcal{N} \right)^{-1} \mathcal{N} = \mathcal{N}.
 \end{aligned}
 \tag{2.33}$$

This shows that the desired random velocity \mathbf{U} in Eq. (2.8) can be generated by solving Eq. (2.23), which is the first-kind boundary integral formulation of the SSBVP (2.10). While at first sight it may appear that we have simply formally rederived Eq. (2.19) by appealing to a first-kind BVP formulation, we will demonstrate next that the formal continuum formulation presented here has a well-defined finite-dimensional truncation that is very suitable for numerical computations.

3. Fluctuating boundary integral method

This section presents the fluctuating boundary integral method (FBIM) for suspensions of Brownian rigid particles in two dimensions. Our discrete formulation closely follows the (formal) continuum first-kind formulation presented in Sec. 2.2. We have found this to be much more effective than following the more general BVP formulation presented in Sec. 2.1. Specifically, one approach to discretizing the SSBVP Eq. (2.10) is to first generate a (smooth) random surface velocity *function* $\check{\mathbf{v}}$ using a *regularized*⁴ variant of Eq. (2.11), and then to use a standard spectrally-accurate second-kind boundary integral formulation to solve the resulting boundary-value problem. Our preliminary investigations of such an approach have revealed that regularization (truncation) of the covariance in Eq. (2.11) leads to a drastic loss of accuracy in numerical fluctuation–dissipation balance. Instead, by relating the covariance of the random surface velocity to the first-kind operator as in Eq. (2.27), the regularization of the *distribution* $\check{\mathbf{v}}$ becomes directly connected to the singular quadrature used to discretize \mathcal{M} . As we demonstrate here, this leads to a first-kind formulation that satisfies discrete fluctuation–dissipation (DFDB) to within solver tolerances, while preserving the underlying accuracy of the singular quadrature.

We begin by presenting a discrete formulation of the mobility problem (2.23) by first discretizing the continuum operators \mathcal{M} , \mathcal{K} and \mathcal{K}^* , to obtain a discrete saddle-point linear system, whose solution strictly obeys DFDB without any approximation. To efficiently solve the saddle-point linear system with Krylov iterative methods, we present the two key components of FBIM: a fast routine for computing matrix-vector product of the single-layer matrix \mathbf{M} (i.e., the discretized operator \mathcal{M}), and a fast method for generating the random surface velocity $\check{\mathbf{v}} = \mathbf{M}^{1/2} \mathbf{W}$. To address the inherent ill-conditioning of the linear system arising from the first-kind integral formulation, we will also discuss block-diagonal preconditioning for the iterative solvers.

3.1. Discrete formulation of the mobility problem

We present the discrete formulation of the mobility problem (2.23) by first discretizing the continuum operators \mathcal{K} , \mathcal{K}^* and \mathcal{M} . In the discrete formulation, for generality, we describe our method for many-body suspensions. Let us assume that Γ_β is parametrized by $\boldsymbol{\gamma}_\beta : [0, 2\pi] \rightarrow \mathbb{R}^2$. We introduce a collection of N_p equispaced points $s_j = j\Delta s$, $j \in \{1, \dots, N_p\}$, where $\Delta s = 2\pi/N_p$, and Γ_β is discretized by the collection of nodes $\mathbf{x}_\beta = \{\mathbf{x}_{\beta,j}\}_{j=1}^{N_p}$, where $\mathbf{x}_{\beta,j} = \boldsymbol{\gamma}_\beta(s_j)$. We also denote the composite vector of all nodes or points by $\mathbf{X} = \{\mathbf{x}_\beta\}_{\beta=1}^N$.

The discrete operator \mathbf{K} is a geometric matrix defined by

$$(\mathbf{K}\mathbf{U})(\mathbf{x}_{\beta,j}) = (\mathbf{K}_\beta \mathbf{U}_\beta)_j = \mathbf{u}_\beta + \boldsymbol{\omega}_\beta \times (\mathbf{x}_{\beta,j} - \mathbf{q}_\beta),
 \tag{3.1}$$

where \mathbf{K}_β is the sub-block of \mathbf{K} that maps the particle velocity $\mathbf{U}_\beta = \{\mathbf{u}_\beta, \boldsymbol{\omega}_\beta\}$ to the velocity at the node $\mathbf{x}_{\beta,j}$ on Γ_β .

The adjoint operator \mathcal{K}^* defined by Eq. (2.26) can be discretized by the periodic trapezoidal rule for each body,

$$\int_{\Gamma_\beta} \boldsymbol{\psi}(\mathbf{x}) \, dS_{\mathbf{x}} \approx \sum_{j=1}^{N_p} \boldsymbol{\mu}_{\beta,j},
 \tag{3.2a}$$

$$\int_{\Gamma_\beta} (\mathbf{x} - \mathbf{q}_\beta) \times \boldsymbol{\psi}(\mathbf{x}) \, dS_{\mathbf{x}} \approx \sum_{j=1}^{N_p} (\mathbf{x}_{\beta,j} - \mathbf{q}_\beta) \times \boldsymbol{\mu}_{\beta,j},
 \tag{3.2b}$$

where $\boldsymbol{\mu}_{\beta,j} = \boldsymbol{\psi}(\mathbf{x}_{\beta,j}) \Delta s$ denotes an unknown discrete boundary force at the node $\mathbf{x}_{\beta,j}$ on Γ_β . We can therefore write

$$(\mathcal{K}^* \boldsymbol{\psi})(\mathbf{X}) \approx \mathbf{K}^{\top} \boldsymbol{\mu},
 \tag{3.3}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_\beta\}_{\beta=1}^N$ is a composite vector that collects all the boundary forces.

⁴ The most direct way to regularize the singular Green's function is to represent it in Fourier space and then simply truncate the finite-dimensional sum to a finite number of Fourier modes.

To approximate the single-layer operator \mathcal{M} , we need to employ a singular quadrature that can handle the singularity of \mathbb{G} . The development of quadrature rules for singular or near-singular integrals is an active research area of its own [40,41,54–57]. In two dimensions, for simplicity, we will use the Alpert quadrature which is based on a modification of the trapezoidal rule with auxiliary nodes whose weights are configured to achieve the desired order of accuracy [54].

In matrix notation, the approximation of the single-layer integral operator evaluated on the vector of quadrature nodes \mathbf{X} can be compactly written as

$$(\mathcal{M}\psi)(\mathbf{X}) \approx \mathbf{M}\boldsymbol{\mu}, \quad (3.4)$$

where \mathbf{M} is the discretized matrix operator of \mathcal{M} , hereinafter referred to as the single-layer matrix. The details of constructing \mathbf{M} will be discussed in Sec. 3.2. We have chosen to keep track of surface forces in the discrete representation (rather than surface tractions as in the continuous case) as this yields both a symmetric saddle-point system in the usual sense, as well as the desirable property that $\mathbf{M}_{ij} \rightarrow \mathbb{G}(\mathbf{x}_i - \mathbf{x}_j)$ as $|\mathbf{x}_i - \mathbf{x}_j| \rightarrow \infty$. The matrix \mathbf{M} has a physical interpretation of a mobility matrix relating surface forces to surface velocities, which in turn suggests that \mathbf{M} should be symmetric (self-adjoint) in the standard L_2 sense.

Following the continuum formulation (see Eq. (2.27)), we require the discrete random surface velocity⁵ to satisfy

$$\langle \check{\mathbf{v}}\check{\mathbf{v}} \rangle = \frac{2k_B T}{\Delta t} \mathbf{M}. \quad (3.5)$$

More precisely, we need to generate a vector of Gaussian random variables $\check{\mathbf{v}}$ whose covariance is given by Eq. (3.5), and we denote it by $\check{\mathbf{v}} = \mathbf{M}^{1/2}\mathbf{W}$, where \mathbf{W} is a finite-dimensional vector of Gaussian random variables with mean zero and covariance

$$\langle W_i W_j \rangle = \frac{2k_B T}{\Delta t} \delta_{ij}, \quad (3.6)$$

and $\mathbf{M}^{1/2}$ satisfies

$$\left(\mathbf{M}^{1/2}\right) \left(\mathbf{M}^{1/2}\right)^\top = \mathbf{M}. \quad (3.7)$$

The discrete formulation of Eq. (2.23) is a saddle-point linear system for the boundary forces $\boldsymbol{\mu}$ and the rigid body motion \mathbf{U} ,

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{U} \end{bmatrix} = - \begin{bmatrix} \mathbf{M}^{1/2}\mathbf{W} \\ \mathbf{F} \end{bmatrix}. \quad (3.8)$$

By taking the Schur complement of \mathbf{M} and eliminating $\boldsymbol{\mu}$, we obtain

$$\begin{aligned} \mathbf{U} &= \mathbf{N}\mathbf{F} + \mathbf{N}^{\frac{1}{2}}\mathbf{W} \\ &= (\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{K})^{-1} \mathbf{F} + (\mathbf{N}\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{M}^{1/2})\mathbf{W}, \end{aligned} \quad (3.9)$$

from which we can define

$$\begin{aligned} \mathbf{N} &= (\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{K})^{-1}, \\ \mathbf{N}^{\frac{1}{2}} &= \mathbf{N}\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{M}^{1/2}, \end{aligned} \quad (3.10)$$

where \mathbf{M}^\dagger is the pseudo-inverse of \mathbf{M} , and \mathbf{N} is an approximation of \mathcal{N} up to the order of accuracy of Alpert quadrature. Under the definition of \mathbf{N} and $\mathbf{N}^{\frac{1}{2}}$, we can verify that our discrete formulation satisfies DFDB without any approximation,

$$\begin{aligned} \mathbf{N}^{\frac{1}{2}} \left(\mathbf{N}^{\frac{1}{2}}\right)^\top &= \mathbf{N}\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{M}^{1/2} \left(\mathbf{M}^{1/2}\right)^\top \mathbf{M}^\dagger \mathbf{K}\mathbf{N}, \\ &= \mathbf{N}(\mathbf{K}^\top \mathbf{M}^\dagger \mathbf{K})\mathbf{N}, \\ &= \mathbf{N}(\mathbf{N})^{-1} \mathbf{N} = \mathbf{N}. \end{aligned} \quad (3.11)$$

Note that these relations are well-defined finite-dimensional versions of the formal continuum equations (2.32) and (2.33).

To generate the random velocity given by Eq. (3.9) efficiently, we solve the saddle-point linear system (3.8) by GMRES. In the remaining sections, we present efficient numerical that compute the matrix-vector product $\mathbf{M}\boldsymbol{\mu}$ and generate the random surface velocity $\check{\mathbf{v}} = \mathbf{M}^{1/2}\mathbf{W}$.

⁵ The discrete $\check{\mathbf{v}}$ can be thought of as being a suitably scaled finite-volume representation of the distribution $\check{\mathbf{v}}$.

3.2. Fast matrix-vector multiplication for the single-layer matrix

In this section we develop a fast method to efficiently perform the matrix-vector product $\mathbf{M}\boldsymbol{\mu}$. The key idea for achieving linear-scaling is to use Ewald splitting to decompose the periodic Stokeslet as

$$\mathbb{G} = \mathbb{G}_\xi^{(w)} + \mathbb{G}_\xi^{(r)} = H * \mathbb{G} + (\mathbb{G} - H * \mathbb{G}), \tag{3.12}$$

where “*” denotes convolution, and $\mathbb{G}_\xi^{(w)}$ is a far-field (long-ranged) smooth kernel that decays exponentially in Fourier space, and $\mathbb{G}_\xi^{(r)}$ is a near-field (short-ranged) singular kernel that decays exponentially in real space. The choice of splitting function $H(r; \xi)$ by Hasimoto [31] is defined in Fourier space as

$$\widehat{H}(k; \xi) = \left(1 + \frac{k^2}{4\xi^2}\right) e^{-k^2/4\xi^2}, \tag{3.13}$$

where ξ is the splitting parameter that controls the rate of exponential decay.

Using the Fourier representation of the periodic Green’s function of Stokes flow we can compute the far-field kernel in wave space as

$$\mathbb{G}_\xi^{(w)}(\mathbf{r}) = \frac{1}{\eta V} \sum_{\mathbf{k} \neq 0} \frac{\widehat{H}(k; \xi)}{k^2} (\mathbf{I} - \widehat{\mathbf{k}}\widehat{\mathbf{k}}^\top) e^{-i\mathbf{k}\cdot\mathbf{r}}, \tag{3.14}$$

where $V = |\mathcal{V}| = L^2$, $\mathbf{k} \in \{2\pi\kappa_i/L : \kappa_i \in \mathbb{Z}, i = 1, 2\}$, and $\widehat{\mathbf{k}} = \mathbf{k}/k$ for $k = |\mathbf{k}|$. The near-field kernel is analytically computed in real space using the inverse Fourier transform,⁶

$$\mathbb{G}_\xi^{(r)}(\mathbf{r}) = \frac{1}{4\pi\eta} \left[\frac{1}{2} E_1(\xi^2 r^2) \mathbf{I} + \left(\frac{\mathbf{r} \otimes \mathbf{r}}{r^2} - \mathbf{I} \right) e^{-\xi^2 r^2} \right], \tag{3.15}$$

where $E_1(z)$ is the exponential integral defined by

$$E_1(z) = \int_1^\infty \frac{e^{-zt}}{t} dt = \int_z^\infty \frac{e^{-t}}{t} dt. \tag{3.16}$$

We observe from Eq. (3.15) that $\mathbb{G}_\xi^{(r)}$ also has the logarithmic singularity of \mathbb{G} , since in the limit $z \rightarrow 0$,

$$E_1(z) = -\gamma - \log z + O(z), \tag{3.17}$$

where γ is the Euler–Mascheroni constant. An important remark on the Hasimoto splitting is that it ensures both $\mathbb{G}_\xi^{(r)}$ and $\mathbb{G}_\xi^{(w)}$ are SPD, because $0 \leq \widehat{H}(k; \xi) \leq 1$ for all \mathbf{k} and ξ [24].

The splitting of \mathbb{G} naturally induces the splitting of the single-layer integral operator $\mathcal{M} = \mathcal{M}^{(r)} + \mathcal{M}^{(w)}$, and subsequently, the corresponding splitting of the single-layer matrix

$$\mathbf{M} = \mathbf{M}^{(r)} + \mathbf{M}^{(w)}, \tag{3.18}$$

where the elements of $\mathbf{M}^{(w)}$ are obtained by applying the regular trapezoidal rule to $\mathcal{M}^{(w)}$, which gives

$$\left(\mathbf{M}^{(w)}\right)_{mn} = \mathbb{G}_\xi^{(w)}(\mathbf{x}_m - \mathbf{x}_n). \tag{3.19}$$

The elements of $\mathbf{M}^{(r)}$ are obtained by applying Alpert’s hybrid Gauss–trapezoidal quadrature to $\mathcal{M}^{(r)}$, and can be further decomposed as

$$\mathbf{M}^{(r)} = \mathbf{M}^{(t)} + \mathbf{M}^{(a)}, \tag{3.20}$$

where $\mathbf{M}^{(t)}$ is the trapezoidal rule, i.e., $\left(\mathbf{M}^{(t)}\right)_{mn} = \mathbb{G}_\xi^{(r)}(\mathbf{x}_m - \mathbf{x}_n)$ for $m \neq n$, and we define it to be zero for $m = n$. The singular quadrature correction $\mathbf{M}^{(a)}$ is a banded matrix that contains Alpert weights for the logarithmic singularity of $\mathbb{G}_\xi^{(r)}$. The Alpert weights are defined on a set of auxiliary nodes that do not coincide with the trapezoidal nodes, so a local Lagrangian interpolation from the auxiliary nodes to the trapezoidal nodes is needed to obtain the elements of $\mathbf{M}^{(a)}$. Note that $\mathbf{M}^{(r)}$, $\mathbf{M}^{(w)}$ and $\mathbf{M}^{(a)}$ depend on ξ , but for conciseness of notation, the subscript ξ is omitted.

⁶ We gratefully thank Anna-Karin Tornberg for sharing with us notes on the Hasimoto splitting of the Stokeslet in two dimensions.

A key ingredient of FBIM is a fast method to compute the matrix-vector product

$$\mathbf{M}\boldsymbol{\mu} = \left(\mathbf{M}^{(a)} + \mathbf{M}^{(t)} + \mathbf{M}^{(w)} \right) \boldsymbol{\mu}. \quad (3.21)$$

We recall that the Alpert quadrature assigns only local correction weights to the trapezoidal nodes. Since $\mathbf{M}^{(a)}$ is block-diagonal and banded, matrix-vector products involving $\mathbf{M}^{(a)}$ can be computed in $\mathcal{O}(N)$ operations using vector rotations and sparse matrix-vector multiplications,

$$\left(\mathbf{M}^{(a)} \boldsymbol{\mu} \right)_\beta = \mathbf{R}_\beta \mathbf{M}_{\text{ref}}^{(a)} \mathbf{R}_\beta^\top \boldsymbol{\mu}_\beta, \quad \beta = 1, \dots, N, \quad (3.22)$$

where $\mathbf{M}_{\text{ref}}^{(a)}$ is a precomputed Alpert matrix for some reference configuration, and \mathbf{R}_β is the rotation matrix from the chosen reference configuration to the configuration of Γ_β .

To accelerate matrix-vector products involving $\mathbf{M}^{(r)}$ and $\mathbf{M}^{(w)}$ which are not sparse, we rely on the Spectral Ewald method [21]. For the near-field contribution, due to the short-ranged nature of $\mathbb{G}_\xi^{(r)}$, the action of $\mathbf{M}^{(t)}$ can be computed by using the cell list algorithm, commonly-used in Molecular Dynamics [58]. First, we partition the computational domain into $N_{\text{box}} \times N_{\text{box}}$ cells and sort the points into these cells, which takes $\mathcal{O}(N)$ work. The splitting parameter ξ is chosen such that the real-space sum converges to within a prescribed tolerance ϵ , at a cutoff radius $r_c = L/N_{\text{box}}$. For each target point, the real-space sum is reduced to a local interaction with source points in its own cell and in all adjacent cells (with periodicity), i.e., nine cells in two dimensions. If the density of points in each cell is approximately held fixed as the system size grows, then the complexity of the direct summation in the near field is also $\mathcal{O}(N)$.

In the SE method, the far-field contribution given by Eqs. (3.14) and (3.19) can be factorized as

$$\mathbf{M}^{(w)} = \mathbf{D}^* \mathbf{B} \mathbf{D}, \quad (3.23)$$

where the block-diagonal matrix \mathbf{B} is defined in the Fourier space as

$$\mathbf{B}(\mathbf{k}, \xi) = \frac{\widehat{H}(\mathbf{k}; \xi)}{k^2} (\mathbf{I} - \widehat{\mathbf{k}} \widehat{\mathbf{k}}^\top), \quad (3.24)$$

which essentially maps the Fourier representation of forces to velocities. The operator \mathbf{D} is the non-uniform Discrete Fourier Transform (NUDFT) that converts point forces $\boldsymbol{\mu} = \{\boldsymbol{\mu}_n\}$ on a collection of non-uniform source points $\{\mathbf{x}_n\}$ to Fourier space. The operator \mathbf{D} and its adjoint \mathbf{D}^* can be efficiently applied using the non-uniform Fast Fourier Transform (NUFFT) (see [59] and the references therein). In operator notation, we can express the NUFFT as

$$\mathbf{D} = \mathcal{C} \mathcal{F} \mathcal{S} \quad \text{and} \quad \mathbf{D}^* = \mathcal{S}^* \mathcal{F}^* \mathcal{C}^*, \quad (3.25)$$

where \mathcal{S} and \mathcal{S}^* are a pair of spreading and interpolation operators using Gaussian kernels, and \mathcal{F} and \mathcal{F}^* are the forward and inverse FFT operators on a uniform grid, and \mathcal{C} and \mathcal{C}^* are “deconvolution” operators.

The main idea of NUFFT is to first smear (spread) the point forces to a uniform grid using a Gaussian kernel, then make use of the FFT on the uniform grid, and finally apply the deconvolution (see [59]). In the SE method, the Gaussian kernel with the Fourier representation $e^{-\eta k^2/8\xi^2}$ is used for spreading and interpolation, where η is a free parameter that sets the shape of Gaussian [21,60]. We observe that this Gaussian with parameter η is different from the one in the Hasimoto function for Ewald splitting. The main purpose for introducing an extra parameter η is that it allows the SE method to independently control the width of spreading and interpolation, so that the accuracy of evaluating the wave-space sum with NUFFT is decoupled from the accuracy of the Ewald sum.

Next we address the choice of parameters in the SE method. The parameters that need to be set by the user include the number of partition cells N_{box} (or r_c), the splitting parameter ξ , the size of the Fourier grid M (even), the number of points P (odd) for spreading and interpolation in the NUFFT, and the Gaussian shape parameter m (which is related to η , see [21, Eq. (22)]). First, we set the number of partition cells N_{box} , primarily based on balancing the computational work between the real- and wave-space sums (see Sec. 4.3 for details), and set $r_c = L/N_{\text{box}}$. For a user-specified error tolerance level ϵ , the splitting parameter ξ and M are determined by the truncation error estimates in the real and wave spaces [21], respectively,

$$C_r e^{-\xi^2 r_c^2} \leq \epsilon, \quad (3.26a)$$

$$C_w e^{-k_{\text{max}}^2/4\xi^2} \leq \epsilon, \quad (3.26b)$$

where k_{max} corresponds to the largest mode of a grid of size $M \times M$. The constants in the error estimates are estimated empirically as in [21], and we have found that $C_r \approx 100$ and $C_w \approx 1$ are suitable for the Stokeslet in two dimensions. The remaining parameters P and m (or η) together determine the accuracy of evaluating the wave-space sum using NUFFT. From the error estimation of Lindbo and Tornberg [21,60], the optimal choice is $m \sim \sqrt{\pi P}$, so that the approximation error arising from the NUFFT (including quadrature error and Gaussian truncation error) is approximately $e^{-\pi P/2} \lesssim \epsilon$, from which P is determined.

The overall complexity of the wave-space sum is $\mathcal{O}(N)$ for spreading and interpolation and $\mathcal{O}(M^2 \log M^2)$ for the FFTs. For dense suspensions in two dimensions, we have found in our implementation that the computational work is dominated by spreading and interpolation instead of the FFTs.

3.3. Fast sampling of the random surface velocity

The second key component of FBIM is a fast routine for sampling the random surface velocity $\check{\mathbf{v}} = \mathbf{M}^{1/2} \mathbf{W}$. The action of $\mathbf{M}^{1/2}$ can be computed using iterative methods such as the Chebyshev polynomial approximation [22] or the Lanczos algorithm for matrix square root [23]. For the Stokeslet in two dimensions, applying iterative methods directly for \mathbf{M} are expected to scale poorly for dense suspensions because of its logarithmic growth in the far field. The condition number of \mathbf{M} , as well as the number of iterations required to converge for a given tolerance level, would grow with the system size. This would make the overall complexity for applying the action of $\mathbf{M}^{1/2}$ superlinear, even though the action of \mathbf{M} can be applied with $\mathcal{O}(N)$ work.

To improve the superlinear complexity for generating the random surface velocity with iterative methods, we apply the *Positively Split Ewald* (PSE) method (developed for the RPY tensor) to the two-dimensional periodic Stokeslet [24]. The main idea of PSE is to split the action of $\mathbf{M}^{1/2}$ into a near-field part $(\mathbf{M}^{(r)})^{1/2}$ and a far-field part $(\mathbf{M}^{(w)})^{1/2}$, and generate the random surface velocity as

$$\mathbf{M}^{1/2} \mathbf{W} \stackrel{\text{d.}}{=} (\mathbf{M}^{(r)})^{1/2} \mathbf{W}^{(r)} + (\mathbf{M}^{(w)})^{1/2} \mathbf{W}^{(w)}, \tag{3.27}$$

so that the near-field contribution can be rapidly generated by the Lanczos algorithm [23] in the real space, and the far-field contribution can be efficiently handled in the wave space by NUFFT. The right-hand-side of Eq. (3.27) defines one way of computing the action of $\mathbf{M}^{1/2}$ provided that $\mathbf{W}^{(r)}$ and $\mathbf{W}^{(w)}$ are two independent Gaussian random vectors.

For a sufficiently small cut-off radius or a sufficiently large ξ , the real-space kernel $\mathbb{G}_\xi^{(r)}$ decays exponentially, so that the near-field interaction is localized and the condition number of $\mathbf{M}^{(r)}$ does not grow with the number of bodies, while the packing fraction is held fixed. However, due to the singular nature of $\mathbb{G}_\xi^{(r)}$, the condition number of $\mathbf{M}^{(r)}$ may grow if the number of points per body increases. We have found that the Lanczos algorithm [61] with block-diagonal preconditioning (see Sec. 3.4) can significantly reduce the number of iterations. In the case when two bodies nearly touch, so that the problem itself becomes ill-conditioned, the block-diagonal approximation becomes worse and the number of iterations for all iterative solvers increases. Nevertheless, we have found that the block-diagonal preconditioner is still effective, and it takes a reasonable number of iterations for the Lanczos algorithm to converge, even for dense suspensions (see Fig. 4).

Note that the validity of Eq. (3.27) relies on the property that both $\mathbf{M}^{(r)}$ and $\mathbf{M}^{(w)}$ need to be SPD. We observe that, even though $\mathbb{G}_\xi^{(r)}$ is a SPD kernel, its singular contribution given by the Alpert correction matrix $\mathbf{M}^{(a)}$ is not symmetric for a general-shaped particle. We have confirmed numerically that using only the symmetric part of $\mathbf{M}^{(a)}$ does not affect the accuracy of the Alpert quadrature. In practice, we observe that the Lanczos algorithm is rather insensitive to spurious negative eigenvalues of small magnitude that may exist for $(\mathbf{M}^{(r)} + (\mathbf{M}^{(r)})^\top)/2$.

The far-field matrix $\mathbf{M}^{(w)}$ is SPD by construction because of Eq. (3.14), and we can rewrite Eq. (3.23) as

$$\mathbf{M}^{(w)} = (\mathbf{D}^* \mathbf{B}^{1/2}) (\mathbf{D}^* \mathbf{B}^{1/2})^*, \tag{3.28}$$

with $\mathbf{B}^{1/2}$ defined in the wave space as

$$\mathbf{B}^{1/2}(\mathbf{k}, \xi) = \frac{1}{k} \widehat{H}^{1/2}(k; \xi) (\mathbf{I} - \widehat{\mathbf{k}} \widehat{\mathbf{k}}^\top). \tag{3.29}$$

The far-field contribution of the random surface velocity can be generated by

$$(\mathbf{M}^{(w)})^{1/2} \mathbf{W}^{(w)} = \mathbf{D}^* \mathbf{B}^{1/2} \mathbf{W}^{(w)} = \mathbf{S}^* \mathcal{F}^* \mathbf{C}^* \mathbf{B}^{1/2} \mathbf{W}^{(w)}, \tag{3.30}$$

where $\mathbf{W}^{(w)}(\boldsymbol{\kappa})$ is a complex-valued random vector in the wave space, and $\boldsymbol{\kappa} = (\kappa_1, \kappa_2)$ for $\kappa_i \in \{-\frac{M}{2}, \dots, \frac{M}{2} - 1\}$. The sequence of operations in Eq. (3.30) can be interpreted as follows. We first generate random numbers in the wave space, and project onto the divergence-free subspace by the projection $\mathbf{I} - \widehat{\mathbf{k}} \widehat{\mathbf{k}}^\top$, then scale by $\widehat{H}^{1/2}(k; \xi)/k$, and apply deconvolution and (inverse) FFTs to obtain velocities in the real space on the grid, and finally, perform interpolation using a Gaussian kernel to obtain the random surface velocities on the particles. This is equivalent to how random velocities are generated in methods like FIB [27] and fluctuating FCM [28,29]. We remark that the cost of applying the action of $(\mathbf{M}^{(w)})^{1/2}$ in Eq. (3.30) is even cheaper than the action of $\mathbf{M}^{(w)}$, since it only requires half the work.

We note that certain complex-conjugate symmetry of $\mathbf{W}^{(w)}(\boldsymbol{\kappa})$ must be maintained to ensure its Fourier transform gives a real-valued Gaussian random vector with the correct covariance in the real space. Specifically, we require that, the zeroth mode $\boldsymbol{\kappa} = (0, 0)$ is set to be zero, and the Nyquist modes $\boldsymbol{\kappa} \in \{(-\frac{M}{2}, 0), (0, -\frac{M}{2}), (-\frac{M}{2}, -\frac{M}{2})\}$ are real-valued and generated from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{2 \times 2})$. All the remaining modes are generated by $\mathbf{W}^{(w)}(\boldsymbol{\kappa}) = \mathbf{a} + i\mathbf{b}$ for $\mathbf{a}, \mathbf{b} \in \mathcal{N}(\mathbf{0}, \frac{1}{2} \mathbf{I}_{2 \times 2})$, and have the complex-conjugate symmetry: $\mathbf{W}^{(w)}(\boldsymbol{\kappa}) = (\mathbf{W}^{(w)}(\boldsymbol{\kappa}'))^*$, where $\boldsymbol{\kappa}' = -\boldsymbol{\kappa} \pmod M$.

3.4. Block-diagonal preconditioning

This section presents the block-diagonal preconditioning technique introduced in [39] for solving the ill-conditioned linear system (3.8) with GMRES, and for generating the near-field contribution of the random surface velocity with the Lanczos algorithm [23,61]. The block-diagonal preconditioner for the linear system (3.8) is obtained by neglecting all hydrodynamic interactions between different bodies, i.e.,

$$\mathbf{P} = \begin{bmatrix} \tilde{\mathbf{M}} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix}, \quad (3.31)$$

where $\tilde{\mathbf{M}}$ is a block-diagonal approximation of \mathbf{M} obtained by setting elements of \mathbf{M} corresponding to pairs of points on *distinct* bodies to zero,

$$\tilde{\mathbf{M}}_{\alpha\beta} = \delta_{\alpha\beta} \mathbf{M}_{\alpha\beta}, \quad (3.32)$$

and the subscripts with Greek letters denote the sub-block containing interactions between Γ_α and Γ_β . Applying the preconditioner to Eq. (3.8) amounts to solving a linear system,

$$\begin{bmatrix} \tilde{\mathbf{M}} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{U} \end{bmatrix} = - \begin{bmatrix} \check{\mathbf{v}} \\ \mathbf{F} \end{bmatrix}, \quad (3.33)$$

which requires the action of the approximate body mobility matrix (Schur complement),

$$\tilde{\mathbf{N}} = \left(\mathbf{K}^\top \tilde{\mathbf{M}}^\dagger \mathbf{K} \right)^{-1}. \quad (3.34)$$

The approximate body mobility matrix $\tilde{\mathbf{N}}$ is also block-diagonal, and can be efficiently applied for each body,

$$\tilde{\mathbf{N}}_{\beta\beta} = \left(\mathbf{K}_\beta^\top \mathbf{M}_{\beta\beta}^\dagger \mathbf{K}_\beta \right)^{-1}. \quad (3.35)$$

The matrix block $\mathbf{M}_{\beta\beta}$ in Eq. (3.35) is a small matrix with size $2N_p \times 2N_p$, which is precomputed for the reference configuration,

$$\mathbf{M}_{\text{ref}} = \mathbf{M}_{\text{ref}}^{(a)} + \mathbf{M}_{\text{ref}}^{(t)} + \mathbf{M}_{\text{ref}}^{(w)}. \quad (3.36)$$

The action of $\mathbf{M}_{\beta\beta}^\dagger$ can be efficiently applied by using $\mathbf{M}_{\text{ref}}^\dagger$, which is also precomputed using a dense SVD decomposition or eigenvalue decomposition, and by using rotation matrices for different bodies because of the translational and rotational invariance of the free-space Stokeslet,

$$\mathbf{M}_{\beta\beta}^\dagger \approx \mathbf{R}_\beta \mathbf{M}_{\text{ref}}^\dagger \mathbf{R}_\beta^\top. \quad (3.37)$$

We note that the two sides of Eq. (3.37) do not equal exactly, since the Alpert quadrature is not rotation-invariant for a general-shaped body. This is not an issue, since the error introduced by this artifact is within the error tolerance, and the preconditioner does not need to be exactly inverted to work effectively.

For the near-field contribution of the random surface velocity, we use the preconditioned Lanczos algorithm [61] to generate

$$\check{\mathbf{v}}^{(r)} = \mathbf{G}^\dagger \left(\mathbf{G} \mathbf{M}^{(r)} \mathbf{G}^\top \right)^{1/2} \mathbf{W}^{(r)}, \quad (3.38)$$

where \mathbf{G} is a block-diagonal preconditioner, whose diagonal blocks can be precomputed as a dense matrix for the reference configuration using the eigenvalue decomposition,

$$\mathbf{M}_{\text{ref}}^{(r)} = \mathbf{V} \boldsymbol{\Sigma} \mathbf{V}^*, \quad (3.39)$$

and we set

$$\mathbf{G}_{\text{ref}} = (\boldsymbol{\Sigma}^\dagger)^{1/2} \mathbf{V}^* \quad \text{and} \quad \mathbf{G}_{\text{ref}}^\dagger = \mathbf{V} \boldsymbol{\Sigma}^{1/2}, \quad (3.40)$$

where the diagonal elements of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^\dagger$ corresponding to the spurious eigenvalues of $\boldsymbol{\Sigma}$ are set to be zero. This gives the desired discrete fluctuation–dissipation balance,

$$\left\langle \check{\mathbf{v}}_{\text{ref}}^{(r)} \check{\mathbf{v}}_{\text{ref}}^{(r)} \right\rangle = \mathbf{G}_{\text{ref}}^\dagger \mathbf{G}_{\text{ref}} \mathbf{M}^{(r)} \mathbf{G}_{\text{ref}}^\top \left(\mathbf{G}_{\text{ref}}^\dagger \right)^\top = \mathbf{M}_{\text{ref}}^{(r)}. \quad (3.41)$$

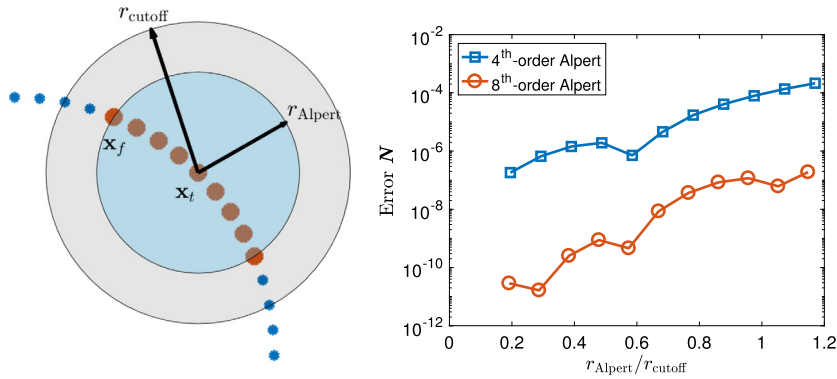


Fig. 1. (Left panel) A cartoon illustration of radius of Alpert correction r_{Alpert} and the cut-off radius of Ewald summation r_c . Both blue and red nodes represent the quadrature nodes in the trapezoidal rule, and the red nodes are the special nodes with Alpert correction weights centered a target node \mathbf{x}_t . The radius of Alpert correction is defined by $r_{\text{Alpert}} = |\mathbf{x}_t - \mathbf{x}_f|$, where \mathbf{x}_f is the last node with Alpert correction weight for a chosen order of Alpert quadrature. (Right panel) Normalized error (in matrix 2-norm) of \mathbf{N} with respect to \mathcal{N} (approximated with 12 digits of accuracy) versus the ratio r_{Alpert}/r_c for the 4th- and 8th-order Alpert quadratures. The mobility solver gradually loses accuracy because the singular kernel (with support r_c) is not sufficiently resolved by the Alpert quadrature as the ratio increases. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Generating $\check{\mathbf{v}}^{(r)}$ as in Eq. (3.38) is efficient, since we can reuse the precomputed matrices in Eq. (3.40) throughout the simulation, and apply rotation matrices for each body,

$$\mathbf{G}_{\beta\beta} = \mathbf{R}_{\beta} \mathbf{G}_{\text{ref}} \mathbf{R}_{\beta}^{\text{T}}. \tag{3.42}$$

We remark that the use of block-diagonal preconditioners does not increase the overall complexity of FBIM since they can be applied with $\mathcal{O}(N)$ work, and, furthermore, the cost is amortized over the length of the BD simulations.

4. Numerical results

This section presents numerical results of applying the FBIM to a number of benchmark problems in two dimensions. We first address the effect of the Ewald splitting parameter ξ on the accuracy of the first-kind mobility solver. Next we test the first-kind mobility solver by applying it to the steady Stokes flow through a square periodic array of disks, and compare the results to well-known analytical solutions. In the third test, we consider suspensions of Brownian rigid disks, and assess the effectiveness of FBIM by its accuracy and convergence, the robustness of iterative solvers, and its scalability to simulate suspensions of many-body particle systems. In the last set of numerical examples, we perform *Brownian Dynamics* (BD) simulations by combining the FBIM with previously-developed stochastic temporal integrators [27,52]. By simulating the free diffusion of a non-spherical (starfish-shaped) particle, we confirm that correctly handling the stochastic drift term in the temporal integrator is necessary in order to reproduce the equilibrium Gibbs–Boltzmann distribution. As a simple but nontrivial benchmark problem for many-body suspension, we investigate the dynamics of a pair of starfish-shaped particles connected by a harmonic spring, which includes interaction through the spring potential, hydrodynamic interaction between the particles and with their periodic images, as well as Brownian noises.

4.1. Choosing the Ewald splitting parameter

The first important aspect of FBIM that needs to be addressed is how the choice of ξ affects the accuracy of the first-kind mobility solver. In the *Spectral Ewald* (SE) method [21] the choice of ξ is only based on balancing the computational work between the real- and Fourier-space sums. In practice, we want to choose a sufficiently large ξ (i.e., a short-ranged singular kernel $\mathbb{G}_{\xi}^{(r)}$), so that the computational work in the real-space can be made cheap at the expense of the FFT in Fourier space. In this work, we demonstrate that the choice of ξ is also limited by the accuracy of Alpert quadrature used to resolve the logarithmic singularity of \mathbb{G} (diagonal elements of \mathbf{M}).

The Alpert quadrature can be viewed as assigning (interpolated) local correction weights to nearby quadrature nodes of a target point \mathbf{x}_t , as illustrated in the left panel of Fig. 1. For example, the number of (one-sided) quadrature nodes with nonzero correction weights is 4 and 8 for the 4th- and 8th-order Alpert quadrature, respectively. Since $\mathbb{G}_{\xi}^{(r)}$ decays exponentially with length scale ξ^{-1} , the Alpert quadrature grid must resolve length scales smaller than ξ^{-1} in order to capture the logarithmic singularity of $\mathbb{G}_{\xi}^{(r)}$ at the origin. Thus for a fixed grid, as ξ increases, we expect the accuracy of the Alpert quadrature to become progressively worse.

To demonstrate this issue, we study how the error of \mathbf{N} changes with ξ as follows. We fix the packing fraction $\phi = \pi/16 \approx 0.196$ and $N_p = 64$ but use different values of ξ (i.e., different r_c). The radius of Alpert correction r_{Alpert} is defined by the distance between the target point \mathbf{x}_t and the farthest quadrature node \mathbf{x}_f with nonzero Alpert weight, i.e., $r_{\text{Alpert}} =$

$|\mathbf{x}_t - \mathbf{x}_f|$. Note that r_{Alpert} is fixed once the quadrature order is fixed. The Ewald sum is computed with accuracy $\epsilon_{\text{Ewald}} = 10^{-9}$ and the tolerance level of GMRES is $\epsilon_{\text{tol}} = 10^{-9}$. The body mobility matrix \mathbf{N} is computed by solving the deterministic mobility problem with force and torque set to be the columns of the 3×3 identity matrix, which gives the corresponding columns of \mathbf{N} . We can also compute the exact body mobility matrix \mathcal{N} to 12 digits of accuracy with $N_p = 256$ by using the second-kind boundary integral formulation (attached as supplementary material of this paper). In two dimensions the resulting linear system of the second-kind formulation is well-conditioned and its solution is spectrally accurate. In the right panel of Fig. 1, we show the normalized error of \mathbf{N} (in matrix 2-norm) with respect to the 12-digit accurate approximation of \mathcal{N} for the 4th- and 8th-order Alpert quadrature, and the error increases by at least two orders of magnitude for the range of r_{Alpert}/r_c considered. We conclude that choosing ξ with $r_{\text{Alpert}}/r_c \lesssim 0.6$ maintains the accuracy of Alpert quadrature sufficiently well.

4.2. Square lattice of disks

Steady Stokes flow around a square periodic array of fixed disks in two dimensions is one of the classical problems in fluid mechanics, and its analytic solution is a thoroughly studied topic in the literature. Notably, Hasimoto [31] obtained an analytical expression for the drag force F on a dilute array of disks moving with velocity U by solving the steady Stokes equations with Fourier series expansions. Later, Sangani and Acrivos [62] extended Hasimoto's solution to the semi-dilute regime by including higher-order correction terms, and obtained the expansion

$$\frac{F}{\eta U} = \frac{4\pi}{-\ln \sqrt{\phi} - 0.738 + \phi - 0.887\phi^2 + 2.039\phi^3 + O(\phi^4)}, \quad (4.1)$$

where a is the radius of disk, l is the spacing of the square lattice of disks, and $\phi = \pi a^2/l^2$ is the packing fraction. Another important regime for which there are theoretical solutions is the densely-packed regime. In this regime, the disks almost touch, so that there is flow through a narrow gap between two neighboring disks, and a lubrication correction is generally required to extend the solution to this regime. Sangani and Acrivos [63] obtained using lubrication theory the asymptotic formula

$$\frac{F}{\eta U} \approx \frac{9\pi}{2\sqrt{2}} \varepsilon^{-\frac{5}{2}} \quad (4.2)$$

where $\varepsilon = (l - 2a)/l = 1 - \sqrt{4\phi/\pi}$ is the relative gap between two neighboring disks.

In the following test, we numerically solve the Stokes mobility problem (2.23) for the rigid body motion \mathbf{U} with applied force and torque $\mathbf{F} = (1, 0, 0)$ to determine the relationship between the drag force and velocity. We set $a = 1.0$ and $\epsilon_{\text{Ewald}} = \epsilon_{\text{tol}} = 10^{-9}$, while varying the length of square lattice l to achieve different packing fractions. For this simple test problem, we do not seek to optimize code performance, and we set $N_{\text{box}} = 4^2$ ($\xi \approx 20.13$) for all the test cases. In Sec. 4.3 we will address the choice of optimal ξ in the example of dense suspension of rigid particles. For a dilute or semi-dilute suspension ($\phi < 0.2$), the boundary Γ is discretized with $N_p = 64$ quadrature nodes, which is a sufficient resolution for the first-kind solver to give at least 6 digits of accuracy. For a higher packing fraction ($\phi > 0.2$), the number of quadrature nodes is determined by the ratio of the smallest gap between any two neighboring disks $d_g = l - 2a$ to the spacing between quadrature nodes $d_s = 2\pi a/N_p$. In a moderately-resolved computation, we require that the quadrature node spacing is comparable or smaller than the gap spacing, i.e., $d_s/d_g \lesssim 1$. For example, for the highest packing fraction considered in this test $\phi = 0.76$ (note that for a close-packed square lattice $\phi_{\text{max}} \approx 0.7854$), the number of quadrature nodes set by the ratio $d_s/d_g = 1$ is $N_p \approx 190$.

The linear system (3.8) is solved by GMRES with block-diagonal preconditioning. Numerical results for the normalized drag force over a broad range of packing fractions are shown in Fig. 2. We have obtained very good agreement with the dilute theory (4.1) ($\phi < 0.2$). For the dense packing fractions, the dilute theory no longer provides a good description for the drag, but our solutions from the first-kind solver match the dense theory (4.2) very well, as shown in the right panel of Fig. 2. The solutions from the first-kind solver are also in excellent agreement with the highly-accurate solutions (at least 9 digits of accuracy) from the second-kind solver for the range of packing fractions considered.

4.3. Suspension of Brownian rigid disks

In this section our primary goal is to study the performance of FBIM applying to suspensions of Brownian rigid particles in two dimensions. We will assess the effectiveness of FBIM from three aspects: accuracy and convergence, robustness of iterative solvers, and its efficiency and scalability to many-body particle systems. For simplicity, we focus on suspensions of disks only, however, there is no significant difficulty in applying the FBIM to more general bodies. In the top panel of Fig. 3, we show two random configurations of $N = 100$ disks with different packing fractions: a dilute suspension with $\phi = 0.25$ and a dense suspension with $\phi = 0.5$. These configurations are generated by using an event-driven molecular dynamics code. Since the random disks may nearly touch in the absence of (electrostatic) repulsive forces, especially when the packing fraction is high, we first generate a random configuration of disks with radius a_0 at a higher packing fraction ϕ_0 , and then adjust the actual radius of disks a to achieve the desired packing fraction ϕ . In this approach, the

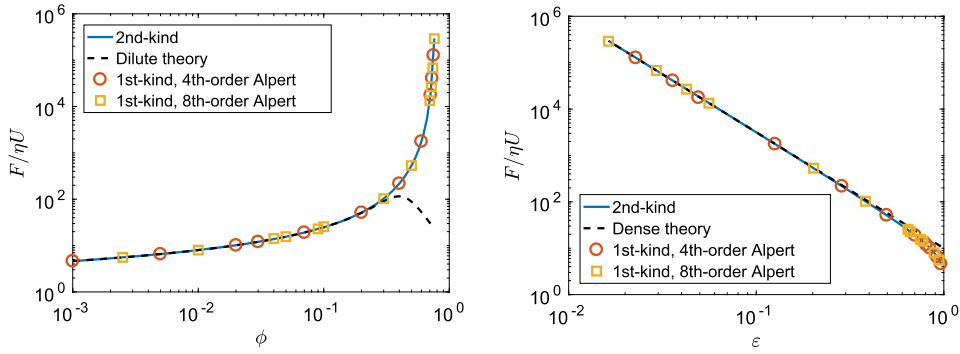


Fig. 2. The drag coefficients for a square periodic array of disks in steady Stokes flow, (Left panel) as a function of packing fraction ϕ , and compared to the dilute theory (4.1); (Right panel) as a function of the relative gap between disks ϵ , and compared to the dense theory (4.2). In both panels, the numerical results obtained from the first-kind solver with Alpert quadrature match very well with the spectrally-accurate results from the second-kind solver. (For details of the second-kind boundary integral formulation of the Stokes BVP, the reader is referred to the supplementary material of this paper.)

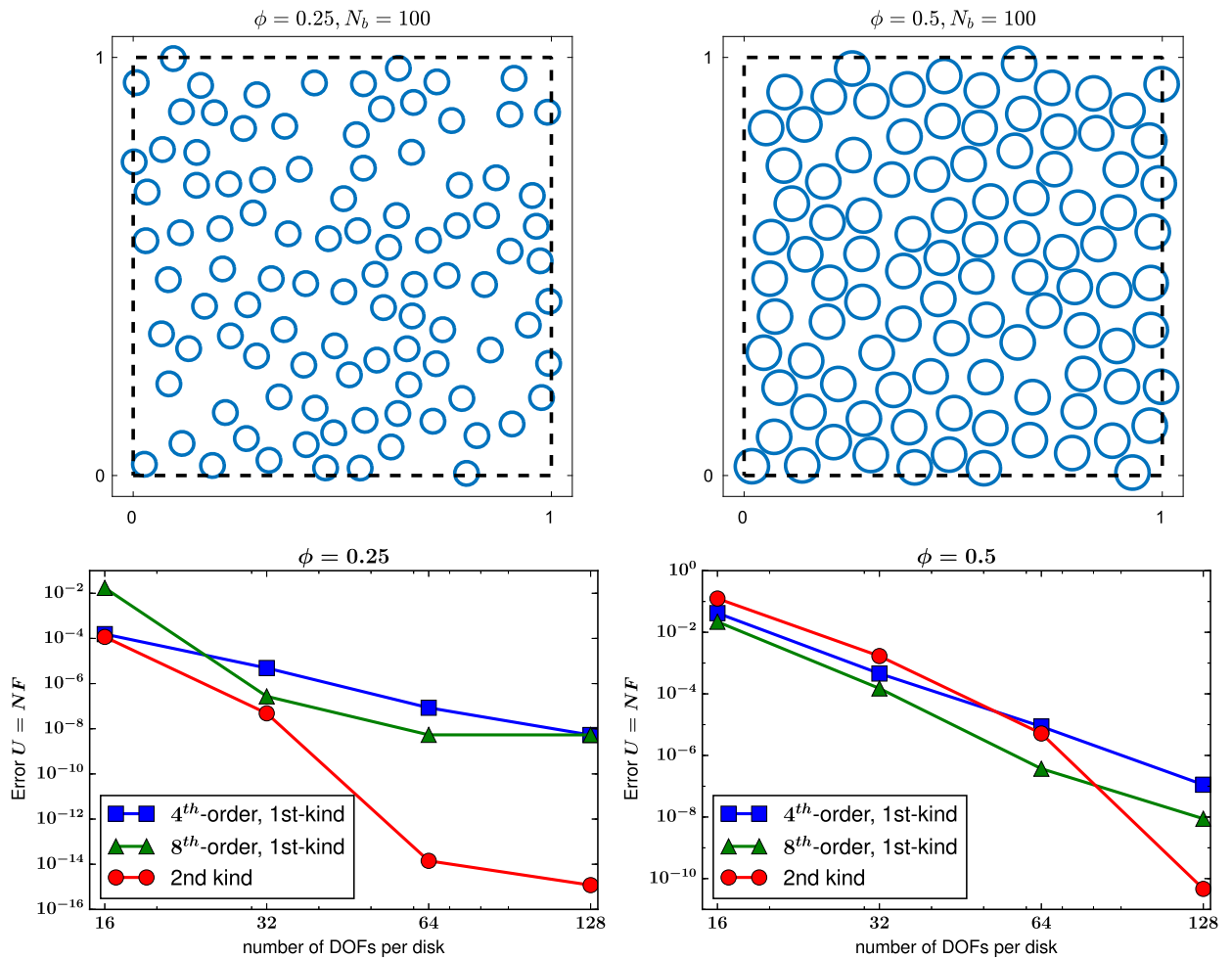


Fig. 3. (Top panel) Two random configurations of 100 rigid disks with packing fractions $\phi = 0.25$ (dilute) and $\phi = 0.5$ (dense) in a periodic unit cell. (Bottom panel) Normalized error of the mobility $U = NF$ versus number of degrees of freedom (DOFs) per disk. For the dilute suspension, the second-kind solver converges spectrally fast and is generally more accurate than the first-kind solver. For the dense suspension, while the second-kind solver converges faster, the first-kind solver is actually more accurate at lower DOFs per disk.

pairs of random disks are separated by a relative minimum distance $d_{\min}/a = 2(\sqrt{\phi_0/\phi} - 1)$. For the dilute suspension ($\phi = 0.25$), we use $\phi_0 = 0.4$ so that $d_{\min}/a \approx 0.523$, and for the dense suspension ($\phi = 0.5$), we use $\phi_0 = 0.6$ so that $d_{\min}/a \approx 0.191$.

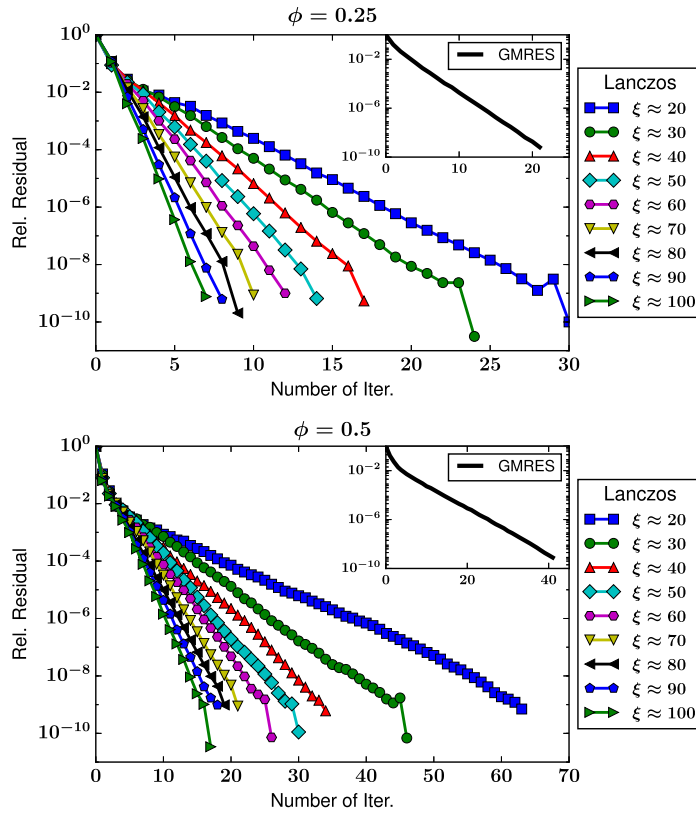


Fig. 4. Convergence of the Lanczos iteration for generating $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ and the GMRES iteration (inset) for solving the saddle-point linear system Eq. (3.8) with block-diagonal preconditioning for both iterative solvers. Generally, it requires more GMRES iterations for the dense packing ($\phi = 0.5$), as shown in the inset. The number of GMRES iteration does not depend on ξ , since the choice of ξ does not change \mathbf{M} in the linear system (3.8). The number of Lanczos iteration decreases with ξ , since the real-space kernel $\mathbb{G}_{\xi}^{(r)}$ becomes more short-ranged as ξ increases, and therefore, the block-diagonal preconditioner gets progressively better in approximating the inverse of $(\mathbf{M}^{(r)})^{1/2}$.

First, we investigate the accuracy and convergence of the first-kind mobility solver by applying it to the random configurations of disks shown in Fig. 3, subject to random forces and torques \mathbf{F} (without random surface velocity $\check{\mathbf{v}}$). In the first-kind mobility solver, we set $\epsilon_{\text{Ewald}} = \epsilon_{\text{tol}} = 10^{-9}$, and choose the splitting parameter $\xi \approx 50$ (or $N_{\text{box}} = 10^2$). Although this value of ξ does not achieve the minimum CPU time (see Fig. 5), it ensures $r_{\text{Alpert}}/r_c \lesssim 0.5$ for the 8th-order Alpert quadrature, so that the singularity is sufficiently resolved in all test cases for convergence study purpose. For this set of computations, we consider different numbers of quadrature nodes, or degrees of freedom (DOFs) per body, $N_p \in \{16, 32, 64, 128\}$. The normalized error of $\mathbf{U} = \mathbf{N}\mathbf{F}$ is computed with respect to a 12-digit accurate solution computed from the second-kind solver with 256 DOFs per disk. For the dilute suspension (bottom left panel of Fig. 3), the second-kind solver converges spectrally fast and is more accurate than the solutions from the first-kind solver. For the dense suspension (bottom right panel of Fig. 3), while the second-kind solver converges faster and gives more accurate solutions for large number of DOFs per body, the first-kind solver is more accurate for low resolutions. This can be explained as follows. When two disks nearly touch, the singular kernel of the first-kind integral (Stokeslet) grows as $\log r$, but the singular kernel of the second-kind integral (stresslet) grows as r^{-1} in two dimensions. This observation implies that the first-kind mobility solver is more practical than the second-kind solver for simulating suspensions of rigid particles with high packing fractions, since it can produce sufficiently accurate solutions with a smaller number of DOFs per body.

Next, we study the robustness of iterative methods for solving the saddle-point linear system (3.8). For the random configurations shown in Fig. 3, in addition to random forces and torques, we also generate random surface velocity $\check{\mathbf{v}} = \mathbf{M}^{1/2}\mathbf{W}$, and include it on the right-hand-side of Eq. (3.8). We show in the inset of Fig. 4 the residual versus the number of GMRES iterations for solving Eq. (3.8) with block-diagonal preconditioning. It would take more than 3 times the number of GMRES iterations to converge to the same tolerance level without preconditioning. In general, when the packing fraction grows or when two disks get closer, the hydrodynamic interaction between the disks becomes stronger, and hence, the condition number of the linear system grows. As a result, it requires more GMRES iterations for the dense suspensions, as expected. The number of GMRES iteration is independent of ξ , since the choice of ξ does not change $\mathbf{M} = \mathbf{M}^{(r)} + \mathbf{M}^{(w)}$ in the linear system (3.8). We expect that GMRES will converge faster for three-dimensional problems because the Stokeslet decays faster (r^{-1} in the far field) in three dimensions. We also show in Fig. 4 the residual versus the number of Lanczos

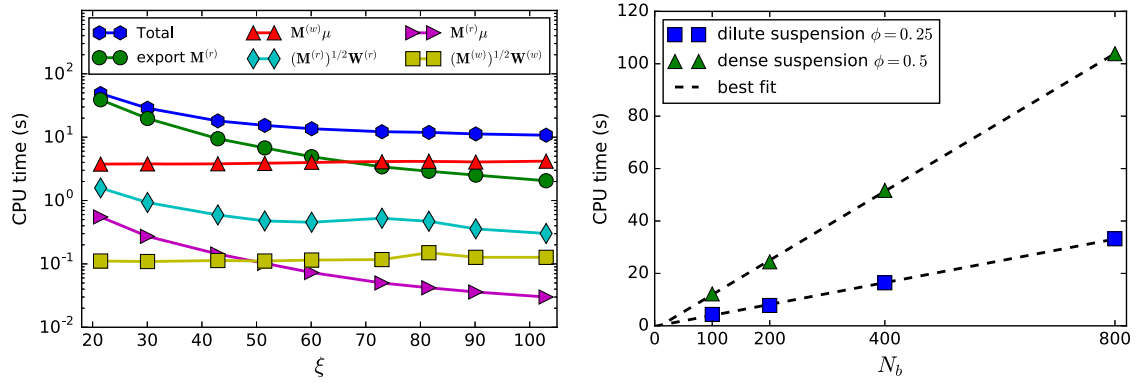


Fig. 5. (Left panel) Execution time of FBIM and its algorithmic components versus ξ for the dense packing $\phi = 0.5$. In our implementation, the computational work of FBIM is dominated by the subroutines for exporting $\mathbf{M}^{(r)}$ sparsely and for evaluating the matrix-vector product $\mathbf{M}^{(w)}\boldsymbol{\mu}$ (iteratively in GMRES). The total execution time of FBIM decreases with ξ , and the minimum CPU time is achieved near $\xi \approx 70$, where the CPU time for exporting $\mathbf{M}^{(r)}$ and evaluating $\mathbf{M}^{(w)}\boldsymbol{\mu}$ crosses. (Right panel) Linear scaling of FBIM with growing number of disks in the suspension, while the packing fraction is held fixed.

iterations for generating $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ with block-diagonal preconditioning, for different values of ξ . The number of Lanczos iterations decreases with ξ for both the dilute and dense suspensions. This can be explained by the observation that $\mathbb{G}_\xi^{(r)}$ becomes more short-ranged as ξ increases, and therefore, the block-diagonal preconditioner gets progressively better in approximating the inverse of $(\mathbf{M}^{(r)})^{1/2}$.

We now present profiling and scaling analysis of FBIM. First, we present the profiling results of FBIM and its algorithmic components in Fig. 5, and analyze the optimal choice of the splitting parameter ξ . We focus on the densely-packed configuration with 64 DOFs per disk ($d_s/d_g \approx 0.5$), and use the 4th-order Alpert quadrature. The accuracy of the first-kind mobility solver with these parameters is about 10^{-5} (see bottom right panel of Fig. 3), so we set $\epsilon_{\text{Ewald}} = \epsilon_{\text{tol}} = 10^{-6}$. We note that the execution time depends heavily on the choice of programming language, implementation and hardware. Our proof-of-concept serial implementation of FBIM in two dimensions is written in MATLAB with some subroutines accelerated by C with the aid of MEX files. As previously discussed in Sec. 3, the main ingredients of FBIM are evaluating the matrix-vector products $\mathbf{M}^{(r)}\boldsymbol{\mu}$ and $\mathbf{M}^{(w)}\boldsymbol{\mu}$ in GMRES, and generating $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ and $(\mathbf{M}^{(w)})^{1/2}\mathbf{W}^{(w)}$ using the Lanczos iteration. In our implementation, we found it optimal to export and store $\mathbf{M}^{(r)}$ sparsely for rapid matrix-vector multiplication in Lanczos and GMRES. In our profiling analysis, we profile the time to export $\mathbf{M}^{(r)}$ sparsely, the cumulative time to evaluate $\mathbf{M}^{(r)}\boldsymbol{\mu}$ and $\mathbf{M}^{(w)}\boldsymbol{\mu}$ in GMRES separately, the time to generate $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ and $(\mathbf{M}^{(w)})^{1/2}\mathbf{W}^{(w)}$ separately, and the total execution time of FBIM. We note that the total execution time of FBIM also includes the time of applying the preconditioners and other overhead time. The left panel of Fig. 5 shows the profiling results of the densely-packed configuration for different values of ξ . First, we observe that our implementation of FBIM is dominated by two subroutines: exporting $\mathbf{M}^{(r)}$ and evaluating $\mathbf{M}^{(w)}\boldsymbol{\mu}$ (iteratively). The CPU time of the sparse matrix-vector product $\mathbf{M}^{(r)}\boldsymbol{\mu}$ and generating $(\mathbf{M}^{(w)})^{1/2}\mathbf{W}^{(w)}$ (non-iteratively) is negligible. The CPU time for generating $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ is also small because of the rapid matrix-vector multiplication of the sparse matrix $\mathbf{M}^{(r)}$. The CPU time for generating $(\mathbf{M}^{(r)})^{1/2}\mathbf{W}^{(r)}$ also includes the time for applying the preconditioner, which accounts for about 20%–30% of the computational work.

Generally, the execution time of the real-space subroutines decreases with ξ , because the amount of work in the real space reduces as $\mathbb{G}_\xi^{(r)}$ becomes more short-ranged with ξ . On the other hand, the execution time of Fourier-space subroutines remains almost constant for the range of ξ considered in this test. This is because the cost of grid operations in Fourier-space sums (spreading/interpolation of a Gaussian to grid in NUFFT) dominates the cost of FFT in two dimensions. We expect the FFT cost would eventually become dominant in three dimensions. In our implementation, we observe that even the total CPU time of FBIM decreases with ξ , whereas Fiore et al. [24] report an optimal ξ for their GPU implementation of the PSE method in three dimensions. There is a wide range of ξ that approximately minimizes the total CPU time (from $\xi \approx 70$ to $\xi \approx 100$), as shown in the left panel of Fig. 5. We recall, however, that the choice of ξ in FBIM is also limited by the accuracy of Alpert quadrature. Using the criterion that $r_{\text{Alpert}}/r_c \lesssim 0.6$ for the densely-packed configuration, we obtain that $\xi \lesssim 150$ for the 4th-order Alpert quadrature, and that $\xi \lesssim 75$ for the 8th-order Alpert quadrature. Similar optimal range of ξ is also obtained in the profiling analysis for the dilute configuration.

Another important computational aspect that needs to be addressed is how the FBIM scales as the number of rigid particles grows while the packing fraction is held fixed. This aspect of FBIM is particularly important for applications involving a large number of particles. In the right panel of Fig. 5, we report the total execution time of FBIM with increasing number of rigid disks, while the packing fraction is held fixed at $\phi = 0.25$ and $\phi = 0.5$, and ξ is also fixed ($\xi \approx 70$) for both configurations. We conclude that FBIM scales linearly in the number of rigid particles for both dilute and dense suspensions.

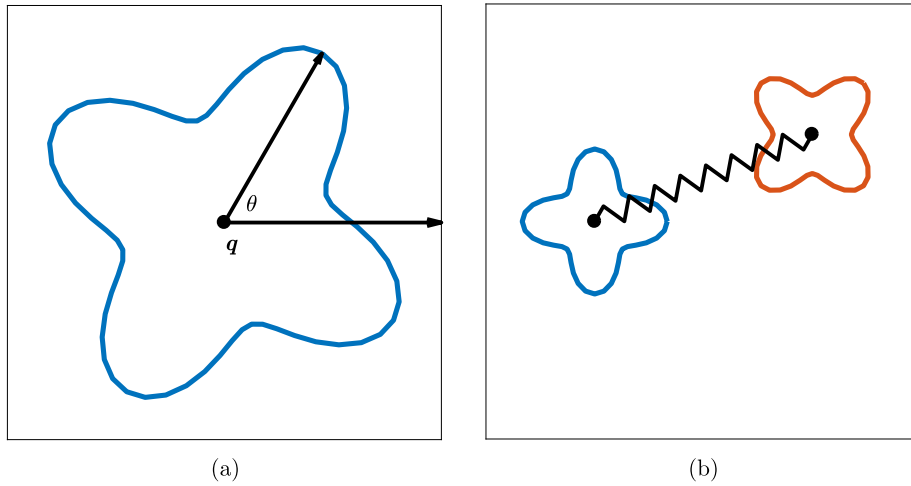


Fig. 6. (Left panel) A single starfish particle with four-fold rotational symmetry, described by the position of tracking point \mathbf{q} and its rotation θ . (Right panel) Equilibrium position for a pair of starfish particles connected by a harmonic spring. The enclosing box shown in the figure is the periodic unit cell.

4.4. Brownian dynamics of rigid particles

In the following tests, we combine the FBIM with stochastic temporal integrators to perform Brownian Dynamics (BD) for rigid particles. Applying the weakly first-order accurate Euler–Maruyama (EM) scheme to Eq. (2.6), we obtain the BD algorithm,

$$\begin{aligned} \mathbf{Q}^{n+1} = & \mathbf{Q}^n + \Delta t \mathbf{N}^n \mathbf{F}^n + \sqrt{2k_B T \Delta t} (\mathbf{N}^n)^{\frac{1}{2}} \mathbf{W}^n \\ & + \Delta t \frac{k_B T}{\delta} \left[\mathbf{N} \left(\mathbf{Q}^n + \frac{\delta}{2} \tilde{\mathbf{W}}^n \right) \tilde{\mathbf{W}}^n - \mathbf{N} \left(\mathbf{Q}^n - \frac{\delta}{2} \tilde{\mathbf{W}}^n \right) \tilde{\mathbf{W}}^n \right], \end{aligned} \quad (4.3)$$

where Δt is the time step size, the superscript denotes the time step level at which quantities are evaluated (e.g., $\mathbf{Q}^n = \mathbf{Q}(t = n\Delta t)$ and $\mathbf{N}^n = \mathbf{N}(\mathbf{Q}^n)$), δ is a small parameter, and \mathbf{W}^n and $\tilde{\mathbf{W}}^n$ are uncorrelated vectors of i.i.d standard Gaussian random variables.

The last term in Eq. (4.3) is a centered *random finite difference* (RFD) approximation to the stochastic drift term that is equal in expectation to $(\Delta t k_B T)(\partial_{\mathbf{Q}} \cdot \mathcal{N})^n$ for sufficiently small δ [27,43,44]. The RFD term guarantees that the EM scheme is a consistent stochastic integrator of Eq. (2.6), but is simpler and more efficient in practice than the Fixman midpoint scheme [51], in which the action of $\mathbf{N}^{-\frac{1}{2}}$ is required. The choice of δ is determined by a balance between truncation and roundoff error in the centered RFD, which gives $\delta/a \sim \epsilon^{1/3}$, where ϵ is the accuracy of the matrix-vector product $\mathbf{N}\mathbf{F}$ and a is the characteristic length of the particle. We note that the EM scheme requires solving the saddle-point linear system (3.8) three times: once for generating the velocity $\mathbf{N}^n \mathbf{F}^n + \sqrt{2k_B T / \Delta t} (\mathbf{N}^n)^{\frac{1}{2}} \mathbf{W}^n$, and twice for the RFD approximation.

The EM scheme is not particularly accurate even for the deterministic motion. Another weakly first-order accurate temporal integrator that has been observed to give a better accuracy is the stochastic Adams–Bashforth (AB) scheme [44],

$$\begin{aligned} \mathbf{Q}^{n+1} = & \mathbf{Q}^n + \Delta t \left(\frac{3}{2} \mathbf{N}^n \mathbf{F}^n - \frac{1}{2} \mathbf{N}^{n-1} \mathbf{F}^{n-1} \right) + \sqrt{2k_B T \Delta t} (\mathbf{N}^n)^{\frac{1}{2}} \mathbf{W}^n \\ & + \Delta t \frac{k_B T}{\delta} \left[\mathbf{N} \left(\mathbf{Q}^n + \frac{\delta}{2} \tilde{\mathbf{W}}^n \right) \tilde{\mathbf{W}}^n - \mathbf{N} \left(\mathbf{Q}^n - \frac{\delta}{2} \tilde{\mathbf{W}}^n \right) \tilde{\mathbf{W}}^n \right], \end{aligned} \quad (4.4)$$

in which the deterministic mobility $\mathcal{N}\mathbf{F}$ in Eq. (2.6) is approximated by the second-order Adams–Bashforth approximation. We observe, however, that the AB scheme is more expensive to use in practice, because it requires four mobility problem solves instead of three in the EM scheme. More efficient schemes can be developed but are not the focus of our work [53].

4.4.1. Free diffusion of a single starfish

We consider a starfish-shaped particle freely diffusing (i.e., no applied force and torque) in a periodic unit lattice. A similar benchmark problem was studied by Delong et al. [27] using the FIB method, and by Delmotte et al. [29] using the FCM method for a single spherical particle in a periodic domain. In their case the body mobility matrix does not depend on the position of the particle, and therefore $\partial_{\mathbf{Q}} \cdot \mathcal{N} = 0$. However, for a starfish \mathcal{N} depends on orientation due to interactions with periodic images and $\partial_{\mathbf{Q}} \cdot \mathcal{N}$ is nonzero.

The starfish particle has four-fold rotational symmetry (left panel of Fig. 6), and is described by

$$\Gamma : (\mathbf{x}(s), \mathbf{y}(s)) = r_s (1 + b \cos(4s)) \cdot (\cos s, \sin s), \quad s \in [0, 2\pi], \quad (4.5)$$

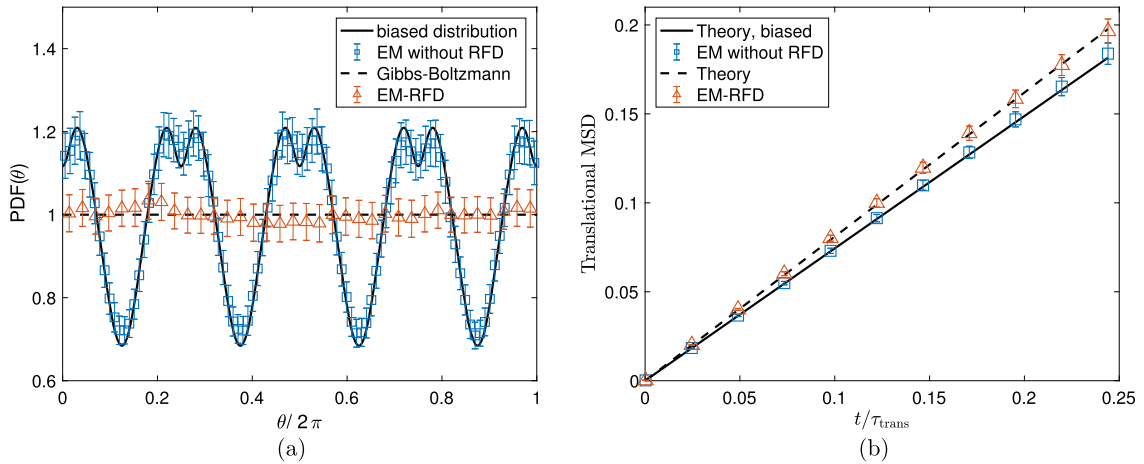


Fig. 7. (Left panel) Equilibrium probability distribution of a single starfish particle freely diffusing in a periodic square lattice. The EM scheme without RFD produces a biased equilibrium distribution, and matches the theory Eq. (4.9) (solid). The EM scheme with RFD produces the correct Gibbs–Boltzmann distribution (dashed). (Right panel) Translational mean square displacement (MSD) for the freely-diffusing starfish particle using the biased and unbiased EM schemes.

where the characteristic length of the starfish particle is its maximum radius $a = r_s(1 + b)$. In the continuum setting, the body mobility matrix of the four-fold starfish particle is a diagonal matrix and depends only on the rotation θ , i.e.,

$$\mathcal{N}(\theta) = \begin{bmatrix} \mathcal{N}_{xx}(\theta) & & \\ & \mathcal{N}_{yy}(\theta) & \\ & & \mathcal{N}_{\theta\theta}(\theta) \end{bmatrix}, \tag{4.6}$$

where \mathcal{N}_{xx} , \mathcal{N}_{yy} and $\mathcal{N}_{\theta\theta}$ are the rotational and translational self-mobilities. Because of the symmetry of the starfish particle, we also have $\mathcal{N}_{xx} = \mathcal{N}_{yy}$. The elements of $\mathcal{N}(\theta)$ can be computed to 12 digits of accuracy using the second-kind solver by solving the deterministic mobility problem with \mathbf{F} set to be the columns of the identity matrix for different θ .

Applying the EM scheme without RFD to the freely-diffusing starfish particle, we obtain the biased scheme,

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + \sqrt{2k_B T \Delta t} (\mathcal{N}^n)^{\frac{1}{2}} \mathbf{W}^n. \tag{4.7}$$

In the limit $\Delta t \rightarrow 0$, the biased scheme Eq. (4.7) is consistent with the Ito SDE

$$\begin{aligned} \frac{d\mathbf{Q}}{dt} &= \sqrt{2k_B T \Delta t} (\mathcal{N}(\theta))^{\frac{1}{2}} \mathcal{W}(t) \\ &= -\mathcal{N}(\theta) (\partial_{\mathbf{Q}} \tilde{U}) + \sqrt{2k_B T \Delta t} (\mathcal{N}(\theta))^{\frac{1}{2}} \mathcal{W}(t) + (k_B T) \partial_{\mathbf{Q}} \cdot (\mathcal{N}(\theta)), \end{aligned} \tag{4.8}$$

where the bias potential is $\tilde{U}(\mathbf{Q} = \{\mathbf{q}, \theta\}) = k_B T \log(\mathcal{N}_{\theta\theta})$. By examining the corresponding Fokker–Plank equation, we can show that the biased SDE (4.8) preserves the biased equilibrium distribution

$$\tilde{P}_{eq}(\mathbf{Q}) \equiv \tilde{P}_{eq}(\theta) = Z^{-1} \exp(-\tilde{U}(\theta)/k_B T) = (Z \mathcal{N}_{\theta\theta})^{-1}. \tag{4.9}$$

We note that the correct Gibbs–Boltzmann distribution preserved by the unbiased scheme (EM with RFD) is a uniform distribution $P_{eq}(\mathbf{Q}) = \text{constant}$.

In our computation, we set $b = 0.3$ and $a = 1.3r_s = 0.45$, which gives a relatively high packing fraction $\phi \approx 0.393$ for the starfish particle, in order to amplify the difference between the biased distribution \tilde{P}_{eq} and the correct Gibbs–Boltzmann distribution P_{eq} (see left panel of Fig. 7). The starfish particle is discretized by $N_p = 64$ quadrature points using the 4th-order Alpert quadrature. In the first-kind mobility solver, we set the error tolerance level $\epsilon = 10^{-7}$. The RFD parameter δ is set by $\delta/a \sim \epsilon^{1/3}$. The short-time translational χ_{trans} and rotational χ_{rot} diffusion coefficients are determined by the Stokes–Einstein relations,

$$\begin{aligned} \chi_{\text{trans}} &= k_B T \langle \mathcal{N}_{xx} \rangle \approx 2.47 \times 10^{-3}, \\ \chi_{\text{rot}} &= k_B T \langle \mathcal{N}_{\theta\theta} \rangle \approx 2.081 \times 10^{-1}, \end{aligned} \tag{4.10}$$

where the average $\langle \cdot \rangle$ is taken with respect to the equilibrium distribution. We use a small time step size $\Delta t = 0.02$ to minimize truncation errors, and each simulation is run for $T = 100$, where τ_{rot} is the rotational Brownian time scale.

Fig. 7 shows the estimated biased and unbiased equilibrium distributions (with error-bars of 2 standard deviations) obtained from 600 independent trajectories of the starfish particle freely diffusing in a periodic unit lattice. The numerical

results using EM without RFD indeed match the biased equilibrium distribution $\tilde{P}_{eq}(\theta)$, whereas EM with RFD preserves the correct Gibbs–Boltzmann distribution at equilibrium. In the right panel of Fig. 7, we compare the translational mean square displacement (MSD) for the biased and unbiased schemes, and also compare to the Einstein formula

$$\langle ||\mathbf{q}(t+s) - \mathbf{q}(t)||^2 \rangle = 4k_B T \langle \mathcal{N}_{xx} \rangle t, \quad (4.11)$$

where the average $\langle \cdot \rangle$ is taken with respect to the biased and unbiased equilibrium distribution, respectively. We conclude that consistent approximation to the stochastic drift term, such as the RFD approximation, is not only important for the equilibrium dynamics, but is also necessary for producing the short-time dynamics (MSD) correctly.

4.4.2. A pair of interacting starfish

In this test, we perform BD simulation of a pair of starfish particles connected by a Hookean spring with rest length l_s in a periodic square lattice of length $l = 2$ (right panel of Fig. 6). The two starfish particles interact through the spring connecting the two tracking points $\mathbf{q}_1, \mathbf{q}_2$ via a potential U_{spring} , and the rotation of each particle is also attached to a preferred angle through a harmonic potential $U_{\text{rot}}(\theta)$. The total potential $U(\mathbf{Q})$ is given by

$$\begin{aligned} U(\mathbf{Q}) &= U(\mathbf{q}_1, \theta_1, \mathbf{q}_2, \theta_2) \\ &= U_{\text{spring}}(\mathbf{q}_1, \mathbf{q}_2) + U_{\text{rot}}(\theta_1) + U_{\text{rot}}(\theta_2) \\ &= \frac{k_s}{2} (|\mathbf{q}_1 - \mathbf{q}_2| - l_s)^2 + \frac{k_\theta}{2} \left(\theta_1 - \frac{\pi}{4} \right)^2 + \frac{k_\theta}{2} \left(\theta_2 - \frac{\pi}{2} \right)^2, \end{aligned} \quad (4.12)$$

where k_s, k_θ are the stiffness coefficients. In the equilibrium, the Gibbs–Boltzmann preserved by Eq. (2.6) with $\mathbf{F} = -\partial_{\mathbf{Q}} U$ is

$$\begin{aligned} P_{eq}(\mathbf{Q}) &\propto \exp(-U(\mathbf{Q})/k_B T) \\ &\propto \exp\left(-\frac{U_{\text{spring}}(\mathbf{q}_1, \mathbf{q}_2)}{k_B T}\right) \cdot \mathcal{N}\left(\frac{\pi}{2}, \frac{k_B T}{k_\theta}\right) \cdot \mathcal{N}\left(\frac{\pi}{4}, \frac{k_B T}{k_\theta}\right), \end{aligned} \quad (4.13)$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2 .

In the computation, each particle has size $a/l = 1.3r_s/l = 1/12$, and is discretized by $N_p = 48$ quadrature points using the 4th-order Alpert quadrature. The spring rest length is $l_s = 5a$ and the stiffness coefficient k_s is set based on the criterion that 3 standard deviations of the distance $d = |\mathbf{q}_1 - \mathbf{q}_2|$ is approximately $2a$, i.e., $3\sqrt{k_B T/k_s} \approx 2a$, so that the particles very rarely overlap. The presence of harmonic springs defines the spring relaxation time scales,

$$\tau_s = \frac{1}{k_s \langle \mathcal{N}_{xx} \rangle} \quad \text{and} \quad \tau_\theta = \frac{1}{k_\theta \langle \mathcal{N}_{\theta\theta} \rangle}, \quad (4.14)$$

where \mathcal{N}_{xx} and $\mathcal{N}_{\theta\theta}$ are the translational and rotational self-mobilities of a single particle, which can be computed with high accuracy using the second-kind solver. The value of the parameters are $k_s = 81$, $k_\theta \approx 2.446$, and $\tau_s = \tau_\theta \approx 0.1189$. Recall that the dominant computational work of FBIM is to compute the mobility \mathbf{NF} (see left panel of Fig. 5). To reduce the amount of computational work, we found that solving the mobility problems in RFD with a lower tolerance level $\epsilon = 10^{-3}$ while maintaining a higher tolerance $\epsilon = 10^{-6}$ in the deterministic mobility does not introduce any observable statistical errors in the computation.

In Fig. 8, we compare errors of the mean and covariance of θ_1, θ_2 and the distance $d = |\mathbf{q}_1 - \mathbf{q}_2|$ with respect to the equilibrium statistics calculated analytically from Eq. (4.13). The numerical results are generated from 16 independent trajectories with length $T \approx 14.86$. We observe in Fig. 8 that AB2 is more accurate than EM (see the panel that shows $\text{cov}(d, d)$). We also observe that the cross covariances are statistically indistinguishable from zero for sufficiently small time step sizes, indicating that the distance between particles and their rotations are uncorrelated, as expected from the equilibrium distribution Eq. (4.13).

5. Conclusions

In this paper we presented a fluctuating boundary integral method (FBIM) for simulating the overdamped Brownian Dynamics (BD) of rigid particles of complex shape in periodic domains. To the best of our knowledge, it is the first boundary integral method that accounts for Brownian motion of nonspherical particles immersed in a viscous incompressible fluid. Its main advantages are that particles of complex shape can be directly discretized with a surface mesh, and the deterministic mobility of the particles can be computed with high accuracy by using high-order singular quadrature techniques. Importantly, the Brownian displacements of the particles are computed along the way with only a marginal increase in the overall cost, and strictly satisfy discrete fluctuation–dissipation balance. To accomplish this, instead of adding a stochastic stress tensor to the fluid equations as done in fluctuating hydrodynamics, we eliminated the fluid in the spirit of boundary integral representations. This led to a Stochastic Stokes Boundary Value Problem (SSBVP) in which we prescribed a random surface velocity distribution that has covariance proportional to the (periodic) Stokeslet. We found that using a first-kind

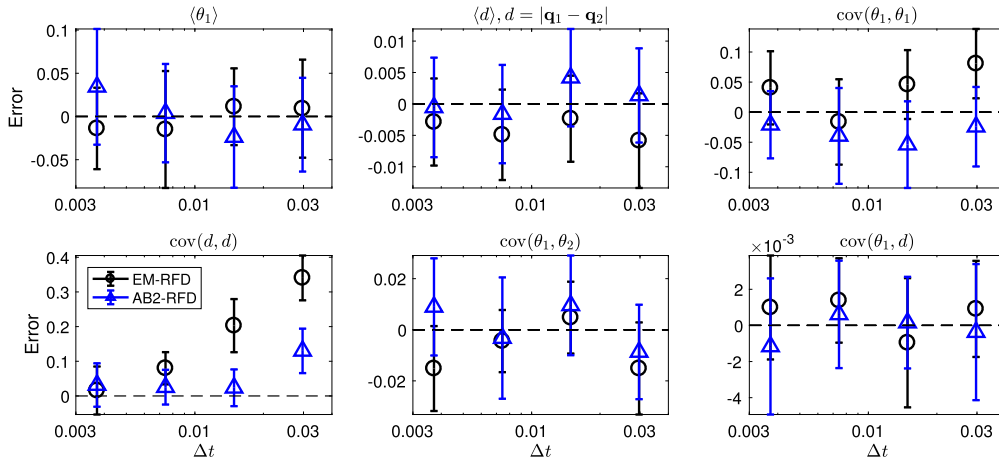


Fig. 8. Error bars (with two standard deviations) of the mean and covariance of distance between two tracking points $d = |\mathbf{q}_1 - \mathbf{q}_2|$ and rotations θ_1, θ_2 for a pair of starfish particles interacting with the potential Eq. (4.12). The numerical results are produced using the EM and AB schemes, with both including the RFD term. The AB scheme is more accurate than the EM scheme for $\text{cov}(d, d)$. The cross covariances are statistically indistinguishable from zero, indicating the particle positions and their rotations are uncorrelated, as expected from the equilibrium distribution Eq. (4.13).

boundary integral formulation is simplest since the first-kind integral operator inherits the SPD property from the Green’s function for Stokes flow. The matrix discretizing the single-layer operator directly gives the covariance of the required fluctuating surface velocity, including a suitable handling of the singularity of the Oseen tensor. While formal analysis suggests that second-kind boundary integral methods are to be preferred over first-kind methods, we found that first-kind methods can be more accurate for dense suspensions, and showed that a simple block-diagonal preconditioner can effectively handle the ill-conditioning of the first-kind formulation. We confirmed through different benchmark problems that FBIM can efficiently compute both deterministic and Brownian motion of rigid particles in accordance with the order of accuracy of the singular quadrature scheme. Our preconditioned iterative solvers (GMRES and Lanczos) converged within a small number of iterations and only grew slowly with the packing fraction. We also confirmed that the computational cost of FBIM scales linearly with the number of particles, even for moderately dense packing fractions. Finally, we coupled FBIM with stochastic temporal integrators, and showed that it reproduced the correct equilibrium Gibbs–Boltzmann distribution of Brownian suspensions of particles of complex shape.

The FBIM presented in this work is a only a first step toward the overarching goal of performing accurate, efficient and robust BD simulations of a large collection of rigid particles of complex shape. Our proof-of-concept implementation of FBIM and numerical examples were presented in two spatial dimensions only. The continuum formulation of FBIM, which generates both deterministic and stochastic velocities in agreement with Eq. (2.20), applies directly to three dimensions. In principle, our discrete formulation of FBIM can also be extended to three spatial dimensions. This extension requires, however, developing a suitable quadrature rule for the single-layer potential. While in two dimensions we were able to use the trapezoidal rule as the underlying quadrature rule and apply Alpert corrections to account for the singularity of the Oseen tensor, in three dimensions these pieces need to be developed anew. First, a suitable discretization of the particles’ surfaces (e.g., using higher-order triangular elements) and a suitable non-singular quadrature rule need to be developed. For special particles shapes, notably spheres or spheroids, one can use specially chosen surface grids with the trapezoidal rule [36,37]; however, for general particle shapes it is not straightforward to achieve spectral accuracy. One of the desired properties of the quadrature rule is to ensure that the far-field component of the discrete single-layer matrix \mathbf{M} is symmetric and positive definite. In particular, we expect that a good quadrature rule would yield the SPD matrix $\mathbf{M} = \Psi \mathbf{G} \Psi^T$, where $\mathbf{G}_{ij} = \mathbb{G}(\mathbf{x}_i - \mathbf{x}_j)$, and Ψ encodes the quadrature weights and mesh connectivity. Second, a singular quadrature near-field correction for the single-layer kernel needs to be constructed. A recently developed option is quadrature-by-expansion (QBX) [40,41]. Recently, af Klinteberg and Tornberg [37] applied QBX to simulate a collection of non-Brownian spheroids immersed in a Stokes flow; however, the generalizations to more complex particle shapes requires an underlying, high-order smooth quadrature rule.

A particular challenge in developing singular quadrature schemes is preserving underlying symmetry properties of the single layer operator. Namely, the single layer integral operator is SPD and is rotationally invariant, meaning that if the body is rotated the result of applying the operator is also rotated in the same way. The Alpert quadrature correction used in this work gives a banded high-order log-singularity correction for the near-field component of \mathbf{M} that is neither symmetric, nor positive definite, nor rotationally invariant. We found all of these artifacts to be numerical errors below the order of accuracy of the quadrature scheme, as expected. However, it would be much better to have a singular quadrature scheme that does not have such unphysical artifacts *by construction*. The traditional focus in boundary integral methods has been on achieving higher-order accuracy. What is more important for Brownian suspensions is preserving physical properties of the continuum operators in their discrete “mimetic” counterparts, so that even coarse resolutions give physically-consistent (even if not very accurate) discretizations.

We note that a Galerkin boundary integral formulation combined with multipole expansions has been used by Singh and Adhikari and collaborators to model active suspensions of spherical particles in an unbounded domain [64]. In some sense, this is an extension of the traditional Stokesian Dynamics method to include moments higher than the stresslet, as necessary to account for active flows. Recently the method has been extended to include Brownian contributions and to account for confinement near a no-slip boundary [65]. While also based on a boundary integral formulation, this class of methods differs in significant ways from the one proposed here. Most importantly, our approach uses numerical quadrature instead of analytical integration, and therefore generalizes to arbitrary (smooth) particle shapes. Furthermore, we do not truncate a multipole hierarchy at a finite number of moments, and can therefore achieve controlled accuracy (i.e., a desired number of digits of accuracy). Nevertheless, generalizations of FBIM to three dimensions require a nontrivial amount of effort, and for spherical particles the Galerkin approach may be an effective alternative that yields sufficiently accurate answers in practice.

Another key aspect of Brownian Dynamics simulations is the temporal integration schemes. There are three main challenges in this regard. The first is that in three dimensions orientation cannot be represented by a vector. This can most straightforwardly be handled by using normalized quaternions to represent particle orientation, as shown in [43]. In the end, as long as a method to compute particle velocities in agreement with Eq. (2.20) is provided, handling the quaternion constraint simply amounts to using quaternion multiplication, rather than addition, to update orientations [43,53]. The second challenge is capturing the stochastic drift term proportional to the divergence of the body mobility matrix using linear-scaling iterative methods. In this work we used methods that perform random finite difference on the body mobility. This requires solving two mobility problems per time step just to capture the stochastic drift term, therefore at least doubling the cost of a time step. In future work we will describe novel temporal integrators that can be used with FBIM to give more accurate answers for larger time step sizes, and which use only a single mobility solve to capture the stochastic drift term [53]. A third challenge is to handle the fact that Brownian displacements can lead to overlaps between the particles, even for small time step sizes. Unlike the rigid multiblob method [39], which builds on a regularized first-kind boundary integral formulation, traditional boundary integral methods based on singular quadratures break down when particles overlap. Even if particles do not overlap, unless the surface discretization is refined adaptively (which would be too costly for denser suspensions), traditional boundary integral representations will give unphysical answers that can easily lead to breakdown especially in the presence of noise. A possible solution to this is to use a regularization of the formulation for particle gaps below the resolution of the method. This is done naturally in the rigid multiblob method [53], and also more recently in a Galerkin multipole method [65], by using the Rotne–Prager–Yamakawa regularization of the Oseen tensor.

Another important direction of future work is to extend FBIM to model Brownian suspensions in other geometries, notably in unbounded domains. One option is to adapt our method to use a newly developed spectral Ewald summation for the free-space Stokeslet [66]. We note, however, that the computational cost of the FFTs in Ewald-type methods may become non-trivial for three dimensional problems. A challenge of great interest is incorporating the Fast Multipole Method (FMM) to generate the random surface velocity, thus developing a grid-free (near) linear-scaling method for Brownian suspension in an unbounded domain. For simple confined geometries, such as colloids sedimented in the vicinity of a single no-slip wall, one can use known analytical Green's functions for Stokes flow as done in [65]. It turns out that because of the more rapid decay of the Green's function in the presence of a wall, iterative methods can efficiently generate the random surface velocity without requiring special handling of the far-field interactions [44]. Nevertheless, achieving both linear scaling and controlled accuracy are challenges even in such simple geometries. For finite domains of more complicated geometry, one can discretize the domain boundary explicitly [67], and then employ the free-space Spectral Ewald method [66].

Acknowledgements

We gratefully thank Anna-Karin Tornberg for numerous informative discussions and for sharing with us notes on the Hasimoto splitting of the Stokeslet and Stresslet in two dimensions. YB and AD were supported in part by the National Science Foundation under award DMS-1418706, and by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0008271. EEK acknowledges support from EPSRC grant EP/P013651/1. LG and MR were supported in part by the Office of the Assistant Secretary of Defense for Research and Engineering and AFOSR under NSSEFF Program Award FA9550-10-1-0180 and in part by the U.S. Department of Energy under contract DEFG0288ER25053. MR was also supported in part by the Office of Naval Research under award N00014-14-1-0797/16-1-2123.

Appendix A. Body mobility matrix and Lorentz reciprocal theorem

In this appendix, we derive the following expression for elements of the body mobility matrix \mathcal{N} in terms of the periodic Green's function $\mathbb{G}(\mathbf{x}, \mathbf{y})$ of Stokes flow,

$$(\mathcal{N})_{ij} \equiv \mathcal{N}_{ij} = \int_{\Gamma} \int_{\Gamma} \boldsymbol{\lambda}^{(j)}(\mathbf{x}) \cdot \mathbb{G}(\mathbf{x} - \mathbf{y}) \cdot \boldsymbol{\lambda}^{(i)}(\mathbf{y}) \, dS_{\mathbf{y}} \, dS_{\mathbf{x}}, \quad (\text{A.1})$$

where the precise definition of $\boldsymbol{\lambda}^{(i)}, \boldsymbol{\lambda}^{(j)}$ appears later.

For simplicity, we consider only a single rigid body Γ immersed in a Stokes fluid with periodic boundary conditions. The generalization of (A.1) to account for many bodies is straightforward. First, we recall that, the mobility problem that solves for the translational velocity \mathbf{u} and the rotational velocity $\boldsymbol{\omega}$ of the body, in response to the force \mathbf{f} and torque $\boldsymbol{\tau}$ exerted on the body is described by the steady Stokes equations with no-slip boundary condition, and force and torque balance conditions:

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma} &= \nabla \pi - \eta \nabla^2 \mathbf{v} = \mathbf{0}, \\ \nabla \cdot \mathbf{v} &= 0, \\ \mathbf{v}(\mathbf{x}) &= \mathbf{u} + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{q}), \quad \forall \mathbf{x} \in \Gamma, \\ \int_{\Gamma} \boldsymbol{\lambda}(\mathbf{x}) \, dS_{\mathbf{x}} &= \mathbf{f} \quad \text{and} \quad \int_{\Gamma} (\mathbf{x} - \mathbf{q}) \times \boldsymbol{\lambda}(\mathbf{x}) \, dS_{\mathbf{x}} = \boldsymbol{\tau}, \end{aligned} \tag{A.2}$$

where η is the fluid viscosity, $\boldsymbol{\sigma}$ the fluid stress tensor, and \mathbf{v} is the fluid velocity, respectively. Here $\boldsymbol{\lambda} = (\boldsymbol{\sigma} \cdot \mathbf{n})(\mathbf{x})$ is the surface traction of the body with \mathbf{n} being the unit normal vector to the surface. The mobility problem (A.2) can be viewed as a linear mapping

$$\mathbf{U} = \mathcal{N} \mathbf{F}, \tag{A.3}$$

where \mathcal{N} is the body mobility matrix that relates the rigid body motion $\mathbf{U} = \{\mathbf{u}, \boldsymbol{\omega}\}$ to the applied force and torque $\mathbf{F} = \{\mathbf{f}, \boldsymbol{\tau}\}$.

Let $\{\mathbf{v}^{(i)}, \boldsymbol{\sigma}^{(i)}, \mathbf{u}^{(i)}, \boldsymbol{\omega}^{(i)}\}$ and $\{\mathbf{v}^{(j)}, \boldsymbol{\sigma}^{(j)}, \mathbf{u}^{(j)}, \boldsymbol{\omega}^{(j)}\}$ denote the solutions to the mobility problem (A.2) with applied force and torque $\mathbf{F} = \{\mathbf{f}^{(i)}, \boldsymbol{\tau}^{(i)}\} = \mathbf{e}^{(i)}$ and $\mathbf{F} = \{\mathbf{f}^{(j)}, \boldsymbol{\tau}^{(j)}\} = \mathbf{e}^{(j)}$ respectively. Here $\mathbf{e}^{(i)}$ and $\mathbf{e}^{(j)}$ are the standard basis vectors (in two dimensions, $\mathbf{e}^{(i)} \in \mathbb{R}^3$). It is not difficult to see that $\mathbf{U}^{(j)} = \{\mathbf{u}^{(j)}, \boldsymbol{\omega}^{(j)}\}$ corresponds to the j^{th} column of \mathcal{N} . It is understood that whenever the force or torque is made dimensionless in the canonical problem, the other quantities' dimensions are adjusted accordingly. Thus, if the force is made dimensionless, velocities have units of the force-velocity mobility, while the tractions have units of inverse area. If the torque is made dimensionless, then the velocities have units of the torque-velocity mobility and the tractions have units of inverse length.

Invoking the Lorentz Reciprocal Theorem (LRT) [12] and eliminating boundary terms arise from integration-by-parts using periodic BCs, we obtain

$$\int_{\Gamma} \mathbf{v}^{(i)} \cdot \boldsymbol{\lambda}^{(j)} \, dS = \int_{\Gamma} \mathbf{v}^{(j)} \cdot \boldsymbol{\lambda}^{(i)} \, dS. \tag{A.4}$$

After substituting the no-slip BC for $\mathbf{v}^{(j)}$ on the RHS of (A.4), and make use of the force and torque balance condition for $\boldsymbol{\lambda}^{(i)}$, we obtain that

$$\begin{aligned} \int_{\Gamma} \mathbf{v}^{(i)} \cdot \boldsymbol{\lambda}^{(j)} \, dS &= \int_{\Gamma} [\mathbf{u}^{(j)} + \boldsymbol{\omega}^{(j)} \times (\mathbf{x} - \mathbf{q})] \cdot \boldsymbol{\lambda}^{(i)} \, dS_{\mathbf{x}} \\ &= \mathbf{u}^{(j)} \cdot \mathbf{f}^{(i)} + \boldsymbol{\omega}^{(j)} \cdot \boldsymbol{\tau}^{(i)} \\ &= \mathbf{U}^{(j)} \cdot \mathbf{e}^{(i)} = \mathcal{N}_{ij}. \end{aligned} \tag{A.5}$$

We recall that $\mathbf{v}^{(i)}(\mathbf{x})$ for $\mathbf{x} \in \Gamma$ can be written as

$$\mathbf{v}^{(i)}(\mathbf{x}) = \int_{\Gamma} \mathbb{G}(\mathbf{x} - \mathbf{y}) \cdot \boldsymbol{\lambda}^{(i)}(\mathbf{y}) \, dS_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma, \tag{A.6}$$

where \mathbb{G} is the Green's function of Stokes flow with unit viscosity and periodic BCs (periodic Stokeslet), as dictated by the first-kind boundary integral formulation of the mobility problem [12]. Lastly, we substitute (A.6) for $\mathbf{v}^{(i)}$ on the LHS of (A.5) to conclude that

$$\mathcal{N}_{ij} = \int_{\Gamma} \int_{\Gamma} \boldsymbol{\lambda}^{(i)}(\mathbf{x}) \cdot \mathbb{G}(\mathbf{x} - \mathbf{y}) \cdot \boldsymbol{\lambda}^{(j)}(\mathbf{y}) \, dS_{\mathbf{y}} \, dS_{\mathbf{x}}. \tag{A.7}$$

Appendix B. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.jcp.2018.08.021>.

References

- [1] Ambarish Ghosh, Peer Fischer, Controlled propulsion of artificial magnetic nanostructured propellers, *Nano Lett.* 9 (6) (2009) 2243–2245.
- [2] C.S. Peskin, G.M. Odell, G.F. Oster, Cellular motions and thermal fluctuations: the Brownian ratchet, *Biophys. J.* 65 (1) (1993) 316–324.
- [3] H. Rafii-Tabar, R. Tavakoli-Darestani, Modelling the stochastic dynamics of biological nano-motors: an overview of recent results, *J. Comput. Theor. Nanosci.* 6 (4) (2009) 806–819.
- [4] Hong-Ren Jiang, Natsuhiko Yoshinaga, Masaki Sano, Active motion of a Janus particle by self-thermophoresis in a defocused laser beam, *Phys. Rev. Lett.* 105 (26) (2010) 268302.
- [5] Donald L. Koch, Ganesh Subramanian, Collective hydrodynamics of swimming microorganisms: living fluids, *Annu. Rev. Fluid Mech.* 43 (2011) 637–659.
- [6] Jeremie Palacci, Stefano Sacanna, Asher Preska Steinberg, David J. Pine, Paul M. Chaikin, Living crystals of light-activated colloidal surfers, *Science* 339 (6122) (2013) 936–940.
- [7] Walter F. Paxton, Kevin C. Kistler, Christine C. Olmeda, Ayusman Sen, Sarah K. St. Angelo, Yanyan Cao, Thomas E. Mallouk, Paul E. Lammert, Vincent H. Crespi, Catalytic nanomotors: autonomous movement of striped nanorods, *J. Am. Chem. Soc.* 126 (41) (2004) 13424–13431.
- [8] Daisuke Takagi, Adam B. Braunschweig, Jun Zhang, Michael J. Shelley, Dispersion of self-propelled rods undergoing fluctuation-driven flips, *Phys. Rev. Lett.* 110 (3) (2013) 038301.
- [9] Jonathan R. Howse, Richard A.L. Jones, Anthony J. Ryan, Tim Gough, Reza Vafabakhsh, Ramin Golestanian, Self-motile colloidal particles: from directed propulsion to random walk, *Phys. Rev. Lett.* 99 (4) (2007) 48102.
- [10] Stephen J. Ebbens, Jonathan R. Howse, In pursuit of propulsion at the nanoscale, *Soft Matter* 6 (4) (2010) 726–738.
- [11] Sangtae Kim, Seppo J. Kärtilä, *Microhydrodynamics: Principles and Selected Applications*, Courier Corporation, 2013.
- [12] Constantine Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow*, Cambridge University Press, 1992.
- [13] R.M. Jendrejack, J.J. de Pablo, M.D. Graham, Stochastic simulations of DNA in flow: dynamics and the effects of hydrodynamic interactions, *J. Chem. Phys.* 116 (17) (2002) 7752–7759.
- [14] Richard M. Jendrejack, David C. Schwartz, Michael D. Graham, Juan J. de Pablo, Effect of confinement on DNA dynamics in microfluidic devices, *J. Chem. Phys.* 119 (2003) 1165.
- [15] J.P. Hernandez-Ortiz, J.J. de Pablo, M.D. Graham, Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry, *Phys. Rev. Lett.* 98 (14) (2007) 140602.
- [16] Yu Zhang, Juan J. de Pablo, Michael D. Graham, An immersed boundary method for Brownian dynamics simulation of polymers in complex geometries: application to dna flowing through a nanoslit with embedded nanopits, *J. Chem. Phys.* 136 (2012) 014901.
- [17] Zhi Liang, Zydrunas Gimbutas, Leslie Greengard, Jingfang Huang, Shidong Jiang, A fast multipole method for the Rotne–Prager–Yamakawa tensor and its applications, *J. Comput. Phys.* 234 (2013) 133–139.
- [18] A. Sierou, J.F. Brady, Accelerated Stokesian dynamics simulations, *J. Fluid Mech.* 448 (2001) 115–146.
- [19] Adolfo J. Banchio, John F. Brady, Accelerated Stokesian dynamics: Brownian motion, *J. Chem. Phys.* 118 (2003) 10323.
- [20] Mu Wang, John F. Brady, Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions, *J. Comput. Phys.* 306 (2016) 443–477.
- [21] Dag Lindbo, Anna-Karin Tornberg, Spectrally accurate fast summation for periodic Stokes potentials, *J. Comput. Phys.* 229 (23) (2010) 8994–9010.
- [22] Marshall Fixman, Construction of Langevin forces in the simulation of hydrodynamic interaction, *Macromolecules* 19 (4) (1986) 1204–1207.
- [23] Tadashi Ando, Edmond Chow, Yousef Saad, Jeffrey Skolnick, Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations, *J. Chem. Phys.* 137 (6) (2012).
- [24] A.M. Fiore, F. Balboa Usabiaga, A. Donev, J.W. Swan, Rapid sampling of stochastic displacements in Brownian dynamics simulations, *J. Chem. Phys.* 146 (12) (2017) 124116, software available at <https://github.com/stochasticHydroTools/PSE>.
- [25] R. Adhikari, K. Stratford, M.E. Cates, A.J. Wagner, Fluctuating lattice Boltzmann, *Europhys. Lett.* 71 (2005) 473–479.
- [26] B. Dünweg, A.J.C. Ladd, Lattice Boltzmann simulations of soft matter systems, in: *Adv. Comp. Sim. for Soft Matter Sciences III*, 2009, pp. 89–166.
- [27] S. Delong, F. Balboa Usabiaga, R. Delgado-Buscailoni, B.E. Griffith, A. Donev, Brownian dynamics without Green’s functions, *J. Chem. Phys.* 140 (13) (2014) 134110, software available at <https://github.com/stochasticHydroTools/FIB>.
- [28] Eric E. Keaveny, Fluctuating force-coupling method for simulations of colloidal suspensions, *J. Comput. Phys.* 269 (2014) 61–79.
- [29] Blaise Delmotte, Eric E. Keaveny, Simulating Brownian suspensions with fluctuating hydrodynamics, *J. Chem. Phys.* 143 (24) (2015) 244109.
- [30] Jens Rotne, Stephen Prager, Variational treatment of hydrodynamic interaction in polymers, *J. Chem. Phys.* 50 (1969) 4831.
- [31] H. Hasimoto, On the periodic fundamental solutions of the stokes equations and their application to viscous flow past a cubic array of spheres, *J. Fluid Mech.* 5 (02) (1959) 317–328.
- [32] D. Liu, E.E. Keaveny, Martin R. Maxey, George E. Karniadakis, Force-coupling method for flows with ellipsoidal particles, *J. Comput. Phys.* 228 (10) (2009) 3559–3581.
- [33] Anna-Karin Tornberg, Leslie Greengard, A fast multipole method for the three-dimensional stokes equations, *J. Comput. Phys.* 227 (3) (2008) 1613–1619.
- [34] Rachh Manas, Leslie Greengard, Integral equation methods for elastance and mobility problems in two dimensions, *SIAM J. Numer. Anal.* 54 (5) (2016) 2889–2909.
- [35] Eduardo Corona, Leslie Greengard, Manas Rachh, Shravan Veerapaneni, An integral equation formulation for rigid bodies in stokes flow in three dimensions, *J. Comput. Phys.* 332 (2017) 504–519.
- [36] Ludvig Af Klinteberg, Anna-Karin Tornberg, Fast Ewald summation for Stokesian particle suspensions, *Int. J. Numer. Methods Fluids* 76 (10) (2014) 669–698.
- [37] Ludvig Af Klinteberg, Anna-Karin Tornberg, A fast integral equation method for solid particles in viscous flow using quadrature by expansion, *J. Comput. Phys.* 326 (2016) 420–445.
- [38] David J. Smith, A boundary element regularized stokeslet method applied to cilia-and flagella-driven flow, *Proc. R. Soc. Lond. Ser. A* 465 (2112) (2009) 3605–3626.
- [39] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A.P.S. Bhalla, B.E. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, *Commun. Appl. Math. Comput. Sci.* 11 (2) (2016) 217–296, software available at <https://github.com/stochasticHydroTools/RigidMultiblobsWall>.
- [40] Charles L. Epstein, Leslie Greengard, Andreas Klöckner, On the convergence of local expansions of layer potentials, *SIAM J. Numer. Anal.* 51 (5) (2013) 2660–2679.
- [41] Andreas Klöckner, Alexander Barnett, Leslie Greengard, Michael O’Neil, Quadrature by expansion: a new method for the evaluation of layer potentials, *J. Comput. Phys.* 252 (2013) 332–349.
- [42] Eric E. Keaveny, Michael J. Shelley, Applying a second-kind boundary integral equation for surface tractions in Stokes flow, *J. Comput. Phys.* 230 (5) (2011) 2141–2159.
- [43] S. Delong, F. Balboa Usabiaga, A. Donev, Brownian dynamics of confined rigid bodies, *J. Chem. Phys.* 143 (14) (2015) 144107, software available at <https://github.com/stochasticHydroTools/RigidMultiblobsWall>.
- [44] Florencio Balboa Usabiaga, Blaise Delmotte, Aleksandar Donev, Brownian dynamics of confined suspensions of active microrollers, *J. Chem. Phys.* 146 (13) (2017) 134104, software available at <https://github.com/stochasticHydroTools/RigidMultiblobsWall>.

- [45] R.F. Fox, G.E. Uhlenbeck, Contributions to non-equilibrium thermodynamics. I. Theory of hydrodynamical fluctuations, *Phys. Fluids* 13 (1970) 1893.
- [46] E.J. Hinch, Application of the Langevin equation to fluid suspensions, *J. Fluid Mech.* 72 (03) (1975) 499–511.
- [47] J.N. Roux, Brownian particles at different times scales: a new derivation of the Smoluchowski equation, *Physica A* 188 (1992) 526–552.
- [48] E.H. Hauge, A. Martin-Lof, Fluctuating hydrodynamics and Brownian motion, *J. Stat. Phys.* 7 (3) (1973) 259–281.
- [49] R. Zwanzig, M. Bixon, Compressibility effects in the hydrodynamic theory of Brownian motion, *J. Fluid Mech.* 69 (1975) 21–25.
- [50] Grigorios A. Pavliotis, Andrew M. Stuart, *Multiscale Methods: Averaging and Homogenization*, vol. 53, Springer, 2008.
- [51] M. Fixman, Simulation of polymer dynamics. I. General theory, *J. Chem. Phys.* 69 (1978) 1527.
- [52] S. Delong, Y. Sun, B.E. Griffith, E. Vanden-Eijnden, A. Donev, Multiscale temporal integrators for fluctuating hydrodynamics, *Phys. Rev. E* 90 (2014) 063312, software available at <https://github.com/stochasticHydroTools/MixingIBAMR>.
- [53] Brennan Sprinkle, Florencio Balboa Usabiaga, Neelesh A. Patankar, Aleksandar Donev, Large scale Brownian dynamics of confined suspensions of rigid particles, *J. Chem. Phys.* 147 (24) (2017) 244103, software available at <https://github.com/stochasticHydroTools/RigidMultiblobsWall>.
- [54] Bradley K. Alpert, Hybrid Gauss-trapezoidal quadrature rules, *SIAM J. Sci. Comput.* 20 (5) (1999) 1551–1584.
- [55] J. Thomas Beale, Wenjun Ying, Jason R. Wilson, A simple method for computing singular or nearly singular integrals on closed surfaces, *Commun. Comput. Phys.* 20 (3) (2016) 733–753.
- [56] James Bremer, Zdrunas Gimbutas, A Nyström method for weakly singular integral operators on surfaces, *J. Comput. Phys.* 231 (14) (2012) 4885–4903.
- [57] P. Kolm, V. Rokhlin, Numerical quadratures for singular and hypersingular integrals, *Comput. Math. Appl.* 41 (3) (2001) 327–352.
- [58] M.P. Allen, D.J. Tildesley, *Computer Simulations of Liquids*, Oxford Science Publications, 1987.
- [59] L. Greengard, J. Lee, Accelerating the nonuniform fast Fourier transform, *SIAM Rev.* 46 (3) (2004) 443–454.
- [60] Dag Lindbo, Anna-Karin Tornberg, Spectral accuracy in fast Ewald-based methods for particle simulations, *J. Comput. Phys.* 230 (24) (2011) 8744–8761.
- [61] Edmond Chow, Yousef Saad, Preconditioned Krylov subspace methods for sampling multivariate gaussian distributions, *SIAM J. Sci. Comput.* 36 (2) (2014) A588–A608.
- [62] A.S. Sangani, A. Acrivos, Slow flow through a periodic array of spheres, *Int. J. Multiph. Flow* 8 (4) (1982) 343–360.
- [63] A.S. Sangani, A. Acrivos, Slow flow past periodic arrays of cylinders with application to heat transfer, *Int. J. Multiph. Flow* 8 (3) (1982) 193–206.
- [64] Rajesh Singh, Somdeb Ghose, R. Adhikari, Many-body microhydrodynamics of colloidal particles with active boundary layers, *J. Stat. Mech. Theory Exp.* 2015 (6) (2015) P06017.
- [65] Rajesh Singh, R. Adhikari, Fluctuating hydrodynamics and the Brownian motion of an active colloid near a wall, *Eur. J. Comput. Mech.* 26 (1–2) (2017) 78–97.
- [66] Ludvig af Klinteberg, Davoud Saffar Shamshirgar, Anna-Karin Tornberg, Fast Ewald summation for free-space Stokes potentials, *Res. Math. Sci.* 4 (1) (2017) 1.
- [67] Oana Marin, Katarina Gustavsson, Anna-Karin Tornberg, A highly accurate boundary treatment for confined stokes flow, *Comput. Fluids* 66 (2012) 215–230.