J. Fluid Mech. (2019), *vol.* 878, *pp.* 544–597. © Cambridge University Press 2019
doi:10.1017/jfm.2019.640

Fast Stokesian dynamics

Andrew M. Fiore[1] **and James W. Swan**[1,†]

[1]Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

(Received 21 February 2019; revised 31 July 2019; accepted 1 August 2019; first published online 17 September 2019)

We present a new method for large scale dynamic simulation of colloidal particles with hydrodynamic interactions and Brownian forces, which we call fast Stokesian dynamics (FSD). The approach for modelling the hydrodynamic interactions between particles is based on the Stokesian dynamics (SD) algorithm (*J. Fluid Mech.*, vol. 448, 2001, pp. 115–146), which decomposes the interactions into near-field (short-ranged, pairwise additive and diverging) and far-field (long-ranged many-body) contributions. In FSD, the standard system of linear equations for SD is reformulated using a single saddle point matrix. We show that this reformulation is generalizable to a host of particular simulation methods enabling the self-consistent inclusion of a wide range of constraints, geometries and physics in the SD simulation scheme. Importantly for fast, large scale simulations, we show that the saddle point equation is solved very efficiently by iterative methods for which novel preconditioners are derived. In contrast to existing approaches to accelerating SD algorithms, the FSD algorithm avoids explicit inversion of ill-conditioned hydrodynamic operators without adequate preconditioning, which drastically reduces computation time. Furthermore, the FSD formulation is combined with advanced sampling techniques in order to rapidly generate the stochastic forces required for Brownian motion. Specifically, we adopt the standard approach of decomposing the stochastic forces into near-field and far-field parts. The near-field Brownian force is readily computed using an iterative Krylov subspace method, for which a novel preconditioner is developed, while the far-field Brownian force is efficiently computed by linearly transforming those forces into a fluctuating velocity field, computed easily using the positively split Ewald approach (*J. Chem. Phys.*, vol. 146, 2017, 124116). The resultant effect of this field on the particle motion is determined through solution of a system of linear equations using the same saddle point matrix used for deterministic calculations. Thus, this calculation is also very efficient. Additionally, application of the saddle point formulation to develop high-resolution hydrodynamic models from constrained collections of particles (similar to the immersed boundary method) is demonstrated and the convergence of such models is discussed in detail. Finally, an optimized graphics processing unit implementation of FSD for mono-disperse spherical particles is used to demonstrated performance and accuracy of dynamic simulations of $O(10^5)$ particles, and an open source plugin for the HOOMD-blue suite of molecular dynamics software is included in the supplementary material.

Key words: Stokesian dynamics

† Email address for correspondence: jswan@mit.edu

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

1. Introduction

The dynamics of colloidal particles immersed in a fluid is moderated by the hydrodynamic interactions (HI) between the immersed bodies and driven in part by the fluctuating Brownian forces exerted on the particles due to collisions with individual solvent molecules, which are coupled to the HI through the fluctuation dissipation theorem (Kubo 1966). Fluid flows at the colloidal scale are dominated by viscous effects because the inertial relaxation time is much shorter than the viscous relaxation time, i.e. the Reynolds number is much less than one, so modelling dispersions via dynamic simulations requires generating efficient solutions to the Stokes equations subject to the no-slip boundary condition on the surfaces of all the immersed particles. A variety of methods, summarized in the next few paragraphs, have been developed for solving the Stokes equations around particles. These solution methods coupled with the Langevin equation, which describes the time evolution of particle positions, provide a sufficient description of particle motion to integrate particle coordinates in dynamic simulations.

The oldest and most established class of approximations applied to spherical particles are called implicit solvent methods. In these methods, the hydrodynamic mobility tensor, which is a linear transformation that describes how forces on one particle induce fluid flows that entrain another particle, is directly computed. The flow field produced by a given particle is approximated using the Green's function for Stokes flow and a multipole expansion of the particle surface force density. In this way, the non-uniform hydrodynamic traction on each particle's surface is represented as the sum of a force, torque, stresslet, etc., which sum together to propagate fluid flow (Kim & Karrila 2005). The motion of other particles entrained by this flow field is inferred via Faxén's laws, which define how particles translate, rotate and deform in response to ambient flows.

In their seminal paper on Brownian dynamics (BD) simulations with HI, Ermak & McCammon (1978) used the Rotne–Prager–Yamawkawa (RPY) mobility (diffusion) tensor (Rotne & Prager 1969) to describe the coupling of particle motions in the underlying Langevin equation. The RPY tensor approximates the HI with the first-order terms in the multipole expansion of the force density on the surface of spherical particles (force monopole and isotropic quadrupole) summed pairwise over all particles. Three-body and higher-order multibody effects are neglected in the RPY tensor, but it is nevertheless an accurate description of the long-ranged coordination in colloidal dispersions. Higher-order approximations follow directly by extending the multipole expansion to arbitrary order. However, the hydrodynamic scattering series associated with the multipole expansion is slowly converging, especially for nearly touching particles, for which an infinite number of terms would be required to resolve the lubrication interactions (Ladd 1990). The Stokesian dynamics (SD) methodology (Brady & Bossis 1988) overcomes this problem by truncating the multipole expansion after the dipole (also retaining the isotropic components of the quadrupole and octupole), then closing the series with the exact expression for the hydrodynamic resistance between nearly touching particles (Jeffrey & Onishi 1984; Jeffrey 1992*b*). This scheme accurately reproduces the transport properties of colloidal dispersions across a wide range of volume fractions and Péclet numbers when lubrication flows dominate. For some transport properties, including sedimentation, higher-order multipoles may be necessary. Ladd (1990) and Cichocki *et al.* (1994) discuss approaches to account for these additional multipoles. The original implementation of the SD algorithm explicitly built the HI tensors, requiring $O(N^2)$ calculations to build and $O(N^3)$ to invert the matrices in a simulation of

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

N particles. The far-field HI stemming from the multipole expansion are slowly decaying and constitute the primary challenge in developing efficient simulation methods. The accelerated Stokesian dynamics (ASD) algorithm was developed by Sierou & Brady (2001) address this problem. They applied the Ewald sum splitting of Hasimoto (1959) and the particle-mesh-Ewald (PME) algorithm (Darden, York & Pedersen 1993) to accelerate the far-field calculation, reducing the computational cost to $O(N \log N)$. Wang & Brady (2016) later replaced the PME acceleration of the wave-space sum in ASD with spectral Ewald (SE) acceleration (Lindbo & Tornberg 2010). The non-uniform fast Fourier transform (NUFFT) employed in the SE technique is spectrally accurate with independent and decoupled errors, and represents a substantial improvement in efficiency and error control over PME, although with no better asymptotic scaling.

Even with these accelerated techniques to compute the HI, the fluctuations associated with BD are challenging compute because they must be sampled from a distribution whose covariance is proportional to the hydrodynamic resistance tensor. This sampling requires computing the square root of the hydrodynamic resistance tensor, which is computationally expensive. In the original SD approach, direct Cholesky decomposition was used to compute the square root, requiring $O(N^3)$ computations. Banchio & Brady (2003) applied the Chebyshev polynomial approximation of Fixman (1986) to compute the far-field Brownian forces, reducing the computational cost to $O(N^{1.25} \log N)$, which is still super-linear, limiting deployment in dynamic simulations to modest numbers of particles, typically $O(10^2)$. We recently published the positively split Ewald (PSE) method for rapidly sampling stochastic displacements in BD simulations with the RPY tensor, which evaluates the Brownian displacements with $O(N)$ computations (Fiore *et al.* 2017). We extended this method to the dipole level of the multipole expansion (Fiore & Swan 2018), which is equivalent to SD without lubrication. In this work, we develop a method incorporating the PSE sampling technique to achieve an $O(N)$ algorithm for SD with Brownian fluctuations.

In contrast to implicit solvent methods, explicit solvent techniques approximately solve the Navier–Stokes equations, rather than the Stokes equations, by representing the fluid as a collection of momentum conserving particles with prescribed rules for advection and collision with other fluid particles and with immersed colloidal particles. The lattice Boltzmann (Ladd, Colvin & Frenkel 1988), dissipative particle dynamics (Hoogerbrugge & Koelman 1992) and multi-particle collision dynamics (Gompper *et al.* 2009) methods belong to this class of approximations. The overdamped behaviour characteristic of Stokes flow is then obtained by setting a suitably short time scale for vorticity diffusion and then integrating the fluid and particle dynamics on time scales that are short relative to the inertial relaxation time scale. Notably, Brownian forces are straightforward to include by introducing random forces on the fluid particles. Although these techniques nominally require $O(N)$ calculations per time step to simulate N colloidal particles in a box of volume L^3, $L^2/\nu \sim N^{2/3}$ time steps are required for vorticity to diffuse quickly throughout the system volume, where ν is the kinematic viscosity of the fluid, resulting in $O(N^{5/3})$ calculations if the vorticity diffusion rate is to be controlled at all length scales, as has been previously discussed (Fiore *et al.* 2017). To accurately model Stokes flow, for which vorticity diffusion is instantaneous, the algorithmic scaling of explicit solvent techniques is clearly undesirable. Furthermore, the true computational cost is actually $O(N + N_s)$, where N_s is the total number of momentum-carrying solvent particles in the simulation volume. In many cases, particularly in dilute systems where more fluid particles are

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

required to fill the void space between the immersed colloids, $N_s \gg N$ and the cost of evaluating the solvent-solving collisions can dominate the computational cost. Additionally, and importantly for the present work, the accuracy of the approximation used in these approaches is hard to systematically improve. For example it is clear that smaller fluid particles are required to more accurately resolve the flows between nearly touching particles and dense suspensions, but rigorous bounds on the error incurred by this approximation are challenging to impose.

Grid-based approaches such as the fluctuating immersed boundary method (FIB) (Delong *et al.* 2014) and the force coupling method (FCM) (Lomholt & Maxey 2003; Keaveny 2014; Delmotte & Keaveny 2015) explicitly solve the continuum Stokes or Navier–Stokes equations throughout the simulation volume. These approaches use kernel, or envelope, functions to project particle forces onto a grid on which standard partial differential equation (PDE) solvers (finite difference, finite volume, spectral methods, etc.) can be used to compute the fluid response. Solving the Stokes equations in this way does not require the Green's function, which is particularly advantageous for simulations with arbitrary boundaries. Adding Brownian forces is accomplished by introducing a fluctuating stress on the fluid grid. A shortcoming of these approaches is that, while the far-field hydrodynamic interaction is effectively RPY-like, the near-field or short-range hydrodynamic interactions depend on the specific kernel used to transfer data between the particles and the grid, and are therefore difficult to control independently. Yeo & Maxey (2010) showed how to incorporate more accurate expressions for the HI between nearly touching particles in the spirit of Stokesian dynamics, but it was unclear how these lubrication corrections could be combined with Brownian fluctuations in the context of that work.

Here we report the extension of the PSE method to include near-field lubrication corrections. This extension includes reformulating Stokesian dynamics itself as a saddle point problem, for which a highly effective constraint preconditioner is developed and applied. In this formulation, the Brownian stress is decomposed into a far-field Brownian slip velocity, computed using the PSE approach (Fiore & Swan 2018), and a near-field Brownian force, computed using a fast preconditioned Krylov subspace approximation (Chow & Saad 2014). We show that this approach maintains the previously reported $O(N)$ scaling, and discuss algorithmic changes and optimization for graphics processing unit (GPU) implementation. This framework, while constructed for spherical particles, is generic and readily applied to many SD-like simulations of colloidal materials.

This paper is structured as follows. In § 2 we summarize Stokesian dynamics and present our reformulation of the equations for particle motion. In § 3 we discuss approximations for the hydrodynamic interactions between immersed particles. In § 4 we present fast methods to sample the near-field and far-field Brownian variables, as well as the Brownian drift. We describe the fast Stokesian dynamics (FSD) algorithm and give performance and validation results in § 5, with implementation details presented in appendix B. Finally, we present an application of the FSD method to simulations of particles with arbitrary shape in § 6, and discuss how to systematically improve the hydrodynamic approximation.

As the work in the following sections weaves together a number of existing results, extensions thereof and entirely new work into a cohesive framework, before commencing with the main discussion, it is worthwhile to pause and distinguish the upcoming information. The methods to be presented comprise five separate calculations:

(i) calculation of the far-field hydrodynamic interactions;
(ii) calculation of the near-field hydrodynamic interactions;

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

(iii) sampling of the stochastic far-field Brownian increments;
(iv) sampling of the stochastic near-field Brownian increments;
(v) solution of the hydrodynamic resistance problem coupling the far-field and near-field interactions.

The decision to decompose the hydrodynamic interactions into near-field and far-field contributions is based on the Stokesian dynamics approach (Brady & Bossis 1988). The far-field hydrodynamic interactions are computed using a mobility tensor constructed via multipole expansion, which is a well-known approach with an extensive history of use in the literature (see e.g. work by Rotne & Prager (1969), Ladd (1990), Cichocki *et al.* (1994), Ichiki (2002), Kim & Karrila (2005) and Fiore & Swan (2018), among others). Although we will discuss the multipole expansion specifically, the framework developed herein admits use of other methods for computing the hydrodynamic mobility, e.g. the approaches developed by Delong *et al.* (2014) and Delmotte & Keaveny (2015).

Following the development of Stokesian dynamics and the ensuing improvements (Brady & Bossis 1988; Sierou & Brady 2001; Wang & Brady 2016), the near-field hydrodynamic interactions are approximated using the exact two-body results of Jeffrey & Onishi (1984) and Jeffrey (1992*b*) modified to account for the interaction contribution already included in the far-field description. As with the far-field calculations, however, the framework presented here permits a wide range of near-field interaction descriptions.

Similar to the deterministic far-field interactions, the mobility perspective is used to compute the far-field contribution to the Brownian motion of the particles. Our previously published PSE method (Fiore *et al.* 2017; Fiore & Swan 2018), which borrows ideas from fluctuating hydrodynamics (Delong *et al.* 2014; Keaveny 2014), is used to efficiently generate these stochastic velocities. The near-field contribution to Brownian is treated from the resistance perspective, resulting in stochastic forces which are sampled using a Krylov subspace method and a novel preconditioner. While the idea of accelerating stochastic sampling problems through preconditioning is not novel (see e.g. Chow & Saad 2014), this particular approach and the preconditioners we present are new.

Finally, assembling the various far-field and near-field calculations into a cohesive framework for SD-like simulations that leverages the strengths of each approach is a challenge. Conventional SD approaches formulate the entire problem in the resistance perspective, requiring costly inversions of the far-field mobility (Brady & Bossis 1988; Sierou & Brady 2001; Wang & Brady 2016). Furthermore, while fast approaches exist to compute the far-field increments in the form of velocities, conventional SD requires that these increments be in the form of forces (Banchio & Brady 2003). To solve the former issue, we cast the deterministic hydrodynamic problem as a saddle point matrix, unifying the far-field mobility and near-field resistance problems. Saddle point matrices find extensive use in science and engineering (Benzi, Golub & Liesen 2005) and have been previously applied in the context of colloidal hydrodynamics for pure mobility problems (Balboa Usabiaga *et al.* 2016; Sprinkle *et al.* 2017) and pure resistance problems (Meng & Higdon 2008*a*), but have not been used before in mixed mobility and resistance formulations of colloidal hydrodynamics, as we do here. The Brownian sampling issue is addressed by computing a far-field slip velocity, as previously done by ourselves and others in the context of pure mobility problems (Delmotte & Keaveny 2015; Balboa Usabiaga *et al.* 2016; Sprinkle *et al.* 2017; Fiore & Swan 2018). This is the first application of this technique to Stokesian dynamics,

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where it is enables a mixed mobility–resistance formulation of the hydrodynamic interactions. The new methods we have developed and the particular combination of existing techniques with which we have integrated them collectively represent a formulation for Stokesian dynamics with Brownian motion that for the first time maintains an overall linear scaling in computational complexity with respect to the number of particles modelled.

2. Stokesian dynamics

As particles move through a viscous fluid, they propagate flows that entrain other particles. Hydrodynamic forces couple together the trajectories of these particles, creating a complex dynamical scenario. At the colloidal scale, inertia can be neglected and these flows are governed by the Stokes equations. In an infinite, dilute dispersion, the strength of these disturbance flows decays to leading order with the inverse of the distance from the particle that propagated the flow. The flow propagated by the whole dispersion is in principle divergent, but can be regularized by considering principles of mass conservation within the whole fluid–solid system. A consequence of this regularization is that the motion of all the particles in the dispersion is coupled. The coupling is linear because the Stokes equations are linear, and is described using either the mobility tensor $\boldsymbol{M}_{UF}$, which maps the hydrodynamic forces on particles $\boldsymbol{F}^H$ to the resulting velocity $\boldsymbol{U} = -\boldsymbol{M}_{UF} \cdot \boldsymbol{F}^H$, or its inverse, the resistance tensor $\boldsymbol{R}_{FU} = \boldsymbol{M}_{UF}^{-1}$, which maps particle velocities to the resulting hydrodynamic drag force $\boldsymbol{F}^H = -\boldsymbol{R}_{FU} \cdot \boldsymbol{U}$. The discrete particle displacement equation used for our dynamic simulations is

$$\boldsymbol{U} - \boldsymbol{U}^\infty = \boldsymbol{R}_{FU}^{-1} \cdot (\boldsymbol{F}^P + \boldsymbol{R}_{FE} : \boldsymbol{E}^\infty) + \boldsymbol{F}^B + k_B T \nabla \cdot \boldsymbol{R}_{FU}^{-1}, \tag{2.1}$$

which can be derived as the linearly accurate solution to the inertial N-body Langevin equation in the over-damped limit (Hinch 1975; Roux 1992). In (2.1), $\boldsymbol{U} = \Delta \boldsymbol{x}/\Delta t$ is the set of particle velocities, where $\Delta \boldsymbol{x}$ is the set of particle displacements over the discrete time step Δt, $\boldsymbol{F}^H$ is the hydrodynamic force experienced by the particles, $\boldsymbol{F}^B$ is the stochastic Brownian force, given by $\sqrt{2k_BT/\Delta t}\,\boldsymbol{R}_{FU}^{1/2} \cdot \boldsymbol{\Psi}$ in accordance with the fluctuation–dissipation theorem, where $\boldsymbol{\Psi}$ is a vector of independent standard Gaussian random variables, and $\boldsymbol{F}^P$ is the set of forces applied on particles due to e.g. external fields or interparticle potentials. Here, we consider the possibility of a linear background flow field characterized by position-dependent velocity $\boldsymbol{U}^\infty$ and constant rate of strain $\boldsymbol{E}^\infty$. The right-hand side of (2.1) evaluates the translational and rotational velocities of the particles relative to the background flow. The quantity $\boldsymbol{R}_{FE} : \boldsymbol{E}^\infty$ is the additional hydrodynamic force experienced by the particles due to the imposed flow. The last term, $\nabla \cdot \boldsymbol{R}_{FU}^{-1}$ is commonly called the Brownian drift. Physically, it serves as an additional flux that drives particles away from low mobility configurations, ensuring they do not accumulate there. Its presence is required to ensure stationarity under the Gibbs–Boltzmann distribution and generate particle configurations with the correct statistics at equilibrium. We have previously shown that neglecting the drift introduces spurious long-range fluctuations in particle density, observed as a depression in the structure factor at small wavenumbers for non-interacting particles and an enhancement in the small wavenumber structure factor for hard sphere particles (Fiore & Swan 2018).

With a suitable approximation for $\boldsymbol{R}_{FU}$ and $\boldsymbol{R}_{FE}$, equation (2.1) specifies the velocity of a collection of particles used to integrate particle trajectories in dynamic

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

simulations of colloidal dispersions. In Stokesian dynamics, $\boldsymbol{R}_{FU}$ is constructed by combining a far-field approximation for the mobility $\mathcal{M}$, which is accurate for relatively large interparticle separations, with specific near-field expressions for the interactions between nearly touching particles $\boldsymbol{R}_{FU}^{nf}$,

$$\boldsymbol{R}_{FU} = \mathcal{B}^{\mathrm{T}} \cdot \mathcal{M}^{-1} \cdot \mathcal{B} + \boldsymbol{R}_{FU}^{nf}. \tag{2.2}$$

For rigid particles, the matrix $\mathcal{B}$ maps the allowable set of particle motions, e.g. translation and rotation, onto the (potentially) higher-order set of local motions and deformations described in the mobility coupling $\mathcal{M}$, and its adjoint $\mathcal{B}^{\mathrm{T}}$ projects the forces associated with the allowed motions, e.g. linear force and torque, out of the general moments of the hydrodynamic force density (traction) considered in $\mathcal{M}$. Examples of the projections due to $\mathcal{B}$ and $\mathcal{B}^{\mathrm{T}}$ are as follows:

$$\begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{0} \end{bmatrix} = \mathcal{B} \cdot \boldsymbol{U}, \quad \boldsymbol{F} = \mathcal{B}^{\mathrm{T}} \cdot \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{S} \end{bmatrix}. \tag{2.3a,b}$$

In the traditional SD formulation, $\mathcal{M}$ is constructed from the multipole expansion of the particle surface force density truncated after the dipole, and $\boldsymbol{R}^{nf}$ is built pairwise from analytical results developed by Jeffrey & Onishi (1984) for the lubrication interactions between a pair of particles less a far-field corrective term (Brady & Bossis 1988). In this case, $\mathcal{B}$ corresponds to the stresslet constraint,

$$\mathcal{M} \cdot \mathcal{F} + \mathcal{B} \cdot \boldsymbol{U} = \boldsymbol{0}, \tag{2.4}$$

where $\mathcal{F}$ is the set of hydrodynamic force moments, including the force, torque and stresslet, as we have discussed previously (Fiore & Swan 2018). However, equation (2.2) is not limited to these approximations. The far-field multipole expansion could be truncated earlier, after the monopole and irreducible quadrupole, producing Rotne–Prager interactions (Rotne & Prager 1969) in the far field, or it could be extended to an arbitrary number of terms, as shown by Ichiki (2002). Other approaches to solving the Stokes equations for a collection of immersed particles, such as iterative multigrid solvers (Griffith 2009) or the fast multipole method (Tornberg & Greengard 2008) could alternatively be used to compute $\mathcal{M}$. This formulation is also compatible with composite bead immersed boundary methods (Swan & Wang 2016). Similarly, while $\boldsymbol{R}^{nf}$ typically represents the additional dissipation due to lubricating flows, it could instead be constructed to represent frictional contacts between particles, an approximation for the near-field interactions between immersed boundary composite bead structures, or any other short-ranged dissipative interaction one wished to model, provided it were linear in the particle velocities. Regardless of the approximation, $\mathcal{M}$ is a dense matrix containing long-range interactions, and $\boldsymbol{R}^{nf}$ is a sparse matrix containing interactions between nearly touching particles whose separation is smaller than some short cutoff distance r_{cut}. The sum of these matrices is required to be positive semi-definite and can be formulated in such a way that it is symmetric as well. Additionally, one can ensure that $\mathcal{M}$ is positive-definite when it is built from a multipole expansion (Kim & Karrila 2005). One should design $\boldsymbol{R}^{nf}$ to be positive-definite as well because this guarantees the sum with $\mathcal{M}$ is indeed positive. Ideally, $\mathcal{M}$ and $\boldsymbol{R}^{nf}$ should represent physics at well-separated scales. For example, this is the case in traditional SD, where the multipole expansion defining $\mathcal{M}$ is most accurate for well-separated particles, while the lubrication interactions

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

contained within $\boldsymbol{R}^{nf}$ are only non-negligible for particle pairs separated by a distance of at most a few particle radii, for which they become very large in magnitude. The near-field/far-field decomposition is not well defined if the near field and far field are not well separated.

It is important to note that any pairwise construction of $\boldsymbol{R}^{nf}_{FU}$ is guaranteed to be positive semi-definite provided the pair resistance tensors used to build it, $(\boldsymbol{R}^{nf}_{FU})_{2B}$, are positive semi-definite. This is straightforward to show using the definition of positive semi-definiteness, i.e. $\boldsymbol{v}^{\mathrm{T}} \cdot \boldsymbol{R}^{nf}_{FU} \cdot \boldsymbol{v} \geqslant 0 \ \forall \boldsymbol{v}$. For pairwise construction, $\boldsymbol{R}^{nf}_{FU} = \sum_p \boldsymbol{S}_p \cdot (\boldsymbol{R}^{nf}_{FU})^p_{2B} \cdot \boldsymbol{S}_p^{\mathrm{T}}$, the sum is evaluated over all particle pairs p, $\boldsymbol{S}_p$ maps the forces for a given particle pair into the list of all particle forces, its transpose projects the velocities for the pair out of the set of all particle velocities, and $(\boldsymbol{R}^{nf}_{FU})^p_{2B}$ is the resistance tensor for the pair. Therefore, $\boldsymbol{v}^{\mathrm{T}} \cdot \boldsymbol{R}^{nf}_{FU} \cdot \boldsymbol{v} = \sum_p \boldsymbol{v}_p^{\mathrm{T}} \cdot (\boldsymbol{R}^{nf}_{FU})^p_{2B} \cdot \boldsymbol{v}_p \geqslant 0$, where $\boldsymbol{v}_p = \boldsymbol{S}_p^{\mathrm{T}} \cdot \boldsymbol{v}$, if and only if $\boldsymbol{v}^{\mathrm{T}} \cdot (\boldsymbol{R}^{nf}_{FU})_{2B} \cdot \boldsymbol{v} \geqslant 0 \ \forall \boldsymbol{v}$, which completes the proof.

Equation (2.2) has the form of a Schur complement. Schur complements arise in the solution of saddle point problems, which suggests that a general SD-like approach can be based on inverting a saddle point matrix. Specifically, equation (2.2) corresponds to the Schur complement, $\mathcal{S}$, of the upper left block of the saddle point matrix

$$\mathcal{A} = \begin{bmatrix} \mathcal{M} & \mathcal{B} \\ \mathcal{B}^{\mathrm{T}} & -\boldsymbol{R}^{nf}_{FU} \end{bmatrix}, \tag{2.5}$$

where

$$\mathcal{S} = -(\boldsymbol{R}^{nf}_{FU} + \mathcal{B}^{\mathrm{T}} \cdot \mathcal{M}^{-1} \cdot \mathcal{B}), \tag{2.6}$$

and

$$\boldsymbol{R}_{FU} = -\mathcal{S}. \tag{2.7}$$

Solution methods for saddle point matrices are well characterized (Benzi *et al.* 2005). In particular, saddle point matrices permit block factorizations and are block diagonalizable using the Schur complement. Therefore, the product of $\boldsymbol{R}^{-1}_{FU}$ with a vector, which is needed to integrate the overdamped equations of motion, is readily and efficiently computed by inverting the saddle point matrix, as discussed in § 2.1. The saddle point equation defining the particle velocities in (2.1) is

$$\begin{bmatrix} \mathcal{M} & \mathcal{B} \\ \mathcal{B}^{\mathrm{T}} & -\boldsymbol{R}^{nf}_{FU} \end{bmatrix} \cdot \begin{bmatrix} \mathcal{F} \\ \boldsymbol{U} - \boldsymbol{U}^{\infty} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{E}^{\infty} \end{bmatrix} \\ -(\boldsymbol{F}^{P} + \boldsymbol{R}^{nf}_{FE} : \boldsymbol{E}^{\infty} + \boldsymbol{F}^{B}) \end{bmatrix}, \tag{2.8}$$

where $\mathcal{F}$ represents the force density in the far-field hydrodynamic approximation, which is composed of the force, torque and stresslet, $\mathcal{F} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{S} \end{bmatrix}$. Its kinematic counterpart on the right-hand side of (2.8) is the set of particle translations, rotations and rates of strain, denoted $\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{E}^{\infty} \end{bmatrix}$. Only the near-field force-strain resistance coupling appears on the right-hand side of (2.8) because the far-field component is accounted for by inverting the saddle point matrix. Furthermore, the total hydrodynamic stresslet is

$$\boldsymbol{S} = \boldsymbol{S}^{ff} - \boldsymbol{R}^{nf}_{SU} \cdot (\boldsymbol{U} - \boldsymbol{U}^{\infty}) + \boldsymbol{R}^{nf}_{SE} : \boldsymbol{E}^{\infty} - k_B T \boldsymbol{\nabla} \cdot (\boldsymbol{R}_{FU} \cdot \boldsymbol{R}^{-1}_{SU}), \tag{2.9}$$

where $\boldsymbol{S}^{ff} = [\boldsymbol{0} \quad \boldsymbol{I}] \cdot \mathcal{F}$ is the far-field hydrodynamic stresslet and $S^B = -k_B T \boldsymbol{\nabla} \cdot (\boldsymbol{R}_{FU} \cdot \boldsymbol{R}^{-1}_{SU})$ is the mean Brownian stresslet (Brady 1993). An imposed shear flow, $\boldsymbol{E}^{\infty}$,

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

generates lubrication forces between nearly touching particles, as represented by the near-field resistance tensors $\boldsymbol{R}_{FE}^{nf}$ and $\boldsymbol{R}_{SE}^{nf}$, which describe the shear flow contribution to the generalized force and stresslet, respectively. With an efficient method to sample $\boldsymbol{F}^B$, equation (2.8) represents a straightforward approach to solving for the velocities required of dynamic SD simulations.

Various hydrodynamic approximations with their associated constraints can be used in (2.8) to simulate a wide range of materials at different levels of accuracy. Typically, higher-order approximations are constructed by increasing the number of degrees of freedom per particle then constraining those degrees of freedom. For example, the near-field resistance tensor can be viewed as a constraint on the relative motion of nearly touching particles. Additional constraints and corrections, e.g. the three-body correction of Cichocki, Ekiel-Jeżewska & Wajnryb (1999), are readily incorporated within this framework. Meng & Higdon (2008*a*) performed simulations of platelet-like particles that were constructed from assemblies of beads constrained to move collectively as a rigid body. The hydrodynamic interactions between the beads were approximated using the far-field plus lubrication correction as is typical in SD. In this case, $\mathcal{B}$ maps rigid body motions to the surface bead velocities and its transpose sums bead force moments to produce the net force and torque on the full rigid assembly. Section 6 discusses rigid body formulations of the Stokesian dynamics problem in more detail. Another application in which the saddle point formulation could have been applied is in the sedimenting fibre simulations of Saintillan, Darve & Shaqfeh (2005), where $\mathcal{B}$ would represent the line integrals over the fibre contour, and $\mathcal{M}$ and $\boldsymbol{R}_{FU}^{nf}$ would be given by their far-field and near-field expressions for the fibre–fibre coupling.

2.1. *Saddle point preconditioner*

The saddle point matrices described in § 2 are efficiently solved using iterative techniques such as the generalized minimum residuals method (GMRES) (Saad & Schultz 1986). Preconditioners are commonly applied to improve the convergence of iterative matrix solvers, particularly in applications where the matrix is ill-conditioned. The main idea behind preconditioning is to apply a linear transformation $\mathcal{P}^{-1} \approx \mathcal{A}^{-1}$ to the matrix equation $\mathcal{A} \cdot \boldsymbol{x} = \boldsymbol{b}$, so that $\mathcal{P}^{-1} \cdot \mathcal{A}$ has a smaller condition number than $\mathcal{A}$, and whose solution therefore converges more quickly. Saddle point matrices commonly arise in the formulation of engineering and mathematical problems with constraints, and their preconditioning and solution are well characterized (Benzi *et al.* 2005). To develop a preconditioner, we start with the block triangular factorization of the saddle point matrix

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \mathcal{B}^{\mathrm{T}} \cdot \mathcal{M}^{-1} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \mathcal{M} & \boldsymbol{0} \\ \boldsymbol{0} & \mathcal{S} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I} & \mathcal{M}^{-1} \cdot \mathcal{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{2.10}$$

where $\boldsymbol{I}$ is the identity matrix. A good preconditioner provides a reasonably accurate approximation of the actual matrix while remaining inexpensive to compute. In the preconditioner, we set $\mathcal{M} = \zeta^{-1}\boldsymbol{I}$, where ζ is an effective drag coefficient, because $\mathcal{M}$ is dense and ill-conditioned, so its inverse is challenging to compute. The projector $\mathcal{B}$ is generally straightforward to compute so it is used in the preconditioner without approximation. To apply the preconditioner, it is necessary to invert the approximation for $\mathcal{A}$, so $\mathcal{S}$ is approximated by $\mathcal{L} \cdot \mathcal{L}^{\mathrm{T}}$, where $\mathcal{L}$ is the zero fill-in incomplete lower Cholesky factor of $\boldsymbol{P} \cdot \mathcal{S} \cdot \boldsymbol{P}^{\mathrm{T}}$ or a reasonable approximation to it, and $\boldsymbol{P}$ is the unitary

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

permutation matrix representing the reverse Cuthill–McKee (RCM) reordering (Cuthill & McKee 1969; George & Liu 1981) of $\boldsymbol{R}_{FU}^{nf}$. The RCM reordering reduces the sparse matrix bandwidth and is applied in this case to reduce the number of dropped entries in the incomplete Cholesky factor. This improves the stability of the method because the incomplete Cholesky factorization is not guaranteed to exist even for symmetric positive-definite matrices. The approximation for $\mathcal{S}$ used in the preconditioner is

$$\tilde{\mathcal{S}} = \zeta \boldsymbol{I} + \tilde{\boldsymbol{R}}_{FU}^{nf}, \tag{2.11}$$

where $\tilde{\boldsymbol{R}}_{FU}^{nf}$ is an approximation to $\boldsymbol{R}_{FU}^{nf}$ built from particle pairs whose separation is less than $r_p < r_{cut}$, and ζ is an approximation for the drag coefficient. For monodisperse spheres one might choose the Stokes drag $\zeta = 6\pi\eta a$, but Viera (2002) notes that larger values can improve performance in various applications. Here we will simply use the Stokes drag coefficient. This approximation increases the sparsity of $\boldsymbol{R}_{FU}^{nf}$, reducing memory requirements and accelerating computation, while simultaneously remaining an effective preconditioner. We will show in § 3 that $\tilde{\boldsymbol{R}}_{FU}^{nf}$ is an effective preconditioner because it retains the largest values from $\boldsymbol{R}_{FU}^{nf}$, which originate with the most narrowly separated particles, and that rather small values of r_p are sufficient to accelerate convergence. The preconditioner is

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \mathcal{B}^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \zeta^{-1}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \mathcal{L} \cdot \mathcal{L}^{\mathrm{T}} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I} & \mathcal{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{2.12}$$

and its inverse

$$\mathcal{P}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\mathcal{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \zeta\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \mathcal{L}^{-\mathrm{T}} \cdot \mathcal{L}^{-1} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\mathcal{B}^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix}, \tag{2.13}$$

can be quickly and inexpensively applied to a vector using the triangular solves of $\mathcal{L}$ and the simple projector $\mathcal{B}$.

2.2. *Brownian dynamics in the saddle point formulation*

In the SD framework, the stochastic force $\boldsymbol{F}^B$ is also decomposed into near-field and far-field parts (Banchio & Brady 2003)

$$\boldsymbol{F}^B = \boldsymbol{F}_{nf}^B + \boldsymbol{F}_{ff}^B, \tag{2.14}$$

where

$$\boldsymbol{F}_{nf}^B = \sqrt{\frac{2k_BT}{\Delta t}} (\boldsymbol{R}_{FU}^{nf})^{1/2} \cdot \boldsymbol{\Psi}_{nf} \tag{2.15}$$

and

$$\boldsymbol{F}_{ff}^B = \sqrt{\frac{2k_BT}{\Delta t}} (\mathcal{B}^{\mathrm{T}} \cdot \mathcal{M}^{-1} \cdot \mathcal{B})^{1/2} \cdot \boldsymbol{\Psi}_{ff}, \tag{2.16}$$

where $\boldsymbol{\Psi}_{nf}$ and $\boldsymbol{\Psi}_{ff}$ are vectors of standard independent Gaussian random variables. The stochastic quantities can also be represented as stochastic variables with zero mean and a covariance set by the fluctuation dissipation theorem, namely

$$\langle \boldsymbol{F}_{nf}^B(\mathrm{i}\Delta t) \boldsymbol{F}_{nf}^B(j\Delta t) \rangle = \frac{2k_BT}{\Delta t} \boldsymbol{R}_{FU}^{nf} \delta_{ij}, \tag{2.17}$$

$$\langle \boldsymbol{F}_{ff}^B(\mathrm{i}\Delta t) \boldsymbol{F}_{ff}^B(j\Delta t) \rangle = \frac{2k_BT}{\Delta t} \mathcal{B}^{\mathrm{T}} \cdot \mathcal{M}^{-1} \cdot \mathcal{B} \delta_{ij}, \tag{2.18}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where the angle brackets denote an ensemble average over realizations of the stochastic variables and δ_{ij} is the Kronecker delta. In this way, computing $\boldsymbol{F}^B_{nf}$ and $\boldsymbol{F}^B_{ff}$ can be thought of as generating samples of an instantaneously correlated random vector whose covariance is proportional to a specific matrix. The near-field Brownian forces in (2.15) are reasonably straightforward to compute because the near-field resistance tensor is known and can be directly computed. The far-field Brownian forces, however, are more challenging because the far-field interactions are described in terms of the mobility, whose inverse is therefore required to compute the forces, as shown in (2.16). Banchio & Brady (2003) compute these forces directly using the Chebyshev polynomial approximation developed by Fixman (1986), which is computationally expensive because $\mathcal{M}$ is ill-conditioned. The saddle point formulation enables a more efficient approach, namely the computation of a far-field slip velocity

$$\mathcal{U}^B = \sqrt{\frac{2k_BT}{\Delta t}}\mathcal{M}^{1/2}\cdot\hat{\boldsymbol{\Psi}}_{ff}, \tag{2.19}$$

that emerges by rearranging (2.8) to give

$$\begin{bmatrix}\mathcal{M} & \mathcal{B}\\ \mathcal{B}^{\mathrm{T}} & -\boldsymbol{R}^{nf}_{FU}\end{bmatrix}\cdot\begin{bmatrix}\hat{\mathcal{F}}\\ \boldsymbol{U}\end{bmatrix} = \begin{bmatrix}\mathcal{U}^B\\ -(\boldsymbol{F}^P+\boldsymbol{F}^B_{nf})\end{bmatrix}, \tag{2.20}$$

where $\hat{\mathcal{F}} = \mathcal{F}+\mathcal{M}^{-1}\cdot\mathcal{U}^B$ is the set of far-field hydrodynamic forces plus an additional force required to satisfy the no-slip boundary condition on the particle surfaces and $\hat{\boldsymbol{\Psi}}_{ff}$ is another white noise vector. Introducing $\mathcal{U}^B$ in this way is physically equivalent to imposing a fluctuating stress in the far-field model of the fluid, whose associated particle displacements satisfy the fluctuation–dissipation theorem. For many representations of $\mathcal{M}$, $\mathcal{U}^B$ is far easier to compute than $\boldsymbol{F}^B_{ff}$, and accelerated techniques with $O(N)$ or nearly $O(N)$ scaling have been developed for various approximations to $\mathcal{M}$. Examples include the fluctuating immersed boundary method published by Delong *et al.* (2014), the fluctuating force coupling method of Delmotte & Keaveny (2015) and our own positively split Ewald approach (Fiore *et al.* 2017; Fiore & Swan 2018).

Others have also used the concept of a far-field slip velocity to efficiently model the thermal motion of dispersed particles. For example, Delmotte & Keaveny (2015) considered spherical particles with force, torques and stresslets, and introduced a slip velocity that included a non-zero rate of strain on each particle. Sprinkle *et al.* (2017) considered rigid multi-blob particles and introduced a fluid slip velocity in an approach that is a special case of the one presented here. In particular, equation (6) in their paper is the special case of (2.20) in which lubrication interactions are neglected and $\mathcal{B}$ represents their geometric block matrix, $-\mathcal{K}$, describing rigid body motions. While the saddle point matrix equation has been previously used by Sprinkle *et al.* (2017) and others in the context of hydrodynamic mobility problems, equation (2.20) is the first time these ideas have been applied to SD or resistance-type problems, in which fluctuating and deterministic near-field forces also need be considered. Equation (2.20) represents a completely general framework for SD simulations of arbitrary particle shapes with relatively few restrictions on the model imposed for near-field HI. No prescription on the particle sizes or shapes has been made, and the method is valid for any symmetric positive semi-definite approximations for $\mathcal{M}$ and $\boldsymbol{R}^{nf}_{FU}$. In §§ 3 and 4 we discuss techniques specific to spherical particles in periodic geometries, both for computing the deterministic HI and the stochastic forces and slip velocities.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

2.3. *Stochastic sampling preconditioning*

As mentioned in the previous section, accelerated sampling methods for the effective far-field Brownian force can be developed by leveraging the algebraic structure of various approximations for $\mathcal{M}$ to generate a far-field slip velocity which is then projected onto the set of hydrodynamic constraints. Further discussion of such methods is deferred to § 4. In this section, we describe an efficient and completely general approach to sample the near-field Brownian forces. This approach relies on the iterative Krylov subspace approximations of Chow & Saad (2014) and leverages the preconditioner that was already developed in § 2.1 for the saddle point problem.

The preconditioning is more simply described using an alternative, but equivalent, notation for the stochastic variables, where an arbitrary random vector $\boldsymbol{x}$ drawn from a normal distribution with mean $\langle \boldsymbol{x}(\mathrm{i}\Delta t)\rangle = \boldsymbol{\mu}$ and covariance $\langle \boldsymbol{x}(\mathrm{i}\Delta t)\boldsymbol{x}^{\mathrm{T}}(j\Delta t)\rangle = \boldsymbol{V}\delta_{ij}$ is denoted as

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{V}). \tag{2.21}$$

Standard random number generators are only able to draw random samples with a covariance proportional to $\boldsymbol{I}$,

$$\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{\mu}, \alpha\boldsymbol{I}), \tag{2.22}$$

where α is a positive scalar. Sampling a distribution with arbitrary covariance $\boldsymbol{V}$ is achieved by computing the product of the square root of the covariance matrix with the independent and identically distributed vector $\boldsymbol{n}$ computed the random number generator

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{V}^{1/2} \cdot \boldsymbol{n}. \tag{2.23}$$

If $\boldsymbol{V}$ is known explicitly, direct calculation of $\boldsymbol{V}^{1/2}$ can be evaluated with eigenvalue or singular value decomposition, incurring a computational cost of $O(N^3)$. Iterative Krylov subspace approximations for $\boldsymbol{V}^{1/2} \cdot \boldsymbol{n}$, such as the commonly used Chebyshev polynomial approach of Fixman (1986) and the more recently developed Lanczos method of Chow & Saad (2014), can be applied for an arbitrary $\boldsymbol{V}$. While these iterative schemes may converge even for ill-conditioned matrices, they are especially efficient when $\boldsymbol{V}$ is well-conditioned and $\boldsymbol{V} \cdot \boldsymbol{x}$ can be rapidly computed. Preconditioning can be used to improve the condition number and accelerate convergence. We will show how this can be achieved to sample $\boldsymbol{R}_{FU}^{nf}$.

Important to any sampling approach is that $\boldsymbol{R}_{FU}^{nf}$ is positive semi-definite, which guarantees that the resulting forces are real-valued. Preconditioning is necessary because $\boldsymbol{R}_{FU}^{nf}$ is ill-conditioned. The large condition number arises because it has both very large eigenvalues originating in the divergent forces between nearly touching particles and very small eigenvalues for particles with few neighbours whose separations are close to the near-field cutoff radius, and whose entries in the resistance tensor are nearly zero. The large and small eigenvalues are most effectively preconditioned independently.

The preconditioner developed in § 2.1 is primarily designed to condition the large eigenvalues present in $\boldsymbol{R}_{FU}^{nf}$, so it can be recycled here with some careful modifications that ensure the resulting variables have the proper covariance. Here, the relevant piece of the saddle point preconditioner is the incomplete Cholesky factor $\mathcal{L}$ of the matrix $\boldsymbol{P} \cdot (\boldsymbol{R}_{FU}^{nf} + \zeta\boldsymbol{I}) \cdot \boldsymbol{P}^{\mathrm{T}}$, which is applied to $\boldsymbol{R}_{FU}^{nf}$ symmetrically in order to reduce the magnitude of the large eigenvalues,

$$\mathcal{L}^{-1} \cdot \boldsymbol{P} \cdot \boldsymbol{R}_{FU}^{nf} \cdot \boldsymbol{P}^{\mathrm{T}} \cdot \mathcal{L}^{-\mathrm{T}}. \tag{2.24}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

To precondition the small eigenvalues, we use a diagonal matrix that is designed to scale each row and column of $\boldsymbol{R}_{FU}^{nf}$ with suitably small non-zero diagonals, $0 < (\boldsymbol{R}_{FU}^{nf})_{ii} < \zeta$, by the diagonal element, so that the relative magnitude of the entries within each row and column is preserved but the absolute magnitudes cluster around ζ. This preconditioner does not alter the rows and columns of $\boldsymbol{R}_{FU}^{nf}$ with diagonals that are either zero or greater than or equal to a nominal drag coefficient, say $\zeta = 6\pi\eta a$ for mono-disperse spherical particles, i.e.

$$D_{ii} = \begin{cases} (\boldsymbol{R}_{FU}^{nf})_{ii}^{1/2}, & 0 < (\boldsymbol{R}_{FU}^{nf})_{ii} < 6\pi\eta a, \\ 1, & (\boldsymbol{R}_{FU}^{nf})_{ii} = 0, \\ 1, & (\boldsymbol{R}_{FU}^{nf})_{ii} \geqslant 6\pi\eta a. \end{cases} \tag{2.25}$$

Here, $\boldsymbol{D}$ is primarily expected to be important at low particle volume fractions, for which particles are more likely to be well separated and have only a few neighbours within the near-field cutoff radius. Combining this preconditioner with the incomplete Cholesky preconditioner for large eigenvalues yields the preconditioned matrix

$$\mathcal{L}^{-1} \cdot \boldsymbol{P} \cdot \boldsymbol{D}^{-1} \cdot \boldsymbol{R}_{FU}^{nf} \cdot \boldsymbol{D}^{-\mathrm{T}} \cdot \boldsymbol{P}^{\mathrm{T}} \cdot \mathcal{L}^{-\mathrm{T}}. \tag{2.26}$$

Unfortunately, this preconditioner is not a great approximation for $\boldsymbol{R}_{FU}^{nf}$ because $\boldsymbol{R}_{FU}^{nf}$ only has non-zero rows for particles that are closely separated, whereas the addition of $\zeta \boldsymbol{I}$ in the computation of $\mathcal{L}$ guarantees that $\mathcal{L}$ has no rows or columns that are purely zero. The effect of the preconditioner can be improved by instead sampling forces from the distribution

$$\hat{\boldsymbol{F}}_{nf}^{B} \sim \mathcal{N}\left(\boldsymbol{0}, \frac{2k_B T}{\Delta t}(\boldsymbol{R}_{FU}^{nf} + \zeta \boldsymbol{I}_{nn})\right), \tag{2.27}$$

where $\boldsymbol{I}_{nn}$ is a modified identity tensor with ones where the diagonal elements of $\boldsymbol{R}_{FU}^{nf}$ are zero, i.e. for the rows/columns associated with neighbourless particles, and zero otherwise, i.e.

$$(\boldsymbol{I}_{nn})_{ii} = \begin{cases} 1, & (\boldsymbol{R}_{FU}^{nf})_{ii} = 0, \\ 0, & (\boldsymbol{R}_{FU}^{nf})_{ii} \neq 0. \end{cases} \tag{2.28}$$

Neighbourless particles experience no near-field Brownian force, so the proper force distribution can be obtained by projecting the corresponding elements of $\boldsymbol{F}_{nf}^{B}$ to zero

$$\boldsymbol{F}_{nf}^{B} = (\boldsymbol{I} - \boldsymbol{I}_{nn}) \cdot \hat{\boldsymbol{F}}_{nf}^{B} \sim \mathcal{N}\left(\boldsymbol{0}, \frac{2k_B T}{\Delta t}\boldsymbol{R}_{FU}^{nf}\right). \tag{2.29}$$

The operator $(\boldsymbol{I} - \boldsymbol{I}_{nn})$ is a projector that removes the additional noise introduced by adding $\zeta \boldsymbol{I}_{nn}$ to $\boldsymbol{R}_{FU}^{nf}$. This operation works because we project out a subset of the Brownian forces. The preconditioned sampling produces random vectors

$$z \sim \mathcal{N}\left(\boldsymbol{0}, \frac{2k_B T}{\Delta t}\mathcal{L}^{-1} \cdot \boldsymbol{P} \cdot \boldsymbol{D}^{-1} \cdot (\boldsymbol{R}_{FU}^{nf} + \zeta \boldsymbol{I}_{nn}) \cdot \boldsymbol{D}^{-\mathrm{T}} \cdot \boldsymbol{P}^{\mathrm{T}} \cdot \mathcal{L}^{-\mathrm{T}}\right), \tag{2.30}$$

where

$$\boldsymbol{F}_{nf}^{B} = (\boldsymbol{I} - \boldsymbol{I}_{nn}) \cdot \boldsymbol{D} \cdot \boldsymbol{P}^{\mathrm{T}} \cdot \mathcal{L} \cdot z \sim \mathcal{N}\left(\boldsymbol{0}, \frac{2k_B T}{\Delta t}\boldsymbol{R}_{FU}^{nf}\right). \tag{2.31}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

The use of $\boldsymbol{I}_{nn}$ and $\boldsymbol{I}-\boldsymbol{I}_{nn}$ bears some additional discussion. As mentioned, adding $\boldsymbol{I}_{nn}$ to $\boldsymbol{R}_{FU}^{nf}$ improves the efficacy of the incomplete Cholesky preconditioner but increases the variance of the stochastic forces. Fortunately, particles that have no neighbours in the near field cannot experience a near-field Brownian force; the operator $\boldsymbol{I}-\boldsymbol{I}_{nn}$ projects these forces out of $\boldsymbol{F}_{nf}^{B}$ and restores the expected covariance. This approach only works because those forces should be exactly zero. Additional preconditioning may be possible for specific forms of $\boldsymbol{R}_{FU}^{nf}$, where knowledge of the specific interactions can be used to further compress the eigenspectrum and improve convergence. The efficacy of the preconditioner presented in this section is demonstrated in § 4 for the Stokesian dynamics lubrication approximation for the near-field resistance.

It is important to note that the approach advocated in this section is not the same as simply omitting the particles that have no neighbours. Rather, Brownian forces are computed for all particles then projected onto the set of particles with neighbours. The motivation behind this approach is to leverage the preconditioner already developed for the saddle point problem in § 2.1, which effectively preconditions many of the features that contribute to large and small eigenvalues in the covariance matrix of the stochastic sampling problem, so as to avoid repeating those calculations. In order to use the saddle point preconditioner, which considers all particles and not solely the ones with neighbours, the Brownian sampling method must at some stage generate noise for these particles without neighbours.

3. Hydrodynamic models

3.1. *Far-field hydrodynamics for monodisperse spheres*

The far-field mobility tensor $\mathcal{M}$ is constructed by multipole expansion. We have previously developed a compact representation of $\mathcal{M}$ (Fiore *et al.* 2017; Fiore & Swan 2018), which will be summarized here for completeness. Equivalent representations have also been developed by Ladd (1990) and Mizerski *et al.* (2014). This representation is derived from the integral formulation of the Stokes equations for rigid suspended bodies (Ladyzhenskaya & Silverman 1969)

$$u_i(\boldsymbol{x}) = \int_{S_\beta} \mathrm{d}S_\beta(\boldsymbol{y}) J_{im}(\boldsymbol{x},\boldsymbol{y}) f_m(\boldsymbol{y}), \tag{3.1}$$

where $u_i(\boldsymbol{x})$ is the ith component of the fluid velocity at a point $\boldsymbol{x}$ in the fluid propagated by the force density $\boldsymbol{f}(\boldsymbol{y})$ at point $\boldsymbol{y}$ on the particle surface S_β through the Green's function $\boldsymbol{J}$. In this and following equations, Einstein summation notation is assumed for repeated indices. All particles are assumed to be spherical and the first few multipole moments are used to approximate the force density on the surface of each particle

$$f_j = \frac{1}{4\pi a^2} F_j + \frac{3}{4\pi a^3} \hat{n}_n C_{nj} + \cdots, \tag{3.2}$$

where $\hat{\boldsymbol{n}}$ is the normal to the particle surface, a is the particle radius, $\boldsymbol{F}$ is the total force on the particle and $\boldsymbol{C}$ is the traceless couplet, whose symmetric and antisymmetric components are the stresslet and torque, respectively. The multipole expansion is truncated after the couplet for computational efficiency, but can in principle be extended to arbitrary order. The force and couplet are defined by

$$F_m = \int_{S_\beta} \mathrm{d}S_\beta f_m, \tag{3.3}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

$$C_{mn} = \int_{S_\beta} \mathrm{d}S_\beta \left[(y_m - x_{\beta,m}) f_n - \tfrac{1}{3}\delta_{mn}(y_p - x_{\beta,p}) f_p \right], \tag{3.4}$$

where $\boldsymbol{x}_\beta$ is the location of the centre of particle β, and higher-order moments are defined by higher-order outer products of the distance vector $(\boldsymbol{y} - \boldsymbol{x})$ with $\boldsymbol{f}$. Faxeń laws infer particle motion in response to the stresses exerted on its surface by a flow field, and are effectively integral statements of the no-slip boundary condition,

$$U_i = \frac{1}{4\pi a^2} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) u_i(\boldsymbol{x}), \tag{3.5}$$

$$D_{ij} = \frac{3}{4\pi a^3} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) u_i \hat{n}_j, \tag{3.6}$$

where S_α is the surface of particle α, and V_α is its volume. The particle velocity gradient, $\boldsymbol{D}$, is composed of a symmetric part, the rate of strain $\boldsymbol{E}$, and an anti-symmetric part, the angular velocity $\boldsymbol{\Omega}$.

The linearity of the Stokes equations allows independent coupling and superposition of the total force or couplet to the velocity or velocity gradient. Substituting the fluid velocity (3.1) and the multipole expansion (3.2) into the Faxén laws (3.5), (3.6) gives the following relations between the force moments on particle β and the velocity moments induced on particle α

$$\begin{bmatrix} U_i^\alpha \\ D_{ij}^\alpha \end{bmatrix} = \begin{bmatrix} M_{UF,im}^{\alpha\beta} & M_{UC,imn}^{\alpha\beta} \\ M_{DF,ijm}^{\alpha\beta} & M_{DC,ijmn}^{\alpha\beta} \end{bmatrix} \cdot \begin{bmatrix} F_m^\beta \\ C_{mn}^\beta \end{bmatrix}, \tag{3.7}$$

where the elements of the pair mobility tensors are given in terms of integrals of the Green's function over the particle surfaces

$$M_{UF,im}^{\alpha\beta} = \frac{1}{4\pi a^2} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) \frac{1}{4\pi a^2} \int_{S_\beta} \mathrm{d}S_\beta(\boldsymbol{y}) J_{im}(\boldsymbol{x}, \boldsymbol{y}), \tag{3.8}$$

$$M_{UC,imn}^{\alpha\beta} = \frac{1}{4\pi a^2} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) \frac{3}{4\pi a^3} \int_{S_\beta} \mathrm{d}S_\beta(\boldsymbol{y}) J_{im}(\boldsymbol{x}, \boldsymbol{y}) \hat{n}_n^\beta, \tag{3.9}$$

$$M_{DF,ijm}^{\alpha\beta} = \frac{3}{4\pi a^3} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) \hat{n}_j^\alpha \frac{1}{4\pi a^2} \int_{S_\beta} \mathrm{d}S_\beta(\boldsymbol{y}) J_{im}(\boldsymbol{x}, \boldsymbol{y}), \tag{3.10}$$

$$M_{DC,ijmn}^{\alpha\beta} = \frac{3}{4\pi a^3} \int_{S_\alpha} \mathrm{d}S_\alpha(\boldsymbol{x}) \hat{n}_j^\alpha \frac{3}{4\pi a^3} \int_{S_\beta} \mathrm{d}S_\beta(\boldsymbol{y}) J_{im}(\boldsymbol{x}, \boldsymbol{y}) \hat{n}_n^\beta. \tag{3.11}$$

For periodic systems, which are commonly used in simulation, the appropriate Green's function is given by Hasimoto (1959),

$$\boldsymbol{J}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\eta V} \sum_{\boldsymbol{k} \neq \boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}-\boldsymbol{y})} \frac{1}{k^2} (\boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}), \tag{3.12}$$

where $\boldsymbol{J}(\boldsymbol{x}, \boldsymbol{y}) \cdot \boldsymbol{F}$ is the fluid velocity at point $\boldsymbol{x}$ propagated by a force $\boldsymbol{F}$ located at point $\boldsymbol{y}$ and its periodic images on a lattice, $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{k}$ are drawn from the set of reciprocal (Fourier) space lattice vectors excluding the zero vector, η is fluid viscosity and V is the periodic cell volume. With Hasimoto's solution, the integrals in (3.8)–(3.11) can be evaluated in principle. The integrals do not necessarily have

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

analytically tractable solutions for arbitrary particle shapes, but are straightforward to evaluate for spherical particles, which are the focus of this work. Substituting (3.12) into (3.8)–(3.11) and evaluating the surface integrals in Fourier space gives algebraic expressions for the pair mobilities of spherical particles with equal, finite size

$$\boldsymbol{M}_{pq}^{\alpha\beta} = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(ka). \tag{3.13}$$

The shape factors are proportional to spherical Bessel functions of the first kind, and can be related to the order of the moment in the multipole expansion p as,

$$\boldsymbol{s}^p(ka) = (-1)^{p/2}(2p+1)!!\mathrm{j}_p(ka)\,\hat{\boldsymbol{k}}^p, \tag{3.14}$$

where $\hat{\boldsymbol{k}}^p$ is an irreducible tensor of rank p (Ladd 1990) and $\mathrm{j}_m(x)$ is the spherical Bessel function of the first kind of order m (Abramowitz & Stegun 1965). The shape factors for the force and linear velocity ($p=0$) are $\boldsymbol{s}^0(ka) = \mathrm{j}_0(ka)$ and the shape factors for the couplet and velocity gradient ($p=1$) are $\boldsymbol{s}^1(ka) = 3\sqrt{-1}\boldsymbol{k}\mathrm{j}_1(ka)$.

The sum in (3.13) is slowly converging, but the convergence can be accelerated using the Ewald sum splitting (Ewald 1921), which filters the slowly decaying, large wavenumber terms out of the sum and explicitly reconstructs them in real space, where they converge quickly because they are associated with high-frequency (short wavelength) motions. Mathematically, the mobility sum $\boldsymbol{M}_{pq}^{\alpha\beta} = \sum_{\boldsymbol{k}\neq\boldsymbol{0}}\boldsymbol{M}_k$ is written in two parts $\boldsymbol{M}_{pq}^{\alpha\beta} = \sum_{\boldsymbol{k}\neq\boldsymbol{0}}(1-H(k,\xi))\boldsymbol{M}_k + \sum_{\boldsymbol{k}\neq\boldsymbol{0}}H(k,\xi)\boldsymbol{M}_k$, where the first term is analytically reconstructed in real space by applying the inverse Fourier transform and the second sum is left in wave space. Hasimoto (1959) derived the Ewald sum for periodic Stokes flow using the filter

$$H(k,\xi) = (1+k^2/4\xi^2)\mathrm{e}^{-k^2/4\xi^2}, \tag{3.15}$$

and showed that the terms in the resulting real-space sum are exponentially decaying functions of distance, while the terms in the wave-space sum are exponentially decaying functions of wavenumber. The variable ξ is an arbitrary parameter that controls how many terms are filtered out of the wave-space sum; in the limit that $\xi\to 0$, the entire sum is evaluated in real space. Using this filter, the real-space and wave-space contributions to the pair mobility tensor, $\boldsymbol{M}_{pq}^{\alpha\beta} = (\boldsymbol{M}_{pq}^{\alpha\beta})_r + (\boldsymbol{M}_{pq}^{\alpha\beta})_w$, are respectively

$$(\boldsymbol{M}_{pq}^{\alpha\beta})_r = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}(1-H(k,\xi))\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(ka), \tag{3.16}$$

$$(\boldsymbol{M}_{pq}^{\alpha\beta})_w = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}H(k,\xi)\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(ka). \tag{3.17}$$

The real-space representation of the pair mobility tensors are computed by taking the inverse Fourier transform of (3.16) for each set of shape factors. The compact representations of these tensors are

$$(M_{UF,im}^{\alpha\beta})_r = c_{UF}^{(1)}(r)(\delta_{im}-\hat{r}_i\hat{r}_m) + c_{UF}^{(2)}(r)\hat{r}_i\hat{r}_m, \tag{3.18}$$

$$(M_{UC,imn}^{\alpha\beta})_r = c_{UC}^{(1)}(r)(\delta_{im}\hat{r}_n - \hat{r}_i\hat{r}_m\hat{r}_m) + c_{UC}^{(2)}(r)(\delta_{in}\hat{r}_m + \delta_{mn}\hat{r}_i - 4\hat{r}_i\hat{r}_m\hat{r}_n), \tag{3.19}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

$$(M^{\alpha\beta}_{DC,ijmn})_r = c^{(1)}_{DC}(r)(\delta_{ij}\delta_{mn} + \delta_{im}\delta_{jn} - 4\delta_{in}\delta_{jm}) + c^{(2)}_{DC}(r)(\delta_{jm}\hat{r}_i\hat{r}_n - \hat{r}_i\hat{r}_j\hat{r}_m\hat{r}_n) + c^{(3)}_{DC}(r)(\delta_{ij}\hat{r}_m\hat{r}_n + \delta_{im}\hat{r}_j\hat{r}_n + \delta_{jn}\hat{r}_i\hat{r}_m + \delta_{mn}\hat{r}_i\hat{r}_j + \delta_{in}\hat{r}_j\hat{r}_m - 6\hat{r}_i\hat{r}_j\hat{r}_m\hat{r}_n - \delta_{in}\delta_{jm}), \tag{3.20}$$

where $\hat{\boldsymbol{r}}$ is the unit distance vector between the particle centres, r is its magnitude, δ_{ij} is the Kronecker delta function and the scalar coefficients $c^{(1)}_{UF}$, $c^{(2)}_{UF}$, $c^{(1)}_{UC}$, $c^{(2)}_{UC}$, $c^{(1)}_{DC}$, $c^{(2)}_{DC}$, $c^{(3)}_{DC}$ are exponentially decaying functions of the interparticle distance derived by analytically evaluating the inverse Fourier transform and projecting onto a minimal tensor basis. These functions are detailed in our previous work (Fiore & Swan 2018). It is important to note that the pair mobility tensor presented here, combined with the Hasimoto filter for the Ewald sum applied in Fourier space produces what we term a positive splitting, i.e. a splitting for which the real-space and wave-space contributions to the operator are independently positive semi-definite, as discussed previously (Fiore *et al.* 2017; Fiore & Swan 2018).

The far-field grand mobility tensor, which describes the set of interactions between all particle pairs

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_{UF} & \mathcal{M}_{UC} \\ \mathcal{M}_{DF} & \mathcal{M}_{DC} \end{bmatrix}, \tag{3.21}$$

whose sub-blocks are built from the pair mobility tensors, e.g.

$$\mathcal{M}_{UF} = \begin{bmatrix} \boldsymbol{M}^{11}_{UF} & \boldsymbol{M}^{12}_{UF} & \cdots & \boldsymbol{M}^{1N}_{UF} \\ \boldsymbol{M}^{21}_{UF} & \boldsymbol{M}^{22}_{UF} & & \boldsymbol{M}^{2N}_{UF} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{M}^{N1}_{UF} & \boldsymbol{M}^{N2}_{UF} & \cdots & \boldsymbol{M}^{NN}_{UF} \end{bmatrix}, \tag{3.22}$$

is also positive semi-definite for all particle configurations and its Ewald sum

$$\mathcal{M} = \mathcal{M}_r + \mathcal{M}_w, \tag{3.23}$$

also represents a positive splitting. This is an important feature and is crucial for fast sampling of the far-field Brownian noise, as discussed in § 4.1.

The wave-space component of the mobility can be written compactly using linear operators

$$\mathcal{M}_w = \mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B} \cdot \mathbb{P} \cdot \mathbb{D}, \tag{3.24}$$

where $\mathbb{D}$ is the unitary non-uniform discrete Fourier transform (NUDFT) operator, and $\mathbb{D}^\dagger$ is its adjoint/inverse. $\mathbb{D}$ transforms the force moments applied on particles from real space to Fourier space. $\mathbb{B}$ is a symmetric block diagonal matrix whose ith diagonal block is

$$\mathbb{B}_i = \frac{H(k_i, \xi)}{\eta V k_i^2}(\boldsymbol{I} - \hat{\boldsymbol{k}}_i\hat{\boldsymbol{k}}_i), \tag{3.25}$$

where $\hat{\boldsymbol{k}}_i$ is ith wavevector; $\mathbb{B}$ multiplies the Fourier components of the force by the Green's function $1/\eta V k_i^2$ and Hasimoto $H(k_i, \xi)$ scaling factors and projects them onto the space of divergence-free velocity fields via the orthogonal projector $(\boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}})$. The projector $\mathbb{P}$ contains the non-negative shape factors that map the Fourier-space force moments onto the force density, and its adjoint $\mathbb{P}^\dagger$ maps the Fourier-space velocity

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

field onto the Fourier-space velocity moments; $\mathbb{P}$ is a rectangular matrix whose only non-zero entries are rectangular blocks along the main diagonal,

$$\mathbb{P}_i = \left[\mathrm{j}_0(k_i a)\boldsymbol{I} \quad 3\sqrt{-1}\mathrm{j}_1(k_i a)\boldsymbol{k}_i \right] . \tag{3.26}$$

The matrix $\mathbb{P}_i^{\dagger} \cdot \mathbb{B}_i \cdot \mathbb{P}_i$ is the wave-space representation of the mobility tensor at the ith wave vector, i.e. the summand in (3.17) except for the exponential factor, where the exponential factors and the sum over $\boldsymbol{k} \neq \boldsymbol{0}$ come from the application of the NUDFT and it inverse, $\mathbb{D}$ and $\mathbb{D}^{\dagger}$.

The results presented in this section hold for monodisperse suspensions of spherical particles. Polydispersity can be incorporated in the expressions presented in this section through subtle modifications to the shape factors and scalar functions. This application is discussed in appendix A. Non-spherical particles can be built from rigid assemblies of spherical beads represented using the shape factors presented here. A detailed discussion of this method is deferred to § 6.

3.2. *Near-field hydrodynamics*

We construct the near-field lubrication tensor $\boldsymbol{R}_{\alpha\beta}^{nf}$ separately for each pair of particles α and β whose distance is shorter than a cutoff radius of r_{lub}. The exact series expansion for the hydrodynamic resistance between two spheres (Jeffrey & Onishi 1984; Jeffrey 1992*b*) provides the basis for the SD lubrication tensor. The series expansion is equivalent to carrying the multipole expansion for the isolated particle pair out to an infinite number of terms, so the corresponding resistance tensor $\boldsymbol{R}_{\alpha\beta}^{2B}$ contains within it all the multipole moments associated with each particle pair. The monopole, dipole and isotropic quadrupole and octupole moments are already included in the far-field approximation to the hydrodynamic mobility for each pair, $\mathcal{M}_{\alpha\beta}^{2B}$, so their contribution to the resistance tensor for each particle pair, $\mathcal{R}_{\alpha\beta}^{2B} = (\mathcal{M}_{\alpha\beta}^{2B})^{-1}$, is subtracted from the exact result to avoid double counting them. Therefore, the pair resistance tensor used is

$$\boldsymbol{R}_{\alpha\beta}^{nf} = \boldsymbol{R}_{\alpha\beta}^{2B} - \mathcal{R}_{\alpha\beta}^{2B}. \tag{3.27}$$

This pair resistance tensor is comprised of four sub-blocks describing the *FU*, *FE*, *SU* and *SE* couplings for the particle pair,

$$\boldsymbol{R}_{\alpha\beta}^{nf} = \begin{bmatrix} (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta} & (\boldsymbol{R}_{FE}^{nf})_{\alpha\beta} \\ (\boldsymbol{R}_{SU}^{nf})_{\alpha\beta} & (\boldsymbol{R}_{SE}^{nf})_{\alpha\beta} \end{bmatrix} , \tag{3.28}$$

where

$$(\boldsymbol{R}_{FU}^{nf})_{\alpha\beta} = \begin{bmatrix} (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta}^{11} & (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta}^{12} \\ (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta}^{21} & (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta}^{22} \end{bmatrix} \tag{3.29}$$

is the *FU* component of the resistance tensor for the pair of particles α and β. In this expression, the indices α and β are the global indices of the two particles constituting the pair, i.e. with N particles $1 \leqslant \alpha, \beta \leqslant N$, while within the pair particles α and β are denoted by the indices 1 and 2, respectively. Similar definitions hold for the other three sub-blocks.

The near-field lubrication tensor for all particles is constructed by summing the interactions over all particle pairs

$$\boldsymbol{R}_{FU}^{nf} = \sum_{\alpha} \sum_{\beta > \alpha} \boldsymbol{S}_{\alpha\beta} \cdot (\boldsymbol{R}_{FU}^{nf})_{\alpha\beta} \cdot \boldsymbol{S}_{\alpha\beta}^{\mathrm{T}}, \tag{3.30}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where $\boldsymbol{S}_{\alpha\beta}$ maps the forces for a given particle pair into the list of all particle forces, its transpose projects the velocities for the pair out of the set of all particle velocities and $(\boldsymbol{R}^{nf}_{FU})_{\alpha\beta} = \boldsymbol{0}$ if $r_{\alpha\beta} > 4a$. Written explicitly in matrix notation,

$$\boldsymbol{R}^{nf}_{FU} = \begin{bmatrix} \displaystyle\sum_{\beta\neq 1} (\boldsymbol{R}^{nf}_{FU})^{11}_{1\beta} & (\boldsymbol{R}^{nf}_{FU})^{12}_{12} & \cdots & (\boldsymbol{R}^{nf}_{FU})^{12}_{1N} \\ (\boldsymbol{R}^{nf}_{FU})^{12}_{21} & \displaystyle\sum_{\beta\neq 2} (\boldsymbol{R}^{nf}_{FU})^{11}_{2\beta} & & (\boldsymbol{R}^{nf}_{FU})^{12}_{2N} \\ \vdots & & \ddots & \vdots \\ (\boldsymbol{R}^{nf}_{FU})^{12}_{N1} & (\boldsymbol{R}^{nf}_{FU})^{12}_{N2} & \cdots & \displaystyle\sum_{\beta\neq N} (\boldsymbol{R}^{nf}_{FU})^{11}_{N\beta} \end{bmatrix}. \tag{3.31}$$

The SU, FE and SE components are constructed similarly. In these expressions, and throughout the rest of this paper, we use $\boldsymbol{F}$ to denote the generalized force, which is comprised of the force and torque, and $\boldsymbol{U}$ to denote the generalized velocity, which is comprised of the linear and angular velocity. These can also be thought of as the forces and velocities associated with rigid body motion. In the notation of Jeffrey & Onishi (1984),

$$(\boldsymbol{R}_{FU})_{\alpha\beta} = \begin{bmatrix} \boldsymbol{A}_{\alpha\beta} & \tilde{\boldsymbol{B}}_{\alpha\beta} \\ \boldsymbol{B}_{\alpha\beta} & \boldsymbol{C}_{\alpha\beta} \end{bmatrix}, \tag{3.32}$$

and in the notation of Jeffrey (1992*b*),

$$(\boldsymbol{R}_{SU})_{\alpha\beta} = \begin{bmatrix} \boldsymbol{G}_{\alpha\beta} & \boldsymbol{H}_{\alpha\beta} \end{bmatrix}. \tag{3.33}$$

The far-field two-body contribution to the near-field resistance tensor is computed from the inverse of the mobility tensor for a pair of particles

$$\mathcal{R}^{2B}_{\alpha\beta} = \begin{bmatrix} \boldsymbol{R}^{\alpha\alpha}_{FU} & \boldsymbol{R}^{\alpha\beta}_{FU} & \boldsymbol{R}^{\alpha\alpha}_{SU} & \boldsymbol{R}^{\alpha\beta}_{SU} \\ \boldsymbol{R}^{\beta\alpha}_{FU} & \boldsymbol{R}^{\beta\beta}_{FU} & \boldsymbol{R}^{\beta\alpha}_{SU} & \boldsymbol{R}^{\beta\beta}_{SU} \\ \boldsymbol{R}^{\alpha\alpha}_{FE} & \boldsymbol{R}^{\alpha\beta}_{FE} & \boldsymbol{R}^{\alpha\alpha}_{SE} & \boldsymbol{R}^{\alpha\beta}_{SE} \\ \boldsymbol{R}^{\beta\alpha}_{FE} & \boldsymbol{R}^{\beta\beta}_{FE} & \boldsymbol{R}^{\beta\alpha}_{SE} & \boldsymbol{R}^{\beta\beta}_{SE} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}^{\alpha\alpha}_{UF} & \boldsymbol{M}^{\alpha\beta}_{UF} & \boldsymbol{M}^{\alpha\alpha}_{US} & \boldsymbol{M}^{\alpha\beta}_{US} \\ \boldsymbol{M}^{\beta\alpha}_{UF} & \boldsymbol{M}^{\beta\beta}_{UF} & \boldsymbol{M}^{\beta\alpha}_{US} & \boldsymbol{M}^{\beta\beta}_{US} \\ \boldsymbol{M}^{\alpha\alpha}_{EF} & \boldsymbol{M}^{\alpha\beta}_{EF} & \boldsymbol{M}^{\alpha\alpha}_{ES} & \boldsymbol{M}^{\alpha\beta}_{ES} \\ \boldsymbol{M}^{\beta\alpha}_{EF} & \boldsymbol{M}^{\beta\beta}_{EF} & \boldsymbol{M}^{\beta\alpha}_{ES} & \boldsymbol{M}^{\beta\beta}_{ES} \end{bmatrix}^{-1}. \tag{3.34}$$

As both the mobility and resistance tensors can be written in terms of scalar functions and orthogonal tensors, the inversion in (3.34) can be used to relate the scalar mobility and resistance functions, and therefore does not have to be performed during the simulation. Because the terms of $\boldsymbol{R}^{nf}_{\alpha\beta} = \boldsymbol{R}_{\alpha\beta} - \mathcal{R}_{\alpha\beta}$ decay very quickly (for $r \gg 2a$, $\boldsymbol{R}_{\alpha\beta} \approx \mathcal{R}_{\alpha\beta}$) when constructing the near-field resistance tensor for all particles, only the entries associated with particle pairs whose distance is within a cutoff radius of $r_{lub} = 4a$ are retained, consistent with previous implementations of Stokesian dynamics (Sierou & Brady 2001).

3.3. *Hydrodynamic saddle point solution*

As described in § 2, the far-field and near-field contributions to the hydrodynamic interactions can be combined in a saddle point matrix to solve for the unknown particle velocities and far-field hydrodynamic force moments. Iterative solutions of this matrix equation can be effectively preconditioned with a block preconditioner based on the Schur factorization. The efficacy of this block saddle preconditioner,

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

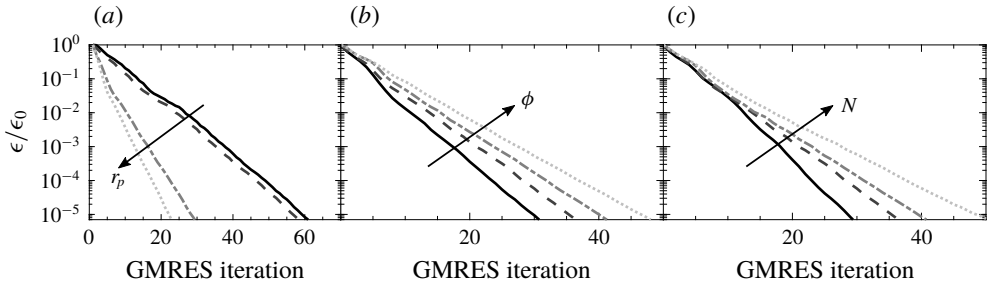


FIGURE 1. Convergence rate of the relative error in the iterative GMRES solution of the saddle point formulation for the hydrodynamic interactions. (*a*) Convergence rate dependence on the preconditioner cutoff radius (in units of particle radius) $r_p = \{2.0, 2.01, 2.1, 2.5\}$ for a random hard sphere dispersion with $N = 1000$ and $\phi = 0.2$. Although not shown on the plot, the convergence rate with $r_p = 3.0$ is identical to that for $r_p = 2.5$. (*b*) Convergence rate dependence on the volume fraction $\phi = \{0.1, 0.2, 0.3, 0.4\}$ for a random hard sphere dispersion with $N = 1000$ and $r_p = 2.1$. (*c*) Convergence rate dependence on the system size $N = \{10^2, 10^3, 10^4, 10^5\}$ for random hard sphere dispersions with $\phi = 0.2$ and $r_p = 2.1$.

equation (2.12), for iterative GMRES inversion of the saddle point matrix, equation (2.8), is shown in figure 1. Specifically, the convergence rate of the relative residual as a function of r_p, normalized by the particle radius a, is given in figure 1(*a*) for a random dispersion comprised of 1000 hard spheres at a volume fraction of $\phi = 0.2$. A relatively short cutoff radius, $r_p = 2.1$, is sufficient to drastically improve the convergence rate, and is almost as effective as larger cutoffs, $r_p = 2.5$ or $r_p = 3.0$. We choose $r_p = 2.1$ because it is very effective and requires minimal memory and operation counts to evaluate. It effectively only considers particles within the first neighbour shell. Within a distance of $r_p = 2.1$, monodisperse spheres can have at most 12 neighbours, so the memory required to construct the preconditioner is limited. For fixed r_p, the preconditioner performance is expected to deteriorate with increasing ϕ as particles gain more neighbours within r_{lub} that are not also within r_p. Figure 1(*b*) shows the convergence rate of the preconditioner with $r_p = 2.1$ across a range of volume fractions. The convergence rate decreases modestly with increasing volume fraction but the number of iterations remains acceptably small. For large scale simulations, the most important metric is the convergence rate as a function of system size. This is shown in figure 1(*c*) for $r_p = 2.1$ and $\phi = 0.2$. The convergence rate is effectively independent of system size, given that the number of iterations required to converge to a given tolerance increases by less than a factor of two when the number of particles is increased by three orders of magnitude.

4. Stochastic sampling for Brownian dynamics

4.1. *Far-field Brownian slip velocity*

The positively split Ewald (PSE) approach to sampling the far-field stochastic velocities has been described in great detail previously (Fiore *et al.* 2017; Fiore & Swan 2018), but will be summarized here for clarity. In the PSE approach, the Brownian velocity $\mathcal{U}^B \sim \mathcal{N}(\mathbf{0}, \mathcal{M})$ is decomposed into two components, corresponding to the real-space and wave-space contributions to the Ewald sum

$$\mathcal{U}^B = \mathcal{U}^B_r + \mathcal{U}^B_w, \tag{4.1}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where

$$\mathcal{U}_r^B \sim \mathcal{N}\left(\mathbf{0}, \frac{2k_BT}{\Delta t}\mathcal{M}_r\right), \quad \mathcal{U}_w^B \sim \mathcal{N}\left(\mathbf{0}, \frac{2k_BT}{\Delta t}\mathcal{M}_w\right). \tag{4.2a,b}$$

Samples from these two distributions are drawn using separate techniques tailored to their different operator structures.

The real-space contribution to the displacements is sampled from $\mathcal{M}_r$,

$$\mathcal{U}_r^B = \mathcal{M}_r^{1/2} \cdot \boldsymbol{\Psi}_r, \tag{4.3}$$

using the iterative Lanczos sampling technique developed by Chow & Saad (2014). The rate of convergence of the Lanczos iteration is controlled by the condition number of the sampled matrix. Because $\mathcal{M}_r$ contains exponentially decaying interactions evaluated only for particles within a small neighbourhood of each other, its condition number is both small and independent of the total number of particles modelled (Fiore *et al.* 2017; Fiore & Swan 2018). Therefore, the Lanczos iterates converge quickly for any number of particles.

The wave-space contribution to the displacements is sampled from $\mathcal{M}_w$ whose compact operator representation is,

$$\mathcal{M}_w = \mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B} \cdot \mathbb{P} \cdot \mathbb{D}, \tag{4.4}$$

which can be written as the product of a matrix and its adjoint

$$\mathcal{M}_w = (\mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B}^{1/2}) \cdot (\mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B}^{1/2})^\dagger. \tag{4.5}$$

The square root of a matrix $\boldsymbol{X}$ useful for stochastic sampling is defined as any matrix $\boldsymbol{X}^{1/2}$ such that $\boldsymbol{X}^{1/2} \cdot (\boldsymbol{X}^{1/2})^\dagger = \boldsymbol{X}$. Therefore, one possible (and very useful) square root of $\mathcal{M}_w$ is given analytically by

$$\mathcal{M}_w^{1/2} = \mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B}^{1/2}, \tag{4.6}$$

so that

$$\mathcal{U}_w^B = \mathbb{D}^\dagger \cdot \mathbb{P}^\dagger \cdot \mathbb{B}^{1/2} \cdot \boldsymbol{\Psi}_w, \tag{4.7}$$

which is no more challenging to compute than the product of the mobility tensor with a vector.

Asserting that $\boldsymbol{\Psi}_r$ and $\boldsymbol{\Psi}_w$ are statistically independent guarantees that $\mathcal{U}_r^B$ and $\mathcal{U}_w^B$ are also statistically independent. Furthermore, if $\boldsymbol{\Psi}_r$ and $\boldsymbol{\Psi}_w$ are computed such that $\boldsymbol{\Psi}_r \sim \boldsymbol{\Psi}_w \sim \mathcal{N}(\mathbf{0}, (2k_BT/\Delta t)\,\boldsymbol{I})$, then

$$\mathcal{U}_r^B + \mathcal{U}_w^B \sim \mathcal{N}\left(\mathbf{0}, \frac{2k_BT}{\Delta t}(\mathcal{M}_r + \mathcal{M}_w)\right), \tag{4.8}$$

as required by the fluctuation–dissipation theorem, where $\mathcal{M} = \mathcal{M}_r + \mathcal{M}_w$. It is important to note that this split sampling approach only works when $\mathcal{M}_r$ and $\mathcal{M}_w$ are independently positive semi-definite, which guarantees that they each have a well-defined square root. To our knowledge, the only representation of $\mathcal{M}_r$ and $\mathcal{M}_w$ for which this is true is the formulation we have presented previously (Fiore & Swan 2018) and reviewed in § 3.1.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

4.2. *Near-field Brownian force*

Banchio & Brady (2003) first developed a method to sample the near-field Brownian force in the context of ASD, avoiding the $O(N^3)$ cost for direct evaluation of the square root by individually computing the force for each set of neighbouring particles α and β,

$$\begin{bmatrix} (\boldsymbol{F}^B_{nf})^\alpha \\ (\boldsymbol{F}^B_{nf})^\beta \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \frac{2k_BT}{\Delta t} \begin{bmatrix} (\boldsymbol{R}^{nf}_{FU})_{\alpha\alpha} & (\boldsymbol{R}^{nf}_{FU})_{\alpha\beta} \\ (\boldsymbol{R}^{nf}_{FU})_{\beta\alpha} & (\boldsymbol{R}^{nf}_{FU})_{\beta\beta} \end{bmatrix} \right), \tag{4.9}$$

and showed that the total force vector is appropriately distributed, i.e. $\boldsymbol{F}^B_{nf} \sim \mathcal{N}(\boldsymbol{0},\, 2k_BT\,\boldsymbol{R}^{nf}_{FU})$. The sampling for each pair of particles within r_{lub} is evaluated using the complete lower Cholesky decomposition $\boldsymbol{L}_{\alpha\beta}$ of the pair resistance tensor

$$\begin{bmatrix} (\boldsymbol{F}^B_{nf})^\alpha \\ (\boldsymbol{F}^B_{nf})^\beta \end{bmatrix} = \boldsymbol{L}^{\alpha\beta} \cdot \begin{bmatrix} \boldsymbol{\Psi}^\alpha_{nf} \\ \boldsymbol{\Psi}^\beta_{nf} \end{bmatrix}, \tag{4.10}$$

where $\boldsymbol{\Psi}^\alpha_{nf}$ and $\boldsymbol{\Psi}^\beta_{nf}$ are uncorrelated and $\boldsymbol{\Psi}^\alpha_{nf} \sim \boldsymbol{\Psi}^\beta_{nf} \sim \mathcal{N}(\boldsymbol{0},\, (2k_BT/\Delta t)\,\boldsymbol{I})$. This approach is efficient from a global memory perspective, because only the pairwise resistance tensors need to be stored, but is inefficient from a performance perspective because each particle pair must be individually computed, requiring $O(N)$ Cholesky factorizations. Some parallelization of this scheme is possible at the cost of increased memory usage, but large scale parallelization and acceleration of the pair Cholesky calculation is infeasible on current computer hardware.

Instead, we choose to directly sample $\boldsymbol{F}^B_{nf}$ for all particles simultaneously from $\boldsymbol{R}^{nf}_{FU}$ using the iterative Lanczos scheme of Chow & Saad (2014). $\boldsymbol{R}^{nf}_{FU}$ is positive semi-definite, which guarantees that the square root exists and the resulting forces are real valued. Although the Lanczos scheme can be applied directly to sample Brownian forces, $\boldsymbol{R}^{nf}_{FU}$ is ill-conditioned, so effective preconditioning schemes are required to accelerate convergence. The preconditioner described in § 2.3 is particularly well suited to this problem.

Figure 2(*a*) shows how these preconditioners compress the spectral radius of the near-field resistance tensor. Specifically, as intended, the incomplete Cholesky factor removes the largest magnitude eigenvalues, while the diagonal preconditioner shifts all eigenvalues that are less than one to be clustered around one. The effect of this eigenspectrum compression on the condition number is shown in figure 2(*b*) for hard sphere suspensions of 1000 particles across a wide range of volume fractions. The preconditioners produce a well-conditioned operator for all volume fraction and the condition number of the preconditioned operator is independent of ϕ. Figure 2(*c*) shows the number of iterations required for the Krylov subspace sampling technique to converge to a step-norm relative error of 10^{-6}. With our preconditioning scheme, the iteration count is independent of the volume fraction, and is 3–15 times smaller than without preconditioning. Most impressively, figure 2(*d*) shows that the iteration count is effectively independent of the system size; increasing the number of particles of particles by a factor of 1000 only increases the number of iterations by 50 %.

4.3. *Brownian drift*

We use the random finite differencing (RFD) scheme of Delong *et al.* (2014) to compute the Brownian drift, $k_BT\nabla \cdot \boldsymbol{R}_{FU}$. This method was already clearly described

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

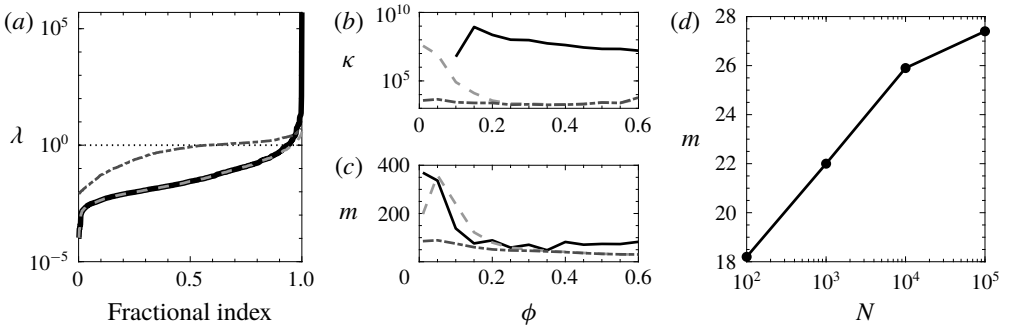


FIGURE 2. Effectiveness of the preconditioner for iterative Krylov subspace approximation to the square root of the near-field resistance tensor. (*a*) Eigenspectrum of the near-field lubrication tensor for a random suspension of $N = 1000$ hard spheres at $\phi = 0.2$ (solid black line), modified eigenspectrum after applying the incomplete Cholesky preconditioner (dashed light grey line, overlaps the black line for much of the range), modified eigenspectrum after applying both the incomplete Cholesky and diagonal preconditioners (dashed-dot medium grey line). The dotted black line denotes a uniform eigenspectrum with $\lambda_i = 1$, which would result from 'perfect' preconditioning. (*b*) Condition number of resistance tensor before and after preconditioning for random hard sphere suspension comprised of $N = 1000$ particles at various volume fractions. Same line definitions as in panel (*a*). (*c*) Number of iterations required for convergence of the method to a relative step-norm error of 10^{-6} for the hard sphere suspensions in panel (*b*). (*d*) Number of iterations required for convergence to a relative step-norm error of 10^{-3}, typical of many simulations, for the fully preconditioned sampling scheme applied to hard sphere dispersions with $\phi = 0.2$ at varying system sizes.

by Delong *et al.*, but will be summarized here for completeness. The idea of RFD is to approximate the divergence of a matrix $\boldsymbol{A}(\boldsymbol{q})$, which is a function of the set of particle coordinates $\boldsymbol{q}$, as the difference between the matrix evaluated at two nearby points. The central random finite difference is

$$\frac{\left\langle \boldsymbol{A}\left(\boldsymbol{q}+\frac{\epsilon}{2}\Delta\boldsymbol{q}\right)\cdot\Delta\boldsymbol{q}-\boldsymbol{A}\left(\boldsymbol{q}-\frac{\epsilon}{2}\Delta\boldsymbol{q}\right)\cdot\Delta\boldsymbol{q}\right\rangle}{\epsilon}=\boldsymbol{\nabla}\cdot\boldsymbol{A}+O(\epsilon^2), \tag{4.11}$$

where $\Delta\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, ϵ is a very small discretization parameter, and the finite difference approximates the divergence of $\boldsymbol{A}$ in expectation. Methods for choosing ϵ that maximize the accuracy of the RFD calculation are detailed by Sprinkle *et al.* (2017). This scheme applied to FSD nominally requires two saddle point solves, one at each of the displaced points. However, because the displacements are small, the particle positions are quite close and the solution at one set of coordinates serves as an excellent initial guess for the solution at the other set. The second solve converges very rapidly ($O(1)$ iterations are required in our tests, compared to $O(10)$ for the first solve), and therefore contributes a negligible amount to the overall computational cost of the algorithm.

Using RFD, the mean drift of the Brownian displacement and stresslet can be computed from saddle point solves,

$$\begin{bmatrix}\mathcal{M}(\boldsymbol{q}\pm\Delta\boldsymbol{q}) & \boldsymbol{B}\\ \boldsymbol{B}^{\mathrm{T}} & -\boldsymbol{R}_{FU}^{nf}(\boldsymbol{q}\pm\Delta\boldsymbol{q})\end{bmatrix}\cdot\begin{bmatrix}\mathcal{F}_{\pm}\\ \boldsymbol{U}_{\pm}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}\\ -\Delta\boldsymbol{q}\end{bmatrix}, \tag{4.12}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

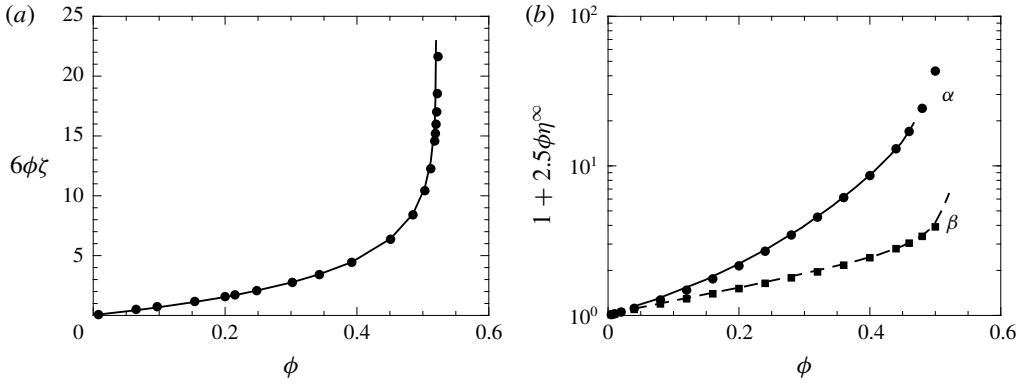


FIGURE 3. Transport coefficients for simple cubic lattices as a function of volume fraction. (*a*) Spin viscosity, ζ, computed using the FSD algorithm (circles) compared to ASD results of Sierou & Brady (2001) (line). (*b*) Independent components, α (circles) and β (squares), of the high-frequency shear viscosity, η^S. The solid and dashed lines are the values of α and β, respectively, computed by Brady *et al.* (1988).

where $\mathcal{F}_\pm$ and $\boldsymbol{U}_\pm$ are the solutions at the positive and negative displacements for the central difference. The far-field hydrodynamic stresslet is $\boldsymbol{S}^{ff}_\pm = [\boldsymbol{0} \quad \boldsymbol{I}] \cdot \mathcal{F}_\pm$ and the total stresslet is

$$\boldsymbol{S}_\pm = \boldsymbol{S}^{ff}_\pm + \boldsymbol{R}^{nf}_{SU}(\boldsymbol{q} \pm \Delta\boldsymbol{q}) \cdot \boldsymbol{U}_\pm. \tag{4.13}$$

The mean drift of the Brownian displacement is

$$\nabla \cdot \boldsymbol{R}^{-1}_{FU} = \frac{1}{\epsilon}\langle \boldsymbol{U}_+ - \boldsymbol{U}_- \rangle + O(\epsilon^2), \tag{4.14}$$

and the mean drift of the Brownian stresslet is

$$-\nabla \cdot (\boldsymbol{R}_{SU} \cdot \boldsymbol{R}^{-1}_{FU}) = \frac{1}{\epsilon}\langle \boldsymbol{S}_+ - \boldsymbol{S}_- \rangle + O(\epsilon^2). \tag{4.15}$$

5. Results and discussion

5.1. *Algorithm validation*

A specific implementation of the FSD algorithm for monodisperse spheres is given in appendix B. We validate our FSD algorithm against known static and dynamic results for monodisperse dispersions. Figure 3 shows the spin viscosity and high-frequency shear viscosity of simple cubic lattices of particles. The spin viscosity $\boldsymbol{\eta}^L$ for a cubic lattice defines the hydrodynamic torque $\boldsymbol{L}^H$ experienced by the particles as they rotate with angular velocity $\boldsymbol{\Omega}$,

$$\boldsymbol{L}^H = -\boldsymbol{\eta}^L \cdot \boldsymbol{\Omega}. \tag{5.1}$$

Because the cubic lattice is isotropic, the spin viscosity can be represented by a single scalar value,

$$\eta^L_{ij} = \zeta\,\delta_{ij}. \tag{5.2}$$

Similarly, the shear viscosity defines the relationship between the hydrodynamic stresslet $\boldsymbol{S}^H$ acquired by particles in response to a rate of strain $\boldsymbol{E}$,

$$\boldsymbol{S}^H = -\boldsymbol{\eta}^S : \boldsymbol{E}. \tag{5.3}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

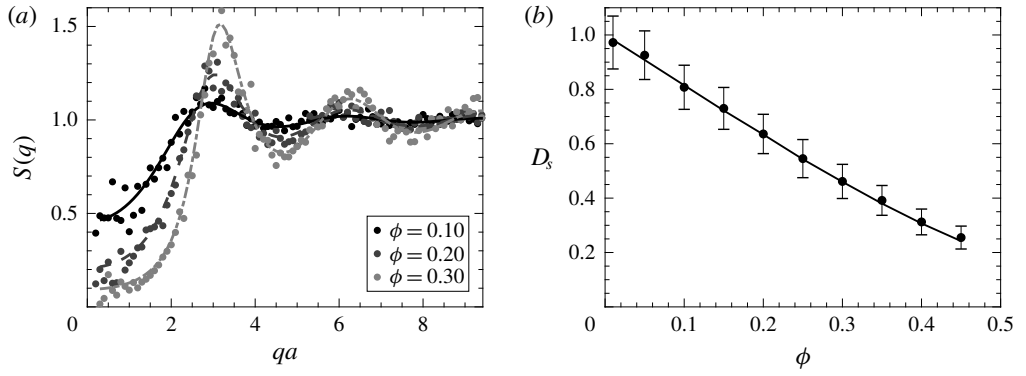


FIGURE 4. Equilibrium and transport characteristics for random hard sphere suspensions. (*a*) Static structure factor for $N = 1000$ particles computed from dynamic simulations using the FSD algorithm (lines) compared to the analytical Percus–Yevick result for hard spheres (Regnaut & Ravey 1989, 1990). (*b*) Short-time self-diffusion coefficient computed from the constrained stochastic displacements $D_s = (\Delta t/6k_BT)\langle \boldsymbol{U}^B \cdot \boldsymbol{U}^B \rangle$ generated by the FSD algorithm (circles) compared to the analytical results of Ladd (1990). Error bars represent standard deviations of the measurements.

Due to symmetry, the fourth-order tensor $\boldsymbol{\eta}^S$ describing the shear viscosity can be decomposed in terms of two scalar coefficients (Brady *et al.* 1988),

$$\eta^S_{ijkl} = \tfrac{1}{2}(1+\beta)\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \tfrac{2}{3}\delta_{ij}\delta_{kl}\right) + (\alpha-\beta)\left(\delta_{ij}\delta_{ik}\delta_{il}\delta_{jk}\delta_{jl}\delta_{kl} - \tfrac{1}{3}\delta_{ij}\delta_{kl}\right). \tag{5.4}$$

In general, these viscosity expressions require an ensemble average over all particles, but all particles are identical in a cubic lattice so the averages are omitted here. The scalar coefficients of the spin and shear viscosities were computed using a relative error tolerance of 10^{-3} in all approximations. The scalar spin viscosity ζ is plotted as a function of volume fraction in figure 3(*a*) and is compared with the values computed by Sierou & Brady (2001) using the accelerated Stokesian dynamics (ASD) algorithm. The two scalar components to the shear viscosity, α and β, are plotted as a function of volume fraction in figure 3(*b*) and are compared with the values obtained by Brady *et al.* (1988) using the original SD algorithm. In all cases, excellent quantitative agreement is observed between our FSD result and the standard Stokesian dynamics approaches.

To validate the proposed sampling and integration methods, we performed dynamic simulations of hard sphere particles with a relative error tolerance of 10^{-3} for all hydrodynamics calculations and a time step of $\Delta t = 10^{-3}$. To avoid unphysical overlap of the hard sphere particles, we used a short-ranged repulsive potential (Sierou & Brady 2001). We computed the structure factor of the resulting equilibrium microstructures from 100 configurations extracted from the particle trajectory every 10 bare particle diffusion times. Figure 4(*a*) shows the structure factor computed from our simulations compared to analytical results derived using the Percus–Yevick closure (Regnaut & Ravey 1989, 1990). The quantitative agreement between our simulations and the analytical theory indicates that the stochastic displacements are consistent with fluctuation–dissipation and produce the correct equilibrium particle configurations. Figure 4(*b*) shows the short-time self-diffusion coefficient computed

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

from the ensemble mean of the constrained Brownian velocity, $\boldsymbol{U}^B$,

$$D_s = \frac{1}{3}\frac{\Delta t}{2k_BT}\langle \boldsymbol{U}^B \cdot \boldsymbol{U}^B \rangle, \tag{5.5}$$

compared to the high-resolution calculations of Ladd (1990). Again, quantitative agreement is observed between our calculations and established literature results, indicating that the simulation produces the correct dynamic displacements.

5.2. *Algorithm performance*

The SD algorithm presented here, which we call fast Stokesian dynamics, is designed for efficient scaling and computations in large system. As detailed in § 4 and appendix B, each component of the FSD algorithm has a computational cost that scales linearly with the number of particles. The most time-consuming calculation is the saddle point inversion because no more effective preconditioners for solution of the stresslet constraint are known. Further acceleration to the algorithm could be achieved by the development of such a preconditioner, should one exist, and would reduce the number of iterations required for the solve. However, the improvement would be modest, as the solution is already evaluated in $O(10)$ iterations, as outlined in § 3.3. In this section, we describe the performance characteristics of the FSD algorithm and compare them to our previous work with simpler approximations of the HI.

The expected performance of the FSD algorithm can be estimated from the previously observed performance of our RPY code. This value will serve as a performance baseline. At the RPY level of approximation for the hydrodynamic interactions, the PSE algorithm has a peak performance of around 3×10^6 particle time steps per second (PTPS) (Fiore *et al.* 2017), which is independent of the system size. PTPS is equal to the total number of particles in the simulation divided by the time (in seconds) required to performed a single simulation time step's worth of calculations. The RPY performance can be used to estimate the optimal performance of the PSE algorithm including lubrication by comparing the operation counts between the two methods. At the RPY level of approximation, there are $O(3N)$ degrees of freedom associated with the three independent components of the hydrodynamic force on each particle, and a single matrix-vector multiply is required to evaluate the hydrodynamic interactions. Therefore, computing the RPY approximation for the hydrodynamics requires $O(3N)$ operations. In contrast, the FSD approximation includes $O(17N)$ degrees of freedom: $3N$ from the hydrodynamic force, $8N$ from the traceless hydrodynamic couplet, and $6N$ from the near-field hydrodynamic force and torque. Additionally, three solves of the saddle point problem need to be evaluated to compute the particle displacements: one solve for the deterministic contribution to particle motion, and two for the RFD approximation to the Brownian drift. However, as previously discussed, the solution of the first RFD solve can be used as the initial guess in the second solve to reduce the second solve to a negligible cost. Therefore, from an operation cost perspective, only two solves are effectively required. Each solve requires approximately 30 iterations regardless of system size, so the total operation count for the FSD algorithm is $O(17N \times 30 \times 2) = O(1020N)$, which is 340 times that of the RPY approximation. With this estimate, the expected peak performance of the FSD algorithm is 10^4 PTPS.

Figure 5 shows the performance of the PSE algorithm applied to dynamic simulations of hard sphere dispersions performed with a global relative error tolerance

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

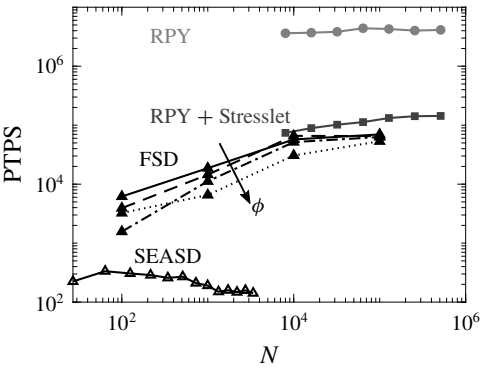


FIGURE 5. Performance of the PSE algorithm at three levels of hydrodynamic approximation: RPY (light grey circles), RPY plus stresslets (medium grey squares) and FSD (solid black triangles). Performance of the SEASD algorithm of Wang & Brady (2016) is shown in open black triangles. The calculations are performed with a relative error tolerance of 10^{-3}. The performance is measured in particle time steps per second, which is equal to the total number of particles divided by the time required to computed one time steps worth of calculations. The performance of the RPY and RPY+stresslet algorithms is independent of ϕ, so only one representative curve is shown. Because the FSD performance is a function of ϕ, the performance for $\phi = \{0.1, 0.2, 0.3, 0.4\}$ is computed separately and denoted by the solid, dashed, dashed-dot and dotted lines, respectively.

of 10^{-3}, which is typical in many applications (Wang & Brady 2016). Performance curves are shown for the FSD algorithm presented here, as well as for the RPY level of hydrodynamic approximation both with and without the stresslet constraints. In all cases, the computational cost includes the cost of performing the Brownian sampling. The RPY performance, with and without the stresslet constraint, has been previously published and was evaluated on an NVIDIA Tesla K40 graphics card (Fiore *et al.* 2017; Fiore & Swan 2018). The FSD performance presented here was evaluated on an NVIDIA GTX 980Ti graphics card. The GTX has a maximum memory bandwidth of 337 GB s^{-1}, compared to 288 GB s^{-1} for the Tesla, and a theoretical maximum performance of 6 teraflops for single precision calculations, compared to 5 teraflops for the Tesla. Whether the application is memory-bound or not, the GTX is expected to be approximately 20 % faster than the Tesla. In our case, we believe the computations are not memory bound. As previously shown, the performance of the RPY stresslet-constrained RPY algorithms is independent of particle volume, so only one curve is plotted for each of these methods.

In contrast to our previous work, the FSD algorithm displays a modest dependence on volume fraction, which is expected based on the convergence rates shown in figure 1(*b*). As shown in figure 5, there is a factor of 1.3 difference in performance between the $\phi = 0.1$ and $\phi = 0.4$ results with $N = 10^5$. The largest difference between volume fractions is a factor of 3 for $N = 10^3$. For the largest system sizes studied, $N = 10^5$, the FSD algorithm is about two times slower than RPY with stresslets. This is substantially better than what would be expected purely from an operation count perspective, even accounting for the difference in maximum theoretical performance between the GPUs. This discrepancy is likely due to the fact that the full computational bandwidth of the GPU is not being used by the RPY or

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

stresslet systems. We have previously noted that the PTPS measured with the RPY algorithm continues to increase up to 4×10^6 particles (Fiore *et al.* 2017), which is evidence of bandwidth underutilization. Furthermore, it is obvious in figure 5 that the throughput of the FSD simulation increases systematically from $N = 10^2$ to $N = 10^5$, which again indicates that more of the available GPU hardware is being utilized by the larger systems. Additionally, some minor improvements have been made to the far-field algorithms since the method for RPY with stresslet constraints was published, which may contribute slightly to the relative performance differences observed.

Nevertheless, when considered with the performance data for the RPY method, figure 5 clearly demonstrates that the algorithm implementation is optimal and exhibits linear scaling in the total computational complexity with the number of particles modelled. Our algorithm reaches 6×10^4 PTPS with $N = 10^5$ particles, which for the first time enables such large scale simulations of colloidal materials with full hydrodynamic interactions, including lubrication. This is made plain by comparing the relative performance of our FSD algorithm with the spectral Ewald accelerated Stokesian dynamics (SEASD) algorithm (Wang & Brady 2016), which also leverages GPUs, the non-uniform fast Fourier transform (NUFFT) acceleration of Lindbo & Tornberg (2010), and a modestly effective preconditioner for the far-field matrix inversion. The SEASD performance data were obtained for bidisperse suspensions of spheres, for which the performance is expected to be only slightly lower than the monodisperse case considered here. Wang & Brady (2016) report that their GPU implementation achieves an order of magnitude speedup over an equivalent CPU implementation, which means that the two orders of magnitude difference between FSD and SEASD shown in figure 5 are due primarily, if not exclusively, to the algorithmic improvements presented in this work (Wang & Brady used an NVIDIA GTX 680 GPU, which has a maximum memory bandwidth of 192 GB s^{-1} and a theoretical maximum performance of 3.25 teraflops for single precision calculations. Therefore, hardware differences amount to at most a factor of 2 difference between their performance and the FSD performance presented here). Specifically, the saddle point hydrodynamic formulation, which avoids nested iterative matrix inversions and for which highly effective preconditioners are developed, and the Brownian force decomposition, which leverages the PSE sampling technique in the far field and a novel preconditioner applied to Krylov subspace techniques in the near field, drastically improve calculation speeds. Furthermore, FSD maintains linear scaling of the computation cost up to 10^5 particles, while SEASD performance deteriorates for systems larger than 10^3 particles.

6. Stokesian dynamics with arbitrarily shaped particles

The flexibility of the saddle point formulation to incorporate mixed sets of constraints will be illustrated using an important class of hydrodynamic approximations known as composite bead assemblies or rigid multiblobs. In this class of approximation, the surface of an immersed colloidal particle is tessellated with hydrodynamically interacting beads and the motion of the composite particle is computed by solving the resulting immersed boundary problem subject to the constraint that beads belonging to the same particle move together as a rigid body (Balboa Usabiaga *et al.* 2016; Swan & Wang 2016). This approach is particularly useful for simulating heterogeneous particle mixtures (size and shape polydispersity), and also provides a convergent method to systematically improve the hydrodynamic approximation. The accuracy of the composite bead approximation is improved by increasing the number of beads

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

used to discretize the particle surface, which effectively accounts for higher-order multipole moments (i.e. more non-uniform surface traction) without requiring explicit derivation of the analytical couplings between them.

The composite particles are constructed by discretizing the particle surface using a set of vertices on which hydrodynamically interactions beads are placed. For example, a sphere's surface can be approximated by a Goldberg polyhedron (Goldberg 1937), or the surface of a generic shape can be approximated by a triangular mesh. The constraint that beads in the same particle move collectively as a rigid body is enforced by imposing a set of constraint forces $\boldsymbol{F}^c$ on the beads within each body that act as Lagrange multipliers. The mathematical formulation of the rigid body problem begins with the statement that the constraint forces produce no net force or torque on the composite body of which they are a part, i.e.

$$\boldsymbol{\Sigma} \cdot \boldsymbol{F}^c = \boldsymbol{0}, \tag{6.1}$$

where $\boldsymbol{\Sigma}$ is a tensor whose action sums the total force $\boldsymbol{F}$ on the beads to produce the resulting generalized force on the composite particle, its adjoint $\boldsymbol{\Sigma}^{\mathrm{T}}$ computes the bead velocities $\boldsymbol{U}$ from the generalized velocity $\mathbb{U}$ (translation and rotation) of the rigid body. Including the constraint force, the overdamped force balance on the beads is

$$\boldsymbol{F}^P + \boldsymbol{F}^B + \boldsymbol{F}^H + \boldsymbol{F}^c = \boldsymbol{0}, \tag{6.2}$$

where $\boldsymbol{F}^P$ are the externally imposed conservative forces, $\boldsymbol{F}^B$ are the set of Brownian forces and $\boldsymbol{F}^H$ are the hydrodynamic drag forces experienced by the beads. The equivalent force balance on the bodies is obtained by taking the product of $\boldsymbol{\Sigma}$ with (6.2)

$$\mathbb{F}^P + \mathbb{F}^B + \mathbb{F}^H = \boldsymbol{0}, \tag{6.3}$$

where $\mathbb{F} = \boldsymbol{\Sigma} \cdot \boldsymbol{F}$ and $\mathbb{F}^c = \boldsymbol{0}$.

In general, each bead interacts hydrodynamically with all other beads according to the mobility relation introduced in § 2, $\mathcal{M} \cdot \mathcal{F} + \mathcal{B} \cdot \boldsymbol{U} = \boldsymbol{0}$, where $\mathcal{F}$ is the set of far-field hydrodynamic force moments on the bead (force, torque, stresslet, etc.) and the generalized hydrodynamic force (force and torque) is given by $\boldsymbol{F}^H_{ff} = \mathcal{B}^{\mathrm{T}} \cdot \mathcal{F}$. Here, $\mathcal{B}^{\mathrm{T}}$ projects the generalized force (force and torque) associated with rigid body motion out of the general set of force moments, and its transpose maps rigid body motions onto the higher-order set of considered motions. In the traditional Stokesian dynamics formulation, the hydrodynamic force on the beads is decomposed into near-field and far-field contributions $\boldsymbol{F}^H = \boldsymbol{F}^H_{nf} + \boldsymbol{F}^H_{ff}$. This approach is taken by Meng & Higdon (2008*a*) to construct platelet-like particles as rigid bead assemblies, but a more general approach decomposes the hydrodynamic force on the particles into a near-field and far-field part, $\mathbb{F}^H = \mathbb{F}^H_{nf} + \mathbb{F}^H_{ff}$, and imposes a specific form for the near-field interactions between particles, not beads. The generic, as yet unspecified, near-field resistance between bodies is denoted $\mathbb{R}^{nf}_{FU}$. In this case, the force balance on the bodies is

$$\mathbb{F}^P + \mathbb{F}^B + \mathbb{F}^H_{nf} + \mathbb{F}^H_{ff} = \boldsymbol{0}, \tag{6.4}$$

where

$$\mathbb{F}^H_{nf} = -\mathbb{R}^{nf}_{FU} \cdot \mathbb{U}, \tag{6.5}$$

and

$$\mathbb{F}^H_{ff} = \boldsymbol{\Sigma} \cdot \mathcal{B}^{\mathrm{T}} \cdot \mathcal{F}. \tag{6.6}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

As in (2.8), the hydrodynamic mobility relation and (6.4)–(6.6) can be combined into a saddle point matrix

$$\begin{bmatrix} \mathcal{M} & \mathcal{B}\cdot\boldsymbol{\Sigma}^{\mathrm{T}} \\ \boldsymbol{\Sigma}\cdot\mathcal{B}^{\mathrm{T}} & -\mathbb{R}_{FU}^{nf} \end{bmatrix}\cdot\begin{bmatrix} \mathcal{F} \\ \mathbb{U} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -(\mathbb{F}^{P}+\mathbb{F}_{nf}^{B}+\mathbb{F}_{ff}^{B}) \end{bmatrix}. \tag{6.7}$$

Here, $\mathbb{F}_{nf}^{B}$ is the near-field Brownian force, $\mathbb{F}_{nf}^{B}\sim\mathcal{N}(\mathbf{0},\,(2k_BT/\Delta t)\mathbb{R}_{FU}^{nf})$, which can be quickly computed using the techniques described in § 4.2. Equation (6.7) is the specific case of (2.8) applied to composite bead particles. Balboa Usabiaga *et al.* (2016) developed a block diagonal preconditioner for fast simulations of so-called rigid multi-blobs, which are equivalent to the rigid bead assemblies considered here. Their preconditioner may be useful to accelerate the solution of (6.7). Meng & Higdon (2008*b*) compute the Brownian forces for rigid particle assemblies using different techniques for the far-field and near-field forces. The far-field Brownian forces are generated using a second-order rational function approximation, which, while faster than the standard Chebyshev polynomial approximation (Fixman 1986), is also less accurate. The near-field Brownian forces are computed by factorizing $\boldsymbol{R}_{FU}^{nf}$, presumably with a Cholesky factorization, and computing the unconstrained near-field forces using the Cholesky factor, then projecting the resulting forces onto the constrained manifold through the action of $\mathcal{B}^{\mathrm{T}}$. In other words, the forces are originally generated on the individual beads comprising each assembly as though the beads were free to move independently, then projected onto the set of allowed rigid body motions. Here we propose a more efficient sampling approach. In the same spirit as (2.20) and the rigid multi-blob work of Balboa Usabiaga *et al.* (2016) and Sprinkle *et al.* (2017), the far-field Brownian force can be replaced with a far-field slip velocity $\mathcal{U}^{B}$ giving $\mathbb{F}_{ff}^{B}=\boldsymbol{\Sigma}\cdot\mathcal{B}^{\mathrm{T}}\cdot\mathcal{M}^{-1}\cdot\mathcal{U}^{B}$, so the resulting saddle point equation is

$$\begin{bmatrix} \mathcal{M} & \mathcal{B}\cdot\boldsymbol{\Sigma}^{\mathrm{T}} \\ \boldsymbol{\Sigma}\cdot\mathcal{B}^{\mathrm{T}} & -\mathbb{R}_{FU}^{nf} \end{bmatrix}\cdot\begin{bmatrix} \hat{\mathcal{F}} \\ \mathbb{U} \end{bmatrix} = \begin{bmatrix} \mathcal{U}^{B} \\ -(\mathbb{F}^{P}+\mathbb{F}_{nf}^{B}) \end{bmatrix}, \tag{6.8}$$

where $\mathcal{U}^{B}$ is now a Brownian slip velocity on all the beads within the rigid bodies, and can be rapidly generated using the PSE algorithm as summarized in § 4.1, and $\hat{\mathcal{F}}=\mathcal{F}+\mathcal{M}^{-1}\cdot\mathcal{U}^{B}$.

Equation (6.8) represents a completely general equation for Stokesian dynamics simulations of arbitrarily shaped bodies comprised of rigid bead assemblies. Mixed sizes and shapes are allowed, and only enter the equation through modifications to $\boldsymbol{\Sigma}$. In the case where the specific near-field interactions are given by bead–bead lubrication, the near-field resistance tensor is given by $\mathbb{R}_{FU}^{nf}=\boldsymbol{\Sigma}\cdot\boldsymbol{R}_{FU}^{nf}\cdot\boldsymbol{\Sigma}^{\mathrm{T}}$. The associated Brownian force is computed by generating a random force on each particle, $\boldsymbol{F}_{nf}^{B}=(\boldsymbol{R}_{FU}^{nf})^{1/2}\cdot\boldsymbol{\Psi}$, where $\boldsymbol{\Psi}\sim\mathcal{N}(\mathbf{0},\,(2k_BT/\Delta t)\boldsymbol{I})$, then projecting that set of forces onto the constrained set of forces consistent with rigid body motion, $\mathbb{F}_{nf}^{B}=\boldsymbol{\Sigma}\cdot(\boldsymbol{R}_{FU}^{nf})^{1/2}\cdot\boldsymbol{\Psi}$. Figure 6 provides an illustration of the various approximations to the hydrodynamics of spherical particles using the techniques summarized here.

An important distinction between the physical meaning of the beads used in the general approach advocated here and the more specific approach taken by Meng & Higdon (2008*a*) bears additional discussion and illuminates some advantages of the proposed method. In the approach taken by Meng & Higdon (2008*a*), each bead is treated as a rigid object with lubricating interactions, i.e. the stresslet constraint associated with rigid motion is solved for each bead. In other words, the composite particles are meant to literally model assemblies of rigid spherical particles; one

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

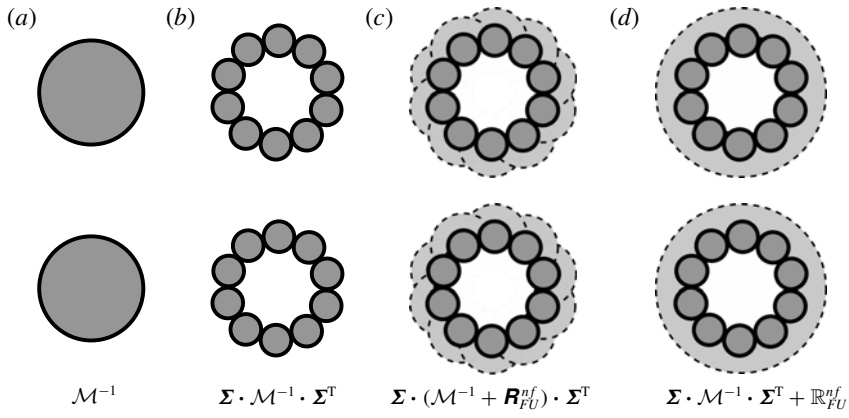


FIGURE 6. Schematic illustration for various approximations to the hydrodynamic resistance tensor when using composite bead particles. The bottom row indicates the mathematical representation for the resistance tensor describing the HI between the particles. (*a*) Individual beads whose interactions are described using the RPY or higher-order multipole expansion for the mobility, $\mathcal{M}$. (*b*) Particles represented using the composite bead model, where each bead propagates flows according to RPY or higher-order multipole approximations. (*c*) Same as in (*b*), with an additional near-field interaction between pairs of composite particles is constructed from the sum of the bead–bead near-field interactions between beads in different bodies. (*d*) Same as in (*b*), with the near-field interaction between pairs of composite particles represented by coating the particles in lubricating sheets, i.e. body–body near-field interactions rather than bead–bead as in (*c*).

can think of gluing hard spheres together to achieve a desired shape. This may not be the desired hydrodynamic model in more general circumstances. A schematic illustration of different composite bead approaches is shown in figure 6. A simpler computational method can be developed by noting that the flow field around a rigid particle can be expressed solely in terms of a single layer potential (Kim & Karrila 2005). This means that the force dipole (torque and stresslet) on the beads (but not on the particles) is irrelevant to the particle dynamics and can be neglected, and the beads can be allowed to interact with one another through the simple RPY approximation. This approach has been previously demonstrated by Swan & Wang (2016) and Balboa Usabiaga *et al.* (2016). In this way, each bead can be thought of as a regularized point force, and the collection of beads within each particle an approximation to the non-uniform force distribution on its surface. Adding lubrication between particles (not beads) improves the effective resolution of the force distribution without resorting to finer discretization. In the results presented in this section, we use this simpler approach and set $\mathcal{M}=\mathcal{M}_{RPY}$ everywhere. Using the RPY approximation includes the additional simplification that $\mathcal{B}=\boldsymbol{I}$. The rigid body constraint implicity captures dipoles, quadrupoles, and higher-order multipole force moments by producing a non-uniform distribution of hydrodynamic bead forces on each particle. Sequential resolution of higher-order moments is achieved by increasing the number of beads used to discretize each particle. This approach has an added benefit of reducing the algorithm's computational cost. With the force dipole and lubrication interactions considered at the bead–bead level, there are 17 degrees of freedom per bead, or $17N_bN_p$ total degrees of freedom, where N_b is the number

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

of beads per particle and N_p is the total number of particles. When only RPY interactions are considered between beads, and lubrication interactions are considered between particles, there are only 3 degrees of freedom per bead and 6 per particle, i.e. $(3N_b + 6)N_p$ in total. Therefore, the speedup expected by using particle–particle lubrication is $17N_b/(3N_b + 6) \approx 17/3$ for modestly large N_b ($N_b \gtrsim 10$), which are typical in many simulations.

Next, the rigid body construction with body–body near-field lubrication interactions (figure 6*d*) will be combined with the stochastic sampling schemes described in § 4 to produce accurate and physically consistent simulations. The effect of increasing resolution and adding lubrication on the convergence of the hydrodynamic approximation is also illustrated.

Some minor modifications to the near-field calculation described in § 3 are required here and are detailed for clarity. In the traditional Stokesian dynamics approach, the lubrication interactions are computed by

$$\boldsymbol{R}_{2B}^{nf} = \boldsymbol{R}_{2B} - \mathcal{R}_{2B}, \tag{6.9}$$

where $\boldsymbol{R}_{2B}^{nf}$ is the pair contribution to the near-field resistance tensor, $\boldsymbol{R}_{2B}$ is the exact lubrication expression for a pair of spherical particles, typically computed using the expressions given by Jeffrey & Onishi (1984), and $\mathcal{R}_{2B}$ is the far-field contribution to the pair resistance tensor. When the particles are represented in the far field by a collection of RPY beads, the lubrication tensor between two composite bead assemblies is built by

$$\mathbb{R}_{2B}^{nf} = \boldsymbol{R}_{2B} - \boldsymbol{\Sigma}_{2B} \cdot \mathcal{M}_{2B}^{-1} \cdot \boldsymbol{\Sigma}_{2B}^{\mathrm{T}}, \tag{6.10}$$

where $\mathcal{M}_{2B}$ is the mobility tensor for the beads comprising both rigid bodies and $\boldsymbol{\Sigma}_{2B}$ contains the rigid body tensor for the two bodies in the pair, and the radii used in the expressions for $\boldsymbol{R}_{2B}$ are the hydrodynamic radius of the composite particle. For spherical particles, it is obvious that the hydrodynamic radius corresponds to the effective radius computed from the translational diffusivity (Swan & Wang 2016), but for other shapes the relevant length scale is less clear. In other words, the far-field contribution to the pair resistance tensor is computed by solving the hydrodynamic rigid body problem for that particle pair. It is necessary to choose approximations for the tensors in (6.10) that result in a positive semi-definite matrix. For particles with a finite radius of curvature, there will always be a cutoff distance within which the lubrication interactions dominate the far-field interactions and a positive-definite $\mathbb{R}_{2B}^{nf}$ is guaranteed.

The PSE technique is used to generate the far-field slip velocity in (6.8) and the preconditioned near-field sampling scheme described in § 4 is used to sample $\mathbb{F}_{nf}^{B}$ from $\mathbb{R}^{nf}$. Finally, random finite differences, which have previously been successfully applied to Brownian dynamics simulations of rigid bodies (Delong, Balboa Usabiaga & Donev 2015), are used to compute the Brownian drift and are combined with an Euler–Maruyama integration scheme. While previous efforts have developed RFD-free integration schemes to implicitly account for the thermal drift in pure mobility (Delmotte & Keaveny 2015; Sprinkle *et al.* 2017; Fiore & Swan 2018) and resistance (Banchio & Brady 2003) problems, it is substantially more challenging to develop such integrators for the mixed problem considered here, where the near-field stochastic quantity is a force and far-field stochastic quantity is a velocity. This representation

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

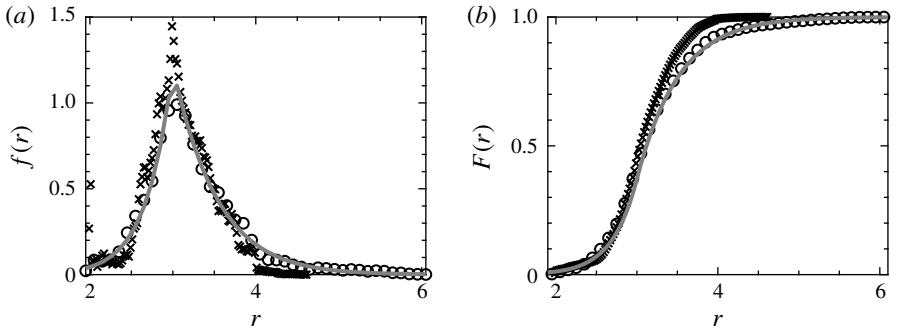


FIGURE 7. Probability functions (symbols) for the centre-to-centre distance of two particles connected by a linear spring with spring constant $k/k_BT = 2.5$ and equilibrium separation $r_0 = 3a$ compared to the theoretical Boltzmann distribution (lines). Circles and crosses represent calculations with and without the thermal drift included, respectively. (*a*) Probability density function. (*b*) Cumulative distribution function. A short-ranged repulsive interaction is added to prevent particle overlap.

of the noise is necessary for performance but not easily reconciled with the RFD-free methods.

A simple test of the proposed integration scheme is to connect two particles with a spring potential, $U(r)$, which varies with the interparticle separation, then compute the distribution of the observed centre-to-centre distance r between the particles. This is a useful test because the Boltzmann distribution provides an analytical result for the equilibrium probability density of the centre-to-centre distance $f(r)$,

$$f(r) = C_f r^2 \mathrm{e}^{-U(r)/k_BT}, \tag{6.11}$$

where the probability of a given state i with energy U_i is e^{-U_i/k_BT} and the number of available states at a given distance r is proportional to r^2. The constant C_f ensures that the total probability density integrates to one, $C_f = \int_0^\infty \mathrm{d}r r^2 \mathrm{e}^{-U(r)/k_BT}$. The associated cumulative distribution function, $F(r)$, is defined as

$$F(r) = \int_0^r \mathrm{d}x f(x). \tag{6.12}$$

These distributions were computed for a pair of particles, each comprised of 12 beads, interacting via a constant force spring for which $U(r)/k_BT = k|r - r_0|$ (with the parameters $k = 2.5$ and $r_0 = 3a$) by integrating their trajectories over 100 bare particle diffusion times. The results are shown in figure 7 and agree quantitatively with the analytical results in (6.11) and (6.12). If fluctuation–dissipation were not satisfied or if the thermal drift were incorrectly computed, then the equilibrium distribution would not be obtained and the distributions presented in figure 7, and would have noticeable error in either the variance or mean. The fact that the simulated distributions are correct indicate that the decomposition of the HI into near-field and far-field contributions and the associated decomposed stochastic sampling techniques produces stochastic displacements that are consistent with the fluctuation–dissipation theorem, and which correctly account for the thermal drift.

The distributions presented in figure 6 only test the equilibrium statistics of the stochastic integration scheme, and therefore do not probe the effect of bead resolution

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

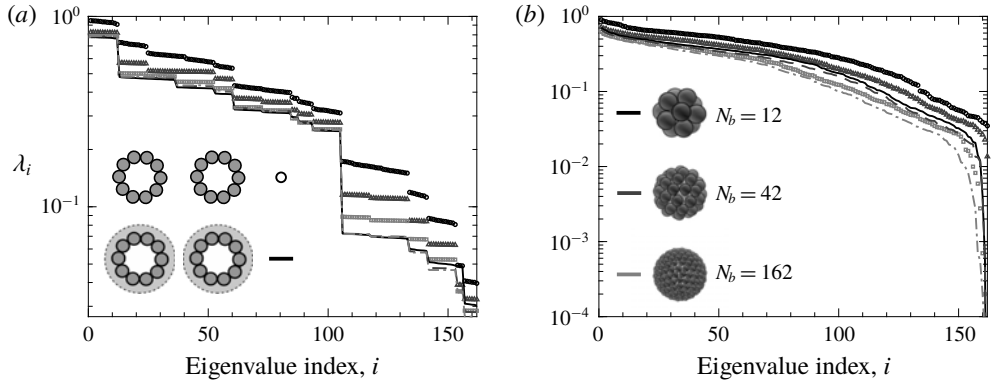


FIGURE 8. Distribution of eigenvalues of the constrained mobility tensor, $\mathbb{R}_{FU}^{-1}$, computed from the constrained saddle point solution, for composite bead particles with (lines) and without (symbols) particle–particle lubrication interactions. Particles are represented by composite beads comprised of $N_B = 12, 41, 162$ beads, denoted by black, dark grey and light grey, respectively. Two types of configurations are used, both comprised of $N = 27$ particles at $\phi = 0.9\pi/6$: (*a*) simple cubic crystal and (*b*) random hard sphere fluid.

or lubrication on the dynamics. Increasing the number of beads used to tesselate a particle's surface improves the approximation to the hydrodynamic interactions and increases the hydrodynamic resistance, especially between nearly touching particles. The lubrication corrections introduced in Stokesian dynamics also improve the approximation because they are designed to precisely capture the pair contribution to those near-field interactions. Therefore, by including lubrication with the composite bead approach, it would be expected that the number of tessellating beads required to represent the HI at a given level of accuracy would be lower than if the lubrication terms were not included. To quantify this effect, we computed the eigenspectrum of the constrained mobility tensor for spherical particles, $\mathbb{R}^{-1}$ (whose action is inferred by solving the saddle point problem for rigid body motion, equation (6.8)), at various levels of surface discretization, i.e. different numbers of beads per particle N_b, both with and without the near-field particle–particle lubrication interactions.

Figure 8(*a*) shows the eigenspectrum of $\mathbb{R}_{FU}^{-1}$, computed from the constrained saddle point solve, for a periodic simple cubic lattice comprised of 27 particles per unit cell at a volume fraction of $\phi = (0.9)\pi/6$, where $\phi = \pi/6$ is the closest-packing volume fraction for the simple cubic lattice. Because the surface of each particle is approximated by a discrete set of beads, a secondary lattice comprised of the beads is superimposed over the lattice comprised of the particle centres of mass and is unavoidable. To minimize the effect of this secondary lattice on the calculation, each composite particle is rotated randomly about its centre of mass so that the bead coordinates on any two particles are uncorrelated. For perfectly spherical particles, the eigenvalues exist at discrete levels due to lattice symmetry. The secondary bead lattice effects causes these discrete levels to smear out, a phenomena which is most noticeable at coarse discretizations, particularly $N_b = 12$. When particle–particle lubrication is neglected, increasing N_b shifts the eigenspectra down to smaller eigenvalues. When lubrication is accounted for, the eigenspectra for all resolutions collapse onto a single curve. The exception is with the eigenvalues whose indices are above 140, which do not collapse to the same value for different

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

resolutions even when lubrication is added. These eigenvalues are associated with sedimentation modes of the lattice, for which it is known that pairwise lubrication does not improve calculation accuracy because there is no relative motion between particles sedimenting in a lattice. For all modes except the sedimentation mode, a simple 12-bead surface discretization with particle–particle lubrication interactions is more accurate than a 162-bead surface tesselation without lubrication. The effect of discretization on transport properties computed in suspensions of adhesive hard spheres has been considered previously in some specific applications (Fiore, Wang & Swan 2018; Wang, Fiore & Swan 2019), but a general and systematic study of suspension transport properties computed using composite bead assemblies appears to be absent from the literature.

Part of the dramatic collapse of the eigenspectra with pair lubrication is due to the symmetry of the simple cubic lattice. For a random hard sphere fluid with the same number of particles and volume fraction, shown in figure 8(*b*), it's clear that adding the pairwise lubrication approximation does substantially improve results even at low-order surface discretizations, but the collapse of all the spectra with the addition of lubrication that was observed for the crystal is no longer seen, and improvements in accuracy are still made by increasing N_b. Without the symmetry of the lattice, high-order many-body interactions need to be resolved to compute the full dynamic spectrum. The only way to achieve this using the composite bead method is to increase the particle resolution. Resolving the many-body effect of lubricating flows, for example using the method proposed by Lefebvre-Lepot, Merlet & Nguyen (2015), might lead to the same eigenspectra collapse as was observed in the lattice, enabling very high fidelity simulations at small N_b. (The random hard sphere fluid configuration used in the calculation is included in the supplementary material, available at https://doi.org/10.1017/jfm.2019.640, so that eigenspectra computed using various methods can be directly compared.) Nevertheless, and importantly for fast simulations, the pairwise lubrication approximation qualitatively captures the diminutive eigenvalues associated with very nearly touching particles even for low-resolution discretizations. Without lubrication, many surface elements are needed to resolve these eigenvalues, therefore including particle–particle lubrication dramatically improves the simulation accuracy for a marginal increase in computational cost. As demonstrated in §§ 3.3 and 4.2, the total computational cost of the proposed simulation method with Brownian dynamics is linear in the total number of hydrodynamic objects, in this case beads. As each bead has three associated degrees of freedom the simulation cost without lubrication is $O(3N_pN_b)$ for N_p total particles. Introducing particle–particle lubrication increases the cost to $O((3N_b + 6)N_p)$ because lubrication adds six degrees of freedom per particle. At the lowest bead resolutions used here, $N_b = 12$, the simulation cost without lubrication is $O(36N_p)$ and $O(42N_p)$ with it. The 12-bead model with lubrication produces results that are qualitatively similar to the 162-bead model without lubrication, for which the cost is $O(486N_p)$. The results in figure 8(*b*), combined with these simple estimates, show that a 12-bead model with particle–particle lubrication maintains quantitatively similar dynamics to the 162-bead model, but is an order of magnitude faster to compute.

7. Conclusion

Our recently developed PSE algorithm for rapid Brownian dynamics simulations with HI represented at the RPY levels of approximations with stresslet constraints has been extended to incorporate the near-field lubrication correction required for

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

Stokesian dynamics. The hydrodynamic approximation is reformulated in a generic saddle point matrix, which encompasses many existing hydrodynamic approximations for colloidal particles. Critically, the saddle point formulation obviates the need to solve a resistance problem, i.e. invert the hydrodynamic mobility tensor, which is an ill-conditioned problem. Combined with efficient preconditioners based on the Schur complement of the hydrodynamic saddle point matrix, the saddle point matrix solve exhibits a linear computational complexity in the total number of particles modelled. We have also shown that the stochastic Brownian force, which is traditionally decomposed into near-field and far-field contributions, is more effectively computed when decomposed into the sum of a near-field force and a far-field slip velocity. The slip velocity can be readily computed using fast techniques such as the PSE algorithm and the near-field Brownian force is computed with a preconditioned Krylov subspace approximation for the square root of the resistance tensor containing the near-field lubrication interactions of all particles. The preconditioner used for this scheme can be recycled from the saddle point solve. In contrast to the standard approach for Stokesian dynamics with Brownian forces, this approach is readily parallelizable and is therefore extensible to large systems. The resulting algorithm, which we call fast Stokesian dynamics (FSD), has linear overall computational complexity and is nearly as fast as the RPY algorithm with stresslet constraints. The speed of the algorithm enables dynamics simulations of $O(10^5)$ particles in reasonable wall times on a single consumer-grade GPU.

We believe that the saddle point formulation presented here is a very general representation for constrained problems in colloidal hydrodynamics. This formulation is especially powerful because it enables the pairing of arbitrary near-field and far-field HI approximations with a general procedure for generating stochastic velocities consistent with a given set of constraints.

Acknowledgements

J.W.S. and A.M.F. gratefully acknowledge funding from the MIT Energy Initiative Shell Seed Fund and NSF Career Award CBET-1554398.

Supplementary material

Supplementary material is available at https://doi.org/10.1017/jfm.2019.640.

Appendix A. Polydispersity in Rotne–Prager-like models of hydrodynamics in dispersions of spherical particles

For polydisperse suspensions the effect of differing sizes on the hydrodynamic interactions can be included directly in the far-field mobility shape factor and the near-field resistance tensor expressions. The near-field lubrication series can be used without modification because the formulae given by Jeffrey & Onishi (1984) and Jeffrey (1992*b*) are already defined for unequal particle size ratios.

The far-field mobility couplings modified to account for polydispersity begin with the Fourier-space representation for the pair mobility coupling between two particles of unequal sizes a and b is

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

$$\boldsymbol{M}^{\alpha\beta}_{pq} = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(kb). \tag{A 1}$$

The different sizes enter the problem through the shape factors, $\boldsymbol{s}$. After applying Hasimoto's splitting function, the real-space and wave-space contributions are given by

$$(\boldsymbol{M}^{\alpha\beta}_{pq})_r = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}(1-H(k,\xi))\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(kb), \tag{A 2}$$

$$(\boldsymbol{M}^{\alpha\beta}_{pq})_w = \frac{1}{\eta V}\sum_{\boldsymbol{k}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\alpha-\boldsymbol{x}_\beta)}\frac{1}{k^2}H(k,\xi)\boldsymbol{s}^p(ka)(\boldsymbol{I}-\hat{\boldsymbol{k}}\hat{\boldsymbol{k}})\boldsymbol{s}^q(kb). \tag{A 3}$$

A.1. *Real-space sum*

As before, the real-space terms are expressed as the product of scalar functions of distance, and now particle sizes, and a minimal set of tensors, derived using appropriate symmetry and tracelessness conditions. The scalar coefficients are computed by analytically evaluating the integral corresponding to the inverse Fourier transform in (A 2). The integration can be split into two parts: an integral over the solid angle and an integral over the wavenumber. The solid angle integrals give the tensorial structure of each mobility relation and are straightforward to compute because they only involve exponentials and trigonometric functions. The wavenumber integral is much more challenging to compute and results in the complicated closed form expressions given previously for monodisperse suspensions (Fiore & Swan 2018). Although the expressions are valid for all particle separations, the integrals are simpler to evaluate, and the resulting expressions are more numerically stable, if the integrals are evaluated independently for each of five cases,

$$r \geqslant a+b, \tag{A 4}$$

$$a-b<b-a<r<a+b, \tag{A 5}$$

$$0<a-b<r<b-a, \tag{A 6}$$

$$0<b-a<r<a-b, \tag{A 7}$$

$$0=r. \tag{A 8}$$

These cases correspond to widely separated particles, partially overlapping particles, and the smaller particle enveloped by the larger particle. All the integrals presented below produce scalar mobility functions that are of the form

$$\begin{aligned} f^{(i)} = {} & f_0^{(i)} + f_1^{(i)}\mathrm{e}^{-(a+b+r)^2\xi^2} + f_2^{(i)}\mathrm{e}^{-(a+b-r)^2\xi^2} + f_3^{(i)}\mathrm{e}^{-(-a+b+r)^2\xi^2} \\ & + f_4^{(i)}\mathrm{e}^{-(a-b+r)^2\xi^2} + f_5^{(i)}\,\mathrm{erfc}((-a-b+r)\xi) + f_6^{(i)}\,\mathrm{erfc}((a-b+r)\xi) \\ & + f_7^{(i)}\,\mathrm{erfc}((-a+b+r)\xi) + f_8^{(i)}\,\mathrm{erfc}((a+b+r)\xi). \end{aligned} \tag{A 9}$$

The functions $f_j^{(i)}$ are various combinations of products, sums, differences, and quotients of a, b, ξ and r, and various powers thereof. Their tabulation would take many pages and are omitted here for brevity, but are included in the supplementary material. In the following sections, we show how to construct the integrals from which these coefficients can be computed.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

A.1.1. *Translation-force coupling*

The pair mobility tensor for the translation-force coupling is a function only of the unit vector between the two particle centres, $\hat{\boldsymbol{r}}$, the magnitude of the particle centre-to-centre separation, r, the radii of the two particles, a and b, and the Ewald splitting parameter ξ. The translation-force coupling, $\boldsymbol{M}_{UF}$ is a rank-2 tensor and is constructed in a minimal form as the sum of two orthogonal basis sets,

$$M_{ij}^{UF} = f_1(r, a, b, \xi)(\delta_{ij} - \hat{r}_i\hat{r}_j) + f_2(r, a, b, \xi)\hat{r}_i\hat{r}_j. \tag{A 10}$$

The scalar functions f_1 and f_2 are computed by evaluating the tensor products with the integral corresponding to the inverse Fourier transform of the wave-space representation for the real-space contribution to the Ewald sum of M_{ij}^{UF}. This representation is

$$M_{ij}^{UF} = -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M_q^{UF}(\delta_{ij} - \hat{q}_i\hat{q}_j), \tag{A 11}$$

where

$$M_q^{UF} = \frac{1}{q^2}\mathrm{j}_0(qa)\mathrm{j}_0(qb)H(q\xi^{-1}) \tag{A 12}$$

is the scale factor at wavenumber q, j_m is the mth order spherical Bessel function of the first kind and $H(x) = 1 - (1 + x^2/4)\mathrm{e}^{-x^2/4}$.

The first integral is the inverse Fourier transform of $M_{ij}^{UF}(\delta_{ij} - \hat{r}_i\hat{r}_j)$, and its evaluation will be used to illustrate how the more complicated integrals later in this appendix are performed,

$$\begin{aligned} f_1 &= -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M_q^{UF}(\delta_{ij} - \hat{q}_i\hat{q}_j)(\delta_{ij} - \hat{r}_i\hat{r}_j), \\ &= -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M_q^{UF}(\delta_{ij}\delta_{ij} - \delta_{ij}\hat{q}_i\hat{q}_j - \delta_{ij}\hat{r}_i\hat{r}_j + \hat{q}_i\hat{q}_j\hat{r}_i\hat{r}_j), \\ &= -\frac{1}{(2\pi)^3\mu}\int \mathrm{d}\boldsymbol{q}\mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M_q^{UF}(\delta_{ij}\delta_{ij} - \delta_{ij}\hat{q}_i\hat{q}_j - \delta_{ij}\hat{r}_i\hat{r}_j + \hat{q}_i\hat{q}_j\hat{r}_i\hat{r}_j). \end{aligned} \tag{A 13}$$

To evaluate these tensor products, the distance vector $\hat{\boldsymbol{r}}$ can be aligned along the z-axis in the spherical coordinates of integration, so that $\hat{\boldsymbol{q}}\cdot\hat{\boldsymbol{r}} = \cos\theta$, where θ is the polar angle,

$$\begin{aligned} f_1 &= -\frac{1}{(2\pi)^3\mu}\int \mathrm{d}\boldsymbol{q}\mathrm{e}^{\mathrm{i}qr\cos\theta} M_q^{UF}(1 + \cos^2\theta), \\ &= -\frac{1}{(2\pi)^3\mu}\int \mathrm{d}q q^2 M_q^{UF}\int_{\Omega}\mathrm{d}\Omega\mathrm{e}^{\mathrm{i}qr\cos\theta}(1 + \cos^2\theta), \end{aligned} \tag{A 14}$$

where Ω represents the integral over the solid angle, $\int_\Omega \mathrm{d}\Omega = \int_0^\pi \mathrm{d}\theta\ \sin\theta\int_0^{2\pi}\mathrm{d}\phi$. Because none of the parameters depend on the azimuthal angle, ϕ, that integral only contributions a multiplicative factor of 2π so that

$$f_1 = -\frac{1}{(2\pi)^2\mu}\int \mathrm{d}q q^2 M_q^{UF}\int_0^\pi \mathrm{d}\theta\,\sin\theta(1 + \cos^2\theta)\mathrm{e}^{\mathrm{i}qr\cos\theta}. \tag{A 15}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

The second integral is the inverse Fourier transform of $M^{UF}_{ij}\,\hat{r}_i\hat{r}_j$

$$\begin{aligned} f_2 &= -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{UF}_q(\delta_{ij}-\hat{q}_i\hat{q}_j)\hat{r}_i\hat{r}_j, \\ &= -\frac{1}{(2\pi)^3\mu}\int \mathrm{d}q q^2 M^{UF}_q \int_\Omega \mathrm{d}\Omega \mathrm{e}^{\mathrm{i}qr\cos\theta}(1-\cos^2\theta), \\ &= -\frac{1}{(2\pi)^2\mu}\int \mathrm{d}q q^2 M^{UF}_q \int_0^\pi \mathrm{d}\theta \sin^3\theta \mathrm{e}^{\mathrm{i}qr\cos\theta}. \end{aligned} \tag{A 16}$$

A.1.2. *Minimal tensor representation of translation-dipole coupling*

The translation-dipole coupling can be written using two scalar mobility functions,

$$M^{UC}_{imn} = g_1(\delta_{im}\hat{r}_n - \hat{r}_i\hat{r}_m\hat{r}_n) + g_2(\delta_{in}\hat{r}_m + \delta_{mn}\hat{r}_i - 4\hat{r}_i\hat{r}_m\hat{r}_n). \tag{A 17}$$

From a general expression for M^{UC}_{imn}, the scalar functions g_1 and g_2 can be isolated by evaluating the inner product of the mobility tensor with the appropriate orthogonal tensors. The products of M^{UC}_{imn} with these tensors are

$$M^{UC}_{imn}\hat{r}_i\hat{r}_m\hat{r}_n = -2g_2 \tag{A 18}$$

$$M^{UC}_{imn}(\delta_{im}-\hat{r}_i\hat{r}_m)\hat{r}_n = 2g_1. \tag{A 19}$$

The coefficients g_1 and g_2 will be determined by evaluating the independent integral corresponding to the inverse Fourier transform of the wave-space representation of real-space contribution to M^{UC}_{imn}.

The wave-space representation of the real-space contribution to the Ewald sum of M^{UC}_{imn} is

$$M^{UC}_{imn} = -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{UC}_q(\delta_{im}-\hat{q}_i\hat{q}_m)\hat{q}_n, \tag{A 20}$$

where

$$M^{UC}_q = \frac{1}{q^2}\mathrm{j}_0(qa)\left(\frac{3\mathrm{i}}{b}\mathrm{j}_1(qb)\right)H(q\xi^{-1}) \tag{A 21}$$

is the scale factor at wavenumber q.

The integral for g_1 is the inverse Fourier transform of $M^{UC}_{imn}(\delta_{im}-\hat{r}_i\hat{r}_m)\hat{r}_n$

$$\begin{aligned} g_1 &= -\frac{1}{2}\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{UC}_q(\delta_{im}-\hat{q}_i\hat{q}_m)\hat{q}_n(\delta_{im}-\hat{r}_i\hat{r}_m)\hat{r}_n, \\ &= -\frac{1}{2}\frac{(2\pi)^{-3}}{\mu}\int_0^\infty \mathrm{d}q q^2 \int_\Omega \mathrm{d}\Omega \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{UC}_q(1+\cos^2\theta)\cos\theta, \\ &= -\frac{1}{2}\frac{(2\pi)^{-2}}{\mu}\int_0^\infty \mathrm{d}q q^2 M^{UC}_q \int_0^\pi \mathrm{d}\theta \sin\theta(1+\cos^2\theta)\cos\theta \mathrm{e}^{\mathrm{i}qr\cos\theta}. \end{aligned} \tag{A 22}$$

The integral for g_2 is the inverse Fourier transform of $M^{UC}_{imn}\hat{r}_i\hat{r}_m\hat{r}_n$

$$\begin{aligned} g_2 &= \frac{1}{2}\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{UC}_q(\delta_{im}-\hat{q}_i\hat{q}_m)\hat{q}_n\hat{r}_i\hat{r}_m\hat{r}_n \\ &= \frac{1}{2}\frac{(2\pi)^{-2}}{\mu}\int_0^\infty \mathrm{d}q q^2 M^{UC}_q \int_0^\pi \mathrm{d}\theta \sin^3\theta\cos\theta \mathrm{e}^{\mathrm{i}qr\cos\theta}. \end{aligned} \tag{A 23}$$

Note that the gradient-force coupling for particle pair α, β can be obtained from the velocity-dipole coupling through symmetry relations, i.e. $M^{UC}_{imn}(\alpha,\beta) = M^{DF}_{nmi}(\beta,\alpha)$.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

A.1.3. *Minimal tensor representation of gradient-dipole coupling*

The pair mobility tensor for the gradient-dipole coupling can be written using three scalar mobility functions,

$$\begin{aligned} M^{DC}_{ijkl} &= h_1(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} - 4\delta_{il}\delta_{jk}) + h_2(\delta_{jk}\hat{r}_i\hat{r}_l - \hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l) \\ &\quad + h_3(\delta_{ij}\hat{r}_k\hat{r}_l + \delta_{ik}\hat{r}_j\hat{r}_l + \delta_{jl}\hat{r}_i\hat{r}_k + \delta_{kl}\hat{r}_i\hat{r}_j + \delta_{il}\hat{r}_j\hat{r}_k - 6\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l - \delta_{il}\delta_{jk}). \end{aligned} \tag{A 24}$$

Where the variables have been renamed for simplicity, as there are now only three scalar functions to be determined. A full-rank set of linear combinations of the scalar functions can be isolated by the appropriate tensor inner products,

$$M^{DC}_{ijkl}\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l = -2h_1 - 2h_3, \tag{A 25}$$

$$M^{DC}_{ijkl}\delta_{il}\hat{r}_j\hat{r}_k = -10h_1 - 2h_3, \tag{A 26}$$

$$M^{DC}_{ijkl}\delta_{jk}\hat{r}_i\hat{r}_l = -10h_1 + 2h_2 - 4h_3. \tag{A 27}$$

Now there is a linear combination of integrals required to separate each scalar function, so a linear system must be solved. The linear system is

$$\begin{bmatrix} M^{DC}_{ijkl}\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l \\ M^{DC}_{ijkl}\delta_{il}\hat{r}_j\hat{r}_k \\ M^{DC}_{ijkl}\delta_{jk}\hat{r}_i\hat{r}_l \end{bmatrix} = \begin{bmatrix} -2 & 0 & -2 \\ -10 & 0 & -2 \\ -10 & 2 & -4 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}, \tag{A 28}$$

and the inverse is

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & -1 & 0 \\ -5 & -3 & 1 \\ -5 & 1 & 0 \end{bmatrix} \begin{bmatrix} M^{DC}_{ijkl}\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l \\ M^{DC}_{ijkl}\delta_{il}\hat{r}_j\hat{r}_k \\ M^{DC}_{ijkl}\delta_{jk}\hat{r}_i\hat{r}_l \end{bmatrix}. \tag{A 29}$$

The coefficients h_1, h_2 and h_3 are therefore given by linear combinations of these tensor products with the integral corresponding to the inverse Fourier transform of the wave-space representation of the real-space contribution to M^{DC}_{ijkl}.

The wave-space representation of the real-space contribution to the Ewald sum of M^{DC}_{ijkl} is

$$M^{DC}_{ijkl} = -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{DC}_q \hat{q}_i(\delta_{jk} - \hat{q}_j\hat{q}_k)\hat{q}_l, \tag{A 30}$$

where

$$M^{DC}_q = \frac{1}{q^2}\left(\frac{3\sqrt{-1}}{a}\mathrm{j}_1(qa)\right)\left(\frac{3\sqrt{-1}}{b}\mathrm{j}_1(qb)\right)H(q\xi^{-1}) \tag{A 31}$$

is the scale factor at wavenumber q, and j_m is the mth-order spherical Bessel function of the first kind and $H(x) = 1 - (1 + x^2/4)\mathrm{e}^{-x^2/4}$ is Hasimoto's splitting function.

The first integral is the inverse Fourier transform

$$\begin{aligned} M^{DC}_{ijkl}\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l &= -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{DC}_q \hat{q}_i(\delta_{jk} - \hat{q}_j\hat{q}_k)\hat{q}_l\hat{r}_i\hat{r}_j\hat{r}_k\hat{r}_l \\ &= -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}} \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}} M^{DC}_q \sin^2\theta\cos^2\theta \end{aligned}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

$$= -\frac{(2\pi)^{-3}}{\mu}\int_0^\infty \mathrm{d}qq^2\int_\Omega \mathrm{d}\Omega \mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}}M_q^{DC}\sin^2\theta\cos^2\theta$$
$$= -\frac{(2\pi)^{-2}}{\mu}\int_0^\infty \mathrm{d}qq^2M_q^{DC}\int_0^\pi \mathrm{d}\theta\sin^3\theta\cos^2\theta\mathrm{e}^{\mathrm{i}qr\cos\theta}. \quad (A\,32)$$

The second integral is the inverse Fourier transform

$$M_{ijkl}^{DC}\delta_{il}\hat{r}_j\hat{r}_k = -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}}\mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}}M_q^{DC}\hat{q}_i(\delta_{jk}-\hat{k}_j\hat{k}_k)\hat{q}_l\delta_{il}\hat{r}_j\hat{r}_k$$
$$= -\frac{(2\pi)^{-2}}{\mu}\int_0^\infty \mathrm{d}qq^2M_q^{DC}\int_0^\pi \mathrm{d}\theta\sin^3\theta\mathrm{e}^{\mathrm{i}qr\cos\theta}. \quad (A\,33)$$

The third integral is the inverse Fourier transform

$$M_{ijkl}^{DC}\delta_{jk}\hat{r}_i\hat{r}_l = -\frac{1}{\mu V}\sum_{\boldsymbol{q}\neq\boldsymbol{0}}\mathrm{e}^{\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{r}}M_q^{DC}\hat{q}_i(\delta_{jk}-\hat{q}_j\hat{q}_k)\hat{q}_l\delta_{jk}\hat{r}_i\hat{r}_l$$
$$= -2\frac{(2\pi)^{-2}}{\mu}\int_0^\infty \mathrm{d}qq^2M_q^{DC}\int_0^\pi \mathrm{d}\theta\sin\theta\cos^2\theta\mathrm{e}^{\mathrm{i}qr\cos\theta}. \quad (A\,34)$$

The analytical expressions resulting from evaluating these integrals can be found using symbolic integration packages such as Mathematica, but the resulting expressions are very sensitive at very small values of r and ξ. Specifically, errors in floating point arithmetic come into play and destabilize the result. In this case, asymptotic expressions at small ξ can be derived and used instead. For general purpose use, quadruple precision arithmetic can minimize the floating point arithmetic error when evaluating the algebraic expressions resulting from the preceding integrals, and single or double precision can be used to tabulate them for use in fast simulations.

A.2. *Wave-space sum*

The wave-space sum could be computed using the non-uniform discrete Fourier transform (NUDFT), however the associated cost would be $O(N^2)$. To use the accelerated NUFFT, we have to derive quadrature operations that account for the differing finite sizes. Following the approach of Lindbo & Tornberg (2010), and using the shape factors in (A 1), we can define kernels that map particle forces and couplets onto the fluid force density, $\boldsymbol{H}$, and whose adjoints map the impinging velocity field, $\tilde{\boldsymbol{H}}$ onto particle velocities and velocity gradients. The math is straightforward and the result is that for finite size particles, the force is spread onto the gridded force density according to

$$H_i(\boldsymbol{y}) = \sum_\beta F_i^\beta\left(\frac{2\xi^2}{\pi\mu}\right)^{3/2}\mathrm{e}^{-2\xi^2r^2/\mu}K_1(r,b), \quad (A\,35)$$

and the couplet is similarly spread by

$$H_i(\boldsymbol{y}) = \sum_\beta C_{ip}^\beta r_p\left(\frac{2\xi^2}{\pi\mu}\right)^{3/2}\mathrm{e}^{-2\xi^2r^2/\mu}K_2(r,b), \quad (A\,36)$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where the sum is over all particles, $\boldsymbol{y}$ is a point in the fluid, and $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{x}_\beta$ is the minimum image distance between the particle centre $\boldsymbol{x}_\beta$ and the fluid point $\boldsymbol{y}$. The parameter μ was introduced by Lindbo & Tornberg (2010) to control the resolution of the NUFFT kernels on the underlying grid representing the fluid. (Lindbo & Tornberg used η to denote this parameter, but we use μ here to avoid confusion with the fluid viscosity, which has been represented by η throughout this work.) The adjoint contraction/interpolation operators are similarly

$$(U_i^\alpha)_w = \frac{1}{\mu}\int_\Omega \mathrm{d}\boldsymbol{y}\tilde{H}_i(\boldsymbol{y})\left(\frac{2\xi^2}{\pi\mu}\right)^{3/2} \mathrm{e}^{-2\xi^2 r^2/\mu} K_1(r, a) \tag{A 37}$$

and

$$(D_{ij}^\alpha)_w = \frac{1}{\mu}\int_\Omega \mathrm{d}\boldsymbol{y}\tilde{H}_i(\boldsymbol{y}) r_j \left(\frac{2\xi^2}{\pi\mu}\right)^{3/2} \mathrm{e}^{-2\xi^2 r^2/\mu} K_2(r, a). \tag{A 38}$$

Here, K_1 and K_2 are the kernels that define how the first and second force and velocity moments are mapped to and from the fluid force and velocity field,

$$K_1(r, b) = \left(\mathrm{e}^{-2\xi^2 b^2/\mu}\frac{\sinh(4\xi^2 br/\mu)}{4\xi^2 br/\mu}\right), \tag{A 39}$$

$$K_2(r, b) = 3\left(\mathrm{e}^{-2\xi^2 b^2/\mu}\frac{\sinh(4\xi^2 br/\mu) - (4\xi^2 br/\mu)\cosh(4\xi^2 br/\mu)}{(4\xi^2 br/\mu)^3}\right). \tag{A 40}$$

Note that this is the exact same result as obtained by Lindbo & Tornberg (2010) modified by an effective shape factor accounting for the way the finite size distributes the force moment. Physically, the point (Gaussian) kernels of Lindbo & Tornberg distribute forces about the centre of each particle, whereas the finite size kernels presented here distribute forces around the edges. This fact is more apparent when the summand in (A 35) is rewritten using the kernel from (A 39) and rearranged using the definition of sinh,

$$F_i^\beta\left(\frac{2\xi^2}{\pi\mu}\right)^{3/2}\left(\frac{1}{8\xi^2 br/\mu}\right)(\mathrm{e}^{-2\xi^2(r-b)^2/\mu} - \mathrm{e}^{-2\xi^2(r+b)^2/\mu}). \tag{A 41}$$

We can follow the approach of Lindbo & Tornberg (2010) to derive bounds for the quadrature error associated with the modified kernels in (A 37) and (A 38). Bounds will be derived only for the kernel in (A 39) because it decays the most slowly with r, so the error associated with kernels of higher order moments will be contained by the same bound.

The quadrature approximation to the integral in (A 37) is

$$\boldsymbol{Q} = h^3 \sum_{\boldsymbol{n}=-P/2}^{P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h)K(\boldsymbol{n}h, a), \tag{A 42}$$

where h is the grid spacing, h^3 is the volume of a grid element, P is the number of grid points within the support of each particle in each direction and $K(\boldsymbol{r}, b) = (2\xi^2/\pi\mu)^{3/2}\mathrm{e}^{-2\xi^2 r^2/\mu}K_1(r, b)$. It is understood that in the sum each element of $\boldsymbol{n}$ goes from $-P/2$ to $P/2$. The error associated with this quadrature approximation is

$$\epsilon_q = \left\| \boldsymbol{Q} - \int_\Omega \mathrm{d}\boldsymbol{y}\tilde{\boldsymbol{H}}(\boldsymbol{y})K(\boldsymbol{y}, a)\right\|, \tag{A 43}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

which is most conveniently evaluated by splitting into two pieces and applying the triangle inequality

$$\epsilon_q \leqslant \left\| h^3 \sum_{\|\boldsymbol{n}\|_\infty \leqslant P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h) K(\boldsymbol{n}h) - \int_\Omega \mathrm{d}\boldsymbol{y} \tilde{\boldsymbol{H}}(\boldsymbol{y}) K(\boldsymbol{y}) \right\| + \left\| h^3 \sum_{\|\boldsymbol{n}\|_\infty > P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h) K(\boldsymbol{n}h) \right\|. \tag{A 44}$$

The two terms in this equation,

$$A = \left\| h^3 \sum_{n \leqslant P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h) K(\boldsymbol{n}h) - \int_\Omega \mathrm{d}\boldsymbol{y} \tilde{\boldsymbol{H}}(\boldsymbol{y}) K(\boldsymbol{y}) \right\|, \tag{A 45}$$

$$B = \left\| h^3 \sum_{n > P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h) K(\boldsymbol{n}h) \right\|, \tag{A 46}$$

are bound using two different approaches.

To derive a bound on the value of A, we start with Poisson's summation formula, $h^3 \sum_{\boldsymbol{n}} \boldsymbol{f}(\boldsymbol{n}h) = \sum_{\boldsymbol{k}} \hat{\boldsymbol{f}}(\boldsymbol{k}/h)$, where $\boldsymbol{f} = \tilde{\boldsymbol{H}} K$,

$$A = \left\| \sum_{\boldsymbol{k}} \hat{\boldsymbol{f}}(\boldsymbol{k}/h) - \int_\Omega \mathrm{d}\boldsymbol{y} \tilde{\boldsymbol{H}}(\boldsymbol{y}) K(\boldsymbol{y}) \right\|. \tag{A 47}$$

By the definition of the Fourier transform, $\hat{\boldsymbol{f}}(\boldsymbol{k}) = \int_\Omega \mathrm{d}\boldsymbol{y} \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y})$, it is obvious that $\hat{\boldsymbol{f}}(\boldsymbol{0}) = \int_\Omega \mathrm{d}\boldsymbol{y} \tilde{\boldsymbol{H}}(\boldsymbol{y}) K(\boldsymbol{y})$ so that:

$$A = \left\| \sum_{|k|>0} \hat{\boldsymbol{f}}(\boldsymbol{k}/h) \right\|. \tag{A 48}$$

Here, $\hat{\boldsymbol{f}} = \widehat{\tilde{\boldsymbol{H}} K} = \tilde{\boldsymbol{H}} * \hat{K}$ is a quantity that decays at least as quickly as $\hat{K}$ with increasing $|k|$ because $\tilde{\boldsymbol{H}}$ is an approximately uniform quantity. Therefore

$$A \leqslant C_A \left\| \sum_{|k|>0} \hat{K}(\boldsymbol{k}/h) \right\| = C_A \left\| \sum_{|k|>0} \frac{\sin(ka/h)}{ka/h} \mathrm{e}^{-\mu k^2/8\xi^2 h^2} \right\|. \tag{A 49}$$

Noting that $\|\sin(ka/h)/(ka/h)\| \leqslant 1$,

$$A \leqslant C_A \left\| \sum_{|k|>0} \mathrm{e}^{-\mu k^2/8\xi^2 h^2} \right\|. \tag{A 50}$$

Replacing the non-integer wavevector k with the integer wavenumber κ, $k = 2\pi\kappa$, allows the sum to be evaluated explicitly,

$$\sum_{\kappa=1}^{\infty} \mathrm{e}^{-\pi^2 \mu \kappa^2/2\xi^2 h^2} = \tfrac{1}{2}(\vartheta_3(0, \mathrm{e}^{-\pi^2 \mu/2\xi^2 h^2}) - 1) \leqslant c \mathrm{e}^{-\pi^2 \mu/2\xi^2 h^2}, \tag{A 51}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

where $\vartheta_a(x, q)$ is the elliptic (Jacobi) theta function and c is an unspecified constant. Therefore,

$$A \leqslant C'_A \mathrm{e}^{-\pi^2\mu/2\xi^2h^2}. \tag{A 52}$$

The value of B is bound by the integral

$$B = \left\| h^3 \sum_{|\boldsymbol{n}|>P/2} \tilde{\boldsymbol{H}}(\boldsymbol{n}h)K(\boldsymbol{n}h) \right\| \leqslant \int_{hP/2}^{\infty} \mathrm{d}\boldsymbol{y}\tilde{\boldsymbol{H}}(\boldsymbol{y})K(\boldsymbol{y}) \leqslant C_B \int_{hP/2}^{\infty} \mathrm{d}\boldsymbol{y}K(\boldsymbol{y}), \tag{A 53}$$

where the second inequality holds because $\tilde{\boldsymbol{H}}$ is bounded over all space. Evaluating the last integral in spherical coordinates and substituting the definition that $w = hP/2$ yields

$$\begin{aligned} B \leqslant C_B \Bigg[& \left(\frac{8w^2\xi^2}{\pi\mu}\right)^{1/2} \mathrm{e}^{-2\xi^2(a^2+w^2)/\mu} \frac{\sinh(4aw\xi^2/\mu)}{4aw\xi^2/\mu} \\ & + \frac{1}{2}\left(\operatorname{erfc}\left(\sqrt{2/\mu}\xi(w+a)\right) + \operatorname{erfc}\left(\sqrt{2/\mu}\xi(w-a)\right)\right)\Bigg]. \end{aligned} \tag{A 54}$$

Combining (A 52) and (A 54), the bound for the total quadrature error is

$$\begin{aligned} \epsilon_q \leqslant C \Bigg[& \mathrm{e}^{-\pi^2\mu/2\xi^2h^2} + \left(\frac{8w^2\xi^2}{\pi\mu}\right)^{1/2} \mathrm{e}^{-2\xi^2(a^2+w^2)/\mu} \frac{\sinh(4aw\xi^2/\mu)}{4aw\xi^2/\mu} \\ & + \frac{\operatorname{erfc}\left(\sqrt{2/\mu}\xi(w+a)\right) + \operatorname{erfc}\left(\sqrt{2/\mu}\xi(w-a)\right)}{2}\Bigg], \end{aligned} \tag{A 55}$$

where $w = hP/2$ is the half-width of a particle's support. The first term in (A 55) is related to the resolution of the Gaussians on the grid, and the last two terms are associated with error of truncating the Gaussians after a finite number of standard deviations. Equation (A 55) can be written completely in terms of the support size P, the Gaussian resolution m and the grid spacing $\gamma = h/a$ using the definition of μ given by Lindbo & Tornberg (2010), $\mu = (2w\xi/m)^2$,

$$\begin{aligned} \epsilon_q \leqslant C \Bigg[& \mathrm{e}^{-\pi^2P^2/2m^2} + \left(\frac{2m^2}{\pi}\right)^{1/2} \mathrm{e}^{-2m^2/P^2\gamma^2} \mathrm{e}^{-m^2/2} \frac{\sinh(2m^2/\gamma P)}{2m^2/\gamma P} \\ & + \frac{\operatorname{erfc}\left(\dfrac{m}{\sqrt{2}}(1+2/\gamma P)\right) + \operatorname{erfc}\left(\dfrac{m}{\sqrt{2}}(1-2/\gamma P)\right)}{2}\Bigg]. \end{aligned} \tag{A 56}$$

Equation (A 56) makes it clear that the resolution error (the first term) is independent of particle size, while the truncation error is greater for finite size particles. The practical effect of this difference is that larger supports are required to control the truncation error, and refining the grid, i.e. decreasing γ, requires expanding the support to maintain that the support size is commensurate with the particle size.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

Equation (A 56) can be used to choose the parameters P and m given a desired error. To present the bounds in a simpler, more useful form, further simplification is necessary. The complementary error functions can be eliminated by using the fact that $\mathrm{erfc}(x) \leqslant \mathrm{e}^{-x^2}$ for $x > 0$, so that

$$\epsilon_q \leqslant C\left[\mathrm{e}^{-\pi^2 P^2/2m^2} + \mathrm{e}^{-2m^2/P^2\gamma^2}\mathrm{e}^{-m^2/2}\left(\sqrt{\frac{2m^2}{\pi}}\frac{\sinh(2m^2/\gamma P)}{2m^2/\gamma P} + \cosh\left(\frac{2m^2}{\gamma P}\right)\right)\right]. \tag{A 57}$$

Further, noting that $\sinh(x)/x \leqslant \cosh(x)$ for $x > 0$ gives

$$\epsilon_q \leqslant C\left[\mathrm{e}^{-\pi^2 P^2/2m^2} + \mathrm{e}^{-(m^2/2)(1+(4/\gamma^2 P^2))}\left(\left(\frac{2m^2}{\pi}\right)^{1/2} + 1\right)\cosh(2m^2/\gamma P)\right]. \tag{A 58}$$

This expression can be simplified further using the fact that $\cosh x < \mathrm{e}^x$ for $x > 0$:

$$\epsilon_q \leqslant C\left[\mathrm{e}^{-\pi^2 P^2/2m^2} + \left(1 + \sqrt{\frac{2m^2}{\pi}}\right)\mathrm{e}^{-(m^2/2)(1-(2/\gamma P))^2}\right]. \tag{A 59}$$

We seek the set of parameters such that the errors bounded by each of the terms in (A 59) are comparable in magnitude, i.e. we equate the exponents,

$$\frac{\pi^2 P^2}{2m^2} = \frac{m^2}{2}\left(1 - \frac{2}{\gamma P}\right)^2. \tag{A 60}$$

Equation (A 60) defines m as a function of P,

$$m^2 = \frac{\pi P}{\left(1 - \dfrac{2}{\gamma P}\right)}, \tag{A 61}$$

which simplifies the error estimate to

$$\epsilon_q \leqslant C\left(2 + \sqrt{\frac{2P}{1 - 2/\gamma P}}\right)\mathrm{e}^{-(\pi P/2)(1-(2/\gamma P))}. \tag{A 62}$$

The choice of P required to satisfy a specified error tolerance is given by the solution to this equation, which is does not have an analytically tractable form. An analytical expression for a slight underestimate of the required value of P can be obtained by neglecting the $\sqrt{2P/(1-2/\gamma P)}$ term in (A 62) and solving the equation

$$P^2 - \frac{4}{\gamma}P + \left(\frac{4}{\gamma^2} - \frac{4}{\pi^2}\left(\log\frac{\epsilon}{2C}\right)^2\right) = 0. \tag{A 63}$$

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

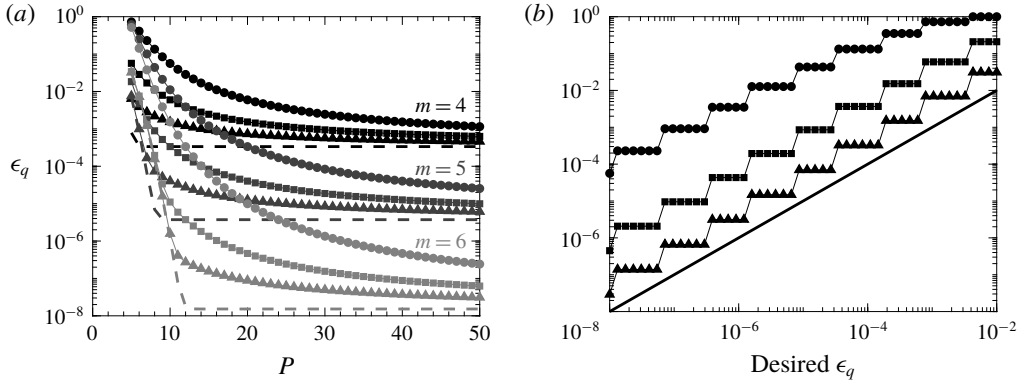


FIGURE 9. Comparison of estimates of the quadrature error associated with the kernels for point particles and particles with finite shape. (*a*) Error estimates as a function of support size P for various Gaussian width resolutions, $m = \{4, 5, 6\}$ and grid resolutions $\gamma = 0.5, 1.0, 2.0$ in circles, squares and triangles, respectively. Dashed lines show the same estimates for the point particle ($\gamma \to \infty$) limit. (*b*) Quadrature error for finite size spreading functions as a function of the specified error tolerance when the error bounds for point particles are used to choose m and P. Discrete levels are a result of using the ceiling function to ensure that P is an integer. Curves are shown for $\gamma = 0.5$ (circles), $\gamma = 1.0$ (squares) and $\gamma = 2.0$ (triangles). The solid line denotes parity.

Because of the $\log \epsilon$ term, the negative root of this equation guarantees that the choice of P is positive,

$$P = \text{Ceiling}\left[\frac{2}{\gamma} - \frac{2}{\pi}\log\frac{\epsilon}{2C}\right]. \tag{A 64}$$

Here, P needs to be an integer so the ceiling function is used to ensure a large enough value, and m can be found by using this result in (A 61),

$$m = \left\lceil \frac{\dfrac{2\pi}{\gamma} - 2\log\dfrac{\epsilon}{2C}}{\left(1 - \dfrac{2}{\gamma P}\right)} \right\rceil^{1/2}. \tag{A 65}$$

Equations (A 64) and (A 65) reduce to the point source suggestions provided by Tornberg *et al.* in the case that $a \to 0$ ($\gamma \to \infty$). For very large grid spacings, i.e. $h \to \infty$, again we have that $\gamma \to \infty$. This is the case for small values of the splitting parameter, ξ, for which grid cells are much larger than particles. Figure 9 shows the error estimates for the quadrature kernels associated with finite size particles and point sources (stokeslets). Is it clear that the error estimates for point particles are insufficient to control the error for finite sized particles. In general, larger supports are required in the finite size case, and the support size increases as the grid density increases because the force must be distributed over the entire particle.

Appendix B. Algorithm implementation

We have implemented an algorithm for monodisperse suspensions of spheres as a plugin for the molecular dynamics software suite HOOMD-blue (Anderson, Lorenz & Travesset 2008; Glaser *et al.* 2015). HOOMD is a set of highly optimized

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

object-oriented software tools to perform molecular dynamics simulations on graphic processing units, and permits implementation of novel integrators as new methods within its integrator class. The supplementary material contains a GPU implementation of the FSD plugin for HOOMD-blue. Included in the software is a sample script to perform a dynamic simulation using the plugin.

For large scale simulations, both memory and execution costs need to be controlled to deploy efficient implementations. To this end, all matrix operations are represented with kernels that compute the action of a given matrix on a vector without explicitly constructing the matrix itself. The one exception is the highly sparse preconditioner $\tilde{\boldsymbol{R}}_{FU}^{nf}$ and its associated incomplete Cholesky factor $\mathcal{L}$, which must be explicitly built and stored in sparse format. Because the preconditioner is sparse and a small value of r_p is chosen, its memory requirement is manageable. A simple estimate of the memory requirement is: the pair resistance tensor for a given particle pair is a 12×12 matrix, so the total memory requirement is $O(144N_{pair})$ numbers, where N_{pair} is the number of particle pairs. In monodisperse suspensions with sufficiently small r_p, each particle has at most 12 neighbours, so the memory requirement can be given in terms of the number of particles: $O(1728N)$ numbers. For $N = 10^5$, the total memory required for single precision calculations is approximately 0.7 GB. These so-called matrix-free kernels are fully compatible with the matrix inversion and square root schemes used throughout this paper, which are based on iterative methods that only require computing matrix-vector products. Furthermore, the matrix-vector products can be efficiently parallelized by exploiting the structure of each operator. For example, the lubrication interactions are computed by assigning a thread to each particle that loops over the nearest neighbours for that particle, adding their contribution to the total near-field force and executing threads in parallel on the GPU.

In the far-field calculation, the Ewald splitting parameter $\xi > 0$ is a free parameter that is chosen to optimize the overall calculation speed. Smaller values of ξ require more real-space calculations, while the larger values require more wave-space calculations. In the limit that $\xi \to 0$, the entire calculation is performed in real space, and in the limit that $\xi \to \infty$ the entire calculation is performed in wave space. The proper balance between real space and wave space to achieve optimal performance depends on implementation specific software and hardware details, but as is discussed in §§ B.1 and B.2, does not affect the calculation accuracy. We have previously given detailed estimates of the optimal values of ξ (Fiore *et al.* 2017), but for many applications in soft matter $\xi \approx 0.5$ is a reasonable choice.

B.1. *Far-field hydrodynamics: wave space*

With Hasimoto's splitting function, the magnitude of each term in the wave-space sum is $O(\mathrm{e}^{-k^2/4\xi^2})$, so the error incurred by evaluating the sum only for wavevectors with magnitude less than k_{max} is $O(\mathrm{e}^{-k_{max}^2/4\xi^2})$. Therefore, for a given value of ξ, the number of terms included in the sum is specified by the desired error tolerance. The wave-space sum calculation is accelerated by applying the spectral Ewald (SE) algorithm of Lindbo & Tornberg (2010). The SE algorithm applies the NUFFT to compute the wave space sum in $\mathbb{D}$ and $\mathbb{D}^{\dagger}$, reducing the cost of evaluating the Fourier sum for N_k wave vectors from $O(N_k^2)$ to $O(N_k \log N_k)$ by leveraging the fast Fourier transform (FFT) to compute the sum on a uniformly spaced grid. The NUFFT uses Gaussian kernel functions to transfer particle forces to an overlaid uniform grid on which the FFT can be applied. These kernel functions are also used to interpolate the particle velocities from the fluid velocity on the grid. The Gaussian kernel is truncated after m standard deviations, where m is chosen to resolve the error associated with

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

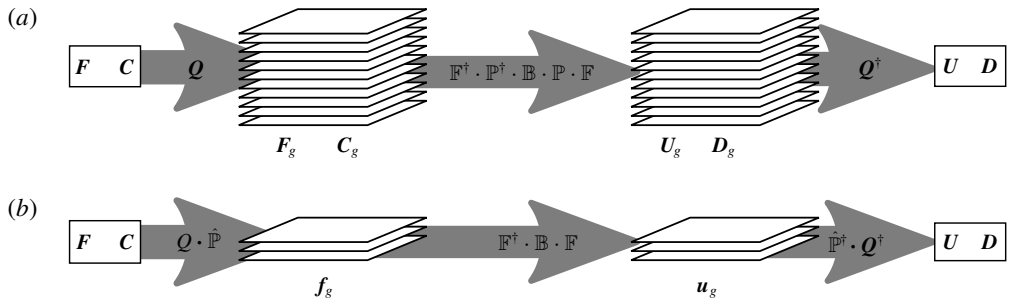


FIGURE 10. Schematic illustration of the NUFFT algorithm for monodisperse (*a*) and polydisperse (*b*) suspensions. In the monodisperse case, the point particle quadrature kernels $\boldsymbol{Q}$ are used so an FFT is applied to each component of the force and couplet, $\boldsymbol{F}_g$ and $\boldsymbol{C}_g$, and the shape factors are applied in wave space. In the polydisperse case, the shape factors are applied with the quadrature kernels, $\boldsymbol{Q} \cdot \hat{\mathbb{P}}$, so that FFTs are applied to the independent components of the force density, $\boldsymbol{f}_g$. The subscript g represents the quantity values on the uniform FFT grids.

spreading data to, and interpolating data from, the grid. These quadrature operations are performed by convoluting the particle data with the Gaussian kernel on a sub-grid of P data points in each direction, where, as with m, P is chosen consistent with a desired error tolerance using error bounds derived by Lindbo & Tornberg (2010). Importantly, the quadrature error decays exponentially with increasing m and P. The FFTs are performed using NVIDIA's cuFFT library and custom implementations of the Gaussian kernels, described below.

Even though we deal with simulations of particles with finite size, the Gaussian kernels derived by Lindbo & Tornberg (2010) for point particles can be used provided all particles are the same size (polydisperse particle suspensions are discussed in appendix A). Particles with the same size and shape scatter fluid in the same way, i.e. the shape factors in $\mathbb{P}$ are identical, so the wave-space particle interactions, $\mathbb{P}^\dagger \cdot \mathbb{B} \cdot \mathbb{P}$, are identical for all particle pairs. Therefore, we can treat each particle as a point and use the quadrature kernels devised by Lindbo & Tornberg (2010) for Stokeslets and account for the finite particle size and shape via the wave-space projectors. This approach requires eleven FFTs, one for each independent component of the force (three) and traceless couplet (eight). Each component is spread to its own grid, and the FFT is applied independently to each grid to obtain the Fourier coefficients for that force component. Applying $\mathbb{P}^\dagger \cdot \mathbb{B} \cdot \mathbb{P}$ to the Fourier-space coefficients produces the wave-space velocity moments which, after application of the inverse FFT, become the real-space velocity moments on the grid. The gridded velocity moments are then interpolated from the grid to particles to determine the particle velocity moments using the adjoint of the spreading operator. There is a different version of this algorithm, discussed in appendix A, that projects all of the force moments onto the force density, which has only 3 independent components, and therefore requires only three FFTs. A visual comparison of these two approaches to computing the NUFFT is given in figure 10. We find that the cuFFT algorithm is so efficient compared to the modified quadrature operations associated with the force density that it is more efficient to use eleven FFTs and the Stokeslet kernels. Furthermore, we

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

optimize the quadrature operations on the GPU using a particle-based approach which employs a block of P^3 threads per particle to simultaneously transfer data to/from the particle and all grid points within its support using atomic operations. As shown in appendix A, P is smaller for the Stokeslet kernel than for the finite size kernels at fixed error tolerance, which minimizes the cost of quadrature operations. A different algorithmic formulation is a grid-based approach, which employs a block of GPU threads per grid point that loop over particles nearby that point (Guo *et al.* 2015). Efficient implementation of this approach requires maintaining adjacency lists of the particles near each grid point, which can be costly. We find that the particle-based approach is efficient for all volume fractions seen in colloidal systems.

B.2. *Far-field hydrodynamics: real space*

The scalar functions in (3.18)–(3.20) are tabulated for a fixed set of distances once at the beginning of the simulation, and are thereafter interpolated from the table. The cost of constructing the table is amortized over many simulation time steps and produces very efficient matrix-vector multiplications. These scalar functions decay exponentially with the distance between the particles, with a decay rate that is controlled by the Ewald parameter ξ. The strength of the interaction is $\mathrm{e}^{-\xi^2 r^2}$, so only particle pairs whose distance is of the order of ξ^{-1} or smaller appreciably affect one another. Therefore, we define a cutoff radius r_{cut} beyond which particle pair interactions are ignored, and in doing so incur an error of $\epsilon_r \sim \mathrm{e}^{-\xi^2 r_{cut}^2}$. Such truncated interactions are efficiently evaluated using linked-cell lists (Allen & Tildesley 1989) to divide the simulation volume into cells of size r_{cut} and only consider interactions between particles within the same or adjacent cells. This reduces the number of interactions each particle has from $O(N)$ to $O(\phi r_{cut}^{d_f})$, where ϕ is the suspension volume fraction and d_f is the local fractal dimension, requiring only $O(N\phi r_{cut}^{d_f})$ computations to evaluate all pair interactions at a prescribed level of accuracy.

B.3. *Near-field hydrodynamics*

The same ideas applied to computing the real-space far-field interactions in § B.2 can be applied to computing the near-field lubrication interactions. The scalar resistance functions are tabulated and interpolated, to avoid computing the extensive sums presented by Jeffrey & Onishi (1984) within the dynamic simulation, and the interactions are computed for particle pairs within the lubrication cutoff of $r_{lub} = 4a$ using linked-cell lists. The lubrication cutoff radius of $4a$ is standard in Stokesian dynamics simulation of monodisperse suspensions (Brady & Bossis 1988; Sierou & Brady 2001). We avoid constructing the near-field resistance tensor explicitly because, even with sparse matrix storage formats, the memory requirements for large systems become impracticable, and therefore only compute the action of $\boldsymbol{R}_{FU}^{nf}$ on a vector.

The pair lubrication interactions are computed using the series expressions given by Jeffrey & Onishi (1984) and Jeffrey (1992*b*) including terms up to order $(r/a)^{-300}$, from which the far-field contribution to the pair resistance is subtracted. The series are evaluated using the FORTRAN code published by Jeffrey (1992*a*), which was translated to MATLAB. The resulting values were recorded in a hard-coded table with logarithmically spaced entries. Logarithmically spacing the table data improves the interpolation error because the interactions diverge when particles are nearly touching. For pairs of particles separated by sufficiently small distance, we switch to using the asymptotic expressions for lubricating interactions instead of the table interpolation.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

B.4. *Saddle point solves and preconditioners*

The sparse matrix products, incomplete Cholesky decompositions, and triangular Cholesky solves required for the near-field preconditioner are performed using NVIDIA's cuSPARSE library, which contains optimized subroutines for these operations. The reverse Cuthill–McKee ordering is computed using software published by Burkardt (2011), and the associated permutations are computed using subroutines from the NVIDIA cuSOLVER library. The cuSPARSE and cuSOLVER libraries are both part of the NVIDIA's CUDA toolkit (NVIDIA 2018). We found that the subroutine to compute the RCM ordering vector executed orders of magnitude more quickly using Burkardt's code than with the implementations available in the cuSOLVER and Boost libraries. Additionally, Burkardt's implementation exhibits a computational cost that scales linearly in the total number of particles, which does not appear to be true of the other libraries. The CUSP library (Dalton *et al.* 2014), which consists of C++ wrappers for fast sparse linear operations, is used for its preconditioned GMRES routine.

Appendix C. Notation

This appendix provides a list of the abbreviations and symbols used throughout this work.

$\boldsymbol{R}_{FU}$	Hydrodynamic resistance tensor, force-velocity component
$\boldsymbol{R}_{FE}$	Hydrodynamic resistance tensor, force-strain component
$\mathcal{B}$	Operator to project multipoles into force and velocity
$\mathcal{M}$	Far-field Hydrodynamic mobility tensor
$\mathcal{M}_r$	Real-space contribution to the far-field hydrodynamic mobility tensor
$\mathcal{M}_w$	Wave-space contribution to the far-field hydrodynamic mobility tensor
$\boldsymbol{J}$	Green's function
$\mathbb{D}$	Non-uniform discrete Fourier transform operator
$\mathbb{P}$	Wave-space operator of particle shape factors
$\mathbb{B}$	Wave-space Stokes flow divergence-free projector
$\mathcal{R}_{2B}$	Far-field contribution to the two-body near-field resistance tensor
$\boldsymbol{R}^{nf}$	Near-field contribution to the hydrodynamic resistance tensor
$\boldsymbol{R}^{nf}_{FU}$	Force-velocity component of the near-field resistance tensor
$\boldsymbol{R}^{nf}_{SU}$	Stresslet-velocity component of the near-field resistance tensor
$\boldsymbol{R}^{nf}_{SE}$	Stresslet-strain component of the near-field resistance tensor
$\boldsymbol{P}$	Reverse Cuthill–McKee permutation matrix
$\tilde{\boldsymbol{R}}^{nf}_{FU}$	Preconditioner approximation to $\boldsymbol{R}^{nf}_{FU}$
$\mathcal{L}$	Lower incomplete Cholesky factor
$\boldsymbol{D}$	Brownian sampling diagonal scaling preconditioner
$\boldsymbol{\Sigma}$	Rigid body projection matrix
$\mathbb{R}^{nf}_{FU}$	Force-velocity component of the near-field resistance tensor for composite particles

TABLE 1. List of tensors and matrices.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

a	Particle radius
D_s	Short-time self-diffusion coefficient
ϵ	Random finite different displacement magnitude
η	Fluid viscosity
η^L	Spin viscosity of a cubic lattice
η^S	Shear viscosity of a cubic lattice
δ_{ij}	Kronecker delta
$\boldsymbol{F}^P$	External conservative forces
$\boldsymbol{F}^B$	Brownian force
$\boldsymbol{F}^B_{nf}$	Near-field contribution to the Brownian force
$\boldsymbol{F}^B_{ff}$	Far-field contribution to the Brownian force
$\boldsymbol{F}^c$	Rigid body constraint forces
$\mathbb{F}^P$	External force on composite bead particle
$\mathbb{F}^B$	Brownian force on composite bead particle
$\mathbb{F}^H$	Hydrodynamic force on composite bead particle
$\mathbb{F}^H_{nf}$	Near-field contribution to the hydrodynamic force on composite particles
$\mathbb{F}^B_{nf}$	Near-field contribution to the Brownian force on composite particles
$\mathcal{F}$	Far-field hydrodynamic force moments (force, torque, stresslet, etc)
H	Hasimoto's Ewald filter
k_B	Boltzmann's constant
$\boldsymbol{k}$	Wavevector
$\boldsymbol{L}^H$	Hydrodynamic torque
N	Number of particles
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{V})$	Normally distributed variable with mean $\boldsymbol{\mu}$ and co-variance $\boldsymbol{V}$
ϕ	Volume fraction
$\boldsymbol{\Psi}$	Vector of independent Gaussian random variables
$\boldsymbol{q}$	Random finite difference displacement vector
qa	Wavenumber, normalized by particle radius
$\boldsymbol{S}$	Hydrodynamic stresslet
$\boldsymbol{S}^{ff}$	Far-field contribution to the hydrodynamic stresslet
$S(q)$	Structure factor
Δt	Size of discrete time step
T	Temperature
$\boldsymbol{U}$	Rigid body velocity, translation and rotation
$\boldsymbol{U}^\infty$	Ambient velocity field
$\mathcal{U}^B$	Far-field Brownian slip velocity
$\mathcal{U}^B_r$	Real-space contribution to the far-field Brownian slip velocity
$\mathcal{U}^B_w$	Wave-space contribution to the far-field Brownian slip velocity
$\mathbb{U}$	Generalized velocity of composite bead particle
ξ	Ewald splitting parameter
$\boldsymbol{x}$	Particle coordinates
$\Delta\boldsymbol{x}$	Particle displacements
ζ	Drag coefficient

TABLE 2. List of scalars and vectors.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

ASD	Accelerated Stokesian dynamics
FCM	Force coupling method
FIB	Fluctuating immersed boundary
FSD	Fast Stokesian dynamics
GMRES	Generalized minimum residuals
GPU	Graphics processing unit
NUFFT	Non-uniform fast Fourier transform
PSE	Positively split Ewald
PTPS	Particle time steps per second
RPY	Rotne–Prager–Yamakawa
SD	Stokesian dynamics
SEASD	Spectral Ewald accelerated Stokesian dynamics

TABLE 3. List of abbreviations.

REFERENCES

ABRAMOWITZ, M. & STEGUN, I. A. 1965 *Handbook of Mathematical Function: with Formulas, Graphs and Mathematical Tables*. Dover.

ALLEN, M. P. & TILDESLEY, D. J. 1989 *Computer Simulation of Liquids*. Oxford University Press.

ANDERSON, J. A., LORENZ, C. D. & TRAVESSET, A. 2008 General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.* **227** (10), 5342–5359.

BALBOA USABIAGA, F., KALLEMOV, B., DELMOTTE, B., BHALLA, A., GRIFFITH, B. & DONEV, A. 2016 Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach. *Commun. Appl. Maths Comput. Sci.* **11** (2), 217–296.

BANCHIO, A. J. & BRADY, J. F. 2003 Accelerated Stokesian dynamics: Brownian motion. *J. Chem. Phys.* **118** (22), 10323–10332.

BENZI, M., GOLUB, G. H. & LIESEN, J. 2005 Numerical solution of saddle point problems. *Acta Numerica* **14**, 1–137.

BRADY, J. F. 1993 Brownian motion, hydrodynamics, and the osmotic pressure. *J. Chem. Phys.* **98** (4), 3335–3341.

BRADY, J. F. & BOSSIS, G. 1988 Stokesian dynamics. *Annu. Rev. Fluid Mech.* **20**, 111–157.

BRADY, J. F., PHILLIPS, R. J., LESTER, J. C. & BOSSIS, G. 1988 Dynamic simulation of hydrodynamically interacting suspensions. *J. Fluid Mech.* **195**, 257–280.

BURKARDT, J. 2011 RCM Reverse Cuthill McKee Ordering C++ Library. https://people.sc.fsu.edu/~jburkardt/f_src/rcm/rcm.html.

CHOW, E. & SAAD, Y. 2014 Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions. *SIAM J. Sci. Comput.* **36** (2), A588–A608.

CICHOCKI, B., EKIEL-JEŻEWSKA, M. L. & WAJNRYB, E. 1999 Lubrication corrections for three-particle contribution to short-time self-diffusion coefficients in colloidal dispersions. *J. Chem. Phys.* **111** (7), 3265–3273.

CICHOCKI, B., FELDERHOF, B. U., HINSEN, K., WAJNRYB, E. & BŁAWZDZIEWICZ, J. 1994 Friction and mobility of many spheres in Stokes flow. *J. Chem. Phys.* **100** (5), 3780–3790.

CUTHILL, E. & MCKEE, J. 1969 Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, pp. 157–172. ACM.

DALTON, S., BELL, N., OLSON, L. & GARLAND, M. 2014 Cusp: generic parallel algorithms for sparse matrix and graph computations. Version 0.5.0.

DARDEN, T., YORK, D. & PEDERSEN, L. 1993 Particle mesh Ewald: an $N\log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* **98** (12), 10089–10092.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

DELMOTTE, B. & KEAVENY, E. E. 2015 Simulating Brownian suspensions with fluctuating hydrodynamics. *J. Chem. Phys.* **143** (24), 244109.

DELONG, S., BALBOA USABIAGA, F. & DONEV, A. 2015 Brownian dynamics of confined rigid bodies. *J. Chem. Phys.* **143** (14), 144107.

DELONG, S., USABIAGA, F. B., DELGADO-BUSCALIONI, R., GRIFFITH, B. E. & DONEV, A. 2014 Brownian dynamics without Green's functions. *J. Chem. Phys.* **140** (13), 134110.

ERMAK, D. L. & MCCAMMON, J. A. 1978 Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.* **69** (4), 1352–1360.

EWALD, P. P. 1921 Die berechnung optischer und elektrostatischer gitterpotentiale. *Ann. Phys.* **369** (3), 253–287.

FIORE, A. M. & SWAN, J. W. 2018 Rapid sampling of stochastic displacements in Brownian dynamics simulations with stresslet constraints. *J. Chem. Phys.* **148** (4), 044114.

FIORE, A. M., USABIAGA, F. B., DONEV, A. & SWAN, J. W. 2017 Rapid sampling of stochastic displacements in Brownian dynamics simulations. *J. Chem. Phys.* **146** (12), 124116.

FIORE, A. M., WANG, G. & SWAN, J. W. 2018 From hindered to promoted settling in dispersions of attractive colloids: simulation, modeling, and application to macromolecular characterization. *Phys. Rev. Fluids* **3** (6), 063302.

FIXMAN, M. 1986 Construction of Langevin forces in the simulation of hydrodynamic interaction. *Macromolecules* **19** (4), 1204–1207.

GEORGE, A. & LIU, J. 1981 *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall.

GLASER, J., NGUYEN, T. D., ANDERSON, J. A., LUI, P., SPIGA, F., MILLAN, J. A., MORSE, D. C. & GLOTZER, S. C. 2015 Strong scaling of general-purpose molecular dynamics simulations on GPUs. *Comput. Phys. Commun.* **192**, 97–107.

GOLDBERG, M. 1937 A class of multi-symmetric polyhedra. *Tohoku Math. J.* (1) **43**, 104–108.

GOMPPER, G., IHLE, T., KROLL, D. M. & WINKLER, R. G. 2009 Multi-particle collision dynamics: a particle-based mesoscale simulation approach to the hydrodynamics of complex fluids. In *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, pp. 1–87. Springer.

GRIFFITH, B. E. 2009 An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner. *J. Comput. Phys.* **228** (20), 7565–7595.

GUO, X., LIU, X., XU, P., DU, Z. & CHOW, E. 2015 Efficient particle-mesh spreading on GPUs. *Proc. Comput. Sci.* **51**, 120–129.

HASIMOTO, H. 1959 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**, 317–328.

HINCH, E. J. 1975 Application of the Langevin equation to fluid suspensions. *J. Fluid Mech.* **72** (3), 499–511.

HOOGERBRUGGE, P. J. & KOELMAN, J. M. V. A. 1992 Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhys. Lett.* **19** (3), 155–160.

ICHIKI, K. 2002 Improvement of the Stokesian dynamics method for systems with a finite number of particles. *J. Fluid Mech.* **452**, 231–262.

JEFFREY, D. 1992*a* Programs for Stokes resistance functions. https://www.uwo.ca/apmaths/faculty/jeffrey/research/Resistance.html.

JEFFREY, D. J. 1992*b* The calculation of the low Reynolds number resistance functions for two unequal spheres. *Phys. Fluids* A **4** (1), 16–29.

JEFFREY, D. J. & ONISHI, Y. 1984 Calculation of the resistance and mobility functions for two unequal rigid spheres in low-Reynolds-number flow. *J. Fluid Mech.* **139**, 261–290.

KEAVENY, E. E. 2014 Fluctuating force-coupling method for simulations of colloidal suspensions. *J. Comput. Phys.* **269**, 61–79.

KIM, S. & KARRILA, S. J. 2005 *Microhydrodynamics: Principles and Selected Applications*. Dover.

KUBO, R. 1966 The fluctuation–dissipation theorem. *Rep. Prog. Phys.* **29** (1), 255–284.

LADD, A. J. C. 1990 Hydrodynamic transport coefficients of random dispersions of hard spheres. *J. Chem. Phys.* **93** (5), 3484–3494.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press

LADD, A. J. C., COLVIN, M. E. & FRENKEL, D. 1988 Application of lattice-gas cellular automata to the Brownian motion of solids in suspension. *Phys. Rev. Lett.* **60** (11), 975–978.

LADYZHENSKAYA, O. A. & SILVERMAN, R. A. 1969 *The Mathematical Theory of Viscous Incompressible Flow*, vol. 12. Gordon & Breach.

LEFEBVRE-LEPOT, A., MERLET, B. & NGUYEN, T. N. 2015 An accurate method to include lubrication forces in numerical simulations of dense Stokesian suspensions. *J. Fluid Mech.* **769**, 369–386.

LINDBO, D. & TORNBERG, A.-K. 2010 Spectrally accurate fast summation for periodic Stokes potentials. *J. Comput. Phys.* **229** (23), 8994–9010.

LOMHOLT, S. & MAXEY, M. R. 2003 Force-coupling method for particulate two-phase flow: Stokes flow. *J. Comput. Phys.* **184** (2), 381–405.

MENG, Q. & HIGDON, J. J. L. 2008*a* Large scale dynamic simulation of plate-like particle suspensions. Part I. Non-Brownian simulation. *J. Rheol.* **52** (1), 1–36.

MENG, Q. & HIGDON, J. J. L. 2008*b* Large scale dynamic simulation of plate-like particle suspensions. Part II. Brownian simulation. *J. Rheol.* **52** (1), 37–65.

MIZERSKI, K. A., WAJNRYB, E., ZUK, P. J. & SZYMCZAK, P. 2014 The Rotne–Prager–Yamakawa approximation for periodic systems in a shear flow. *J. Chem. Phys.* **140** (18), 184103.

NVIDIA 2018 CUDA Toolkit Documentation.

REGNAUT, C. & RAVEY, J. C. 1989 Application of the adhesive sphere model to the structure of colloidal suspensions. *J. Chem. Phys.* **91** (2), 1211–1221.

REGNAUT, C. & RAVEY, J. C. 1990 Erratum: application of the adhesive sphere model to the structure of colloidal suspensions [*J. Chem. Phys.* 9 1, 1211 (1989)]. *J. Chem. Phys.* **92** (5), 3250–3250.

ROTNE, J. & PRAGER, S. 1969 Variational treatment of hydrodynamic interaction in polymers. *J. Chem. Phys.* **50** (11), 4831–4837.

ROUX, J.-N. 1992 Brownian particles at different times scales: a new derivation of the Smoluchowski equation. *Physica* A **188** (4), 526–552.

SAAD, Y. & SCHULTZ, M. H. 1986 GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **7** (3), 856–869.

SAINTILLAN, D., DARVE, E. & SHAQFEH, E. S. G. 2005 A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: the sedimentation of fibers. *Phys. Fluids* **17** (3), 033301.

SIEROU, A. & BRADY, J. F. 2001 Accelerated Stokesian dynamics simulations. *J. Fluid Mech.* **448**, 115–146.

SPRINKLE, B., BALBOA USABIAGA, F., PATANKAR, N. A. & DONEV, A. 2017 Large scale Brownian dynamics of confined suspensions of rigid particles. *J. Chem. Phys.* **147** (24), 244103.

SWAN, J. W. & WANG, G. 2016 Rapid calculation of hydrodynamic and transport properties in concentrated solutions of colloidal particles and macromolecules. *Phys. Fluids* **28** (1), 011902.

TORNBERG, A.-K. & GREENGARD, L. 2008 A fast multipole method for the three-dimensional Stokes equations. *J. Comput. Phys.* **227** (3), 1613–1619.

VIERA, M. N. 2002 Large scale simulations of Brownian suspensions. PhD thesis, University of Illinois at Urbana-Champaign.

WANG, G., FIORE, A. M. & SWAN, J. W. 2019 On the viscosity of adhesive hard sphere dispersions: critical scaling and the role of rigid contacts. *J. Rheol.* **63** (2), 229–245.

WANG, M. & BRADY, J. F. 2016 Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions. *J. Comput. Phys.* **306**, 443–477.

YEO, K. & MAXEY, M. R. 2010 Simulation of concentrated suspensions using the force-coupling method. *J. Comput. Phys.* **229** (6), 2401–2421.

https://doi.org/10.1017/jfm.2019.640 Published online by Cambridge University Press