



# A generalised drift-correcting time integration scheme for Brownian suspensions of rigid particles with arbitrary shape



Timothy A. Westwood<sup>a</sup>, Blaise Delmotte<sup>b</sup>, Eric E. Keaveny<sup>a,\*</sup>

<sup>a</sup> Department of Mathematics, Imperial College London, South Kensington Campus, London, SW7 2AZ, UK

<sup>b</sup> LadHyX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France

## ARTICLE INFO

### Article history:

Received 5 May 2021

Received in revised form 23 February 2022

Accepted 27 June 2022

Available online 8 July 2022

### Keywords:

Brownian motion

Suspensions

Time integration

Fluctuations

Simulation

## ABSTRACT

The efficient computation of the overdamped, random motion of micron and nanometre scale particles in a viscous fluid requires novel methods to obtain the hydrodynamic interactions, random displacements and Brownian drift at minimal cost. Capturing Brownian drift is done most efficiently through a judiciously constructed time-integration scheme that automatically accounts for its contribution to particle motion. In this paper, we present a generalised drift-correcting (gDC) scheme that accounts for Brownian drift for suspensions of rigid particles with arbitrary shape. The scheme seamlessly integrates with fast methods for computing the hydrodynamic interactions and random increments and requires a single full mobility solve per time-step. As a result, the gDC provides increased computational efficiency when used in conjunction with grid-based methods that employ fluctuating hydrodynamics to obtain the random increments. Further, for these methods the additional computations that the scheme requires occur at the level of individual particles, and hence lend themselves naturally to parallel computation. We perform a series of simulations that demonstrate the gDC obtains similar levels of accuracy as compared with the existing state-of-the-art. In addition, these simulations illustrate the gDC's applicability to a wide array of relevant problems involving Brownian suspensions of non-spherical particles, such as the structure of liquid crystals and the rheology of complex fluids.

© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Contents

1. Introduction	2
2. Brownian Dynamics of rigid particles of arbitrary shape	3
2.1. The overdamped Langevin equation for rigid body motion	4
2.2. Mobility matrix	5
3. The generalised drifter-corrector (gDC)	7
4. Applying $\mathbf{M}$ and computing $\dot{\mathbf{v}}$	8
4.1. RPY tensor	8
4.2. Fluctuating Force Coupling Method	8
5. Computational cost of the gDC	9
6. Simulations	10

\* Corresponding author.

E-mail address: [e.keaveny@imperial.ac.uk](mailto:e.keaveny@imperial.ac.uk) (E.E. Keaveny).

6.1. Implementations . . . . .	11
6.2. Convergence study: boomerang above no-slip boundary . . . . .	11
6.3. Confined liquid crystal . . . . .	11
6.4. Rheology of suspensions of Czech hedgehog colloids . . . . .	14
7. Summary and conclusions . . . . .	16
CRedit authorship contribution statement . . . . .	17
Declaration of competing interest . . . . .	17
Acknowledgements . . . . .	17
Appendix A. Quaternions . . . . .	17
Appendix B. Expressing torques in terms of the Lie algebra element . . . . .	19
Appendix C. Equivalence between the Lie algebra and quaternion SDEs . . . . .	19
Appendix D. Algorithm for Euler-Maruyama scheme . . . . .	21
Appendix E. Proof of consistency for gDC . . . . .	21
References . . . . .	22

## 1. Introduction

Brownian motion, the random motion exhibited by colloidal particles due to thermal fluctuations in the surrounding viscous fluid, is a common and important feature of fluidic systems at the micron and nanometre scales [1–3]. It plays a fundamental role in determining the distribution and configuration of polymers and particles that comprise complex fluids [4,5], which then in turn affect suspension rheology and bulk mechanical response to applied stresses [6–8]. Further, Brownian motion and thermal fluctuations play a critical role in biological and cellular transport processes, often in conjunction with the cell's active mechanisms that utilise chemical energy [9].

To simulate these systems, one often considers the equations of motion in the Brownian dynamics [10], or overdamped [11], limit where momentum variables are taken to have reached thermal equilibrium and the velocity field in the surrounding fluid is governed by the steady Stokes equations. In the context of rigid particles, this means that the dynamics of the particle positions and orientations are provided by the overdamped Langevin equations, and accordingly, the dynamics of the corresponding probability distribution of the particle positions and orientations is described by Smoluchowski's equation.

The hydrodynamic interactions between the particles due to the surrounding fluid are embodied in the dense, position-dependent mobility matrix [12,13] that relates the generalised forces on the particles to the resulting particle velocities. The hydrodynamic interactions between the particles also impact the fluctuations that they experience. The fluctuation-dissipation theorem [14] stipulates that the covariance of the random particle velocities is proportional to the mobility matrix and hence so too is the particle diffusion matrix. This requires the random particle velocities to be proportional to the square root of the mobility matrix. Finally, due to hydrodynamic interactions, the overdamped limit produces a nontrivial thermal, or Brownian, drift term [10] in the equations of motion that is proportional to the divergence of the mobility matrix with respect to the particle positions and orientations.

Initially, the computations of these three quantities – the mobility matrix, random velocities, and Brownian drift – in approaches such as Stokesian Dynamics [15] utilised a variety of direct methods. The entries of the mobility matrix were determined by evaluating analytical expressions based on multipole expansions and lubrication theory, and the random particle velocities were computed using a Cholesky decomposition of the resulting matrix. Finally, Brownian drift was incorporated directly by evaluating expressions for the derivatives of the mobility matrix entries. A reliance on these direct approaches limited system sizes, as well as particle shapes to spheres.

Since these first computations, there have been several key advances, some of which have been incorporated into Stokesian dynamics [16,17], including fast, matrix-free methods for determining the particle velocities given the forces applied to them. These methods provide the action of the mobility matrix on a vector in  $O(N \log N)$  operations, where  $N$  is the vector length, without ever computing the matrix itself. Such approaches are built around fast summation techniques such as the FMM [18–20], or the FFT and include spectrally-accurate Ewald summation [21], positively split Ewald summation [22,23], Accelerated and Fast Stokesian Dynamics [16,17,24,25], in addition to the immersed boundary [26] or similar methods, such as the force-coupling method [27,28], that utilise fast, grid-based solvers.

Recent work on computational methods for Brownian suspensions has focused on developing similarly rapid computations of the random particle velocities that can be used with the matrix-free methods described above. While the early work of Fixman [29] relying on a Chebyshev expansion of the spectrum of the mobility matrix has been adopted in implementations of Accelerated Stokesian Dynamics [17] and Brownian Dynamics [30], only more recently have techniques been developed that can be used more seamlessly with fast summation methods. These include the Lanczos algorithm [31] which computes iteratively an approximation of the matrix-square root using the Ritz values and vectors generated at the final iteration. For methods that utilise grid-based solvers, the random particle velocities can be computed rapidly from the flows generated by the spatially uncorrelated fluctuating stress originally considered by Landau and Lifshitz [32]. This fluctuating hydrodynamics approach provides the foundation for several methods that capture particle Brownian motion including Lattice Boltzmann [33], distributed Lagrange multiplier technique [34], finite element methods [35], the fluctu-

ating [36] and stochastic [37,38] immersed boundary methods and the fluctuating force-coupling method [39,40]. Further, computations that rely on decomposing the mobility matrix, as is the case with positively split Ewald summation [22,23] or discretisations of the fluctuating boundary integral equations [41], may use the fluctuating hydrodynamics approach and the Lanczos algorithm side-by-side to compute different contributions to the particle stochastic velocities resulting from the decomposition.

The final aspect of the computation is to account for Brownian drift when advancing the particle positions. This is done most efficiently using a well-designed time integration scheme that automatically accounts for Brownian drift, yielding a numerical solution whose moments converge to their true values as the time step size goes to zero. Fixman again provides [42,43] an early example of a midpoint scheme which, to leading order in the time step size, emits the Brownian drift term as part of its error expansion. This scheme, however, relies on a resistance formulation, where the dense, long-ranged mobility matrix must be inverted, introducing costly additional linear systems. Recent work, however, has shown that similar integration schemes can be constructed that avoid these additional linear systems. It has been shown [36] that the drift can be incorporated into the Euler-Maruyama scheme using random finite differences (RFD). This involves applying the mobility matrix evaluated at randomly displaced positions to a random force vector with an appropriate covariance. This limits additional costs due to mobility matrix multiplications. For the grid-based methods built around fluctuating hydrodynamics, the drifter-corrector (DC) midpoint scheme [40] eliminates the additional mobility matrix multiplication completely. Here, the drift term is incorporated by advancing the particle positions to the midpoint using the particle velocity extracted from the flow generated by the fluctuating stress. The full mobility computation is then performed at the midpoint.

While RFD and the DC represent important advances in designing schemes that capture Brownian drift and naturally interface with matrix-free methods, they were originally limited to the case of spherical particles with mobility matrices obtained through singular multipole expansions, or their regularised equivalents. Recent work has extended the RFD approach for simulation of rigid particles of arbitrary shape that are represented by multiple discrete degrees of freedom constrained to move as a rigid body. Sprinkle et al. [44,45] proposed a family of time integration schemes that use the chain rule to split the computation of the divergence of the body mobility matrix into three contributions that can be obtained at a lower cost using RFD. The resulting schemes, called Euler-Maruyama Traction (EM-T) and Trapezoidal Slip (T-S), are both weakly first-order accurate, but achieve first (resp. second) order accuracy for deterministic problems and require two (resp. three) full mobility solves per time step.

In this work, we generalise the DC scheme to simulate rigid bodies of arbitrary shape. Like the original DC, the generalised DC (gDC) scheme requires a single full mobility solve per time step. This significantly accelerates time integration for schemes built around grid-based solvers that can take advantage of the flows generated by a fluctuating stress. The gDC may also be used for matrix-based computations at a cost comparable to the existing state-of-the-art. The main idea behind the scheme is to advance to the mid-step using particle velocities obtained by orthogonally projecting the fluctuating velocities onto the space of rigid body motions. At the midstep, the full mobility is treated and important factors based on the divergence of the projected fluctuating velocity are incorporated into the update. The gDC has the nice property that many of the additional computations needed to capture Brownian drift occur at the level of individual particles, and hence naturally lend themselves to parallel computation and larger-scale simulation. We show that the resulting scheme is weakly first-order accurate and, in practice, provides errors similar in magnitude to the T-S scheme. Finally, we demonstrate the applicability of the scheme for larger-scale simulation by considering confined suspensions of rod-like particles, as well as the rheology of Czech hedgehog particle suspensions.

## 2. Brownian Dynamics of rigid particles of arbitrary shape

Consider a suspension of  $N$  rigid particles where the centre of mass position of particle  $p$  at time  $t$  is  $\mathbf{Y}_p(t)$ , while the rotation relative to its initial orientation is given by the unit quaternion  $\mathbf{q}_p(t)$  (see Fig. 1). We provide a detailed overview of using quaternions to represent rotations in Appendix A, but present some of the key facts here. Any vector  $\mathbf{b}$  in the body frame of particle  $p$  is given by  $\mathbf{B}(t) = \mathbf{R}(\mathbf{q}_p(t))\mathbf{b}$  in the lab frame at time  $t$ . The rotation matrix  $\mathbf{R}$  is related to entries of the quaternion through

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & 1 - 2q_1^2 - 2q_3^2 & 2(q_3q_2 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_3q_2 + q_1q_0) & 1 - 2q_2^2 - 2q_1^2 \end{bmatrix}. \quad (1)$$

The quaternions are advanced in time using the formula

$$\mathbf{q}_p(t) = \exp(\mathbf{u}_p(t)) \bullet \mathbf{q}_p(0), \quad (2)$$

where  $\bullet$  is the Hamiltonian product (see Appendix A),  $\mathbf{u} \in \mathbb{R}^3$  is the Lie algebra element, and

$$\exp(\mathbf{u}) = \left( \cos\left(\frac{\|\mathbf{u}\|}{2}\right), \sin\left(\frac{\|\mathbf{u}\|}{2}\right) \frac{\mathbf{u}}{\|\mathbf{u}\|} \right). \quad (3)$$

We will be interested in numerically integrating the overdamped Langevin equations for hydrodynamically interacting rigid bodies that govern the positions,  $\mathbf{Y}_p$ , and Lie algebra elements,  $\mathbf{u}_p$ , for each rigid body  $p$ .

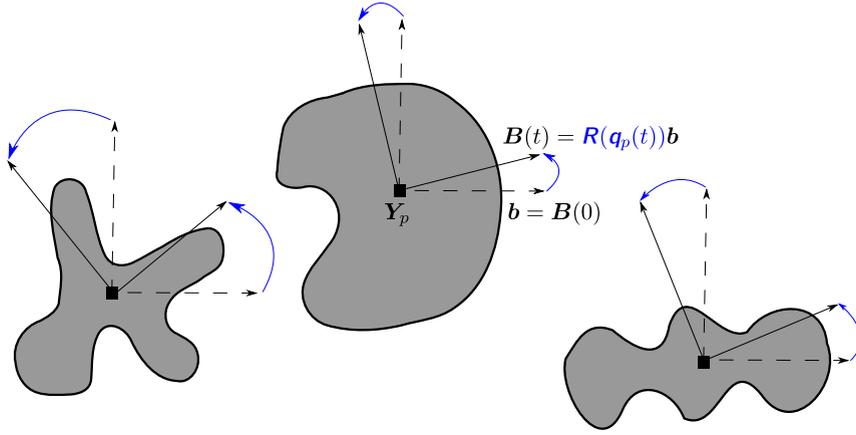


Fig. 1. Illustration of a rigid body with position  $\mathbf{Y}_p$  and orientation  $\mathbf{q}_p$ .

### 2.1. The overdamped Langevin equation for rigid body motion

Suppose that the particles are subject to an external potential  $U(\mathbf{Y}_1, \mathbf{u}_1, \dots, \mathbf{Y}_N, \mathbf{u}_N)$ , as well as non-conservative forces and torques collected into a single  $6N \times 1$  vector  $\mathbf{F}_{nc}$ . The conservative force on particle  $p$  is then  $-\partial_{\mathbf{Y}_p} U$ , while the corresponding torque is  $-\mathbf{D}_{\mathbf{u}_p}^\top \partial_{\mathbf{u}_p} U$ . A derivation of this expression for the conservative torque is presented in Appendix B. The matrix  $\mathbf{D}_{\mathbf{u}_p}^{-1}$  is the  $3 \times 3$  ‘dexpinv’ matrix defined by

$$\mathbf{D}_{\mathbf{u}_p}^{-1} = \mathbf{I} - \frac{1}{2} [\mathbf{u}_p \times] - \frac{1}{2\|\mathbf{u}_p\|^2} \left( \|\mathbf{u}_p\| \cot\left(\frac{\|\mathbf{u}_p\|}{2}\right) - 2 \right) [\mathbf{u}_p \times]^2, \quad (4)$$

where  $[\mathbf{v} \times]_{ij} = \varepsilon_{ikj} v_k$  and  $[\times \mathbf{v}] = -[\mathbf{v} \times] = [\mathbf{v} \times]^\top$  for any  $\mathbf{v} \in \mathbb{R}^3$ . The ‘dexpinv’ matrix relates the angular velocity of particle  $p$ ,  $\boldsymbol{\Omega}_p$ , to the time derivative of  $\mathbf{u}_p$  through

$$\frac{d\mathbf{u}_p}{dt} = \mathbf{D}_{\mathbf{u}_p}^{-1} \boldsymbol{\Omega}_p. \quad (5)$$

In the overdamped limit, the force and torque vectors are linearly related to their contributions to the particle velocity and angular velocity vectors via the  $6N \times 6N$  mobility matrix,  $\mathbf{N}$ , whose entries depend on the particle configuration. Specifically, we have

$$\mathbf{V} = \mathbf{N}\mathbf{F}, \quad (6)$$

where  $\mathbf{V}$  is the  $6N \times 1$  vector of all particle velocity and angular components, and  $\mathbf{F}$  is the vector containing the forces and torques that the particles exert on the surrounding fluid. The application of the mobility matrix is equivalent to solving the Stokes boundary value problem for a collection of rigid particles subject to applied forces and torques. Thus, in the absence of Brownian motion, the equations of motion are given by the system of differential equations

$$\frac{d\mathbf{x}}{dt} = -\tilde{\mathbf{N}} \left( \partial_{\mathbf{x}} U - \boldsymbol{\Phi}^{-\top} \mathbf{F}_{nc} \right), \quad (7)$$

where  $\mathbf{x} = [\mathbf{Y}_1^\top, \mathbf{u}_1^\top, \dots, \mathbf{Y}_N^\top, \mathbf{u}_N^\top]^\top$  describes the positions and orientations of all particles and  $\tilde{\mathbf{N}} = \boldsymbol{\Phi} \mathbf{N} \boldsymbol{\Phi}^\top$  with

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{\mathbf{u}_1}^{-1} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{D}_{\mathbf{u}_N}^{-1} \end{bmatrix}. \quad (8)$$

From (7) one can identify the vector of total non-hydrodynamic force and torque acting on the particles as  $\mathbf{F} = -\boldsymbol{\Phi}^\top \partial_{\mathbf{x}} U + \mathbf{F}_{nc}$ .

When Brownian motion is present, the particles will also move randomly due to thermal fluctuations in the surrounding fluid. In the overdamped limit, the effects of Brownian motion are captured through the inclusion of random increments of

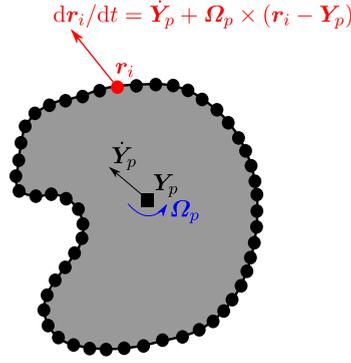


Fig. 2. Discretisation of the rigid body surface with discrete degrees of freedom.

the positions and Lie algebra elements, turning the equations of motion into a system of stochastic differential equations. Specifically, these increments are given by  $\sqrt{2k_B T} \tilde{\mathbf{N}}^{1/2} d\mathbf{W}$ , where  $k_B T$  is the thermal energy,  $\tilde{\mathbf{N}}^{1/2} = \Phi \mathbf{N}^{1/2}$  with  $\mathbf{N}^{1/2}$  being the matrix square root of the mobility matrix, and  $\mathbf{W}$  is a  $6N \times 1$  vector of independent Wiener processes. The dependence on the mobility matrix ensures that the fluctuation-dissipation theorem is satisfied, a necessary condition for the Boltzmann distribution to be recovered at equilibrium.

Finally, along with the random velocities, the overdamped equations of motion require the inclusion of the thermal drift term,  $k_B T \partial_{\mathbf{x}} \cdot \tilde{\mathbf{N}} dt$  [10]. This ensures that the stochastic differential equation with an Itô interpretation of the stochastic integral yields dynamics consistent with those of Smoluchowski's equation for the corresponding probability distribution. Putting these terms together, we arrive at the equations of motion for the particle positions and orientations,

$$d\mathbf{x} = \left[ -\tilde{\mathbf{N}} \left( \partial_{\mathbf{x}} U - \Phi^{-\top} \mathbf{F}_{nc} \right) + k_B T \partial_{\mathbf{x}} \cdot \tilde{\mathbf{N}} \right] dt + \sqrt{2k_B T} \tilde{\mathbf{N}}^{1/2} d\mathbf{W}. \quad (9)$$

## 2.2. Mobility matrix

The purpose of this paper is to develop time integration schemes for eq. (9) that avoid the direct computation of the thermal drift term and naturally interface with matrix-free methods for computing the random increments and particle velocities arising from the applied forces. In doing so, we will assume that the particles are discretised into  $M$  total degrees of freedom, as depicted in Fig. 2. This discretisation can involve the surface of the particles, as is done when considering the first-kind boundary integral representation of the Stokes equations [46,12], or can be related to a volume discretisation where elements of the particle volume are represented by regularised distributions of force, for example.

Let  $\mathbf{r}_i$  denote the position of discrete degree of freedom  $i$ ,  $\mathbf{v}_i \equiv d\mathbf{r}_i/dt$  be its velocity and  $\lambda_i$  be the force it exerts on the fluid. The hydrodynamic interactions between the discrete degrees of freedom provide a linear relationship between  $\mathbf{v}_i$  and  $\lambda_i$  such that

$$\mathbf{v}_i = \sum_{j=1}^M \mathbf{M}_{ij} \lambda_j, \quad (10)$$

where  $\mathbf{M}_{ij}$  is the matrix that relates the force on discrete degree of freedom  $j$  to the velocity of discrete degree of freedom  $i$ . Additionally, the requirement that all discrete degrees of freedom belonging to particle  $p$  move as a single rigid body (see Fig. 2) stipulates that

$$\mathbf{v}_i = \dot{\mathbf{Y}}_p + \boldsymbol{\Omega}_p \times (\mathbf{r}_i - \mathbf{Y}_p), \quad \forall i \in \mathcal{B}_p, \quad (11)$$

where  $\dot{\mathbf{Y}}_p$  is the translational velocity of particle  $p$ ,  $\boldsymbol{\Omega}_p$  is its angular velocity and  $\mathcal{B}_p$  is the set of discrete degrees of freedom belonging to particle  $p$ .

Introducing the vectors  $\boldsymbol{\lambda} = [\lambda_1^\top, \dots, \lambda_M^\top]^\top$  and  $\mathbf{V} = [\mathbf{V}_1^\top, \dots, \mathbf{V}_N^\top]^\top$  with  $\mathbf{V}_p = [\dot{\mathbf{Y}}_p^\top, \boldsymbol{\Omega}_p^\top]^\top$ , eqs. (10) and (11) for all particles  $p = 1, \dots, N$  can be equated to yield

$$\mathbf{M} \boldsymbol{\lambda} = \mathbf{K} \mathbf{V}, \quad (12)$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \dots & \mathbf{M}_{1M} \\ \vdots & \dots & \vdots \\ \mathbf{M}_{M1} & \dots & \mathbf{M}_{MM} \end{bmatrix} \quad (13)$$

is the  $3M \times 3M$  mobility matrix for the discrete degrees of freedom and  $\mathbf{K}$  is a  $3M \times 6N$  block matrix that maps the velocity and angular velocity from the space of rigid body motions to the velocities of the discrete degrees of freedom. Specifically

$$\mathbf{K}_i^{(p)} = \begin{cases} \begin{bmatrix} \mathbf{I}_3 & [\times(\mathbf{r}_i - \mathbf{Y}_p)] \end{bmatrix}, & \text{if } i \in \mathcal{B}_p, \\ \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, & \text{otherwise,} \end{cases} \quad (14)$$

is the  $3 \times 6$  block that relates the rigid body motion of particle  $p$  to the translational velocity of discrete degree of freedom  $i$ . Note that the specific form of  $\mathbf{M}$  will depend on how the discretisation is performed, however, here we will assume that the discretisation results in a positive definite  $\mathbf{M}$ , thereby preserving the structure of the underlying continuous equations. This is an important ingredient for the fluctuation-dissipation theorem to be satisfied exactly in the discrete setting [36].

In the absence of inertia, the conservation of linear and angular momentum reduces to the balance of force and torque, which can be written compactly for the  $N$  particles as

$$\mathbf{K}^\top \boldsymbol{\lambda} = \mathbf{F}, \quad (15)$$

where  $\mathbf{F}$  is the total, non-hydrodynamic, force and torque vector applied on the particles.

The mobility problem given by eqs. (12) and (15) forms a saddle point system

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{F} \end{bmatrix}. \quad (16)$$

After eliminating  $\boldsymbol{\lambda}$ , one obtains

$$\mathbf{V} = (\mathbf{K}^\top \mathbf{M}^{-1} \mathbf{K})^{-1} \mathbf{F} \quad (17)$$

and therefore the mobility matrix is given by

$$\mathbf{N} = (\mathbf{K}^\top \mathbf{M}^{-1} \mathbf{K})^{-1}. \quad (18)$$

As we describe later, our discretisation will follow from the rigid blob framework [47], but the gDC time integration scheme is applicable to any spatial discretisation provided that the resulting mobility matrix  $\mathbf{N}$  can be related to the saddle point system in eq. (16). This particular decomposition has the advantage that the computation of the stochastic particle velocities can be readily incorporated into the system by including a random velocity for the discrete degrees of freedom [44]. Specifically, we consider the system

$$\begin{bmatrix} \mathbf{M} & -\mathbf{K} \\ -\mathbf{K}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} -\check{\mathbf{v}} \\ -\mathbf{F} \end{bmatrix}, \quad (19)$$

with the random velocity,

$$\check{\mathbf{v}} = \sqrt{2k_B T} \mathbf{M}^{1/2} \mathbf{W}(t), \quad (20)$$

where  $\mathbf{M}^{1/2}$  is the square root of  $\mathbf{M}$  which satisfies  $\mathbf{M}^{1/2}(\mathbf{M}^{1/2})^\top = \mathbf{M}$ , and  $\mathbf{W}(t)$  is a  $3M \times 1$  vector of Wiener processes. After eliminating  $\boldsymbol{\lambda}$ , one obtains

$$\mathbf{V} = \mathbf{N}\mathbf{F} + \check{\mathbf{V}} \quad (21)$$

where  $\check{\mathbf{V}} = \mathbf{N}\mathbf{K}^\top \mathbf{M}^{-1} \check{\mathbf{v}}$ . As a result the fluctuating body velocities are given by

$$\check{\mathbf{V}} = \sqrt{2k_B T} \mathbf{N}\mathbf{K}^\top \mathbf{M}^{-1} \mathbf{M}^{1/2} \mathbf{W}(t) = \sqrt{2k_B T} \mathbf{N}^{1/2} \mathbf{W}(t), \quad (22)$$

where  $\mathbf{N}^{1/2}$  satisfies the fluctuation dissipation balance with  $\mathbf{N}^{1/2}(\mathbf{N}^{1/2})^\top = \mathbf{N}$ .

As shown by Balboa-Usabiaga et al. [48], the symmetric saddle point problem (19) can be efficiently solved with an iterative Krylov subspace method, such as GMRES, using a simple block diagonal preconditioner  $\mathbf{P}$ . The matrix  $\mathbf{P}$  is constructed by setting to zero the entries of  $\mathbf{M}$  in (19) if they correspond to interactions between discrete degrees of freedom from different rigid bodies. As  $\mathbf{K}$  is a block diagonal matrix, the cost of solving the linear system (19) will be related primarily to the costs of applying the matrix  $\mathbf{M}$  and computing  $\check{\mathbf{v}}$ .

### 3. The generalised drifter-corrector (gDC)

The primary result of this paper is the following generalisation of the DC scheme to rigid bodies of arbitrary shape. The gDC updates the particle positions and Lie algebra elements while automatically accounting for the Brownian drift term that arises in the equations of motion. It has the advantage of requiring only a single full mobility solve per time step. For grid-based solvers, the gDC incurs nearly the same cost as applying the well-known Euler-Maruyama scheme, outlined in Appendix D, which does not capture the Brownian drift. For matrix-based approaches the cost will be similar to the existing state-of-the-art [44,45].

Below we present the gDC algorithm. The time-step is indicated by superscript,  $n$ , e.g.  $\mathcal{V}^n$  would correspond to  $\mathcal{V}(\mathbf{x}^n) = \mathcal{V}(\mathbf{x}(t^n))$  in the continuous setting. For the purposes of implementing the algorithm, it should be noted that the collection of Lie algebra elements contained in  $\mathbf{x}^n$  is set to zero at the start of each time step (see Appendix A) and hence, as seen in Appendix E, the gDC algorithm requires no explicit mention of  $\Phi$ .

For each time step  $n = 0, 1, 2, \dots$

1. Generate a vector  $\mathbf{W}$  (or tensor  $\mathbf{W}$  if using fluctuating hydrodynamics) of  $\mathcal{N}(0, 1)$  random variables.
2. Solve the following saddle point problem at the start of the time-step,

$$\begin{bmatrix} \mathbf{I} & -\mathbf{K}^n \\ -(\mathbf{K}^\top)^n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \mathcal{V}^n \end{bmatrix} = \begin{bmatrix} -\check{\mathbf{v}}^n = -\sqrt{\frac{2k_B T}{\Delta t}} (\mathbf{M}^{1/2})^n \mathbf{W} \\ \mathbf{0} \end{bmatrix}, \quad (23)$$

yielding

$$\mathcal{V}^n = \left( (\mathbf{K}^\top)^n \mathbf{K}^n \right)^{-1} (\mathbf{K}^\top)^n \check{\mathbf{v}}^n. \quad (24)$$

Note that this is also the solution of the least-squares problem  $\min_{\mathcal{V}^n} \|\mathbf{K}^n \mathcal{V}^n - \check{\mathbf{v}}^n\|$ . As discussed in section 5, we use the direct solution given by eq. (24) in practice.

3. Define  $\nu = 1 + \frac{\Delta t}{2} (\partial_{\mathbf{x}} \cdot \mathcal{V})^n$ . For a small parameter  $\delta$ , the divergence can be calculated via<sup>1</sup>
  - (a) finite-differencing:

$$(\partial_{\mathbf{x}} \cdot \mathcal{V})^n = \sum_{i=1}^6 \sum_{p=1}^N \frac{\mathcal{V}_i^{(p)}(\mathbf{x}^n + \delta \mathbf{e}_i^{(p)}) - \mathcal{V}_i^{(p)}(\mathbf{x}^n)}{\delta} + \mathcal{O}(\delta^2) \quad (25)$$

where  $\mathcal{V}_i^{(p)}$  is the  $i$ -th component of the velocity of particle  $p$ , and  $\mathbf{e}_i^{(p)}$  is a  $6N \times 1$  vector whose nonzero entries, corresponding to the position and orientation of particle  $p$ , are given by the  $i$ -th Cartesian basis vector of  $\mathbb{R}^6$ ; or

- (b) random finite-differences:

$$(\partial_{\mathbf{x}} \cdot \mathcal{V})^n = \left\langle \sum_{i=1}^6 \sum_{p=1}^N \frac{\tilde{\mathbf{W}} \cdot \mathbf{e}_i^{(p)}}{\delta} \left( \mathcal{V}_i^{(p)}(\mathbf{x}^n + \delta \tilde{\mathbf{W}}) - \mathcal{V}_i^{(p)}(\mathbf{x}^n) \right) \right\rangle + \mathcal{O}(\delta^2), \quad (26)$$

where  $\tilde{\mathbf{W}}$  is a vector of independent  $\mathcal{N}(0, 1)$  random variables which is also independent of  $\mathbf{W}$ . By replacing the divergence in the definition of  $\nu$  with the term inside the expectation in eq. (26), the appropriate value of  $\nu$  will be achieved in expectation, which is ultimately all that is required to demonstrate weak-accuracy.

We find that option (a) is best partnered with grid-based mobility methods, whilst option (b) is best suited to matrix-based methods to reduce the computational cost. We discuss this in more detail in section 5.

4. Move the body positions and orientations to the mid-step according to

$$\mathbf{x}^m = \mathbf{x}^n + \frac{\Delta t}{2} \mathcal{V}^n. \quad (27)$$

5. Solve the full mobility problem at the mid-step,

$$\begin{bmatrix} \mathbf{M}^m & -\mathbf{K}^m \\ -(\mathbf{K}^\top)^m & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^m \\ \mathbf{V}^m \end{bmatrix} = \begin{bmatrix} -\check{\mathbf{v}}^m = -\sqrt{\frac{2k_B T}{\Delta t}} (\mathbf{M}^{1/2})^m \mathbf{W} \\ -\mathbf{F}^m \end{bmatrix}, \quad (28)$$

for

<sup>1</sup> In order to respect the physical units of the problem and to minimize the variance of the RFD approximation, we multiply  $\delta$  by a typical length scale  $L_p$  when computing the translational part of  $(\partial_{\mathbf{x}} \cdot \mathcal{V})^n$  [44].  $L_p$  is the maximum distance between two discrete degrees of freedom in the body and represents the typical size of body  $p$ .

$$\mathbf{V}^m = \mathbf{N}^m \left( \mathbf{F}^m + \left( \mathbf{K}^\top \right)^m \left( \mathbf{M}^{-1} \right)^m \check{\mathbf{v}}^m \right) = \mathbf{N}^m \mathbf{F}^m + \check{\mathbf{V}}^m, \quad (29)$$

where  $\mathbf{F}^m$  contains the forces and torques acting on the bodies at the mid-step. Note that quantities are evaluated at the mid-step using the mid-step quaternions  $\mathbf{q}_p^m = \exp(\mathbf{u}_p^m) \bullet \mathbf{q}_p^n$ .

6. Update the positions and orientations of the bodies according to

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \nu \Delta t \mathbf{V}^m, \quad (30)$$

with the new orientation quaternion of each particle  $p$  defined by  $\mathbf{q}_p^{n+1} = \exp(\mathbf{u}_p^{n+1}) \bullet \mathbf{q}_p^n$ .

In words, the gDC first computes a set of particle velocities by solving a least squares problem to find the appropriate stochastic velocity,  $\mathcal{V}$ , in the space of rigid body motions. This velocity is used to advance the particle positions to the mid-step where the full computation is performed. The positions are then updated using the velocity computed at the mid-step scaled by a factor based on the divergence of  $\mathcal{V}$ . As shown in Appendix E, the scheme recovers the first and second moments of the increment  $\Delta \mathbf{x} = \mathbf{x}^{n+1} - \mathbf{x}^n$  to first-order in expectation.

#### 4. Applying $\mathbf{M}$ and computing $\check{\mathbf{v}}$

As shown in section 2.2, hydrodynamic interactions between bodies are directly obtained from the matrix  $\mathbf{M}$ . The action of  $\mathbf{M}$  on a vector is required to compute the deterministic velocities  $\mathbf{V} = \mathbf{N}\mathbf{F}$ , while the product  $\mathbf{M}^{1/2}\mathbf{W}$  is necessary to obtain the Brownian velocities  $\check{\mathbf{V}}$ . It is therefore essential to use and develop efficient methods to compute the action of  $\mathbf{M}$  and its square root.

In this section, we outline two distinct strategies with similar resolution of hydrodynamic interactions between the discrete degrees of freedom. The first follows directly from an implementation of the rigid multiblob model [47] where the rigid particles are discretised into surface or volume elements that interact via the Rotne-Prager-Yamakawa (RPY) tensor. The other approach called the fluctuating Force Coupling Method (FCM) is matrix-free and simultaneously applies  $\mathbf{M}$  and computes  $\check{\mathbf{V}}$  by solving the fluctuating Stokes equations on a grid with a forcing term that accounts for the presence of the particles.

##### 4.1. RPY tensor

The well-known RPY mobility matrix was originally developed to provide a symmetric positive definite, pairwise approximation of the mobility matrix for a collection of spherical particles of equal radii in an unbounded domain [49]. Extensions of the RPY matrix for particles of different radii [50], in a background shear flow [49], and above a no-slip boundary [51] are available in the literature. Following the rigid multiblob model, the RPY tensor can be used to provide the hydrodynamic interactions between the discrete degrees of freedom making up the rigid particles. When discretising a particle surface, the rigid multiblob model provides a first-order accurate approximation to the particle mobility.

In general, the computational cost of these matrix-vector products scales quadratically with the number of discrete degrees of freedom. More sophisticated methods, such as the fast multipole methods (FMM) [52] and Ewald methods, achieve a linear scaling. For periodic domains, one can use the positively split Ewald method [22,23] to compute the action of  $\mathbf{M}$  on a vector. In our computations below, we perform a direct pairwise evaluation of the wall-corrected RPY tensor [51].

The action of  $\mathbf{M}^{1/2}$  on the random vector  $\mathbf{W}$  is obtained through the Lanczos algorithm [31]. To achieve convergence, the Krylov subspace  $\mathbf{K}$  is enriched iteratively with basis vectors that are linear combinations of the powers of the mobility matrix times the random vector:  $\mathbf{K} = \text{span}\{\mathbf{W}, \mathbf{M}\mathbf{W}, \mathbf{M}^2\mathbf{W}, \dots\}$ . The cost of the method therefore depends on the number of basis vectors, and thus mobility-vector products, required to reach a given tolerance  $\epsilon$ .

##### 4.2. Fluctuating Force Coupling Method

The other approach relies on a matrix-free method called fluctuating Force Coupling Method (FCM) [53], which combines FCM [54] with fluctuating hydrodynamics [32]. With fluctuating FCM, the coupling between the discrete degrees of freedom and the fluid is achieved through a forcing term added to the fluctuating Stokes equations,

$$\nabla p - \eta \nabla^2 \mathbf{u} = \nabla \cdot \mathbf{W} + \mathbf{f}, \quad (31)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (32)$$

where  $\eta$  is the dynamic fluid viscosity,  $\mathbf{u}$  the fluid velocity and  $p$  is the pressure.

The first term in the RHS of eq. (31) is the divergence of the fluctuating stress tensor,  $\mathbf{W}$ .  $\mathbf{W}$  is delta correlated in time and space with the following statistics,

$$\langle W_{\alpha\beta}(\mathbf{x}, t) \rangle = 0, \quad (33)$$

$$\langle W_{\alpha\beta}(\mathbf{x}, t) W_{\gamma\chi}(\mathbf{y}, t') \rangle = 2\eta k_B T (\delta_{\alpha\gamma} \delta_{\beta\chi} + \delta_{\alpha\chi} \delta_{\beta\gamma}) \delta(\mathbf{x} - \mathbf{y}) \delta(t - t'). \quad (34)$$

The second term on the RHS of eq. (31) is the forcing transferred to the fluid with a spreading operator  $\mathbf{S}$ ,

$$\mathbf{f}(\mathbf{x}) = \mathbf{S}[\boldsymbol{\lambda}](\mathbf{x}) = \sum_{i=1}^M \lambda_i \Delta_i(\mathbf{x}), \quad (35)$$

where the finite size of the discrete degrees of freedom is accounted for with a Gaussian spreading envelope,

$$\Delta_i(\mathbf{x}) = (2\pi\sigma^2)^{-3/2} \exp\left(-\frac{\|\mathbf{x} - \mathbf{r}_i\|^2}{2\sigma^2}\right). \quad (36)$$

The length scale  $\sigma$  functions in a similar way to the bead radius,  $a$ , in the RPY tensor. The diagonal entries of the FCM mobility matrices will match those of the RPY tensor if  $\sigma = a/\sqrt{\pi}$  [54].

The fluid velocity, obtained after solving eqs. (31) and (32), is the sum of a deterministic part, due to the forcing  $\boldsymbol{\lambda}$ , and a fluctuating term, stemming from the fluctuating stress. We may express the total fluid velocity as

$$\mathbf{u} = \mathbf{L}^{-1}(\mathbf{S}\boldsymbol{\lambda} + \mathbf{D}\mathbf{W}) \quad (37)$$

$$= \mathbf{u}^D + \check{\mathbf{u}}, \quad (38)$$

where  $\mathbf{S}$  is the spreading operator in eq. (35),  $\mathbf{L}^{-1}$  is the inverse Stokes operator (i.e. the fluid solver), and  $\mathbf{D}$  the divergence operator applied to the fluctuating stress [36].

The velocities of the discrete degrees of freedom are then obtained from the fluid velocity using an averaging operator,  $\mathbf{J}$ , such that

$$\mathbf{v}_i = (\mathbf{J}[\mathbf{u}])_i = \int \mathbf{u} \Delta_i(\mathbf{x}) d^3\mathbf{x} = \mathbf{v}_i^D + \check{\mathbf{v}}_i, \quad i = 1, \dots, M, \quad (39)$$

where  $\mathbf{J} = \mathbf{S}^*$  is adjoint to the spreading operator.

We see then that the FCM mobility matrix can be written as the composition of three linear operators:  $\mathbf{M}_{FCM} = \mathbf{J}\mathbf{L}^{-1}\mathbf{S}$ . Additionally, as demonstrated in [39] the velocity  $\check{\mathbf{v}}$  satisfies the fluctuation–dissipation theorem with the covariance given by the FCM approximation of the mobility matrix,

$$\langle \check{\mathbf{v}}(t)\check{\mathbf{v}}(t') \rangle = 2k_B T \mathbf{M}_{FCM} \delta(t - t'), \quad (40)$$

where  $\check{\mathbf{v}} = \sqrt{2k_B T} \mathbf{M}_{FCM}^{1/2} \mathbf{W}(t)$ , with  $\mathbf{M}_{FCM}^{1/2} = \mathbf{J}\mathbf{L}^{-1}\mathbf{D}$ .

Thus, we see that fluctuating FCM simultaneously computes the actions of  $\mathbf{M}$  and  $\mathbf{M}^{1/2}$  on  $\boldsymbol{\lambda}$  and  $\mathbf{W}$ , respectively, by considering solutions to the forced fluctuating Stokes equations.

Following [39,40], we implement fluctuating FCM in periodic domains. This involves first evaluating the forcing,  $\mathbf{f}(\mathbf{x})$ , and the fluctuating stress,  $\mathbf{W}$ , on a regular grid. Next, the Stokes equations are solved using a Fourier spectral method. Finally, the trapezoidal rule is used to numerically integrate eq. (39) to obtain the translational velocity for each of the discrete degrees of freedom making up the rigid bodies.

## 5. Computational cost of the gDC

Due to the differences in their implementations, there is a distinct difference in cost between RPY and grid-based fluctuating FCM that manifests itself when computing the random velocity vector,  $\check{\mathbf{v}}$ . For both implementations, the most expensive computation is applying the matrix  $\mathbf{M}$ . With RPY, as we are directly handling  $\mathbf{M}$ , any time the particle positions change, we must compute  $\check{\mathbf{v}} = \mathbf{M}^{1/2}\mathbf{W}$  using the Lanczos algorithm, for which each iteration requires a matrix-vector multiplication involving  $\mathbf{M}$ .

For fluctuating FCM, however, we are instead working with a decomposition of  $\mathbf{M}_{FCM}^{1/2}$  such that  $\check{\mathbf{v}} = \sqrt{2k_B T} \mathbf{J}\mathbf{L}^{-1}\mathbf{D}\mathbf{W}(t)$ . In this decomposition, the only matrix that depends on the particle configuration is  $\mathbf{J}$ . Thus, provided that  $\mathbf{W}$  remains the same, changing the particle positions simply requires reaveraging the fluid velocity  $\check{\mathbf{u}}$  at the new positions, and avoids having to recompute the velocity field itself.

*Cost of Step 2:* Along with the computation of  $\check{\mathbf{v}}$ , Step 2 of the gDC algorithm in section 5 involves solving a simplified mobility problem with  $\mathbf{M}$  replaced by  $\mathbf{I}$ . This is equivalent to solving the least squares problem  $\min_{\mathbf{v}} \|\mathbf{K}\mathbf{v} - \check{\mathbf{v}}\|$ , the solution of which is given by eq. (24) and involves the  $6N \times 6N$  symmetric, block-diagonal matrix  $\mathbf{K}^\top \mathbf{K}$ . The  $6 \times 6$  block associated with body  $p$  is given by

$$(\mathbf{K}^\top \mathbf{K})^{(p)} = \begin{bmatrix} M^{(p)} \mathbf{I}_3 & \sum_{i \in \mathcal{B}_p} [\times(\mathbf{r}_i - \mathbf{Y}_p)] \\ \sum_{i \in \mathcal{B}_p} [(\mathbf{r}_i - \mathbf{Y}_p) \times] & - \sum_{i \in \mathcal{B}_p} [(\mathbf{r}_i - \mathbf{Y}_p) \times]^2 \end{bmatrix}, \quad (41)$$

where  $\mathcal{B}_p$  is the set of  $M^{(p)}$  discrete degrees of freedom belonging to body  $p$ . Therefore, computing  $(\mathbf{K}^\top \mathbf{K})^{-1}$  just requires inverting  $N$  small  $6 \times 6$  blocks using dense linear algebra, which is negligible in terms of computational cost and, additionally, naturally lends itself to parallel computation.

Thus, the primary cost associated with Step 2 involves computing  $\check{\mathbf{v}}$ , which for fluctuating FCM incurs the cost of one Stokes solve and  $M$  averaging operations. The RPY implementation will require  $M_{\text{Lanczos}}$  mobility-vector products with the Lanczos method.

*Cost of Step 3:* The main cost of this step is computing the scalar term  $\nu$  using finite differences (FD) or RFD to generate  $\partial_{\mathbf{x}} \cdot \mathcal{V}$ . Finite differencing involves perturbing the positions and orientations of each rigid body and calculating the new fluctuating velocities for the discrete degrees of freedom at the disturbed states. With RFD, the new fluctuating velocities are only calculated once, after disturbing the particles' positions and orientations with a random vector  $\tilde{\mathbf{W}}$ .

For fluctuating FCM, FD is straight forward and incurs minimal cost. Since the body velocities are uncorrelated in eq. (23), the finite-differencing step in eq. (25) requires  $6M$  averaging operations. As discussed above, however, changing the particle position with the RPY-based implementation will require recomputing  $\mathbf{M}^{1/2} \mathbf{W}$  with each displacement. Thus, we require  $6N \times M_{\text{Lanczos}}$  mobility-vector products. This is a cost that could rise quickly as the particle number increases. On the other hand, using RFD in eq. (26) only requires  $M_{\text{Lanczos}}$  mobility-vector products with matrix-based methods and  $M$  averaging operations with fluctuating FCM.

*Cost of Step 5:* This step involves the costliest computation of solving the full mobility problem, eq. (19), along with the additional computation of  $\check{\mathbf{v}}$ .

For the fluctuating FCM implementation, each GMRES iteration requires a full fluctuating FCM computation that involves spreading the force vector  $\lambda$ , solving the Stokes equations, and finally averaging the resulting flow field to obtain the velocities for the discrete degrees of freedom. A single additional averaging is needed to incorporate the fluctuating velocities,  $\check{\mathbf{v}}^m$ , on the RHS of eq. (28).

For matrix-based methods, each GMRES iteration will incur the cost of a matrix-vector product involving  $\mathbf{M}$ . There will also be the cost of  $M_{\text{Lanczos}}$  Lanczos interactions needed to obtain  $\check{\mathbf{v}}^m$ .

For a tolerance of  $\epsilon = 10^{-3}$ , both the Lanczos and GMRES with block preconditioning require approximately  $M_{\text{Lanczos}} = M_{\text{GMRES}} = 5$  interactions on average. The tolerance of  $\epsilon = 10^{-3}$  has been shown [44] to be a suitable choice for simulations involving fluctuations. Adding these costs together assuming  $\epsilon = 10^{-3}$ , we see that the cost for one gDC-FD time iteration with fluctuating FCM is  $(1 + 5 =) 6$  Stokes solves,  $5M$  spreading operations and  $(1 + 6 + 1 + 5 =) 13M$  averaging operations. For the RPY based implementation, we have  $(5 + 6N \times 5 + 5 + 5 =) 5 \times (6N + 3)$  matrix-vector products involving  $\mathbf{M}$ . With gDC-RFD the number of averaging operations reduces to  $8M$  for fluctuating FCM, and the total number of matrix-vector products involving  $\mathbf{M}$  with the RPY based implementation lowers to 20, becoming independent of the particle number  $N$ .

As shown in section 6.2, the gDC is more stable with FD than with RFD. For grid-based methods, the  $5M$  additional averaging operations incurred by FD are negligible compared to the Stokes solve. This is not true for matrix based methods for which RFD represents a significant cost reduction.

Based on these costs, we see that the gDC with FD is well-suited for grid-based methods utilising fluctuating hydrodynamics such as fluctuating FCM [53,55], PSE [22,23] or the fluctuating and stochastic Immersed Boundaries methods [56,57,36,45], while gDC with RFD is the best compromise between numerical stability and computational cost for matrix-based methods.

In order to better understand the computational effort associated with the gDC, we compare its computational cost with other state-of-the-art schemes. For matrix-based methods, the Trapezoidal-Slip (T-S) scheme from Sprinkle et al. [44] requires 1 Lanczos call, 3 full mobility solves and 2 additional mobility-vector products, representing a total of 22 mobility-vector products per time-step (see Algorithm 2, p8 in [44]), which is similar to the cost of gDC-RFD (20 matrix-vector products). For grid-based methods relying on fluctuating hydrodynamics, we compare with a recently developed first-order deterministically accurate scheme called the Split-Euler-Maruyama (SEM) [45]. Per time-step, the SEM algorithm requires 2 full and 1 unconstrained mobility solves (approximately equivalent to  $2 \times 5 + 2 = 12$  Stokes solves), in addition to  $12M$  spreading and  $13M$  averaging operations (see Algorithm 1, p8 [45]). The gDC requires 1 full mobility solve and 1 unconstrained solve per time-step ( $\approx 5 + 1 = 6$  Stokes solves), half as many as SEM, as well as  $5M$  spreading and  $13M$  (or  $8M$  if using RFD in step 3) averaging operations. Additionally, in practice, the gDC yields errors similar to the second-order deterministically accurate T-S scheme (see Section 6.2).

## 6. Simulations

In this section, we demonstrate the performance of the gDC by performing simulations of particulate suspensions under both dynamic and equilibrium conditions. In Section 6.2, we investigate the accuracy of the gDC by examining the equilibrium distributions of the position and orientation of a boomerang-shaped particle in a gravitational field above a planar no-slip boundary. Owing to the small size of this system, we can integrate for long times and acquire many realisations to quantify the temporal accuracy of the gDC. We compare these results with those obtained using the state-of-the-art integrators developed in Sprinkle et al. [44]. After this, we demonstrate the suitability of the gDC for large scale simulations of anisotropic Brownian particles. In Section 6.3, we study the equilibrium properties of confined suspensions of rod-shaped particles, a model for a liquid crystal or stiff-polymer system. Finally, inspired by recent experimental work [58],

in Section 6.4 we use the gDC to perform simulations to obtain the rheology of colloidal suspensions made from Czech hedgehog-shaped (CH) particles.

### 6.1. Implementations

The simulations that follow rely on two implementations: one using direct evaluations of the wall-corrected RPY tensor and the other using fluctuating FCM.

- **Python with RPY:** for the single boomerang simulations we use the collaborative code, called “RigidMultiblobWall”, that one of us co-developed [48] and used for large scale simulations of colloidal active particles [59,47]. The code contains most of the time integration schemes recently developed for matrix-based approaches [60,47,44]. It relies on the pairwise evaluation of the wall-corrected RPY tensor. The mobility computations can be accelerated in different ways: using C++ routines, Numba or GPUs via the PyCUDA interface. We implemented the gDC in the code to compare its performance with the most accurate scheme developed by Sprinkle et al.: the Trapezoidal Slip scheme (T-S).
- **C++ with fluctuating FCM:** for the confined liquid crystal and rheology simulations, we use an FFT-based fluid solver with fluctuating FCM in a periodic box. FFTs are parallelised with the scalable MPI library FFTW. MPI is also used to parallelise the spreading and averaging operations as well as the nearest neighbour search necessary to compute short-ranged, pairwise interactions between discrete degrees of freedom. At each grid point, the entries of the fluctuating stress are independent Gaussian random variables. Since the fluctuating stress  $\mathbf{W}$  is symmetric, at each grid point we generate six random numbers based on a Gaussian distribution with zero mean and unit variance. We then multiply the off-diagonal entries by  $\sqrt{2k_B T \eta / (\Delta x^3)} \Delta t^{-1/2}$  and the diagonal ones by  $2\sqrt{k_B T \eta / (\Delta x^3)} \Delta t^{-1/2}$ . We truncate the Gaussian envelopes by setting  $\Delta_i(\mathbf{x}) = 0$  for  $\|\mathbf{x} - \mathbf{r}_i\| > 3a$ , and the length scale of the envelope,  $\sigma$ , is related to the grid size,  $\Delta x$ , through  $\sigma / \Delta x = 1.86$ . For more details, we refer the reader to our previous work [40].

### 6.2. Convergence study: boomerang above no-slip boundary

In this section we study the accuracy of the gDC scheme by examining the equilibrium distributions of a boomerang particle in a gravitational field above a no-slip boundary. Micron-size boomerangs have been used extensively to study the diffusion of colloidal anisotropic particles in experiments [61] and simulations [60,48]. Following Delong et al. [60] and as shown in Fig. 3, we discretise the boomerang using 15 RPY-particles separated by distance  $a$ , the radius of the RPY-particle. The boomerang is subject to a gravitational force of magnitude  $mg = 0.18k_B T/a$  and each RPY-particle interacts with the planar boundary through a short-ranged, repulsive potential,

$$V(h) = V_0 a \frac{\exp\left(-\frac{h-a}{b}\right)}{h-a}, \quad (42)$$

where  $h$  is the height of the RPY-particle’s centre from the surface,  $V_0 = 23k_B T$  sets the potential strength, and  $b = 0.5a$  is the range of the potential.

Simulations are performed using both the T-S and gDC schemes for three different time-step sizes  $\Delta t = 0.1, 0.2, 0.3\tau_D$ , where  $\tau_D = a^2/D = 6\pi\eta a^3/k_B T$  is the typical diffusive timescale associated with an RPY-particle. The divergence term,  $\nu$ , in Step 3 of the gDC algorithm (see section 3) is computed using both FD (25) and RFD (26). The solver tolerance for the GMRES and Lanczos algorithm is set to  $10^{-4}$ , and the finite difference parameter is  $\delta = 10^{-6}$ .

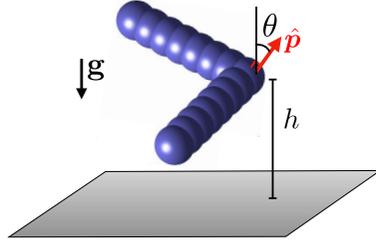
In order to obtain sufficient statistics, we run 36 different simulations for each value of  $\Delta t$ . Each simulation is initialized using a Monte Carlo generated sample of the Gibbs-Boltzmann distribution and is run to a final time of  $30000\tau_D$ , where the solution is recorded at every  $t = 0.3\tau_D$ .

Fig. 4 compares the height distributions from the T-S and gDC simulations with the true distribution obtained using Markov Chain Monte Carlo (MCMC). We see that, as expected, both schemes converge to the MCMC distribution as  $\Delta t$  decreases. The  $L_2$  errors for the distributions shown in the right panel indicate that the gDC achieves a similar accuracy and convergence rate as the T-S. However, the RFD computation of the  $\nu$  term in the gDC algorithm leads to larger errors than FD. This is particularly true for the largest time-step,  $\Delta t = 0.3\tau_D$ , for which gDC-RFD exhibit numerical stability issues due to particle overlaps across the wall. Similar stability problems were also encountered with the T-S scheme, resulting in several simulations being discarded and restarted when  $\Delta t = 0.3\tau_D$ . We suspect that the loss of stability with gDC-RFD is due to overestimations of  $(\partial_x \cdot \mathcal{V})^n$  caused by the random sampling  $\widetilde{\mathbf{W}}$ , which leads to large values of the corrective term  $\nu$ , therefore potentially moving the particle too far into the wall at the next time step.

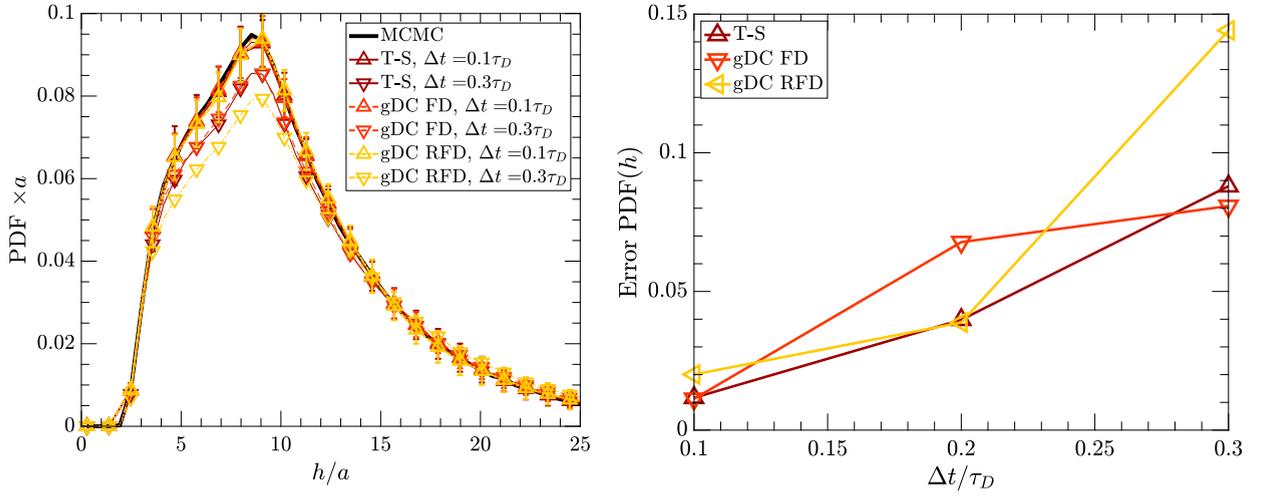
The orientation distribution, shown in Fig. 5, displays a similar trend, though the gDC exhibits a smaller error which does not change considerably with the time-step size.

### 6.3. Confined liquid crystal

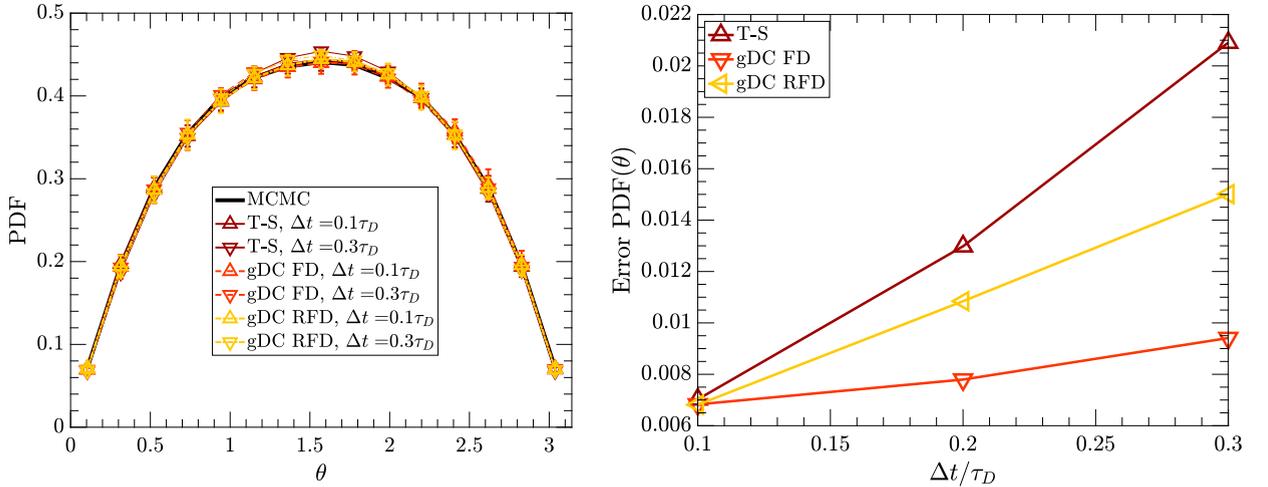
Inspired by the alignment of slender molecules in liquid crystals, this section examines the equilibrium distributions of confined suspensions of rod-shaped particles. In this set of simulations, each rod is constructed from 22 particles of radius



**Fig. 3.** Colloidal boomerang made of 15 RPY-particles above a no-slip boundary. The height,  $h$ , is the distance between the centre of the boomerang-point RPY-particle and the no-slip surface. The angle  $\theta$  is that between the vertical axis and the unit vector,  $\hat{\mathbf{p}}$ , that is orthogonal to both boomerang arms.



**Fig. 4.** (left panel) Height distribution of the boomerang above a no-slip surface; error bars (only shown for  $\Delta t = 0.1\tau_D$  for clarity) denote 95% confidence intervals. (right panel)  $L_2$ -error between the simulated distributions and the true distribution generated using MCMC.



**Fig. 5.** (left panel) Orientational distribution of the boomerang above a no-slip surface; error bars (only shown for  $\Delta t = 0.1\tau_D$  for clarity) denote 95% confidence intervals. (right panel)  $L_2$ -error between the simulated distributions and the true distribution generated using MCMC.

$a$  as illustrated in Fig. 6. The simulations are performed using fluctuating FCM in periodic computational domains with dimensions  $[0, L_x] \times [0, 2L_y] \times [0, L_z]$ . Confinement is introduced by first including slip boundaries at  $y = 0$  and  $y = L_y$  through boundary conditions  $\mathbf{u} \cdot \hat{\mathbf{e}}_y = 0$  and  $(\mathbf{I} - \hat{\mathbf{e}}_y \hat{\mathbf{e}}_y^T) \nabla \mathbf{u} = \mathbf{0}$  on the flow field. These conditions are enforced using an image system [40] for both the forces on the bodies, as well as the fluctuating stress, in the  $y > L_y$  half of the computational domain. Second, the rods are kept away from the channel walls by including a repulsive harmonic potential

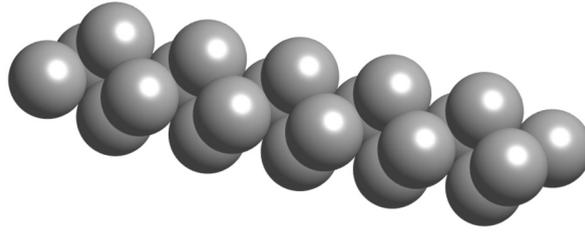


Fig. 6. A rod formed of 22 FCM-particles.

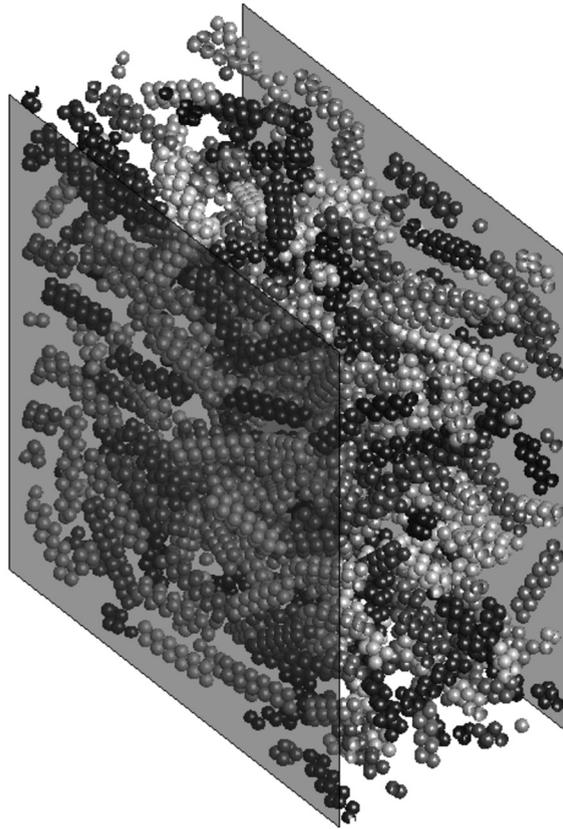


Fig. 7. A snapshot from a rod simulation with channel width  $L_y = L/2$ . Each rod has a randomly-selected greyscale colouring and the semi-transparent grey boundaries depict the locations of the slip boundaries.

$$V(r) = \begin{cases} \frac{K}{2} (r - b)^2, & r < b, \\ 0, & r \geq b, \end{cases} \tag{43}$$

on each FCM-particle making up the rod. Here,  $r$  is the distance between an FCM-particle centre and the wall,  $b = 2.2a$  is the cut-off distance, and  $K = \frac{40k_B T}{ab}$ . Along with this potential, overlap between FCM-particles is also discouraged using the soft potential

$$U(r) = \begin{cases} U_0 \left(1 + \frac{2a-r}{b}\right), & r < 2a, \\ U_0 \exp\left(\frac{2a-r}{b}\right), & r \geq 2a, \end{cases} \tag{44}$$

with  $r$  now denoting the centre-to-centre distance between 2 FCM-particles,  $U_0 = 20k_B T$  and  $b = 0.5a$ .

In the following simulations, the domain length is set to  $L_x = L_z = L = 77.65a$ . We consider three different channel widths, namely  $L_y = L/8$ ,  $L_y = L/4$  and  $L_y = L/2$ , to explore how confinement impacts the rod distributions. The number of rods in these simulations is taken to be 63, 125 and 250 respectively to maintain a constant volume fraction of 10%. A representative simulation snapshot shown in Fig. 7. Each simulation was run until a final time of  $50\tau_D/3$  and the time step size was set to  $\Delta t = \tau_D/600$ . Finally, the initial conditions for the gDC simulations were generated using samples from independent MCMC runs.

To measure the distribution of the rods within the channel, we compute the distribution of the  $y$ -coordinates of geometric centre of the rods. Additionally, we compute the distribution of the angle  $\theta$  between the rod-axis and the  $xz$ -plane, such that  $\theta = 0$  corresponds to the rod being parallel to the channel boundaries. Fig. 8 compares the  $y$ -coordinate distributions from the gDC simulations with the Gibbs-Boltzmann distributions calculated using MCMC. For all channel widths that were considered, good agreement is observed between these distributions. Similarly close agreement is found for the  $\theta$ -distributions in Fig. 9; the shape of the distribution is well-preserved even when viewed on a logarithmic scale. All of the  $y$ -coordinate distributions exhibit the expected symmetry about the midpoint of the channel at  $y = L_y/2$ , and have peaks in probability density near the slip boundaries. The position of the peaks depends on the width of the slip channel because of the finite thickness of the rods. It is also clear that the narrower the channel, the higher the probability of the  $y$ -positions associated with these peaks. The  $\theta$ -distributions all decrease from a maximum value at  $\theta \approx 0$ , although for the narrowest channel, the distribution exhibits a second local maximum just below  $\theta = \pi/8$ . The wider the channel, the more slowly the probability density decays, and hence the larger the range of likely orientations.

For the narrowest channel, the channel width,  $L_y = L/8 \approx 9.7a$ , is slightly larger than twice the diameter of the rod cross-section ( $2 \times 4a = 8a$ ), but less than the rod length of  $16.14a$ . As a result, the harmonic potential which keeps rods away from the slip surfaces will also induce a torque on the rods even for small deviations from  $\theta = 0$ . Thus, the rods are most likely to be found with an orientation  $\theta \approx 0$  in one of the two layers displaced from the channel centre. The distance between these layers balances the repulsive potentials between different rods and between particles and the walls. The smaller peak at the channel centre represents the small number of rods that get ‘jammed’ between the two layers. Additionally, as there is insufficient space for all rods to have  $\theta \approx 0$ , they also tend to span the channel diagonally with their ends trapped between the channel walls, leading to the secondary peak in the  $\theta$ -distribution below  $\theta = \pi/8$ .

As the channel width increases when we have  $L_y = L/4 \approx 19.4a$ , the central peak in the  $y$ -coordinate distribution becomes more pronounced, as shown in Fig. 8. Additionally, we observe a wider range of possible  $\theta$ -value for rods near the channel centre, although we note that a small, secondary peak in the  $\theta$  distribution remains as seen in the log-scale plot in Fig. 9. Note that this peak occurs at a larger  $\theta$  value than for the narrowest channel due to the increased channel width. Rods in the outer layers between this central layer and the slip boundaries are still most likely to have  $\theta \approx 0$ . As a consequence, the rods in the middle layer are also likely to align with  $\theta = 0$  with the two outer rod layers acting like the channel walls. Therefore, we see that the angle distribution retains its maximum at low  $\theta$ .

For the largest width, the  $\theta$  distribution still exhibits its peak value at  $\theta \approx 0$  as the rods nearest the slip boundaries continue to align parallel to the boundaries. We again observe peaks in the  $y$ -coordinate distribution close to the channel boundaries, though their magnitude is now reduced as shown in Fig. 8. The central peak observed in the two previous cases, however, has now spread out to form a series of peaks of decreasing magnitude in the interior of the channel. Note that this persistent pattern of density peaks which are largest near the walls and which decrease towards the channel centre is consistent with existing observations on the layering of colloidal particles near repulsive boundaries [62–66]. Rods appear to spend comparable amounts of time between and in these internal layers, seemingly making layers away from the slip surfaces more transient than those near the boundaries where rods tend to spend significant amounts of time. Finally, rods in the channel interior achieve a broader distribution of  $\theta$  values with no discernible secondary peak in the  $\theta$  distribution.

#### 6.4. Rheology of suspensions of Czech hedgehog colloids

In this section, we use fluctuating FCM with the gDC to perform nonequilibrium simulations that examine the rheology of Brownian Czech hedgehog-shaped (CH) particles, depicted in Fig. 10. This set of simulations is inspired by the recent experimental work of Bourrienne et al. [67], where suspensions of dendritic, silica particles, either hydrophobic or hydrophilic, were found to exhibit interesting rheological behaviour, including discontinuous shear thickening, depending on the particle interactions, as well as the relative strengths of shear and particle diffusion (Peclet number) and volume fraction.

In our simulations, the CH particles are formed of 19 FCM-particles of radius  $a$ , as shown in Fig. 10. This shape was chosen to reproduce, at some level, the complex structure and high specific surface area of the silica particles in the experiments. We consider two types of interactions between the CH particles, intended to offer holistic descriptions of both the hydrophobic and hydrophilic interactions in the motivating experimental work. For the first type, the FCM-particles comprising the CH particles repel at short-range through a soft potential, while attracting at longer range. Specifically, defining

$$\phi_{rc}(r) = A \exp\left(-\left(\frac{r-r_c}{\lambda}\right)^2\right) \quad (45)$$

for  $A = 5k_B T$  and  $\lambda = 0.5a$ , the hydrophobic-inspired ‘repel-attract’ potential is given by

$$U_1(r) = U(r) - \phi_{3a}(r), \quad (46)$$

where  $U(r)$  is the soft-sphere potential defined in eq. (44). For the second type of interaction, we have the hydrophilic-inspired ‘repel-attract-repel’ potential

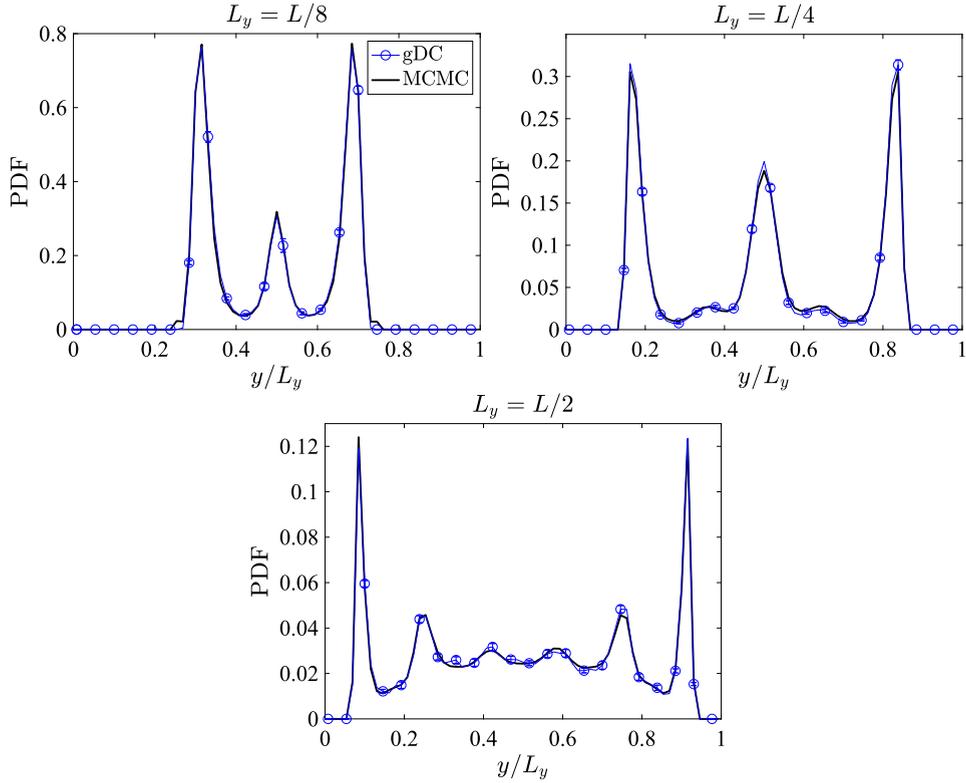


Fig. 8. The equilibrium distributions of rod positions across slip channels of widths  $L_y = L/8$ ,  $L_y = L/4$  and  $L_y = L/2$ .

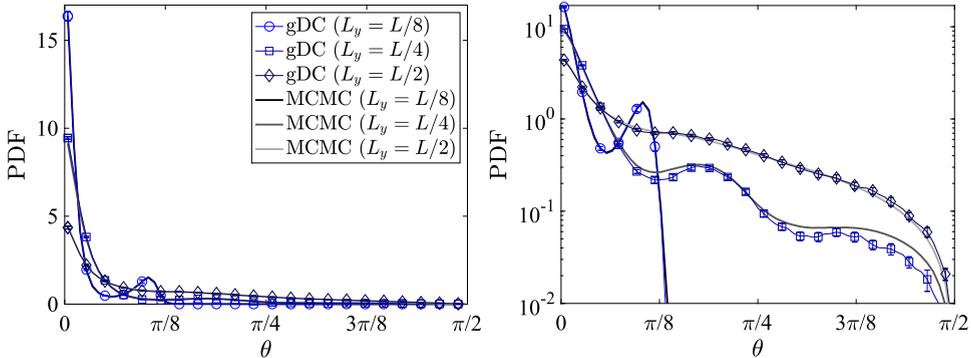
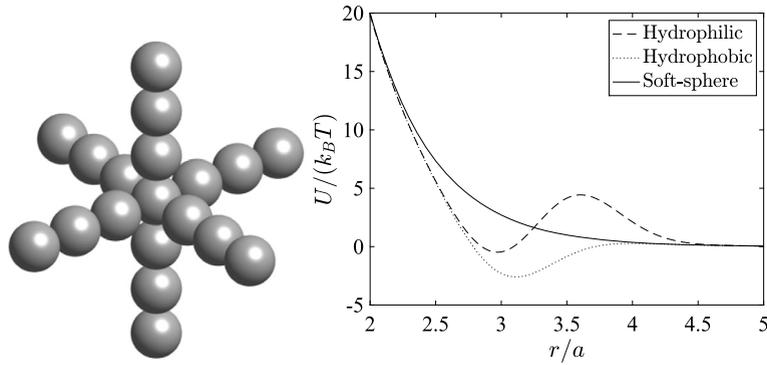


Fig. 9. The equilibrium distributions of rod orientations in slip channels of various widths, shown in both (left) linear and (right) logarithmic scales.

$$U_2(r) = U(r) - \phi_{3a}(r) + \phi_{7a/2}(r), \tag{47}$$

where a barrier is introduced in the potential just beyond the well. The resulting shapes of these potentials are shown in Fig. 10. Note that strictly these definitions are only  $r \geq 2a$ , with the potentials continuously extended by linear functions for  $r < 2a$  to ensure that the force is constant for blobs which overlap, just as for the original soft-sphere potential  $U(r)$ .

The simulations are performed in a triply-periodic computational domain with dimensions  $L_x = L_y = L_z = 77.65a$ . In order ensure that a strong rheological response is observed, a high volume fraction of 20% was used for all simulations. This corresponds to 1177 CH particles, yielding a mobility problem for the  $1177 \times 19 = 22363$  FCM-particles in each simulation. Simulations were run with a time step length of  $\Delta t = \tau_D/600$ . The initial conditions for the full simulations were constructed by running simulations that ignore thermal fluctuations and hydrodynamic interactions, i.e. using a diagonal mobility matrix, but retain interactions due to the potentials. In practice, we find that by allowing the CH particles to settle in this way before the shear is applied, the ‘long-time’ velocity profile is realised more quickly. The simulations were run to final times between  $35\tau_D/3$  and  $50\tau_D$  and in all cases, a regular oscillatory velocity profile emerges long before the end of the simulation.



**Fig. 10.** An illustration of a CH particle formed of 19 FCM-particles of radius  $a$  (left), and the different interaction potentials for suspended CH particles (right).

To measure the suspension viscosity from this emergent velocity field, we adopt the approach described by Vázquez et al. [68]. Here, the background fluid is forced using the periodic force density

$$\mathbf{f}(x, y, z) = f_0 \sin(ky) \hat{\mathbf{e}}_x \quad (48)$$

where  $f_0$  is the force magnitude and  $k = 2\pi/L_y$  is the wavenumber. The  $x$ -component of the resulting velocity field of the suspension is averaged over time and in the  $x$ - and  $z$ -directions, resulting in a sinusoidal velocity profile in  $y$  with magnitude  $v_x$ . The effective viscosity of the suspension can then be defined as

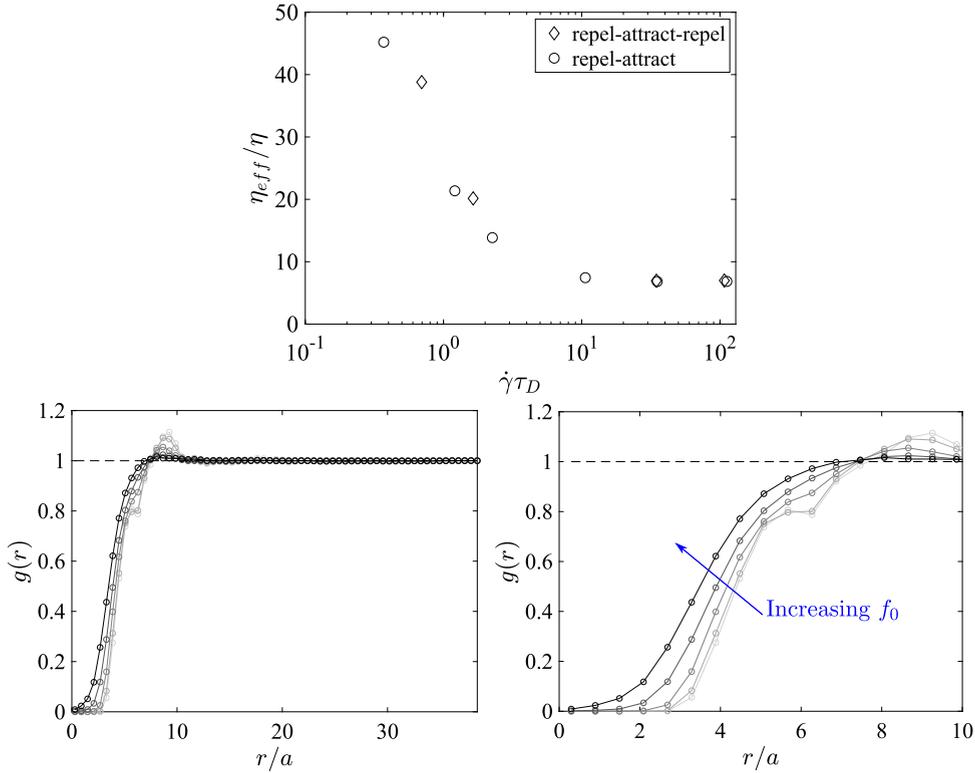
$$\eta_{eff} = \frac{f_0}{k^2 v_x}. \quad (49)$$

In Fig. 11, the ratio of the effective viscosity  $\eta_{eff}$  to the underlying fluid viscosity  $\eta$  is compared to the shear rate  $\dot{\gamma}$  for suspensions with both types of particle interactions. In this work,  $\dot{\gamma}$  is taken as the maximum slope of the sinusoidal velocity profile. We find that both the ‘repel-attract’ and ‘repel-attract-repel’ exhibit rapid shear thinning at lower shear rates, as found in the original experiments of Bourrienne et al. We do not observe, however, any shear thickening for the ‘repel-attract-repel’ CH particle suspensions whose interactions are meant to be similar to those of the hydrophilic particles in the experiments. Bourrienne et al. suggest that solid friction is required for the onset of shear thickening and that hydrogen bonding is additionally required for DST to be observed. Both of these interactions are absent from our simulations.

Given that this is a non-equilibrium problem, it is of particular interest to examine distributions of the particle states since they cannot be sampled using equilibrium procedures such as MCMC. Fig. 11 shows the radial distribution function  $g(r)$  where  $r$  is the centre-to-centre distance of  $r$  for the CH particles with ‘repel-attract’ interactions; the distribution for ‘repel-attract-repel’ interactions is similar. Simulations with higher values of  $f_0$  correspond to darker curves in the figure. At the smallest  $f_0$ , where the particle motion is dominated by the Brownian motion,  $g(r)$  increases from zero to a peak value at separations between  $r/a \approx 8$  and  $r/a \approx 10$ . At larger separations,  $g(r)$  gradually decreases to approach values for a uniform distribution. As  $f_0$ , and hence the shear rate, is increased, this peak diminishes and the distribution approaches a monotonically increasing function.

## 7. Summary and conclusions

In this paper, we developed the gDC scheme for the efficient time integration of hydrodynamically interacting rigid Brownian particles. The gDC automatically accounts for Brownian drift in advancing particle positions and orientations while retaining the need to perform only a single, full mobility problem at each time step, while achieving accuracy comparable to the current state-of-the-art. The gDC has been designed to be used with fast methods for applying the mobility matrix and generating random increments with the correct covariance and is ideally suited for grid-based computations that take advantage of fluctuating hydrodynamics. For some approaches, such as positively-split Ewald [22] and discretisation of the fluctuating boundary integral equations [41], the mobility matrix  $\mathbf{M}$  is split with one (or more) of the resulting parts being treated using the Lanczos method to generate their contribution to the random velocity and the remainder handled using a fluctuating hydrodynamics-like approach. The matrices handled using the Lanczos algorithm are typically sparse, allowing for rapid matrix-vector multiplication, and hence an efficient application of Lanczos. As such, we expect the gDC to provide efficient time-integration for these methods as well. More generally, any fast method to compute the random velocities is compatible with the gDC. Additionally, many of the additional computations that the gDC requires are local to each particle, enabling parallelisation and large-scale computation. Our simulations demonstrate each of these features, as well as the applicability of gDC to enable simulations that complement experiments in modern suspension mechanics and colloidal science.



**Fig. 11.** The effective viscosity of both types of suspension for different dimensionless shear rates (top), and the radial distribution functions for the different ‘repel-attract’ simulations (bottom). The bottom right plot is displaying the same data as the bottom left plot but over a smaller  $r$  range. The colour of the curves fades from black to white as  $f_0$  decreases.

### CRediT authorship contribution statement

All authors contributed equally to the research design and writing of the manuscript. BD performed the simulations in Section 6.2. TAW performed all other simulations.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

We thank Aleksandar Donev for his comments and valuable discussions on the use of the gDC with the Positively Split Ewald method. EEK and TAW gratefully acknowledge support from EPSRC Grant EP/P013651/1. TAW is also thankful for funding through an EPSRC Studentship (Ref: 1832024). BD acknowledges support from the French National Research Agency (ANR), under award ANR-20-CE30-0006. B.D. also thanks the NVIDIA Academic Partnership program for providing GPU hardware for performing some of the simulations reported here.

### Appendix A. Quaternions

In order to describe the state of an arbitrarily shaped particle, its orientation must be provided along with its position in space. This requires providing the rotation mapping from some reference configuration to the current orientation of the particle and thus a parametrisation for three dimensional rotations must be chosen. Perhaps the most natural choice is the axis-angle representation  $\theta = \theta\hat{e}$ , which describes the anti-clockwise rotation through an angle  $\theta$  about the axis  $\hat{e}$ , but subsequent rotations cannot be easily combined under this representation. This problem is shared by the Euler angles, which further suffer from the need to specify a convention for their application as well as the gimbal lock phenomenon, wherein certain choices of one angle result in a loss of a degree of freedom. These issues are avoided when using rotation matrices, but they require significant amounts of redundant storage – storing nine variables to represent three degrees of

freedom – and they must be projected back to the nearest orthonormal matrix in the event of accumulated numerical error. An alternative representation, and the one used in this work, is that of the unit quaternions.

The unit quaternions can be identified with the unit sphere in  $\mathbb{R}^4$ , although they are often viewed as having separate scalar and vector parts and thus written as  $\mathbf{q} = (q_0, \mathbf{q})$ , where  $q_0 \in \mathbb{R}$  and  $\mathbf{q} \in \mathbb{R}^3$ ; the unit length constraint is then expressed as  $q_0^2 + \|\mathbf{q}\|^2 = 1$ . They form a group under the non-commuting, associative multiplication known as the Hamilton product,

$$\begin{aligned} \mathbf{p} \bullet \mathbf{q} &= (p_0, \mathbf{p}) \bullet (q_0, \mathbf{q}) \\ &= (p_0q_0 - \mathbf{p} \cdot \mathbf{q}, p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}), \end{aligned} \quad (\text{A.1})$$

with inverses  $\mathbf{q}^{-1} = (q_0, -\mathbf{q})$  and identity  $\mathbf{l}_q = (1, \mathbf{0})$ . It is through the Hamilton product and the polar form for unit quaternions,

$$\mathbf{q} = \left( \cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\hat{\mathbf{e}} \right), \quad (\text{A.2})$$

that the unit quaternions can be associated with the anti-clockwise rotation through an angle  $\theta$  about the axis  $\hat{\mathbf{e}}$ , with the image,  $\mathbf{x}'$ , of a vector  $\mathbf{x}$  under this rotation satisfying

$$(0, \mathbf{x}') = \mathbf{q} \bullet (0, \mathbf{x}) \bullet \mathbf{q}^{-1}. \quad (\text{A.3})$$

Thus it is clear that the Hamilton product allows us to describe successive quaternion-represented rotations as another unit quaternion; rotating according to  $\mathbf{p} \bullet \mathbf{q}$  is equivalent to first rotating according to  $\mathbf{q}$  before rotating according to  $\mathbf{p}$ . Consequently, the unit quaternions improve on the rotation matrix representation of rotations by reducing the storage cost from nine to four variables, reducing the cost of correcting accumulated errors to normalising a vector and by providing an intuitive correspondence with the axis-angle description.

If  $\boldsymbol{\Omega}$  denotes the angular velocity of the body, then the unit quaternion associated with the orientation of the body, viewed as an element of  $\mathbb{R}^4$ , satisfies

$$\frac{d\mathbf{q}}{dt} = \boldsymbol{\Psi}_q \boldsymbol{\Omega}, \quad (\text{A.4})$$

where

$$\boldsymbol{\Psi}_q = \frac{1}{2} \begin{bmatrix} -\mathbf{q}^\top \\ q_0 \mathbf{I} + [\times \mathbf{q}] \end{bmatrix} \in \mathbb{R}^{4 \times 3} \quad (\text{A.5})$$

satisfies  $\boldsymbol{\Psi}_q \boldsymbol{\Omega} = \frac{1}{2} (0, \boldsymbol{\Omega}) \bullet \mathbf{q}$ .

Given an initial orientation quaternion  $\mathbf{q}_0$  at time  $t = 0$ , this ODE could be solved using standard linear techniques. However, this will produce a quaternion with non-unit norm in general. The resulting quaternion could be projected back to unit length, but this represents an unnecessary source of error. Instead, we utilise the Lie algebra associated with the group of unit quaternions to perform multiplicative updates which preserve the norm to machine precision. Following the general frameworks of Iserles et al. [69] and Faltinsen et al. [70], we observe that the solution of the ODE for the orientation quaternion,

$$\begin{aligned} \frac{d\mathbf{q}}{dt} &= \boldsymbol{\Psi}_q \boldsymbol{\Omega}, \\ \mathbf{q}(0) &= \mathbf{q}_0, \end{aligned} \quad (\text{A.6})$$

is given by

$$\mathbf{q}(t) = \exp(\mathbf{u}(t)) \bullet \mathbf{q}_0, \quad (\text{A.7})$$

where  $\mathbf{u}$  is the so-called Lie algebra element satisfying

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{D}_u^{-1} \boldsymbol{\Omega}, \\ \mathbf{u}(0) &= 0. \end{aligned} \quad (\text{A.8})$$

The exponential function for this Lie algebra corresponds to the polar form of the unit quaternions, i.e.

$$\exp(\mathbf{u}) = \left( \cos\left(\frac{\|\mathbf{u}\|}{2}\right), \sin\left(\frac{\|\mathbf{u}\|}{2}\right) \frac{\mathbf{u}}{\|\mathbf{u}\|} \right), \quad (\text{A.9})$$

and  $\mathbf{D}_u^{-1}$  is the ‘dexpinv’ matrix defined by

$$\mathbf{D}_{\mathbf{u}}^{-1} = \mathbf{I} - \frac{1}{2} [\mathbf{u} \times] - \frac{1}{2\|\mathbf{u}\|^2} \left( \|\mathbf{u}\| \cot\left(\frac{\|\mathbf{u}\|}{2}\right) - 2 \right) [\mathbf{u} \times]^2; \quad (\text{A.10})$$

the Lie algebra considered here is isomorphic to  $(\mathbb{R}^3, \times)$  and so  $\mathbf{u}$  should simply be thought of as a 3-vector. It should also be noted that  $\mathbf{D}_{\mathbf{u}}^{-1}$  is continuously extended by  $\mathbf{D}_{\mathbf{0}}^{-1} = \mathbf{I}$ .

As discussed by Falinsen et al. [70], the numerical properties of the solution are preserved by in fact using  $\mathbf{u}$  to describe the rotation we apply to the current quaternion rather than the initial condition  $\mathbf{q}_0$ . That is, at a given discrete time  $n$  (i.e.  $t = t^n = n\Delta t$ ) we solve for  $\mathbf{u}^{n+1}$  assuming that  $\mathbf{u}^n = \mathbf{0}$  and define  $\mathbf{q}^{n+1} = \exp(\mathbf{u}^{n+1}) \bullet \mathbf{q}^n$ . The fact that the Lie algebra element is always expressed in a coordinate system such that it is zero at the start of the current time-step will allow us to make some useful simplifications in Appendix E.

## Appendix B. Expressing torques in terms of the Lie algebra element

Delong et al. [60] considered the change in energy due to an infinitesimal rotation to argue that if the quaternion  $\mathbf{q}$  describes the orientation of a particle then the torque it experiences due to a potential  $U$  can be written

$$\mathbf{T} = -\Psi_{\mathbf{q}}^{\top} \partial_{\mathbf{q}} U, \quad (\text{B.1})$$

where the matrix  $\Psi_{\mathbf{q}}$  is as defined in eq. (A.5). In this section, we employ a similar argument to show that the analogous result holds for the corresponding Lie algebra element.

Consider the torque  $\mathbf{T} = -\partial_{\varphi} U$  generated by the potential  $U$ , where  $\varphi$  is the oriented angle. We want to express this torque directly in terms of the Lie algebra element  $\mathbf{u}$ . Recalling eq. (A.8), we have

$$\begin{aligned} d\mathbf{u} &= \mathbf{D}_{\mathbf{u}}^{-1} d\varphi \\ &= d\varphi - \frac{1}{2} \mathbf{u} \times d\varphi - g(\mathbf{u}) \mathbf{u} \times (\mathbf{u} \times d\varphi), \end{aligned} \quad (\text{B.2})$$

where we have defined  $g(\mathbf{u}) = \left( \|\mathbf{u}\| \cot\left(\frac{\|\mathbf{u}\|}{2}\right) - 2 \right) / (2\|\mathbf{u}\|^2)$  for legibility. Hence, the change in energy  $dU$  due to an infinitesimal rotation  $d\varphi$  satisfies

$$\begin{aligned} -\mathbf{T} \cdot d\varphi &= dU = \partial_{\mathbf{u}} U \cdot d\mathbf{u} \\ &= \partial_{\mathbf{u}} U \cdot d\varphi - \frac{1}{2} (\mathbf{u} \times d\varphi) \cdot \partial_{\mathbf{u}} U - g(\mathbf{u}) [\mathbf{u} \times (\mathbf{u} \times d\varphi)] \cdot \partial_{\mathbf{u}} U. \end{aligned} \quad (\text{B.3})$$

Since the scalar triple product is invariant under cyclic permutation of the vectors we observe that  $(\mathbf{u} \times d\varphi) \cdot \partial_{\mathbf{u}} U = (\partial_{\mathbf{u}} U \times \mathbf{u}) \cdot d\varphi$  and  $[\mathbf{u} \times (\mathbf{u} \times d\varphi)] \cdot \partial_{\mathbf{u}} U = [\partial_{\mathbf{u}} U \times \mathbf{u}] \cdot (\mathbf{u} \times d\varphi) = [(\partial_{\mathbf{u}} U \times \mathbf{u}) \times \mathbf{u}] \cdot d\varphi$ , giving us

$$dU = \left[ \partial_{\mathbf{u}} U - \frac{1}{2} (\partial_{\mathbf{u}} U \times \mathbf{u}) - g(\mathbf{u}) (\partial_{\mathbf{u}} U \times \mathbf{u}) \times \mathbf{u} \right] \cdot d\varphi. \quad (\text{B.4})$$

This permits the identification of the torque as

$$\begin{aligned} \mathbf{T} &= - \left[ \partial_{\mathbf{u}} U - \frac{1}{2} (\partial_{\mathbf{u}} U \times \mathbf{u}) - g(\mathbf{u}) (\partial_{\mathbf{u}} U \times \mathbf{u}) \times \mathbf{u} \right] \\ &= - \left[ \mathbf{I} - \frac{1}{2} [\times \mathbf{u}] - g(\mathbf{u}) [\times \mathbf{u}]^2 \right] \partial_{\mathbf{u}} U \\ &= -\mathbf{D}_{\mathbf{u}}^{-\top} \partial_{\mathbf{u}} U. \end{aligned} \quad (\text{B.5})$$

## Appendix C. Equivalence between the Lie algebra and quaternion SDEs

Having shown that the torque on a body can be written in terms of the unit quaternion describing its orientation,  $\mathbf{q}$ , according to eq. (B.1), Delong et al. [60] argue that  $\mathbf{q}$  should satisfy the Itô Langevin SDE

$$d\mathbf{q} = \left( k_B T \partial_{\mathbf{q}} \cdot \left( \Psi_{\mathbf{q}} \mathbf{N}_{\Omega T} \Psi_{\mathbf{q}}^{\top} \right) - \Psi_{\mathbf{q}} \mathbf{N}_{\Omega T} \Psi_{\mathbf{q}}^{\top} \partial_{\mathbf{q}} U \right) dt + \sqrt{2k_B T} \Psi_{\mathbf{q}} \mathbf{N}_{\Omega T}^{1/2} d\mathbf{W}, \quad (\text{C.1})$$

where  $\mathbf{N}_{\Omega T}$  is the rotational mobility matrix mapping the torque on the body,  $\mathbf{T} = -\Psi_{\mathbf{q}}^{\top} \partial_{\mathbf{q}} U$ , to its angular velocity,  $\boldsymbol{\Omega}$ , and  $\mathbf{N}_{\Omega T}^{1/2}$  satisfies  $\mathbf{N}_{\Omega T}^{1/2} \left( \mathbf{N}_{\Omega T}^{1/2} \right)^{\top} = \mathbf{N}_{\Omega T}$  as required by the fluctuation-dissipation theorem. In this work, we integrate unit quaternions in time using their associated Lie algebra elements rather than handle the quaternions directly. Thus, we seek an Itô SDE for the Lie algebra element such that the resulting quaternion, defined by eq. (A.7), satisfies eq. (C.1) and hence that the rotational dynamics in our approach are equivalent to those in the work of Delong et al. [60]. Given our result in

Appendix B, namely that the torque on the body can be expressed using the Lie algebra element as  $\mathbf{T} = -\mathbf{D}_{\mathbf{u}}^{-\top} \partial_{\mathbf{u}} U$ , we propose that the appropriate Itô SDE is

$$d\mathbf{u} = \left( k_B T \partial_{\mathbf{u}} \cdot \left( \mathbf{D}_{\mathbf{u}}^{-1} \mathbf{N}_{\Omega T} \mathbf{D}_{\mathbf{u}}^{-\top} \right) - \mathbf{D}_{\mathbf{u}}^{-1} \mathbf{N}_{\Omega T} \mathbf{D}_{\mathbf{u}}^{-\top} \partial_{\mathbf{u}} U \right) dt + \sqrt{2k_B T} \mathbf{D}_{\mathbf{u}}^{-1} \mathbf{N}_{\Omega T}^{1/2} d\mathbf{W}, \quad (\text{C.2})$$

the direct analogue of eq. (C.1). In the remainder of this section, we demonstrate that this does indeed result in the correct Itô SDE for the unit quaternion.

Loosely speaking, the approach here is to convert the Itô SDE for  $\mathbf{u}$  to the Stratonovich interpretation, for which many standard calculus results are recovered, then to mirror the deterministic result “ $d\mathbf{u}/dt = \mathbf{D}_{\mathbf{u}}^{-1} \boldsymbol{\Omega} \implies d\mathbf{q}/dt = \boldsymbol{\Psi}_{\mathbf{q}} \boldsymbol{\Omega}$ ” and finally to recover the Itô interpretation. To that end, recall that if  $\mathbf{y}$  satisfies the Itô SDE

$$d\mathbf{y} = \mathbf{h}(\mathbf{y}) dt + \boldsymbol{\gamma}(\mathbf{y}) d\mathbf{W}, \quad (\text{C.3})$$

then the equivalent Stratonovich form of the SDE is

$$d\mathbf{y} = \left( \mathbf{h}(\mathbf{y}) - \frac{1}{2} \mathbf{c}(\mathbf{y}) \right) dt + \boldsymbol{\gamma}(\mathbf{y}) \circ d\mathbf{W}, \quad (\text{C.4})$$

where  $c_i = \gamma_{jk} \partial \gamma_{ik} / \partial y_j$  and  $\circ$  denotes the Stratonovich product. Writing  $\mathbf{D}^{-1} = \mathbf{D}_{\mathbf{u}}^{-1}$  and  $\mathbf{N} = \mathbf{N}_{\Omega T}$  (and hence  $\mathbf{N}^{1/2} = \mathbf{N}_{\Omega T}^{1/2}$ ) to simplify notation, we make the identification  $\gamma_{ij} = \sqrt{2k_B T} D_{ik}^{-1} N_{kj}^{1/2}$  by comparing eq. (C.2) to eq. (C.3) for  $\mathbf{y} = \mathbf{u}$  and hence obtain

$$\begin{aligned} \frac{1}{2} c_i &= k_B T D_{jb}^{-1} N_{bk}^{1/2} \frac{\partial \left( D_{ia}^{-1} N_{ak}^{1/2} \right)}{\partial u_j} \\ &= k_B T \left( \frac{\partial \left( D_{ia}^{-1} N_{ak}^{1/2} N_{kb}^{1/2} D_{jb}^{-1} \right)}{\partial u_j} - D_{ia}^{-1} N_{ak}^{1/2} \frac{\partial \left( D_{jb}^{-1} N_{bk}^{1/2} \right)}{\partial u_j} \right) \\ &= k_B T \left( \frac{\partial \tilde{N}_{ij}}{\partial u_j} - D_{ia}^{-1} N_{ak}^{1/2} \frac{\partial \left( D_{jb}^{-1} N_{bk}^{1/2} \right)}{\partial u_j} \right). \end{aligned} \quad (\text{C.5})$$

Now we claim that  $\partial D_{jb}^{-1} / \partial u_j = 0$  and hence that  $\partial \left( D_{jb}^{-1} N_{bk}^{1/2} \right) / \partial u_j = D_{jb}^{-1} \partial N_{bk}^{1/2} / \partial u_j$ . For  $\mathbf{u} \neq \mathbf{0}$ , we can simply differentiate eq. (4) and observe that  $\partial D_{jb}^{-1} / \partial u_j \propto \left( \mathbf{u} \times \mathbf{j} \right)_b = 0$ . The case  $\mathbf{u} = \mathbf{0}$  can be shown easily using the limit definition of the partial derivative. Thus

$$\frac{1}{2} c_i = k_B T \left( \frac{\partial \tilde{N}_{ij}}{\partial u_j} - D_{ia}^{-1} N_{ak}^{1/2} D_{jb}^{-1} \frac{\partial N_{bk}^{1/2}}{\partial u_j} \right), \quad (\text{C.6})$$

which allows us to cast eq. (C.2) in Stratonovich form as

$$\begin{aligned} du_i &= \left( k_B T D_{ia}^{-1} N_{ak}^{1/2} D_{jb}^{-1} \frac{\partial N_{bk}^{1/2}}{\partial u_j} - \tilde{N}_{ij} \frac{\partial U}{\partial u_j} \right) dt + \sqrt{2k_B T} D_{ia}^{-1} N_{aj}^{1/2} \circ dW_j \\ &= D_{ia}^{-1} \left[ \left( k_B T N_{ak}^{1/2} D_{jb}^{-1} \frac{\partial N_{bk}^{1/2}}{\partial u_j} - N_{ab} D_{jb}^{-1} \frac{\partial U}{\partial u_j} \right) dt + \sqrt{2k_B T} N_{aj}^{1/2} \circ dW_j \right]. \end{aligned} \quad (\text{C.7})$$

Applying Theorem 5.1 of Malham and Wiese [71] (recognizing the  $\boldsymbol{\xi}$  as the columns of  $\mathbf{N}^{1/2}$ ) we find that the corresponding orientation quaternion satisfies the Stratonovich equation

$$dq_i = \psi_{ia} \left[ \left( k_B T N_{ak}^{1/2} D_{jb}^{-1} \frac{\partial N_{bk}^{1/2}}{\partial u_j} - N_{ab} D_{jb}^{-1} \frac{\partial U}{\partial u_j} \right) dt + \sqrt{2k_B T} N_{aj}^{1/2} \circ dW_j \right], \quad (\text{C.8})$$

where we have dropped the  $\mathbf{q}$  subscript from  $\boldsymbol{\Psi}_{\mathbf{q}}$  for legibility. Recalling Appendix B, we make the identification  $D_{jb}^{-1} \partial / \partial u_j = \psi_{jb} \partial / \partial q_j$  and thus the above becomes

$$dq_i = \psi_{ia} \left[ \left( k_B T N_{ak}^{1/2} \psi_{jb} \frac{\partial N_{bk}^{1/2}}{\partial q_j} - N_{ab} \psi_{jb} \frac{\partial U}{\partial q_j} \right) dt + \sqrt{2k_B T} N_{aj}^{1/2} \circ dW_j \right]. \quad (\text{C.9})$$

Observing that the form of this equation is analogous to eq. (C.7) and noting that  $\partial\Psi_{jb}/\partial q_j = 0$  since the  $j^{\text{th}}$  row of  $\Psi$  doesn't depend on  $q_j$ , we simply perform the above procedure in reverse to obtain eq. (C.1).

Having established the appropriate Itô SDE for the Lie algebra element, it is simple to couple this to the corresponding equation for the position of the body to produce the complete Itô SDE for the state of the body we presented in eq. (9).

#### Appendix D. Algorithm for Euler-Maruyama scheme

For each time-step  $n = 0, 1, 2, \dots$

1. Generate a vector  $\mathbf{W}$  (or tensor  $\mathbf{W}$  if using fluctuating hydrodynamics) of  $\mathcal{N}(0, 1)$  random variables.
2. Solve the full problem

$$\begin{bmatrix} \mathbf{M}^n & -\mathbf{K}^n \\ -(\mathbf{K}^\top)^n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \mathbf{V}^n \end{bmatrix} = \begin{bmatrix} -\check{\mathbf{v}}^n = -\sqrt{\frac{2k_B T}{\Delta t}} (\mathbf{M}^{1/2})^n \mathbf{W} \\ -\mathbf{F}^n \end{bmatrix} \quad (\text{D.1})$$

for

$$\mathbf{V}^n = \mathbf{N}^n \left( \mathbf{F}^n + (\mathbf{K}^\top)^n (\mathbf{M}^{-1})^n \check{\mathbf{v}}^n \right) = \mathbf{N}^n \mathbf{F}^n + \check{\mathbf{V}}^n, \quad (\text{D.2})$$

3. Move the rigid bodies to the next time-step according to

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{V}^n, \quad (\text{D.3})$$

$$\mathbf{q}_p^{n+1} = \exp(\mathbf{u}_p^{n+1}) \bullet \mathbf{q}_p^n \text{ for } 1 \leq p \leq N. \quad (\text{D.4})$$

#### Appendix E. Proof of consistency for gDC

In this section we offer a proof of the consistency of the gDC algorithm (section 3); i.e. that the increment  $\Delta \mathbf{x}^n = \mathbf{x}^{n+1} - \mathbf{x}^n = \nu \Delta t \mathbf{V}^m$  produces the correct first and second moments for freely-diffusing rigid bodies,

$$\langle \Delta \mathbf{x}^n \rangle = k_B T \Delta t (\partial_{\mathbf{x}} \cdot \tilde{\mathbf{N}})^n, \quad (\text{E.1})$$

$$\langle \Delta \mathbf{x}^n (\Delta \mathbf{x}^n)^\top \rangle = 2k_B T \Delta t \tilde{\mathbf{N}}^n, \quad (\text{E.2})$$

to first-order in time. We begin by observing that  $\mathbf{D}_0^{-1} = \mathbf{I} \implies \boldsymbol{\Phi}^n = \mathbf{I}$  (recall Appendix A) and hence that eq. (E.2) reduces to

$$\langle \Delta \mathbf{x}^n (\Delta \mathbf{x}^n)^\top \rangle = 2k_B T \Delta t \mathbf{N}^n. \quad (\text{E.3})$$

To similarly simplify eq. (E.1) we introduce the shorthand  $\partial_i \equiv (\partial_{\mathbf{x}})_i = \partial/\partial x_i$  and, summing over repeated indices, expand

$$\begin{aligned} (\partial_j \tilde{\mathbf{N}}_{ij})^n &= (\partial_j (\Phi_{ia} N_{ab} \Phi_{jb}))^n \\ &= \Phi_{ia}^n N_{ab}^n (\partial_j \Phi_{jb})^n + \Phi_{ia}^n \Phi_{jb}^n (\partial_j N_{ab})^n + N_{ab}^n \Phi_{jb}^n (\partial_j \Phi_{ia})^n \\ &= N_{ib}^n (\partial_j \Phi_{jb})^n + (\partial_j N_{ij})^n + N_{aj}^n (\partial_j \Phi_{ia})^n, \end{aligned} \quad (\text{E.4})$$

since  $\Phi_{kl}^n = \delta_{kl}$ . In Appendix C we show that  $\partial_j (\mathbf{D}_u^{-1})_{jb}^n = 0$  and hence the first term vanishes. Furthermore, applying the definition of partial differentiation yields  $(\partial_j (\mathbf{D}_u^{-1})_{ia}^n)^n = -\frac{1}{2} \epsilon_{ija}$ , and hence the components of the third term which don't vanish in the differentiation vanish as the inner product of a symmetric and skew-symmetric matrix. Thus we obtain the simpler expression

$$\langle \Delta \mathbf{x}^n \rangle = k_B T \Delta t (\partial_{\mathbf{x}} \cdot \mathbf{N})^n \quad (\text{E.5})$$

for the first moment in expectation. Note that these simplified forms for eqs. (E.3) and (E.5) mean that the gDC scheme can be implemented without any explicit mention of  $\boldsymbol{\Phi}$  (and hence  $\mathbf{D}_u^{-1}$ ).

To simplify notation, for the remainder of this section all quantities are evaluated at time  $n$  unless explicitly indicated otherwise. We start by expanding  $\mathbf{V}^m$  about time  $t_n$  to obtain

$$V_i^m = V_i + \frac{\Delta t}{2} \mathcal{V}_j \partial_j V_i + \frac{\Delta t^2}{8} \mathcal{V}_k \mathcal{V}_j \partial_k \partial_j V_i + \mathcal{O}(\Delta t). \quad (\text{E.6})$$

Recalling that  $\nu = 1 + \frac{\Delta t}{2} \partial_j \mathcal{V}_j$  we obtain

$$\begin{aligned} vV_i^m &= V_i + \frac{\Delta t}{2} \mathcal{V}_j \partial_j V_i + \frac{\Delta t^2}{8} \mathcal{V}_k \mathcal{V}_j \partial_k \partial_j V_i \\ &\quad + \frac{\Delta t}{2} V_i \partial_j \mathcal{V}_j + \frac{\Delta t^2}{4} \mathcal{V}_j \partial_j (V_i) \partial_k (\mathcal{V}_k) + \mathcal{O}(\Delta t). \end{aligned} \quad (\text{E.7})$$

We recognize the second and fourth terms on the right hand side as a product rule expansion and re-write the above as

$$\begin{aligned} vV_i^m &= V_i + \frac{\Delta t}{2} \partial_j (\mathcal{V}_j V_i) + \frac{\Delta t^2}{8} \mathcal{V}_k \mathcal{V}_j \partial_k \partial_j V_i \\ &\quad + \frac{\Delta t^2}{4} \mathcal{V}_j \partial_j (V_i) \partial_k (\mathcal{V}_k) + \mathcal{O}(\Delta t). \end{aligned} \quad (\text{E.8})$$

All but the second term vanish in expectation, leaving us with

$$\langle vV_i^m \rangle = \frac{\Delta t}{2} \partial_j \langle \mathcal{V}_j V_i \rangle + \mathcal{O}(\Delta t). \quad (\text{E.9})$$

Note that if we are calculating the divergence using RFD instead, eqs. (E.7) and (E.8) only hold in expectation and are obtained by exploiting the independence of  $\mathbf{W}$  and  $\tilde{\mathbf{W}}$  to separate terms. Nevertheless, we obtain eq. (E.9).

Now, recalling the definitions of  $\mathcal{V}$  and  $\mathbf{V}$  we have

$$\langle \mathcal{V}_j V_i \rangle = \frac{2k_B T}{\Delta t} N_{ik} K_{kl}^\top M_{lm}^{-1} M_{mn}^{1/2} (K^\top K)^{-1}_{jq} K_{qr}^\top M_{rs}^{1/2} \langle W_n W_s \rangle. \quad (\text{E.10})$$

Using  $\langle W_n W_s \rangle = \delta_{ns}$  it is easy to verify that

$$\langle \mathcal{V}_j V_i \rangle = \frac{2k_B T}{\Delta t} N_{ij}. \quad (\text{E.11})$$

Substituting this into eq. (E.9) and multiplying through by  $\Delta t$  reveals that eq. (E.5) is satisfied to first-order. To show the same for eq. (E.3), recall eq. (E.8) and observe that

$$\begin{aligned} \Delta x_i \Delta x_a &= \Delta t^2 v^2 V_i^m V_a^m \\ &= \Delta t^2 V_i V_a + \frac{\Delta t^3}{2} [\partial_j (V_i \mathcal{V}_j V_a) + V_i V_a \partial_j \mathcal{V}_j] + \mathcal{O}(\Delta t^2). \end{aligned} \quad (\text{E.12})$$

The second term vanishes in expectation and we have

$$\langle \Delta x_i \Delta x_a \rangle = \Delta t^2 \langle V_i V_a \rangle + \mathcal{O}(\Delta t^2). \quad (\text{E.13})$$

If we are using RFD to calculate the divergence, then the expansion in eq. (E.12) only holds in expectation but we obtain eq. (E.13) nonetheless.

Again recalling the definition of  $\mathbf{V}$  from section 3 we find that

$$\Delta t^2 \langle V_i V_a \rangle = 2k_B T \Delta t N_{ik} K_{kl}^\top M_{lm}^{-1} M_{mn}^{1/2} N_{ab} K_{bc}^\top M_{cd}^{-1} M_{de}^{1/2} \langle W_n W_e \rangle. \quad (\text{E.14})$$

In a similar way to before, substituting  $\langle W_n W_e \rangle = \delta_{ne}$  and summing over repeated indices yields

$$\Delta t^2 \langle V_i V_a \rangle = 2k_B T \Delta t N_{ia}, \quad (\text{E.15})$$

giving us the desired result.

## References

- [1] W.B. Russel, Brownian motion of small particles suspended in liquids, *Annu. Rev. Fluid Mech.* 13 (1) (1981) 425–455, <https://doi.org/10.1146/annurev.fl.13.010181.002233>.
- [2] W.B. Russel, W. Russel, D.A. Saville, W.R. Schowalter, *Colloidal Dispersions*, Cambridge University Press, 1991.
- [3] M.D. Graham, *Microhydrodynamics, Brownian Motion, and Complex Fluids*, Vol. 58, Cambridge University Press, 2018.
- [4] M. Doi, S.F. Edwards, *The Theory of Polymer Dynamics*, International Series of Monographs on Physics, Oxford Science Publications, 1986.
- [5] R.G. Larson, *The Structure and Rheology of Complex Fluids*, Vol. 150, Oxford University Press, New York, 1999.
- [6] G.K. Batchelor, The effect of Brownian motion on the bulk stress in a suspension of spherical particles, *J. Fluid Mech.* 83 (1977) 97–117, <https://doi.org/10.1017/S0022112077001062>.
- [7] G. Bossis, J.F. Brady, The rheology of Brownian suspensions, *J. Chem. Phys.* 91 (3) (1989) 1866–1874.
- [8] D.R. Foss, J.F. Brady, Structure, diffusion and rheology of Brownian suspensions by Stokesian dynamics simulation, *J. Fluid Mech.* 407 (2000) 167–200, <https://doi.org/10.1017/S0022112099007557>.
- [9] P.C. Bressloff, J.M. Newby, Stochastic models of intracellular transport, *Rev. Mod. Phys.* 85 (1) (2013) 135.
- [10] D.L. Ermak, J.A. McCammon, Brownian dynamics with hydrodynamic interactions, *J. Chem. Phys.* 69 (4) (1978) 1352–1360, <https://doi.org/10.1063/1.436761>, <https://aip.scitation.org/doi/10.1063/1.436761>.
- [11] G.A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, Vol. 60, Springer, 2014.

- [12] S. Kim, S.J. Karrila, *Microhydrodynamics: Principles and Selected Applications*, Courier Dover Publications, 1991.
- [13] J. Happel, H. Brenner, *Low Reynolds Number Hydrodynamics: with Special Applications to Particulate Media*, Vol. 1, Springer Science & Business Media, 2012.
- [14] R. Kubo, The fluctuation-dissipation theorem, *Rep. Prog. Phys.* 29 (1) (1966) 255–284, <https://doi.org/10.1088/0034-4885/29/1/306>.
- [15] J.F. Brady, G. Bossi, Stokesian dynamics, *Annu. Rev. Fluid Mech.* 20 (1) (1988) 111–157, <https://doi.org/10.1146/annurev.fl.20.010188.000551>.
- [16] A. Sierou, J.F. Brady, Accelerated Stokesian dynamics simulations, *J. Fluid Mech.* 448 (2001) 115–146.
- [17] A.J. Banchio, J.F. Brady, Accelerated Stokesian dynamics: Brownian motion, *J. Chem. Phys.* 118 (22) (2003) 10323–10332, <https://doi.org/10.1063/1.1571819>.
- [18] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* 73 (2) (1987) 325–348.
- [19] A.-K. Tornberg, L. Greengard, A fast multipole method for the three-dimensional Stokes equations, *J. Comput. Phys.* 227 (3) (2008) 1613–1619.
- [20] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, *J. Comput. Phys.* 196 (2) (2004) 591–626.
- [21] D. Lindbo, A.-K. Tornberg, Spectral accuracy in fast Ewald-based methods for particle simulations, *J. Comput. Phys.* 230 (24) (2011) 8744–8761.
- [22] A.M. Fiore, F. Balboa Usabiaga, A. Donev, J.W. Swan, Rapid sampling of stochastic displacements in Brownian dynamics simulations, *J. Chem. Phys.* 146 (12) (2017) 124116.
- [23] A.M. Fiore, J.W. Swan, Rapid sampling of stochastic displacements in Brownian dynamics simulations with stresslet constraints, *J. Chem. Phys.* 148 (4) (2018) 044114.
- [24] M. Wang, J.F. Brady, Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions, *J. Comput. Phys.* 306 (2016) 443–477.
- [25] A.M. Fiore, J.W. Swan, Fast Stokesian dynamics, *J. Fluid Mech.* 878 (2019) 544–597.
- [26] C. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [27] M.R. Maxey, B.K. Patel, Localized force representations for particles sedimenting in Stokes flow, *Int. J. Multiph. Flow* (2001) 24.
- [28] S. Lomholt, M.R. Maxey, Force-coupling method for particulate two-phase flow: Stokes flow, *J. Comput. Phys.* 184 (2) (2003) 381–405, [https://doi.org/10.1016/S0021-9991\(02\)00021-9](https://doi.org/10.1016/S0021-9991(02)00021-9), <http://linkinghub.elsevier.com/retrieve/pii/S0021999102000219>.
- [29] M. Fixman, Construction of Langevin forces in the simulation of hydrodynamic interaction, *Macromolecules* 19 (4) (1986) 1204–1207, <https://doi.org/10.1021/ma00158a043>.
- [30] R.M. Jendrejack, M.D. Graham, J.J. de Pablo, Hydrodynamic interactions in long chain polymers: application of the Chebyshev polynomial approximation in stochastic simulations, *J. Chem. Phys.* 113 (7) (2000) 2894–2900.
- [31] E. Chow, Y. Saad, Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions, *SIAM J. Sci. Comput.* 36 (2) (2014) A588–A608, <https://doi.org/10.1137/130920587>, <http://epubs.siam.org/doi/10.1137/130920587>.
- [32] L. Landau, E. Lifshitz, *Fluid Mechanics*, Pergamon Press, 1959.
- [33] A.J.C. Ladd, Short-time motion of colloidal particles: numerical simulation via a fluctuating lattice-Boltzmann equation, *Phys. Rev. Lett.* 70 (1993) 1339–1342, <https://doi.org/10.1103/PhysRevLett.70.1339>.
- [34] N. Sharma, N.A. Patankar, Direct numerical simulation of the Brownian motion of particles by using fluctuating hydrodynamic equations, *J. Comput. Phys.* 201 (2) (2004) 466–486, <https://doi.org/10.1016/j.jcp.2004.06.002>.
- [35] M. De Corato, J. Slot, M. Hütter, G. D'Avino, P.L. Maffettone, M.A. Hulsen, Finite element formulation of fluctuating hydrodynamics for fluids filled with rigid particles using boundary fitted meshes, *J. Comput. Phys.* 316 (2016) 632–651.
- [36] S. Delong, F.B. Usabiaga, R. Delgado-Buscalioni, B.E. Griffith, A. Donev, Brownian dynamics without Green's functions, *J. Chem. Phys.* 140 (13) (2014) 134110.
- [37] P.J. Atzberger, P.R. Kramer, C.S. Peskin, A stochastic immersed boundary method for fluid-structure dynamics at microscopic length scales, *J. Comput. Phys.* 224 (2) (2007) 1255–1292.
- [38] P.R. Kramer, C.S. Peskin, P.J. Atzberger, On the foundations of the stochastic immersed boundary method, *Comput. Methods Appl. Mech. Eng.* 197 (25–28) (2008) 2232–2249.
- [39] E.E. Keaveny, Fluctuating force-coupling method for simulations of colloidal suspensions, *J. Comput. Phys.* 269 (2014) 61–79, <https://doi.org/10.1016/j.jcp.2014.03.013>, <http://www.sciencedirect.com/science/article/pii/S0021999114001867>.
- [40] B. Delmotte, E.E. Keaveny, Simulating Brownian suspensions with fluctuating hydrodynamics, *J. Chem. Phys.* 143 (24) (2015) 244109, <https://doi.org/10.1063/1.4938173>, <http://aip.scitation.org/doi/10.1063/1.4938173>.
- [41] Y. Bao, M. Rachh, E.E. Keaveny, L. Greengard, A. Donev, A fluctuating boundary integral method for Brownian suspensions, *J. Comput. Phys.* 374 (2018) 1094–1119.
- [42] M. Fixman, Simulation of polymer dynamics. I. General theory, *J. Chem. Phys.* 69 (4) (1978) 1527–1537, <https://doi.org/10.1063/1.436725>.
- [43] P.S. Grassia, E.J. Hinch, L.C. Nitsche, Computer simulations of Brownian motion of complex systems, *J. Fluid Mech.* 282 (1995) 373–403, <https://doi.org/10.1017/S0022112095000176>.
- [44] B. Sprinkle, F.B. Usabiaga, N.A. Patankar, A. Donev, Large scale Brownian dynamics of confined suspensions of rigid particles, *J. Chem. Phys.* 147 (24) (2017) 244103, <https://doi.org/10.1063/1.5003833>, arXiv:1709.02410.
- [45] B. Sprinkle, A. Donev, A.P.S. Bhalla, N. Patankar, Brownian dynamics of fully confined suspensions of rigid particles without Green's functions, *J. Chem. Phys.* 150 (16) (2019) 164116.
- [46] C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow*, Cambridge University Press, 1992.
- [47] F. Balboa Usabiaga, B. Delmotte, A. Donev, Brownian dynamics of confined suspensions of active microrollers, *J. Chem. Phys.* 146 (13) (2017) 134104.
- [48] F.B. Usabiaga, B. Kallembom, B. Delmotte, A.P.S. Bhalla, B.E. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, *Commun. Appl. Math. Comput. Sci.* 11 (2) (2016) 217–296, <https://doi.org/10.2140/camcos.2016.11.217>, arXiv:1602.02170.
- [49] E. Wajnryb, K.A. Mizerski, P.J. Zuk, P. Szymczak, Generalization of the Rotne-Prager-Yamakawa mobility and shear disturbance tensors, *J. Fluid Mech.* 731 (2013), arXiv:1307.7312, <https://doi.org/10.1017/jfm.2013.402>.
- [50] P.J. Zuk, E. Wajnryb, K.A. Mizerski, P. Szymczak, Rotne-Prager-Yamakawa approximation for different-sized particles in application to macromolecular bead models, *J. Fluid Mech.* 741 (2014), <https://doi.org/10.1017/jfm.2013.668>, [http://www.journals.cambridge.org/abstract\\_S002211201300668X](http://www.journals.cambridge.org/abstract_S002211201300668X).
- [51] J.W. Swan, J.F. Brady, Simulation of hydrodynamically interacting particles near a no-slip boundary, *Phys. Fluids* 19 (11) (2007) 113306, <https://doi.org/10.1063/1.2803837>, <http://aip.scitation.org/doi/10.1063/1.2803837>.
- [52] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, S. Jiang, A fast multipole method for the Rotne-Prager-Yamakawa tensor and its applications, *J. Comput. Phys.* 234 (2013) 133–139.
- [53] E.E. Keaveny, Fluctuating force-coupling method for simulations of colloidal suspensions, *J. Comput. Phys.* 269 (2014) 61–79, <https://doi.org/10.1016/j.jcp.2014.03.013>, <http://linkinghub.elsevier.com/retrieve/pii/S0021999114001867>.
- [54] M.R. Maxey, B.K. Patel, Localized force representations for particles sedimenting in Stokes flow, *Int. J. Multiph. Flow* 27 (2001) 1603–1626.
- [55] B. Delmotte, E. Climent, F. Plouraboué, A general formulation of Bead Models applied to flexible fibers and active filaments at low Reynolds number, *J. Comput. Phys.* 286 (2015) 14–37, <https://doi.org/10.1016/j.jcp.2015.01.026>.
- [56] P.J. Atzberger, P.R. Kramer, C.S. Peskin, A stochastic immersed boundary method for fluid-structure dynamics at microscopic length scales, *J. Comput. Phys.* 224 (2) (2007) 1255–1292, <https://doi.org/10.1016/j.jcp.2006.11.015>.

- [57] P.J. Atzberger, Stochastic Eulerian Lagrangian methods for fluid-structure interactions with thermal fluctuations, *J. Comput. Phys.* 230 (8) (2011) 2821–2837, <https://doi.org/10.1016/j.jcp.2010.12.028>.
- [58] P. Bourrienne, V. Niggel, G. Polly, T. Divoux, G.H. McKinley, Unifying disparate experimental views on shear-thickening suspensions, arXiv:2001.02290, 2020.
- [59] M. Driscoll, B. Delmotte, M. Youssef, S. Sacanna, A. Donev, P. Chaikin, Unstable fronts and stable “critters” formed by microrollers, *Nat. Phys.* 13 (4) (2017) 375–379, <https://doi.org/10.1038/nphys3970>, arXiv:1609.08673.
- [60] S. Delong, F. Balboa Usabiaga, A. Donev, Brownian dynamics of confined rigid bodies, *J. Chem. Phys.* 143 (14) (2015) 144107, <https://doi.org/10.1063/1.4932062>, <http://aip.scitation.org/doi/10.1063/1.4932062>.
- [61] A. Chakrabarty, A. Konya, F. Wang, J.V. Selinger, K. Sun, Q.-H. Wei, Brownian motion of boomerang colloidal particles, *Phys. Rev. Lett.* 111 (16) (2013) 160603.
- [62] M. Eshraghi, J. Horbach, Molecular dynamics simulation of charged colloids confined between hard walls: pre-melting and pre-freezing across the BCC–fluid coexistence, *Soft Matter* 14 (20) (2018) 4141–4149, <https://doi.org/10.1039/C8SM00398J>, <https://pubs.rsc.org/en/content/articlelanding/2018/sm/c8sm00398j>.
- [63] I.K. Snook, D. Henderson, Monte Carlo study of a hard-sphere fluid near a hard wall, *J. Chem. Phys.* 68 (5) (1978) 2134–2139, <https://doi.org/10.1063/1.436036>, <https://aip.scitation.org/doi/10.1063/1.436036>.
- [64] J. Mittal, J.R. Errington, T.M. Truskett, Does confining the hard-sphere fluid between hard walls change its average properties?, *J. Chem. Phys.* 126 (24) (2007) 244708, <https://doi.org/10.1063/1.2748045>, <https://aip-scitation-org.iclibezp1.cc.ic.ac.uk/doi/10.1063/1.2748045>.
- [65] D. Deb, A. Winkler, M.H. Yamani, M. Oettel, P. Virnau, K. Binder, Hard sphere fluids at a soft repulsive wall: a comparative study using Monte Carlo and density functional methods, *J. Chem. Phys.* 134 (21) (2011) 214706, <https://doi.org/10.1063/1.3593197>, <https://aip-scitation-org.iclibezp1.cc.ic.ac.uk/doi/10.1063/1.3593197>.
- [66] D.H. Van Winkle, C.A. Murray, Layering in colloidal fluids near a smooth repulsive wall, *J. Chem. Phys.* 89 (6) (1988) 3885–3891, <https://doi.org/10.1063/1.454864>, <https://aip-scitation-org.iclibezp1.cc.ic.ac.uk/doi/10.1063/1.454864>.
- [67] P. Bourrienne, V. Niggel, G. Polly, T. Divoux, G.H. McKinley, Unifying disparate experimental views on shear-thickening suspensions, arXiv:2001.02290.
- [68] A. Vázquez-Quesada, F. Balboa Usabiaga, R. Delgado-Buscalioni, A multiblob approach to colloidal hydrodynamics with inherent lubrication, *J. Chem. Phys.* 141 (20) (2014) 204102, <https://doi.org/10.1063/1.4901889>, <https://aip.scitation.org/doi/10.1063/1.4901889>.
- [69] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, A. Zanna, Lie-group methods, *Acta Numer.* 9 (2000) 215–365, <https://www.cambridge.org/core/journals/acta-numerica/article/liegroup-methods/856125FF1EAF7762DEF6E37EEBA9CA5F>.
- [70] S. Faltinsen, A. Marthinsen, H.Z. Munthe-Kaas, Multistep methods integrating ordinary differential equations on manifolds, *Appl. Numer. Math.* 39 (3) (2001) 349–365, [https://doi.org/10.1016/S0168-9274\(01\)00103-9](https://doi.org/10.1016/S0168-9274(01)00103-9), <http://www.sciencedirect.com/science/article/pii/S0168927401001039>.
- [71] S.J.A. Malham, A. Wiese, Stochastic Lie group integrators, *SIAM J. Sci. Comput.* 30 (2) (2008) 597–617, <https://doi.org/10.1137/060666743>.