Large Scale Brownian Dynamics of Confined Suspensions of Rigid Particles

Brennan Sprinkle,[1] Florencio Balboa Usabiaga,[2, 3] Neelesh A. Patankar,[1] and Aleksandar Donev[2, *]

[1]*McCormick School of Engineering, Northwestern University, Evanston, IL 60208*

[2]*Courant Institute of Mathematical Sciences,*
New York University, New York, NY 10012

[3]*Center for Computational Biology, Flatiron Institute,*
Simons Foundation, New York 10010, USA

We introduce methods for large scale Brownian Dynamics (BD) simulation of many rigid particles of arbitrary shape suspended in a fluctuating fluid. Our method adds Brownian motion to the rigid multiblob method [F. Balboa Usabiaga *et al.*, Communications in Applied Mathematics and Computational Science, 11(2):217-296, 2016] at a cost comparable to the cost of deterministic simulations. We demonstrate that we can efficiently generate deterministic and random displacements for many particles using preconditioned Krylov iterative methods, if kernel methods to efficiently compute the action of the Rotne-Prager-Yamakawa (RPY) mobility matrix and it "square" root are available for the given boundary conditions. These kernel operations can be computed with near linear scaling for periodic domains using the Positively Split Ewald method. Here we study particles partially confined by gravity above a no-slip bottom wall using a graphical processing unit (GPU) implementation of the mobility matrix vector product, combined with a preconditioned Lanczos iteration for generating Brownian displacements. We address a major challenge in large-scale BD simulations, capturing the stochastic drift term that arises because of the configuration-dependent mobility. Unlike the widely-used Fixman midpoint scheme, our methods utilize random finite differences and do not require the solution of resistance problems or the computation of the action of the inverse square root of the RPY mobility matrix. We construct two temporal schemes which are viable for large scale simulations, an Euler-Maruyama traction scheme and a Trapezoidal Slip scheme, which minimize the number of mobility solves per time step while capturing the required stochastic drift terms. We validate and compare these schemes numerically by modeling suspensions of boomerang shaped particles sedimented near a bottom wall. Using the trapezoidal scheme, we investigate the steady-state active motion in a dense suspensions of confined microrollers, whose height above the wall is set by a combination of thermal noise and active flows. We find the existence of two populations of active particles, slower ones closer to the bottom and faster ones above them, and demonstrate that our method provides quantitative accuracy even with relatively coarse resolutions of the particle geometry.

I. INTRODUCTION

The study of Brownian suspensions of passive and active particles has become ubiquitous, particularly in the biological and physical sciences [1–3]. Thermal fluctuations play an integral role in determining the dynamics of active suspensions, for example, they affect the development of a recently-discovered fingering instability [4] in suspensions of microrollers partially confined by gravity above a no-slip wall bottom wall [5]. Efficient simulations of these systems, which correctly capture the effects of Brownian motion, are essential for designing and understanding experimental results as well as discovering new collective phenomena. The purpose of this work is to design scalable and accurate numerical methods for Brownian Dynamics (BD) simulation of suspensions of many passive and/or active rigid particles of arbitrary shape. By scalable, we mean that the computational complexity of the method should scale (nearly) *linearly* with the number of particles considered. By accurate, here we mean that the temporal integrators used should be second-order accurate in the deterministic setting (i.e., without Brownian motion), and maximize the weak accuracy for a given computational cost per unit simulation time. To our knowledge, this is the first time either of these goals have been accomplished with *controlled* accuracy for a suspension of particles of arbitrary shape. Existing methods based on uncontrolled multipole truncations [6–8] are focused on spheres and spheroids and are difficult to generalize to general particle shapes, though some special shapes like thin rods have been tackled using slender body theory [9].

In [10] some of us proposed a flexible and scalable *rigid multiblob* method for simulation of many rigid bodies (not necessarily spherical) in Stokes flow, in the absence of Brownian motion. In the rigid multiblob method [10, 11], complex particle shapes are constructed with tunable resolution (accuracy) as a rigidly-connected cluster of spherical "blobs". In [10] some of us developed efficient preconditioned Krylov methods for solving mobility problems in computational cost that scales (nearly) linearly with the number of blobs used to construct the rigid particles. Here we focus on efficient computation of stochastic displacements consistent with the overdamped Langevin equation for non-spherical particles proposed by some of us in [11]. In the prior work [11] we assumed a small number of particles and performed direct (Cholesky) factorization of mobility matrices to compute Brownian increments, which scales as the cube of the number of blobs and is infeasible for many-body suspensions. In this work we develop linear-scaling preconditioned iterative methods for computing the stochastic increments of particle positions and orientations.

---

*Electronic address: donev@courant.nyu.edu

A second nontrivial challenge we address is the construction of consistent and accurate temporal integrators. The widely-used Fixman midpoint temporal integrator, generalized to include particle orientations in [11], requires solving resistance problems, which cannot be done in linear time with present methods [10]. Here we construct two temporal integrators that correctly capture stochastic drift terms proportional to the divergence of the mobility matrix, and require only solving mobility problems. While here we only test these novel schemes with the rigid multiblob method [10], it is important to note that the same temporal integrators apply to highly-accurate boundary integral formulations [43] for Stokesian suspensions [12–14]. Furthermore, while we focus here on suspensions confined above a no-slip wall, the methods we present here are rather general and can be applied to other systems such as bulk passive or active suspensions.

In section II B, we develop a scalable method to generate the Brownian increments for the particles from the Brownian increments of the individual blobs, which can themselves be computed using a preconditioned Lanczos method [15], as previously described for particles above a no-slip wall in [5], and for periodic suspensions in [16]. In section III A we propose a novel modification of the Euler-Maruyama (EM) scheme, which involves solving only a *single* additional mobility problem in order to capture the Ito stochastic (thermal) drift required to maintain the Gibbs-Boltzmann distribution at equilibrium. This is a notable improvement over the EM method proposed in [11] which requires *two* additional mobility solves to compute the drift using a *random finite difference* (RFD). In section III B we propose a novel trapezoidal scheme which also captures the correct thermal drift by solving only a single additional mobility problem, and is second order accurate in time for deterministic calculations. While the scheme is formally only first-order weakly accurate in the stochastic setting, the improved deterministic accuracy translates to substantially improved weak accuracy, as we demonstrate numerically.

In sections IV A 1 and IV B, we validate the new temporal integrators and compare their efficiency/accuracy tradeoffs by examining equilibrium statistics for suspensions of passive colloidal boomerangs confined above a wall. In section IV C we revisit some experimental and computational investigations done in [5] for dense uniform suspensions of rotating colloids (microrollers) above a planar wall. In this prior work [5], a large mismatch was observed between experimental measurements of the steady-state mean suspension velocity and estimates based on the minimally-resolved Brownian dynamics computations [5]. Here we are able to simulate a dense uniform suspension of microrollers with much higher resolution. The higher resolution allows us to better resolve the hydrodynamic interactions between the particles and make quantitative predictions that are sufficiently accurate to be directly compared to experiments.

II. BROWNIAN DYNAMICS FOR RIGID BODIES

We consider a suspension of N_b passive or active rigid bodies (particles) suspended in a fluctuating Stokesian fluid. For body $p \in [1, \ldots, N_b]$, we will follow a reference *tracking point* with Cartesian position, $\boldsymbol{q}_p(t)$. The orientation of body p relative to the tracking point will be denoted by $\boldsymbol{\theta}_p(t)$. For simplicity and increased generality, we make the bulk of the discussion in this work agnostic to the choice of coordinates for $\boldsymbol{\theta}_p$ and assume that the representation is a scalar in two dimensions or a three-dimensional vector in three dimensions. In practice, however, we use unit quaternions in three dimensions, as discussed in detail in [11]. The unit norm constraint of the quaternion can be handled simply by updating orientation using quaternion multiplication (rotations) instead of addition, as detailed in Appendix B 1. We denote the generalized position of body p as $\boldsymbol{Q}_p(t) = \left[\boldsymbol{q}_p(t), \boldsymbol{\theta}_p(t)\right]$ and denote the many-body configuration with $\boldsymbol{Q} = \left[\boldsymbol{Q}_p\right]$. To each body p, we prescribe an applied force $\boldsymbol{f}_p$, and an applied torque $\boldsymbol{\tau}_p$, and denote the generalized force on body p with $\boldsymbol{F}_p = \left[\boldsymbol{f}_p, \boldsymbol{\tau}_p\right]$ and write $\boldsymbol{F} = [\boldsymbol{F}_p]$. The prescription of external (non-conservative) forces and torques is one way in which we may model active bodies, the other, active slip, is discussed in more detail in [10] and summarized in section II A.

Given forces and torques, our aim is to find the rigid body velocities $\boldsymbol{U} = [\boldsymbol{U}_p]$, where the generalized velocity $\boldsymbol{U}_p = [\boldsymbol{u}_p, \boldsymbol{\omega}_p]$ is composed of a translational velocity $\boldsymbol{u}_p$ and a rotational (angular) velocity $\boldsymbol{\omega}_p$. A central object in the overdamped Langevin equations for the suspension is the configuration-dependent *body mobility matrix* $\boldsymbol{\mathcal{N}}(\boldsymbol{Q})$. In a deterministic setting, the symmetric positive-definite (SPD) matrix $\boldsymbol{\mathcal{N}}$, relates the generalized velocities with the generalized forces, $\boldsymbol{U} = \boldsymbol{\mathcal{N}}\boldsymbol{F}$. The application of the body mobility matrix, i.e., the computation of $\boldsymbol{U} = \boldsymbol{\mathcal{N}}\boldsymbol{F}$ given $\boldsymbol{F}$, is referred to as the *mobility problem.* Its inverse problem, the *resistance problem*, involves finding the forces and torques given prescribed rigid-body motions, i.e., computing $\boldsymbol{F} = \boldsymbol{\mathcal{N}}^{-1}\boldsymbol{U}$. By combining the rigid multiblob method with preconditioned iterative solvers, one can solve a mobility problem efficiently in linear time, however, the solution of resistance problems is much more expensive and does not scale linearly [10].

For a suspension of rigid bodies, the configuration evolves according to the overdamped Langevin Ito BD equation,

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

where $\boldsymbol{\mathcal{W}}$ is a collection of independent white noise processes [11]. Here the "square root" of the mobility $\boldsymbol{\mathcal{N}}^{1/2}$ is any matrix, *not* necessarily square, that satisfies the fluctuation-dissipation

relation $\boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{N}}^{1/2} \left(\boldsymbol{\mathcal{N}}^{1/2} \right)^T$. We will refer to the term $k_B T \, \partial_{\boldsymbol{Q}} \cdot \boldsymbol{\mathcal{N}}$ as the *stochastic* or *thermal drift* (or sometimes simply the *drift*) since it has its origin in the stochastic interpretation of the noise; this drift term would disappear if the so-called kinetic or Klimontovich interpretation of the noise is used [17]. Efficient generation of this term will be the most challenging part of this work and is discussed in detail in section III. As a prelude, in subsection II A we briefly review the methods proposed in [10] to efficiently compute the deterministic displacements, $\boldsymbol{\mathcal{N}} \boldsymbol{F}$. Then, in subsection II B, we propose a scalable iterative method for computing the Brownian displacements over a time interval Δt, $\sqrt{2 k_B T \, \Delta t} \, \boldsymbol{\mathcal{N}}^{1/2} \boldsymbol{W}$, where $\boldsymbol{W}$ is a vector of independent standard Gaussian random variables.

A. Solving Mobility Problems

We discretize the rigid bodies using a rigid multi-blob model, wherein rigid bodies are treated as rigid conglomerations of beads, or "blobs", of hydrodynamic radius a. The blobs comprising a given rigid body $\mathcal{B}_p$ have positions $\boldsymbol{r}^{(p)} = [\boldsymbol{r}_i \mid i \in \mathcal{B}_p]$. Given a rigid body velocity $\boldsymbol{U}_p = [\boldsymbol{u}_p, \boldsymbol{\omega}_p]$, the *geometric* block matrix $\boldsymbol{\mathcal{K}}$ that converts rigid body motion into blob motion is defined as [18]

$$(\boldsymbol{\mathcal{K}} \boldsymbol{U})_i = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times \left(\boldsymbol{r}_i - \boldsymbol{q}_p \right), \quad p \in 1, \dots, N_B \text{ and } i \in \mathcal{B}_p. \tag{2}$$

Using $\boldsymbol{\mathcal{K}}$, a slip condition on the rigid bodies can be compactly expressed as

$$\dot{\boldsymbol{r}} = \boldsymbol{\mathcal{K}} \boldsymbol{U} - \breve{\boldsymbol{u}}, \tag{3}$$

where $\breve{\boldsymbol{u}}$ is a prescribed slip velocity of the fluid at the locations of the blobs. Physically, $\breve{\boldsymbol{u}}$ could account for an active boundary layer [8, 19]. However, in this work, we will find a great deal of utility in prescribing $\breve{\boldsymbol{u}}$ in such a way as to help generate the stochastic terms in equation (1). The force and torque balance conditions on the particles can be expressed using the adjoint of $\boldsymbol{\mathcal{K}}$, $\boldsymbol{\mathcal{K}}^T \boldsymbol{\lambda} = \boldsymbol{F}$ [18].

The hydrodynamic interactions between blob i and j are captured by the 3×3 mobility matrix $\boldsymbol{M}_{ij}$, which gives the velocity $\dot{\boldsymbol{r}}_i$ of blob i given a force $\boldsymbol{\lambda}_j$ on blob j, $\dot{\boldsymbol{r}}_i = \boldsymbol{M}_{ij} \boldsymbol{\lambda}_j$. The symmetric, positive semi-definite matrix $\boldsymbol{\mathcal{M}}$ composed of the blocks $\boldsymbol{M}_{ij}$ is termed the *blob-blob mobility matrix*. The construction of $\boldsymbol{\mathcal{M}}$ for a rigid multiblob must account for the finite hydrodynamic radius of the blobs, a, as well as the geometry of the domain. In the case of a three dimensional unbounded domain, the well-known Rotne-Prager-Yamakawa (RPY) tensor [20, 21] can be used to construct $\boldsymbol{M}_{ij}$, and the action of $\boldsymbol{\mathcal{M}}$ on a vector can be computed in linear time using a fast

multipole method [22]. For periodic domains, we can use the Positively Split Ewald (PSE) method [16] to compute the action of the RPY-based mobility $\mathcal{M}$ on a vector. A generalization of the RPY kernel to particles confined above a single no-slip wall, the Rotne-Prager-Blake tensor, is given in [23] and we will use it in section IV. For general fully-confined domains, an on-the-fly procedure to calculate $\mathcal{M}$ has been proposed in [24, 25]. Note that the action of $\mathcal{M}$ can be interpreted as a physically-regularized single-layer (first-kind) boundary integral operator (see appendix A of [10]).

Given $\mathcal{M}$, we can write the mobility problem as a linear system

$$\mathcal{M}\boldsymbol{\lambda} = \mathcal{K}\boldsymbol{U} - \breve{\boldsymbol{u}} \tag{4}$$

$$\mathcal{K}^T\boldsymbol{\lambda} = \boldsymbol{F}, \tag{5}$$

which can be written as the *saddle-point* linear system,

$$\begin{bmatrix} \mathcal{M} & -\mathcal{K} \\ -\mathcal{K}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} -\breve{\boldsymbol{u}} \\ -\boldsymbol{F} \end{bmatrix}. \tag{6}$$

Using Schur complements, we can compactly write the solution to (6) as

$$\boldsymbol{U} = \mathcal{N}\boldsymbol{F} + \mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\breve{\boldsymbol{u}}, \tag{7}$$

where we have identified the *body mobility matrix*

$$\mathcal{N} = \left(\mathcal{K}^T\mathcal{M}^{-1}\mathcal{K}\right)^{-1}. \tag{8}$$

We note that exactly the same saddle-point system, with a mobility matrix $\mathcal{M}$ computed using singular quadratures instead of the RPY kernel, appears in a recently-developed first-kind Fluctuating Boundary Integral Method (FBIM) for suspensions [26].

In the case of many bodies, computing (the action of) $\mathcal{N}$ directly from equation (8) is very inefficient if at all feasible. In practice, we will solve mobility problems by solving (6) directly. Efficient, preconditioned Krylov solvers to solve this system were developed in [10]. The efficiency of these solvers is dependent, primarily, on the speed at which the matrix, $\mathcal{M}$, can be applied to a vector. If a linear-scaling method such as a fast-multipole-method (FMM) [22, 27] or the PSE method [16] are used, these methods will scale near linearly (to within logarithmic factors) with the total number of blobs. Following [10], here we will use direct dense matrix-vector products implemented on a GPU to apply the Rotne-Prager-Blake mobility. While this in principle scales quadratically with the total number of blobs, modern GPUs are typically powerful enough for a direct implementation of a matrix-vector product to outperform more sophisticated techniques up

to a fairly large number (hundreds of thousands) of blobs [5]. No matter how fast the matrix-vector products with $\boldsymbol{\mathcal{N}}$ (or equivalently $\boldsymbol{\mathcal{M}}$) can be computed, solving the system (6) is one of two bottlenecks in designing efficient integrators to solve (1). We discuss the other bottleneck next.

B. Computing Brownian increments

As mentioned in section II A, direct computation of $\boldsymbol{\mathcal{N}} = \left(\boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}}\right)^{-1}$ is computationally infeasible for many bodies due to the dense matrix inversions required. Direct computation of $\boldsymbol{\mathcal{N}}^{1/2}$, therefore, is still less practical in these situations. Our key insight to overcome this is that $\boldsymbol{\mathcal{N}}^{1/2}$ is not unique and doesn't need to be a square matrix, it only needs to satisfy $\boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{N}}^{1/2} \left(\boldsymbol{\mathcal{N}}^{1/2}\right)^T$. This gives great freedom in choosing $\boldsymbol{\mathcal{N}}^{1/2}$ so that its action can be computed in linear time. We will assume here that we were able to efficiently compute Brownian displacements for the individual blobs, i.e., to compute $\boldsymbol{\mathcal{M}}^{1/2} \boldsymbol{W}$, where $\boldsymbol{W}$ is a vector of independent standard Gaussian random variables. This can be done using preconditioned iterative methods for bodies near a no-slip wall [5], using the PSE method [16] for periodic suspensions, or using the FBIM [26] for fully confined or periodic suspensions.

Let us impose the random slip velocity $\breve{\boldsymbol{u}} = \sqrt{2k_B T/\Delta t}\, \boldsymbol{\mathcal{M}}^{1/2} \boldsymbol{W}$ in (6) [44], to get the saddle-point linear system

$$\begin{bmatrix} \boldsymbol{\mathcal{M}} & -\boldsymbol{\mathcal{K}} \\ -\boldsymbol{\mathcal{K}}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} -\sqrt{2k_B T/\Delta t}\, \boldsymbol{\mathcal{M}}^{1/2} \boldsymbol{W} \\ \mathbf{0} \end{bmatrix}. \tag{9}$$

The solution of this system can be written using equation (7) as

$$\boldsymbol{U} = \sqrt{2k_B T/\Delta t}\, \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{M}}^{1/2} \boldsymbol{W} = \sqrt{2k_B T/\Delta t}\, \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1/2} \boldsymbol{W}. \tag{10}$$

It is not hard to see that we can identify the matrix

$$\boldsymbol{\mathcal{N}}^{1/2} \equiv \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1/2} \tag{11}$$

as a "square root" of the mobility, since

$$\boldsymbol{\mathcal{N}}^{1/2} \left(\boldsymbol{\mathcal{N}}^{1/2}\right)^T = \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1/2} \left(\boldsymbol{\mathcal{M}}^{-1/2}\right)^T \boldsymbol{\mathcal{K}} \boldsymbol{\mathcal{N}} \tag{12}$$

$$= \boldsymbol{\mathcal{N}} \left(\boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}}\right) \boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{N}}^{-1} \boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{N}}. \tag{13}$$

Thus, equation (10) becomes

$$\boldsymbol{U} = \sqrt{\frac{2k_B T}{\Delta t}}\, \boldsymbol{\mathcal{N}}^{1/2} \boldsymbol{W}. \tag{14}$$

Hence the Brownian "velocities" (more precisely, the Brownian displacements $\boldsymbol{U}\Delta t$) for the rigid bodies can be computed by solving a mobility problem with random slip given by Brownian velocities for the blobs. Observe that we need only a *single* application of $\boldsymbol{\mathcal{M}}^{1/2}$ to a vector, and the solution of a *single* mobility problem, to compute *both* the deterministic and the Brownian increments (but not including the stochastic drift terms yet). Note that the same construction of $\boldsymbol{\mathcal{N}}^{1/2}$ is used in the recently-developed FBIM [26], with the slip velocity $\breve{\boldsymbol{u}}$ interpreted as a random surface velocity distribution with covariance equal to the Green's function for periodic Stokes flow (i.e., the periodic Stokeslet).

Given an efficient routine to compute the product $\boldsymbol{\mathcal{M}}\boldsymbol{\lambda}$ for a given $\boldsymbol{\lambda}$, as discussed in section II A, a preconditioned Lanczos-type iterative method to compute the product $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$ was proposed in [15]. In unbounded or periodic domains the number of iterations increases with the size of $\boldsymbol{\mathcal{M}}$, and the cost of computing $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$ is many times than that of computing $\boldsymbol{\mathcal{M}}\boldsymbol{\lambda}$. In the PSE method an additional splitting of $\boldsymbol{\mathcal{M}}$ into a near-field and far-field components is introduced, and the Lanczos method is only applied to the near field, while the far-field component is handled using fluctuating hydrodynamics. For particles confined close to a no-slip wall, the friction with the floor screens the hydrodynamic interactions to decay like inverse distance cubed. This makes the Lanczos iteration converge in a small number of iterations independent of the number of blobs [5]. However, for rigid multiblobs the number of iterations is higher than for single blobs because of the increased ill-conditioning of $\boldsymbol{\mathcal{M}}$ due to the presence of (nearly-)touching blobs.

In this work, we employ a block diagonal preconditioner $\widetilde{\boldsymbol{\mathcal{M}}} \approx \boldsymbol{\mathcal{M}}$ for the Lanczos algorithm [15] that substantially reduces the number of iterations in the computation of $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$ for rigid multiblobs. Similar block-diagonal or diagonal preconditioners have been used for Stokesian suspensions by other authors [14, 28–31]. In the preconditioner, which is also used to solve the saddle-point system (6) [10], we ignore hydrodynamic interactions between *distinct* bodies p and q, $\widetilde{\boldsymbol{\mathcal{M}}}^{(pq)} = \delta_{pq}\boldsymbol{\mathcal{M}}^{(pp)}$. For each body p we explicitly form a dense blob-blob mobility matrix $\boldsymbol{\mathcal{M}}^{(pp)}$ (equal to the diagonal block of $\boldsymbol{\mathcal{M}}$ corresponding to body p), ignoring the presence of other bodies. The preconditioner for the Lanczos method is a block diagonal matrix $\boldsymbol{L} = \widetilde{\boldsymbol{\mathcal{M}}}^{\frac{1}{2}}$ composed of the Cholesky factors of $\boldsymbol{\mathcal{M}}^{(pp)}$. We pre-compute $\boldsymbol{L}$ once per time step (or less frequently if desired) and then reuse it in the iterative solves in that time step.

In Fig. 1 we probe the convergence of our preconditioned solvers in a suspension of boomerang colloidal particles sedimented over a rigid wall for surface area fraction $\phi \approx 0.25$ (see details in Sec. IV). The figure shows the number of iterations required to reach a desired tolerance for both the solution of (6), as well as for computing the product $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$, for three different problem sizes.
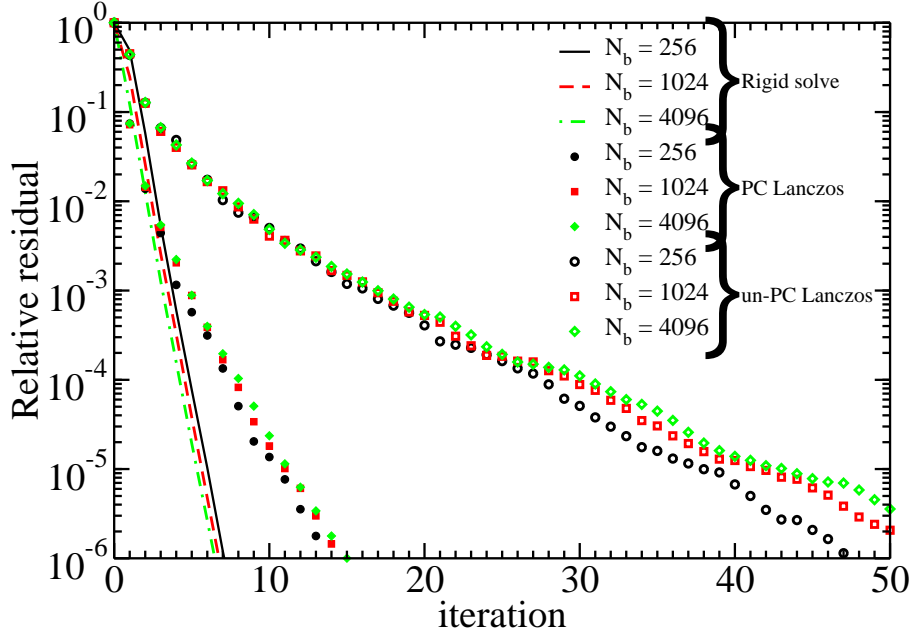
Figure 1: Convergence of iterative solvers for the problem described in section IV B, a suspension of 256, 1024 or 4096 colloidal boomerangs (each containing 15 blobs) sedimented near a bottom wall. Convergence of the preconditioned GMRES iteration to solve equation (6), labeled as 'Rigid solve', is demarcated by solid lines. Convergence of the preconditioned Lanczos method to compute $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$, labeled as 'PC Lanczos', is demarcated by filled symbols, while the corresponding results without preconditioning, labeled as 'un-PC Lanczos', are demarcated by un-filled symbols.

Also shown is the effect of preconditioning on the convergence of the matrix root computation. We can see that up to a relative tolerance of around 10^{-3}, both the saddle point solve, (6) and $\boldsymbol{\mathcal{M}}^{1/2}\boldsymbol{W}$ require roughly the same number of iterations to converge, with the latter taking more iterations when smaller tolerances are required. When preconditioning is used, both iterative methods are shown to have convergence rates independent of problem size. Nevertheless, computing matrix roots represents another major bottleneck in integrating (1) for rigid multiblobs confined above a no-slip floor, with cost similar to that of solving the saddle point system (6). Observe that both the computation of the deterministic and the fluctuating velocities involves repeated applications of $\boldsymbol{\mathcal{M}}$, which dominates the cost. Therefore, we seek to integrate (1) to a desired accuracy in as few total number of applications of $\boldsymbol{\mathcal{M}}$ as possible.

III. TEMPORAL INTEGRATORS AND THE THERMAL DRIFT

Our goal is to numerically integrate the overdamped Langevin equation (1) as efficiently as possible. In section II we discussed efficient means of computing $\boldsymbol{\mathcal{N}}\boldsymbol{F} + \sqrt{2k_B T/\Delta t}\,\boldsymbol{\mathcal{N}}^{1/2}\boldsymbol{W}$, all that remains is to find a way to efficiently generate the thermal drift term $k_B T\,\partial_{\boldsymbol{Q}}\cdot\boldsymbol{\mathcal{N}}$. Capturing this drift term is a common challenge in all methods for Brownian dynamics, and the methods developed here are general and apply to any approach based on solving mobility problems.

A widely-used method to capture $k_B T\,\partial_{\boldsymbol{Q}}\cdot\boldsymbol{\mathcal{N}}$ is due to Fixman [32], and can be seen as a midpoint method to capture the Stratonovich product in a mixed Stratonovich-Ito (also known as Klimontovich or kinetic interpretation [17]) re-formulation of (1) [11]. The generalization of Fixman's method to account for particle orientations is given in Section III of [11]. The problem with the Fixman scheme in the context of many-body suspensions is that it requires the computation of $\boldsymbol{\mathcal{N}}^{-1/2}\boldsymbol{W}$. This is related to solving resistance problems and is infeasible for many body simulations. In particular, there is no known method to compute $\boldsymbol{\mathcal{M}}^{-1}$ which scales linearly with the problem size. Hence, the Fixman's scheme has to be ruled out for use in many body simulations. Here we will only use Fixman's method as a reference method for small problems involving at most on the order of a hundred blobs, where dense linear algebra is practicable [11].

In [11, 33], some of us proposed a means of capturing the drift term in (1) using a modification of Fixman's approach. This idea, termed random finite difference (RFD), is as follows. Given two Gaussian random vectors, $\Delta\boldsymbol{P}$ and $\Delta\boldsymbol{Q}$, such that $\left\langle\Delta\boldsymbol{P}\Delta\boldsymbol{Q}^T\right\rangle = (k_B T)\,\boldsymbol{I}$, the following relation holds for a configuration dependent matrix $\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}\right)$,

$$\lim_{\delta\to 0}\frac{1}{\delta}\left\langle\left\{\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}+\delta\Delta\boldsymbol{Q}\right)-\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}\right)\right\}\Delta\boldsymbol{P}\right\rangle = \tag{15}$$

$$\lim_{\delta\to 0}\frac{1}{\delta}\left\langle\left\{\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}+\frac{\delta}{2}\Delta\boldsymbol{Q}\right)-\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}-\frac{\delta}{2}\Delta\boldsymbol{Q}\right)\right\}\Delta\boldsymbol{P}\right\rangle = \tag{16}$$

$$\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}\right)\right\}:\left\langle\Delta\boldsymbol{P}\Delta\boldsymbol{Q}^T\right\rangle = k_B T\,\partial_{\boldsymbol{Q}}\cdot\boldsymbol{\mathcal{B}}\left(\boldsymbol{Q}\right), \tag{17}$$

where $\langle\rangle$ denotes an ensemble average. In practice, we will implement random finite differences by simply taking δ to be a *small* number. Thus we recognize, by analogy with standard finite differences, equations (15) and (16) as one-sided and centered approximations to (17) with truncation errors of $O\left(\delta\right)$ and $O\left(\delta^2\right)$ respectively. Note that Fixman's scheme can be viewed as an RFD where $\delta = \sqrt{\Delta t}$, $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{N}}$, $\Delta\boldsymbol{Q} = \sqrt{k_B T}\boldsymbol{\mathcal{N}}^{1/2}\boldsymbol{W}$, and $\Delta\boldsymbol{P} = \sqrt{k_B T}\boldsymbol{\mathcal{N}}^{-1/2}\boldsymbol{W}$ [33]; see Section III.D in [6] for the first use of a δ independent of Δt in order to "avoid particle 'overlaps' in the intermediate configuration." A simpler choice, used in [5, 11], is to take $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{N}}$, and use $\Delta\boldsymbol{P} = \Delta\boldsymbol{Q} = \sqrt{k_B T}\,\boldsymbol{W}$. Other more efficient choices have been constructed in a number of specific

contexts [7, 24**?**]. In order to best pick δ, we must balance the truncation error with other sources of error introduced from the inexact multiplication of $\boldsymbol{\mathcal{B}}$. At best, multiplication by $\boldsymbol{\mathcal{B}}$ is calculated to machine precision and δ may be taken to be quite small. At worst, multiplication of $\boldsymbol{\mathcal{B}}$ is only computed approximately to within some relative tolerance ϵ, as would be the case when we take $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{N}}$ and matrix vector multiplications are computed using the iterative method described in section (II A). In this case, using one-sided differencing can lead to large truncation errors when loose solver tolerances are used, and we recommend that only central differencing be used.

In [11], an Euler-Maruyama (EM) RFD (EM-RFD) scheme is presented to solve (1), using a one-sided RFD on $\boldsymbol{\mathcal{N}}$. A scalable variant of this using a central RFD is a trivial extension summarized in appendix A, where we clarify how to do this using iterative solvers and also with care for different units for length and orientation. This scheme requires *three* solutions of the saddle point system (6) and one application of $\boldsymbol{\mathcal{M}}^{1/2}$ per timestep, and is only first-order accurate even deterministically. By using different choices for $\Delta\boldsymbol{P}$ and $\Delta\boldsymbol{Q}$ in (16), we will reduce the cost of capturing the stochastic drift term considerably. In section III A we will present an EM Traction (EM-T) scheme which only requires *two* solutions of the saddle point system. The trapezoidal slip (T-S) scheme presented in section III B still requires three solutions of the saddle point system but achieves higher accuracy, notably, it is second-order accurate deterministically just like the Fixman midpoint scheme given in [11]. We will empirically compare these two schemes in terms of accuracy per computational effort in Section IV B.

A.   Euler-Maruyama Traction (EM-T) Scheme

To improve the efficiency of the scheme given in appendix A, we propose a different means of computing the drift term. Using the chain rule, we can split the divergence of the body mobility matrix into three pieces,

$$
\begin{aligned}
\partial_{\boldsymbol{Q}} \cdot \boldsymbol{\mathcal{N}} &= -\boldsymbol{\mathcal{N}} \left(\partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{N}}^{-1} \right) : \boldsymbol{\mathcal{N}} = -\boldsymbol{\mathcal{N}} \left(\partial_{\boldsymbol{Q}} \left\{ \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}} \right\} \right) : \boldsymbol{\mathcal{N}} = \\
&\quad -\boldsymbol{\mathcal{N}} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}}^T \right\} : \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}} \boldsymbol{\mathcal{N}} - \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{M}}^{-1} \right\} : \boldsymbol{\mathcal{K}} \boldsymbol{\mathcal{N}} - \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}} \right\} : \boldsymbol{\mathcal{N}} = \\
&\quad -\boldsymbol{\mathcal{N}} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}}^T \right\} : \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}} \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{M}} \right\} : \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{K}} \boldsymbol{\mathcal{N}} - \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}} \right\} : \boldsymbol{\mathcal{N}}.
\end{aligned}
\tag{18}
$$

where colon denotes contraction; this calculation is done more precisely using index notation in appendix B 1. Unlike $\boldsymbol{\mathcal{N}}$, we can efficiently compute the action of $\boldsymbol{\mathcal{K}}^T$, $\boldsymbol{\mathcal{M}}$, and $\boldsymbol{\mathcal{K}}$, without the need for a linear solver [45]. Thus, we can use a random finite difference to compute the three derivatives $\partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}}^T$, $\partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{M}}$, and $\partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}}$ in equation (18) separately. When selecting the value of δ

for these computations, we must balance the truncation error of the RFD (we will only consider centered differences) with the relative accuracy in computing the product of the operator (i.e $\mathcal{K}$, $\mathcal{M}$). If the matrix-vector products are computed directly, then we balance the truncation error with the machine precision and take $\delta \sim 10^{-3}$ when single precision is used [46], and $\delta \sim 10^{-6}$ for double precision. However, if we only compute the action of $\mathcal{M}$ to within some relative accuracy ϵ, as would be the case if we used the FMM or PSE method, we must take $\delta \sim \epsilon^{1/3}$.

To utilize equation (18), we first generate random forces and torques for each body p

$$\boldsymbol{W}_p^{FT} = k_B T \begin{bmatrix} L_p^{-1} \boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix}, \tag{19}$$

where $\boldsymbol{W}_p^f$, $\boldsymbol{W}_p^\tau$ are standard Guassian random vectors, and L_p is a measure of the body length. Note the choice of length scale used in the blocks of (19) is to minimize the variance of the RFD estimate, as we explain in appendix A. We then solve a mobility problem with random applied forces and torques $\boldsymbol{W}^{FT} = \begin{bmatrix} \boldsymbol{W}_p^{FT} \end{bmatrix}$, for both the random traction force $\boldsymbol{\lambda}^{\text{RFD}}$, and the random rigid velocity $\boldsymbol{U}^{\text{RFD}}$,

$$\boldsymbol{\lambda}^{\text{RFD}} = \mathcal{M}^{-1} \mathcal{K} \mathcal{N} \, \boldsymbol{W}^{FT} \tag{20}$$

$$\boldsymbol{U}^{\text{RFD}} = \mathcal{N} \, \boldsymbol{W}^{FT}. \tag{21}$$

To compute the relevant random finite difference terms, we randomly displace the particles to $\boldsymbol{Q}^{\pm}$, where

$$\boldsymbol{Q}_p^{\pm} = \boldsymbol{Q}_p \pm \frac{\delta}{2} \Delta \boldsymbol{Q}_p = \boldsymbol{Q}_p \pm \frac{\delta}{2} \begin{bmatrix} L_p \boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix}. \tag{22}$$

Using this and equation (18), we are able to compute the necessary drift term using random finite differences as

$$\begin{aligned} \text{Drift} = &-\frac{1}{\delta} \mathcal{N} \left\{ \mathcal{K}^T(\boldsymbol{Q}^+) - \mathcal{K}^T(\boldsymbol{Q}^-) \right\} \boldsymbol{\lambda}^{\text{RFD}} \\ &+ \frac{1}{\delta} \mathcal{N} \mathcal{K}^T \mathcal{M}^{-1} \left\{ \mathcal{M}(\boldsymbol{Q}^+) - \mathcal{M}(\boldsymbol{Q}^-) \right\} \boldsymbol{\lambda}^{\text{RFD}} \\ &- \frac{1}{\delta} \mathcal{N} \mathcal{K}^T \mathcal{M}^{-1} \left\{ \mathcal{K}(\boldsymbol{Q}^+) - \mathcal{K}(\boldsymbol{Q}^-) \right\} \boldsymbol{U}^{\text{RFD}} \\ \approx &\left(-\mathcal{N} \left\{ \partial_{\boldsymbol{Q}} \mathcal{K}^T \right\} \mathcal{M}^{-1} \mathcal{K} \mathcal{N} \right. \\ &+ \mathcal{N} \mathcal{K}^T \mathcal{M}^{-1} \left\{ \partial_{\boldsymbol{Q}} \mathcal{M} \right\} \mathcal{M}^{-1} \mathcal{K} \mathcal{N} \\ &\left. - \mathcal{N} \mathcal{K}^T \mathcal{M}^{-1} \left\{ \partial_{\boldsymbol{Q}} \mathcal{K} \right\} \mathcal{N} \right) : \left[\boldsymbol{W}^{FT} \, (\Delta \boldsymbol{Q})^T \right] \\ = &\partial_{\boldsymbol{Q}} \mathcal{N} : \left[\boldsymbol{W}^{FT} \, (\Delta \boldsymbol{Q})^T \right], \end{aligned} \tag{23}$$

where all operators and derivatives are evaluated at the same point $\boldsymbol{Q}$ unless otherwise noted and $(\Delta \boldsymbol{Q})^T$ denotes the transpose of $\Delta \boldsymbol{Q}$. Hence, in expectation, we have

$$\langle \text{Drift} \rangle \approx k_B T \ \partial_{\boldsymbol{Q}} \cdot \boldsymbol{\mathcal{N}}. \tag{24}$$

This computation is detailed in index notation, accounting for the constrained quaternion representation of orientations, in Appendix B 2.

To leading order in δ, the method of computing the drift proposed in equation (23), termed the *traction-corrected* RFD, is equivalent to the direct RFD on $\boldsymbol{\mathcal{N}}$ used in appendix A when exact linear algebra is used. However, using the traction-corrected RFD allows the use of inexact, iterative mobility solvers, without incurring additional restrictions on the small parameter δ from the prescribed solver tolerance. Furthermore, we are able to capture the drift term in equation (1) with only two saddle point solves rather than the three required if we were to use an RFD on $\boldsymbol{\mathcal{N}}$ directly. Our Euler-Maruyama Traction (EM-T) Scheme is summarized in algorithm 1, and is analyzed in Appendix B 2.

B. Trapezoidal Slip (T-S) Scheme

In section III A, we developed a method to efficiently and accurately generate the necessary drift term in an Euler-Maruyama scheme. However, when second order deterministic accuracy is desired, we may wish to use a midpoint or trapezoidal scheme [33]. Some higher order methods, however, will generate additional drift terms due to the Brownian increment being evaluated at multiple time levels. As an example, consider a naive two-solve implementation of the trapezoidal scheme:

$$\widetilde{\boldsymbol{Q}} = \boldsymbol{Q}^n + \Delta t \boldsymbol{\mathcal{N}}^n \boldsymbol{F}^n + \sqrt{2\Delta t k_B T} \left(\boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \right)^n \left(\boldsymbol{\mathcal{M}}^{1/2} \right)^n \boldsymbol{W}^n \tag{25}$$

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n + \frac{\Delta t}{2} \left(\boldsymbol{\mathcal{N}}^n \boldsymbol{F}^n + \widetilde{\boldsymbol{\mathcal{N}}} \widetilde{\boldsymbol{F}} \right) \tag{26}$$
$$+ \sqrt{\frac{\Delta t k_B T}{2}} \left\{ \left(\boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \right)^n + \widetilde{\boldsymbol{\mathcal{N}}} \widetilde{\boldsymbol{\mathcal{K}}}^T \widetilde{\boldsymbol{\mathcal{M}}}^{-1} \right\} \left(\boldsymbol{\mathcal{M}}^{1/2} \right)^n \boldsymbol{W}^n,$$

where superscripts and tildes indicate the point at which quantities are evaluated, e.g., $\widetilde{\boldsymbol{\mathcal{N}}} \equiv \boldsymbol{\mathcal{N}} \left(\widetilde{\boldsymbol{Q}} \right)$.

As shown in Appendix B 3, the thermal drift produced by the final velocity update in equation (26) (in expectation) is

$$\langle \text{Drift part 1} \rangle = \left\langle \frac{\boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n}{\Delta t} \right\rangle \approx (k_B T) \ \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}^T \boldsymbol{\mathcal{M}}^{-1} \left\{ \partial_{\boldsymbol{Q}} \boldsymbol{\mathcal{K}} \right\} : \boldsymbol{\mathcal{N}}, \tag{27}$$

Algorithm 1 Euler-Maruyama Traction (EM-T) Scheme

---

1. Compute relevant quantities for capturing drift:

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

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

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

   (b) Solve RFD mobility problem:

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

   (c) Randomly displace particles to:

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

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

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

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

2. Compute $\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n$ using a preconditioned Lancoz method or PSE.

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

   $$\begin{bmatrix} \boldsymbol{\mathcal{M}}^n & -\boldsymbol{\mathcal{K}}^n \\ -\left(\boldsymbol{\mathcal{K}}^T\right)^n & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \boldsymbol{U}^n \end{bmatrix} = \begin{bmatrix} -\boldsymbol{D}^S - \sqrt{2k_B T/\Delta t}\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n \\ -\boldsymbol{F}^n + \boldsymbol{D}^F \end{bmatrix}.$$

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

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

We recognize this as the third term in equation (18) and hence, we may use it to generate the full, desired drift. Examining equation (18) reveals that we must generate the remaining two terms

$$-\boldsymbol{\mathcal{N}}\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{K}}^T\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{N}}\boldsymbol{\mathcal{K}}^T\boldsymbol{\mathcal{M}}^{-1}\left\{\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{M}}\right\} : \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{N}}, \tag{28}$$

in order to capture the desired drift.

In section III A, we generated random traction forces of the form $\mathcal{M}^{-1}\mathcal{K}\mathcal{N}\boldsymbol{W}^{FT}$, and used these as the $\Delta\boldsymbol{P}$ in equation (17) to compute a traction-corrected RFD approximation to (28). Here we propose a different *slip-corrected* RFD method to compute the two terms in (28). For each body, we generate a vector of random blob displacements $\breve{\boldsymbol{W}}^D = \left[L_p \boldsymbol{W}_p^s\right]$, and random blob forces $\breve{\boldsymbol{W}}^F = \left[\frac{k_B T}{L_p}\boldsymbol{W}_p^s\right]$, where L_p is a length scale for body p, and $\boldsymbol{W}^s$ is a random Gaussian vector. We may then compute rigid body displacements, $\Delta\boldsymbol{Q}^{\mathrm{RFD}} = \mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\breve{\boldsymbol{W}}^D$, which may be used as $\Delta\boldsymbol{Q}$ in equation (17) to compute an RFD approximation to (28). That is, we may approximate the missing drift terms (28) by computing

$$
\begin{aligned}
\text{Drift part 2} = &-\frac{1}{\delta}\mathcal{N}\left\{\mathcal{K}^T(\boldsymbol{Q}^+) - \mathcal{K}^T(\boldsymbol{Q}^-)\right\}\ \breve{\boldsymbol{W}}^F \\
&+\frac{1}{\delta}\mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\left\{\mathcal{M}(\boldsymbol{Q}^+) - \mathcal{M}(\boldsymbol{Q}^-)\right\}\ \breve{\boldsymbol{W}}^F \\
\approx &\left(-\mathcal{N}\left\{\partial_{\boldsymbol{Q}}\mathcal{K}^T\right\} + \mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\left\{\partial_{\boldsymbol{Q}}\mathcal{M}\right\}\right) : \left[\breve{\boldsymbol{W}}^F\ \left(\Delta\boldsymbol{Q}^{\mathrm{RFD}}\right)^T\right] \\
= &\left(k_B T\right)\left(-\mathcal{N}\left\{\partial_{\boldsymbol{Q}}\mathcal{K}^T\right\} + \mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\left\{\partial_{\boldsymbol{Q}}\mathcal{M}\right\}\right) : \mathcal{M}^{-1}\mathcal{K}\mathcal{N}\left[\boldsymbol{W}^s\left(\boldsymbol{W}^s\right)^T\right],
\end{aligned}
\tag{29}
$$

where as before $\boldsymbol{Q}_p^{\pm} = \boldsymbol{Q}_p \pm \frac{\delta}{2}\Delta\boldsymbol{Q}_p^{\mathrm{RFD}}$. Hence, in expectation we obtain the missing drift terms (28),

$$
\langle\text{Drift part 2}\rangle \approx k_B T\left(-\mathcal{N}\left\{\partial_{\boldsymbol{Q}}\mathcal{K}^T\right\} : \mathcal{M}^{-1}\mathcal{K}\mathcal{N} + \mathcal{N}\mathcal{K}^T\mathcal{M}^{-1}\left\{\partial_{\boldsymbol{Q}}\mathcal{M}\right\} : \mathcal{M}^{-1}\mathcal{K}\mathcal{N}\right),
$$

which combined with (27) gives us the desired drift $k_B T\ \partial_{\boldsymbol{Q}}\cdot\mathcal{N}$.

Our Trapezoidal Slip (T-S) scheme is summarized in algorithm 2, and is analyzed in Appendix B 3. It involves three mobility solves and one Lanczos computation per time step, just like the EM-RFD scheme given in Algorithm 3, however, T-S is second order deterministically just like the Fixman midpoint scheme.

It is important to point out that by using either the traction-corrected RFD (i.e., applying random uncorrelated forces and torques on the particles) or the slip-corrected RFD (i.e., applying random uncorrelated slip on the particles' surfaces), one can construct a multitude of schemes that give the desired stochastic drift term in expectation for sufficiently small Δt. For example, an alternative method to generate the remaining drift terms in (28), while still using the trapezoidal rule, would be to compute $\boldsymbol{D}^S$ and $\boldsymbol{D}^F$ analogous to step 1 of algorithm 1 but without the term involving $\boldsymbol{U}^{\mathrm{RFD}}$ (which is already included via the trapezoidal corrector step). In numerical tests, we have found such a Trapezoidal Traction (T-T) scheme to perform very similarly to the (T-S)

Algorithm 2 Trapezoidal Slip (T-S) scheme

---

1. Compute relevant quantities for capturing drift:

   (a) Generate random Gaussian directions $\boldsymbol{W}^s$ for each blob, and form the composite vectors of blob displacements $\breve{\boldsymbol{W}}^D = \left[L_p \boldsymbol{W}_p^s\right]$ and blob forces $\breve{\boldsymbol{W}}^F = \left[\frac{k_B T}{L_p} \boldsymbol{W}_p^s\right]$.

   (b) Solve RFD mobility (more precisely, displacement) problem:

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

   (c) Randomly displace particles to $\boldsymbol{Q}^\pm$:

   $$\boldsymbol{Q}^\pm = \boldsymbol{Q}^n \pm \frac{\delta}{2} \Delta \boldsymbol{Q}^{\mathrm{RFD}}$$

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

   $$\boldsymbol{D}^F = \frac{1}{\delta} \left\{ \boldsymbol{\mathcal{K}}^T \left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{K}}^T \left(\boldsymbol{Q}^-\right) \right\} \breve{\boldsymbol{W}}^F,$$
   $$\boldsymbol{D}^S = \frac{1}{\delta} \left\{ \boldsymbol{\mathcal{M}} \left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{M}} \left(\boldsymbol{Q}^-\right) \right\} \breve{\boldsymbol{W}}^F.$$

   Note that different δ may be used for the two RFDs depending on the relative accuracy with which the action of $\boldsymbol{\mathcal{M}}$ is evaluated.

2. Compute $\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n$ using a preconditioned Lancoz method or PSE.

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

   $$\begin{bmatrix} \boldsymbol{\mathcal{M}}^n & -\boldsymbol{\mathcal{K}}^n \\ -\left(\boldsymbol{\mathcal{K}}^T\right)^n & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \boldsymbol{U}^n \end{bmatrix} = \begin{bmatrix} -\sqrt{2k_B T/\Delta t} \left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n \\ -\boldsymbol{F}^n \end{bmatrix}.$$

4. Update configurations to predicted position $\widetilde{\boldsymbol{Q}}$:

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

5. Evaluate forces and torques at $\widetilde{\boldsymbol{F}} = \boldsymbol{F}\left(\widetilde{\boldsymbol{Q}}, t\right)$ and solve corrector mobility problem at the predicted position $\widetilde{\boldsymbol{Q}}$:

   $$\begin{bmatrix} \widetilde{\boldsymbol{\mathcal{M}}} & -\widetilde{\boldsymbol{\mathcal{K}}} \\ -\widetilde{\boldsymbol{\mathcal{K}}}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\lambda}} \\ \widetilde{\boldsymbol{U}} \end{bmatrix} = \begin{bmatrix} -2\boldsymbol{D}^S - \sqrt{\frac{2k_B T}{\Delta t}} (\boldsymbol{\mathcal{M}}^{1/2})^n \boldsymbol{W}^n \\ -\widetilde{\boldsymbol{F}} + 2\boldsymbol{D}^F \end{bmatrix}.$$

6. Update configurations to corrected position $\boldsymbol{Q}^{n+1}$:

   $$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n + \frac{\Delta t}{2} \left(\boldsymbol{U}^n + \widetilde{\boldsymbol{U}}\right).$$

---

scheme for all solver tolerances and time step sizes examined. We also examined midpoint variants [34] of the (T-T) and (T-S) schemes, both of which require an additional application of $\mathcal{M}^{1/2}$ in the corrector step, and found them to be inferior in terms of cost-accuracy balance than the T-S scheme presented here [47]. Modifications of the ideas presented in section III A can be used to write a second order Adams-Bashforth traction (AB-T) scheme, in which the second-order AB multistep rule is used for the deterministic terms, while the drift term is computed analogously to the EM-T scheme [5]. We found, however, that the AB-T scheme was inferior in accuracy compared to the T-S scheme, especially for larger Δt.

IV.   RESULTS

In this section, we study the accuracy and efficiency of the numerical schemes presented in sections III A and III B for suspensions of rigid particles sedimented above a no-slip bottom wall. As mentioned in section II B, the bottom wall acts to screen the hydrodynamic interactions, thereby reducing the number of iterations required for iterative methods to converge to a desired tolerance [5, 10]. To prevent unphysical particle overlaps with the wall due to the Brownian motion, we include a soft repulsive wall-particle potential, and employ a regularized form of the blob-blob mobility which ensures that $\mathcal{M}$ is SPD and physical even when some blobs overlap the wall [5].

In sections IV A and IV B, we investigate the weak accuracy of our methods on suspensions of colloidal right-angle "boomerangs". Colloidal boomerangs have been manufactured using lithography [35], and the diffusion of a single boomerang above a wall was studied numerically in [11]. We will model a colloidal boomerang as an L-shaped body composed of 15 blobs, with each 'arm' of the boomerang being composed of 7 blobs in straight line, plus a common eighth blob shared by both arms, see Fig. 2. In [11] it was found that blobs centers should be spaced approximately a distance of a apart, where a is the hydrodynamic radius of a blob. Although geometrically simple, boomerangs do not have spherical, axial, or skew symmetry and therefore proper treatment of orientations is essential to correctly model colloidal diffusion [11]. In subsection IV A 1, we study two colloidal boomerangs connected by an elastic string. The small problem size allows us to reduce sampling (statistical) errors enough to accurately measure temporal accuracy, and also to compare the schemes developed in this work to Fixman's scheme [32], which requires dense linear algebra to be used. In subsection IV B we examine a suspension of many boomerangs to more effectively assess the accuracy-efficiency tradeoff for the schemes developed here.

In section IV C, we revisit some of the computational investigations reported in [5, 36] for active

suspensions of rotating colloids [4, 36]. In these suspensions thermal motion sets the equilibrium gravitational height of the colloids, and it is necessary to include Brownian motion to enable quantitative comparisons to experiments [5, 36]. At the same time, previous studies [5, 36] used a minimally-resolved representation of the hydrodynamics, with each particle represented by a single blob. This is not quantitatively accurate when the microrollers are close to the wall or other colloids, as in recent experiments [4, 36]. We represent the spheres using either 12 or 42 blobs [10] in order to improve the accuracy of the hydrodynamic interactions, and choose a as roughly half the distance between vertices in the multiblob sphere model following the recommendations in Sections IV and V of [10].
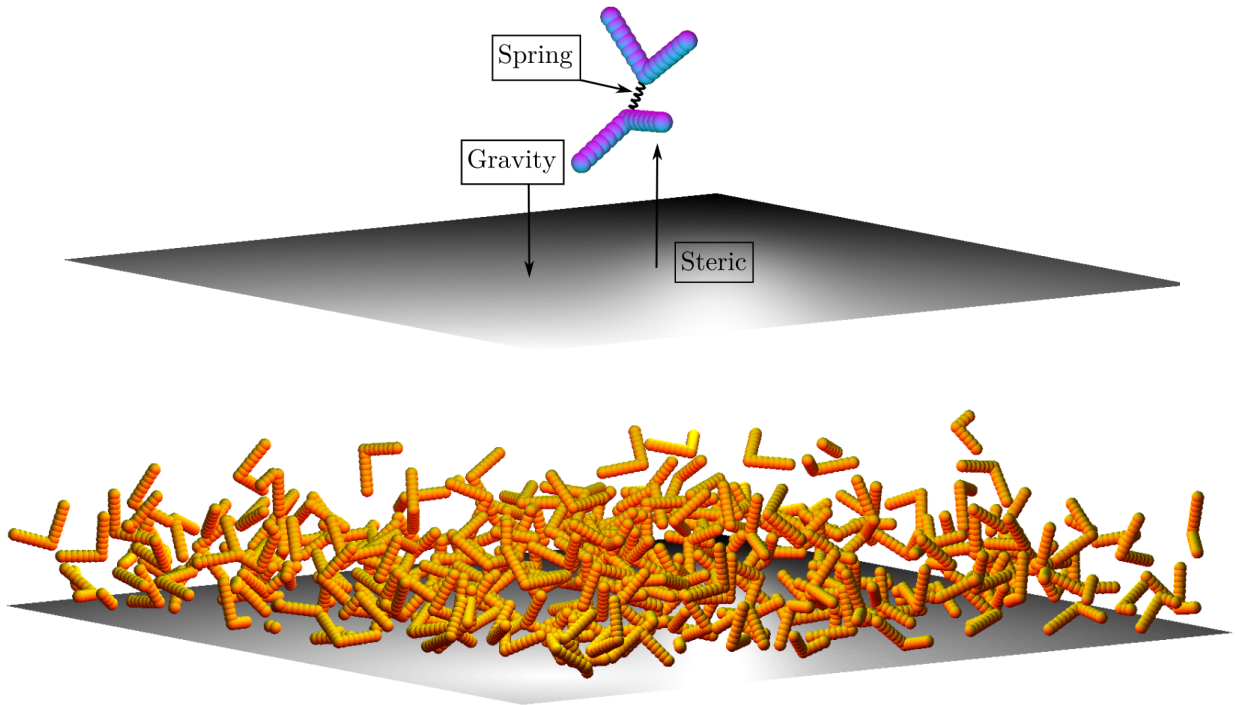
Figure 2: Illustrations of the test problems involving colloidal boomerangs. (Top panel) Sample configuration of a boomerang dimer for the numerical experiments conducted in section IV A 1. (Bottom panel) Sample configuration of a boomerang suspension for the numerical experiments conducted in section IV B. The shaded area is the part of the bottom wall that belongs to the central unit cell used for the pseudo-periodic boundary conditions.

A. Weak accuracy for a dimer of boomerangs

In this section we examine the equilibrium dynamics of a boomerang dimer formed from two colloidal boomerangs connected by a harmonic spring, as illustrated in the top panel of Fig. 2. The form of the potential connecting the two boomerangs is

$$U_{\text{spring}}(d) = \frac{\kappa}{2} \left(d - l_0\right)^2, \tag{30}$$

where d is the distance between the two boomerang's "cross-points" (the elbow of the L-shaped bodies), l_0 is a rest length, and κ is the spring constant. The steric interactions of the individual blobs are captured through a repulsive Yukawa potential,

$$U_{\text{steric}}(r) = \gamma \frac{e^{-r/b}}{r}, \tag{31}$$

where b is the Debye length and γ is the repulsion strength. This Yukawa potential potential is also used for the steric interactions of the bodies with the wall. In this section, we take $l_0 = 1\mu$m, $\gamma = \kappa = 0.096 \frac{\text{mg}}{\text{s}^2}$, and $b = 0.162\mu$m. We will take the blob radius to be $a = 0.324\mu$m and each blob will have a buoyant (excess) mass $m_e = 1.57 \times 10^{-11}$mg, giving a net gravitational force $m_e g$ on each blob, where $g = 9.81\frac{\text{m}}{\text{s}^2}$. The total force and torque on each body are computed by adding contributions of the spring, gravity, and steric repulsion over all the blobs comprising the body [10]. The bodies are suspended in water, $\eta = 1$ mPa·s, at approximately room temperature, $T = 300$K. In these investigations, we nondimensionalize the time step using the diffusive time scale for a single blob,

$$\Delta\tau = \frac{k_B T}{6\pi\eta a^3} \Delta t.$$

In the absence of non-conservative forces (i.e., for passive suspensions), the equilibrium distribution for the particles' configuration is the familiar Gibbs-Boltzmann (GB) distribution

$$P_{\text{eq}}\left(\boldsymbol{Q}\right) = P_{\text{GB}}\left(\boldsymbol{Q}\right) = Z^{-1} \exp\left(-U\left(\boldsymbol{Q}\right)/k_B T\right),$$

where $U\left(\boldsymbol{Q}\right)$ is the conservative potential from which the external forces and torques are obtained. As demonstrated in our prior work [11, 24], failure to consistently include the stochastic drift term in BD simulations leads to strong deviations from $P_{\text{GB}}\left(\boldsymbol{Q}\right)$ in the presence of confinement. Therefore, a strong test that our methods are consistent with the overdamped Langevin equation (1) is to examine how closely they reproduce (marginals of) the GB equilibrium distribution, as we do in subsection IV A 1. We use a Markov-chain Monte Carlo (MCMC) method to very accurately

sample the GB equilibrium distribution and use this data to compute the error produced by each scheme. At the same time, it is important to also confirm that our schemes, unlike MCMC, correctly reproduce the *dynamics* of the particles even for time steps that are on the order of the diffusive time scale, as we do in subsection IV A 2.

1. Static Accuracy

The stability limit for the EM-T scheme for the chosen parameters was empirically estimated to be $\Delta\tau \lesssim 0.3$. In Figure 3 we study how well our numerical methods reproduce selected the Gibbs-Boltzmann equilibrium distribution for $\Delta\tau = 0.072, 0.144, 0.288$. We have examined a number of marginals of the equilibrium distribution, but we focus here on the equilibrium distributions of the boomerang cross-point to cross-point distance. We use a relative tolerance of 10^{-4} in all iterative methods for the computations done in this section. An investigation into the effect of solver tolerance on the accuracy of the EM-T and T-S schemes showed no change in temporal accuracy for all solver tolerances less than or equal to 10^{-3}, and overall accuracy was only slightly affected for solver tolerances $10^{-3} - 10^{-2}$, but then degraded rapidly for looser tolerances. Note however, that using the same solver tolerance for all iterative methods is perhaps not necessary to maintain temporal accuracy, and looser tolerances may be used for the RFD-related linear solves. We take the random finite difference parameter $\delta = 10^{-6}$ for both schemes as double precision was used for these calculations. Results were obtained by averaging 20 independent realizations containing 10^5 samples, initialized from unique configurations sampled from the equilibrium distribution using the MCMC algorithm. In addition to the proposed EM-T and T-S schemes, we also compare with Fixman's scheme given in Section III.B of [24], implemented using dense linear algebra.

Figure 3 shows that even for the smallest time step size considered, the T-S scheme is substantially more accurate than the EM-T scheme. Note that no data is included for Fixman's scheme for the largest time step size considered, because the scheme was seen to be numerically unstable. While all of the schemes are seen to be $\mathcal{O}(\Delta t)$ in the cummulative L_2 error, the order constant for the T-S and Fixman schemes are clearly much lower than the EM scheme. In terms of accuracy alone, the T-S scheme compares very favorably with Fixman's scheme, while also enabling scalable computations for suspensions of many rigid bodies.
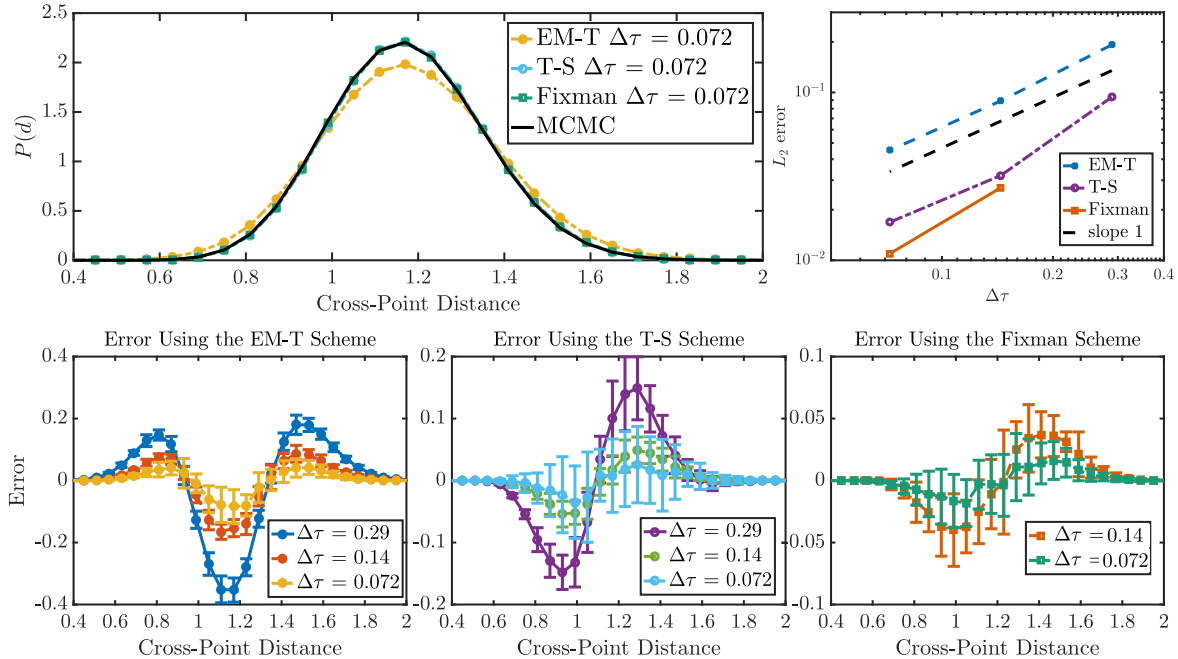
Figure 3: Numerical errors in the equilibrium distribution for a boomerang dimer. (Upper left panel) Comparison between the correct (marginal of the) Gibbs-Boltzmann distribution of the cross-point distance, computed using an MCMC method, and numerical results from the T-S and EM-T schemes for normalized time step size $\Delta\tau = 0.072$. (Upper right panel) Cumulative error in the distribution of the cross-point distance, as measured by the L_2 norm of the error in the histogram $P(d)$, for the EM-T, T-S and Fixman schemes. (Lower panels) Error in the distribution of the cross-point distance for the EM-T (left panel), T-S (middle panel), and the Fixman scheme (right panel) for several different time step sizes (see legend). Note that the scale of the plots changes and that the Fixman scheme is unstable for $\Delta\tau > 0.29$. Error bars indicate 95% confidence intervals and are estimated from multiple independent runs.

2. Dynamic Accuracy

We now turn our attention to time-dependent statistics by examining the equilibrium translational mean squared displacement (MSD) of the cross point of one of the two connected boomerangs,

$$\boldsymbol{D}(t) = \langle \Delta\boldsymbol{q}_p(t) \left(\Delta\boldsymbol{q}_p(t)\right)^T \rangle = \langle \left(\boldsymbol{q}_p(t) - \boldsymbol{q}_p(0)\right) \left(\boldsymbol{q}_p(t) - \boldsymbol{q}_p(0)\right)^T \rangle,$$

where the average is an ensemble average over equilibrium trajectories and $p = 1$ or $p = 2$. Since the trajectories of the two boomerangs are statistically identical, we will average results over the two particles to improve statistical accuracy. Here and in what follows we will assume that the cross point is chosen as the tracking point around which the boomerang rotates. We may define

short-time and *long-time* translational diffusion tensors,

$$\boldsymbol{\chi}_{st} = \frac{1}{2} \lim_{t \to 0} \frac{\boldsymbol{D}(t)}{t}, \quad \boldsymbol{\chi}_{lt} = \frac{1}{2} \lim_{t \to \infty} \frac{\boldsymbol{D}(t)}{t} \tag{32}$$

respectively. The Stokes-Einstein relationship implies

$$\boldsymbol{\chi}_{st} = (k_B T) \left\langle \boldsymbol{\mathcal{N}}_{pp}^{(tt)} \right\rangle_{GB}, \tag{33}$$

where we have taken an average over the Gibbs-Boltzmann distribution of the 3×3 translation-translation diagonal block of the mobility matrix corresponding to body p. However, $\boldsymbol{\chi}_{lt}$ admits no such simple characterizations and is typically challenging to compute accurately, requiring many samples from long simulations, as discussed extensively in [11].

Since we are investigating diffusion near an infinite wall (placed at $z = 0$ with normal in the positive z direction), under the influence of gravity, we may define the parallel ($D_p^{\parallel}$) and perpendicular ($D_p^{\perp}$) MSD of body p as

$$D_p^{\parallel}(t) = \boldsymbol{D}_{xx}(t) + \boldsymbol{D}_{yy}(t), \qquad D_p^{\perp}(t) = \boldsymbol{D}_{zz}(t). \tag{34}$$

At long times the perpendicular MSD $D_p^{\perp}(t)$ asymptotically tends towards a finite value, related to the gravitational height of the body [11]. We focus here on the parallel MSD $D_p^{\parallel}(t)$ as this is typically what is measured in experiments [35, 37].

At short times, we can use the Stokes-Einstein formula $D_p^{\parallel}(t) = 2 \, (k_B T) \left\langle \boldsymbol{\mathcal{N}}_{11}^{(xx)} \right\rangle_{GB} t$ to validate our simulations. To estimate the long-time MSD, we use a non-equilibrium method based on linear response theory [38]. Specifically, if we pull one of the boomerangs with a force $\boldsymbol{F} = F \, \hat{\boldsymbol{x}}$ applied to the cross (tracking) point,

$$\langle x_p(t) - x_p(0) \rangle_{\boldsymbol{F}} = -\frac{F}{k_B T} \int_0^t \langle x_p(0) \dot{x}_p(t - t') \rangle_0 \, \mathrm{d}t' = \frac{F}{2k_B T} \langle (x_p(t) - x_p(0))^2 \rangle_0. \tag{35}$$

Here the average on the left hand side is an average over nonequilibrum trajectories initialized from the GB distribution, while the average on the right hand side is an average over equilibrium trajectories. The formula (35) relates the MSD at equilibrium with the mean displacement under a external force. The nonequilibrium method offers better statistical accuracy at long times over computing the MSD if the applied force F is sufficiently large but still small enough to remain in the linear-response regime (for the simulations reported below the Péclet number is Pe $= L F / (k_B T) \approx$ 0.5, where $L = 2.1 \mu$m is the boomerang arm length). To see this, consider a one dimensional diffusion process with constant mobility μ,

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \mu F + \sqrt{2k_B T \mu} \, \mathcal{W}(t),$$

whose solution has mean $\langle x(t) \rangle = \mu F t$ and standard deviation $\sqrt{2 k_B T \mu t}$. The relative statistical uncertainty in the mean displacement $\langle x(t) \rangle$ is $\sqrt{2 k_B T / (\mu F^2 t)}$, and therefore decays as time grows. By contrast, in the absence of the force the mean MSD is $2 k_B T \mu t$ while the standard deviation of the MSD is $\sqrt{3} (2 k_B T \mu t)$, and therefore the relative statistical uncertainty in the MSD is independent of time. This assumes we have an infinitely-long trajectory. In practice, however, the finite length of the trajectories makes the MSD most statistically accurate at short times, and it is beneficial to use the nonequilibrium method to estimate the long-time diffusion coefficient.

Figure 4 shows the results for the MSD of the cross point of one of the boomerangs obtained using the EM-T and T-S schemes. We can see that both schemes produce the correct slope of the MSD at short times (short-time diffusion coefficient), as compared with the Stokes-Einstein estimate obtained by computing $\left\langle \boldsymbol{\mathcal{N}}_{11}^{(xx)} \right\rangle_{GB}$ accurately using a Monte-Carlo method (solid black line). At long times, to within statistical uncertainty, both schemes produce the same slope of the MSD (long-time diffusion coefficient) as the non-equilibrium method (dashed black line). Because the short time MSD is computed as an equilibrium average over the Gibbs-Boltzmann distribution (see (33)), the temporal accuracy with which a given scheme samples the equilibrium GB distribution (as measured in section IV A 1) directly effects the accuracy of the short time MSD. In particular, the stochastic displacement produced by the EM-T scheme has covariance proportional to $\Delta t \boldsymbol{\mathcal{N}}(\boldsymbol{Q})$. Hence, the short time diffusion coefficient produced by the EM-T scheme is independent of Δt, and the only source of error stems from the error in the equilibrium distribution. The inset of Fig. 4 shows the short time MSD produced by the EM-T scheme for different time steps. Here we see clear improvement as the time step is reduced, analogous to the results shown in Fig. 3.

B.   Accuracy and Efficiency for Many-Body Suspensions

To compare the accuracy-efficiency tradeoff for the T-S scheme versus the EM-T scheme, we investigate a dense suspension of freely-diffusing colloidal boomerangs. All of the physical parameters and interaction potentials of the simulation are taken to be the same as for the boomerang dimer studied in Section (IV A), except that we take the solver tolerance for all iterative solvers to be 10^{-3} as this was found to give statistically indistinguishable results from tolerance 10^{-4}. Further, to reduce the computation time, we perform the arithmetic on the GPU (multiplication by $\boldsymbol{\mathcal{M}}$) in single precision and thus take random the finite difference parameter $\delta = 10^{-3}$ for both schemes. For suspensions we employ pseudo periodic boundary conditions [5]. Specifically, for each blob i, we sum the hydrodynamic interactions with every other blob j (using the minimum
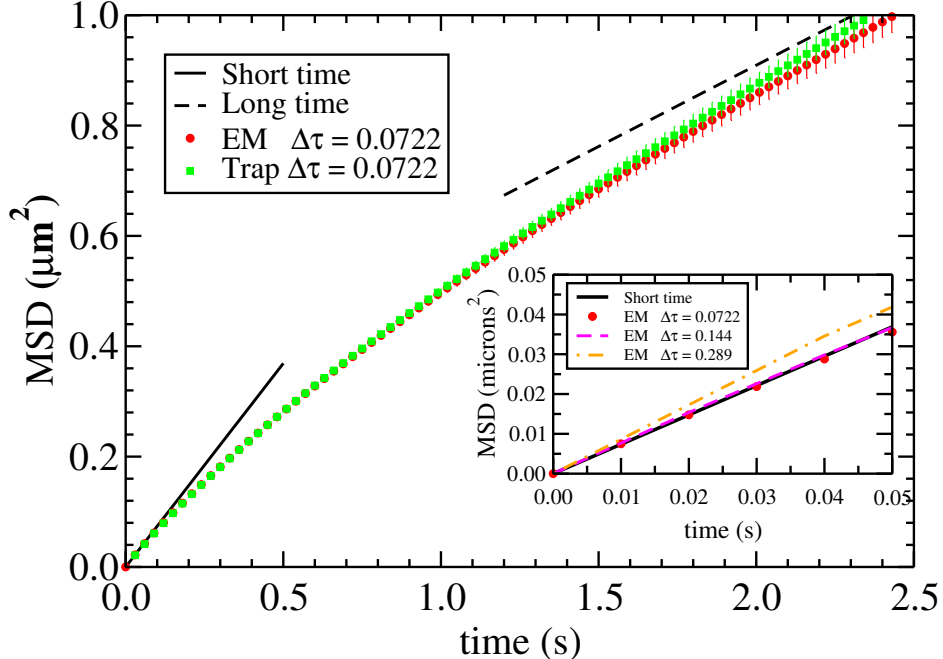
Figure 4: Mean square planar displacement of the cross point of one of the boomerangs as obtained from trajectories sampled using the EM-T and T-S schemes. Error bars indicate 95% confidence intervals. Black lines show the reference slope of the MSD at short times (solid), as computed using the Stokes-Einstein formula, and long times (dashed), as computed using a non-equilibrium method. (Inset) Short time parallel MSD computed by the EM-T method for different time step sizes.

image convention) and also the 8 nearest periodic images of blob j. This is fairly effective in capturing the hydrodynamics of an infinitely periodic suspension, without requiring involved Ewald summation techniques [39], since the presence of the wall screens the hydrodynamic interactions to decay as inverse *cubed* distance. In order to challenge the temporal integrators we consider a dense suspension in which steric exclusion plays a large role in the dynamics.

Specifically, we will simulate 256 boomerangs in a domain which is semi-infinite in z and periodic with unit cell of length 45.3μm in both the x and y directions, as illustrated in the bottom panel of Fig. 2. We have examined a number of relevant statistics (marginals of the equilibrium distribution) and found the radial distribution function $g(r)$ to be the most sensitive measure. We compute $g(r)$ using the minimum Euclidean distance between two boomerangs, approximated as the smallest distance between a pair of blobs taken from distinct bodies. We normalize $g(r)$ as if the suspension were two dimensional, as was done in [5], in order to ensure that $g(r) \to 1$ for large r. For each scheme and value of $\Delta\tau$, we simulate 8 independent trajectories with 10^4 samples in each, initialized using unique configurations sampled from the equilibrium distribution using an MCMC algorithm.
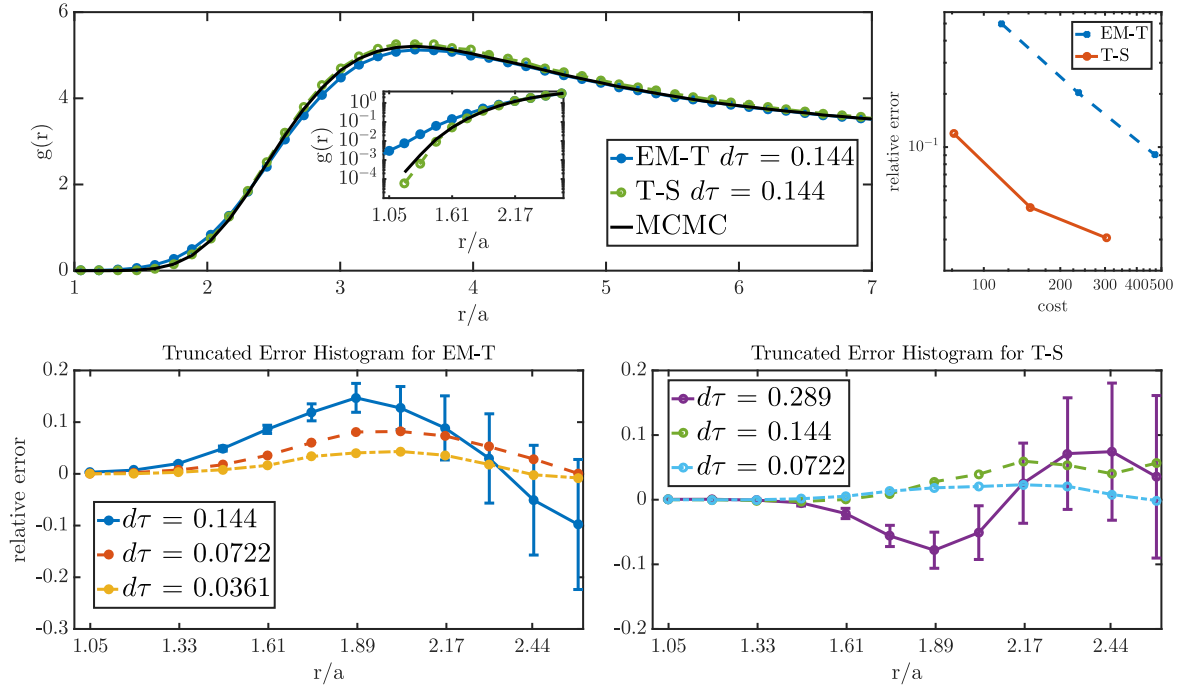
Figure 5: Accuracy of the radial distribution function $g(r)$ for a dense suspension of boomerangs (illustrated in the bottom panel of Fig. 2), for several values of the normalized time step size $\Delta\tau$. (Upper left panel) Comparison with the correct $g(r)$ computed using an MCMC method, for $\Delta\tau = 0.144$ for the EM-T and T-S schemes. The semi-log scale inset zooms in on nearly touching boomerangs to reveal a much larger amount of particle (near) overlaps for the EM-T scheme. (Upper right panel) Cumulative error as a function of cost per unit time (see main text). (Lower panels) Error in the radial distribution function $g(r)$ for the EM-T (left panel) and T-S (right panel) schemes for several different time step sizes (see legend). Error bars indicate 95% confidence intervals.

The bottom two panels of Fig. 5 show that the T-S scheme is notably more accurate than the EM-T scheme for the same value of the time step $\Delta\tau$. It is also worthy of note that we were able to run the T-S scheme with fair accuracy using $\Delta\tau = 0.29$, which was seen to be unstable for the EM-T scheme. However, while the T-S scheme is more accurate, it also requires one more mobility solve per time step than the EM-T scheme. Therefore, to really determine which scheme is best for large-scale simulation, we must define a notion of accuracy and cost and determine which scheme achieves a given level of accuracy for a smaller computational cost.

We define a cumulative measure of accuracy using a modified L_2 error of the $g(r)$ histograms relative to reference values computed with high statistical accuracy using an MCMC algorithm. We

account for statistical errors by considering a weighted L_2 norm proportional to the log-likelihood,

$$\text{Error} := \sqrt{\frac{1}{2} \int_0^R \left(\frac{g^{\Delta t}(r) - g_{\text{MCMC}}(r)}{\sigma(r)} \right)^2 dr}, \tag{36}$$

where $\sigma(r)$ is the standard deviation estimated empirically using multiple independently-seeded simulations. We take $R = 2.5a$ since for $r \gtrsim 2.5a$ the error in $g(r)$ is dominated by sampling (statistical) error for both schemes.

Since in our specific case the computational cost is dominated by (dense) multiplications of the blob-blob mobility matrix with a vector, we define the cost per unit time as the (average) total number of multiplications by $\boldsymbol{\mathcal{M}}$ per time step, divided by $\Delta \tau$. We observe that when a solver tolerance of 10^{-3} is used, the preconditioned iterative methods to solve the saddle point system, and to compute $\boldsymbol{\mathcal{M}}^{1/2}$, will both converge in 5 iterations most of the time. Thus, the total number of multiplications per time step for the T-S scheme is 22 ($3 \times 5 + 5 + 2$ for three mobility solves, one Lanczos iteration, and one RFD on $\boldsymbol{\mathcal{M}}$), while the EM-T scheme requires 17 ($2 \times 5 + 5 + 2$).

The upper right panel of Fig. 5 shows that the T-S scheme costs less per unit time than the EM-T scheme for any desired accuracy. We will therefore use the T-S schemes in Section IV C, and recommend it for suspensions confined above a no-slip wall. Nevertheless, the cost of each scheme depends heavily on how expensive it is to compute the action of $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{M}}^{\frac{1}{2}}$, and we recommend repeating the cost-accuracy balance computations reported here for each specific application/code.

C. Uniform suspensions of Brownian Rollers

Active suspensions of rotating spherical colloids (microrollers) sedimented above a bottom wall have been recently investigated using both experiments and simulations [4]. The colloids have an embedded hematite which makes them weakly ferromagnetic and thus easily rotated by an external magnetic field, as illustrated in Fig. (6). Because of the presence of a nonzero rotation-translation coupling due to the bottom wall, micro-rollers translate parallel to the wall. Collective flow effects dominate the dynamics of many-body suspensions, and the particles translate much faster at larger densities. For non-uniform suspensions, shocks were observed to form and destabilize into fingering instabilities, and deterministic simulations were performed to interrogate the observations. In [5], the effects of Brownian motion were included in the simulations to demonstrate the quantitative effect that fluctuations have on the the development and progression of the fingering instability. In particular, it is important to note that Brownian motion sets the equilibrium gravitational height of the colloids, and therefore must be included to obtain quantitative predictions that can directly

be compared to experiments. In [36], the nonlocal nature of the shock front was further elucidated, and propagation of density waves in a uniform suspension translating parallel to the wall was investigated using both experiments and simulation. One of the key parameters that enters in the simplified equations describing the dynamics of density fluctuations (see Eq. (4) in [36]) around a uniform state is the mean suspension velocity $\bar{V}$.
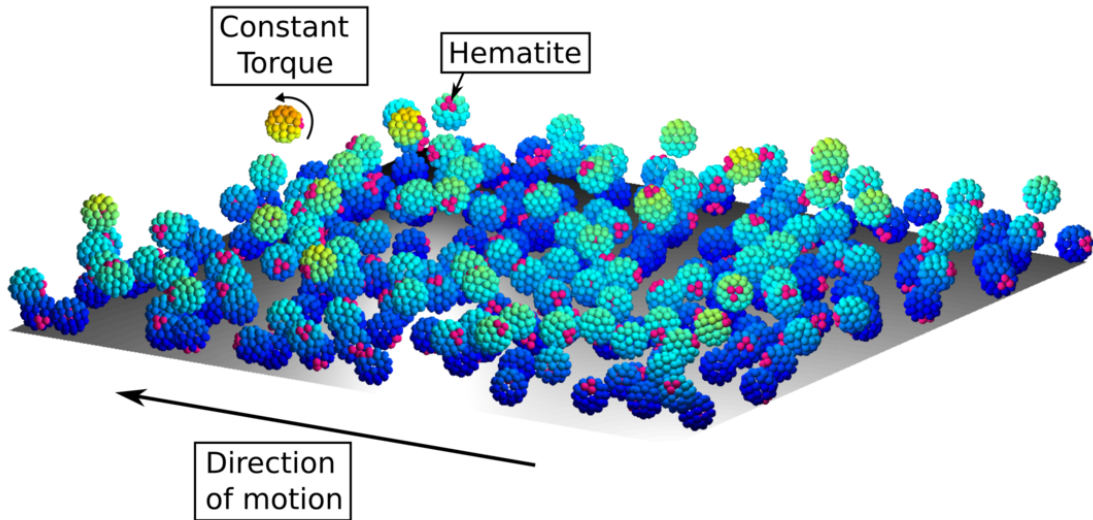

Figure 6: A snapshot of a steadily-translating uniform suspension of 256 microrollers, each made up of 42 blobs (colored by their height above the floor), at planar packing density $\phi = 0.4$. Each particle has an embedded magnet, illustrated as a cluster of fuchsia blobs. Note that although for constant applied torque the particle orientation does not enter in the equations of motion for translation, our algorithm keeps track of the orientation of each colloid, which can be used to more accurately compute a time-dependent magnetic torque on the particles if desired.

At higher densities, the mean velocity $\bar{V}$ is dominated by collective effects and near-field hydrodynamic interactions between the particles and between the particles and the wall. In all prior work [5, 36], rollers were represented using only one blob, and the Rotne-Prager-Blake tensor was used to add the active translation as a deterministic forcing term. In [10], it was demonstrated that using more blobs to discretize spherical particles gives much greater accuracy for hydrodynamics. We are here able to, for the first time, consistently and sufficiently accurately resolve hydrodynamics *and* account for thermal fluctuations, and thus obtain *quantitative* predictions that can be compared to experiments. Following [5], we take $\eta = 1$ mPa·s, the hydrodynamic radius of the particles $R_h = 0.656$ μm, excess (buoyant) mass $m_e g = 1.24 \times 10^{-14} \frac{\text{kg m}}{\text{s}^2}$, and apply a constant, identical torque on every particle, $\boldsymbol{T} = 8\pi\eta\omega R_h^3 \widehat{\boldsymbol{y}}$, where we take the angular frequency $\omega = 10$Hz. We use

the T-S scheme with $\Delta t = 0.008$s. The particle-particle and particle-wall interaction potentials are as described in [5]. We discretize the rollers using 1, 12, or 42 blobs (illustrated in Fig. (6)), following [10]. It is important to note that for 12 or 42 blobs per particle the translation-rotation coupling inducing the active motion is captured by the multiblob model itself rather than added by hand as it is for a single blob. After an initial, transient period, we computed individual particle velocities over intervals of 1/24s, and collected histograms of particles' velocities at the steady translating state [48]. Different time intervals to compute the velocity were also explored but no substantial change was observed.
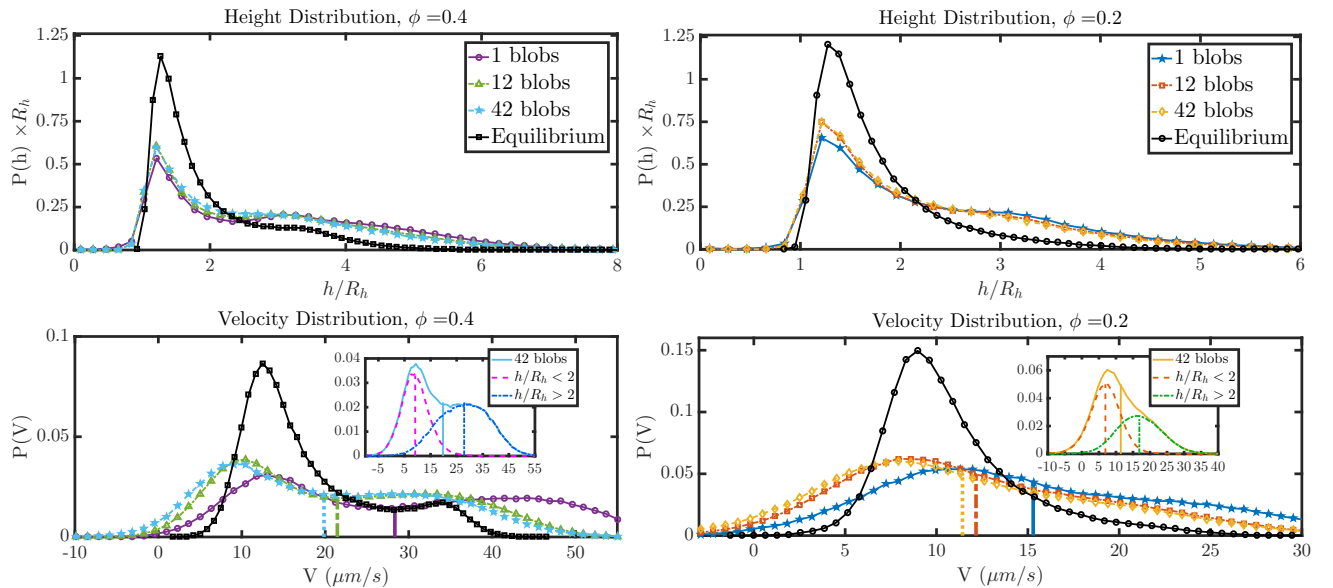

Figure 7: Histograms of the microrollers' heights above the wall ($P(h)$, top panels) and their velocities ($P(V)$, bottom panels), for two packing densities in the plane, $\phi = 0.2$ (left panels) and $\phi = 0.4$ (right panels), for a (pseudo)periodic active suspension at steady state. Solid vertical lines demarcate the mean of the velocity distributions. Curves marked "equilibrium" use particle positions sampled from the equilibrium GB distribution (in the absence of activity) using an MCMC method. All other curves are results of dynamic BD simulations using the T-S scheme and either 1, 12 or 42 blobs to resolve each spherical colloid. (Top panels) Comparison of the height distribution $P(h)$ for $\phi = 0.2$ (right) and $\phi = 0.4$ (left), as set by a balance of thermal noise, active vertical flows and gravity. (Bottom panels) Comparison of the velocity distribution $P(V)$ for $\phi = 0.2$ (right) and $\phi = 0.4$ (left). For the curves marked as equilibrium, velocities were generated by solving a deterministic mobility problem with particles discretized by 42 blobs, using configurations sampled by an MCMC method. Insets show $P(V)$ for the finest resolution split into two groups based on particle height ($h \lessgtr 2R_h$), where the normalization factor for the distributions is based on the fraction of the total number of particles in the given subgroup.

Figure 7 shows histograms of the particles' heights above the wall ($P(h)$, top panel) and their velocities ($P(V)$, bottom panels), for two packing densities in the plane, $\phi = n\left(\pi R_h^2\right) = 0.2$ (left panels) and $\phi = 0.4$ (right panels), where n is the number density in the plane. We use 256 particles for each case; no significant change in the results was observed when 1024 particles were used, confirming that finite size effects are small. The equilibrium Gibbs-Boltzmann height distributions were computed by using an MCMC method, and are not affected by the resolution of the multiblobs, which only changes the (hydro)dynamics of the suspension. To understand the effects on $P(V)$ due to changes in $P(h)$ caused by the active motion, we estimate the distribution of particles velocities at a fictitious "equilibrium" state by sampling particle positions from the equilibrium GB distribution using MCMC. We then apply a constant torque $\boldsymbol{T}$ on each particle and solve a *deterministic* mobility problem (using the 42-blob model of the particles) for the particles' velocities. The bottom panels in Fig. 7 show a large mismatch between these static predictions and the actual dynamics of the particles. This underscores the importance of explicitly simulating the Brownian dynamics in this system in order to sample the out-of-equilibrium steady state distribution, which is quite different from the GB distribution due to the strong active flows.

The bottom panels in Fig. 7 demonstrate that the more resolved results for $P(V)$ obtained using 12 and 42 blobs closely match for both packing densities, while the minimally-resolved distributions using just one blob per particle are fairly dissimilar from the others. This same mismatch in distribution based on particle resolution is also seen in the height distributions for both packing densities, though it is less pronounced. While this certainly indicates that minimally-resolved simulations are not sufficient make quantitative predictions, all of the particle resolutions considered produce qualitatively similar results. More specifically, we estimate the mean velocity among all particles in μm/s for $\phi = 0.2$ to be $\bar{V} = 15.3$ for 1 blob, $\bar{V} = 12.2$ for 12 blobs, $\bar{V} = 11.4$ for 42 blobs per sphere, compared to $\bar{V} = 10.9$ predicted by "equilibrium" static sampling (with 42 blobs). For $\phi = 0.4$, we estimate $\bar{V} = 28.4$ for 1 blob, $\bar{V} = 21.3$ for 12 blobs, $\bar{V} = 19.8$ for 42 blobs per sphere, compared to $\bar{V} = 17.9$ predicted by "equilibrium" static sampling. These results also indicate that using as few as 12 blobs per sphere can give sufficiently accurate predictions (with relative error less than 10%) to be quantitatively compared to experimental measurements. Note that the mean velocity obtained from the most-resolved computations leads to an estimate of the Péclet number $\text{Pe} = \left(6\pi\eta\right) R_h^2 \bar{V} / \left(k_B T\right)$ of $\text{Pe} \approx 22$ for $\phi = 0.2$ and $\text{Pe} \approx 39$ for $\phi = 0.4$. This indicates that the horizontal motion is dominated by the active flow. However, the Brownian motion is important for setting the height of the particles above the bottom wall, as can be seen from the fact that the distribution of heights $P\left(h\right)$ is not changed significantly by the active motion.

All of the simulation results in Fig. 7 show bimodal distributions for both the particles' heights and velocities. Particularly prominent in the particle velocity distribution for $\phi = 0.4$, but present for $\phi = 0.2$ as well, are two peaks indicating the existence of two distinct populations of "fast" and "slow" particles. Close examination of the height distributions reveals a similar bimodality, and our simulations indicate a strong correlation between particle height and velocity. We separate particles into two subgroups roughly corresponding to the two peaks in $P(h)$, and identify the fast particles as the group corresponding to $h > 2R_h$, while the remaining particles are slower, as seen in the inset figures in the lower panels of Fig. 7. This separation is surprising as we might expect the opposite given that a single particle will translate faster if it is placed closer to the wall. This indicates the importance of collective flows and packing effects in these suspensions. Physically, the higher packing density causes a relatively dense monolayer of particles to form around the gravitational height, $h_G = R_h + k_B T / m_e g \approx 1 \mu\text{m}$. The rest of the particles form a sparser and more diffuse (in the vertical direction) monolayer above the first at height of roughly $2h_G$, and are rapidly advected by the collective flow as they "slide" on top of the bottom layer.

The presence of two populations of particles at different heights and moving at different velocities makes experimental measurements of $P(V)$ or even $\bar{V}$ more difficult. Namely, at these packing densities it is not possible to track individual particles to measure individual particle velocities, and indirect method such as particle image velocimetry (PIV) are used to estimate $\bar{V}$, which can lead to bias in the presence of fast and slow particles. Direct comparison of our computational estimates to experimental measurements is therefore deferred for future work.

V.   CONCLUSIONS

In this work we designed efficient and robust temporal integrators for the simulation of many rigid particles suspended in a fluctuating viscous fluid. Hydrodynamic interactions were computed using a rigid multiblob model [10] of the particles, and here we proposed a method to generate the Brownian increments of the particles at a computational cost that is no larger than that of solving a mobility problem. We demonstrated that the block-diagonal preconditioner used to solve mobility problems in [10] is equally effective as a preconditioner for the Lanczos algorithm to compute Brownian increments for the blobs. The stochastic drift term arising from the configuration-dependent mobility matrix were computed using traction-corrected or slip-corrected random finite differences. We presented a traction-corrected Euler-Maruyama scheme (EM-T) (algorithm 1), as well as a slip-corrected Trapezoidal scheme (T-S) (algorithm 2). We have made our python

implementation (with PyCUDA acceleration) of the methods described here available at `https://github.com/stochasticHydroTools/RigidMultiblobsWall`. Both the EM-T and T-S schemes scale linearly in complexity with the number of rigid particles being simulated if the iterative methods used to compute deterministic and Brownian blob velocities are based on fast methods. We confirmed that both schemes correctly reproduce the equilibrium Gibbs-Boltzmann distribution for sufficiently small time step sizes, and found the T-S scheme to be notably superior in accuracy for the same computational effort for large-scale problems. We used the T-S scheme to study the non-equilibrium dynamics of an active suspension of microrollers confined above a no-slip bottom wall, and demonstrated that as few as 12 blobs per sphere gives numerical errors on the order of 10% or less, unlike previous simulations of existing microroller experiments [5, 36]. The use of particle image velocimetry (PIV) in [4] to experimentally estimate the average particle velocity can be biased from the bimodality of the particle velocities, rendering a direct comparison with experimental results currently unobtainable. In section IV C, we identified two well separated, unimodal, populations of microrollers demarcated by their heights. This separation can perhaps be used to design new experimental techniques to accurately measure the particles' velocities within each population.

There are number of application-dependent parameters that can be tweaked to improve efficiency. For instance if small particle displacements are expected over a time step, one can wait several time steps before recomputing the (Cholesky) factorizations of $\boldsymbol{\mathcal{M}}$ that enter in the block-diagonal preconditioner. Here we used the same relative error tolerance for all iterative methods, and found a relatively loose tolerance of 10^{-3} to be sufficient. However, one could use a different solver tolerances in, for example, steps 1,2, and 3 of algorithm 1 for the EM-T scheme. Further, while the EM-T and T-S schemes were found to be optimal for the applications considered herein, other schemes such as an Adams-Bashforth variant of the EM-T scheme, or a midpoint variant of the T-S scheme are straightforward extensions and may prove optimal for other applications and implementations. In fact, our biggest contribution here is the development of the traction-corrected and slip-corrected RFDs, which can be used as tools to construct other schemes. It is important to realize that showing theoretically that a certain scheme is consistent in the limit $\Delta t \to 0$ is *not* sufficient – establishing numerically that the scheme is robust for Δt on the order of the *diffusive* time scale, as we have done here for the EM-T and T-S schemes, is crucial. While we do not have a detailed theoretical understanding of the errors that arise for finite Δt, one important consideration is that RFDs only give the stochastic drift term in expectation, and it is important to control and reduce their variance and not just their mean.

The methods presented in this work are fairly general and can be extended to other geometries and ways of computing hydrodynamic interactions. In periodic domains, we can use the Positively Split Ewald (PSE) method [16] to compute deterministic and Brownian blob velocities, and no change is made in algorithms 1 or 2 to account for this. However, in this case, generating Brownian blob displacements becomes rather inexpensive compared to a mobility solve, and it is possible that a midpoint split scheme would become preferable over the T-S scheme in terms of efficiency-accuracy balance. Note that the PSE method could be extended to other geometries such as doubly-periodic domains (e.g., membranes) by building on recently-developed Spectral Ewald methods [40, 41]. To the best of our knowledge, there is presently no known method to compute Brownian blob increments for infinite unbounded domains in (near) linear time; we relied here explicitly on the the simplicity of the Rotne-Prager-Blake tensor and the screening by the wall to handle particles confined in a half-space.

All of the computations performed in this work used a rather coarsely resolved multiblob model to represent the rigid bodies. In future work, we will apply the temporal integrators proposed here to more accurate representations of the geometry and hydrodynamics using the recently-developed Fluctuating Boundary Integral Method (FBIM) [26]. Both of the temporal integrators presented here can be used without modification with FBIM, but a midpoint scheme may be preferable because of the low-cost of computing Brownian terms compared to solving mobility problems.

All of the methods presented herein relied extensively on an explicit representation of $\mathcal{M}$, which restricts the choice of domain and boundary conditions that we may use to those for which an analytical expressions (and preferably a fast method to compute its action) for the RPY mobility is available. In [24] the Stokes equations are solved explicitly on an Eulerian grid for fully confined domains such as slit channels, and Immersed Boundary (IB) interpolation and spreading operators are used to couple the blobs to the fluid solver. This Green's-function-free or "explicit-solvent" (but still inertia-less) approach implicitly computes the action of $\mathcal{M}$ in linear time in the number of fluid grid cells. Some of us demonstrated in [24] that the action of $\mathcal{M}^{\frac{1}{2}}$ can also be computed using the IB approach at minimal additional cost by using fluctuating hydrodynamics. In [10, 25], the IB approach was extended to rigid bodies (multiblobs), but without accounting for Brownian motion. The schemes presented herein can, in principle, be used with only minor modification with the rigid-body IB method to simulate Brownian motion of rigid particles in fully confined domains, when explicit representation of $\mathcal{M}$ is not available. However, the efficiency of the methods used in this work hinged on the action of $\mathcal{M}$ being computed rapidly, and hence the temporal integrators should be modified to account for the comparatively expensive explicit-solvent IB solvers introduced

in [10]. Efficient simulation of rigid, Brownian particles in general confined domains will be the subject of future work.

Acknowledgments

Appendix

Appendix A: EULER-MARUYAMA SCHEME

In section III we noted that a simple means of computing an RFD on $\boldsymbol{\mathcal{B}} \equiv \boldsymbol{\mathcal{N}}$, is to use $\Delta\boldsymbol{P} = \Delta\boldsymbol{Q} = \sqrt{k_B T}\,\boldsymbol{W}$ in equation (17), where $\boldsymbol{W}$ is a vector drawn from the standard normal distribution. However, incrementing the translational and rotational components of the configuration $\boldsymbol{Q}$ by the same quantity may cause translation and rotation of a body by very different magnitudes. This, in turn, may result in large variance of the quantity $\partial_{\boldsymbol{Q}}\boldsymbol{\mathcal{N}} : \left[\boldsymbol{W}\boldsymbol{W}^T\right]$ (see (17)), and hence slow the convergence to $\partial_{\boldsymbol{Q}} \cdot \boldsymbol{\mathcal{N}}$ in expectation. To remedy this, we simply ensure that a given body is being randomly translated and rotated by the same amount.

Specifically, we choose a length scale, L_p, to represent the size of body p, and compute the random displacement,

$$\Delta\boldsymbol{Q}_p = \begin{bmatrix} L_p \boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix} \tag{A1}$$

as well as the random forces and torques,

$$\boldsymbol{W}_p^{FT} = k_B T \begin{bmatrix} L_p^{-1} \boldsymbol{W}_p^f \\ \boldsymbol{W}_p^\tau \end{bmatrix} \tag{A2}$$

where $\boldsymbol{W}_p^f, \boldsymbol{W}_p^\tau$ are both three dimensional standard Gaussian random vectors, and we form the composite vectors $\boldsymbol{W}^{FT} = \left[\boldsymbol{W}_p^{FT}\right]$ and $\Delta\boldsymbol{Q} = \left[\Delta\boldsymbol{Q}_p\right]$. The drift can then be computed using a simple RFD,

$$\frac{1}{\delta} \left\langle \left\{ \boldsymbol{\mathcal{N}}\left(\boldsymbol{Q} + \frac{\delta}{2}\Delta\boldsymbol{Q}\right) - \boldsymbol{\mathcal{N}}\left(\boldsymbol{Q} - \frac{\delta}{2}\Delta\boldsymbol{Q}\right) \right\} \boldsymbol{W}^{FT} \right\rangle = k_B T\,\partial_{\boldsymbol{Q}} \cdot \boldsymbol{\mathcal{N}} + O(\delta^2), \tag{A3}$$

which respects the physical units of the problem and minimizes the variance of the approximation. A one-sided difference approximation can be defined analogously.

Algorithm 3 summarizes a scalable implementation of the Euler-Maruyama RFD scheme from [11] to solve (1), using a random finite difference similar to (A3) to compute the drift. Implementation of this scheme requires three solutions of the saddle point system (6) and one Lanczos application per timestep.

Appendix B: PROOFS OF CONSISTENCY OF TEMPORAL INTEGRATORS

In this appendix we demonstrate that the two temporal integrators presented in this work generate the required stochastic drift terms. In all of the calculations of this section, we will

---

Algorithm 3 Euler-Maruyama-RFD scheme

1. Compute RFD terms:

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

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

    (b) Displace the particles by small random amounts:

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

    (c) Solve *two* mobility problems for $\boldsymbol{U}^+, \boldsymbol{U}^-$:

$$\begin{bmatrix} \boldsymbol{\mathcal{M}}^{\pm} & -\boldsymbol{\mathcal{K}}^{\pm} \\ -\left(\boldsymbol{\mathcal{K}}^T\right)^{\pm} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{\pm} \\ \boldsymbol{U}^{\pm} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{W}^{FT} \end{bmatrix}.$$

2. Compute $\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n$ using a preconditioned Lancoz method or PSE.

3. Solve mobility problem:

$$\begin{bmatrix} \boldsymbol{\mathcal{M}}^n & -\boldsymbol{\mathcal{K}}^n \\ -\left(\boldsymbol{\mathcal{K}}^T\right)^n & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^n \\ \boldsymbol{U}^n \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{2k_B T}{\Delta t}}\left(\boldsymbol{\mathcal{M}}^{1/2}\right)^n \boldsymbol{W}^n \\ -\boldsymbol{F}^n \end{bmatrix}.$$

4. Update configuration:

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n + \Delta t \left\{\boldsymbol{U}^n + \frac{1}{\delta}(\boldsymbol{U}^+ - \boldsymbol{U}^-)\right\}.$$

---

use index notation with the convention of summing over repeated indices. We will also use the convention that superscripted indices correspond to a matrix inverse, i.e $\left[\boldsymbol{\mathcal{A}}^{-1}\right]_{ij} \equiv \boldsymbol{\mathcal{A}}^{ij}$. To avoid possible confusion with our superscript notation for the time level at which an operator is evaluated, we will assume that all terms and operators are evaluated at the base configuration $\boldsymbol{Q} \equiv \boldsymbol{Q}^n$ unless otherwise noted. We denote partial derivatives with the shorthand notation $\partial_k \equiv \partial/\partial Q_k$.

1. Overdamped Langevin Equation using Quaternions

In principle, any means of representing orientation could be used with the techniques detailed in this work. In previous work [11], unit quaternions were found to be a particularly favorable choice, and we will use normalized quaternions to represent the orientation $\boldsymbol{\theta}$ hereafter. Here we briefly review key notation and results regarding quaternions; details can be found in [11]. A normalized

quaternion is a vector $\boldsymbol{\theta} = [s, \boldsymbol{p}] \in \mathbb{R}^4$ such that $\|\boldsymbol{\theta}\|_2 = 1$. We define an orientation dependent 4×3 "projection" matrix $\boldsymbol{\Psi}$ as

$$\boldsymbol{\Psi}(\boldsymbol{\theta}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{p}^T \\ s\boldsymbol{I} - \boldsymbol{P} \end{bmatrix}, \tag{B1}$$

where $\boldsymbol{P}$ is the cross product matrix defined by $\boldsymbol{Px} = \boldsymbol{p} \times \boldsymbol{x}$. For a configuration, $\boldsymbol{Q} = [\boldsymbol{q}, \boldsymbol{\theta}]$ and a rigid body velocity $\boldsymbol{U} = [\boldsymbol{u}, \boldsymbol{\omega}]$, we introduce a matrix

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi} \end{bmatrix}, \tag{B2}$$

so that we may write $d\boldsymbol{Q}/dt = \boldsymbol{\Xi U}$. In [11] some of us showed that the overdamped Langevin Ito equation (1) may be written using unit quaternions as

$$\begin{aligned} \frac{d\boldsymbol{Q}_l}{dt} &= \boldsymbol{\Xi}_{li}\boldsymbol{\mathcal{N}}_{ij}\boldsymbol{F}_j + k_B T \ (\boldsymbol{\Xi}_{li}\{\partial_k\boldsymbol{\mathcal{N}}_{ij}\}\boldsymbol{\Xi}_{kj} + \{\partial_k\boldsymbol{\Xi}_{li}\}\boldsymbol{\mathcal{N}}_{ij}\boldsymbol{\Xi}_{kj}) + \sqrt{2k_B T}\ \boldsymbol{\Xi}_{li}\boldsymbol{\mathcal{N}}_{it}^{1/2}\boldsymbol{\mathcal{W}}_t \quad \text{(B3)} \\ &= \boldsymbol{\Xi}_{li} \circ \left(\boldsymbol{\mathcal{N}}_{ij}\boldsymbol{F}_j + k_B T \ \{\partial_k\boldsymbol{\mathcal{N}}_{ij}\}\boldsymbol{\Xi}_{kj} + \sqrt{2k_B T}\ \boldsymbol{\mathcal{N}}_{it}^{1/2}\boldsymbol{\mathcal{W}}_t\right) \\ &= \boldsymbol{\Xi}_{li} \circ \boldsymbol{U}_i, \end{aligned}$$

where $\circ$ denotes the Stratonovich product. As shown in [11], the forces and torques of the system may be written as $\boldsymbol{F}_j = -\boldsymbol{\Xi}_{kj}\partial_k U$, where $U(\boldsymbol{Q})$ is the conservative potential. This, combined with the fact that $\partial_k\boldsymbol{\Xi}_{kj} = 0$, allows us to recognize (B3) as having the same form as equation (1) but with $\boldsymbol{\mathcal{N}}$ replaced by $\boldsymbol{\Xi}\boldsymbol{\mathcal{N}}\boldsymbol{\Xi}^T$.

In order to discretize (B3), we write an expansion in Δt for the procedure which rotates a quaternion $\boldsymbol{\theta}$ by an angular displacement $\boldsymbol{\omega}\Delta t$ as

$$\text{Rotate}(\boldsymbol{\theta}_k, \boldsymbol{\omega}_k\Delta t) \approx \boldsymbol{\theta}_k + \Delta t\boldsymbol{\Psi}_{kj}\boldsymbol{\omega}_j - \frac{\Delta t^2}{8}\|\boldsymbol{\omega}\|_2^2\ \boldsymbol{\theta}_k. \tag{B4}$$

The second order term in (B4) is responsible for the fact that that a simple (inconsistent) Euler-Maruyama scheme without an RFD, using a velocity

$$\boldsymbol{U}_i = \boldsymbol{\mathcal{N}}_{ij}\boldsymbol{F}_j + \sqrt{\frac{2k_B T}{\Delta t}}\boldsymbol{\mathcal{N}}_{it}^{1/2}\boldsymbol{W}_t^n \tag{B5}$$

to update the configurations to the next time step, will generate the drift term $(k_B T)\{\partial_k\boldsymbol{\Xi}_{li}\}\boldsymbol{\mathcal{N}}_{ij}\boldsymbol{\Xi}_{kj}$ in (B3) to leading order in Δt [11]. This is to say that the rotate procedure (B4) naturally captures the Stratonovich product in the second line of (B3). A one-step numerical scheme is first-order weakly accurate if the first three moments of the numerical update are correct to $O(\Delta t)$ [42]; the third moment in our schemes is easily seen to be at least $O(\Delta t^{3/2})$.

Hence, to show that the schemes introduced in this work are weakly first order in solving (B3), we will show that velocity used to update the configurations to the next time step is of the form

$$U_i = \mathcal{N}_{ij} F_j + \sqrt{\frac{2k_B T}{\Delta t}} \mathcal{N}_{it}^{1/2} W_t^n + k_B T \ \{\partial_k \mathcal{N}_{ij}\} \Xi_{kj} + \mathcal{R}\left(\Delta t, \ \Delta t^{1/2}\right),$$ (B6)

where the notation $\mathcal{R}\left(a, b^{1/2}\right)$ denotes a Gaussian random error term with mean $O(a)$, and variance $O(b)$. Notice that the leading order term in (B6), the Brownian velocity $\sqrt{\frac{2k_B T}{\Delta t}} \mathcal{N}_{it}^{1/2} W_t^n$, controls the second moment of the velocity update. This term is easily identified in the velocity update produced by the schemes introduced in this work. The main difficulty is showing that both schemes generate in expectation value the required drift term $k_B T \ \{\partial_k \mathcal{N}_{ij}\} \Xi_{kj}$ for sufficiently small Δt. For this, we will use the chain rule to expand the derivative,

$$
\begin{aligned}
\partial_k \mathcal{N}_{ij} = & -\mathcal{N}_{im} \left\{\partial_k \mathcal{N}^{ml}\right\} \mathcal{N}_{lj} = -\mathcal{N}_{im} \left(\partial_k \left\{\mathcal{K}_{pm} \mathcal{M}^{pq} \mathcal{K}_{ql}\right\}\right) \mathcal{N}_{lj} \\
= & -\mathcal{N}_{im} \left\{\partial_k \mathcal{K}_{pm}\right\} \mathcal{M}^{pq} \mathcal{K}_{ql} \mathcal{N}_{lj} - \mathcal{N}_{im} \mathcal{K}_{pm} \left\{\partial_k \mathcal{M}^{pq}\right\} \mathcal{K}_{ql} \mathcal{N}_{lj} - \mathcal{N}_{im} \mathcal{K}_{pm} \mathcal{M}^{pq} \left\{\partial_k \mathcal{K}_{ql}\right\} \mathcal{N}_{lj} \\
= & -\mathcal{N}_{im} \left\{\partial_k \mathcal{K}_{sm}\right\} \mathcal{M}^{sq} \mathcal{K}_{ql} \mathcal{N}_{lj} \\
& + \mathcal{N}_{im} \mathcal{K}_{pm} \mathcal{M}^{pr} \left\{\partial_k \mathcal{M}_{rs}\right\} \mathcal{M}^{sq} \mathcal{K}_{ql} \mathcal{N}_{lj} \\
& - \mathcal{N}_{im} \mathcal{K}_{pm} \mathcal{M}^{pr} \left\{\partial_k \mathcal{K}_{rl}\right\} \mathcal{N}_{lj}.
\end{aligned}
$$ (B7)

Note that the expression for $(\partial_Q \cdot \mathcal{N})_i$ can be obtained by replacing every instance of the index k with j in the above. For clarity but without loss of generality, we take $F = 0$ hereafter. Note that a nonzero value of F may affect the variance of a velocity update and thus the accuracy of the scheme for finite values of Δt. Still, the weak *order* of accuracy of the schemes considered will be maintained for nonzero F.

2. Traction-Corrected Euler-Maruyama Scheme

It is immediate from the discussion in section II B, that the Brownian velocities produced by Algorithm 1 are $\sqrt{\frac{2k_B T}{\Delta t}} \mathcal{N}_{ij}^{1/2} W_j^n$ to leading order in Δt, and hence produce the correct second moment. We will now show that the velocity produced in step 3 of Algorithm 1 is such that

$$\langle U_i^n \rangle = k_B T \ \{\partial_k \mathcal{N}_{ij}\} \Xi_{kj} + O\left(\delta^2\right).$$ (B8)

First, we may write step 3 as

$$
\begin{aligned}
U_i^n = & -\mathcal{N}_{im} D_m^F + \mathcal{N}_{im} \mathcal{K}_{pm} \mathcal{M}^{pr} \left(D_r^S + \sqrt{\frac{2k_B T}{\Delta t}} \mathcal{M}_{rj}^{1/2} W_j^n\right) \\
= & -\mathcal{N}_{im} D_m^F + \mathcal{N}_{im} \mathcal{K}_{pm} \mathcal{M}^{pr} D_r^S + \sqrt{\frac{2k_B T}{\Delta t}} \mathcal{N}_{ij}^{1/2} W_j^n
\end{aligned}
$$

and thus

$$\langle \boldsymbol{U}_i^n \rangle = -\boldsymbol{\mathcal{N}}_{im} \langle \boldsymbol{D}_m^F \rangle + \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \langle \boldsymbol{D}_r^S \rangle . \tag{B9}$$

We can expand the quantities $\boldsymbol{D}_m^F$ and $\boldsymbol{D}_r^S$, defined in step 1d, using the definition of $\boldsymbol{\lambda}^{\mathrm{RFD}}$ from step 1b,

$$\boldsymbol{U}_l^{\mathrm{RFD}} = \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{W}_j^{FT}$$

$$\boldsymbol{\lambda}_s^{\mathrm{RFD}} = \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{U}_l^{\mathrm{RFD}} = \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{W}_j^{FT} .$$

Beginning with $\boldsymbol{D}_m^F$,

$$\langle \boldsymbol{D}_m^F \rangle = \frac{1}{\delta} \left\langle \{ \boldsymbol{\mathcal{K}}_{sm} \left(\boldsymbol{Q}^+ \right) - \boldsymbol{\mathcal{K}}_{sm} \left(\boldsymbol{Q}^- \right) \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{W}_j^{FT} \right\rangle \tag{B10}$$

$$= \{ \partial_k \boldsymbol{\mathcal{K}}_{sm} \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \langle \Delta \boldsymbol{Q}_k \, \boldsymbol{W}_j^{FT} \rangle + O(\delta^2) . \tag{B11}$$

Using the expansion for the rotation procedure from (B4), we may write $\Delta \boldsymbol{Q}_k = [\Delta \boldsymbol{q}_k, \Delta \theta_k]$ as

$$\Delta \boldsymbol{q}_k = \frac{\delta}{2} L_p \boldsymbol{W}_k^f$$

$$\Delta \boldsymbol{\theta}_k = \frac{\delta}{2} \boldsymbol{\Psi}_{kl} \boldsymbol{W}_l^\tau + \mathcal{R} \left(\delta^2, \delta^2 \right) ,$$

from which it is straightforward to verify that

$$\left\langle \Delta \boldsymbol{Q}_k \, \boldsymbol{W}_j^{FT} \right\rangle = k_B T \, \boldsymbol{\Xi}_{kj} + O(\delta^2) . \tag{B12}$$

Using equations (B11) and (B12), we see that the terms produced by $\boldsymbol{D}^F$ give one part of the required stochastic drift term,

$$\left\langle \boldsymbol{D}_m^F \right\rangle = k_B T \, \{ \partial_k \boldsymbol{\mathcal{K}}_{sm} \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj} + O(\delta^2) . \tag{B13}$$

Now, for the contributions from terms involving $\boldsymbol{D}_r^S$, we have

$$\left\langle \boldsymbol{D}_r^S \right\rangle = \frac{1}{\delta} \left\langle \{ \boldsymbol{\mathcal{M}}_{rs} \left(\boldsymbol{Q}^+ \right) - \boldsymbol{\mathcal{M}}_{rs} \left(\boldsymbol{Q}^- \right) \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{W}_j^{FT} \right\rangle$$

$$- \frac{1}{\delta} \left\langle \{ \boldsymbol{\mathcal{K}}_{rl} \left(\boldsymbol{Q}^+ \right) - \boldsymbol{\mathcal{K}}_{rl} \left(\boldsymbol{Q}^- \right) \} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{W}_j^{FT} \right\rangle$$

$$= (\{ \partial_k \boldsymbol{\mathcal{M}}_{rs} \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} - \{ \partial_k \boldsymbol{\mathcal{K}}_{rl} \}) \boldsymbol{\mathcal{N}}_{lj} \left\langle \Delta \boldsymbol{Q}_k \, \boldsymbol{W}_j^{FT} \right\rangle + O(\delta^2) , \tag{B14}$$

and hence, using equations (B14) and (B12), we see that from $\boldsymbol{D}^S$ we get two more pieces of the required stochastic drift term,

$$\left\langle \boldsymbol{D}_r^S \right\rangle = k_B T \, (\{ \partial_k \boldsymbol{\mathcal{M}}_{rs} \} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} - \{ \partial_k \boldsymbol{\mathcal{K}}_{rl} \}) \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj} + O(\delta^2) . \tag{B15}$$

Plugging the results from equations (B13) and (B15) into equation (B9), and using (B7), gives

$$\langle \boldsymbol{U}_i^n \rangle = -\,\boldsymbol{\mathcal{N}}_{im} \langle \boldsymbol{D}_m^F \rangle + \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \langle \boldsymbol{D}_r^S \rangle + O(\delta^2) \tag{B16}$$

$$= -\,k_B T \,\left(\boldsymbol{\mathcal{N}}_{im} \left\{ \partial_k \boldsymbol{\mathcal{K}}_{sm} \right\} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj} \right.$$

$$+\, \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \left\{ \partial_k \boldsymbol{\mathcal{M}}_{rs} \right\} \boldsymbol{\mathcal{M}}^{sq} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj}$$

$$\left. -\, \boldsymbol{\mathcal{N}} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \left\{ \partial_k \boldsymbol{\mathcal{K}}_{rl} \right\} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj} \right) + O(\delta^2)$$

$$= k_B T \, \left\{ \partial_k \boldsymbol{\mathcal{N}}_{ij} \right\} \boldsymbol{\Xi}_{kj} + O(\delta^2),$$

where the last equality comes directly from equation (B7). This is the desired result (B8).

3. Slip-Corrected Trapezoidal Scheme

We must show that the velocity used to update the position in the corrector step in Algorithm 2,

$$\boldsymbol{U}_i^{(c)} = \frac{1}{2} \left(\widetilde{\boldsymbol{U}}_i + \boldsymbol{U}_i^n \right), \tag{B17}$$

satisfies (B6) in law to leading order. We first show that the predictor-corrector update already gives part of the required drift term. For this we set $\boldsymbol{D}^F = \boldsymbol{D}^S = \boldsymbol{0}$ (recall that we take $\boldsymbol{F} = 0$ for simplicity), and show that to leading order in Δt, the velocity update in step 6 of Algorithm 2 is such that [49]

$$\left\langle \boldsymbol{U}_i^{(c)} \right\rangle = -k_B T \, \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \left\{ \partial_k \boldsymbol{\mathcal{K}}_{rl} \right\} \boldsymbol{\mathcal{N}}_{lj} \boldsymbol{\Xi}_{kj} + O \left(\Delta t \right). \tag{B18}$$

To show this we first write steps 3 and 5 as

$$\boldsymbol{U}_j^n = \sqrt{\frac{2 k_B T}{\Delta t}} \, \boldsymbol{\mathcal{N}}_{jl} \boldsymbol{\mathcal{K}}_{ql} \boldsymbol{\mathcal{M}}^{qs} \boldsymbol{\mathcal{M}}_{su}^{1/2} \, \boldsymbol{W}_u^n, \tag{B19}$$

$$\widetilde{\boldsymbol{U}}_i = \sqrt{\frac{2 k_B T}{\Delta t}} \, \widetilde{\boldsymbol{\mathcal{N}}}_{im} \widetilde{\boldsymbol{\mathcal{K}}}_{pm} \widetilde{\boldsymbol{\mathcal{M}}}^{pr} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n, \tag{B20}$$

and write the velocity update of step 6 as

$$\sqrt{\frac{2 \Delta t}{k_B T}} \boldsymbol{U}_i^{(c)} = \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n + \widetilde{\boldsymbol{\mathcal{N}}}_{im} \widetilde{\boldsymbol{\mathcal{K}}}_{pm} \widetilde{\boldsymbol{\mathcal{M}}}^{pr} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n$$

$$= \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n$$

$$+ \left(\boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} + \partial_k \left\{ \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \right\} \Delta \widetilde{\boldsymbol{Q}}_k + O \left(\Delta \widetilde{\boldsymbol{Q}}^2 \right) \right) \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n \tag{B21}$$

$$= 2 \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \boldsymbol{W}_t^n + \partial_k \left\{ \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \right\} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \, \Delta \widetilde{\boldsymbol{Q}}_k \boldsymbol{W}_t^n + \mathcal{R} \left(\Delta t^{3/2}, \Delta t \right)$$

$$= 2 \boldsymbol{\mathcal{N}}_{it}^{1/2} \boldsymbol{W}_t^n + \partial_k \left\{ \boldsymbol{\mathcal{N}}_{im} \boldsymbol{\mathcal{K}}_{pm} \boldsymbol{\mathcal{M}}^{pr} \right\} \boldsymbol{\mathcal{M}}_{rt}^{1/2} \, \Delta \widetilde{\boldsymbol{Q}}_k \boldsymbol{W}_t^n + \mathcal{R} \left(\Delta t^{3/2}, \Delta t \right). \tag{B22}$$

In equation (B21), we have used the fact that all operators with tilde are evaluated at $\widetilde{\boldsymbol{Q}}$, with $\Delta\widetilde{\boldsymbol{Q}} = \widetilde{\boldsymbol{Q}} - \boldsymbol{Q}^n$.

All that remains is to compute,

$$\left\langle \boldsymbol{U}_i^{(c)} \right\rangle = \sqrt{\frac{k_B T}{2\Delta t}}\, \partial_k \{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\}\, \boldsymbol{\mathcal{M}}_{rt}^{1/2} \left\langle \Delta\widetilde{\boldsymbol{Q}}_k \boldsymbol{W}_t^n \right\rangle + O\left(\Delta t\right). \tag{B23}$$

Using the expansion for the rotate procedure (B4), we may write

$$\Delta\tilde{\boldsymbol{Q}}_k = \Delta t \boldsymbol{\Xi}_{kj}\boldsymbol{U}_j^n + \mathcal{R}\left(\Delta t^2, \Delta t^2\right),$$

which combined with (B19) gives

$$\left\langle \Delta\widetilde{\boldsymbol{Q}}_k \boldsymbol{W}_t^n \right\rangle = \sqrt{2\Delta t k_B T}\, \boldsymbol{\Xi}_{kj}\boldsymbol{\mathcal{N}}_{jl}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{M}}^{qs}\boldsymbol{\mathcal{M}}_{su}^{1/2} \langle \boldsymbol{W}_u^n \boldsymbol{W}_t^n \rangle + O(\Delta t^2) \tag{B24}$$

$$= \sqrt{2\Delta t k_B T}\, \boldsymbol{\Xi}_{kj}\boldsymbol{\mathcal{N}}_{jl}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{M}}^{qs}\boldsymbol{\mathcal{M}}_{st}^{1/2} + O(\Delta t^2),$$

where we used $\langle \boldsymbol{W}_u^n \boldsymbol{W}_t^n \rangle = \delta_{tu}$. Hence, equation (B23) becomes,

$$\left\langle \boldsymbol{U}_i^{(c)} \right\rangle = k_B T\, \partial_k \{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\}\, \boldsymbol{\mathcal{M}}_{rt}^{1/2}\boldsymbol{\mathcal{M}}_{ts}^{1/2}\boldsymbol{\mathcal{M}}^{sq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right) \tag{B25}$$

$$= k_B T\, \partial_k \{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\}\, \boldsymbol{\mathcal{M}}_{rs}\boldsymbol{\mathcal{M}}^{sq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right)$$

$$= k_B T\, \partial_k \{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\}\, \boldsymbol{\mathcal{K}}_{rl}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right)$$

where we used that $\boldsymbol{\mathcal{M}}_{rs}\boldsymbol{\mathcal{M}}^{sq} = \delta_{rq}$. After expanding $\partial_k \{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\}$ using the chain rule and recalling that $\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\boldsymbol{\mathcal{K}}_{rl} = \boldsymbol{\mathcal{N}}^{ml}$, we get

$$\left\langle \boldsymbol{U}_i^{(c)} \right\rangle = k_B T\, \left(\{\partial_k \boldsymbol{\mathcal{N}}_{im}\} \boldsymbol{\Xi}_{km} \right. \tag{B26}$$

$$+ \boldsymbol{\mathcal{N}}_{im} \{\partial_k \boldsymbol{\mathcal{K}}_{pm}\} \boldsymbol{\mathcal{M}}^{pr}\boldsymbol{\mathcal{K}}_{rl}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj}$$

$$+ \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm} \{\partial_k \boldsymbol{\mathcal{M}}^{pr}\} \boldsymbol{\mathcal{K}}_{rl}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} \left.\right) + O\left(\Delta t\right)$$

$$= -k_B T\, \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr} \{\partial_k \boldsymbol{\mathcal{K}}_{rl}\} \boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right). \tag{B27}$$

where we have used (B7). This gives the desired result (B18).

We now include the contribution from nonzero $\boldsymbol{D}^F$ and $\boldsymbol{D}^S$ to show that $\boldsymbol{U}_i^{(c)}$ satisfies equation (B6) as desired. Including the contributions from $\boldsymbol{D}^F$ and $\boldsymbol{D}^S$, we may write the velocity update of step 6 as,

$$\boldsymbol{U}_i^{(c)} = -\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F + \widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S$$

$$+ \sqrt{\frac{k_B T}{2\Delta t}}\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\boldsymbol{\mathcal{M}}_{rt}^{1/2}\boldsymbol{W}_t^n + \sqrt{\frac{k_B T}{2\Delta t}}\widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{\mathcal{M}}_{rt}^{1/2}\boldsymbol{W}_t^n. \tag{B28}$$

Using equations (B22) and (B24), we may write this as

$$\boldsymbol{U}_i^{(c)} = -\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F + \widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S + \sqrt{\frac{2k_BT}{\Delta t}}\boldsymbol{\mathcal{N}}_{it}^{1/2}\boldsymbol{W}_t^n \tag{B29}$$
$$+ \partial_k\left\{\boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\right\}\boldsymbol{\mathcal{M}}_{rt}^{1/2}\left(\boldsymbol{\Xi}_{kj}\boldsymbol{\mathcal{N}}_{jl}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{M}}^{qs}\boldsymbol{\mathcal{M}}_{su}^{1/2}\boldsymbol{W}_u^n\boldsymbol{W}_t^n\right) + \mathcal{R}\left(\Delta t, \Delta t^{1/2}\right).$$

In this form, it is easy to see that the scheme produces the correct Brownian velocity, $\sqrt{\frac{2k_BT}{\Delta t}}\boldsymbol{\mathcal{N}}_{it}^{1/2}\boldsymbol{W}_t^n$, and all that is left to verify is the first moment. Equation (B18) allows us to immediately write the mean of (B29) as

$$\left\langle\boldsymbol{U}_i^{(c)}\right\rangle = \left\langle-\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F + \widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S\right\rangle \tag{B30}$$
$$- k_BT\ \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\left\{\partial_k\boldsymbol{\mathcal{K}}_{rl}\right\}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right).$$

We now examine the terms involving $\boldsymbol{D}_m^F$ and $\boldsymbol{D}_r^S$, separately. First, we note that using the expanded rotate procedure (B4), steps 1c and 1b in algorithm 2 give

$$\boldsymbol{Q}_k^{\pm} - \boldsymbol{Q}_k = \pm\delta\ \boldsymbol{\Xi}_{kj}\Delta\boldsymbol{Q}_k^{\mathrm{RFD}} + \mathcal{R}\left(\delta^2, \delta^2\right) = \pm\delta\ \boldsymbol{\Xi}_{kj}\boldsymbol{\mathcal{N}}_{jl}\boldsymbol{\mathcal{K}}_{lq}\boldsymbol{\mathcal{M}}^{qs}\breve{\boldsymbol{W}}_s^D + \mathcal{R}\left(\delta^2, \delta^2\right), \tag{B31}$$

Using the definition of $\boldsymbol{D}_m^F$ from step 1d of Algorithm 2,

$$-\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F = -\frac{1}{\delta}\ \widetilde{\boldsymbol{\mathcal{N}}}_{im}\left\{\boldsymbol{\mathcal{K}}_{pm}\left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{K}}_{pm}\left(\boldsymbol{Q}^-\right)\right\}\breve{\boldsymbol{W}}_p^F$$
$$= -\widetilde{\boldsymbol{\mathcal{N}}}_{im}\left\{\partial_k\boldsymbol{\mathcal{K}}_{pm}\right\}\ \boldsymbol{\Xi}_{kj}\Delta\boldsymbol{Q}_k^{RFD}\breve{\boldsymbol{W}}_p^F + \mathcal{R}\left(\delta^2, \delta^2\right)$$
$$= -\boldsymbol{\mathcal{N}}_{im}\left\{\partial_k\boldsymbol{\mathcal{K}}_{pm}\right\}\ \boldsymbol{\Xi}_{kj}\boldsymbol{\mathcal{N}}_{jl}\boldsymbol{\mathcal{K}}_{lq}\boldsymbol{\mathcal{M}}^{qs}\breve{\boldsymbol{W}}_s^D\breve{\boldsymbol{W}}_p^F + \mathcal{R}\left(\Delta t, \Delta t^{1/2}\right) + \mathcal{R}\left(\delta^2, \delta^2\right), \tag{B32}$$

where we used (B31) in (B32). In expectation, we get the drift term

$$\left\langle-\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F\right\rangle = -\boldsymbol{\mathcal{N}}_{im}\left\{\partial_k\boldsymbol{\mathcal{K}}_{pm}\right\}\left\langle\breve{\boldsymbol{W}}_s^D\breve{\boldsymbol{W}}_p^F\right\rangle\boldsymbol{\mathcal{M}}^{sq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\delta^2\right) + O\left(\Delta t\right)$$
$$= -k_BT\ \boldsymbol{\mathcal{N}}_{im}\left\{\partial_k\boldsymbol{\mathcal{K}}_{pm}\right\}\boldsymbol{\mathcal{M}}^{pq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\delta^2\right) + O\left(\Delta t\right), \tag{B33}$$

where we have used that $\left\langle\breve{\boldsymbol{W}}_s^D\breve{\boldsymbol{W}}_p^F\right\rangle = k_BT\delta_{sp}$.

Similarly, using the definition of $\boldsymbol{D}_r^S$ from step 1d of Algorithm 2,

$$\widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S = \frac{1}{\delta}\ \widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\left\{\boldsymbol{\mathcal{M}}_{rt}\left(\boldsymbol{Q}^+\right) - \boldsymbol{\mathcal{M}}_{rt}\left(\boldsymbol{Q}^-\right)\right\}\breve{\boldsymbol{W}}_t^F \tag{B34}$$
$$= \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\left\{\partial_k\boldsymbol{\mathcal{M}}_{rt}\right\}\ \boldsymbol{\Xi}_{kj}\Delta\boldsymbol{Q}_k^{RFD}\breve{\boldsymbol{W}}_t^F + \mathcal{R}\left(\Delta t, \Delta t^{1/2}\right) + \mathcal{R}\left(\delta^2, \delta^2\right).$$

Hence, taking the mean of (B34), gives

$$\left\langle\widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S\right\rangle = \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\left\{\partial_k\boldsymbol{\mathcal{M}}_{rt}\right\}\boldsymbol{\Xi}_{kj}\left\langle\Delta\boldsymbol{Q}_k^{RFD}\breve{\boldsymbol{W}}_t^F\right\rangle + O\left(\Delta t\right) + O(\delta^2)$$
$$= \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\left\{\partial_k\boldsymbol{\mathcal{M}}_{rt}\right\}\left\langle\breve{\boldsymbol{W}}_s^D\breve{\boldsymbol{W}}_t^F\right\rangle\boldsymbol{\mathcal{M}}^{sq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right) + O(\delta^2)$$
$$= k_BT\ \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\left\{\partial_k\boldsymbol{\mathcal{M}}_{rs}\right\}\boldsymbol{\mathcal{M}}^{sq}\boldsymbol{\mathcal{K}}_{ql}\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj} + O\left(\Delta t\right) + O(\delta^2). \tag{B35}$$

Combining equations (B7), (B35) and (B33), we may write

$$
\left\langle -\widetilde{\boldsymbol{\mathcal{N}}}_{im}\boldsymbol{D}_m^F + \widetilde{\boldsymbol{\mathcal{N}}}_{im}\widetilde{\boldsymbol{\mathcal{K}}}_{pm}\widetilde{\boldsymbol{\mathcal{M}}}^{pr}\boldsymbol{D}_r^S \right\rangle = k_B T \ \left(\{\partial_k\boldsymbol{\mathcal{N}}_{ij}\}\,\boldsymbol{\Xi}_{kj} + \boldsymbol{\mathcal{N}}_{im}\boldsymbol{\mathcal{K}}_{pm}\boldsymbol{\mathcal{M}}^{pr}\{\partial_k\boldsymbol{\mathcal{K}}_{rl}\}\,\boldsymbol{\mathcal{N}}_{lj}\boldsymbol{\Xi}_{kj}\right)
$$
$$
+ O\left(\Delta t\right) + O(\delta^2). \tag{B36}
$$

Combining this with (B30) and using equation (B7) gives the desired result (B6).

---

[1] R. M. Jendrejack, J. J. de Pablo, and M. D. Graham. Stochastic simulations of DNA in flow: Dynamics and the effects of hydrodynamic interactions. *J. Chem. Phys.*, 116(17):7752–7759, 2002.

[2] W. B. Russel. Brownian Motion of Small Particles Suspended in Liquids. *Ann. Rev. Fluid Mech.*, 13:425–455, 1981.

[3] Raymond Kapral. Perspective: Nanomotors without moving parts that propel themselves in solution. *J. Chem. Phys.*, 138:020901, 2013.

[4] Michelle Driscoll, Blaise Delmotte, Mena Youssef, Stefano Sacanna, Aleksandar Donev, and Paul Chaikin. Unstable fronts and motile structures formed by microrollers. *Nature Physics*, 13:375–379, 2017.

[5] Florencio Balboa Usabiaga, Blaise Delmotte, and Aleksandar Donev. Brownian dynamics of confined suspensions of active microrollers. *J. Chem. Phys.*, 146(13):134104, 2017. Software available at `https://github.com/stochasticHydroTools/RigidMultiblobsWall`.

[6] Adolfo J Banchio and John F Brady. Accelerated stokesian dynamics: Brownian motion. *The Journal of chemical physics*, 118:10323, 2003.

[7] Blaise Delmotte and Eric E Keaveny. Simulating brownian suspensions with fluctuating hydrodynamics. *The Journal of chemical physics*, 143(24):244109, 2015.

[8] Rajesh Singh and R. Adhikari. Fluctuating hydrodynamics and the brownian motion of an active colloid near a wall. *European Journal of Computational Mechanics*, 26(1-2):78–97, 2017.

[9] Brendan D Hoffman and Eric SG Shaqfeh. The effect of brownian motion on the stability of sedimenting suspensions of polarizable rods in an electric field. *Journal of Fluid Mechanics*, 624:361–388, 2009.

[10] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A. P. S. Bhalla, B. E. Griffith, and A. Donev. Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach. *Communications in Applied Mathematics and Computational Science*, 11(2):217–296, 2016. Software available at `https://github.com/stochasticHydroTools/RigidMultiblobsWall`.

[11] S. Delong, F. Balboa Usabiaga, and A. Donev. Brownian dynamics of confined rigid bodies. *J. Chem. Phys.*, 143(14):144107, 2015. Software available at `https://github.com/stochasticHydroTools/RigidMultiblobsWall`.

[12] Constantine Pozrikidis. *Boundary integral and singularity methods for linearized viscous flow*. Cambridge University Press, 1992.

[13] Ludvig Af Klinteberg and Anna-Karin Tornberg. Fast ewald summation for stokesian particle suspensions. *International Journal for Numerical Methods in Fluids*, 76(10):669–698, 2014.

[14] Ludvig af Klinteberg and Anna-Karin Tornberg. A fast integral equation method for solid particles in viscous flow using quadrature by expansion. *Journal of Computational Physics*, 326:420–445, 2016.

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

[16] A. M. Fiore, F. Balboa Usabiaga, A. Donev, and J. W. Swan. Rapid sampling of stochastic displacements in brownian dynamics simulations. *J. Chem. Phys.*, 146(12):124116, 2017. Software available at `https://github.com/stochasticHydroTools/PSE`.

[17] M. Hütter and H.C. Öttinger. Fluctuation-dissipation theorem, kinetic stochastic integral and efficient simulations. *J. Chem. Soc., Faraday Trans.*, 94(10):1403–1405, 1998.

[18] James W Swan, John F Brady, Rachel S Moore, et al. Modeling hydrodynamic self-propulsion with Stokesian Dynamics. Or teaching Stokesian Dynamics to swim. *Physics of Fluids*, 23:071901, 2011.

[19] Rajesh Singh, Somdeb Ghose, and R Adhikari. Many-body microhydrodynamics of colloidal particles with active boundary layers. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(6):P06017, 2015.

[20] Jens Rotne and Stephen Prager. Variational treatment of hydrodynamic interaction in polymers. *The Journal of Chemical Physics*, 50:4831, 1969.

[21] Eligiusz Wajnryb, Krzysztof A Mizerski, Pawel J Zuk, and Piotr Szymczak. Generalization of the rotne–prager–yamakawa mobility and shear disturbance tensors. *Journal of Fluid Mechanics*, 731:R3, 2013.

[22] Zhi Liang, Zydrunas Gimbutas, Leslie Greengard, Jingfang Huang, and Shidong Jiang. A fast multipole method for the rotne–prager–yamakawa tensor and its applications. *Journal of Computational Physics*, 234:133–139, 2013.

[23] James W. Swan and John F. Brady. Simulation of hydrodynamically interacting particles near a no-slip boundary. *Physics of Fluids*, 19(11):113306, 2007.

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

[25] B. Kallemov, A. Pal Singh Bhalla, B. E. Griffith, and A. Donev. An immersed boundary method for rigid bodies. *Communications in Applied Mathematics and Computational Science*, 11(1):79–141, 2016. Software available at `https://github.com/stochasticHydroTools/RigidBodyIB`.

[26] Y. Bao, M. Rachh, E. E. Keaveny, L. Greengard, and A. Donev. A fluctuating boundary integral method for Brownian suspensions. *J. Comp. Phys.*, 374:1094 – 1119, 2018.

[27] Z. Gimbutas, L. Greengard, and S. Veerapaneni. Simple and efficient representations for the fundamental solutions of Stokes flow in a half-space. *Journal of Fluid Mechanics*, 776:R1, 2015. Code available at `http://www.cims.nyu.edu/cmcl/fmm3dlib/fmm3dlib.html`.

[28] David Saintillan, Eric Darve, and Eric SG Shaqfeh. A smooth particle-mesh ewald algorithm for stokes suspension simulations: The sedimentation of fibers. *Physics of Fluids*, 17(3):033301, 2005.

[29] Mu Wang and John F Brady. Spectral ewald acceleration of stokesian dynamics for polydisperse suspensions. *Journal of Computational Physics*, 306:443–477, 2016.

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

[31] James W Swan and Gang Wang. Rapid calculation of hydrodynamic and transport properties in concentrated solutions of colloidal particles and macromolecules. *Physics of Fluids (1994-present)*, 28(1):011902, 2016.

[32] M. Fixman. Simulation of polymer dynamics. I. General theory. *J. Chem. Phys.*, 69:1527, 1978.

[33] S. Delong, Y. Sun, B. E. Griffith, E. Vanden-Eijnden, and A. Donev. Multiscale temporal integrators for fluctuating hydrodynamics. *Phys. Rev. E*, 90:063312, 2014. Software available at `https://github.com/stochasticHydroTools/MixingIBAMR`.

[34] S. Delong, B. E. Griffith, E. Vanden-Eijnden, and A. Donev. Temporal Integrators for Fluctuating Hydrodynamics. *Phys. Rev. E*, 87(3):033302, 2013.

[35] Ayan Chakrabarty, Andrew Konya, Feng Wang, Jonathan V Selinger, Kai Sun, and Qi-Huo Wei. Brownian motion of boomerang colloidal particles. *Physical review letters*, 111(16):160603, 2013.

[36] Blaise Delmotte, Michelle Driscoll, Paul Chaikin, and Aleksandar Donev. Hydrodynamic shocks in microroller suspensions. *Phys. Rev. Fluids*, 2:092301, 2017.

[37] Ayan Chakrabarty, Andrew Konya, Feng Wang, Jonathan V. Selinger, Kai Sun, and Qi-Huo Wei. Brownian motion of arbitrarily shaped particles in two dimensions. *Langmuir*, 30(46):13844–13853, 2014.

[38] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, 2002.

[39] Hoang-Ngan Nguyen and Karin Leiderman. Computation of the singular and regularized image systems for doubly-periodic stokes flow in the presence of a wall. *Journal of Computational Physics*, 297:442–461, 2015.

[40] Dag Lindbo and Anna-Karin Tornberg. Fast and spectrally accurate summation of 2-periodic stokes potentials. *arXiv preprint arXiv:1111.1815*, 2011.

[41] Oana Marin, Katarina Gustavsson, and Anna-Karin Tornberg. A highly accurate boundary treatment for confined stokes flow. *Computers & Fluids*, 66:215–230, 2012.

[42] G.N. Milstein and M.V. Tretyakov. *Stochastic numerics for mathematical physics*. Springer, 2004.

[43] As explained in detail in Appendix A of [10], the rigid multiblob method can be seen as a first-kind boundary integral method regularized in a physically-consistent way so as to ensure the symmetry and positive definiteness of the mobility matrix, as required for adding Brownian motion.

[44] The random slip velocity $\breve{\boldsymbol{u}}$ can be derived by using fluctuating hydrodynamics. For example, one can start with the coupled fluid+particle equations given in Eq. (20) in [10], and then add a stochastic stress tensor to the Stokes equation for the fluid velocity [24, 26]. After elimination of the fluid one

obtains (9). This can most simply be done by starting from the fully discrete saddle-point system (25) in [10], adding the stochastic stress tensor as done in [24], and the using standard Schur complement techniques to eliminate the fluid velocity and pressure.

[45] In this paper, we evaluate the action of $\mathcal{M}$ directly using a summation on the GPU, which gives accuracy comparable to roundoff. But other more scalable implementations of the action of the RPY mobility would use approximate methods like the fast multipole method (FMM) [22, 27] or the Spectral Ewald method [16], which would have an input relative error tolerance of their own, larger than the roundoff error.

[46] On many GPUs it is more efficient to use single-precision arithmetic.

[47] The midpoint slip scheme was much more robust than the midpoint traction scheme for larger Δt.

[48] Note that the analysis in [36] shows that the steady uniform translating state is stable, and that density fluctuations propagate as waves without growing or shrinking in time.

[49] Note that this quantity gives the drift term produced by equation (26) in section III B.