The Journal of Chemical Physics

Computing collision stress in assemblies of active spherocylinders: Applications of a fast and generic geometric method

Cite as: J. Chem. Phys. **150**, 064109 (2019); https://doi.org/10.1063/1.5080433
Submitted: 08 November 2018 . Accepted: 28 January 2019 . Published Online: 13 February 2019

Wen Yan, Huan Zhang, and Michael J. Shelley

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Computing collision stress in assemblies of active spherocylinders: Applications of a fast and generic geometric method

Cite as: J. Chem. Phys. 150, 064109 (2019); doi: 10.1063/1.5080433
Submitted: 8 November 2018 • Accepted: 28 January 2019 •
Published Online: 13 February 2019

Wen Yan,[1,2,a)] Huan Zhang,[2,3] and Michael J. Shelley[1,2]

AFFILIATIONS

[1]Center for Computational Biology, Flatiron Institute, Simons Foundation, New York, New York 10010, USA
[2]Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA
[3]Zhiyuan College and Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China

Note: This article is part of the Special Topic "Chemical Physics of Active Matter" in J. Chem. Phys.
[a)]wyan@flatironinstitute.org and wenyan4work@gmail.com

ABSTRACT

In this work, we provide a solution to the problem of computing collision stress in particle-tracking simulations. First, a formulation for the collision stress between particles is derived as an extension of the virial stress formula to general-shaped particles with uniform or non-uniform mass density. Second, we describe a collision-resolution algorithm based on geometric constraint minimization which eliminates the stiff pairwise potentials in traditional methods. The method is validated with a comparison to the equation of state of Brownian spherocylinders. Then we demonstrate the application of this method in several emerging problems of soft active matter.

I. INTRODUCTION

Computing bulk collision stress is one of the key statistical tasks in simulations of many particle systems for both underdamped and overdamped, ranging from the molecular to the granular-flow scale. Collision stress is important because it contributes significantly to the Equation of State (EOS) and rheological properties of such systems. Notable examples include phase transitions in liquid crystals[1] and Active Brownian Particles (ABPs),[2] and the jamming and glassy states of spherical colloids.[3]

In simulations involving point particles, the collision stress can be computed with the usual virial formula $\langle \boldsymbol{x}\mathbf{F}_C \rangle$, where the moment $\boldsymbol{x}$ is the vector connecting each pair of point particles and $\mathbf{F}_C$ is the collision force between each pair. Collision stress in spherical particles of uniform density can be computed in the same way. A large volume of work can be found in the literature discussing all aspects of how to compute collision stress for various systems, but two problems remain. First, there remains some disagreement about how to compute the stress generated from one pair of colliding aspherical particles or spherical particles with nonuniform density. Some earlier work uses the same virial formula as in the point particle case, where the moment vector $\boldsymbol{x}$ is the vector connecting the center-of-mass of two particles.[4] In some work for slender rods, the moment vector $\boldsymbol{x}$ is taken to be the minimal distance between two centerlines of the colliding pair of rods.[5] In work for granular flow involving spherical particles, the virial contribution is integrated over the two particles' volumes, instead of picking only one point on each particle.[6,7] To the best of our knowledge, such different approaches have not been systematically examined.

Another crucial problem is how to detect and resolve the collisions. Traditionally, collisions are resolved by including a pairwise repulsive force, usually governed by Lennard-Jones (LJ) or Weeks-Chandler-Andersen (WCA) potential, and particle trajectories are integrated over time. There are two

key problems in this traditional approach. First, the pairwise repulsive potentials cause stiffness in the time-integrator and require very small time-step sizes. Second, such pairwise potentials always extend repulsive forces over a finite range, and therefore the collisions are resolved as if the particles were soft and deformable. For example, in work on Brownian rods,[8] the authors reported an "effective" diameter that is equal to around 90% of the imposed rod diameter because the repulsive forces cannot be infinitely stiff. Other collision-resolving methods have been developed upon the idea of geometric constraints. In these methods, the collision forces are not computed using an intermediate repulsive potential. Instead, the forces are *solved* for by imposing the geometric constraint that at the end of the current time-step, the particles cannot overlap. The method by Maury[9] is one notable example in this style, but his formulation does not preserve the pairwise collision network and therefore the necessary information to compute collision stress is lost. Another method by Tasora, Negrut, and Anitescu[10] follows similar ideas, but constructs the geometrical constraint problem in a way that the pairwise collision network and Newton's third law are all preserved. This method has been successfully applied in underdamped granular flow problems.

In this work, we present a complete and efficient solution to resolve collisions and to compute collision stress. We first resolve the discrepancies in the pairwise contribution to collision stress in Sec. II. The formula is derived as an extension to the virial stress formula in the most general settings, considering the momentum transfer throughout the entire volume of the particles. We then describe a collision resolution method for overdamped systems in Sec. III, together with a fast and parallel solver, as a generalization of the method by Tasora and Anitescu.[11] In particular, we allow the mobility matrix $\mathcal{M}$ to be computed by any method or approximations which keeps $\mathcal{M}$ symmetric-positive-definite (SPD). Our method is validated in Sec. IV by simulating Brownian spherocylinders and comparing the measured EOS with the classic work by Bolhuis and Frenkel.[1] In Sec. V, we demonstrate the application of our solution by measuring the collision stress in soft active matter systems, including self-propelled rods (SPRs) and growing-dividing cells.

II. PAIRWISE COLLISION STRESS

In this section, we consider the collision stress generated by one pair of particles in the most general setting, for both underdamped and overdamped systems. We make only the following assumptions of the collision between two rigid bodies:

- The collision force is between one point on particle 1 and one point on particle 2.
- The collision process is almost instantaneous.
- Newton's third law is satisfied.

In particular, *no* assumptions are made for the shape, friction, and density of the two particles. We shall also see that the existence of other forces like gravity does not change the formulae. Also, the two points where the collision force is transmitted do not have to be on the two particles' surfaces.

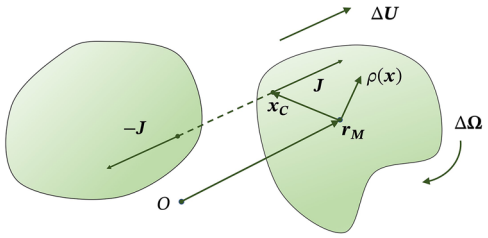

FIG. 1. Collision geometry of two arbitrary-shaped rigid bodies. $\boldsymbol{r}_M$ is the center of mass, and $\rho(\boldsymbol{x})$ is the mass density of the particle in the particle frame. $\boldsymbol{x}_C$ is the point where the transfer of momentum $\boldsymbol{J}$ happens. $\Delta\boldsymbol{U}$ and $\Delta\boldsymbol{\Omega}$ are the changes in the center of the mass velocity and the angular velocity, respectively, due to the action of $\boldsymbol{J}$.

We consider the collision geometry shown in Fig. 1. O is the origin of lab frame, and $\mathbf{r}_\mathrm{M}$ is the center of mass in the lab frame. $\boldsymbol{x}$ is the location of a mass point relative to the center of mass, and $\boldsymbol{x}_C$ is the location of collision in that frame. $\boldsymbol{J}$ is the impulse due to this collision event. For a small duration of collision, $\mathbf{J} = \mathbf{F}_C \delta t$.

A. Governing equations

Due to symmetry, it is sufficient to consider the motion of only one body of the collision pair. Let the change in velocity and angular velocity due to collision be $\Delta\mathbf{U}$ and $\Delta\boldsymbol{\Omega}$, respectively. With Newton's laws, we have two equations for translational motion

$$\int_\mathrm{V} \rho(\mathbf{U} + \boldsymbol{\Omega} \times \boldsymbol{x}) d\mathrm{V} + \mathbf{J} = \int_\mathrm{V} \rho[(\mathbf{U} + \Delta\mathbf{U}) + (\boldsymbol{\Omega} + \Delta\boldsymbol{\Omega}) \times \boldsymbol{x}] d\mathrm{V} \quad (1)$$

and rotational motion

$$\int_\mathrm{V} (\mathbf{r}_\mathrm{M} + \boldsymbol{x}) \times \rho(\mathbf{U} + \boldsymbol{\Omega} \times \boldsymbol{x}) d\mathrm{V} + (\mathbf{r}_\mathrm{M} + \mathbf{x}_\mathrm{C}) \times \mathbf{J} = \int_\mathrm{V} (\mathbf{r}_\mathrm{M} + \boldsymbol{x}) \times \rho[(\mathbf{U} + \Delta\mathbf{U}) + (\boldsymbol{\Omega} + \Delta\boldsymbol{\Omega}) \times \boldsymbol{x}] d\mathrm{V}. \quad (2)$$

We have the definition of mass M and the moment of inertia tensor $\mathbf{G}_\mathrm{M}$

$$\int_\mathrm{V} \rho d\mathrm{V} = \mathrm{M}, \quad (3)$$

$$\int_\mathrm{V} \rho \left(x^2 \mathbf{I} - \boldsymbol{x}\boldsymbol{x}\right) d\mathrm{V} = \mathbf{G}_\mathrm{M}. \quad (4)$$

By definition $\mathbf{G}_\mathrm{M}$ is always symmetric positive definite. Because $\boldsymbol{x}$ is the location in the particle frame relative to the center of mass, we have

$$\int_\mathrm{V} \rho \boldsymbol{x} d\mathrm{V} = \mathbf{0}. \quad (5)$$

We further define the tensors $\mathbf{N}$ and $\mathbf{Q}$ to simplify the tensor notations in the derivation, using

$$\mathbf{N} = \int_V \rho \mathbf{x}\mathbf{x} dV, \tag{6}$$

$$\mathbf{Q} = \mathbf{G}_M^{-1}. \tag{7}$$

Physically, the stress generated by this pair of particles colliding is related to the momentum transfer during the collision, which, quantitatively, is the integral of the "point-wise virial contribution $\mathbf{x}\mathbf{f}\delta t$" over the entire volume of the rigid body, denoted by the tensor $\mathbf{s}$, for both objects in the collision pair. In other words, the task is to determine $\mathbf{s}$ defined as

$$\mathbf{s} = \int_V \rho(\mathbf{r}_M + \mathbf{x})(\Delta\mathbf{U} + \Delta\boldsymbol{\Omega} \times \mathbf{x}) dV, \tag{8}$$

given the collision force and geometry. Once $\mathbf{s}$ is known for both particles 1 and 2, the collision stress generated by this pair is simply

$$\boldsymbol{\sigma}^{12} = \frac{1}{\delta t}\left(\mathbf{s}^1 + \mathbf{s}^2\right). \tag{9}$$

B. General results

Equations (1) and (2) can be simplified as

$$\mathbf{J} = M\Delta\mathbf{U}, \tag{10}$$

$$\mathbf{x}_C \times \mathbf{J} = \mathbf{G}_M \cdot \Delta\boldsymbol{\Omega}, \tag{11}$$

where we used the definition of the center of mass. Then $\mathbf{s}$ can be simplified

$$\mathbf{s} = \mathbf{r}_M\mathbf{J} + \int_V \rho\mathbf{x}(\Delta\mathbf{U} + \Delta\boldsymbol{\Omega} \times \mathbf{x}) dV. \tag{12}$$

The first term $\mathbf{r}_M\mathbf{J}$ simply corresponds to the virial stress. Since $\mathbf{U}$ is the center-of-mass velocity independent of $\mathbf{x}$, the integral $\int_V \rho\mathbf{x}\Delta\mathbf{U}$ in the second term vanishes by the definition of the center of mass. We define the integral as $\mathbf{s}^G$, i.e.,

$$\begin{aligned}\mathbf{s}^G &= \int_V \rho\mathbf{x}[\Delta\boldsymbol{\Omega} \times \mathbf{x}] dV \quad \text{or} \\ &= \int_V \rho\mathbf{x}[\mathbf{Q} \cdot (\mathbf{x}_C \times \mathbf{J}) \times \mathbf{x}] dV,\end{aligned} \tag{13}$$

where the superscript G stands for the geometric part of $\mathbf{s}$. Hence,

$$\mathbf{s} = \mathbf{r}_M\mathbf{J} + \mathbf{s}^G. \tag{14}$$

In tensor notation, $\mathbf{s}^G$ is

$$s_{ij}^G = N_{il}\epsilon_{jkl}\Delta\Omega_k = \epsilon_{jkl}N_{il}[\mathbf{Q} \cdot (\mathbf{x}_C \times \mathbf{J})]_k. \tag{15}$$

Here, ϵ_{jkl} is the Levi-Civita permutation symbol.

Up to this point, the derivation is for one rigid body in the collision pair. Due to symmetry and Newton's third law, the collision stress generated by this pair of particles, 1 and 2, is simply

$$\begin{aligned}\sigma_{ij}^{12} = &\left(r_{M,i}^2 - r_{M,i}^1\right)F_j^C + \epsilon_{jkl}N_{il}^2\left[\mathbf{Q}^2 \cdot \left(\mathbf{x}_C^2 \times \mathbf{F}_C\right)\right]_k \\ &+ \epsilon_{jkl}N_{il}^1\left[\mathbf{Q}^1 \cdot \left(\mathbf{x}_C^1 \times \mathbf{F}_C\right)\right]_k.\end{aligned} \tag{16}$$

Here, $\mathbf{F}_C$ points from particle 1 to particle 2.

Again, the first term in Eq. (16) is simply the virial stress, computed with the center of mass of the two particles. The extra terms are contributions due to the particles' shape and mass distribution. For objects with homogeneous density ρ, the formula [Eq. (16)] is purely geometric because the density ρ in N_{il} and $\mathbf{Q} = \mathbf{G}_M^{-1}$ cancel. Also, since the equations of motion [Eqs. (1) and (2)] are linear, the stress generated by multiple collisions between two particles, or several particles colliding with one particle, can all be simply summed over each $\mathbf{F}_C$.

In the above derivation, we made no assumption about how $\mathbf{F}_C$ is computed. In general, $\mathbf{F}_C$ can be computed in many different ways, depending on the physical setting and the collision resolution algorithms. For example, for simple smooth spheres, $\mathbf{F}_C$ can be computed with WCA potentials. While for more realistic granular flow models,[6] $\mathbf{F}_C$ can be computed with considerations for having coefficient of restitution and friction. The derivation of Eq. (16) is straightforward, but surprisingly not appreciated in the literature, except for a few special cases in which we will show that Eq. (16) reproduces those results.

C. Mechanical pressure of σ_{ij}^{12}

The mechanical pressure is defined as the isotropic diagonal part of the stress. For σ_{ij}^{12} given by Eq. (16), we can show that

$$\delta_{ij}\sigma_{ij}^{12} = \delta_{ij}\left(r_{M,i}^2 - r_{M,i}^1\right)F_{C,j}. \tag{17}$$

In other words, the extra geometric part of σ_{ij}^{12} changes only the deviatoric part of the collision stress. This is because $\delta_{ij}\epsilon_{jkl}N_{il}\Omega_k = \epsilon_{jkl}N_{jl}\Omega_k = 0$, for any Ω_k, due to the symmetry of N_{jl} and antisymmetry of ϵ_{jkl}.

Therefore, the mechanical collision pressure follows the usual virial formula

$$\Pi^{12} = \frac{1}{3}\left(r_{M,i}^2 - r_{M,i}^1\right)F_{C,i}. \tag{18}$$

D. Homogeneous frictionless spheres

In the case of homogeneous frictionless spheres, we always have $\mathbf{F}_C \parallel (\mathbf{r}_M^1 - \mathbf{r}_M^2) \parallel \mathbf{x}_C^1 \parallel \mathbf{x}_C^2$. Also $\mathbf{r}_M$ coincides with the geometric sphere center due to homogeneity. Therefore, the geometric contribution to stress is zero, and we have the usual virial formula

$$\sigma_{ij}^{12} = \left(r_{M,i}^2 - r_{M,i}^1\right)F_{C,j}^1, \tag{19}$$

as has been widely used in many studies on the rheology of spherical suspensions.[3,12]

E. Homogeneous frictional spheres

In the case of homogeneous frictional spheres, the collision force $\mathbf{F}_C$ is applied at the point of contact between the

two spheres. In the special case of two equal spheres, we have $\boldsymbol{x}^1_C = -\boldsymbol{x}^2_C$, and Eq. (16) reduces to

$$\sigma^{12}_{ij} = \left(r^2_{M,i} - r^1_{M,i}\right)F^1_{C,j}. \tag{20}$$

However, unlike the frictionless case, $\mathbf{F}^C$ is not necessarily parallel to $\boldsymbol{r}^2_M - \boldsymbol{r}^1_M$. Equation (20) reproduces the formula used by Campbell.[7]

F. Homogeneous frictionless long and thin rod

In the case of homogeneous frictionless long and thin rod, the shape and orientation of each body is solely determined by an orientation norm vector $\boldsymbol{n}$. Taking the rod simply as a line segment, any point $\boldsymbol{x}$ on the rod can be specified by

$$\boldsymbol{x} = x\boldsymbol{n}, \quad x \in [-L/2, L/2]. \tag{21}$$

In this case, head-to-head collision is negligible because of the assumption of being long and thin. Then in the absence of friction, we always have $\mathbf{J} \perp \boldsymbol{n}$. Therefore, $\Delta\boldsymbol{\Omega} = (\boldsymbol{x}_C \times \mathbf{J})/\gamma$, with $\gamma = \rho \int_{-L/2}^{L/2} x^2 dx$, and we have

$$\mathbf{s}^G = \frac{\rho}{\gamma} x_C \boldsymbol{n}\mathbf{J} \int_{-L/2}^{L/2} x^2 dx = \boldsymbol{x}_C \mathbf{J}. \tag{22}$$

Furthermore, Eq. (16) reduces to

$$\sigma^{12}_{ij} = \left(r^2_{M,i} + x^2_{C,i} - r^1_{M,i} - x^1_{C,i}\right)F^1_{C,j}, \tag{23}$$

which reproduces the formula used in the work by Snook *et al.*[5].

III. COLLISION RESOLUTION IN DYNAMIC SIMULATIONS

The other ingredient in our calculation of the collision stress is how to stably and efficiently compute the collision force $\mathbf{F}_C$ needed for Eq. (16). For underdamped systems with inertia, significant progress has been made by Tasora, Negrut, and Anitescu.[10] In this work, we extend this approach to overdamped systems because most active matter systems we are interested in are in this regime. Accordingly, we also focus on the completely inelastic collision case, where colliding bodies can remain in contact after collisions. Here, we ignore friction.

A. The mobility problem

We start from the mobility problem because having the mobility matrix being symmetric-positive-definite (SPD)[13] is one of the keys to the success of our method. Due to the linearity of Stokes equation, the dynamics of n_b rigid bodies is specified compactly by a linear equation

$$\mathcal{U} = \mathcal{M}\mathcal{F}, \tag{24}$$

where $\mathcal{U} = (\mathbf{U}_1, \boldsymbol{\Omega}_1, \mathbf{U}_2, \boldsymbol{\Omega}_2, \ldots)$ consists of translational and rotational velocities of each rigid body, and $\mathcal{F} = (\mathbf{F}_1, \mathbf{T}_1, \mathbf{F}_2, \mathbf{T}_2, \ldots)$ consists of the forces and torques on each rigid body. They are both column vectors with $6n_b$ entries. $\mathcal{M}$ is the *mobility matrix*, which contains all the solution information given by the Stokes equation and the no-slip boundary condition. Physically, the positive-definiteness can be explained by a simple observation that any non-zero force $\mathcal{F}$ applied to the rigid bodies dissipates energy into the viscous fluid, that is,

$$\mathcal{F} \cdot \mathcal{U} = \mathcal{F}^T \mathcal{M}\mathcal{F} > 0. \tag{25}$$

It is important that all the derivations in this work make no assumption about the shape of the rigid bodies or of the numerical method used to solve the mobility problem. Also, our approach does not require that the matrix $\mathcal{M}$ be explicitly constructed. As long as $\mathcal{U}$ can be computed with given force $\mathcal{F}$ for a given geometry, the method derived in this work can be applied. At the most crude level of description, the many-body coupling can be completely ignored and $\mathcal{M}$ becomes block-diagonal, describing isolated Brownian particles. With many-body coupling, the Rotne-Prager-Yamakawa tensor is a fairly inexpensive SPD approximation to $\mathcal{M}$ and can be used here straightforwardly. Stokesian Dynamics[14] can also be used in this method as a full hydrodynamics solver. The recent progress in boundary integral methods provides the most accurate solvers to the mobility problem, for which spheres[15,16] and rigid slender bodies[17,18] are examples.

B. Complementarity formulation for contact dynamics

The evolution of the geometric configuration $\boldsymbol{q}$ of a collection of rigid bodies is uniquely defined by the translational and rotational velocities $\mathbf{U}_k$ and $\boldsymbol{\Omega}_k$ for each particle k. Their velocities can be partitioned as the "known" velocities and the "collision" velocities

$$\mathbf{U}_k = \mathbf{U}_{k,known} + \mathbf{U}_{k,C}, \tag{26}$$

$$\boldsymbol{\Omega}_k = \boldsymbol{\Omega}_{k,known} + \boldsymbol{\Omega}_{k,C}, \tag{27}$$

where "known" stands for the known velocities before resolving the collisions. For example, for Brownian colloids, $\mathbf{U}_{k,known}$ and $\boldsymbol{\Omega}_{k,known}$ are Brownian displacements which can be computed without resolving the consequent collisions. Also for swimming bacteria, $\mathbf{U}_{k,known}$ and $\boldsymbol{\Omega}_{k,known}$ arise from the swimming motion.

The collision motion $\mathcal{U}_C = \mathcal{M}\mathcal{F}_C$ is governed by the mobility problem [Eq. (24)]. The collision velocities $\mathcal{U}_C$ are governed by the mobility problem $\mathcal{U}_C = \mathcal{M}_C\mathcal{F}_C$, i.e., Eq. (24). The equations of motion for the rigid bodies can be written as the evolution of configuration $\boldsymbol{q}$ with velocity $\mathcal{U}$

$$\dot{\boldsymbol{q}} = \mathcal{U}, \tag{28}$$

$$\mathcal{U} = \mathcal{U}_{known} + \mathcal{M}\mathcal{F}_C. \tag{29}$$

In this formulation, both $\mathcal{F}_C$ and $\mathcal{U}_C$ are the unknowns to be solved for, with the geometric constraint that $\boldsymbol{q}$ satisfies the non-overlap condition at all time t.

The geometric non-overlap condition can be defined as having a positive minimal separation, that is, $\Phi_\ell(\boldsymbol{q}) > 0$ between each close pair ℓ of rigid bodies, as a function of geometry

configuration $\boldsymbol{q}$. For each contact pair indexed ℓ, the positivity of minimal separation distance Φ_ℓ and the collision force magnitude γ_ℓ are mutually exclusive situations:

- No contact: $\Phi_\ell > 0$ and $\gamma_\ell = 0$.
- Contact: $\Phi_\ell = 0$ and $\gamma_\ell > 0$.

Mathematically, this is called a complementarity condition and is usually denoted by the following special notation combining all ℓ:

$$0 \le \boldsymbol{\Phi} \perp \boldsymbol{\gamma} \ge 0, \tag{30}$$

where $\boldsymbol{\Phi} = (\Phi_0, \Phi_1, \ldots)$ denotes the collection of minimal distances, and $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots)$ denotes the collection of all contact force magnitudes, for all possible contacts in the system. The dimension of both $\boldsymbol{\Phi}$ and $\boldsymbol{\gamma}$ is n_C, the total number of possible collisions in the system. n_C is identified by tracking the separation distance between pairs of rigid bodies that are close to collision. That is, once a pair of particles' separation Φ_ℓ is larger than a positive distance δ, this pair is then excluded from the collision resolution algorithm because they are far apart and cannot collide. This threshold distance δ is chosen empirically according to the system dynamics and is not necessarily a constant for all pairs over time. For example, we usually pick $\delta = 0.5(R_i + R_j)$ for a pair of spheres with radius R_i and R_j.

Now, for n_b rigid bodies appearing in the mobility problem, let $\mathbf{D}_\ell \in \mathbb{R}^{6n_b}$ be a sparse column vector containing geometric information mapping the magnitudes γ_ℓ to the collision force (and torque) vector on each rigid body. $\mathbf{D}_\ell$ defined in this way gives the force and torque on the two rigid bodies in this collision pair ℓ, as a linear function to the collision force magnitude γ_ℓ. Therefore, $\mathbf{D}_\ell$ has 12 non-zero entries for aspherical shapes, corresponding to 3 translational and 3 rotational degrees of freedom for each rigid body in the contact pair. For two spheres in contact without friction, $\mathbf{D}_\ell$ has only 6 non-zero entries because the normal collision forces induce no torques in this case. Then we can define a matrix $\mathcal{D} \in \mathbb{R}^{6n_b \times n_C}$ as the assembly of all $\mathbf{D}_\ell$ column vectors, mapping $\boldsymbol{\gamma}$ to the collision forces $\mathcal{F}_C$

$$\mathcal{F}_C = \mathcal{D}\boldsymbol{\gamma}, \tag{31}$$

$$\mathcal{D} = [\mathbf{D}_0\, \mathbf{D}_1\, \ldots\, \mathbf{D}_{n_C}] \in \mathbb{R}^{6n_b \times n_C}. \tag{32}$$

The details about the entries of $\mathcal{D}$ can be found in the work by Tasora, Negrut, and Anitescu.[10]

Then, the equations of motion result in the differential variational inequality

$$\dot{\boldsymbol{q}} = \mathcal{U}(\boldsymbol{q}), \tag{33}$$

$$\mathcal{U}(\boldsymbol{q}) = \mathcal{U}_{known}(\boldsymbol{q}) + \mathcal{M}(\boldsymbol{q})\mathcal{D}(\boldsymbol{q})\boldsymbol{\gamma}, \tag{34}$$

$$0 \le \boldsymbol{\Phi}(\boldsymbol{q}) \perp \boldsymbol{\gamma} \ge 0. \tag{35}$$

Here, $\mathcal{U}_{known}(\boldsymbol{q})$, $\mathcal{M}(\boldsymbol{q})$, and $\mathcal{D}(\boldsymbol{q})$ are all directly solvable with given geometry $\boldsymbol{q}$, without information about the collision force magnitudes $\boldsymbol{\gamma}$. This equation set is then solvable and integrable in time once a relation between the configuration $\boldsymbol{q}$ and the collision force $\boldsymbol{\gamma}$ is supplied, that is, a timestepping scheme.

With temporal discretization, the complementarity condition is satisfied at each time step instead of over all time t. The convergence theory for the trajectories of the rigid bodies has been analyzed in the work by Anitescu, Potra, and Stewart.[19] Here, we move forward to discuss the formulation and solution algorithm, without repeating the discussion of convergence to the continuous differential variational inequality. Higher order schemes such as the Runge-Kutta and Adams-Bashforth families can all be used, but for simplicity of derivation, we employ a first-order Euler scheme. Given position $\boldsymbol{q}^k$ and velocity $\mathcal{U}^k_{known}$ at a given time step t^k and step size Δt, velocity $\mathcal{U}^k$ and contact forces $\boldsymbol{\gamma}^k$ are solved via the nonlinear complementarity problem (NCP)

$$\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \Delta t\left(\mathcal{U}^k_{known} + \mathcal{M}(\boldsymbol{q})^k\mathcal{D}^k\boldsymbol{\gamma}^k\right), \tag{36}$$

$$0 \le \boldsymbol{\Phi}(\boldsymbol{q}^{k+1}) \perp \boldsymbol{\gamma}^k \ge 0. \tag{37}$$

The velocity $\mathcal{U}^k$ is then used to evolve the position in time.

This is an NCP because the minimum gap $\boldsymbol{\Phi}$ is in general a nonlinear function of $\boldsymbol{q}$. NCPs can often be solved iteratively by a series of linear complementarity problems (LCPs) with superlinear or quadratic convergence rate.[20] Here, we follow a simpler route rather than solving the NCP exactly. The time step size δt must be reasonably small to integrate $\mathcal{U}^k_{known}$ accurately, and so $\boldsymbol{\Phi}(\boldsymbol{q}^{k+1}) \ge 0$ can be linearized (and scaled with $1/\Delta t$) to yield

$$\frac{1}{\Delta t}\boldsymbol{\Phi}(\boldsymbol{q}^k) + \left(\nabla_{\boldsymbol{q}}\boldsymbol{\Phi}\right)^k\left[\mathcal{U}^k_{known} + \mathcal{M}^k\mathcal{D}^k\boldsymbol{\gamma}^k\right] \ge 0, \tag{38}$$

where the matrix $\left(\nabla_{\boldsymbol{q}}\boldsymbol{\Phi}\right)^k$ is simply the coefficients of the Taylor expansion of $\boldsymbol{\Phi}$ over $\boldsymbol{q}$ at time step t^k.

For rigid objects, it is straightforward to show that $\nabla_{\boldsymbol{q}}\boldsymbol{\Phi} = \mathcal{D}^T$. This is the same relation utilized in the work by Tasora, Negrut, and Anitescu.[10] The LCP problem can be written in the standard form

$$0 \le \mathbf{A}^k\boldsymbol{\gamma}^k + \boldsymbol{b}^k \perp \boldsymbol{\gamma}^k \ge 0, \tag{39}$$

where

$$\mathbf{A} = \mathcal{D}^T\mathcal{M}\mathcal{D}, \tag{40}$$

$$\boldsymbol{b} = \frac{1}{\Delta t}\boldsymbol{\Phi}(\boldsymbol{q}) + \mathcal{D}^T\mathcal{U}_{known}. \tag{41}$$

The term $\mathcal{D}^T\mathcal{U}_{known}$ computes the (linearized) changes in the minimal separation $\boldsymbol{\Phi}$ before the contact constraints are considered. We also note that each application of $\mathbf{A}$ corresponds to the solution of a mobility problem for the contact force $\mathcal{F}_C = \mathcal{D}\boldsymbol{\gamma}$. For large enough numbers of particles, it may thus be preferable to use matrix-free methods instead of constructing $\mathbf{A}$ explicitly.

The procedures of this collision resolution method based on LCP are as follows:

1. Compute $\mathcal{U}^k_{known}$ at time step t^k.
2. Compute the sparse matrix $\mathcal{D}^k$ with given geometric configuration $\boldsymbol{q}^k$ and the threshold δ for possible contacts.
3. Solve for $\boldsymbol{\gamma}^k$ with Eq. (39). $\mathcal{U}^k_C$ and $\mathcal{F}^k_C$ are solved simultaneously.
4. Evolve to $\boldsymbol{q}^{k+1}$ with $\mathcal{U}^k_{known} + \mathcal{U}^k_C$.

C. LCP solvers

In this section, we briefly discuss the solution methods to Eq. (39). The superscripts k denoting the time step are dropped to simplify the notation, since the LCP solution algorithms discussed here are generic methods not limited to collision resolution problems.

The matrix $\mathbf{A}$ defined in the LCP formulation [Eq. (39)] is symmetric-positive-semi-definite (SPSD) because the mobility matrix $\mathcal{M}$ is symmetric-positive-definite (SPD). Therefore, the LCP problem can be conveniently converted to a Constrained Quadratic Programming (CQP)[21]

$$\boldsymbol{\gamma} = \arg\min_{\boldsymbol{\gamma}\geq 0} f(\boldsymbol{\gamma}) = \frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{A}\boldsymbol{\gamma} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\gamma}. \tag{42}$$

From the physics perspective, the minimization of $f(\boldsymbol{\gamma})$ can be understood qualitatively as the minimization of the total virtual work done by the collision forces (and torques). This CQP formulation allows a wide range of algorithms. It can be solved with first order methods based on Projected Gradient Descent (PGD), where the projection is used to impose the constraint $\boldsymbol{\gamma} \geq 0$ during the gradient-descent minimization process. It can also be solved with second order Newton-type methods, for example, the minimum-map Newton method.[21]

It is beyond the scope of this work to discuss these methods in detail. Here, we solve the LCP problem with first order methods because we found that PGD methods are much more efficient since the gradient $\boldsymbol{g} = \nabla f = \mathbf{A}\boldsymbol{\gamma} + \boldsymbol{b}$ is inexpensive to compute for every gradient descent step. In particular, we found that Barzilai-Borwein Projected Gradient Descent (BBPGD) is much more efficient than the previously reported Accelerated Projected Gradient Descent (APGD)[22] because BBPGD does not rely on the estimation by back-tracking of the Lipschitz parameter of the function f. The BBPGD algorithm has been analyzed mathematically for generic CQP by Dai and Fletcher.[23] The procedures of BBPGD can be found in Appendix B.

The convergence of CQP solvers can be checked at each step by computing the L_2-norm $\phi(\boldsymbol{\gamma}, \boldsymbol{g}(\boldsymbol{\gamma}))$ of the minimum-map function $\mathbf{H}$

$$\phi(\boldsymbol{\gamma}, \boldsymbol{g}(\boldsymbol{\gamma})) = \|\mathbf{H}(\boldsymbol{\gamma}, \boldsymbol{g}(\boldsymbol{\gamma}))\|_2 < \epsilon_{tol}, \tag{43}$$

$$\mathbf{H}(\boldsymbol{\gamma}, \boldsymbol{g}(\boldsymbol{\gamma})) = \min(\boldsymbol{\gamma}, \mathbf{A}\boldsymbol{\gamma} + \mathbf{b}), \tag{44}$$

because the solution to the CQP is reached when $\phi = 0$. In this work, $\epsilon_{tol} = 10^{-5}$ is used unless otherwise noted. This criteria function ϕ is also efficient to compute because $\mathbf{A}\boldsymbol{\gamma} + \boldsymbol{b}$ is already computed as the gradient of the quadratic function at each gradient descent step.

D. Performance

The collision resolution algorithm based on the LCP [Eq. (39)] allows the time step size Δt to be increased by 10–100 times in comparison to the traditional method with LJ or WCA potentials because the stiffness induced by the potentials is eliminated. For each time step, the explicit construction of Eq. (39) has approximately the same cost as computing the pairwise repulsive force. After the construction, $\mathbf{A}\boldsymbol{\gamma} + \boldsymbol{b}$ must be computed once during each BBPGD minimization step. The total number of iterations increases slowly with the number of actual collisions, i.e., the number of positive entries in the solution $\boldsymbol{\gamma}$. Empirically, 5–10 iterations is enough for dilute systems. Since $\mathbf{A}\boldsymbol{\gamma} + \boldsymbol{b}$ can be computed with standard sparse matrix-vector multiplication operations (`spmv`) efficiently, the solution of Eq. (39) is usually not a significant extra cost unless the system is densely packed and close to the random-close-packing (RCP) limit, where O(1000) iterations is necessary. Therefore, overall this LCP-based method significantly increases both the stability and efficiency of resolving collisions compared to repulsive potential methods. We also implemented this algorithm with full MPI (Message Passing Interface) and OpenMP parallelism, and the program scales efficiently to $O(10^7)$ particles on O(100) cores.

IV. VALIDATION

To validate our derivation for the collision stress and resolution algorithm, particles with aspherical shapes should be used because otherwise the geometric part in Eq. (16) varnishes. Unfortunately, such widely accepted and available benchmark data are only available for a few systems, partially because the difficulty of handling collisions between aspherical particles and computing the stress.

In this section, we extract the Equation-of-State (EOS) of monodisperse Brownian spherocylinders of length L and diameter D (Fig. 2) and compare the results with the benchmark data reported by Bolhuis and Frenkel.[1] In this purely Brownian system, the many-body hydrodynamics coupling in the mobility matrix is ignored, i.e., $\mathcal{M}$ becomes block diagonal, with each block being the translational and rotational mobility matrix $\mathbf{M}_{tt}$ and $\mathbf{M}_{rr}$ for each spherocylinder. The coupling between rotational and translational motion is also ignored

$$\mathbf{M}_{tt} = \frac{1}{\zeta_{\parallel}}\boldsymbol{nn}^{\mathrm{T}} + \frac{1}{\zeta_{\perp}}\left(\mathbf{I} - \boldsymbol{nn}^{\mathrm{T}}\right), \tag{45}$$

$$\mathbf{M}_{rr} = \frac{1}{\zeta_r}\mathbf{I}. \tag{46}$$

Here, $\boldsymbol{n}$ is the orientation norm vector of the spherocylinder. The drag coefficients are approximated by slender body theory of straight rigid fibers[17]

$$1/\zeta_{\parallel} = 2b/(8\pi L\mu), \tag{47}$$

$$1/\zeta_{\perp} = (b+2)/(8\pi L\mu), \tag{48}$$

$$1/\zeta_r = 3(b+2)/(2\pi L^3\mu), \tag{49}$$

$$b = -(1 + 2\log[D/(2L)]). \tag{50}$$

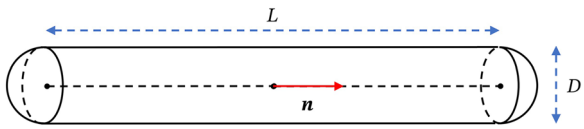

FIG. 2. The geometry of a spherocylinder of length L, width D, and orientation $\boldsymbol{n}$.

Similar but different drag coefficients are often used in previous studies.[8,24]

The Brownian velocities $\mathbf{U}^{\mathrm{B}}$ and $\boldsymbol{\Omega}^{\mathrm{B}}$ for each spherocylinder are computed by the Random-Finite-Difference (RFD) algorithm[25] treating $\mathbf{M}_{tt}$ and $\mathbf{M}_{rr}$ independently because the many-body coupling has been ignored. Then the "known" velocity $\mathcal{U}_{known}$ in Eq. (29) is just the Brownian velocity $\mathcal{U}^{\mathrm{B}} = (\mathbf{U}_1^{\mathrm{B}}, \boldsymbol{\Omega}_1^{\mathrm{B}}, \mathbf{U}_2^{\mathrm{B}}, \boldsymbol{\Omega}_2^{\mathrm{B}}, \ldots)$. The necessary geometric quantities in Eq. (16) and the sparse matrix $\mathcal{D}$ in the LCP collision resolution algorithm are computed with the method described in Appendixes A and C. The BBPGD algorithm is then used to solve the CQP (equivalent to the LCP) for $\mathcal{U}_{\mathrm{C}}$ and $\mathcal{F}_{\mathrm{C}}$. The system stress is then computed with $\mathcal{F}_{\mathrm{C}}$ according to Eq. (16) for each pair in the collision.

Periodic boundary conditions are imposed in each direction of the rectangular simulation box of size $L_x \times L_y \times L_z$, containing N spherocylinders. $n = \mathrm{N}/(L_x L_y L_z)$ is the number density. The system total stress and pressure are computed with a simple average of all collision pairs' contributions

$$\boldsymbol{\Sigma} = nk_{\mathrm{B}}\mathrm{T}\mathbf{I} + \frac{1}{\mathrm{N}}\sum \boldsymbol{\sigma}^{col}, \tag{51}$$

$$\Pi = \frac{1}{3}\mathrm{Tr}\boldsymbol{\Sigma}, \quad \Pi^{col} = \frac{1}{3\mathrm{N}}\sum \mathrm{Tr}\boldsymbol{\sigma}^{col}. \tag{52}$$

The kinetic part $nk_{\mathrm{B}}\mathrm{T}\mathbf{I}$ is *imposed* with given $k_{\mathrm{B}}\mathrm{T}$ through the Brownian motion moves in overdamped simulations.

A. The isotropic phase

We first present results in the isotropic phase, as shown in Fig. 3. The simulations start from a random placement and orientation of N = 2000 spherocylinders of varying aspect ratio L/D in a cubic periodic box and are equilibrated with a fixed box size until the measured stress reaches a steady state. This process usually takes about 10^5 time steps. Then the system pressure Π is averaged over another 2000 time steps. The method described in this work accurately reproduces the standard data reported by Bolhuis and Frenkel.[1]

B. The isotropic-nematic phase transition

Beyond the isotropic phase, the simulations are much more demanding because the system relaxation time becomes significantly longer. In this regime, if a simulation is simply started from a random configuration, it remains "jammed" in this structure for a long time, even when the system density is in the nematic phase regime. Limited by computing resources, we conduct dense simulations starting from N randomly located, but all aligned configuration of spherocylinders. The fixed simulation box is fixed with $L_x > L_y = L_z$, and the spherocylinders are aligned in the x direction. N = 2000 is fixed but the box sizes are varied around $72D \times 15D \times 15D$ for different volume fractions. Simulations with N = 6000 spherocylinders in a cubic periodic box are also performed, and the results reported here are not impacted by the box shape.

We focus on the isotropic-nematic transition for $L/D = 5$, where a nematic phase can stably exist, because it is not too close to the isotropic-nematic-smectic triple point at around $L/D \approx 3.7$ estimated by Bolhuis and Frenkel.[1] The pressure and its standard deviation are also calculated with equilibrated systems in the same way as described above. The results for the measured pressure agrees well with the results by Bolhuis and Frenkel,[1] as shown in Fig. 4.

Figure 4 shows a jump in pressure at $\phi \approx 0.4$. More information about this isotropic-nematic transition can be extracted by measuring the orientation order parameter

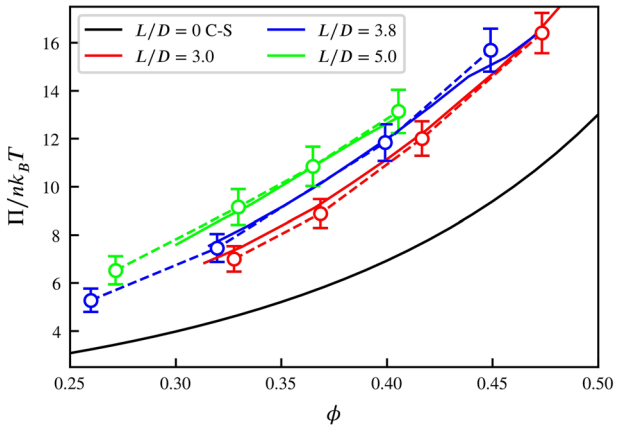

FIG. 3. The pressure of Brownian spherocylinders in the isotropic phase. The open circles with error bars connected by dashed lines are measured from simulations. The error bars show the standard deviation of pressure within the time-average window. The solid lines are data extracted from the work by Bolhuis and Frenkel.[1] The black line shows the Carnahan-Starling equation for hard spheres ($L/D = 0$) as a reference.

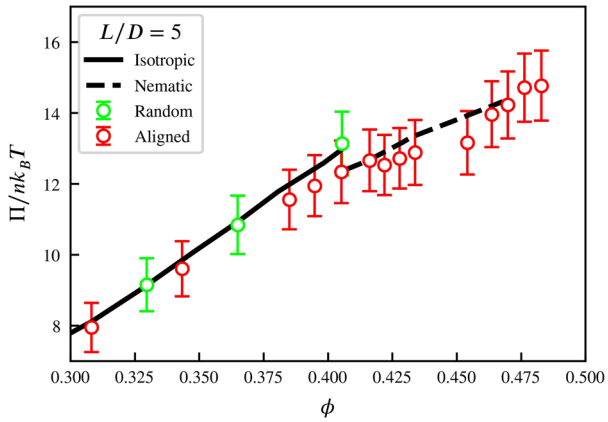

FIG. 4. The pressure of Brownian spherocylinders close to the isotropic-nematic phase transition. The green symbols are simulations starting from a randomly oriented configuration in a cubic box, and the red symbols are simulations starting from a random center location but aligned orientation in a rectangular box. The error bars show the range of standard deviation of pressure within the time-average window. The solid line representing the isotropic phase and the dashed line representing the nematic phase are both data extracted from the work by Bolhuis and Frenkel.[1]

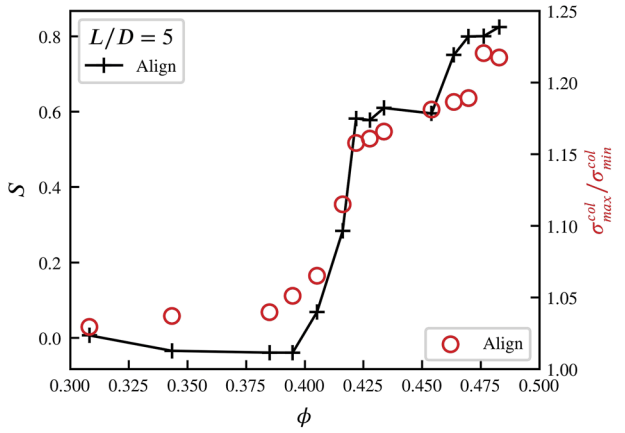

FIG. 5. The jump in orientation order parameter S and the collision stress anisotropy during the isotropic-nematic phase transition for $L/D = 5$. The black symbols connected by a solid line show the order parameter S, and the red open circles show the anisotropy. All data are extracted from the same set of simulations starting from a random but aligned state, as in Fig. 4.

$S = \langle P_2(\boldsymbol{n}\cdot\bar{\boldsymbol{n}})\rangle$, where P_2 is the order-2 Legendre polynomial and $\bar{\boldsymbol{n}}$ is the average orientation of spherocylinders at a specific time in the simulation. Furthermore, the anisotropy of the system pressure can be quantitatively investigated by computing the ratio of the maximum to the minimum of the eigenvalues $\sigma^{col}_{max}/\sigma^{col}_{min}$ of the collision stress tensor $\boldsymbol{\sigma}^{col}$. As shown in Fig. 5, the anisotropy ratio $\sigma^{col}_{max}/\sigma^{col}_{min}$ closely follows the jump in S, which shows that the isotropic-nematic phase transition for $L/D = 5$ happens at $\phi \approx 0.42$. Last, but not least, the computed stress tensor $\boldsymbol{\sigma}^{12}$ is exactly symmetric without Brownian noise for each pair of spherocylinders at each time step, as required by the general principal of continuum mechanics. This would not be satisfied if the geometric part in Eq. (16) is not included in the stress calculation.

V. APPLICATION

In this section, we demonstrate a few applications of the computational framework described in this work to the area of soft active matter, namely, self-propelled rods and growing-dividing cells.

A. Self-propelled rods

The Active Brownian Particle (ABP) model has attracted much attention because despite being a minimal model it can be used to explain many important features of soft active matter systems. However, the similar Self-Propelled Rod (SPR) model has not been investigated in such detail in the literature. Almost all related work focuses on 2D systems,[26–32] mostly because the collisions are difficult to handle in 3D. In particular, an EOS has not been quantitatively measured. In this work, we report briefly on the enhancement of collision pressure for dilute Brownian SPR systems. The Brownian SPR model we consider here is exactly the same as the Brownian spherocylinders considered in Sec. IV except that each spherocylinder has a propulsion speed $\mathbf{U}_0$ along its orientation norm vector $\boldsymbol{n}$.

The virial expansion of the EOS can be written as[33]

$$\frac{\Pi}{nk_BT} = 1 + B_2 n + B_3 n^2 + \cdots, \tag{53}$$

or,

$$1 + \frac{\Pi^{col}}{nk_BT} = 1 + B_2\frac{\phi}{v_0} + \frac{B_3}{B_2^2}\left(B_2\frac{\phi}{v_0}\right)^2 + \cdots, \tag{54}$$

where $v_0 = \pi\left(\frac{1}{4}LD^2 + \frac{1}{6}D^3\right)$ is the volume of a single rod (spherocylinder). In the limit of $\phi \to 0$, the higher order terms varnish and the EOS can be approximately written as

$$\frac{\Pi^{col}}{nk_BT} \approx B_2\frac{\phi}{v_0}. \tag{55}$$

When $U_0 = 0$, $B_2 = \pi\left(\frac{2}{3}D^3 + LD^2 + L^2D/4\right)$ is analytically known.[34,35] Therefore, we measure the enhancement of collision pressure Π^{col} due to self-propulsion with simulations at a given L/D and ϕ, with varying U_0. We simulate $N = 4\times10^5$ SPRs in a fixed cubic periodic box to overcome the strong effect of Brownian noise in such dilute systems and guarantee that the persistence length U_0/D_R is much smaller than the box size. We take $\phi = 0.0052$ for $L/D = 5$, $\phi = 0.0065$ for $L/D = 10$, and $\phi = 0.0065$ for $L/D = 20$. Such dilute systems remain isotropic with varying U_0. We plot the measured $\Pi^{col}/(B_2 n)$ as a function of dimensionless velocity $U_0/(LD_R)$, where $D_R = k_BT/\zeta_r$ is computed as in Sec. IV when $\phi \to 0$.

The results of this measurement are shown in Fig. 6. The collision pressure increases almost linearly as the propulsion speed U_0. Some work[36] proposed an "effective length" $L_U = \sqrt{L(L + U_0/D_R)}$ to approximate the effect of propulsion. Substituting L_U into the analytic expression for $B_2 = \pi\left(\frac{2}{3}D^3 + LD^2 + L^2D/4\right)$ does generate a linear scaling as

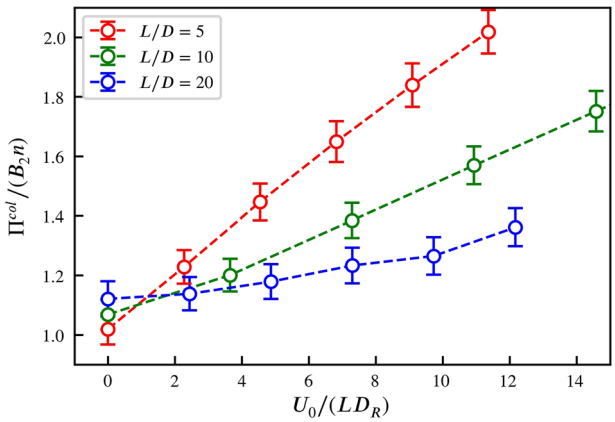

FIG. 6. The enhancement of collision pressure due to self-propelled velocity U_0 in the dilute limit. Here, $\phi = 0.0052$ for $L/D = 5$, $\phi = 0.0065$ for $L/D = 10$, and $\phi = 0.0065$ for $L/D = 20$. The results and error bars are averaged for 2000 time steps over equilibrated systems.

U_0 when $L/D \to \infty$, but we found that quantitatively this simple scaling law fails in predicting both the value and the trends of the data shown in Fig. 6.

Ideally, $\Pi^{col}/(B_2 n) \to 1$ at $U_0 = 0$, which is approximately the case of $L/D = 5$. For $L/D = 10$ and 20, there is about 10% error because the contributions from B_3 and B_4 remain important. Using a more dilute system could help resolve this issue, but a larger number of SPRs are necessary to overcome the Brownian noise, which is currently beyond our computing power. However, this slight mismatch does not change our conclusion of the linear scaling between Π^{col} and U_0.

B. Growing and dividing cells

The collision stress [Eq. (16)] and the LCP method [Eq. (39)] are derived for rigid bodies in Secs. II and III. However, this assumption only means that they are rigid in response to collision forces. Besides this, they can freely deform and Eqs. (16) and (39) are still applicable. Growing and dividing cells are one of the examples with which we can demonstrate the applications where the objects are changing their shapes, even discontinuously. In the following, we present some interesting stress measurement for systems of a minimal model of growing and dividing cells. The model is unrealistic because the growing and dividing process is assumed to be synchronized for all cells and the time between divisions is very short. We use this model only to demonstrate the capability of the computational method. More realistic biological parameters can be straightforwardly added to this minimal model in our future study.

We model biological cells as spherocylinders where the diameter D remains constant but the length L grows linearly in time. All cells start to grow from a specified original length L_0 at $t = 0$. Once the length reaches the specified division length L_D, each cell splits into two shorter cells with equal length L_0. This division is assumed to occur instantaneously. As shown in Fig. 7, we choose $L_D = 2L_0 + D$. The new cells continue this growing-dividing cycle. The number of cells in the simulation box therefore exponentially grows over time. The division time τ_{div} denotes the time one cell grows from L_0 to L_D, i.e., the time between two consecutive division events.

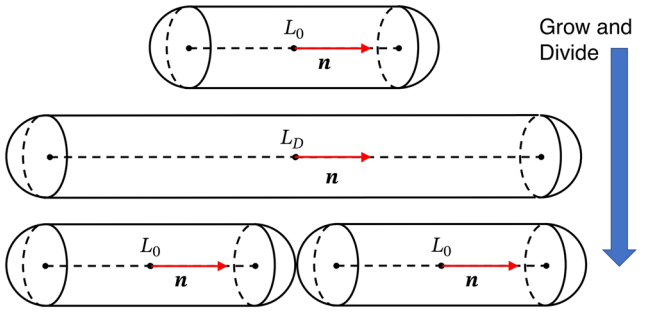

FIG. 7. The division of cells modelled as splitting of spherocylinders. The cell with length L_D divides into two cells with equal length L_0. The total cell volume slightly decreases in this division process due to the shape change in the center. The orientation norm vector $\boldsymbol{n}$ remains unchanged after the division.

We use dimensional units $D = 1\,\mu$m, $L_0 = 2.5\,\mu$m, $L_D = 6\,\mu$m, and viscosity $\mu = 0.001$ Pa s, close to the viscosity of water at room temperature. The Brownian motion is also computed as in Sec. IV, where at room temperature $k_BT = 0.004\,11$ pN μm. All cells are assumed to divide at the same time. They are also assumed to swim in the direction $\boldsymbol{n}$ with velocity $\mathbf{U}_0 = U_0\boldsymbol{n}$ as the SPR model. All simulations start from 100 cells randomly and homogeneously distributed in a periodic cubic $100 \times 100 \times 100\ \mu$m^3 box.

In this problem, there are a variety of time scales, including the Brownian time scale D_R^{-1}, the swimming time scale L_0/U_0, the cell division time scale τ_{div}, and the system relaxation time scale where the cell number density relaxes to a homogeneous distribution after each division. A thorough investigation is beyond the scope of the current work, and we only report the results for a fast growing case where τ_{div} is longer than D_R^{-1} but is much shorter than the density relaxation time scale. We choose the rotational diffusion time D_R^{-1} for cells with length L_0 as the unit of time. $D_R^{-1} = 1.89$ s and we pick $\tau_{div} = 3.5$ s.

The results are reported in dimensionless numbers in Fig. 8, where a moving average window of 100 time steps is applied to the measured Π^{col} to filter the Brownian fluctuations. The measured collision pressure shows a peak, at almost the same height, at every division event before the volume fraction ϕ reaches 10%. This is because in dilute systems most collisions are contributed by those "newborn" pairs of cells with length L_0. This contribution is proportional to the total number of cells in the system,

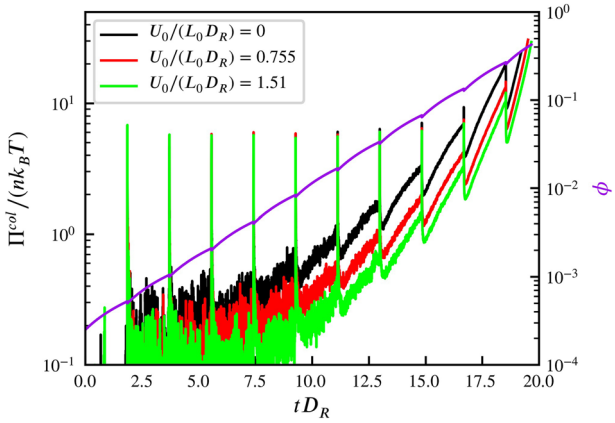

FIG. 8. The increase in collision pressure Π^{col} for dividing cells with different self-propelled velocities $\boldsymbol{U}_0$. The time step $\delta t = 5.3 \times 10^{-5} D_R^{-1}$. A moving average window of 100 time steps is applied to the measured Π^{col} to filter the Brownian fluctuations. The purple line shows the exponential growth of the volume fraction ϕ over time. The tiny dips in ϕ at each collision event correspond to the slight decrease in the total cell volume as shown in Fig. 7.

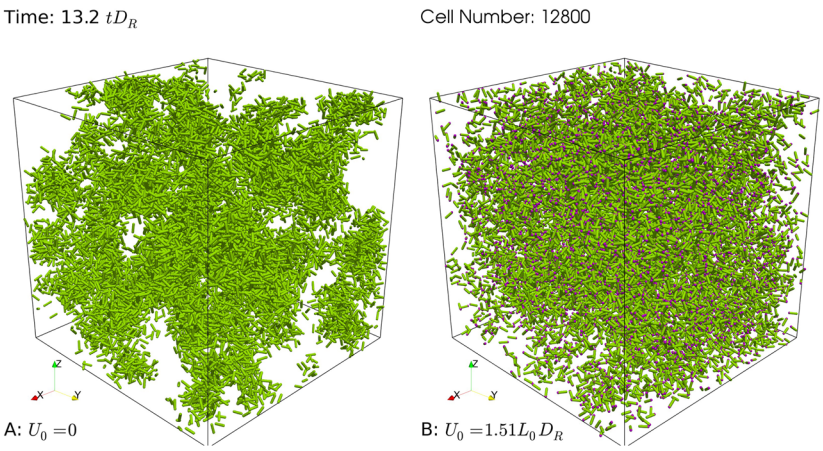

FIG. 9. The snapshot of dividing cells for (a) U_0 = 0 and (b) U_0 = 1.51L_0D_R at time tD_R = 13.2. The purple dots mark the heads of the moving cells.

and therefore, when Π^{col} is scaled by nk_BT, the total number is scaled out and the peaks are of almost the same height.

Another notable feature is that the system with faster swimming velocity U_0 has lower collision pressure. This is because the density relaxation time scale decreases with increasing U_0. As shown in Figs. 9(a) and 10(a), when U_0 = 0 the cells form local clusters because the division time τ_{div} is not sufficiently long for them to diffuse translationally. Such high density clusters increase the system collision

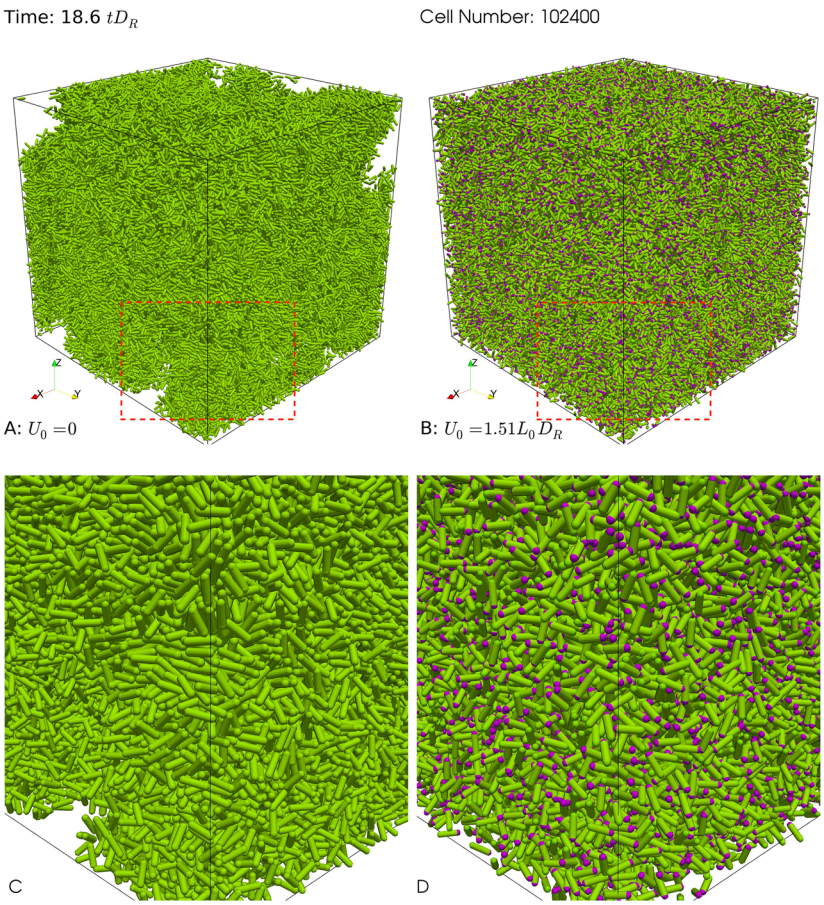

FIG. 10. The snapshot of dividing cells for (a) U_0 = 0 and (b) U_0 = 1.51L_0D_R at time tD_R = 18.6. The purple dots mark the heads of the moving cells. The red box marks the region shown in (c) and (d).

pressure significantly. While in Figs. 9(b) and 10(b) when $U_0 = 2\ \mu m\ s^{-1} = 1.51 L_0 D_R$, the system number density remains approximately homogeneous because of the swimming motion.

VI. CONCLUSION

In this work, we described a complete solution for computing the collision stress for moving rigid particle assemblies. We first developed the general expression [Eq. (16)] to compute the collision stress for each colliding pair of particles, based on the idea of volumetric integration of momentum transfer in that collision event. Equation (16) is then demonstrated in Sec. II to reproduce known expressions in various simplified cases. This task can be completed by the LCP based collision resolution algorithm described in Sec. III. The idea is to utilize the geometric non-overlapping constraints and to remove the stiff pairwise repulsive potentials. Our method is validated in Sec. IV by measuring the system EOS for Brownian spherocylinders and finding accurate agreement with the work by Bolhuis and Frenkel.[1] We further demonstrated briefly the applications of this method in Sec. V for (i) self-propelled rods and (ii) growing-dividing cells. This new method allows us to measure mechanical properties in such soft active matter systems straightforwardly.

The method described in this work can be applied to various systems, as long as (i) the collision geometry for a pair of particles can be computed and (ii) the mobility matrix can be computed. We designed the method such that the mobility matrix $\mathcal{M}$ appears only as an abstract matrix-vector multiplication operator. In this way, $\mathcal{M}$ can be computed with *any* method without the necessity to explicitly construct the matrix, as long as the method keeps $\mathcal{M}$ symmetric positive definite. In this paper, we focused on the case where the many-body coupling in $\mathcal{M}$ is ignored, i.e., $\mathcal{M}$ is block-diagonal. The same algorithm [Eq. (39)] also works for the cases with full hydrodynamics. For example, Rotne-Prager-Yamakawa tensor,[37] Stokesian Dynamics,[14] and Boundary Integral method[16] can all be used depending on the required accuracy for hydrodynamics for rigid particle suspensions. We leave the analysis about the cases with full hydrodynamics to other forthcoming studies.

Last, but not least, Eq. (16) is applicable not only to the collision stress. It is applicable to all cases where some form of momentum transfer happens from a point on one object to a point on another object. Furthermore, the impulse $\mathbf{J}$ does not have to be along the direction between the two points of momentum transfer. As long as the force $\mathbf{F}_C$ and the geometry during the event can be computed, the stress follows Eq. (16). For example, in a microtubule network driven by motor proteins,[38] the stress between microtubules generated by motor proteins can be computed with Eq. (16) by replacing the force $\mathbf{F}_C$ with the protein pushing or pulling force. This paves the way to more fundamental understanding of the mechanical properties of such biological active networks.

ACKNOWLEDGMENTS

M.J.S. thanks the support from NSF Grant Nos. DMR-1420073 (NYUMRSEC), DMS-1463962, and DMS-1620331.

APPENDIX A: GEOMETRY OF SPHEROCYLINDERS

Spherocylinders are cylinders of length L and diameter D, capped with two hemispheres. We define $\beta = L/D = L/(2R)$. In the coordinate system where the spherocylinder is aligned with the z axis, the integral $\mathbf{N}$ and moment of inertia tensor $\mathbf{G}_M$ are diagonalized

$$\mathbf{N} = \rho \begin{bmatrix} N_\perp & 0 & 0 \\ 0 & N_\perp & 0 \\ 0 & 0 & N_\parallel \end{bmatrix}, \tag{A1}$$

$$\mathbf{G}_M = \rho \begin{bmatrix} G_{M,\perp} & 0 & 0 \\ 0 & G_{M,\perp} & 0 \\ 0 & 0 & G_{M,\parallel} \end{bmatrix}, \tag{A2}$$

where

$$N_\perp = \frac{1}{30}(15\beta + 8)\pi R^5, \tag{A3}$$

$$N_\parallel = \frac{1}{15}\left(10\beta^3 + 20\beta^2 + 15\beta + 4\right)\pi R^5, \tag{A4}$$

$$G_{M,\perp} = \frac{1}{30}\left(20\beta^3 + 40\beta^2 + 45\beta + 16\right)\pi R^5, \tag{A5}$$

$$G_{M,\parallel} = \frac{1}{15}(15\beta + 8)\pi R^5. \tag{A6}$$

APPENDIX B: BBPGD

This method is summarized in Algorithm 1.

In this algorithm, α_k^{BB1} (next to the last line) is not the only choice. $\alpha_k^{BB2} = \mathbf{s}_{k-1}^T \mathbf{y}_{k-1} / \mathbf{y}_{k-1}^T \mathbf{y}_{k-1}$ can also be used. We find that there is no significant difference in performance of different choices of α_k^{BB1} or α_k^{BB2} in solving our problems, and α_k^{BB1} is used for all results reported in this work.

Algorithm 1. The Barzilai-Borwein projected gradient descent method.

Solve Eq. (42) with initial guess $\boldsymbol{\gamma}_0$, residual tolerance ϵ_{tol}, and k_{max}.
$\boldsymbol{g}_0 = \mathbf{A}\boldsymbol{\gamma}_0 + \boldsymbol{b}$.
if $\phi(\boldsymbol{\gamma}_0, \boldsymbol{g}_0) < \epsilon$ **then**
Solution is $\boldsymbol{\gamma}_0$.
end if
Simple gradient-descent step size $\alpha_0 = \boldsymbol{g}_0^T \boldsymbol{g}_0 / \boldsymbol{g}_0^T \mathbf{A} \boldsymbol{g}_0$.
for $k = 1$: $k_{\max}$ **do**
The descent step: $\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} - \alpha_{k-1} \boldsymbol{g}_{k-1}$.
The projection step: $\boldsymbol{\gamma}_k = \Pi_{\boldsymbol{\gamma} \geq 0}[\boldsymbol{\gamma}_k]$.
Compute the gradient $\boldsymbol{g}_k = \mathbf{A}\boldsymbol{\gamma}_k + \boldsymbol{b}$.
if $\varphi(\boldsymbol{\gamma}_k, \boldsymbol{g}_k) \leq \epsilon_{tol}$ **then**
Stop iteration, solution is $\boldsymbol{\gamma}_k$.
end if
$\mathbf{s}_{k-1} = \boldsymbol{\gamma}_k - \boldsymbol{\gamma}_{k-1}$, $\mathbf{y}_{k-1} = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$.
$\alpha_k^{BB1} = \mathbf{s}_{k-1}^T \mathbf{s}_{k-1} / \mathbf{s}_{k-1}^T \mathbf{y}_{k-1}$.
end for

APPENDIX C: COLLISION BETWEEN SPHEROCYLINDERS

This appendix describes how to find the minimum separation between a pair of spherocylinders. Geometrically, this task can be reduced to find the minimum distance between two line segments $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{Q}_0$, $\mathbf{Q}_1$ in 3D space, where $\mathbf{P}_0$, $\mathbf{P}_1$ (also $\mathbf{Q}_0$, $\mathbf{Q}_1$) are the two end points of the cylindrical section of one spherocylinder, as shown in Fig. 11.

We parameterized the two spherocylinders with scalars $0 < s, t < 1$: $\mathbf{P}(s) = (1 - s)\mathbf{P}_0 + s\mathbf{P}_1$ and $\mathbf{Q}(t) = (1 - t)\mathbf{Q}_0 + t\mathbf{Q}_1$. Then the square distance between two points on the segments is the quadratic function

$$R(s,t) = |\mathbf{P}(s) - \mathbf{Q}(t)|^2 \tag{C1}$$

$$= as^2 - 2bst + ct^2 + 2ds - 2et + f \tag{C2}$$

$$= \boldsymbol{p}^T\mathbf{M}\boldsymbol{p} + 2\mathbf{K}^T\boldsymbol{p} + f, \tag{C3}$$

where

$$\boldsymbol{p}^T = \begin{bmatrix} s & t \end{bmatrix}, \tag{C4}$$

$$\mathbf{M} = \begin{bmatrix} a & -b \\ -b & c \end{bmatrix}, \tag{C5}$$

$$\mathbf{K}^T = \begin{bmatrix} d & -e \end{bmatrix}, \tag{C6}$$

$$a = (\mathbf{P}_1 - \mathbf{P}_0) \cdot (\mathbf{P}_1 - \mathbf{P}_0), \tag{C7}$$

$$b = (\mathbf{P}_1 - \mathbf{P}_0) \cdot (\mathbf{Q}_1 - \mathbf{Q}_0), \tag{C8}$$

$$c = (\mathbf{Q}_1 - \mathbf{Q}_0) \cdot (\mathbf{Q}_1 - \mathbf{Q}_0), \tag{C9}$$

$$d = (\mathbf{P}_1 - \mathbf{P}_0) \cdot (\mathbf{P}_0 - \mathbf{Q}_0), \tag{C10}$$

$$e = (\mathbf{Q}_1 - \mathbf{Q}_0) \cdot (\mathbf{P}_0 - \mathbf{Q}_0), \tag{C11}$$

$$f = (\mathbf{P}_0 - \mathbf{Q}_0) \cdot (\mathbf{P}_0 - \mathbf{Q}_0). \tag{C12}$$

$R(s, t)$ is a quadratic function to minimize on unit square $(s, t) \in [0, 1]^2$. Observe that

$$\det \mathbf{M} = ac - b^2 = |(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{Q}_1 - \mathbf{Q}_0)|^2 \geq 0. \tag{C13}$$

The minimization of $R(s, t)$ is straightforward, unless the two line segments are close to parallel, i.e., $\det \mathbf{M} \to 0$. In this special case, numerical instabilities may occur due to the singularity of $\mathbf{M}$. To handle all cases robustly, we follow the method described in the computational geometry library `Geometric Tools`,[39] where a constrained conjugate gradient approach is used. In our tests, this method computes the solution s_N, t_N both efficiently and robustly.

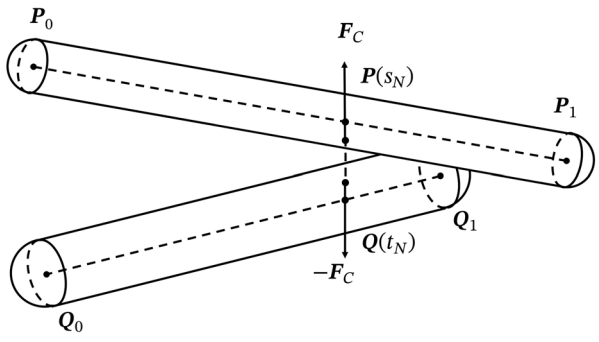

FIG. 11. Collision geometry of two spherocylinders.

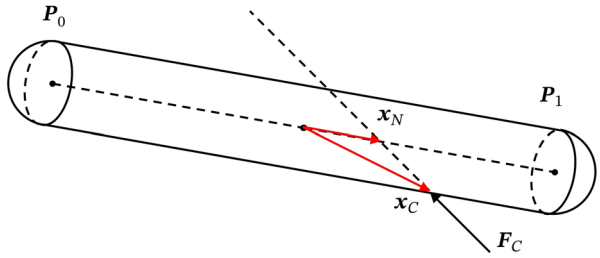

FIG. 12. The relation between $\boldsymbol{x}_N$ and $\boldsymbol{x}_C$.

After we find s_N and t_N on each spherocylinder, we could easily compute the locations of minimal distance $\mathbf{P}(s_N)$ and $\mathbf{Q}(t_N)$. The intersection points of vector $\mathbf{P}(s_N) - \mathbf{Q}(t_N)$ and surfaces of spherocylinders are the collision points. However, for the sake of convenience, it is not necessary to find the exact collision points on surfaces. When computing the stress tensor using Eq. (16), only the torque relative to the center of mass $\boldsymbol{x}_C \times \mathbf{F}^C$ is necessary. Geometrically, it is straightforward to realize that $\boldsymbol{x}_C \times \mathbf{F}^C = \boldsymbol{x}_N \times \mathbf{F}^C$, as shown in Fig. 12. Therefore, there is no need to compute $\boldsymbol{x}_C$.

REFERENCES

[1] P. Bolhuis and D. Frenkel, J. Chem. Phys. **106**, 666 (1997).
[2] S. C. Takatori, W. Yan, and J. F. Brady, Phys. Rev. Lett. **113**, 028103 (2014).
[3] M. Wang and J. F. Brady, Phys. Rev. Lett. **115**, 158301 (2015).
[4] D. W. Rebertus and K. M. Sando, J. Chem. Phys. **67**, 2585 (1977).
[5] B. Snook, L. M. Davidson, J. E. Butler, O. Pouliquen, and E. Guazzelli, J. Fluid Mech. **758**, 486 (2014).
[6] C. S. Campbell and A. Gong, J. Fluid Mech. **164**, 107 (1986).
[7] C. S. Campbell, J. Fluid Mech. **203**, 449 (1989).
[8] Y.-G. Tao, W. K. den Otter, J. T. Padding, J. K. G. Dhont, and W. J. Briels, J. Chem. Phys. **122**, 244903 (2005).
[9] B. Maury, Numer. Math. **102**, 649 (2006).
[10] A. Tasora, D. Negrut, and M. Anitescu, Proc. Inst. Mech. Eng., Part K: J. Multi-Body Dyn. **222**, 315 (2008).
[11] A. Tasora and M. Anitescu, Comput. Methods Appl. Mech. Eng. **200**, 439 (2011).
[12] D. R. Foss and J. F. Brady, J. Rheol. **44**, 629 (2000).
[13] S. Kim and S. J. Karrila, *Microhydrodynamics: Principles and Selected Applications* (Courier Corporation, 2005).
[14] M. Wang and J. F. Brady, J. Comput. Phys. **306**, 443 (2016).
[15] E. Corona, L. Greengard, M. Rachh, and S. Veerapaneni, J. Comput. Phys. **332**, 504 (2017).
[16] E. Corona and S. Veerapaneni, J. Comput. Phys. **362**, 327 (2018).
[17] A.-K. Tornberg and K. Gustavsson, J. Comput. Phys. **215**, 172 (2006).
[18] K. Gustavsson and A.-K. Tornberg, Phys. Fluids **21**, 123301 (2009) (1994-present).
[19] M. Anitescu, F. A. Potra, and D. E. Stewart, Comput. Methods Appl. Mech. Eng. **177**, 183 (1999).
[20] S. Fang, IEEE Trans. Autom. Control **29**, 930 (1984).
[21] S. Niebe and K. Erleben, *Numerical Methods for Linear Complementarity Problems in Physics-Based Animation* (Morgan & Claypool Publishers, San Rafael, California, 2015), oCLC: 904469157.
[22] H. Mazhar, T. Heyn, D. Negrut, and A. Tasora, ACM Trans. Graph. **34**(32), 1 (2015).

[23]Y.-H. Dai and R. Fletcher, Numer. Math. **100**, 21 (2005).
[24]H. Löwen, Phys. Rev. E **50**, 1232 (1994).
[25]S. Delong, F. B. Usabiaga, and A. Donev, J. Chem. Phys. **143**, 144107 (2015).
[26]A. Baskaran and M. C. Marchetti, Phys. Rev. Lett. **101**, 268101 (2008).
[27]F. Ginelli, F. Peruani, M. Bär, and H. Chaté, Phys. Rev. Lett. **104**, 184502 (2010).
[28]S. Orozco-Fuentes and D. Boyer, Phys. Rev. E **88**, 012715 (2013).
[29]H.-S. Kuan, R. Blackwell, L. E. Hough, M. A. Glaser, and M. D. Betterton, Phys. Rev. E **92**, 060501 (2015).
[30]S. Weitz, A. Deutsch, and F. Peruani, Phys. Rev. E **92**, 012322 (2015).
[31]F. Peruani, Eur. Phys. J. Spec. Top. **225**, 2301 (2016).
[32]R. Großmann, F. Peruani, and M. Bär, Phys. Rev. E **94**, 050602 (2016).
[33]G. J. Vroege and H. N. W. Lekkerkerker, Rep. Prog. Phys. **55**, 1241 (1992).
[34]L. Onsager, Ann. N. Y. Acad. Sci. **51**, 627 (1949).
[35]H. Graf and H. Löwen, J. Phys.: Condens. Matter **11**, 1435 (1999).
[36]P. Kraikivski, R. Lipowsky, and J. Kierfeld, Phys. Rev. Lett. **96**, 258103 (2006).
[37]J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[38]P. J. Foster, W. Yan, S. Fürthauer, M. J. Shelley, and D. J. Needleman, New J. Phys. **19**, 125011 (2017).
[39]D. Eberly, "Robust computation of distance between line segments," https://www.geometrictools.com/.