Methods for Stokesian dynamics simulations of nonspherical particles and chains

Ramzi Kutteh*
Department of Physics and Astronomy, Johns Hopkins University, Baltimore, Maryland 21218, USA

(Received 8 August 2003; published 30 January 2004)

The microstructure and properties of suspensions of nonspherical particles are influenced by the specific particle shapes through hydrodynamic interactions. We describe algorithms for Stokesian dynamics simulations of arbitrary shape particles, rigid or flexible, constructed with appropriate constraints among rigid spherical particles whose hydrodynamic mobility is computable by various available schemes, including the one that we recently described [J. Chem. Phys. **112**, 2548 (2000)]. The optimal algorithm also provides for rigid attachment among particles during simulation, by aggregation for example. Its implementation for a system with a general combination of internal coordinate constraints (available in a routine from the author) is tested in simulations of sedimentation of spheroids and chains in bounded and unbounded geometries.

DOI: 10.1103/PhysRevE.69.011406 PACS number(s): 82.70.−y, 83.80.Hj, 02.70.−c

I. INTRODUCTION

Recently we described [1] a force multipole algorithm for computing the hydrodynamic mobility matrix of finite size spherical particles, which accounts for the many-body, long-range, and lubrication effects and, in a fluid bounded by a hard wall, the additional hydrodynamic wall effects. The scheme was used to perform Stokesian dynamics (SD) simulations of colloidal sedimentation near a wall [2] and irreversible deposition onto a wall [3]. However, many industrial and biological processes involve nonspherical suspended particles, such as disk-shaped red blood cells, acicular colloids common in particulate magnetic recording media [4], rodlike fiber suspensions, macromolecules, and colloidal aggregates. Away from sphericality, the influence of particle orientation and shape on the hydrodynamic interactions (HI) leads to dynamical effects not exhibited by suspensions of spheres under similar conditions. We describe an algorithm for performing two types of simulations on suspensions of nonspherical particles. In type (1), particles are defined at the outset and rigid bonding between them forbidden during simulation. Particles could be rigid, such as rigid platelike particles, prolate and oblate spheroids, or flexible, such as deformable particle models and polymer chain models consisting of particles connected by rods. In type (2), rigid bonding between primary particles may occur during simulation, as a result of aggregation for example. The algorithm allows us to perform simulations on spherocylindrical magnetic colloidal dispersions [5], where formation of magnetic agglomerates can influence the dispersion quality, and to study the sedimentation [2] and deposition [3] of nonspherical particles.

Numerical techniques for solving the Stokes equations starting from boundary integral representations of the velocity field, such as the boundary element method [6,7] or the completed double layer solution [8,9], can handle nonspherical particles but are limited to small numbers by their high computational cost, aside from issues of inherent analytical difficulties. Collocation techniques [10–13] are also computationally expensive and confined to problems with special configurations, leaving SD with a mobility algorithm, as the sole continuum-based approach for simulating more realistic systems. While it is feasible to extend analytically our multipole mobility scheme [1] to some nonspherical rigid particle shapes, as was accomplished in a SD extension to prolate spheroids [14], this strategy entails an increased mathematical complexity, must be undertaken for each distinct shape required, and allows for neither flexible particles nor type (2) simulations. Therefore we follow an alternative strategy with three algorithms for SD simulations of arbitrary shape particles, rigid or flexible, formed by imposing suitably chosen holonomic constraints among rigid spherical particles whose hydrodynamic mobility matrix is obtainable by a number of schemes [1,15–20]. By virtue of its decoupled structure, the optimal algorithm handles easily type (2) simulations, during which both the number and forms of the constraints may vary. This algorithm consists of two parts, the first of which is iterative and comprises two stages. Because the first stage resembles formally the SHAKE scheme [21,22] for holonomic constraints in molecular dynamics (MD) simulations, we refer to the entire algorithm as hydrodynamics SHAKE (HSHAKE). As iteration over internal coordinate constraints is more rapidly convergent [22] than that over equivalent distance constraints, HSHAKE was implemented [23] for a general combination of distance, angle, and torsion constraints.

The paper is organized as follows. In Sec. II, we describe the direct approach of incorporating holonomic constraints into SD simulations of spherical particles with HI. A correction method for eliminating the numerical constraint drift in this approach is given in the Appendix. In Sec. III, we describe the alternative undetermined parameters approach and two implementation techniques: a coupled scheme and HSHAKE. The HSHAKE implementation is then described for a system of spherical particles with distance, angle, and torsion constraints, possibly with unconstrained particles. In Sec. IV we validate the methodology underlying HSHAKE and illustrate its performance with simulations of sedimentation of a pair of nonspherical particles, of a nonspherical particle near a tilted wall, and of a chain in unbounded and bounded fluids.

*Electronic address: r.kutteh@mailaps.org

II. SD, CONSTRAINTS, AND THE DIRECT APPROACH

Consider N submicron size spherical non-Brownian particles in an incompressible Newtonian fluid, and assume a low-Reynolds-number regime $(\mathrm{Re}\ll 1)$ so that fluid flow is described by the Stokes or creeping-flow equations [24]. Assume also stick boundary conditions at the particle surfaces and no external flow field, so that fluid motion is due only to that of the particles. The motion of any particle in the fluid creates a flow pattern which causes a drag force on the other particles. These HI between the particles appear in the form of a configuration and geometry dependent mobility tensor $\boldsymbol{\mu}$. The Stokesian equation of motion for this system is given in terms of $\boldsymbol{\mu}$ as

$$\begin{pmatrix} \mathbf{U} \\ \boldsymbol{\Omega} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}^{\mathrm{tt}} & \boldsymbol{\mu}^{\mathrm{tr}} \\ \boldsymbol{\mu}^{\mathrm{rt}} & \boldsymbol{\mu}^{\mathrm{rr}} \end{pmatrix} \begin{pmatrix} \mathbf{F} \\ \mathbf{T} \end{pmatrix}, \tag{1}$$

where $\mathbf{U}$, $\boldsymbol{\Omega}$, $\mathbf{F}$, and $\mathbf{T}$ are $3N$-dimensional vectors containing the linear and angular velocities $\mathbf{U}_i$ and $\boldsymbol{\Omega}_i$, and interparticle and external forces and torques $\mathbf{F}_i$ and $\mathbf{T}_i$, where $i = 1, \ldots, N$. The coefficient matrix is $\boldsymbol{\mu}$, and the $\boldsymbol{\mu}^{\mathrm{ab}}$ are $(3N\times 3N)$ mobility matrices, with superscripts t and r denoting translation and rotation, respectively. We performed SD simulations on such a system [1–3], by taking $\mathbf{T}=\mathbf{0}$, computing at every time step $\mathbf{F}$ and $\boldsymbol{\mu}$, obtaining $\mathbf{U}$ from Eq. (1), and integrating numerically for the coordinates. Consider now the system subject to the l general holonomic constraints,

$$\sigma_k(\mathbf{r})=0 \quad (k=1,\ldots,l), \tag{2}$$

where $\mathbf{r}$ denotes all center of mass coordinates appearing in σ_k. A general system contains constrained and unconstrained coordinates, and we define P_c as the set of labels of the constrained particles. The Stokesian equation of motion for this system is given by

$$\begin{pmatrix} \mathbf{U} \\ \boldsymbol{\Omega} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}^{\mathrm{tt}} & \boldsymbol{\mu}^{\mathrm{tr}} \\ \boldsymbol{\mu}^{\mathrm{rt}} & \boldsymbol{\mu}^{\mathrm{rr}} \end{pmatrix} \begin{pmatrix} \mathbf{F} \\ \mathbf{T} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\mu}^{\mathrm{tt}} & \boldsymbol{\mu}^{\mathrm{tr}} \\ \boldsymbol{\mu}^{\mathrm{rt}} & \boldsymbol{\mu}^{\mathrm{rr}} \end{pmatrix} \begin{pmatrix} \mathbf{F}^c \\ \mathbf{T}^c \end{pmatrix}, \tag{3}$$

where $\mathbf{F}^c$ and $\mathbf{T}^c$ are $3N$-dimensional vectors containing the constraint forces and torques $\mathbf{F}_i^c$ and $\mathbf{T}_i^c$. Absence of orientational degrees of freedom from Eq. (2) implies $\mathbf{T}^c=\mathbf{0}$, and it follows from Eq. (3) that

$$\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}_{11}^{\mathrm{tt}} & \boldsymbol{\mu}_{12}^{\mathrm{tt}} & \boldsymbol{\mu}_{13}^{\mathrm{tt}} & \cdots \\ \boldsymbol{\mu}_{21}^{\mathrm{tt}} & \boldsymbol{\mu}_{22}^{\mathrm{tt}} & \boldsymbol{\mu}_{23}^{\mathrm{tt}} & \cdots \\ \boldsymbol{\mu}_{31}^{\mathrm{tt}} & \boldsymbol{\mu}_{32}^{\mathrm{tt}} & \boldsymbol{\mu}_{33}^{\mathrm{tt}} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \vdots \end{pmatrix} + \begin{pmatrix} \boldsymbol{\mu}_{11}^{\mathrm{tt}} & \boldsymbol{\mu}_{12}^{\mathrm{tt}} & \boldsymbol{\mu}_{13}^{\mathrm{tt}} & \cdots \\ \boldsymbol{\mu}_{21}^{\mathrm{tt}} & \boldsymbol{\mu}_{22}^{\mathrm{tt}} & \boldsymbol{\mu}_{23}^{\mathrm{tt}} & \cdots \\ \boldsymbol{\mu}_{31}^{\mathrm{tt}} & \boldsymbol{\mu}_{32}^{\mathrm{tt}} & \boldsymbol{\mu}_{33}^{\mathrm{tt}} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{F}_1^c \\ \mathbf{F}_2^c \\ \mathbf{F}_3^c \\ \vdots \end{pmatrix}, \tag{4}$$

where the $\boldsymbol{\mu}_{ij}^{\mathrm{tt}}$ are (3×3) mobility tensors. With $\dot{\mathbf{r}}_i \equiv \mathbf{U}_i$, Eq. (4) can be written in dyadic notation

$$\begin{aligned} \dot{\mathbf{r}}_i &= \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \mathbf{F}_j + \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \mathbf{F}_j^c \\ &= \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \mathbf{F}_j + \delta\dot{\boldsymbol{r}}_i \quad (i=1,\ldots,N), \end{aligned} \tag{5}$$

where $\mathbf{F}_j^c = -\Sigma_{k=1}^{l} \lambda_k \nabla_j \sigma_k$, with λ_k a Lagrange multiplier. The $(N+l)$ Eqs. (2) and (5) can be solved numerically for the N $\mathbf{r}$'s and l λ's. Differentiating with respect to time Eq. (2) and inserting Eq. (5) $\forall\, i \in P_c$ gives

$$\sum_{i=1}^{N} [\boldsymbol{\nabla}_i \sigma_k] \cdot \left[\sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \mathbf{F}_j - \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \sum_{k'=1}^{l} \lambda_{k'} [\boldsymbol{\nabla}_j \sigma_{k'}] \right] = 0$$

$$(k=1,\ldots,l). \tag{6}$$

Defining n_k as the set of labels of particles with coordinates appearing in σ_k, Eq. (6) can be recast in matrix notation as $\mathbf{A}\boldsymbol{\lambda}=\mathbf{b}$, where the vector $\boldsymbol{\lambda}$ contains the l λ's, and $b_k = \Sigma_{i \in n_k} \Sigma_{j=1}^{N} [\boldsymbol{\nabla}_i \sigma_k] \cdot \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot \mathbf{F}_j$. For subsequent reference, we define the matrix $\widetilde{A}_{kk'}(\zeta,\theta) \equiv \Sigma_{i \in n_k} \Sigma_{j \in n_{k'}} [\boldsymbol{\nabla}_i \sigma_k](\zeta) \cdot \boldsymbol{\mu}_{ij}^{\mathrm{tt}}(\theta) \cdot [\boldsymbol{\nabla}_j \sigma_{k'}](\theta)$ $(k,k'=1,\ldots,l)$. We shall be concerned in this paper with two special cases of the arguments ζ and θ. In the first case, θ is a time argument and $\zeta=\theta$. In this case $\widetilde{A}_{kk'}(\theta,\theta)$ implies that all terms in its expression are evaluated at the same time step, or equivalently using the final coordinates from that time step. The symmetric matrix $\mathbf{A}$ is given by $A_{kk'} = \widetilde{A}_{kk'}(\theta,\theta)$. In the second case, θ is again a time argument, but ζ is a coordinate argument. We shall encounter this case in Secs. III A and III B, but as an example, $\widetilde{A}_{kk'}(\mathbf{r}'(t_0+\delta t),t_0)$ implies that terms in its expression corresponding to the first argument are evaluated using the coordinates $\mathbf{r}'(t_0+\delta t)$ (defined in Sec. III) while those corresponding to the second argument are evaluated using the final coordinates from the time step t_0. The λ's can be obtained with standard numerical library routines, $\mathbf{F}^c$ computed from its expression above, and Eq. (5) numerically integrated using $\mathbf{F}^c$ and $\mathbf{F}$, to generate the $\mathbf{r}$'s. From Eq. (5) we write

$$\begin{aligned} \delta\dot{\boldsymbol{r}}_i &= -\sum_{k=1}^{l} \lambda_k \sum_{j \in n_k} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot [\boldsymbol{\nabla}_j \sigma_k] \\ &= -\sum_{k \in C_i} \lambda_k \sum_{j \in n_k} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot [\boldsymbol{\nabla}_j \sigma_k] - \sum_{k \notin C_i} \lambda_k \sum_{j \in n_k} \boldsymbol{\mu}_{ij}^{\mathrm{tt}} \cdot [\boldsymbol{\nabla}_j \sigma_k] \\ &= \delta\dot{\boldsymbol{r}}_i^{\mathrm{I}} + \delta\dot{\boldsymbol{r}}_i^{\mathrm{II}} \quad (\forall\, i \in P_c), \end{aligned} \tag{7}$$

where C_i is defined as the set of labels of constraints containing $\mathbf{r}_i$ $\forall\, i \in P_c$, and superscripts I and II denote the contributions from the constraints including and excluding $\mathbf{r}_i$, respectively. Note that $\forall\, i \in P_c$ (a1) $\delta\dot{\boldsymbol{r}}_i^{\mathrm{I}}$ does not arise solely from $\mathbf{F}_i^c$ and (a2) $\delta\dot{\boldsymbol{r}}_i^{\mathrm{II}} \neq \mathbf{0}$, and (a3) $\forall\, i \notin P_c$, $\delta\dot{\boldsymbol{r}}_i \neq \mathbf{0}$. The counterpart MD system of N atomic particles with coordinates $\mathbf{r}$ subject to Eq. (2) obeys the equations $\ddot{\mathbf{r}}_i=(1/m_i)(\mathbf{F}_i+\mathbf{F}_i^c)=(1/m_i)\mathbf{F}_i+\delta\ddot{\mathbf{r}}_i \quad (i=1,\ldots,N)$, where $\mathbf{F}_i^c=-\Sigma_{k=1}^{l}\lambda_k\boldsymbol{\nabla}_i\sigma_k$ and m_i is the mass of particle i. The MD counterpart of Eq. (7) is given by $\delta\ddot{\mathbf{r}}_i=-(1/m_i)\Sigma_{k\in C_i}\lambda_k\boldsymbol{\nabla}_i\sigma_k=\delta\ddot{\mathbf{r}}_i^{\mathrm{I}}$ $(\forall\, i\in P_c)$, where we note that $\forall\, i\in P_c$ $(\overline{\text{a1}})$ $\delta\ddot{\mathbf{r}}_i^{\mathrm{I}}$ arises solely from $\mathbf{F}_i^c$ and $(\overline{\text{a2}})$ $\delta\ddot{\mathbf{r}}_i^{\mathrm{II}}=\mathbf{0}$, and $(\overline{\text{a3}})$ $\forall\, i\notin P_c$, $\delta\ddot{\mathbf{r}}_i=\mathbf{0}$. The formal differences between this SD method and the MD direct method [22] result from the contrast between (a1,a2,a3) and $(\overline{\text{a1}},\overline{\text{a2}},\overline{\text{a3}})$.

A numerical constraint method must [25,26] (A) ensure that the constraints are satisfied during simulation at least within a desired tolerance, and (B) given an integration algorithm, avoid introducing through (A) numerical truncation errors into the coordinate trajectories of an order in the time step lower than that present in the direct method (i.e., present in the integration algorithm). Such additional numerical errors to those in the direct method artificially alter the constrained dynamics, as illustrated in the simulations of Secs. IV C and IV D. Incorporating (a1), (a2), and (a3) in the direct approach ensures that $\delta\dot{\mathbf{r}}_i$ in Eq. (5) is complete $\forall\, i$, and consequently that the direct approach satisfies requirement (B). Absence of a single constraint contribution from $\delta\dot{\mathbf{r}}_i$, for a single i, would have implied violation of requirement (B). The proof of this implication is analogous to that in the error analysis of Sec. III C. As in the MD case [22], application of this direct method in SD simulations leads to constraints that drift progressively from their constraint values, mainly because of the truncation error in integration algorithms. Hence this direct method violates requirement (A). Accordingly, we describe in the Appendix a correction method [27] for the constraint drift, more conveniently discussed after the material in Sec. III has been presented, and show that the [direct + correction] method satisfies both requirements (A) and (B). Next, we present an alternative constraint method and two techniques of implementation, which also satisfy requirements (A) and (B). The need for this alternative method is discussed in Sec. III B, where we compare the [direct+correction] method and the two techniques.

III. THE UNDETERMINED PARAMETERS APPROACH

A truncated Taylor series solution of Eq. (5) can be written as

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\lambda(t_0))&=\mathbf{r}_i(t_0)+\delta t\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot\mathbf{F}_j(t_0)\\&\quad-\delta t\sum_{k=1}^{l}\lambda_k(t_0)\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\\&=\mathbf{r}_i'(t_0+\delta t)+\delta\mathbf{r}_i(t_0+\delta t,\lambda(t_0))\\&\quad(i=1,\ldots,N),\end{aligned}\tag{8}$$

where $\delta\mathbf{r}_i(t_0+\delta t,\lambda(t_0))$ denotes the term containing the λ's. To compute the coordinates, first the $\mathbf{r}'(t_0+\delta t)$ are evaluated by numerically integrating Eq. (5) using $\mathbf{F}$ only (i.e., $\delta\dot{\mathbf{r}}_i=\mathbf{0}$). Two integrators were used in the simulations of Sec. IV: an Euler integrator

$$\begin{aligned}\mathbf{r}_i'(t_0+\delta t)&=\mathbf{r}_i(t_0)+\delta t\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot\mathbf{F}_j(t_0)+O(\delta t^2)\\&\quad(i=1,\ldots,N),\end{aligned}\tag{9}$$

and a fourth order Runge-Kutta algorithm

$$\begin{aligned}\mathbf{r}_i'(t_0+\delta t)&=\mathbf{r}_i(t_0)+\tfrac{1}{6}(\Delta\mathbf{r}_i^{(1)}+2\Delta\mathbf{r}_i^{(2)}+2\Delta\mathbf{r}_i^{(3)}+\Delta\mathbf{r}_i^{(4)})\\&\quad+O(\delta t^5)\quad(i=1,\ldots,N),\end{aligned}\tag{10}$$

where

$$\begin{aligned}\Delta\mathbf{r}_i^{(1)}&=\delta t\dot{\mathbf{r}}_i[\mathbf{r}(t_0)]=\delta t\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}[\mathbf{r}(t_0)]\cdot\mathbf{F}_j[\mathbf{r}(t_0)],\\\Delta\mathbf{r}_i^{(n+1)}&=\delta t\dot{\mathbf{r}}_i[\mathbf{r}(t_0)+c_n\Delta\mathbf{r}^{(n)}]\\&=\delta t\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}[\mathbf{r}(t_0)+c_n\Delta\mathbf{r}^{(n)}]\cdot\mathbf{F}_j[\mathbf{r}(t_0)+c_n\Delta\mathbf{r}^{(n)}],\\&\qquad n=1,2,3;\ c_1=c_2=\frac{1}{2},\ c_3=1,\end{aligned}\tag{11}$$

with square brackets delimiting arguments. Note that the Runge-Kutta integrator requires four evaluations per time step of the computationally expensive mobility matrix, whereas the Euler scheme requires just one. Second, the $\delta\mathbf{r}$ are chosen to satisfy the constraints. Replacing the $\lambda(t_0)$'s by undetermined parameters γ's, Eq. (8) becomes

$$\begin{aligned}&\mathbf{r}_i(t_0+\delta t,\gamma)\\&\quad=\mathbf{r}_i'(t_0+\delta t)+\delta\mathbf{r}_i(t_0+\delta t,\gamma)\\&\quad=\mathbf{r}_i'(t_0+\delta t)-\delta t\\&\qquad\times\sum_{k=1}^{l}\gamma_k\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\quad(i=1,\ldots,N),\end{aligned}\tag{12}$$

where the γ's are chosen to satisfy Eq. (2). Therefore, inserting Eq. (12) $\forall\, i\in P_c$ into Eq. (2) gives

$$\begin{aligned}&\sigma_k(\mathbf{r}(t_0+\delta t,\gamma))\\&\quad=\sigma_k\left(\mathbf{r}'(t_0+\delta t)-\delta t\sum_{k'=1}^{l}\gamma_{k'}\sum_{j=1}^{N}\boldsymbol{\mu}_{\{\}j}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\right)\\&\quad=0\quad(k=1,\ldots,l),\end{aligned}\tag{13}$$

where the first subscript in $\boldsymbol{\mu}_{\{\}j}^{\mathrm{tt}}$ runs over n_k. The l γ's obtained by solving this generally nonlinear system are substituted into Eq. (12) to provide the final coordinates. By analogy with Eq. (7), from Eq. (12) we write

$$\delta\mathbf{r}_i(t_0+\delta t,\gamma)=-\delta t\sum_{k\in C_i}\gamma_k\sum_{j\in n_k}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)$$
$$-\delta t\sum_{k\notin C_i}\gamma_k\sum_{j\in n_k}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)$$
$$=\delta\mathbf{r}_i^{\mathrm{I}}+\delta\mathbf{r}_i^{\mathrm{II}}\qquad(\forall\ i\in P_c).\tag{14}$$

Note that $\forall\ i\in P_c$ (b1) $\delta\mathbf{r}_i^{\mathrm{I}}$ does not arise solely from $\mathbf{F}_i^c$ and (b2) $\delta\mathbf{r}_i^{\mathrm{II}}\neq 0$, and (b3) $\forall\ i\notin P_c$, $\delta\mathbf{r}_i\neq 0$. When the undetermined parameters method is applied to the MD system, the counterpart of Eq. (14) is given by [22] $\delta\mathbf{r}_i(t_0+\delta t,\gamma)=-([\delta t]^2/m_i)\Sigma_{k\in C_i}\gamma_k[\boldsymbol{\nabla}_i\sigma_k](t_0)=\delta\mathbf{r}_i^{\mathrm{I}}$ $(\forall\ i\in P_c)$, where we note that $\forall\ i\in P_c$ $(\overline{\mathrm{b1}})$ $\delta\mathbf{r}_i^{\mathrm{I}}$ arises solely from $\mathbf{F}_i^c$ and $(\overline{\mathrm{b2}})$ $\delta\mathbf{r}_i^{\mathrm{II}}=0$, and $(\overline{\mathrm{b3}})$ $\forall\ i\notin P_c$, $\delta\mathbf{r}_i=0$. The contrast between (b1,b2,b3) and $(\overline{\mathrm{b1}},\overline{\mathrm{b2}},\overline{\mathrm{b3}})$ follows directly from that between (a1,a2,a3) and $(\overline{\mathrm{a1}},\overline{\mathrm{a2}},\overline{\mathrm{a3}})$. The undetermined parameters approach satisfies requirement (A) by construction, and incorporating (b1), (b2), and (b3) ensures that $\delta\mathbf{r}_i(t_0+\delta t,\gamma)$ in Eq. (12) is complete $\forall\ i$, which in turn implies that the approach satisfies requirement (B), as shown by the following error analysis.

The integration algorithm in the direct method can be represented by the expansion in Eq. (8). Assuming henceforth that the integration algorithm has a local error in the coordinates of $O(\delta t^{m+1})$, the $\{\delta t\,\lambda_k(t_0)\Sigma_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\}$ term in Eq. (8) is of $O(\delta t^m)$. If $[\Sigma_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)]$ is of $O(\delta t^a)$, then $\lambda(t_0)$ is of $O(\delta t^{m-a-1})$, or

$$\lambda_k(t_0)=\beta_k+O(\delta t^{m-a}),\tag{15}$$

where the β's are some estimated values of the $\lambda(t_0)$'s. In the undetermined parameters approach, the $\lambda(t_0)$'s are replaced by the γ's, as described before. Accordingly, replacing the β's by the γ's, Eq. (15) becomes $\gamma_k=\lambda_k(t_0)+O(\delta t^{m-a})$, which after insertion back into Eq. (12) gives the coordinates from the method of undetermined parameters as

$$\mathbf{r}_i(t_0+\delta t,\gamma)=\mathbf{r}_i'(t_0+\delta t)$$
$$-\delta t\sum_{k=1}^{l}\lambda_k(t_0)\sum_{j=1}^{N}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)$$
$$+O(\delta t^{m+1})\quad(i=1,\ldots,N).\tag{16}$$

By comparing Eq. (16) with Eq. (8) we can write

$$\mathbf{r}_i[\text{undetermined parameters}]=\mathbf{r}_i[\text{direct}]+O(\delta t^{m+1}),$$
$$(i=1,\ldots,N).\tag{17}$$

From the assumed order of error $O(\delta t^{m+1})$ of the coordinates in the integration algorithm, we can write

$$\mathbf{r}_i[\text{direct}]=\mathbf{r}_i[\text{exact}]+O(\delta t^{m+1})\qquad(i=1,\ldots,N),\tag{18}$$

where $\mathbf{r}_i[\text{exact}]$ is the trajectory obtained ideally from an exact solution of the equations of motion. Inserting Eq. (18) into Eq. (17) gives

$$\mathbf{r}_i[\text{method}]=\mathbf{r}_i[\text{exact}]+O(\delta t^p)\qquad(i=1,\ldots,N),\tag{19}$$

where method="undetermined parameters" and $p=m+1$. Comparison with Eq. (18) shows that the method of undetermined parameters satisfies requirement (B). Two techniques for computing the γ's are described next.

A. Coupled technique

This technique consists of two parts, the first is iterative and corrects the constrained coordinates, the second is noniterative and corrects the unconstrained coordinates. To solve Eq. (13) for the γ's, we Taylor expand in the first part $\sigma_k(\mathbf{r}(t_0+\delta t,\gamma))$ about $\mathbf{r}'(t_0+\delta t)$ and recast the expansions in the matrix form

$$\boldsymbol{\sigma}+\mathbf{L}\boldsymbol{\gamma}+\mathbf{Q}+\cdots=\mathbf{0},\tag{20}$$

where the vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$ contain the l $\sigma_k(\mathbf{r}'(t_0+\delta t))$ and $\gamma_{k'}$, respectively, $L_{kk'}=L_{kk'}(\mathbf{r}'(t_0+\delta t),t_0)$ with $L_{kk'}(\zeta,\theta)\equiv-\delta t\tilde{A}_{kk'}(\zeta,\theta)$, and $\mathbf{Q}$ is quadratic in the γ's,

$$Q_k=\sum_{i,j\in n_k}\frac{[\delta t]^2}{2}\sum_{k',k''=1}^{l}\gamma_{k'}\gamma_{k''}\left[\sum_{j'\in n_{k'}}\boldsymbol{\mu}_{ij'}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_{j'}\sigma_{k'}](t_0)\right]\left[\sum_{j''\in n_{k''}}\boldsymbol{\mu}_{jj''}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_{j''}\sigma_{k''}](t_0)\right]:[\boldsymbol{\nabla}_i\boldsymbol{\nabla}_j\sigma_k](\mathbf{r}'(t_0+\delta t))$$
$$(k=1,\ldots,l).\tag{21}$$

Neglecting nonlinear terms, Eq. (20) is solved initially for $\gamma^{(0)}$'s, which are then inserted into Eq. (21), and Eq. (20) solved for $\gamma^{(1)}$'s, and so on until the γ's converge within desired tolerances on the constraints in Eq. (13). Assuming convergence at some time step after r iterations, we then have

$$\mathbf{r}_i(t_0+\delta t,\gamma)=\mathbf{r}'_i(t_0+\delta t)-\delta t\sum_{k'=1}^{l}\gamma_{k'}\times\sum_{j\in n_{k'}}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\quad(\forall\, i\in P_c),\tag{22}$$

where $\gamma_{k'}=\gamma^{(r)}_{k'}$ $(k'=1,\ldots,l)$. Using these γ's, the unconstrained $\mathbf{r}_i$ are corrected in the second noniterative part

$$\mathbf{r}_i(t_0+\delta t,\gamma)=\mathbf{r}'_i(t_0+\delta t)-\delta t\sum_{k'=1}^{l}\gamma_{k'}\times\sum_{j\in n_{k'}}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\quad(\forall\, i\notin P_c).\tag{23}$$

Note that the first part incorporates (b1) and (b2), and the second (b3). The formal differences between this technique and the MD "matrix technique" [22] result from the contrast between (b1,b2,b3) and $(\overline{\mathrm{b1}},\overline{\mathrm{b2}},\overline{\mathrm{b3}})$. Next, we describe an alternative decoupled scheme and discuss its advantages.

B. Decoupled scheme: HSHAKE

This technique also consists of an iterative first part which corrects the constrained coordinates, and a noniterative second part which corrects the unconstrained coordinates. Each iteration of the first part comprises two stages. Because $\delta\mathbf{r}^{\mathrm{I}}_i$ in Eq. (14) contains only contributions from constraints involving $\mathbf{r}_i$, it is accumulated in stage I by decoupling these constraints and successively satisfying each with corrections to $\mathbf{r}_i$. We consider a certain $\sigma_k(\mathbf{r})$ and rewrite Eq. (14) as

$$\begin{aligned}\delta\mathbf{r}_i(t_0+\delta t,\gamma)={}&-\delta t\,\gamma_k\sum_{j\in n_k}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\\&-\delta t\sum_{k'\in C_i,k'\neq k}\gamma_{k'}\\&\times\sum_{j\in n_{k'}}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)-\delta t\sum_{k'\notin C_i}\gamma_{k'}\\&\times\sum_{j\in n_{k'}}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\={}&\delta\mathbf{r}^{\mathrm{I}}_i+\delta\mathbf{r}^{\mathrm{II}}_i\quad(\forall\, i\in n_k),\end{aligned}\tag{24}$$

where $\delta\mathbf{r}^{\mathrm{II}}_i$ denotes the last of the three terms. To satisfy separately $\sigma_k(\mathbf{r})=0$, only the first term of Eq. (24) is required:

$$\delta\mathbf{r}_i(t_0+\delta t,\gamma_k)=-\delta t\,\gamma_k\sum_{j\in n_k}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\quad(\forall\, i\in n_k).\tag{25}$$

During stage I of iteration q, HSHAKE successively selects each constraint $\sigma_k=0$ and corrects its positions according to Eq. (25), to satisfy it. With $\gamma_k(q)$ denoting the γ_k computed in iteration q, and $\mathbf{r}^{\mathrm{old}}(t_0+\delta t)$ including all preceding changes, the new positions when considering some σ_k,

$$\begin{aligned}\mathbf{r}^{\mathrm{new}}_i(t_0+\delta t)&=\mathbf{r}^{\mathrm{old}}_i(t_0+\delta t)+\delta\mathbf{r}_i(t_0+\delta t,\gamma_k(q))\\&=\mathbf{r}^{\mathrm{old}}_i(t_0+\delta t)-\delta t\,\gamma_k(q)\\&\quad\times\sum_{j\in n_k}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\quad(\forall\, i\in n_k),\end{aligned}\tag{26}$$

should satisfy

$$\begin{aligned}\sigma_k(\mathbf{r}^{\mathrm{new}}(t_0+\delta t))&=\sigma_k\Bigg(\mathbf{r}^{\mathrm{old}}(t_0+\delta t)-\delta t\,\gamma_k(q)\\&\quad\times\sum_{j\in n_k}\boldsymbol{\mu}^{\mathrm{tt}}_{\{\}j}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\Bigg)\\&=0,\end{aligned}\tag{27}$$

which is generally nonlinear in $\gamma_k(q)$. Taylor expanding about $\mathbf{r}^{\mathrm{old}}(t_0+\delta t)$,

$$\begin{aligned}&\sigma_k\Bigg(\mathbf{r}^{\mathrm{old}}(t_0+\delta t)-\delta t\,\gamma_k(q)\sum_{j\in n_k}\boldsymbol{\mu}^{\mathrm{tt}}_{\{\}j}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_k](t_0)\Bigg)\\&\quad=\sigma_k(\mathbf{r}^{\mathrm{old}}(t_0+\delta t))+\gamma_k(q)L_{kk}(\mathbf{r}^{\mathrm{old}}(t_0+\delta t),t_0)+\cdots=0,\end{aligned}\tag{28}$$

and neglecting terms nonlinear in $\gamma_k(q)$, yields

$$\gamma_k(q)=-\frac{\sigma_k(\mathbf{r}^{\mathrm{old}}(t_0+\delta t))}{L_{kk}(\mathbf{r}^{\mathrm{old}}(t_0+\delta t),t_0)}.\tag{29}$$

Stage I of iteration q consists therefore of successively selecting each constraint and applying to it Eqs. (26) and (29). Using the $\gamma_k(q)$ $(k=1,\ldots,l)$ computed in this stage, the coordinates corrected by Eq. (26) are corrected in stage II by

$$\mathbf{r}^{\mathrm{new}}_i(t_0+\delta t)=\mathbf{r}^{\mathrm{old}}_i(t_0+\delta t)-\delta t\sum_{k'\notin C_i}\gamma_{k'}(q)\times\sum_{j\in n_{k'}}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\quad(\forall\, i\in P_c).\tag{30}$$

Additional iterations, each consisting of stages I and II, are performed until all constraints are satisfied within desired tolerances. Stages I and II accumulate $\delta\mathbf{r}^{\mathrm{I}}_i$ and $\delta\mathbf{r}^{\mathrm{II}}_i$ $\forall\, i\in P_c$, respectively, and iterating compensates for the decoupling of $\delta\mathbf{r}^{\mathrm{I}}_i$, the separate inclusion of $\delta\mathbf{r}^{\mathrm{II}}_i$ in stage II, and the linearization of Eq. (28). At convergence we again have Eq. (22) but with the γ's given here by

TABLE I. Structure and properties of SHAKE, HSHAKE, and SHAKE*. From the last two rows, SHAKE and HSHAKE satisfy requirement (B) but SHAKE* does not.

	HSHAKE (SD)	SHAKE(MD)	SHAKE*(SD)
$\delta \mathbf{r}_i^{\mathrm{I}}$ $(\forall\, i \in P_c)$	Stage I of first part, Eqs. (26) and (29)	Entire iteration, Eq. (32)	Entire iteration, Eqs. (26) and (29)
$\delta \mathbf{r}_i^{\mathrm{II}}$ $(\forall\, i \in P_c)$	Stage II of first part, Eq. (30)	0	0
$\delta \mathbf{r}_i$ $(\forall\, i \notin P_c)$	Second part, Eq. (23)	0	0
Extra errors in $\mathbf{r}_i, \forall\, i \in P_c$	No	No	Yes
Extra errors in $\mathbf{r}_i, \forall\, i \notin P_c$	No	No	Yes

$$\gamma_{k'} = \sum_{q=1}^{r} \gamma_{k'}(q) \quad (k'=1,\dots,l). \tag{31}$$

In the second noniterative part of HSHAKE, the unconstrained $\mathbf{r}_i$ are corrected according to Eq. (23) with the γ's obtained by Eq. (31). Stages I and II of the first part incorporate (b1) and (b2), respectively, the second part incorporates (b3), and combining Eqs. (22) and (23) yields Eq. (12). Hence $\delta \mathbf{r}_i(t_0 + \delta t, \gamma)$ generated by either the coupled technique or HSHAKE is complete $\forall\, i$, and consequently both techniques satisfy requirement (B). Because they satisfy requirement (A) by construction, they are true implementations of the undetermined parameters approach.

The formal differences between HSHAKE and SHAKE, noted in Table I, result from the contrast between (b1,b2,b3) and $(\overline{\text{b1}},\overline{\text{b2}},\overline{\text{b3}})$. In particular, due to the contrast between (b1) and $(\overline{\text{b1}})$, the correction in stage I of a HSHAKE iteration, Eqs. (26) and (29), is more involved than that of a SHAKE iteration [22]

$$\mathbf{r}_i^{\mathrm{new}}(t_0+\delta t) = \mathbf{r}_i^{\mathrm{old}}(t_0+\delta t) - \frac{\sigma_k(\mathbf{r}^{\mathrm{old}}(t_0+\delta t))[\boldsymbol{\nabla}_i \sigma_k](t_0)}{\sum_{i \in n_k} [\boldsymbol{\nabla}_i \sigma_k](\mathbf{r}^{\mathrm{old}}(t_0+\delta t)) \cdot [\boldsymbol{\nabla}_i \sigma_k](t_0)} \quad (\forall\, i \in n_k). \tag{32}$$

In the MD limit, Eqs. (26) and (29) reduce identically to Eq. (32), stage II and the second part of HSHAKE vanish, and HSHAKE reduces to SHAKE. Like SHAKE, it also satisfies requirements (A) and (B) and hence is an extension of SHAKE to SD simulations with HI, as summarized in Table I.

We favor HSHAKE over the [direct+correction] method and the coupled technique for two reasons. First, numerical solution of $\mathbf{A}\boldsymbol{\lambda} = \mathbf{b}$ or Eq. (20) every time step becomes costly for a large number of constraints. The drift correction scheme (e.g., first part of HSHAKE) operates on coordinates from the direct method, which incorporate already the effect of constraints, and therefore it converges faster than HSHAKE. To satisfy the constraints, the correction scheme effectively adjusts the actual constraint forces from the direct method, while HSHAKE computes directly the approximate constraint forces. This difference in convergence rates occurs also in MD simulations [27]. Despite its slower convergence, HSHAKE is more efficient than the [direct+correction] method because of the extra cost of computing the λ's. Second, application of the [direct+correction] method or the coupled technique to type (2) simulations, where the constraints may vary during runtime, requires construction of a new and usually different size matrix $\mathbf{A}$ or $\mathbf{L}$, respectively, at each time step involving change in constraints. Such change however is generally random, requiring a complex procedure to repeatedly construct a corresponding $\mathbf{A}$ or $\mathbf{L}$. In contrast, HSHAKE easily accommodates changes in constraints during simulation because of its decoupled structure. This contrast occurs also in colloidal deposition simulations [3].

C. On alternative algorithms to HSHAKE

Consider a hypothetical algorithm alternative to HSHAKE [in the sense that it also satisfies requirement (A)], which we denote by SHAKE* and which, in contrast with HSHAKE, consists of a single iterative part comprising only one stage. Specifically, in iteration q, SHAKE* selects every constraint and corrects its coordinates according to Eqs. (26) and (29), with γ's denoted here by ξ's. Additional iterations are performed until all constraints are satisfied within desired tolerances. Iteration to convergence hence accumulates only $\delta \mathbf{r}_i^{\mathrm{I}}$ $\forall\, i \in P_c$. At convergence, we therefore have

$$\mathbf{r}_i(t_0+\delta t, \xi) = \mathbf{r}_i'(t_0+\delta t) - \delta t \sum_{k' \in C_i} \xi_{k'} \sum_{j \in n_{k'}} \boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0) \cdot [\boldsymbol{\nabla}_j \sigma_{k'}](t_0) \quad (\forall\, i \in P_c,\ \text{SHAKE}^\star), \tag{33}$$

where $\xi_{k'} = \Sigma_{q=1}^{r} \xi_{k'}(q)$. The unconstrained coordinates are not corrected by SHAKE*, thus

$$\mathbf{r}_i(t_0+\delta t) = \mathbf{r}_i'(t_0+\delta t) \quad (\forall\, i \notin P_c, \text{SHAKE}^\star). \tag{34}$$

From Table I we see that in addition to (b1), SHAKE* incorporates $(\overline{\text{b2}})$ and $(\overline{\text{b3}})$ rather than (b2) and (b3), leading to an incomplete $\delta \mathbf{r}_i(t_0+\delta t, \xi) \forall\, i$, as seen by combining Eqs. (33) and (34) and comparing them with Eq. (12), which implies that SHAKE* violates requirement (B), as shown by the following error analysis.

For the N spheres subject to Eq. (2) and with an arbitrary $\boldsymbol{\mu}$, SHAKE* gives from Eq. (33),

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\xi)=\mathbf{r}_i'(t_0+\delta t)&-\delta t\sum_{k'=1}^{l}\xi_{k'}\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&+\delta t\sum_{k'\notin C_i}\xi_{k'}\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&(\forall\, i\in P_c).\end{aligned}\tag{35}$$

In SHAKE*, the $\lambda(t_0)$'s are replaced by the ξ's. Replacing thus the β's by the ξ's, Eq. (15) becomes $\xi_k=\lambda_k(t_0)+O(\delta t^{m-a})$, which after insertion back into Eq. (35) gives

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\xi)=\mathbf{r}_i'(t_0+\delta t)&-\delta t\sum_{k'=1}^{l}\lambda_{k'}(t_0)\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&+\delta t\sum_{k'\notin C_i}\lambda_{k'}(t_0)\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&+O(\delta t^{m+1})\quad(\forall\, i\in P_c).\end{aligned}\tag{36}$$

Comparing Eq. (36) with Eq. (8) $\forall\, i\in P_c$, we can write

$$\mathbf{r}_i[\text{SHAKE}^\star]=\mathbf{r}_i[\text{direct}]+O(\delta t^m)\quad(\forall\, i\in P_c),\tag{37}$$

where the $O(\delta t^{m+1})$ term in Eq. (36) has been dominated by the $O(\delta t^m)$ term preceding it. From Eq. (34), the unconstrained coordinates are unmodified by SHAKE*, hence

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t)=\mathbf{r}_i'(t_0+\delta t)&-\delta t\sum_{k'=1}^{l}\lambda_{k'}(t_0)\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&+\delta t\sum_{k'=1}^{l}\lambda_{k'}(t_0)\sum_{j\in n_{k'}}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\sigma_{k'}](t_0)\\&(\forall\, i\notin P_c).\end{aligned}\tag{38}$$

Comparing Eq. (38) with Eq. (8) $\forall\, i\notin P_c$, we can write

$$\mathbf{r}_i[\text{SHAKE}^\star]=\mathbf{r}_i[\text{direct}]+O(\delta t^m)\quad(\forall\, i\notin P_c).\tag{39}$$

Inserting Eq. (18) $\forall\, i\in P_c$ into Eq. (37) and $\forall\, i\notin P_c$ into Eq. (39), and combining the pair gives Eq. (19) with method = "SHAKE*" and $p=m$. Comparison with Eq. (18) shows that SHAKE* violates requirement (B) for constrained and unconstrained particles.

The purpose of the above discussion is not purely academic. Indeed, the SHAKE-HI algorithm [28] for enforcing distance constraints among spherical particles in Brownian dynamics (BD) simulations with Rotne-Prager [18] HI, can also be applied to SD simulations because it satisfies requirement (A). In the case of SD simulations, SHAKE-HI is a special case of SHAKE* however, and hence violates requirement (B). SHAKE-HI has already been shown [29] to be inaccurate for BD simulations, and as we just noted, should not be used for SD simulations either as it leads to incorrect trajectories, as illustrated in the simulations of Secs. IV C and IV D.

D. HSHAKE for internal coordinate constraints

A system of N spherical particles is subject to Eq. (2), with distance, angle, and torsional constraints [23], labeled by the sets C^{d}, C^{a}, and C^{t}, respectively. For distance constraints, Eq. (2) reduces to

$$\sigma_k(\mathbf{r})=[\mathbf{r}_{ij}(t)]^2-d_{ij}^2=0\quad(\forall\, k\in C^{\mathrm{d}}),\tag{40}$$

where d_{ij} is the constant separation between particles i and j, and $\mathbf{r}_{ij}\equiv\mathbf{r}_i-\mathbf{r}_j$. By means of Eq. (40), Eqs. (26) and (29) reduce to

$$\begin{aligned}\mathbf{r}_j^{\mathrm{new}}(t_0+\delta t)&=\mathbf{r}_j^{\mathrm{old}}(t_0+\delta t)-2\,\delta t\,\gamma_k(q)\,\bar{\boldsymbol{\mu}}_{j,ji}^{\mathrm{tt}}(t_0)\cdot\mathbf{r}_{ji}(t_0),\\\mathbf{r}_i^{\mathrm{new}}(t_0+\delta t)&=\mathbf{r}_i^{\mathrm{old}}(t_0+\delta t)-2\,\delta t\,\gamma_k(q)\,\bar{\boldsymbol{\mu}}_{i,ij}^{\mathrm{tt}}(t_0)\cdot\mathbf{r}_{ij}(t_0),\end{aligned}\tag{41}$$

and

$$\gamma_k(q)=[\delta t]^{-1}\frac{[\mathbf{r}_{ji}^{\mathrm{old}}(t_0+\delta t)]^2-d_{ij}^2}{4\mathbf{r}_{ji}^{\mathrm{old}}(t_0+\delta t)\cdot[\bar{\boldsymbol{\mu}}_{i,ij}^{\mathrm{tt}}(t_0)+\bar{\boldsymbol{\mu}}_{j,ji}^{\mathrm{tt}}(t_0)]\cdot\mathbf{r}_{ji}(t_0)},\tag{42}$$

respectively, where $\bar{\boldsymbol{\mu}}_{i,jk}^{\mathrm{tt}}(t_0)\equiv\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)-\boldsymbol{\mu}_{ik}^{\mathrm{tt}}(t_0)$. For angle constraints, Eq. (2) becomes

$$\sigma_k(\mathbf{r})=\phi_{abc}(\mathbf{r})-\alpha_{abc}=0\quad(\forall\, k\in C^{\mathrm{a}}),\tag{43}$$

where $\phi_{abc}\equiv\arccos(\hat{\mathbf{r}}_{ab}\cdot\hat{\mathbf{r}}_{cb})$ is the angle at b formed by particles a, b, and c, $\hat{\mathbf{r}}_{ab}\equiv\mathbf{r}_{ab}/|\mathbf{r}_{ab}|$, and α_{abc} is the constant constraint angle. Using Eq. (43), Eqs. (26) and (29) become

$$\begin{aligned}\mathbf{r}_i^{\mathrm{new}}(t_0+\delta t)=\mathbf{r}_i^{\mathrm{old}}(t_0+\delta t)&-\delta t\,\gamma_k(q)\\&\times\sum_{j=a,b,c}\boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j\phi_{abc}](t_0)\quad(i=a,b,c),\end{aligned}\tag{44}$$

and

$$\gamma_k(q)=[\delta t]^{-1}\frac{\phi_{abc}(\mathbf{r}^{\mathrm{old}}(t_0+\delta t))-\alpha_{abc}}{\sum_{i,j=a,b,c}[\boldsymbol{\nabla}_i\phi_{abc}](\mathbf{r}^{\mathrm{old}}(t_0+\delta t))\cdot\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\phi_{abc}](t_0)}, \tag{45}$$

respectively. For torsional constraints, Eq. (2) becomes

$$\sigma_k(\mathbf{r})=\tau_{abcd}(\mathbf{r})-\beta_{abcd}=0 \quad (\forall\, k\in C^{\mathrm{t}}), \tag{46}$$

where

$$\tau_{abcd} \equiv \arccos[(\hat{\mathbf{r}}_{ab}\times\hat{\mathbf{r}}_{cb})\cdot(\hat{\mathbf{r}}_{bc}\times\hat{\mathbf{r}}_{dc})/(\sin\phi_{abc}\sin\phi_{bcd})]$$

is the dihedral angle formed by particles *a*, *b*, *c*, and *d*, and β_{abcd} is its constant constraint value. By means of Eq. (46), Eqs. (26) and (29) reduce to

$$\mathbf{r}_i^{\mathrm{new}}(t_0+\delta t)=\mathbf{r}_i^{\mathrm{old}}(t_0+\delta t)-\delta t\,\gamma_k(q)\times\sum_{j=a,b,c,d}\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\tau_{abcd}](t_0)\quad(i=a,b,c,d) \tag{47}$$

and

$$\gamma_k(q)=[\delta t]^{-1}\frac{\tau_{abcd}(\mathbf{r}^{\mathrm{old}}(t_0+\delta t))-\beta_{abcd}}{\sum_{i,j=a,b,c,d}[\boldsymbol{\nabla}_i\tau_{abcd}](\mathbf{r}^{\mathrm{old}}(t_0+\delta t))\cdot\boldsymbol{\mu}^{\mathrm{tt}}_{ij}(t_0)\cdot[\boldsymbol{\nabla}_j\tau_{abcd}](t_0)}, \tag{48}$$

respectively. Expressions for $\boldsymbol{\nabla}_i\phi_{abc}$ and $\boldsymbol{\nabla}_i\tau_{abcd}$ are furnished by the Wilson vectors [30] for angle and torsional internal coordinates, respectively. In the first part and stage I of iteration *q*, HSHAKE successively selects each distance, angle, and torsion constraint, and corrects its coordinates according to Eqs. (41) and (42), Eqs. (44) and (45), and Eqs. (47) and (48), respectively. In stage II of iteration *q*, HSHAKE corrects these coordinates according to Eq. (30),

$$\begin{aligned}\mathbf{r}_i^{\mathrm{new}}(t_0+\delta t)=\mathbf{r}_i^{\mathrm{old}}(t_0+\delta t)&-2\,\delta t\sum_{k'\in(C^{\mathrm{d}}-C_i)}\gamma_{k'}(q)D_{k'}\\&-\delta t\sum_{k'\in(C^{\mathrm{a}}-C_i)}\gamma_{k'}(q)A_{k'}\\&-\delta t\sum_{k'\in(C^{\mathrm{t}}-C_i)}\gamma_{k'}(q)T_{k'}\quad(\forall\, i\in P_c),\end{aligned} \tag{49}$$

where

$$D_{k'}\equiv\{\overline{\boldsymbol{\mu}}^{\mathrm{tt}}_{i,j'i'}(t_0)\cdot\mathbf{r}_{j'i'}(t_0)\}_{k'},$$

$$A_{k'}\equiv\{\boldsymbol{\mu}^{\mathrm{tt}}_{ia}(t_0)\cdot[\boldsymbol{\nabla}_a\phi_{abc}](t_0)+\boldsymbol{\mu}^{\mathrm{tt}}_{ib}(t_0)\cdot[\boldsymbol{\nabla}_b\phi_{abc}](t_0)+\boldsymbol{\mu}^{\mathrm{tt}}_{ic}(t_0)\cdot[\boldsymbol{\nabla}_c\phi_{abc}](t_0)\}_{k'},$$

$$\begin{aligned}T_{k'}\equiv\{&\boldsymbol{\mu}^{\mathrm{tt}}_{ia}(t_0)\cdot[\boldsymbol{\nabla}_a\tau_{abcd}](t_0)+\boldsymbol{\mu}^{\mathrm{tt}}_{ib}(t_0)\cdot[\boldsymbol{\nabla}_b\tau_{abcd}](t_0)\\&+\boldsymbol{\mu}^{\mathrm{tt}}_{ic}(t_0)\cdot[\boldsymbol{\nabla}_c\tau_{abcd}](t_0)+\boldsymbol{\mu}^{\mathrm{tt}}_{id}(t_0)\cdot[\boldsymbol{\nabla}_d\tau_{abcd}](t_0)\}_{k'}.\end{aligned} \tag{50}$$

After iteration to convergence, the second part of HSHAKE corrects the unconstrained $\mathbf{r}_i$ according to Eq. (23):

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\gamma)=\mathbf{r}'_i(t_0+\delta t)&-2\,\delta t\sum_{k'\in C^{\mathrm{d}}}\gamma_{k'}D_{k'}\\&-\delta t\sum_{k'\in C^{\mathrm{a}}}\gamma_{k'}A_{k'}-\delta t\sum_{k'\in C^{\mathrm{t}}}\gamma_{k'}T_{k'}\\&(\forall\, i\notin P_c),\end{aligned} \tag{51}$$

with γ's given by Eq. (31).

IV. SIMULATION RESULTS AND DISCUSSION

To verify HSHAKE's methodology and report its performance, we present here results obtained with it from type (1) simulations of chains and rigid nonspherical particles. We recall from Sec. I that in type (1) simulations rigid bonding between particles is not allowed after the start of simulation and hence the constraints are determined at the outset, whereas in type (2) simulations rigid bonding is possible during simulation and consequently both the number and forms of the constraints may vary during the run. Results

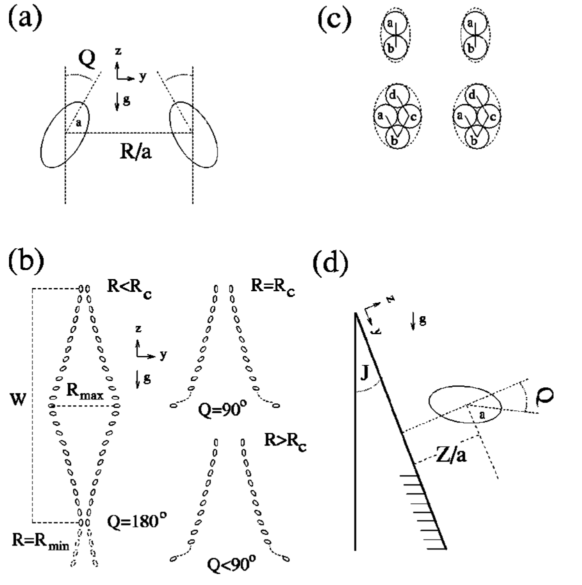

FIG. 1. (a) The instantaneous configuration of a pair of sedimenting spheroids, with axes in a vertical plane, is determined by their separation R/a and inclination angle Q. (b) Types of motion for a pair of sedimenting prolate spheroids with axes initially parallel to gravity. W is the spatial period for $R(t_0)<R_c$, and the dashed lines for $R(t_0)=R_c$ and $R(t_0)>R_c$ denote asymptotic trajectories. (c) The two pairs or singles of Secs. IV A and IV B, respectively, with dashed contours to convey overall shapes. From top to bottom, $a\approx D$ and $1.37D$, respectively. (d) The instantaneous configuration of a spheroid sedimenting near an inclined wall, with axis in a plane normal to it, is determined by the separation Z/a and orientation angle Q.

from type (2) simulations are described elsewhere. In Secs. IV A–IV D, the spherical particles have diameter $D=1\ \mu$m, are in a fluid with the shear viscosity η of water, and their hydrodynamic mobility is computed using our force multipole algorithm [1], which includes the hydrodynamic wall effect in the bounded geometry cases. The boundary in these cases is a planar hard wall with stick boundary conditions, and lies in the xy plane with the fluid occupying the half space $z>0$. We adhere for accuracy to a multipole truncation order $L_t\geqslant 3$ [1,2,17], which is always specified. Both the Euler integrator in Eq. (9) and the fourth order Runge-Kutta in Eq. (10) were used to generate the $\mathbf{r}'(t_0+\delta t)$ [see Eq. (12)]. Results were generated using the former, except where stated otherwise. Test simulations were performed to ensure that the chosen δt values yielded accurate results.

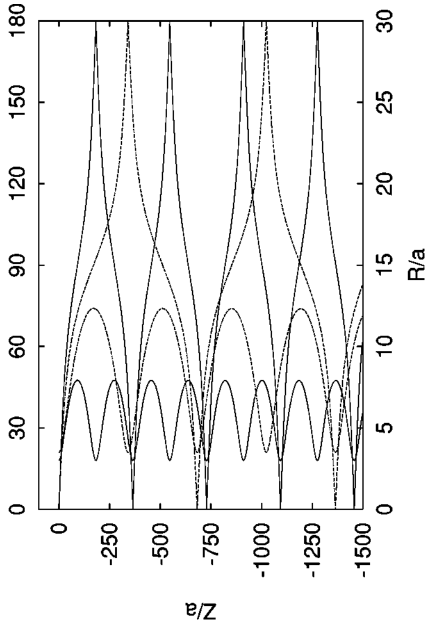

FIG. 2. Sedimentation distance Z/a vs separation R/a and rotation angle Q, from SD simulations with HSHAKE of a pair of sedimenting RSD in an unbounded fluid, with $\delta t=0.005$ s, $L_t=3$, and $Q(t_0)=0$. The solid and dashed lines correspond to $[R/a](t_0)=3$ and 3.5, respectively, and Q is measured from the upward pointing vertical such that $0\leqslant Q\leqslant 180°$.

A. Sedimentation of a pair of nonspherical particles

Referring to Fig. 1(a), consider two identical prolate spheroids with axes initially vertically oriented $[Q(t_0)=0]$ and aligned with gravity. Depending on the initial value of their scaled center-center separation $[R/a](t_0)$, where a is the length of the major semiaxis, the released spheroids can execute three types of sedimenting motion [31]: (1) if $[R/a](t_0)>R_c/a$, where R_c/a is some critical separation, the particles steadily separate while Q increases, and in the limit of infinite separation $Q(t_\infty)<90°$; (2) if $[R/a](t_0)=R_c/a$, the particles steadily separate with Q increasing to $Q(t_\infty)=90°$; (3) if $[R/a](t_0)<R_c/a$, the particles rotate past the horizontal $(Q>90°)$ at some finite maximum separation R_{max}, and then approach each other along trajectories mirroring those preceding the maximum separation. Once

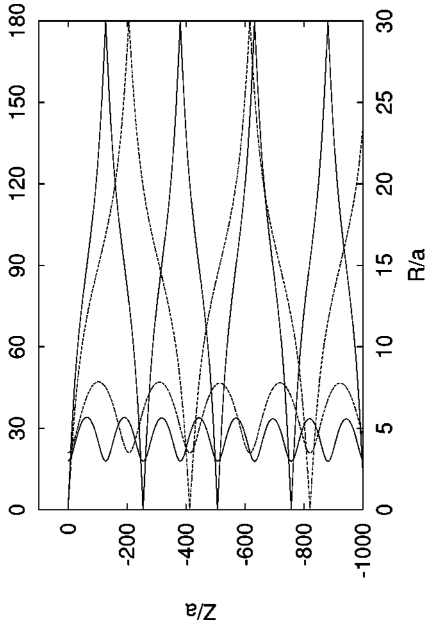

FIG. 3. Same as Fig. 2, but for a pair of sedimenting RSQ with the solid and dashed lines corresponding to $[R/a](t_0)=3$ and 3.5, respectively.

TABLE II. Average number of HSHAKE iterations per step for the SD simulations in an unbounded fluid of sedimenting pairs of RSD and RSQ, taken over $\approx 10^6$ and 9.3×10^5 time steps, respectively, where $\delta t = 0.005$ s.

	$[R/a](t_0)=3$	3.5	4	4.5	5	6.6
RSD	1.044	1.015	1.005	1.001	1.0	
RSQ	30.32	29.25	28.39		25.35	20.86

$R/a = [R/a](t_0)$ again, the motion proceeds periodically and indefinitely in an unbounded fluid. The three regimes are depicted in Fig. 1(b). Referring to Fig. 1(c), consider two pairs of identical nonspherical particles: rigid spheroidal dimers (RSD) of aspect ratio ≈ 2, and rigid spheroidal quadrimers (RSQ) of aspect ratio ≈ 1.37. We define R/a as the scaled (center of mass)-(center of mass) separation between a pair, Z/a as the scaled z coordinate of the center of mass of either particle in a pair, the two equal by symmetry, and Q as the angle between the z axis and ab for a dimer, and bd for a quadrimer.

We first performed SD simulations on each pair in an unbounded fluid with different $[R/a](t_0)$, $Q(t_0)=0$, and a sedimentation force $-F_g\hat{\boldsymbol{z}}$ on each spherical particle, where $F_g = 5.1313\times 10^{-15}$ N. Using HSHAKE, dimer rigidity was maintained by a distance constraint as in Eq. (40), with $d_{ab} = 1.001D$, and quadrimer rigidity by three distance constraints, two angle constraints as in Eq. (43), and a torsion constraint as in Eq. (46), with $d_{ab}=d_{bc}=d_{cd}=1.001D$, $\alpha_{abc}=60°$, $\alpha_{bcd}=120°$, and $\beta_{abcd}=0$. Convergence of a distance, angle, or torsion constraint is reached when [Eq. (42) numerator] $\leqslant \chi_d \times d_{ij}^2$, [Eq. (45) numerator] $\leqslant \chi_a \times \alpha_{abc}$, or [Eq. (48) numerator] $\leqslant \chi_t \times \beta_{abcd}$, respectively, where we take $\chi_d=\chi_a=\chi_t=10^{-6}$. The simulations generate the three dynamical regimes of Fig. 1(b). The pair of RSD in Fig. 2 is released from an initial reference height of $Z/a=0$ and sediments down into the negative range of Z/a. For each initial value of the pair separation, $[R/a](t_0)$, the sedimenting motion generates two curves plotted with the same line type in Fig. 2. The curve sampling the whole range of the horizontal represents $[Z/a](Q)$ while the other represents $[Z/a](R/a)$. Similar remarks apply to the pair of RSQ in Fig. 3. Figures 2 and 3 show that for a given aspect ratio, a larger $[R/a](t_0)$ leads to larger R_{max} and W (and a larger temporal period T), whereas for a given $[R/a](t_0)$, a larger aspect ratio leads to larger R_{max} and W (and a larger T). Lengthy simulations suggest $R_c/a \approx 4.93$ for the dimers and $R_c/a \approx 6.15$ for the quadrimers. Table II gives the convergence rates of HSHAKE in the simulations. The above simulation results for the pairs of RSD and RSQ constructed with HSHAKE are consistent with those obtained for spheroids by the method of reflections [31] or the ana-

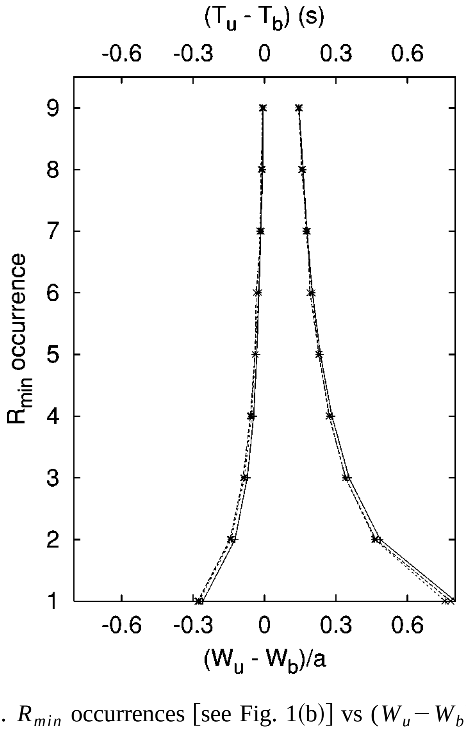

FIG. 4. R_{min} occurrences [see Fig. 1(b)] vs $(W_u - W_b)/a$ (right curves) and $(T_u - T_b)$ (left curves) from SD simulations with HSHAKE, in unbounded and bounded fluid, of a pair of sedimenting RSD with $Q(t_0)=0$, $[R/a](t_0)=3$, $[Z/a](t_0)=2000$, and $L_t=3$. The dotted and dashed curves were generated with $\delta t = 0.005$ s, using the Euler integrator in Eq. (9) and the fourth order Runge-Kutta in Eq. (10), respectively. The solid curve was generated with $\delta t = 0.001$ s using the Euler integrator. Occurrence 9 is the initial and 1 the final before the pair impacts the wall.

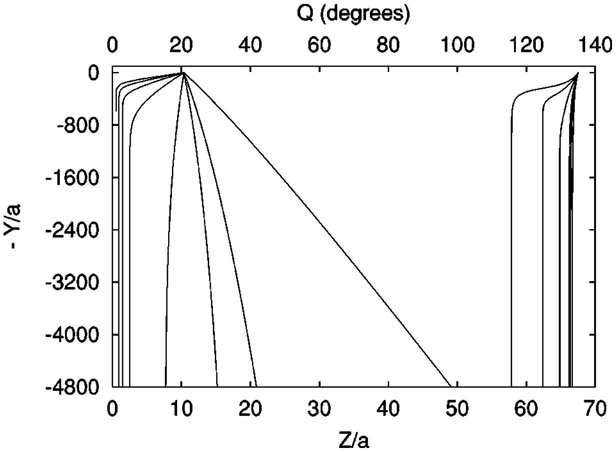

FIG. 5. Sedimentation distance $-Y/a$ along the wall vs separation Z/a and rotation angle Q, from SD simulations with HSHAKE of a sedimenting RSD near a tilted wall, with $\delta t = 0.01$ s, $L_t=3$, $[Z/a](t_0)=10.36$, and $Q(t_0)=135°$. The Z/a curves (left) and the Q curves (right) correspond from right to left (f.r.t.l.) to tilt angle $J=2.5°, 2.9°, 3°, 3.2°, 4°, 5°, 6°$, and $7°$, with the $J_w=7°$ Q curve omitted, where w denotes particle contact with the wall.

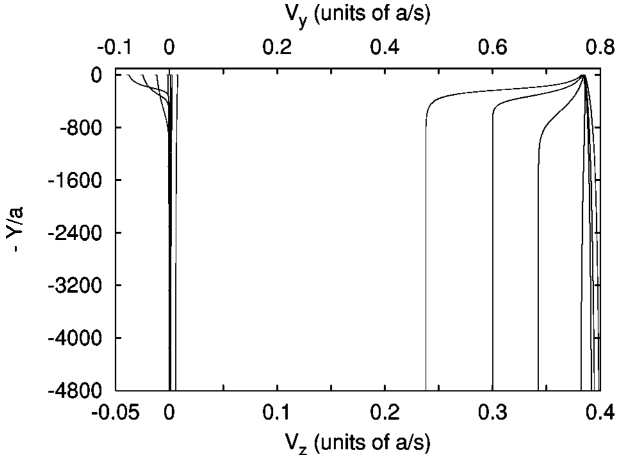

FIG. 6. $-Y/a$ vs normal and parallel velocities V_z and V_y, respectively, from the simulations of Fig. 5. The V_z curves (left) and the V_y curves (right) correspond f.r.t.l. to $J=2.5°$, 2.9°,3°,3.2°,4°,5°, and 6°. The $J_w=7°$ velocity curves are omitted.

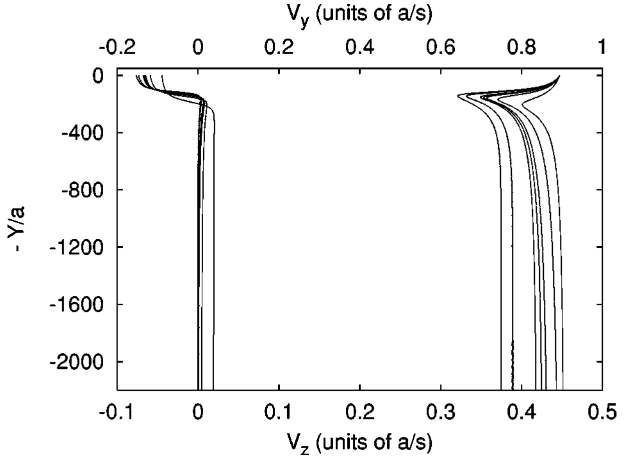

FIG. 8. Same as Fig. 6, but from the simulations of Fig. 7. The V_z curves (left) and the V_y curves (right) correspond f.r.t.l. to $J=1°$,2°,2.4°,2.5°,2.6°,3°, and 3.2°. The $J_w=4°$ velocity curves are omitted.

lytical extension of a mobility scheme for SD to prolate spheroids [14], noted in Sec. I.

We also performed two SD simulations, in a bounded and unbounded fluid, respectively, of a pair of sedimenting RSD with $[R/a](t_0)=3$, $Q(t_0)=0$, $[Z/a](t_0)=2000$, and a force $-F_g\hat{\boldsymbol{z}}$ on each spherical particle. Comparison of results reveals a sedimenting motion in the bounded fluid qualitatively similar to that in the unbounded fluid, depicted in Fig. 2 for $[R/a](t_0)=3$. With u and b denoting unbounded and bounded fluid, respectively, Fig. 4 shows that as the pair sediments, the hydrodynamic effect of the wall leads to a decrease in W_b and an increase in T_b, compared to the constant $W_u\approx 182.7a$ and $T_u\approx 211.5$ s, so that the oscillating trajectories become more "compressed," the motion more retarded, and no longer strictly periodic. Given the magnitudes of T_u and W_u, the effect of the wall on the trajectories is seen to be relatively small in this case.

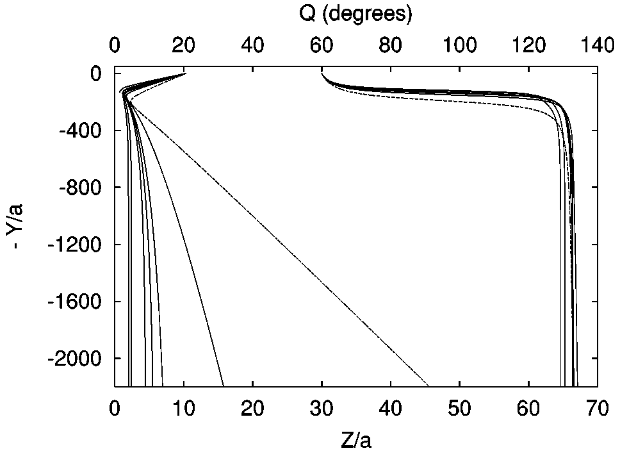

FIG. 7. Same as Fig. 5, but for a sedimenting RSQ near a tilted wall, with $[Z/a](t_0)=10.32$ and $Q(t_0)=60°$. The Z/a solid curves (left) and the Q solid curves (right) correspond f.r.t.l. to $J=2°$,2.4°,2.5°,2.6°,3°,3.2°, and 4°, with the $J_w=4°$ Q curve omitted. The dashed curves correspond to $J=1°$.

B. Sedimentation of a nonspherical particle near a tilted wall

Referring to Fig. 1(d), consider a prolate spheroid near an infinite plane wall inclined by an angle J with respect to gravity. For arbitrary initial orientation $Q(t_0)$ and scaled center-wall separation $[Z/a](t_0)$, the released spheroid can execute three types of sedimenting motion, depending on the value of J [32]: (1) if $J<J_1$, where J_1 is some critical angle, the particle separates from the wall while Q varies and in the limit of infinite separation the rotation ceases and $Q(t_\infty)$ becomes constant; (2) if $J_1<J<J_2$, where J_2 is another critical angle, the particle eventually reaches a stable equilibrium where it translates uniformly at $[Z/a]=[Z/a]^{\mathrm{eq}}$ and $Q=Q^{\mathrm{eq}}$; finally, (3) if $J>J_2$, the particle monotonically approaches the wall until it makes contact. Referring to Fig. 1(c), we consider a single particle in a bounded fluid from each pair and define Z/a as the scaled (center of mass)-wall separation, Y/a as the scaled y coordinate of the center of mass, V_z and V_y as the corresponding center of mass velocities, and Q as the orientation angle relative to the z axis, specified as in Sec. IV A.

We performed SD simulations of each particle sedimenting near a wall inclined by an angle J relative to a force of magnitude F_g on each sphere, with the same (arbitrary) values of $[Z/a](t_0)$ and $Q(t_0)$ and different values of J. Particle rigidity was maintained using HSHAKE, as specified in Sec. IV A. As seen in Figs. 5–8, the simulations produce the three dynamical regimes described above. In particular, for a given aspect ratio, a larger J leads to a smaller $[Z/a]^{\mathrm{eq}}$, as seen from Fig. 5 or 7; for a given J, a larger aspect ratio leads to a larger $[Z/a]^{\mathrm{eq}}$, as seen from Figs. 5 and 7 for $J=3°$ and 3.2°. In addition, Figs. 6 and 8 show that for $J<J_1$, the velocity normal to the wall, V_z, is nonzero, while for $J_1<J<J_2$, V_z eventually vanishes and the parallel component V_y becomes constant. Simulations we performed with other arbitrary values of $[Z/a](t_0)$ and $Q(t_0)$ yielded qualitatively

TABLE III. Average number of HSHAKE iterations per step for the SD simulations of sedimentation near a tilted wall of a RSD and a RSQ, taken over 10^6 and 3×10^5 time steps, respectively, where $\delta t=0.01$ s. Averages for $J=J_w$ are not shown.

	$J=1°$	2°	2.4°	2.5°	2.6°	2.9°	3°	3.2°	4°	5°	6°
RSD				1.028		1.074	1.11	1.31	1.96	1.98	1.99
RSQ	13.71	15.7	15.1	14.84	17.19		12.57	12.94			

similar results. Table III gives the convergence rates of HSHAKE in the simulations. The above simulation results for the RSD and RSQ constructed with HSHAKE are consistent with those obtained for spheroids by the boundary integral method [32] or a combination of SD with a nonconstraint scheme for forming dimers [33].

C. Comparative simulations of a sedimenting chain of spheres

We performed a pair of SD simulations in an unbounded fluid on a sedimenting chain of 32 spherical particles connected by distance constraints, as in Eq. (40), maintained by HSHAKE and SHAKE*, respectively. The center-center distance between adjacent particles was constrained to 1.4D, with $\chi_d=10^{-6}$. Each simulation started from the same random nonlinear chain configuration shown in Fig. 10 and ran for 4300 steps with $\delta t=0.01$ s, $L_t=3$, a sedimentation force $F_g\hat{\boldsymbol{y}}$ on each particle, and a Lennard-Jones 6-12 potential with $\sigma=D$ and $\varepsilon=kT$, where $T=300$ K. Absence of unconstrained particles from the current system means that HSHAKE and SHAKE* differ only in the missing $\delta\mathbf{r}_i^{\mathrm{II}}$ contribution from the latter, representing contributions from all constraints that do not contain $\mathbf{r}_i$. Hence application of SHAKE* is equivalent to a "nearest rods" approximation along the chain.

For the mth order integration algorithm we can write based on Secs. III B and III C,

$$\mathbf{r}_i[\text{HSHAKE}]=\mathbf{r}_i[\text{exact}]+\alpha_i O(\delta t^{m+1}),$$

$$\mathbf{r}_i[\text{SHAKE}^\star]=\mathbf{r}_i[\text{exact}]+\beta_i O(\delta t^{m}) \qquad (i=1,\ldots,N), \tag{52}$$

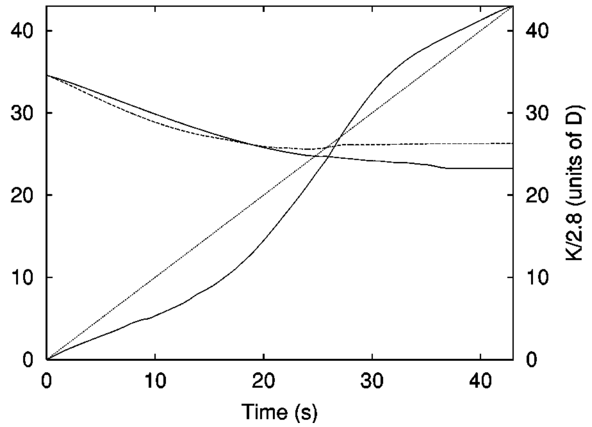

FIG. 9. The solid curve through the origin is the RMS difference $K(t)$ (/2.8) between the trajectories from SD simulations in unbounded fluid of a sedimenting 32-particle chain using HSHAKE and SHAKE*. The dotted straight line highlights its quasilinear dependence. The remaining curves represent the magnitude G of the end-to-end vector from the simulations with HSHAKE (solid) and SHAKE* (dashed).

where α and β are prefactors of the local truncation errors in one step of HSHAKE and SHAKE*, respectively. After $N_s=t/\delta t$ time steps, the global truncation error in HSHAKE is given by $\Sigma_{j=1}{}^{N_s}[\alpha_i^j O(\delta t^{m+1})]=O(\delta t^{m+1})\Sigma_{j=1}{}^{N_s}\alpha_i^j=O(\delta t^{m+1})N_s\overline{\alpha_i}=\overline{\alpha_i}tO(\delta t^m)$, where the bar denotes an average over N_s steps. With a similar calculation for SHAKE*, the trajectories from the two schemes can be written as

$$\mathbf{r}_i[\text{HSHAKE}](t)=\mathbf{r}_i[\text{exact}](t)+\overline{\alpha_i}tO(\delta t^m),$$

$$\mathbf{r}_i[\text{SHAKE}^\star](t)=\mathbf{r}_i[\text{exact}](t)+\overline{\beta_i}tO(\delta t^{m-1})$$

$$(i=1,\ldots,N). \tag{53}$$

We computed the RMS difference between the trajectories from SD simulations with HSHAKE and SHAKE*, $K(t)=(\Sigma_{i=1}^N\{\mathbf{r}_i^c[\text{HSHAKE}](t)-\mathbf{r}_i^c[\text{SHAKE}^\star](t)\}^2/N)^{1/2}$, where $N=32$ and the computed trajectories can be written as

$$\mathbf{r}_i^c[\text{HSHAKE}](t)=\mathbf{r}_i[\text{HSHAKE}](t)+\rho_i,$$

$$\mathbf{r}_i^c[\text{SHAKE}^\star](t)=\mathbf{r}_i[\text{SHAKE}^\star](t)+\rho_i' \qquad (i=1,\ldots,N), \tag{54}$$

with ρ and ρ' denoting the global round-off errors due to finite machine precision in executing the codes containing the HSHAKE and SHAKE* routines, respectively. As these codes are identical except for the difference between the HSHAKE and SHAKE* routines, which consists of the missing implementation of $\delta\mathbf{r}_i^{\mathrm{II}}$ from the latter, to a good approximation $\rho_i\approx\rho_i'(i=1,\ldots,N)$. From Eq. (54) we then have $\{\mathbf{r}_i^c[\text{HSHAKE}](t)-\mathbf{r}_i^c[\text{SHAKE}^\star](t)\}\approx\{\mathbf{r}_i[\text{HSHAKE}](t)-\mathbf{r}_i[\text{SHAKE}^\star](t)\}$, and by means of Eq. (53) $K(t)\approx BtO(\delta t^{m-1})$, where $B=(\Sigma_{i=1}^N\overline{\beta_i^2}/N)^{1/2}$ is expected to be bounded and slowly varying in time because of the averaging over both N_s and N. For the Euler integrator $m=1$ and $K(t)\approx Bt$. Thus the SHAKE* violation of requirement (B), summarized by Eq. (52), should lead to a computed $K(t)$ quasilinear in time. Figure 9 confirms this prediction and also compares the magnitudes of the end-to-end vectors from the two simulations. Figure 10 highlights the marked difference between the chain configurations generated in the simulations using HSHAKE and SHAKE*, illustrating the point made in Secs. II and III C that violation of requirement (B), here by SHAKE*, would artificially alter the dynamics of the system.

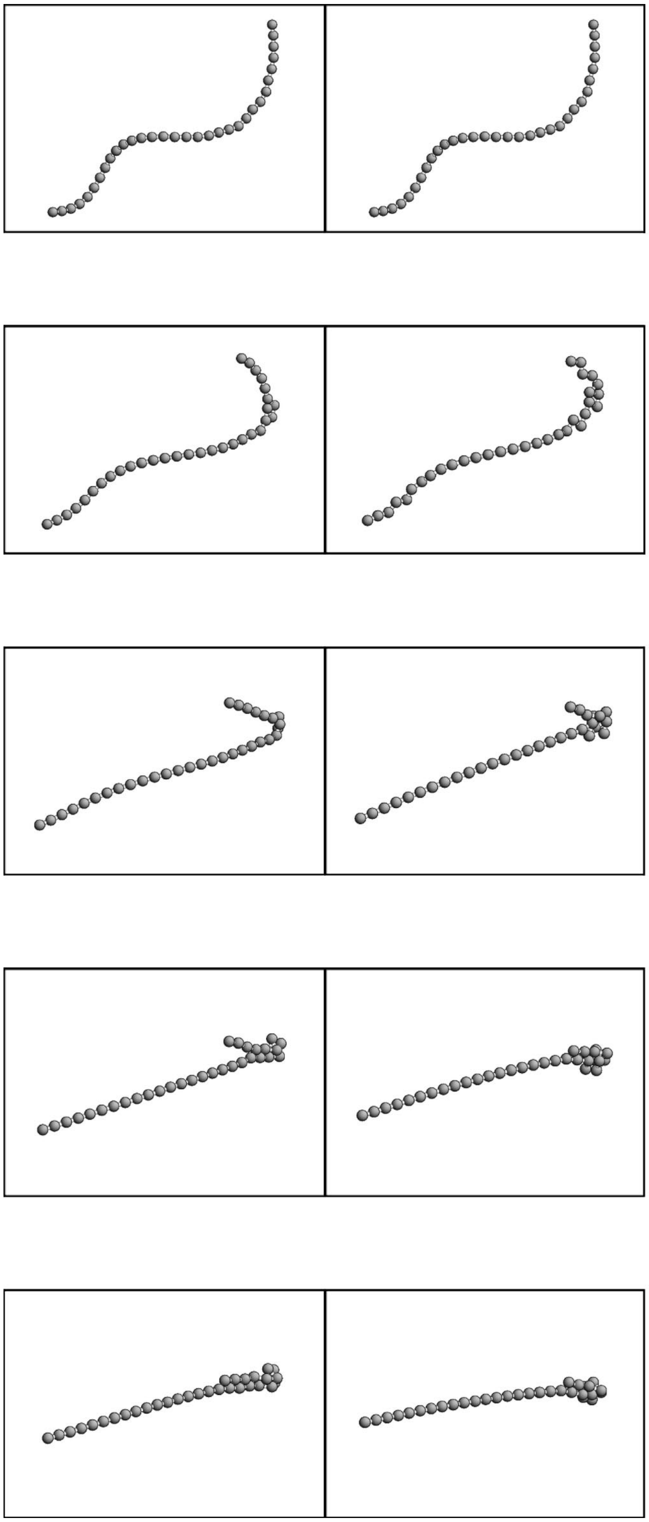

FIG. 10. Snapshot pairs from the simulations of Fig. 9 with HSHAKE (left) and SHAKE* (right), taken from top to bottom at 0 s, 10.75 s, 21.5 s, 32.25 s, and 43 s. The sedimentation force $F_g\hat{y}$ points from left to right along the horizontal, with a slight tilt into the page.

D. Friction coefficient of a linear chain of spheres near a wall

Consider a rigid linear chain of N spherical particles with center-center separation between adjacent particles of $(D+S)$, placed in a fluid parallel to the x axis, at a height H above a wall in the xy plane. The normalized friction coefficient of the chain for motion parallel to its axis is given by

$$C=\frac{F_x}{3\pi\eta D U_x}, \tag{55}$$

where F_x is the force on the chain and U_x its translational velocity. Analytical approaches [34,35] for evaluating C are limited to chains with large aspect ratios and at relatively large H. We first describe two numerical approaches for computing C. The first can be cast as either a friction or equivalent mobility problem, and requires a single evaluation of elements of the (inverse) friction or (inverse) mobility tensors, respectively, at the chain configuration specified above. The second approach requires short SD simulations on the chain using HSHAKE and makes direct use of Eq. (55). C values for the chain computed by both approaches, for several values of N and H, are then compared to verify the accuracy of HSHAKE. For conciseness we define the chain as translationally rigid (TR) when (1) it remains parallel to the x axis at height H above the wall, ensured by imposing $U_{iy}=U_{iz}=0$ $(i=1,\ldots,N)$, and (2) its spheres all move with the same translational velocity, $U_{ix}=U_x$. The chain is completely rigid (CR) if it is also rotationally rigid, ensured by setting $\mathbf{\Omega}=\mathbf{0}$, which implies that in general $\mathbf{T}\neq\mathbf{0}$. For an only translationally rigid (OTR) chain, we set $\mathbf{T}=\mathbf{0}$, so that in general $\mathbf{\Omega}\neq\mathbf{0}$.

The friction formulation of the first approach for computing C employs the inverse of Eq. (1),

$$\begin{pmatrix}\mathbf{F}\\ \mathbf{T}\end{pmatrix}=\begin{pmatrix}\boldsymbol{\zeta}^{\mathrm{tt}} & \boldsymbol{\zeta}^{\mathrm{tr}}\\ \boldsymbol{\zeta}^{\mathrm{rt}} & \boldsymbol{\zeta}^{\mathrm{rr}}\end{pmatrix}\begin{pmatrix}\mathbf{U}\\ \mathbf{\Omega}\end{pmatrix}, \tag{56}$$

where the $\boldsymbol{\zeta}^{\mathrm{ab}}$ are $(3N\times3N)$ friction matrices. From the first row of Eq. (56),

$$\mathbf{F}=\boldsymbol{\zeta}^{\mathrm{tt}}\mathbf{U}+\boldsymbol{\zeta}^{\mathrm{tr}}\mathbf{\Omega}, \tag{57}$$

or in dyadic notation $\mathbf{F}_i=\Sigma_{j=1}^{N}(\boldsymbol{\zeta}_{ij}^{\mathrm{tt}}\cdot\mathbf{U}_j+\boldsymbol{\zeta}_{ij}^{\mathrm{tr}}\cdot\mathbf{\Omega}_j)$ $(i=1,\ldots,N)$, where the $\boldsymbol{\zeta}_{ij}^{\mathrm{ab}}$ are (3×3) friction tensors. For a CR chain, Eq. (57) leads to $F_{ix}=U_x\Sigma_{j=1}^{N}[\boldsymbol{\zeta}_{ij}^{\mathrm{tt}}]_{xx}$. Summing over N and using Eq. (55) gives

$$C=\frac{1}{3\pi\eta D}\sum_{i,j=1}^{N}[\boldsymbol{\zeta}_{ij}^{\mathrm{tt}}]_{xx}\quad(\mathrm{CR}). \tag{58}$$

For an OTR chain, the second row of Eq. (56) yields $\mathbf{\Omega}=-[\boldsymbol{\zeta}^{\mathrm{rr}}]^{-1}\boldsymbol{\zeta}^{\mathrm{rt}}\mathbf{U}$, Eq. (57) then gives $F_{ix}=U_x\Sigma_{j=1}^{N}\{(\boldsymbol{\zeta}^{\mathrm{tt}}-\boldsymbol{\zeta}^{\mathrm{tr}}[\boldsymbol{\zeta}^{\mathrm{rr}}]^{-1}\boldsymbol{\zeta}^{\mathrm{rt}})_{ij}\}_{xx}$, and we get

$$C=\frac{1}{3\pi\eta D}\sum_{i,j=1}^{N}\{(\boldsymbol{\zeta}^{\mathrm{tt}}-\boldsymbol{\zeta}^{\mathrm{tr}}[\boldsymbol{\zeta}^{\mathrm{rr}}]^{-1}\boldsymbol{\zeta}^{\mathrm{rt}})_{ij}\}_{xx}\quad(\mathrm{OTR}). \tag{59}$$

In the mobility formulation, the first row of Eq. (1) gives

$$\mathbf{U}=\boldsymbol{\mu}^{\mathrm{tt}}\mathbf{F}+\boldsymbol{\mu}^{\mathrm{tr}}\mathbf{T}. \tag{60}$$

TABLE IV. Computed friction coefficients of the chain described in Sec. IV D, with N spheres, length $L=ND+0.1D(N-1)$, and height $H=(0.5+10^{-6})D$ above the wall. C and C^* are the friction coefficients of the OTR chain, computed by Eq. (63) from short SD simulations using HSHAKE and SHAKE*, respectively, with $L_t=4$, $\delta t=10^{-4}$ s, and $\chi_d=10^{-10}$. From the simulations with HSHAKE, $\langle \Omega_y \rangle$ is the average angular velocity of the spheres about the y axis, U_x the chain translational velocity, and I the average number of HSHAKE iterations per step.

N	C	Eq. (62)	Eq. (58)	C^*	$\langle \Omega_y \rangle$(rad/s)	$U_x(D/\mathrm{s})$	I
5	33.616	33.616	35.549	33.515	0.0361	0.0810	11
10	65.964	65.964	69.972	65.8078	0.0374	0.0825	32
15	98.295	98.295	104.381	98.119	0.0378	0.0831	56
20	130.6209	130.6209	138.786	130.435	0.0380	0.0834	81.5
25	162.944	162.944	173.189	162.753	0.0382	0.0835	107.75
30	195.266	195.266	207.591	195.0707	0.0383	0.0836	128.75

For a CR chain, the second row of Eq. (1) leads to $\mathbf{T}=-[\boldsymbol{\mu}^{\mathrm{rr}}]^{-1}\boldsymbol{\mu}^{\mathrm{rt}}\mathbf{F}$, and Eq. (60) yields $F_{ix}=U_x\Sigma_{j=1}^{N}\{(\boldsymbol{\mu}^{\mathrm{tt}}-\boldsymbol{\mu}^{\mathrm{tr}}[\boldsymbol{\mu}^{\mathrm{rr}}]^{-1}\boldsymbol{\mu}^{\mathrm{rt}})_{ij}^{-1}\}_{xx}$, so that

$$C=\frac{1}{3\pi\eta D}\sum_{i,j=1}^{N}\{(\boldsymbol{\mu}^{\mathrm{tt}}-\boldsymbol{\mu}^{\mathrm{tr}}[\boldsymbol{\mu}^{\mathrm{rr}}]^{-1}\boldsymbol{\mu}^{\mathrm{rt}})_{ij}^{-1}\}_{xx}\quad (\mathrm{CR}). \tag{61}$$

For an OTR chain, Eq. (60) leads to $F_{ix}=U_x\Sigma_{j=1}^{N}[(\boldsymbol{\mu}^{\mathrm{tt}})_{ij}^{-1}]_{xx}$, and we get

$$C=\frac{1}{3\pi\eta D}\sum_{i,j=1}^{N}[(\boldsymbol{\mu}^{\mathrm{tt}})_{ij}^{-1}]_{xx}\quad (\mathrm{OTR}). \tag{62}$$

Equations (58) and (59) are equivalent to Eqs. (61) and (62), respectively. Given the chain configuration above and our friction/mobility algorithm [1] for spheres in a bounded fluid, the more convenient Eqs. (58) and (62) are adopted however for computing C by this first approach below. Calculations of this type, requiring no simulations, have been performed [36,37] on chains of spheres but using Eq. (58) only.

In the second approach for computing C, we perform a short SD simulation on the same chain with an external force $F_g\hat{\boldsymbol{x}}$ on each sphere. By means of HSHAKE, $(N-1)$ distance constraints, as in Eq. (40), fix the separation between adjacent particles to $(D+S)$. By symmetry $U_{iy}=0$ $(i=1,\ldots,N)$, and for $H\gg L$, where the chain length $L=ND+(N-1)S$, the medium is effectively unbounded so that $U_{iz}\approx 0$ also by symmetry, whereas for $H\ll L$, HI with the wall lead to $U_{iz}\approx 0$, as illustrated below. Hence for $H\gg L$ or $H\ll L$, $U_{iy}=0$, $U_{iz}\approx 0$, and by virtue of HSHAKE $U_{ix}=U_x$, so the chain is TR. Absence of orientational coordinates from the constraints implies $\mathbf{T}^c=\mathbf{0}$, hence in general $\boldsymbol{\Omega}\neq\mathbf{0}$, so the chain is OTR. The force on the chain is $F_x=NF_g+\Sigma_{i=1}^{N}F_{ix}^{c}$, where the sum of the internal constraint forces must vanish. The approximate constraint forces $\mathbf{F}_i^c=-\Sigma_{k=1}^{l}\gamma_k\boldsymbol{\nabla}_i\sigma_k$ were computed at every time step and the vanishing of their sum verified as a numerical check. The velocity of each sphere was computed by the central difference formula $\mathbf{U}_i(t_0)=[\mathbf{r}_i(t_0+\delta t)-\mathbf{r}_i(t_0-\delta t)]/2\delta t+O(\delta t^2)$, $U_{ix}(t_0)=U_x(t_0)$ was verified, and C computed in this second approach by Eq. (55) as

$$C=C(t_0)=\frac{NF_g}{3\pi\eta DU_x(t_0)}. \tag{63}$$

Tables IV and V compare the friction coefficients computed by Eqs. (63), (62), and (58), for $H=(0.5+10^{-6})D$ and $H=10L$, respectively, and various chain lengths. As a dilute aqueous suspension of micron-diameter polystyrene spheres is observed [38] to aggregate into linear chains with $S\approx 0.1D$ under a strong electric field, and similar separation has been predicted [39] for field-induced chains of dipolar colloidal particles, we took $S=0.1D$. For C computed with SD simulations by Eq. (63), we generated the $\mathbf{r}'(t_0+\delta t)$ [see Eq. (12)] with the fourth order Runge-Kutta integrator in Eq. (10) and verified at every time step that $U_{ix}=U_x$, $U_{iy}=0$, $U_{iz}\approx 0$ $(i=1,\ldots,N)$, hence that the chain is TR as anticipated. That the chain is specifically OTR is highlighted by the nonzero $\langle\Omega_y\rangle$ in Table IV. Tables IV and V show that the friction coefficients computed by Eq. (63) using HSHAKE agree exactly with those from Eq. (62), for both values of H. In addition, for $H=10L$ the medium around the chain is effectively unbounded, weak HI with the wall give $\langle\Omega_y\rangle\approx 0$, and the OTR chain is effectively CR. We expect then the friction coefficients computed by Eq. (58) to agree with

TABLE V. Same as Table IV, but with $H=10L$.

N	C	Eq. (62)	Eq. (58)	C^*	$\langle \Omega_y \rangle$(rad/s)	$U_x(D/\mathrm{s})$	I
5	2.0840	2.0840	2.0840	2.0494	≈ 0	1.3062	15
10	3.116	3.116	3.116	3.0270	≈ 0	1.747	25
15	4.0348	4.0348	4.0348	3.9064	≈ 0	2.0240	22
20	4.890	4.890	4.890	4.733	≈ 0	2.227	31
25	5.7025	5.7025	5.7025	5.524	≈ 0	2.387	41
30	6.484	6.484	6.484	6.288	≈ 0	2.519	55

those from Eq. (63) or (62), as Table V confirms. HSHAKE generates through Eq. (63) the correct friction coefficients in Tables IV and V because it satisfies both requirements (A) and (B). Any alternative scheme satisfying requirement (A) but not (B) will result in a chain which is also OTR under the above simulation conditions but with an incorrect U_x, hence incorrect friction coefficients. We illustrate this point by recomputing the friction coefficients with Eq. (63) but using SHAKE*. As discussed in Sec. IV C, the difference for the chain system between HSHAKE and SHAKE* is the missing $\delta\mathbf{r}_i^{\mathrm{II}}$ contribution from the latter, which leads to violation of requirement (B). Tables IV and V show that SHAKE* always underestimates in this case the correct value of the friction coefficient.

V. CONCLUSION

We described constraint algorithms for SD simulations of arbitrary shape particles, rigid or flexible, consisting of spheres connected by appropriate constraints and with HI obtained by any of a number of available mobility schemes for spheres. The optimal algorithm HSHAKE [23] was verified with simulations of nonspherical particles and chains, which employed different forms of constraints and numerical integrators. In the simulations of rigid particles, two simple shapes were considered, but simulations with other shapes are possible with appropriate internal coordinate constraints, perhaps even using the simple shapes as building blocks. Similarly, simulations of chains of nonspherical particles or simulations of other flexible particles are feasible. Additional simulation results obtained with HSHAKE are described elsewhere [40]. Finally we note that a situation may arise where the analytical development of a mobility algorithm tailored to a specific nonspherical particle shape (e.g., oblate or prolate spheroids), as discussed in Sec. I, is being considered. Even in this case, we have shown that the approach described here provides a complementary quick means of generating preliminary simulation results for the approximate particle shape and situation at hand, thus aiding to decide whether to pursue the more involved analytical approach.

APPENDIX: DRIFT CORRECTION

We describe a method for eliminating the constraint drift from the direct method at every time step. $\forall\, i \in P_c$, we add to Eq. (8) a term containing l unknown η's,

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\eta) &= \mathbf{r}_i(t_0+\delta t,\lambda(t_0)) \\ &\quad - \delta t \sum_{k=1}^{l} \eta_k \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j \sigma_k](t_0) \\ &= \mathbf{r}_i(t_0+\delta t,\lambda(t_0)) + \delta\mathbf{r}_i(t_0+\delta t,\eta) \\ &\qquad (\forall\, i \in P_c). \qquad \text{(A1)}\end{aligned}$$

To compute the coordinates, the $\mathbf{r}_i(t_0+\delta t,\lambda(t_0))$ are generated by the direct method, and the $\delta\mathbf{r}$, and therefore the η's, chosen to satisfy Eq. (2) at every time step. Therefore, inserting Eq. (A1) into Eq. (2) gives

$$\begin{aligned}\sigma_k(\mathbf{r}(t_0+\delta t,\eta)) = \sigma_k\Bigg(&\mathbf{r}(t_0+\delta t,\lambda(t_0)) \\ &- \delta t \sum_{k'=1}^{l} \eta_{k'} \sum_{j=1}^{N} \boldsymbol{\mu}_{\{\}j}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j \sigma_{k'}](t_0)\Bigg) \\ = 0 \quad (k=1,\ldots,l). \qquad \text{(A2)}\end{aligned}$$

The η's obtained by solving Eq. (A2) are substituted into Eq. (A1) to provide the final $\mathbf{r}_i$ $\forall\, i \in P_c$. We set $\mathbf{r}_i(t_0+\delta t,\eta) = \mathbf{r}_i(t_0+\delta t,\lambda(t_0))$ $\forall\, i \notin P_c$. Consequently, the first part of the coupled technique or of HSHAKE [23] is used to compute the η's, replacing $\mathbf{r}_i'(t_0+\delta t)$, $\mathbf{r}_i(t_0+\delta t,\gamma)$, and $\delta\mathbf{r}_i(t_0+\delta t,\gamma)$ by $\mathbf{r}_i(t_0+\delta t,\lambda(t_0))$, $\mathbf{r}_i(t_0+\delta t,\eta)$, and $\delta\mathbf{r}_i(t_0+\delta t,\eta)$, respectively

The drift correction method approximates $\lambda_k(t_0)$ by $[\lambda_k(t_0)+\eta_k]$ $\forall\, i \in P_c$, as seen by comparing Eq. (8) with Eq. (A1). Accordingly, replacing β_k by $[\lambda_k(t_0)+\eta_k]$, Eq. (15) becomes $[\lambda_k(t_0)+\eta_k] = \lambda_k(t_0) + O(\delta t^{m-a})$, keeping in mind the equality is only approximate because the β's are estimates of the λ's. Inserting this expression back into Eq. (A1), the $\mathbf{r}_i$ from the correction method can be written as

$$\begin{aligned}\mathbf{r}_i(t_0+\delta t,\eta) = \mathbf{r}_i(t_0) &+ \delta t \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot\mathbf{F}_j(t_0) \\ &- \delta t \sum_{k=1}^{l} \lambda_k(t_0) \sum_{j=1}^{N} \boldsymbol{\mu}_{ij}^{\mathrm{tt}}(t_0)\cdot[\boldsymbol{\nabla}_j \sigma_k](t_0) \\ &+ O(\delta t^{m+1}) \quad (\forall\, i \in P_c). \qquad \text{(A3)}\end{aligned}$$

Comparing Eq. (A3) with Eq. (8) $\forall\, i \in P_c$, we can write

$$\mathbf{r}_i[\text{direct}+\text{correction}] = \mathbf{r}_i[\text{direct}] + O(\delta t^{m+1}) \quad (\forall\, i \in P_c). \qquad \text{(A4)}$$

The unconstrained $\mathbf{r}_i$ are unmodified by the correction scheme, hence

$$\mathbf{r}_i[\text{direct}+\text{correction}] = \mathbf{r}_i[\text{direct}] \quad (\forall\, i \notin P_c). \qquad \text{(A5)}$$

Inserting Eq. (18) $\forall\, i \in P_c$ into Eq. (A4) and $\forall\, i \notin P_c$ into Eq. (A5), and combining the resulting equations gives Eq. (19) with method = "direct + correction" and $p = m+1$. Comparison with Eq. (18) shows that the [direct + correction] method satisfies requirement (B). It satisfies also requirement (A) by construction.

[1] B. Cichocki, R.B. Jones, R. Kutteh, and E. Wajnryb, J. Chem. Phys. **112**, 2548 (2000).
[2] R.B. Jones and R. Kutteh, Phys. Chem. Chem. Phys. **1**, 2131 (1999).
[3] R.B. Jones and R. Kutteh, J. Chem. Phys. **112**, 11 080 (2000).
[4] J.C. Mallinson, *The Foundations of Magnetic Recording*, 2nd ed. (Academic Press, San Diego, 1993).
[5] R. Kutteh and P.B. Visscher, in *Computer Simulation Studies in Condensed Matter Physics*, edited by D.P. Landau, S.P. Lewis, and H.B. Schüttler (Springer-Verlag, Heidelberg, 2001), Vol. XIV.
[6] G.K. Youngren and A. Acrivos, J. Fluid Mech. **69**, 377 (1975).
[7] J. Vincent, N. Phanthien, and T. Trancong, J. Rheol. **35**, 1 (1991).
[8] S.T. Kim, Y.O. Fuentes, and S.J. Karrila, J. Stat. Phys. **62**, 1197 (1991).
[9] P. Pakdel and S. Kim, J. Rheol. **35**, 797 (1991).
[10] M.J. Gluckman, R. Pfeffer, and S. Weinbaum, J. Fluid Mech. **50**, 705 (1971).
[11] M.J. Gluckman, S. Weinbaum, and R. Pfeffer, J. Fluid Mech. **55**, 677 (1972).
[12] W.H. Liao and D.A. Krueger, J. Fluid Mech. **96**, 223 (1980).
[13] Q. Hassonjee, P. Ganatos, and R. Pfeffer, J. Fluid Mech. **197**, 1 (1988).
[14] I.L. Claeys and J.F. Brady, J. Fluid Mech. **251**, 411 (1993).
[15] A.J.C. Ladd, J. Chem. Phys. **93**, 3484 (1990).
[16] A.S. Sangani and G. Mo, Phys. Fluids **8**, 1990 (1996).
[17] B. Cichocki, B.U. Felderhof, K. Hinsen, E. Wajnryb, and J. Blawzdziewicz, J. Chem. Phys. **100**, 3780 (1994).
[18] J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[19] J.F. Brady and G. Bossis, Annu. Rev. Fluid Mech. **20**, 111 (1988).
[20] S. Bhattacharya and J. Blawzdziewicz, J. Math. Phys. **43**, 5720 (2002).
[21] J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, J. Comput. Phys. **23**, 327 (1977).
[22] R. Kutteh and T. P. Straatsma, in *Reviews in Computational Chemistry*, edited by K.B. Lipkowitz and D.B. Boyd (Wiley, New York, 1998), Vol. 12.
[23] A HSHAKE FORTRAN routine for a system of spheres with distance, angle, and torsion constraints, and possibly unconstrained spheres, usable with any algorithm for computing the hydrodynamic mobility matrix of spheres, is available from the author. Its iterative first part serves alternatively as a drift correction scheme for the direct method of Sec. II.
[24] J. Happel and H. Brenner, *Low Reynolds Number Hydrodynamics* (Prentice-Hall, New Jersey, 1965).
[25] R. Kutteh, J. Chem. Phys. **111**, 1394 (1999).
[26] R. Kutteh, Comput. Phys. Commun. **119**, 159 (1999).
[27] R. Kutteh and R.B. Jones, Phys. Rev. E **61**, 3186 (2000).
[28] S.A. Allison and J.A. McCammon, Biopolymers **23**, 167 (1984).
[29] H.C. Öttinger, Phys. Rev. E **50**, 2696 (1994).
[30] E.B. Wilson, Jr., J.C. Decius, and P.C. Cross, *Molecular Vibrations* (Dover, New York, 1955).
[31] S. Kim, Int. J. Multiphase Flow **11**, 699 (1985).
[32] R. Hsu and P. Ganatos, J. Fluid Mech. **268**, 267 (1994).
[33] H. Binous and R.J. Phillips, J. Non-Newtonian Fluid Mech. **85**, 63 (1999).
[34] G.K. Batchelor, J. Fluid Mech. **44**, 419 (1970).
[35] H. Brenner, J. Fluid Mech. **12**, 35 (1962).
[36] G. Bossis, A. Meunier, and J.D. Sherwood, Phys. Fluids A **3**, 1853 (1991).
[37] A. Meunier, J. Phys. II **4**, 561 (1994).
[38] S. Fraden, A.J. Hurd, and R.B. Meyer, Phys. Rev. Lett. **63**, 2373 (1989).
[39] M.C. Miguel and J.M. Rubi, J. Chem. Phys. **110**, 6007 (1999).
[40] R. Kutteh, J. Chem. Phys. **119**, 9280 (2003).