Accelerated Stokesian dynamics: Brownian motion

Adolfo J. Banchio[a] and John F. Brady[b]

Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125

A new Stokesian dynamics (SD) algorithm for Brownian suspensions is presented. The implementation is based on the recently developed accelerated Stokesian dynamics (ASD) simulation method [Sierou and Brady, J. Fluid Mech. **448**, 115 (2001)] for non-Brownian particles. As in ASD, the many-body long-range hydrodynamic interactions are computed using fast Fourier transforms, and the resistance matrix is inverted iteratively, in order to keep the computational cost $O(N \log N)$. A fast method for computing the Brownian forces acting on the particles is applied by splitting them into near- and far-field contributions to avoid the $O(N^3)$ computation of the square root of the full resistance matrix. For the near-field part, representing the forces as a sum of pairwise contributions reduces the cost to $O(N)$; and for the far-field part, a Chebyshev polynomial approximation for the inverse of the square root of the mobility matrix results in an $O(N^{1.25} \log N)$ computational cost. The overall scaling of the method is thus roughly of $O(N^{1.25} \log N)$ and makes possible the simulation of large systems, which are necessary for studying long-time dynamical properties and/or polydispersity effects in colloidal dispersions. In this work the method is applied to study the rheology of concentrated colloidal suspensions, and results are compared with conventional SD. Also, a faster approximate method is presented and its accuracy discussed. © *2003 American Institute of Physics.* [DOI: 10.1063/1.1571819]

I. INTRODUCTION

The understanding and prediction of the rheology and dynamics of Brownian suspensions has been (and still is) a subject of both fundamental and technological interest. In the last decade, considerable effort has been made to investigate, experimentally and theoretically, the structure and dynamics of colloidal suspensions. The equilibrium structure and short-time dynamics of monodisperse suspensions have been thoroughly studied and, in many aspects, are well understood.[1,2] The long-time dynamics, rheology, and the nonequilibrium structure, on the other hand, are still the subject of ongoing research. The presence of many-body hydrodynamic interactions (HI) greatly complicates the theoretical treatment of these systems, and for this reason the development of simulation techniques capable of addressing the hydrodynamic interactions has been a significant advance. In particular, since its first implementation for Brownian suspensions, the Stokesian dynamics (SD) simulation method has been successfully applied for more than ten years to a wide range of problems, such as the short-time diffusion and rheology of Brownian suspensions, as well as the nonequilibrium properties of suspensions under flow[3–7] to mention but a few. Long-time dynamics and large-scale structure formation, however, were normally out of the scope of SD simulations owing to the large computational costs involved [an $O(N^3)$ scheme, where N is the number of particles in the system], which limited its use to small systems (typically 27–125 particles in the simulation box).

Recently, Sierou and Brady[8] presented the accelerated Stokesian dynamics (ASD) algorithm for nonBrownian suspensions, a new implementation of the SD method in which the computational cost scales as $O(N \log N)$. This new scheme allows simulations of the order of 1000 or more particles, and makes possible the study of highly concentrated suspensions, large-scale structure formation, etc. The method can also be readily extended to study nonspherical particles and mixtures or polydispersity effects.

The aim of this work is to develop a new SD simulation technique for Brownian suspensions in which the computational cost scales roughly as $O(N^{1.25} \log N)$. This will be achieved by extending the ASD code for the efficient treatment of Brownian forces, which in the original SD implementation involved $O(N^3)$ operations. This new algorithm will now make possible the simulation of much larger systems, and will also allow much longer simulations (on the particle time scale), which is necessary for assessing long-time dynamics and time-correlation functions.

This article is organized as follows: In Sec. II, we describe in detail the new simulation method. Results for the rheology of concentrated colloidal suspensions are presented in Sec. III. Conclusions are given in Sec. IV.

II. METHOD

First, we briefly review the conventional Stokesian dynamics algorithm for Brownian suspensions, emphasizing the steps in which the computational cost is larger than

[a]Present address: Facultad de Matemática, Astronomía y Física, Universidad Nacional de Códoba, Ciudad Universitaria, 5000 Córdoba, Argentina.
[b]Electronic mail: jfbrady@caltech.edu

$O(N)$. We will then discuss the changes introduced in the ASD algorithm, and finally we describe in detail the further modifications needed for the inclusion of Brownian motion in order to keep the scaling of the computations as low as possible.

For a system of N rigid particles (of characteristic size a) suspended in an incompressible Newtonian fluid of viscosity η_0 and density ρ, the particle motion is governed by the coupled N-body Langevin equation:

$$\mathbf{m} \cdot \frac{d\mathbf{U}}{dt} = \mathbf{F}^h + \mathbf{F}^b + \mathbf{F}^p, \tag{1}$$

where $\mathbf{m}$ is the generalized mass/moment of inertia tensor, $\mathbf{U}$ is the particle translational/rotational velocity vector, and on the right-hand side are the forces/torques acting on the particles, which for convenience we split into: hydrodynamic forces ($\mathbf{F}^h$) due to the motion of the particles relative to the fluid; the stochastic forces ($\mathbf{F}^b$) responsible for the Brownian motion; and deterministic nonhydrodynamic forces ($\mathbf{F}^p$), which may be either interparticle or external.

When the particle Reynolds number is small, the hydrodynamic forces/torques acting on the particles in a suspension undergoing a bulk linear flow are given by

$$\mathbf{F}^h = -\mathbf{R}_{FU} \cdot (\mathbf{U} - \mathbf{u}^\infty) + \mathbf{R}_{FE} : \mathbf{E}^\infty. \tag{2}$$

Here, $\mathbf{u}^\infty$ is the velocity of the bulk linear flow evaluated at the center of a particle, $\mathbf{u}^\infty(\mathbf{x}) = \dot{\mathbf{\Gamma}} \cdot \mathbf{x}$, where $\dot{\mathbf{\Gamma}}$ is the bulk velocity gradient tensor, with $\mathbf{E}^\infty$ its rate of strain. The resistance tensors $\mathbf{R}_{FU}(\mathbf{x})$ and $\mathbf{R}_{FE}(\mathbf{x})$ give the hydrodynamic forces/torques on the particles due to their motion relative to the fluid and due to the imposed flow, respectively. $\mathbf{R}_{FU}(\mathbf{x})$ and $\mathbf{R}_{FE}(\mathbf{x})$ depend on the particle positions (both translational and orientational), represented by the generalized configuration vector $\mathbf{x}$.

The deterministic, nonhydrodynamic forces $\mathbf{F}^p$ are arbitrary, and we assume that they can be computed in $O(N)$ operations. This might not be the case when particles interact via a long-range potential with no cutoff. Long-range forces of, e.g., electrostatic origin, however, may be evaluated in $O(N \log N)$ operations by following a method similar to the one used in ASD for the hydrodynamic interactions.

The Brownian force $\mathbf{F}^b$ arises from the thermal fluctuations in the fluid and is a Gaussian stochastic variable, defined by

$$\overline{\mathbf{F}^b} = 0, \quad \overline{\mathbf{F}^b(0)\mathbf{F}^b(t)} = 2kT\mathbf{R}_{FU}\delta(t), \tag{3}$$

where the overbar denotes an ensemble average over the thermal fluctuations in the fluid, k is Boltzmann's constant, T is the absolute temperature, and $\delta(t)$ is the delta function. The correlation at 0 and t is a consequence of the fluctuation–dissipation theorem for the N-particle system.

In the conventional SD algorithm, an evolution equation for the particle configuration is obtained by integrating Eq. (1) twice over a time step Δt (larger than the inertial relaxation time, $\tau_B = m/6\pi\eta_0 a$, but small compared with the time over which the configuration changes), leading to

$$\Delta\mathbf{x} = \{\mathbf{u}^\infty + \mathbf{R}_{FU}^{-1} \cdot [\mathbf{R}_{FE} : \mathbf{E}^\infty + \mathbf{F}^p]\}\Delta t + kT\nabla \cdot \mathbf{R}_{FU}^{-1}\Delta t$$
$$+ \mathbf{X}(\Delta t) + o(\Delta t), \tag{4}$$

with

$$\overline{\mathbf{X}} = 0, \quad \overline{\mathbf{X}(t)\mathbf{X}(t)} = 2kT\mathbf{R}_{FU}^{-1}\Delta t. \tag{5}$$

Here, $\Delta\mathbf{x}$ is the change in particle position during the time step Δt and $\mathbf{X}(t)$ is a Gaussian random displacement computed as

$$\mathbf{X}(t) = \sqrt{2}kT\Delta t\mathbf{A} \cdot \mathbf{\Psi}(t)$$

with

$$\mathbf{A} \cdot \mathbf{A}^T = \mathbf{R}_{FU}^{-1}, \tag{6}$$

where $\mathbf{\Psi}(t)$ is a normal deviate, i.e., $\overline{\mathbf{\Psi}(t)} = 0$ and $\overline{\mathbf{\Psi}(t)\mathbf{\Psi}(t)} = \mathbf{1}$. In conventional SD, the matrix $\mathbf{A}$ was obtained as a byproduct of the inversion of $\mathbf{R}_{FU}$, and did not result in any additional computational cost.

To complete our brief description of the SD algorithm we still have to discuss how the resistance matrices are calculated. For this purpose, we define the grand resistance matrix $\mathcal{R}$ as

$$\mathcal{R} = \begin{bmatrix} \mathbf{R}_{FU} & \mathbf{R}_{FE} \\ \mathbf{R}_{SU} & \mathbf{R}_{SE} \end{bmatrix}, \tag{7}$$

where $\mathbf{R}_{SU}$ and $\mathbf{R}_{SE}$ are similar to $\mathbf{R}_{FU}$ and $\mathbf{R}_{FE}$ and relate the particle stresslet, $\mathbf{S}$—the symmetric first moment of the force density on a particle—to the velocity and the rate of strain by

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{S} \end{bmatrix} = -\mathcal{R} \cdot \begin{bmatrix} \mathbf{U} - \mathbf{u}^\infty \\ -\mathbf{E}^\infty \end{bmatrix} + \begin{bmatrix} \mathbf{F}^b + \mathbf{F}^p \\ -\mathbf{r}\mathbf{F}^p \end{bmatrix}. \tag{8}$$

The inverse of the grand resistance matrix is the grand mobility matrix $\mathcal{M}$ and gives the particle velocities and the rate of strain ($\mathbf{U} - \mathbf{u}^\infty, \mathbf{E}^\infty$) in terms of the total forces/torques and stresslets ($\mathbf{F}, \mathbf{S}$).

In Eq. (8) the definition of the stresslet contribution due to the interparticle forces as $\mathbf{r}\mathbf{F}^p$ may need clarification. First, it is assumed that the interparticle forces do not result in a net force on the suspension, i.e., $\Sigma\mathbf{F}^p = 0$, where Σ stands for a sum over all particles. A net force should be considered an external force and generates bulk translational motion, not stress. Thus, $\mathbf{r}\mathbf{F}^p$ is independent of the origin and the sum over all particles and over all unit cells can be performed. In the special case in which the forces are pairwise additive, the stresslet can be written as $(1/2)(1/N)\Sigma_{ij}\mathbf{r}_{ij}\mathbf{F}_{ij}^p$, where the sum is over all particles, $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$, and $\mathbf{F}_{ij}^p$ is the interparticle force for the pair ij. If the forces are not pairwise additive, as in electrorheological fluids for example, then the more general form implied by $\mathbf{r}\mathbf{F}^p$ must be used. Care must also be exercised when only summing over a single unit cell; the nearest image must be used.

Conventional SD exploits the fact that hydrodynamic interactions among particles can be decomposed into long-range mobility interactions and short-range lubrication interactions and splits the grand resistance matrix according to

$$\mathcal{R} = (\mathcal{M}^\infty)^{-1} + \mathcal{R}_{\mathrm{nf}}, \tag{9}$$

with

$$\mathcal{R}_{\mathrm{nf}} = \mathcal{R}_{2B} - \mathcal{R}_{2B}^\infty = \begin{bmatrix} \mathbf{R}_{FU,\mathrm{nf}} & \mathbf{R}_{FE,\mathrm{nf}} \\ \mathbf{R}_{SU,\mathrm{nf}} & \mathbf{R}_{SE,\mathrm{nf}} \end{bmatrix}. \tag{10}$$

J. Chem. Phys., Vol. 118, No. 22, 8 June 2003

Accelerated Stokesian dynamics: Brownian motion   10325

Here, $\mathcal{M}^{\infty}$ is the grand mobility matrix from the infinite (via periodic boundary conditions) system formed by a truncated multipole expansion in combination with Faxén laws, and $\mathcal{R}_{nf}$ contains the near-field lubrication interactions. $\mathcal{R}_{nf}$ is formed in a pairwise additive fashion from the exact two-body resistance interactions, $\mathcal{R}_{2B}$, minus the two-body interactions already included in $(\mathcal{M}^{\infty})^{-1}$, denoted as $\mathcal{R}_{2B}^{\infty}$.

Due to the near-field character of lubrication interactions, only pairs of particles that are closer than a certain cutoff distance (typically 4 particle radii) have nonzero entries in $\mathcal{R}_{nf}$. For this reason, $\mathcal{R}_{nf}$ is a sparse matrix, and its construction involves only $O(N)$ operations. $\mathcal{M}^{\infty}$, on the other hand, is not sparse, and its construction requires $O(N^2)$ calculations.

From Eqs. (4)–(6), and the above-presented discussion, we can identify the computations in conventional SD that require $O(N^2)$ or larger operations:

(i) $\mathcal{M}^{\infty}$: its construction is of $O(N^2)$.
(ii) $\mathbf{R}_{FU}^{-1}$: matrix inversion of $O(N^3)$.
(iii) $\nabla \cdot \mathbf{R}_{FU}^{-1}$: uses $\mathbf{R}_{FU}^{-1}$ and $\mathcal{M}^{\infty}$.
(iv) $\mathbf{X}$: uses the factorization: $\mathbf{A} \cdot \mathbf{A}^T = \mathbf{R}_{FU}^{-1}$, an $O(N^3)$ operation.

In order to obtain a method with a more favorable scaling, we need to devise alternate approaches to avoid these computations, or to compute them in a more efficient manner.

In ASD Sierou and Brady[8] developed an alternative algorithm that avoided the explicit computation of $\mathcal{M}^{\infty}$ and the inversion of $\mathbf{R}_{FU}^{-1}$, the computationally more expensive parts in absence of Brownian motion. In what follows, we briefly describe their approach and then present different schemes for addressing (iii) and (iv) in order to keep the overall scaling as low as possible.

A. Far-field hydrodynamics: $\mathcal{M}^{\infty}$

In the Stokesian dynamics algorithm the far-field many-body hydrodynamic interactions are accounted (in the resistance formulation) by the inverse of the far-field mobility matrix $\mathcal{M}^{\infty}$. As already mentioned, the explicit computation of $\mathcal{M}^{\infty}$ involves $O(N^2)$ operations; for this reason, we avoid computing $\mathcal{M}^{\infty}$ explicitly and, instead, find a way to compute $\mathcal{M}^{\infty} \cdot \mathbf{y}$, for a given (far-field) force/torque/stresslet vector $\mathbf{y} = (\mathbf{f}_{ff}, \mathbf{t}_{ff}, \mathbf{s}_{ff})$.

For this purpose, Sierou and Brady use Faxén laws to obtain an expression for the particle velocities and rate of strain as a function of the particles' force/torque/stresslet and the far-field fluid velocity and its rate of strain at the center of the particles, namely,

$$\mathbf{u} - \mathbf{u}^{\infty}(\mathbf{x}) = -\frac{1}{6\pi\eta_0 a}\mathbf{f}_{ff} + \left(1 + \frac{a^2}{6}\nabla^2\right)\mathbf{u}_{ff}, \tag{11a}$$

$$\boldsymbol{\omega} - \boldsymbol{\omega}^{\infty}(\mathbf{x}) = -\frac{1}{8\pi\eta_0 a^3}\mathbf{t}_{ff} + \frac{1}{2}\nabla\times\mathbf{u}_{ff}, \tag{11b}$$

$$-\mathbf{E}^{\infty} = -\frac{3}{20\pi\eta_0 a^3}\mathbf{s}_{ff} + \left(1 + \frac{a^2}{10}\nabla^2\right)\mathbf{e}_{ff}. \tag{11c}$$

Here, $\mathbf{u}_{ff}$ corresponds to the far-field fluid velocity evaluated at the center of the particle, and $\mathbf{e}_{ff}$ is its rate of strain, arising

from the presence of all other particles (represented as force/force-moment densities). Both are functions of the far-field force/torque/stresslet vector $(\mathbf{f}_{ff}, \mathbf{t}_{ff}, \mathbf{s}_{ff})$, In Eqs. (11a) and (11b) $\mathbf{u}$ and $\boldsymbol{\omega}$ stand for the particle translational and rotational velocity, respectively, i.e., $\mathbf{U} = (\mathbf{u}, \boldsymbol{\omega})$. We will further omit the details on how $\mathbf{u}_{ff}$ and $\mathbf{e}_{ff}$ are obtained from the force/torque/stresslet $\mathbf{y}$; we only mention that this can be done in $O(N\log N)$ by representing the particle moments as forces on a grid, and then solving for the far-field velocity field on that grid using fast Fourier transforms and the particle-mesh-Ewald technique. For a detailed description see Ref. 8.

B. Computing particle velocities: R_{FU}^{-1}

As is well known, the most efficient way of solving a large sparse linear system of equations is using iterative methods. Conjugate gradient methods have proven to be a powerful technique for solving large linear systems, and they share the advantage of referencing the matrix only through its product with a vector, a necessary condition in our case, since we know $\mathcal{M}^{\infty}$ only through its action on the vector $\mathbf{y}$.

Not having $\mathbf{R}_{FU}$ explicitly, forces us to solve for the velocities and (far-field) stresslet simultaneously. In the absence of particle inertia (particle inertia can be easily included if desired), Eqs. (1), (2), (7), and (8) give

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{S} \end{bmatrix} = -\mathcal{R} \cdot \begin{bmatrix} \mathbf{U} - \mathbf{u}^{\infty} \\ -\mathbf{E}^{\infty} \end{bmatrix} + \begin{bmatrix} \mathbf{F}^b + \mathbf{F}^p \\ -\mathbf{r}\mathbf{F}^p \end{bmatrix}. \tag{12}$$

Note that the unknowns are $\mathbf{U}$ and $\mathbf{S}$. We further split $\mathbf{U} - \mathbf{u}^{\infty}$ and $\mathbf{S}$ into a so-called hydrodynamic part, $\mathbf{U}^h$ and $\mathbf{S}^h$, a Brownian part, $\mathbf{U}^b$ and $\mathbf{S}^b$, and an interparticle force part, $\mathbf{U}^p$ and $\mathbf{S}^p$, satisfying the following:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{S}^h \end{bmatrix} = -\mathcal{R} \cdot \begin{bmatrix} \mathbf{U}^h \\ -\mathbf{E}^{\infty} \end{bmatrix}, \tag{13}$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{S}^b \end{bmatrix} = -\mathcal{R} \cdot \begin{bmatrix} \mathbf{U}^b \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{F}^b \\ \mathbf{0} \end{bmatrix}, \tag{14}$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{S}^p \end{bmatrix} = -\mathcal{R} \cdot \begin{bmatrix} \mathbf{U}^p \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{F}^p \\ -\mathbf{r}\mathbf{F}^p \end{bmatrix}. \tag{15}$$

By doing this, we can compute the particle motion and the rheological properties arising from the Brownian motion separately from the hydrodynamic and interparticle force contributions. The extra cost of solving Eq. (13) [and/or Eq. (15); actually, they are combined together in one step] is usually not considerable since the hydrodynamic quantities change slowly and one can use the solution from the previous time step as an initial guess, leading to convergence in a few iterations. Solving Eq. (14), on the other hand, must be done anew at each time step, since $\mathbf{F}^b$ and $\mathbf{S}^b$ are completely uncorrelated (as they must be) from one time step to the next.

Note that the splitting defined in Eqs. (13)–(15) corresponds to the forces in Eq. (1). The linearity of the hydrodynamic forces at low Reynolds number allows one to superimpose effects. The physical interpretation is that $\mathbf{S}^h$ is directly proportional to the rate of strain $\mathbf{E}^{\infty}$, $\mathbf{S}^b$ to the Brownian forces $\mathbf{F}^b$, and $\mathbf{S}^p$ to the interparticle forces $\mathbf{F}^p$.

For example, in a flow cessation experiment, $\mathbf{E}^\infty$ is suddenly set to zero and the hydrodynamic stresslet $\mathbf{S}^h$ drops instantaneously to zero, while $\mathbf{S}^b$ and $\mathbf{S}^p$ relax to zero as the structure relaxes back to equilibrium. Calling $\mathbf{S}^h$ the hydrodynamic stresslet is actually a misnomer, since both $\mathbf{S}^b$ and $\mathbf{S}^p$ contain direct contributions due to hydrodynamic interactions (indeed $\mathbf{S}^b$ is purely hydrodynamic in origin); a better name would be $\mathbf{S}^E$, indicating its proportionality to the rate of strain. Note also, that each stresslet depends on the structure (i.e., the configuration of particles), which results from the combined effects of hydrodynamic, Brownian, and interparticle forces and this ''indirect'' effect is not superposable.

Using Eqs. (9), (10), and (13), we find the following equation for $\mathbf{U}^h$,

$$\mathbf{R}_{FU,\text{nf}} \cdot \mathbf{U}^h + \mathbf{F}_{\text{ff}}^h = \mathbf{R}_{FE,\text{nf}} : \mathbf{E}^\infty, \tag{16}$$

with the far-field hydrodynamic forces/torques, $\mathbf{F}_{\text{ff}}^h$, and stresslets, $\mathbf{S}_{\text{ff}}^h$, satisfying

$$\mathcal{M}^\infty \cdot \begin{bmatrix} \mathbf{F}_{\text{ff}}^h \\ \mathbf{S}_{\text{ff}}^h \end{bmatrix} = \begin{bmatrix} \mathbf{U}^h \\ -\mathbf{E}^\infty \end{bmatrix}. \tag{17}$$

Replacing $\mathbf{U}^h$ in Eq. (17) from Eq. (16) and rearranging terms leads to the linear system of equations for $\mathbf{F}_{\text{ff}}^h$ and $\mathbf{S}_{\text{ff}}^H$:

$$\widetilde{\mathcal{M}}^\infty \cdot \begin{bmatrix} \mathbf{F}_{\text{ff}}^h \\ \mathbf{S}_{\text{ff}}^h \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{FU,\text{nf}}^{-1} \cdot \mathbf{F}_{\text{nf}}^h \\ -\mathbf{E}^\infty \end{bmatrix}, \tag{18}$$

where

$$\widetilde{\mathcal{M}}^\infty = \mathcal{M}^\infty + \begin{bmatrix} \mathbf{R}_{FU,\text{nf}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{19}$$

and

$$\mathbf{F}_{\text{nf}}^h = \mathbf{R}_{FE,\text{nf}} : \mathbf{E}^\infty. \tag{20}$$

Note that $\mathbf{U}^h$ does not appear explicitly in Eq. (18), but is computed when evaluating the left-hand side.

To solve the linear system of equations (18) we use a generalized conjugate gradient method, and for computing the action of $\mathbf{R}_{FU,\text{nf}}^{-1}$ (needed to evaluate the right-hand side and to calculate the action of $\widetilde{\mathcal{M}}^\infty$) we employ an incomplete Cholesky preconditioned conjugate gradient method as described in Ref. 8. Note that we have two nested iterative solvers, a preconditioned conjugate gradient for the inner iterations (inversion of $\mathbf{R}_{FU,\text{nf}}$) and a generalized conjugate gradient for the outer iterations (inversion of $\widetilde{\mathcal{M}}^\infty$).

Having $\mathbf{F}_{\text{ff}}^h$ and $\mathbf{S}_{\text{ff}}^h$ satisfying Eq. (18), we can obtain the hydrodynamic contributions to the particle velocities and stresslets as

$$\mathbf{U}^h = \mathbf{R}_{FU,\text{nf}}^{-1} \cdot \mathbf{F}_{\text{nf}}^h - \mathbf{R}_{FU,\text{nf}}^{-1} \cdot \mathbf{F}_{\text{ff}}^h, \tag{21}$$

$$\mathbf{S}^h = -\mathbf{R}_{SU,\text{nf}} \cdot \mathbf{U}^h + \mathbf{R}_{SE,\text{nf}} : \mathbf{E}^\infty + \mathbf{S}_{\text{ff}}^h. \tag{22}$$

The operation count is $O(N \log N)$ (shown in the following) times the number of iterations necessary for convergence. Through the use of the preconditioner, a typical number of iterations for the inner solver is $O(20)$ (with a tolerance of 10^{-5} and a time step of 10^{-3}), for the outer solver the number of iterations needed is $O(25)$ (with a tolerance of 10^{-4} for $\phi = 0.45$, Pe$=1$). These representative

values are for the computation of the hydrodynamic velocity, and may be considerably reduced (down to a few) by using a larger tolerance (taking care that this does not alter the average results).

C. Brownian displacement: X

Unlike the deterministic velocity $\mathbf{U}^h$ in Eq. (13), the Brownian velocity $\mathbf{U}^b$ fluctuates with a characteristic time equal to the Brownian relaxation time τ_B, which usually is several orders of magnitude smaller than the time interval needed for the particles to move a significant fraction of their size. Here, we are only interested in the diffusive limit, i.e., when the time step is much larger than τ_B.

In order to compute the Brownian displacement $\mathbf{X}$ that satisfies Eq. (5), we determine the Brownian velocity $\mathbf{U}^b$ from Eq. (14). (A similar approach was suggested in Ref. 9.) Note that here, and in the rest of this paper, $\mathbf{U}^b$ (and also the random forces $\mathbf{F}^b$) are fixed during the time step.

The algorithm is as follows: First, a set of random forces/torques $\mathbf{F}^b$ is generated according to the discretized version of Eq. (3). Then, Eq. (14) for $\mathbf{U}^b$ is solved iteratively and finally $\mathbf{X}$ is given as $\mathbf{X} = \mathbf{U}^b \Delta t$. It is straightforward to show that $\mathbf{X}$ computed this way fulfills Eq. (5). To solve Eq. (14), we proceed as with Eq. (13). (For an alternative approach for low densities see Appendix A.)

In the rest of this section, we describe how to generate the random Brownian forces/torques $\mathbf{F}^b$, while keeping the overall scaling of the algorithm as low as possible.

The Brownian forces are split into a near- and a far-field part, according to

$$\mathbf{F}^b = \mathbf{F}_{\text{nf}}^b + \mathbf{F}_{\text{ff}}^b, \tag{23}$$

with

$$\overline{\mathbf{F}_{\text{nf}}^b} = \overline{\mathbf{F}_{\text{ff}}^b} = 0, \tag{24}$$

and

$$\overline{\mathbf{F}_{\text{nf}}^b \mathbf{F}_{\text{nf}}^b} = 2kT\mathbf{R}_{FU,\text{nf}}/\Delta t, \tag{25}$$

$$\overline{\mathbf{F}_{\text{ff}}^b \mathbf{F}_{\text{ff}}^b} = 2kT(\mathcal{M}^\infty)_{FU}^{-1}/\Delta t, \tag{26}$$

$$\overline{\mathbf{F}_{\text{nf}}^b \mathbf{F}_{\text{ff}}^b} = 0. \tag{27}$$

Here, $(\mathcal{M}^\infty)_{FU}^{-1}$ represents the part of $\mathbf{R}_{FU}$ arising from the inverse of $\mathcal{M}^\infty$.

1. Brownian forces: Near-field part

To generate the near-field part of the random forces/torques, $\mathbf{F}_{\text{nf}}^b$, obeying Eqs. (24), (25), and (27) using $O(N)$ operations, we write the Brownian force/torque acting on particle i, $\mathbf{F}_{\text{nf},i}^b$ as a pairwise sum over all of its near neighbors according to

$$\mathbf{F}_{\text{nf},i}^b = \sum_{\substack{\gamma = (i,j) \\ |\mathbf{r}_i - \mathbf{r}_j| < 4a}} (\mathbf{L}_{ii}^\gamma \cdot \mathbf{\Psi}_{\gamma,i}^{\text{nf}} + \mathbf{L}_{ij}^\gamma \cdot \mathbf{\Psi}_{\gamma,j}^{\text{nf}}), \tag{28}$$

with

$$
\begin{bmatrix} \mathbf{L}_{ii}^{\gamma} & \mathbf{L}_{ij}^{\gamma} \\ \mathbf{L}_{ji}^{\gamma} & \mathbf{L}_{jj}^{\gamma} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{L}_{ii}^{\gamma} & \mathbf{L}_{ij}^{\gamma} \\ \mathbf{L}_{ji}^{\gamma} & \mathbf{L}_{jj}^{\gamma} \end{bmatrix}^T = 2kT(\mathbf{R}_{2B,FU,ij} - \mathbf{R}_{2B,FU,ij}^{\infty}),
$$

(29)

and $\boldsymbol{\Psi}_{\gamma,i}^{nf}$ is a random Gaussian deviate of length 6 for particle i in pair γ, satisfying

$$
\overline{\boldsymbol{\Psi}_{\gamma,i}^{nf}} = 0, \quad \overline{\boldsymbol{\Psi}_{\gamma,i}^{nf} \boldsymbol{\Psi}_{\nu,j}^{nf}} = \mathbf{1}\delta_{\gamma\nu}\delta_{ij}.
$$

(30)

In Eq. (29), $(\mathbf{R}_{2B,FU,ij} - \mathbf{R}_{2B,FU,ij}^{\infty})$ is the 12×12 two-body pairwise contribution to $\mathbf{R}_{FU,\mathrm{nf}}$ arising from pair $\gamma = (i,j)$, which we factorize using Cholesky decomposition to obtain the $\mathbf{L}_{ij}^{\gamma}$.

2. Brownian forces: Far-field part

For the far-field part of the Brownian forces/torques, we employ Fixman's method,[10,11] consisting of replacing the square root of $(\mathcal{M}^{\infty})^{-1}$ by its Chebyshev polynomial approximation, and exploit the fact that only its action on a vector is needed. Note that in our case, because we compute the random forces instead of the random displacements, we need to approximate the square root of the inverse of the (far-field) mobility matrix.

To generate a set of random Gaussian deviates with a given covariance matrix $\mathbf{M}$, the most common way is to factorize the covariance matrix as in Eq. (6). This factorization, however, is not unique and any choice of the factors satisfying $\mathbf{M} = \mathbf{A} \cdot \mathbf{A}^T$ is valid. In particular, one can choose $\mathbf{A}$ to be the square root of $\mathbf{M}$, i.e., $\mathbf{M} = \mathbf{A} \cdot \mathbf{A}$, with $\mathbf{A} = \mathbf{A}^T$. This has the advantage that a polynomial approximation can be used to approximate the square root.

In the case of the far-field Brownian forces/torques, $\mathbf{F}_{\mathrm{ff}}^b$, the covariance matrix is proportional to the FU block of the inverse of $\mathcal{M}^{\infty}$ [cf. Eq. (26)]. Since we do not have $\mathcal{M}^{\infty}$ explicitly, and we only can compute its action on vector (a force/torque/stresslet vector), we generate $\mathbf{F}_{\mathrm{ff}}^b$ as the forces/torques part of

$$
\begin{bmatrix} \mathbf{F}_{\mathrm{ff}}^b \\ \mathbf{S}_{\mathrm{ran}} \end{bmatrix} = \sqrt{\frac{2kT}{\Delta t}} (\mathcal{M}^{\infty})^{-1/2} \cdot \begin{bmatrix} \boldsymbol{\Psi}_F^{\mathrm{ff}} \\ \boldsymbol{\Psi}_S^{\mathrm{ff}} \end{bmatrix}.
$$

(31)

Here, $\boldsymbol{\Psi}^{\mathrm{ff}} = [\boldsymbol{\Psi}_F^{\mathrm{ff}}, \boldsymbol{\Psi}_S^{\mathrm{ff}}]$ is an $11N$ normal deviate. If we further require each component of $\boldsymbol{\Psi}^{\mathrm{ff}}$ to be independent of any component of $\boldsymbol{\Psi}_{\gamma,i}^{nf}$, it is straightforward to show that $\mathbf{F}_{\mathrm{ff}}^b$ as defined in Eq. (31) fulfills Eqs. (24), (26), and (27). We will approximate the inverse of the square root of $\mathcal{M}^{\infty}$ in Eq. (31) by its polynomial approximation as described in the following.

The Chebyshev polynomial approximation of the scalar function $1/\sqrt{x}$ over the range $[\lambda_{\min}, \lambda_{\max}]$ reads

$$
1/\sqrt{x} \approx \sum_{j=0}^{N_{\mathrm{Cheb}}} a_j C_j(x),
$$

(32)

where $C_j(x)$ are the shifted Chebyshev polynomials, which have the following recursion relations:

$$
\begin{aligned}
C_{l+1}(y) &= 2yC_l(y) - C_{l-1}(y), \\
C_0(y) &= 1, \quad C_1(y) = y,
\end{aligned}
$$

(33)

with

$$
y = \frac{2x}{\lambda_{\max} - \lambda_{\min}} - \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}.
$$

(34)

The Chebyshev coefficients, a_j in Eq. (32), depend on $\lambda_{\min}$ and $\lambda_{\max}$, and are usually computed for an approximation of much larger order than N_{Cheb} to estimate the truncation error.[12]

The approximation in Eq. (32) can be generalized to a function of matrices, keeping the same Chebyshev coefficients a_j, if the eigenvalues of the matrix are bounded by $[\lambda_{\min}, \lambda_{\max}]$. This results in the following approximate expression for the action of $(\mathcal{M}^{\infty})^{-1/2}$ on a vector $\mathbf{y}$:

$$
(\mathcal{M}^{\infty})^{-1/2} \cdot \mathbf{y} \approx \mathbf{z} = \sum_{j=0}^{N_{\mathrm{Cheb}}} a_j C_j(\mathcal{M}^{\infty}) \cdot \mathbf{y}.
$$

(35)

The factors $C_j(\mathcal{M}^{\infty}) \cdot \mathbf{y}$ can be computed recursively, and involve evaluations of the action of $\mathcal{M}^{\infty}$ on vectors, which can be computed in $O(N \log N)$ operations (see Appendix B).

The number of terms, N_{Cheb}, to be included in Eq. (35) to achieve the desired accuracy [typically of $O(10^{-2})$] depends on the ratio $\lambda_{\max}/\lambda_{\min}$. This dependence was found to scale roughly as $(\lambda_{\max}/\lambda_{\min})^{1/2}$, in agreement with similar findings for the square root of the mobility matrix in simulations of polymer chains.[10,11] Taking into account that the ratio $\lambda_{\max}/\lambda_{\min}$ was found to scale with the number of particles, N, roughly as $N^{1/2}$, the overall scaling of the computation of the far-field part of the Brownian forces is nearly of order $O(N^{1.25} \log N)$.

Before calculating the Chebyshev approximation of $(\mathcal{M}^{\infty})^{-1/2} \cdot \mathbf{y}$, we need to know the $\lambda_{\min}$ and $\lambda_{\max}$ bounding the eigenvalues of $\mathcal{M}^{\infty}$. Following Jendrejack *et al.*[11] and noting that if we assume we are using a valid eigenvalue range, we have

$$
\lim_{N_{\mathrm{Cheb}} \to \infty} [\mathbf{z} \cdot \mathcal{M}^{\infty} \cdot \mathbf{z}] = \mathbf{y} \cdot \mathbf{y},
$$

(36)

which allows us to define an estimate of the relative error resulting from the polynomial approximation as

$$
E_{\mathrm{Cheb}} = \sqrt{\frac{|\mathbf{z} \cdot \mathcal{M}^{\infty} \cdot \mathbf{z} - \mathbf{y} \cdot \mathbf{y}|}{\mathbf{z} \cdot \mathcal{M}^{\infty} \cdot \mathbf{z}}}.
$$

(37)

We compute E_{Cheb} for each configuration and use its value to decide whether or not it is necessary to compute a new eigenvalue range for the given configuration. The upper and lower eigenvalues are calculated in $O(N \log N)$ operations using the free software package ARPACK.[13]

D. Mean drift term: $\nabla \cdot \mathbf{R}_{FU}^{-1}$

The presence of the mean drift term in Eq. (4) is a consequence of the simple forward time-stepping integration scheme used. Replacing it by any higher order scheme, such as a mid-point or end-point algorithm, leads to an evolution equation without a mean drift term.[14] In this way, the explicit computation of $\nabla \cdot \mathbf{R}_{FU}^{-1}$, of order $O(N^3)$, may be completely avoided. However, the price for this is that one has to compute two velocities by iteratively inverting $\mathbf{R}_{FU}$ at each time step. (More sophisticated methods do not improve the accuracy because of the discontinuities in the random forces.)

We chose a slightly modified mid-point scheme for the computation of the Brownian velocity (and similarly for the Brownian stresses which also include divergences in their expressions; see the following). Instead of evaluating an intermediate velocity at the middle of the time step, we preferred to move the particles to an intermediate position by using a fraction of the time step $1/n$, with n typically on the order of 100. This is used to avoid particle "overlaps" in the intermediate configuration.

If at time t_0 the system has the configuration $\mathbf{x}_0$, and the corresponding grand resistance matrix of the system is $\mathcal{R}_0$, then we compute the Brownian velocity for a given set of random forces $\mathbf{F}^b$ (with $\overline{\mathbf{F}^b\mathbf{F}^b} = 2kT\mathcal{R}_0/\Delta t$) in the following manner.

(1) Compute the random velocity $\mathbf{U}_0^b$ (and $\mathbf{S}_0^b$) satisfying

$$\mathcal{R}_0 \cdot \begin{bmatrix} \mathbf{U}_0^b \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^b \\ -\mathbf{S}_0^b \end{bmatrix}. \tag{38}$$

(2) Move the particles to the intermediate position $\mathbf{x}'$, according to

$$\mathbf{x}' = \mathbf{x}_0 + \mathbf{U}_0^b \frac{\Delta t}{n}. \tag{39}$$

(3) Compute a new random velocity $\mathbf{U}^{b\prime}$ (and $\mathbf{S}^{b\prime}$) satisfying

$$\mathcal{R}' \cdot \begin{bmatrix} \mathbf{U}^{b\prime} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^b \\ -\mathbf{S}^{b\prime} \end{bmatrix}. \tag{40}$$

Note, that here $\mathcal{R}'$ represents the grand resistance matrix at the configuration $\mathbf{x}'$, and the forces $\mathbf{F}^b$ are the same as used in Eq. (38).

(4) Compute a realization of the mean drift and of the Brownian stress, $\overline{\mathbf{S}^b} = -kT\boldsymbol{\nabla}\cdot(\mathbf{R}_{SU}\cdot\mathbf{R}_{FU}^{-1})$, as

$$\mathbf{U}_{\text{drift}}^b \Delta t = \frac{n}{2}(\mathbf{U}^{b\prime} - \mathbf{U}_0^b)\Delta t, \tag{41}$$

$$\mathbf{S}^b = -\frac{n}{2}(\mathbf{R}_{SU}'\cdot\mathbf{U}^{b\prime} - \mathbf{R}_{SU}\cdot\mathbf{U}_0^b). \tag{42}$$

It is straightforward to show that

$$\overline{\mathbf{U}_{\text{drift}}^b} = kT\boldsymbol{\nabla}\cdot\mathbf{R}_{FU}^{-1} + o(\Delta t), \tag{43}$$

and

$$\overline{\mathbf{S}^b} = -kT\boldsymbol{\nabla}\cdot(\mathbf{R}_{SU}\cdot\mathbf{R}_{FU}^{-1}) + o(\Delta t), \tag{44}$$

as desired. From Eqs. (41) and (42), it is apparent that we preferred to numerically compute a realization of the drift term (and Brownian stress), instead of actually using a true mid-point-like scheme, which would have given the same results. We found this approach to reduce the *noise* arising from terms that average to zero in the computation of $\overline{\mathbf{S}^b}$.

Note that $\mathbf{U}_0^b$ in Eq. (41) is the Brownian velocity used for generating the Brownian displacement $\mathbf{X}$ (see Sec. II C).

For comparison, if one chooses $n=2$ and uses a mid-point scheme, the Brownian velocity would be

$$\mathbf{U}_{\text{mid}}^b = \tfrac{1}{2}(\mathbf{U}^{b\prime} + \mathbf{U}_0^b), \tag{45}$$

and would replace $\mathbf{U}_0^b + \mathbf{U}_{\text{drift}}^b$ in our scheme.

To solve Eqs. (38) and (40) we proceed as before for Eq. (13) (see Sec. II B and Appendix A).

E. Approximate method

As we show in the following, the above-presented method has a favorable scaling of roughly the order of $N^{1.25}\log N$ as desired, but the computational overhead for achieving this performance still renders this method a computationally intensive scheme, preventing, for the moment, its use for large systems on personal computers (PC). Parallelization and faster CPU speeds should in the near future allow one to take full advantage of the method. For this reason, we also have developed an approximate scheme that has the same favorable scaling with N, but that allows the study of large systems on a single PC.

The most time-consuming parts of the algorithm are the (iterative) inversions of the resistance matrix. In particular, solving for the Brownian velocities consumes most of the time, since the velocities of the previous time step may not be used as initial guesses (the Brownian forces are random and uncorrelated), in contrast to the hydrodynamic velocity, which is strongly correlated from one time step to the next. Furthermore, two Brownian velocities are needed at each time step, and needed with good accuracy in order to compute the mean drift and Brownian stress.

An alternative approximate faster algorithm is the following near-field scheme (ASDB-nf):

(1) For the hydrodynamic velocity $\mathbf{U}^h$ solve with the *full* resistance matrix (as in ASD). For consistency, all hydrodynamic quantities are computed with full hydrodynamic interactions.
(2) For the computation of the Brownian velocities (and other Brownian quantities) use a mean-field-like approximation for the hydrodynamic far-field interactions, i.e., replace $\mathcal{M}^\infty$ by a diagonal matrix with effective values that depend on volume fraction.
(3) For the diagonal approximation to $\mathcal{M}^\infty$ use the values of the translational and rotational short-time self-diffusivities for an equilibrium system in absence of near-field hydrodynamic interactions.

Replacing $\mathcal{M}^\infty$ by a diagonal matrix dramatically simplifies the computations since the inverse of this matrix (trivially obtained) can be added to $\mathbf{R}_{FU,\text{nf}}$, and then the iterative inversion can simply be done using the Cholesky preconditioned conjugate gradient algorithm.

To obtain the equilibrium translational and rotational short-time self-diffusivities in absence of near-field hydrodynamic interactions, we generated for each volume fraction a set of equilibrium configurations (obtained using molecular dynamics) and computed the diffusivities and the high frequency dynamic viscosities (η_∞') as an average over all configurations (here, we followed Sierou and Brady[8] for computing the self-diffusivities).

The translational self-diffusivity shows a strong $N^{-1/3}$ dependence,[15] and for this reason we used an extension of Ladd's[16] proposed expression to extract the infinite system
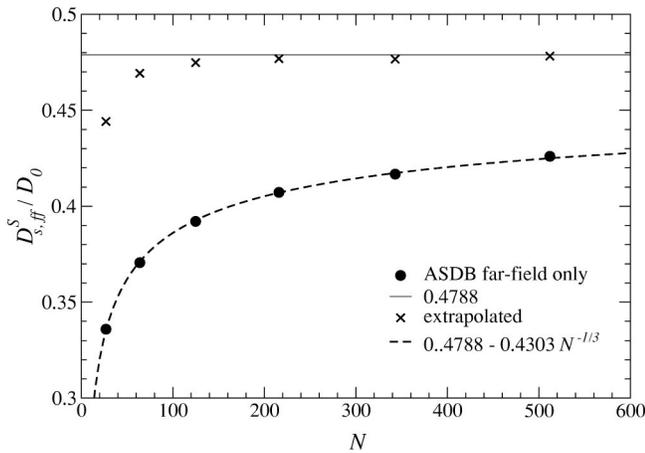
FIG. 1. Short-time self-diffusion coefficient in absence of near-field hydro-dynamic interactions vs the number of particles in the system. The closed circles are ASD results, the crosses are extrapolated values according to Eq. (46), the dotted line is a fitted curve of the form $a + bN^{-1/3}$, and the solid line represents the ordinate of this fit. Here, results are shown for $\phi = 0.45$.

FIG. 3. The CPU time (in arbitrary units) for 100 time steps of a shearing simulation at $\phi = 0.45$ and Pe = 1, starting from an equilibrium configuration. Dashed lines are a guide to the eye to show the approximate scalings (as labeled).

diffusivity from the N-particle periodic diffusion coefficient, which reads

$$D_{s,\text{ff}}^s = D_{s,\text{ff}}^s(N) + (\eta_0 / \eta_{\infty,\text{ff}}') D_0 [1.7601(\phi/N)^{1/3} - \phi/N]. \tag{46}$$

Here, we have replaced the self-diffusivities and the suspension viscosity from Ladd's expression by their counterparts in the absence of near-field interactions. In Fig. 1 we plot the translational self-diffusion coefficient versus number of particles for $\phi = 0.45$, compared with the corresponding extrapolated translational self-diffusivities from Eq. (46). For comparison, a fitted curve of the form $a + bN^{-1/3}$ is also shown; the value of a is shown by the solid line. Figure 1 shows that Eq. (46) gives a good estimate for the infinite system self-diffusivity.

Figure 2 shows the results obtained for the short-time translational and rotational self-diffusivities for different vol-
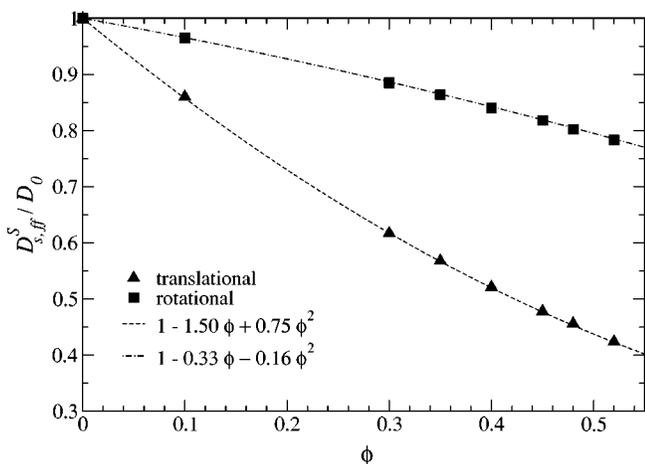
ume fractions. (The rotational diffusivities do not have a strong N-dependence[15] and need not be corrected for finite N.) Polynomial fits of the diffusivities are also shown. These might be useful for the simulation of other volume fractions without the need to compute the far-field diffusivities. We used these functions for our ASDB-nf runs. It is interesting to note that according to the fitted polynomia, the first-order (in ϕ) term of the far-field translational self-diffusion coefficient has a factor 1.5, which is to be compared with the exact value 1.83, first obtained by Batchelor.[17] Batchelor's result contains the near-field interactions and is therefore larger, implying a smaller diffusivity.

The scaling of this approximate method is dominated by the computation of the hydrodynamic velocity, which scales as $O(N \log N)$. In case of an equilibrium suspension, the approximate method is an $O(N)$ scheme.

III. RESULTS AND DISCUSSION

In this section we present results obtained with both the new ASD for Brownian suspensions (ASDB) and the approximate version ASDB-nf, illustrating the scaling of the methods and also comparing their results with those obtained by the conventional SD algorithm. The aim of this section is to verify the predicted scalings, to assess the accuracy of the ASDB-nf, as well as to verify the proper implementation of the ASDB scheme.

Figure 3 shows the CPU time needed for 100 time steps of a typical simulation of a system with a volume fraction $\phi = 0.45$ and Pe = 1, starting from an equilibrium configuration, as a function of the number of particles in the simulation box, N. (The Peclet number measures the relative importance of shear to thermal forces and is given by Pe $= 6\pi\eta_0 a^3 \dot{\gamma}/kT$, with the shear rate $\dot{\gamma}$ given by the magnitude of the velocity gradient tensor.) We have chosen 100 time steps to try to capture the average computational cost, since at the beginning the eigenvalues need to be computed and in many cases they are not computed again for hundreds

FIG. 2. Short-time self-diffusion coefficients in absence of near-field hydro-dynamic interactions vs volume fraction [values for the translational diffu-sivities have been corrected according to Eq. (46) from values for $N = 427-512$]. Dotted lines represent the quadratic fits of the data.
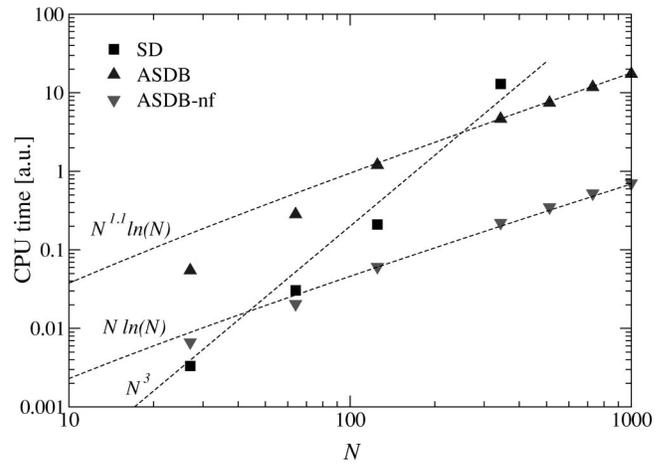
of time steps (depending on volume fraction and Pe). Remember also that the SD algorithm usually computes the inverse of $\mathcal{M}^\infty$ only every 100 time steps.

Two aspects are to be noted in Fig. 3. First, the observed scalings of the different algorithms with N are roughly those expected. For small systems, however, the ASDB scheme scales nearly as N^2, since the size of the simulation box is smaller than the cutoff distance for the real sum contribution to the far-field fluid velocity (see Sierou and Brady[8]). Second, the computational overhead for achieving the favorable $N^{1.25}\log N$ scaling is quite important, such that the break-even point is for a system of about 300 particles. The ASDB-nf approximate scheme, on the other hand, is about 20 times faster than ASDB for the system under consideration; the factor of 20 is the cost of computing the far-field random Brownian forces and resulting velocities.

For a suspension under steady simple shear flow the viscosity is related to the xy component [if (x,y) define the velocity–velocity-gradient plane] of the bulk stress $\mathbf{\Sigma}$ and rate of strain in the following manner:

$$\eta = \frac{\Sigma_{xy}}{2E_{xy}^\infty}, \tag{47}$$

where the bulk stress is defined as an average over the volume V containing the N particles and is given by

$$\mathbf{\Sigma} = -\langle p \rangle \mathbf{I} + 2\eta_0 \langle \mathbf{E}^\infty \rangle - nkT\mathbf{I} + n(\langle \mathbf{S}^h \rangle + \langle \overline{\mathbf{S}^b} \rangle + \langle \mathbf{S}^p \rangle). \tag{48}$$

Here, $-nkT\mathbf{I}$ is the isotropic stress associated with the thermal energy of the Brownian particles, $\mathbf{I}$ is the isotropic tensor, and n is the number density of the particles. The angular brackets denote an average over all particles and over time in a dynamic simulation.

The individual hydrodynamic ($\mathbf{S}^h$) and Brownian ($\overline{\mathbf{S}^b}$) contributions to the viscosity are denoted as η_h and η_b, respectively; hence, in absence of interparticle forces,

$$\eta = \eta_0 + \eta_h + \eta_b. \tag{49}$$

Results for the Brownian and hydrodynamic viscosities as a function of Pe for a typical suspension with $\phi=0.45$ are presented in Fig. 4. Here, the number of particles used in the simulation box was 64 in order to compare with conventional SD results. In this plot we include data for the three schemes SD, ASDB, and ASDB-nf. The agreement between data obtained using SD and ASDB confirms the proper implementation of the scheme. ASDB-nf values are also in near perfect quantitative agreement with the ASDB results. At the lowest Pe, there is an appreciable scatter of the data, but this can be attributed to the quite large error associated with these points. Similarly, for high Pe the SD ($N=27$) data seem to lie systematically above the ASDB and ASDB-nf data. This can be related to the fact that SD ($N=27$) results were obtained allowing some slight particle overlap, while the other methods do not allow any particle overlap, and furthermore, a minimum interparticle separation of $2.0002a$ was enforced.

For equilibrium suspensions it is possible to compute the Brownian contribution to the zero-shear limiting viscosity η_b employing the Green–Kubo formula[18]
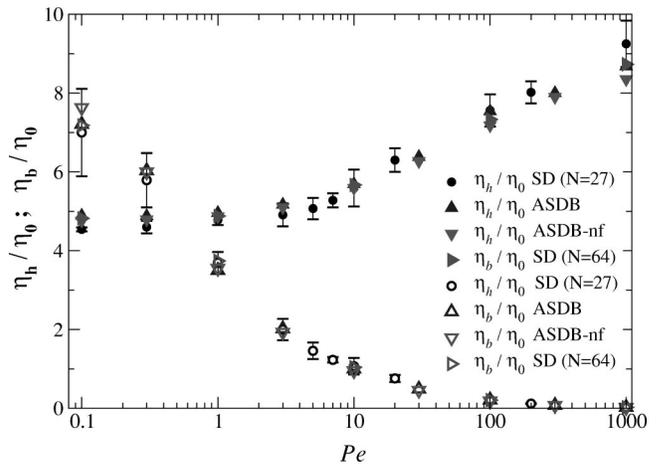

FIG. 4. Hydrodynamic and Brownian viscosity contributions for a Brownian suspension with $\phi=0.45$, vs Pe. Results are shown for systems with $N=64$, using SD, ASDB, and ASDB-nf, also included are SD data from Foss and Brady (Ref. 7) for systems with 27 particles.

$$\eta_b = \eta - \eta_\infty' = \frac{V}{kT}\int_0^\infty \langle \sigma_{xy}(t)\sigma_{xy}(0)\rangle dt. \tag{50}$$

Here, $\sigma_{xy}(t)$ represents the instantaneous Brownian shear stress, and $\langle \sigma_{xy}(t)\sigma_{xy}(0)\rangle$ is the Brownian shear–stress autocorrelation function.

In Fig. 5 we compare ASDB, ASDB-nf, and SD (taken from Foss and Brady[7]) results for the Brownian contribution to the zero-shear limiting viscosity as a function of the volume fraction. All sets of data are obtained using Eq. (50). As in the nonequilibrium case, we find good agreement between the approximate ASDB-nf and the more accurate SD (or ASDB) algorithm. For $\phi=0.2$ there is some discrepancy between the SD and the ASDB results, but we atribute this to the difficulty in computing η_b using Eq. (50), since very long runs and a large number of particles are needed to obtain an accurate stress-autocorrelation function. The exact low-density limit,[19,20] $\eta_b/\eta_0 = 0.913\phi^2 + O(\phi^3)$, is also included for comparison.
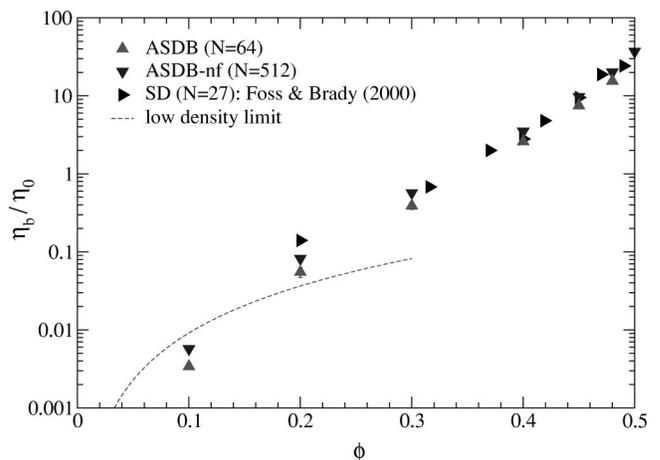

FIG. 5. Brownian viscosity contribution as a function of the volume fraction ϕ. Results obtained using ASDB-nf are compared with SD data from Foss and Brady (Ref. 7).

J. Chem. Phys., Vol. 118, No. 22, 8 June 2003

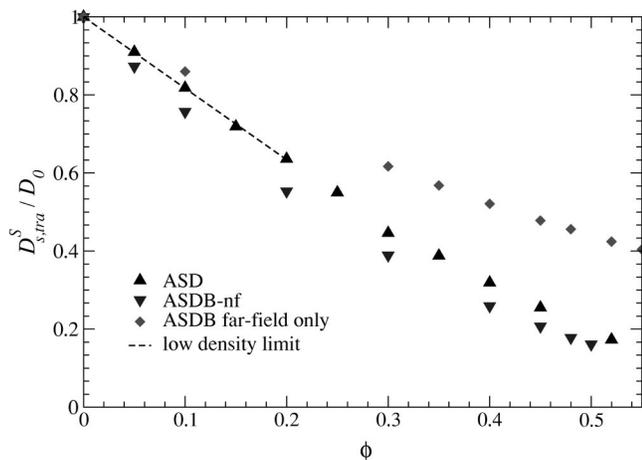Accelerated Stokesian dynamics: Brownian motion   10331

FIG. 6. Short-time translational self-diffusion coefficient for equilibrium Brownian suspensions as a function of volume fraction. ASD data taken from Ref. 8 are compared with ASDB-nf results (ASD values are corrected according to Ladd's proposed expression). For comparison, the extrapolated short-time self-diffusion coefficient in absence of near-field HI are included, as well as the rigorous low density limit (Ref. 17).

Figures 6 and 7 show results for the short-time translational and rotational self-diffusion coefficients, respectively, as a function of the volume fraction for equilibrium colloidal suspensions. In the case of ASDB and ASDB-nf, the self-diffusivities were obtained as

$$D_{s,ii}^S \equiv \langle \mathbf{R}_{FU,ii}^{-1} \rangle = \frac{1}{2\Delta t} \langle \mathbf{u}_i^B \mathbf{u}_i^B \rangle. \tag{51}$$

Here $\langle \dots \rangle$ denotes an average over all particles and an ensemble average (over configurations during the simulation), and $\mathbf{u}_i^B$ represents the i component of the particle Brownian velocity, so that depending on i we compute the translational or rotational diffusivity with respect to one axis. Since the suspension is isotropic we average over the three axes to obtain the plotted results. Note that computing the diffusivities according to Eq. (51) also constitutes a test for the statistics of the Brownian forces.
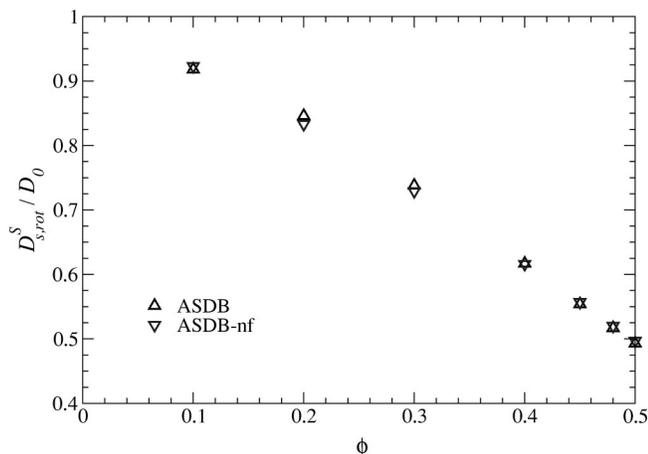

FIG. 7. Short-time rotational self-diffusion coefficient for equilibrium Brownian suspensions as a function of volume fraction. Results are from simulations for $N=64$.

As was mentioned,[15] the rotational diffusivities (Fig. 7) do not have an appreciable size dependence and we simply present results obtained for systems with $N=64$. Translational diffusivities obtained using ASDB-nf are also not sensitive to the size of the system since the far-field hydrodynamic interactions (responsible for the $N^{-1/3}$ dependence) are approximated in a mean-field-like manner. Data for ASDB-nf in Fig. 6 correspond to systems with $N=64$.

In Fig. 6 the rigorous dilute limit $D_{s,\text{tra}}^S/D_0 = 1 - 1.83\phi$ is also included for comparison. Note that the inclusion of the near-field interactions in the Stokesian dynamics algorithm (compared with the far-field only computations shown in Fig. 2) recovers the correct dilute limit.

IV. CONCLUSIONS

In this paper we have presented a new Stokesian dynamics algorithm for Brownian suspensions (ASDB) with a favorable scaling of roughly $O(N^{1.25}\log N)$. This new algorithm is seen to be accurate, reproducing results obtained with conventional SD. While accurate, ASDB is computationally still involved, and is only faster than conventional SD for systems of the order of 200–300 particles. For this reason, full advantage of this new scheme has only been possible for the study of static quantities where a set of configurations are given and static and short-time dynamic properties can be computed using ASDB without moving the particles. For dynamic simulations, which may require very long runs, it is still very demanding using the current state of the art PCs.

A very fast approximate algorithm (ASDB-nf) was also presented, whose scaling is of order $N\log N$ for nonequilibrium simulations and is $O(N)$ for equilibrium suspensions. This approximate algorithm was tested by comparing both static and dynamic quantities and showed, in general, a remarkable agreement with the ASDB (or SD) data. Using ASDB-nf, dynamic simulations of systems of the order of 500 particles are possible on a 1.5GHz PC. The study of larger systems, or long-time dynamics of Brownian suspensions, is therefore now accessible with ASDB-nf.

The scope of this article was limited to introducing the new algorithms, and we leave the application of them for the study of long-time dynamics and rheology of highly concentrated colloidal suspensions to future works.

As a final note, ASDB is constructed for infinite periodic systems and achieves its favorable scaling through the use of the FFT. If one wishes to study an isolated assembly of particles, such as a bead-spring model of a polymer in an unbounded fluid, then the method cannot be directly applied. However, to model an isolated system it might prove more advantageous to employ a very dilute periodic representation with ASDB, than to treat a truly isolated system for which the algorithm would scale roughly as $O(N^{2.25})$.[11] [Fast multipole methods might, in principle, allow one to reduce this to $O(N^{1.25})$.]

ACKNOWLEDGMENTS

APPENDIX A: ALTERNATIVE APPROACH FOR THE INVERSION OF $\mathcal{R}$

Solving Eq. (13) as described in Sec. II B is inadequate when the matrix $\mathbf{R}_{FU,\text{nf}}$ is ill-conditioned, which may occur for low volume fractions. In particular, in the limit of infinite dilution $\mathbf{R}_{FU,\text{nf}} \equiv \mathbf{0}$. To overcome this inconvenience, we present here an alternative approach that does not involve the inversion of $\mathbf{R}_{FU,\text{nf}}$, which is then more appropriate for low volume fraction systems (typically $\phi \leq 0.3$).

From Eqs. (16), (17), and (20), it follows that $\mathbf{F}_{\text{ff}}^h = \mathbf{F}_{\text{nf}}^h - \mathbf{R}_{FU,\text{nf}} \cdot \mathbf{U}^h$, which we replace in Eq. (17) to obtain

$$\mathcal{M}^\infty \cdot \begin{bmatrix} \mathbf{F}_{\text{nf}}^h - \mathbf{R}_{FU,\text{nf}} \cdot \mathbf{U}^h \\ \mathbf{S}_{\text{ff}}^h \end{bmatrix} = \begin{bmatrix} \mathbf{U}^h \\ -\mathbf{E}^\infty \end{bmatrix}. \tag{A1}$$

We now define $\widetilde{\mathbf{F}}$ and $\widetilde{\mathbf{S}}$ according to

$$\mathcal{M}^\infty \cdot \begin{bmatrix} \widetilde{\mathbf{F}} \\ \widetilde{\mathbf{S}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{E}^\infty \end{bmatrix}. \tag{A2}$$

Note that this can be solved without knowing the velocity, and, if desired, the values of $\widetilde{\mathbf{F}}$ and $\widetilde{\mathbf{S}}$ can be kept unchanged for many time steps since $\mathcal{M}^\infty$ varies slightly with small configuration changes (in conventional SD the inverse of $\mathcal{M}^\infty$ was kept constant for up to 100 time steps).

Subtracting Eq. (A2) from Eq. (A1) yields

$$\mathcal{M}^\infty \cdot \begin{bmatrix} \mathbf{F}_{\text{nf}}^h - \widetilde{\mathbf{F}} \\ \mathbf{0} \end{bmatrix} + \mathcal{M}^\infty \cdot \begin{bmatrix} -\mathbf{R}_{FU,\text{nf}} \cdot \mathbf{U}^h \\ \mathbf{S}_{\text{ff}}^h - \widetilde{\mathbf{S}} \end{bmatrix} = \begin{bmatrix} \mathbf{U}^h \\ \mathbf{0} \end{bmatrix}, \tag{A3}$$

and rearranging terms leads to the final equation for $\mathbf{U}^h$ and $(\mathbf{S}_{\text{ff}}^h - \widetilde{\mathbf{S}})$,

$$\widehat{\mathcal{M}^\infty} \cdot \begin{bmatrix} \mathbf{U}^h \\ \mathbf{S}_{\text{ff}}^h - \widetilde{\mathbf{S}} \end{bmatrix} = -\mathcal{M}^\infty \cdot \begin{bmatrix} \mathbf{F}_{\text{nf}}^h - \widetilde{\mathbf{F}} \\ \mathbf{0} \end{bmatrix}, \tag{A4}$$

where

$$\widehat{\mathcal{M}^\infty} = \mathcal{M}^\infty \cdot \begin{bmatrix} -\mathbf{R}_{FU,\text{nf}} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{A5}$$

For the Brownian velocity the approach is completely analogous, the only difference is that since the rate of strain does not appear there is no need to define (and compute) the analogues of $\widetilde{\mathbf{F}}$ and $\widetilde{\mathbf{S}}$.

The advantage of solving Eq. (A4) instead of Eq. (18) is that there are no inner iterations in this approach, since we

have replaced the inversion of $\mathbf{R}_{FU,\text{nf}}$ (which was done iteratively with a Cholesky preconditioned conjugate gradient algorithm) by just a dot product of $\mathbf{R}_{FU,\text{nf}}$ and a vector. This makes this approach more convenient for low to intermediate concentrations. For concentrated suspensions, however, the extra work of the inner solver for inverting $\mathbf{R}_{FU,\text{nf}}$ pays off since it reduces the number of (outer) iterations considerably.

APPENDIX B: CHEBYSHEV POLYNOMIAL APPROXIMATION

$C_j(\mathcal{M}^\infty)$ are the (matrix-) generalized shifted Chebyshev polynomials, which satisfy the following recursion relations:

$$C_{l+1}(\mathbf{M}') = 2\mathbf{M}' C_l(\mathbf{M}') - C_{l-1}(\mathbf{M}'),$$
$$C_0(\mathbf{M}') = \mathbf{1}, \quad C_1(\mathbf{M}') = \mathbf{M}', \tag{B1}$$

with

$$\mathbf{M}' = \frac{2}{\lambda_{\max} - \lambda_{\min}} \mathcal{M}^\infty - \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \mathbf{1}. \tag{B2}$$

Here, $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalue of $\mathcal{M}^\infty$, respectively.

Using these recursion relations, it is possible to compute the action of $C_j(\mathcal{M}^\infty)$ on an arbitrary vector $\mathbf{y}$, by j repetitions of computations of the form $\mathbf{M}' \cdot \mathbf{x}$, which are $O(N \log N)$.

[1] W. B. Russel, D. A. Saville, and W. R. Schowalter, *Colloidal Dispersions* (Cambridge University Press, Cambridge, 1989).
[2] J. K. G. Dhont, *An Introduction to Dynamics of Colloids* (Elsevier, Amsterdam, 1996).
[3] G. Bossis and J. F. Brady, J. Chem. Phys. **87**, 5437 (1987).
[4] G. Bossis and J. F. Brady, J. Chem. Phys. **91**, 1866 (1989).
[5] T. N. Phung, J. F. Brady, and G. Bossis, J. Fluid Mech. **313**, 181 (1996).
[6] D. R. Foss and J. F. Brady, J. Fluid Mech. **401**, 243 (1999).
[7] D. R. Foss and J. F. Brady, J. Fluid Mech. **407**, 167 (2000).
[8] A. Sierou and J. F. Brady, J. Fluid Mech. **448**, 115 (2001).
[9] R. C. Ball and J. R. Melrose, Physica A **247**, 444 (2001).
[10] M. Fixman, Macromolecules **19**, 1204 (1986).
[11] R. M. Jendrejack, M. D. Graham, and J. de Pablo, J. Chem. Phys. **113**, 2894 (2000).
[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, UK, 1992).
[13] R. B. Lehoucq, D. C. Sorensen, and C. Yang, ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods, 1997, ftp://ftp.caam.rice.edu/pub/software/ARPACK
[14] P. S. Grassia, E. J. Hinch, and L. C. Nitsche, J. Fluid Mech. **282**, 373 (1995).
[15] R. J. Phillips, J. F. Brady, and G. Bossis, Phys. Fluids **31**, 3462 (1988).
[16] A. J. C. Ladd, J. Chem. Phys. **93**, 3484 (1990).
[17] G. K. Batchelor, J. Fluid Mech. **74**, 1 (1976).
[18] G. Nägele and J. Bergenholtz, J. Chem. Phys. **108**, 9893 (1998).
[19] G. K. Batchelor and J. T. Green, J. Fluid Mech. **56**, 401 (1972).
[20] B. Cichocki and B. U. Felderhof, J. Chem. Phys. **101**, 7850 (1994).