

Dynamic simulation of concentrated macromolecular solutions with screened long-range hydrodynamic interactions: Algorithm and limitations

Tadashi Ando, Edmond Chow, and Jeffrey Skolnick

Citation: *J. Chem. Phys.* **139**, 121922 (2013); doi: 10.1063/1.4817660

View online: <http://dx.doi.org/10.1063/1.4817660>

View Table of Contents: <http://jcp.aip.org/resource/1/JCPSA6/v139/i12>

Published by the [AIP Publishing LLC](#).

Additional information on *J. Chem. Phys.*

Journal Homepage: <http://jcp.aip.org/>

Journal Information: http://jcp.aip.org/about/about_the_journal

Top downloads: http://jcp.aip.org/features/most_downloaded

Information for Authors: <http://jcp.aip.org/authors>

ADVERTISEMENT



Explore the **Most Cited**
Collection in Applied Physics

AIP
Publishing

Dynamic simulation of concentrated macromolecular solutions with screened long-range hydrodynamic interactions: Algorithm and limitations

Tadashi Ando,¹ Edmond Chow,² and Jeffrey Skolnick¹

¹Center for the Study of Systems Biology, School of Biology, Georgia Institute of Technology, 250 14th Street NW, Atlanta, Georgia 30318-5304, USA

²School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, Georgia 30332-0765, USA

(Received 3 April 2013; accepted 17 July 2013; published online 13 August 2013)

Hydrodynamic interactions exert a critical effect on the dynamics of macromolecules. As the concentration of macromolecules increases, by analogy to the behavior of semidilute polymer solutions or the flow in porous media, one might expect hydrodynamic screening to occur. Hydrodynamic screening would have implications both for the understanding of macromolecular dynamics as well as practical implications for the simulation of concentrated macromolecular solutions, e.g., in cells. Stokesian dynamics (SD) is one of the most accurate methods for simulating the motions of N particles suspended in a viscous fluid at low Reynolds number, in that it considers both far-field and near-field hydrodynamic interactions. This algorithm traditionally involves an $O(N^3)$ operation to compute Brownian forces at each time step, although asymptotically faster but more complex SD methods are now available. Motivated by the idea of hydrodynamic screening, the far-field part of the hydrodynamic matrix in SD may be approximated by a diagonal matrix, which is equivalent to assuming that long range hydrodynamic interactions are completely screened. This approximation allows sparse matrix methods to be used, which can reduce the apparent computational scaling to $O(N)$. Previously there were several simulation studies using this approximation for monodisperse suspensions. Here, we employ newly designed preconditioned iterative methods for both the computation of Brownian forces and the solution of linear systems, and consider the validity of this approximation in polydisperse suspensions. We evaluate the accuracy of the diagonal approximation method using an intracellular-like suspension. The diffusivities of particles obtained with this approximation are close to those with the original method. However, this approximation underestimates intermolecular correlated motions, which is a trade-off between accuracy and computing efficiency. The new method makes it possible to perform large-scale and long-time simulation with an approximate accounting of hydrodynamic interactions. © 2013 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4817660>]

I. INTRODUCTION

How can we describe the dynamical motions of molecules in a fluid? This simple question has been one of the fundamental topics in various fields including fluid mechanics, chemical engineering, and biology. In biology, since intracellular transport and diffusive mechanisms underlie almost all cellular processes, answering this question is essential to understanding the mechanisms of cellular dynamics. In this regard, molecular dynamics (MD) is an important tool for simulating molecular motions in a fluid.¹ MD provides detailed information on the dynamical behavior of solute as well as solvent molecules at an atomic level of description. However, even though we are interested in the long-time dynamical behavior of solute molecules, to account for the effect of the solvent molecules on the solute requires a large number of solvent molecules and short time steps to handle their fast motions. This makes the simulations very expensive. In such a case, an approximate approach may be adopted. Pioneering work on coarse-grained simulations as applied to biological macromolecules was carried out by Ermak and McCammon in 1978, who developed a Brownian dynamics (BD) algorithm that can incorporate hydrodynamics.² In BD, so-

lute molecules are treated as Brownian particles and solvent molecules are not treated explicitly; rather, their dynamical effects on the solute are incorporated in a stochastic manner consistent with hydrodynamics.² Their BD algorithm can include the simplest hydrodynamic interactions (HI) through a position-dependent inter-particle diffusion tensor such as the Oseen³ and Rotne-Prager-Yamakawa^{4,5} (RPY) tensors. However, these tensors only represent the far-field part of HI, and thus, they are only valid for dilute solutions. To overcome this limitation, Stokesian dynamics (SD) was developed by Brady's group in 1987.⁶ In SD, both many-body far-field HI and the near-field HI (so-called "lubrication forces") are modeled. SD can reproduce the properties of monodisperse suspensions at high volume fractions.⁷ SD has been successfully applied to the rheology of the Brownian particles under flow.^{8,9} Recently, we applied the SD method to explore the nature of macromolecular dynamics in crowded intracellular environments comprised of a polydisperse collection of macromolecules.¹⁰ Our simulations with coarse-grained models of macromolecules showed that excluded volume effects and HI are both required to account for the large reduction in diffusion coefficient from that at infinite dilution that is

observed in experiments.¹⁰ We also observed collective intermolecular motions in such dense systems, which were not seen in simulations without HI.

While SD is accurate, it is computationally demanding, which limits the size of systems that can be simulated in practice to about one thousand particles. Overcoming this limitation is especially important for biological applications; the complexity of these systems is much higher than simple colloidal suspensions and long-time simulations are typically required for analyzing relevant biological motions. In SD, each time step involves solves with a hydrodynamic matrix which is a dense matrix due to the long-range nature of HI, decaying with particle distance r as $1/r$.^{6,11} To take account of Brownian motions, the square root of the matrix is required to calculate the random forces. Both calculations typically require $O(N^3)$ operations for an N particle system. To tackle the computational complexity, several ideas have been proposed. To update each time step, iterative solution methods have been suggested to solve the linear systems of equations.^{12–14} To calculate the random forces, a Chebyshev polynomial approximation was introduced by Fixman.¹⁵ We recently proposed using Krylov subspace methods to compute the Brownian displacement for BD, which scales very nearly as $O(N^2)$.¹⁶ These methods are related to the Chebyshev method, but do not require eigenvalue estimates. These iterative and polynomial methods involve $O(N^2)$ matrix-vector multiplications for dense matrix cases. For efficient multiplication with the dense hydrodynamic matrix, particle-mesh Ewald (PME)^{12,13} and smooth PME¹⁷ methods can be employed for periodic boundary conditions, which scale as $N \log(N)$. Brady's group implemented a version of PME into SD for non-Brownian suspensions called accelerated SD (ASD)¹³ and for Brownian suspensions called ASDB¹² that scales $N^{1.25} \log N$. Fast-multipole methods might compute the multiplication with $O(N)$ computational cost for free-boundary conditions. Recently developed methods, such as general geometry Ewald-like method (GGEM)¹⁸ and immersed boundary GGEM,¹⁹ are able to calculate long-range hydrodynamic effects with $O(N \log N)$ or $O(N)$ computational cost for complex geometries in a confined space.

While hydrodynamic interactions are long range and decay like $1/r$, these bare interactions may be screened by many-body effects in colloidal suspensions, resulting in effective pair interactions that are no longer long range. There are several studies on hydrodynamic screening. The theory of the dynamics of semidilute polymer solutions, originally due to DeGennes, established that as polymer solutions become more concentrated, hydrodynamic interactions are screened beyond a polymer concentration dependent screening length, and then the dynamics becomes Rouse like, viz., the free draining approximation holds and long range hydrodynamic interactions can be ignored.²⁰ The mechanism of this screening has been studied by computer simulations.²¹ In a porous medium, long-range hydrodynamic interactions are screened and decay faster than $1/r$, and appear to decay as $1/r^3$ as described by the Brinkman equation,²² which is also seen by simulation.²³ Snook, van Meegen, and Tough introduced a simple volume fraction-dependent screened two-body tensor, which provides reasonably accurate descriptions of various

properties of diffusive particles compared with experimental results.^{24–28} However, theoretical analysis suggested that hydrodynamic screening does not occur for freely moving diffusive particles.²⁹ Here, we examine the validity of the screening of far-field hydrodynamic interactions in the context of dense, polydisperse solutions of macromolecules. If this approximation were shown to be valid, this would not only provide insight into the forces dominating macromolecular motion, but would have significant practical consequences as it would significantly reduce the computational burden.

For periodic boundary conditions, since the long-range nature of HI is similar to electrostatic interactions, Ewald summation of the Oseen or RPY tensors to obtain the far-field hydrodynamic matrix is necessary not only for accuracy but also to ensure that the matrix is positive definite.¹¹ Otherwise, the transport properties of the system are non-physical, e.g., a negative diffusion coefficient results. If hydrodynamic interactions are strongly screened, the far-field hydrodynamic matrix can be approximated by a diagonal matrix, corresponding to the neglect of intermolecular, long range hydrodynamic interactions. The idea of using a diagonal approximation for the far-field hydrodynamic matrix is not new. This approximation was already used to develop fast SD algorithms for non-Brownian¹⁴ and Brownian particles.^{30–33} Hereafter, using the name coined by Bybee,³² we call these methods Fast Lubrication Dynamics (FLD). Combining the diagonal approximation with a sparse iterative solver for linear systems, such as the conjugate gradient (CG) method, as well as the Chebyshev method for computing Brownian forces, FLD methods can reduce the computational cost to $O(N)$. Banchio and Brady used the approximation only for computing Brownian quantities in the ASD framework called ASDB-nf, which scales as $O(N \log N)$ for non-equilibrium suspensions and as $O(N)$ for equilibrium suspensions.¹² In the previous studies using FLD, monodisperse suspensions were used for algorithm development. The purpose of this work is to design an alternative $O(N)$ computational method for FLD and to see limitations of the diagonal approximation idea. We evaluate this approximation in a polydisperse suspension, which is essential for coarse-grained simulations of biological systems. In our FLD method, a Krylov subspace method is used for computing Brownian forces, where we use preconditioning to accelerate the computations, a possibility that is not well known in this context. Our FLD algorithms are described in Sec. II. The computational efficiency of the algorithms and simulation results for a model of the intracellular environment are shown in Sec. III. Our conclusions are presented in Sec. IV.

II. THEORY AND METHOD

In this paper, we compared several algorithms for simulating Brownian particles. For clarity, we summarize names of the algorithms used here. SD denotes the Stokesian dynamics method that considers both many-body far-field HI and near-field lubrication forces at low Reynolds number that was introduced by Brady *et al.*¹¹ The original SD algorithm denotes a simulation algorithm involving an $O(N^3)$ matrix inversion and Cholesky factorization of dense matrices. BD represents the simulation algorithm considering HI at the level of

Oseen or RPY tensors developed by Ermak and McCammon.² Free-draining simulations (FD) are BD simulations without HI, where the diffusion matrix only has diagonal elements and these are constant. FLD is the method where the far-field hydrodynamic matrix is approximated by a diagonal matrix and the lubrication forces are explicitly considered.

We first describe the original SD algorithm and then introduce our FLD approximations. Throughout the paper, we only consider particle translation and forces, and ignore particle rotations and torques. This corresponds to the so-called ‘‘F version’’ of the SD paper.⁷

A. Stokesian dynamics for Brownian particles

For a system of N Brownian particles suspended in an incompressible Newtonian fluid, the translational motion of particles is represented by the coupled N -body Langevin equation:^{34,35}

$$\mathbf{m} \frac{d\mathbf{u}}{dt} = \mathbf{f}^H + \mathbf{f}^P + \mathbf{f}^B, \quad (1)$$

where \mathbf{m} is the $3N \times 3N$ mass tensor, \mathbf{u} is the particle translational velocity vector of length $3N$, and on the right-hand side are the forces acting on the particles, which can be split into three parts: the hydrodynamic forces \mathbf{f}^H , the deterministic non-hydrodynamic forces \mathbf{f}^P , and the stochastic Brownian forces \mathbf{f}^B ; all are vectors of length $3N$. When the particle Reynolds number is small, the hydrodynamic forces acting on the particles in the absence of an external bulk flow are given by

$$\mathbf{f}^H = -\mathbf{R}\mathbf{u}. \quad (2)$$

Here \mathbf{R} is the configuration dependent hydrodynamic resistance tensor, with size $3N \times 3N$. The deterministic non-hydrodynamic forces are arbitrary and can include electrostatic and van der Waals forces. The Brownian forces \mathbf{f}^B arise from thermal fluctuations in the fluid and are characterized by the fluctuation-dissipation theorem³⁶

$$\langle \mathbf{f}^B \rangle = 0, \quad \langle \mathbf{f}^B(0)\mathbf{f}^B(t) \rangle = 2k_B T \mathbf{R} \delta(t), \quad (3)$$

where the angle brackets denote an ensemble average over independent configurations, k_B is Boltzmann’s constant, T is the absolute temperature, t is time, and $\delta(t)$ is the Dirac delta function.

The time evolution equation of the particles is obtained by following Ermak and McCammon and integrating Eq. (1) over a time step Δt that is larger than the inertial relaxation time, ($\tau_B = m/6\pi\eta a$, for a particle of radius a and mass m , and solvent viscosity η), but smaller than the time over which the forces change significantly.² This leads to

$$\mathbf{r} = \mathbf{r}^0 + k_B T (\nabla \cdot (\mathbf{R}^0)^{-1}) \Delta t + (\mathbf{R}^0)^{-1} \mathbf{f}^{P,0} \Delta t + \mathbf{x}^0(\Delta t) \quad (4)$$

with

$$\langle \mathbf{x}^0(\Delta t) \rangle = 0, \quad \langle \mathbf{x}^0(\Delta t) \mathbf{x}^0(\Delta t) \rangle = 2k_B T (\mathbf{R}^0)^{-1} \Delta t. \quad (5)$$

Here, \mathbf{r} is the particle position vector of $3N$ length, $\mathbf{x}(\Delta t)$ is a random displacement due to Brownian noise, and the superscript ‘‘0’’ indicates that each quantity is computed at the beginning of the time interval. The inverse of the resistance

matrix is the N -particle mobility matrix \mathbf{M} , which, when multiplied by $k_B T$, is the N -particle diffusion tensor

$$k_B T \mathbf{R}^{-1} = k_B T \mathbf{M} = \mathbf{D}. \quad (6)$$

Using this relation, Eq. (4) is identical to the Ermak and McCammon algorithm for Brownian dynamics with HI.² These matrices are symmetric and should be positive definite. The simplest way to compute $\mathbf{x}(\Delta t)$ is through the use of a Cholesky factorization, $\mathbf{C}\mathbf{C}^T = \mathbf{R}^{-1}$, which is an $O(N^3)$ operation.

1. Hydrodynamic interactions

Hydrodynamic interactions between particles are described by the matrix \mathbf{R} (or \mathbf{M} or \mathbf{D}). The RPY diffusion or mobility tensor is commonly used to model the HI in dynamic simulations because this tensor retains positive definiteness even when particles overlap.^{2,37,38} The RPY tensor contains the two-body and long-range or far-field contributions to the particle mobility up to order $1/r^3$. For dilute systems, two-body interactions may be sufficient, but in concentrated systems, e.g., the inside of cells, far-field many-body HI as well as near-field lubrication forces play important roles in determining mobility. Durlofsky, Brady, and Bossis developed a method to include not only the far-field HI but also the many-body and near-field HI in simulations, which is the key idea of SD.^{6,11,35} The resistance tensor \mathbf{R} could be described by the sum of far-field HI and near-field lubrication interactions as

$$\mathbf{R} = (\mathbf{M}^\infty)^{-1} + \mathbf{R}_{\text{lub}} \quad (7)$$

with

$$\mathbf{R}_{\text{lub}} = \mathbf{R}_{2B} - \mathbf{R}_{2B}^\infty. \quad (8)$$

Here, $(\mathbf{M}^\infty)^{-1}$ represents the contribution of many-body, far-field interactions, which is the inverse of the RPY mobility matrix. \mathbf{R}_{lub} contains the near-field lubrication forces, which is formed in a pair wise additive fashion of two-body HI. \mathbf{R}_{2B} , which represents the exact two-body HI, includes both near-field and far-field interactions. \mathbf{R}_{2B}^∞ is the resistance tensor that represents two-body, far-field interactions. The far-field part has already been included in $(\mathbf{M}^\infty)^{-1}$. Thus, in order not to count these interactions twice, we must subtract these two-body interactions. This is the standard method to correct for the lubrication effects in the resistance tensor.

\mathbf{M}^∞ can be estimated by the Ewald summation of the Oseen or RPY tensor developed by Hasimoto³⁹ and Beenakker,⁴⁰ respectively, for the case of periodic boundary conditions. Due to the long-range nature of HI, use of the Ewald summation technique is necessary not only for accuracy but also for obtaining positive definite matrices in the calculation of the mobility tensor under periodic boundary conditions. In addition, inverting \mathbf{M}^∞ corresponds to including many-body interactions.⁶ \mathbf{R}_{2B} is calculated by the exact two-body solution of Jeffrey and Onishi.⁴¹ \mathbf{R}_{2B}^∞ is obtained by simply inverting a two-body mobility matrix containing terms to the same order as $1/r$ in \mathbf{M}^∞ . The far-field \mathbf{M}^∞ is a dense matrix, and its explicit construction and inversion require $O(N^2)$ and $O(N^3)$ computations, respectively. Due to the

near-field interactions of lubrication interactions, on the other hand, \mathbf{R}_{lub} is sparse and its construction is an $O(N)$ computational task using a cut-off method.

2. Mid-point algorithm

The positions of Brownian particles can be computed through the evolution equations (4) and (5). For BD simulations with the RPY and Oseen tensors, since the matrices have the property of $\nabla \cdot \mathbf{M} = \nabla \cdot \mathbf{R}^{-1} = 0$, we can neglect the second term on the right-hand side of Eq. (4).² In contrast, for SD simulations, the divergence term must be taken into account, whose explicit computation is $O(N^3)$. To avoid this expensive calculation, one can use instead the ‘‘mid-point algorithm’’ introduced by Banchio and Brady,¹² following an idea of Fixman.⁴² In this method, Eqs. (4) and (5) are re-written as

$$\mathbf{r} = \mathbf{r}^0 + k_B T (\nabla \cdot (\mathbf{R}^0)^{-1}) \Delta t + (\mathbf{R}^0)^{-1} (\mathbf{f}^{\text{P},0} + \mathbf{f}^{\text{B},0}) \Delta t \quad (9)$$

with

$$\langle \mathbf{f}^{\text{B}} \rangle = 0, \quad \langle \mathbf{f}^{\text{B}}(0) \mathbf{f}^{\text{B}}(t) \rangle = 2k_B T \mathbf{R} \delta(t). \quad (10)$$

The procedure for this mid-point algorithm is the following steps:

- (i) Compute the resistance matrix \mathbf{R}^0 using an initial configuration \mathbf{r}^0

$$\mathbf{R}^0 = (\mathbf{M}^{\infty,0})^{-1} + \mathbf{R}_{\text{lub}}^0. \quad (11)$$

- (ii) Compute the deterministic forces $\mathbf{f}^{\text{P},0}$ and Brownian forces $\mathbf{f}^{\text{B},0}$. The latter is given by

$$\mathbf{f}^{\text{B},0} = \sqrt{\frac{2k_B T}{\Delta t}} \mathbf{B} \mathbf{z}, \quad (12)$$

where \mathbf{z} is a standard normal vector and \mathbf{B} is

$$\mathbf{R}^0 = \mathbf{B} \mathbf{B}^T. \quad (13)$$

The most conventional way to compute \mathbf{B} is use of the Cholesky factorization.

- (iii) Compute the velocity \mathbf{u}^0 of $3N$ length at \mathbf{r}^0 by solving the linear equation

$$\mathbf{R}^0 \mathbf{u}^0 = \mathbf{f}^{\text{P},0} + \mathbf{f}^{\text{B},0}. \quad (14)$$

- (iv) Move the particles to intermediate positions \mathbf{r}' by a small fraction of a time step, $\Delta t/n$

$$\mathbf{r}' = \mathbf{r}^0 + \frac{\Delta t}{n} \mathbf{u}^0, \quad (15)$$

where n is typically on the order of 100.

- (v) Calculate a resistance matrix, \mathbf{R}' , at the intermediate positions \mathbf{r}' .
(vi) Calculate a new velocity \mathbf{u}' at the intermediate positions using the forces evaluated at \mathbf{r}^0

$$\mathbf{R}' \mathbf{u}' = \mathbf{f}^{\text{P},0} + \mathbf{f}^{\text{B},0}. \quad (16)$$

- (vii) Calculate the drift velocity, $\mathbf{u}^{\text{drift}}$,

$$\mathbf{u}^{\text{drift}} = \frac{n}{2} (\mathbf{u}' - \mathbf{u}^0). \quad (17)$$

- (viii) Finally, update the positions of the particles for time step Δt ,

$$\mathbf{r} = \mathbf{r}^0 + (\mathbf{u}^0 + \mathbf{u}^{\text{drift}}) \Delta t. \quad (18)$$

3. Computational cost in original SD

In the mid-point algorithm just presented, the computations that scale as $O(N^2)$ or worse are the following:

- (i) Construction of \mathbf{M}^∞ at steps (I) and (V): $O(N^2)$;
- (ii) Inversion of matrix \mathbf{M}^∞ at step (I) and (V): $O(N^3)$;
- (iii) Cholesky factorization of \mathbf{R}^0 at step (II): $O(N^3)$;
- (iv) Solving linear equations at steps (III) and (VI): $O(N^3)$.

However, there are several ways to avoid these computations without any advanced techniques. First, since \mathbf{M}^∞ is less sensitive to small changes of particle positions compared to \mathbf{R}_{lub} , the inverse of the matrix calculated at step (I) can be reused at step (V) without significant accuracy loss. Furthermore, infrequent construction of the mobility tensor and its inversion are possible.³⁵ Second, the linear equation at step (III) can be solved by $O(N^2)$ operations by using the Cholesky factor \mathbf{B} obtained in the previous step. Third, the second linear equation at step (VI) could be simply solved by iterative methods, such as the CG method.^{43,44} For this case, since \mathbf{R}' is not significantly changed from \mathbf{R}^0 , the Cholesky factor \mathbf{B} obtained at step (II), would work as a preconditioner for solving the equation by the iterative method.^{43,44} These three ideas significantly reduce computational cost of the original SD method. However, an $O(N^3)$ Cholesky factorization is still required at each time step, which prohibits large scale SD simulations. Construction of \mathbf{M}^∞ and its inversion are also a problem for large systems.

B. Approximations

We introduce three approximations into the original SD algorithm: (1) the diagonal approximation of far-field HI, (2) a preconditioned Krylov subspace approximation for computing Brownian forces, and (3) a preconditioned conjugate gradient method for solving linear equations. We explain each approximation in the following sections.

1. The diagonal approximation of far-field HI

In dense systems, such as inside of cells, lubrication interactions between particles are more important than the precise details of the far-field HI. For such cases, mean-field-like approximations for the far-field interactions have been proposed, in which \mathbf{M}^∞ is replaced by a diagonal matrix^{12,14,30–33}

$$\mathbf{M}^\infty \approx \frac{1}{6\pi\eta_F a} \mathbf{I}. \quad (19)$$

Here, η_F is a parameter representing the ‘‘effective far-field viscosity’’ and \mathbf{I} is the identity matrix. This is equivalent to ignoring long range HI, except in a mean field sense. We now avoid the computationally intensive matrix inversions

and \mathbf{M}^∞ construction in the original SD algorithm. The resistance matrix \mathbf{R} is now sparse. This is particularly advantageous since the iterative methods introduced in the next sections involve matrix-vector multiplies with the matrix \mathbf{R} .

For monodisperse suspensions, η_F may be chosen by the Einstein viscosity $\eta_F = \eta(1 + 5\phi/2)$ with volume fraction ϕ .¹⁴ In our case, since the RPY tensor in \mathbf{M}^∞ contains terms up to $O(1/r^3)$ and \mathbf{R}_{lub} contains the rest of the higher order terms, reduction of the effective viscosity is completely included in the lubrication correction term. Therefore, we simply use

$$\mathbf{M}_{ii}^\infty \approx \frac{1}{6\pi\eta a_i} \mathbf{I}, \quad (20)$$

where \mathbf{M}_{ii}^∞ is the 3×3 diagonal block for the i th particle in \mathbf{M}^∞ and a_i is the radius of particle i .

2. Preconditioned Krylov subspace method for computing Brownian forces

The Brownian force in SD is computed as the square root of the resistance matrix times an uncorrelated random vector with mean zero. Hence, we may use Krylov subspace methods proposed for computing the function of a matrix times a vector, see for example Refs 45 and 46, and the references therein. We recently introduced and studied the use of Krylov subspace methods for computing Brownian displacements in the BD algorithm at the level of the RPY diffusion tensor.¹⁶ Here, the method approximates the product of the principal square root of a given symmetric positive definite matrix and a random vector via a Lanczos process, that is,

$$\mathbf{y} = \mathbf{M}^{1/2} \mathbf{z} \approx \|\mathbf{z}\|_2 \mathbf{V}_m \mathbf{H}_m^{1/2} \mathbf{e}_1, \quad (21)$$

where \mathbf{M} is the $3N \times 3N$ mobility matrix, \mathbf{z} is a Gaussian random noise vector, and $\|\mathbf{z}\|_2$ is the 2-norm of vector \mathbf{z} . \mathbf{V}_m is a $3N \times m$ orthonormal basis for the Krylov subspace, \mathbf{H}_m is the $m \times m$ symmetric tridiagonal matrix equaling $\mathbf{V}_m^T \mathbf{M} \mathbf{V}_m$ computed after m Lanczos steps, and \mathbf{e}_1 is the first column of the $m \times m$ identity matrix. The approximation is based on computing the matrix square root on a much smaller subspace, of dimension $m \ll 3N$, where it is inexpensive to compute exactly, and then mapping the result to the original space. A more accurate approximation is attained for larger dimensions m of the Krylov subspace. See our previous paper¹⁶ for details.

The convergence of the above approximation with m can be improved through preconditioning. This possibility has been suggested in the past,^{47–49} but has not been applied to Brownian simulations. The approach is analogous to preconditioning for accelerating the convergence of iterative methods for solving systems of linear equations. To explain this method for SD, we first note that Brownian forces can be expressed as

$$\mathbf{F}^{\text{B},0} = \sqrt{\frac{2k_B T}{\Delta t}} \mathbf{R}^{1/2} \mathbf{z} = \sqrt{\frac{2k_B T}{\Delta t}} \mathbf{y}. \quad (22)$$

In this case, \mathbf{M} in Eq. (21) is the resistance tensor \mathbf{R} . To emphasize the generality of the method, the notation \mathbf{A} is used in this section. Now, consider the approximation $\mathbf{L}\mathbf{L}^T \approx \mathbf{A}$. It is required that one can readily solve with \mathbf{L} , for example, \mathbf{L}

may be a sparse triangular matrix. Let us denote

$$\mathbf{L}^{-1} \mathbf{A} \mathbf{L}^{-T} = \mathbf{X}. \quad (23)$$

Since \mathbf{X} is a symmetric positive definite matrix, \mathbf{X} has a unique principal square root. Then \mathbf{A} can be decomposed by

$$\begin{aligned} \mathbf{A} &= \mathbf{L} \mathbf{X} \mathbf{L}^T \\ &= \mathbf{L} \mathbf{X}^{1/2} \mathbf{X}^{1/2} \mathbf{L}^T \\ &= (\mathbf{L} \mathbf{X}^{1/2}) (\mathbf{L} \mathbf{X}^{1/2})^T \\ &= \mathbf{S} \mathbf{S}^T. \end{aligned} \quad (24)$$

The matrix \mathbf{S} is not symmetric or triangular matrix in general, but can be used like \mathbf{B} for calculating Brownian forces described in Eq. (13). Then, \mathbf{y} is given by

$$\begin{aligned} \mathbf{y} &= \mathbf{S} \mathbf{z} \\ &= \mathbf{L} \mathbf{X}^{1/2} \mathbf{z}. \end{aligned} \quad (25)$$

To compute the approximation $\tilde{\mathbf{y}} = \mathbf{S} \mathbf{z}$, we first compute $\tilde{\mathbf{y}}' = \mathbf{X}^{1/2} \mathbf{z}$ via Eq. (21). This involves the matrix $\mathbf{L}^{-1} \mathbf{A} \mathbf{L}^{-T}$, which is designed to be well-conditioned, and thus the Krylov subspace approximation converges quickly. Finally, $\tilde{\mathbf{y}} = \mathbf{L} \tilde{\mathbf{y}}'$ is computed. Note that matrices \mathbf{S} and \mathbf{X} are never formed in the method. The factorization $\mathbf{L}\mathbf{L}^T \approx \mathbf{A}$ may be considered a ‘‘preconditioner.’’

The overall algorithm for the approximate correlated vector \mathbf{y} is shown in the Appendix. The most expensive operations are the matrix-vector multiplication and solving linear equations with the triangular matrix \mathbf{L} , which are both $O(N^2)$ operations for the dense matrix case. If the matrix \mathbf{A} contains $O(N)$ nonzero elements, which is the case in our FLD algorithm, then the time for one iteration scales linearly with N .

The Krylov subspace method iteratively improves the accuracy of the approximate correlated vector. We use the following quantity based on two consecutive iterates for monitoring the convergence of the method:¹⁶

$$E_k = \frac{\|\tilde{\mathbf{y}}_k - \tilde{\mathbf{y}}_{k-1}\|_2}{\|\tilde{\mathbf{y}}_{k-1}\|_2}, \quad (26)$$

where k is the iteration number. We also checked the relative norm of the exact error of the k th approximation as

$$E_k^{\text{exact}} = \frac{\|\tilde{\mathbf{y}}_k - \mathbf{L} \mathbf{X}^{1/2} \mathbf{z}\|_2}{\|\mathbf{L} \mathbf{X}^{1/2} \mathbf{z}\|_2}, \quad (27)$$

where \mathbf{z} is the same standard normal vector used to compute $\tilde{\mathbf{y}}$. This of course cannot be computed in practice.

3. Solving linear equations with a preconditioned iterative method

For solving linear systems $\mathbf{R} \mathbf{u} = \mathbf{f}$ at steps (III) and (VI) of the mid-point algorithm, the preconditioned CG method can be used as mentioned above. The preconditioner \mathbf{L} used for the Krylov subspace method at step (II) can be reused for solving these linear systems. For the second linear equation at step (VI), the solution \mathbf{u}^0 obtained at step (III) could be used as an initial guess. On the other hand, calculating an initial

guess for the first linear equation is nontrivial, since the Brownian forces on the right-hand side do not have correlations at different time steps. However, by using the fact that the resistance matrix on the left-hand side is slightly perturbed between consecutive time steps, initial guesses for the first linear system can be efficiently constructed.⁵⁰ When a basis for the Krylov subspace for the Brownian forces is available, an alternative way of constructing initial guesses for \mathbf{u}^0 is possible with almost no additional computational cost. To explain this method, first note that in the absence of non-hydrodynamic forces, that is $\mathbf{f}^p = 0$, the first linear equation is written as

$$\mathbf{R}^0 \mathbf{u}^0 = \mathbf{f}^B. \quad (28)$$

Given the preconditioner \mathbf{L} , the above equation is equivalent to

$$(\mathbf{L}^{-1} \mathbf{R}^0 \mathbf{L}^{-T})(\mathbf{L}^T \mathbf{u}^0) = \mathbf{L}^{-1} \mathbf{f}^B. \quad (29)$$

Using $\mathbf{X} = \mathbf{L}^{-1} \mathbf{R}^0 \mathbf{L}^{-T}$ and $\mathbf{f}^B = \alpha \mathbf{L} \mathbf{X}^{1/2} \mathbf{z}$ with $\alpha = \sqrt{2k_B T / \Delta t}$, then the solution \mathbf{u}^0 is written as

$$\begin{aligned} \mathbf{X}(\mathbf{L}^T \mathbf{u}^0) &= \mathbf{L}^{-1}(\alpha \mathbf{L} \mathbf{X}^{1/2} \mathbf{z}) \\ \mathbf{X}(\mathbf{L}^T \mathbf{u}^0) &= \alpha \mathbf{X}^{1/2} \mathbf{z} \\ \mathbf{L}^T \mathbf{u}^0 &= \alpha \mathbf{X}^{-1/2} \mathbf{z} \\ \mathbf{u}^0 &= \alpha \mathbf{L}^{-T} \mathbf{X}^{-1/2} \mathbf{z}. \end{aligned} \quad (30)$$

Then, the approximate \mathbf{u}^0 is calculated by

$$\mathbf{u}^0 \approx \tilde{\mathbf{u}}^0 = \alpha \|\mathbf{z}\|_2 \mathbf{L}^{-T} \mathbf{V}_m \mathbf{H}_m^{-1/2} \mathbf{e}_1. \quad (31)$$

The additional computational cost to compute $\tilde{\mathbf{u}}^0$ is negligible, since \mathbf{V}_m and \mathbf{H}_m are already calculated in the Lanczos process for computing the Brownian forces. This initial guess may reduce the number of iterations required for solving the first linear equation of each time step. We will see the results of this in Sec. III.

C. Simulation models

To validate our approximations, we used two models: (1) an intracellular-like polydisperse suspension and (2) monodisperse particle suspensions. The former is the same model that we used in our previous study of a model of a cell cytosol.¹⁰ The latter was used to evaluate the scaling of our algorithm. The intracellular-like suspension model has a total of 1152 particles, comprised of 15 different particle radii ranging from 21 Å to 115 Å, placed in a (100 nm)³ box giving a volume fraction ϕ of 0.51. Each particle radius corresponds to Stokes radius of a representative protein or tRNA molecule estimated by a rigid-particle theory. See our previous paper¹⁰ for details. The time step was set to 20 ps, which is obtained from $\sim 0.0005 (a^{\min})^2 / D_0^{\min}$, where a^{\min} and D_0^{\min} are the smallest radius in the system and its self-diffusion coefficient at infinite dilution ($D_0 = k_B T / 6\pi\eta a$), respectively. For the monodisperse suspension model, the number of particles N ranges from 200 to 4000 at a volume fraction $\phi = 0.5$. For this model, a time step of $0.0005 a^2 / D_0$ was used. All simulations were done at 298 K under periodic boundary conditions. In the mid-point algorithm, n was set to 100. In

calculating the lubrication forces, a cut-off distance in units of $s = 2r_{ij}/(a_i + a_j) = 4$ was used. For each condition, 10 independent simulations were done with different initial configurations. For the intracellular-like suspension model, simulations were done for 10^8 steps and the first 2.5×10^5 steps were discarded from the analysis. Deterministic non-hydrodynamic forces were not included; that is, $\mathbf{f}^p = 0$. It is worth mentioning that particles do not overlap in the presence of lubrication forces. Simulations using the original SD algorithm were also performed for comparison. For these cases, the far-field HI \mathbf{M}^∞ was estimated by Ewald summation of the RPY tensor.⁴⁰ \mathbf{M}^∞ was updated every 200 steps. All timing tests were performed on an AMD Opteron processor Model 2387. The Intel Math Kernel Library was used for computing matrix factorizations, matrix-matrix multiplications, and matrix-vector multiplications.

The short-time and long-time self-diffusion coefficients for translational motion of a particle (D_S and D_L , respectively) were estimated as in Ref. 35. It is known that the translational diffusion coefficients estimated by the Ewald summation technique under periodic boundary conditions show a $N^{-1/3}$ dependence.⁷ To correct for this finite-size effect, the formula proposed by Ladd is used for homogeneous monodisperse suspensions.⁵¹ In principle, it may be possible to correct these values even for the heterogeneous systems in this study using the same formula. To check this dependency, we calculated D_S for systems having varying numbers of particles, with $N > 9000$. However, a clear $N^{-1/3}$ dependence was not observed. Additionally, since the number of particles in the systems considered here is more than 1000, finite-size effects might be small. Therefore, we will show uncorrected data.

Finally, to analyze the correlations between particles in time and space, we calculated the normalized pair correlation function, C_{ij} , described in Ref. 10. C_{ij} ranges from -1.0 to 1.0 . When two particles are positively correlated, C_{ij} assumes a positive value and when negatively correlated, C_{ij} is negative.

III. RESULTS AND DISCUSSION

In this section, we first show the convergence of the two iterative approximation methods, the preconditioned Krylov subspace method for computing Brownian noise vectors, and preconditioned CG methods for solving linear systems involving sparse resistance matrices. Then, we present simulation results obtained by the new SD method and discuss the limitations of the approximations.

A. Convergence in the iterative methods

1. Convergence rate and error estimates for computing Brownian noise

In Fig. 1, the effect of preconditioning on the convergence of the preconditioned Krylov subspace method measured by E_k^{exact} is shown for the intracellular-like suspension. Here, we first employ the simplest preconditioner that just consists of the diagonal of the matrix, which works if

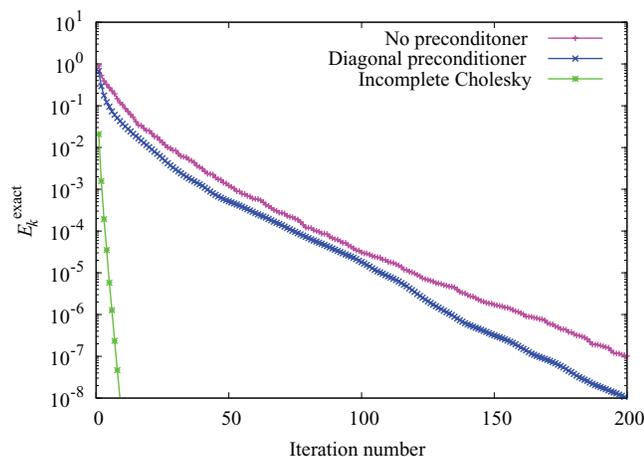


FIG. 1. Effect of preconditioning on the convergence rate in the Krylov subspace method for computing Brownian forces. For this test, an intracellular suspension of 1152 particles with a volume fraction of 0.51 is used.

the matrix is diagonal dominant. This reduces the number of iterations only slightly. The next preconditioner we employ is an incomplete Cholesky factorization, which was also used in previous studies.^{12–14} The incomplete Cholesky factor of a positive definite matrix \mathbf{A} , which we will denote by \mathbf{L}_0 , is given by a sparse lower triangular matrix that approximates the complete Cholesky factor, but is as sparse as \mathbf{A} .⁴⁴ As suggested by Torres and Gilbert,¹⁴ the reverse Cuthill-McKee ordering⁵² was applied to the sparse matrix not only to improve convergence but also to minimize breakdown in calculating the preconditioner. Use of the incomplete Cholesky preconditioner dramatically accelerates the convergence rate in the preconditioned Krylov subspace method. For a stopping tolerance of $E_k^{\text{exact}} = 10^{-4}$, this preconditioning reduces the number of iterations by a factor of 20.

The E_k^{exact} and E_k estimates during the preconditioned Krylov subspace iterations with the incomplete Cholesky factor for the intracellular-like suspension are shown in Fig. 2. Although E_k underestimates the exact errors by one order of magnitude throughout the iterations, it monotonically decreases and follows E_k^{exact} as was previously seen in the BD case.¹⁶ This result suggests that E_k is useful for estimating the accuracy of the preconditioned Krylov subspace approximation. The iterations can be stopped when the estimate E_k falls below a user-supplied threshold.

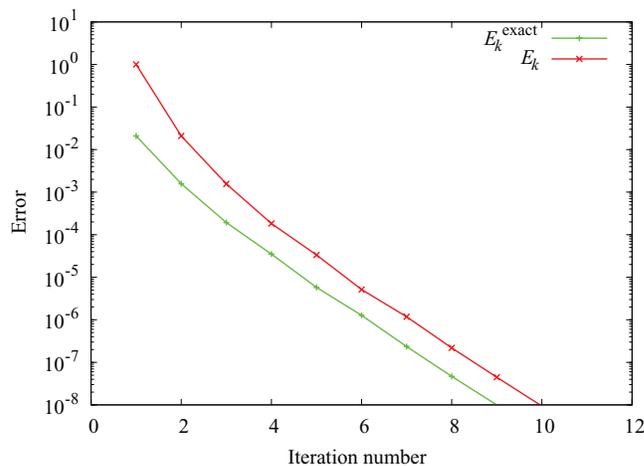


FIG. 2. Comparison of E_k^{exact} and E_k in the preconditioned Krylov subspace method for computing Brownian noise. An incomplete Cholesky preconditioner was used. The simulated system is an intracellular-like suspension. E_k with $k = 1$ was set to 1.

2. Convergence rate for solving linear systems

Once the correlated Brownian force vector is obtained, the next computationally demanding task in the mid-point algorithm is solving the two linear systems at steps (III) and (VI). To solve these systems, we use the preconditioned CG method, in which we can reuse the preconditioner constructed for the Krylov Brownian noise approximation at step (II). Also, as explained in Sec. II, we may calculate initial guesses $\tilde{\mathbf{u}}^0$ for the first linear system without additional computational cost; this may reduce the number of necessary iterations in the CG process.

Timings and iteration counts for our simulation method, FLD, are summarized in Table I. Different preconditioners are tested, as well as the use of initial guesses. The simulated system is the intracellular-like suspension model. The results show that the incomplete Cholesky preconditioner significantly reduces the number of iterations (by a factor of about 25) required to achieve convergence of the linear solutions. The initial guesses for the first linear system of each time step also reduces the number of iterations required to achieve the desired accuracy, from 10 to an average of 5.7 iterations. The initial guesses for the second linear system of each time step are the solutions from the first linear system. The results show that this second linear system can generally

TABLE I. Breakdown of computational time for FLD with different preconditioning methods, and with and without initial guesses for the first linear system. Timings are shown in seconds. Iteration counts are shown in brackets. For iterative convergence criteria, $E_k < 10^{-3}$ for the Brownian noise approximation and the relative residual norm $< 10^{-6}$ for CG are used. The simulated system is an intracellular suspension of 1152 particles with volume fraction 0.51. Occupancy of non-zero elements in the resistance tensor is 2%. All values are averaged over ten independent calculations using different particle configurations.

	No preconditioner		Diagonal preconditioner		Incomplete Cholesky	
	No initial guess	With initial guess	No initial guess	With initial guess	No initial guess	With initial guess
Preconditioner construction	...		Negligible		0.08	
Krylov Brownian noise approximation	0.05 (21)		0.06 (22)		0.04 (4.1)	
CG for the first linear system	0.15 (238)	0.13 (210)	0.12 (192)	0.10 (164)	0.02 (10)	0.01 (5.7)
CG for the second linear system	0.06 (101)		0.05 (75)		0.01 (5.0)	
Total	0.26	0.24	0.23	0.21	0.15	0.14

TABLE II. Number of iterations (and time in units of seconds) in the Krylov Brownian noise approximation for various convergence criteria E_k and the subsequent preconditioned CG method using the initial guesses constructed from the available Krylov subspace. Convergence criterion of preconditioned CG is relative residual norm $< 10^{-6}$. For this test, an intracellular suspension of 1152 particles with a volume fraction of 0.51 is used. All values are averaged over ten independent calculations using different particle configurations.

	$E_k = 0.1$	$E_k = 0.01$	$E_k = 0.001$	$E_k = 0.0001$
Krylov Brownian noise approximation	2.0 (0.022)	3.0 (0.027)	4.1 (0.037)	5.2 (0.042)
Preconditioned CG for the first linear system	7.8 (0.029)	6.9 (0.018)	5.7 (0.013)	4.7 (0.010)
Total	9.8 (0.051)	9.9 (0.045)	9.8 (0.050)	9.9 (0.052)

be solved with fewer than half the number of iterations when this initial guess is used. From the viewpoint of total computational time, the use of incomplete Cholesky preconditioning and the use of initial guesses give the best results among all the options tested. We note in particular that these results indicate that the incomplete Cholesky factors work very well in our model.

Since the quality of the initial guess for the first linear system of each time step depends on the size of the Krylov subspace used for computing the Brownian noise vector, we investigate the effect of different values of the threshold E_k (which affects the size of the Krylov subspace) on the number of iterations required to solve the first linear system. These results are shown in Table II. As expected, the number of iterations required for computing Brownian vectors increases with smaller values of E_k . For solving the first linear system, the number of iterations is reduced. Interestingly, the total number of iterations as well as the total computational time of the two iterative methods remains almost constant. The two methods compensate each other in computational cost via the initial guess $\tilde{\mathbf{u}}^0$. To obtain more accurate solutions in an iterative method, more iterations are required in general, resulting in an increase of total computational cost. However, our result indicates that in the mid-point algorithm with the above iterative approximation methods employed, we generally do not incur additional computational time for computing Brownian forces with higher accuracy.

B. Timing and scaling

We now turn to the scaling of the computational time of FLD as the number of simulated particles increases. For the following timing tests, the monodisperse suspension model was used. Figure 3 shows the computational time required for 10 time steps for simulated systems from 200 to 4000 particles. Timings for the original SD algorithm are also shown for comparison, where the far-field HI and its inverse were calculated only in the first time step and the second linear system was solved by the CG preconditioned by the complete Cholesky factor obtained for computing Brownian forces. The results show that FLD scales very nearly as $O(N)$ over the range of N tested. Although the number of iterations generally grows with N for many types of linear systems arising from physical problems, this was not observed in our tests, probably because only a small number of iterations were required. The original SD method scales as $O(N^3)$ for large N , which is expected due to using the Cholesky factorization at each time step and the inversion of the far-field HI matrix \mathbf{M}^∞ . For

small N , the Ewald summation of far-field HI, which is an $O(N^2)$ operation, dominates the computational time in original SD. Thus, the observed scaling for small N is less than cubic.

C. SD simulations using the approximation methods

In this section, we evaluate FLD with respect to diffusion coefficients and the pair correlation function using the intracellular-like suspension model. In this examination, the stopping threshold E_k is fixed to 0.001. For this value of E_k , the long-time self-diffusion coefficient D_L and inter-particle correlated motions obtained from the simulations were almost indistinguishable from the results when very stringent values of E_k are used. It is worth noting that the simulation with $E_k = 0.1$ was numerically unstable due to the inaccuracy of Brownian forces. In the following, we show simulation results using the original SD, FLD, and BD without HI (FD).

We first evaluate the short-time and long-time self-diffusion coefficients of particles. Figure 4 shows normalized short-time diffusion coefficients (D_S/D_0) and long-time diffusion coefficients (D_L/D_0) obtained from the SD simulations, where D_0 is the value at infinite dilution. Here, D_S was calculated using 100 different configurations randomly picked from the trajectories. The diagonal approximation to \mathbf{M}^∞ overestimates the short-time diffusivity compared to

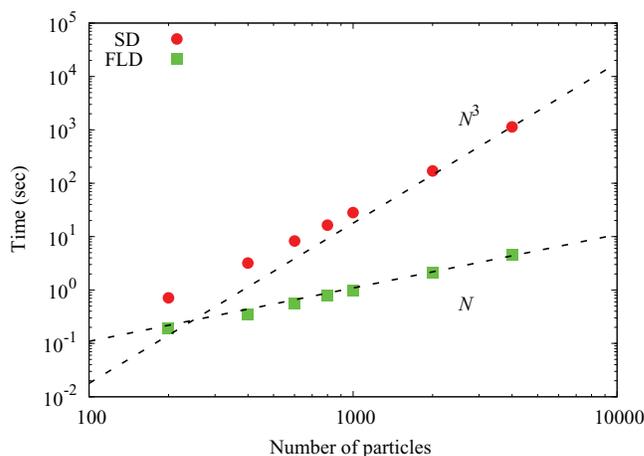


FIG. 3. Scaling of computational time for 10 time steps of FLD and SD simulations of a monodisperse suspension model at a volume fraction of 0.5 with periodic boundary conditions. In the SD simulation, the far-field hydrodynamic interaction term and its inverse were calculated only in the first time step. Dashed lines are a guide to the eye to show the approximate scaling with the number of particles.

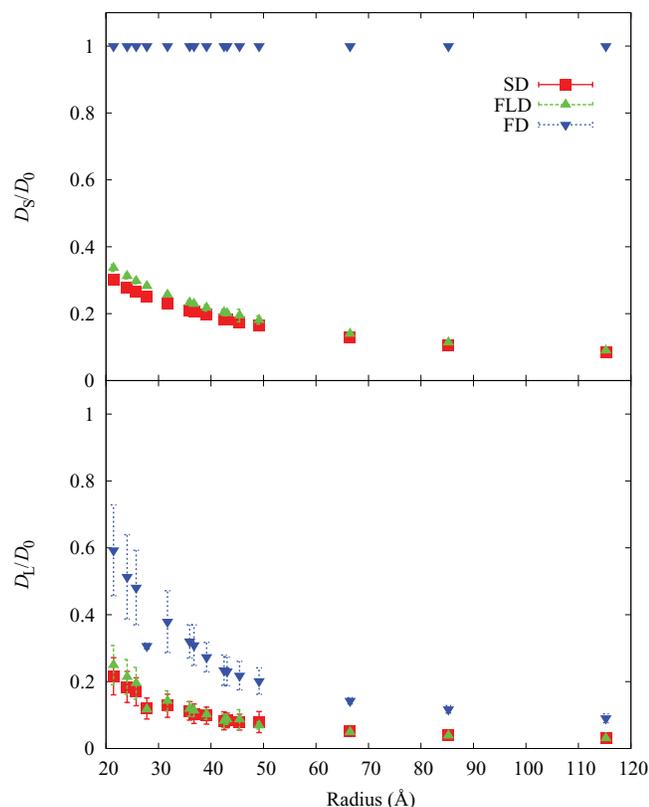


FIG. 4. (Upper) Normalized short-time and (lower) long-time diffusion coefficients of various particle radii in the intracellular-like suspension model obtained from SD, FLD, and FD simulations.

results with the original \mathbf{M}^∞ estimated by the Ewald method, with the deviation being most obvious for the smallest particles ($\sim 12\%$ for the smallest particle). The same trend is observed in the long-time diffusivity. FLD provides slightly higher D_L for small particles ($\sim 15\%$ for the smallest particle). However, these values are much more accurate than those from FD simulations. Although FLD gives slightly larger diffusion coefficients for small particles, the method still well represents the effects of HI on particle diffusivity. Thus, with regards to the both the short time and long time diffusion constants, long range hydrodynamic interactions are effectively screened.

Next, the normalized pair correlation function, C_{ij} , for particles are evaluated. Inter-particle correlated motions in time and space are also an important property caused by HI. In Fig. 5, C_{ij} for 24 Å–24 Å and 66 Å–66 Å particle pairs, which correspond to green fluorescent protein pairs and RNA-polymerase pairs, respectively, are shown as examples. In this figure, time and surface distance are normalized by a^2/D_0 and a , respectively. In SD simulations, correlated motions are obvious for both pairs. For small particle pairs, the correlated motions ($C_{ij} > 0.2$) range up to 2 in time and 0.75 in surface distance, which correspond to 110 ns and 18 Å, respectively. For large particle pairs, the correlation spans 12 in time and 0.4 in surface distance, which correspond to 14 μs and 26 Å, respectively. On the other hand, C_{ij} from FLD simulations are much smaller than in the original SD and are intermediate between the original SD and FD simulations. This underestimation is obvious for the small particle pair. Since small particles readily diffuse away from each other as com-

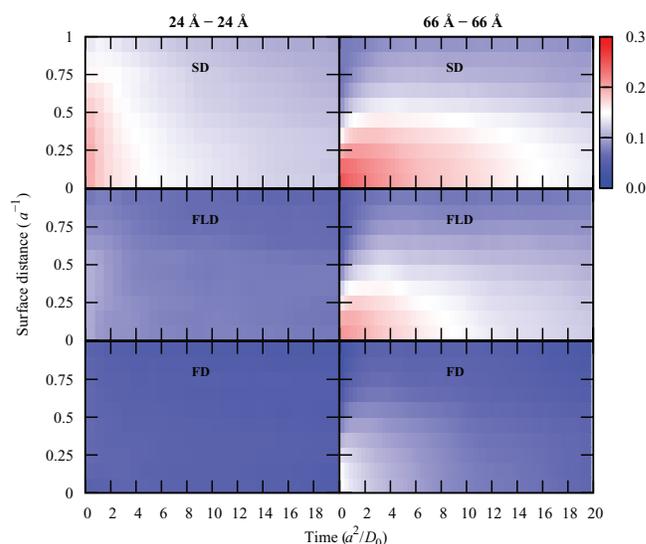


FIG. 5. Normalized pair correlation function, C_{ij} , for the 24 Å–24 Å and 66 Å–66 Å pairs in the intracellular-like suspension model obtained from SD, FLD, and FD simulations. Time and surface distance are normalized by a^2/D_0 and a , respectively.

pared to large particles and their time to stay near to each other is relatively short, they are sensitive to the absence of cross coupling terms of the far-field HI. This underestimation is a clear limitation of using a diagonal approximation for \mathbf{M}^∞ .

In the FD simulation, the correlations for both pairs are very weak and almost zero. For the large pairs in the FD simulation, weak but obvious correlated motions spanning 2 in time and 0.25 in surface distance are seen even in the absence of HI, which may be caused by a depletion attraction as described by Asakura-Oosawa theory on entropic forces between particles suspended in a solution of macromolecules.⁵³

IV. CONCLUSIONS

In this paper, we have described an efficient algorithm for performing large scale Stokesian dynamics simulations of Brownian particles that scales as $O(N)$. This new method reproduces the diffusive properties of the particles obtained with original SD. The key advance in our method is the use of preconditioning in various ways to reduce overall computation time. The preconditioner, an incomplete Cholesky factorization in our case, is computed once and used three times per time step, to accelerate the Brownian noise calculation and the solves of two linear systems. (Although not demonstrated in this paper, the preconditioner may also be reused for several time steps, which amortizes the cost of the constructing the preconditioner.) Additionally, the Krylov subspace generated for computing Brownian noise can be used to construct a good initial guess for the first linear system of each time step. We computed initial guesses under the assumption that $\mathbf{f}^p = 0$, but the technique may work even in the presence of non-hydrodynamic forces. We note that since our resistance matrix approximation is sparse, sparse direct methods, i.e., complete sparse Cholesky factorization, may also be used for computing Brownian noise and solving linear systems. These methods may be expected to be faster than our preconditioned

iterative approach for small numbers of particles, but we expect the iterative approach to be faster for large scale simulations, particularly when exploiting reuse of the preconditioner as described in this paper.

Banchio and Brady also proposed an $O(N)$ SD algorithm, where \mathbf{M}^∞ is replaced by a diagonal matrix and the incomplete Cholesky preconditioned CG method is used to solve linear equations.¹² The main difference between their algorithm and our algorithm is how to compute Brownian forces. In their algorithm, the Brownian forces are calculated as a pairwise sum over all near neighbors. An advantage of using a Krylov subspace method here is that this subspace can be reused to generate a good initial guess for the first linear equation of each time step without additional computational cost as described in Sec. II B 3. The use of the initial guess reduced the number of iterations in the preconditioned CG for the equation by 50% as shown in Sec. III A 2.

The generality of the Krylov subspace approximation for computing Brownian noise is also an important point. The approximation is not only valid for the SD algorithm, but one can use this idea in the BD algorithm as well. Here, the diffusion matrix is dense; thus, each iteration involves an $O(N^2)$ matrix-vector multiplication. We observed that the simple diagonal preconditioner slightly accelerates the convergence rate for polydisperse suspensions (data not shown). We are looking for proper preconditioners for large-scale and long-time BD simulations.

Fixman's Chebyshev method has been used in the past for computing Brownian noise in large scale simulations.¹⁵ In principle, we can apply the preconditioning idea to this method in the same manner we did for the Krylov subspace approximation. In addition, the Chebyshev method can also be used to construct initial guesses for the first linear system by computing these guesses at the same time as when Brownian noise is calculated. However, the major advantage of the Krylov subspace approximation over the Chebyshev approximation is that the former method does not require estimates of the extreme eigenvalues of a given matrix. This is a very important point when the Krylov subspace method is applied to approximate the square root of the resistance matrix \mathbf{R} , since the lubrication interactions are sensitive to small changes in particle distances so that the eigenvalue spectra of \mathbf{R} frequently change. If the Chebyshev approximation is applied to the problem, we must update these estimates frequently or suffer degraded convergence rates when conservative eigenvalue bounds are estimated *a priori* and then used throughout the simulation.

Although the diagonal approximation of the far-field HI may give particle diffusion constants within a reasonable error range, thereby implying neglect of long range HI is a reasonable approximation to describe diffusion, intermolecular correlated motions, especially for small particles, are significantly underestimated in the intracellular-like suspension. A similar deficiency was observed in monodisperse suspensions at low volume fractions (data not shown). If we optimize the diagonal elements of \mathbf{M}^∞ , the method may provide a diffusivity of particles that more closely match the SD results. However, this does not recover correlated motions in FLD. One way to fix this problem while keeping the overall scaling of

the algorithm as low as possible is the use of PME method to treat the far-field HI, which scale as $M \log(N)$.^{12,13,17} In this case, computation of the Brownian forces with covariance \mathbf{R} can be done by separately computing two Brownian forces with covariance $(\mathbf{M}^\infty)^{-1}$ and \mathbf{R}_{lub} .¹² The Krylov subspace method can be adapted for $(\mathbf{M}^\infty)^{-1}$ by seeking an approximation to the inverse square root of \mathbf{M}^∞ .

In conclusion, SD is an important simulation technique for understanding the dynamical motions of particles in fluids comprised not only of colloidal particles but also biological molecules. We proposed an SD algorithm based on iterative methods and the diagonal approximation of the hydrodynamic matrix for large-scale and long-time SD simulations. The method scales as $O(N)$. With regards to diffusion, long-range HI are significantly screened. However, the neglect of long range HI significantly underestimates intermolecular dynamical correlations. The complete hydrodynamic screening for freely moving particles is not theoretically validated. Therefore, the use of the diagonal approximation in our model has its limitations. We need further improvements to address this issue in an approach that retains the computational complexity of FLD and yet includes the appropriate level of HI so as to retain the temporal and spatial dependence of the intermolecular correlations.

ACKNOWLEDGMENTS

This research was supported in part by NIH Grant No. GM-37408 of the Division of General Medical Sciences of the National Institutes of Health.

APPENDIX: PRECONDITIONED KRYLOV SUBSPACE ALGORITHM FOR COMPUTING BROWNIAN FORCES

The preconditioned Krylov subspace algorithm for computing approximate vector $\tilde{\mathbf{y}} \approx \mathbf{S}\mathbf{z}$ with $\mathbf{S}\mathbf{S}^T = \mathbf{A}$ is described in the following.

Algorithm: Preconditioned Krylov subspace algorithm for computing $\tilde{\mathbf{y}} \sim N(0, \mathbf{A})$

```

1  Generate random  $\mathbf{z} \sim N(0, \mathbf{I})$ 
2   $\mathbf{v}_1 = \mathbf{z} / \|\mathbf{z}\|_2$ 
3  for  $j = 1$  to  $m$  do
4     $\mathbf{w} = \mathbf{L}^{-1} \mathbf{A} \mathbf{L}^{-T} \mathbf{v}_j$ 
5    if  $j > 1$  then
6       $\mathbf{w} = \mathbf{w} - h_{j-1,j} \mathbf{v}_{j-1}$ 
7    end
8     $h_{j,j} = \mathbf{w}^T \mathbf{v}_j$ 
9    if  $j < m$  then
10    $\mathbf{w} = \mathbf{w} - h_{j,j} \mathbf{v}_j$ 
11    $h_{j+1,j} = h_{j,j+1} = \|\mathbf{w}\|_2$ 
12    $\mathbf{v}_{j+1} = \mathbf{w} / h_{j+1,j}$ 
13   end
14   end
15   return  $\tilde{\mathbf{y}} = \|\mathbf{z}\|_2 \mathbf{L} \mathbf{V}_m \mathbf{H}_m^{1/2} \mathbf{e}_1$ 

```

¹M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, New York, 1987).

²D. L. Ermak and J. A. McCammon, *J. Chem. Phys.* **69**, 1352 (1978).

- ³C. W. Oseen, *Neuere Methoden und Ergebnisse in der Hydrodynamik* (Akademische Verlagsgesellschaft, Leipzig, 1927).
- ⁴J. Rotne and S. Prager, *J. Chem. Phys.* **50**, 4831 (1969).
- ⁵H. Yamakawa, *J. Chem. Phys.* **53**, 436 (1970).
- ⁶L. Durlofsky, J. F. Brady, and G. Bossis, *J. Fluid Mech.* **180**, 21 (1987).
- ⁷R. J. Phillips, J. F. Brady, and G. Bossis, *Phys. Fluids* **31**, 3462 (1988).
- ⁸D. R. Foss and J. F. Brady, *J. Fluid Mech.* **407**, 167 (2000).
- ⁹T. N. Phung, J. F. Brady, and G. Bossis, *J. Fluid Mech.* **313**, 181 (1996).
- ¹⁰T. Ando and J. Skolnick, *Proc. Natl. Acad. Sci. U.S.A.* **107**, 18457 (2010).
- ¹¹J. F. Brady, R. J. Phillips, J. C. Lester, and G. Bossis, *J. Fluid Mech.* **195**, 257 (1988).
- ¹²A. J. Banchio and J. F. Brady, *J. Chem. Phys.* **118**, 10323 (2003).
- ¹³A. Sierou and J. F. Brady, *J. Fluid Mech.* **448**, 115 (2001).
- ¹⁴F. E. Torres and J. R. Gilbert, "Large-Scale Stokesian Dynamics Simulations of Non-Brownian Suspensions," Technical Report C9600004, Xerox Research Centre of Canada, 1996.
- ¹⁵M. Fixman, *Macromolecules* **19**, 1204 (1986).
- ¹⁶T. Ando, E. Chow, Y. Saad, and J. Skolnick, *J. Chem. Phys.* **137**, 064106 (2012).
- ¹⁷D. Saintillan, E. Darve, and E. S. G. Shaqfeh, *Phys. Fluids* **17**, 033301 (2005).
- ¹⁸J. P. Hernández-Ortiz, J. J. de Pablo, and M. D. Graham, *Phys. Rev. Lett.* **98**, 140602 (2007).
- ¹⁹Y. Zhang, J. J. de Pablo, and M. D. Graham, *J. Chem. Phys.* **136**, 014901 (2012).
- ²⁰P. G. de Gennes, *Macromolecules* **9**, 594 (1976).
- ²¹P. Ahlrichs, R. Everaers, and B. Dunweg, *Phys. Rev. E* **64**, 040501(R) (2001).
- ²²H. C. Brinkman, *Appl. Sci. Res., Sect. A* **1**, 27 (1949).
- ²³L. Durlofsky and J. F. Brady, *Phys. Fluids* **30**, 3329 (1987).
- ²⁴I. Snook and W. van Meegen, *J. Colloid Interface Sci.* **100**, 194 (1984).
- ²⁵I. Snook, W. van Meegen, and R. J. A. Tough, *J. Chem. Phys.* **78**, 5825 (1983).
- ²⁶W. van Meegen and I. Snook, *Faraday Discuss.* **76**, 151 (1983).
- ²⁷W. van Meegen and I. Snook, *J. Chem. Soc., Faraday Trans. 2* **80**, 383 (1984).
- ²⁸W. van Meegen and I. Snook, *J. Chem. Phys.* **88**, 1185 (1988).
- ²⁹C. W. J. Beenakker and P. Mazur, *Physica A* **126**, 349 (1984).
- ³⁰R. C. Ball and J. R. Melrose, *Physica A* **247**, 444 (1997).
- ³¹A. Kumar and J. J. L. Higdon, *Phys. Rev. E* **82**, 051401 (2010).
- ³²M. Bybee, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2009.
- ³³A. Kumar, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2010.
- ³⁴G. Bossis and J. F. Brady, *J. Chem. Phys.* **87**, 5437 (1987).
- ³⁵J. F. Brady and G. Bossis, *Annu. Rev. Fluid Mech.* **20**, 111 (1988).
- ³⁶R. Kubo, *Rep. Prog. Phys.* **29**, 255 (1966).
- ³⁷T. Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide* (Springer, New York, 2002).
- ³⁸P. Szymczak and M. Cieplak, *J. Phys.: Condens. Matter* **23**, 033102 (2011).
- ³⁹H. Hasimoto, *J. Fluid Mech.* **5**, 317 (1959).
- ⁴⁰C. W. J. Beenakker, *J. Chem. Phys.* **85**, 1581 (1986).
- ⁴¹D. J. Jeffrey and Y. Onishi, *J. Fluid Mech.* **139**, 261 (1984).
- ⁴²M. Fixman, *J. Chem. Phys.* **69**, 1527 (1978).
- ⁴³W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing*, 2nd ed. (Cambridge University Press, New York, 1996).
- ⁴⁴Y. Saad, *Iterative Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2003).
- ⁴⁵V. Druskin and L. Knizhnerman, *SIAM J. Matrix Anal. A.* **19**, 755 (1998).
- ⁴⁶N. J. Higham, *Functions of Matrices: Theory and Computation* (Society for Industrial and Applied Mathematics, Philadelphia, 2008).
- ⁴⁷A. Parker and C. Fox, *SIAM J. Sci. Comput.* **34**, B312 (2012).
- ⁴⁸Y. Saad, private communication (2012).
- ⁴⁹M. K. Schneider and A. S. Willis, *Multidimens. Syst. Signal Process.* **14**, 295 (2003).
- ⁵⁰X. Liu, E. Chow, K. Vaidyanathan, and M. Smelyanskiy, *IEEE 26th International Parallel and Distributed Processing Symposium (IPDPS)* (IEEE, Shanghai, China, 2012), p. 36.
- ⁵¹A. J. C. Ladd, *J. Chem. Phys.* **93**, 3484 (1990).
- ⁵²A. George and J. W. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems* (Prentice Hall Professional Technical Reference, 1981).
- ⁵³S. Asakura and F. Oosawa, *J. Polym. Sci.* **33**, 183 (1958).