Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations

Tadashi Ando, Edmond Chow, Yousef Saad, and Jeffrey Skolnick

Citation: J. Chem. Phys. **137**, 064106 (2012); doi: 10.1063/1.4742347
View online: http://dx.doi.org/10.1063/1.4742347
View Table of Contents: http://jcp.aip.org/resource/1/JCPSA6/v137/i6
Published by the American Institute of Physics.

Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations

Tadashi Ando,[1] Edmond Chow,[2] Yousef Saad,[3] and Jeffrey Skolnick[1]
[1]*Center for the Study of Systems Biology, School of Biology, Georgia Institute of Technology, 250 14th Street NW, Atlanta, Georgia 30318-5304, USA*
[2]*School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, Georgia 30332-0765, USA*
[3]*Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, Minnesota 55455, USA*

(Received 8 June 2012; accepted 20 July 2012; published online 10 August 2012)

Hydrodynamic interactions play an important role in the dynamics of macromolecules. The most common way to take into account hydrodynamic effects in molecular simulations is in the context of a Brownian dynamics simulation. However, the calculation of correlated Brownian noise vectors in these simulations is computationally very demanding and alternative methods are desirable. This paper studies methods based on Krylov subspaces for computing Brownian noise vectors. These methods are related to Chebyshev polynomial approximations, but do not require eigenvalue estimates. We show that only low accuracy is required in the Brownian noise vectors to accurately compute values of dynamic and static properties of polymer and monodisperse suspension models. With this level of accuracy, the computational time of Krylov subspace methods scales very nearly as $O(N^2)$ for the number of particles N up to 10 000, which was the limit tested. The performance of the Krylov subspace methods, especially the "block" version, is slightly better than that of the Chebyshev method, even without taking into account the additional cost of eigenvalue estimates required by the latter. Furthermore, at $N = 10\,000$, the Krylov subspace method is 13 times faster than the exact Cholesky method. Thus, Krylov subspace methods are recommended for performing large-scale Brownian dynamics simulations with hydrodynamic interactions. [http://dx.doi.org/10.1063/1.4742347]

I. INTRODUCTION

Brownian dynamics (BD) is a computational technique for simulating the motion of macromolecules in a fluid environment.[1] Globular molecules such as proteins may be treated as coarse-grained particles, possibly of different sizes, while polymers such as DNA/RNA or proteins may be treated, for example, by a bead-spring model. Interactions between particles (or beads) with the solvent are modeled by random forces corresponding to particle collisions with solvent molecules, as well as a Stokes drag force proportional to particle velocity.

The solvent also mediates interactions between the particles themselves. This gives rise to so-called hydrodynamic interactions (HI), where the motion of one particle through the fluid induces a force on all the other particles. If one is only interested in equilibrium thermodynamic properties, HI do not play any role and can be neglected.[1] On the other hand, it is essential to include HI to correctly capture the dynamics of colloidal spheres, macromolecules, and swimming bacteria at a low Reynolds number; in particular, their collective, intermolecular motions can give rise to qualitatively different dynamic behavior.[2–4]

Hydrodynamic interactions are modeled by a configuration-dependent diffusion matrix, $\mathbf{D}$, of size $3N \times 3N$ for a system of N particles. This matrix is dense, owing to the long-ranged nature of HI. The BD propagation equation can be expressed as[1]

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \frac{\Delta t}{k_B T}\mathbf{D}\mathbf{F} + (\nabla \cdot \mathbf{D})\,\Delta t + \mathbf{g}, \tag{1}$$

$$\langle \mathbf{g} \rangle = 0, \quad \langle \mathbf{g}\mathbf{g}^T \rangle = 2\mathbf{D}\Delta t. \tag{2}$$

Here, $\mathbf{r}$ is the position vector of the N particles, t is the time, Δt is the time step length, k_B is Boltzmann's constant, T is the temperature, and $\mathbf{F}$ is the $3N$-dimensional force vector determined by the gradient of potential energy. The Rotne-Prager-Yamakawa (RPY) tensor has been widely used for estimating $\mathbf{D}$, since the tensor is positive definite for all particle configurations even when particles overlap.[5,6] The RPY tensor also has the property that $\nabla \cdot \mathbf{D} = 0$ so that in this case the third term in the right hand side of Eq. (1) is zero.[1] At each time step of the BD algorithm, a Brownian displacement vector $\mathbf{g}$ must be computed from a multivariate Gaussian distribution with mean zero and covariance $2\mathbf{D}\Delta t$, which can be calculated by

$$\mathbf{g} = \sqrt{2\Delta t}\mathbf{y} = \sqrt{2\Delta t}\mathbf{B}\mathbf{z}, \tag{3}$$

with

$$\mathbf{D} = \mathbf{B}\mathbf{B}^T. \tag{4}$$

Here, $\mathbf{z}$ is a standard normal vector. By the change-of-variable formula for probability distributions, the correlated vector $\mathbf{y} = \mathbf{B}\mathbf{z}$ has the Gaussian distribution $N(0, \mathbf{D})$. The factorization

in Eq. (4) is not unique, and any $\mathbf{B}$ satisfying Eq. (4) can be used.[7,8] The computation of the correlated vector $\mathbf{y}$ is generally the bottleneck in a BD simulation with HI.[9,10]

The standard technique for computing the correlated vector $\mathbf{y}$ is to compute $\mathbf{B}$ as the lower triangular Cholesky factor of $\mathbf{D}$, and to then form $\mathbf{y} = \mathbf{B}\mathbf{z}$. This technique was used by Ermak and McCammon in their original BD algorithm.[1] The cost of computing the Cholesky factorization is $O(N^3)$, although this cost in practice can be amortized over many time steps if $\mathbf{D}$ changes slowly.

The other major technique used in BD with HI is the Chebyshev polynomial approximation proposed by Fixman.[7] In this approximation technique, an approximate correlated vector is computed as $p(\mathbf{D})\mathbf{z}$, where $p(\mathbf{D})$ is a polynomial in $\mathbf{D}$ that approximates the principal square root of $\mathbf{D}$. This square root corresponds to the factorization in Eq. (4), where $\mathbf{B} = \mathbf{B}^T$. The technique is based on Chebyshev polynomials and requires estimates of the extreme eigenvalues of $\mathbf{D}$. The matrix $p(\mathbf{D})$ itself is not necessary and is never formed explicitly, and thus $O(N^3)$ matrix-matrix multiplications are avoided. The arithmetic complexity is observed to grow as $O(N^{2.25})$.[7,8]

Recently, a new method called the truncated expansion approximation (TEA) was proposed for calculating correlated vectors in BD.[11] TEA assumes a particular form for these correlated vectors and is particularly effective in cases where the correlations among all particles are approximately equal and relatively weak. This method has spurred much interest, attesting to the growing importance of fast methods for computing HI.[12] TEA has been found to work efficiently for bead-spring random polymers.[11,13] However, when multiple beads are assembled into compact structures, the TEA method does not show the correct scaling of translational diffusivity with N.[14]

In this paper, we study Krylov subspace methods for computing correlated Brownian displacements for use in BD. The methods are not new in the numerical analysis community,[15–17] but to the best of our knowledge, they appear to be unknown in the BD literature. These methods have two major advantages over Chebyshev polynomial approximations. First, estimates of the extreme eigenvalues of $\mathbf{D}$, required in the Chebyshev approximation, are not required in Krylov subspace methods. This is a great simplification over Chebyshev approximations. Second, block versions of Krylov subspace methods converge faster than Chebyshev approximations, and therefore require fewer computations for the same level of accuracy. Block versions of Krylov subspace methods are applicable when $\mathbf{D}$ changes slowly and can be reused for several consecutive time steps. Finally, in this paper, we also study the accuracy required by Krylov subspace methods in BD simulations. This is done for three different simulation models.

II. THEORY

A. Chebyshev polynomial approximations

The principal square root of a symmetric, positive definite matrix $\mathbf{D}$ may be approximated by a polynomial, $p(\mathbf{D})$, where $p(\lambda)$ is small, when λ is an eigenvalue of $\mathbf{D}$. Fixman[7] proposed an approximation to the Brownian correlated vector $\mathbf{y} \approx p(\mathbf{D})\mathbf{z}$ based on Chebyshev polynomials. Such polynomial approximations of the square root of a matrix times a vector (and in general, any function of a matrix times a vector) appeared soon afterward in the numerical analysis literature,[15–17] but these studies were unaware of Fixman's contribution. An essential feature of these approximations is that $p(\mathbf{D})$ is not needed and is never computed explicitly; rather, only $p(\mathbf{D})\mathbf{z}$ is required, which can be computed much more efficiently.

Chebyshev polynomials have the property that they are small in the interval $[-1, 1]$. To approximate $f(\mathbf{D})$, where the function f in our case is the square root, and where the spectrum of $\mathbf{D}$ lies in the interval [a, b], we approximate instead the function $g(\mathbf{D}_s)$, where

$$\mathbf{D}_s = \frac{2}{b-a}\mathbf{D} - \frac{a+b}{b-a}\mathbf{I}, \tag{5}$$

which has eigenvalues in the interval $[-1, 1]$. The function g is then defined as

$$g(\mathbf{D}_s) = f\left(\frac{b-a}{2}\mathbf{D}_s + \frac{a+b}{2}\mathbf{I}\right), \tag{6}$$

where $\mathbf{I}$ in the above expressions denotes the identity matrix. In other words, we use Chebyshev polynomials to approximate $g(\mathbf{D}_s)$, which is equal to $f(\mathbf{D})$. Thus, the extreme eigenvalues of $\mathbf{D}$ are required to perform the above change of variables. The more accurate the estimates of these extreme eigenvalues, the faster is the convergence. Convergence can be very poor if any eigenvalue of $\mathbf{D}$ lies outside these estimates. (The convergence rate is the inverse of the degree of the polynomial required for a given level of accuracy.)

Beginning with Fixman, procedures have been developed for estimating the extreme eigenvalues of $\mathbf{D}$ and also for updating these estimates as a BD simulation progresses.[8,13] The updates may be performed dynamically, according to measures of the accuracy of the Chebyshev approximation. A recent comparison of these techniques found that the run time may differ significantly with different approaches.[13] As shown below, Krylov subspace approximations employed in this paper do not require eigenvalue estimates.

The Chebyshev polynomial expansion up to degree L is

$$p_L(\mathbf{D}_s) = \frac{c_0}{2} + \sum_{k=1}^{L} c_k T_k(\mathbf{D}_s), \tag{7}$$

where c_k denotes the expansion coefficients and T_k denotes the kth Chebyshev polynomial. The expansion coefficients are computed by interpolating the function g at $L+1$ points, generally selected to be the Chebyshev nodes. Due to the discrete orthogonality property of Chebyshev polynomials, the coefficients are easily computed.

B. Truncated expansion approximation

In the TEA proposed by Geyer and Winter,[11] the correlated vector $\mathbf{y}$ is assumed to be of a specific form (an ansatz),

namely, $\mathbf{y} = \mathbf{S}_{\mathrm{TEA}}\mathbf{z}$, where

$$\mathbf{S}_{\mathrm{TEA}} = \mathbf{C}\,(\mathbf{D}_d)^{-1}\,(\mathbf{B}\cdot\mathbf{D})\,(\mathbf{D}_d)^{1/2}, \tag{8}$$

where $\mathbf{D}_d$ is the matrix that is the diagonal part of $\mathbf{D}$, the matrix $\mathbf{B}$ has diagonal entries 1 and off-diagonal entries β, the "dot" operator represents an element-wise product, and $\mathbf{C}$ is a diagonal matrix. TEA chooses the entries in $\mathbf{C}$ and the value of β so that

$$\mathbf{S}_{\mathrm{TEA}}\mathbf{S}_{\mathrm{TEA}}^{T} \approx \mathbf{D}, \tag{9}$$

which is the requirement that $\mathbf{y}$ approximately has covariance $\mathbf{D}$. The value of β is chosen based on the assumption that the off-diagonal couplings in $\mathbf{D}$ are small relative to the diagonal of $\mathbf{D}$; specifically, they are chosen by replacing the off-diagonal entries in $\mathbf{D}$ by their average value. The entries in $\mathbf{C}$ are chosen such that the diagonal of $\mathbf{S}_{\mathrm{TEA}}\mathbf{S}_{\mathrm{TEA}}{}^{T}$ matches the diagonal of $\mathbf{D}$. The overall procedure is $O(N^2)$. The computational cost of the method is somewhat more than that of three matrix-vector products with $\mathbf{D}$: computing the entries in $\mathbf{C}$, computing β, and multiplying by $\mathbf{D}$.

Details of this method are available elsewhere.[11] What we have presented here is an algebraic description of the method. The method has been shown to be very efficient and effective, in particular, for bead-spring chain models.[11,13]

C. Krylov subspace approximation

We now present Krylov subspace approximations for computing the correlated vector $\mathbf{y}$. Consider first the exact computation of $\mathbf{y}$ via the principal square root of $\mathbf{D}$, which is given by

$$\mathbf{y} = \mathbf{D}^{1/2}\mathbf{z} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\mathbf{U}^{T}\mathbf{z}, \tag{10}$$

where $\boldsymbol{\Lambda}$ is the $3N \times 3N$ diagonal matrix whose elements are the eigenvalues of $\mathbf{D}$, and $\mathbf{U}$ is the $3N \times 3N$ matrix whose columns are eigenvectors of $\mathbf{D}$. Computing the correlated vector directly this way requires an eigenvalue decomposition of $\mathbf{D}$, which is $O(N^3)$ computations and not any better than using the Cholesky factorization approach.

In the Krylov subspace approach, instead of the exact method, an approximation $\tilde{\mathbf{y}}$ to $\mathbf{D}^{1/2}\mathbf{z}$ is constructed from the Krylov subspace

$$K_m(\mathbf{D}, \mathbf{z}) = \mathrm{span}\{\mathbf{z}, \mathbf{D}\mathbf{z}, \ldots, \mathbf{D}^{m-1}\mathbf{z}\}, \tag{11}$$

where $m \leq 3N$. The approximation $\tilde{\mathbf{y}}$ can be observed to be a linear combination of vectors of the form $\mathbf{D}^{i}\mathbf{z}$, with $0 \leq i \leq m-1$, and thus, such an approximation has the form $p_{m-1}(\mathbf{D})\mathbf{z}$, where p_{m-1} is a polynomial of degree $m-1$ or less. Krylov subspace approximations are thus a form of polynomial approximation. Like the Chebyshev approximation, the coefficients of this polynomial are chosen by interpolating the square root function, although implicitly and at different points than those used by the Chebyshev method.[17] Thus, we expect Krylov subspace approximations and Chebyshev polynomial approximations of similar degree to have similar quality, although Krylov subspace approximations may be somewhat more efficient because $p_{m-1}(\lambda)$ is designed to be small when λ is an eigenvalue of $\mathbf{D}$, rather than uniformly small over the entire interval from the smallest to largest eigenvalues of $\mathbf{D}$, as in the Chebyshev case.

An important advantage of the Krylov subspace approximation approach over the Chebyshev approach is that estimates of the spectrum of $\mathbf{D}$ are not necessary.

Since $\mathbf{D}$ is symmetric, the Lanczos process can be used for constructing an orthonormal basis for the Krylov subspace. The approximation $\tilde{\mathbf{y}}$ is then constructed in this basis. Let a $3N \times m$ matrix $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ be an orthonormal basis for the Krylov subspace. The optimal approximation, one that minimizes the 2-norm of the error from this subspace, is

$$\mathbf{y}^{*} = \mathbf{V}_m\mathbf{V}_m^{T}\mathbf{D}^{1/2}\mathbf{z}, \tag{12}$$

which is the orthogonal projection of the exact solution onto the Krylov subspace. Let us now choose the basis $\mathbf{V}_m$ such that the first vector of this basis is $\mathbf{v}_1 = \mathbf{z}/\|\mathbf{z}\|_2$. Here, $\|\mathbf{z}\|_2$ is the 2-norm of vector $\mathbf{z}$, which is the square root of the sum of squares of the entries of the vector. Thus, $\mathbf{z} = \|\mathbf{z}\|_2\mathbf{V}_m\mathbf{e}_1$ with $\mathbf{e}_1$ being the first column of the $m \times m$ identity matrix. Then we can write the optimal approximation as

$$\mathbf{y}^{*} = \|\mathbf{z}\|_2\,\mathbf{V}_m\mathbf{V}_m^{T}\mathbf{D}^{1/2}\mathbf{V}_m\mathbf{e}_1. \tag{13}$$

Now, we define the symmetric tridiagonal matrix $\mathbf{H}_m = \mathbf{V}_m^{T}\mathbf{D}\mathbf{V}_m$, which is automatically calculated in the Lanczos process as shown below. After the matrix $\mathbf{H}_m$ is obtained, one can easily compute its eigenvalue decomposition when $m \ll 3N$,

$$\mathbf{H}_m = \mathbf{P}_m\boldsymbol{\Sigma}_m\mathbf{P}_m^{T}. \tag{14}$$

Here, $\boldsymbol{\Sigma}_m$ is the $m \times m$ diagonal matrix whose diagonal elements are eigenvalues of $\mathbf{H}_m$ and $\mathbf{P}_m$ is the $m \times m$ matrix whose columns are eigenvectors of $\mathbf{H}_m$. The eigenvalues of $\mathbf{H}_m$ are known to approximate the extremal eigenvalues of $\mathbf{D}$, and $\mathbf{V}_m\mathbf{P}_m$ are the corresponding approximations to the eigenvectors.[18,19] Therefore, we approximate $\mathbf{V}_m^{T}\mathbf{D}^{1/2}\mathbf{V}_m$ as

$$\begin{aligned}\mathbf{V}_m^{T}\mathbf{D}^{1/2}\mathbf{V}_m &= \mathbf{V}_m^{T}(\mathbf{U}\boldsymbol{\Lambda}^{1/2}\mathbf{U}^{T})\mathbf{V}_m \\ &\approx \mathbf{V}_m^{T}\,(\mathbf{V}_m\mathbf{P}_m)\,\boldsymbol{\Sigma}_m^{1/2}\,(\mathbf{V}_m\mathbf{P}_m)^{T}\,\mathbf{V}_m \\ &= \mathbf{P}_m\boldsymbol{\Sigma}_m^{1/2}\mathbf{P}_m^{T} \\ &= \mathbf{H}_m^{1/2}.\end{aligned} \tag{15}$$

Here, we used $\mathbf{V}_m^{T}\mathbf{V}_m = \mathbf{I}$. Thus, an approximation for Eq. (13) can be written as

$$\tilde{\mathbf{y}} \approx \|\mathbf{z}\|_2\,\mathbf{V}_m\mathbf{H}_m^{1/2}\mathbf{e}_1. \tag{16}$$

The approximation is thus based on computing the matrix square root on a much smaller subspace, where it is inexpensive to compute exactly, and then mapping the result to the original space. Note that like the Chebyshev polynomial approximation, an approximation to $\mathbf{D}^{1/2}$ is never computed explicitly.

ALGORITHM I. Krylov subspace algorithm for computing $\tilde{\mathbf{y}} \sim N(0, \mathbf{D})$.

```
1   Generate random z ~ N(0,I)
2   v_1 = z/||z||_2
3   for j = 1 to m do
4       w = D v_j
5       if j > 1 then
6           w = w − h_{j−1,j} v_{j−1}
7       end
8       h_{j,j} = w^T v_j
9       if j < m then
10          w = w − h_{j,j} v_j
11          h_{j+1,j} = h_{j,j+1} = ||w||_2
12          v_{j+1} = w/h_{j+1,j}
13      end
14  end
15  return ỹ = ||z||_2 V_m H_m^{1/2} e_1
```

1. Lanczos process

The matrix $\mathbf{D}$ is symmetric and thus an orthonormal basis $\mathbf{V}_m$ for the Krylov subspace can be computed using the Lanczos process. The matrix $\mathbf{H}_m$ is computed automatically in this method. This is the same process used in the Lanczos method for solving symmetric eigen-problems, where the spectrum of $\mathbf{H}_m$ approximates the spectrum of $\mathbf{D}$. The overall algorithm for computing the approximate correlated vector $\tilde{\mathbf{y}}$ with Gaussian distribution $N(0, \mathbf{D})$ is shown in Algorithm I. In the algorithm, m is the number of Lanczos steps.

The most expensive operation in this algorithm is the $O(N^2)$ dense matrix-vector multiplication by $\mathbf{D}$, which is performed m times. For large N, the overall running time of this algorithm is $O(N^2)$ due to this matrix-vector multiplication. The result of Algorithm I is a polynomial approximation of degree $m-1$. In comparison to the Chebyshev technique, an approximation of degree $m-1$ requires $m-1$ matrix-vector multiplies, i.e., one fewer than required by the Krylov subspace approximation. The extra matrix-vector multiplication required by the Krylov subspace approximation is due to the need to form $\mathbf{H}_m$, which is equal to $\mathbf{V}_m^T \mathbf{D} \mathbf{V}_m$.

The number of steps m in the Lanczos process may be chosen beforehand, or the Lanczos process may be iterated until a certain convergence criterion is satisfied. A convergence criterion may be applied after the computation of $h_{j,j}$ using an approximation $\tilde{\mathbf{y}}_k$ computed at each Lanczos step, and using the basis vectors computed thus far. This approximation is used in the convergence criterion to be described later.

2. Block-Lanczos process for multiple vectors

In BD simulations, the covariance matrix generally changes slowly, making it possible to use the same covariance matrix for several time steps. When the Cholesky factorization approach is used, this avoids the need to compute the factorization at every time step. For further computational efficiency, the correlated vectors for several time steps should be computed simultaneously, as one *block* of vectors, rather than one vector at a time. The multiplication of the Cholesky factor by a *block* of standard normal vectors should be carried out such that all the multiplications are performed while traversing the elements of the Cholesky factor only once. This reduces data movement which has relatively high cost compared to arithmetic computations on modern processors.[20] If the Chebyshev polynomial approach is used, it is likewise advantageous to compute multiple vectors simultaneously because of the efficiency of computing matrix-vector products with a block of vectors.

For the Lanczos approach, a block variant can also be used when multiple sample vectors can be computed for the same covariance matrix. Like the above, computational efficiency is gained by operating on a block of vectors simultaneously. However, there is an additional important advantage: the solution for each vector can be sought in a larger subspace (larger than in the single-vector case) for very little additional cost. Thus, the solutions converge much more quickly and require fewer matrix-vector multiplications in total.

Consider the block-Lanczos process for a block of s independent standard normal vectors, $\mathbf{z}_1, \ldots, \mathbf{z}_s$. After m steps, the block-Lanczos process computes an orthonormal basis $\mathbf{V}_{ms}$ for the combined subspaces

$$K_m(\mathbf{D}, \mathbf{z}_1) + \cdots + K_m(\mathbf{D}, \mathbf{z}_s). \tag{17}$$

Each step of the algorithm produces s basis vectors, one vector for each of the subspaces in the "sum" above. The block-Lanczos process also computes the $ms \times ms$ banded matrix $\mathbf{H}_{ms} = \mathbf{V}_{ms}^T \mathbf{D} \mathbf{V}_{ms}$.

Let $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_s]$ denote a block of s standard normal vectors. Let $\mathbf{V}_j$ denote the jth block of s vectors computed and available at the beginning of step j of the block-Lanczos process. An approximation to a block of correlated vectors with covariance $\mathbf{D}$ in the space spanned by $\mathbf{V}_{ms}$ is given by

$$\mathbf{Y}^* = \mathbf{V}_{ms} \mathbf{V}_{ms}^T \mathbf{D}^{1/2} \mathbf{Z}. \tag{18}$$

Let $\mathbf{Z} = \mathbf{QR}$ be the reduced QR factorization of $\mathbf{Z}$ and choose $\mathbf{V}_1 = \mathbf{Q}$. Then

$$\mathbf{Y}^* = \mathbf{V}_{ms} \mathbf{V}_{ms}^T \mathbf{D}^{1/2} [\mathbf{V}_1 \cdots \mathbf{V}_m] \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix}, \tag{19}$$

where the quantity in square brackets is $\mathbf{V}_{ms}$ and where $\mathbf{R}$ is $s \times s$. As in the single-vector case, we now make the approximation

$$\tilde{\mathbf{Y}} = \mathbf{V}_{ms} \mathbf{H}_{ms}^{1/2} \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix}. \tag{20}$$

This procedure is embodied in the algorithm below. We use $\mathbf{H}_{i,j}$ to denote the (i, j) block (of size $s \times s$) of matrix $\mathbf{H}_{ms}$. Like Algorithm I, the cost of the algorithm is dominated by the matrix-vector multiplication by $\mathbf{D}$.

III. MODELS AND SIMULATION METHODS

A. Random polymer chain model

Random polymer bead-spring models have been widely used not only for theoretical studies of hydrodynamic interactions but also for evaluating simulation accuracy.[1,11,13,21,22]

ALGORITHM II. Block-Krylov subspace algorithm for computing a block of s correlated vectors, $\tilde{\mathbf{Y}}$, each vector with distribution $N(0, \mathbf{D})$.

```
Generate a block of s vectors Z, each vector with distribution N(0,I)
Compute reduced QR factorization V_1 R = Z
for j = 1 to m do
    W = D V_j
    if j > 1 then
        W = W − V_{j−1} H_{j−1,j}
    end
    H_{j,j} = V_j^T W
    if j < m then
        W = W − V_j H_{j,j}
        Compute reduced QR factorization V_{j+1} H_{j+1,j} = W
        H_{j,j+1} = H_{j+1,j}
    end
end
return Ỹ = V_ms H_ms^{1/2} (R; 0)
```

The polymer consists of N beads of radius a, each connected to their first neighbors by harmonic springs with potential

$$V_{\mathrm{s}} = \frac{1}{2} k_{\mathrm{s}} (x - 2a)^2 . \tag{21}$$

Here, k_{s} is the spring constant and x is the distance between the beads. To prevent bead overlap, a repulsive harmonic potential between beads i and j with $|i - j| \geq 2$ is applied:

$$V_{\mathrm{r}} = \begin{cases} \frac{1}{2} k_{\mathrm{r}} (x - 2a)^2 & \text{for } x < 2a, \\ 0 & \text{for } x \geq 2a, \end{cases} \tag{22}$$

where k_{r} is the force constant. In this study, $k_{\mathrm{s}} = k_{\mathrm{r}} = 125\ k_{\mathrm{B}}T/a^2$ was used. Polymers with $N = 10$, 20, 40, 60, 80, 100, 200 were examined in BD simulations. For timing tests and to study convergence of the Krylov subspace methods, we used much longer polymers. Five independent initial configurations were generated for each chain length, where beads were randomly placed without significant overlaps under the constraint of distances between beads i and $i + 1$ of $2a$. These configurations were then subjected to short-time energy minimization. This model does not have any attractive interactions between beads and thus corresponds to a polymer in a good solvent.[23]

B. Collapsed chain model

For many applications of BD, especially for biological simulations, attractive interactions are applied to particles or beads to analyze the dynamics of self-organization of molecules and molecular associations.[24–26] Therefore, testing a model with attractive interactions is quite important. Here, we examine a simple chain model that collapses to a compact conformation. This model corresponds to a polymer in a poor solvent.[23]

Adjacent beads are connected by the harmonic springs described by Eq. (21). Between beads i and j with $|i - j| \geq 2$, a Lennard-Jones 12-6 potential is applied,

$$V_{\mathrm{LJ}} = \varepsilon_{\mathrm{LJ}} \left[\left(\frac{\sigma_{\mathrm{LJ}}}{x} \right)^{12} - 2 \left(\frac{\sigma_{\mathrm{LJ}}}{x} \right)^{6} \right], \tag{23}$$

where $\varepsilon_{\mathrm{LJ}}$ is the energy depth and σ_{LJ} is the distance at the energy minimum. In this study, $\varepsilon_{\mathrm{LJ}} = 1\ k_{\mathrm{B}}T$ and $\sigma_{\mathrm{LJ}} = 2a$ were used. Polymers of length $N = 10$, 20, 40, 60, 80, 100, 200 were considered in BD simulations. Completely extended configurations were used as initial states and five independent BD simulations were performed with different random number seeds. To further study the convergence of the Krylov subspace methods, we used long polymers with $N = 1000$. For these long polymers, initial configurations were generated by the same procedure as in the random polymer model, and then BD simulations with HI were performed to obtain equilibrated states.

C. Monodisperse suspension model

Another model we used for evaluating simulation accuracy is a monodisperse suspension of N particles of radius a. To help prevent bead overlap, a repulsive harmonic potential between particles as described by Eq. (22) was used. Five different volume fractions Φ of 0.1, 0.2, 0.3, 0.4, and 0.5 were considered in periodic boxes. A value of N of 200 was used in BD simulations. For convergence tests, N of 1000 was also examined. Five independent initial configurations were generated for each condition, where particles were randomly placed without significant overlaps in simulation boxes and subjected to short-time energy minimization. As described in Sec. III D, we use the RPY tensor to account for HI in the BD simulations. The RPY tensor represents only the far-field part of HI and the tensor is not appropriate by itself for simulations at high volume fractions. For simulations at high volume fractions, a more sophisticated formulation that also incorporates near-field HI is necessary.[27,28] In these formulations, the RPY tensor corresponding to high volume fraction is utilized to represent the far-field part of HI (technically, it is the *inverse* of this RPY tensor that is the far-field component of the covariance matrix).

D. Brownian dynamics simulation and analysis

The integration algorithm for BD described in Eq. (1) was used. In this work, we employ the RPY tensor for estimating $\mathbf{D}$,[5,6]

$$\mathbf{D}_{ij} = \begin{cases} \frac{k_{\mathrm{B}}T}{6\pi\eta a} \mathbf{I} & i = j, \\ \frac{k_{\mathrm{B}}T}{8\pi\eta r_{ij}} \left[(\mathbf{I} + \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}) + \frac{2a^2}{r_{ij}^2} \left(\frac{1}{3}\mathbf{I} - \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij} \right) \right] & i \neq j, r_{ij} \geq 2a, \\ \frac{k_{\mathrm{B}}T}{6\pi\eta a} \left[\left(1 - \frac{9}{32}\frac{r_{ij}}{a} \right) \mathbf{I} + \frac{3}{32}\frac{r_{ij}}{a} \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij} \right] & i \neq j, r_{ij} < 2a, \end{cases} \tag{24}$$

where i and j are the indices of particles, $\mathbf{r}_{ij}$ is $\mathbf{r}_i - \mathbf{r}_j$, r_{ij} is the length of $\mathbf{r}_{ij}$, $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$, and $\mathbf{I}$ is the unit tensor. For periodic boundary conditions, since HI have a long-range nature similar to electrostatic interactions, an Ewald summation of the RPY tensor to obtain $\mathbf{D}$ of the system is necessary not only for accuracy but also for giving a positive definite matrix $\mathbf{D}$. We used the Ewald sum technique originally derived by Beenakker[29] and modified by Zhou and Chen[30] to allow for particle overlap.

For the sake of simplicity, all quantities are expressed in dimensionless units. Length is in units of the bead radii, a, and time is units of a^2/D_0, where $D_0 = k_B T/6\pi\eta a$ is the diffusion coefficient of a single bead in dilute solution. Simulations were performed for 5.0×10^6 steps with a time step Δt of 0.002 for the random polymer chain and monodisperse suspension models. For the following analysis of the random polymer and monodisperse suspension models, the first 5.0×10^5 steps were discarded. For the collapsed chain model, at least 4.0×10^6 steps were run after the chains collapsed into their compact structures with Δt of 0.001 in the presence of HI and Δt of 0.0005 in the absence of HI.

Translational diffusion coefficients of the centers of masses, D_{cm}, were estimated by

$$6D_{\mathrm{cm}}\tau = \langle(\mathbf{r}_{\mathrm{cm}}(t+\tau) - \mathbf{r}_{\mathrm{cm}}(t))^2\rangle, \tag{25}$$

where the brackets indicate an average over configurations separated by a time difference τ and $\mathbf{r}_{\mathrm{cm}}$ is the center of mass of the polymer. Translational diffusion coefficients of particles, D, were calculated by the same equation where $\mathbf{r}_{\mathrm{cm}}$ is replaced by the position, $\mathbf{r}_i$, of particle i, and the brackets indicate an average over configurations separated by a time difference τ and over all particles.

IV. RESULTS AND DISCUSSION

A. Convergence

1. Convergence rate and error estimates

The Krylov subspace method iteratively improves the accuracy of the approximate correlated vector. An estimate of the error of this correlated vector is desired. We define the relative norm of the exact error of the kth approximation as

$$E_k^{\mathrm{exact}} = \frac{\|\tilde{\mathbf{y}}_k - \mathbf{D}^{1/2}\mathbf{z}\|_2}{\|\mathbf{D}^{1/2}\mathbf{z}\|_2}, \tag{26}$$

where $\mathbf{z}$ is the same standard normal vector used to compute $\tilde{\mathbf{y}}$. This of course cannot be computed in practice. We propose an approximation based on two consecutive iterates

$$E_k = \frac{\|\tilde{\mathbf{y}}_k - \tilde{\mathbf{y}}_{k-1}\|_2}{\|\tilde{\mathbf{y}}_{k-1}\|_2}. \tag{27}$$

This estimate is natural when convergence is monotonic, which is generally the case. The estimate is particularly good if convergence is rapid. In Figure 1(a), convergence of the Krylov method measured by E_k^{exact} and E_k for a random polymer with $N = 1000$ is shown as a representative example. (In this and later figures, the iteration count on the x axis is equal to the polynomial degree plus one of the Krylov subspace or Chebyshev approximation.) We observe that E_k closely follows E_k^{exact}. This result indicates that E_k may be used for monitoring convergence in real simulations. We thus propose using E_k as the convergence criterion of the Krylov subspace methods. When the estimate falls below a user-supplied threshold, then convergence is assumed and the iterations are stopped.

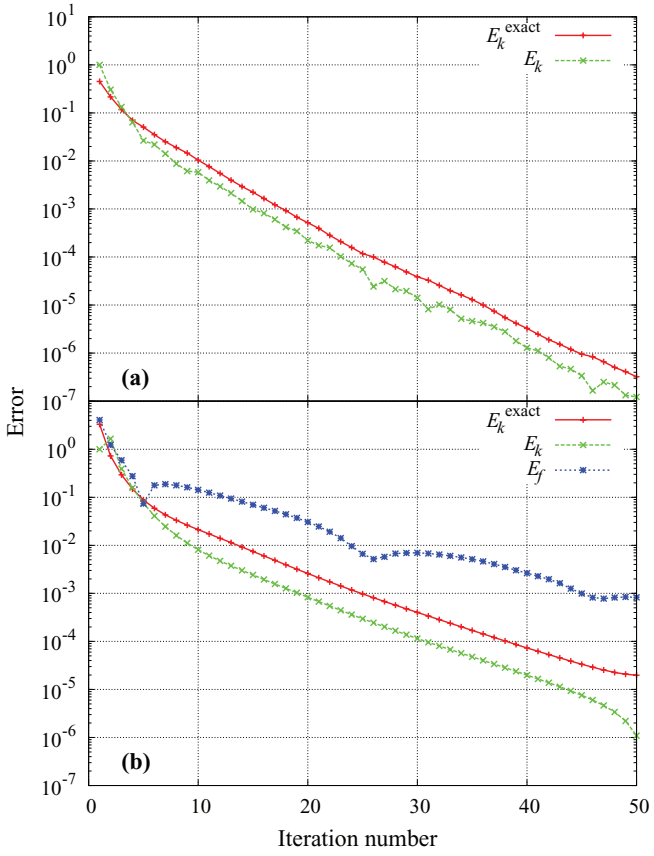

FIG. 1. Convergence of (a) the Krylov subspace method and (b) the Chebyshev method measured by various error estimates, E_k^{exact}, E_k, and E_f. Diffusion matrices were constructed from five configurations of random polymer chains with length $N = 1000$ and the results represent the average over the five configurations. Standard deviations for all data points are so small that they are not displayed. E_k with $k = 1$ was set to 1.

Jendrejack *et al.*[8] proposed the following quantity for monitoring the convergence of the Chebyshev method:

$$E_f = \sqrt{\frac{|\tilde{\mathbf{y}}^T\tilde{\mathbf{y}} - \mathbf{z}^T\mathbf{D}\mathbf{z}|}{\mathbf{z}^T\mathbf{D}\mathbf{z}}}. \tag{28}$$

If $\tilde{\mathbf{y}} = \mathbf{D}^{1/2}\mathbf{z}$ exactly, then $E_f = 0$. When E_f is large, then $\tilde{\mathbf{y}}$ is inaccurate. The quantity E_f can thus be used to adaptively control the Chebyshev polynomial approximation as a simulation progresses. For example, when E_f is large, this may indicate that the eigenvalue estimates are no longer accurate and/or that the polynomial degree is not large enough. The extreme eigenvalues are then recomputed and/or the polynomial degree is then adjusted and the current time step is repeated. A threshold of 10^{-3} has been suggested to indicate sufficient accuracy, although no strong justification has been given.[8,11,13]

The quantity E_f, however, turns out to be inappropriate for monitoring the Krylov subspace approximation, as this

approximation always produces a result such that $E_f = 0$. This is because the iterates produced by Krylov subspace methods are already scaled such that $E_f = 0$. To see this, we form

$$\begin{aligned}\tilde{\mathbf{y}}^T\tilde{\mathbf{y}} &= \|\mathbf{z}\|_2^2\,\mathbf{e}_1^T\left(\mathbf{H}_m^{1/2}\right)^T\mathbf{V}_m^T\mathbf{V}_m\mathbf{H}_m^{1/2}\mathbf{e}_1\\ &= \|\mathbf{z}\|_2^2\,\mathbf{e}_1^T\mathbf{H}_m\mathbf{e}_1\\ &\approx \|\mathbf{z}\|_2^2\,\mathbf{e}_1^T\left(\mathbf{V}_m^T\mathbf{D}\mathbf{V}_m\right)\mathbf{e}_1\\ &= \|\mathbf{z}\|_2^2\,\mathbf{v}_1^T\mathbf{D}\mathbf{v}_1\\ &= \mathbf{z}^T\mathbf{D}\mathbf{z}. \end{aligned} \tag{29}$$

In general, a small value of E_f does not always imply that $\tilde{\mathbf{y}}$ is accurate; any $\tilde{\mathbf{y}}$ can be "improved" by a scaling so that $E_f = 0$.

It is also possible for the Chebyshev approximation to use the error estimate Eq. (27). In this case, $\tilde{\mathbf{y}}_k$ and $\tilde{\mathbf{y}}_{k-1}$ correspond to the degree k and degree $k-1$ polynomial approximations, respectively. Convergence of the Chebyshev method as measured by ${E_k}^{\text{exact}}$, E_k, and E_f for the random polymer with $N = 1000$ is shown in Figure 1(b). We observe again that E_k closely tracks ${E_k}^{\text{exact}}$, suggesting that E_k may be useful for estimating the accuracy of the Chebyshev approximation. The quantity E_f, on the other hand, does not appear monotonic although the exact error decreases monotonically.

Comparing Figures 1(a) and 1(b), we observe that the convergence rate of the Krylov subspace approximation is somewhat faster than that of the Chebyshev polynomial approximation.

2. Effect of block size on convergence

In Figure 2(a), the effect of block size on the convergence of the block-Krylov method is shown for the random polymer model with $N = 1000$. The convergence rate increases with block size as expected. The ${E_k}^{\text{exact}}$ and E_k estimates (for the first vector of the block of vectors) during the block-Krylov iteration are shown in Figure 2(b). We observe that E_k tracks ${E_k}^{\text{exact}}$, indicating that E_k would be again useful for checking convergence in practice.

Another effect of the block size is the improved computational performance when matrix-vector products are performed with a block of vectors simultaneously, compared to performing multiple matrix-vector products with single vectors. We will study this computational efficiency of the block-Krylov method in Sec. IV C.

3. Simulation model properties affecting convergence

Differences in convergence between the random and collapsed polymer models, and between $N = 200$ and 1000 are compared in Figure 3(a). Convergence is slower for larger N and for collapsed polymers. In Figure 3(b), the dependence of convergence rate on volume fraction for the monodisperse suspension model with $N = 200$ and 1000 are shown. Slower convergence for the higher volume fractions and the larger systems is observed. If convergence rates in the Krylov subspace methods are strongly dependent on N, the algorithmic

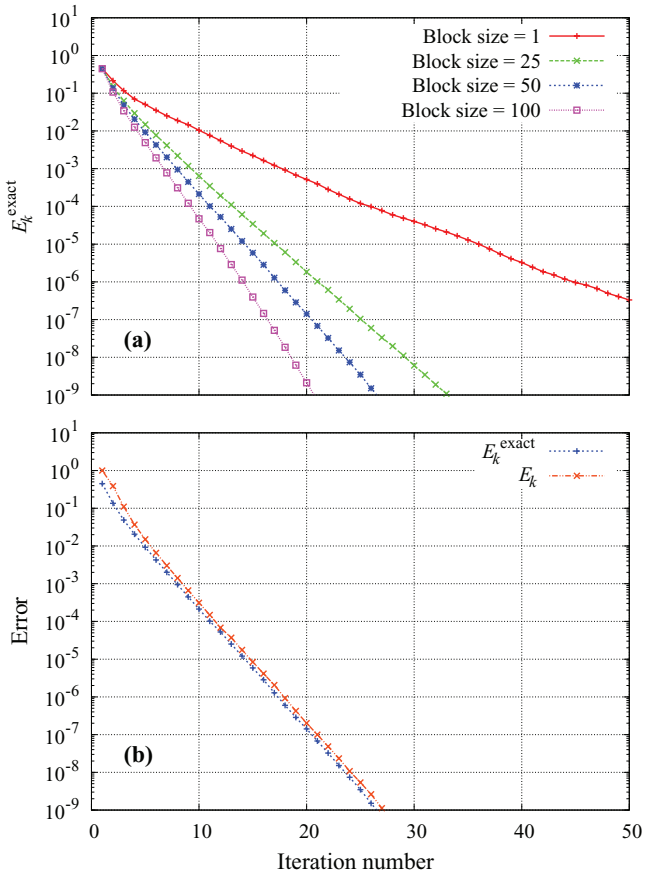

FIG. 2. (a) Effect of block size on convergence rate in the block Krylov subspace method. The errors are computed for the first vector of the block of vectors. Results are averages of the five random polymer chains with $N = 1000$. (b) Comparison of ${E_k}^{\text{exact}}$ and E_k in the block Krylov subspace method. Results for polymers with $N = 1000$ and block size = 50 are shown. All results are the average over the five independent configurations. Standard deviations for all data points are so small that they are not displayed. E_k with $k = 1$ was set to 1.

scaling is larger than $O(N^2)$. However, the convergence is insensitive to N in the Krylov methods when $E_k < \sim 0.01$, as seen later in Figure 7. Therefore, we might expect near $O(N^2)$ scaling for the Krylov methods if low accuracy of Brownian noise vectors are sufficient for BD simulations. We will discuss the accuracy of Brownian noise vectors and the scaling of the Krylov subspace methods in Secs. IV B and IV C.

B. BD simulations

In this section, we perform BD simulations with the three different models using Krylov subspace methods and check if dynamic properties of the model systems obtained from the simulations reproduce the results using the standard Cholesky factorization method. Statistical errors in the translational diffusion coefficients D_{cm} and D are less than 5% on average. We also evaluated the radius of gyration, R_g, as a static polymer property. Conclusions obtained from analysis of R_g are essentially the same as those for D_{cm}. Therefore, we show results only on D_{cm} for the polymer models in this text.

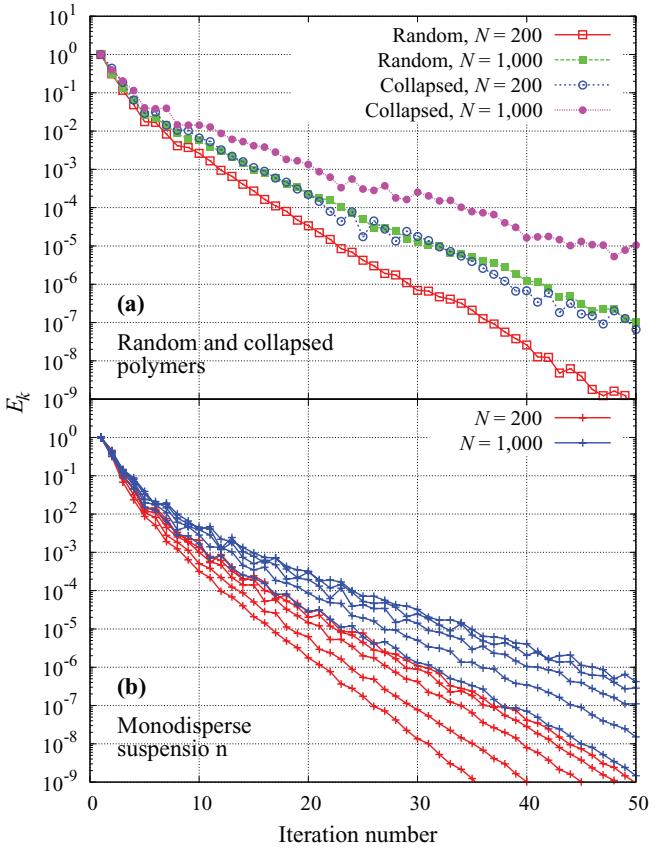

FIG. 3. Effect of model, number of particles, and volume fraction on convergence of the Krylov subspace method. (a) Convergence of the random and collapsed polymer models with $N = 200$ and 1000. (b) Convergence of the monodisperse suspension model at various volume fractions $\Phi = 0.1$, 0.2, 0.3, 0.4, and 0.5 with $N = 200$ and $N = 1000$. For monodisperse suspensions, convergence is slower for larger volume fractions and for larger numbers of particles.

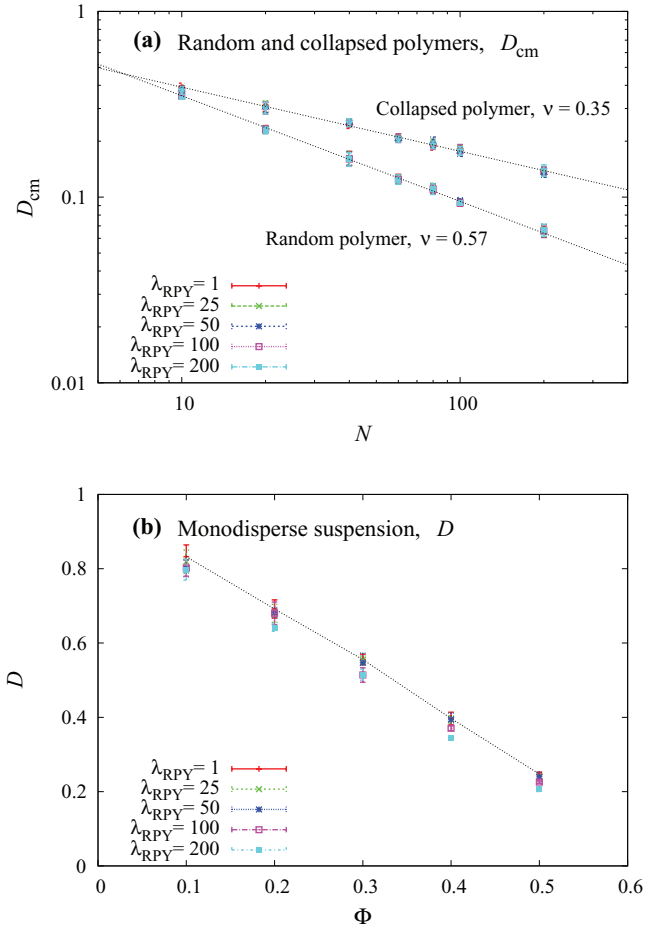

FIG. 4. Effect of λ_{RPY} on dynamic properties for the random and collapsed polymers, and monodisperse suspension model. (a) D_{cm} for the random and collapsed polymers with various polymer lengths N obtained from BD simulation with various λ_{RPY}. Lines are fit to the data for $\lambda_{RPY} = 1$ with $D_{cm} \propto N^{-\nu}$ and their exponents are shown. For both polymer models, results with different λ_{RPY} are so close that their plots overlap and are almost indistinguishable in the figure. (b) D for the monodisperse suspension model with number of particles of 200 at various volume fractions Φ obtained from BD simulations with various λ_{RPY}. The results with $\lambda_{RPY} = 1$ are connected by a broken line to guide the eye. The Cholesky factorization method was used in the BD simulations.

1. Update interval of diffusion tensor

In BD simulations, the diffusion matrix changes slowly, making it possible to use the same matrix for several time steps, significantly reducing computational cost. This also allows us to use block versions of Krylov subspace methods. In this section, we determine an appropriate update interval, λ_{RPY}, for the RPY diffusion matrix in BD simulations. We use the Cholesky factorization for computing correlated vectors in order to not confound these results with further approximations.

Figure 4(a) shows a log-log plot of D_{cm} as a function of length N for random polymers obtained from BD simulation using various values of λ_{RPY}. Theoretical scaling for this property is $D_{cm} \propto N^{-\nu}$. In a good solvent and in the presence of HI, as $N \to \infty$, the scaling exponent has a theoretical value of $\nu \approx 0.588$ from a perturbation analysis.[31] Our simulation of the random polymer model using $\lambda_{RPY} = 1$ gives $\nu = 0.57$ for D_{cm} in the presence of HI, in good agreement with the prediction. BD simulations with $\lambda_{RPY} = 25$–200 also provide D_{cm} of random polymers close to those obtained from the simulation with $\lambda_{RPY} = 1$ for all chain lengths examined in this study. For all conditions with $\lambda_{RPY} = 25$–200, the relative error is less than 2% for D_{cm}. Even if $\lambda_{RPY} = 800$ was used, the error was less than 4%. These results indicate that the random polymer model is quite insensitive to the update interval of the RPY tensor.

Effects of λ_{RPY} on D_{cm} for the collapsed chain model are also shown in Figure 4(a). This model corresponds to polymers in poor solvent, where the scaling exponent of $\nu = 0.33$ is predicted.[23] BD simulations with $\lambda_{RPY} = 1$ provide $\nu = 0.35$ for D_{cm}, which is also good agreement with theory. Like the above, the relative errors in D_{cm} are small for all λ_{RPY} conditions. The collapsed polymer model is thus also quite insensitive to the update interval of the RPY tensor.

The diffusion coefficients of particles, D, obtained from BD simulations in a monodisperse suspension at various volume fractions with various λ_{RPY} are shown in Figure 4(b). Errors in D obtained with various λ_{RPY} relative to values of D with $\lambda_{RPY} = 1$ are listed in Table I. In contrast to the single chain polymer models, D values are significantly affected by λ_{RPY} in the monodisperse suspension model. With λ_{RPY}

TABLE I. Errors (%) in D obtained with various diffusion matrix update intervals, λ_{RPY}, relative to results with $\lambda_{RPY} = 1$ for the monodisperse model at various volume fractions Φ with $N = 200$.

Φ	$\lambda_{RPY} = 25$	$\lambda_{RPY} = 50$	$\lambda_{RPY} = 100$	$\lambda_{RPY} = 200$	$\lambda_{RPY} = 400$	$\lambda_{RPY} = 800$
0.1	− 1.6	− 3.8	− 3.6	− 4.2	− 4.6	− 10.0
0.2	− 1.7	− 1.5	− 1.7	− 7.2	− 10.6	− 14.1
0.3	− 0.2	− 1.6	− 7.5	− 7.3	− 14.9	− 23.6
0.4	− 0.6	− 0.7	− 6.6	− 12.3	− 17.7	− 25.7
0.5	− 2.8	− 3.1	− 8.8	− 16.7	− 22.3	− 31.9
⟨|Error|⟩	1.4	2.1	5.6	9.7	14.0	21.1

= 200, the errors are more than 10% at volume fractions of 0.4 and 0.5, which exceeds the statistical error in our analysis. The errors in the results with $\lambda_{RPY} = 100$ are less than 10% for all volume fractions. However, at moderate to high volume fractions (Φ = 0.3–0.5), the errors are slightly higher than those at low volume fractions but are still greater than 5%. With $\lambda_{RPY} = 25$ and 50, the errors are less than 5% for all volume fractions. Compared with the polymer models, particles in the monodisperse suspension model can move easily due to the lack of harmonic constraints. In addition, attractive interactions between beads in the collapsed polymer model restrict the motions of beads. Therefore, the diffusion tensor of the monodisperse suspension model may change more rapidly than in the single chain polymer models.

2. Required accuracy of Brownian noise vectors

In this section, we study the accuracy of the correlated Brownian noise vectors required for accurate simulation results. The use of an appropriate level of accuracy is critical for obtaining maximum efficiency of approximate methods such as the Krylov subspace and Chebyshev methods. We used the Krylov subspace method to generate correlated Brownian noise vectors with accuracy controlled by values of E_k of 0.1, 0.01, and 0.001. The choice of $\lambda_{RPY} = 50$ was adopted which was shown in Sec. IV B 1 to give results comparable to those of $\lambda_{RPY} = 1$ for all models. We also performed BD simulations using the TEA method for comparison.

The accuracy of D_{cm} for the random polymer chain model obtained from BD simulations using the Krylov subspace method with various E_k as well as using the TEA method are listed in Table II. All values of E_k examined here for the Krylov subspace method matched the Cholesky results with relative errors in D_{cm} of less than 4%. The TEA method results are within the average error of 8%, consistent with the results reported by Geyer and Winter.[11] Values of R_g obtained from BD simulations with the Krylov subspace method and TEA were also close to the Cholesky results (data not shown).

Since the collapsed polymers are packed into compact structures, small errors in Brownian noise vectors may cause significant clashes between beads. Therefore, the model would be sensitive to the accuracy of correlated Brownian noise vectors. Errors in D_{cm} of the collapsed polymer chains obtained from the BD simulations using the Krylov method with various values of E_k as well as using the TEA method are listed in Table III. The results of the Krylov method with E_k of 0.001–0.1 have a relative error less than 5%. For TEA, errors in D_{cm} are about −16%, for all chain lengths. Geyer also reported the low estimation of D_{cm} by the TEA method for spherical objects consisting of many small particles.[14]

We also evaluated the relaxation time τ_{corr} for the autocorrelation function $\langle \mathbf{r}_{ee}(t) \cdot \mathbf{r}_{ee}(0) \rangle$ of the end-to-end vector $\mathbf{r}_{ee}$ for the polymer models, which may be sensitive to the accuracy of Brownian noise vectors. Although the values of relaxation times obtained from the BD simulations have large noise due to limited simulation length, the Krylov subspace method even with $E_k = 0.1$ gave close values to the Cholesky results, with no qualitative differences (data not shown). We

TABLE II. Errors (%) in D_{cm} obtained from the simulations using the Krylov method with various levels of Brownian noise accuracies and the TEA method relative to results using the Cholesky method with $\lambda_{RPY} = 1$ for various chain lengths for the random polymer model.

	Krylov with $\lambda_{RPY} = 50$			TEA with
N	$E_k = 0.001$	$E_k = 0.01$	$E_k = 0.1$	$\lambda_{RPY} = 50$
10	1.3	1.3	1.0	− 5.4
20	− 0.8	− 0.8	− 1.2	− 8.8
40	− 2.7	− 3.4	− 3.8	− 9.8
60	− 0.4	− 0.2	− 1.3	− 7.5
80	− 1.1	− 1.2	− 2.2	− 8.5
100	1.3	1.0	− 0.5	− 5.7
200	− 2.0	− 1.1	− 3.9	− 8.8
⟨|Error|⟩	1.4	1.3	2.0	7.8

TABLE III. Errors (%) in D_{cm} at equilibrated states obtained from the simulations using the Krylov method with various Brownian noise accuracies and the TEA method relative to results using the Cholesky method with $\lambda_{RPY} = 1$ for various chain lengths for the collapsed polymer model.

	Krylov with $\lambda_{RPY} = 50$			TEA with
N	$E_k = 0.001$	$E_k = 0.01$	$E_k = 0.1$	$\lambda_{RPY} = 50$
10	− 2.1	− 2.4	− 3.0	− 11.7
20	− 2.8	− 3.1	− 4.9	− 16.1
40	− 1.5	− 1.8	− 4.2	− 16.2
60	− 1.0	− 1.9	− 1.9	− 16.0
80	− 0.2	− 1.3	− 1.5	− 16.6
100	− 2.1	− 1.5	− 3.2	− 16.8
200	0.2	2.5	− 2.0	− 15.6
⟨|Error|⟩	1.4	2.1	2.9	15.6

also studied polymers modeled using a finite extensible nonlinear elastic potential with a soft-core repulsive potential function. We performed the same analysis as above and obtained the same conclusions (data not shown).

Diffusion coefficients of particles in the monodisperse suspension system at various volume fractions obtained from the BD simulation using the Krylov subspace method with various E_k as well as using the TEA method are listed in Table IV. For the monodisperse suspension model, diffusivities of particles obtained from the Krylov subspace method with three different E_k are within a 5% error for all volume fractions. On the other hand, results at high volume fractions obtained by the TEA method significantly deviate from the Cholesky results (27%–50%). This defect in TEA is not surprising since an assumption in the TEA method is that the hydrodynamic coupling is weak; at low volume fractions, this assumption is correct, but at high volume fractions, the average distances between particles become small, resulting in strong hydrodynamic coupling.

TABLE IV. Errors (%) in D obtained from the simulations using the Krylov method with various Brownian noise accuracies and the TEA method relative to results using the Cholesky method with $\lambda_{\mathrm{RPY}} = 1$ for monodisperse model at various volume fractions Φ with number of particles $N = 200$.

	Krylov with $\lambda_{\mathrm{RPY}} = 50$			TEA with
Φ	$\mathrm{E_k} = 0.001$	$\mathrm{E_k} = 0.01$	$\mathrm{E_k} = 0.1$	$\lambda_{\mathrm{RPY}} = 50$
0.1	− 1.1	− 3.0	− 2.3	− 4.4
0.2	0.8	− 2.3	− 2.2	− 5.4
0.3	0.0	− 1.9	− 3.0	− 14.2
0.4	1.5	− 3.1	− 1.6	− 27.3
0.5	− 4.7	− 4.4	− 3.4	− 51.8
⟨|Error|⟩	1.6	3.0	2.5	20.6

In this section, we estimated the required accuracy of Brownian noise vectors in the Krylov subspace method. Results for the three different models show that a value of E_k of 0.1 would be practically adequate and a value of E_k of 0.01

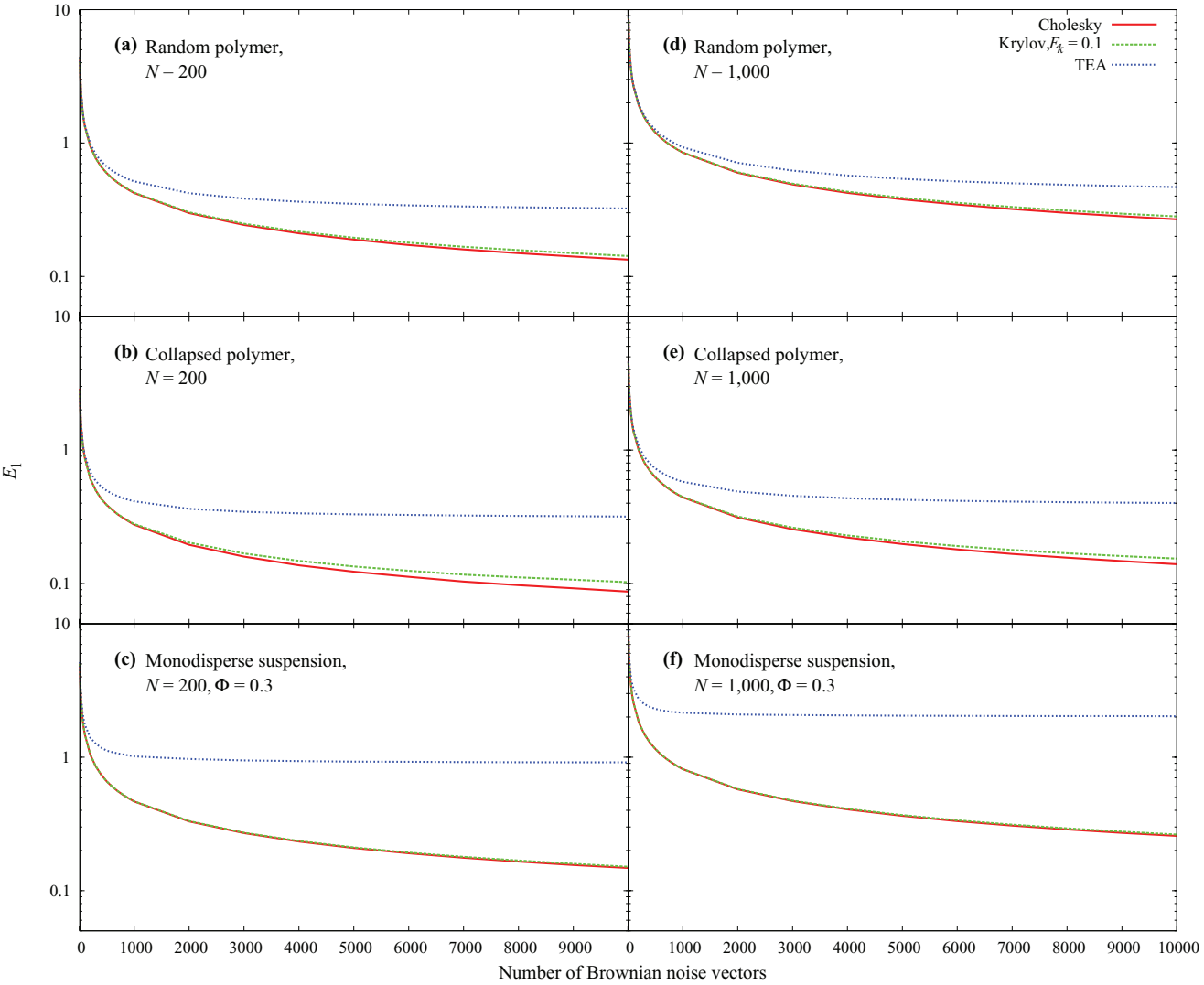

FIG. 5. Values of E_1 for the covariance matrices constructed from sets of Brownian noise vectors generated by the Cholesky, Krylov subspace with $E_k = 0.1$, and TEA methods. Results of the random polymer chain model (a, d), the collapsed polymer model (b, e), and the monodisperse suspension model at volume fraction of 0.3 (c, f) are shown. Left subfigures (a, b, c) are for $N = 200$ and right subfigures (d, e, f) are for $N = 1000$. For the monodisperse suspension model, results of the Cholesky (red lines) and Krylov subspace with $E_k = 0.1$ (green lines) are so close that their lines overlap and are indistinguishable in the figure. All results are the average over five independent configurations. Standard deviations for all data points are so small that they are not displayed.

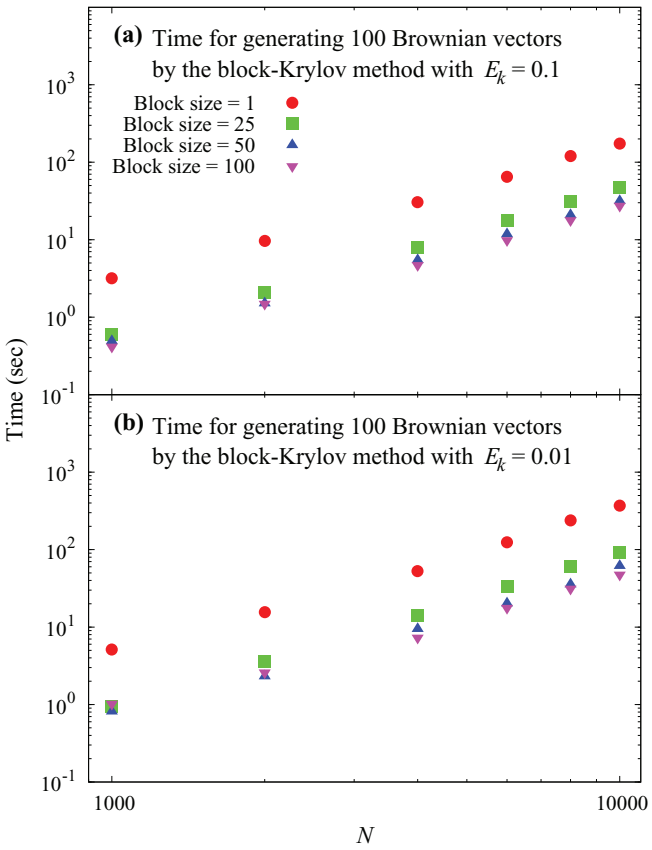

FIG. 6. Computational time required for generating 100 correlated Brownian noise vectors by the block-Krylov subspace method with different block sizes. Timings for (a) $E_k = 0.1$ and (b) $E_k = 0.01$ are shown. The random polymer model with length $N = 1000$–$10\,000$ was used for this timing test. For block size = 1, the DSYMV BLAS routine was employed for matrix-vector multiplications. For other block sizes, the DSYMM BLAS routine was employed for matrix-matrix multiplications. The latter routine is not optimized for matrix-vector multiplications.

would be sufficient to reproduce the Cholesky results within statistical error. In this study, a λ_{RPY} of 50 was used for all models, where the relative errors in diffusivities of all three models are less than 5%. However, as shown in Figure 4 and Table I, much larger values of λ_{RPY} could be used for polymers at dilute solution in good and poor solvents, resulting in great saving in computational time without a significant loss of simulation accuracy.

3. Comparison of covariance matrix generated from Brownian noise vectors

In this section, we seek to understand why low levels of accuracy can be used in Krylov subspace and Chebyshev polynomial approximations yet computed model properties from a BD simulation are essentially unaffected.

Given a set of correlated vectors $\tilde{\mathbf{Y}}^X$ generated by a method X, its average $\langle \tilde{\mathbf{Y}}^X \rangle$ should be ~ 0 and a covariance matrix $\mathbf{C}^X$ constructed from $\tilde{\mathbf{Y}}^X$ should be close to $\mathbf{D}$,

$$\mathrm{cov}(\tilde{\mathbf{Y}}^X, \tilde{\mathbf{Y}}^X) = \mathbf{C}^X \approx \mathbf{D}. \tag{30}$$

The difference between $\mathbf{C}^X$ and $\mathbf{D}$ can be quantified by the relative error

$$E_1^X = \frac{\|\mathbf{C}^X - \mathbf{D}\|_F}{\|\mathbf{D}\|_F}. \tag{31}$$

Here, $\|\cdot\|_F$ is the Frobenius norm, which is defined to be the square root of the sum of the squares of the entries of a matrix. In this analysis, an identical set of uncorrelated Gaussian noise vectors is used for all methods as input. The relative error when the Cholesky factorization is used, ${E_1}^{\mathrm{Cholesky}}$, is the lowest value of E_1 that can be expected for a given number of correlated vectors. The values of E_1 for the Cholesky, Krylov subspace with $E_k = 0.1$, and TEA methods as a function of the number of correlated vectors up to 10 000 are shown in Figure 5. In addition, we observed that ${E_1}^{\mathrm{Krylov}}$ with $E_k = 0.01$ and 0.001 always track ${E_1}^{\mathrm{Cholesky}}$ for all models with $N = 200$ and 1000 and they are indistinguishable from each other for any number of input noise vectors (data not shown). Values of ${E_1}^{\mathrm{Krylov}}$ with $E_k = 0.1$ slightly deviate from ${E_1}^{\mathrm{Cholesky}}$ for the collapsed polymer model, although the differences are smaller than 0.01. The average $\langle \tilde{\mathbf{Y}}^X \rangle$ for all cases is close to zero (data not shown). We also did the same analysis for the random polymer $N = 10$ with up to 10^7 input noise vectors for $E_k = 0.4$, 0.1, and 0.01 to see converged E_1 values, showing that E_1 converged to 0.08, 0.02, and 0.003, respectively, which are much smaller than their E_k values in input vectors. These results seem to suggest that using $E_k = 0.01$ in the Krylov subspace methods would be sufficient to generate correlated Brownian noise vectors whose accuracy is indistinguishable from that of the Cholesky method. Even for $E_k = 0.1$ in the Krylov subspace methods, the error in the generated covariance matrix would have about 1%, which would be also sufficient for BD simulations. For Chebyshev, results similar to the Krylov methods are observed (data not shown).

The values of E_1 for TEA are always larger than those of the Cholesky method for the polymer and monodisperse suspension models. These results indicate that each correlated vector calculated by TEA has small errors and these errors accumulate, resulting in significant deviation from the exact covariance matrix. Geyer and Winter,[11] and Schmidt *et al.*[13] reported that the TEA method could reproduce Cholesky results for random polymers in a good solvent and a theta solvent. The polymers in these conditions are not closely packed. Therefore, since the models may be insensitive to noise in the correlated Brownian vectors, the TEA method might reproduce the Cholesky results for these models.

C. Computational time

Finally, we evaluate the computational efficiency of the Krylov subspace methods. For the following timing tests, the random polymer model with $N = 1000$–$10\,000$ was used. Values of E_k of 0.1 and 0.01 were used for the stopping criterion. Cholesky, Chebyshev, and TEA methods are also examined for comparison. These tests were performed on a quad-core AMD Opteron processor and the algorithms were parallelized by hand. The GOTO BLAS library[32] was used

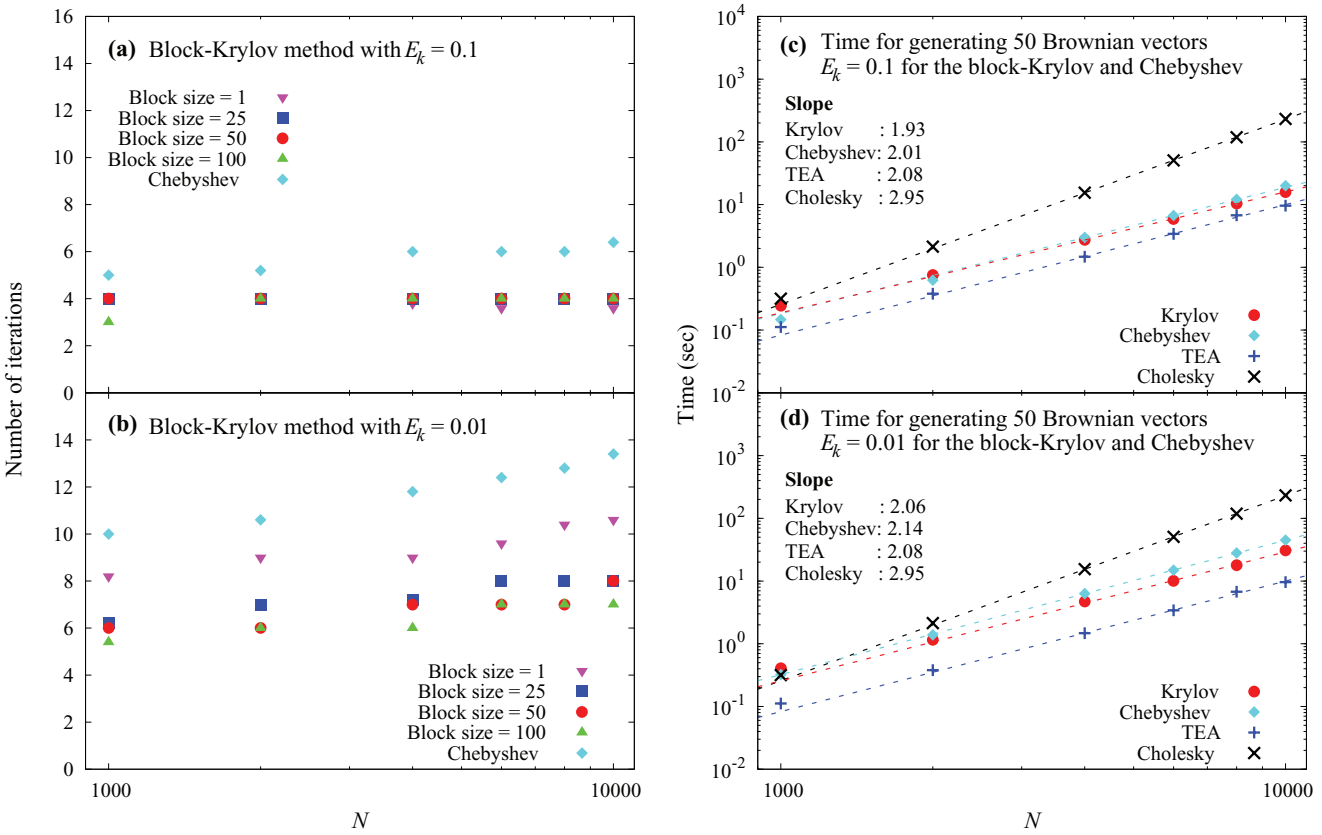

FIG. 7. Scaling of number of iterations and computational time of the block-Krylov subspace method with the number of particles. The random polymer model with length $N = 1000$–$10\,000$ was used for this timing test. Number of iterations required with thresholds (a) $E_k = 0.1$ and (b) $E_k = 0.01$ for the block-Krylov subspace and Cholesky methods. Computational time required for generating 50 correlated Brownian noise vectors by the block-Krylov subspace and Cholesky methods with (c) $E_k = 0.1$ and (d) $E_k = 0.01$. A block size of 50 was used for the block-Krylov subspace method. Results for the Cholesky and TEA methods are also shown for comparison (also computed in block fashion). Dashed lines are fitted linear slopes for a range of $N = 4000$–$10\,000$. The values of slopes for these methods are shown in inside the figures.

for matrix factorization and matrix-matrix and matrix-vector multiplications in the algorithms. It is important to note that the time for estimating eigenvalues is not included in the Chebyshev results.

For the block-Krylov subspace method, we expect good performance of the algorithm due to enhanced convergence rate and computational efficiency. The effect of block size in the block-Krylov subspace method on computational time is shown in Figure 6. At $N = 10\,000$, the time for generating 100 Brownian noise vectors simultaneously, i.e., a block size of 100, with $E_k = 0.1$ and 0.01 is much lower than that for separately generating 100 vectors by factors of 6.4 and 7.8, respectively.

The number of iterations required for convergence below pre-defined accuracy thresholds E_k of 0.1 and 0.01 in the block-Krylov and Chebyshev methods are shown in Figures 7(a) and 7(b). Even at $N = 10\,000$, both methods converge (with $E_k = 0.01$) within 14 iterations. Comparing different block sizes in the Krylov subspace method, using larger block sizes accelerates the convergence rate in the case of $E_k = 0.01$. When $E_k = 0.1$ is used, this effect of block size on convergence rate is not observed since only a very small number of iterations is necessary for convergence. Comparing the Krylov and Chebyshev methods, the number of iterations required for the former method is less than that for the latter. In addition, the block-Krylov subspace method appears more insensitive to N than the Chebyshev method.

In Figures 7(c) and 7(d), the computational time required for generating 50 correlated Brownian noise vectors by various methods is shown. The block-Krylov method scales very nearly as $O(N^2)$ over the range of N tested, with both values of E_k tested. The Chebyshev method also scales very nearly as $O(N^2)$, again when E_k of 0.1 and 0.01 are used. Such scaling implies that the time is dominated by the cost of matrix-vector multiplications by $\mathbf{D}$ (as it should be for large N), and that the number of iterations is essentially insensitive to N. This latter fact holds for the values of N and E_k we tested; we do expect the number of iterations to grow more noticeably with N when the stopping tolerance is more stringent.

The performance of the block-Krylov method is 1.2 and 1.5 times better than the Chebyshev method (also computed using matrix multiplications with a block of vectors simultaneously) at large N, with E_k of 0.1 and 0.01, respectively. The main reason for this improvement is the reduced number of iterations required by the block-Krylov method. For the Chebyshev method, additional computational cost is required to estimate the extreme eigenvalues. Jendrejack *et al.* proposed a method where the eigenvalues are updated dynamically using the Arnoldi method in $O(N^2)$ operations,[33] according to measures of the accuracy of the Chebyshev approximation.[8]

This cost may be amortized over several time steps. Kroger *et al.* used twice the maximum and half the minimum eigenvalues coming from a pre-averaged HI tensor as upper and lower bounds, respectively, in the Chebyshev method for entire simulations.[34] We took eigenvalues averaged over five configurations instead of the values of the pre-averaged HI tensor in Kroger's idea. For this case, additional 19% and 25% computational costs were required on average over $N = 1000$–$10\,000$ for $E_k = 0.1$ and 0.01, respectively, due to slow convergence caused by use of the wider spectrum range (data not shown).

Compared to the Cholesky method, which scales as $O(N^3)$, the block-Krylov method with E_k of 0.1 and 0.01 outperforms it by factors of 13 and 7, respectively, at large N. When it is applicable, the TEA method shows the best performance among the four methods examined here, which scales as $O(N^2)$ with a small constant, as observed in other reports.[11,13]

V. CONCLUSIONS

This paper has studied a class of methods based on Krylov subspaces for computing correlated Brownian noise vectors in BD simulations with HI. The existing methods that have been used for this purpose in the past are Cholesky factorizations, Chebyshev polynomial approximations, and the TEA method. For small numbers of particles, the Cholesky factorization method is most efficient. For large numbers of particles, the main alternative in the past has been Chebyshev approximations. The Krylov subspace methods studied here also have their niche in large-scale problems. Indeed, Krylov subspace methods are also polynomial approximations and have similar computational cost as Chebyshev approximations for polynomials of the same degree. Krylov subspace methods, however, have the potential to converge faster than Chebyshev approximations, and thus require lower degree approximations, especially for large problems or when high accuracy is required. This was observed experimentally (see Figures 1, 6, 7(a), and 7(b)). From the view point of memory usage, the Krylov subspace methods as well as the Chebyshev method require half of the memory size of the Cholesky method, which is essentially just for the diffusion matrix. This also helps for large-scale BD simulations.

There are, however, two much more important advantages of Krylov subspace methods compared to Chebyshev approximations. First, Krylov subspace methods do not require estimates of the extreme eigenvalues of **D**, making them very easy to use. In contrast, Chebyshev approximations must intermittently update these estimates potentially at high cost[8] or suffer degraded convergence rates when conservative eigenvalue bounds are estimated *a priori* and then used throughout the simulation. Overall, simulations with Chebyshev approximations may lead to a longer time-to-solution than simulations with Krylov subspace approximations.

The second major advantage of Krylov subspace methods over Chebyshev approximations arises in the usual case when **D** changes sufficiently slowly and it is possible to compute Brownian noise vectors for several time steps using the same **D**. In all methods, it is much more computationally efficient to compute all vectors simultaneously (i.e., using products of a matrix with a block of vectors) than to compute the vectors individually. However, Krylov subspace methods with a block of vectors can be reformulated so that each solution can exploit the Krylov subspace associated with other vectors. The result is faster convergence compared to the single-vector case without a significant increase in cost. This was observed experimentally (see Secs. IV A and IV C). We thus expect that block versions of Krylov subspace methods will become very useful for large or ill-conditioned **D** (e.g., large volume fractions), where a large Krylov subspace dimension is required. In this paper, we have studied the error in macroscopic quantities as a function of the update interval of **D**, and thus the block size. The results are that surprisingly large update intervals (e.g., tens to hundreds of time steps) can be tolerated with little impact on the computed macroscopic quantities. Such large update intervals can make block-Krylov subspace methods very effective.

In this paper, we have also studied how the accuracy of the Brownian noise vector affects the accuracy of computed macroscopic quantities. We found that only low levels of accuracy are required to match the diffusion rate D_{cm} and radius of gyration R_{g} as computed by simulations using the full accuracy Cholesky factorization. These levels of accuracy are lower than what has been proposed in the past.[8] Such low levels of accuracy reduce the effective cost of the approximate methods and make them more competitive with Cholesky factorization on smaller problems. One reason why such low levels of accuracy in the Krylov subspace methods are acceptable is that quality of the noise vectors generated by this method and the Cholesky factorization might be indistinguishable even at these levels of accuracy from the point of view of the effective covariance matrix (see Sec. IV B 3 and Figure 5). We believe that the same low levels of accuracy in Brownian noise vectors would be sufficient also for much larger problems.

Our study of simulation accuracy also included the TEA method. This method is very inexpensive, with a cost comparable to a polynomial approximation of degree 3. TEA is a fixed approximation, without adjustable accuracy. For model systems where the hydrodynamic interaction is weak (low volume fraction particle suspensions and polymers in a good solvent), TEA is able to accurately compute macroscopic quantities. For other systems, as shown in our results, and potentially for large systems, results from TEA are much less accurate than results from Chebyshev and Krylov methods.

Concerning the computational efficiency for large systems, the computational time for Krylov subspace methods scales very nearly as $O(N^2)$ for values of N up to 10 000 (which was the limit we tested) using sufficient levels of accuracy for simulation purposes. This is in contrast to reported computational time scaling of $O(N^{2.5})$ (Ref. 11) and higher[13] for entire simulations using the Chebyshev method when eigenvalue estimates are computed adaptively based on monitoring E_f. We note that a value for E_f of 10^{-3} corresponds to E_k of 10^{-5}–10^{-4} for the example shown in Figure 1. We believe this level of accuracy is not necessary for producing accurate simulation results.

For large systems, the computational time scaling can be further reduced to $O(N \log N)$ or $O(N)$ by replacing the matrix-vector multiplications by fast approximations such as particle-mesh Ewald,[35] generalized Ewald,[36] and potentially fast multipole methods. These methods also avoid the $O(N^2)$ cost of forming and storing **D**, which may in some cases be unavailable in explicit form. The Krylov subspace and Chebyshev approaches are especially important and useful when combined with the above fast methods.

In conclusion, HI play important roles in the dynamics of a given system, whose effects are well-studied in the fields of colloids and random polymers.[37,38] On the other hand, the understanding of their role in biological reactions is limited.[2,9,24–26] The main reason for this is the high complexity of biological systems. As is often the case, computational approaches that can simulate large systems for long time scales are very desirable. The Krylov subspace method is a simpler alternative to Chebyshev polynomial approximations that can help carry out very large-scale simulations with HI.

ACKNOWLEDGMENTS

This research was supported in part by National of Institutes of Health (NIH) Grant No. GM-37408 of the Division of General Medical Sciences of the National Institutes of Health. Dr. Saad was supported by National Science Foundation (NSF) Grant No. NSF/DMS-0810938.

[1] D. L. Ermak and J. A. McCammon, J. Chem. Phys. **69**, 1352 (1978).
[2] T. Ando and J. Skolnick, Proc. Natl. Acad. Sci. U.S.A. **107**, 18457 (2010).
[3] T. Ishikawa, G. Sekiya, Y. Imai, and T. Yamaguchi, Biophys. J. **93**, 2217 (2007).
[4] J. Kotar *et al.*, Proc. Natl. Acad. Sci. U.S.A. **107**, 7669 (2010).
[5] J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[6] H. Yamakawa, J. Chem. Phys. **53**, 436 (1970).
[7] M. Fixman, Macromolecules **19**, 1204 (1986).
[8] R. M. Jendrejack, M. D. Graham, and J. J. de Pablo, J. Chem. Phys. **113**, 2894 (2000).
[9] P. Mereghetti, D. Kokh, J. A. McCammon, and R. C. Wade, BMC Biophys. **4**, 2 (2011).
[10] T. Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide* (Springer, New York, 2002).
[11] T. Geyer and U. Winter, J. Chem. Phys. **130**, 114905 (2009).
[12] A. H. Elcock, Curr. Opin. Struct. Biol. **20**, 196 (2010).
[13] R. R. Schmidt, J. G. H. Cifre, and J. G. de la Torre, J. Chem. Phys. **135**, 084116 (2011).
[14] T. Geyer, BMC Biophys. **4**, 7 (2011).
[15] E. Gallopoulos and Y. Saad, SIAM J. Sci. Stat. Comput. **13**, 1236 (1992).
[16] L. A. Knizhnerman, Comput. Math. Math. Phys. **31**, 1 (1991).
[17] Y. Saad, SIAM J. Numer. Anal. **29**, 209 (1992).
[18] S. Kaniel, Math. Comput. **20**, 369 (1966).
[19] Y. Saad, SIAM J. Numer. Anal. **17**, 687 (1980).
[20] X. Liu, E. Chow, K. Vaidyanathan, and M. Smelyanskiy, in *Proceedings of the 26th IEEE International Parallel and Distributed Processing Symposium* (IEEE, Shanghai, 2012), p. 12.
[21] P. E. Rouse, J. Chem. Phys. **21**, 1272 (1953).
[22] B. H. Zimm, J. Chem. Phys. **24**, 269 (1956).
[23] K. A. Dill and S. Bromberg, *Molecular Driving Forces: Statistical Thermodynamics in Chemistry and Biology* (Garland Science, New York, 2003).
[24] T. Frembgen-Kesner and A. H. Elcock, J. Chem. Theory Comput. **5**, 242 (2009).
[25] T. Frembgen-Kesner and A. H. Elcock, Biophys. J. **99**, L75 (2010).
[26] J. Antosiewicz and J. A. McCammon, Biophys. J. **69**, 57 (1995).
[27] J. F. Brady and G. Bossis, Annu. Rev. Fluid Mech. **20**, 111 (1988).
[28] D. J. Jeffrey, and Y. Onishi, J. Fluid Mech. **139**, 261 (1984).
[29] C. W. J. Beenakker, J. Chem. Phys. **85**, 1581 (1986).
[30] T. Zhou and S. B. Chen, J. Chem. Phys. **124**, 034904 (2006).
[31] B. Li, N. Madras, and A. D. Sokal, J. Stat. Phys. **80**, 661 (1995).
[32] See http://www.tacc.utexas.edu/tacc-projects/gotoblas2 for information about the GOTO BLAS library.
[33] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* (SIAM, 1998).
[34] M. Kroger, A. Alba-Perez, M. Laso, and H. C. Ottinger, J. Chem. Phys. **113**, 4767 (2000).
[35] A. J. Banchio and J. F. Brady, J. Chem. Phys. **118**, 10323 (2003).
[36] J. P. Hernandez-Ortiz, J. J. de Pablo, and M. D. Graham, Phys. Rev. Lett. **98**, 140602 (2007).
[37] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics* (Oxford University Press, Oxford, 1988).
[38] W. B. Russel, D. A. Saville, and W. R. Schowalter, *Colloidal Dispersions* (Cambridge University Press, Cambridge, 1989).