Efficient Brownian dynamics simulation of DNA molecules with hydrodynamic interactions in linear flows

Szu-Pei Fu,* Y.-N. Young,† and Shidong Jiang‡
Department of Mathematical Sciences and Center for Applied Mathematics and Statistics, New Jersey Institute of Technology, Newark, New Jersey 07102, USA
(Received 28 September 2014; published 17 June 2015)

The coarse-grained molecular dynamics (MD) or Brownian dynamics (BD) simulation is a particle-based approach that has been applied to a wide range of biological problems that involve interactions with surrounding fluid molecules or the so-called hydrodynamic interactions (HIs). In this paper, an efficient algorithm is proposed to simulate the motion of a single DNA molecule in linear flows. The algorithm utilizes the integrating factor to cope with the effect of the linear flow of the surrounding fluid and applies the Metropolis method (MM) by Bou-Rabee, Donev, and Vanden-Eijnden [Multiscale Model. Simul. **12**, 781 (2014)] to achieve more efficient BD simulation. Thus our method permits much larger time step size than previous methods while still maintaining the stability of the BD simulation, which is advantageous for long-time BD simulation. Our numerical results on λ-DNA agree very well with both experimental data and previous simulation results. Finally, when combined with fast algorithms such as the fast multipole method which has nearly optimal complexity in the total number of beads, the resulting method is parallelizable, scalable to large systems, and stable for large time step size, thus making the long-time large-scale BD simulation within practical reach. This will be useful for the study of membranes, long-chain molecules, and a large collection of molecules in the fluids.

DOI: 10.1103/PhysRevE.91.063008 PACS number(s): 47.27.eb, 05.10.−a, 87.15.A−

I. INTRODUCTION

The dynamics of a single DNA or polymer macromolecule in fluid flow has been extensively investigated experimentally ([1,2] and references therein), theoretically [3–5], and numerically [6,7]. Bulk rheological experiments such as flow birefringence and light scattering measurements give inference of polymer conformation, orientation, and chain stretch in fluid flows. The advent of single molecule visualizations using fluorescence microscopy allows for the direct observation of complex dynamics of individual macromolecules in dilute solutions under shear, extensional, and general two-dimensional mixed flows [2,8–11]. These measurements provide data for direct comparison against fully parametrized models of macromolecules, such as the bead-spring model for DNA with finite extensibility, excluded volume (EV) [12] effects, and hydrodynamic interactions (HI) [7]. Brownian dynamics (BD) simulations of bead-spring and bead-rod models with free-draining assumption (no hydrodynamic interactions) give quantitative agreement with short chains of double stranded DNA experiments, for example, ∼21 μm long λ-DNA [13–15]. However, for longer chains of DNA, HI needs to be included for quantitative agreement. For truly flexible polymers such as single stranded DNA or synthetic polymers, one can expect that HI will be important even for short chains.

Following Ermak and McCammon [4], Schroeder *et al.* modeled the DNA macromolecule as a system of N particles subject to interparticle forces, fluctuating HI and EV forces [7,13,14]. They designed a semi-implicit predictor-corrector scheme for simulating the Brownian system, and illustrated how effects of HI and EV between monomers in a flexible polymer chain influence both the equilibrium and non-equilibrium physical properties of DNA macromolecules [7], consistent with the experimental observations. The non-local HI between the DNA macromolecule and the surrounding fluid involves an integral of hydrodynamic forces between a point and the rest of the macromolecule. Within the coarse-grained framework, this integral is equivalent to a sum of all hydrodynamic forces between a bead and the rest of the system. Here we adopt the Rotne-Prager-Yamakawa (RPY) tensor [16] (i.e., the mobility tensor) for HI effects:

$$D_{ij} = \frac{k_B T}{\zeta_{\text{res}}} I_{ij}, \quad \text{if} \quad i = j, \tag{1}$$

$$D_{ij} = \frac{k_B T}{8\pi\eta r_{ij}}\left[\left(1+\frac{2a^2}{3r_{ij}^2}\right)I_{ij} + \left(1-\frac{2a^2}{r_{ij}^2}\right)\frac{\mathbf{r}_{ij}\mathbf{r}_{ij}}{r_{ij}^2}\right], \quad \text{if} \quad i \neq j, \quad r_{ij} \geqslant 2a, \tag{2}$$

$$D_{ij} = \frac{k_B T}{\zeta_{\text{res}}}\left[\left(1-\frac{9r_{ij}}{32a}\right)I_{ij} + \frac{3\mathbf{r}_{ij}\mathbf{r}_{ij}}{32ar_{ij}}\right], \quad \text{if} \quad i \neq j, \quad r_{ij} < 2a, \tag{3}$$

where D_{ij} is the mobility of bead i due to bead j in three dimensions, I_{ij} the 3×3 identity matrix, and $\zeta_{\text{res}} = 6\pi\eta a$ is the bead resistivity with η the solvent viscosity and a the radius of beads.

There are two main challenges for the long-time large-scale BD simulations with HI and EV effects. First, the correlated random noises in the change of displacement vectors at each time step are proportional to $\sqrt{\Delta t}$ with Δt the time step size. This makes the design of high-order marching scheme very difficult and forces very small Δt for many explicit or semi-implicit numerical schemes in order to avoid the numerical instability. The problem becomes much more severe for long-time BD simulations since it then requires

*sf47@njit.edu
†yyoung@njit.edu
‡shidong.jiang@njit.edu

a very large total number of time steps for the system to reach the desired state, which very often leads to weeks of simulation time even for one run.

Second, the direct evaluation of the particle interaction at each time step requires $O(N^2)$ operations where N is the total number of particles in the system; and the generation of the correlated random displacements requires $O(N^3)$ operations if the standard Choleski factorization is used or $O(KN^2)$ if the Chebyshev spectral approximation is used for computing the product of the matrix square root and an arbitrary vector (here K is the condition number of the covariance matrix) (see, for example, [17]).

To summarize, in order to efficiently utilize the BD simulation as a practical tool to study the properties of large systems, say, many polymers or a large collection of DNA molecules in a fluid, it is essential to address the following two questions: how to numerically integrate the system with greater accuracy and better stability property which enables much large time step size? How to expedite the calculations of long-range particle interactions and associated correlated random effects in BD simulations with HI, especially for large N?

For BD simulations near equilibrium, a Metropolis scheme for the temporal integration has been recently proposed [18,19] for a Markov process whose generator is self-adjoint (with respect to a density function) to expedite simulations to reach equilibrium in a timely fashion. Under this scheme, stable and accurate BD simulations of DNA in a solvent are obtained using time step sizes that are orders of magnitude larger than those for predictor-corrector schemes [6,7,14]. However, such a Metropolis scheme relies heavily on the self-adjointness of the Markov process generator for a quiescent flow.

In this work, we present an efficient algorithm for the simulations of the dynamics of DNA macromolecules under linear flows. Our method is based upon the Metropolis scheme developed in [19] for self-adjoint diffusions, which is applicable for the study of the DNA molecule to its equilibrium configurations in a quiescent flow. When a linear flow such as an extensional or a shear flow is present in the surrounding fluid, the diffusion process is not self-adjoint anymore. We first apply the method of integrating factors to recast the associated system of stochastic differential equations (SDE) into a form such that the effect of the linear flow is taken into account by the integrating factor. We then modify the Metropolis scheme in [19] to update the displacements of beads which are the coarse-grained representation of the long chain DNA molecule. Our numerical experiments show that our scheme allows much greater time step size in the BD simulation and avoids the numerical instability. The numerical results on the study of λ-DNA agree very well with the experimental data [2,11] and previous simulation results [7]. Moreover, the total simulation time is significantly reduced in our methods as compared with the semi-implicit predictor-corrector scheme [7].

For BD simulations that involve a "large" number of interacting particles (so large that the calculation of their mutual interactions becomes the computational bottleneck), recent work in [17,20] reduces the computational cost of particle interactions from $O(N^2)$ to $O(N)$ and the cost of generating the correlated random displacements from $O(N^3)$ or $O(KN^2)$ to $O(KN)$. These works yield an essentially linear algorithm with respect to the total number of particles in the BD simulation of interacting particles. The method developed in [17,20] extends the original fast multipole method (FMM) [21] to the case of the RPY tensor and combines it with the spectral Lanczos decomposition method (SLDM) to generate correlated random vectors whose correlation is determined by the RPY tensor. To demonstrate that long-time large-scale BD simulations (with or without linear flows) for large systems of interacting particles are within practical reach when our modified Metropolis scheme is combined with the fast method in [17,20], we use two examples to illustrate that our algorithms do efficiently capture the HI effects in a large BD system when compared with experimental results: One is the hysteretic extension of a long DNA molecule in a linear extensional flow, and the other is the multiple DNA molecules in an oscillatory shear flow.

This paper is organized as follows. In Sec. II the formulation for the BD simulation is presented along with a discussion on the relevant physical parameters and forces. Section III provides a detailed description of the numerical method used in this paper. In Sec. IV we demonstrate the performance of our numerical scheme by comparing our numerical results with the experimental data [2,11] and previous simulation results [7] where the motion of a single DNA molecule in a quiescent, extensional, or shear flow is studied and the DNA molecule is modeled via 29 beads. In Sec. V we briefly discuss the extension of our method to the study of large systems by combining it with the FMM for the RPY tensor and other fast algorithms. Finally Sec. VI contains a short conclusion and discussion for future work.

II. BROWNIAN DYNAMIC SIMULATION OF A DNA MOLECULE WITH HI

The DNA or polymer macromolecule is coarse-grained into a system of N beads described by the Langevin equation [4] with hydrodynamic interactions. The governing equation for the position vector $\mathbf{r}_i$ of the ith bead is

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \sum_j \zeta_{ij} \cdot \left(\mathbf{v}_j - \frac{d\mathbf{r}_j}{dt} \right) + \mathbf{F}_i + \sqrt{2} \sum_j \sigma_{ij} \cdot W_j, \quad (4)$$

where m_i is the mass of bead i, $\mathbf{v}_j$ is the solvent velocity, and ζ_{ij} is the friction coefficient tensor. The coefficient matrix σ connects the thermal fluctuations of the particles through hydrodynamic interactions. In the Ermak-McCammon model [4], it is related to ζ with $\zeta = \sigma^\top \sigma / k_B T$, where $k_B T$ is the thermal energy. W_j is the thermal fluctuation modeled as a Wiener process with mean 0 and variance dt. Thus, the RHS of Eq. (4) is the total force acting on the bead i including the drag force, total inter-particle force and the thermal fluctuating HI.

Ignoring the bead inertia, Eq. (4) can be written as a first-order stochastic differential equation (SDE):

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} \frac{\partial D_{ij}}{\partial \mathbf{r}_j} + \sum_{j=1}^{N} \frac{D_{ij} \cdot \mathbf{F}_j}{k_B T} \right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j, \quad (5)$$

where κ is the transpose of the constant velocity gradient tensor of the linear far-field flow velocity and $\mathbf{v}_i = \kappa \cdot \mathbf{r}_i$ ($\mathbf{v}_j = 0$ in a quiescent flow). The random Wiener process in the SDE dW_j is related to dt as: $dW_j = \sqrt{dt}\mathbf{n}_j$ where $\mathbf{n}_j$ is a random vector with the standard Gaussian distribution.

D is the mobility tensor of size $3N \times 3N$ and for the N-bead chain the tensor D is related to the thermal energy through the friction coefficient tensor ζ_{ij} as $\sum_l \zeta_{il} D_{lj} = k_B T \delta_{ij}$. As in [4,7], we use the RPY tensor for D.

In the absence of external driving forces, the covariance between the bead displacements satisfy the following relation:

$$\langle d\mathbf{r}_i d\mathbf{r}_j \rangle = 2D_{ij}dt. \tag{6}$$

Hence, the coefficient matrix α is connected with D via the formula $D = \alpha^\top \alpha$. We remark here that the choice of α is not unique and fast algorithms for generating these correlated random displacements actually take advantage of this fact. Finally, we observe that for the RPY tensor, $\sum_{j=1} \frac{\partial D_{ij}}{\partial \mathbf{r}_j}$ is always zero and Eq. (5) is reduced to

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} \frac{D_{ij} \cdot \mathbf{F}_j}{k_B T}\right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j. \tag{7}$$

A. Nondimensionalization of the SDE (7)

The bead-spring chain model is widely used for BD simulations of a DNA molecule. In the bead-spring chain model, the DNA molecule is represented as a chain of N beads of radius a with adjacent beads connected by a spring. Each spring contains $N_{k,s}$ Kuhn steps of length b_k. So the maximum length of each spring is $N_{k,s}b_k$, and the characteristic contour length of the double stranded DNA molecule L is approximately $(N-1)N_{k,s}b_k$ as the size of each bead is much smaller than the length of each spring and thus neglected. We denote the Hookean spring constant by H. The characteristic length l_s is chosen to be $l_s = \sqrt{k_B T/H}$ and the characteristic time t_s is chosen to be $t_s = \zeta_{\text{res}}/4H$, where ζ_{res} is the bead resistivity appeared in the RPY tensor (3). We scale the length and time by l_s and t_s, respectively and nondimensionalize Eq. (7) into the following dimensionless form:

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} D_{ij} \cdot \mathbf{F}_j\right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j, \tag{8}$$

Here with a slight abuse of notation, we have used the same notation to denote all corresponding dimensionless quantities.

B. Choices of the velocity gradient tensor κ

We now specify the velocity gradient tensor κ in Eq. (8) and restrict our attention to the following two linear planar flows. The first one is the extensional flow where $v_x = \dot{\epsilon}x, v_y = -\dot{\epsilon}y$ with $\dot{\epsilon}$ the extension rate. The second is the shear flow where $v_x = \dot{\gamma}y, v_y = 0$ with $\dot{\gamma}$ the shear rate. We define the Peclet number $Pe = \dot{\epsilon}\zeta/4H$ for the extensional flow and $Pe = \dot{\gamma}\zeta/4H$ for the shear flow, respectively. Then the dimensionless velocity gradient tensor κ in Eq. (8) is given by the following formulas:

$$\kappa_{\text{ext}} = \begin{pmatrix} Pe & 0 & 0 \\ 0 & -Pe & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \kappa_{\text{shear}} = \begin{pmatrix} 0 & Pe & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{9}$$

Here $\kappa = \kappa_{\text{ext}}$ for the extensional flow and $\kappa = \kappa_{\text{shear}}$ for the shear flow.

C. Specification of the forcing term $\mathbf{F_i}$

The force $\mathbf{F_j}$ in Eq. (8) contains two parts: the force exerted by the connected springs and the force due to the finite size of the beads. We adopt the Marko-Siggia's wormlike chain (WLC) spring law [5] to model the spring force between beads. In the WLC model, the dimensionless spring force acting on the ith bead by the ith spring is

$$\mathbf{F}_i^s = \sqrt{\frac{N_{k,s}}{3}} \left[\frac{1}{2}\frac{1}{\left(1 - \frac{Q_i}{Q_0}\right)^2} - \frac{1}{2} + \frac{2Q_i}{Q_0}\right] \frac{\mathbf{Q}_i}{Q_i}, \tag{10}$$

where $i = 1, \ldots, N-1$, $\mathbf{Q}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$ is the distance vector between bead $\mathbf{r}_{i+1}$ and $\mathbf{r}_i$, Q_i is the length of $\mathbf{Q}_i$, and Q_0 is the maximum distance between these two beads. Since all interior beads are connected with two springs from two sides, the net entropic spring force acting on the ith bead is

$$\mathbf{F}_i^{\text{entropy}} = \mathbf{F}_i^s - \mathbf{F}_{i-1}^s, \quad \mathbf{F}_0^s = \mathbf{F}_N^s = 0, \tag{11}$$

with $i = 1, \ldots, N$. For later use, we also record the potential for the ith spring below

$$U_{\text{WLC}}(\mathbf{Q}_i) = \frac{1}{2}\sqrt{\frac{N_{k,s}}{3}} \left(\frac{Q_0^2}{Q_0 - Q} - Q + \frac{2Q^2}{Q_0}\right). \tag{12}$$

For the force due to the finite size of the beads, we adopt the excluded volume force in [7,12] given by the formula

$$\mathbf{F}_i^{\text{EV}} = -\sum_{j=1, i\neq j}^{N} \frac{9\sqrt{3}z}{2} \exp\left(-\frac{3r_{ij}^2}{2}\right) \mathbf{r}_{ij} \tag{13}$$

where $z = (\frac{1}{2\pi})^{3/2} \tilde{v} N_{k,s}^2$, and $\tilde{v} = 2ab_k^2/l_s^3$ is the dimensionless excluded volume parameter. And the excluded volume potential between bead i and bead j is given by

$$U_{ij}^{\text{EV}} = \frac{3\sqrt{3}z}{2} \exp\left(-\frac{3r_{ij}^2}{2}\right). \tag{14}$$

Finally, the total force acting on bead i is the sum of the spring forces and the excluded volume forces, that is,

$$\mathbf{F}_i = \mathbf{F}_i^{\text{entropy}} + \mathbf{F}_i^{\text{EV}}. \tag{15}$$

III. NUMERICAL ALGORITHM FOR BD SIMULATIONS IN LINEAR FLOWS

In the past, a semi-implicit predictor-corrector scheme [7,9,15] was often used for the temporal integration in BD simulations. A major problem associated with that scheme is that a very small time step size has to be used in order to avoid the numerical instability, which leads to an excessively large number of time steps and a very long total simulation

time. Recently, a Metropolis integrator has been developed to integrate the self-adjoint diffusion equations [19] for BD simulations in a quiescent flow.

Here we extend the algorithm in [19] to study BD simulations in linear flows. We first introduce an integrating factor $e^{-\kappa t}$ and rewrite Eq. (8) as follows:

$$d(e^{-\kappa t}\mathbf{r}_i) = \mathrm{e}^{-\kappa t}[D(\mathbf{r}_i)\mathbf{F}(\mathbf{r}_i)dt + \sqrt{2}\alpha(\mathbf{r}_i)dW_i]. \tag{16}$$

Let $K(t) = \exp(-\kappa t)$. We now introduce a new variable $\mathbf{x}_i = K(t)\mathbf{r}_i$ [i.e., $\mathbf{r}_i = K(-t)\mathbf{x}_i$]. Then the original SDE Eq. (16) can be rewritten in terms of $\mathbf{x}_i$ as follows:

$$\begin{aligned} d\mathbf{x}_i &= K(t)[D(K(-t)\mathbf{x}_i)\mathbf{F}(K(-t)\mathbf{x}_i)dt \\ &\quad + \sqrt{2}\alpha(K(-t)\mathbf{x}_i)dW]. \end{aligned} \tag{17}$$

The generator of Eq. (17) is given by the following formula:

$$Lf(\mathbf{x}_i) = \frac{1}{\nu(\mathbf{x}_i)}\mathrm{div}\{\nu(\mathbf{x}_i)K(t)D[K(-t)\mathbf{x}_i]K(t)^T Df(\mathbf{x}_i)\}, \tag{18}$$

where we introduce the stationary density given by

$$\nu(\mathbf{x}_i) = \exp\{-U[K(-t)\mathbf{x}_i]\}. \tag{19}$$

We denote the total energy by U which is the sum of WLC spring energy U_{WLC} and EV potential energy U^{EV}. It is easy to see that the generator of the transformed stochastic differential equation with respect to $\nu(\mathbf{x}_i)$ is self-adjoint. Thus the algorithm in [19] can be directly applied to Eq. (17). We now update the position vector as follows:

(1) Compute the vector $\tilde{\mathbf{x}}_i^{n+1}$ and update $\mathbf{x}_i^n$ by the following formulas:

$$\begin{aligned} \tilde{\mathbf{x}}_i^{n+1} &= \mathbf{x}_i^n + K(t_n)G\big[K(-t_n)\mathbf{x}_i^n\big]\Delta t \\ &\quad + \sqrt{2\Delta t}K(t_n)B\big[K(-t_n)\mathbf{x}_i^n\big]dW_i, \end{aligned} \tag{20}$$

where the functions G and B are defined by the formulas

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x} + \tfrac{2}{3}D(\mathbf{x})\mathbf{F}(\mathbf{x})\Delta t, \\ G(\mathbf{x}) &= \tfrac{5}{8}D(\mathbf{x})\mathbf{F}(\mathbf{x}) - \tfrac{3}{8}D(\mathbf{x})\mathbf{F}(\mathbf{x}_1) \\ &\quad - \tfrac{3}{8}D(\mathbf{x}_1)\mathbf{F}(\mathbf{x}) + \tfrac{9}{8}D(\mathbf{x}_1)\mathbf{F}(\mathbf{x}_1), \end{aligned} \tag{21}$$

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{x} - \tfrac{2}{3}D(\mathbf{x})\mathbf{F}(\mathbf{x})\Delta t, \\ B(\mathbf{x})B(\mathbf{x})^{\top} &= \tfrac{1}{4}D(\mathbf{x}) + \tfrac{3}{4}D(\mathbf{x}_2). \end{aligned} \tag{22}$$

We then apply the Metropolis integrator to obtain the updated $\mathbf{x}_i^{n+1}$.

(2) Calculate the acceptance probability α as follows:

$$\begin{aligned} \alpha\big(\mathbf{x}_i^n, \tilde{\mathbf{x}}_i^{n+1}\big) = \min\Bigg(1, C\exp\Bigg[&-\frac{|d\tilde{W}_i|^2}{2} + \frac{|dW_i|^2}{2} \\ &- U\big(\tilde{\mathbf{x}}_i^{n+1}\big) + U\big(\mathbf{x}_i^n\big)\Bigg]\Bigg), \end{aligned} \tag{23}$$

where $C = \det B(\mathbf{x}_i^n)/\det B(\tilde{\mathbf{x}}_i^{n+1})$, $U = U_{\mathrm{WLC}} + U^{\mathrm{EV}}$ is the total potential energy, and $d\hat{W}_i$ is obtained via the formula

$$B\big(\tilde{\mathbf{x}}_i^{n+1}\big)d\tilde{W}_i = B\big(\mathbf{x}_i^n\big)dW_i + \sqrt{2\Delta t}G\big(\tilde{\mathbf{x}}_i^{n+1}\big). \tag{24}$$

(3) Generate a Bernoulli random number γ, that is, generate a uniformly distributed random number β on $[0,1]$ and set γ to 1 if $\beta \leqslant \alpha$ and 0 otherwise.

(4) Compute the updated position vector at time $t = t_{n+1}$ by the formula

$$\mathbf{r}_i^{n+1} = \gamma K(-t_{n+1})\tilde{\mathbf{x}}_i^{n+1} + (1-\gamma)\mathbf{r}_i^n. \tag{25}$$

In other words, the position vector will be updated only if the Bernoulli random number γ is equal to 1. This is the essence of the Metropolis algorithm for Monte Carlo simulations.

IV. NUMERICAL RESULTS

Common measures of the "stretch" of a DNA molecule under flow are the molecular fractional extension ($\hat{x}$ is the unit vector in the x direction)

$$X \equiv \max_i(\mathbf{r}_i \cdot \hat{x}) - \min_i(\mathbf{r}_i \cdot \hat{x}), \tag{26}$$

and its ensemble average $\langle X\rangle \equiv \frac{1}{M}\sum X$, where M is the total number of experiments (or simulations). Here we first compare the transient fractional extensions of a λ-DNA between the experimental data, semi-explicit numerical simulations [7], and our Metropolis scheme simulations. The initial DNA configurations in these simulations are the equilibrium DNA configurations in the absence of flow from the Metropolis scheme.

For the purpose of comparison, we use the same values of physical and model parameters as in [7]. That is, the viscosity η of solvent is 8.4 $cP(=mPa\cdot s)$ and the relaxation time τ is 21.0 s. The λ-DNA is modeled with $N = 29$ beads of radius $a = 0.101$ μm connected by 28 springs, where each spring has $N_{k,s} = 40$ Kuhn steps of size $b_k = 0.132$ μm and the contour length L is 150 μm. Finally, the excluded volume parameter $v = 0.0034$ $\mu\mathrm{m}^3$.

To mimic the experimental configurations, it is essential [7] to first simulate the DNA molecule to its equilibrium in a quiescent flow, i.e., $\kappa \cdot \mathbf{r}_i = 0$ in Eq. (8), which is now a self-adjoint stochastic differential equation that can be efficiently solved to an equilibrium state using the Metropolis scheme in Sec. III. At the beginning of the no-flow simulations, beads are equally spaced on the x-axis. The Metropolis scheme allows for relatively large time step Δt (an order of magnitude larger), consequently saving a significant amount of computation time for running no-flow simulations compared to the semi-implicit predictor-corrector scheme in [7]. The flow-free simulation is continued until an equilibrium configuration is reached, which is often 10–20 relaxation times (τ). After the equilibrium is reached for a DNA in a quiescent flow, we then turn flow on in the simulations and sum up $d\mathbf{r}_i$ to obtain the updated configuration and the mean fractional extension of a DNA molecule under linear flow.

In the figures below, we use the Deborah number to label different flows. As a dimensionless flow strength, the Deborah number De is equal to $\dot{\epsilon}\tau$ for the extensional flow and $\dot{\gamma}\tau$ for the shear flow. We would like to remark here that $\dot{\gamma}\tau$ is also called the Weissenberg number in the case of the shear flow in many literature. The transient fractional extension from these simulations is summarized in Fig. 1, which shows two sets of comparison for Deborah number

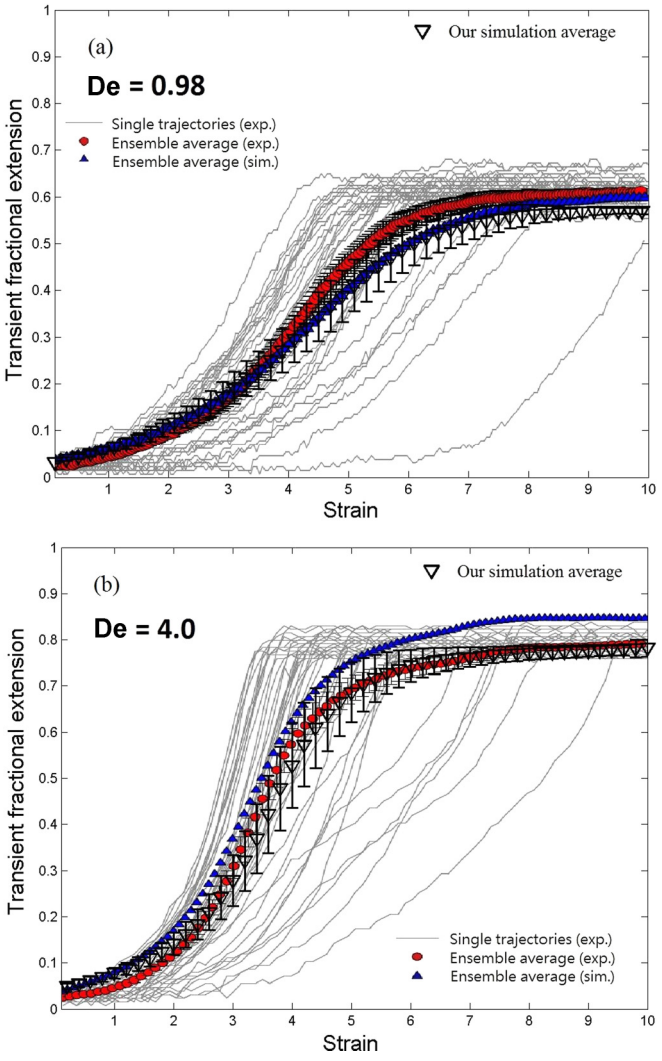

FIG. 1. (Color online) Transient fractional extension for a seven-lambda ($L = 150$ μm) DNA in a planar extensional flow. 60 trajectories from simulations are used for ensemble average.

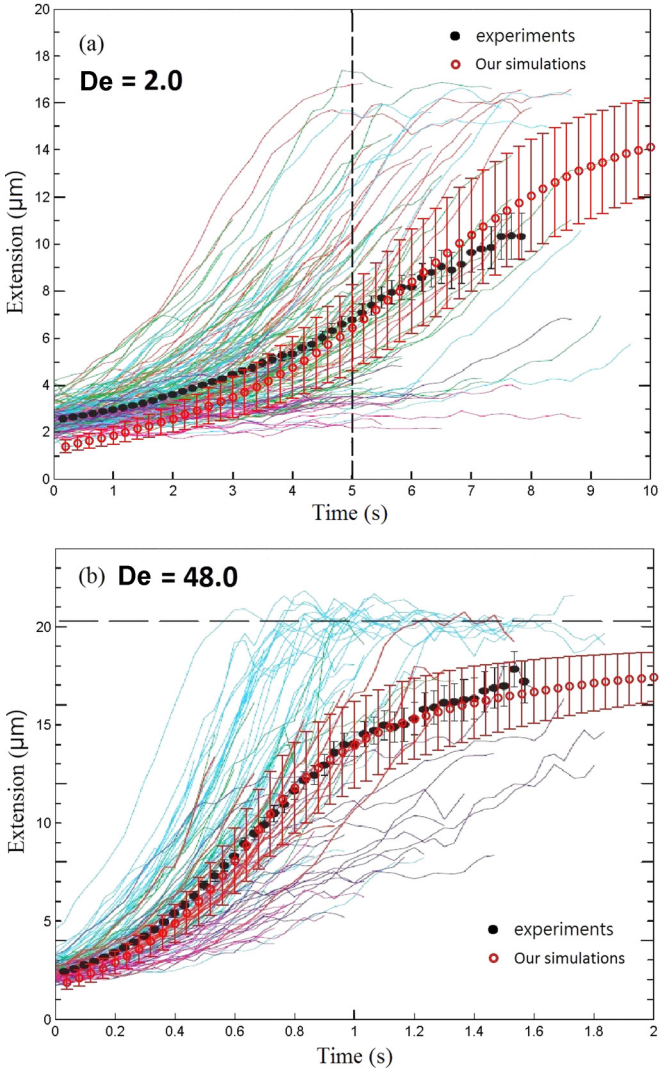

FIG. 2. (Color online) Comparison between experiments [11] (thin curves for individual trajectories and filled circles for the average) and our numerical simulations (empty circles). The vertical dashed line in (a) shows the point below which continuous data could not be collected in some experiments. The horizontal dashed line in (b) shows the steady-state of the stretched ~22 μm λ-DNA.

De $= 0.98$ ($\dot{\epsilon} \approx 0.0467$ s^{-1}) and De $= 4.0$ ($\dot{\epsilon} \approx 0.1905$ s^{-1}) for panels (a) and (b), respectively. Figure 1 is simulated by using the modified Metropolis integrator scheme with an integrating factor [Eq. (8) in Sec. III, $\kappa = \kappa_{\text{ext}}$]. Thin curves are individual trajectories from experiments, filled circles are the ensemble average from experiments, filled triangles are ensemble average from Schroeder *et al.* [7], and our results are the empty triangles. We observe that, in both panels, our results are in good agreement with the experiment results. However, our simulations are orders of magnitude more efficient because a time-step $\Delta t = 10^{-4}\tau = 2.1 \times 10^{-3}$ s is used for results in panels (a), and $\Delta t = 10^{-3}/\dot{\epsilon} = 5.25 \times 10^{-3}$ s is used for panel (b). In comparison, a much smaller time step for De $= 4.0$ and De $= 0.98$ cases are necessary for the predictor-corrector scheme [7]. The advantage of using Metropolis integrator is to capture the physical phenomena of the model. Error bars in the figures denote the standard deviation calculated from our numerical simulation data (triangles). No error bars are provided for the data from the experiment (filled circles) and Schroeder's simulation (empty circles).

Similar comparison of a single DNA molecule in a planar extensional flow between experiment and simulation are also conducted in [6]. Figure 2 compares our results against those from [11] for a 21 μm DNA molecule in an extensional flow with $N = 11$, $b_k = 0.106$ μm, $N_{k,s} = 19.8$, $a = 0.077$ μm, 23 °C for the temperature and $v = 0.0012$ μm^3. Figure 2(a) is for De $= 2.0$, $\dot{\epsilon} = 0.5$ s^{-1}, $\tau = 4.1$ s, and $\eta = 43.3$ cP. Figure 2(b) is for De $= 48.0$, $\dot{\epsilon} = 2.8$ s^{-1}, $\tau = 17.3$ s, and $\eta =$ 182 cP. Thin curves are trajectories from experiments [11], filled circles are the ensemble average of experimental results, and empty circles are the ensemble average from our modified Metropolis integrator simulations. For De $= 2.0$ [Fig. 2(a)] our average is almost identical to the simulation average from [6] (bottom panel of their figure 2). For De $= 48.0$ [Fig. 2(b)], Our simulation results are in better agreement with experimental results than those from Jendrejack *et al.* [6] and we show

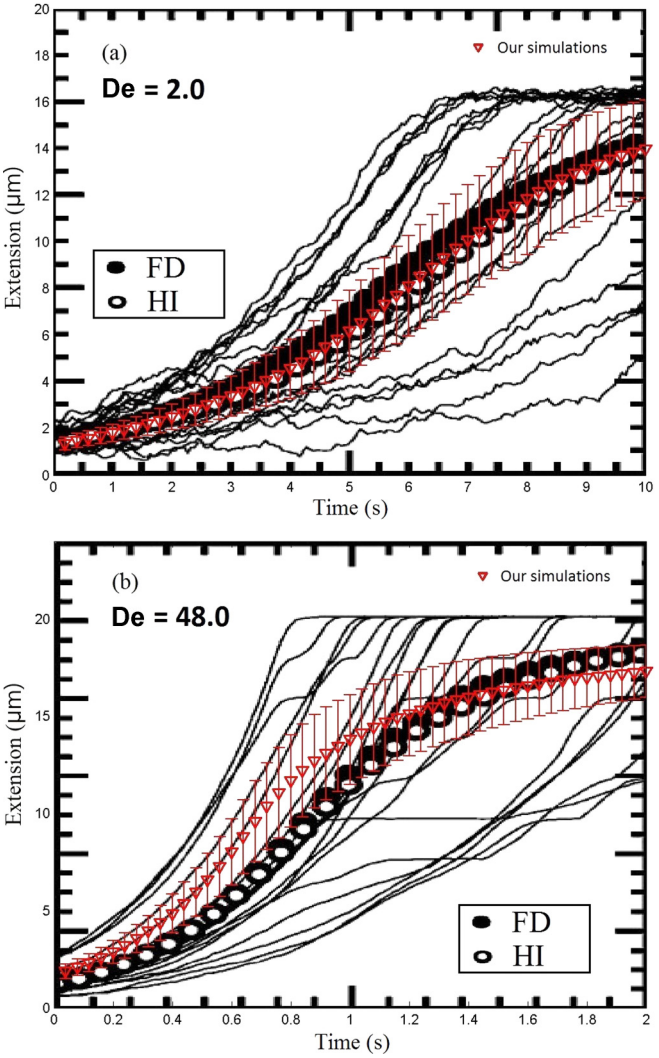

FIG. 3. (Color online) Comparison between numerical results [6] (filled and empty circles for the assemble averages of FD and HI) and our numerical simulations (empty triangles). Solid curves are single trajectories of HI simulations from [6].

these comparisons in Fig. 3. In these Metropolis integrator simulations $\Delta t = 10^{-3}$ s for both De = 2.0 in Fig. 3(a) and De = 48.0 in Fig. 3(b). Even though this time step is slightly smaller than those used in [6], our Metropolis algorithm with the integrating factor is second-order accurate [18,19] and no matrix inversion is needed. In Sec. V we describe how our numerical algorithm can be further improved when the system size is large by using FMM to efficiently calculate the HI.

Next we compare the mean fractional extension of a DNA molecule against experiments [2] and Jendrejack *et al.*'s simulations [6]. The parameters for simulations are [2]: bead radius $a = 0.077$ μm, and temperature is fixed at 20 °C. Two viscosities are considered in the experiments, $\eta = 60$ cP and 220 cP for the shear flow cases, while only $\eta = 60$ cP is used for the case of extensional flow (based on the experiments in [2]). For the corresponding simulations in [6] the number of beads is 11, Kuhn step size $b_k = 0.106$ μm, the number of springs per Kuhn step $N_{ks} = 21$, and the contour length $L = 22$ μm.

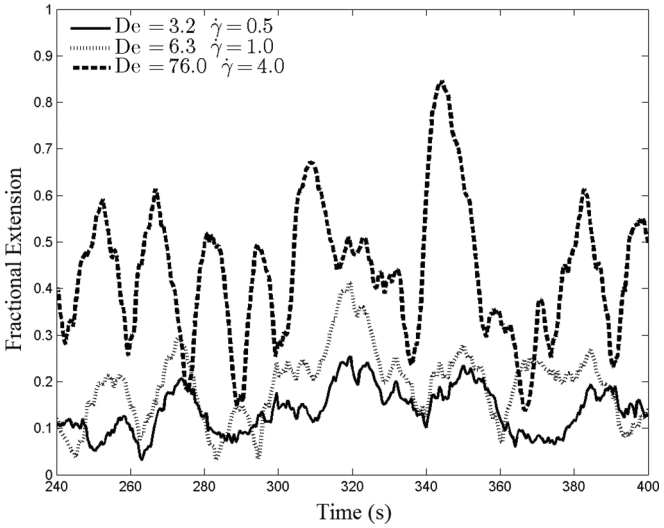

FIG. 4. Fractional extensions for different De and shear rate $\dot{\gamma}$ from our simulations with $(\mathrm{De}, \dot{\gamma}) = (3.2, 0.5)$, (6.3,1.0), (76.0,4.0), respectively. The relaxation time τ is 6.3 s for the first two cases and 19.0 s for the third case. The time steps are: $\Delta t = 10^{-3}$ s for De = 3.2, $\Delta t = 5 \times 10^{-4}$ s for De = 6.3, and $\Delta t = 2.5 \times 10^{-4}$ s for De = 76.0.

Figure 4 shows the fractional extension versus time for three cases of Deborah numbers (De = 3.2, 6.3, and 76.0) when the DNA molecule is under the simple shear flow. Since the relaxation time τ is fixed, shear rate $\dot{\gamma}$ is higher at higher De. As expected, larger mean extension of the DNA molecule is expected at a higher shear rate. From these results the mean fractional extension is computed by taking the averages over a long duration.

Figure 5 shows the comparison of mean fractional extension between experiments [2], Jendrejack *et al.*'s simulations [6]

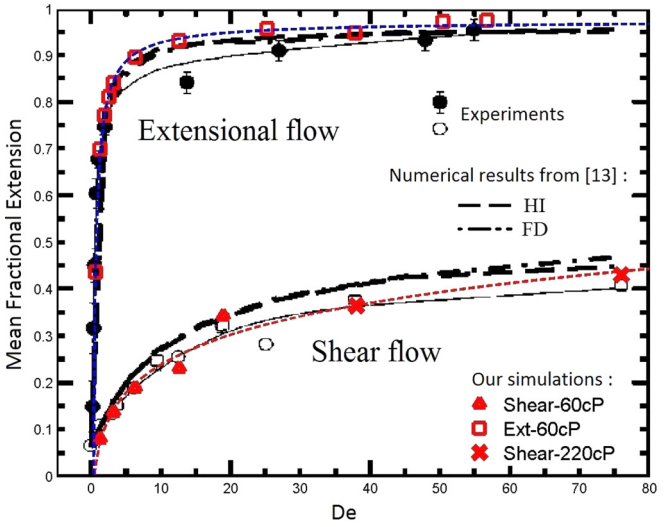

FIG. 5. (Color online) Mean fractional extensions for shear flow and extensional flow. Experimental data [2] are symbols with error bars, bead model with and without HI (see legend) are from [6] and our results as red symbols (empty circles for the extensional flow; triangles and crosses for the shear flow).

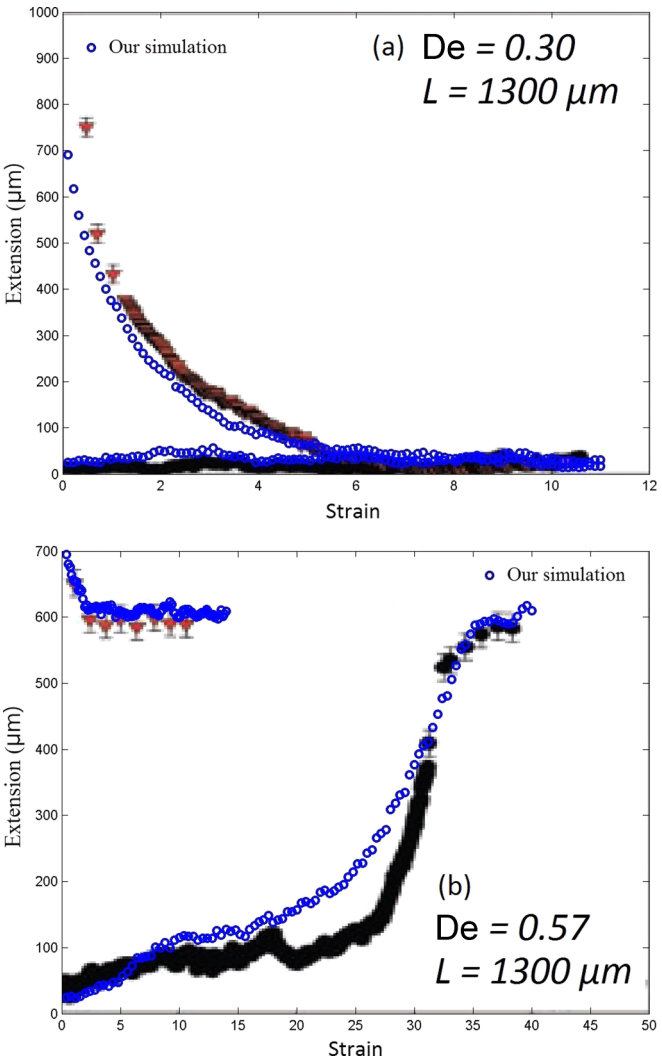

FIG. 6. (Color online) Molecular extensions for 1.3 mm DNA in an extensional flow for De = 0.30 (top) and De = 0.57 (bottom). Filled symbols are experimental data from [10]. Blue circles are our simulation results. Time steps $\Delta t = 10^{-2}$ s for De = 0.30 (top), and $\Delta t = 5 \times 10^{-3}$ s for De = 0.57 (bottom).

and our simulations. Experimental data are shown in filled dark disks for the extensional flow and dark circles for the shear flow, and the thin solid curves are their best fits. Simulation results from [6] are thick dashed (with HI) and dash-dotted [without HI, or free-draining (FD)] curves. Our simulation results are denoted by red symbols in the legends, and their best fits are the thin dashed curves. It is clear that our results agree well with experimental data for the shear flow cases. For the extensional flow cases, our results agree better with simulation results from [6] for all values of De. At larger De (De $\geqslant$ 40), all three agree well for the extensional flow cases.

Schroeder *et al.* also investigated the hysteresis of stretch-coil transition of a long λ-DNA (∼1300 μm) in an extensional flow [10]. Figure 6 shows the comparisons of single trajectories of DNA extensions over strains between their experimental data [10] and our numerical results. For the initially coiled DNA the simulation starts without flow for several relaxation times as done in previous numerical simulations [10]. Also following their procedure for the initially stretched DNA, we first run the simulations with a high Deborah number (De = 15) until equilibrium, and then gradually lower the flow strength (Deborah number) until the desired values are reached (at $t = 0$): De = 0.30 for Fig. 6(a) and De = 0.57 for Fig. 6(b). The parameters for simulations are the following: $a = 0.28\,\mu$m, $N_s = 123$, $N_{ks} = 80$, $b_k = 0.132$ μm, $\nu = 0.00032$ μm^3, and $\tau = 126.0$ s. We set the solvent viscosity to be 1 cP and time step size (a) $\Delta t = 10^{-2}$ s and (b) $\Delta t = 5 \times 10^{-3}$ s are used for simulations. The agreement with the experimental data demonstrates that our numerical methods are able to capture the hysteric transition between stretched and coiled DNA in an extensional flow.

V. EXTENSION TO LARGE SYSTEMS

In the numerical algorithm described in Sec. III, the RPY tensor D is constructed explicitly, the matrix vector product $D\mathbf{F}$ is computed directly, and the upper-triangular matrix B is obtained by the Cholesky decomposition with its determinant simply the product of its diagonal entries. This is affordable for the numerical experiments presented in Sec. IV since the total number of beads $N = 29$. However, for large systems the computational cost of these standard direct algorithms becomes prohibitively expensive since the matrix vector product $D\mathbf{F}$ requires $O(N^2)$ operations, the Cholesky factorization requires $O(N^3)$ operations, and each BD simulation often requires more than 10^5 time steps. Thus, fast algorithms become a necessity in order to make long-time large-scale BD simulations practical.

As mentioned in Sec. I, recently a fast multipole method for the RPY tensor (RPYFMM) has been developed in [20]. The fundamental observation in [20] is that the RPY tensor can be decomposed as follows:

$$D_{ij} = C_1\left[\frac{\delta_{ij}}{|\mathbf{x}-\mathbf{y}|} - (x_j - y_j)\frac{\partial}{\partial x_i}\frac{1}{|\mathbf{x}-\mathbf{y}|}\right] + C_2\frac{\partial}{\partial x_i}\frac{x_j - y_j}{|\mathbf{x}-\mathbf{y}|^3}, \quad (27)$$

where $C_1 = \frac{k_B T}{8\pi\eta}, C_2 = \frac{k_B T a^2}{12\pi\eta}$.

With this decomposition, the matrix vector product $D\mathbf{v}$ for a given vector $\mathbf{v}$ can be interpreted as a linear combination of four harmonic sums with suitably chosen source charges and dipoles. In other words, the matrix vector product $D\mathbf{v}$ can be evaluated by four calls of the classical FMM for Coulomb interactions in three dimensions [22]. Thus, the RPYFMM avoids the explicit construction of the RPY tensor and reduces

TABLE I. Timing results (sec) for computing $T = D\mathbf{v}$ by RPYFMM.

N	T_{RPYFMM}	T_{Direct}	E_{RPYFMM}
1000	0.20897	0.31495	1.6008×10^{-02}
10 000	1.6058	30.6643	5.5339×10^{-02}
100 000	16.172	2738.48	8.3803×10^{-02}
1 000 000	160.24	271009.4	1.1603×10^{-01}

TABLE II. Timing results (sec) for computing $T = \sqrt{D}\mathbf{v}$ by RPYFMM-SLDM.

N	m	T_{SLDM}	E_{relative}
1000	4	0.54192	6.21032×10^{-06}
10 000	4	9.03360	6.24604×10^{-04}
100 000	6	111.80	7.92857×10^{-04}
1 000 000	12	2180.8	2.91239×10^{-04}

the computational cost of $D\mathbf{v}$ to $O(N)$ in both CPU time and memory storage.

We observe further that the Cholesky factor B of the RPY tensor D can be replaced by any matrix C which satisfies the same matrix equation $CC^{\top} = D$ (note that there are actually infinitely many matrices satisfying this matrix equation, see, for example, [17] for details). Indeed, [20] also proposed to replace the Cholesky factor B by $\sqrt{D}$ and compute $\sqrt{D}\mathbf{v}$ by combining the classical Spectral Lanczos Decomposition Method (SLDM) with the RPYFMM. The resulting algorithm has $O(\kappa N)$ complexity with κ the condition number of the RPY tensor D. We remark here that for most BD simulations with HIs, the beads do not overlap with each other due to the EV force and our numerical experiments show that the condition number of the RPY tensor in this case is fairly low. This indicates that the RPYFMM-SLDM method is essentially a linear algorithm for computing $\sqrt{D}\mathbf{v}$. The timing results presented in Tables I and II clearly demonstrate of linear scaling of the RPYFMM and RPYFMM-SLDM methods.

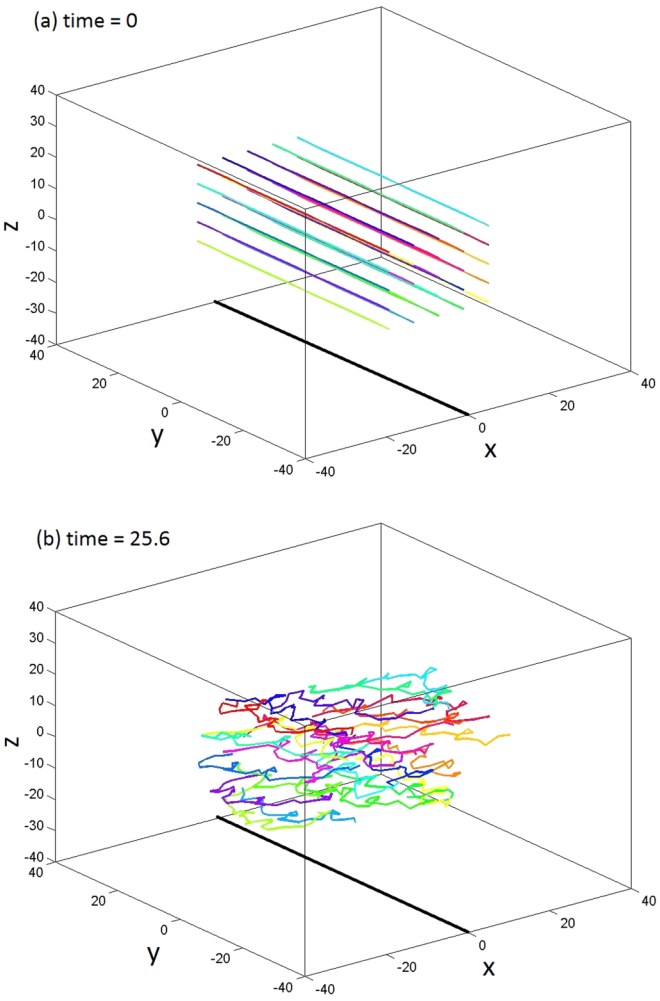

FIG. 7. (Color online) Numerical experiments of many-DNA in an oscillatory shear flow at $t = 0$ and $t = 25.6$, when shear flow velocity is zero.

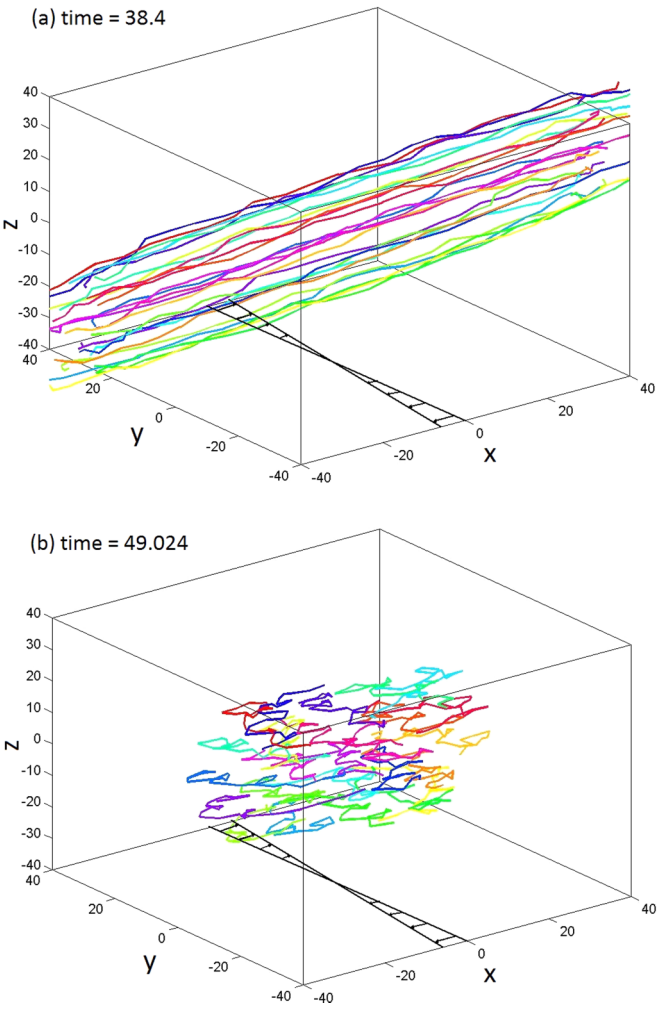

FIG. 8. (Color online) Numerical experiments of many-DNA in an oscillatory shear flow at $t = 38.4$ and $t = 49.024$.

Finally, we would like to remark here that recent developments in the fast multipole methods and fast direct solvers also enable a linear algorithm for computing the determinant of a matrix with certain hierarchical low-rank structure [23–25]. By incorporating all these fast methods into our current numerical scheme, we obtain a numerical algorithm which is stable even for relatively large time-step size and scales linearly with respect to the number of particles (or beads) in the system.

Figures 7 and 8 are simulation snapshots of many DNA in an oscillatory shear flow. Similar to the previous work [26] we define the background oscillatory shear as $\mathbf{U}_0 = [\dot{\gamma}\sin(2\pi\omega t)y, 0, 0]$ (on the $x-y$ plane in each panel), where $\omega = (20000\Delta t)^{-1}$, $\Delta t = 0.00128$, and $\dot{\gamma} = 1.0$ is the shear rate for simulations. For this simulation we include 25 DNA molecules, each of which has a rest contour length of

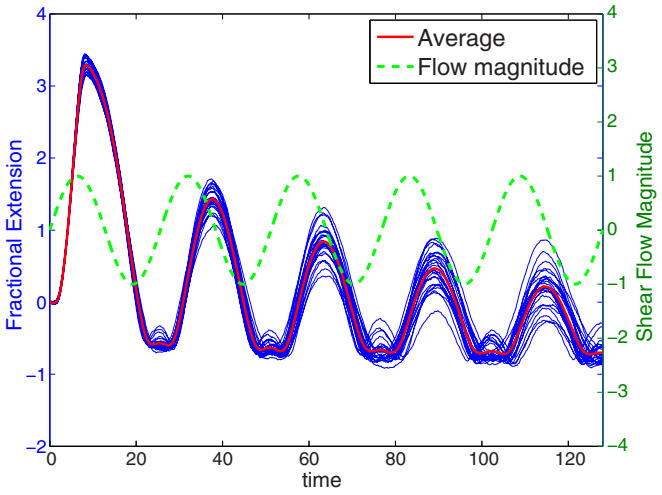

FIG. 9. (Color online) Numerical simulations for 25 DNA molecules in an oscillatory shear flow. Blue trajectories are fractional extensions of each molecule, dashed green curve is the magnitude of periodic shear flow and solid red curve is the assemble average.

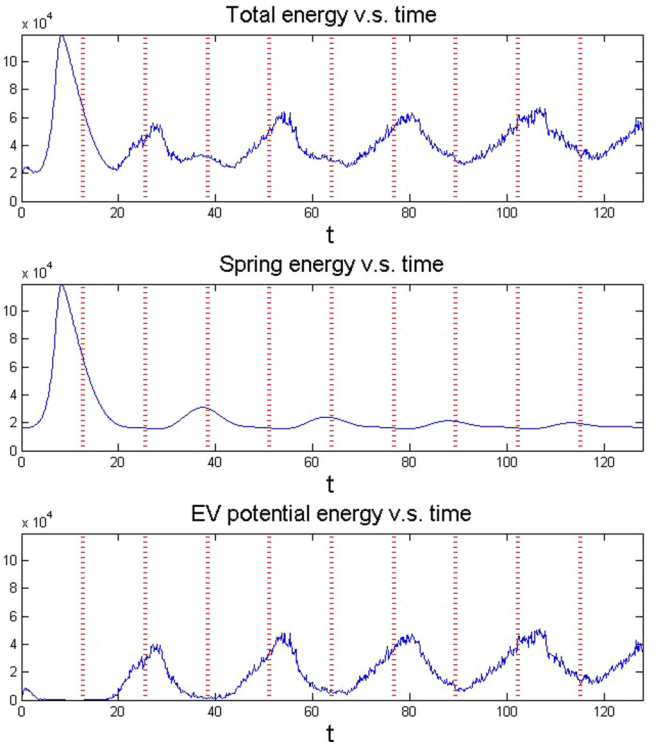

FIG. 10. (Color online) Energies versus time t for 25 DNA molecules in an oscillatory shear flow. Total energy (top) is the sum of WLC spring energy (middle) and EV potential energy (bottom). Red dotted vertical lines represent half period of flow oscillation.

150 μm. The parameter set of 150 μm long DNA (29 beads) is used (total number of beads is 725). In the simulation, each molecule has the same initial extension. Figure 9 illustrates the correlation between mean molecule extension and the oscillating shear flow magnitude. Figure 10 shows that the spring energy dominates the total energy during the first period ($t = 25.6$) since the DNA molecules stretch under shear flow. After one period of time, DNA molecules turn to coiled states and EV potential energy is dominant due to molecule-molecule interactions.

Finally, the timing result for one time step Δt including forcing calculations and matrix-vector multiplications with the use of RPYFMM and RPYFMM-SLDM is ∼0.3 s. However, it takes ∼98.0 s at each time step for direct calculations involving construction of RPY tensor and direct factorization using Cholesky decomposition. This result shows that one can reduce much computational cost by using the $O(N)$-operation algorithms when N is large.

VI. CONCLUSION AND DISCUSSION

We have extended the Metropolis integrator in [19] to study BD simulations with HIs in linear flows. The method utilizes the integrating factor to absorb the effect of the linear flow and permits much larger time step sizes for BD simulations with HIs in linear flows. We have applied our method to study the fractional stretch and the mean stretch of a single λ-DNA molecule in planar linear flows. Our numerical results agree very well with experimental data [2,11] and other simulation results [7] in the literature.

We have also discussed the extension of our method to large systems in Sec. V. By incorporating the RPYFMM and other fast algorithms into the scheme, the resulting algorithm admits large time step sizes and has nearly optimal complexity [i.e., $O(N)$ or $O(N \log N)$] in the number of particles in the system. Thus, even though many of these fast algorithms have a large prefactor (say, $C \geqslant 1000$) in front of N, the combination of our fast algorithm with modern computers makes long-time large-scale BD simulations with HIs within practical reach. We are currently incorporating these fast algorithms into the modified Metropolis integrator and applying the resulting algorithm to study the lipid bilayer membrane of the red blood cells in the blood flow. Results from these ongoing work are being analyzed now and will be reported in a timely fashion.

ACKNOWLEDGMENTS

The authors acknowledge useful comments and suggestions from the referees. The authors thank N. Bou-Rabee and C. Schroeder for helpful discussions. S.-P.F. and Y.-N.Y. are supported by NSF under Grant No. DMS-1222550. S.J. is supported by NSF under Grant No. DMS-1418918.

[1] T. T. Perkins, D. E. Smith, and S. Chu, Science **276**, 2016 (1997).
[2] D. E. Smith, H. P. Babcock, and S. Chu, Science **283**, 1724 (1999).
[3] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics* (Oxford Science Publications, New York, 1986).
[4] D. L. Ermak and J. A. McCammon, J. Chem. Phys. **69**, 1352 (1978).

[5] J. F. Marko and E. D. Siggia, Macromolecules **28**, 8759 (1995).
[6] R. M. Jendrejack, J. J. de Pablo, and M. D. Graham, J. Chem. Phys. **116**, 7752 (2002).
[7] C. M. Schroeder, E. S. G. Shaqfeh, and S. Chu, Macromolecules **37**, 9242 (2004).
[8] H. P. Babcock, D. E. Smith, J. Hur, E. S. G. Shaqfeh, and S. Chu, Phys. Rev. Lett. **85**, 2018 (2000).
[9] C.-C. Hsieh, L. Li, and R. G. Larson, J. Non-Newton Fluid **113**, 147 (2003).
[10] C. M. Schroeder, H. P. Babcock, E. S. G. Shaqfeh, and S. Chu, Science **301**, 1515 (2003).
[11] D. E. Smith and S. Chu, Science **281**, 1335 (1998).
[12] J. R. Prakash, J. Rheol. **46**, 1353 (2002).
[13] R. M. Jendrejack, M. D. Graham, and J. J. de Pablo, J. Chem. Phys. **113**, 2894 (2000).
[14] R. G. Larson, H. Hua, D. E. Smith, and S. Chu, J. Rheol. **46**, 267 (1999).
[15] M. Somasi, B. Khomami, N. J. Woo, J. S. Hur, and E. S. G. Shaqfeh, J. Non-Newtonian Fluid Mech. **108**, 227 (2002).
[16] J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[17] S. Jiang, Z. Liang, and J. Huang, Math Comput. **82**, 1631 (2013).
[18] N. Bou-Rabee, Entropy **16**, 138 (2014).
[19] N. Bou-Rabee, A. Donev, and E. Vanden-Eijnden, Multiscale Model. Simul. **12**, 781 (2014).
[20] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, and S. Jiang, J. Comput. Phys. **234**, 133 (2013).
[21] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[22] H. Cheng, L. Greengard, and V. Rokhlin, J. Comput. Phys. **155**, 468 (1999).
[23] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil, arXiv:1403.6015.
[24] K. L. Ho and L. Ying, arXiv:1307.2895.
[25] K. L. Ho and L. Ying, arXiv:1307.2666.
[26] A.-K. Tornberg and M. Shelley, J. Comput. Phys. **196**, 8 (2004).