

# Brownian dynamics simulations with stiff finitely extensible nonlinear elastic-Fraenkel springs as approximations to rods in bead-rod models

Chih-Chen Hsieh,<sup>a)</sup> Semant Jain, and Ronald G. Larson*Department of Chemical Engineering, University of Michigan, Ann Arbor, Michigan 48109*

(Received 4 October 2005; accepted 28 November 2005; published online 30 January 2006; publisher error corrected 2 February 2006)

A very stiff finitely extensible nonlinear elastic (FENE)-Fraenkel spring is proposed to replace the rigid rod in the bead-rod model. This allows the adoption of a fast predictor-corrector method so that large time steps can be taken in Brownian dynamics (BD) simulations without over- or understretching the stiff springs. In contrast to the simple bead-rod model, BD simulations with beads and FENE-Fraenkel (FF) springs yield a random-walk configuration at equilibrium. We compare the simulation results of the free-draining bead-FF-spring model with those for the bead-rod model in relaxation, start-up of uniaxial extensional, and simple shear flows, and find that both methods generate nearly identical results. The computational cost per time step for a free-draining BD simulation with the proposed bead-FF-spring model is about twice as high as the traditional bead-rod model with the midpoint algorithm of Liu [*J. Chem. Phys.* **90**, 5826 (1989)]. Nevertheless, computations with the bead-FF-spring model are as efficient as those with the bead-rod model in extensional flow because the former allows larger time steps. Moreover, the Brownian contribution to the stress for the bead-FF-spring model is isotropic and therefore simplifies the calculation of the polymer stresses. In addition, hydrodynamic interaction can more easily be incorporated into the bead-FF-spring model than into the bead-rod model since the metric force arising from the non-Cartesian coordinates used in bead-rod simulations is absent from bead-spring simulations. Finally, with our newly developed bead-FF-spring model, existing computer codes for the bead-spring models can trivially be converted to ones for effective bead-rod simulations merely by replacing the usual FENE or Cohen spring law with a FENE-Fraenkel law, and this convertibility provides a very convenient way to perform multiscale BD simulations.

© 2006 American Institute of Physics. [DOI: [10.1063/1.2161210](https://doi.org/10.1063/1.2161210)]

## I. INTRODUCTION

The response of dilute solutions of long-chain polymers to imposed flows has been of interest for decades, not only because of its fundamental importance in the characterization of the polymer, but also because of applications such as DNA electrophoresis, DNA microarray analyses, and turbulent drag reduction. To simulate polymer molecules in dilute solutions, molecular-dynamics simulations could be used and quantitative predictions may be obtained. However, it is still far from affordable to simulate with this method a typical polymer or DNA molecule consisting of tens of thousands atoms with numerous surrounding solvent molecules. Fortunately, since the phenomena happening at very short-time (femtosecond to picosecond) and very small length scales ( $\leq 1$  nm) are not of primary importance for rheological properties of dilute polymer solutions, Brownian dynamics (BD), which is more coarse grained, can be used. Taking advantage of the fact that the configuration relaxation time of a long-chain polymer is much longer than its momentum relaxation time, one can replace effectively the velocity distribution of the polymer with a stochastic process with zero correlation

time and thus avoid resolving the very fast dynamics caused by the extremely frequent bombardment of solvent molecules.

With Brownian dynamics, linear polymer molecules in dilute solutions are typically modeled by a chain of beads connected by either freely jointed links (freely jointed chain model) or entropic springs (bead-spring model). In the freely jointed chain (FJC) model, a polymer is assumed to remain locally at equilibrium even it has been disturbed at longer length scales. Although the details of the atomic structure of the polymer are lost in such a coarse-grained model, their collective effect can be attributed to the representative length scale of a Kuhn step, which is chosen to be the length of a link in the FJC model. The freely jointed chain model has been shown to reproduce the statistical properties of real polymers on length scales much larger than that of a Kuhn step.<sup>1</sup>

To simulate a polymer molecule more efficiently, one can use the more coarse-grained bead-spring model in which each entropic spring represents an ensemble of links in the FJC model. The entropic spring force arises from a reduction of the configuration space available to a chain of freely jointed links which is stretched. Consequently, the spring force can be derived from the FJC model and usually can be expressed by a simple function of the stretch of the spring

<sup>a)</sup>Electronic mail: [cchsieh@umich.edu](mailto:cchsieh@umich.edu)

that depends on the degree of the coarse graining.<sup>2</sup> Thus, the bead-spring model and the FJC model can be considered equivalent at long-time and long length scales. Both models have been used to simulate DNA/polymer behavior and have been shown to yield results in quantitative agreement with experimental data. More detailed information can be found in recent reviews of Larson<sup>3</sup> and Shaqfeh.<sup>4</sup>

Lately, there is growing interest in understanding the behavior of DNA/polymer near a solid boundary and in micro- and nanofluidic devices. However, the traditional spring laws that perform well in simulating polymers in bulk flow are no longer valid for such confined chains due to the loss of the configuration space available to the FJC chain. As a result, one should either develop different spring laws for different degrees of confinements<sup>5</sup> or use the FJC model instead of the bead-spring model. While the latter method provides a more general solution for BD simulations in confined geometries, it is computationally more expensive than the former method. Therefore, improving the efficiency of the FJC model, especially when hydrodynamic interaction is present, is a key problem that needs to be solved.

The freely jointed chain model is usually implemented as the bead-rod model. In the bead-rod model proposed by Kramers,<sup>6</sup> a rod represents a rigid link with a fixed length equal to that of a Kuhn step, and a bead is a frictional center. However, the bead-rod model is not the only way to implement a freely jointed chain, and it has been debated whether a stiff Fraenkel spring or a rigid rod is the better choice for simulating the links in a freely jointed chain.<sup>7</sup> A Fraenkel spring generates a force proportional to the deviation between the actual spring length and its natural spring length.<sup>8</sup> The Fraenkel spring can be so stiff that its length can be kept nearly constant even in a strong flow. Thus, a bead-spring model with Fraenkel springs can be effectively considered as a freely jointed chain. Using this approach, however, in traditional algorithms very small time steps must be used to resolve stably and accurately the rapid change of spring lengths even though one is only interested in the large-scale long-time behavior of the chains. Hence, the bead-rod model has been overwhelmingly preferred over the bead-spring model with stiff Fraenkel springs because much larger time steps can be used with the former without breaking the constraints on rod lengths. However, there are some subtle differences between the bead-rod and the bead-spring models even in the limit of infinite spring stiffness, and the behavior of the bead-spring model with stiff Fraenkel springs is actually closer to that of a freely jointed chain than is the Kramers bead-rod model. Consider, for example, a trumbbell consisting of three beads and two rods. For the bead-spring trumbbell with elastic springs, the probability distribution for the included angle  $\theta$  is given by the random-walk configuration,<sup>6,9,10</sup>

$$p(\theta) = C_1 \sin(\theta). \quad (1)$$

This probability distribution is valid for any elastic spring even in the limit of infinite stiffness. However, Kramers bead-rod trumbbell yields a different probability distribution for the included angle, namely,

$$p(\theta) = C_2 \sin \theta \sqrt{1 - \frac{1}{4} \cos^2 \theta}, \quad (2)$$

where  $C_1=0.5$  and  $C_2=0.522\ 678$  are normalization constants.

To retain the random-walk configuration while preserving the efficiency of the bead-rod model, a modified bead-rod model that uses an additional pseudopotential to recover the random-walk configuration at equilibrium has been proposed.<sup>11–13</sup> However, most workers have simply accepted the deviation from random-walk behavior as appropriate behavior for the Kramers chain.<sup>6,14–18</sup> This “traditional” approach has been carried out both with and without hydrodynamic interaction, while the alternative approach, which restores the random-walk configuration, has only been implemented for the free-draining case.

In addition to the above issue, constructing a correct Brownian dynamics simulation procedure for the bead-rod model is not trivial. One must be careful to construct the correct stochastic differential equation (i.e., using Ito’s or Stratonovich’s interpretation; using generalized coordinates or Cartesian coordinates) and to choose an appropriate numerical method (Lagrange multipliers or projection operators). Many studies have been criticized because they ignored some subtle issues associated with these methods.

The best known algorithm for simulating the free-draining bead-rod chain is the midpoint algorithm proposed by Ryckaert *et al.*<sup>19</sup> and first implemented by Liu.<sup>15</sup> In Liu’s method, Lagrange multipliers determined by an iterative scheme are used to enforce constraints on rod lengths. Although everything is evaluated using information obtained at the beginning of each time step, Liu’s algorithm is equivalent to a midpoint algorithm. The advantage of this method is its efficiency. For each iteration, a system of linear equations for the Lagrange multipliers needs to be solved. Since the coefficient matrix is (nonsymmetrically) tridiagonal and constant at each time step over all iterations, the Lagrange multipliers can very efficiently be obtained using backward substitution. The weakness of Liu’s algorithm is in the calculation of the stresses in transient flows, but Doyle *et al.*<sup>17</sup> developed a variance reduction method for the stresses that makes only moderate sacrifice of efficiency. In addition to Liu’s method, a similar algorithm called SHAKE was introduced by Ryckaert *et al.*,<sup>19</sup> in which the Lagrange multipliers are also used to enforce the constraints, but not as efficiently as in Liu’s algorithm. This algorithm is named “SHAKE” because the fulfillment of each constraint partially destroys all previous constraints. Both the midpoint algorithm and the SHAKE algorithm follow the traditional approach (the Kramers bead-rod model) and contain no error in their formulation.

Besides using the Lagrange multipliers, the constraints can also be fulfilled by a second method using projection operators.<sup>13,20–22</sup> In this method, one projects the unconstrained motion of the chain onto the constrained space and thus rejects the motion along the forbidden directions. The drawback of this method is that the trajectory will slowly move away from the constrained space due to the accumulation of the discretization errors and thus will result in a growing deviation in the rod length from its expected value. This

subtle effect must be suppressed to obtain the correct simulation results.<sup>13,21</sup> However, in doing so, care must be taken to avoid introducing additional errors. The advantage of the projection operator is its speed since there is no need for iterations (although constructing the projection operators is more complicated than carrying out the matrix operations to determine the Lagrange multipliers). Although no direct comparison in speed between Liu's algorithm and the projection-operator method is available in the literature, the latter method should be at least as efficient as the former one.

Grassia and Hinch<sup>20</sup> followed the approach that recovers the random-walk equilibrium distribution and performed simulations of polymer chain relaxation with the midstep algorithm proposed by Fixman.<sup>11</sup> In the midstep algorithm, a predictor step is first used to evaluate the midpoint, and then the second corrector step is performed using the information obtained at the midpoint. A problem in Grassia and Hinch's algorithm is the way they suppress the numerical drift from the constrained space that leads to an increase in rod length from its natural value. Grassia and Hinch<sup>20</sup> and Montesi *et al.*<sup>13</sup> simply rescaled the rod vectors whenever the rod length deviates to a preset upper bound so rod length can be kept nearly constant. However, this simple correction unintentionally alters the equilibrium configuration distribution of the chain and results in a deviation in, for example, equilibrium end-to-end distance and the distribution of the included angles between rods.<sup>13,21</sup> This problem becomes more serious when the time step is larger and the simulation is longer.

To simulate non-free-draining bead-rod chains, the most widely used algorithm is "SHAKE-HI" originally proposed by Allison and McCammon<sup>14</sup> or a variation of it. This algorithm is a straightforward extension of the SHAKE algorithm to include hydrodynamic interaction (HI) and has been used by Agarwal *et al.*,<sup>23</sup> Agarwal,<sup>24</sup> Lyulin *et al.*,<sup>25</sup> and Neelov *et al.*<sup>26</sup> Although the SHAKE-HI algorithm may seem correct, it does not reproduce the corresponding kinetic theory.<sup>16</sup> First, a metric force arising from the conversion of generalized coordinates to Cartesian coordinates was missed. Second, the constraint conditions are fulfilled at the end of the time step rather than at midpoint. The midpoint and SHAKE algorithms, on the other hand, are correct because, for the free-draining case with identical beads and isotropic mobility tensors, the metric force is canceled, and fulfilling the constraints at the end of the step is equivalent to fulfilling those at the midpoint. Öttinger<sup>16</sup> realized these errors and derived correct stochastic differential equations for bead-rod simulations with HI using Ito's interpretation. Muthukumar and co-workers<sup>27,28</sup> implemented Öttinger's method to simulate a bead-rod chain with hydrodynamic interaction in both shear and extensional flows.

In this study, we revisit the approach of using stiff springs to mimic links with constant lengths. With a newly proposed spring law [the "finitely extensible nonlinear elastic (FENE)-Fraenkel spring"] and a stable numerical method, the bead-spring model can be efficiently used to simulate a freely jointed chain. The advantages are threefold. First, using a bead-spring model avoids imposing rigid constraints on the rod length and thus avoids the complexity and ambiguity introduced by these constraints. Second, the absence of the

rigid constraints significantly simplifies the calculation of the polymer stresses. Third, existing computer codes for the bead-spring model with nonlinear springs can be trivially switched to simulate the bead-FENE-Fraenkel-spring model and may simplify the development of a multiscale BD simulation platform.

This article is organized as follows. In Sec. II, we propose a FENE-Fraenkel spring to replace the Fraenkel spring as an approximation to a rod in the bead-rod model. A stable predictor-corrector method is then adapted for the newly developed bead-FENE-Fraenkel model. In Sec. III, we compare predictions for the Kramers bead-rod model and for the bead-FENE-Fraenkel spring model in equilibrium, relaxation, start-up of uniaxial extensional flow, and steady shear flow in the free-draining limit and steady shear flow in the non-free-draining limit. In Sec. IV we summarize the results and give the conclusions.

## II. MODEL

### A. FENE-Fraenkel spring law

Due to the Hookean nature of the Fraenkel spring, very small time steps have to be used to resolve the fast fluctuations of the spring and to prevent the spring from being overly stretched or compressed. To develop a more efficient algorithm, we need to invent a new spring law and a corresponding implicit numerical method. More specifically, we intend to construct a spring whose length can only fluctuate within a very small range around the value of the rod length  $Q_0$  so that the spring behaves similarly to a rigid link in the freely jointed chain.

To construct such a spring, we mimic the formulation of the FENE spring law and the Fraenkel spring law, where the former is finitely extensible and the latter has a nonzero equilibrium length. The FENE and the Fraenkel springs are defined by

$$\text{FENE spring: } \mathbf{F} = \frac{H\mathbf{Q}}{1 - Q^2}, \quad (3)$$

$$\text{Fraenkel spring: } \mathbf{F} = H(Q - 1)\frac{\mathbf{Q}}{Q}. \quad (4)$$

We thus propose a FENE-Fraenkel (FF) spring force law that is very similar to the FENE spring force law but has a nonzero natural length.

$$\mathbf{F} = \frac{H(Q - 1)}{1 - (1 - Q)^2/s^2} \frac{\mathbf{Q}}{Q} \text{ for } (1 - s) < Q < (1 + s), \quad (5)$$

where  $\mathbf{Q}$  is the spring vector,  $Q$  is the spring length,  $H$  is the spring constant, and  $s$  is the extensibility parameter that defines the maximum possible deviation between the actual spring length and the natural spring length. All variables used in Eqs. (3), (4), and (5), and the following are nondimensionalized using appropriate length and time scales. We follow the common procedure for the bead-rod model<sup>17</sup> and choose  $Q_0$ ,  $Q_0^2\zeta/k_B T$ , and  $k_B T/Q_0$  as the length, time, and force scales, respectively. Here  $\zeta$  is the drag coefficient of a

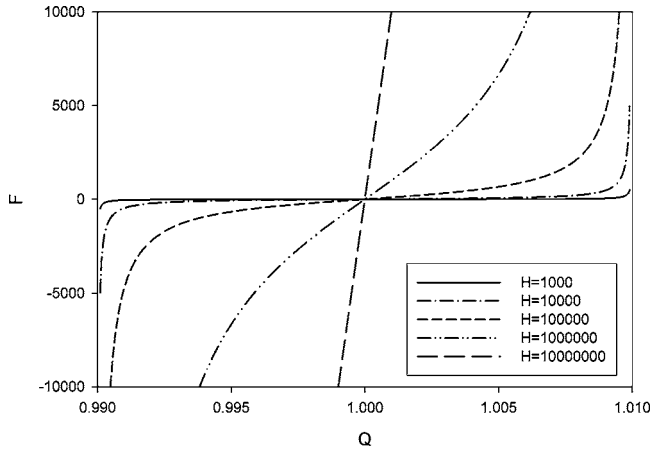


FIG. 1. The spring force of a FENE-Fraenkel spring vs the spring length with the extensibility parameter  $s$  set to be 0.01. Here and all subsequent figures, all quantities are nondimensionalized as described in the text with  $Q_0$ ,  $Q_0^2 \zeta / k_B T$ , and  $k_B T / Q_0$  as the length, time, and force scales, respectively.

bead and can be related to the bead radius  $a$  and the solvent viscosity  $\eta_s$  by  $\zeta = 6\pi\eta_s a$ .

For the FENE-Fraenkel spring, the spring force diverges when the spring length  $Q$  approaches either the lower bound  $(1-s)$  or the upper bound  $(1+s)$  so that the spring length is confined between  $(1-s)$  and  $(1+s)$ . We illustrate how the spring force changes with the spring extension for different spring constants with  $s$  fixed as 0.01 in Fig. 1. As can be seen, the spring length is always confined within the preset range for any spring constant  $H$ . The effect of  $H$  can be understood by taking the derivative of the spring force  $F$  with respect to the spring length  $Q$ . That is,

$$\frac{dF}{dQ} = \frac{H[1 + (1-Q)^2/s^2]}{[1 - (1-Q)^2/s^2]^2}. \quad (6)$$

At  $Q=1$ , the spring becomes Hookean with

$$\frac{dF}{dQ}(Q=1) = H. \quad (7)$$

For the Fraenkel spring, the spring constant  $H$  should be chosen so that the spring equilibrates its length faster than a rod equilibrates its orientation in the freely jointed chain

model. Otherwise, the relaxation spectrum of a freely jointed chain will be truncated by the relaxation time of a spring. This might not be a serious problem when the imposed flow gradient is weak because only the slowest modes are important. However, the criterion for the spring constant must be met when the flow gradient is strong and the fast modes become important. For the FENE-Fraenkel spring, the effective spring constant changes with the stretch of the spring [see Eq. (6)], and thus the effective spring constant increases with flow strength. Apart from the above constraint, the value of the spring constant does not have any physical meaning and can be chosen to optimize the numerical method.

## B. Brownian dynamics simulation with bead-FENE-Fraenkel-spring model

We start from Brownian dynamics simulations with hydrodynamic interaction for which the force balance on the  $i$ th bead can be described by<sup>29</sup>

$$\frac{d\mathbf{r}_i}{dt} = \boldsymbol{\kappa} \cdot \mathbf{r}_i + \sum_{j=1}^N \frac{\partial}{\partial \mathbf{r}_j} \cdot \mathbf{D}_{ij}^T + \sum_{j=1}^N \mathbf{D}_{ij} \cdot \mathbf{F}_j^{\text{sp},b} + \left(\frac{6}{\Delta t}\right)^{1/2} \sum_{j=1}^i \boldsymbol{\sigma}_{ij} \cdot \mathbf{n}_j, \quad (8)$$

where  $\boldsymbol{\kappa} = \nabla \mathbf{v}^T$  is the transpose of the velocity gradient tensor,  $N$  is the number of beads,  $\mathbf{n}_j$  is a random vector uniformly distributed in each of three directions over the interval  $[-1, 1]$ ,  $\mathbf{r}_i$  is the coordinate vector of bead  $i$  and  $\mathbf{F}_j^{\text{sp},b}$  is the overall spring force acting on bead  $j$ . That is, for interior beads,

$$\mathbf{F}_j^{\text{sp},b} = \mathbf{F}_j^{\text{sp}} - \mathbf{F}_{j-1}^{\text{sp}}, \quad (9)$$

where  $\mathbf{F}_j^{\text{sp}}$  is the force that spring  $j$  exerts on bead  $j$ . For the end beads,

$$\mathbf{F}_1^{\text{sp},b} = \mathbf{F}_1^{\text{sp}}, \mathbf{F}_N^{\text{sp},b} = -\mathbf{F}_N^{\text{sp}}. \quad (10)$$

The diffusion tensor  $\mathbf{D}_{ij}$  describes the effect of hydrodynamic interaction on bead  $i$  induced by a force acting on bead  $j$ . For BD simulations, hydrodynamic interaction is usually described by the Rotne-Prager-Yamakawa tensor,<sup>30,31</sup>

$$\mathbf{D}_{ii} = \mathbf{I}, \quad \mathbf{D}_{ij} = \frac{3a}{4R_{ij}} \begin{cases} \left[ \left(1 + \frac{2a^2}{3R_{ij}^2}\right) \mathbf{I} + \left(1 - \frac{2a^2}{R_{ij}^2}\right) \frac{\mathbf{R}_{ij}\mathbf{R}_{ij}}{R_{ij}^2} \right] & \text{for } R_{ij} \geq 2a \\ \left[ \frac{R_{ij}}{2a} \left[ \left(\frac{8}{3} - \frac{3R_{ij}}{4a}\right) \mathbf{I} + \frac{R_{ij}}{4a} \frac{\mathbf{R}_{ij}\mathbf{R}_{ij}}{R_{ij}^2} \right] \right] & \text{for } R_{ij} < 2a \end{cases}, \quad i \neq j, \quad (11)$$

where  $\mathbf{R}_{ij}$  is the vector between beads  $i$  and  $j$ , and  $R_{ij} = |\mathbf{R}_{ij}|$ . We notice that the hydrodynamic interaction parameter  $h^*$  is not used in the above expression because its definition is not applied to the bead-rod model which has no ‘‘spring length.’’

From the fluctuation-dissipation theorem, the weighting factor  $\boldsymbol{\sigma}_{ij}$  in Eq. (8) must satisfy the condition

$$\mathbf{D}_{ij} = \sum_{l=1}^N \boldsymbol{\sigma}_{il} \cdot \boldsymbol{\sigma}_{jl}^T. \quad (12)$$

For a given  $\mathbf{D}_{ij}$ ,  $\boldsymbol{\sigma}_{ij}$  is not unique. Most studies, including the current one, use the Cholesky decomposition, an  $O(N^3)$  scheme, to calculate it.<sup>32</sup> A faster spectral approximation method that evaluates the term of  $\boldsymbol{\sigma}_{ij} \cdot \mathbf{n}_j$  rather than  $\boldsymbol{\sigma}_{ij}$  has been proposed

by Fixman<sup>33</sup> and implemented by Jendrejack *et al.*<sup>34</sup> When beads rarely overlap, this approximation scheme roughly scales as  $N^{2.25}$  and is faster than the Cholesky decomposition. However, when beads overlap very frequently, the Cholesky decomposition may be more efficient.

The divergence of the diffusion tensor in Eq. (8) arises due to the use of Ito's interpretation while it will vanish if the Stratonovich interpretation is used instead. In many papers, this term is dropped because it is zero for the widely used Oseen and Rotne-Prager-Yamakawa tensors. However, when a different diffusion tensor is employed or a wall effect is present, this term cannot be neglected.

To prevent overstretching or overcompressing the springs, we employ an implicit predictor-corrector method originally designed for the bead-spring model with finitely extensible springs.<sup>32,35,36</sup> The key aspect of the predictor-corrector method used here is that for each spring length  $Q$  a cubic equation can be constructed that has only one real root between  $(1-s)$  and  $(1+s)$ . The FENE-Fraenkel spring law meets the above criterion so the predictor-corrector method can be applied.

Since the predictor-corrector method has been described in detail by Somasi *et al.*<sup>36</sup> for the free-draining case and by Hsieh *et al.*<sup>32</sup> for the non-free-draining case, we only briefly summarize the method here. To apply the predictor-corrector method, Eq. (8) has to be rewritten as a function of the spring vectors  $\mathbf{Q}$  rather than a function of positions of beads  $\mathbf{r}$ . The predictor step is an explicit Euler integration scheme which provides an initial guess. In corrector steps, spring lengths  $Q$  are treated implicitly so that they never become forbidden values. Thus, one obtains the following equation for  $\mathbf{Q}_i$

$$\left(1 + \frac{B(1-Q_i)}{[s^2 - (1-Q_i)^2]Q_i}\right)\mathbf{Q}_i = \mathbf{Y}_i, \quad (13)$$

where we define  $B = -2s^2H\Delta t$ ,  $\mathbf{Y}_i$  is actually  $\mathbf{R}_i$  in Eq. (3.15) of Hsieh *et al.*,<sup>32</sup> and  $Q_i$  is the magnitude of  $\mathbf{Q}_i$ . By taking the magnitude of each side of Eq. (13), one can reduce it to a set of third-order scalar polynomial equations

$$Q_i^3 - (2 + Y_i)Q_i^2 + (2Y_i - B + (1 - s^2))Q_i + (Y_i(s^2 - 1) + B) = 0. \quad (14)$$

This cubic equation can be easily solved and always has a unique real root between  $(1-s)$  and  $(1+s)$ . Once  $Q_i$  is obtained, the spring vector can be calculated from Eq. (13) by

$$\mathbf{Q}_i = \mathbf{Y}_i / \left(1 + \frac{B(1-Q_i)}{[s^2 - (1-Q_i)^2]Q_i}\right). \quad (15)$$

Although this method developed for the earlier bead-spring model can be applied to our new bead-FENE-Fraenkel-spring model, care should be taken because the prefactor of  $\mathbf{Q}_i$  in Eq. (13), in contrast to that for ordinary bead-FENE-spring model, is not always positive, and its sign cannot be determined before  $Q_i$  is solved. To guarantee that the correct root is obtained, one should check if  $\mathbf{Q}_i$  obtained from Eq. (15) is consistent with the value of  $Q_i$  obtained from Eq. (14). If not, one should re-solve Eq. (14) with  $Y_i$  replaced by  $-Y_i$  to obtain the correct value of  $Q_i$ . (Although without this

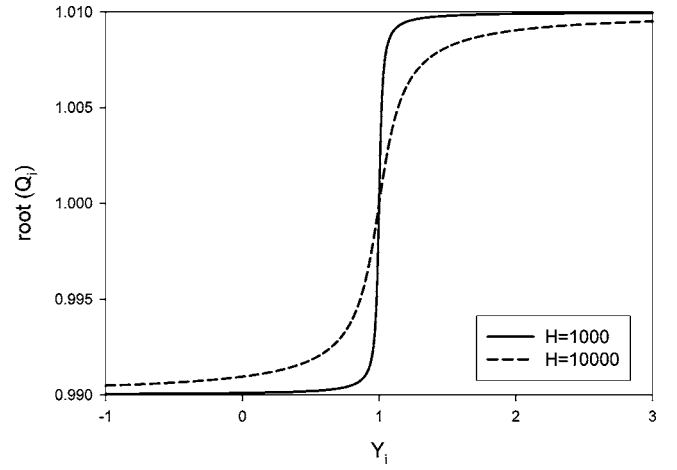


FIG. 2. The root  $Q_i$  of the cubic equation Eq. (14) as a function of the parameter  $Y_i$ , which is the magnitude of the right-hand side of Eq. (13).

check obtaining the wrong root is possible, this has never happened in our simulations.)

The corrector steps should be repeated until the residual of  $Q_i$  has reached a prespecified tolerance. The residual is calculated as the summation of the squares of the differences between the values of  $Q_i$  obtained from two consecutive iterations. A typical tolerance used in this study for a ten-spring chain is  $10^{-7}$ . Using too large a tolerance, such as  $10^{-5}$ , yields wrong equilibrium configurations.

In the above procedure, Eq. (14) is solved several times each time step so that a considerable amount of computer time can be saved if one can accelerate this scheme. Since  $s$  and  $B$  in Eq. (14) are constants for all the springs, any value of  $Y_i$  corresponds to a unique root ( $Q_i$ ). Figure 2 shows the root of Eq. (14) as a function of  $Y_i$  given  $s=0.01$  and  $\Delta t=0.001$ . Since the curve is very smooth, we can build a “lookup table” of roots of Eq. (14) for different values of  $Y_i$  and use interpolation to obtain approximate solutions instead of solving the cubic equations each time step.<sup>21,36</sup> Figure 2 also shows that the value of  $Q_i=|Q_i|$  is always bounded between  $(1-s)$  and  $(1+s)$ .

If hydrodynamic interaction is neglected, the diffusion tensor  $\mathbf{D}$  becomes isotropic and Eq. (8) can be reduced to

$$\frac{d\mathbf{r}_i}{dt} = \boldsymbol{\kappa} \cdot \mathbf{r}_i + \mathbf{F}_i^{\text{sp},b} + \left(\frac{6}{\Delta t}\right)^{1/2} \mathbf{n}_i. \quad (16)$$

One can solve  $\mathbf{Q}_i$  the same way as described above.

### C. Kramers bead-rod model

We will compare simulation results for our bead-FF-spring model with those of a Kramers bead-rod model. For the free-draining case, Liu's algorithm will be used to generate results of bead-rod model for comparison. We refer the readers to Liu<sup>15</sup> and Doyle *et al.*<sup>17</sup> for more details about the algorithm. All simulation results presented here for the free-draining case use ten-spring (rod) chains with a contour length of ten dimensionless units. We note that the definition of the tolerance for the bead-FF-spring model is different from that for the bead-rod model. For the bead-rod model, we use

$$\varepsilon = \sum_{i=1}^{N_s} (Q_i - 1)^2. \quad (17)$$

For ten-spring (rod) chains, the tolerance is set to  $10^{-7}$  for both models, if not specified otherwise.

Since developing an algorithm for the bead-rod model with hydrodynamic interaction is not trivial and also not the main goal of this study, we choose to compare our results for shear flow with HI to those obtained by Petera and Muthukumar,<sup>27</sup> who implemented Öttinger's algorithm.<sup>16</sup>

#### D. Evaluation of stress

For the bead-FENE-Fraenkel spring model, we calculate the polymer stresses  $\boldsymbol{\sigma}$  from

$$\boldsymbol{\sigma} = \nu \sum_{i=1}^{N_s} \langle \mathbf{F}_i^{\text{sp}} \mathbf{Q}_i \rangle. \quad (18)$$

For the bead-rod model, the stress is calculated using the Kramers-Kirkwood expression,

$$\boldsymbol{\sigma} = \nu \sum_{i=1}^{N+1} \langle \mathbf{R}_i \mathbf{F}_i^H \rangle, \quad (19)$$

where  $\nu$  is the number density of polymer chains and is set to unity in this study,  $\mathbf{R}_i$  is the position vector of bead  $i$  and  $\mathbf{F}_i^H$  is the hydrodynamic force acting on bead  $i$  which can be calculated as the negative summation of the Brownian and constraint forces.

In this study, no variance (or noise) reduction technique is used for either model. However, we point out that the variance reduction techniques developed for the regular bead-spring model, such as the control variate method<sup>35,37</sup> can be applied to the bead-FF-spring model without any modification.

#### E. Initial configuration and general setup

As will be shown later, the bead-FF-spring model yields a random-walk configuration at equilibrium, so we can easily generate initial equilibrium configurations. To obtain initial configurations for the bead-rod model, we start from a random-walk configuration and run the simulation under a no-flow condition. We discard the results obtained in the initial period of 200 relaxation times and then record the configuration of the polymer chain once every two relaxation times until enough configurations are obtained.

### III. RESULTS AND DISCUSSION

#### A. Choosing spring constant $H$ and extensibility parameter $s$

As mentioned above, the spring constant  $H$  does not have a physical meaning and the choice of the value of  $H$  can be solely based on minimizing computer time. To demonstrate this, we first test the influence of  $H$  on the equilibrium end-to-end distance squared for value of  $H$  ranging from 100 to 1 000 000 with  $s=0.01$  and time step  $\Delta t=0.001$ . The simulated equilibrium end-to-end distance squared is an average over a period of 100 000 relaxation times and deviates

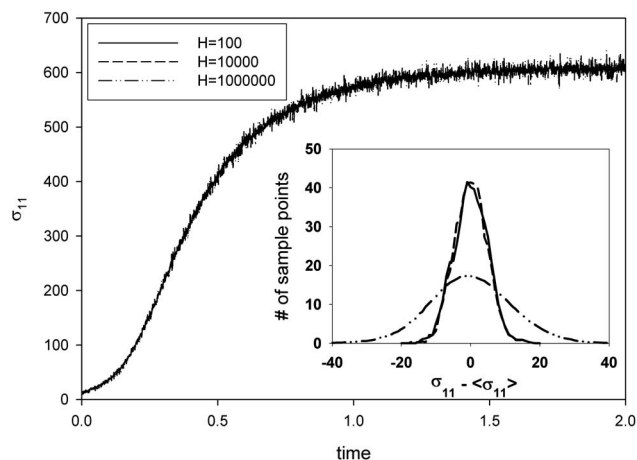


FIG. 3. Comparison of the average value of  $\sigma_{11}$  vs time in start-up of extensional flow for different values of  $H$  at Weissenberg number  $Wi=10$ . The inserted plot is a histogram of  $\sigma_{11,\text{plateau}} - \langle \sigma_{11,\text{plateau}} \rangle$ .

less than 0.3% from its expected value. Thus,  $H$  has negligible effect on the equilibrium end-to-end distance squared. We next test the influence of  $H$  on the polymer stresses in a start-up uniaxial extensional flow. In Fig. 3, we compare the simulated  $\sigma_{11}$  versus time in transient extensional flow with Weissenberg number  $Wi=10$  for different values of the spring constant  $H$ . The Weissenberg number  $Wi$  is defined as the product of the strain rate and the longest relaxation time of the chain. The relaxation time is evaluated using Eq. (20) that will be introduced in a later section. Each curve in Fig. 3 represents an ensemble average of 10 000 independent runs. The time step  $\Delta t$  and extensibility parameter  $s$  are set to  $10^{-4}$  and 0.01, respectively. Although different values of  $H$  are used, the simulation results are essentially the same. We then analyze the degrees of fluctuation of  $\sigma_{11}$  after it reached the steady state. The results are shown as a histogram of  $\sigma_{11,\text{plateau}} - \langle \sigma_{11,\text{plateau}} \rangle$  in the insert of Fig. 3. We find that  $\sigma_{11,\text{plateau}}$  is much noisier for simulations using  $H=10^6$  than for those using  $H=10^2$  and  $H=10^4$ . The latter two have essentially the same magnitude of noise. The computational costs are 6.5, 6.4, and 8.1 iterations per step for  $H=10^2$ ,  $10^4$ , and  $10^6$ , respectively. We thus confirm that any reasonable choice of  $H$  is safe while there is an optimum range of  $H$  that can keep the computational cost low. For the simulations in this study for a ten-spring chain, a rule of the thumb we find is that  $H=1000Wi$ . A more general method for choosing an appropriate value of  $H$  is to perform a test run. One can choose any value of  $H$  for the test run and compute the approximate average spring force  $\langle \mathbf{F} \rangle$ .  $H$  can then be chosen by setting  $H \sim 1.5 \langle \mathbf{F}^{\text{sp}} \rangle / s$  which implies that the average deviation in the spring length from its natural length (i.e., the rod length) is around  $0.5s$ .

The extensibility parameter  $s$  determines the maximum and minimum spring lengths. If  $s$  is very small, the spring becomes very stiff and a large number of iterations are required to achieve convergence. On the other hand, using a large value of  $s$  results in a large deviation in the spring length  $Q$  from the natural spring length. We test the influence of the extensibility parameter  $s$  using free-draining BD simulation of a chain in a start-up uniaxial extensional flow and

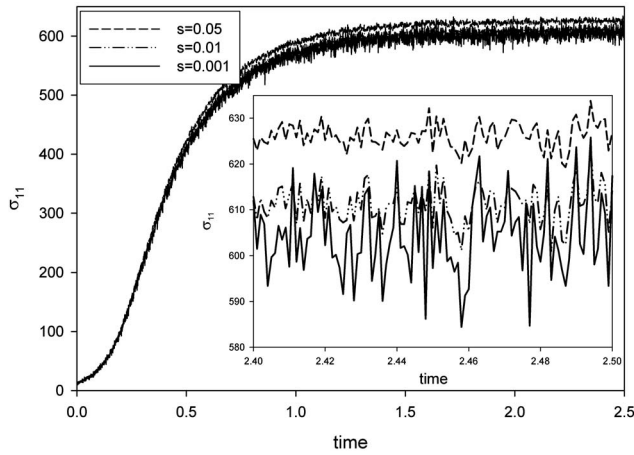


FIG. 4. Comparison of the average value of  $\sigma_{11}$  vs time in start-up extensional flow with different values of  $s$  at Weissenberg number  $Wi=10$ . The inserted plot is magnification of this plot over a small time window in the plateau region.

show the simulated  $\sigma_{11}$  versus time in Fig. 4. Each curve in Fig. 4 represents an ensemble average of 10 000 independent runs. The time step  $\Delta t$  and the spring constant  $H$  are set to be  $10^{-4}$  and  $10^4$ , respectively. The inserted plot is an enlarged view of the stress plateau. As can be seen, when  $s$  becomes smaller, the simulation results converge on average although noise increases for smaller  $s$ . The average iterations per step are 6.4, 7.6, and 12.0 for  $s=0.05$ ,  $s=0.01$ , and  $s=0.001$ , respectively. Although the simulation is fastest for  $s=0.05$ , this value leads to an inaccurate plateau value of  $\sigma_{11}$  due to the large deviation in the spring length from its natural length. We therefore use  $s=0.01$  for the following simulations because it provides an acceptable result at a reasonable computational cost.

## B. The bead-FENE-Fraenkel-spring model versus the bead-rod model in the free-draining limit

### 1. Equilibrium and relaxation

We first perform simulations at an equilibrium state to verify that the bead-FF-spring model yields a random-walk configuration. Figure 5 compares the simulation results of the bead-rod model and the bead-FF-spring model consisting of three beads and two rods with theoretical predictions for the included angle. The bead-FF-spring model yields a random-walk configuration while bead-rod model does not. Both models yield probability distributions in agreement with the theoretical predictions given by Eqs. (1) and (2).

Figure 6 plots the square of the stretch  $\langle x^2 \rangle$  simulated by the bead-rod model and the bead-FF-spring model versus time. We start from a fully stretched configuration and run the simulations under a no-flow condition until the chains are completely relaxed. Each curve is an ensemble average of 500 runs. As can be seen, the two curves almost overlap perfectly with each other. Therefore, we can infer that the two models give very similar relaxation times. For convenience, the longest relaxation time  $\tau_1$  for the bead-rod model in the free-draining limit will be estimated using the formula provided by Doyle *et al.*,<sup>17</sup>

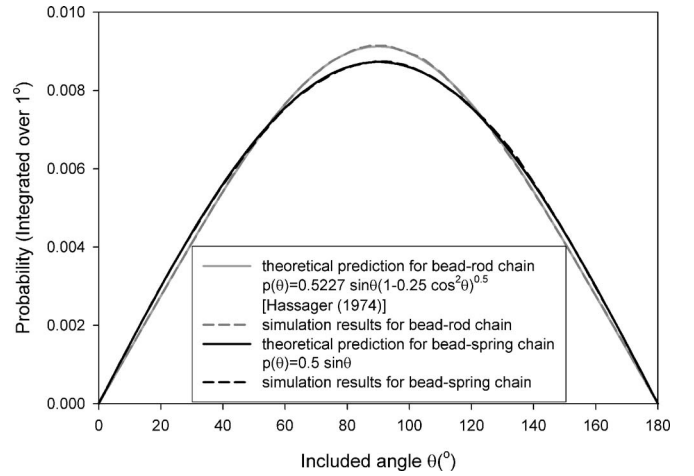


FIG. 5. The probability distribution for the included angle of a trumbell. Solid lines represent the theoretical predictions and dashed lines the simulation results. The gray lines were obtained using the bead-rod model and black lines using the bead-FF-spring model. The parameters for the bead-FF-spring model are  $s=0.01$ ,  $H=1000$ , and  $\Delta t=0.0001$ .

$$\tau_1 = 0.0142(N_s + 1)^2. \quad (20)$$

This will be used to calculate the Weissenberg numbers for both the bead-FF-spring and the bead-rod models in the following.

### 2. Start-up of uniaxial extensional flow

Figure 7 compares predictions of  $\sigma_{11}$  for the bead-rod model with the midpoint algorithm to that of the bead-FF-spring model with the predictor-corrector method after start-up of uniaxial extensional flow at  $Wi=10$ . Each curve is an average over 1000 individual runs. Only one time step  $\Delta t=0.001$  is used for the bead-rod model because it does not converge if a larger time step is employed. Actually, even with  $\Delta t=0.001$ , the bead-rod simulations often become unstable after ten dimensionless time units (or roughly 60 strain units). Therefore, this is the largest time step allowed for bead-rod model simulations with the specified tolerance. On

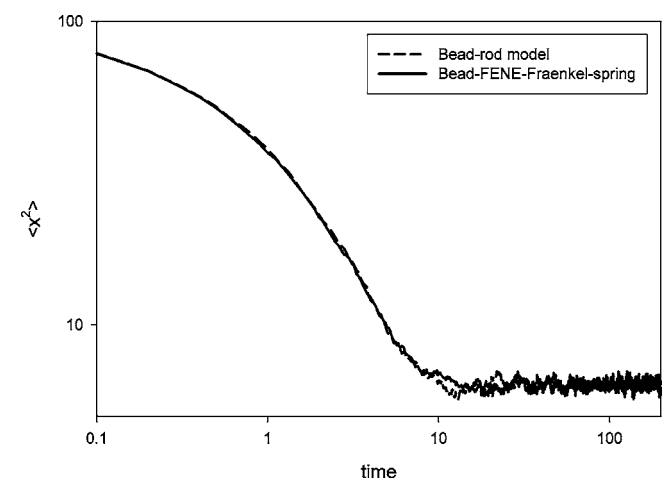


FIG. 6. Comparison of relaxation of stretch square for a ten-rod (spring) chain vs time for the bead-rod model with Liu's algorithm and the bead-FF-spring model with the predictor-corrector method. Each curve represents an ensemble of 500 independent runs. Hydrodynamic interaction is not included.

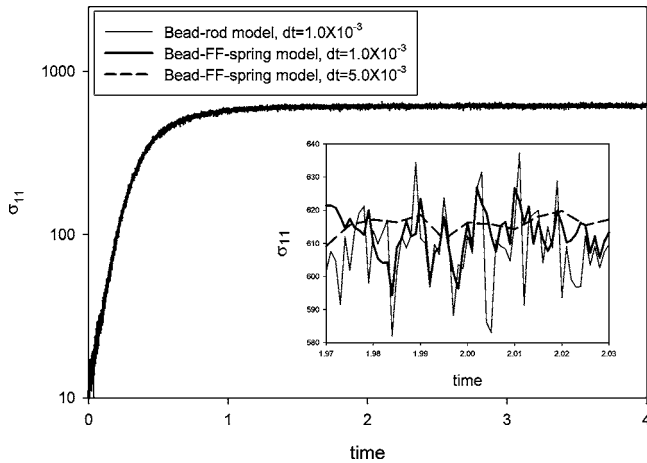


FIG. 7. Comparison of polymer contribution to  $\sigma_{11}$  vs time in start-up extensional flow with Weissenberg number  $Wi=10$  for the bead-rod model with Liu's algorithm and the bead-FF-spring model with the predictor-corrector method. Each curve is an average over an ensemble of 1000 runs. Hydrodynamic interaction is not included. The inserted plot shows an enlarged view of the fluctuation about the plateau value of  $\sigma_{11}$ .

the other hand, the bead-FF-spring model is very stable even with  $\Delta t=0.005$ . We therefore present results for both  $\Delta t=0.005$  and  $\Delta t=0.001$ . It can be observed that both methods predict a very similar value of  $\sigma_{11}$ . However, the bead-FF-spring model yields an ensemble average value of  $\sigma_{11}$  with smaller fluctuations (see the inserted plot of Fig. 7).

Figure 8 shows the histogram of the deviation between the instantaneous value of  $\sigma_{11,plateau}$  and its average value  $\langle \sigma_{11,plateau} \rangle$ . We assume that  $\sigma_{11}$  reaches its plateau after  $t=3$  (17.5 strain units) and collect data at every time step between  $t=3$  to  $t=4$  in each realization until  $2 \times 10^6$  data points are obtained. From Fig. 8, we observe two trends. First, using a larger time step effectively reduces the noise in the prediction of stresses for the bead-FF-spring model. Second, the bead-FF-model yields a narrower distribution of  $\sigma_{11,plateau} - \langle \sigma_{11,plateau} \rangle$  than does the bead-rod model when the same time step and tolerance are used.

The standard errors of the mean of these data are 0.267, 0.160, and 0.091 for the bead-rod model with  $\Delta t=0.001$ , the bead-FF-spring model with  $\Delta t=0.001$ , and the bead-FF-

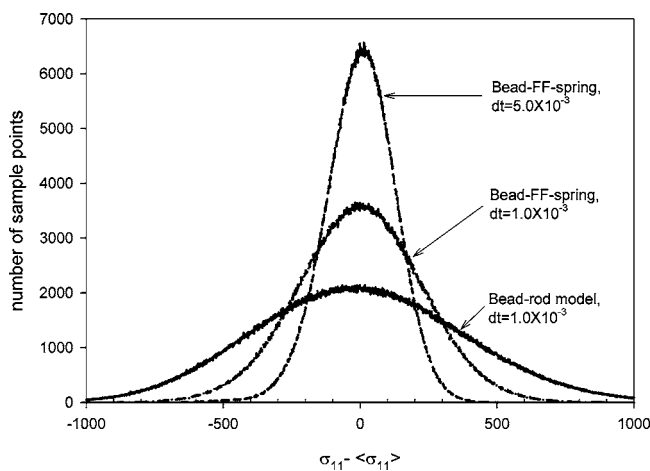


FIG. 8. Histogram of the deviation between the instantaneous value of  $\sigma_{11, plateau}$  and its average value  $\langle \sigma_{11, plateau} \rangle$ .

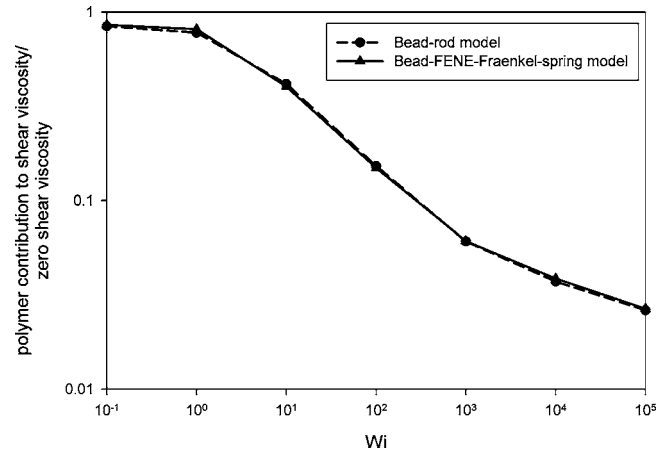


FIG. 9. Comparison of steady-state shear viscosity scaled with the zero-shear viscosity vs Weissenberg number in shear flow for the bead-rod model with Liu's algorithm and the bead-FENE-Fraenkel-spring model with the predictor-corrector method. Hydrodynamic interaction is not included.

spring model with  $\Delta t=0.005$ , respectively. The absolute values of the standard errors of the mean are very small because we collect a large set of data. Since the standard error of the mean decreases proportionally to the square root of the number of the samples, we can infer that only about one-third of realizations are needed for the bead-FF-spring model with time step  $\Delta t=0.001$  to obtain the results with the same accuracy as those of the bead-rod model provided that the time step and the tolerance are the same. Furthermore, because the bead-FF-spring model allows a larger time step,  $\Delta t=0.005$ , it needs about only one-eighth of realizations to obtain results with the same accuracy as those of the bead-rod model with  $\Delta t=0.001$ .

### 3. Steady shear flow

In Fig. 9, we compare the polymer contribution to the shear viscosity obtained using the bead-FF-spring model with that using the bead-rod model in steady shear flow versus the Weissenberg number  $Wi$  averaged over at least 20 000 strain units. The Weissenberg number for shearing flow is defined as the product of the longest relaxation time and the shear rate. The parameters are listed in Table I. We note that the simulation parameters are not optimized, but each result is verified using an additional set of simulations with a smaller tolerance and smaller time step. In shear flow, the bead-FF-spring model does not have the advantage of

TABLE I. Simulation parameters for the bead-rod model and the bead-FENE-Fraenkel-spring model in shear flows.

Wi	Bead-rod model		Bead-FENE-Fraenkel-spring model		
	$\Delta t$	$H$	$s$	$\Delta t$	
0.1	$10^{-3}$	$10^3$	0.01	$10^{-3}$	
1	$10^{-3}$	$10^3$	0.01	$10^{-3}$	
10	$10^{-3}$	$10^4$	0.01	$10^{-3}$	
100	$3 \times 10^{-4}$	$10^4$	0.01	$3 \times 10^{-4}$	
1 000	$10^{-4}$	$2 \times 10^4$	0.01	$10^{-4}$	
10 000	$10^{-5}$	$10^5$	0.01	$10^{-5}$	
100 000	$10^{-6}$	$10^6$	0.01	$10^{-6}$	



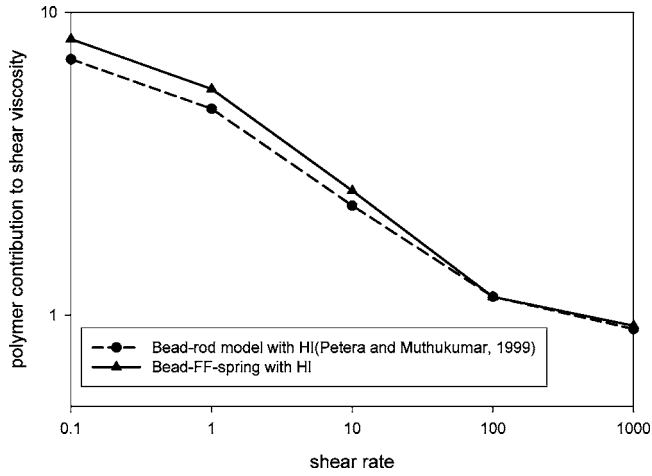


FIG. 10. Comparison of steady-state shear viscosity vs Weissenberg number in shear flow for the bead-rod model and the bead-FENE-Fraenkel-spring model. Hydrodynamic interaction is included.

allowing larger time steps. The shear viscosities shown in Fig. 9 are normalized using the zero-shear viscosity given by  $((N_s+1)^2-1)/36$ .<sup>8</sup> As can be seen, the two models once again generate very similar results. We also check the standard errors of the mean  $\sigma_M$  of the shear viscosities generated by both models and find that the ratio of  $\sigma_{M,\text{bead-FF-spring}}$  to  $\sigma_{M,\text{bead-rod}}$  is 0.9. Thus, the bead-FF-spring model predicts slightly less noisy shear stresses than does the bead-rod model.

### C. The bead-FENE-Fraenkel-spring model versus the bead-rod model with hydrodynamic interaction

Since developing a BD code for the bead-rod model with hydrodynamic interaction is not trivial and also is not the main interest of this study (in fact, part of our purpose in developing the FF spring is to avoid needing to develop a bead-rod model with HI), we choose to compare our simulation results with those of Peters and Muthukumar.<sup>27</sup> A 20-rod chain was used for their simulations in shear flow. The hydrodynamic radius  $a$  is taken from Peters and

Muthukumar<sup>27</sup> and is set to be 0.5. Figure 10 compares the predicted shear viscosities from both models. All results represent averages over at least 200 000 dimensionless time units. From Fig. 10, it is a little surprising that there seems to be a discrepancy between the predicted shear viscosities given that two models predict very similar results for the free-draining case. However, we need more data for comparison to determine if this discrepancy is generic or merely caused by inaccurate simulations. Since bead-rod simulations with HI are complicated to formulate and implement while with the FENE-Fraenkel spring, there is no added complication beyond that of a normal bead-spring chain, we believe that if a formulation or coding error is responsible for the difference in Fig. 10, the error is more likely to reside in the code for the bead-rod chain than in that for the bead-FF-spring chain.

### D. Efficiency of the bead-FF-spring model

For the free-draining bead-FF-spring model, the computational cost per time step scales roughly linearly with the number of springs  $N_s$  if the time step does not change with  $N_s$ . This scaling is the same as that for the bead-rod model with the midpoint algorithm, the SHAKE algorithm, or projection operators. Because of the iterative nature of the predictor-corrector method, the number of iterations differs in each step and also changes with the choice of parameters. Therefore, the scaling provided is only a rough guide, not a precise result.

Table II shows the computer times corresponding to Fig. 7 as well as those for simulating the equilibrium state. Each set of data represents the computer time for simulating a ten-spring (rod) chain over an ensemble of 2000 runs with 4000 steps in each run for those with  $\Delta t=0.001$  and with 800 steps for those with  $\Delta t=0.005$ . The programs were implemented in FORTRAN 90, compiled with PGI FORTRAN compiler 3.2, and run on an Intel Pentium III 1.0 GHz processor. From Table II, we can estimate that the computational cost per time step for a free-draining BD simulation with the proposed bead-FF-spring model is about twice as high as the

TABLE II. Comparison of computational time per time step for the bead-rod model with Liu's algorithm and the bead-FENE-Fraenkel model with the predictor-corrector method.

	At equilibrium			Start-up of extensional flow (Wi=10)		
	CPU time (s)	Number of iterations	Average error in rod length	CPU time (s)	Number of iterations	Average error in rod length
Bead-rod model with Liu's algorithm, $\Delta t=0.001$	93	5.32	$6.41E-4$	110	6.22	$6.84E-4$
Bead-FF-spring with predictor-corrector method $\Delta t=0.001$	187	6.15	$2.30E-3$	212	7.00	$4.14E-3$
Bead-FF-spring with predictor-corrector method $\Delta t=0.005$	43	7.07	$1.29E-3$	53	8.92	$4.12E-3$

traditional bead-rod model with the midpoint algorithm of Liu<sup>15</sup> for a given time step and tolerance. However, the bead-FF-spring model can be as efficient because it allows use of a larger time step and it yields less noisy results as demonstrated earlier.

If the hydrodynamic interaction is included, the computer time is dominated by the decomposition of the diffusion tensor  $\mathbf{D}$  which scales roughly as  $(N_s+1)^3$ . Therefore, the bead-FF-spring model should be faster for extensional flow because it allows larger time steps.

#### IV. CONCLUSIONS

We propose to use the bead-spring model with stiff FENE-Fraenkel springs as an alternative to the bead-rod model for simulating freely jointed chains in Brownian dynamics simulations. We developed a FENE-Fraenkel spring that is finitely extensible and has a natural length equal to the rod length in the bead-rod model. A predictor-corrector method originally developed for the bead-spring model was developed and tested. In the free-draining limit, we compared results from BD simulations of the bead-FENE-Fraenkel-spring model against those for the bead-rod model in relaxation, steady-state shear, and start-up of extensional flow and find that both methods generate very similar results. Both models are also comparably efficient in use of computer time. When hydrodynamic interaction is included, a slight discrepancy is observed between the predicted shear viscosities. However, more data are needed for a more detailed comparison.

Summarizing the above findings, we conclude that the bead-FENE-Fraenkel-spring model can be used as an alternative method to simulate a freely jointed chain. With this new model, existing computer codes for bead-spring models can trivially be converted to ones for effective bead-rod simulations merely by replacing the usual FENE or Cohen spring law with a FENE-Fraenkel law. This convertibility provides a very convenient way to perform multiscale BD simulations.

#### ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Grant No. NSF DMR-0072101. Any opinions, findings and conclusions or recommendations expressed in this material are those of the

authors and do not necessarily reflect the views of the National Science Foundation (NSF). The authors also thank Professor David D. Morse for helpful discussions.

- <sup>1</sup>P. J. Flory, *Statistical Mechanics of Chain Molecules* (Oxford University Press, New York, 1988).
- <sup>2</sup>P. T. Underhill and P. S. Doyle, *J. Rheol.* **49**, 963 (2005).
- <sup>3</sup>R. G. Larson, *J. Rheol.* **49**, 1 (2005).
- <sup>4</sup>E. S. G. Shaqfeh, *J. Non-Newtonian Fluid Mech.* **130**, 1 (2005).
- <sup>5</sup>N. J. Woo, E. S. G. Shaqfeh, and B. Khomami, *J. Rheol.* **48**, 281 (2004).
- <sup>6</sup>H. A. Kramers, *J. Chem. Phys.* **14**, 415 (1946).
- <sup>7</sup>J. M. Rallison, *J. Fluid Mech.* **93**, 251 (1979).
- <sup>8</sup>R. B. Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids* (Wiley-Interscience, New York, 1987), Vol. 2.
- <sup>9</sup>O. Hassager, *J. Chem. Phys.* **60**, 2111 (1974).
- <sup>10</sup>M. Gottlieb and R. B. Bird, *J. Chem. Phys.* **65**, 2467 (1976).
- <sup>11</sup>M. Fixman, *J. Chem. Phys.* **69**, 1527 (1978).
- <sup>12</sup>E. J. Hinch, *J. Fluid Mech.* **271**, 219 (1994).
- <sup>13</sup>A. Montesì, D. C. Morse, and M. Pasquali, *J. Chem. Phys.* **122**, 084903 (2005).
- <sup>14</sup>S. A. Allison and J. A. McCammon, *Biopolymers* **23**, 167 (1984).
- <sup>15</sup>W. T. Liu, *J. Chem. Phys.* **90**, 5826 (1989).
- <sup>16</sup>H.-C. Öttinger, *Phys. Rev. E* **50**, 2696 (1994).
- <sup>17</sup>P. S. Doyle, E. S. G. Shaqfeh, and A. P. Gast, *J. Fluid Mech.* **334**, 251 (1997).
- <sup>18</sup>J. S. Hur, E. S. G. Shaqfeh, and R. G. Larson, *J. Rheol.* **44**, 713 (2000).
- <sup>19</sup>J. P. Ryckaert, G. Ciccotti, and H. J. Berendsen, *J. Comput. Phys.* **23**, 327 (1977).
- <sup>20</sup>P. S. Grassia and E. J. Hinch, *J. Fluid Mech.* **282**, 373 (1995).
- <sup>21</sup>E. A. J. F. Peters, Ph.D. thesis, Delft University of Technology, 2000.
- <sup>22</sup>D. C. Morse, *Adv. Chem. Phys.* **128**, 65 (2004).
- <sup>23</sup>U. S. Agarwal, R. Bhargava, and R. A. Mashelkar, *J. Chem. Phys.* **108**, 1610 (1998).
- <sup>24</sup>U. S. Agarwal, *J. Chem. Phys.* **113**, 3397 (2000).
- <sup>25</sup>A. V. Lyulin, D. B. Adolf, and G. R. Davies, *J. Chem. Phys.* **111**, 758 (1999).
- <sup>26</sup>I. M. Neelov, D. B. Adolf, A. V. Lyulin, and G. R. Davies, *J. Chem. Phys.* **117**, 4030 (2002).
- <sup>27</sup>D. Petera and M. Muthukumar, *J. Chem. Phys.* **111**, 7614 (1999).
- <sup>28</sup>S. Liu, B. Ashok, and M. Muthukumar, *Polymer* **45**, 1383 (2004).
- <sup>29</sup>D. L. Ermak and J. A. McCammon, *J. Chem. Phys.* **69**, 1352 (1978).
- <sup>30</sup>J. Rotne and S. Prager, *J. Chem. Phys.* **50**, 4831 (1969).
- <sup>31</sup>H. Yamakawa, *J. Chem. Phys.* **51**, 436 (1970).
- <sup>32</sup>C.-C. Hsieh, L. Li, and R. G. Larson, *J. Non-Newtonian Fluid Mech.* **113**, 141 (2003).
- <sup>33</sup>M. Fixman, *Macromolecules* **19**, 1195 (1986).
- <sup>34</sup>R. M. Jendrejack, M. D. Graham, and J. J. de Pablo, *J. Chem. Phys.* **113**, 2894 (2000).
- <sup>35</sup>H.-C. Öttinger, *Stochastic Processes in Polymeric Fluids* (Springer, Berlin, 1996).
- <sup>36</sup>M. Somasi, B. Khomami, N. J. Woo, J. S. Hur, and E. S. G. Shaqfeh, *J. Non-Newtonian Fluid Mech.* **108**, 2275 (2002).
- <sup>37</sup>M. Herrchen and H.-C. Öttinger, *J. Non-Newtonian Fluid Mech.* **68**, 17 (1997).