An efficient algorithm for metric correction forces in simulations of linear polymers with constrained bond lengths

Matteo Pasquali[a] and David C. Morse[b]
Department of Chemical Engineering & Materials Science and University of Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, Minnesota 55455

(Received 17 August 2001; accepted 26 October 2001)

The dynamical behavior and equilibrium distribution of linear bead-rod and bead-spring polymers differ even in the limit of infinitely stiff springs. Imposing metric pseudopotential forces on the bead-rod chains yields the behavior of bead-spring chains in Langevin and Brownian Dynamics simulations. Here we present a simple, compact, and efficient algorithm for computing the required metric correction forces at minimal computational cost. © *2002 American Institute of Physics.* [DOI: 10.1063/1.1428747]

When designing a dynamical simulation of constrained classical mechanical system, one is faced with two mathematically similar, but conceptually distinct, sets of subtle questions, both of which arise from the existence of mechanical constraints: physical questions about the nature of the equilibrium state and mathematical questions about the formulation of simulation algorithms.

The first set of questions arises from the observation[1–6] that the thermal equilibrium distribution for a ''rigid'' classical mechanical system of point masses whose positions are subject to rigorous mechanical constraints generally differs from that of a corresponding ''stiff'' system, in which the constraints are approximately maintained by a stiff potential energy, even in the limit of an infinitely stiff potential. Specifically, the theoretically predicted equilibrium distribution of bond orientations for a linear bead-rod polymer with rigorously constrained bond lengths is different from that of a bead-spring chain in the limit of infinite spring stiffness. Infinitely stiff bead-spring models yield a simple Boltzmann distribution for the bond orientations, giving uncorrelated bond orientations when the chain is freely jointed, whereas models with rigidly constrained bond lengths lead to subtle correlations between bond orientations even for a freely jointed chain, whose detailed form depends on the relative bead masses.

The second set of questions arises in the analysis of algorithms for Brownian dynamics simulations of inertialess diffusion of systems with either variable mobility (including unconstrained systems with hydrodynamic interactions), or mechanical constraints, or both. In systems with a variable mobility but without constraints, two choices are possible to account properly for terms in the diffusion equation which involve derivatives of the mobility: A simple explicit Euler algorithm in which the random bead displacements are biased by adding derivatives of the mobility,[7] or a mid-step or predictor-corrector algorithm[8] that (in the absence of constraints) generates automatically the required terms. For constrained systems, Fixman[8] found that a mid-step algorithm for a polymer with constrained bond lengths and a potential energy that depends on the bond orientations would yield an equilibrium distribution that differs from the Boltzmann distribution of bond orientations expected for a bead-spring chain of stiff springs, and that happens to be equal to the one predicted by equilibrium statistical mechanics for bead-rod chains with equal bead masses. Fixman also noted, however, that a Brownian dynamics simulation of a chain with constrained bond lengths can be made to mimic that of a stiff bead-spring chain by adding an appropriate metric-pseudopotential to the potential energy from which the forces on the beads are derived. More generally, the predicted equilibrium distribution of an infinitely stiff bead-spring chain can be obtained from a constrained simulation, with or without inertia, by adding an appropriate metric pseudo-potential to the potential energy used in the simulation.

To design a simulation of linear polymers with constant or nearly constant bond lengths, one must choose both an underlying physical model (i.e., chains with rigidly constrained bond lengths or chains with stiff bonds), and a computational method (i.e., imposing constraints via Lagrange multipliers or using stiff potentials). To date, Brownian dynamics simulations of chains with constrained bond lengths have been carried out both without metric pseudopotential forces, for freely jointed chains without[9,10] and with hydrodynamic interactions,[11] and with the metric pseudo-potential forces, for free-draining freely jointed,[12] and wormlike[13,14] chains.

In practice, the choice of physical model and computational method must be based on a combination of considerations of physical realism and mathematical convenience. Algorithms which enforce rigorously constraints on bond length are generally computationally preferable to ones based on very stiff bonds, because they permit the use of longer time steps than those required to resolve rapid bond fluctuations. We believe, however, that there are both physical and mathematical reasons to prefer simulations that mimic the behavior of an infinitely stiff bead-spring chains. The question of which type of model is more physically realistic has been the subject of a long debate;[1–6] classical

[a]Current address: Department of Chemical Engineering, Rice University MS-362, 6100 Main St., Houston, TX 77005. Electronic mail: mp@rice.edu
[b]Electronic mail: morse@cems.umn.edu

bead-spring models appear to us to be somewhat less wrong[4,5] than classical bead-rod models in most applications. The mathematical advantages of the analysis of bead-spring models arise primarily from the simple analytic form of their equilibrium distribution of bond orientations—e.g., random orientations in a freely jointed chain. For example, in the analysis of the origins of stress in solutions of wormlike chains,[15] the ability to obtain a simple algebraic expression for the stress tensor, and a simple physical interpretation of its components, relied directly upon the use of a bead-spring model for the analysis. More generally, theoretical analysis and the analysis of simulations is often significantly simplified by using a model that has an algebraically simple equilibrium distribution.

Here we present a compact formula and efficient algorithm for calculating the metric pseudo-potential forces required in simulations of chains with constrained bond lengths to mimic the behavior of infinitely stiff bead-spring chains. An algorithm for these forces has been given previously by Grassia and Hinch[12] and used in simulations by Grassia and Hinch and by Everaers *et al.*[13] The computational cost of both algorithms scales linearly with the number of beads in a chain. The formula and algorithm presented here are, however, significantly simpler than those given by Grassia and Hinch, and yield efficient code.

The Langevin equations of motion of a linear polymer of N beads with positions $\mathbf{R}_1, \ldots, \mathbf{R}_N$, connected by $N-1$ rods of length a with unit tangent vectors $\mathbf{u}_i \equiv (\mathbf{R}_{i+1} - \mathbf{R}_i)/a$ are

$$m_i \frac{d^2\mathbf{R}_i}{dt^2} = \mathbf{F}_i^{\text{pot}} + \mathbf{F}_i^{\text{const}} + \mathbf{F}_i^{\text{met}} + \mathbf{F}_i^{\text{hydro}} + \mathbf{F}_i^{\text{rand}}, \tag{1}$$

for $i=1, \ldots, N$. Here, m_i is the mass of bead i, $\mathbf{F}_i^{\text{pot}} \equiv -\partial U/\partial \mathbf{R}_i$ is a force derived from a potential energy U ($U \equiv 0$ in a freely jointed chain)

$$\mathbf{F}_i^{\text{const}} = \mathbf{u}_i \lambda_i - \mathbf{u}_{i-1} \lambda_{i-1}, \tag{2}$$

is a constraint force, which depends upon bond tensions $\lambda_1, \ldots, \lambda_{N-1}$ that must be computed to maintain constant bond lengths,

$$\mathbf{F}_i^{\text{hydro}} = -\sum_{j=1}^{N} \boldsymbol{\zeta}_{ij} \cdot (\dot{\mathbf{R}}_j - \boldsymbol{\kappa} \cdot \mathbf{R}_j), \tag{3}$$

is the hydrodynamic drag force on a molecule in a flow field with a homogeneous velocity gradient $\nabla \boldsymbol{v} = \boldsymbol{\kappa}^T$, $\boldsymbol{\zeta}_{ij}$ is a friction tensor (e.g., either a local friction tensor or the Rotne–Prager tensor[16]), $\mathbf{F}_i^{\text{rand}}$ is a random force, and $\mathbf{F}_i^{\text{met}}$ is the metric force of primary interest here. The equations of motion used in inertialess Brownian dynamics simulations are obtained by setting the masses to zero; they are a set of first order stochastic differential equations in which the bead velocities appearing in $\mathbf{F}_i^{\text{hydro}}$ can be expressed in terms of other forces through Eq. (1).

Hereafter we consider the calculation of metric forces for either: (i) Inertialess Brownian dynamics simulations using a mid-step algorithm,[8,12,17,18] or (ii) inertial Langevin dynamics simulations of Eq. (1) with nonzero bead masses. In either case the metric forces can be expressed as[2,8,17]

$$\mathbf{F}_i^{\text{met}} \equiv -\frac{\partial U_{\text{met}}}{\partial \mathbf{R}_i}, \tag{4}$$

where

$$U_{\text{met}} \equiv \tfrac{1}{2} kT \ln(\det \mathsf{G}), \tag{5}$$

and G is a $(N-1)\times(N-1)$ tridiagonal matrix

$$\mathsf{G} = \begin{bmatrix} d_1 & c_2 & 0 & & \ddots & \\ c_2 & d_2 & c_3 & 0 & \ddots & \\ 0 & c_3 & \ddots & \ddots & 0 & \cdots \\ \ddots & 0 & \ddots & d_{N-3} & c_{N-2} & 0 \\ & \ddots & 0 & c_{N-2} & d_{N-2} & c_{N-1} \\ & & \ddots & 0 & c_{N-1} & d_{N-1} \end{bmatrix}. \tag{6}$$

In inertialess Brownian dynamics, G has a purely geometrical meaning; its diagonal elements are $d_i = 2$, and its off diagonal elements are

$$c_i = -\mathbf{u}_i \cdot \mathbf{u}_{i-1}, \tag{7}$$

i.e., the negative of the cosine of the angle between neighboring bonds that share bead i. In inertial Langevin dynamics, the elements of G depend on the bead masses

$$d_i = m_{i+1}^{-1} + m_i^{-1}, \tag{8}$$

$$c_i = -m_i^{-1} \mathbf{u}_i \cdot \mathbf{u}_{i-1}. \tag{9}$$

The inertialess case can be recovered from the inertial case by setting $m_i = 1$ for all beads. The metric forces obtained with inertialess and inertial dynamics coincide when the beads have identical masses ($m_i = m$) because the matrices G obtained in these two cases differ only by a multiplicative factor m, which does not affect derivatives of $\ln \det \mathsf{G}$.

Differentiating Eq. (5) with respect to the position of bead i and using Eqs. (6), (8), (9), and the formula for the derivative of the determinant of a tensor with respect to the tensor components[19] yields the expression of the metric forces

$$\begin{aligned} \mathbf{F}_k^{\text{met}} &= -\frac{1}{2} kT \frac{\partial \ln \det \mathsf{G}}{\partial \mathbf{R}_k} \\ &= -\frac{1}{2} kT \sum_{i,j=1}^{N-1} \frac{1}{\det \mathsf{G}} \frac{\partial \det \mathsf{G}}{\partial G_{ij}} \frac{\partial G_{ij}}{\partial \mathbf{R}_k} \\ &= -\frac{1}{2} kT \sum_{i,j=1}^{N-1} G_{ji}^{-1} \frac{\partial G_{ij}}{\partial \mathbf{R}_k} \\ &= kT \sum_{i=2}^{N-1} G_{i-1,i}^{-1} m_i^{-1} \frac{\partial (\mathbf{u}_i \cdot \mathbf{u}_{i-1})}{\partial \mathbf{R}_k}. \end{aligned} \tag{10}$$

Equation (10) can be evaluated by inverting G, at a cost of $\mathcal{O}(N^3)$. An $\mathcal{O}(N)$ (linear) efficient algorithm can be obtained by noting that only the super- or sub-diagonal elements of G^{-1} are required; they can be expressed in terms of cofactors of the super-diagonal elements of G by Cramer's rule

$$G_{i-1,i}^{-1} = \frac{\text{cof}\, G_{i,i-1}}{\det \mathsf{G}}, \tag{11}$$

where the cofactor $\mathrm{cof}\, G_{i-1,i}$ of a subdiagonal element is minus the determinant of the $(N-2)\times(N-2)$ matrix obtained by removing row i and column $i-1$ from G

$$\mathrm{cof}\, G_{i-1,i} \equiv -\det \begin{bmatrix} d_1 & c_2 & 0 & \dots & & & & & & & \\ c_2 & d_2 & c_3 & 0 & \dots & & & & & & \\ 0 & \ddots & \ddots & \ddots & 0 & \dots & & & & & \\ \dots & 0 & c_{i-3} & d_{i-3} & c_{i-2} & 0 & \dots & & & & \\ & \dots & 0 & c_{i-2} & d_{i-2} & 0 & \dots & & & & \\ & & \dots & 0 & c_{i-1} & c_i & 0 & \dots & & & \\ & & & \dots & 0 & c_{i+1} & d_{i+1} & c_{i+2} & 0 & \dots & \\ & & & & \dots & 0 & c_{i+2} & d_{i+2} & c_{i+3} & 0 & \dots \\ & & & & & \dots & 0 & \ddots & \ddots & \ddots & 0 \\ & & & & & & \dots & 0 & c_{N-2} & d_{N-2} & c_{N-1} \\ & & & & & & & \dots & 0 & c_{N-1} & d_{N-1} \end{bmatrix} \tag{12}$$

$$= -c_i \det \begin{bmatrix} d_1 & c_2 & 0 & \dots & & \\ c_2 & d_2 & c_3 & 0 & \dots & \\ 0 & \ddots & \ddots & \ddots & 0 & \dots \\ & \dots & 0 & c_{i-3} & d_{i-3} & c_{i-2} \\ & & \dots & 0 & c_{i-2} & d_{i-2} \end{bmatrix} \det \begin{bmatrix} d_{i+1} & c_{i+2} & 0 & \dots & & \\ c_{i+2} & d_{i+2} & c_{i+3} & 0 & \dots & \\ 0 & \ddots & \ddots & \ddots & 0 & \dots \\ & \dots & 0 & c_{N-2} & d_{N-2} & c_{N-1} \\ & & \dots & 0 & c_{N-1} & d_{N-1} \end{bmatrix}. \tag{13}$$

Equation (13) is obtained by expanding the determinant of the large matrix along the row containing c_i. When $i>2$, this row contains only the two nonzero elements c_i and c_{i-1}, whereas when $i=2$ it contains only the single nonzero element c_i. The cofactor of c_{i-1} vanishes for all $i>2$, leaving only the term arising from c_i and its cofactor. In the generic case $(3<i<N-1)$, the cofactor of c_{i-1} in Eq. (12) is the determinant of an $(N-3)\times(N-3)$ matrix that contains two rectangular (and thus rank-deficient) blocks of nonzero elements: A top left block containing $i-2$ rows with nonzero elements in the first $i-3$ columns of the original matrix, whose rows cannot be linearly independent, and a bottom right block containing $N-i$ columns with nonzero elements in the last $N-1-i$ columns, whose columns cannot be linearly independent. In the special case $i=3$, the cofactor of $c_{i-1}=c_2$ in Eq. (12) is the determinant of an $(N-3)\times(N-3)$ matrix containing all zeroes in its first row, yielding a zero determinant. In the remaining case $i=N-1$, the cofactor of $c_{i-1}=c_{N-2}$ is the determinant of a matrix whose last column contains all zeroes.

For all $i=2,\dots,N-1$, the determinant of the $(N-2)\times(N-2)$ matrix shown in Eq. (12) is thus given by c_i times the cofactor of c_i in that matrix. The cofactor of c_i in this matrix is given for $2<i<N-1$, by the determinant of an $(N-3)\times(N-3)$ block diagonal matrix with a top left block containing the intersection of rows and columns $1,\dots,i-2$ of the original matrix G and a bottom right block containing rows and columns $i+1,\dots,N-1$ of G [these are the two matrices in Eq. (13)]. In more compact notation

$$G^{-1}_{i-1,i} = -c_i \frac{\det \mathsf{T}^{i-2} \det \mathsf{B}^{i+1}}{\det \mathsf{G}}, \tag{14}$$

where T^j denotes the top left $j\times j$ square matrix containing rows and columns $1,\dots,j$ of G, and B^j is the bottom right $(N-j)\times(N-j)$ square matrix containing rows and columns $j,\dots,N-1$ of G, with the convention $\det \mathsf{T}^0 = \det \mathsf{B}^N = 1$. The full matrix G is given by $\mathsf{G}=\mathsf{T}^{N-1}=\mathsf{B}^1$.

The determinants of the top left and bottom right matrices T and B can be calculated efficiently with a simple recursion relation[2]

$$\det \mathsf{T}^{i+1} = d_{i+1} \det \mathsf{T}^i - (c_{i+1})^2 \det \mathsf{T}^{i-1}, \tag{15}$$

$$\det \mathsf{B}^{j-1} = d_{j-1} \det \mathsf{B}^{\,j} - (c_j)^2 \det \mathsf{B}^{\,j+1}, \tag{16}$$

where $i=1,\dots,N-2$ and $j=2,\dots,N-1$.

At each time step, the metric forces can be computed by calculating the determinants of all of the submatrices T and B with the recursion relations[20] (15) and (16), then using Eqs. (10) and (14) to calculate the metric forces. This algorithm is the main result of this paper.

Equation (10) for the metric forces is nearly identical to the formula of the bending forces $\mathbf{F}_k^{\mathrm{pot}}$ arising from the bending energy

$$U = -\frac{\kappa}{a} \sum_{i=2}^{N-1} \mathbf{u}_i \cdot \mathbf{u}_{i-1}, \tag{17}$$

of a discrete wormlike chain with bending rigidity κ, viz.

$$\mathbf{F}_k^{\mathrm{pot}} \equiv -\frac{\partial U}{\partial \mathbf{R}_k} = \frac{\kappa}{a} \sum_{i=2}^{N-1} \frac{\partial(\mathbf{u}_i \cdot \mathbf{u}_{i-1})}{\partial \mathbf{R}_k}. \tag{18}$$

For such linear wormlike chains, the bending and metric forces are calculated most effectively by first computing

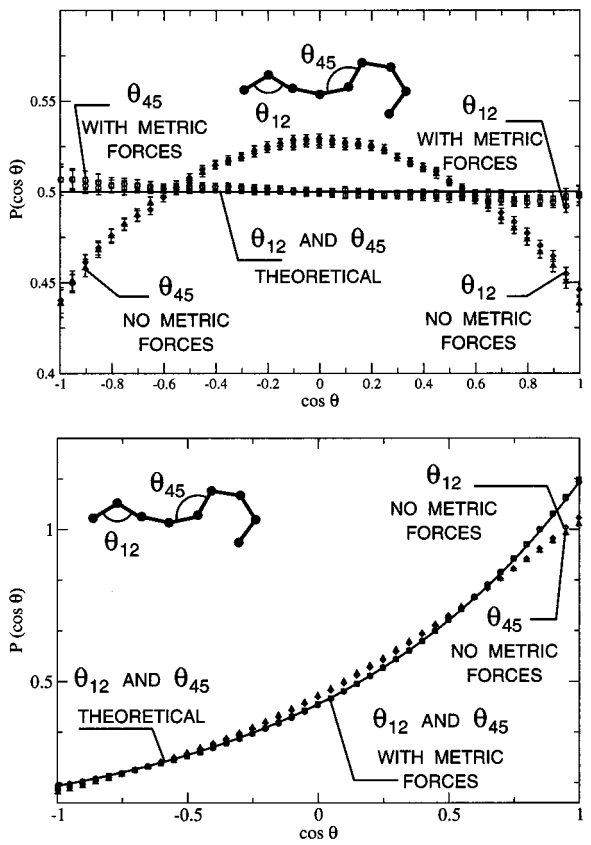

FIG. 1. Distribution of cosine of angles between rods 1 and 2 and rods 4 and 5 in a free-draining chain of 9 beads, for $\kappa=0$ (top) and $\kappa/(akT)=1$ (bottom). The solid line denotes the normalized theoretical prediction $P(\cos\theta) \propto e^{\kappa\cos\theta/(akT)}$. The symbols were computed by averaging the configuration of 400 free-draining molecules for ~14 times the rotational diffusion time of a multi-bead rod of equal length, or 7×10^6 time steps. Initial configurations of the molecules were generated by sampling the theoretical distribution, then letting the system equilibrate for ~3 rotational diffusion times before collecting data.

$G^{-1}_{i-1,i}$ with the recursion relations (15), (16), and Eq. (14), and then by computing the bending and metric forces together, as

$$\mathbf{F}_k^{\text{pot}}+\mathbf{F}_k^{\text{met}}=\frac{1}{a}\sum_{i=2}^{N-1}\kappa_i^{\text{eff}}\frac{\partial(\mathbf{u}_i\cdot\mathbf{u}_{i-1})}{\partial\mathbf{R}_k}, \tag{19}$$

by replacing the true bending rigidity κ with a conformation-dependent effective rigidity

$$\kappa_i^{\text{eff}}\equiv\kappa+kTa\,m_i^{-1}G^{-1}_{i-1,i} \tag{20}$$

for the joint connecting bonds i and $i-1$ at bead i, In a freely jointed chain, of course, $\kappa=0$. The derivative with respect to $\mathbf{R}_k$ in Eq. (19) can be evaluated with the identity

$$\frac{\partial}{\partial\mathbf{R}_k}\mathbf{u}_i=\frac{1}{a}(\delta_{k,i+1}-\delta_{ki})(\mathbf{I}-\mathbf{u}_i\mathbf{u}_i), \tag{21}$$

which is easily derived by differentiating $\mathbf{u}_i\equiv(\mathbf{R}_{i+1}-\mathbf{R}_i)/|\mathbf{R}_{i+1}-\mathbf{R}_i|$ with respect to $\mathbf{R}_k$, and setting $|\mathbf{R}_{i+1}-\mathbf{R}_i|=a$ after evaluating the derivative.

We have confirmed in Brownian dynamics simulations of free-draining wormlike chains, with a mid-step algorithm,[12] that this new algorithm for the metric forces (as that of Grassia and Hinch) yields the expected equilibrium distribution of angles between neighboring bonds in a bead-spring polymer: We obtain uncorrelated bond orientations for a freely jointed chains, with $\kappa=0$, and a Boltzmann distribution $P(\cos\theta_i)\propto e^{\kappa\cos\theta_i/(akT)}$ of the cosine $\cos\theta_i=\mathbf{u}_i\cdot\mathbf{u}_{i-1}$ of the angle between neighboring bonds when $\kappa\neq0$ (Fig. 1, bottom). When the metric forces are neglected, the distribution deviates measurably from this prediction (Fig. 1, top), and, for $N=3$, it agrees with the theoretical prediction for a trimer with beads of equal masses.[6]

In our simulations, computing the metric forces in this way adds only 15% computation time to a mid-step algorithm for a free-draining bead-rod (Kramers) chain without bending forces—see Appendix for details. The marginal cost of the metric forces is even lower in simulations of free-draining wormlike chains, because the metric forces are evaluated with the bending forces at virtually no extra computational cost other than the time to calculate the subdeterminants. The $\mathcal{O}(N)$ cost of calculating the metric forces would become completely negligible in any simulation with hydrodynamic interactions. Because the procedural complexity of our algorithm is also minimal, there is no longer any real penalty for including the metric forces in simulations of linear polymers with constrained bond lengths, if one has a reason to prefer a constrained simulation that mimics the dynamics of infinitely stiff, rather than rigid, polymers.

ACKNOWLEDGMENTS

This work was supported by NSF DMR-9973976 and the Minnesota Supercomputing Institute. We also thank Shriram Ramanathan for help with the test calculations and the figure.

APPENDIX A: COMPUTATIONAL DETAILS

The inertialess equations of motion of a free-draining, wormlike chain were integrated with a midstep algorithm[12] and the procedure and formulas for the metric and bending forces [Eqs. (14)–(16) and (19)–(21)] described in this article. The tensions were computed by solving by fast LU factorization without pivoting[21] of the symmetric, diagonally dominant matrix G; uniformly distributed random numbers were generated with a Tausworthe long-period generator.[22] The algorithm was programmed in Fortran 90, compiled and optimized with IBM's XL Fortran 7.1, and run on an IBM SP with Power3-II 375 MHz processors running IBM AIX 4.3. The optimized code performed 10 000 time steps for 100 Kramers chains of 128 beads in 119 s. Computing the metric forces required additional 20 s, i.e., 17% more CPU time. When bending forces were added, the simulation time was 123 and 139 s without and with metric forces, respectively, i.e., the metric forces required 13% extra computational effort.

[1] H. A. Kramers, J. Chem. Phys. **14**, 415 (1946).
[2] M. Fixman, Proc. Natl. Acad. Sci. U.S.A. **74**, 3050 (1974).
[3] N. Go and H. A. Scheraga, Macromolecules **9**, 535 (1976).
[4] E. Helfand, J. Chem. Phys. **71**, 5000 (1979).
[5] J. M. Rallison, J. Fluid Mech. **93**, 251 (1979).
[6] R. Bird, C. Curtiss, R. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids*, Vol. 2 (Wiley, New York, 1987).
[7] D. L. Ermak and J. McCammon, J. Chem. Phys. **69**, 1352 (1978).

[8] M. Fixman, J. Chem. Phys. **69**, 1527 (1978).
[9] T. W. Liu, J. Chem. Phys. **90**, 5826 (1989).
[10] P. S. Doyle, E. S. G. Shaqfeh, and A. P. Gast, J. Fluid Mech. **334**, 251 (1997).
[11] D. Peters and M. Muthukumar, J. Chem. Phys. **111**, 7614 (1999).
[12] P. S. Grassia and E. J. Hinch, J. Fluid Mech. **308**, 255 (1996).
[13] R. Everaers, F. Julicher, A. Ajdari, and A. C. Maggs, Phys. Rev. Lett. **82**, 3717 (1999).
[14] M. Pasquali, V. Shankar, and D. C. Morse, Phys. Rev. E **64**, 020802(R) (2001).
[15] D. C. Morse, Macromolecules **31**, 7030 (1998).
[16] J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[17] E. J. Hinch, J. Fluid Mech. **271**, 219 (1994).
[18] P. S. Grassia, E. J. Hinch, and L. C. Nitsche, J. Fluid Mech. **282**, 373 (1995).
[19] C. Truesdell and W. Noll, *The Non-Linear Field Theories of Mechanics*, *Encyclopedia of Physics* (Springer Verlag, Berlin, 1965), Vol. III/3, p. 26.
[20] If hydrodynamic interactions are absent and the beads have equal masses, then the matrix of the linear system defining the tensions is proportional to G, and it is symmetric positive definite; thus, G must be LU-factorized without pivoting to compute the tensions. In this case, the faster recursion relation $\det \mathsf{T}^{i+1} = U_{i+1} \det \mathsf{T}^{i}$ should be used to compute the top subdeterminants, where U_{i+1} are the diagonal elements of the U factor.
[21] J. J. Lambiotte and R. G. Voigt, ACM Trans. Math. Softw. **1**, 308 (1975).
[22] R. C. Tausworthe, Math. Comput. **19**, 201 (1965).