Finite-element approach to Brownian dynamics of polymers

Christian J. Cyron and Wolfgang A. Wall*
Lehrstuhl für Numerische Mechanik, Technische Universität München, Boltzmannstr. 15, D-85748 Garching, Germany
(Received 10 August 2009; revised manuscript received 11 November 2009; published 9 December 2009)

In the last decades simulation tools for Brownian dynamics of polymers have attracted more and more interest. Such simulation tools have been applied to a large variety of problems and accelerated the scientific progress significantly. However, the currently most frequently used explicit bead models exhibit severe limitations, especially with respect to time step size, the necessity of artificial constraints and the lack of a sound mathematical foundation. Here we present a framework for simulations of Brownian polymer dynamics based on the finite-element method. This approach allows simulating a wide range of physical phenomena at a highly attractive computational cost on the basis of a far-developed mathematical background.

DOI: 10.1103/PhysRevE.80.066704 PACS number(s): 02.70.Dh, 36.20.−r, 83.10.Mj

I. INTRODUCTION

For several decades Brownian dynamics of polymers has been a field attracting increasing interest. In many cases progress in active research areas such as material science, macromolecular chemistry, bioengineering, and biophysics is intrinsically tied to the understanding of the dynamics of polymers under thermal forces. Often the complexity of the systems under study makes computer simulations the preferred, sometimes even the only scientific tool. Since the 1970s simulation of Brownian polymer dynamics predominantly resorts to so-called bead-spring or bead-rod models. Although these models have successfully been used to deal with a broad variety of problems, e.g., [1–3], their limitations are well known and have become a serious obstacle on the way to further progress.

Classical explicit bead-spring simulations exhibit severe difficulties in dealing with stiff or semiflexible polymers due to their explicit time integration scheme and the resulting tremendous reduction of time step size. To overcome this problem bead-rod models were developed where the highest eigenfrequencies were cancelled out of the system by artificially imposed rigid constraints. However, these bead-rod models mainly suffer from two drawbacks: on the one hand these models cannot be applied straightforward [4] as the artificial constraints would tamper with the simulated dynamics. Intricate extensions are needed [5] to circumvent this difficulty. On the other hand bead-rod models cannot account for any axial strain making a large variety of phenomena plain inaccessible for these models. Furthermore all bead-based models have problems in accounting for the full range of mechanical phenomena present in polymer physics. For example, shear deformation, as it may arise in thick polymers or polymer bundles [6], cannot be captured in a coarse-grained fashion by bead models. In addition to that, modeling polymers as a chain of beads entails severe difficulties in accounting correctly for anisotropic friction of polymers [7]. Finally, both the method and the quality of the solutions gained by bead models lack any careful mathematical analysis. This complicates using simulations in a predictive sense separate from concomitant experimental and analytical work significantly.

There exist a number of key issues especially in biophysics and bioengineering, which could not yet be addressed by proper simulation tools, especially in the field of cell division, cell motility, and mechanotransduction. The above-mentioned shortcomings of state-of-the-art Brownian dynamics simulation tools may be considered to be among the main reasons for this deficiency of simulation technology.

From the above arguments we may conclude that there is a significant need for a new framework for Brownian dynamics simulations in polymer physics. In this paper we introduce such a new framework: we demonstrate how the finite-element method can be applied to Brownian polymer dynamics. In opposition to the above described simulation techniques this method can be derived from a clearly defined set of assumptions in a strictly mathematical manner. As a consequence it allows for establishing rigorous theorems about the quality of the numerical results. The finite-element method is well known to be capable of accounting for all relevant mechanical effects in one-dimensional continua. Therefore, it has already been applied to polymer mechanics [8–12]. However, previous studies have always been restricted to static elasticity or at the most to deterministic dynamics of polymers and polymer networks. None of these studies has included stochastic mechanics of polymers so far. However, it is generally accepted that the stochastic mechanics of polymers may not be neglected for the quantitative and often even qualitative understanding of polymer mechanics. In this paper we describe how to make use of finite elements for the spatial discretization of Brownian polymer dynamics. Together with an implicit time integration scheme this forms an ideal framework for large scale simulations of even complete polymer networks such as studied experimentally in [13]. On the basis of several examples we demonstrate the method's efficiency and reliability. The finite-element method is a *de facto* standard in many fields of computational engineering. For the present work an in-house finite-element code was employed. However, an extension of commonly available finite-element codes with the aim of capturing Brownian dynamics as described in this paper is to the authors' estimation straightforward. This represents an essential advantage over other Brownian dynamics simula-

*wall@lnm.mw.tum.de

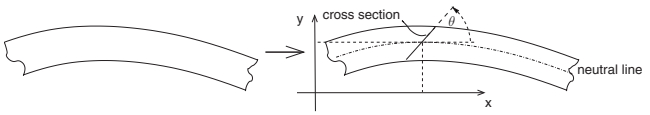

FIG. 1. A one-dimensional mechanical continuum (left) may be modeled according to Reissner's theory (right).

tion tools which often can be used after long and tedious in-house coding only.

This paper is organized as follows: in Sec. II we describe the mathematical model of Brownian polymer dynamics used in this paper. As the finite-element method has not yet been applied to Brownian dynamics, in Sec. III we give a brief introduction into the basics of the finite-element method for the reader not yet familiar with this method. Finally, in Sec. IV we present a finite-element formulation of the mathematical model of Sec. II. Throughout this paper we will usually employ lower case letters for continuous functions and variables whereas we will use upper case letters for discrete variables and functions.

II. BROWNIAN POLYMER DYNAMICS

In this section we describe a mathematical model for Brownian polymer dynamics. As common in the theoretical analysis of polymer dynamics [11,14,15], we model polymers on a coarse-grained scale as one-dimensional beamlike continua. For simplicity of this presentation we restrict ourselves to the consideration of the two-dimensional case in the x-y plane. However, an extension to three dimensions is straight forward and already realized in our research code.

A. Static polymer model

There is rich theoretical work on beamlike continua. Rather simplified models such as the Euler-Bernoulli model account for axial stress and bending only assuming a zero shear deformation. Although for many polymers the Euler-Bernoulli model is often a good approximation due to their large slenderness ratio, for thick filaments and especially bundles one may have to account for shear deformation, too [6]. Therefore, we resort to Reissner's beam theory, which is well known to allow for axial stress, bending, and shear in a geometrically exact manner. The basis of that theory is the kinematic assumption that a beamlike continuum can be modeled as a curve in space, to each point on which a plain cross section is assigned (cf. Fig. 1). The curve itself is called the neutral line. Let $\xi \in [0;L]$ be the curve parameter with curve length L in space and let $\Omega=\{[x(\xi)y(\xi)]^T:0\leq\xi\leq L\}$ be the set of all points on the neutral line. Note that in this case the Jacobi determinant of the map between parameter space and physical space is one. Let $\theta(\xi)$ be the angle of the cross section in space at the point $[x(\xi)y(\xi)]^T$ on the neutral line. In the classical Euler-Bernoulli model this cross section is assumed to remain constantly orthogonal to the neutral line as a zero shear deformation is assumed. Therefore within an Euler-Bernoulli model the orientation of the cross section in space has not to be described explicitly by an angle $\theta(\xi)$ but is rather given implicitly by the shape of the neutral line itself. In opposition to that Reissner's beam theory allows for a rotation of the cross section relative to the neutral line due to shear deformation. According to Reissner's model the continuum in some current configuration may be described completely by vectors $\boldsymbol{x}(\xi)=[x(\xi)y(\xi)\theta(\xi)]^T$. With some known stress-free reference configuration given by $\boldsymbol{X}(\xi)=[X(\xi)Y(\xi)\Theta(\xi)]^T$, the continuum can be described by the displacement vector $\boldsymbol{u}(\xi)$ between the reference configuration and the current configuration with

$$\boldsymbol{u}(\xi) = (\Delta x(\xi), \Delta y(\xi), \Delta\theta(\xi))^T. \tag{1}$$

The method of sections [16] reveals the internal elastic forces N_{int} orthogonal to the cross section and Q_{int} parallel to the cross section as well as the internal moment M_{int} as crucial for the elasticity of beams. According to [17] these quantities are given for a stress-free reference configuration by

$$\begin{pmatrix} N_{int} \\ Q_{int} \\ M_{int} \end{pmatrix} = \begin{pmatrix} EA & 0 & 0 \\ 0 & GA & 0 \\ 0 & 0 & EI \end{pmatrix} \left[\boldsymbol{T}^T(\boldsymbol{x}) \frac{d\boldsymbol{x}}{d\xi} - \boldsymbol{T}^T(\boldsymbol{X}) \frac{d\boldsymbol{X}}{d\xi} \right], \tag{2}$$

$$\boldsymbol{T}(\boldsymbol{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

Here the area of the beam cross section is denoted by A, Young's modulus of the beam material by E, and the shear modulus by G. The products EA, GA, and EI denote the stretching, shear, and bending stiffness, respectively. The triad $\boldsymbol{T}(\boldsymbol{x})$ is a rotation matrix rotating from a local coordinate system which is attached to the beam cross section into the global x-y coordinate system. Let the external forces per unit length in x and y directions be given by $f_{ext,x}$ and $f_{ext,y}$ and the external moment per unit length by m_{ext}. Then the static elasticity of the beam can be described according to [16] by

$$\boldsymbol{f}_{el}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{f}_{ext}(\boldsymbol{x}), \tag{4}$$

$$\boldsymbol{f}_{el}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x}) \left(\frac{dN_{int}}{d\xi} \quad \frac{dQ_{int}}{d\xi} \quad \frac{dM_{int}}{d\xi} - Q_{int} \right)^T, \tag{5}$$

$$\boldsymbol{f}_{ext}(\boldsymbol{x}) = (f_{ext,x} \quad f_{ext,y} \quad m_{ext})^T. \tag{6}$$

B. Dynamic polymer model

The static model of Sec. II A can be extended to cover also dynamics. The field of Brownian polymer dynamics is mostly concerned with problems where the equation of motion of a polymer is determined by elastic forces $\boldsymbol{f}_{el}(\boldsymbol{u},\boldsymbol{x})$ due to structural deformation, hydrodynamic forces $\boldsymbol{f}_{visc}(\boldsymbol{u},\dot{\boldsymbol{u}},\boldsymbol{x})$ due to viscous damping by a surrounding fluid, and some external forces. The external forces acting on the polymer may be considered as the sum of the two forces $\boldsymbol{f}_{ext}(\boldsymbol{x})$ and $\boldsymbol{f}_{stoch}(\boldsymbol{x})$. Here $\boldsymbol{f}_{stoch}(\boldsymbol{x})$ represents the stochastic thermodynamic forces causing Brownian motion. The variable $\boldsymbol{f}_{ext}(\boldsymbol{x})$ just collects all other forces, e.g., those caused by some force field which the polymer is subject to. Inertia can be neglected

usually in Brownian polymer dynamics. Therefore Eq. (4) can be generalized in the dynamic case by

$$\boldsymbol{f}_{el}(\boldsymbol{u},\boldsymbol{x}) + \boldsymbol{f}_{visc}(\boldsymbol{u},\dot{\boldsymbol{u}},\boldsymbol{x}) = \boldsymbol{f}_{ext}(\boldsymbol{x}) + \boldsymbol{f}_{stoch}(\boldsymbol{x}). \tag{7}$$

All variables in the above equation should be interpreted as forces or moments per unit length. A brief remark as to the notation seems worthwhile at this point: Eq. (7) has to be satisfied at each point in time. Therefore all quantities in Eq. (7) depend in general on the time t. For simplicity we will omit the parameter t whenever it is not essential for understanding the following equations. Yet we implicitly assume that all the following equations always refer to some specific point in time.

In order to explicitly determine the friction force $\boldsymbol{f}_{visc}(\boldsymbol{u},\dot{\boldsymbol{u}},\mathbf{x})$ we need to agree on a friction model first. Indeed several different friction models have been employed so far in polymer physics. Simple models are based on just one single friction coefficient [3]. More intricate ones account for hydrodynamic interactions by means of different friction coefficients parallel and perpendicular to the polymer backbone [2,7] or by a consideration of the Navier-Stokes equations' characteristics [18,19]. As this paper represents a mere introduction into the simulation of Brownian polymer dynamics by means of finite elements, we restrict ourselves to the simple model of homogeneous isotropic friction coefficients ζ entailing velocity proportional damping forces. In this case the friction force per length is given by

$$\boldsymbol{f}_{visc}(\dot{\boldsymbol{u}}) = \boldsymbol{c}\dot{\boldsymbol{u}}, \quad \boldsymbol{c} = \begin{pmatrix} \zeta & 0 & 0 \\ 0 & \zeta & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{8}$$

Note that in Eq. (8) we assume that the velocity of the polymer backbone is given by the velocity of the neutral line in the Reissner beam model which is a suitable approximation for slender continua such as polymers. It is underlined that an isotropic friction model makes all spurious drift terms vanish and allows for abstaining from any discussion of the Itō-Stratonovich dilemma. Yet we would like to briefly mention that a careful analysis reveals that even for a more realistic anisotropic friction model the here proposed framework can be applied without difficulties.

Studying Brownian dynamics of polymers, a typical choice for ζ would be setting $\zeta = 4\pi\eta$ with fluid viscosity η and adding some correction factor [20]. In Brownian dynamics viscous drag forces always go along with stochastic thermal forces $\boldsymbol{f}_{stoch}$ according to the fluctuation-dissipation theorem. These forces are usually modeled as Gaussian random variables with zero correlation time, mean value zero, and a mean square depending on temperature and friction coefficients. Precisely [14], they are described by

$$\langle \boldsymbol{f}_{stoch}(\boldsymbol{x},t) \rangle = 0, \tag{9a}$$

$$\langle \boldsymbol{f}_{stoch}(\boldsymbol{x},t) \otimes \boldsymbol{f}_{stoch}(\boldsymbol{x}',t') \rangle = 2k_B T \boldsymbol{c}\, \delta_{tt'} \delta_{\boldsymbol{x}\boldsymbol{x}'}, \tag{9b}$$

where t, t' and $\boldsymbol{x}$, $\boldsymbol{x}'$ represent in general different points in time and the polymer continuum, respectively, and $\delta_{tt'}$ is the Dirac-function with argument $t-t'$. The temperature of the fluid is denoted by T, the Boltzmann constant is denoted by k_B, and a mean value is denoted by $\langle . \rangle$. Thus, the covariance matrix of the thermal forces equals the scaled damping matrix $\boldsymbol{c}$.

In the end of this section it seems worthwhile dropping a brief remark about Eq. (9): in Eq. (9) we model thermal forces as Gaussian random variables. This means that even for an arbitrary large constant value, there is a nonzero probability that it is exceeded by the amplitude of the thermal forces at least one point in time. In reality, of course, we may assume that there is a certain sharp limit for the forces and that no larger forces arise. This subtlety does not matter much as we are talking about events of extremely small likelihood and thus of minor importance for the numerical suitability of the approach. However, one should keep this detail in mind for Sec. IV B 2. There we will use weighted integrals in space over the stochastic line load and derive a respective finite-element formulation. A corner stone for convergence theorems in the finite-element theory is Céa's lemma which can be established only on the basis of square integrable right-hand side terms in Eq. (9) [21]. Hence, both for a realistic model and for making rigorous mathematical statements quasi-Gaussian random variables with a cut-off far from the expected value should be employed. Those readers taking a strict point of view may substitute in mind all the Gaussian random variables in this paper by such quasi-Gaussian random variables.

III. BASICS OF THE FINITE-ELEMENT METHOD

The finite-element method is a rather general method for solving partial differential equations. As it has not yet been applied to the Brownian dynamics of polymers, we give a brief introduction into the basics of finite elements for the reader not yet familiar with this method. We abstain from mathematical discussions for which the reader is referred to text books such as [21–23]. Rather we restrict ourselves to the essential ideas and take a merely descriptive point of view, leaving a series of intricacies completely out of discussion.

A. Strong form

On the domain $\Omega \subset \mathbb{R}^d$ an n-vector valued function $\boldsymbol{u}(\boldsymbol{x})$ is defined implicitly by the nonlinear partial differential equation

$$\boldsymbol{f}_{int}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{f}_{ext}(\boldsymbol{x}). \tag{10}$$

Here $\boldsymbol{f}_{int}(.)$ is an n-vector valued differential operator, which is in general nonlinear. The function $\boldsymbol{f}_{ext}(\boldsymbol{x})$ may be some inhomogeneity. Of course, for the differential Eq. (10) boundary conditions have to be specified. However, as the issue of boundary conditions is not essential for our applications, we just neglect them in this brief introduction as we will do for the rest of this paper. Usually Eq. (10) is referred to as the strong form of a partial differential equation. In principle an arbitrary partial differential equation may be written in form (10) and the differential operator $\boldsymbol{f}_{int}$ and inhomogeneity $\boldsymbol{f}_{ext}$ have to be assigned the respective physical interpretation to. In the context of this paper $\boldsymbol{f}_{int}$ and $\boldsymbol{f}_{ext}$

may be associated with the left and right-hand side of Eq. (7), respectively.

B. Weak form

It can be shown [23] that this strong form is completely equivalent to the so-called weighted residual formulation. This formulation is gained by multiplying Eq. (10) by some so-called weighting function $w(\boldsymbol{x})$ and integrating over the whole domain

$$\int_{\Omega} w(\boldsymbol{x})\boldsymbol{f}_{int}(\boldsymbol{u},\boldsymbol{x})d\Omega = \int_{\Omega} w(\boldsymbol{x})\boldsymbol{f}_{ext}(\boldsymbol{x})d\Omega. \tag{11}$$

It is important that Eqs. (10) and (11) are only completely equivalent if we demand Eq. (11) not to be satisfied for one specific function $w(\boldsymbol{x})$ only but for any function $w(\boldsymbol{x})$ out of a sufficiently large function space $\mathcal{V}$. Integrating Eq. (11) by parts, one arrives at the so-called weak form.

C. Discretization and Galerkin approximation

The domain Ω with boundary Γ may be of arbitrary shape. We set up a so-called discretization by subdividing the domain into n_{el} so-called elements $\Omega_{(e)}$, which form a nonoverlapping decomposition of the domain. Furthermore we define m points $\boldsymbol{x}_b \subset \Omega$, $1 \leq b \leq m$, which are referred to as nodes. In principle these nodes may be situated anywhere on the domain. The m_e nodes situated either in the interior of the element $\Omega_{(e)}$ or on its boundary are denoted by $\boldsymbol{x}_{a(e)}$, $1 \leq a \leq m_e$. The crucial step from Eq. (11) toward a finite-element formulation is the so-called Galerkin approximation: we assume that the solution function $\boldsymbol{u}(\boldsymbol{x})$ is not some arbitrary function. Rather we assign on each element $\Omega_{(e)}$ to each element node $\boldsymbol{x}_{a(e)}$ a nodal value $\boldsymbol{U}_{a(e)}$, whose n components are unknown at first. Defining some polynomial interpolation functions $N_{a(e)}(\boldsymbol{x})$, the solution itself is assumed to be representable on each element by

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{a=1}^{m_e} N_{a(e)}(\boldsymbol{x})\boldsymbol{U}_{a(e)}, \quad \forall\ \boldsymbol{x} \in \Omega_{(e)}. \tag{12}$$

Defining

$$\boldsymbol{U}_{(e)} = [\boldsymbol{U}_{1(e)}^T \cdots \boldsymbol{U}_{m_e(e)}^T]^T, \tag{13}$$

$$\boldsymbol{N}_{(e)} = \underbrace{\begin{bmatrix} N_{1(e)} & 0 & \ddots & \cdots & N_{m_e(e)} & 0 & \ddots \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ \ddots & 0 & N_{1(e)} & \cdots & \ddots & 0 & N_{m_e(e)} \end{bmatrix}}_{n \times (nm)} \tag{14}$$

we can rewrite Eq. (12) as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{N}_{(e)}(\boldsymbol{x})\boldsymbol{U}_{(e)}, \quad \forall\ \boldsymbol{x} \in \Omega_{(e)}. \tag{15}$$

As we interpolated the solution function, we also interpolate the geometry elementwise between the nodes by means of the same polynomial functions $N_{a(e)}(\boldsymbol{x})$. Therefore, actually not the exact domain Ω with arbitrary boundary Γ is considered in a standard finite-element model but rather a domain with interpolated boundary. In slight abuse of notation we denote this approximated domain still by Ω.

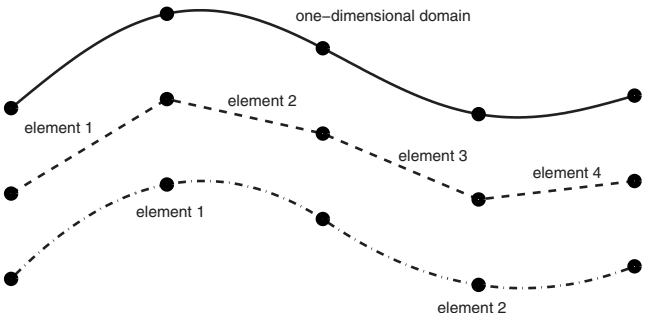

FIG. 2. From top to bottom: one-dimensional domain Ω (continued line) embedded in $\mathbb{R}^2$ discretized by nodes (black dots). Between the nodes the original domain may be interpolated, e.g., linearly (dashed line) or quadratically (dot-dashed line).

To illustrate the above definitions we consider the example of a polymer confined to plane motion by means of two plates. Such a polymer may be modeled by a one-dimensional domain Ω, which is a curve embedded in the two-dimensional space $\mathbb{R}^2$. We may place an arbitrary number of nodes $\boldsymbol{x}_b$ on that curve and interpolate it between the nodes by polynomial functions. Using polynomial interpolation of order p, $m_e = p+1$ nodes are assigned to each element. As the domain, also each single element is in general a curved line segment embedded in $\mathbb{R}^2$. The discretized domain is depicted in Fig. 2 for linear and quadratic interpolation between the nodes. For those readers familiar with common Brownian dynamics simulation models we would like to briefly mention that at least in some sense the nodes play a similar role as the beads in a bead-spring or bead-rod model. At this point already one of the crucial advantages of the finite-element method compared to classical bead models becomes evident: bead models approximate the geometry of a polymer usually by linearly connecting points in space. In opposition to that, a finite-element model can capture a virtually curved geometry between the nodes thereby allowing for a much more realistic representation of a fluctuating polymer.

A corner stone of the finite-element method is switching between an element perspective and a global perspective whenever convenient. We may refer to a specific node in an element $\Omega_{(e)}$ either in elementwise notation as $\boldsymbol{x}_{a(e)}$ or in global notation as $\boldsymbol{x}_b$, where both notations refer to one and the same node. Formally this relation may be described via a map $\Phi(\boldsymbol{x}_{a(e)}) = \boldsymbol{x}_b$. When, for example, numbering the nodes both globally and locally from the left to the right, one would refer to the rightmost node in the linear discretization in Fig. 2 in global notation as $\boldsymbol{x}_5$ and in elementwise notation as $\boldsymbol{x}_{2(4)}$. In analogy to the nodes themselves one can define global counterparts $\boldsymbol{U}_b$ of the element nodal values $\boldsymbol{U}_{a(e)}$ and concatenate them to the global nodal value vector $\boldsymbol{U}$ similarly to Eq. (13). Furthermore, on a global level we can define for each node $\boldsymbol{x}_b$ a global basis function

$$N_b(\boldsymbol{x}) = \begin{cases} N_{a(e)}(\boldsymbol{x}) & \text{if } \exists\{a,e\}:\Phi(\boldsymbol{x}_{a(e)}) = \boldsymbol{x}_b, \boldsymbol{x} \in \Omega_{(e)} \\ 0 & \text{else.} \end{cases} \tag{16}$$

Thus, if a point $\boldsymbol{x}$ lies within some element $\Omega_{(e)}$ within which the global node $\boldsymbol{x}_b$ has some element counterpart $\boldsymbol{x}_{a(e)}$, the global basis function $N_b(\boldsymbol{x})$ takes the same value as does $N_{a(e)}(\boldsymbol{x})$. If the point $\boldsymbol{x}$ lies outside any element within which the global node corresponds to a local one, the global shape function becomes zero. By means of Eq. (16) one can define a global $n \times nm$ basis function matrix $\boldsymbol{N}(\boldsymbol{x})$ in complete analogy to Eq. (14) and rewrite Eq. (15) as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x})\boldsymbol{U}, \quad \forall\, \boldsymbol{x} \in \Omega. \tag{17}$$

As we assumed for the solution function a certain kind of structure in Eq. (12), we require that also the weighting functions $w(\boldsymbol{x})$ in Eq. (11) are not arbitrary functions. Rather we admit as weighing functions only the global interpolation functions $N_b(\boldsymbol{x})$. By these assumptions Eq. (11) may be rewritten as

$$\int_\Omega N_b(\boldsymbol{x})\boldsymbol{f}_{int}[\boldsymbol{N}(\boldsymbol{x})\boldsymbol{U}]d\Omega = \int_\Omega N_b(\boldsymbol{x})\boldsymbol{f}_{ext}(\boldsymbol{x})d\Omega. \tag{18}$$

Looping through all admitted weighting functions $N_b(\boldsymbol{x})$, Eq. (18) leads to a system of mn scalar equations for the nm so far unknown elements of the global nodal value vector $\boldsymbol{U}$. This system of equations can be written as

$$\boldsymbol{F}_{int} = \boldsymbol{F}_{ext}, \tag{19}$$

with

$$\boldsymbol{F}_{int} = \int_\Omega \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{f}_{int}[\boldsymbol{N}(\boldsymbol{x})\boldsymbol{U}]d\Omega, \tag{20}$$

$$\boldsymbol{F}_{ext} = \int_\Omega \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{f}_{ext}(\boldsymbol{x})d\Omega. \tag{21}$$

Equation (19) can be interpreted as an equilibrium between a discrete internal and external force vector $\boldsymbol{F}_{int}$ and $\boldsymbol{F}_{ext}$, respectively. The in general nonlinear equation system (19) can be solved, e.g., by means of Newton-Raphson iterations. Therefore we need to evaluate the discrete force vectors $\boldsymbol{F}_{int}$, $\boldsymbol{F}_{ext}$ as well as the derivative

$$\boldsymbol{K} = \frac{\partial \boldsymbol{F}_{int}}{\partial \boldsymbol{U}}. \tag{22}$$

In structural mechanics the matrix $\boldsymbol{K}$ is usually called the stiffness matrix. As the quantities $\boldsymbol{F}_{int}$, $\boldsymbol{F}_{ext}$, and $\boldsymbol{K}$ are integrals over the whole domain Ω and the elements $\Omega_{(e)}$ form a nonoverlapping decomposition of Ω, we may compute the contributions of each element to the integrals $\boldsymbol{F}_{int}$, $\boldsymbol{F}_{ext}$, and $\boldsymbol{K}$ separately. The contributions on element level are the so-called element force vectors $\boldsymbol{F}_{int(e)}$, $\boldsymbol{F}_{ext(e)}$, and element stiffness matrices $\boldsymbol{K}_{(e)}$ defined by

$$\boldsymbol{F}_{int(e)} = \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^T(\boldsymbol{x})\boldsymbol{f}_{int}[\boldsymbol{N}_{(e)}(\boldsymbol{x})\boldsymbol{U}_{(e)}]d\Omega, \tag{23}$$

$$\boldsymbol{F}_{ext(e)} = \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^T(\boldsymbol{x})\boldsymbol{f}_{ext}(\boldsymbol{x})d\Omega, \tag{24}$$

$$\boldsymbol{K}_{(e)} = \frac{\partial \boldsymbol{F}_{int(e)}}{\partial \boldsymbol{U}_{(e)}}. \tag{25}$$

The above introduction to the basics of the finite-element method can be summed up as follows: any nonlinear partial differential Eq. (10) can be written in weak form. Discretizing this formulation by polynomial interpolation leads to a nonlinear system of algebraic Eqs. (19). This system of equations can be solved by iterative evaluation of the element force vectors and element stiffness matrices given in Eqs. (23)–(25).

In the end of this brief introduction into the finite-element method we would like to mention some important facts: as shown above the finite-element method is a numerical method which can be derived in a rigorous mathematical manner from an in principle arbitrary nonlinear partial differential equation. This sound theoretical foundation entails two important consequences. First, applying the finite-element method the set of assumptions underlying to the simulation method is completely transparent: essentially it is the same set used to formulate the partial differential Eq. (10). Second rigorous statements about the properties of the method (accuracy, convergence, stability,…) can be derived. This is in striking contrast to the simulation models applied so far to Brownian dynamics: neither for bead-spring nor for bead-rod models any such clear statements about the simulation results or assumptions underlying to the model have been presented so far.

IV. FINITE-ELEMENT APPROACH TO BROWNIAN POLYMER DYNAMICS

To numerically simulate the phenomenon of Brownian polymer dynamics described in Sec. II one needs to introduce a discretization both in space and time. The finite-element method as introduced in Sec. III is only used for a discretization in space here so that we have to perform a discretization in time separately.

A. Discretization in time

For the discretization in time Brownian dynamics simulations usually resort to the so called explicit Euler scheme. Although this time integration scheme goes along with low computational cost per time step, it suffers crucially from numerical stability problems. To alleviate this drawback several strategies have been worked out [2,7]. These strategies typically try to modify the discretization in space in order to gain numerical stability in the time integration. In the simulation framework presented in this paper we pursue a completely different approach. Mainly two reasons may be identified why explicit time integration is that popular for bead models despite its severe stability problems: on one hand there are several pitfalls both with respect to the theory and the numerics of implicit stochastic time integration. On the

other hand the implementation of implicit schemes is not as straightforward as of explicit schemes. Concerning the here presented finite-element model, the theoretical and numerical issues of implicit time integration are comprehensively discussed and resolved in [24]. As explained in the end of Sec. IV B 2, the implementation is easily possible on the basis of already existing finite-element codes. Hence, in this paper we apply the implicit backward Euler time integration scheme, thereby avoiding any numerical stability problem. Discretizing time by steps of length Δt allows writing the velocity $\dot{\boldsymbol{u}}(\boldsymbol{x})$ as

$$\dot{\boldsymbol{u}}^{i+1}(\boldsymbol{x}) = \frac{\boldsymbol{u}^{i+1}(\boldsymbol{x}) - \boldsymbol{u}^{i}(\boldsymbol{x})}{\Delta t}. \tag{26}$$

Here an upper index i means that a variable is taken at the point in time $t^i := i\Delta t$. Modeling time by discrete steps Δt requires rewriting Eq. (9) using a time discrete Dirac function so that we arrive at

$$\langle \boldsymbol{f}_{stoch}(\boldsymbol{x}, t^i) \rangle = 0, \tag{27a}$$

$$\langle \boldsymbol{f}_{stoch}(\boldsymbol{x}, t^i) \otimes \boldsymbol{f}_{stoch}(\boldsymbol{x}', t^j) \rangle = \frac{2k_B T}{\Delta t} \boldsymbol{c}\, \delta_{ij} \delta_{\boldsymbol{x}\boldsymbol{x}'}. \tag{27b}$$

Here δ_{ij} is a Kronecker-δ function. In Eq. (27) we implicitly assume that the stochastic forces are constant during each time step, respectively. In simulation methods where not stochastic forces are applied but rather Brownian displacements in each time step, these are computed neglecting the variation in the noise term during the time step. Turning the equation of motion Eq. (7) into the equivalent stochastic partial differential equation reveals that this common way of applying Brownian displacements is equivalent to the assumption of stochastic forces constant over each time step, respectively. We briefly would like to mention in the end of this section that for a finite-element approach an explicit time integration scheme such as used for classical bead-spring models could be employed as well. The decision for the implicit scheme is just based on the fact that in polymer physics implicit time integration is often more efficient in terms of computational cost.

B. Discretization in space

In Eqs. (19)–(21) an equilibrium between discrete force vectors $\boldsymbol{F}_{int}$ and $\boldsymbol{F}_{ext}$ was shown to lead to an approximated solution of Eq. (10). Therefore for the partial differential Eq. (7) a finite-element formulation amounts to an equilibrium

$$\boldsymbol{F}_{el} + \boldsymbol{F}_{visc} = \boldsymbol{F}_{ext} + \boldsymbol{F}_{stoch}. \tag{28}$$

Here $\boldsymbol{F}_{el}$ and $\boldsymbol{F}_{visc}$ can be derived from $\boldsymbol{f}_{el}$ in Eq. (5) and $\boldsymbol{f}_{visc}$ in Eq. (8) in the same way $\boldsymbol{F}_{int}$ in Eq. (20) has been derived from $\boldsymbol{f}_{int}$ in Eq. (10). Similarly, the external force vectors $\boldsymbol{F}_{ext}$ and $\boldsymbol{F}_{stoch}$ can be derived from $\boldsymbol{f}_{ext}$ in Eq. (6) and $\boldsymbol{f}_{stoch}$ in Eq. (9) as $\boldsymbol{F}_{ext}$ in Eq. (21) was derived from $\boldsymbol{f}_{ext}$ in Eq. (10). As described in Sec. III the finite-element solution just requires the evaluation of the respective element force vectors $\boldsymbol{F}_{el(e)}$, $\boldsymbol{F}_{visc(e)}$, $\boldsymbol{F}_{stoch(e)}$, $\boldsymbol{F}_{ext(e)}$, and element stiffness matrices $\boldsymbol{K}_{el(e)}$ and $\boldsymbol{K}_{visc(e)}$. In the following subsections we describe in greater detail how to evaluate these quantities. To describe the dynamics of a beamlike continuum as introduced in Sec. II A the solution function (1) has to be approximated according to Eq. (17) by means of nodal values

$$\boldsymbol{U}_b = (\Delta x_b, \Delta y_b, \Delta \theta_b)^T. \tag{29}$$

The first two coordinates describe the displacement of the node $\boldsymbol{x}_b$ in space. Each node represents a point on the neutral line of the continuum. The third coordinate $\Delta \theta_b$ gives the difference in orientation of the cross section between the current configuration and the reference configuration.

1. Static polymer model

Static finite-element formulations for beamlike continua have long been studied in computational engineering. Thus there exists rich literature about how to evaluate the elastic and external element force vectors and stiffness matrices $\boldsymbol{F}_{el(e)}$, $\boldsymbol{F}_{ext(e)}$ and $\boldsymbol{K}_{el(e)}$ following from Reissner's beam theory. For a detailed description the reader is referred to [17].

2. Dynamic polymer model

Although finite-element simulations have already been applied to polymer networks several times [8–12] these simulations were always limited to either the static case or merely deterministic dynamics. In opposition to that, in this section we demonstrate how to capture also Brownian dynamics by means of a finite-element model. Therefore we will describe in detail how to evaluate the quantities $\boldsymbol{F}_{visc(e)}$, $\boldsymbol{F}_{stoch(e)}$ and $\boldsymbol{K}_{visc(e)}$.

Using Eqs. (15) and (26) a discretization of the velocity both in space and time is given by

$$\dot{\boldsymbol{u}}^{i+1}(\boldsymbol{x}) = \boldsymbol{N}_{(e)}(\boldsymbol{x}) \frac{\boldsymbol{U}_{(e)}^{i+1} - \boldsymbol{U}_{(e)}^{i}}{\Delta t}, \quad \boldsymbol{x} \in \Omega_{(e)}. \tag{30}$$

Making use of Eqs. (8), (23), and (30), we can define an element damping force vector $\boldsymbol{F}_{visc(e)}$ by

$$\begin{aligned} \boldsymbol{F}_{visc(e)} &= \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^T(\boldsymbol{x}) \boldsymbol{f}_{visc} d\Omega \\ &= \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^T(\boldsymbol{x}) \boldsymbol{c} \boldsymbol{N}_{(e)}(\boldsymbol{x}) \frac{\boldsymbol{U}_{(e)}^{i+1} - \boldsymbol{U}_{(e)}^{i}}{\Delta t} d\Omega. \end{aligned} \tag{31}$$

Defining the element nodal velocity vector $\dot{\boldsymbol{U}}_{(e)}(\boldsymbol{x})$ as the derivative of $\boldsymbol{U}_{(e)}$ with respect to time and the element damping matrix

$$\boldsymbol{C}_{(e)} = \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^T(\boldsymbol{x}) \boldsymbol{c} \boldsymbol{N}_{(e)}(\boldsymbol{x}) d\Omega, \tag{32}$$

we may rewrite Eq. (31) by means of Eq. (32) as

$$\boldsymbol{F}_{visc(e)} = \boldsymbol{C}_{(e)} \dot{\boldsymbol{U}}_{(e)}. \tag{33}$$

The partial derivative of $\boldsymbol{F}_{visc(e)}$ with respect to the nodal variables is given by

$$\boldsymbol{K}_{visc(e)} = \frac{1}{\Delta t}\boldsymbol{C}_{(e)}. \tag{34}$$

Finally we discuss the discretization of the stochastic force vector $\boldsymbol{f}_{stoch}$ by a stochastic element force vector $\boldsymbol{F}_{stoch(e)}$. According to Eq. (9) and (24) the discrete stochastic force vector on element level is

$$\boldsymbol{F}_{stoch(e)} = \int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^{T}(\boldsymbol{x})\boldsymbol{f}_{stoch}(\boldsymbol{x})d\Omega. \tag{35}$$

To simplify Eq. (35), we recall that for polymers $\Omega_{(e)}$ is just a curved line of length $L_{(e)}$ embedded into higher dimensions. We subdivide this line into n_c intervals of equal length $h_c = L_{(e)}/n_c$, where $n_c \gg 1$. We define n_c sample points $\hat{\boldsymbol{x}}_k$, with $1 \leq k \leq n_c$ in the middle of each interval, respectively. Whereas in Eq. (27) implicitly a zero correlation length in space was assumed we model the thermal forces now as piecewise constant on each interval of the (small) length h_c and uncorrelated on distinct intervals. Similarly to the discretization in time, this discretization in space goes along with changing from a continuous Dirac function $\delta(\boldsymbol{x}-\boldsymbol{x}')$ in space to the discrete one $\delta(\hat{\boldsymbol{x}}_n - \hat{\boldsymbol{x}}_m) = 1/h_c\delta_{nm}$ which gives

$$\langle \boldsymbol{f}_{stoch}(\hat{\boldsymbol{x}}_n, t^i)\rangle = 0, \tag{36a}$$

$$\langle \boldsymbol{f}_{stoch}(\hat{\boldsymbol{x}}_n, t^i) \otimes \boldsymbol{f}_{stoch}(\hat{\boldsymbol{x}}_m, t^j)\rangle = \frac{2k_BT}{\Delta t h_c}\boldsymbol{c}\,\delta_{ij}\delta_{nm}. \tag{36b}$$

Thus, we may approximate the integral in Eq. (35) at a certain point in time by a sum and write

$$\boldsymbol{F}_{stoch(e)} = \sum_{n=1}^{n_c} \boldsymbol{N}_{(e)}^{T}(\hat{\boldsymbol{x}}_n)\boldsymbol{f}_{stoch}(\hat{\boldsymbol{x}}_n)h_c + \mathcal{O}(h_c^2). \tag{37}$$

The error intrinsic to a finite-element model typically scales with the characteristic element length $L_{(e)}$. Having assumed $h_c \ll L_{(e)}$, we neglect the error scaling with h_c^2 for the rest of this paper. As $\boldsymbol{F}_{stoch(e)}$ is a sum of Gaussian random variables with mean value zero, it is also a Gaussian random variable with mean value zero. Its second moment is given by

$$\begin{aligned}\langle \boldsymbol{F}_{stoch(e)}^{\otimes 2}\rangle &= \left\langle \left[\sum_{n=1}^{n_c} \boldsymbol{N}_{(e)}^{T}(\hat{\boldsymbol{x}}_n)\boldsymbol{f}_{stoch}(\hat{\boldsymbol{x}}_n)h_c\right]^{\otimes 2}\right\rangle \\ &= \sum_{n=1}^{n_c} \boldsymbol{N}_{(e)}^{T}(\hat{\boldsymbol{x}}_n)\boldsymbol{c}\boldsymbol{N}_{(e)}(\hat{\boldsymbol{x}}_n)\frac{2k_BTh_c}{\Delta t} \\ &= \frac{2k_BT}{\Delta t}\int_{\Omega_{(e)}} \boldsymbol{N}_{(e)}^{T}(\boldsymbol{x})\boldsymbol{c}\boldsymbol{N}_{(e)}(\boldsymbol{x})d\Omega_{(e)}.\end{aligned} \tag{38}$$

Here by $(.)^{\otimes 2}$ we denote a dyadic product of a vector with itself. According to the first line of Eq. (38) each element of the covariance matrix of the stochastic element force vectors is theoretically a sum of altogether n_c^2 summands. Each of these summands represents a product of the contributions of two intervals of length h_c. By Eq. (36) the expectation value of each product can be evaluated. Especially the products of contributions of distinct intervals have a zero expectation value. Equation (38) relates to one specific point in time. The correlation between the stochastic forces at distinct discrete points in time is zero. Hence, the element damping matrix $\boldsymbol{C}_{(e)}$ from Eq. (32) allows for uniquely characterizing the stochastic element force vector $\boldsymbol{F}_{stoch(e)}$ by

$$\langle \boldsymbol{F}_{stoch(e)}^{i}\rangle = 0, \tag{39a}$$

$$\langle \boldsymbol{F}_{stoch(e)}^{i} \otimes \boldsymbol{F}_{stoch(e)}^{j}\rangle = \frac{2k_BT\delta_{ij}}{\Delta t}\boldsymbol{C}_{(e)}. \tag{39b}$$

At this point it seems worthwhile dropping some brief remarks about the above calculations: modeling the stochastic forces with a correlation length $h_c = L_{(e)}/n_c$ in space and assuming $n_c \gg 1$ is justified from physics: the thermal forces stem from collisions with molecules in the surrounding fluid and are therefore assumed to have a correlation length much smaller than the polymer or element length. In the above model the stochastic forces are basically considered as stochastic variables resulting from functions piecewise constant over intervals of length h_c in the limit $h_c \to 0$. This point of view has two important implications. First it is the reason why the stochastic forces can be discretized elementwise without accounting for correlations between different elements. Second, in general it is all but obvious, which mathematical consequences a polynomial interpolation of the thermal forces in space may have in the limit $h_c \to 0$. Indeed a rigorous survey of these consequences would require lengthy mathematical discussions and is therefore skipped in this paper. Yet, we would like to briefly mention that the polynomial interpolation of thermal forces according to the above model may be justified in a similar way as demonstrated in [25]. It should also be underlined that the above mathematical derivation of discrete force vectors is in striking contrast to the merely heuristic stochastic forces employed in bead-simulation models.

Another remark should be dedicated to the main finding of Eq. (39): the element damping matrix $\boldsymbol{C}_{(e)}$ is the discrete counterpart of the damping matrix $\boldsymbol{c}$ in Eq. (8). And as the continuous damping matrix $\boldsymbol{c}$ is the scaled covariance matrix for the continuous thermal forces $\boldsymbol{f}_{stoch}$ [see Eq. (9)] the same holds true for the discrete damping matrix $\boldsymbol{C}_{(e)}$ and the discrete thermal forces $\boldsymbol{F}_{stoch(e)}$. One should note carefully that this relation is not a matter of course in the discrete case. Rather one can verify readily that it holds true if and only if the discretization in space is based on a Bubnov-Galerkin method and not on a Petrov-Galerkin method [22].

In the end of this section it seems worthwhile recalling the essentials of Brownian dynamics simulations with finite elements: according to Eq. (28), a finite-element simulation of Brownian dynamics requires in each time step between two points in time t^i and t^{i+1} the following implicit system of equations to be solved:

$$\boldsymbol{F}_{el}^{i+1}(\boldsymbol{U}^{i+1}) + \boldsymbol{C}\frac{\boldsymbol{U}^{i+1} - \boldsymbol{U}^{i}}{\Delta t} = \boldsymbol{F}_{ext}^{i+1}(\boldsymbol{U}^{i+1}) + \boldsymbol{F}_{stoch}^{i+1}(\boldsymbol{U}^{i+1}). \tag{40}$$

Here $\boldsymbol{F}_{visc}^{i+1}$ in Eq. (28) has been replaced according to Eqs. (30) and (33). The system of Eqs. (40) to be solved is different from the systems arising in ordinary finite-element simu-

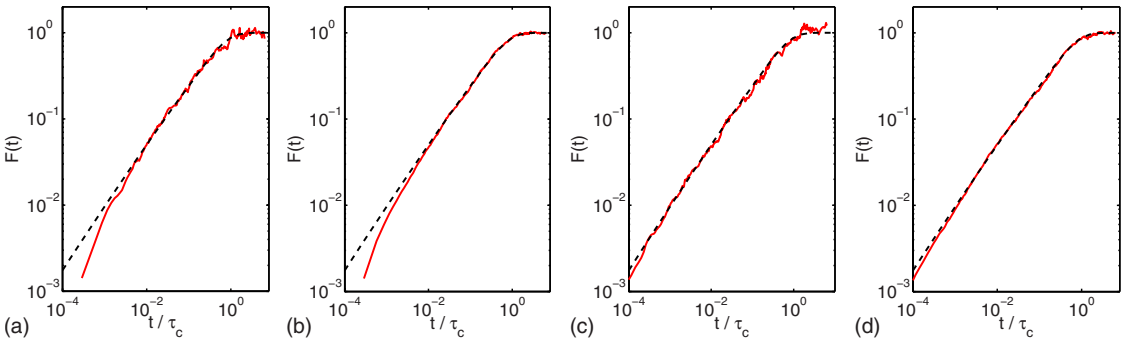

FIG. 3. (Color online) Comparison of analytical solution (dashed line) and finite-element simulation (continued line) of normalized MSD with (from left to right): $N_e=20$, $\Delta t=0.01$, $n_{sample}=400$ and $N_e=20$, $\Delta t=0.01$, $n_{sample}=4000$ and $N_e=40$, $\Delta t=0.001$, $n_{sample}=400$ and $N_e=40$, $\Delta t=0.001$, $n_{sample}=4000$.

lations only by the thermal and viscous forces. Consequently an ordinary finite-element program can be upgraded in order to perform Brownian dynamics simulations just by implementing methods for evaluating the discrete viscous and thermal forces $\boldsymbol{F}_{visc(e)}$, $\boldsymbol{F}_{stoch(e)}$ and the viscous stiffness matrices $\boldsymbol{K}_{visc(e)}$. These quantities can be computed easily in each time step from Eqs. (33), (34), and (39), once the element damping matrix $\boldsymbol{C}_{(e)}$ has been computed from Eq. (32). For the above simple friction model the damping matrix can be computed at the beginning of a simulation once. For example, choosing linear basis functions $N_{a(e)}$ gives

$$\boldsymbol{C}_{lin(e)} = \frac{4\pi\eta L_{(e)}}{3} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{41}$$

For all the examples in Sec. V linear basis functions and thus Eq. (41) were employed.

V. EXAMPLES

The above finite-element model of Brownian polymer dynamics was subject to a rigorous validation and verification process. In this section we present several out of all the simulated examples which show that the model is capable of simulating Brownian polymer dynamics in accordance with theoretical predictions and experimental data. When referring to physical or numerical parameters we omit units as all values are given in terms of the basis units micrometer, second, milligram, and Kelvin.

A. Brownian dynamics of a polymer with hinged ends

A typical test case to study Brownian dynamics of polymers is the dynamics of a filament with hinged ends in an x-y plane. Here we consider a polymer whose ends are freely movable in x direction and attached in y direction. In the absence of bending its backbone is aligned with the x axis. Both end points are assumed to be hinged, which means that a zero bending moment at both end points is assumed. The end-to-end distance $R(t)$ at a certain point in time t changes steadily due to stochastic thermal forces acting on the polymer and causing varying bending deformations. The dynamics of $R(t)$ may be characterized by means of the so-called mean-square difference (MSD) of the end-to-end distance, which is defined by

$$\delta R^2(t) \equiv \langle [R(t) - R(0)]^2 \rangle. \tag{42}$$

Obviously, the MSD starts at zero and increases due to the fact that the stochastic thermal forces continuously change the shape of the polymer and thus the end-to-end distance. At each point in time $t>0$ the random thermal forces affect a filament either in such a way that $[R(t)-R(0)]^2$ increases or decreases. Roughly spoken we may say: the more similar the shape of the polymer at a point in time t is compared to the initial shape the larger the probability that the random forces increase $[R(t)-R(0)]^2$ at that point in time and vice versa. Therefore a saturation of the MSD can be observed after some time when the polymers in the observed ensemble have changed their shape sufficiently compared to the initial shape at time zero. For polymers of contour length L and persistence length l_p in the stiff limit $L/l_p \ll 1$, in [26] a way to compute the MSD analytically was presented. Thus the MSD is an excellent test case for the validation of the finite-element simulation of Brownian polymer dynamics. By Monte Carlo simulations we studied the convergence of the MSD gained from finite-element simulations toward the exact value predicted in [26]. As geometric and material parameters for the polymer we chose a moment of inertia $I=28.74\times10^{-12}$, a cross section $A=19\times10^{-6}$, a Young's modulus $E=1.3\times10^{9}$, a Poisson's ratio $\nu=0.3$ and a contour length $L=10$. The viscosity of the surrounding fluid is $\eta=1\times10^{-3}$ and its temperature is given by the thermal energy $k_BT=4\times10^{-5}$ with the Boltzmann constant k_B. In Fig. 3 the analytical solution of Hallatschek is compared to the results of our finite-element simulation. The time t was normalized with respect to the relaxation time τ_c of the slowest eigenmode, which is computed according to [27] by τ_c

$\approx(\zeta/EI)(L/4.73)^4$. The MSD was normalized by defining

$$F(t)=\frac{90l_p^2}{L^4}\delta R^2(t). \tag{43}$$

Results for a finite-element simulation with $N_e=20$ elements and time step size $\Delta t=0.01$ as well as $N_e=40$ elements and time step size $\Delta t=0.001$ are presented. In either case the MSD, which is an ensemble average, was calculated both from $n_{sample}=400$ and $n_{sample}=4000$ realizations.

Apparently a refinement of the discretization in space and time together with an increase in the number of realizations makes the simulated MSD converge toward the analytically predicted one. With $N_e=40$ elements and $n_{sample}=4000$ excellent agreement over more than four decades in time is achieved already. Although the intention of this paper is by no means a careful study of computational performance, some details seem worthwhile being mentioned at this point: finite-element models with 20 and 40 elements, respectively, may be compared to bead-spring models with 21 and 41 beads, respectively. For the latter ones the explicit time integration scheme usually employed requires a time step size $\Delta t\approx 4\times 10^{-8}$ and $\Delta t\approx 8\times 10^{-9}$, respectively, for the above parameter choice. Thus explicit bead-spring simulations require approximately 1×10^5 more time steps than the here introduced implicit finite-element scheme. Certainly the computational cost of one implicit time step is higher than the one of an explicit one. Abstaining from a rigorous performance comparison we would just like to present some very rough figures about computational cost. In our simulations the implicit time integration scheme required approximately six iteration steps per time step and additionally one predictor step. In each iteration step the time for evaluation of the element vectors and matrices on the one hand and for the solution of the equation system on the other hand were found to be comparable. For linear elements with six degrees of freedom one may roughly estimate the evaluation time for the stiffness matrices six times longer than for the force vectors. In an explicit scheme no stiffness terms have to be evaluated and the force vectors only one time. Therefore one may estimate the CPU time for one implicit time step approximately 91 times higher than for one explicit time step, which would still make the implicit time integration scheme surpass an explicit one by a factor of more than 1000 in the above example with respect to computational cost. The exact amount of computational cost saved depends, of course, crucially on the choice of parameters. Therefore it is emphasized that the above parameter choice is by no means an exotic one. Rather the above parameters are the ones of a 10 μm long actin filament without additional stabilization, e.g., by tropomyosin or phalloidin.

Indeed the almost inacceptable computational cost of bead-spring simulations, which may go along with a parameter choice faithfully modeling the physical reality is a well-known problem. Often it is just circumvented by tampering with the physical parameters such as done in [28]. Such a modification is conducted always at the risk of an improper model, which looses the capability of representing certain kinds of physical effects in a correct manner. The capability of the finite-element method to deal with the above example without any such work around may be considered a significant advantage over so far proposed simulation models.

B. Brownian dynamics of a polymer with free ends

The dynamics of semiflexible polymers in dilute solution has been subject to several careful studies. As a semiflexible polymer of contour length L undergoes thermal undulations its effective end-to-end distance $R(t)$ is practically always smaller than L and changes continuously. Obviously different end-to-end distances arise with a different probability and the amplitude of the thermal undulations may be characterized by the so-called radial distribution function (RDF) $G(R)$, which gives the probability for a certain end-to-end distance $R(t)$. According to [29] the RDF can be computed in two dimensions as

$$G(R)=\frac{1}{N}\sum_{l=0}^{\infty}\frac{(2l)!}{(2^l l!)^2}\times\frac{D_{3/2}\left(2\dfrac{l+0.25}{\sqrt{\phi(R)}}\right)}{\phi(R)^{5/4}\exp\left[\dfrac{(l+0.25)^2}{\phi(R)}\right]}, \tag{44}$$

where $D_{3/2}(\cdot)$ is a parabolic cylinder function, $\phi(R)\equiv 2(1-R/L)\kappa/(Lk_BT)$ and N ensures a normalized distribution. Equation (44) has been verified experimentally in [27] by confining the fluctuations of actin polymers to the narrow gap between two plates and measuring the frequency at which certain end-to-end distances arose. By curve fitting it was possible to determine the experimental parameters to $L=13.4$, $l_p=16.2$, $\kappa=6.55\times 10^{-2}$, $E=2.3\times 10^9$, and $k_BT=4.045\times 10^{-3}$, where the nomenclature of the parameters was chosen as introduced in Sec. V A. Note that the bending stiffness is approximately twice as large as assumed for the numerical example in Sec. V A as the experiments in [27] were conducted with phalloidin stabilized actin filaments.

These numerical data were given as input to our finite-element Brownian dynamics simulation with $N_e=10$ and $N_e=40$ elements, respectively. Defining the function

$$\Gamma(R)=\frac{G(R)R}{\displaystyle\int_{R=0}^{R=L}G(R)RdR}, \tag{45}$$

a comparison between the simulated RDF with experimental data according to [27] and the theoretical prediction according to Eq. (44) is presented in Fig. 4. Apparently the simulation results excellently agree both with the theoretical prediction and experiments already for $N_e=40$ elements and a time step $\Delta t=1\times 10^{-4}$.

C. Rotational diffusion of a microtubule

So far we have always considered a very simple friction model not accounting for any geometric anisotropy of the polymer nor for any hydrodynamic interactions. In this final example we give a brief outlook to a more complex friction model. As pointed out in [7] hydrodynamic interactions may be roughly accounted for by means of the logarithmic correction factor $\ln(L/d)$, where d is the diameter of the poly-

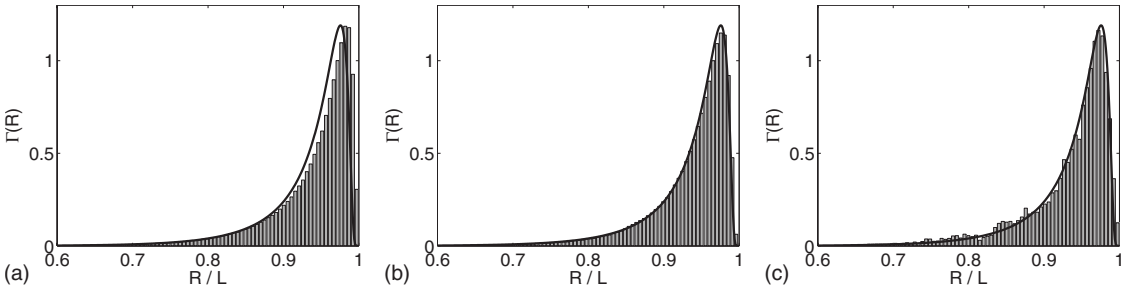

FIG. 4. RDF gained from finite-element simulation with $N_e=10$, $\Delta t=1\times10^{-3}$ (left) and $N_e=40$, $\Delta t=1\times10^{-4}$ (middle) compared with experimental data from [27] (right). The analytical solution according to Eq. (44) is represented by the continued black line, respectively, and simulation and experimental results are illustrated by the gray histograms.

mer backbone and L is its contour length. The simplest way to do so is not to choose $\zeta=4\pi\eta$ as suggested in Sec. II B but $\zeta=4\pi\eta\ln^{-1}(L/d)$. By such a friction model we cannot yet account for the anisotropy of friction parallel and orthogonal to the polymer backbone, however, already this minor modification allows for a reasonable simulation of the rotational diffusion of stiff polymers or rodlike viruses: we consider polymer fluctuations confined to two dimensions. Then a stiff polymer or rodlike virus can be described by the motion of its center of gravity and its orientation angle ϕ in the x-y plane. The rotational diffusion coefficient can be defined as

$$D_r=\frac{\langle\Delta\phi^2\rangle}{2t}, \tag{46}$$

where $\Delta\phi$ is the change in the orientation angle within a time interval of length t and $\langle.\rangle$ again denotes an ensemble average. According to [30] for stiff polymers with moderate to large length-to-diameter ratio the rotational diffusion coefficient D_r can be well predicted by

$$D_r=\frac{3k_BT}{\pi\eta L^3}\left(\ln\frac{L}{d}+2\ln 2-\frac{11}{6}\right). \tag{47}$$

The analytical predictions by Eq. (47) were verified experimentally several times, e.g., in [31]. Here we make use of this relation to predict the rotational diffusion coefficient of microtubles of length $L=20$. Due to the fact that the persistence length of microtubles is approximately $l_p=5.99\times10^3$ and the length-to-diameter ratio is given by $L/d\approx8.49\times10^2$, Eq. (47) can be readily applied to predict the rotational diffusion coefficient of such microtubules in dilute solution. With parameters $\kappa=23.94$, $k_BT=4\times10^{-3}$, and $\eta=1.1146\times10^{-3}$, Eq. (47) leads to $D_{r,microtubule}=2.698\times10^{-3}$.

To capture the rotational diffusion of such a microtuble a finite-element discretization with already one single element is enough (a finer discretization would not change the result). For the simulation we chose a time step $\Delta t=0.076$. In Fig. 5 the angular diffusion $\langle\Delta\phi^2\rangle$ averaged over 2000 realizations is plotted over the time t together with the diffusion expected due to Eq. (47). Obviously the simulation fits well with common formulas for the computation of rotational diffusion. The precise value of the simulated rotational diffusion coefficient may be computed by summing up all the angular increments over all time steps of all realizations and then dividing by the number of realizations and time steps. This procedure gives $D_{r,simulation}=2.895\times10^{-3}$. The relative error of 7.3% compared to the theoretical prediction arises due to the vastly simplified friction model. Yet this numerical experiment confirms the capability of the finite-element method to well resemble diffusion processes of polymers.

VI. CONCLUSIONS

In this paper we have presented a finite-element framework for Brownian dynamics simulations of polymers. As polymers are formed by a large number of monomers, common simulation models such as bead-spring and bead-rod models model polymers as a series of beads. These beads represent in a way the single monomers. In reality the size of each monomer is usually much smaller than the size of the

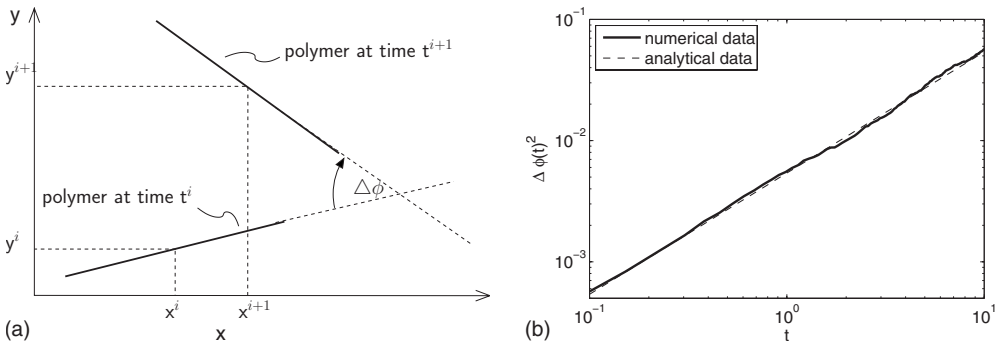

FIG. 5. Angle of rotation of a stiff polymer between two different points in time t^i and t^{i+1} (left). Simulated angular diffusion $\langle\phi(t)^2\rangle$ compared to the one predicted by Eq. (47) (right).

whole polmyer. As a consequence in reality the number of monomers surpasses the number of beads in simulations often vastly, which renders moot the suitability of the approach. As against the bead models the finite-element model takes a completely different point of view: since the polymer is much larger than each single monomer, a continuum mechanical model is employed. The continuum model is based on a clearly formulated set of modeling assumptions. From these assumptions a finite-element formulation is derived in a strictly mathematical manner. Due to this sound theoretical foundation for each parameter present in the numerical model a physical interpretation can be given easily. Therefore finite-element simulations can be applied easily also in a predictive sense. In opposition to that bead-spring models are often confronted with the difficulty that certain numerical parameters such as the friction coefficient of the beads are difficult to be interpreted physically [7]. So far there has been a gap between the partial differential equations employed by theorists in order to describe polymers in an analytical way and the models employed for the simulation of polymers. The finite-element model closes this gap: the finite-element formulation can be proven mathematically to converge to exactly the solution of the partial differential equations of theoretical polymer physics.

The finite-element model described in this paper is based on Reissner's beam theory and thus capable of representing not only thin but also comparatively thick polymers such as microtubles or polymer bundles as arising in certain types of polymer networks. In opposition to that bead-models are suitable for very slender filaments only. Bead models are usually applied together with an explicit time integration scheme. Often this entails severe problems as to the time step size. These can be circumvented only by accepting the inability of resolving phenomena of enthalpic axial elasticity in a bead-rod model. In opposition to that, for the proposed finite-element model an implicit time integration scheme is employed in case that numerical stability is an issue. Together with such an implicit time integration scheme the complete range of polymer elasticity, including both entropic and enthalpic elasticity, can be accounted for at an acceptable computational cost. Especially for polymer networks, where both phenomena may matter significantly for the same filament at different points in time, this represents a striking advantage over currently available simulation models. So far finite-element simulations of polymer networks have been only employed neglecting Brownian dynamics [8–12]. However, it is well known that Brownian dynamics contributes essentially to the mechanics both of single filaments as well as to the mechanics of polymer networks. For this reason recent work [28,32] more and more aims at capturing effects of Brownian dynamics, too. However, computational cost is well known to form a severe obstacle for such simulations. Thus, there is urgent need in material science, biophysics, and bioengineering for a simple and efficient simulation tool capable of capturing Brownian dynamics in large-scale simulations. The above finite-element model represents such a tool and might open up the way to the simulation of a variety of complex processes, e.g., the remodeling of the human cytoskeleton.

By means of several numerical examples we have shown good accordance of the proposed approach both with theoretical predictions and experimental data. Thus, altogether the finite-element model may be considered a powerful alternative to existing simulation methods for Brownian polymer dynamics. Its versatility makes it the ideal tool for simulating both single filaments and large systems of polymers such as the cytoskeleton in biological cells. Especially from the practical point of view an application of the finite-element method allows for exploiting a number of benefits: so far polymer dynamics simulations are often run by means of simple self-written in-house codes. Especially when simulating complex phenomena, not only programming such a code may be cumbersome. Also using such codes, which often lack proper interfaces, is tedious. In opposition to that the finite-element method is a *de facto* standard in computational engineering. Therefore simulations can be set up making use of widely available powerful packages, comfortable user interfaces, and preprocessing and postprocessing tools. Making use of these tools consequently will open up a convenient way to large scale Brownian polymer dynamics simulations in material science and bioengineering.

ACKNOWLEDGMENTS

The support for the first author (C.J.C.) by the International Graduate School for Science and Engineering (IGSSE) of Technische Universität München is gratefully acknowledged. In addition to that, we thank Christof Hirth for the generation and preparation of the numerical data used for the validation and verification of this method. We thank Benedikt Obermayer and Erwin Frey (Ludwig-Maximilians-Universität München) for fruitful discussions.

[1] A. Alexander-Katz, M. F. Schneider, S. W. Schneider, A. Wixforth, and R. R. Netz, Phys. Rev. Lett. **97**, 138101 (2006).
[2] A. Montesi, D. Morse, and M. Pasquali, J. Chem. Phys. **122**, 084903 (2005).
[3] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics* (Clarendon Press, Oxford, 1994).
[4] P. S. Grassia, E. J. Hinch, and L. C. Nitsche, J. Fluid Mech. **282**, 373 (1995).
[5] E. J. Hinch, J. Fluid Mech. **271**, 219 (1994).
[6] C. Heussinger, M. Bathe, and E. Frey, Phys. Rev. Lett. **99**, 048101 (2007).
[7] P. L. Chandran and M. R. K. Mofrad, Phys. Rev. E **79**, 011906 (2009).
[8] P. R. Onck, T. Koeman, T. van Dillen, and E. van der Giessen, Phys. Rev. Lett. **95**, 178102 (2005).
[9] T. van Dillen, P. R. Onck, and E. van der Giessen, e-print arXiv:physics/0611230.
[10] E. M. Huisman, T. van Dillen, P. R. Onck, and E. Van der Giessen, Phys. Rev. Lett. **99**, 208103 (2007).
[11] C. Heussinger and E. Frey, Eur. Phys. J. E **24**, 47 (2007).

[12] J. A. Åström, P. B. Sunil Kumar, I. Vattulainen, and M. Karttunen, Phys. Rev. E **77**, 051913 (2008).
[13] O. Lieleg, K. M. Schmoller, C. J. Cyron, Y. Luan, and A. R. B. W. A. Wall, Soft Matter **5**, 1796 (2009).
[14] O. Hallatschek, E. Frey, and K. Kroy, Phys. Rev. E **75**, 031905 (2007).
[15] F. C. MacKintosh, J. Käs, and P. A. Janmey, Phys. Rev. Lett. **75**, 4425 (1995).
[16] A. Bedford and W. Fowler, *Engineering Mechanics: Statics and Dynamics* (Prentice Hall, Upper Saddle River, 2005).
[17] M. A. Crisfield, *Nonlinear Finite-Element Analysis of Solids and Structures, Volume 1: Essentials* (Wiley & Sons, Chichester, 2003).
[18] J. Rotne and S. Prager, J. Chem. Phys. **50**, 4831 (1969).
[19] D. L. Ermak and J. A. McCammon, J. Chem. Phys. **69**, 1352 (1978).
[20] J. Howard, *Mechanics of Motor Proteins and the Cytoskeleton* (Sinauer Associates, Sunderland, 2001).
[21] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods* (Springer, New York, 2002).
[22] T. J. R. Hughes, *Finite Element Method: Linear Static and Dynamic Finite Element Analysis* (Prentice Hall, Englewood Cliffs, 2000).
[23] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method* (Wellesley-Cambridge Press, Wellesley, 1988).
[24] C. J. Cyron and W. A. Wall (unpublished).
[25] E. J. Allen, S. J. Novosel, and Z. Zhang, Stochastics **64**, 117 (1998).
[26] O. Hallatschek, *Semiflexible Polymer Dynamics* (Shaker Verlag, Aachen, 2004).
[27] L. Le Goff, O. Hallatschek, E. Frey, and F. Amblard, Phys. Rev. Lett. **89**, 258101 (2002).
[28] T. Kim, W. Hwang, and R. D. Kamm, Exp. Mech. **49**, 91 (2009).
[29] J. Wilhelm and E. Frey, Phys. Rev. Lett. **77**, 2581 (1996).
[30] J. Bonet Avalos, J. M. Rubu, and D. Bedeaux, Macromolecules **26**, 2550 (1993).
[31] R. P. J. K. Phalakornkul and A. P. Gast, Macromolecules **32**, 3122 (1999).
[32] H. Lee, B. Pelz, J. M. Ferrer, T. Kim, M. J. Lang, and R. D. Kamm, Cellular and Molecular Bioengineering **2**, 28 (2009).