Computers and Structures 196 (2018) 24–35

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

A framework of finite element procedures for the analysis of proteins

Reza Sharifi Sedeh [a], Giseok Yun [b], Jae Young Lee [b], Klaus-Jürgen Bathe [a], Do-Nyun Kim [b,c,*]

[a] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, United States
[b] Department of Mechanical and Aerospace Engineering, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Republic of Korea
[c] Institute of Advanced Machines and Design, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Republic of Korea

ARTICLE INFO

Article history:
Received 8 July 2017
Accepted 24 October 2017
Available online 20 November 2017

Keywords:
Finite element procedures
Protein
Solvent damping
Friction matrix
Brownian dynamics
Macromolecules

ABSTRACT

Large-scale, functional collective motions of proteins and their supra-molecular assemblies occur in a physiological solvent environment at finite temperatures. The solution of these motions with standard molecular dynamics algorithms is computationally hardly possible when considering macromolecules. Much research has focused on alternative approaches that use coarse-graining to model proteins, but mostly in vacuum. In this paper, we incorporate realistically the physical effects of solvent damping into the finite element model of proteins. The proposed framework is based on Brownian dynamics and shown to be effective. An important advantage of the approach is that the computational cost is not dependent on the molecular size, which makes the long-time simulation of macromolecules possible. Using the proposed procedure, we demonstrate the analysis of a macromolecule in solvent—an analysis that has not been achieved before and could not be performed with a molecular dynamics algorithm.

1. Introduction

Protein motions such as conformational changes and folding/unfolding, generally occur in a physiological solvent, that is, a viscous environment within cells. Hence, to solve for the dynamical behavior of a protein, both the protein and the solvent should ideally be modeled simultaneously, as in all-atom, explicit-solvent molecular dynamics [1]. However, in practice, the time-integration of the full set of governing equations of motion in molecular dynamics is computationally hardly feasible, in particular when large length scale and long time scale motions need to be considered with the effects of the solvent.

Hence, to simulate protein motions, various coarse-grained modeling approaches have been developed. These models can describe approximately important protein motions that are hardly accessible using a molecular dynamics simulation. For example, protein folding and unfolding have been investigated, respectively, using the lattice models to coarse-grain the spatial discretization [2–4] and the steered molecular dynamics procedure to coarse-grain the time discretization [5]. Also, the elastic network model for coarse-grained normal mode analysis has been used to solve for the change of flexibility of proteins in large deformations [6–9]. However, the effects of the solvent on the motion and the flexibility of proteins have been ignored in the elastic network model, and therefore, the predicted in-vacuum frequencies do not correspond to realistic time-scales and the physical normal modes of the protein [10,11].

On the other hand, the Brownian dynamics formalisms include the solvent effects implicitly. The formalisms can be used to simulate biomolecular motions on a computer substantially faster than the molecular dynamics techniques and with finite element procedures [12] open an avenue to significantly advance the field. In 1978, Ermak and McCammon [13] proposed a generalized algorithm to simulate the Brownian dynamics of N particles, where hydrodynamic interactions were described by a $3N \times 3N$ diffusion tensor. In the Ermak-McCammon procedure, the tensor needs to be Cholesky-decomposed [12] at each step to compute random displacements, resulting in a computation-time scaling of $O(N^3)$. Over the past four decades, researchers have developed several approaches to reduce this computational cost [14–17] in order to make the long time-scale Brownian dynamics simulations of large biomolecules feasible. For example, a Chebyshev-polynomial approach was proposed by Fixman [16] for the approximation of the square-root of the diffusion tensor, which results in a computational cost that scales with $O(N^{2.25})$ [18]. Also, as another alternative to the direct Cholesky-decomposition of the diffusion tensor, Geyer and Winter [17] proposed the Truncated Expansion Ansatz, which scales with $O(N^2)$ by truncating the expansion of the hydrodynamic multi-particle correlations as two-body contributions at

* Corresponding author at: Department of Mechanical and Aerospace Engineering, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Republic of Korea.
E-mail address: dnkim@snu.ac.kr (D.-N. Kim).

https://doi.org/10.1016/j.compstruc.2017.10.015
0045-7949/

the second order. Recently, based on Krylov subspaces, Ando and coworkers [14] proposed a new approach to approximately compute the random Brownian displacements with a computation time scaling of $O(N^2)$. As an alternative to approximating the square-root of the diffusion tensor in order to speed up the Brownian dynamics simulations, the tensor may be also kept unchanged for several sequential time-steps [19–22] or throughout the Brownian dynamics simulation as in our own recent work on DNA nanostructures [23].

A variety of Brownian dynamics packages are already available for simulating the protein dynamics from SDA [24] and Browndye [25], which use rigid-body models of proteins, to UHBD [26], BD_Box [27], and Brownmove [28], which use flexible models. Here, coarse-grained Brownian dynamics simulations have been used to analyze protein motions by employing bead models [29–31]. However, these models are complicated, and more importantly, they lead to bead overlapping [32], require volume and viscosity corrections [33,34], and ignore the presence of protein atoms between bead pairs [35]. Additionally, although solvent friction takes place on the surface of proteins [36], the bead models used in the Brownian dynamics simulations assume that the frictional forces act at the centers of the α-carbon atoms (representative atoms in the protein) of amino acids which are the building blocks of proteins.

Here, we propose a novel framework of finite element procedures for the analysis of proteins. In this framework we model the protein and solvent environment more realistically with the frictional forces applied directly on the protein surface and without any overlapping and any correction for the volume and viscosity. The friction matrix due to the solvent damping is computed by embedding a protein in a Stokes fluid and establishing an influence matrix. Due to the specific physics, we do not solve a nonlinear fluid-structure interaction problem, like performed in many other fields, see for example [12,37,38]. The interaction matrix is obtained as usual in finite element analyses [12,39], but of course with the specific conditions encountered in the case here considered, as detailed below. The computational cost to obtain the friction matrix for the Brownian dynamics simulation using ADINA version 9.3 (ADINA R&D, Inc, Watertown, MA, USA) [40] is quite reasonable.

In the following sections, we first discuss how the stiffness, mass, and friction matrices are obtained for the Brownian dynamics simulation using the finite element method, and show how to calculate the diffusion coefficients, which define the translational and rotational mobility of proteins in the solvent, from the friction matrix. Then, we give results obtained using the proposed method considering a simple case for which analytical solutions are available, and compare results for actual proteins with experimental data. Diffusion coefficients calculated for 10 proteins of various molecular weights, ranging from 7 kDa to 233 kDa (with 1 kDa = 1 kilodalton = $1.6605402 \times 10^{-21}$ g) are provided. We also give more detailed results for the proteins Taq polymerase and Lysozyme obtained using our Brownian dynamics, finite element simulation framework. These illustrate that the solvent-damping effects can significantly alter the normal modes of proteins. Finally, we show considering the protein GroEL that our proposed framework can be used efficiently to solve for the response of large proteins when a molecular dynamics solution is not feasible.

2. Finite element framework

In this section, we present a framework of finite element procedures developed for the analysis of protein dynamics in solvents.

2.1. Langevin and Brownian dynamics

The Langevin governing equations are [35]

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Z}\dot{\mathbf{q}} + V'(\mathbf{q}) = \mathbf{f}(t) \tag{1}$$

where $\mathbf{M}$ is the $3N \times 3N$ diagonal mass matrix, $\mathbf{Z}$ is the $3N \times 3N$ friction matrix, V is the potential energy function, $V'(\mathbf{q})$ are the spatial derivatives of the potential energy function with respect to the position vector, $\mathbf{q}$ is the position vector, $\dot{\mathbf{q}}$ is the velocity vector, $\ddot{\mathbf{q}}$ is the acceleration vector, N is the number of particles in the Langevin dynamics model, and $\mathbf{f}(t)$ is the vector of external stochastic forces with the following conditions

$$\begin{aligned} &\langle f_i(t) \rangle = 0 \\ &\langle f_i(t) \cdot f_j(t') \rangle = 2k_B T Z_{ij} \delta(t - t') \end{aligned} \tag{2}$$

Here the bracket notation indicates the time-average value, k_B is Boltzmann's constant, T is the temperature, $\delta(t - t')$ is the Dirac delta function, $f_i(t)$ is the i^{th} component of $\mathbf{f}(t)$, and Z_{ij} is the ij^{th} component of the friction matrix. By expanding the potential energy function in a Taylor series around a minimum state $\mathbf{q^0}$ and neglecting the terms higher than quadratic order, we obtain the Langevin equations governing the linearized protein response

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{Z}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \tag{3}$$

The ij^{th} component of the stiffness matrix $\mathbf{K}$ is

$$K_{ij} = \frac{\partial^2 V}{\partial q_i \partial q_j} = \frac{\partial^2 V}{\partial x_i \partial x_j} \tag{4}$$

and the displacement vector $\mathbf{x}$ is

$$\mathbf{x} = \mathbf{q} - \mathbf{q^0} \tag{5}$$

In our research, we establish the mass matrix and the stiffness matrix as described in Section 2.2. The Brownian dynamics equations are directly obtained by neglecting the inertial forces in Eq. (3).

2.2. Calculation of the stiffness and mass matrices

The finite element method has been used successfully in calculating the lowest normal modes of proteins in vacuum [41–43]. It has been shown that proteins can be modeled simply as homogeneous, isotropic, linear elastic continua because the mode shapes of the lowest frequencies depend predominantly on the overall geometry of the protein [44]. It was also shown that modeling the protein as a heterogeneous material does not improve the results significantly [45]. To calculate the stiffness and mass matrices of a protein, the volume within its molecular surface is discretized with the 10-node tetrahedral elements using ADINA. The mass density of the protein model is defined to be the molecular mass per unit volume, and Poisson's ratio is set to 0.3. The Young's modulus of the protein model is an adjustable parameter, which is determined here by fitting the fluctuation profiles of the α-carbon atoms obtained using the finite element method [46] to those calculated by performing the atomistic block normal mode analysis [47,48] using a molecular dynamics simulation program, CHARMM [49].

2.3. Calculation of the friction matrix using the finite element method

Here we use the finite element method to calculate the solvent friction matrix $\mathbf{Z}$. We embed the protein geometry in the solvent with the boundary of the protein given by the solvent-excluded surface. The solvent-excluded surface of a protein is defined as the closest contact point to the protein van der Waals surface of a solvent probe with the radius of 1.4 Å (1 Å = 10^{-10} m) rolled over