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

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

that surface. We compute the solvent-excluded surface using the freely available program MSMS ver. 2.6.1 [50]. Subsequently, the surface is coarsened using the surface simplification algorithm QSLIM [51–53], as implemented in MeshLab [54] to have a smaller number of surface nodes. Then the coarsened solvent-excluded surface is imported into the finite element analysis program ADINA [40]. The space from the solvent-excluded surface to the surface of a sphere whose diameter is about 400 times the largest dimension of the protein is meshed (Figs. 1 and 2). The element sizes change from the finest (near the solvent-excluded surface) to the coarsest (near the sphere surface) level in eleven spherical layers composed of 10-node tetrahedral elements (Fig. 2). In this study, the radii of spheres are set to 1.4, 2, 3, 4, 5, 6, 8, 10, 20, 100, and 400 times the largest dimension of a protein, but their surfaces are meshed with the same number of triangles. For the calculation of **Z**, the protein itself is not modeled but only the surface geometry is used.

Considering a stationary Cartesian reference frame ($x_i,\ i = 1, 2, 3$) and using index notation, the governing equations of the Stokes flow are [12]

$$\text{momentum}: \quad \partial\tau_{ij}/\partial x_j + f_i^B = 0 \tag{6}$$

$$\text{constitutive}: \quad \tau_{ij} = -p\delta_{ij} + 2\mu e_{ij} \tag{7}$$

$$\text{continuity}: \quad v_{i,i} = 0 \tag{8}$$

where

v_i = velocity of fluid flow in direction x_i
τ_{ij} = components of stress tensor
f_i^B = components of body force vector, here zero
p = pressure
δ_{ij} = Kronecker delta
μ = fluid (laminar) viscosity
e_{ij} = components of velocity strain tensor $= \frac{1}{2}(\partial v_i/\partial x_j + \partial v_j/\partial x_i)$

These equations when discretized, as usual, by finite elements lead to a symmetric coefficient matrix, which is identical to the matrix obtained in the incompressible analysis of a solid material provided μ is equal to the shear modulus and velocities are interpreted as displacements. We use this structural analysis approach and the mixed, displacement-pressure (u/p) finite element formulation since an incompressible analysis is performed. Here, 10/1 tetrahedral elements are used where the displacements are interpolated using 10 nodes (4 at the corners and 6 on the sides of a tetrahedron) and the element pressure is assumed to be constant [12]. We use the notation "10/1 element" for the displacement-pressure mixed element which means that the element contains 10 displacement degrees of freedom and 1 pressure degree of freedom. Likewise, "8/1 element" means that the element contains 8

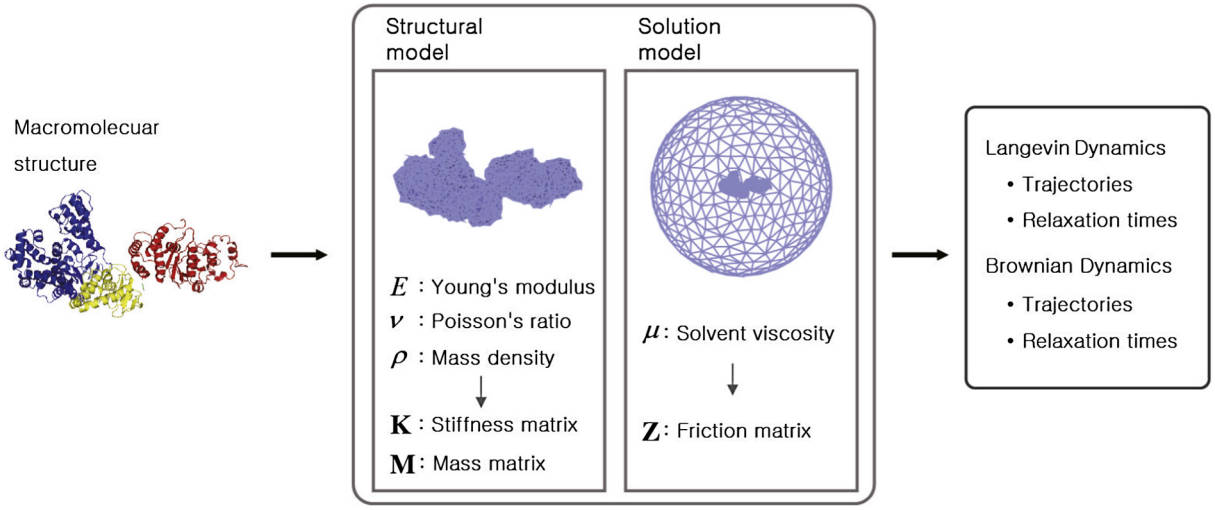

Fig. 1. Finite element framework for the Langevin and Brownian dynamics simulation of proteins. The X-ray crystal structure of Taq polymerase (Protein Data Bank ID 1TAQ) (*left*) is obtained from the Protein Data Bank and its solvent-excluded surface is calculated and coarsened. Then, structural and solvent models of Taq polymerase are defined to calculate the stiffness, the mass, and the friction matrices (*middle*). Finally, the calculated matrices can be used for the Langevin and Brownian dynamics simulations of Taq polymerase (*right*).

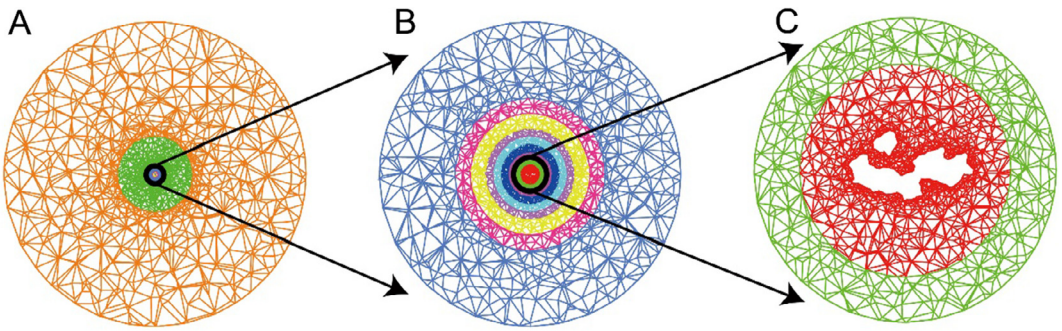

Fig. 2. The mesh between the Taq polymerase surface and the surface of the sphere (in cross-section). Finite element meshes are shown for (A) all the layers, (B) 9 layers, and (C) 2 layers from the Taq polymerase surface. Different colors indicate different layers in the mesh. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

displacement degrees of freedom and 1 pressure degree of freedom.

In the analysis process, we apply unit velocities in succession, to each of the nodal degrees of freedom on the protein surface, with each time all other degrees of freedom on the protein surface set to zero. The resulting nodal forces acting onto the protein nodes for a unit velocity applied to a surface node are the corresponding column entries in the friction matrix. Since the finite element coefficient matrix is constant and symmetric, the solution process requires one initial matrix factorization and then $3M$ forward reductions and back-substitutions, which are performed quite effectively [12], where M represents the number of surface nodes. Considering only the nonzero entries of the friction matrix $\mathbf{Z}$, we see that these correspond to a $3M \times 3M$ matrix, $\widetilde{\mathbf{Z}}$, for the nodes on the protein surface.

Here it is important to use a proper element to obtain a reasonable pressure distribution. We employed the 10/1 tetrahedral element with a single pressure degree of freedom because it satisfies the inf-sup condition [12]. If we use instead the 8/1 hexahedral element with the constant pressure assumption, which does not satisfy the inf-sup condition, abnormal pressure distributions with a checkerboard pattern are obtained as shown in Fig. 3A. This nonphysical result never appeared when we used the 10-node tetrahedral element (Fig. 3B). We tested a hybrid finite element model as well in which only the first layer was discretized using 10/1 tetrahedral elements while the other layers were meshed using 8/1 hexahedral elements, which reduces the computational cost (Fig. 3C). The pressure prediction near the solid body significantly improved, but still, the pressure was badly predicted in the other layers due to the checkerboard-like instability. Hence, for all further results we use finite element models constructed using the 10-node tetrahedral element only.

2.4. Calculation of diffusion coefficients from the friction matrix

With the numerical scheme described above, the finite element friction matrix can be readily established. However, to proceed further, the entries in this matrix should be validated by comparisons with analytical solutions (for simple geometries) and data obtained in physical experiments. We pursued the validation of the matrix in the first instance by calculating the diffusion coefficients corresponding to the friction matrix for a simple case allowing an analytical solution (see Section 3.1.1).

The first step is to establish a 6×6 global resistance tensor, Ξ. Such tensor is usually defined to express the hydrodynamic resistance of an object [36]. To calculate Ξ from the local nodal actions, we partition the $3M \times 3M$ friction matrix $\widetilde{\mathbf{Z}}$ into M^2 3×3 blocks, $\widetilde{\mathbf{Z}}^{(i,j)}$, where i and j denote surface nodes. $\widetilde{\mathbf{Z}}^{(i,j)}_{mn}$ is the mn component of the 3×3 matrix $\widetilde{\mathbf{Z}}^{(i,j)}$, and gives the friction force generated at node i in the m direction ($m = 1, 2, 3$) for the unit velocity applied at node j in the n direction ($n = 1, 2, 3$).

Then Ξ can be obtained using the following equations

$$\Xi_{tt} = \sum_i \sum_j \widetilde{\mathbf{Z}}^{(i,j)} \tag{9}$$

$$\Xi_{tr} = \sum_i \sum_j \mathbf{U}_i \widetilde{\mathbf{Z}}^{(i,j)} \tag{10}$$

$$\Xi_{rr} = \sum_i \sum_j \mathbf{U}_i \widetilde{\mathbf{Z}}^{(i,j)} \mathbf{U}_j^T \tag{11}$$

$$\Xi = \begin{pmatrix} \Xi_{tt} & \Xi_{tr}^T \\ \Xi_{tr} & \Xi_{rr} \end{pmatrix} \tag{12}$$

where $\mathbf{U}_i$ is the matrix converting the force components in $\widetilde{\mathbf{Z}}^{(i,j)}$ to moment components in Ξ_{tr} about the coordinate axes

$$\mathbf{U}_i = \begin{pmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{pmatrix} \tag{13}$$

Thus Ξ_{tt}, Ξ_{rr}, and Ξ_{tr} are the 3×3 blocks of Ξ which correspond to the friction forces and moments generated for the protein in translation, rotation, and translation-rotation coupling, respectively, and x_i, y_i, and z_i are the coordinates of node i in the stationary Cartesian reference frame.

The 6×6 diffusion matrix $\mathbf{D}$ that we seek is obtained from Ξ using the generalized Einstein relationship

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{tt} & \mathbf{D}_{tr}^T \\ \mathbf{D}_{tr} & \mathbf{D}_{rr} \end{pmatrix} = k_B T \Xi^{-1} \tag{14}$$

where $\mathbf{D}$, like Ξ, has been also partitioned into 3×3 blocks [36]. The translational (D_t) and rotational (D_r) diffusion coefficients are then obtained as follows

$$D_t = 1/3 \ \mathrm{Tr}(\mathbf{D}_{tt}) \tag{15}$$

$$D_r = 1/3 \ \mathrm{Tr}(\mathbf{D}_{rr}) \tag{16}$$

where the symbol Tr indicates the trace of a tensor. Consequently, translational (f_t) and rotational (f_r) friction coefficients can be obtained using the following equations [36]

$$f_t = k_B T / D_t \tag{17}$$

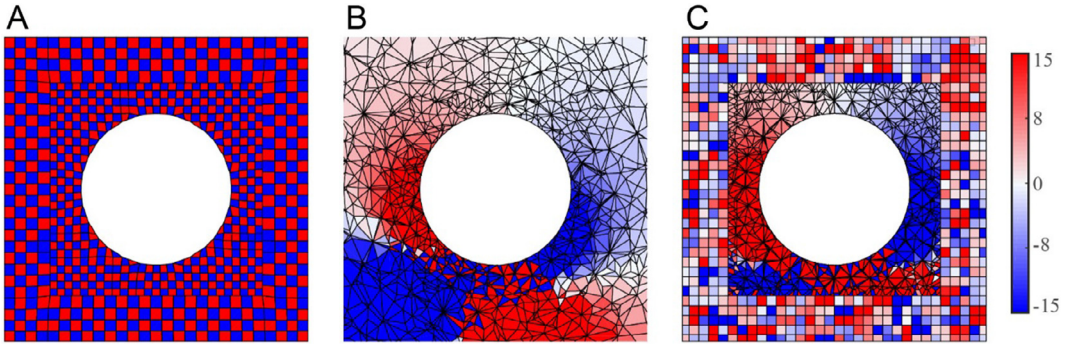

Fig. 3. Pressure distributions of the solvent model around a spherical body. (A) All layers are discretized using 8/1 hexahedral elements. (B) All layers are discretized using 10/1 tetrahedral elements. (C) Only the first layer is discretized using 10/1 tetrahedral elements while other layers are composed of 8/1 hexahedral elements. The unit for the pressure is 10^{-3} Da/(ps^2 Å).

$$f_r = k_B T / D_r \tag{18}$$

The blocks $\mathbf{D}_{tt}$ and $\mathbf{D}_{tr}$ depend on the origin of the Cartesian system, while $\mathbf{D}_{rr}$ does not vary with the change of origin [55]. However, the diffusion matrix $\mathbf{D}$ has to be calculated at the center of diffusion, C_d, to best match the experimental diffusion coefficients [56], where C_d is a location at which $\mathbf{D}_{tr}$ is symmetric [55]. Hence, we first compute the matrix at some arbitrary origin, O, and then transfer the origin to C_d using $\mathbf{r}_{OD}$, the position vector of C_d with respect to O, and recalculate the matrix [36]

$$\mathbf{r}_{OD} = \begin{pmatrix} x_{OD} \\ y_{OD} \\ z_{OD} \end{pmatrix} = \begin{pmatrix} D_{rr}^{yy} + D_{rr}^{zz} & -D_{rr}^{xy} & -D_{rr}^{xz} \\ -D_{rr}^{xy} & D_{rr}^{xx} + D_{rr}^{zz} & -D_{rr}^{yz} \\ -D_{rr}^{xz} & -D_{rr}^{yz} & D_{rr}^{xx} + D_{rr}^{yy} \end{pmatrix}^{-1} \begin{pmatrix} D_{tr}^{yz} - D_{tr}^{zy} \\ D_{tr}^{zx} - D_{tr}^{xz} \\ D_{tr}^{xy} - D_{tr}^{yx} \end{pmatrix} \tag{19}$$

We can use this diffusion matrix to validate the friction matrix as illustrated below.

2.5. Brownian dynamics simulation using the finite element model

The Brownian dynamics equations are obtained by neglecting the inertial forces in Eq. (3)

$$\mathbf{Z}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \tag{20}$$

The Brownian modes are computed solving the generalized eigenvalue problem

$$\mathbf{Z}\boldsymbol{\varphi}_i = \tau_i \mathbf{K}\boldsymbol{\varphi}_i \tag{21}$$

where $\boldsymbol{\varphi}_i$ is the i^{th} Brownian mode, and τ_i is the i^{th} eigenvalue, that is, the relaxation time associated with $\boldsymbol{\varphi}_i$.

The Brownian dynamics Eq. (20) is solved using the trapezoidal rule, an implicit time integration scheme. The difficulties with this scheme enumerated in Ref. [57] are not present in this problem solution and hence the trapezoidal rule is an effective scheme to use.

For the solution, we rewrite the Brownian dynamics equation into the form

$$\mathbf{Z}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{C}\mathrm{d}\mathbf{w} \tag{22}$$

$$\mathbf{C} = \begin{bmatrix} \widetilde{\mathbf{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{23}$$

where $\mathbf{f}(t)$ is expressed in a form that we can more easily deal with in the numerical integration.

In Eq. (23), $\mathbf{C}$ is a $3N \times 3N$ matrix and $\widetilde{\mathbf{C}}$ is a $3M \times 3M$ matrix, where N denotes the total number of nodes in the protein finite element model. The matrix $\widetilde{\mathbf{C}}$ is given by the Cholesky decomposition of $2k_B T\widetilde{\mathbf{Z}}$

$$\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^{\mathrm{T}} = 2k_B T\widetilde{\mathbf{Z}} \tag{24}$$

The vector $\mathrm{d}\mathbf{w}$ is a random vector of Gaussian distribution with zero mean and variance of $1/\Delta t$, where Δt is the time-step size used in the step by step solution [58]. Since this vector is a random vector, we performed multiple Brownian dynamics simulations in Sections 3.2.1 and 3.2.2.

The calculation of the displacement and the velocity vectors is now achieved through the following steps:

(1) Calculate the effective load vector ${}^{t+\Delta t}\hat{\mathbf{R}}$ at time $t + \Delta t$ using

$${}^{t+\Delta t}\hat{\mathbf{R}} = \mathbf{C}\mathrm{d}^{t+\Delta t}\mathbf{w} + \mathbf{Z}[(2/\Delta t)^{t}\mathbf{x} + {}^{t}\dot{\mathbf{x}}] \tag{25}$$

(2) Solve for the displacement vector ${}^{t+\Delta t}\mathbf{x}$ at time $t + \Delta t$ using

$$\hat{\mathbf{K}}^{t+\Delta t}\mathbf{x} = {}^{t+\Delta t}\hat{\mathbf{R}} \quad (\text{where} \quad \hat{\mathbf{K}} = \mathbf{K} + (2/\Delta t)\mathbf{Z}) \tag{26}$$

(3) Compute the velocity vector ${}^{t+\Delta t}\dot{\mathbf{x}}$ at time $t + \Delta t$ using

$${}^{t+\Delta t}\dot{\mathbf{x}} = (2/\Delta t)({}^{t+\Delta t}\mathbf{x} - {}^{t}\mathbf{x}) - {}^{t}\dot{\mathbf{x}} \tag{27}$$

The time trajectories of a protein finite element model are thus obtained for a predefined simulation duration and time step size.

3. Results and discussion

In this section, we present the results that we have obtained using the procedure described above.

3.1. Diffusion coefficients

As we mentioned above, the calculation of the diffusion coefficients is a means to validate the friction matrix. We present here various results.

3.1.1. Diffusion coefficients of a sphere

Since there are analytical solutions for the diffusion coefficients of a sphere [36], we can examine, as a validation, the accuracy of the proposed finite element procedure in calculating the friction matrix of a sphere. This study also helps to identify how to effectively model proteins within water.

We calculate the diffusion coefficients of a sphere of radius r_{in} = 25 Å located in 20 °C water. We embed the sphere in an incompressible medium, modeling water, of spherical geometry with radius r_{out}. Poisson's ratio of the medium is set to 0.4999 and its shear modulus is chosen equal to 60.34 Da/(ps^2 Å) based on the viscosity, μ, of water at 20 °C [59]. Considering the dynamic behavior and molecular properties of proteins, dalton, angstrom, and picosecond (1 ps = 10^{-12} s) are chosen as units for the mass, length, and time scales, respectively.

As described above, the space between the inner and outer spherical surfaces is meshed with 10/1 elements [12], see Fig. 4. The element size varies from the finest (at the inner surface) to the coarsest (at the outer surface) level in several layers.

The error in the calculated diffusion coefficients of the sphere is evaluated using

$$\mathrm{Error}_{t/r} = 100\left|\frac{D_{t/r}^{anal} - D_{t/r}^{calc}}{D_{t/r}^{anal}}\right|\% \tag{28}$$

where $D_{t/r}^{anal}$ is the reference value of the translational (t) or rotational (r) diffusion coefficients obtained from the analytical solutions, and $D_{t/r}^{calc}$ is the calculated value obtained using our finite element procedure. The reference values of the translational and rotational diffusion coefficients of the sphere with the radius of 25 Å in 20 °C water obtained from analytical solutions [36] are

$$D_t^{anal} = \frac{k_B T}{6\pi\mu r_{in}} = 8.57 \times 10^{-3}\ \text{Å}^2/\text{ps} \tag{29}$$

$$D_r^{anal} = \frac{k_B T}{8\pi\mu r_{in}^3} = 1.03 \times 10^{-5}\ \text{ps}^{-1} \tag{30}$$

To simulate the zero velocity and pressure at infinity, the movement of the nodes located on the outer spherical surface is restrained in all three directions, except for a fraction of nodes r_{free} that are free to move. The results show that changing r_{free}, in the range of 0.1–0.9, has almost no effect on the errors in the calculated diffusion coefficients (Fig. 5). We thus used r_{free} = 0.2 in our analyses throughout the paper, and which nodes are actually chosen makes almost no difference to the results.

However, changing the ratio of r_{out} to r_{in} from 5 to 400 significantly decreases the errors in the calculated diffusion coefficients, while the errors remain almost constant for ratios

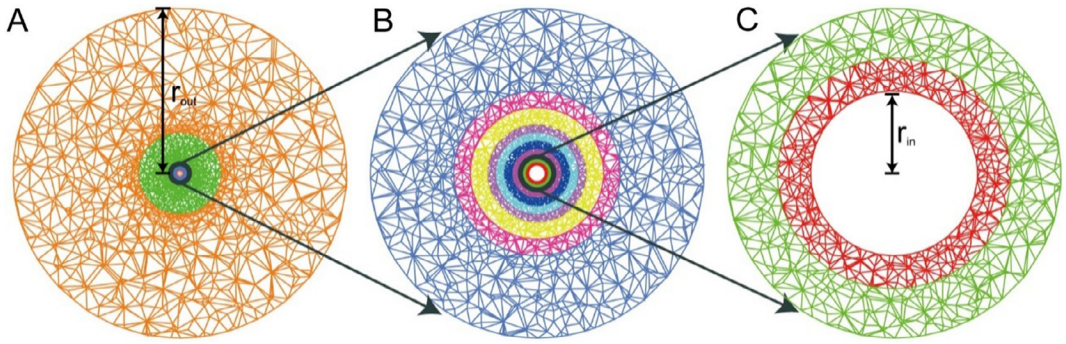

Fig. 4. The mesh between the inner and outer spherical surfaces (in cross-section). Finite element meshes are shown for (A) all the layers, (B) 9 layers, and (C) 2 layers from the inner sphere. Different colors indicate different layers of the mesh.

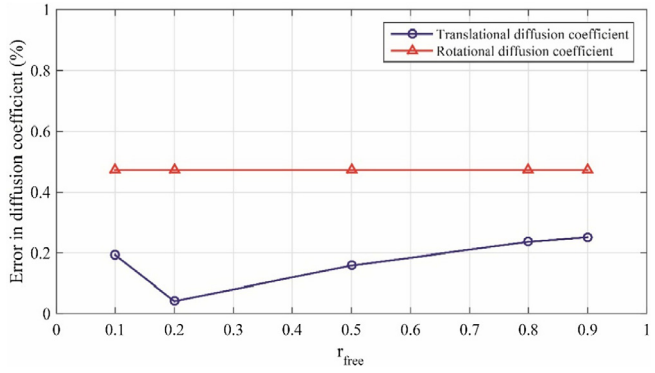

Fig. 5. Errors in the calculated translational and rotational diffusion coefficients of a sphere as a function of r_{free}. Here $r_{out} = 400 \times r_{in}$ and $h/r_{in} = 0.18$ where h is element size near the inner surface.

greater than 400 (Fig. 6). Since larger ratios substantially increase the cost of computations, we use the ratio 400 for all finite element simulations.

Using $r_{free} = 0.2$ and $r_{out}/r_{in} = 400$, we established the convergence of the calculated diffusion coefficients as we refine the mesh. The element sizes of all the layers are defined based on the element size near the inner surface, h. When we change h/r_{in} from 0.33 to 0.18, the error in the translational diffusion coefficient reduces from 0.39% to 0.04%, and error in the rotational diffusion coefficient decreases from 0.67% to 0.47% (Fig. 7). Since the computational cost of using 0.18 is reasonable (the number of elements employed in the finite element model is about 100,000), we use the ratio of about 0.18 for the analysis of all proteins.

3.1.2. Diffusion coefficients of proteins

Using the above procedure, we calculated the rotational and translational diffusion coefficients of 10 different proteins (Table 1). The solvent-excluded surface of the protein is embedded in the incompressible medium (water) modeled with about 100,000 10/1 tetrahedral elements. All the experimental and calculated diffusion coefficients are given for the viscosity of water at 20 °C, except for the rotational diffusion coefficients of myoglobin and hemoglobin (Table 1) which are provided for the viscosity of 66.24 Da/(ps Å). As shown, the calculated diffusion coefficients match very well with the experimentally measured data, particularly considering the case of translation.

3.2. Brownian dynamics simulation

To illustrate the Brownian dynamics simulation, we considered two representative proteins: Lysozyme and Taq polymerase. The initial structures of HEW Lysozyme (14.4 kDa) and the Taq polymerase (94.4 kDa) are taken, respectively, from the works by Vaney et al. [61] (Protein Data Bank ID 193L) and by Kim et al. [62] (Protein Data Bank ID 1TAQ). We first present the results for Taq polymerase because the experimental data providing the time scale to be simulated are available for this point, and the modeling experience is used for the analysis of Lysozyme. A much larger protein, GroEL (810.1 kDa), is also analyzed to demonstrate the computational efficiency of the proposed method.

3.2.1. Taq polymerase

Taq polymerase is comprised of 832 residues, that is, blocks of atoms (Fig. 8) [62]. The largest relaxation time of Taq polymerase is ~537 ps. We simulated the response for 300,050 ps, which is about 559 times the relaxation time. This was intended for the

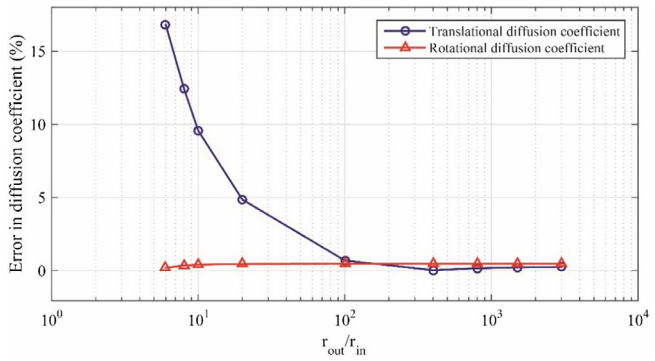

Fig. 6. Errors in the calculated translational and rotational diffusion coefficients of a sphere as a function of the ratio of r_{out} to r_{in}. Here $r_{free} = 0.2$ and $h/r_{in} = 0.18$.

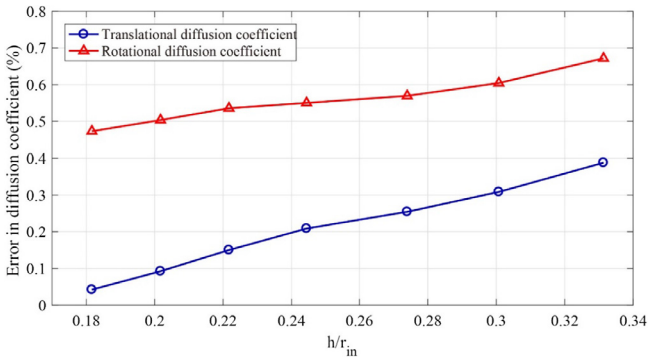

Fig. 7. Errors in the calculated translational and rotational diffusion coefficients of the sphere as a function of the ratio of h to r_{in}. Here, we use r_{free} 0.2 and r_{out}/r_{in} 400.

Table 1
Experimental and calculated values of the translational and rotational diffusion coefficients for 10 different proteins whose atomic structures are obtained from the Protein Data Bank (PDB). All the reported experimental diffusion coefficients are obtained from Ref. [60], except for those of myoglobin and hemoglobin obtained from Ref. [11].

Rotational diffusion coefficient (10^{-5} ps^{-1})		Translational diffusion coefficient (10^{-2} Å^2/ps)		Structure weight (kDa)	PDB ID	Protein name
Model	Experiment	Model	Experiment			
4.87	4.17	1.46	1.29	7	4PTI	BPTI (q)
2.46	N.A.	1.17	1.07	14	1RBX	Ribonuclease A
2.63	2.6	1.17	1.09	14	193L	Lysozyme
1.97	1.72	1.08	1.02	18	1DWR	Myoglobin
1.36	N.A.	0.95	0.93	26	2CGA	Chymotrypsinogen
0.84	0.75	0.81	0.78	37	1BEB	B-Lactoglobulin
0.61	0.44	0.72	0.69	65	2DN2	Hemoglobin
0.23	N.A.	0.52	0.5	143	2GPD	GPD (r)
0.18	N.A.	0.49	0.45	158	1ADO	Adolase
0.16	N.A.	0.46	0.4	233	2MIN	Nitrogenase MoFe

simulation data to be compared with the experimental dynamic form factor data, which are available up to ~30,000 ps (Fig. 8D and Appendix A.1). We performed, in total, 10 Brownian dynamics simulations of Taq polymerase with the time-step size of 1 ps. We excluded the results of the first 50 ps and used the results for the next 300,000 ps for sampling the coordinates.

We also obtained the root-mean-square fluctuations of residues from the Brownian dynamics solution of the protein and compared the results with those calculated using the block normal mode analysis, molecular dynamics simulation, and normal mode analysis based on the finite element method (see Ref. [43], Appendix A.2, and Fig. 8). The root-mean-square fluctuations of each residue is calculated at the location of its corresponding α-carbon atom.

The residues located at the tips of the two main domains of the protein are flexible and residues D496 and P226 have the highest root-mean-square fluctuations in the two domains. It is known

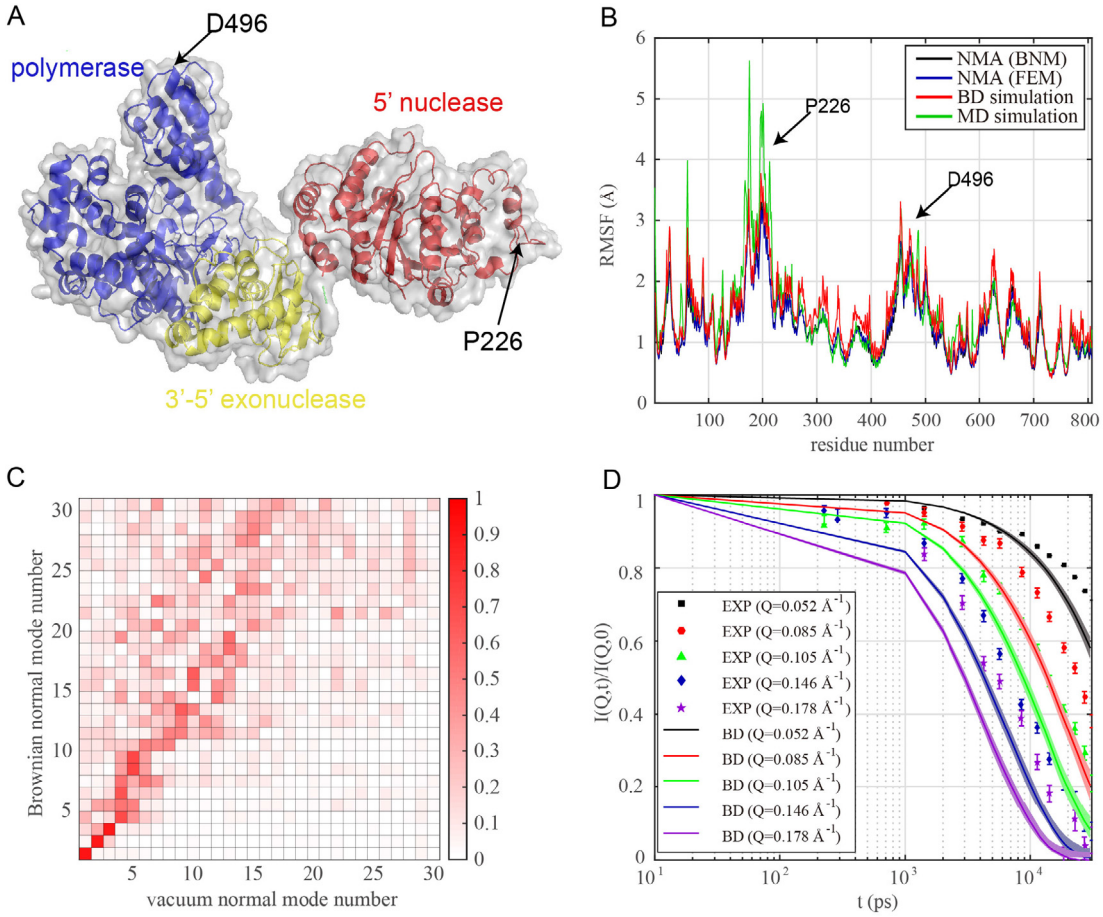

Fig. 8. Results for Taq polymerase. (A) Schematic representation of the molecular structure analyzed. Three main domains of Taq polymerase, 5′ nuclease (*red*), 3′-5′ exonuclease (*yellow*), and polymerase (*blue*), are shown. (B) Root-mean-square fluctuations (RMSFs) of residues obtained using the Brownian dynamics (BD) trajectories (the scheme proposed in this paper, *red curve*) compared with those calculated using the molecular dynamics (MD) simulation and two normal mode analysis (NMA) approaches: block normal mode (BNM) analysis and finite element method (FEM) (*green, black and blue curves*). Residues D496 and P226 and their corresponding root-mean-square fluctuations are denoted in the figure. (C) Mode overlap matrix between the first 30 Brownian normal modes and vacuum normal modes. (D) Normalized dynamic form factor, $I(Q,t)/I(Q,0)$, calculated (*solid lines*) for scattering vector lengths of 0.052 Å^{-1} (*black*), 0.085 Å^{-1} (*red*), 0.105 Å^{-1} (*green*), 0.146 Å^{-1} (*blue*), and 0.178 Å^{-1} (*purple*) and compared with experimental data (*markers*) from Ref. [64]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Pearson correlation [65,66] of root-mean-square fluctuations of Taq polymerase obtained using the Brownian dynamics (BD) simulation, the molecular dynamics (MD) simulation, finite element method (FEM) and block normal mode (BNM) analysis (see Fig. 8B).

	BD	MD	FEM	BNM
BD	1.00	–	–	–
MD	0.99	1.00	–	–
FEM	0.89	0.79	1.00	–
BNM	0.97	0.98	0.88	1.00

that the relative motion of these two domains toward each other is functionally important [63]. We also calculated the Pearson correlation between the results of these modeling approaches (Table 2), where we can see high correlations of predicted root-mean-square fluctuations.

Considering the Brownian dynamics simulation the eigenvectors of $\mathbf{Z}\boldsymbol{\varphi}_i = \tau_i \mathbf{K}\boldsymbol{\varphi}_i$ and the "in vacuum simulation" eigenvectors of $\mathbf{K}\boldsymbol{\psi}_j = \lambda_j \mathbf{M}\boldsymbol{\psi}_j$, we can compare the modes by projecting the Brownian normal modes onto the vacuum normal modes at the specific locations of the α-carbon atoms (where we exclude rigid body modes). The mode overlap coefficient between the Brownian normal mode i and the vacuum normal mode j is defined as $p_{ij} = \boldsymbol{\varphi}_i^{\alpha} \cdot \boldsymbol{\psi}_j^{\alpha} / (|\boldsymbol{\varphi}_i^{\alpha}||\boldsymbol{\psi}_j^{\alpha}|)$, where the superscript denotes that the projected vectors are used. The mode overlap matrix of Taq polymerase shows a significant discrepancy between its Brownian normal modes and vacuum normal modes (Figs. 8C and 9), demonstrating the importance of considering the solvent damping effect. The first three modes are similar to each other, but the higher modes do not show that similarity.

Finally, using the Brownian dynamics trajectories, the dynamic form factors of Taq polymerase (Appendix A.1) were calculated for a wide range of scattering vector lengths and compared with experimental results obtained from Ref. [64] (Fig. 8D). As seen, the simulation results are in good agreement with experimental results, which further illustrates the effectiveness of our proposed framework for finite element simulations of proteins.

3.2.2. Lysozyme

Lysozyme is a 129-residue-long enzyme, whose structure and function have been studied extensively over the last decades [67–72]. The largest relaxation time of Lysozyme, corresponding to the first non-rigid body Brownian normal mode, is ~119 ps. To accurately compute the equilibrium quantities of Lysozyme from its Brownian dynamics trajectories, we simulate each trajectory for ~66,492 ps. We performed, in total, 10 Brownian dynamics simulations of Lysozyme with the time-step size of 0.22 ps. The first 12 ps was excluded and the next 66,480 ps was used for sampling the coordinates. Here we determined the appropriate simulation time and time step by employing the same ratios to the maximum relaxation time as we used in the Taq polymerase simulations. For example, the simulation time is the 559 times the largest relaxation time, and the time step is 1/537 times the largest relaxation time, see Section 3.2.1.

We obtained the root-mean-square fluctuations of residues from the Brownian dynamics trajectories of the protein and compared the results with those calculated using the block normal mode analysis, molecular dynamics simulation, and normal mode analysis based on the finite element method (Fig. 10). The Pearson correlation between the root-mean-square fluctuations results of each modeling approach is provided in Table 3, which shows high correlations between these approaches. In addition, as in the results for Taq polymerase, the mode overlap matrix of Lysozyme shows a significant discrepancy between its Brownian normal modes and vacuum normal modes (Figs. 10C and 11). The first vacuum normal mode is more similar to the fourth Brownian normal mode than the first Brownian normal mode. The effect of solvent damping cannot be ignored in a normal mode analysis of Lysozyme and can significantly change a mode shape. Nevertheless, the vacuum normal modes have been used extensively in the literature to analyze the dynamical behavior of Lysozyme (see for example Refs. [68,71]).

3.2.3. Computational efficiency

The proposed finite element procedure for the analysis of proteins has important advantages over the classical molecular dynamics simulation in computational efficiency. Firstly, the total number of degrees of freedom in the model based on finite elements does not increase with the molecular size or weight unlike for the atomistic model of a molecular dynamics simulation. Since the low mode response of a protein is largely governed by only its overall shape, appropriate finite element models are constructed using a similar number of nodes for proteins of similar shapes regardless of the size and weight of the shapes [12]. Hence, the number of atoms comprising a protein is not a direct consideration in the finite element modeling of a protein for a Brownian dynamics simulation.

Secondly, a much larger time step can be used in the proposed method. In general, the largest relaxation time increases with the molecular size, and hence the physical time to be simulated becomes longer for a larger molecule. In the molecular dynamics simulation, the time step size must be of order of a femtosecond ($=10^{-15}$ s) regardless of molecular size in order to accurately represent short-time scale vibrations of light atoms such as hydrogens. This fact significantly limits the use of molecular dynamics simulations of macromolecular protein structures. In the finite-element-based Brownian dynamics simulation, however, we choose a time step size based on the relaxation time. For example, we can fix the number of time steps to be used to simulate the response within the largest relaxation time. Then the computational effort is independent of the relaxation time and, consequently, the molecular weight. Therefore, we can simulate macromolecular proteins with a similar computational effort as required for small proteins.

To illustrate these points regarding the proposed solution scheme we compare the required computational times for the

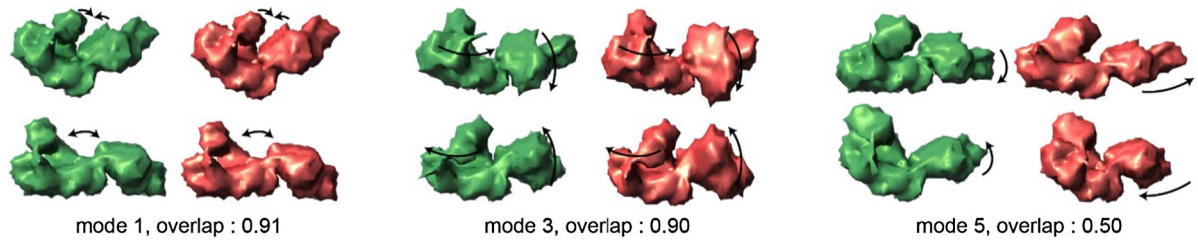

Fig. 9. Several normal modes of Taq Polymerase. Brownian normal modes and vacuum normal modes are depicted in *green* and *red*, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

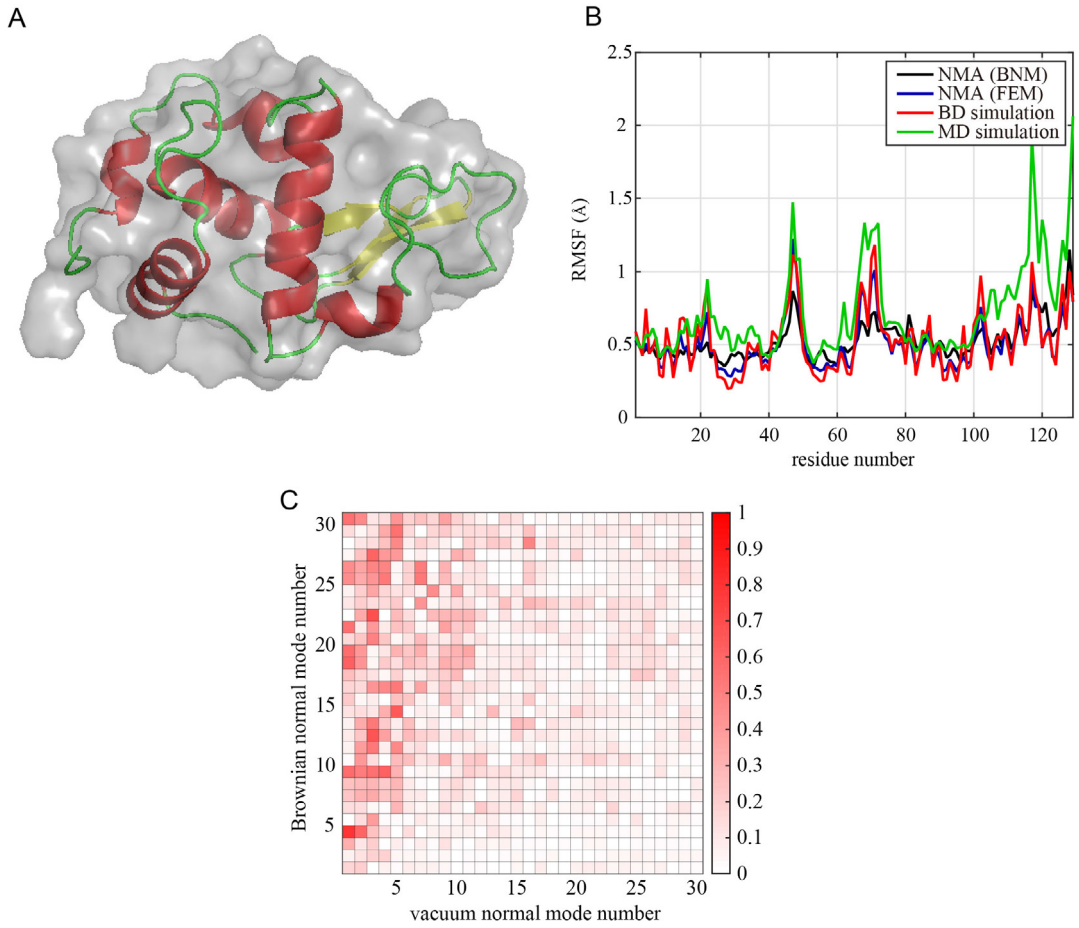

Fig. 10. Results for Lysozyme. (A) Schematic representation of the molecular structure analyzed. (B) Root-mean-square fluctuations (RMSFs) of residues obtained using the Brownian dynamics (BD) trajectories (*red curve*) compared with those calculated using the molecular dynamics (MD) simulation and two normal mode analysis (NMA) approaches: block normal mode analysis (BNM) and finite element method (FEM) (*green, black and blue curves*). (C) Mode overlap matrix between the first 30 Brownian normal modes and vacuum normal modes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Pearson correlation of root-mean-square fluctuations of Lysozyme obtained using the Brownian dynamics (BD) simulation, the molecular dynamics (MD) simulation, finite element method (FEM) and block normal mode (BNM) analysis (see Fig. 10B).

	BD	MD	FEM	BNM
BD	1.00	–	–	–
MD	0.82	1.00	–	–
FEM	0.81	0.79	1.00	–
BNM	0.75	0.95	0.71	1.00

analyses of three proteins including GroEL whose molecular weight is much larger than the molecular weights of Taq polymerase and Lysozyme analyzed in the previous sections (Table 4 and Fig. 12). It took about two days to perform the proposed Brownian dynamics simulation on a PC for any of the three proteins. However, the molecular dynamics simulation for the same physical time spans as used in the Brownian dynamics simulation would take about two days for Lysozyme, a few weeks

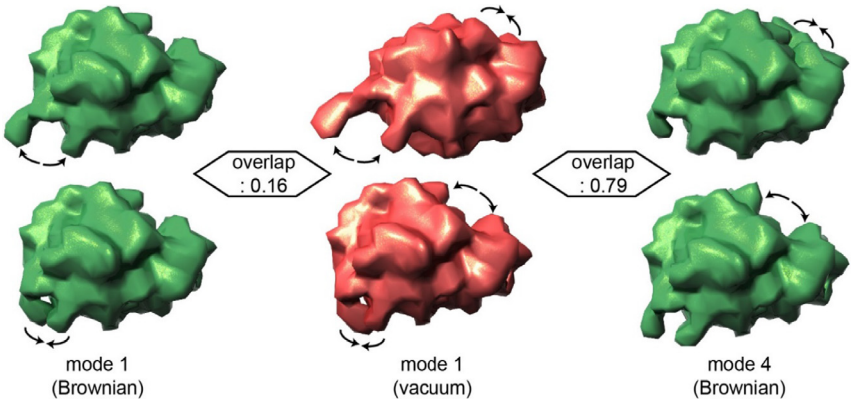

Fig. 11. Several normal modes of Lysozyme.

Table 4
Analysis information of Brownian dynamics simulation and molecular dynamics simulation.

Protein	Lysozyme	Taq polymerase	GroEL
Molecular weight (kDa)	14.4	94.4	810.1
	Brownian dynamics simulation (PC: Intel® Xeon® CPU E5-2407 with 2.2 GHz (CPU))		
Number of nodes	11,279	9,756	12,827
Largest relaxation time (ps)	119	537	60,792
Simulated physical time (ps) (∼559 × largest relaxation time)	66,492	300,050	33,967,671
Time step size (ps)	0.22	1.0	133.21
Computational time (h)	45.79	31.50	49.37
	Molecular dynamics simulation (GPU-cluster: Intel® Xeon® CPU E5-2630 with 2.4 GHz + NVIDIA® Tesla® K80, 6EA)		
Number of atoms (including solvent atoms)	27,536	162,761	535,480
Time step size (ps)	0.002	0.002	0.002
Computational time for 10,000-ps-long simulation (h)	4.96	19.68	68.55
Estimated computational time required to simulate the same physical time as in the Brownian dynamics simulation (h)	33.00	590.58	232,865.90

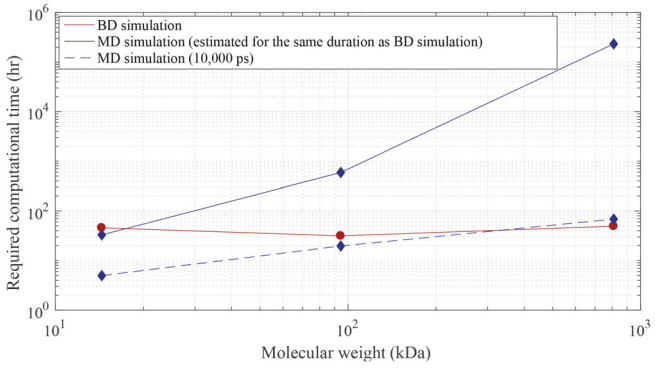

Fig. 12. Comparison of the computational time required in Brownian dynamics (BD) simulation (*solid red*) and molecular dynamics (MD) simulation (*solid blue*) for the simulation times that are ∼559 times the largest relaxation times, while the dashed blue line indicates the computational time required for 10,000-ps-long molecular dynamics simulations for the three proteins (see Table 4). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for Taq polymerase, and many years for GroEL. Indeed, the simulation of GroEL would still take years even when performed on a GPU server.

4. Concluding remarks

The functional motions of proteins in solvents are hardly accessible to traditional molecular dynamics analyses. In this paper, we present a finite element framework to analyze the realistic dynamical behavior of proteins in solvents. We showed that the proposed approach is effective in studying the functional mechanism of proteins such as Lysozyme and Taq polymerase on a regular personal computer (PC) with a reasonable computational effort. An important feature of this approach is that the computational cost does not change with the molecular size, which enables researchers to analyze the long-time-scale dynamics of large bio-macromolecules. We illustrated this feature by the analysis of the protein GroEL. In addition, we have shown that the mode shapes of proteins are highly sensitive to the solvent-damping effect, a condition usually ignored in the analysis of proteins due to computational cost.

Acknowledgement

This work was partly supported by the Creative Materials Discovery Program (Grant No. 2017M3D1A1039422), the Global Frontier R&D Program on Center for Wave Energy Control based on Metamaterials (Grant No. 2014M3A6B3063711), and the Basic Science Research Program (Grant No. 2016R1C1B2011098) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT.

Appendix A

A.1. Dynamic form factor

From the calculated molecular trajectories, we can compute the dynamic form factor $I(Q,t)$ [73] providing insight and information about the predicted protein motions. It can be obtained as follows

$$I(Q,t) = \sum_{i,j}^{L} b_{i,coh} b_{j,coh} \langle \exp[-i\mathbf{Q}\cdot\mathbf{r}_i(0)] \exp[i\mathbf{Q}\cdot\mathbf{r}_j(t)] \rangle \quad \text{(A1)}$$

where $b_{i,coh}$ is the coherent neutron scattering length of effective residue i (Table A1), calculated by adding up the coherent scattering lengths of atoms for each residue, $\mathbf{Q}$ is the scattering vector, $\mathbf{r}_i(t)$ is the average coordinate of the atoms in the effective residue at time t, and L is the number of residues.

Table A1
The coherent scattering lengths of effective residues [74].

Residue	Coherent neutron scattering length (Å)	Residue	Coherent neutron scattering length (Å)
Alanine	16.423×10^{-5}	Leucine	13.933×10^{-5}
Arginine	38.302×10^{-5}	Lysine	19.562×10^{-5}
Asparagine	34.507×10^{-5}	Methionine	17.570×10^{-5}
Asparatic acid	34.680×10^{-5}	Phenylalanine	41.386×10^{-5}
Cysteine	19.227×10^{-5}	Proline	22.224×10^{-5}
Glutamine	33.677×10^{-5}	Serine	22.226×10^{-5}
Glutamic acid	33.850×10^{-5}	Threonine	21.396×10^{-5}
Glycine	17.253×10^{-5}	Tryptophan	60.299×10^{-5}
Histidine	47.635×10^{-5}	Tyrosine	47.170×10^{-5}
Isoleucine	13.933×10^{-5}	Valine	14.763×10^{-5}

The scattering vector can be written in terms of Q and two Euler angles (θ and φ) where θ is the angle between the x_3-axis and the scattering vector, φ is the angle between the x_1-axis and the projection of the scattering vector on the x_1-x_2 plane, and Q is the length of the scattering vector. Also, $\vec{\mathbf{e}}_1$, $\vec{\mathbf{e}}_2$ and $\vec{\mathbf{e}}_3$ are the basis vectors of the Cartesian coordinate system

$$\mathbf{Q} = Q(\sin(\theta)\cos(\varphi)\vec{\mathbf{e}}_1 + \sin(\theta)\sin(\varphi)\vec{\mathbf{e}}_2 + \cos(\theta)\vec{\mathbf{e}}_3) \tag{A2}$$

Hence, Eq. (A1) is simplified as follows.

$$\begin{aligned} I(Q,t) &= \sum_{i,j}^{L} b_{i,coh} b_{j,coh} \langle \exp[i\mathbf{Q}\cdot(\mathbf{r}_j(t) - \mathbf{r}_i(0))] \rangle \\ &= \sum_{i,j}^{L} b_{i,coh} b_{j,coh} \frac{\int_0^{2\pi}\int_0^{\pi} \exp[i\mathbf{Q}\cdot(\mathbf{r}_j(t) - \mathbf{r}_i(0))]\sin\theta\, d\theta\, d\varphi}{\int_0^{2\pi}\int_0^{\pi} \sin\theta\, d\theta\, d\varphi} \\ &= \sum_{i,j}^{L} b_{i,coh} b_{j,coh} \frac{\int_0^{2\pi}\int_0^{\pi} \exp[iQ|\mathbf{r}_j(t) - \mathbf{r}_i(0)|\cos\theta]\sin\theta\, d\theta\, d\varphi}{4\pi} \\ &= \sum_{i,j}^{L} b_{i,coh} b_{j,coh} \frac{\sin(Q|\mathbf{r}_j(t) - \mathbf{r}_i(0)|)}{Q|\mathbf{r}_j(t) - \mathbf{r}_i(0)|} \end{aligned} \tag{A3}$$

A.2. Root-mean-square fluctuations

Root-mean-square fluctuations have been widely used to characterize the conformational variance of a protein structure, and can be computed from the molecular trajectories obtained using a molecular dynamics or Brownian dynamics simulation, or from the mode shapes and frequencies of the protein in vacuum (obtained using a normal mode analysis based on the block normal mode method or finite element method).

The root-mean-square fluctuation values of atom i can be expressed as

$$\langle \Delta r_i^2 \rangle^{1/2} = \langle (x_i(t) - \bar{x}_i)^2 + (y_i(t) - \bar{y}_i)^2 + (z_i(t) - \bar{z}_i)^2 \rangle^{1/2} \tag{A4}$$

where $x_i(t)$, $y_i(t)$, and $z_i(t)$ denote the atomic coordinates of atom i at time t with translational and rotational rigid body motions removed, and $\bar{x}_i$, $\bar{y}_i$, and $\bar{z}_i$ denote the mean position of atom i over the simulation time.

Alternatively, root-mean-square fluctuations can also be computed from the results of normal mode solutions using the following equation derived from the equipartition theorem in statistical mechanics [66]

$$\langle \Delta r_i^2 \rangle^{1/2} = \left(\frac{k_B T}{m_i} \sum_k \frac{1}{\lambda_k} \psi_{ik}^2 \right)^{1/2} \tag{A5}$$

where k_B is the Boltzmann constant, T is the temperature, m_i is the atomic mass of atom i, λ_k is the eigenvalue of mode k, and ψ_{ik}^2 is the squared magnitude of the displacement vector of atom i corresponding to mode k.

References

[1] Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. Nat Struct Biol 2002;9(9):646–52.
[2] Dill KA, Chan HS. From Levinthal to pathways to funnels. Nat Struct Biol 1997;4(1):10–9.
[3] Dinner AR, Šali A, Smith LJ, Dobson CM, Karplus M. Understanding protein folding via free-energy surfaces from theory and experiment. Trends Biochem Sci 2000;25(7):331–9.
[4] Shakhnovich EI. Theoretical studies of protein-folding thermodynamics and kinetics. Curr Opin Struct Biol 1997;7(1):29–40.
[5] Isralewitz B, Gao M, Schulten K. Steered molecular dynamics and mechanical functions of proteins. Curr Opin Struct Biol 2001;11(2):224–30.
[6] Best RB, Chen Y-G, Hummer G. Slow protein conformational dynamics from multiple experimental structures: the helix/sheet transition of arc repressor. Structure 2005;13(12):1755–63.
[7] Chu J-W, Voth GA. Coarse-grained free energy functions for studying protein conformational changes: a double-well network model. Biophys J 2007;93(11):3860–71.
[8] Maragakis P, Karplus M. Large amplitude conformational change in proteins explored with a plastic network model: adenylate kinase. J Mol Biol 2005;352(4):807–22.
[9] Zheng W, Brooks BR, Hummer G. Protein conformational transitions explored by mixed elastic network models. Proteins: Struct Funct Bioinf 2007;69(1):43–57.
[10] Lamm G, Szabo A. Langevin modes of macromolecules. J Chem Phys 1986;85(12):7334–48.
[11] Miller BT, Zheng W, Venable RM, Pastor RW, Brooks BR. Langevin network model of myosin. J Phys Chem B 2008;112(19):6274–81.
[12] Bathe KJ. Finite element procedures. 2nd ed.; 2014. Available at amazon.com and Higher Education Press, China.
[13] Ermak DL, McCammon J. Brownian dynamics with hydrodynamic interactions. J Chem Phys 1978;69(4):1352–60.
[14] Ando T, Chow E, Saad Y, Skolnick J. Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations. J Chem Phys 2012;137(6):064106.
[15] Ando T, Chow E, Skolnick J. Dynamic simulation of concentrated macromolecular solutions with screened long-range hydrodynamic interactions: algorithm and limitations. J Chem Phys 2013;139(12):121922.
[16] Fixman M. Construction of Langevin forces in the simulation of hydrodynamic interaction. Macromolecules 1986;19(4):1204–7.
[17] Geyer T, Winter U. An O (N^2) approximation for hydrodynamic interactions in Brownian dynamics simulations. J Chem Phys 2009;130(11):114905.
[18] Saadat A, Khomami B. Computationally efficient algorithms for incorporation of hydrodynamic and excluded volume interactions in Brownian dynamics simulations: a comparative study of the Krylov subspace and Chebyshev based techniques. J Chem Phys 2014;140(18):184903.
[19] Fayad GN, Hadjiconstantinou NG. Realistic Brownian dynamics simulations of biological molecule separation in nanofluidic devices. Microfluid Nanofluid 2010;8(4):521–9.
[20] Jian H, Vologodskii AV, Schlick T. A combined wormlike-chain and bead model for dynamic simulations of long linear DNA. J Comput Phys 1997;136(1):168–79.
[21] Klenin K, Merlitz H, Langowski J. A Brownian dynamics program for the simulation of linear and circular DNA and other wormlike chain polyelectrolytes. Biophys J 1998;74(2):780–8.
[22] Jian H, Schlick T, Vologodskii A. Internal motion of supercoiled DNA: Brownian dynamics simulations of site juxtaposition. J Mol Biol 1998;284(2):287–96.
[23] Sedeh RS, Pan K, Adendorff MR, Hallatschek O, Bathe KJ, Bathe M. Computing nonequilibrium conformational dynamics of structured nucleic acid assemblies. J Chem Theory Comput 2016;12(1):261–73.
[24] Gabdoulline RR, Wade RC. Brownian dynamics simulation of protein–protein diffusional encounter. Methods 1998;14(3):329–41.
[25] Huber GA, McCammon JA. Browndye: a software package for Brownian dynamics. Comput Phys Commun 2010;181(11):1896–905.
[26] Madura JD, Briggs JM, Wade RC, Davis ME, Luty BA, Ilin A, et al. Electrostatics and diffusion of molecules in solution: simulations with the University of Houston Brownian Dynamics program. Comput Phys Commun 1995;91(1–3):57–95.
[27] Długosz M, Zieliński P, Trylska J. Brownian dynamics simulations on CPU and GPU with BD_BOX. J Comput Chem 2011;32(12):2734–44.
[28] Geyer T. Many-particle Brownian and Langevin dynamics simulations with the Brownmove package. BMC Biophys 2011;4(1):7.
[29] Elcock AH. Molecular simulations of cotranslational protein folding: fragment stabilities, folding cooperativity, and trapping in the ribosome. PLoS Comput Biol 2006;2(7):824–42.
[30] Elcock AH. Molecule-centered method for accelerating the calculation of hydrodynamic interactions in Brownian dynamics simulations containing many flexible biomolecules. J Chem Theory Comput 2013;9(7):3224–39.
[31] Frembgen-Kesner T, Elcock AH. Striking effects of hydrodynamic interactions on the simulated diffusion and folding of proteins. J Chem Theory Comput 2009;5(2):242–56.
[32] Carrasco B, García de la Torre J, Zipper P. Calculation of hydrodynamic properties of macromolecular bead models with overlapping spheres. Eur Biophys J 1999;28(6):510–5.
[33] Carrasco B, García de la Torre J. Improved hydrodynamic interaction in macromolecular bead models. J Chem Phys 1999;111(10):4817–26.
[34] García de la Torre J, Carrasco B. Intrinsic viscosity and rotational diffusion of bead models for rigid macromolecules and bioparticles. Eur Biophys J 1998;27(6):549–57.
[35] Kottalam J, Case D. Langevin modes of macromolecules: applications to crambin and DNA hexamers. Biopolymers 1990;29(10–11):1409–21.
[36] Carrasco B, García de la Torre J. Hydrodynamic properties of rigid particles: comparison of different modeling and computational procedures. Biophys J 1999;76(6):3044–57.
[37] Bathe KJ, Zhang H. Finite element developments for general fluid flows with structural interactions. Int J Numer Meth Eng 2004;60(1):213–32.
[38] Bathe KJ. Advances in the multiphysics analysis of structures. In: Topping BHV, editor. Chapter 1 in computational methods for engineering science, e. B. Stirlingshire (Scotland): Saxe-Coburg Publications; 2012.

[39] Bathe KJ. Frontiers in finite element procedures and applications. In: Topping BHV, editor. Computational methods for engineering science, e. B. Stirlingshire (Scotland): Saxe-Coburg Publications; 2014.
[40] The ADINA finite element program. Watertown (MA, USA): ADINA R&D Inc; 2017. www.adina.com.
[41] Kim D-N, Nguyen C-T, Bathe M. Conformational dynamics of supramolecular protein assemblies. J Struct Biol 2011;173(2):261–70.
[42] Kim J, Kim J-G, Yun G, Lee P-S, Kim D-N. Toward modular analysis of supramolecular protein assemblies. J Chem Theory Comput 2015;11 (9):4260–72.
[43] Sedeh RS, Bathe M, Bathe KJ. The subspace iteration method in protein normal mode analysis. J Comput Chem 2010;31(1):66–74.
[44] Lu M, Ma J. The role of shape in determining molecular motions. Biophys J 2005;89(4):2395–401.
[45] Yun G, Kim J, Kim D-N. A critical assessment of finite element modeling approach for protein dynamics. J Comput Aided Mol Des 2017;31(7):609–24.
[46] Bathe M. A finite element framework for computation of protein normal modes and mechanical response. Proteins: Struct Funct Bioinf 2008;70 (4):1595–609.
[47] Tama F, Gadea FX, Marques O, Sanejouand YH. Building-block approach for determining low-frequency normal modes of macromolecules. Proteins: Struct Funct Bioinf 2000;41(1):1–7.
[48] Li G, Cui Q. A coarse-grained normal mode approach for macromolecules: an efficient implementation and application to Ca^{2+}-ATPase. Biophys J 2002;83 (5):2457–74.
[49] Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. J Comput Chem 1983;4(2):187–217.
[50] Sanner MF, Olson AJ, Spehner JC. Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 1996;38(3):305–20.
[51] Garland M. Quadric-based polygonal surface simplification. Carnegie Mellon University; 1999.
[52] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co; 1997.
[53] Heckbert PS, Garland M. Optimal triangulation and quadric-based surface simplification. Comput Geometry 1999;14(1–3):49–65.
[54] Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G. Meshlab: an open-source mesh processing tool. In: Proceedings of sixth eurographics Italian chapter conference, Fisciano, Italy; 2008. p 129–36.
[55] Brune D, Kim S. Predicting protein diffusion coefficients. Proc Natl Acad Sci 1993;90(9):3835–9.
[56] Harvey S, García De La Torre J. Coordinate systems for modeling the hydrodynamic resistance and diffusion coefficients of irregularly shaped rigid macromolecules. Macromolecules 1980;13(4):960–4.
[57] Bathe KJ, Noh G. Insight into an implicit time integration scheme for structural dynamics. Comput Struct 2012;98–99:1–6.
[58] Grassia P, Hinch E, Nitsche LC. Computer simulations of Brownian motion of complex systems. J Fluid Mech 1995;282:373–403.
[59] García de La Torre J, Huertas M, Carrasco B. HYDRONMR: prediction of NMR relaxation of globular proteins from atomic-level structures and hydrodynamic calculations. J Magn Reson 2000;147(1):138–46.
[60] García de La Torre J, Huertas ML, Carrasco B. Calculation of hydrodynamic properties of globular proteins from their atomic-level structure. Biophys J 2000;78(2):719–30.
[61] Vaney M, Maignan S, Ries-Kautt M, Ducruix A. High-resolution structure (1.33 Å) of a HEW lysozyme tetragonal crystal grown in the APCF apparatus. Data and structural comparison with a crystal grown under microgravity from SpaceHab-01 mission. Acta Crystallogr D Biol Crystallogr 1996;52(3):505–17.
[62] Kim Y, Eom SH, Wang J, Lee D-S. Crystal structure of Thermus aquaticus DNA polymerase. Nature 1995;376(6541):612.
[63] Ho DL, Byrnes WM, Ma W-P, Shi Y, Callaway DJ, Bu Z. Structure-specific DNA-induced conformational changes in Taq polymerase revealed by small angle neutron scattering. J Biol Chem 2004;279(37):39146–54.
[64] Bu Z, Biehl R, Monkenbusch M, Richter D, Callaway DJ. Coupled protein domain motion in Taq polymerase revealed by neutron spin-echo spectroscopy. Proc Natl Acad Sci USA 2005;102(49):17646–51.
[65] Na H, Song G. Conventional NMA as a better standard for evaluating elastic network models. Proteins: Struct Funct Bioinf 2015;83(2):259–67.
[66] Sedeh RS, Fedorov AA, Fedorov EV, Ono S, Matsumura F, Almo SC, et al. Structure, evolutionary conservation, and conformational dynamics of Homo sapiens fascin-1, an F-actin crosslinking protein. J Mol Biol 2010;400 (3):589–604.
[67] Blake CC, Mair G, North A, Phillips D, Sarma V. On the conformation of the hen egg-white lysozyme molecule. In: Proceedings of the royal society of London. Series B, biological sciences, vol. 167(1009); 1967. p. 365–77.
[68] Brooks B, Karplus M. Normal modes for specific motions of macromolecules: application to the hinge-bending mode of lysozyme. Proc Natl Acad Sci 1985;82(15):4995–9.
[69] de Groot B, Hayward S, van Aalten D, Amadei A, Berendsen H. Domain motions in bacteriophage T4 lysozyme: a comparison between molecular dynamics and crystallographic data. Proteins: Struct Funct Bioinf 1998;31(2):116–27.
[70] Horiuchi T, Gō N. Projection of Monte Carlo and molecular dynamics trajectories onto the normal mode axes: human lysozyme. Proteins: Struct Funct Bioinf 1991;10(2):106–16.
[71] Levitt M, Sander C, Stern PS. Protein normal-mode dynamics: trypsin inhibitor, crambin, ribonuclease and lysozyme. J Mol Biol 1985;181(3):423–47.
[72] Mchaourab HS, Oh KJ, Fang CJ, Hubbell WL. Conformation of T4 lysozyme in solution. Hinge-bending motion and the substrate-induced conformational transition studied by site-directed spin labeling. Biochemistry 1997;36 (2):307–16.
[73] Price D, Skold K. Neutron scattering. Elsevier Science; 1987.
[74] "The table of neutron scattering lengths and cross sections available on the NIST website (http://www.ncnr.nist.gov/resources/n-lengths/).".