

MBO(N)D: A Multibody Method for Long-Time Molecular Dynamics Simulations

HON M. CHUN,¹ CARLOS E. PADILLA,¹ DONOVAN N. CHIN,¹ MASAKATSU WATANABE,¹ VALERI I. KARLOV,¹ HOWARD E. ALPER,¹ KETO SOOSAAR,¹ KIM B. BLAIR,¹ OREN M. BECKER,² LEO S. D. CAVES,³ ROBERT NAGLE,^{4,*} DAVID N. HANEY,⁵ BARRY L. FARMER⁶

¹Moldyn, Inc., 955 Massachusetts Avenue, Cambridge, Massachusetts 02139

²School of Chemistry, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel

³Protein Structure Research Group, University of York, York YO15DD, United Kingdom

⁴Department of Chemistry, Harvard University, Cambridge, Massachusetts 02138

⁵Haney Associates, Inc., 5455 Westknoll Drive, La Jolla, California 92037

⁶Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433

Received 24 November 1998; accepted 14 September 1999

ABSTRACT: A modeling approach that can significantly speed up the dynamics simulation of large molecular systems is presented herein. A multigranular modeling approach, whereby different parts of the molecule are modeled at different levels of detail, is enabled by substructuring. Substructuring the molecular system is accomplished by collecting groups of atoms into rigid or flexible bodies. Body flexibility is modeled by a truncated set of body-based modes. This approach allows for the elimination of the high-frequency harmonic motion while capturing the low-frequency anharmonic motion of interest. This results in the use of larger integration step sizes, substantially reducing the computational time required for a given dynamic simulation. The method also

*Also at Orment, Inc., 12 Hubbard Avenue, Cambridge, MA 02140

Correspondence to: H. M. Chun; e-mail: info@moldyn.com

Contract/grant sponsor: Advanced Technology Program; contract/grant number: 70NANB5H1078

Contract/grant sponsor: National Science Foundation; contract/grant number: 9361843

Contract/grant sponsor: National Cancer Institute; contract/grant number: CA60211-03

Contract/grant sponsor: Photon Research Associates, Inc.

This article includes Supplementary Material available from the authors upon request or via the Internet at <ftp.wiley.com/public/journals/jcc/suppmat/21/159> or <http://journals.wiley.com/jcc/>

includes the use of a multiple time scale (MTS) integration scheme. Speed increases of 5- to 30-fold over atomistic simulations have been realized in various applications of the method. © 2000 John Wiley & Sons, Inc. J Comput Chem 21: 159–184, 2000

Keywords: molecular dynamics; normal modes; anharmonicity; macromolecules; numerical integrators

Introduction

BACKGROUND

This article describes the details of a new molecular dynamics (MD) method that substantially improves the computational speed of MD simulations. The goal is to enable simulations over the longer time frames required for the study of many important biomolecular events that are currently beyond routine treatment with classical all-atom MD methods. This new approach, MBO(N)D (Multi-Body Order (N) Dynamics), utilizes a multibody dynamics framework where groups of atoms are organized into interacting bodies to achieve long time steps (Δt). The multibody equations of motion are solved utilizing an $O(N)$ algorithm, the computational requirements of which scale linearly with the number of bodies in the system. MBO(N)D goes beyond rigid body dynamics methods reported to date,^{3–5} with the addition of internal body flexibility. Use of MBO(N)D's flexible body modeling approach facilitates both significant computational speed increases, factors of 5–30 on the 12–3,000 atom molecules studied to date, while reproducing the essential dynamic characteristics of the system.

Classical molecular dynamics (MD) propagates the motion of molecular models by solving the coupled classical equations of motion for all the atoms in the model. Unfortunately, due to the nature of the molecular interactions, solving these equations requires very fine time steps, which limit the simulation time duration. To guarantee stable numerical integration behavior, the time steps must be small enough to resolve the highest frequency of the system, normally about 1/20 the size of the fastest period. Because the highest frequency motions in biomolecules are associated with C—H, O—H, and N—H bond stretching (periods $P = 2\pi/\omega$ of about 10 fs), time steps on the order of $\Delta t = 0.5$ fs must be used. Consequently, MD simulations of biomolecules are severely constrained in time length.

The current length of MD simulations on moderate-to-high end workstations stand at a few

nanoseconds for a 1000-atom system. Although there are interesting transitions in the subnanosecond time range, many more biologically interesting motions occur in the micro- to millisecond time scale (e.g., allosteric transitions) or longer (e.g., folding, which may take seconds). Various techniques (see the next section) have been developed over the years to increase the time step in MD. Although significant, the factors of two to four computational speed improvement attained from these methods in general biomolecular applications are still far short of that required for the microsecond regime. Clearly, there is a need for new techniques that will significantly enhance the computational efficiency of the basic methodology of MD. The MBO(N)D approach we have developed will provide a framework for meeting this need.

CURRENT MD METHODS FOR ENABLING LONGER TIME STEPS

Many approaches have been developed in an attempt to overcome the strict time step limits in MD simulations so that long time-scale simulations can be routinely undertaken. Examples include constrained dynamics, multiple time-scale methods, eigenvector-based schemes, implicit integration schemes, and path optimization schemes. Only a brief overview of these methods is presented herein. For recent in-depth reviews on this topic, see refs. 6–9.

Constrained Dynamics

One way to overcome the MD time step problem is to replace high-frequency vibrations by algebraic constraints. This approach is represented by the SHAKE algorithm,¹⁰ which introduces constraints into the equation of motion using the formalism of Lagrange multipliers. With the fastest vibrations removed from the model, the integration time step can be lengthened to 1–2 fs. The most common implementation of SHAKE is an iterative scheme,¹⁰ where the individual constraints are imposed sequentially on the results of the unconstrained integration step.

A self-starting version of SHAKE, termed RATTLE, was later developed.¹¹ In general, such approaches allow a modest but significant increase in speed by a factor of 2, and are standard techniques applied in most MD studies. Van Gunsteren and Karplus¹² have shown that constraining the bond angles with SHAKE significantly modifies the dynamic properties of the system while constraining bond lengths does not significantly alter the dynamics. Note that computational advantage has to be balanced against the additional work required to solve the constraint equations.

In a different application of constrained dynamics, torsion angle dynamics,¹³ only dihedral angles are allowed to move, maintaining all bonds and bond angles fixed. The dynamic simulations are then performed by integrating the equations of motion in generalized coordinates.^{14–17} Applications to alanine nanopeptide¹⁷ succeeded in reproducing the atomistic simulation results (at $\Delta t = 1$ fs) of all atom trajectories using $\Delta t = 9$ fs for an all-atom representation of the molecule. Time steps as high as $\Delta t = 13$ fs were successfully used for simulations of an extended atom representation (i.e., after removing methyl group rotations). However, despite the significant time-step increase, application of these methods to large biomolecules is limited because a matrix inversion is required at every time step.

Another constraint approach based on internal coordinates was developed by Durup,¹⁸ and used a set of generalized internal coordinates, arranged in a “tree-like” topology. A set of holonomic constraints is used to freeze out the high-frequency motions. Time-step increases from 1 to 9 fs were reported for an application to citrate synthase. Though the method was applied to a search for reaction pathways in citrate synthase,¹⁹ it has not been adopted in general, because the choice of coordinates is not straightforward.

An extreme version of constrained dynamics is rigid body dynamics. In this approach, small groups of atoms are defined as rigid bodies effectively constraining the relative motion of all atoms contained within a body. The resulting dynamics only accounts for the relative translation and rotation between the bodies.³ This approach is often used as a sampling procedure and in the course of refining experimentally determined structures.⁴

Multiple Time-Scale Methods

Another approach to the integration time-step problem takes advantage of the multiscale character of protein motions. In what is known as multi-

ple time-scale (MTS) schemes,²⁰ a hierarchy of time steps is introduced instead of the traditional single time step. The idea is to take advantage of the terms in the biomolecular Hamiltonian that vary more slowly than other terms. Appropriate time steps are assigned for the different terms. A significant step forward in terms of biomolecular simulations was the introduction of MTS variants that exhibit time reversal symmetry.^{21, 22}

By applying MTS methods to a variety of systems, including the protein BPTI, using a splitting scheme based on bonding topology, Watanabe and Karplus^{23, 24} were able to obtain speed-up factors of about 2–4. The essential properties of the dynamics (with $\Delta t = 0.5$ fs as the shortest time step) were successfully replicated. Similar results were reported by Berne and collaborators,²⁵ who obtained a speed-up factor of 4 by splitting the nonbonded terms into fast and slow parts based on distance using a continuous switching function. Forester and Smith²⁶ reported MTS speed-up factors of 2–3 for liquid water and a solvated protein system. The MTS approach shows significantly better performance enhancements in systems where the separation of fast and slow motions is more pronounced than in biomolecules. The C_{60} molecule is a good example of such a system, realizing MTS speed-up factors of 20 or more, enabled by time steps of up to 25 fs for the slow forces.²⁷

In the context of multiscale approaches to biomolecule dynamics, the now common multipole expansion of the electrostatic field should also be mentioned.²⁸ There are also preliminary attempts at applying multigrid techniques to molecular problems.²⁹ Multigrid techniques have proven useful for solving partial differential equations in other contexts, such as the Navier–Stokes equations of fluid dynamics.³⁰

Eigenvector-Based Schemes

Normal modes, which are obtained from diagonalization of the Hessian matrix (second derivative of the potential), describe global deformations around a local minimum on the molecular potential energy surface. However, the harmonic approximation involved in the derivation of the modes limits their usefulness to small amplitudes, making these eigenvectors inappropriate for overall dynamics.

To overcome this problem, principal component analysis of MD trajectories, also known as “quasi-harmonic analysis” or “covariance modes,” was introduced to protein dynamics.^{31–33} In this method, a covariance matrix is constructed from a long

MD simulation, and diagonalized to obtain global modes that describe that motion. The method was later employed with the hope of describing molecular dynamics trajectories in terms of a small "essential subspace."^{34,35} Berendsen and coworkers later developed a technique that performs an adapted form of MD with constraint forces in the approximate "essential subspace."^{36,37} Because time information is lost in this protocol, the method is essentially a sampling technique that cannot directly be interpreted as dynamics. However, it does provide an enhanced sampling of the configuration space compared to usual MD.

Space et al.³⁸ used such subspaces in a dynamics method in which the low frequency motion is propagated via a projective Newton's equation. Applying this method to a Lennard-Jones crystal and a glass, they found convergence to the reference MD trajectory, although notable deviations also emerged. Speed ups of 2–5 relative to atomistic MD were obtained with time steps up to 25 fs, but in simple molecular models, significant energy damping was observed.

A recent study by Schulten and coworkers³⁹ suggests that the principal-component analysis method may not be suitable for describing long-time protein dynamics. The extremely long trajectories needed to extract stable principal component modes are currently impractical.

Implicit Schemes

The Langevin-implicit-Euler scheme was introduced into MD by Peskin and Schlick⁴⁰ to maintain numerical stability for large time steps. In implicit integration schemes, the incorporation of future information helps avoid stability problems associated with extrapolation techniques.⁴¹ Although the Langevin framework was introduced to replenish some of the energy that is damped by the implicit integrator, severe damping of high frequency modes still occurs, which in turn, has a significant effect on global motion.⁴² Therefore, this scheme is not appropriate for atomistic simulations. However, for macroscopic models, where high-frequency motions are irrelevant to the overall behavior, the Langevin-implicit-Euler allows for dynamical simulations of large-scale supercoiled DNA models as an elastic material.⁴³ A variant of this method, in which energy was put back into the system in an *ad hoc* fashion, was used to study possible folding pathways in BPTI.⁴⁴ However, this variant should be regarded as a sampling procedure rather than a dynamic integration scheme.

Implicit schemes, therefore, are not likely to be effective dynamical integrators at large time steps due to the damping effects. Implicit methods are also costly because of nonlinear minimization at each time step, and not likely to be competitive with other approaches in terms of simulation time requirements.⁹

Reaction Path Optimization

Recently, a different approach was suggested by Olender and Elber⁴⁵ to compute long-time molecular dynamics trajectories of fixed length in cases where both the initial and final states are known. The technique is an extension of a reaction path method,⁴⁶ and is based on the stochastic path integral of Onsager and Machlap.⁴⁷ Trajectories are computed by path optimization between the two end points, and modes of motion with periods shorter than the discrete time steps are filtered out making the trajectory stable for very long time steps. Although this approach does not solve the time-step problem for the typical molecular dynamics simulation, it is likely to be useful for specific processes where both initial and final states are known.

As seen from the above brief overview, extending the length of molecular dynamics simulations is an essential issue, leading to a broader application of this very useful computational technique. The available approaches offer only limited speedups of atomistic MD simulations by factors of 2 to 4 (SHAKE, MTS, Langevin–Normal mode), falling significantly short of what is desired. Other approaches have either developed into useful sampling procedures (torsion dynamics, rigid body dynamics, covariance modes) or deal efficiently with a restricted class of problems (reaction path optimization), but do not extend the scope of general purpose MD.

Despite the recent introduction of new approaches to bridge the gap between the time scales accessed by computer and those of more physical processes, the problem is far from being solved. The MBO(N)D flexible-body method presented herein represents a new approach to enabling long time-step MD simulations.

MBO(N)D-Substructured Multibody Molecular Dynamics

MBO(N)D's roots lie in applied mathematics techniques and algorithms developed in the 1980s for solving similar computational efficiency problems associated with dynamics simulations of large

complex mechanical structures (e.g., spacecraft) in the aerospace industry.⁴⁸ Proof-of-concept studies were presented in Turner et al.,⁴⁹ where various aspects of the MBO(N)D approach were discussed, and results for simple systems were reported.

MBO(N)D is a reduced-variables MD approach that seeks to dramatically improve computational efficiency over atomistic simulation methods, while maintaining comparable accuracy in the trajectory and ensemble properties at the structure–function level. The most important feature of MBO(N)D is that of substructuring, whereby groups of atoms in a molecular model are grouped into interconnected flexible and/or rigid bodies. These bodies are allowed to undergo large motions relative to each other. Within flexible bodies, the relative motions among the aggregated atoms are assumed small and at low frequency. For rigid bodies, it is assumed there is no motion between the aggregated atoms. This approach acknowledges that the essential dynamics of molecules are captured by the low-frequency modes.^{32, 34, 38, 50–52}

MBO(N)D's substructuring approach contains two essential elements for substantially improving the computational efficiency of molecular dynamics simulations. First, the number of degrees of freedom in the dynamics equations is drastically reduced, potentially from tens or hundreds of thousands of atoms to tens or hundreds of bodies. Second, both the inter- and intrabody dynamics occur at low frequencies, which allows the simulation time step to be increased from 1 or 2 fs to 10 fs or more. These two key attributes allow MBO(N)D dynamics simulations to execute at computational speeds that are significantly faster than traditional all-atom methods, depending on the nature of the substructuring scheme. For systems studied to date, speed ups of up to 30 have been obtained. As would be expected, higher levels of aggregation (e.g., moderate- to large-sized bodies) typically execute faster than lower levels of aggregation (e.g., small bodies and atomistic regions).

Substructuring schemes for MBO(N)D simulations are determined by the amount of relative motion expected throughout the molecule. For regions where motions are expected to be very small or small enough to be unimportant to the particular event of interest in the simulation, the atoms can be grouped together into rigid bodies. Regions where there are moderate amounts of motion can be modeled as flexible bodies. Regions where large conformational changes are expected are modeled with many small bodies or individual atoms. The development of appropriate substructuring schemes is an

important aspect of MBO(N)D, and is addressed in detail in the Substructuring Strategies section.

The internal motions of MBO(N)D's flexible bodies are modeled by a reduced set of body-based modes, which are added to the motion associated with the three translational and three rotational rigid body degrees of freedom. The eigensolution process needed to generate these flexible body modes is far more computationally efficient than in traditional normal mode analyses. This is because MBO(N)D's modal displacement vectors and frequencies are calculated separately for each individual body, rather than for the entire system. The resulting mode set for each flexible body is then reduced by selecting only the lowest frequency modes that correspond to the overall motions of the body. The high-frequency modes, which correspond to localized vibrations, are typically not important to the event of interest, and are ignored during subsequent simulation.

MBO(N)D's rigid bodies are considered as special cases of flexible bodies where none of the body-based modes are retained. Individual atoms (particles) are modeled classically, with three translational degrees of freedom, and are used in regions of the molecular model where no aggregation into larger bodies is appropriate or desirable.

Compared to individual atoms, rigid bodies have a larger mass, and hence, lower frequency content when subjected to the same level of forcefield interactions. When bodies are made flexible, the truncation of high frequency modes corresponding to localized motions maintains the frequency content of the body at a low level. Thus, substructured modeling allows the use of long time steps in MBO(N)D. Because arbitrarily large motions subject to interaction forces are permitted between bodies, MBO(N)D's substructured modeling approach captures anharmonic motions that other reduced order methods, such as normal modes, cannot properly reproduce.

Substructuring allows multigranular simulations in MBO(N)D where different regions of the molecule are modeled at varying levels of aggregation (flexible/rigid bodies and/or atoms), depending on their importance to the particular event being studied. Parts of the system, such as the region near an active site of a protein, are modeled at a fine level of granularity by using individual atoms and small bodies. Other parts far away from an active site are modeled at a coarse level of granularity by using large bodies.

MBO(N)D's multigranularity can be coupled with integrator implementations that make use of

the multiple time scales present in the physical system. Even for conventional all-atom models, one can take advantage of the separation of time scales that exist between fast interactions such as bond stretching and slow interactions such as long-range electrostatics. For MBO(N)D, a natural separation of time scales arises from differences in body size. The motions and interactions involving individually modeled atoms are expected to have high-frequency content, while bodies are expected to have lower frequency content. Consequently, the motion in regions that are modeled using particles can be integrated using shorter time steps, while the motion in regions that are modeled as bodies (rigid or flexible) can be integrated using longer time steps. As will become apparent, multiple time-scale (MTS) integration is the key to allowing MBO(N)D handle the higher frequency content of atomistic regions, while still achieving maximum computational efficiency from the body-dominated portion of the model. A similar multigranular approach was developed by Head-Gordon and Brooks,³ where small virtual rigid bodies were used in regions surrounding a central atomistic region to account for global motions that are not captured in stochastic boundary molecular dynamics methods.

Force field interactions for MBO(N)D are generally obtained by conventional all-atom calculations, such as the CHARMM all-atom force field.⁵³ One alternative to the conventional scheme, afforded by MBO(N)D's body-based formalism, is to replace interactions that are internal to a body with modal stiffness terms. This allows the bond and nonbond pairlists to be reduced in size by eliminating interaction calculations between atoms that reside in the same body, resulting in a more efficient computational process. In addition, fast multipole algorithms can be applied to speed up nonbond calculations for large molecules.^{28, 54}

MBO(N)D is currently integrated with CHARMM,⁵³ which provides the molecular definition, the force field interactions, and postsimulation analyses. The output from an MBO(N)D simulation is the same as that from all-atom simulations, with the addition of body-based information. The coordinates and velocities of every atom at every step in MBO(N)D trajectories can be determined from the translations, rotations, and modal amplitudes of the bodies. Thus, conventional postsimulation analysis algorithms, such as those in CHARMM, can be applied directly to the trajectories resulting from MBO(N)D simulations.

In the following subsections the elements of the MBO(N)D modeling approach are described, in-

cluding the substructuring and body-based mode methodology, use of constraints, handling of force field interactions, and a new integrator developed for MBO(N)D.

MULTIBODY DYNAMICS

In the MBO(N)D approach, a molecule is generally described as a cluster of multiple, flexible substructures (bodies) that comprise a dynamic system. Member bodies of the molecule are capable of undergoing large relative excursions, and are interconnected by fixed or free bond lengths. This type of multibody system is acted on by inertial forces such as those due to centripetal and Coriolis acceleration as well as by the usual forces derived from empirical potentials.

Figure 1 shows a topological configuration example of a typical multibody system. The five hinges and four bodies shown result in one closed path. For each body of the system, there is a body-fixed, right-handed reference frame located at a user-selected location within the body. A body's elastic deformation is measured relative to this reference frame.

A hinge is defined by locating one point on each of two bodies. Each pair of points, p_i and q_i , in Figure 1, constitutes a hinge. Clearly, a typical body may contain one or more hinge points, but a single hinge defines the relative motion of only two bodies.

An orthogonal reference frame is attached to point p , and another is attached to point q . (The

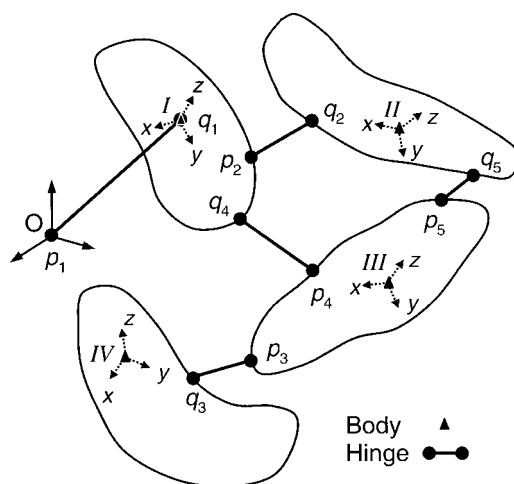


FIGURE 1. A multibody dynamics system substructured into several bodies connected by hinges. Shown are the body-fixed reference frames (dotted lines) and the inertial frame (solid lines). Roman numerals indicate body numbers. p_i and q_i indicate the two points associated with hinge i .

subscripts on p and q are implied.) Now six relative position/orientation coordinates are associated with the hinge defined by points p and q . These coordinates are used to define the relative position/orientation of the bodies “joined” by points p and q . Point q is located relative to point p by a p -frame referenced position vector. This vector may be expressed in Cartesian coordinates or spherical coordinates, depending on the constraints across the hinge. The orientation of the q -frame with respect to the p -frame is represented by three Euler rotations. If N_H is the number of system hinges, $6 \times N_H$ position/orientation coordinates are used in conjunction with modal displacement coordinates for defining the system’s position state. Only the time-variable position/orientation coordinates of the $6 \times N_H$ set of coordinates are included as state-vector elements. The position/orientation coordinates whose rates are constrained to be zero are not included; however, the position/orientation coordinates themselves need not be zero.

The base-body, indicated by *Body I*, and the base hinge, indicated by points p_1 – q_1 of Figure 1, are given special consideration. Point p_1 of the pair is coincident with the inertial origin. Although motions across other hinges represent relative motion between the associated bodies, motion across the base hinge defines the motion of the base body relative to the inertial frame. By traversing consecutive sequences of bodies and hinges, starting from the base body, and calculating positions and orientations along the way, one can obtain the inertial positions and orientations of all of the bodies and particles within the multibody system. Note that the selection of any one body as the base body is purely arbitrary. Atom coordinates and velocities can be calculated from the multibody variables (which are the displacement vectors, Euler angles, modal coordinates, and their time derivatives).

Interaction forces are applied to each of the atoms in each body. The effect of these interaction forces on the body coordinates is determined by: summing up the atomistic forces to obtain the body force vector; summing up the moments about the body reference to get the body torque vector; and multiplying the atomistic forces by modal displacement vectors to obtain the body’s modal force vector [see also the Generalized Force Vector section and eq. (10)].

Multibody Equations of Motion

In MBO(N)D, the rigid or flexible bodies are oriented with respect to one another through a series of vectors that span the hinge points between bodies,

and through a corresponding series of angular coordinates that define the relative orientation of the bodies. These relative hinge coordinate vectors, β , are of dimension 6×1 . For the case of unconstrained relative motion

$$\beta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad x \quad y \quad z]^T, \quad (1)$$

where θ_1 , θ_2 , and θ_3 are Euler angles for unconstrained hinges. For the case of bond-length constraints,

$$\beta = [\phi \quad \theta \quad r \quad x \quad y \quad z]^T, \quad (2)$$

where ϕ , θ , and r are spherical coordinates for bond-length constrained hinges. In both cases, x , y , and z are the components of the vector defined from the hinge’s p frame origin to the hinge’s q frame origin.

For rigid bodies, the positional state of the system is completely described by the above set of relative coordinates. For flexible bodies, an additional set of modal coordinates is used to describe the deformational state of each of the flexible bodies. The deformed position of the i th atom within a flexible body is given by

$$\{x_i\} = \sum_j \phi_{ij} \xi_j + O(\xi_j^2) + O(\xi_j^3) + \cdots, \quad (3)$$

where ϕ_{ij} is the partition of the j th mode shape vector corresponding to the x , y , and z coordinates of atom i , and ξ the j th modal coordinate for the flexible body.

In MBO(N)D and many structural dynamics formulations, the second and higher order terms in eq. (3) are ignored, based on the assumption that deformations are small. Equations of motion involving nonlinear terms in ξ are computationally more intensive, and are, therefore, avoided whenever possible. In MBO(N)D, large motions are allowed between bodies, thus providing a computationally efficient solution to the problem of limited motions inherent to the linear modal representation.

The hinge kinematics of the multibody system can be written as

$$\dot{\beta} = BU, \quad (4)$$

where β is vector of relative velocities across hinges, B is a matrix of kinematic coefficients that relate hinge relative velocities to body velocities, and U is a vector of nonholonomic body velocities consisting of angular velocities, translational velocities, and modal velocities and modal velocities,⁵⁵

$$U = [\omega_x \quad \omega_y \quad \omega_z \quad v_x \quad v_y \quad v_z \quad \dot{\xi}_1 \quad \dot{\xi}_2 \quad \cdots \quad \dot{\xi}_m]^T. \quad (5)$$

The matrix B in eq. (4) is defined in such a way as to simplify the specification of free and constrained degrees of freedom. For bond-length constraints, the appropriate partition of the B matrix is used to transform velocities from Cartesian frames into spherical frames. The radial degree of freedom, representing the bond direction, can be specified as the constrained degree of freedom.

An orthogonal set of selection matrices, Φ and $\overline{\Phi}$ (consisting of ones and zeroes), are created to extract the free and constrained relative degrees of freedom, respectively:

$$\dot{\beta} = \Phi \dot{\beta}^f + \overline{\Phi} \dot{\alpha}. \quad (6)$$

The variable β^f represents the unconstrained hinge degrees of freedom, and the variable α represents the constrained hinge degrees of freedom. For a frozen degree of freedom, which is the most common type of constraint, α is zero. Rheonomic constraints, where α is an explicit function of time, can also be specified within this formulation. Such constraints can be useful for forcing parts of the molecule to move in a predefined manner.

The multibody equations of motion that are used in MBO(N)D are derived from Lagrange's equations of motion,⁴⁸ with the generalized velocities transformed into the set of nonholonomic velocities defined by the U vector. The dynamics of the multibody system are governed by the equation

$$M\dot{U} = G + B^T \overline{\Phi} \lambda, \quad (7)$$

where M is a generalized inertia matrix, G is a vector of generalized forces, $\overline{\Phi}$ is the constraint selection matrix of eq. (6), and λ is a vector of generalized constraint forces. For rigid bodies, the corresponding partitions within U , M , G , and B contain only translational and rotational degrees of freedom; for particles, the corresponding partitions contain only translational degrees of freedom. The terms comprising the generalized force vector are described in the next section. The equations of motion are subject to the following constraint:

$$\dot{\alpha} = \overline{\Phi}^T \dot{\beta}. \quad (8)$$

The O(N) algorithm takes advantage of the sparsity of the matrices M and B ,⁵⁶ and is used to solve eqs. (7) and (8) in MBO(N)D. Body accelerations, U , are transformed into relative accelerations, β , in a recursive procedure that traverses the topology of the multibody system. Note that the computational requirements of the O(N) algorithm scale linearly with the number of bodies (N) in the system. Appendix A (supplementary material) contains a concise description of the algorithm.

Generalized Force Vector

The generalized force vector is composed of the following terms for each body:

$$G = G_{ff} + \Omega MU + \frac{1}{2} U M_j U - \dot{M} U. \quad (9)$$

The first term, G_{ff} , accounts for chemical interactions embedded in forcefield form. The Ω matrix in the second term consists of linear and angular velocities, and accounts for gyroscopic and coriolis effects. The last two terms account for changes in the body's inertia matrix, resulting from the deformation of the body. The subscript j represents the derivative with respect to the j th modal coordinate.

The dynamics of the MBO(N)D flexible multibody model are driven by forcefield interactions between bodies as well as interactions within flexible bodies. The equations of motion deal with body forces, body torques, and modal forces. Currently two alternative methods for calculating these aggregate forces are available within MBO(N)D. Note that interactions between atoms within rigid bodies are not computed, because they do not affect the dynamics. The two methods are described as: (a) Conventional atomistic forcefields—all forcefield evaluations are performed as in all-atom models, and the resulting force vector is processed to obtain the generalized forces, as follows:

$$G_{ff} = \begin{Bmatrix} \sum r_j \times f_j \\ \sum f_j \\ \sum \phi_j^T f_j \end{Bmatrix}. \quad (10)$$

Here, the implied summation over j includes all atoms within the body being considered: r_j is the vector from the body reference origin to atom j , f_j is the total force applied to atom j , and, ϕ_j represents the j th partition of the body-based mode vectors for the flexible body.

Note that any number of fast multipole approximations described in the literature^{28,54} can be applied to speed up the nonbond force field calculations. In this case, certain nonbond components of the forcefield interactions are evaluated by a conventional fast multipole method. The resulting atomistic force vector is processed as in eq. (10) to yield G_{ff} . (b) Modal stiffness calculations—for flexible bodies, interactions between atoms on the same body can be approximated by a linearization about the nominal body structure used for body-based mode generation. This method is equivalent to using the Hessian matrix projected into modal space, and multiplied by the modal coordinates to obtain the modal force. There are no effects on body translation or body rotation that result from

body internal interactions. To use the modal stiffness approach, the force vector is separated into two terms—the interbody interaction term, and the intrabody interaction term,

$$G_{ff} = G_{\text{inter}} + G_{\text{intra}}. \quad (11)$$

The interbody interaction term is computed as in eq. (10), but using a modified bond list and nonbond pair list that is restricted to the interactions involving at least one atom outside body j . For dihedral interactions, this term includes dihedrals where one or more atoms are within body j and at least one atom is outside body j . The same applies for bond angle terms. The intrabody interactions are approximated by a modal stiffness term

$$G_{\text{intra}} = K\xi = \phi^T H \phi \xi. \quad (12)$$

The matrix K is formed by transforming the appropriate Hessian matrix, H , into modal space via a matrix of modal displacement vectors, ϕ . The modal coordinate is represented by ξ . The Hessian matrix includes only those interactions where all atoms involved are within body j .

Bond-Length Constraints

Constraints for bond lengths and bond angles are important elements in MD simulations.¹⁰ Typically, these constraints are used in an attempt to eliminate the time constants that are attributable to bond stretching and angle-bending terms. In MBO(N)D, such constraints are handled by the O(N) recursion algorithm that eliminates the constrained degrees of freedom from the system model.

MBO(N)D's constrained hinge bond lengths are constraints that are applied across covalent bonds that exist between bodies and/or particles. For models involving flexible bodies, the body-based modal displacement vectors do not currently account for fixed bond lengths within the bodies. Thus, the specification of bond length constraints affects only those covalent bonds that are between bodies.

Whenever hinge bond lengths are constrained in an MBO(N)D substructured simulation, there exists the possibility of closed topological loops because MBO(N)D exactly enforces the constraints by removing the corresponding degrees of freedom. Examples involving closed loops include: aromatic side chains if modeled as atomistic regions; disulfide bonds if not completely enclosed within one body; and, in some instances, bodies defined by nonconsecutive groups of atoms, such as in across-strand beta-sheet bodies. MBO(N)D has no inherent difficulty in handling closed topological loops.

However, as the number of closed loops increases, the computational time and required memory increases. In such cases, it is prudent to turn off the hinge bond length constraints

Degrees of Freedom and Temperature

The MBO(N)D substructured model typically involves fewer degrees of freedom than the corresponding atomistic model. The number of degrees of freedom of the system is given by

$$n_{\text{DOF}} = \sum_{i=1}^{NB} (6 + m_i) + 3 \times n_{\text{atom}} - n_{\text{constraints}}. \quad (13)$$

Each multiatom body (i) has six rigid degrees of freedom and m_i modal degrees of freedom. Each individually modeled atom has three degrees of freedom. The total number of degrees of freedom is reduced by the presence of bond-length constraints (described in the previous section).

The calculation of system temperature in MBO(N)D takes into account the reduced number of degrees of freedom in the system. The formula used for the temperature calculation is

$$T = \frac{2K}{k_B n_{\text{DOF}}}, \quad (14)$$

where T is temperature, K is the total kinetic energy of the system, and k_B is the Boltzmann constant.

For comparison between MBO(N)D and atomistic models, both sets of simulations are run at the same temperature. The MBO(N)D model exhibits lower kinetic energy than the atomistic model, but this energy is distributed among correspondingly fewer (but important) degrees of freedom.

Initial Coordinates and Velocities

The initial conditions for MBO(N)D position and velocity variables are obtained by a least-squares fit to the atomistic coordinates and velocities. Consequently, MBO(N)D can be started from the same set of coordinates and velocities from which an atomistic MD run starts. The final coordinates and velocities of an MBO(N)D run can also be read in by an all-atom code to start an atomistic simulation. The least-squares fitting procedure is used only once during initialization.

For the positional variables, the least-squares fit is performed to solve for the following quantities: the displacement vectors from the inertial frame origin to the body frame origin; the rotational transformation matrices that orient the body frames with respect to the inertial frame; and the modal amplitudes of each body. Hinge bond-length constraints

are also imposed for covalent bonds between bodies, if specified for the simulation. An iterative Newton–Raphson⁵³ procedure is followed, where the second-derivative matrix is a function of the positional variables to be solved.

For the velocity least-squares fitting, the velocity vectors, angular velocity vectors, and modal velocities are solved for in a one-time calculation. The fitting problem is linear and, therefore, requires no iteration. The fitting also includes derivatives of the bond-length constraints, if these were used for the position fitting. Furthermore, six additional constraints are applied to the velocities, such that the MBO(N)D model's linear and angular momenta match those of the input atomistic conditions. These can be set to zero.

After the positional and velocity variables have been computed, these are easily converted into relative and modal coordinates, and relative and modal velocities to initialize the MBO(N)D dynamics integration.

MBO(N)D INTEGRATION ALGORITHMS

Numerical integration of the MBO(N)D equations of motion is performed by a specially formulated algorithm. The Verlet-type integrators that are commonly used for molecular dynamics are not directly applicable to the MBO(N)D equations, because such integrators assume that the acceleration variables are not functions of velocity. The acceleration expression in the MBO(N)D equations of motion depends nonlinearly on the velocity variables as shown in eq. (9). These velocity-dependent terms arise from gyroscopic and Coriolis effects, kinematic constraints, and deformation-dependent inertia terms.

Integrators commonly used for multibody dynamics, such as Runge–Kutta and predictor–corrector methods, are not computationally efficient for use in MBO(N)D because of the large number of forcefield evaluations involved. Additionally, the energy conservation characteristics of these integrators over a large number of integration steps are poor. An iterated velocity Verlet, which is sometimes used in atomistic MD for treating velocity dependency,²² also results in poor conservation of energy when utilized in MBO(N)D.

The integration algorithm developed for MBO(N)D to handle the velocity-dependent terms and afford high computational efficiency is similar to the velocity Verlet algorithm. This new integrator, hereinafter called the Lobatto integrator for brevity, is based on the Lobatto III a-b partitioned

Runge–Kutta integrator.⁵⁷ The essential difference in this new integrator lies in the inclusion of velocity-dependent terms that are always evaluated at the half step. The positional dependencies are evaluated at the beginning and end points of the integration interval. The algorithm proceeds as follows. The velocity variables are propagated to the half step,

$$v\left(t + \frac{\Delta t}{2}\right) = v(t) + a\left[x(t), v\left(t + \frac{\Delta t}{2}\right)\right] \frac{\Delta t}{2}, \quad (15)$$

where $v(t)$ is the velocity state, $x(t)$ is the position state, $a[\cdot]$ is the acceleration, and Δt is the integration step size. Note that the acceleration term is evaluated with the position variables at the beginning of the interval, while the velocity variables are evaluated at the half point of the interval. Because the velocity at the half step is present both in the left-hand and right-hand parts of eq. (15), the solution needs to be solved by iteration. The initial iterate is calculated as follows:

$$v^{(0)}\left(t + \frac{\Delta t}{2}\right) = v(t) + a[x(t), v(t)] \frac{\Delta t}{2}. \quad (16)$$

In MBO(N)D simulations, it has been found that one iteration is sufficient for convergence.

The position variables are propagated to the full step:

$$x(t + \Delta t) = x(t) + v\left(t + \frac{\Delta t}{2}\right) \Delta t. \quad (17)$$

The full-step positions are then used to evaluate the full-step accelerations. The full-step velocities are then obtained using the following equation:

$$v(t + \Delta t) = v\left(t + \frac{\Delta t}{2}\right) + a\left[x(t + \Delta t), v\left(t + \frac{\Delta t}{2}\right)\right] \frac{\Delta t}{2}. \quad (18)$$

Note that, as in eq. (15), the position and velocity variables are evaluated at different time points for calculation of the acceleration term. In this case, the position variables are at the full step, while the velocity variables are at the half step. Thus, the same set of velocities is used to compute the acceleration terms in both eqs. (15) and (18). Because of this, the acceleration evaluation at the end of the integration interval is different from that at the beginning of the next integration interval. This is due to the velocity dependencies of the acceleration terms being at the middle of the respective intervals. Nevertheless, the number of forcefield evaluations is the same as that required for Verlet integrators, which is once per integration step. This is because contribution

to the generalized force vector due to force field evaluations, G_{ff} , which are dependent only upon positions, can be evaluated at the end of the current step and saved for reuse at the beginning of the next step. Similarly, the iteration required for solution of eq. (15) is not very time consuming because the iteration is only over the velocity-dependent terms, which are less computationally intensive to calculate than the position-dependent terms.

One can show by Taylor expansion of the acceleration term that the above integrator is accurate to second order. Even though the Lobatto integrator performs exceedingly well compared to many other integrators, there is still a small amount of drift in linear and angular momentum, which needs to be removed from time to time. This is not surprising, given the complexities arising from quadratic dependency on velocity, with position-dependent coefficients, for the MBO(N)D accelerations.

Figure 2 is a comparison of three integrators: fourth order Runge-Kutta, Verlet central difference,⁵⁸ and the Lobatto integrator for an MBO(N)D simulation of ubiquitin using a 1-fs time step. The 1231-atom ubiquitin structure (1UBQ from the Protein DataBank) consists of two alpha helices and five beta strands. To prepare for the MBO(N)D simulations, the structure was minimized, heated, and equilibrated atomistically. This procedure was then followed by MBO(N)D equilibration and production simulations. In the MBO(N)D model, all unstructured regions were treated atomistically (i.e., atoms as particles), each alpha helix was treated as a flexible body, and each beta strand was treated as a flexible body. Vacuum modes with frequencies less than 80 cm^{-1} were selected for each flexible body. MBO(N)D simulations were performed using the

three integration methods, under constant energy conditions. Conservation of energy was used as the criterion for judging the accuracy of the integrators. Note that in the example presented, the Runge-Kutta integrator used four times as many force field evaluations as the Verlet central difference and Lobatto integrators. The figure shows that the total energy is conserved extremely well for the simulation performed by the Lobatto algorithm compared to the other two. The CPU times reported in Figure 2 are for runs performed on an SGI/R4000 computer. The results show that the Lobatto integrator is the most efficient, and accurate of the three tested.

A multiple time-scale version of the Lobatto integrator has been developed to handle multigranular models. Details are shown in supplementary material.

MODELING BODY FLEXIBILITY

Whole-Molecule Modes vs. Body-Based Modes

Body-based or component modes refer to a set of modes used to describe a portion, in this application a flexible body, of the modeled system. For MD applications, the component modes form a basis set for the group of atoms that comprise a body. Component modes are used in the engineering community^{59–61} as an intermediate step in the solution for system eigenvectors. Recently, such methods have been adapted to the problem of solving for the whole-molecule normal modes of large macromolecules.^{62,63} Similar methods for efficient calculation of normal modes have been developed by Durand,⁶⁴ and Mouwad and Perahia.⁶⁵ The methods are similar, with each solving a number of small diagonalization problems, and coupling these solutions to obtain the low-frequency modes of the entire system.

Component modes have been used directly to describe deformational motions of flexible components of articulated mechanical systems, such as mechanical robots with flexible arms, where there are large motions between the component parts.⁶⁶ Although the body-based modes account for small motions within the body, translational and rotational degrees of freedom allow for large motions between bodies.

Mode generation involves several issues: the method of mode generation, the coordinates from which to start the generation process, and the selection of modes to retain. In MBO(N)D, two types of body-based modes have been implemented: vacuum modes and fixed environment modes. These

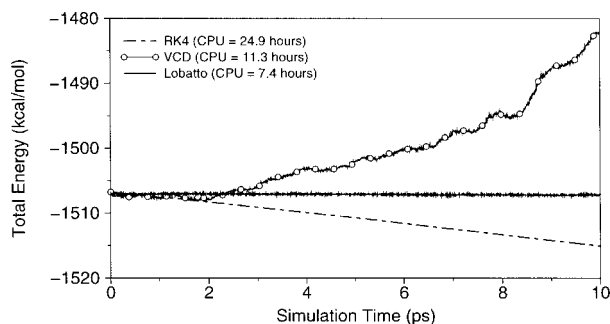


FIGURE 2. Total energy for 10 ps MBO(N)D simulations of ubiquitin using 1-fs time steps for three different integrators: fourth order Runge-Kutta (RK4), Verlet Central Difference (VCD), and Lobatto. The CPU times are reported for runs done on an SGI/R4000 computer.

methods reflect the use of different boundary conditions for the molecular component.

Vacuum and Fixed Environment Modes in MBO(N)D

Calculation of the Hessian matrix for each body in the absence of the rest of the system results in a set of modes referred to as vacuum modes. This set of modes is simple to calculate, as knowledge of other parts of the molecule is not required. However, these modes may be less useful when the body's deformational motions are strongly affected by its surroundings, such as in buried regions.

The second method of generation uses a Hessian matrix that is calculated assuming the rest of the system is fixed in inertial space. This set of modes is referred to as fixed environment modes. This approach accounts for some of the effects of the interaction with the rest of the system; however, the fixed environment modes may be overly restrictive on the low-frequency motions of the body.

The calculation of modes is straightforward for both mode generation methods. The usual Hessian matrix for the entire system can be divided into partitions that represent the bodies of the system:

$$H = \begin{bmatrix} H_{AA} & H_{AB} & H_{AC} & \dots \\ & H_{BB} & H_{BC} & \dots \\ & & H_{CC} & \dots \\ & & & \ddots \end{bmatrix}. \quad (19)$$

To calculate the fixed environment modes for body A, diagonalize the partition H_{AA} after mass weighting it. To obtain the vacuum modes for body A, modify the assembly process to exclude the components that arise from interactions between atoms in body A and atoms outside of it. Diagonalization of the resulting mass-weighted Hessian partition then yields the vacuum modes.

Within any MBO(N)D model, it is possible to use a set of vacuum modes for one body and a set of fixed environment modes for another body, as long as the same type of modes is used within any given body. MBO(N)D simulations using vacuum and fixed environment modes on a variety of molecular systems has not resulted in a clear preference. In general, we have favored the use of vacuum modes because they are more efficient to compute.

Choice of Reference Coordinates

The proper choice of reference coordinates for body-based mode generation is very important for obtaining good simulation results. Normal modes

are computed for the molecule using its minimum-energy state as the reference coordinates for calculation of the system Hessian matrix. The resulting modes are, thus, valid in the narrow harmonic region around this minimum energy state. Often, the minimum-energy structure is in a different conformation than those that are at room temperature. To generate a set of body-based mode vectors that are valid at the temperature of interest, the following procedure was adopted for generating the reference structure. A structure that has been equilibrated at room temperature or other desired simulation condition is subjected to a small number of minimization steps using steepest descents to yield the reference structure. Note that this is only a partial minimization, and serves to relieve instantaneous bad contacts while keeping the conformation close to the desired initial state. Alternative methods of obtaining an initial coordinate set exist such as minimization using a gradient criteria. However, it is unclear which method is most general.

Any modal solution using nonminimum-energy coordinates results in modal displacement vectors that correspond to imaginary or unstable frequencies. These unstable modes are important for describing barrier crossings and transitional motions within the molecule or flexible body.⁶⁷ For MBO(N)D, the purpose of the body-based modes is to provide a set of basis vectors for describing the elastic motions of the body. The imaginary frequencies do not present a problem for MBO(N)D dynamics calculations, because the frequency values themselves are not used in propagating the dynamics of the system. However, these frequency values may influence the selection of the modes to be retained for the simulation (see the next section).

For example, identifying the transition states associated with unstable modes may influence mode selection. Selection of unstable modes associated with desired barrier crossings⁶⁸ would allow for this type of motion in the subsequent simulation.

The procedure of applying a short amount of minimization to obtain the reference structure has the effect of reducing the number of unstable modes that would have resulted if no minimization had been applied at all. Including the remaining unstable modes allows more anharmonic motion during the simulation.

Selection of Body-Based Modes

Currently, there are two alternatives for the selection of body-based modes for use in MBO(N)D dynamics. The first approach is to use the frequency

associated with each mode as a means for selecting or discarding modes. For stable modes, one can select a cutoff frequency such that only modes lower than this frequency are retained. For unstable modes, animation of the modal displacement vectors has shown that unstable modes with large imaginary values correspond to highly localized motions, such as bond stretching and angle bending, and involve only two or three atoms of the body. As the magnitude of the imaginary frequency decreases, there is less localized behavior and more global behavior in the unstable modal displacement vectors. Thus, a selection criterion for unstable modes is to select a cutoff frequency and animate the modes in the vicinity of this cutoff frequency to observe what kinds of motions are described by those modes.

The second mode-selection approach utilizes some form of analysis on the modal displacement vectors to evaluate and rank the modes. A delocalization factor criterion⁷⁰ has been implemented to help determine whether a particular mode should be retained. This factor is used to distinguish modes that have localized behavior, such as local bond stretching and angle bending, from those that have a global behavior, such as helical torsion and bending. The delocalization factor is defined as

$$\text{delocalization factor} = \frac{\sum_i \Phi_i^4}{(\sum_i \Phi_i^2)^2}, \quad (20)$$

where the Φ_i are the elements of the modal displacement vector of interest.

In general, the smaller the value of the delocalization factor, the more global the character of the mode. A minor problem with this criterion is that some types of concerted local behavior, such as localized C—H stretching all over the molecule, may yield a small delocalization factor. Because these modes tend to be of high frequency, it seems that the best approach is first to apply a frequency cutoff, and then to sort the low- to medium-frequency modes by using the delocalization factor.

SUBSTRUCTURING STRATEGIES

The essential idea behind MBO(N)D's substructured modeling methodology is that large relative motions are allowed between bodies, while relative motions within each body are assumed to be small for flexible bodies, or negligible for rigid bodies. The goal of the substructuring procedure, therefore, is to identify groupings of atoms that can be treated as bodies, and to identify the bonds, hinges, or general areas where large motions take place.

For proteins, information on the system's motion can be obtained from several sources: (1) If the Protein Data Bank (PDB) has 3D coordinates for several conformations of a given molecule, analysis of the multiple set of conformations provides information for grouping atoms into bodies or identifying hinges. (2) Short atomistic MD or Monte Carlo trajectories can be analyzed to suggest the dynamical characteristics of the local conformational space. (3) Dynamical information from NMR spectroscopy and crystallographic B-factors (temperature factors) can give a rough indication of which parts of the system have larger motions. (4) Knowledge of particular types of protein dynamics can be applied. For example, alpha helices tend to have small amounts of motion, followed by beta-sheets, and unstructured regions. Prolines tend to introduce kinks into alpha helices, glycines introduce disordered floppy points, and peptide groups tend to stay planar. This type of information can be used to specify the substructuring scheme for molecular systems where other information on structural flexibility is not available.

A number of MBO(N)D substructuring strategies have been explored for proteins, and the following sections describe how these strategies work. The basic MBO(N)D modeling framework allows for the development of many different types of substructuring strategies. The following descriptions can be regarded as examples from which a user can develop a specific strategy for the particular modeling problem at hand.

Molecular Domains

Some protein motions can be characterized as motions between domains. To substructure such a molecule, first identify the two or more domains and the linker regions between domains. The linker regions will need to be more finely substructured than the domain regions. The typically correlated nature of the side chain motions of residues near the interdomain/linker region requires finer substructuring in this region. (See the Side Chain section.) The portions of the domains away from the linker region can be substructured into much larger bodies. The disparity in body sizes and the associated disparity in time scales for the dynamic simulation can be handled best by use of the MTS integrator.

Secondary Structure Elements

It is natural to expect that structurally well-defined regions such as alpha-helices and beta-sheets, whose atoms exhibit concerted motions,

would be good candidates for grouping into rigid or flexible bodies. However, a grouping of the constituent residues of such secondary structures into bodies does not necessarily yield the desired results. This is because statically defined secondary structures may actually exhibit considerable motion at the ends of the helical regions that is not well correlated with motions of the interior residues of that helical section. Interactions between beta strands may also be weak at certain locations within a protein.

One needs a way of evaluating the flexibility of the secondary structure elements to identify the core residues that maintain their stiffness through the hydrogen-bonding structure during molecular dynamics. These subsets are excellent candidates for substructuring into rigid or flexible bodies. The choice between rigid and flexible representation depends on both the amount of internal motion expected and the importance of that motion to the dynamic property being studied.

Alpha Helices. For alpha helices, a good way to identify these stiff secondary-structure elements is to carry out pseudodihedral angle analysis with a preliminary short atomistic simulation. Pseudodihedral angles are formed from the alpha carbon atoms from each set of four consecutive amino acid residues. The analysis involves the calculation of the difference between the largest and smallest values (the maximum difference) of each pseudodihedral angle during the atomistic simulation. Then, the model can be separated into bodies based on a cutoff value on the range of pseudodihedral angle motion.

Pseudodihedral angle analysis can actually be used to identify any region, having small enough relative motions that its component residues can be aggregated into bodies. However, this analysis works best for identifying alpha helical regions whose residues can be grouped into bodies. Our experience had shown that a maximum pseudodihedral angle difference threshold of about 45° works well for differentiating residues with low relative motions from highly mobile regions, such as loops. An example of this analysis is shown in supplementary material.

Beta Sheets. Application of pseudodihedral angle analysis to beta strands or bridges along the protein backbone usually indicates that there is too much motion in a beta-sheet to allow for modeling as a single body. Analysis of the motion of the beta sheet in the cross-strand direction, perpendicular to

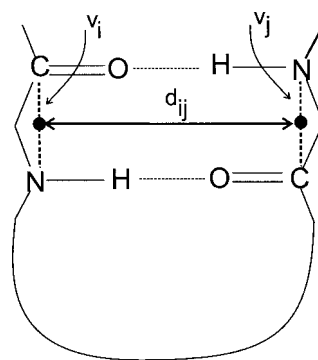


FIGURE 3. Schematic of the vectors used for the beta bridge analysis.

the backbone and along the hydrogen-bonding network, enables the identification of nonconsecutive residues that exhibit low relative motions. These residues can be grouped into rigid or flexible bodies.

The following beta bridge analysis can provide useful information for grouping residues across beta strands into bodies. For each residue, we define a vector within a residue between the main-chain carbon and the amide nitrogen as shown in Figure 3. The motion between each pair of residues can be grossly characterized by the translational and angular motion between these vectors. Considering only pairs of residues that are geometrically close to each other, we can sort the residue pairs by the amount of angular motion between them. Those with small relative angular motions are candidates for grouping into beta bridge bodies. See supplementary material for an example analysis.

In addition to the secondary structure analysis described above, programs such as DSSP can be used to analyze MD trajectories to determine the flexibility of secondary structure elements.⁷¹

Loops and Turns. By their nature, loop and turn regions exhibit highly anharmonic motions due mostly to dihedral motions between residues. When necessary, these regions can be modeled atomistically, but at the expense of limiting the size of the time step that can be used in the MBO(N)D simulation. The loop and turn regions (as well as any other region) should be substructured into the largest possible bodies that still allow the desired essential function of these regions to be expressed.

Several loop-substructuring strategies are useful in achieving this balance: divide the loop region into a few bodies; group two to three consecutive residues into bodies, being careful not to separate the peptide plane into separate bodies; or treat the

peptide planes as bodies and the side chains as bodies, starting at the alpha or beta carbon.

Side Chains. Side chains can be substructured independently of the protein backbone, as several rigid or flexible bodies, or they can be treated atomistically. A χ -angle analysis over a short atomistic run is often effective in helping to decide on a substructuring scheme for side chains. However, for longer side chains, crankshaft motion with little change in excluded volume is possible. Side-chain motion can be more effectively determined by considering the side-chain angle formed by the vector from the N to C atoms of the residue, and the vector from the C_α to an atom at the side-chain terminus. Figure 4 is a schematic diagram showing the vectors that are used to define the side-chain angle. If the analysis of multiple structures, or of frames from an atomistic trajectory, shows that the overall relative motion between the side chain and main chain (as measured by our side-chain angle) of a given residue is large, a candidate substructuring is to cut at the ϕ and ψ angles, and even the χ_1 angle. Additional analyses, such as calculating the dihedral fluctuations for ϕ , ψ and χ_1 angles can be performed on the candidate residues. A large dihedral fluctuation will indicate that a particular torsion angle should not be constrained. See supplementary material for an example plot of side chain angle.

For globular proteins, experience has shown that it is important to allow χ_1 and other torsional side chain motions in buried regions of the protein for certain essential dynamic characteristics to be reproduced. Note that allowing side-chain motion within buried regions does not necessarily require the use of smaller time steps. The atomic fluctuations depend on details of the local packing environment, which may serve to over damp local oscillations.^{72, 73}

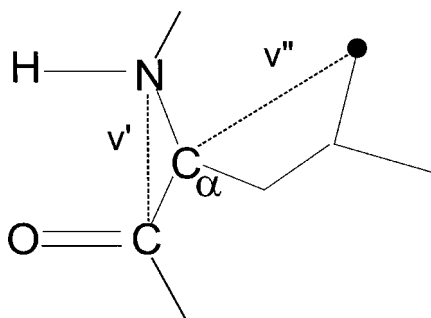


FIGURE 4. Schematic diagram showing the vectors used for defining overall angular motion between side-chain and main-chain groups.

MBO(N)D Test Cases

This section presents the results from a number of simulation analyses on various molecular systems that have been conducted by us to test and evaluate MBO(N)D. We selected these systems to characterize the dynamics of MBO(N)D and validate the use of substructured modeling in a variety of systems that differ substantially in size and range of motion, and to compare the results to corresponding atomistic methods. Due to the larger number of DOF, atomistic systems require significantly more time to stabilize energetically than body-based systems; our test systems are, therefore, relatively small to permit an easier task for meaningful comparisons. Consequently, the systems studied herein do not afford optimum conditions for demonstrating MBO(N)D's computational speed capability, which is favored by much larger molecules with larger bodies than those we will discuss. Our analyses and results are, nevertheless, an important first step in assessing MBO(N)D's capability to yield acceptable agreement with atomistic simulations in terms of the essential dynamics. Once MBO(N)D's capabilities and behavior are well understood at this level, our methodology can be extended to much larger systems where it will have the opportunity to facilitate significantly higher computational speedups compared to atomistic simulations.

All MBO(N)D simulations were performed using the MBO(N)D code. The standard atomistic simulations, to which we compare our results, were performed with the CHARMM molecular modeling program.⁵³ We explored a variety of different substructuring strategies to determine the effects on the dynamics of each of the following five systems. (1) Alanine dipeptide—a relatively small and simple system that has a well-defined and frequent transition between two distinct conformers on the subnanosecond time scale. (2) The terminal fragment of the L7/L12 ribosomal protein from *E. coli*—a globular protein system containing loops, helices, and beta strands that exhibits key motion between two of the three helices. A wide range of substructuring strategies were applied to 1CTF. (3) Dickerson dodecamer—this system is a DNA duplex initially in the B form. The goal is to characterize the ability of MBO(N)D to handle large conformational changes induced by an external shear force in nucleic acids. (4) HIV-1 protease–ligand complex—a protein–ligand complex that represents a model system encountered in rational drug design. We applied a unique pulling protocol to examine ligand–

protein dynamics with MBO(N)D. (5) Liquid crystal polymer—an alternate system to those based on biological molecules. This system is used in optics and represents a problem in polymer dynamics. (See supplementary material.)

ALANINE DIPEPTIDE

Alanine dipeptide ($\text{CH}_3\text{CONHCH}(\text{CH}_3)\text{CONHCH}_3$) is a good model system with which to understand the basic behavior of the MBO(N)D method and to simplify direct comparison with atomistic dynamic simulations. This dipeptide is structurally simple, which significantly reduces the complexity of the analysis, and it exhibits well-defined low frequency anharmonic motion. This motion was analyzed previously using standard atomistic techniques,^{75,76} and is characterized by a transition between two distinct conformations at 300 K. One conformer is referred to as C5, and is proximate to $\phi = -180$ and $\psi = 180$. The other conformer, C7eq, is proximate to $\phi = -60$ and $\psi = 60$. The barrier between these two conformers is on the order of 1 kcal/mol on the CHARMM potential energy surface. Simulation conditions are detailed in supplementary material.

Alanine Dipeptide Results

The following two substructuring schemes were used with MBO(N)D. First, a single, flexible-body for the entire system using all of the body-based modes. The purpose of this run was to characterize MBO(N)D’s ability to capture anharmonic motions through the projection of atomistically calculated force field interactions onto the modal degrees of freedom. Although the transformation of the direction of motion from modal to atomistic is linear, the atomistic forces that are projected onto these DOF are nonlinear. The interplay between the linear and nonlinear effects can be assessed with the single flexible-body example. Second, two rigid bodies for each peptide group (—CONH—) and the remaining atoms treated as particles (single atom bodies). Because peptide groups are quite rigid, we expect that treating such groups as rigid bodies should be consistent with atomistic models.

Table I shows a summary of the relative populations between the two dominant conformations of alanine dipeptide at 300 K (C5 and C7eq) obtained from three different simulations: an atomistic MD simulation, and the two different MBO(N)D simulations. As seen in the rigid body, MBO(N)D simulation shows very good agreement with the

TABLE I. Relative Population of the C5 and C7eq Conformers and Transition Rates of Alanine Dipeptide for Various Simulations (1 ns).

Model	(C5/C7eq)*100	Transition rate (ps ⁻¹)
Atomistic	23.7	0.38
MBO(N)D		
Rigid peptide plane bodies	21.1	0.37
One flex body—all modes	14.1	0.12

atomistic simulation results. The agreement is obtained both in terms of the relative occupation of the two minima and in terms of transition rates, reflecting on MBO(N)D’s success in reproducing aspects of the system’s thermodynamics and kinetics. The flexible body result, however, is more restricted in barrier crossings reflecting the fact that the modes were calculated near the C7eq conformer. In other words, these modes bias the region of the initial coordinates.

Figure 5a and b shows that the distribution of the phi and psi angles for the rigid peptide bodies match the atomistic results well. The distribution from the single flexible body with all modal DOF (c), however, favors the C7eq conformer about which the modes were calculated. These results suggest that the rigid body substructuring used is better suited for this type of anharmonic behavior, while single flexible bodies are good for more local behavior. We note, however, that significant anharmonic motion occurs even when all the DOF are due to modes. We believe this result from the single flexible-body simulation to be significant because it suggests that the linear approximations made with modes can be extended to include anharmonic motion by projecting nonlinear forces onto these DOF. Additional analysis results such as time correlation functions are shown in supplementary material.

C-TERMINAL FRAGMENT OF RIBOSOMAL PROTEIN

We wished to explore a more extensive set of substructuring strategies on a model protein to further characterize MBO(N)D on a typical problem in biophysics: interdomain motion. We, therefore, chose the C-terminal fragment (1CTF) of the L7/L12 ribosomal protein from *E. coli*,⁷⁹ as this model. 1CTF

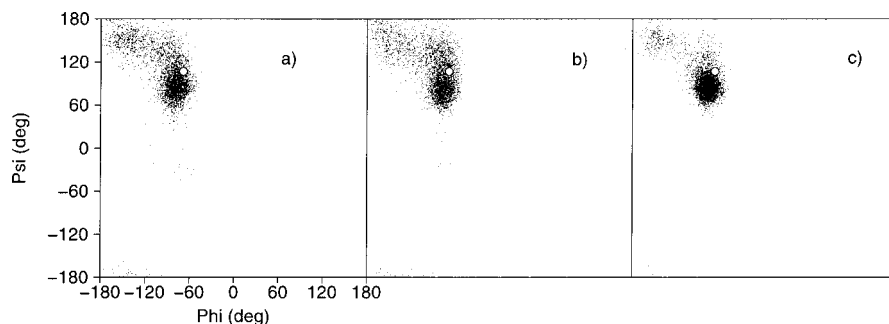


FIGURE 5. ϕ - ψ scatter plots for alanine dipeptide at 300 K for (a) atomistic, (b) MBO(N)D simulations with rigid peptide plane bodies, and (c) single flexible body with all modal DOF. The values from initial coordinates are represented by the white circle.

is a medium-size protein that contains 65 residues, and is essential for efficient polypeptide synthesis in bacteria. 1CTF contains three α -helices and three β -sheets that account for approximately 76% of its structure.

Simulations by Åqvist et al.,^{80,81} using the GRO-MOS force field, found that an important feature of the dynamics of 1CTF is the low frequency motion, around 5 cm^{-1} , of the B helix relative to the rest of the structure. This motion was clearly displayed when helix C is used as a reference for the relative movement. Helix C is tightly bound to the beta sheet and, therefore, has restrained motion. Thus, the motion between helices B and C is considered the essential dynamic behavior of 1CTF. The purpose of this test case was to assess the speedups attainable from various MBO(N)D substructuring strategies, as well as the level of agreement in the essential dynamics with atomistic simulations.

1CTF Results

We explored different values of integration time steps to determine the relationship between each different substructuring scheme and stability of the simulation. The conservation of total energy was monitored so as to evaluate the stability of each simulation. We will only discuss results of simulations exhibiting the highest time step that resulted in stable simulations (defined as $\leq 1\text{ kcal/mol}$ RMS fluctuation of the total energy).

The following substructuring schemes were used for the MBO(N)D simulations. For each scheme, the total number of DOF is listed, together with the maximum time step used. The atomistic model contained 1185 DOF.

Case 1. Loops and β -strands were substructured into several small rigid bodies with hinges defined

at selected ψ dihedrals. Each helix was substructured as a single rigid body. Total of 20 bodies. This scheme is shown in Figure 6. Total number of DOF = 95. Maximum time step = 20 fs.

Case 2. Same substructuring as in case 1, with body-based modes added to each of the 20 bodies. The modes in each body were sorted by delocalization factor [eq. (20)] and the lowest 10 modes were added to each helix body. Similarly, the lowest three modes were added to each of the small bodies. Fifty-eight modes were added. Total number of DOF = 153. Maximum time step = 10 fs.

Case 3. The entire system was substructured into 31 small rigid bodies with hinges at ϕ or ψ an-

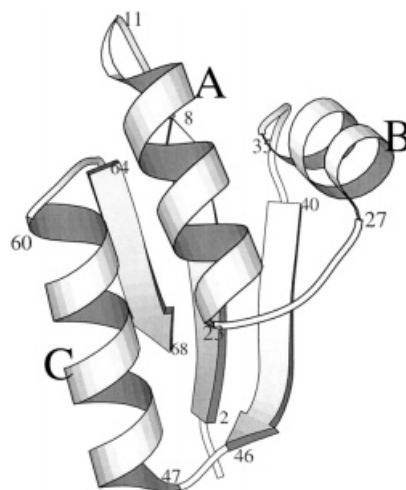


FIGURE 6. Structure of the C-terminal fragment of the ribosomal protein from *E. coli* (1CTF⁷⁹). The three helices are marked as A, B, and C. Substructured according to case 1; the residue ID numbers between bodies are shown.

gles. Total number of DOF = 150. Maximum time step = 15 fs.

Case 4. Based on case 1 substructuring, with each helix further separated into a flexible main-chain body and several side-chain bodies (except for Ala and Gly residues), resulting in 66 bodies. Body-based modes with a natural frequency less than 100 cm⁻¹ were added to bodies with more than 10 atoms. One hundred seven modes were added. Total number of DOF = 432. Maximum time step = 5 fs.

Case 5. Peptide planes and side chains modeled as bodies, and C_α atoms as particles. The five lowest frequency modes were added to each side-chain body. Total of 150 bodies and 180 modes. Total number of DOF = 966. Maximum time step = 1 fs.

Case 6. Domain-based substructuring. Two small domains in 1CTF were separated into subdomains including: αα domain (which includes helices A and B), and the B-sheet domain (which includes the β-sheet and helix C). Those domains were substructured into bodies, which had several residues each. Then, linker residues, connecting between the two domains, were substructured with one-residue bodies (a total of 20 bodies). Total number of DOF = 95. Maximum time step = 20 fs.

We performed additional simulations using the MTS integrator for cases 1, 2, 3, and 6 to evaluate the performance of this methodology with different

substructuring. MTS was not utilized with cases 4 and 5, as the potential for simulation speedup was in the range of that attainable by atomistic techniques. Integration time steps used in these MTS runs are listed in Table II. In these substructuring strategies, case 1 is the coarsest substructure scheme, and case 5 is the finest.

A summary of results from all MBO(N)D and atomistic simulations performed for 1CTF are listed in Table II. Each MBO(N)D simulation shows good total energy conservation during the 140 ps production run; the RMS fluctuation of total energy for each MBO(N)D simulation is well below our criterion for a stable run (<1 kcal/mol). shows the following three general trends. First, smaller time steps result in increased stability. Second, increasing the number of DOF lowers the time step needed for stable simulation. It is possible that two different substructuring strategies can result in different stabilities and speedups, even though they have the same number of DOF. In other words, the number of DOF used is not necessarily a good predictor of MBO(N)D performance; proper substructuring representing the motions of interest is important also. Third, MTS results in energy stabilities comparable to the single time step results, but with increased speed (as we will show).

Figure 7 shows the speedup vs. the ratio of the MBO(N)D to atomistic B-C angle RMS fluctuation. Representative probability density functions of the B-C helical angle are presented in Figure 8. As expected, both figures show that finer MBO(N)D

TABLE II. Comparison of Atomistic and MBO(N)D Simulations of 1CTF.

	Number of bodies	Modes	DOF ^a	Δt ^b (fs)	E _{rms} ^c (kcal/mol)
Atomistic	—	—	1782	1	0.06
Case 1	20	0	95	20 20, 40, 60, 80	0.37 0.35
Case 2	20	58	153	10 10, 20, 30, 40	0.15 0.21
Case 3	31	0	150	15 15, 30, 45, 60	0.64 0.51
Case 4	66	107	432	5	0.59
Case 5	150	180	966	1	0.05
Case 6	20	0	95	20 20, 40, 60, 80	0.59 0.79

^a DOF is the total number of degrees of freedom in the system.
^b The MTS values are also listed for cases 1–3, and 6.
^c “E_{rms}” is the root mean square fluctuation of the total energy over the 140-ps simulation.

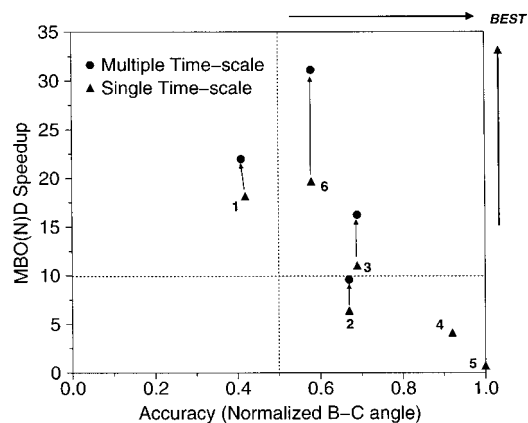


FIGURE 7. Speed up vs. ratio of MBO(N)D to atomistic normalized standard deviation of the B-C interhelical angle.

substructuring leads to better agreement with the atomistic results.

Figure 7 also shows that the MTS integrator increased the CPU speedup over the single time-scale integrator by a factor of 1.2 to 1.6 for the 1CTF simulations. Comparison of the accuracy and speedup values for cases 1 and 2 shows that the inclusion of flexible modes can sometimes result in the need for a smaller integration step size to maintain stability. Including modes yields a smaller speedup, but significantly improves the agreement with atomistic motions, illustrating the speed/accuracy trade-off in MBO(N)D simulations. Note that case 5, which represents a very fine level of substructuring, shows excellent agreement with the atomistic result, albeit with little computational speedup. The domain-based substructuring approach (case 6) shows the greatest level of speedup (factor of 32 with the MBO(N)D/MTS integrator), and still results in a reasonable level of accuracy. In general, one can observe that the finer substructuring schemes require small integration step sizes (1 fs for case 5). Conversely, the coarser substructuring schemes can use large integration step sizes (20 fs base step size for cases 1 and 6). The slowest time-scale bins for the MTS runs were updated at intervals that were as long as 80 fs.

To compare dynamic motion between helices B and C, the power spectra of the helical angle motion for representative cases were calculated for comparison with the atomistic simulation (Fig. 9). As this type of analysis requires a longer simulation for the accurate calculation of the power spectra, 500-ps segments of the MBO(N)D and atomistic production runs were utilized. Both MBO(N)D and atomistic simulations show a very distinctive low fre-

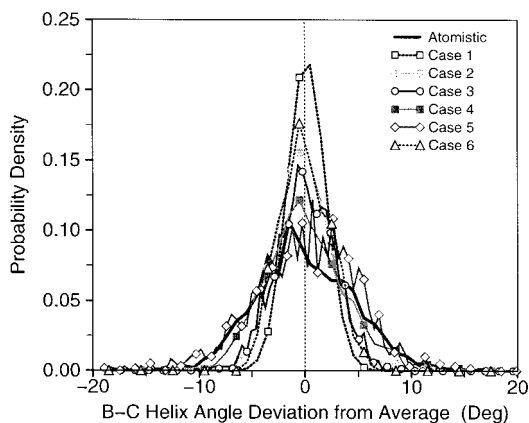


FIGURE 8. Representative probability density functions of the deviation of the B-C interhelical angle about its respective average for several MBO(N)D runs compared with the atomistic result.

quency peak near 8 cm^{-1} . The overall characteristics of case 5 show good agreement with the atomistic results due to the finer substructuring treatment of case 5. Cases 1 and 3 have very simple characteristics in their spectra because substructuring removed most of the atomistic thermal fluctuations in the system. However, each case reproduced the essential dynamic motion observed in atomistic simulations, as can be seen from the agreement of the positions in the strong peaks in each of the spectra.

Figure 10 shows the RMS fluctuations of C_α atoms from the MBO(N)D and atomistic simulations. Except for the finer substructuring schemes, cases 4 and 5, the MBO(N)D results of RMS fluctuations are generally lower in magnitude than the atomistic result because the grouping of the atoms into bodies results in the thermal fluctuations of the atoms being ignored. In general, we would not expect atomistic RMS fluctuations to be a good mea-

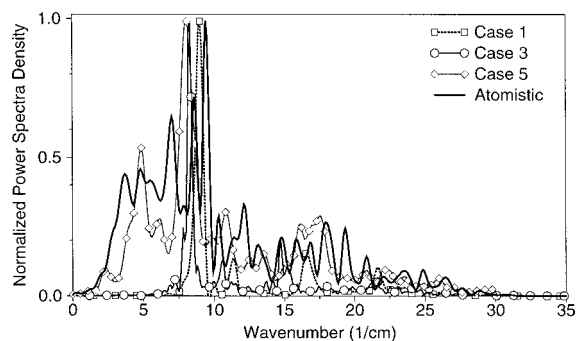


FIGURE 9. Representative power spectra of the B-C interhelical angle motion from MBO(N)D and atomistic simulations.

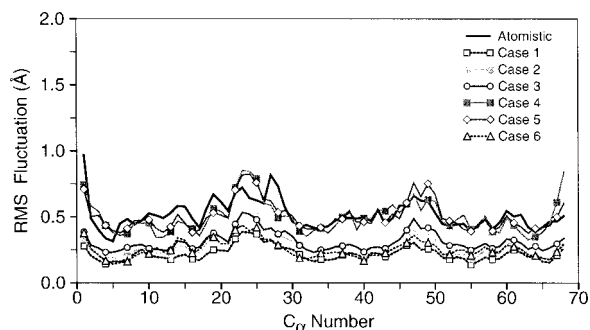


FIGURE 10. Comparisons of the atomistic and MBO(N)D simulations showing RMS fluctuations of C_{α} atoms.

sure of the performance of MBO(N)D, because these detailed motions are ignored in MBO(N)D. Nevertheless, the relative values of the atomistic RMS fluctuation plot are well represented by all of the MBO(N)D simulations.

An analysis of the pseudodihedral angles for the simulations (Fig. 11) shows that the trends in the relative peaks and valleys are consistent for most of the MBO(N)D runs (the coarsest run, case 6, having the least undulations) with the atomistic simulation. The pseudodihedral analysis is a coarser metric than the RMS fluctuation data in Figure 10, yet shows similar trends between the different substructurings.

DNA SHEARING SIMULATION

We carried out a simulated shearing apart of the two strands of a DNA duplex to test the ability of MBO(N)D to reproduce motions in nucleic acids involving large induced conformational changes. The shearing apart of DNA duplexes is an event that occurs, for example, during cell division, and has been studied by Atomic and Chemical Force Microscopy⁸³ and simulation methods.⁸⁴

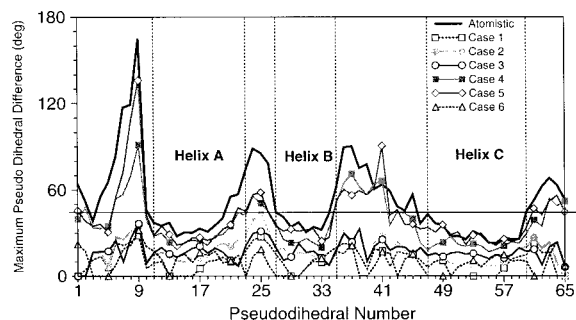


FIGURE 11. Comparisons of the atomistic and MBO(N)D simulations showing pseudodihedral angle range.

Our shearing simulations were designed to be consistent with Atomic or Chemical Force Microscopy experiments, which have suggested that the strands undergo significant conformational changes before separating. The essential dynamics inspired by these previous experiments are the formation of a unique “stretched” form of DNA (S-ladder), and the point at which the strands separate.

The DNA duplex used was the Dickerson dodecamer. The simulation protocol is similar to that described in Konrad and Bolonick.⁶⁴ Initial conditions and simulation protocols are detailed in the supplementary material.

DNA Shearing Results

The first MBO(N)D substructuring scheme, D1, is based on the assumption that the functional groups—phosphodiester ($O(POO^-)O$), sugars, and bases—are essentially rigid, and can be treated as separate bodies. The puckering of the sugars, of course, is an important motion related to certain conformational transitions, but this motion is not critical to the essential dynamics, as we will show. The second substructuring scheme, D2, is similar to D1, except that each phosphodiester group was divided into three bodies: one rigid body, consisting of the phosphorus and two oxygens, and two individual atoms corresponding to the 3' and 5' oxygens. The impetus for the D2 substructuring was based on recent evidence suggesting that the ζ torsion plays a major role in conformational transitions involving the backbone of a nucleic acid.^{89,90} These two substructuring schemes seem to suggest the minimum substructuring needed to reproduce the essential dynamics of DNA shearing while exhibiting significant speedups, as will be shown.

The applied tension and the strand separation distance—measured as the distance between the two terminal atoms to which the pulling forces were applied—as a function of time, are shown in Figure 12 for the MBO(N)D and atomistic simulations. Figure 12 shows that overall, both MBO(N)D simulations capture the essential behavior of the system, although the finer D2 substructuring scheme yields a better quantitative agreement with the atomistic simulation in the comparison of strand separation force. This improvement in the unraveling and separation of the DNA strands is consistent with other evidence that suggests phosphate flexibility is important to the overall dynamics of DNA duplexes under equilibrium conditions.⁹¹ The values of the rupture forces in Figure 12 are almost one order of magnitude higher than those observed by Konrad

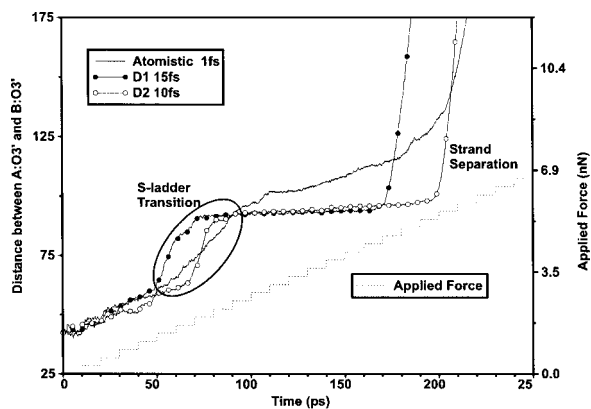


FIGURE 12. DNA strand separation distance as a function of time for three different simulations. Also shown is the applied force at each time step.

and Bolonick. The increased values of the rupture forces in the MBO(N)D simulations may be due to the absence of a good representation of solvent. (See the Discussion section.) Nevertheless, the comparison between MBO(N)D and atomistic simulations is valid because both sets of runs were carried out under the same conditions. Our results are consistent with the experimental observation that the length of the DNA is increased by a factor of two in the stretched conformation over that of the unstretched conformation.⁸³ Additional analyses are shown in the supplementary material.

The more accurate D2 simulation was able to run at a time step of 10 fs, yielding a sevenfold speedup over the atomistic simulation. The less accurate D1 simulation was run at a 15-fs time step with a 10-fold speedup.

HIV-LIGAND PULL

As a demonstration of MBO(N)D's application to a problem of pharmaceutical interest, an MBO(N)D model of HIV protease was developed, and a small molecule (Cbz-Val-Phe-Phe-Val-Cbz, referred to as A74707) was simulated being pulled out of the protein cavity (Fig. 13). The purpose of this test was to see how well MBO(N)D performs on protein-ligand interactions and dynamics. HIV protease is a 3136-atom homodimeric enzyme that cleaves virus-specific polypeptide products, and is critical for virus replication. This aspartyl proteinase is classified as an "all beta" structure, although each polypeptide chain does contain a single nine-residue alpha-helix. The structural features of interest in the protease are the two β -turn structures—flaps—that envelop the ligand. (See supplementary materials for more detail.)

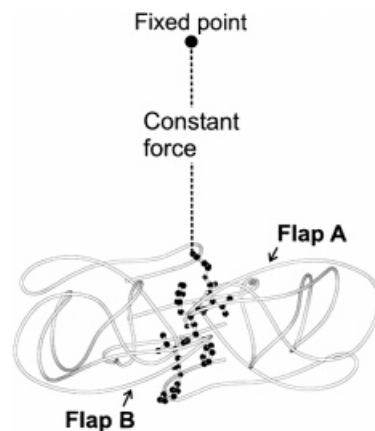


FIGURE 13. A schematic of the pulling experiment for HVP. HVP is drawn as a C_α trace, and the ligand is drawn as a ball and stick. The orientation of the pulling vector is shown; a constant force is applied along this vector between a fixed point 45 Å away in space and a proximate atom on the ligand.

Two distinct conformations identified by X-ray crystallography suggest that HIV protease undergoes a large conformational change upon binding of a substrate or inhibitor. One conformation is an "open flaps" form, which has no inhibitor bound to it; the other a "closed flaps" form, which has an inhibitor bound in the active site.⁹² It is thought that the flap movement is important to the ligand-binding event. An HIV ligand-pull simulation (described below) was developed to investigate this movement. The protocol used in this simulation is similar to atomic force microscopy experiments⁹³ and simulations of streptavidin/biotin⁹⁴ and the DNA shearing described previously.⁸⁸ The initial structure and simulation conditions are described in the supplementary material.

HIV LIGAND UNBINDING RESULTS

The A-74707 was substructured into seven bodies: one body comprised the main-chain atoms, and the remaining side-chain groups were each placed into a separate body. Two substructuring strategies were explored for the HIV protease bodies. The first strategy (H1) groups every three consecutive residues into rigid bodies (cutting at the ψ angle). Our experience with substructuring suggests that cutting every three residues for proteins is a good base substructuring that permits a reasonable compromise between accuracy and speedups. The second strategy (H2) is a finer version of H1. Residues that were within 3.5 Å of the ligand, and showed significant motion from the side-chain analysis (the

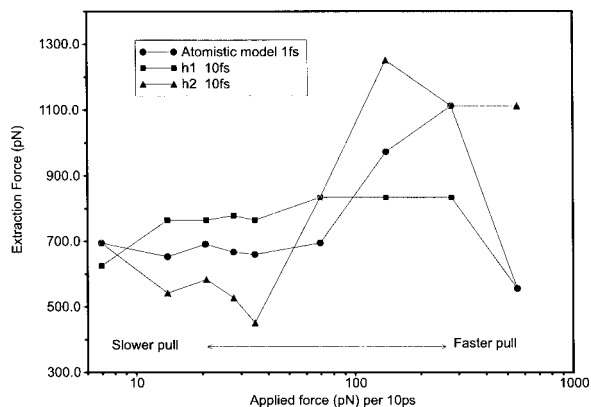


FIGURE 14. Plot of extraction force vs. applied force increment at every 10 ps for atomistic simulations and MBO(N)D simulations using the H1 and H2 substructures.

Side Chains section) of a short atomistic pull simulation were substructured more finely as side-chain bodies (cutting at the ϕ and ψ angles). The H1 and H2 substructuring strategies, along with that for A-74707, resulted in 131 and 145 bodies, respectively, for the protease–ligand system. Figure S7 in supplementary material shows details of the substructured model. In this test case, the amount of motion needed was greater than that allowed by modes; therefore, these simulations were done with small rigid bodies in the hinge-bending regions of the protein.

Figure 14 shows the extraction forces required to pull the A-74707 from HIV protease for a variety of applied forces from the atomistic and MBO(N)D simulations. This figure suggests that both MBO(N)D simulations are converging on the atomistic results in the slow pulling region. The results for H1 converged faster than H2 because fewer degrees of freedom are included. The range of values for the extraction forces in Figure 14 underscores the need to perform multiple simulations with different rates of pulling. Only when the values of the extraction forces have converged within a reasonable range of one another can the contribution to the dynamics from inertial effects be understood. The maximum extraction force was computed as the maximum value of a curve that resulted from averaging every 20 ps over the raw data. We estimate that the errors associated with each point in Figure 14 is 50 pN, which is based on the shift in maximum value when the window of averaging is increased to 40 ps (as was done similarly by Grubmueller⁹⁴).

Figure S8 in supplementary materials shows the ligand escaping the protease, and is representa-

tive of MBO(N)D and atomistic pulling simulations. The range of movement between the flaps was approximately 60° for both methods. These results suggest that MBO(N)D can reproduce the essential features of protein–ligand interactions that involve substantial conformational changes during unbinding. MBO(N)D, however, shows an eightfold CPU speedup over the atomistic simulations.

Discussion

We have developed a modeling methodology based on multibody dynamics, MBO(N)D, for performing long time scale simulations of macromolecules. MBO(N)D is designed to take advantage of the fact that low-frequency motion dominates the overall global motions of macromolecular systems. Our studies on various systems that differ substantially in size and dynamics suggest that the global motions from MBO(N)D are quite comparable to atomistic results. The studies also suggest that our substructuring strategies are applicable to a wide range of molecular systems. Interhelical angle motions, separation forces, order parameters, end-to-end distance, and transitional structures are all properties that seem to be well within the capabilities of MBO(N)D. The maximum speedups associated with these results vary from a factor 8 to 30, depending on the system, type of property, and corresponding substructuring. Detailed or high frequency motions such as RMS atomistic fluctuations are sacrificed for the gains in speed. The size and frequency of the motion of a body affects the maximum time step. That is, some MBO(N)D simulations can tolerate small bodies as long as its frequency of motion is relatively low: for example, buried side chains bodies that were relatively immobile in 1CTF resulted in 10 fs time step, but all side-chain bodies (some of which were relatively mobile) resulted in a much lower time step.

The key elements of MBO(N)D are the grouping of atoms into rigid or flexible bodies (substructuring), and the modeling of body flexibility using a truncated set of body-based mode shape vectors. Both elements serve to eliminate high-frequency motions from the simulation. The remaining low-frequency motions permit the use of long integration time steps that give MBO(N)D a computational speed advantage over atomistic simulations.

Macromolecular systems have diverse local and global dynamical properties, and therefore, no single substructuring strategy works best for all possible cases. We have not identified a single analysis

method that will completely automate the process of substructuring, but have instead relied upon semiautomated approaches. The types of analyses described in this article help in determining the most plausible substructuring for the system under question, with the intent being that the important properties—global properties—will be captured adequately. We have, nevertheless, found that the simple protocol of first defining a “base” substructuring that involves three residues per body for the entire protein, and then refining the substructuring in areas of interest is a reasonable approach to the problem.

We have shown that a multibody protocol coupled with the Lobatto integrator produces very stable dynamics with elevated time steps for constant energy and constant temperature ensembles. As far as we know, the time steps used for the constant energy runs are the largest reported in the literature to date that use atomistic force fields.

The multibody equations of motion in MBO(N)D are solved by an efficient $O(N)$ algorithm that scales linearly with the size of the system. The Lobatto integrator, which is used to propagate the dynamics, is very efficient, and has two main advantages over other integrators for multibody dynamics applications. First, it requires only one force field evaluation per step while being accurate to the second order. This attribute is on par with the Verlet class of integrators for atomistic MD. Second, it accounts for the nonlinear velocity dependency of the acceleration expression. Proper handling of this nonlinear velocity dependency is critical to the efficient and accurate propagation of MBO(N)D dynamics.

The position fluctuations from MBO(N)D simulations at the detailed atomistic level, however, tend to be lower in magnitude than for atomistic simulations. This reduction in atomistic mobility is not surprising because most or all of the internal high-frequency motions have been eliminated. Said differently, the small localized atomistic motions are neglected while the more global motions are retained (e.g., domain–domain bending). Nevertheless, the fluctuation profile—the relative fluctuations among the atoms and residues—is reproduced by the MBO(N)D models. Reducing the degrees of freedom, as noted by others, can also have the undesirable effect of eliminating the coupling between degrees of freedom (e.g., the coupling between bond angles and torsions). This problem can be alleviated, however, by adjusting the force field for the remaining degrees of freedom,¹⁰³ as was done, for example, by Head–Gordon and Brooks for virtual bodies.³

With the macromolecule substructured into bodies, and body flexibility captured with body-based modes, the model allows large relative motions to take place between the bodies, while small deformational motions are assumed within the flexible bodies. Force field interactions are computed in the normal atomistic fashion, with the resulting atom forces being projected into the reduced degrees of freedom, yielding body forces, body torques, and modal forces, for dynamic propagation of the MBO(N)D model. The ability of the bodies to undergo large motions, and the use of the fully nonlinear force field are the features that allow the MBO(N)D modeling approach to capture the important anharmonic effects of molecular dynamics.

The substructuring capability in the MBO(N)D methodology is very versatile. For example, as bodies become smaller (i.e., smaller number of atoms per body, with a corresponding increase in the number of bodies), and more body-based modes are included, the system approaches an atomistic representation. The multiple time scale Lobatto integrator exploits systems with dissimilar sizes and motions, and produces up to a twofold speed improvement over the multibody simulations with single time steps, depending on the system and substructuring used.

There are several potentially useful applications of the MBO(N)D modeling approach, and some examples are briefly discussed in turn.

1. Folding of structural elements in proteins: MBO(N)D substructuring can be used to preserve certain structural features (e.g., an alpha helix) during dynamics. The large-scale motions between these structural elements can then be explored.
2. Protein–ligand, protein–protein, and protein–nucleic acid interactions: the long time scale properties of these complexes continue to be one of the main problems that cannot be treated meaningfully using atomistic methods. The main advantage of MBO(N)D in these cases is speed, and therefore, the ability to explore longer time-scale properties.
3. Supramolecular assemblies and rigid rod polymeric systems: systems that approach mesoscale dimensions are known to have properties that cannot be elucidated from studying the individual component molecules. MBO(N)D provides the framework that can be used to build very large assemblies of molecules and study their dynamical properties.

There are a number of ongoing MBO(N)D development efforts that are being pursued by us and others to broaden its applicability to other types of simulations. The most important of these is the modeling of solvent. Solvent mediates many molecular motions and properties, and correctly accounting for the environment of a molecule is crucial for meaningful results in many cases. Solvent can mediate, for example, bulk properties such as dielectric screening, as well as localized properties such as the geometry and dynamics of side chains at the protein/solvent interface. In special but important cases of localized properties, a molecule of solvent can be considered as part of the ligand binding site in a protein. A good model of solvent will, therefore, address either one or all of these properties adequately, depending on the problem to be studied. The example simulations in this article were performed with a crude implicit solvation protocol (distance-dependent dielectric).

We are considering several other implicit solvation methods approaches for incorporation within the MBO(N)D paradigm that have shown promising simulation results. These methods are the following. First, including only a reduced number of explicit waters that are proximate to the surface of the macromolecule, and that fill the depressions and holes if appropriate. This “thin shell approach” to solvation is the most straightforward approach as each explicit water molecule is treated as a body. Indeed, recent work by Steinbach¹⁰⁴ and Mazur¹⁰⁵ shows that a relatively small number of explicit water molecules is sufficient to solvate myoglobin and DNA with good reproduction of key dynamical properties. A fully solvated system would not be appropriate for MBO(N)D, as the existence of a large number of very small bodies can severely limit the speedups that can be obtained. Second, methods such as electrostatic continuum methods^{106, 107} or empirical solvation potentials,^{112, 113} are consistent with the reduced variable approach of MBO(N)D. These methods have levels of fidelity and time constants that are commensurate with typical MBO(N)D substructured models. Recent results by McCammon, for example, have demonstrated the advantage of the Poisson–Boltzmann Stochastic Dynamics method for reproducing conformational statistics of alanine dipeptide from fully solvated simulations.¹⁰⁸ Studies on macromolecules using these methods, however, are still ongoing. Third, hybrid methods that combine the aspects of explicit molecules of water at the surface of the macromolecule, with the implicit description of water outside the solvent shell. The approaches in this area are

broad, and have been successful in PKa calculations for BPTI,¹⁰⁹ binding free energy calculations for HIV protease inhibitors,¹¹⁰ and the dynamics of enzyme catalysis.¹¹¹

The MBO(N)D modeling methodology has shown promise in providing significant computational speedups while reproducing important dynamical global properties of protein, nucleic acid, and polymeric systems. A number of modifications to the methodology have been identified, that will further enhance MBO(N)D's ability to reproduce other important dynamical behaviors. MBO(N)D represents a new approach to removing the computational bottleneck associated with atomistic methods. The goal of MBO(N)D is to permit the efficient study of very long time-scale properties and of very large molecular systems.

Acknowledgments

The authors wish to express our greatest appreciation to Professor Martin Karplus for his interest in and support of the development of MBO(N)D. He provided many substantial suggestions on methodologies, test cases, and analysis methods. His advice and insights in molecular modeling were invaluable to this effort. In addition, the authors would like to thank the following individuals for their contribution to this work: Dr. Harold P. Frisch for initial discussions and strong encouragement; Dr. Herman van Vlijmen for help on implementing the body-based mode generation capability; Venkataraman Mohan for preprints of his work on nucleic acids; and Dr. Ryszard Czerminski for numerous technical discussions. The National Aeronautics and Space Administration funded initial work for the development of the O(N) algorithm.

Supplementary Materials

A substantial amount of supporting text and figures are contained in the supplementary materials.

References

1. Brooks, III, C. L.; Karplus, M.; Pettitt, B. M. *Advances in Chemical Physics*; John Wiley & Sons: New York, 1988, vol. 71.
2. McCammon, J. A.; Harvey, S. C. *Dynamics of Proteins and Nucleic Acids*; Cambridge University Press: Cambridge, MA, 1987.

3. Head-Gordon, T.; Brooks, III, C. L. *Biopolymers* 1991, 31, 77.
4. Brünger, A. T. *X-PLOR Manual* (version 3); Yale University Press: New Haven, CT, 1992.
5. Jain, A.; Vaidehj, N.; Rodrigues, G. *J Comp Phys* 1993, 106, 258.
6. McCammon, J. A.; Pettitt, B. M.; Scott, L. R. *Comput Math Applic* 1994, 28, 319.
7. Elber, R. *Curr Opin Struct Biol* 1996, 6, 232.
8. Leimkuhler, B.; Reich, S.; Skeel, R. D. *Mathematical Approaches to Biomolecular Structure and Dynamics*; Mesirov, J. P.; Schulten, K.; Sumner, D. W., Eds.; Springer Verlag: New York, 1996, p. 161, vol. 82.
9. Schlick, T.; Barth, E.; Mandziuk, M. *Annu Rev Biophys Biomol Struct* 1997, 26.
10. Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. *J Comp Phys* 1977, 23, 327.
11. Andersen, H. *J Comp Phys* 1983, 52.
12. van Gunsteren, W. F.; Karplus, M. *Macromolecules* 1982, 15, 1528.
13. Rice, L. M.; Brünger, A. T. *Proteins Struct Funct Genet* 1994, 19, 277.
14. Ryckaert, J. P.; Bellemans, A. *Chem Phys Lett* 1975, 30, 123.
15. Mazur, K.; Abagyan, R. A. *Biomol Struct Dynam* 1989, 6, 815.
16. Gibson, K. D.; Scheraga, H. *J Comp Chem* 1990, 1, 468.
17. Mazur, K.; Dorofeev, K. V.; Abagyan, R. A. *J Comp Phys* 1991, 92, 261.
18. Durup, J. *J Phys Chem* 1991, 95, 1817.
19. Durup, J. *Biopolymers* 1992, 32, 561.
20. Streett, W. B.; Tildesley, D. J.; Saville, G. *Mol Phys* 1978, 35, 639.
21. Grubmüller, H.; Heller, H.; Windemuth, A.; Schulten, K. *Mol Sim* 1991, 6, 121.
22. Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J Chem Phys* 1992, 97.
23. Watanabe, M.; Karplus, M. *J Chem Phys* 1993, 99, 8063.
24. Watanabe, M.; Karplus, M. *J Phys Chem* 1995, 99, 5680.
25. Humphreys, D. E.; Friesner, R. A.; Berne, B. J. *J Phys Chem* 1994, 98, 6885.
26. Forester, T.; Smith, W. *Mol Sim* 1994, 13, 195.
27. Procacci, P.; Berne, B. J. *J Chem Phys* 1994, 101, 2421.
28. Greengard, L.; Rokhlin, V. *J Comp Phys* 1987, 73, 325.
29. Schlick, T.; Brandt, A. *Sci Eng* 1996, 3, 78.
30. Briggs, W. L. *A Multi-Grid Tutorial*; SIAM: Lancaster, PA, 1987.
31. Perahia, D.; Levy, R. M.; Karplus, M. *Biopolymers* 1990, 29, 645.
32. Ichiye, T.; Karplus, M. *Proteins* 1991, 11, 205.
33. Hayward, S.; Kitao, A.; Go, N. *Protein Sci* 1994, 3, 936.
34. Amadei, A.; Linssen, A. B. M.; Berendsen, H. J. C. *Proteins Struct Funct Genet* 1993, 17, 412.
35. vanAalten, D. M. F.; et al. *Proteins Struct Funct Genet* 1995, 22, 45.
36. Amadei, A.; Linssen, A. B. M.; deGroot, B. L.; vanAlten, D. M. F.; Berendsen, H. J. C. *J Biomol Struct Dynam* 1996, 13, 615.
37. deGroot, B. L.; Amadei, A.; vanAlten, D. M. F.; Berendsen, H. J. C. *J Biomol Struct Dynam* 1996, 13, 741.
38. Space, B.; Rabitz, H.; Askar, A. *J Chem Phys* 1993, 99, 9070.
39. Balsera, M. A.; Wriggers, W.; Oono, Y.; Schulten, K. *J Phys Chem* 1996, 100, 2567.
40. Peskin, C. S.; Schlick, T. *Commun Pure Appl Math* 1989, 42, 1001.
41. Gear, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*; Prentice Hall: Englewood Cliffs, NJ, 1971.
42. Zhang, G.; Schlick, T. *J Comp Chem* 1993, 14, 1212.
43. Schlick, T.; Olson, W. K. *J Mol Biol* 1992, 223.
44. Hao, M. H.; Pincus, M. R.; Rackovsky, S.; Scheraga, H. *Biochemistry* 1993, 32.
45. Olender, R.; Elber, R. *J Chem Phys* 1996, 105, 9299.
46. Czerminski, R.; Elber, R. *J Quantum Chem* 1990, 24, 167.
47. Onsager, L.; Machlap, S. *Phys Rev* 1953, 91, 1505.
48. Bodley, C. S.; Devers, A. D.; Park, A. C.; Frisch, H. P. *NASA Technical Paper* 1219, 1 (1978).
49. Turner, J. D.; Weiner, P. K.; Chun, H. M.; Lupi, V.; Galion, S.; Singh, U. C. *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*; Van Gunsteren, W. F.; Weiner, P. K.; Wilkinson, A. J., Eds.; ESCOM: Leiden, 1993.
50. Levy, R. M.; Karplus, M.; Kushik, J.; Perahia, D. *Macromolecules* 1984, 17, 1370.
51. Horiuchi, T.; Go, N. *Proteins* 1991, 10, 106.
52. Mizuguchi, K.; Kidera, A.; Go, N. *Proteins* 1994, 18, 34.
53. Brooks, B. R.; Brucoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. *J Comp Chem* 1983, 187.
54. Ding, H. Q.; Karasawa, N.; Goddard, III, W. A. *J Chem Phys* 1992, 97, 4309.
55. de Jalón, J. G.; Bayo, E. *Kinematic and Dynamic Simulation of Multibody Systems*; Springer Verlag: New York, 1994.
56. Chun, H. M.; Turner, J. D.; Frisch, H. P. *Paper AAS 89-457, AAS/AIAA Conf.*, Stowe, VT (1987).
57. Lapidus, L.; Seinfeld, J. H. *Numerical Solution of Ordinary Differential Equations*; Academic Press: New York, 1971.
58. Ferrario, M.; Ryckaert, J. P. *Mol Phys* 1985, 54, 587.
59. Benfield, W. A.; Hrudá, R. F. *AIAA J* 1971, 9, 1255.
60. Craig, R. R., Jr. *Shock Vib Digest* 1977, 9.
61. Lee, A. Y.; Tsuha, W. S. *J Guide Control Dynam* 1994, 17, 69.
62. Hao, M. H.; Harvey, S. C. *Biopolymers* 1992, 32, 1393.
63. Hao, M. H.; Scheraga, H. *Biopolymers* 1994, 34, 321.
64. Durand, P.; Trinquier, G.; Sanejouand, Y. H. *Biopolymers* 1994, 34, 759.
65. Mouawad, L.; Perahia, D. *Biopolymers* 1993, 33, 599.
66. Gerber, P. *Biopolymers* 1992, 32, 1003.
67. Straub, J. E.; Rashkin, A. B.; Thirumalai, D. *JACS* 1994, 116, 2049.
68. Straub, J. E.; Choi, J. K. *J Phys Chem* 1994, 98, 10978.
69. Tirion, M. M. *Phys Rev Lett* 1996, 77, 1905.
70. Ichiye, T.; Karplus, M. *Proteins Struct Funct Genet* 1987, 2, 236.
71. Kabsch, W.; Sander, C. *Biopolymers* 1983, 22, 2577.

72. Swaminathan, S.; Ichiye, T.; Vangunsteren, W.; Karplus, M. *Biochemistry* 1982, 21, 5230.
73. McCammon, J. A.; Karplus, M. *Biopolymers* 1980, 19, 1375.
74. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. *J Chem Phys* 1984, 81, 3684.
75. Rossky, P. J.; Karplus, M. *J Am Chem Soc* 1979, 101, 1913.
76. Czerminski, R.; Elber, R. *J Chem Phys* 1990, 92, 5580.
77. Allen, M. P.; Tildsley, D. J. *Computer Simulation of Liquids*; Oxford Science Publications: New York, 1987.
78. Derreumaux, P.; Schlick, T. *Protiens Struct Funct Genet* 1995, 21, 282.
79. Leijonmarck, M.; Liljas, A. *J Mol Biol* 1987, 195, 555.
80. Aqvist, J.; vanGunsteren, W.; Leijonmarck, M.; Tapia, O. *J Mol Biol* 1985, 183, 461.
81. Aqvist, J.; Leijonmarck, M.; Tapia, O. *Eur Biophys J* 1989, 16, 327.
82. Sanejouand, Y.; Tapia, O. *J Phys Chem* 1995, 99, 5698.
83. Noy, A.; Vezenov, D.; Kayyem, J.; Meade, T.; Lieber, C. *Chem Biol* 1996, 527.
84. Isralewitz, B.; Izrailev, S.; Schulten, K. *Biophys J* 1997, 73, 2972.
85. Dickerson, R. E. *J Biomol Struct Dynam* 1989, 6, 627.
86. Weiner, S. J.; Kollman, P. A.; Case, D.; Singh, U. C.; Ghio, C.; Alagona, G.; Weiner, P. K. *J Am Chem Soc* 1984, 106, 765.
87. Mackerell, unpublished.
88. Konrad, M. W.; Bolonick, J. I. *J Am Chem Soc* 1996, 118, 10989.
89. Chen, Y. Z.; Mohan, V.; Griffee, R. H. *J Biomolec Struct Dynam* 1998, 15, 756.
90. Chen, Y. Z.; Mohan, V.; Griffee, R. H. *Chem Phys Lett* 1998.
91. Bertrand, H. O.; et al. *Nucl Acid Res* 1998, 26, 1261.
92. Collins, J. R.; Burt, S. K.; Erickson, J. W. *Nat Struct Biol* 1995, 2, 334.
93. Florin, E. L.; May, V. T.; Gaub, H. E. *Science* 1994, 264, 415.
94. Grubmüller, H.; Heymann, B.; Tavan, P. *Science* 1996, 271, 997.
95. Erickson, J.; Neidhart, D. J.; VanDrie, J.; Kempf, D. J.; Wang, X. C.; Norbeck, D. W.; Plattner, J. J.; Rittenhouse, J. W.; Turon, M.; Wideburg, N.; et al. *Science* 1990, 249, 527.
96. Kempf, D. J. *J Med Chem* 1990, 33, 2687.
97. Sohn, S. E.; Singer, R. D.; Lamala, S. J.; Kuzyk, M. K. *Polym Mater Sci Eng* 1986, 55, 532.
98. Tsai, M. L.; Chen, S. H.; Jacobs, S. D. *Appl Phys Lett* 1989, 54, 2395.
99. Pojodil, G. M.; Farmer, B. L.; Adams, W. W. *Polymer* 1996, 37, 1825.
100. Grigoros, S.; Lane, T. H. *J Comp Chem* 1988, 9, 25.
101. Caves, L. S. D.; Evanseck, J. D.; Karplus, M. *Protein Sci* 1988, 7, 649.
102. Auffinger, P.; Louise-May, S.; Westhof, E. *J Am Chem Soc* 1995, 117, 6720.
103. Bornemann, F. A.; Schuette, C. *Phys D* 1997, 102, 57.
104. Steinbach, P. J.; Brooks, B. R. *Proc Natl Acad Sci USA* 1993, 90, 9135.
105. Mazur, K. M. *J Am Chem Soc* 1998, 120, 10928.
106. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J Am Chem Soc* 1990, 112, 6127.
107. Scheafer, M.; Karplus, M. *J Phys Chem* 1996, 100, 1578.
108. Gilson, M. K.; McCammon, J. A.; Madura, J. D. *J Comp Chem* 1995, 16, 1081.
109. Russel, A. T.; Warshel, A. *J Mol Biol* 1985, 185, 389.
110. Hansson, T.; Aqvist, J. *Protein Eng* 1995, 8, 1137.
111. Washel, A.; Papazyan, A.; Kollman, P. A. *Science* 1995, 269, 102.
112. Fraternali, F.; van Gunsteren, W. F. *J Mol Biol* 1996, 256, 939.
113. Wesson, L.; Eisenberg, D. *Protein Sci* 1992, 1, 227.