International Journal of Non-Linear Mechanics 43 (2008) 1040 – 1055

Contents lists available at ScienceDirect

International Journal of Non-Linear Mechanics

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

# Substructured molecular dynamics using multibody dynamics algorithms

Rudranarayan M. Mukherjee[a,*,1], Paul S. Crozier[b], Steven J. Plimpton[b], Kurt S. Anderson[a]

[a]Rensselaer Polytechnic Institute, 110 8th Street, Troy NY 12180, USA
[b]Sandia National Laboratories, Albuquerque, New Mexico, USA

ARTICLE INFO

*Article history:*
Received 31 October 2007
Received in revised form 1 April 2008
Accepted 2 April 2008

*Keywords:*
Coarse-grained molecular dynamics
Multibody dynamics algorithms
Open-source software

ABSTRACT

This paper reports a new research effort aimed at using efficient multibody dynamics methods to simulate coarse-grained molecular systems. Various molecular systems are studied and the results of nanosecond-long simulations are analyzed to validate the method. The systems studied include bulk water, alkane chains, alanine dipeptide and carboxyl terminal fragments of calmodulin, ribosomal L7/L12 and rhodopsin proteins. The stability and validity of the simulations are studied through conservation of energy, thermodynamics properties and conformational analysis. In these simulations, a speed up of an order of magnitude is realized for conservative error bounds with a fixed timestep integration scheme. A discussion is presented on the open-source software developed to facilitate future research using multibody dynamics with molecular dynamics.

Published by Elsevier Ltd.

## 1. Introduction

Simulations of the dynamics of biomolecular and polymeric systems are usually characterized by very complex energy surfaces and large conformational spaces. Consequently, there are two fundamental computational challenges which have limited the scope of molecular dynamics (MD) simulations. The first challenge is associated with the spatial size of these systems which makes the calculation of interactions a computationally expensive step in the simulations. The biomolecular or polymeric systems of interest typically contain large number of interaction sites (or atoms) and these systems can span hundreds to many thousand degrees of freedom ($n$).

The calculation of the interaction forces between these sites is a many body problem which traditionally scales as $O(n^2)$. Methods such as [1–3] have been proposed that reduce the cost of this calculation. Although these methods reduce the cost of the force calculations, considering the large value of $n$, this calculation is still computationally expensive and it accounts for most of the computational cost of a simulation.

The second challenge is associated with the diverse frequency content of the system and the large temporal domains that contain the phenomena of interest. As the integration step size is inversely proportional to the highest frequency content of the system, a MD simulation of a fine scale system typically propagates with an integration timestep of only subfemtoseconds. Consequently, the state-of-the-art *long* simulation ranges from nanoseconds to micro-seconds of real time. Many phenomena of interest occur in longer temporal scales ranging from milliseconds to several seconds.

The methods aimed at reducing the computational complexity of MD can be broadly classified into two classes. The first class of methods are those that aim at reducing the complexity of calculating the interaction forces between different elements in a system by re-parameterizing the nature of the force fields [3–6]. These methods have typically focused on producing estimates of the forcing terms, with acceptable accuracy, at greatly reduced cost. The second class of methods is aimed at constraint imposition to truncate the frequency spectrum of the system to allow larger temporal integration step sizes. The different methods that fall under this class include the following.

*Constrained dynamics using explicit constraints* [7–10]: The explicit constraints are coupled with the equations of the unconstrained system resulting in a set of *d*ifferential-*a*lgebraic *e*quations (DAEs) (differential equations of motion and algebraic constraint equations). While representative methods such as SHAKE [7], and others [8–10] have been successfully used, the cost and numerical issues involved in iteratively solving the DAEs can be challenging for larger systems and different types of constraints.

*Constrained dynamics using implicit constraints* [11–13]: This family of methods is commonly called internal coordinate MD. They satisfy the constraints exactly and implicitly through the use of generalized coordinates. The solution of the heavily coupled equations

* Corresponding author. Tel.: +1 818 354 2677.
*E-mail addresses:* Rudranarayan.M.Mukherjee@jpl.nasa.gov (R.M. Mukherjee), pscrozi@sandia.gov (P.S. Crozier), sjplimp@sandia.gov (S.J. Plimpton), anderk5@rpi.edu (K.S. Anderson).
[1] Now at NASA Jet Propulsion Laboratory, Pasadena, CA.

0020-7462/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.ijnonlinmec.2008.04.003

of motion can be computationally expensive unless intelligent, efficient schemes are employed. Reduced computational order algorithms [15–18] have been proposed to reduce the cost of these methods. They use special manipulations to decompose and solve the equations of motion in particular efficient ways (at least for systems involving many generalized coordinates), thus avoiding much of the very costly linear algebra.

*Multirate temporal integration* [18–24]: These methods exploit the multiscale character of biopolymer motions. In what is known as multiple time-scale (MTS) schemes [25] a hierarchy of timesteps is introduced instead of the traditional single timestep. The idea is to take advantage of the terms in the biomolecular Hamiltonian that vary more slowly than other terms. Appropriate timesteps are assigned for the different terms.

*Eigenvector* (*modal*)*-schemes* [26–31]: These schemes use normal modes (free vibration modes), which are obtained from the diagonalization of the Hessian matrix (second derivative of the potential). These modes are used to describe global deformations around a local minimum on the molecular potential energy surface. While these methods allow for a potentially large integration step size, the associated linearization and assumption of modal superposition can restrict the efficacy of these methods.

*Implicit integration schemes* [32–34]: Implicit integration schemes are interpolative in nature, and thus may take much larger stable timesteps than explicit integration schemes which are extrapolative. Compared to explicit integration methods, implicit integration schemes are more expensive as they usually require the solution of a set of coupled algebraic equations. These non-linear systems of equations may be expensive to solve, thereby offsetting any gains made by larger timesteps. Further, the damping introduced by the implicit schemes may pose a problem in maintaining the micro-canonical conditions of the ensembles.

In this paper, a methodology is presented for coarse-grained molecular dynamics (CGMD) simulations by selecting and combining several existing algorithms from different fields of application. Coarse-grained models of molecular systems are developed by imposing constraints that curtail the frequency spectrum, thereby allowing for a larger integration step size. The coarse-grained models of the systems are in the form of articulated multirigid body systems. This method uses efficient multibody dynamics algorithms for solving the equations of motion of constrained MD systems in linear complexity. It uses generalized coordinates and imposes the holonomic constraint implicitly and exactly, not just to integrator tolerance. A two-step iterative integration scheme is used for temporally advancing the states. This scheme requires only one system wide force calculation and takes into account the velocity-dependent gyroscopic and Coriollis forces. The method uses Cartesian force fields and gains from the computational savings obtained from using efficient force field calculation schemes such as Ewald sum and the $P^3M$ methods [1,2]. The use of efficient multibody dynamic algorithms coupled with efficient and highly parallelizable force field calculations is a novel research effort and results in a highly efficient computation method for constrained MD. Two different schemes for constraint imposition are used, viz holonomic distance (and/or) angle constraints as well as aggregation. In the case of holonomic distance constraints, the atomistic representation of the system is retained. However, in the aggregation model, the system is coarse grained into clusters, where each cluster may contain more than one atom. This method is implemented in a C++ object oriented open-source software developed by coupling together the highly parallel MD software LAMMPS (Large-Scale Atomic/Molecular Massively Parallel Simulator) [35] and the efficient multibody software POEMS (Parallelizable open-source Efficient Multibody Software) [36]. The original contributions of this manuscript are (i) the development and validation of the multibody dynamics based method for CGMD simulations and (ii) the development of a research code that embodies this method.

## 2. Method overview

A number of efficient algorithms and models have been synergistically combined together to form a method to facilitate substructured MD simulations. In this method two key aspects, efficient calculation of molecular interaction and efficient solution of multibody equations of motion, have been seamlessly integrated with various aspects of MD simulation protocols. The method supports a whole host of interaction potential fields used for modelling molecular systems. Similarly, the method builds on efficient multibody algorithms such as the $O(n)$ method [43] and the generalized impulse momentum algorithm [45]. The systems studied are coarse grained into articulated system of rigid bodies. As evidenced from the various systems studied, this coarse graining eliminates the high frequency dynamics of the system thereby allowing for a larger integration timestep. This results in significant reduction in the overall computational cost of the simulations.

In this method, the systems of interest are coarse grained into different hierarchies of reduced order models that are subject to implicit constraints. The reduced order models and the constraints eliminate the high frequency components of the system dynamics and allow for a stable, large integration timestep. There are two different representations of the reduced order models, both of which are in the form of an articulated multibody system. The kinematic model used here enforces the constraints implicitly, exactly and non-iteratively using the topological information and generalized (relative) coordinates, thereby reducing the computational complexity.

The first category of models is obtained by imposing holonomic distance-dependent constraints between the interaction sites. The fully atomistic representation of these systems consists of interaction sites connected to each other by stiff springs. In the reduced order model, the axial stiff spring is replaced by a constant length massless rigid link. Each link and the next interaction site it is attached to is treated as a single body with a point mass and negligible moments of inertia. Successive bodies are connected to their inward bodies by kinematic joints allowing only rotational degrees of freedom.

The second category of models is obtained by *aggregation* where a large number of discrete interaction sites are constrained to move as an articulated system of coupled rigid bodies. Each rigid body consists of a number of interaction sites that are coupled together to move as a rigid body. Two rigid bodies are coupled when they share a common interaction site. This common interaction site is treated as a kinematic joint location. The relative motion between two rigid bodies is modelled using joints that allow relative rotational motion and constrain out the translational motion. The relative rotation between bodies in an articulated model is modelled using Euler parameters to avoid any singular configurations. The dynamics of these reduced order models is simulated using the efficient multibody dynamics algorithm described in Section 3.

Among the systems studied using this method, the periodic boundary condition has been used extensively in the simulation of explicit water, alkanes and other polymer simulations. The protein simulations are carried out using the *shrink-wrap* boundary conditions where the simulation box is non-periodic and the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

In this method, the system description is concurrently maintained using two sets of coordinates viz generalized coordinates and Cartesian coordinates. The system is represented in generalized coordinates to describe the kinematics and dynamics of the reduced order models. The generalized coordinates describe the relative orientation and displacement of the reduced order models. The underlying multibody dynamics algorithm also generates the fully atomistic representation of the system in Cartesian coordinates at

negligible cost. This representation is useful in calculating the interaction forces and in implementing the boundary conditions. These Cartesian coordinates reflect the true positions of the system entities and as the system moves, these positions may be outside the simulation box. If forces are calculated based on these true positions, they will not reflect the periodic boundary conditions. To model periodic boundary conditions, in this method the Cartesian coordinates are modified using simple geometric formulae [44] to generate the positions of the images which are inside a simulation box. Forces are then calculated from these modified coordinates using the Cartesian force fields. Under the periodic boundary conditions, the forces acting on the system entities are exactly equal to the forces acting on the images. Hence the forces calculated based on the image locations can be used in conjunction with the articulated body description of the system to solve the equations of motion without having to modify the generalized coordinates.

This method can accommodate both explicit and implicit water models such as Solvent Accessible Surface Area model and the generalized Born model. The implicit solvent models introduce an augmented state-dependent potential and hence are uncoupled from the generation and solution of dynamic equations of motion. In explicit water models, each water molecule is modelled as a single free floating rigid body.

In this work, initial atomic positions for biomolecular simulations are obtained from the protein database (www.pdb.org). Bulk material initial coordinates for small molecules were generated on a regular lattice and subsequently melted and adequately equilibrated. Initial atomic velocities were randomly generated from the Gaussian distribution for the desired simulation temperature.

In this method, the mass properties and the initial state of the system are derived from the fully atomistic model. For the reduced order models, the mass, position of the center of mass and the velocity of the center of mass of any rigid body are generated from the properties of all the atoms that are aggregated into that rigid body. An eigenvalue problem is formulated and solved to generate the temporally invariant principal moments of inertia and the associated directions which form an unambiguous set of body fixed basis vectors. This process gives the properties of each body in the reduced order model as a set of uncoupled rigid bodies.

The generalized coordinates are easily generated from the relative translation and orientation of successive bodies. However, the generation of initial velocities is tricky. Unlike an atomistic formulation, where the velocities are uncoupled, in the articulated model, there exists a recursive relation between the velocities of successive bodies. Hence these cannot be directly generated from a random distribution.

The initial atomistic model of the system is converted to the articulated model by simultaneously and instantaneously imposing a number of kinematic constraints on an initially uncoupled system. The imposition of the constraints is modelled as the application of impulsive loads at each joint. These impulsive loads instantaneous and simultaneously eliminate the high frequency system dynamics. This results in a change in the degrees of freedom and discontinuous jumps in the generalized speeds.

However, as the articulated bodies in the reduced order model are kinematically coupled, the calculation of these impulsive loads and the subsequent effect on the system momentum is not trivial. This method calls upon a generalized impulse momentum algorithm [45] to calculate these impulsive loads and the resulting generalized speeds of the reduced order model from the velocities of the fully atomistic model. This algorithm ensures that the generalized momenta [37] of the system is conserved and results in correct physical behavior of the system after the constraints are imposed. The computational complexity of this algorithm scales linearly and logarithmically with the number of generalized coordinates in the system in serial and parallel implementation, respectively.

For the systems studied in this work, the Lobatto III a-b partitioned Runge–Kutta integration scheme [46] has been used. It is a second order method which iteratively calculates the velocities at the half step. The iteration quickly converges in one or two steps. The method efficiently accommodates the velocity-dependent inertial forces as the velocity is being calculated at the half step. The Lobatto integration scheme requires one force field calculation per integration step and allows for a fairly large timestep with good energy conservation over long durations of simulation. It does not include any excessive data storage problems. Numerical experiments showed that the energy conservation characteristics of the Lobatto integrator is better than the Runge–Kutta and Verlet schemes. However, a small drift can be noticed over time and needs to be corrected by resetting the momentum of the center of mass to zero. The integration scheme proceeds as below where (1) is the half step iterative velocity propagation step with initial guess (2), while (3) and (4) are the full step position and velocity propagation.

$$v(t+\Delta t/2) = v(t) + \Delta t/2 * a[x(t), v(t+\Delta t/2)] \tag{1}$$

$$v^0(t+\Delta t/2) = v(t) + \Delta t * a[x(t), v(t)] \tag{2}$$

$$x(t+\Delta t) = x(t) + \Delta t * v[t+\Delta t/2] \tag{3}$$

$$v(t+\Delta t) = v(t+\Delta t/2) + \Delta t/2 * a[x(t+\Delta t), v(t+\Delta t/2)] \tag{4}$$

$a[x(t), v(t+\Delta t/2)]$ implies that the acceleration is a function of $x$ and $v$ evaluated at instances $t$ and $(t+\Delta t/2)$, respectively.

The use of appropriate force fields for biomolecular systems is essential for generating valid simulation results and this method builds on the progress made in the developments of empirical force fields. The forces can be classified into two types, bonded and non-bonded forces. The bonded forces include the bonds, angles, dihedrals and impropers. These forces are usually responsible for generating the high frequency localized motion of the atoms or interaction sites. In this method, within each component body, all the atoms or interaction sites that comprise the component body have no relative displacements and the forces on these do not change. Consequently, the bonded forces are eliminated and are not accounted for in the model.

The non-bonded interactions include Coulombic and Lennard-Jones forces. While using a rigid-body model, the sum of all non-bonded forces arising between interaction sites within a component body go to zero because of equal and opposite actions. However, the non-bonded forces between the interaction sites in different bodies have to be accurately calculated.

In the calculation of non-bonded forces, there arise two different approaches, both of which can be encompassed in the method. In the first approach, the fully atomistic force fields are retained and all the interactions between the interaction sites that are not on the same body are explicitly calculated. This is computationally expensive, but compared with the fully atomistic case, there are still considerable savings as interactions between sites on the same body are not calculated. The extent of computational efficiency scales with the size of the coarse-grained entities i.e. number of interaction sites per coarse-grained body. There are several different procedures for efficiently calculating the non-bonded interactions and recent reports of gain in efficiency [47] are encouraging. Notable improvements include faster non-bonded electrostatics calculations via meshing [1,2], and use of tabulation and bitmapping schemes [48], some of which [35] are accommodated in the method.

The second approach is based on powerful, yet simple, approaches for deriving CGMD force fields from atomistic force fields [4–6]. The most exciting among these is the multiscale coarse-graining method,

which relies upon a force matching procedure, and easily transitions between quantum, atomistic and CGMD scales. The method obtains CGMD effective pairwise interactions between groups of atoms from short atomistic simulations that use well-parameterized biomolecular force fields. These force fields reduce the number of interaction sites and the resulting coarse-grain models have one interaction site per coarse-grained entity.

From the perspective of the method discussed here, the calculation of the interaction forces is a state-dependent term and does not represent a coupling in solving the dynamics equations. All and any of the different force calculation strategies can be accommodated in the method. The calculation of the force fields and the methods for generating and solving the equations of motion are independent from each other and can function as completely uncoupled algorithms. The first approach discussed above has been used for the systems simulated in this work.

As described above, this method of substructured MD is based on the central idea of implicit imposition of constraints and the use of multibody dynamics algorithms in MD simulation protocols to efficiently simulate molecular systems. The method is obtained by integrating various existing algorithms, some of which were developed for application in a completely different field. The novelty of our work is in the integration of state-of-the-art techniques from different fields to propose an efficient solution to the important problem of computational efficiency of biomolecular simulations. The method embodies the benefits of efficient force calculation using efficient methods such as [3–6] and efficient algorithms for dynamics simulation such as [41,43,45]. The next section presents a brief overview of the multibody dynamics algorithm [41,43] which is central to this work.

## 3. $O(n)$ algorithm overview

This algorithm [41,43] uses the projection method as promoted by Kane and others [37] and uses internal coordinates instead of the Cartesian coordinates to formulate and solve the equations of motion. Formulations of this type [38–42] have been in use for a number of years for macroscopic problems such as vehicle dynamics, robotic, etc. A generic discussion of the particular method used in this work is provided here for the readers' convenience. While the discussion is limited to a systems in chain topologies for ease of explanation, the method is equally applicable for generic tree (or branching) topologies.

The algorithm begins by generating a topology or connectivity map of the system in terms of relative coordinates, joints and body fixed reference frames to form kinematic serial chains or tree topologies. Consider the *parent–child* notation associated with the description of an arbitrary set of interconnected rigid bodies shown in Fig. 1.

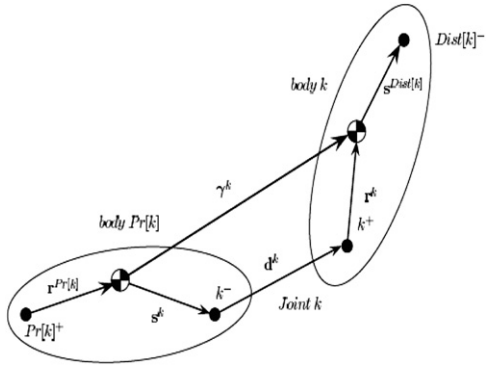



**Fig. 1.** Notation associated with an arbitrary set of interconnected rigid bodies.

The parent body Pr[$k$] is connected to its child body $k$ through joint $k$, via joint points $k^-$ and $k^+$ which reside in bodies Pr[$k$] and $k$, respectively. The set of bodies, Dist[$k$], in the system topology which are downstream from body $k$ (away from the base) are referred to as the *distal* or child bodies of the body $k$. The position vector $\boldsymbol{s}^k$ locates joint-$k$ relative to the mass center of body Pr[$k$], while the position vector $\boldsymbol{r}^k$ locates the mass center of body $k$ with respect to the outward end of this same joint. The position of child mass center $k^*$ relative to proximal mass center Pr[$k$]$^*$ is described by the vector $\boldsymbol{\gamma}^k$.

The angular velocity of any body $k$ and the translational velocity of its associated mass center $k^*$ with respect to the Newtonian reference frame $N$, may always be written in terms of the generalized speeds as

$$\boldsymbol{\omega}^k = \sum_{r=1}^{n} \boldsymbol{\omega}_r^k u_r + \boldsymbol{\omega}_t^k \tag{5}$$

$$\boldsymbol{v}^{k^*} = \sum_{r=1}^{n} \boldsymbol{v}_r^{k^*} u_r + \boldsymbol{v}_t^{k^*} \tag{6}$$

In these expressions $\boldsymbol{\omega}_r^k$ and $\boldsymbol{v}_r^{k^*}$ are termed the *rth partial angular velocity of body k* and *rth partial velocity of point k*$^*$ with respect to the reference frame $N$. These quantities may be thought of as basis vectors for the space of admissible system velocities and angular velocities, while the associated generalized speeds are the velocity space measure numbers. Additionally, the terms $\boldsymbol{\omega}_t^k$ and $\boldsymbol{v}_t^{k^*}$ appearing in Eqs. (5)–(6) are referred to as the *angular velocity remainder term* of body $k$ and *velocity remainder term* of point $k^*$ with respect to the reference frame $N$. These quantities are most often associated with specified/prescribed motion, and thus are not associated with the time derivative of a generalized coordinate.

In spatial form, the *velocity*, *partial velocity* and *velocity remainder term* matrices are represented as

$$\underline{\mathscr{V}}^k = \begin{bmatrix} \underline{\omega}^k \\ \underline{v}^{k^*} \end{bmatrix}, \quad \underline{\mathscr{P}}_r^k = \begin{bmatrix} \underline{\omega}_r^k \\ \underline{v}_r^{k^*} \end{bmatrix} \quad \text{and} \quad \underline{\mathscr{V}}_t^k = \begin{bmatrix} \underline{\omega}_t^k \\ \underline{v}_t^{k^*} \end{bmatrix} \tag{7}$$

With these matrices so defined, Eqs. (5) and (6) may be expressed as

$$\underline{\mathscr{V}}^k = \underline{\bar{\mathscr{V}}}^k + \underline{\mathscr{V}}_t^k = \sum_{r=1}^{n} \underline{\mathscr{P}}_r^k u_r + \underline{\mathscr{V}}_t^k \tag{8}$$

Similarly the generalized acceleration matrix of an arbitrary body $k$ [41,43] can be represented as

$$\underline{\mathscr{A}}^k = \begin{bmatrix} {}^N\underline{\alpha}^k \\ {}^N\underline{a}^{k^*} \end{bmatrix} = \underline{\bar{\mathscr{A}}}^k + \underline{\mathscr{A}}_t^k \tag{9}$$

where $\underline{\mathscr{A}}^k$ is also be divided into two portions. One is $\underline{\bar{\mathscr{A}}}^k$, which contains all terms which are explicit in the unknown state derivatives $\underline{\dot{u}}$ and the other is the *acceleration remainder term* $\underline{\mathscr{A}}_t^k$, which represents all of the state-dependent acceleration terms (gyroscopic, Coriollis and/or prescribed).

The basic recursive $O(n)$ algorithm for performing forward dynamics simulation consists of three principle steps, or "sweeps" of the system topology.

*Kinematics sweep*: The kinematics sweep starts at the base body and works outward to the tip while recursively using the kinematic relations to generate the partial velocities, transformation matrices, translational and rotational velocities and state-dependent acceleration terms in the body basis. With the generalized velocity, generalized acceleration and generalized acceleration remainder term matrices as represented above, it is possible to compactly represent the recursive relationships necessary for determining all system kinematic quantities. As has been demonstrated in [41] we have

$$\underline{\mathscr{V}}^k = [(\underline{\mathscr{S}}^k)^{\mathrm{T}} \underline{\bar{\mathscr{V}}}^{\mathrm{Pr}[k]} + \underline{\mathscr{P}}_k^k \underline{u}_k] + \underline{\mathscr{V}}_t^k \tag{10}$$

and

$$\underline{\mathscr{A}}^k = [(\underline{\mathscr{S}}^k)^{\mathrm{T}}\,\bar{\underline{\mathscr{A}}}^{\mathrm{Pr}[k]} + \underline{\mathscr{P}}^k_k \dot{\underline{u}}_k] + \underline{\mathscr{A}}^k_t \tag{11}$$

The quantity $\underline{\mathscr{S}}^k$ appearing in Eqs. (10)–(11) is the basis consistent linear transformation matrix

$$\underline{\mathscr{S}}^k = \begin{bmatrix} \underline{\mathscr{C}}^k & \underline{\mathscr{C}}^k \underline{\gamma}^k_\times \\ \underline{0} & \underline{\mathscr{C}}^k \end{bmatrix}_{6\times 6} \tag{12}$$

Within this expression $\underline{\mathscr{C}}^k \equiv {}^{\mathrm{Pr}[k]}\underline{\mathscr{C}}^k$ is the direction cosine matrix which relates the body $k$ basis vectors to those fixed in its parent body Pr[$k$]; $\underline{0}$ is a $3\times 3$ zero matrix; and $\underline{\gamma}^k_\times$ is the skew symmetric matrix equivalent to the vector cross product operation $\boldsymbol{\gamma}^k\times$.

The body $k$ *generalized inertia* $\underline{\mathscr{I}}^k$ and the body $k$ *generalized force* $\underline{\mathscr{F}}^k$ matrices are defined as

$$\underline{\mathscr{I}}^k = \begin{bmatrix} \underline{I}^{k/k^*} & \underline{0} \\ \underline{0} & \underline{M}^k \end{bmatrix}_{6\times 6} \tag{13}$$

$$\underline{\mathscr{F}}^k = \begin{bmatrix} \underline{T}^k - (\underline{I}^{k/k^*}\underline{\alpha}^k_t + \underline{\omega}^k_\times \underline{I}^{k/k^*}\underline{\omega}^k) \\ \underline{R}^k - \underline{M}^k \underline{a}^k_t \end{bmatrix}_{6\times 1} \tag{14}$$

Within these expressions, $\underline{I}^{k/k^*}$ is the $3\times 3$ inertia matrix of body $k$ with reference to its center of mass and $\underline{M}^k$ is the diagonal matrix whose diagonal elements are equal to the mass of the body. By comparison $\underline{T}^k$ and $\underline{R}^k$ represent the resultant of all moments and forces, respectively, acting on body $k$ through its center of mass $k^*$.

*Triangularization sweep*: The triangularization procedure works recursively inward (tip to base) to compute the articulated inertia mass matrix $\mathscr{I}^k_3$ and the articulated generalized active force $\mathscr{F}^k_3$ expressed as

$$\tilde{\boldsymbol{I}}^k = \mathscr{I}^k_1 + \sum_{j\in \mathrm{Dist}[k]} \tilde{\boldsymbol{T}}^j \mathscr{I}^j_3 (\mathscr{S}^j)^{\mathrm{T}} \tag{15}$$

$$\tilde{\boldsymbol{F}}^k = \mathscr{F}^k_1 + \sum_{j\in \mathrm{Dist}[k]} \tilde{\boldsymbol{T}}^j \mathscr{F}^j_3 \tag{16}$$

where Dist[$k$] is the *distal* (children) set associated with body $k$. The triangularization operator $\mathscr{T}^j$ and the basis consistent shifting operator $\mathscr{S}^j$ used in (15)–(16) are defined as

$$\mathscr{T}^j = (\mathscr{S}^j)^{\mathrm{T}} \left[ \boldsymbol{U} - \frac{1}{\mathscr{M}_j} \tilde{\boldsymbol{I}}^j \mathscr{P}^j_j (\mathscr{P}^j_j)^{\mathrm{T}} \right] \tag{17}$$

$$\mathscr{S}^j = \begin{bmatrix} \mathscr{C}^j & \boldsymbol{0} \\ \boldsymbol{0} & \mathscr{C}^j \end{bmatrix} \begin{bmatrix} \boldsymbol{U} & \boldsymbol{\gamma}^j_\times \\ \boldsymbol{0} & \boldsymbol{U} \end{bmatrix} \tag{18}$$

The quantity $\mathscr{M}_k$ is also given by

$$\mathscr{M}_k = (\mathscr{P}^k_k)^{\mathrm{T}} \mathscr{I}^k_3 \mathscr{P}^k_k \tag{19}$$

*Back-substitution sweep*: At the base body, information associated with an entire set of outboard bodies has been accumulated and is explicitly available such that the equation $\mathscr{M}_1 \dot{\boldsymbol{u}}_1 = \mathscr{P}^1_1 \tilde{\boldsymbol{F}}^1$ can be isolated and yields the solution of $\dot{\boldsymbol{u}}_1$. The solution for $\dot{\boldsymbol{u}}_1$ is then substituted into the next equation to solve for $\dot{\boldsymbol{u}}_2$. Proceeding in this manner, a generalized function expression for the solution of each generalized acceleration $\dot{\boldsymbol{u}}_k$ is given as follows:

$$\dot{\boldsymbol{u}}_k = \frac{(\mathscr{P}^k_k)^{\mathrm{T}}}{\mathscr{M}_k} [\tilde{\boldsymbol{F}}^k - \tilde{\boldsymbol{I}}^k \ (\mathscr{S}^k)^{\mathrm{T}} \ \mathscr{A}^{k-1}] \tag{20}$$

with $\mathscr{A}^k$ computed from

$$\mathscr{A}^k = (\mathscr{S}^k)^{\mathrm{T}} \mathscr{A}^{k-1} + \mathscr{P}^k_k \dot{\boldsymbol{u}}_k \tag{21}$$

Thus, with the recursive inverse sweep using Eq. (21), the accelerations and the state derivatives of the system can be calculated, thereby solving the equations of motion. As is apparent, the algorithm uses recursive sweeps and efficient manipulations to generate and solve the equations of motion of the multibody system. For further details of this algorithm, refer [41,43].

## 4. Software development

The two research codes used in this work were the classical MD package LAMMPS [35] and the multibody dynamics software POEMS [36]. Both these codes were separately developed by the authors and as a part of this effort were integrated together for CGMD simulations using multibody methods. LAMMPS is distributed freely as open-source software under the GPL. POEMS is also open source, with a BSD-style license. The two codes have different functions which for the work discussed in this paper are complementary.

LAMMPS models an ensemble of particles representing an atomic, polymeric, biological, metallic, or coarse-grained system using a variety of force fields and boundary conditions. It runs either in serial or as a distributed-memory parallel program, dividing atoms across processors by spatially partitioning the simulation box. LAMMPS implements various force fields, including the commonly used CHARMM [61,62] and AMBER [63] force fields for biological molecules, and Ewald or particle-mesh solvers for long-range electrostatics of systems with charged particles. Details of LAMMPS capabilities and simulation results are discussed in [35].

POEMS is a multibody dynamics code which models the dynamics of reduced order or coarse-grained systems. It was designed to efficiently simulate systems with large numbers of articulated rigid bodies. It contains libraries with different dynamics formulations for efficiently generating and solving the equations of motion of articulated systems as well as different time integration schemes such as *Runge Kutta* 4-5, *Verlet and Velocity Verlet* and the *Lobatto Partitioned III a-b Runge Kutta* 4-5 schemes.

Currently POEMS has three algorithms for solving equations of motion of articulated multirigid body systems including a traditional $O(n^3)$ complexity solver based on Kane's method [37], an $O(n)$ complexity recursive solver [43] and an $O(\log(n))$ complexity [64] solver for parallel implementations. POEMS also contains an implementation of a generalized impulse momentum formulation for correct kinematic coarse graining of reduced order models. This is a novel feature that enforces the correct initial conditions required to preserve the essential dynamics of the systems, allowing smooth transitions between models of different resolutions.

To model the problems discussed in this paper, LAMMPS and POEMS were coupled together as described below. The C++ object oriented nature of the two codes facilitated an easy coupling between them.

LAMMPS was used as the driver or parent code, calling POEMS as a library at the appropriate points within each MD timestep. The problem definition (atoms, molecular topology, force fields for pairwise and bond/angle/dihedral/etc. interactions) was formulated as LAMMPS input. LAMMPS allows new C++ classes to be added to the code which define operations to be invoked at prescribed points within a timestep. Parameters controlling these operations can be specified in the input script that runs a simulation. A wrapper around the POEMS library was written that became one of these classes. Arguments to the corresponding input command specify which atoms should be grouped and treated as rigid bodies. Atoms common to two or more rigid bodies are connection points between the bodies, so that the collection of rigid bodies forms one or more articulated bodies with interior *joints* from the POEMS perspective. This topology information is passed to the POEMS library during a setup phase before a dynamics run is launched, along with other needed information, such as the total mass and inertial moments of each body.

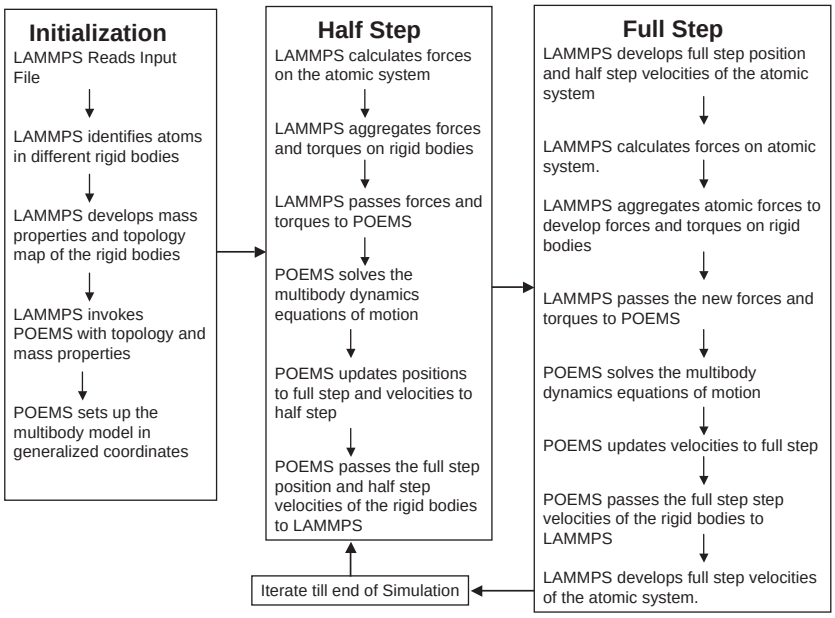



**Fig. 2.** Flow chart of the LAMMPS–POEMS coupling.

LAMMPS and POEMS both use a two-step temporal integration scheme with the same half step and full step integration timestep. LAMMPS uses a Velocity-Verlet scheme for atoms in the system that are not part of rigid bodies (e.g. solvent molecules) while POEMS uses a Lobatto scheme for the rigid bodies. Consequently, the new wrapper class is invoked three times during any integration step in the simulation.

To begin an integration timestep, POEMS solves the equations of motion of the articulated multirigid body system and takes the first half step accounting for the joint constraints, and updates the position, velocity, orientation and angular velocity of each body. LAMMPS also takes a half step and updates the position and velocities of atoms that are not in the rigid bodies. POEMS calls on the wrapper class to pass the updated information for the rigid bodies to LAMMPS. LAMMPS uses this information to update the positions and velocities of the atoms in the rigid bodies. Based on these new atomic positions, LAMMPS calculates the forces on each of the atoms in the system. The new class sums these forces over the atoms in each rigid body to compute the net force and torque on each body. We note that the calculation of forces between atoms within a single body (intra-body forces) can be turned off by appropriate settings in the LAMMPS input script to save computation, since they do not contribute to the external force or torque on the body. This can be a significant savings for rigid bodies containing a large number of atoms. The wrapper class is then invoked for a second time when LAMMPS updates the forces and torques on the rigid bodies in POEMS. POEMS once again solves the equations of motion of the articulated system using the new forces from LAMMPS. It then takes the remaining half step of the two-step integration scheme and updates the translational and angular velocity of the bodies while maintaining the joint constraints. LAMMPS also takes the half step and updates the velocities of the atoms in the system that are not in the rigid bodies. The wrapper class is then invoked for the third time when POEMS passes these updated velocities of the rigid bodies to LAMMPS. Using this information, LAMMPS updates the velocities of the atoms in the rigid bodies. Thus from the LAMMPS perspective, the atoms in bodies maintain their rigid form with respect to each other, and their motion is properly constrained by the defined joints. Fig. 2 summarizes the coupling between the two codes.

All the operations on atoms in this formulation occur in parallel, since the atoms in rigid bodies are distributed across processors by LAMMPS. This includes the computation of body forces and torques and the resetting of atom positions and velocities after POEMS performs its integration. Currently POEMS operates in serial to generate and solve the equations of motion of the constrained system and the temporal integration using the Lobatto scheme. This means each LAMMPS processor passes a list of properties for all rigid bodies defined in the entire simulation domain to POEMS and each processor duplicates the serial POEMS calculation. For simulations we have run thus far, this has not been a bottleneck, since the dominant portion of the overall dynamics calculation is in the MD which scales with the number of atoms, not the number of bodies. The POEMS library and the wrapper class are now distributed with LAMMPS which can be downloaded from the LAMMPS website [35].

## 5. Applications and results

This section describes the systems simulated using this method and presents the results of the simulations. The validation of the simulations is obtained through monitoring the stability of the simulations in conserving energy in micro-canonical (NVE) ensembles, generating thermodynamic properties comparable to atomistic simulations, or by studying the conformational motion of the system. The initial position data for water, alkanes and polymer systems were generated on crystalline lattices and subsequently melted and equilibrated at the desired temperature and pressure. The computational savings achieved by this method are discussed in the next section.

### *5.1. Explicit rigid water*

Simulation of 1369 rigid water molecules was the first simple test that was conducted to examine the validity of the new multibody dynamics simulation capabilities. The TIP3P model of water was used and the box of water molecules was equilibrated by constant pressure (NPT) simulation with a set point at 308 K and 1 atm. Three different test cases were run and the results were compared to validate the method without consideration for computational efficiency. The first test used a rigid water simulation using the

implementation of the SHAKE algorithm in LAMMPS. The second test case used rigid water model in Cartesian coordinates using traditional Newton–Euler formulation. The third used the method discussed in this paper using generalized coordinates to model rigid water molecules. All three test cases were performed in the NVE ensemble using a 2-fs timestep and full long-range electrostatics, with the Lennard-Jones interactions switching smoothly to zero between 8 and 10 å.

Averages and standard deviations of temperature, pressure, kinetic energy and potential energy for the three test cases are shown in Table 1. Good agreement between all three test cases is demonstrated for each of the thermodynamic properties. Note that the large fluctuations in pressure are typical of liquid simulations of this size. The trajectories from the three test cases are not expected to be identical, but the same phase space is being explored as is evidenced by the good statistical agreement in thermodynamic properties. Visual comparison of the trajectories showed very minor difference in trajectories obtained from the three methods, with a minor drift observed due to slight differences in temporal integration.

**Table 1**
Thermodynamic properties of box of rigid TIP3P water molecules.

| Method | $T$ (K) | $P$ (atm) | KE (kmol) | PE (kmol) |
|---|---|---|---|---|
| SHAKE | 307 ± 1 | 215 ± 104 | 2501 ± 11 | 3246 ± 11 |
| RIGID | 306 ± 1 | 31 ± 185 | 2493 ± 9 | 13 239 ± 9 |
| POEMS | 307 ± 1 | 137 ± 128 | 2503 ± 21 | 13 248 ± 20 |

*Note*: $T$ (temperature), $P$ (pressure), KE (kinetic energy), PE (potential energy). SHAKE, RIGID and POEMS correspond to different methods in LAMMPS.

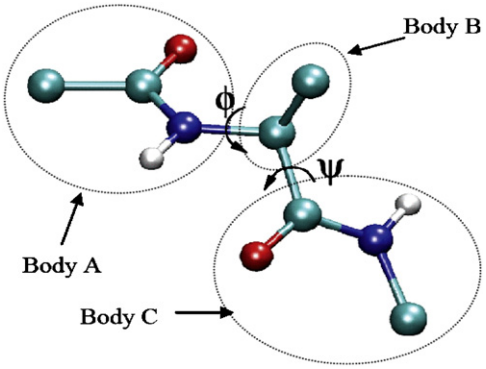



**Fig. 3.** Alanine dipeptide molecule with two 4-atom peptidic groups modelled as rigid bodies as shown. Carbon is shown in cyan, nitrogen in blue, hydrogen in white and oxygen in red.

### 5.2. Alanine dipeptide

The next system that has been tested is alanine dipeptide as shown in Fig. 3. Note that $CH_3$ groups are modelled as single interaction sites and both planar peptide groups are modelled as rigid bodies, as shown in the figure. This system is small and simple enough to afford easy comparison with an atomistic model.

Even with this reduced model, the expected conformational motion of the molecule is observed. At 300 K, alanine dipeptide transitions between two distinct conformations: the C5 conformer (defined here as $\phi = -140 \pm 30$ and $\psi = 140 \pm 30$) and the C7eq conformer ($\phi = -80 \pm 30$ and $\psi = 80 \pm 30$). The system was modelled using the CHARMM 19 force field in implicit solvent (distance-dependent dielectric). Lennard-Jones and Coulombic interactions were smoothly switched to zero between 7 and 7.5 å. Nanosecond-long simulations were performed of both systems using a 1 fs timestep.

Relative populations of the two conformers as well as transition rates between them for both the atomistic and rigid-body models were computed, as shown in Fig. 4. The plot for the rigid-body model is on the left and the plot for the atomistic model is on the right. The computed C5/C7eq ratio was 0.269 for the atomistic model and 0.362 for the rigid-body model. Transition rates were 0.399 and 0.488 $ps^{-1}$, respectively. While there is a difference between the values, overall it can be concluded that there is good agreement between the atomistic and rigid-body models for alanine dipeptide and in agreement with results in [46].

### 5.3. Alkanes

Three simulations of boxes containing 216 chains of alkanes each of chain lengths 8, 16 and 32 were simulated using periodic boundary conditions in an NVE ensemble. The united atom TraPPE force field was used in these simulations. The axial vibrations of the beads were constrained by modelling the stiff carbon–carbon bonds as fixed length massless rigid links. This rendered the model as articulated chains of point masses connected by rigid links and kinematic joints. Three different timesteps (1, 2 and 3 fs) were used to run 1 ns simulations. While the simulation of water was used to validate the modelling approach for discrete rigid bodies, this example was used to simulate the bond constrained model. Stable simulations were achieved with good energy conservation with an increase in timesteps by a factor of 4–6.

Table 2 shows the thermodynamic properties of the systems calculated at different timesteps. The thermodynamic properties were calculated at every 200 fs and the average and standard deviations in temperature, kinetic, potential and total energies are shown.

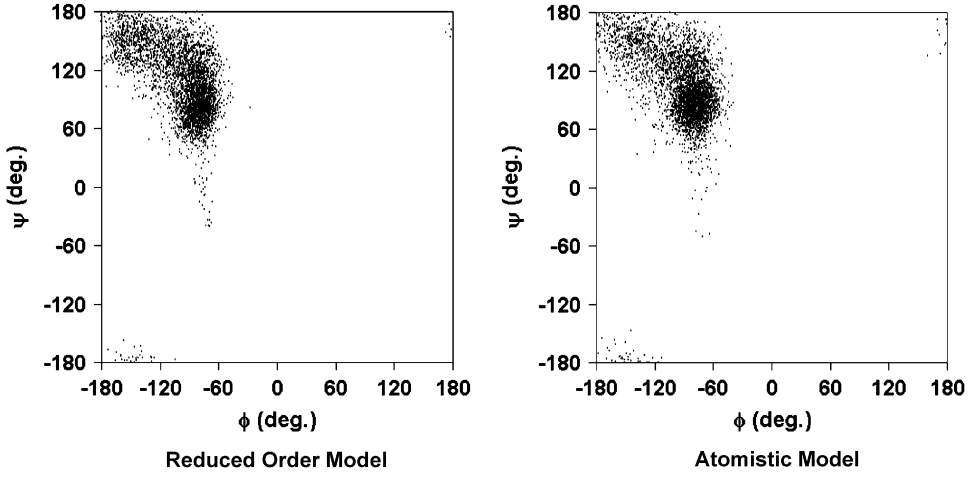



**Fig. 4.** Ramachandran plots from alanine dipeptide simulations.

**Table 2**
Thermodynamic properties of box of alkanes.

| Test-case | $T$ (K) | $E$ (kmol) | KE (kmol) | PE (kmol) |
|---|---|---|---|---|
| Octane 1 fs | 119 ± 5 | −100 ± 2.5 | 617 ± 24 | −720 ± 26 |
| Octane 2 fs | 121 ± 5 | −101 ± 2 | 628 ± 24 | −730 ± 26 |
| Octane 3 fs | 122 ± 5 | −100 ± 3 | 630 ± 23 | −731 ± 23 |
| 16-mer 1 fs | 155 ± 6 | 928 ± 4 | 1600 ± 63 | −671 ± 66 |
| 16-mer 2 fs | 154 ± 6 | 933 ± 2 | 1588 ± 63 | −654 ± 63 |
| 16-mer 3 fs | 155 ± 7 | 951 ± 20 | 1602 ± 75 | −650 ± 62 |
| 32-mer 1 fs | 248 ± 4 | 7078 ± 2 | 5125 ± 74 | 1942 ± 74 |
| 32-mer 2 fs | 255 ± 5 | 7261 ± 142 | 5255 ± 111 | 2006 ± 74 |
| 32-mer 3 fs | 260 ± 7 | 74 255 ± 206 | 5338 ± 150 | 2087 ± 75 |

*Note*: $T$ (temperature), $P$ (pressure), KE (kinetic energy), PE (potential energy).

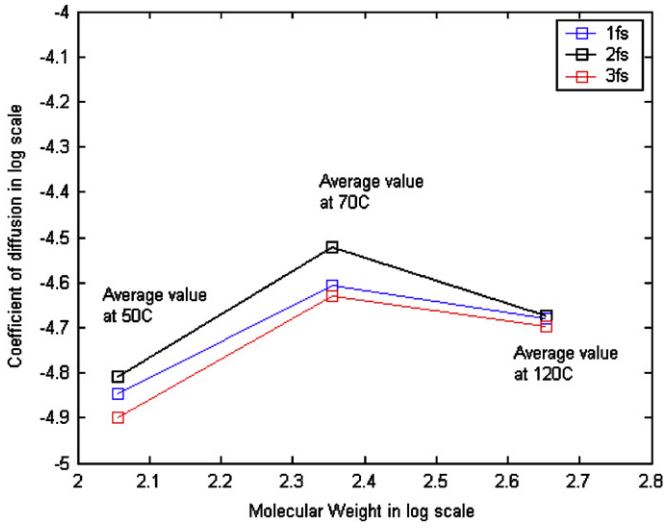



**Fig. 5.** Diffusion coefficients from long time-scale alkane simulations as computed using the Einstein formula.

Good agreement between all three test cases is demonstrated for each of the thermodynamic properties. The variation in temperature is low and in all three cases retains an average value. The total energy of the systems is expected to be conserved. However, as a non-symplectic integrator is used, variations are observed in the total energy. The energy conservation is much better at the lower timestep, and the variations increase with higher timesteps. With a 3 fs timestep, the variation in energy is only about 2% for the 1 ns simulation. This variation is an acceptable variation [44].

Fig. 5 shows diffusion coefficients as calculated by the Einstein formula. It is plotted on a logarithmic scale against molecular weight. The values calculated for three different timesteps are shown and there is good agreement between the three. The calculated values of the coefficient of self-diffusion are in agreement with the results presented in [49].

### 5.4. Polymer melts

The bead spring model of the polymer chains is replaced by an articulated body model made of beads connected by massless rigid links, thereby constraining the inter-bead axial vibration. The connection of successive links to beads is modelled using kinematic joints and relative degrees of freedom. This system is again an example of the bond constrained system.

The system consists of 32 polymer chains and each chain has 16 beads. The interaction between the beads is modelled as non-bonded

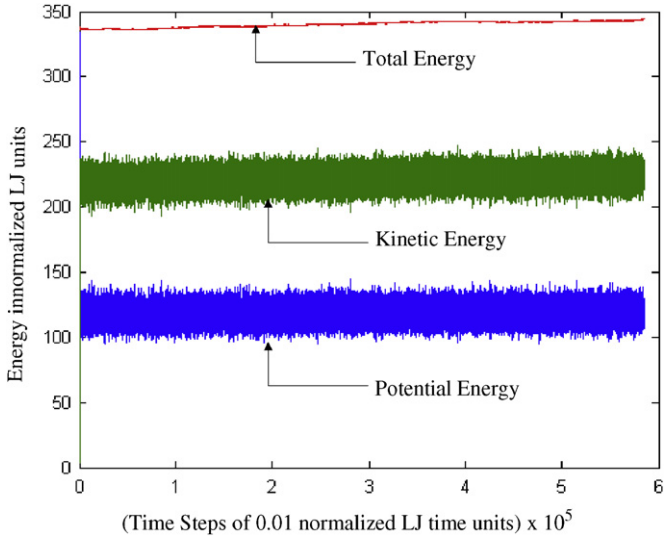



**Fig. 6.** Energy conservation over long temporal scales.

inter-atomic interactions, by using the normalized Lennard-Jones pairwise potential. The system is modelled under periodic boundary conditions and as a constant energy ensemble. For the simulations discussed here, the simulation box is a cube and the material packing fraction is maintained at about 0.85.

Fig. 6 shows a plot of the kinetic, potential and total energy of the system over long temporal scales for a constant energy ensemble. In the figure, a drift in the total energy can be observed. However, the drift is small, the ratio of standard deviation to mean value is only about 0.63%. By eliminating the high frequency axial vibration of the beads, an increase in the timestep by *an order of magnitude* is obtained.

A few critical structural properties commonly studied in polymer melt simulations are also presented. Fig. 7 shows the plot of the mean-square displacement of the center of mass of the system ($g3$) and that of the middle beads in the chain ($g1$) on a log–log scale. The $g3$ values attain a slope of 1 and maintain a constant value of 1. This is in agreement with characteristic features of fluids, further indicating the validity of the model.

The mean square end-to-end vectors ($\langle R_2 \rangle$) and the mean square radius of gyration ($\langle R_{g2} \rangle$) calculated from the trajectory for polymer chains are shown in Fig. 7. As expected, both these values are varying with time but the moving averages of both quantities maintain a constant value. Further, the ratio of $\langle R_2 \rangle / \langle R_{g2} \rangle$ is about 6, as expected of Gaussian chains.

The systems discussed above were either discrete rigid-body simulations or systems with holonomic constant bond length constraint in NVE ensembles. The simulations indicate good energy conservation even with increased timestep size. The structural properties calculated from these simulations agree with the traditional values obtained from the literature. In the following the simulation of systems with coupled rigid bodies where the bodies are rigid aggregates of atoms is discussed.

### 5.5. Calmodulin (1CLL) simulations

Calmodulin (CaM) is an example of a protein that undergoes changes in conformation as reported in [50–52]. CaM is a ubiquitous, calcium-binding protein that can bind to and regulate a multitude of different protein targets, thereby affecting many different cellular functions. CaM is also known to have interactions with a wide variety of proteins including kinases and cyclases.

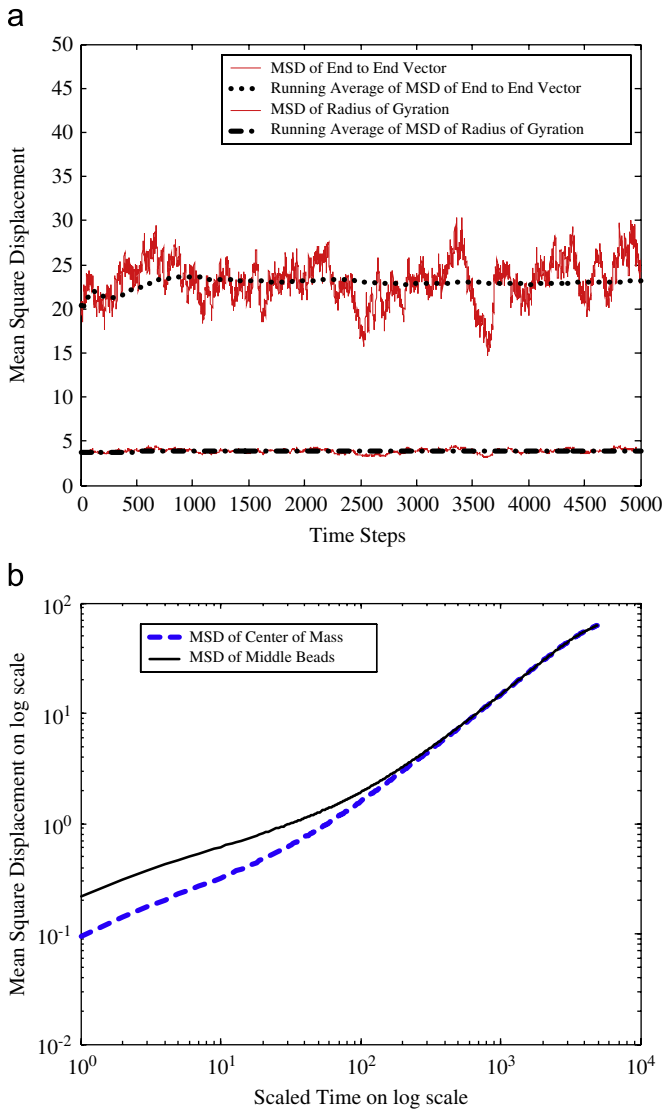



**Fig. 7.** Structural properties of long time polymer melt simulations.

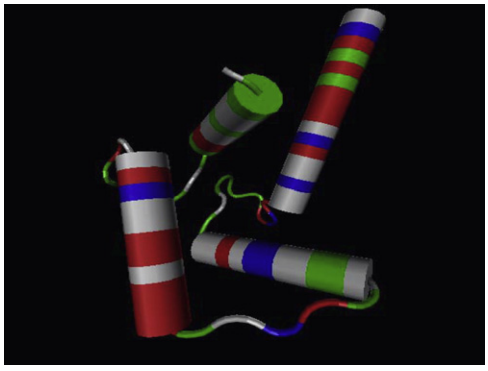


**Fig. 8.** The C-terminal region calmodulin.

A representation of the protein is shown in Fig. 8. It shows the C-terminal region of the protein. Both the C- and N-terminals have two helices in what is called the *E–F* motif. The protein is known to have two conformations, *open* and *closed*. In the *open* conformation, the helices are oriented nearly perpendicular to each other, while in the *closed* conformation, these helices are nearly anti-parallel.

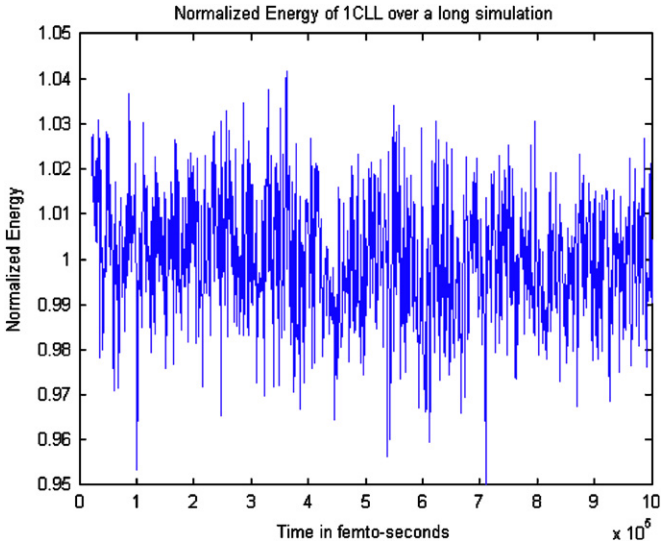



**Fig. 9.** Energy plot of long time simulation of coarse-grained calmodulin protein.

Studies on the relative orientation of the N- and C-terminal domains have indicated that these domains manifest rigid-body behavior and [50] gives a good coarse-grained model of the protein where the $\alpha$ helices and $\beta$ strands are treated as rigid bodies. In the loops, the peptide planes are also treated as rigid bodies.

A similar coarse-graining strategy has been used in simulations reported here. The system consists of about 1100 atoms and was enclosed in a cubic simulation box of length 100 å using non-periodic or *shrink-wrap* boundary conditions. The interactions were modelled using Lennard-Jones potentials and long-range Coulombic interactions were modelled using the CHARMM [53] force field. The implicit solvent version of the CHARMM force field was used in lieu of an explicit model. A timestep of 10 fs was used in the Lobatto integration scheme. The potential energy of the initial configuration of the system was minimized using the minimizer in LAMMPS [35] with a tolerance of $10^{-4}$. The results of a 1 ns simulation of the C-terminal of the protein in NVE ensemble are presented here.

The 1 ns simulation evidenced a good energy conservation with a timestep of 10 fs. Fig. 9 shows the fluctuations in the total energy of the system. The energy has been normalized by the mean energy value over the duration of the simulation. As can be observed, the total energy of the system remains fairly conserved with less than 3% variation in the energy.

The conformational motion of the protein is well depicted in the changes in the $\phi$–$\psi$ torsional angles. Fig. 10 presents the Ramachandran plot. The plot shows the variations of the $\phi$–$\psi$ angles during the entire simulations with the $\phi$ and $\psi$ angles running along the horizontal and vertical axes, respectively, from 180° to −180°. The Ramachandran plot shows three distinct domains of motion shown by the blue smears. These correspond to the relative motion between the $\alpha$ helices, between the $\beta$ strands and in the loop domains.

The root mean squared displacements (RMSD) variations of the $\phi$ and $\psi$ angles at each residue are shown in Fig. 11. The residues with variations in the angles, correspond to the residues in the C-term loop, between the helices and the loops, the loops and the strands and the N-term loop. The residues where there are no changes in the angles are the ones which are constrained into rigid bodies. The plots show separate regions, corresponding to the constrained residues. These corresponds to $\alpha$-helix 1, the first $\beta$ strand, helices 2 and 3, the second $\beta$ strand and the $\alpha$-helix 4, respectively. Though the plots

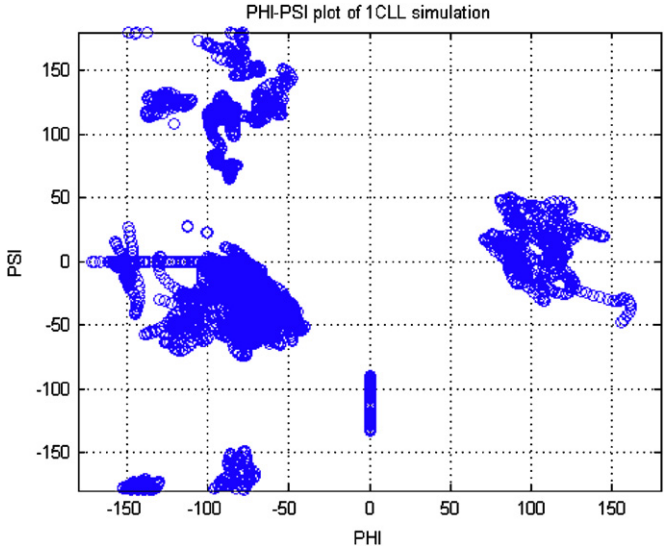



**Fig. 10.** $\phi$–$\psi$ plot of long time simulation of coarse-grained calmodulin protein.

show large localized variations in conformation with as much as 70° variation in angles, the simulation is not able to depict the transition between the *open* and *closed* conformations. This is expected as the time-scale of the simulation is smaller than the transition time-scale (milliseconds). Further, the simulation presented here is only for validating the method and does not include the physically significant hydrophobic surfaces [50] which are expected to initiate the transition.

The per residue B factors for the C-terminal of CaM obtained from the simulation and the theoretical values are plotted in Fig. 12. The B factors were calculated using the relationship [54]

$$B = \frac{8\pi^2}{3}\langle(\Delta r)^2\rangle \tag{22}$$

where $\langle(\Delta r)^2\rangle$ is the mean square of the displacement of $C_\alpha$ atoms averaged over the trajectory. The magnitude of the B factors obtained are smaller than the theoretical values. This can be attributed to the coarse graining of the model, the absence of physically important $Ca^+$ and hydrophobic surfaces [50] and the short time-scale of the simulation. However, the overall trends in the B factor are in agreement with the theoretical values and indicate the validity of the conformational motion.

Fig. 13 shows the contact map of the protein while Fig. 14 RMSD of the protein during the simulation. The contact map shows the average $C_\alpha$ inter-residue distance during the course of the simulation and is in agreement with a similar plot shown in [55]. A black colored square corresponds to a 0.0 å distance indicating near neighbors or at contact distance, white squares correspond to distances greater than 10 å, while the gray ones indicate distance between 0 and 10 å. The plot shows a clear backbone of the protein shown along the diagonal, and also the interactions between different residues shown in gray in the off-diagonal squares. Similarly, the RMSD plot shows that the protein undergoes conformational displacement to attain a different configuration than the starting configuration and is in agreement with a similar plot in [52].

Finally, Fig. 15 shows the spectral plot of the simulation in logarithmic scale. As expected, the plots show dominant low frequency motion corresponding to the orientational changes between the helices and the strands. It also shows low amplitude minor contributions from intermediate frequencies corresponding to the inter-residue movement in the loops where only the peptide planes are constrained as rigid bodies. This is a finer scale model

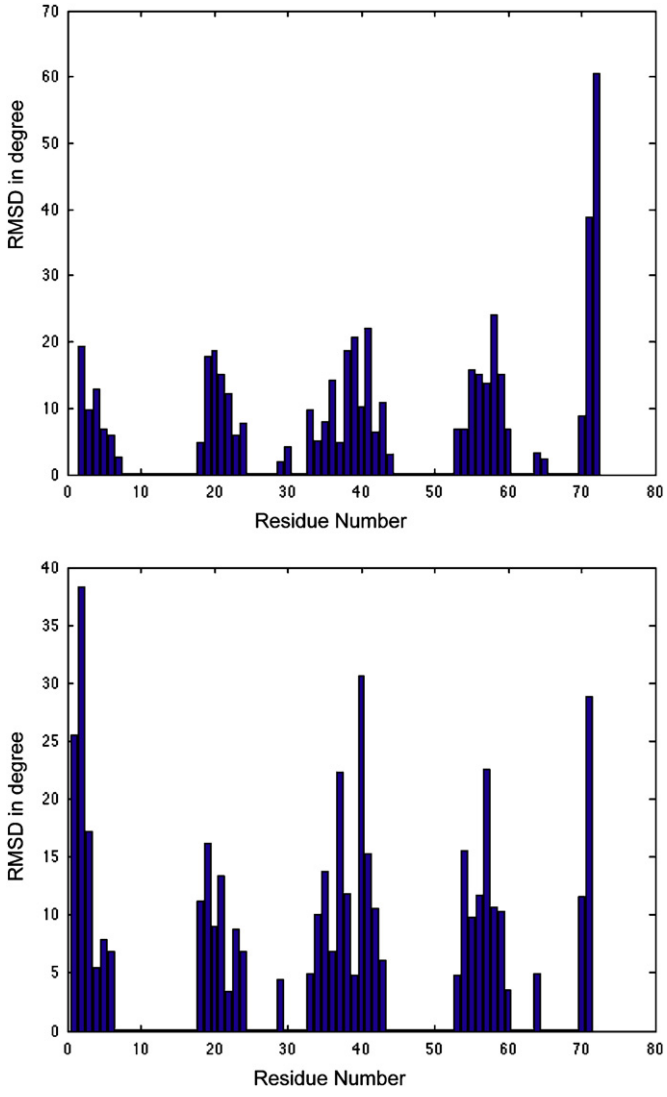



**Fig. 11.** Per residue $\phi$ and $\psi$ angle plot of long time simulation of coarse-grained calmodulin protein.

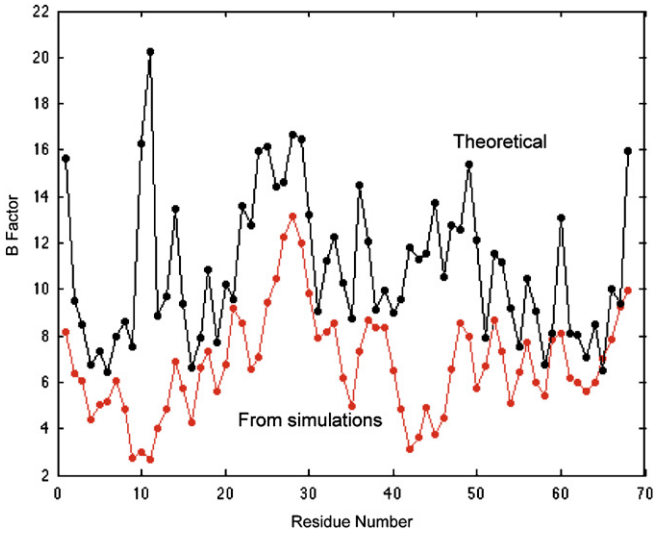



**Fig. 12.** Comparison of B-factor calculated for calmodulin.

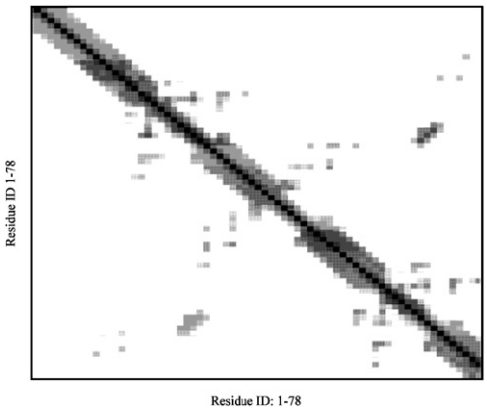



**Fig. 13.** Inter-residue contact map of long time simulation of coarse-grained calmodulin protein.

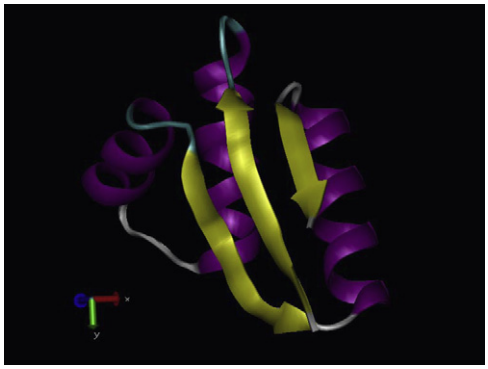


**Fig. 16.** Coarse-grained ribosomal protein L7/L12.

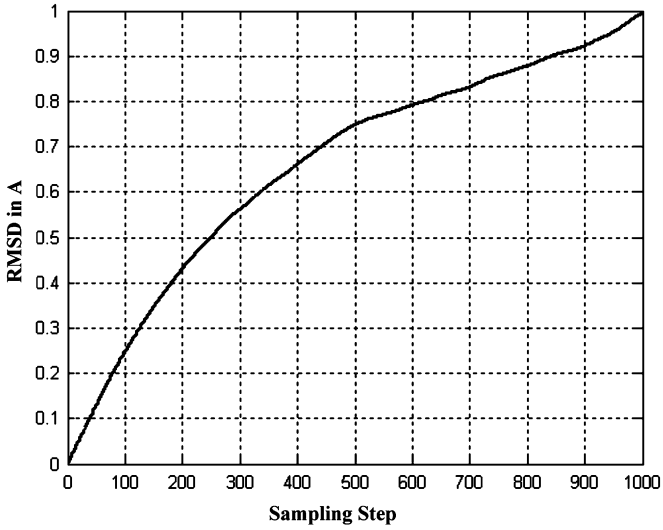



**Fig. 14.** RMSD of long time simulation of coarse-grained calmodulin protein.

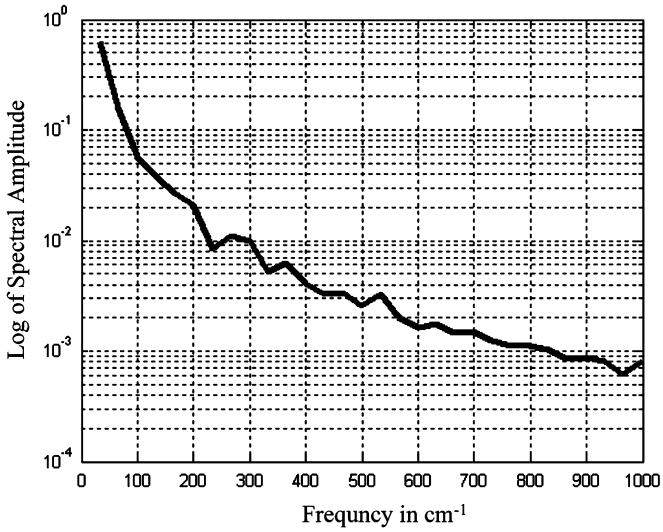



**Fig. 15.** Spectral plot of long time simulation of coarse-grained calmodulin protein.

compared with the rigid helices and strands and hence exhibits some intermediate frequency motion.

### 5.6. C-terminal of ribosomal protein L7/L12

In this section, simulation studies of the C-terminal fragment of the L7/L12 ribosomal protein (1CTF) are presented. The C-terminal end of the protein has interesting functional aspects, including efficient polypeptide synthesis in bacteria. It has been shown [56] that this protein exhibits low frequency motion which appears as a fluctuation of relatively large amplitude between the alpha helices and the beta strands. This low frequency motion can be characterized as rigid-body motion as the residues in the corresponding domains show highly correlated motion.

The C-terminal fragment of the protein is well folded, with secondary structures accounting for approximately 76% of its structure. This secondary structure consists of three $\alpha$ helices and two $\beta$ strand domains. Fig. 16 shows the topology of the protein fragment in the cartoon representation where the topology consists of $\beta$ strand 1, $\alpha$ helices 1 and 2, $\beta$ strand 2 and finally the $\alpha$ helix 3. The C-term consists of 64 residues and in simulations reported here, the system is substructured into 31 small rigid bodies with hinges at the $\phi$ or $\psi$ angles. The simulation is monitored for energy conservation and preservation of essential dynamics.

Fig. 17 shows the fluctuations in the total energy of the system for a 1 ns simulation carried out with a 10 fs integration timestep. Here too, the energy has been normalized with respect to the mean energy value for the entire simulation. As can be observed from the figure, the simulation results in good energy conservation with about 2% variation in energy.

As with the previous test case, the conformational variations of the protein are studied using the Ramachandran plot and the per residue $\phi$ and $\psi$ angle plots. Fig. 18 shows the Ramachandran plot, where the relative motion between the residues are plotted on the $\phi$–$\psi$ angle scale, which is in agreement with the comparable plot presented in [57]. The smear of the dots on the plot shows the variations of respective pairs of the $\phi$–$\psi$ angles.

The plot clearly evidences relative motion between the residues in and off the secondary structures. Note that the data points shown by single points (i.e. not smeared) correspond to angles which are held constant due to coarse graining used in the model. This is further analyzed in Fig. 19 which shows the RMSD variations in the $\phi$ and $\psi$ angles at each residue in the system. The plots clearly separate the regions corresponding to the secondary structures. In contrast with the corresponding Fig. 11 for the CaM simulation, this plot shows a relatively large domain where the RMSD values remain at zero. This is because a large number of residues are contained in the secondary structure which accounts for about 76% of the C-term. The plots

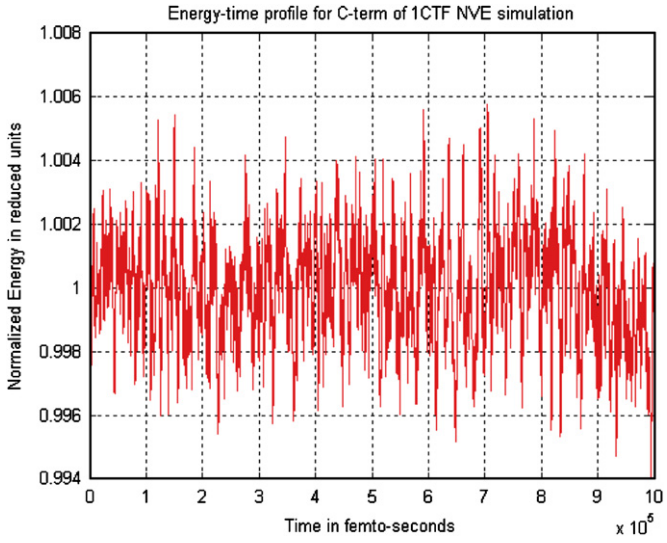



**Fig. 17.** Energy plot of long time simulation of coarse-grained ribosomal protein L7/L12.

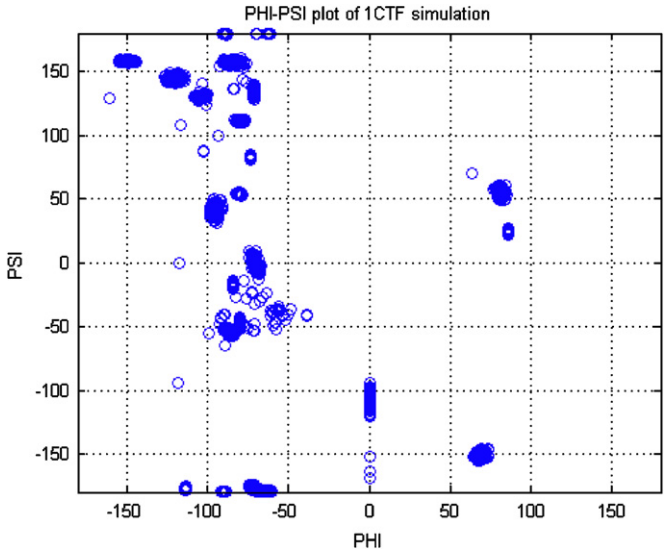



**Fig. 18.** $\phi$–$\psi$ plot of long time simulation of coarse-grained ribosomal protein.

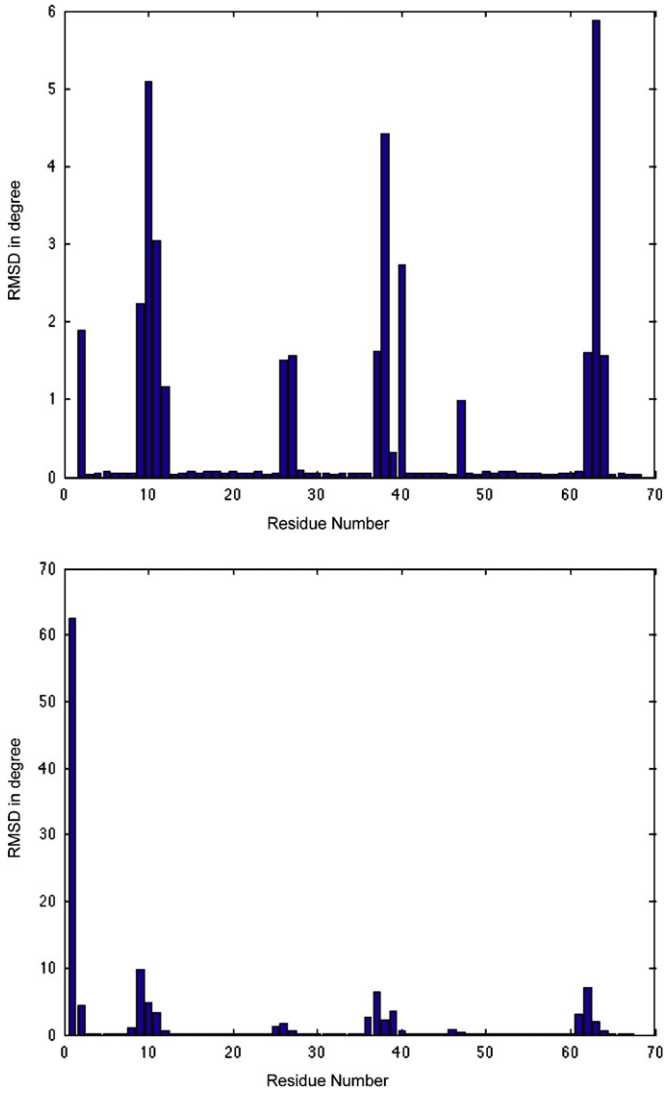



**Fig. 19.** Per residue $\phi$ and $\psi$ angle plot of long time simulation of coarse-grained ribosomal protein.

show variations of about 20° which indicate localized motion. There are no discontinuous changes in the angle values indicating that the simulation does not suffer from abrupt conformational changes.

Fig. 20 shows the spectral plot of the simulation in logarithmic scale. The dominant motion is captured by the low frequency peak at about 5 $cm^{-1}$. This corresponds to the motion between the helices and the beta-strands as has been reported in various previous studies. However, the plot also shows relatively smaller, yet distinct domains of low amplitude relatively higher frequency motion components. These correspond to the localized vibrations in the residues not contained in the secondary structure. As observed in the Ramachandran plot and the per residue $\phi$–$\psi$ RMSD plots, there are several $\phi$–$\psi$ angle variations which do not correspond to the secondary structure motion. The spectral plot presented here clearly shows the dominant low frequency mode of motion and is in agreement with results in [46].

Fig. 21 shows the per residue B factors for the protein obtained from the simulation and the theoretical values. The magnitude of the B factors obtained are smaller than the theoretical values. However, the overall trends in the B factors are in agreement with the theoretical values and supports the validity of the simulated motion.

### 5.7. RuBisCO

A coarse-grained model of the RuBisCO (ribulose-1, 5-bisphosphate carboxylase/oxygenase) enzyme was built and tested. RuBisCO is the primary carbon fixation enzyme in most plants and bacteria. Its poor specificity and inefficiency represent the bottleneck of carbon sequestration and the photosynthetic process. Mutations of residues in the C-terminal region affect the opening/closing rate of the gating mechanism, which in turn affects the enzyme's performance (Fig. 22).

The fully atomistic model of the C-terminal of RuBisCO consists of 510 mobile atoms modelled using harmonic bond potentials, CHARMM angle and dihedral potentials and non-bonding CHARMM Lennard-Jones and Coulomb force fields with cut offs at 8 and 10. The remaining atoms are constrained to remain immobile during the simulation but contribute to the potential acting on the mobile atoms. In the substructured model, the system consists of 11 rigid

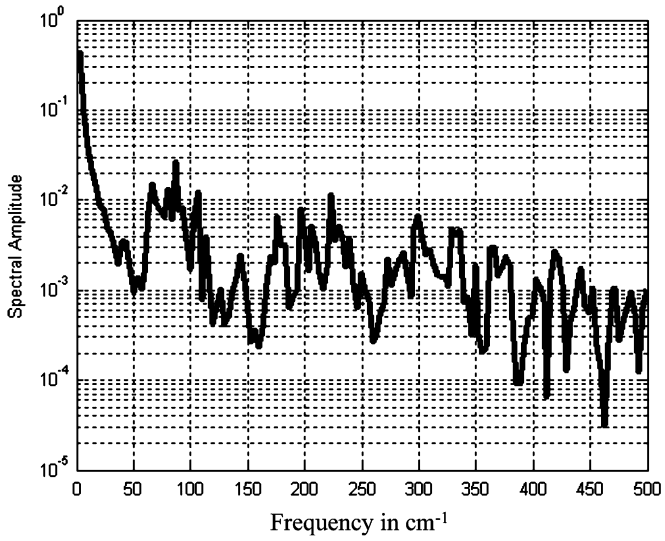



**Fig. 20.** Spectral plot of long time simulation of coarse-grained ribosomal protein.

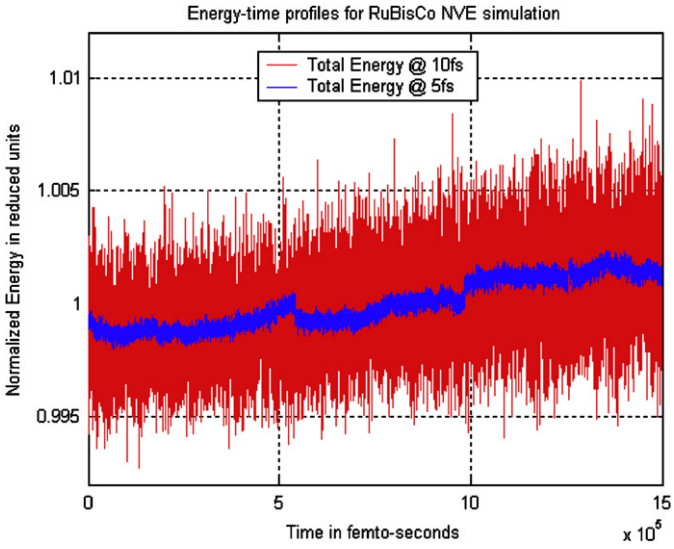



**Fig. 23.** Energy drift plot of long RuBisCO protein simulation.

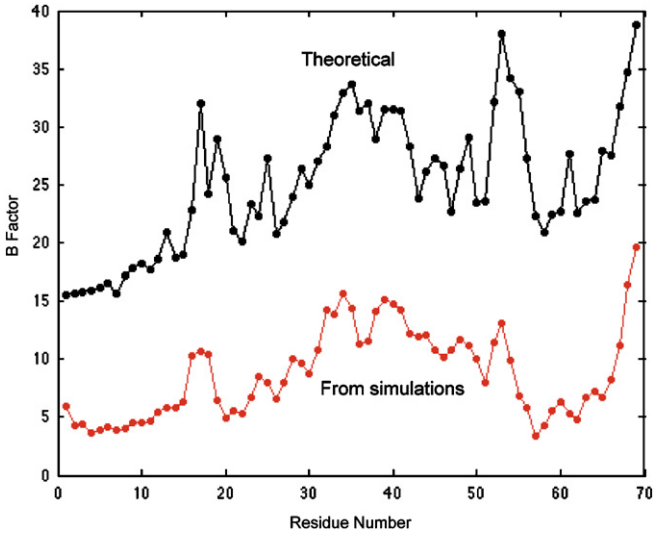



**Fig. 21.** Comparison of B-factor calculated for ribosomal.

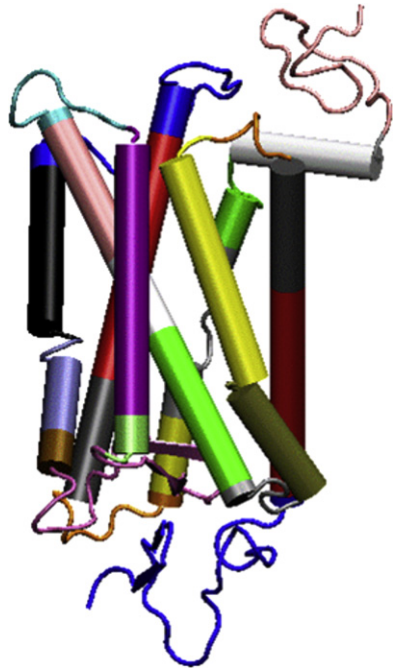


**Fig. 24.** Coarse-grained rhodopsin protein.

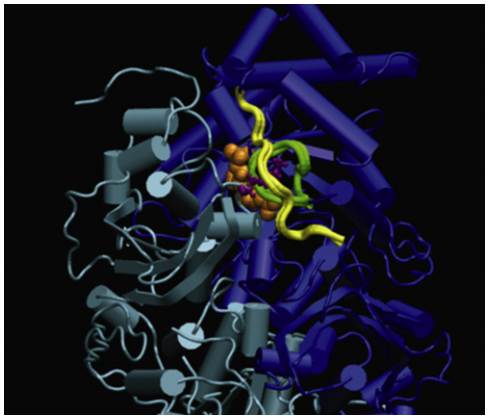


**Fig. 22.** Coarse-grained RuBisCO protein.

bodies connected together to form an articulated serial chain topology. All atom explicit force calculations are supported by this model. However, intra-body interactions between the atoms that make up a rigid body are ignored as these would sum to zero. This is a modest sized problem which is a good one for the validation of this modelling approach. Using this model, NVE simulations were run for 1.5 ns. Different timesteps were used to determine the drift in energy as a function of timestep. Fig. 23 shows the fluctuations in the normalized energy of the system. For the 5 fs timestep, the energy conservation is excellent and even for the 10 fs timestep, the energy conservation is within 1%. Ignoring the intra-body atomistic interactions gives an immediate computational saving. Further, by using a rigid-body model, a 10-fold larger timestep could be used.

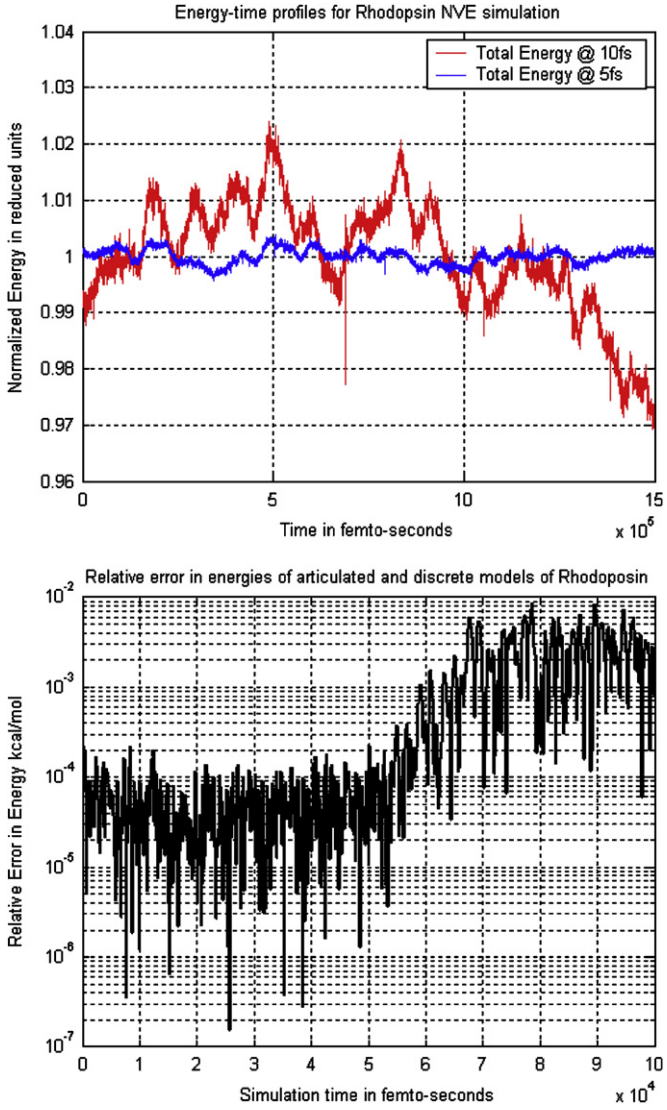



**Fig. 25.** Energy results for long time coarse-grained rhodopsin protein simulation.

### 5.8. Rhodopsin

Rhodopsin is a G-protein coupled receptor (GPCR) with known tertiary structure. It is a prototypical GPCR and a good candidate for the study of GPCR transduction due to the large body of experimental information available about its structure and function. Its light-driven photocycle plays an essential role in generating visual response. In prior studies [58], rhodopsin has been found to exhibit essentially rigid-body motion between the key secondary structural units (Fig. 24).

A substructured articulated rigid-body model of the rhodopsin protein has been developed. This is a fairly large system with the fully atomistic model consisting of about 5000 atoms. This atomistic model is substructured into 26 connected rigid bodies that form an articulated serial chain topology. Similar to what was done with the RuBisCO model, multiple simulations of the rhodopsin model were performed in the NVE ensemble for 1 ns each, with varying timestep sizes. The simulations showed good energy conservation, even for the larger timestep simulations. The total number of degrees of freedom of the system was reduced from about 15 000 to 85. This provided significant computational savings in the calculations of the force interactions as the intra-body interactions were not calculated. Further using this model, the results show a stable simulation with an increase in the timestep size by a factor of 10. The simulations were monitored for energy conservation. Fig. 25 shows the drift in the energy during the simulation. A shorter simulation was run to compare the energy results from the simulations discussed here against the energy results obtained using a model reported in [58]. As can be seen from the two plots, the energy conservation is very good with about 2% change in energy for the 1.5 ns simulation at 10 fs integration timestep. Also, the comparison between the energies of the two models shows very good agreement with relative error being about $10^{-2}$.

## 6. Summary

In the above subsections, results from several test cases of using multibody dynamics algorithms and coarse-grained modelling applied to MD simulations have been discussed. Two different types of coarse-grained models have been pursued, viz models with holonomic distance constraints and models with aggregates of atoms in rigid-body representations. The first model was validated and used for water simulations, simulations of alkanes and simulations of bead-spring polymer systems. In these test cases, a relatively finer scale model is pursued where only the highest frequency content of the system is constrained. Timesteps of several femtoseconds are achieved with good energy conservation and retention of structural properties. The atomistic simulations were stable only with subfemtosecond integration timestep. There are no savings obtained in the force calculations as all the long-range interactions have to be accounted for.

The second type of model is used for modelling and simulating several protein systems. Because the aggregation model is used in these simulations, the system refinement is coarser than the previous models, and the frequency content of the systems is considerably reduced. Consequently stable simulations with preservation of essential dynamics and conservation of energy were achieved with integration timesteps an order of magnitude larger than the required time step for atomistic simulations. The results reported here are for simulations with a conservative 10 fs timestep, an order of magnitude increase in comparison with atomistic simulations. Only 2–3% violation in energy conservation is observed for nanosecond simulations. The conformational statistical results are in agreement with reference studies cited in literature.

While more aggressive timesteps have been used in the studies and can definitely be used to get stable simulations, these results are not shown here. For the results reported here, the bounds on errors and violation of conservation laws are kept tight. However, if an analyst were to accept violation in conservation up to 10%, stable simulations with integration timesteps in the range of 20–40 fs can be achieved. In these models, computational savings are also achieved from reduction in the number of interaction sites for force computations.

Fig. 26 shows a plot summarizing variations in error in energy conservation with increase in integration timesteps as compared with integration step sizes for stable atomistic simulations for the various proteins studied here. As can be observed, a speed up of a factor of 20 can be easily achieved with errors less than 10%. Further, these simulations were run with fixed timestep integration schemes where all degrees of freedom of the system are temporally propagated using the same timestep.

The simulation results were monitored for energy conservation while maintaining a large integration step size. As discussed in the previous subsections, excellent energy conservation characteristics were obtained despite the larger integration step and the use of a non-symplectic integration scheme. In some cases, the thermodynamic and structural properties were calculated and conformational

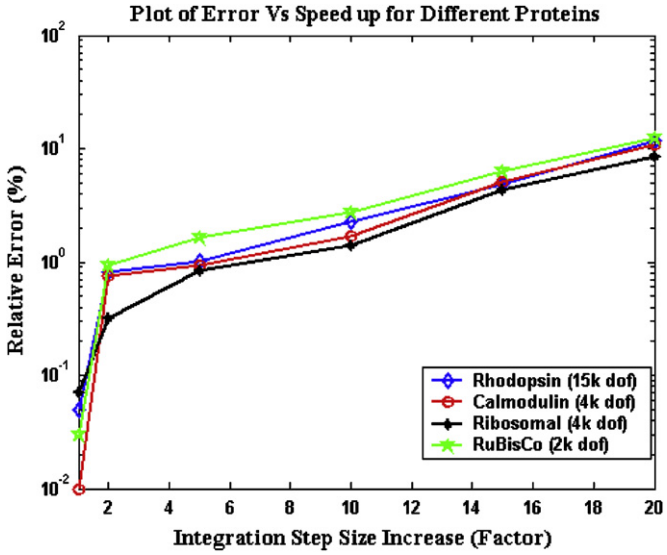



**Fig. 26.** Variations in computational gains with error.

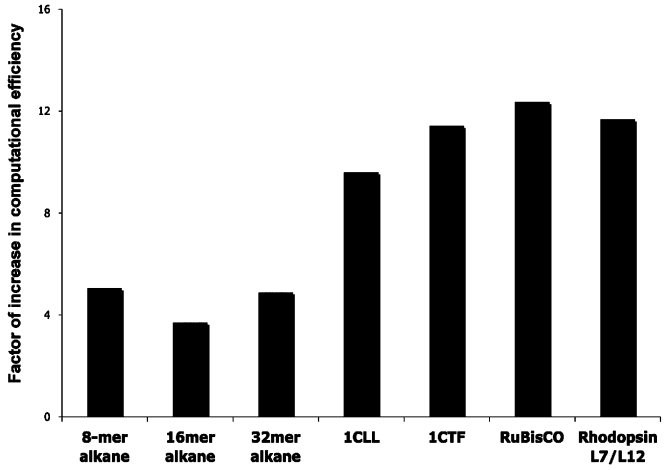



**Fig. 27.** Speedup obtained in simulating various systems studied.

analyses of the proteins were carried out. The results indicate good agreement with the expected essential dynamics of the systems.

The representations of the proteins were developed using VMD [59] and the theoretical values of the B factors for CaM and ribosomal were obtained from [60].

Fig. 27 shows the factor of increase in computational speedup obtained by comparing the average simulation times obtained from the method discussed here and classical MD simulations. The various systems were simulated with the same initial conditions using similar two-step temporal integration schemes. The values depicted in the figure are averaged over three different simulations of 1 ns each.

The timesteps for the different simulations were chosen to result in comparable bounds in energy drift. Consequently, the MD simulations were run with a smaller timestep as compared to the coarse-grained simulations. For example, the rhodopsin simulation was run with a timestep of 1 fs with classical MD, while a timestep of 10 fs was used with the coarse-grained method.

The figure shows the results for two different kinds of systems, coarse-grained alkanes simulation and simulation of proteins. As can be observed, the speed up obtained from the alkanes simulations on a single processor is about a factor of 5. The speedup obtained for the proteins is about a factor of 10. The coarse-grained model of the alkanes is of a finer resolution than the proteins. In the alkanes systems, only the axial vibrations of the atoms were constrained. In the proteins, the constrained peptide plane corresponds to the finest level of refinement. Compared to the alkanes, this is a coarser model and as expected, a larger speedup is obtained in the protein simulations.

The simulation results presented here were obtained for conservative error norms using a simple two-step integration scheme on a single processor machine. It is expected that substantial increase in speedup can be obtained by using MTS algorithms such as the RESPA and r-RESPA [20], particularly when used with the protein simulations where different levels of coarse graining have been used. For example, the loops were simulated with constrained peptide planes, while in the $\alpha$-helices, several residues could potentially be coarse grained into a single rigid body. However, using a fixed timestep integrator, the simulation of the whole system proceeds with a timestep small enough to capture the frequency content of the peptide plane motion. With an MST scheme, it could be possible to propagate different segments of the system at different time scales, resulting in improved simulation times.

## 7. Conclusion

This paper reports a new internal coordinate molecular dynamics research effort. In this research, an efficient $O(n)$ multibody dynamics algorithm is used in conjunction with efficient force field calculation algorithms to simulate coarse-grained molecular dynamics systems. A review of this algorithm is presented and the software developed as a part of this research is also briefly discussed. The results from simulating different materials and biomolecular systems using this method are presented along with the analysis of the simulation results. The systems studied in this work include rigid water, coarse-grained polymer and alkane systems, as well as coarse-grained models of alanine di-peptide and carboxyl terminals of calmodulin, ribosomal, RuBisCo and rhodopsin proteins. The results indicate excellent energy conservation with an integration timestep of an order of magnitude larger than those achievable with atomistic simulations. Results of thermodynamic and configurational analysis indicate the conservation of essential properties of the systems.

## Acknowledgments

This work was supported in part by NSF-NIRT Grant 0303902 and by Sandia LDRD 67017. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000. The authors thank the funding agencies for their support.

## References

[1] M. Deserno, C. Holm, How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines, J. Chem. Phys. 109 (1998) 7678–7693.
[2] M. Deserno, C. Holm, How to mesh up Ewald sums. II. An accurate error estimate for the particle–particle–particle-mesh algorithm, J. Chem. Phys. 109 (1998) 7694–7701.
[3] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[4] D. Reith, M. Pütz, F. Müller-Plathe, Deriving effective mesoscale potentials from atomistic simulation, J. Comput. Chem. 24 (13) (2003) 1624–1636.
[5] S. Ivzekov, G.A. Voth, A multiscale coarse-graining method for biomolecular systems, J. Phys. Chem. B 109 (2005) 2469–2473.
[6] T.D. Hone, S. Ivzekov, G.A. Voth, Fast centroidal molecular dynamics: a force-matching approach for the predetermination of the effective centroidal forces, J. Phys. Chem. 122 (2005) 054105-1–7.
[7] H.C. Anderson, Rattle: a 'velocity' version Shake algorithm for molecular dynamics calculations, J. Comput. Phys. 54 (1983) 24–34.

[8] J.P. Rychaert, G. Cicotti, H.J.C. Berensden, Numerical integration of the Cartesian equations of motion of a system of with constraints: molecular dynamics of *n*-alkanes, J. Comput. Phys. 23 (1977) 327–341.
[9] E. Barth, K. Kuczera, B. Leimkuhler, R.D. Skeel, Algorithms for constrained molecular dynamics, J. Comput. Chem. 16 (1995) 1192–1209.
[10] P.J. Ludovice, U.W. Suter, Molecular dynamics of geometrically constrained polymer systems in generalized coordinates: basic formalism, Comput. Polym. Sci. 1 (1995) 69–79.
[11] R.A.J. Abagyan, A. Mazur, New methodology for computer-aided modelling of biomolecular structure and dynamics, J. Biomolecular Struct. 6 (1989) 833–845.
[12] A.K. Mazur, V.E. Dorofeev, R.A.J. Abagyan, Derivation and testing of explicit equations of motion for polymers described by internal coordinates, J. Comput. Phys. 92 (2) (1991) 261–272.
[13] L.M. Rice, A.T. Brünger, Torsion angle dynamics: reduced variable conformational sampling enhanced crystallographic structure refinement, Proteins Struct. Funct. Genet. 19 (1994) 277–290.
[15] A. Jain, N. Vaidehi, G. Rodriguez, A fast recursive algorithm for molecular dynamics simulation, J. Comput. Phys. 106 (2) (1993) 258–268.
[16] J.D. Turner, P. Weiner, B. Robson, R. Venugopal, H. Schubele III, R. Singh, Reduced variable molecular dynamics, J. Comput. Chem. 16 (1995) 1271–1290.
[17] N. Vaidehi, A. Jain, W.A. Goddard, Constant temperature constrained molecular dynamics: the Newton–Euler inverse matrix operator method, J. Phys. Chem. 100 (1996) 10508–10517.
[18] H.M. Chun, C.E. Padilla, D.N. Chin, M. Watenabe, V.I. Karlov, H.E. Alper, K. Soosaar, K.B. Blair, O.M. Becker, L.S.D. Caves, R. Nagle, D.N. Haney, B.L. Farmer, MBO(N)D: a multibody method for long-time molecular dynamics simulations, J. Comput. Chem. 21 (3) (2000) 159–184.
[19] R.D. Swindoll, J.M. Haile, A multiple time-step method for molecular dynamics simulations of fluids of chains, J. Chem. Phys. 53 (1984) 289–298.
[20] M.E. Tuckerman, B.J. Berne, Molecular dynamics in systems with multiple time-scales: systems with stiff and soft degrees of freedom and with short and long range forces, J. Comput. Chem. 95 (1992) 8362–8364.
[21] J.L. Scully, J. Hermans, Multiple timesteps: limits on the speedup of molecular dynamics simulation of aqueous systems, Mol. Simulation 11 (1993) 67–77.
[22] M. Watanabe, M. Karplus, Dynamics of molecules with internal degrees of freedom by multiple time-step methods, J. Chem. Phys. 99 (1993) 8063–8074.
[23] M. Watanabe, M. Karplus, Simulations of macromolecules by multiple time-step methods, J. Phys. Chem. 103 (1995) 3718–3732.
[24] T. Forester, T. Smith, On multiple time-step methods and the Ewald sum, Mol. Simulation 13 (1994) 195–204.
[25] W.B. Streett, D.J. Tildesley, G. Saville, Multiple timestep methods in molecular dynamics, Mol. Phys. 35 (1978) 639–648.
[26] G. Zhang, T. Schlick, LIN: a new algorithm combining implicit integration and normal mode techniques for molecular dynamics, J. Comput. Chem. 14 (1993) 1212–1233.
[27] K. Hinsen, Analysis of domain motions by approximate normal mode calculations, Proteins Struct. Funct. Genet. 33 (1998) 417–429.
[28] D. Janežič, M. Prapotnik, Symplectic molecular dynamics integration using normal mode analysis, Int. J. Quantum Chem. 84 (2001) 2–12.
[29] F. Tama, O. Miyashita, C.L. Brooks, Normal mode based flexible fitting of high-resolution structure in low resolution experimental data from cryo-EM, J. Struct. Biol. 147 (2004) 315–326.
[30] M. Kaledin, A.L. Kaledin, A. Brown, J.M. Brown, Driven molecular dynamics for normal modes of biomolecules with the Hessian and beyond, Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems, Chapman and Hall/CRC Mathematical and Computational Biology Series, 2006.
[31] P. Petrone, V.S. Pande, Can conformational change be described by only a few normal modes, Biophys. J. 90 (2006) 1583–1593.
[32] C.S. Peskin, T. Schlick, Molecular dynamics by the backward Euler's method, Commun. Pure Appl. Math. 42 (1989) 1001–1031.
[33] G. Zhang, T. Schlick, The Langevin Implicit-Euler Normal-mode scheme for molecular dynamics at large timesteps, J. Chem. Phys. 101 (1994) 4995–5012.
[34] G. Zhang, T. Schlick, Implicit discretization schemes for Langevin dynamics, Mol. Phys. 84 (1995) 1077–1098.
[35] ⟨http://lammps.sandia.gov/⟩.
[36] K. Anderson, R. Mukherjee, J. Critchley, J.L. Ziegler, S. Lipton, Poems: paralleizable open-source efficient multibody software, Eng. Comput. 23 (1) (2007) 11–23.
[37] T.R. Kane, D.A. Levinson, Dynamics: Theory and Application, McGraw-Hill, New York, 1985.
[38] R. Featherstone, The calculation of robotic dynamics using articulated body inertias, Int. J. Robotics Res. 2 (1) (1983) 13–30.
[39] D.S. Bae, E.J. Haug, A recursive formation for constrained mechanical system dynamics: Part I, open loop systems, Mech. Struct. Mach. 15 (3) (1987) 359–382.
[40] D. Rosenthal, An order *n* formulation for robotic systems, J. Astron. Sci. 38 (4) (1990) 511–529.
[41] K.S. Anderson, An order-*n* formulation for motion simulation of general constrained multi-rigid-body systems, Comput. Struct. 43 (3) (1992) 565–572.
[42] A. Jain, Unified formulation of dynamics for serial rigid multibody systems, J. Guidance Control Dyn. 14 (3) (1991) 531–542.
[43] K.S. Anderson, An order-*n* formulation for the motion simulation of general multi-rigid-body tree systems, Comput. Struct. 46 (3) (1993) 547–559.
[44] A.R. Leach, Molecular Modelling Principles and Applications, Prentice-Hall, Englewood Cliffs, NJ, 2001.
[45] R.M. Mukherjee, K.S. Anderson, Efficient methodology for multibody simulations with discontinuous changes in system definition, Multibody Syst. Dyn. 18 (2) (2007) 145–168.
[46] H. Chun, C. Padilla, D. Chin, M. Watenabe, H. Alper, K. Soosaar, K. Blair, O. Becker, L. Caves, R. Nagle, D. Haney, B. Farmer, MBO(N)D: a multibody method for long-time molecular dynamics simulations, J. Comput. Chem. 21 (3) (2000) 159–184.
[47] J. Norberg, L. Nilsson, Advances in biomolecular simulations: methodology and recent applications, Q. Rev. Biophys. 36 (3) (2003) 257–306.
[48] D. Wolf, W.G. Rudd, Tabulated potentials in molecular dynamics simulations, Comput. Phys. Commun. 120 (1999) 200–232.
[49] E. von Meerwalla, S. Beckman, Diffusion of liquid *n*-alkanes: free-volume and density effects, J. Chem. Phys. 108 (10) (1998) 4299–4304.
[50] J. Elezgaray, G. Marcou, Y.H. Sanejouand, Exploring the natural conformational changes of the C-terminal domain of calmodulin, Phys. Rev. E 66 (031908) (2002) 1–8.
[51] C. Yang, K. Kuczera, Molecular dynamics simulations of calcium-free calmodulin in solution, J. Biomolecular Struct. Dyn. 19 (2002) 801–819.
[52] W. Wriggers, E. Mehler, F. Pitici, H. Weinstein, K. Schulten, Structure and dynamics of calmodulin in solution, Biophys. J. 74 (1994) 1622–1639.
[53] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, J. Comput. Chem. 4 (1983) 187–217.
[54] D. Kosztin, S. Izraileve, K. Schulten, Unbinding of retinoic acid from its receptor studied by steered molecular dynamics, Biophys. J. 76 (1999) 188–197.
[55] C.M. Shepherd, H.J. Vogel, A molecular dynamics study of $Ca^{2+}$-calmodulin: evidence of interdomain coupling and structural collapse on the nanosecond timescale, Biophys. J. 87 (2004) 780–791.
[56] Y.-H. Sanejouand, O. Tapiat, Low-frequency domain motions in the carboxy terminal fragment of L7/L12 ribosomal protein studied with molecular dynamics techniques: Are these movements model independent?, J. Phys. Chem. 99 (1995) 5698–5704.
[57] T.X. Hoang, F. Seno, J.R. Banavar, M. Cieplak, A. Maritan, Assembly of protein tertiary structures from secondary structures using optimized potentials, Proteins Struct. Funct. Genet. 52 (2003) 155–165.
[58] P.S. Crozier, M.J. Stevens, L.R. Forrest, T.B. Woolf, Molecular dynamics simulation of dark-adapted rhodopsin in an explicit membrane bilayer: coupling between local retinal and larger scale conformational changes, J. Mol. Biol. 333 (2003) 493–514.
[59] W. Humphrey, A. Dalke, K. Schulten, VMD—visual molecular dynamics, J. Mol. Graphics 14 (1996) 33–38.
[60] ⟨http://ignm.ccbb.pitt.edu⟩.
[61] B.R. Brooks, D.B. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, J. Chem. 16 (1997) 599–610.
[62] M. Karplus, CHARMM (chemistry at Harvard macromolecular mechanics) ⟨http://brooks.scripps.edu/⟩, brooks Group Computational BioPhysics and Chemistry, 2005.
[63] D.A. Pearlman, D.A. Case, J.W. Caldwell, W.R. Ross, T.E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules, Comput. Phys. Commun. 91 (1995) 1–41.
[64] R. Featherstone, A divide-and-conquer articulated body algorithm for parallel $O(\log(n))$ calculation of rigid body dynamics. Part 1: basic algorithm, Int. J. Robotics Res. 18 (9) (1999) 867–875.