

Substructured Molecular Dynamics using Multibody Dynamics Algorithms

Rudranarayan M. Mukherjee ^{a,1}

^a*Rensselaer Polytechnic Institute, 110 8th Street, Troy NY 12180 U.S.A.*

Paul S. Crozier ^b

^b*Sandia National Laboratories, Albuquerque, New Mexico U.S.A.*

Steven J. Plimpton ^c

^c*Sandia National Laboratories, Albuquerque, New Mexico U.S.A.*

Kurt S. Anderson ^d

^d*Rensselaer Polytechnic Institute, 110 8th Street, Troy NY 12180 U.S.A.*

Abstract

This paper presents results obtained from simulating several biomolecular and bulk materials using multibody dynamics algorithms. The systems studied include bulk water, alkane chains, alanine dipeptide and carboxyl terminal fragments of calmodulin, ribosomal protein, and rhodopsin. The atomistic representations of these systems include several thousand degrees of freedom and the results of nano-second long simulations of several of these systems are presented. 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 time step integration scheme. A discussion is presented on the open-source software developed to facilitate future research using multibody dynamics with molecular dynamics.

Key words: Coarse grained molecular dynamics, Multibody dynamics algorithms, Open source software

1. Introduction

Simulations of the dynamics of biomolecular and polymeric systems are usually characterized by very complex energy surfaces and large conformational spaces. The simulation of the fine scale model of a molecular system involves the calculation of the in-

teraction forces between different entities in the system, direct application of Newton's second law of motion to calculate the state derivatives, and the temporal integration of the state derivatives to temporally advance the simulation. Though conceptually a simple process, the computational aspect of this simple process is daunting. There are two fundamental computational challenges which have limited the scope of molecular dynamics 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

Email addresses:

Rudranarayan.M.Mukherjee@jpl.nasa.gov (Rudranarayan M. Mukherjee), pscrozi@sandia.gov (Paul S. Crozier), sjplimp@sandia.gov (Steven J. Plimpton), anderk5@rpi.edu (Kurt S. Anderson).

¹ Now at NASA Jet Propulsion Laboratory, Pasadena CA

that are of interest typically contain large number of interaction sites (or atoms). Further, these systems are typically suspended in some form of aqueous solution and if an explicit water model is used, the number of water molecules further increase the number of degrees of freedom n . Thus the spatial size of these systems can span hundreds to many thousand degrees of freedom i.e. $n = 10^3 \sim 10^6$.

The interactions between these sites include local stiff interactions such as axial and angular spring forces, torsional forces, and dihedral interactions, as well as long-range Coulombic and Lennard Jones (LJ) interactions. These long-range interactions are inter-atomic and hence the calculation is a many body problem which traditionally scales as $O(n^2)$. Methods such as the Ewald summation [1], the Particle-Particle Particle-Mesh (P^3M) [2] and the Fast Multi-pole Method (FMM) [3] have been proposed that reduce the cost of this calculation. Although these methods reduce the cost of 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. It is manifest in the problems associated with the temporal integration of the equations of motion. The state of the art *long* simulation ranges from nano-seconds to micro-seconds of real time. However many phenomena of interest occur in longer temporal scales ranging from micro to several seconds. For example, protein folding and polymer entanglement are typically associated with the micro-seconds to seconds domain, while much longer phenomena such as glass relaxation are found in the seconds to days domain. However, a molecular dynamics simulation of a fine scale system typically propagates with an integration time step of sub-femto seconds. This is because the integration step size has to be inversely proportional to the highest frequency content of the system to obtain a stable simulation. In the biomolecular and polymeric systems, the highest frequency is usually associated with carbon-hydrogen oscillations limiting the time step to 0.5 – 1 femto-seconds and the Debye frequency which limits the time step to about 2 femtoseconds. For only a 1 nanosecond simulation, the fine scale atomistic simulation requires $\sim 10^6$ integration steps, with each integration step expending at worst $O(n^2)$ calculations for the system wide in-

teraction force calculations. This cost quickly becomes computationally prohibitive.

Over the years a large volume of work has been focused on methods for reducing molecular dynamics simulation turn around times, which can be categorized into two broad classes. The first class of methods are those that aim at reducing the complexity of calculating the 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] This approach eliminates the high frequency components of the system by eliminating the degrees of freedom which produce this high frequency behavior. This is accomplished by augmenting the original set of unconstrained equations of motion with a companion set of constraint equations. These explicit constraints are then enforced by solving the resulting systems of **Differential-Algebraic Equations (DAEs)**(differential equations of motion and algebraic constraint equations) simultaneously. This approach is represented by the SHAKE algorithm [7] which introduces constraints into the equations of motion using the formalism of Lagrange multipliers, with similar approaches used by others [8]-[10].

Constrained Dynamics using Implicit Constraints: [11]-[13] This family of approaches satisfies the constraints exactly and implicitly through the use of generalized coordinates. With this type of formulation, the constraints are enforced directly in the formation of the system equations of motion. A more restrictive example of this is termed “torsional dynamics” in which the bond lengths are “frozen” along with relative bond angles, permitting only the dihedral (torsional) angles to change. The dynamic simulations are then performed by solving and integrating the now heavily coupled equations of motion in the generalized coordinates [12]-[14]. The solution of the heavily coupled equations of motion can be computationally expensive unless intelligent, efficient schemes are employed.

Reduced Computational Order Algorithms: [15]-[18] These methods may be considered a special-

ized subset of the more generic *Generalized Coordinate* methods mentioned immediately above. These reduced computational order algorithms 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. Such methods include the so-called $O(n)$ methods, for which the overall computational cost per time step increases linearly with the number of generalized coordinates n .

Multirate Temporal Integration: [18]-[24] Multirate integration methods attempt to exploit the multiscale character of biopolymer motions. In what is known as multiple time-scale (MTS) schemes [25] 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 [20]. By applying MTS methods to a variety of systems, including the protein BPTI, using a splitting scheme based on bonding topology, [22][23] report significant speed-ups.

Eigenvector(Modal)-Schemes: [26]-[31] Significant reductions in model frequency content may be realized through the use of modal (usually eigenvector) based schemes. Eigenvector-based schemes use normal modes (free vibration modes), which are obtained from diagonalization of the Hessian matrix (second derivative of the potential). These modes are then used to describe global deformations around a local minimum on the molecular potential energy surface. This type of approach offers the advantage of directly eliminating the high frequency components of the system, while retaining the lower frequency modes which are important for describing the system conformational motions. As such these methods offer the potential for very large increases in the size of the temporal integration step which may be taken.

Implicit integration schemes: [32]-[34] Implicit integration schemes are interpolative in nature, and thus may take much larger stable time steps than explicit integration schemes which are extrapolative. Implicit integration is inherently more stable, and expensive than explicit integration methods. For all but the most elementary systems, true implicit integration yields a set of coupled algebraic equations

which must be solved at each integration step. However implicit integration methods are particularly problematic when applied to MD systems in which generalized coordinate type reduction schemes have been used. Even in the situation where low order $O(n)$ approaches have been used to solve for the state derivatives in what outwardly appears to be an uncoupled manner, using these state derivatives with a implicit integration scheme yields a system of fully coupled highly nonlinear equations. The cost of solving this nonlinear system of equations is sufficiently great as to more than offset any gains which might be realized through the larger integration step size. Further, the damping introduced by the implicit schemes may pose a problem while maintaining the micro-canonical conditions of the ensembles.

In this paper, a methodology for internal coordinate molecular dynamics is presented. This method uses efficient multibody dynamics algorithms for solving the equations of motion of the constrained molecular dynamics system in linear complexity. The method 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 Coriolis 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. The use of efficient multibody dynamic algorithms coupled with efficient and highly parallelizable force field calculations results in a highly efficient computation method for constrained molecular dynamics. 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 [35] and the efficient multibody software POEMS [36]. In the following section a brief overview of the multibody dynamics algorithm is presented. The implementation details are presented next, followed by the results obtained by using the method on different molecular systems in-

clude materials and biomolecular systems. Finally a discussion is presented on the software that was developed in this research effort, and is distributed freely in the public domain.

2. Algorithm Overview

In this section an overview of the recursive $O(n)$ algorithm is presented. This algorithm 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 have been in use for a number of years including [38]-[42] among others. 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. Each body is associated with its own body fixed dextral set of unit vectors. The joint locations, inertia values and the generalized velocities are expressed with respect to the body fixed reference frames. One end of the chain (or tree) is chosen as the base body and it is connected to the inertial reference frame by a kinematic joint. For a chain topology, each successive body is connected by two kinematic joints, one to an inward (parent) body and the other to the outward body (child) on the chain. The orientation and motion of a body is expressed in terms of the generalized coordinates associated with the admissible degrees of freedom of the joints. The linear and angular velocities of a body are expressed as invertible linear combinations of the time derivatives of the generalized coordinates (referred to as generalized speeds).

The algorithm works in three recursive sweeps or traversals of the system topology. The first sweep begins at the base body and moves outwards to the tip recursively generating the kinematic preliminaries such as partial velocities [37], inertias and applied forces from the known states of the system. The second traversal then begins at the tip (terminal body) and recursively moves inwards towards the base body. This is a triangulation traversal and it recursively generates the articulated composite in-

ertias and forces. This traversal is equivalent to successively shifting the inertias and the forces inwards towards the base body. Since the boundary conditions are known at the base body, at the end of the triangulation sweep the equations of motion for the base body can be solved to generate the state derivatives of the base body. Starting at the base body the third traversal works outward recursively solving for the state derivatives of each successive body. By the time this sweep ends at the tip, the state derivatives are all solved with an efficient $O(n)$ complexity.

2.1. Mathematical Preliminaries

To aid in the subsequent discussion, consider the notation associated with the description of an arbitrary set of interconnected rigid bodies shown in figure (1). For this system, proximal (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. Similarly, the set of bodies in the system topology which are downstream from body k (away from the base) are referred to as the *distal* (child) bodies of body k and are represented as members of the set of bodies $Dist[k]$. As shown in figure (1), the position vector s^k locates joint- k relative to the mass center of body $Pr[k]$, while the position vector r^k locates the mass center of body k with respect to the outward end of this same joint. It will also prove convenient to describe the position of child mass center k^* relative to proximal mass center $Pr[k]^*$ by the vector γ^k .

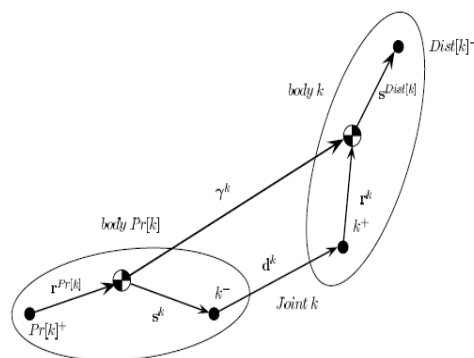


Fig. 1. Notation Associated with an Arbitrary Set of Interconnected Rigid Bodies

The angular velocity of any body k with respect to the Newtonian reference frame N , and velocity of

its associated mass center k^* may always be written in terms of the generalized speeds as

$$\boldsymbol{\omega}^k = \sum_{r=1}^n \boldsymbol{\omega}_r^k u_r + \boldsymbol{\omega}_t^k \quad (1)$$

and

$$\mathbf{v}^{k^*} = \sum_{r=1}^n \mathbf{v}_r^{k^*} u_r + \mathbf{v}_t^{k^*}. \quad (2)$$

In these expressions $\boldsymbol{\omega}_r^k$ and $\mathbf{v}_r^{k^*}$ are termed the r^{th} *partial angular velocity of body k* and r^{th} *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 $\mathbf{v}_t^{k^*}$ appearing in equations (1)–(2), 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.

When presenting this method it is convenient to express quantities in a scalar matrix, as opposed to a tensor (vector and dyadic) form. For this purpose an arbitrary vector $\boldsymbol{\vartheta}^k$ will be represented in matrix form as $\underline{\boldsymbol{\vartheta}}^k$, which is associated with the local dextral orthogonal unit vectors $\hat{k}_1, \hat{k}_2, \hat{k}_3$, fixed in body k . One may then define the *velocity*, *partial velocity*, and *velocity remainder term* matrices as

$$\underline{\boldsymbol{\nu}}^k = \begin{bmatrix} \underline{\boldsymbol{\omega}}^k \\ \underline{\mathbf{v}}^{k^*} \end{bmatrix}, \quad \underline{\mathcal{P}}_r^k = \begin{bmatrix} \underline{\boldsymbol{\omega}}_r^k \\ \underline{\mathbf{v}}_r^{k^*} \end{bmatrix}, \quad \text{and} \quad \underline{\boldsymbol{\nu}}_t^k = \begin{bmatrix} \underline{\boldsymbol{\omega}}_t^k \\ \underline{\mathbf{v}}_t^{k^*} \end{bmatrix}. \quad (3)$$

With these matrices so defined, equations (1) and (2) may be expressed as

$$\underline{\boldsymbol{\nu}}^k = \bar{\underline{\boldsymbol{\nu}}}^k + \underline{\boldsymbol{\nu}}_t^k = \sum_{r=1}^n \underline{\mathcal{P}}_r^k u_r + \underline{\boldsymbol{\nu}}_t^k. \quad (4)$$

One can similarly represent the generalized acceleration matrix of an arbitrary body k as defined in previous works [41][43], as

$$\underline{\mathcal{A}}^k = \begin{bmatrix} N \underline{\boldsymbol{\alpha}}^k \\ N \underline{\boldsymbol{a}}^{k^*} \end{bmatrix}, \quad (5)$$

where $\underline{\mathcal{A}}^k$ may also be divided into two portions. One is $\bar{\underline{\mathcal{A}}}^k$, which contains all terms which are explicit

in the unknown state derivatives $\dot{\underline{\boldsymbol{\nu}}}$ and the other is the *acceleration remainder term* $\underline{\mathcal{A}}_t^k$, which represents all of the other acceleration terms (gyroscopic, Coriolis and/or prescribed) that may be calculated directly from the system state. Thus

$$\underline{\mathcal{A}}^k = \bar{\underline{\mathcal{A}}}^k + \underline{\mathcal{A}}_t^k. \quad (6)$$

3. $O(n)$ Forward Dynamics Analysis

As mentioned previously, the basic recursive $O(n)$ algorithm for performing forward dynamics simulation consists for three principle steps, or “sweeps”. These steps are the *Kinematics Sweep*, the *Triangularization Sweep*, and the *Back Substitution Sweep*.

3.1. Kinematic 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

$$\underline{\boldsymbol{\nu}}^k = [(\underline{\mathcal{S}}^k)^T \bar{\underline{\boldsymbol{\nu}}}^{Pr[k]} + \underline{\mathcal{P}}_k^k \underline{\boldsymbol{u}}_k] + \underline{\boldsymbol{\nu}}_t^k, \quad (7)$$

and

$$\underline{\mathcal{A}}^k = [(\underline{\mathcal{S}}^k)^T \bar{\underline{\mathcal{A}}}^{Pr[k]} + \underline{\mathcal{P}}_k^k \underline{\dot{\boldsymbol{u}}}_k] + \underline{\mathcal{A}}_t^k. \quad (8)$$

The quantity $\underline{\mathcal{S}}^k$ appearing in equations (7)–(8) is the basis consistent linear transformation matrix

$$\underline{\mathcal{S}}^k = \begin{bmatrix} \underline{\mathcal{C}}^k & \underline{\mathcal{C}}^k \underline{\boldsymbol{\gamma}}_{\times}^k \\ \underline{\mathbf{0}} & \underline{\mathcal{C}}^k \end{bmatrix}_{6 \times 6}. \quad (9)$$

Within this expression $\underline{\mathcal{C}}^k \equiv {}^{Pr[k]} \underline{\mathcal{C}}^k$ is the direction cosine matrix which relates the body k basis vectors to those fixed in its parent body $Pr[k]$; $\underline{\mathbf{0}}$ is a 3×3 zero matrix; and $\underline{\boldsymbol{\gamma}}_{\times}^k$ is the skew symmetric matrix equivalent to the vector cross product operation $\boldsymbol{\gamma}^k \times$. The *shift matrix* transformation $\underline{\mathcal{S}}^k$ converts a system of forces and moments acting through the center of mass of k , to an equivalent force system, acting though a point of k which is instantaneously coincident with the center of mass of $Pr[k]$.

At this time, it is also convenient to define the body k *generalized inertia* $\underline{\mathcal{I}}^k$ and the body k *generalized force* $\underline{\mathcal{F}}^k$ matrices

$$\underline{\mathcal{I}}^k = \begin{bmatrix} \underline{\mathcal{I}}^{k/k^*} & \underline{0} \\ \underline{0} & \underline{M}^k \end{bmatrix}_{6 \times 6}, \quad (10)$$

$$\underline{\mathcal{F}}^k = \begin{bmatrix} \underline{T}^k - (\underline{\mathcal{I}}^{k/k^*} \underline{\alpha}_t^k + \underline{\omega}_\times^k \underline{\mathcal{I}}^{k/k^*} \underline{\omega}^k) \\ \underline{R}^k - \underline{M}^k \underline{a}_t^k \end{bmatrix}_{6 \times 1}. \quad (11)$$

Within these expressions, $\underline{\mathcal{I}}^{k/k^*}$ is the 3×3 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^* .

3.2. Triangularization Sweep

The triangularization procedure works recursively inward (tip to base) to compute the articulated inertia mass matrix $\underline{\mathcal{I}}_3^k$ and the articulated generalized active force $\underline{\mathcal{F}}_3^k$ expressed as

$$\underline{\tilde{\mathcal{I}}}^k = \underline{\mathcal{I}}_1^k + \sum_{j \in \text{Dist}[k]} \underline{\tilde{\mathcal{T}}}^j \underline{\mathcal{I}}_3^j (\underline{\mathcal{S}}^j)^T \quad (12)$$

$$\underline{\tilde{\mathcal{F}}}^k = \underline{\mathcal{F}}_1^k + \sum_{j \in \text{Dist}[k]} \underline{\tilde{\mathcal{T}}}^j \underline{\mathcal{F}}_3^j \quad (13)$$

where $\text{Dist}[k]$ is the *distal* (children) set associated with body k . The triangularization operator $\underline{\mathcal{T}}^j$ and the basis consistent shifting operator $\underline{\mathcal{S}}^j$ used in (12)-(13) are defined as

$$\underline{\mathcal{T}}^j = (\underline{\mathcal{S}}^j)^T \left[\underline{U} - \frac{1}{\mathcal{M}_j} \underline{\tilde{\mathcal{I}}}^j \underline{\mathcal{P}}_j^j (\underline{\mathcal{P}}_j^j)^T \right], \quad (14)$$

$$\underline{\mathcal{S}}^j = \begin{bmatrix} \underline{\mathcal{C}}^j & \underline{0} \\ \underline{0} & \underline{\mathcal{C}}^j \end{bmatrix} \begin{bmatrix} \underline{U} & \underline{\gamma}_\times^j \\ \underline{0} & \underline{U} \end{bmatrix}. \quad (15)$$

The quantity \mathcal{M}_k is also given by

$$\mathcal{M}_k = (\underline{\mathcal{P}}_k^k)^T \underline{\mathcal{I}}_3^k \underline{\mathcal{P}}_k^k, \quad (16)$$

3.3. 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 $\mathcal{M}_1 \dot{\mathbf{u}}_1 = \underline{\mathcal{P}}_1^1 \underline{\tilde{\mathcal{F}}}^1$ can be isolated and yields the solution of $\dot{\mathbf{u}}_1$. The solution for $\dot{\mathbf{u}}_1$ is then substituted into the next equation to solve for $\dot{\mathbf{u}}_2$. Proceeding in this manner, a generalized function expression for the solution of each generalized acceleration $\dot{\mathbf{u}}_k$ is given as follows

$$\dot{\mathbf{u}}_k = \frac{(\underline{\mathcal{P}}_k^k)^T}{\mathcal{M}_k} \left[\underline{\tilde{\mathcal{F}}}^k - \underline{\tilde{\mathcal{I}}}^k (\underline{\mathcal{S}}^k)^T \underline{\mathcal{A}}^{k-1} \right], \quad (17)$$

with $\underline{\mathcal{A}}^k$ computed from

$$\underline{\mathcal{A}}^k = (\underline{\mathcal{S}}^k)^T \underline{\mathcal{A}}^{k-1} + \underline{\mathcal{P}}_k^k \dot{\mathbf{u}}_k. \quad (18)$$

Thus, with the recursive inverse sweep using the above equation (18), 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. Implementation Details

In this section, the simulation protocols and their importance are discussed briefly. Boundary conditions and the implementation method are discussed first followed by discussions on the nature of the kinematic models pursued in these simulations. Temporal integration schemes and the generation of initial states are explained. These topics, by themselves, merit elaborate discussion on validity and efficacy which are beyond the scope of this paper. Hence, only a brief discussion is presented for completeness.

4.1. Boundary Conditions

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. The position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Implementing the periodic boundary conditions as is traditionally done for atomistic systems would require breaking the chains when they leave the simulation box and introducing the images from the

other side. Since this model uses an articulated body formulation based on relative coordinates, breaking a chain would require moving between different sets of generalized coordinates and impose unnecessary computational complexity. However, maintaining the system description in generalized coordinates and generating an extra array of Cartesian coordinates helps to overcome this issue. The dynamics of the system are tracked using relative coordinates without any change to the system description. The Cartesian coordinates are generated at little computational expense during the simulation. These Cartesian coordinates reflect the true positions of the system entities and not their images and may reflect positions outside the simulation box. If forces are calculated based on these true positions, they will not reflect the periodic boundary conditions. Consequently, the Cartesian coordinates are modified using simple geometric formulae [44] to generate the positions of the images which are inside the 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. In this way the articulated body description of the system is never altered and the periodic boundary conditions are imposed with very little computational or memory expense.

4.2. Kinematic Model

There are two different representations of the reduced order models, both of which are in articulated multibody form. 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 traditionally used SHAKE or RATTLE formulations require additional nonlinear equations to be solved to impose the constraints. Further, they impose the constraints iteratively to only within a specified tolerance. The number of iterations can be large if tight tolerances are imposed for constraint satisfaction, resulting in increased computational cost. 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.

4.3. Implicit solvents

Solvents are a critical component of simulations of biomolecular system. Explicit solvent models are accurate, but are computationally prohibitive. Consequently implicit solvent models have been developed, which typically consist of the removal of solvent and the modification of the remaining interactions to account for the solvent's effects. Examples of implicit water models range from simple distance-dependent dielectric implicit solvent models to more sophisticated and accurate models such as Solvent Accessible Surface Area model and the Generalized Born model. The method discussed here does not restrict the choice of solvent model and both explicit water models and implicit models may be used. The implicit solvent models introduce an augmented state-dependent potential and hence are uncoupled from the generation and solution of dynamic equations of motion. Explicit water models can also be used, where each water molecule is modelled as a single free floating rigid body.

4.4. *Generating initial properties*

Initial atomic positions for biomolecular simulations are typically obtained via the protein data base (www.pdb.org). Bulk material initial coordinates for small molecules can be generated on a regular lattice and subsequently melted and adequately equilibrated. Initial atomic velocities can be randomly generated from the Gaussian distribution for the desired simulation temperature.

The calculation of the total mass, position of the center of mass and the velocity of the center of mass of any rigid body is a simple process and can be obtained 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.

The above method does not take into account the effect of kinematic coupling between the bodies. The articulated model is obtained by simultaneously and instantaneously imposing a number of kinematic constraints on an initially unconstrained system. Each kinematic joint instantaneously constrains the relative translation between two consecutive bodies while allowing for relative rotation between the bodies. This instantaneous change in the system degrees of freedom introduces impulsive loads at each joint resulting in discontinuous jumps in the remaining generalized speeds. To reduce the model to an articulated system of bodies connected by kinematic joints, an impulse-momentum method [45] is called upon. The computational complexity of this method scales linearly and logarithmically with the number of generalized coordinates in the system in serial and parallel implementation respectively. This method 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.

4.5. *Time Integration*

The suitability of an integration scheme for molecular dynamics problems is based on two issues viz. (i) how many force calculations are required per integration step and (ii) how efficiently it can maintain energy conservation while taking as large a time step as possible.

Velocity-Verlet temporal integration scheme has

been extensively used for atomistic simulations. The Velocity-Verlet scheme requires only one force calculation per integration step. It is symplectic and gives very good energy conservation characteristics albeit at modest integration step sizes. However for reduced order models involving coupled bodies, the performance of the Velocity-Verlet is not as good as with atomistic simulations. This is because the velocity dependent inertial forces such as the gyroscopic and Coriolis forces come into play for the reduced order models. Also, as compared with an atomistic simulation, the integrands are not the Cartesian accelerations and velocities but the time derivatives of the generalized coordinates.

For these reasons, the multibody dynamics equations of motions have been traditionally integrated by higher order methods such as the Runge-Kutta 4-5 schemes. For macroscopic problems, such as aerospace applications, these schemes are typically highly accurate and give good energy conservation. However when applied to reduced order molecular dynamics simulations, these methods quickly become computationally expensive as they require four force calculations per integration step.

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 time step with good energy conservation over long durations of simulation. It does not include any excessive data storage problems. 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 (19) is the half step iterative velocity propagation step with initial guess (20), while (21) and (22) are the full step position and velocity propagation.

$$v(t + \Delta t/2) = v(t) + \Delta t/2 * a[x(t), v(t + \Delta t/2)] \quad (19)$$

$$v^0(t + \Delta t/2) = v(t) + \Delta t * a[x(t), v(t)] \quad (20)$$

$$x(t + \Delta t) = x(t) + \Delta t * v[t + \Delta t/2] \quad (21)$$

$$v(t + \Delta t) = v(t + \Delta t/2) + \Delta t/2 * a[x(t + \Delta t), v(t + \Delta t/2)] \quad (22)$$

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

4.6. Interaction Forces

The use of appropriate force fields for biomolecular systems is essential for generating valid simulation results. As mentioned earlier, 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. When an articulated rigid body model is used, 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 can be eliminated and do not need to be accounted for in the model.

The non-bonded interactions include Coulombic and Lennard Jones forces. These forces need to be carefully accounted for if the conformational motion of the system is to be accurately simulated. While using a rigid body model, the sum of all non-bonded forces arising between interaction sites within a component body will go to zero because of equal and opposite actions. However, the non-bonded forces between the interaction sites in different bodies need to be accurately calculated.

In the calculation of non-bonded forces, there arise two different approaches. 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 is still considerable savings as interactions between sites on the same body need not be 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 have been 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].

The second approach is based on powerful, yet simple, approaches for deriving coarse-grained molecular dynamics (CGMD) force fields from atomistic force fields [4]-[6]. The most exciting among these is the multi-scale 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 or methods.

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 microcanonical (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.

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

Method	T (K)	P (atm)	KE (kmol)	PE (kmol)
SHAKE	307 ± 1	215 ± 104	2501 ± 11	13246 ± 11
RIGID	306 ± 1	31 ± 185	2493 ± 9	13239 ± 9
POEMS	307 ± 1	137 ± 128	2503 ± 21	13248 ± 20

Table 1

Thermodynamic properties of box of rigid TIP3P water molecules. Note: T (temperature), P (pressure), KE (kinetic energy), PE (potential energy). SHAKE, RIGID and POEMS correspond to different methods in LAMMPS.

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.

5.2. Alanine Dipeptide

The next system that has been tested is alanine dipeptide as shown in figure (2). 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 con-

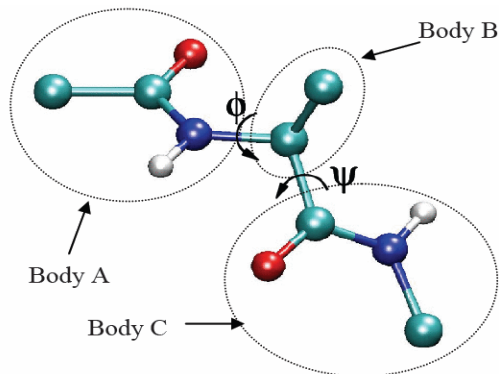


Fig. 2. Alanine dipeptide molecule with two 4-atom peptidic groups modelled as rigid bodies as shown. Carbon is shown in cyan, Nitrogen in blue, Hydrogen in white, and Oxygen in red

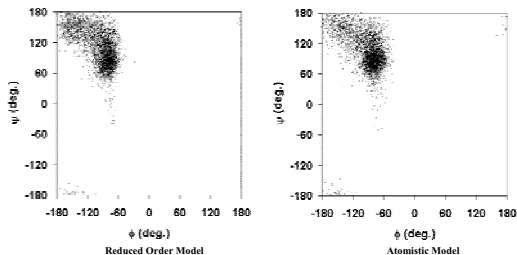


Fig. 3. Ramachandran plots from alanine dipeptide simulations

formational 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 time step.

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 figure (3). 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 ps^{-1} 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

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	74255 ± 206	5338 ± 150	2087 ± 75

Table 2

Thermodynamic properties of box of alkanes. Note: T (temperature), P (pressure), KE (kinetic energy), PE (potential energy)

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 a 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 fs, 2 fs, and 3 fs) were used to run one nanosecond 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 time steps by a factor of 4-6.

Table (2) shows the thermodynamic properties of the systems calculated at different time steps. The thermodynamic properties were calculated at every 200 femto-seconds and the average and standard deviations in temperature, kinetic, potential and total energies are shown. 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 to-

tal energy. The energy conservation is much better at the lower time step, and the variations increase with higher time steps. With a 3 femto-second time step, the variation in energy is only about 2% for the 1 nano-second simulation. This variation is an acceptable variation [44].

Figure (4) 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 time steps 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].

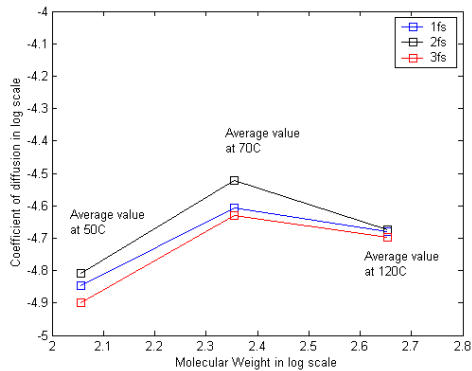


Fig. 4. Diffusion coefficients from long time-scale alkane simulations as computed using the Einstein formula

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 interatomic 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.

Figure (5) shows a plot of kinetic, potential and total energy of the system over long temporal scales

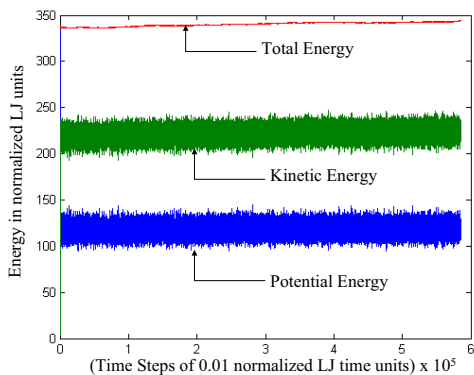


Fig. 5. Energy Conservation over Long Temporal Scales

for a constant energy ensemble. In the figure, a small 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 percent. By eliminating the high frequency axial vibration of the beads, an increase in the time step by *an order of magnitude* is obtained.

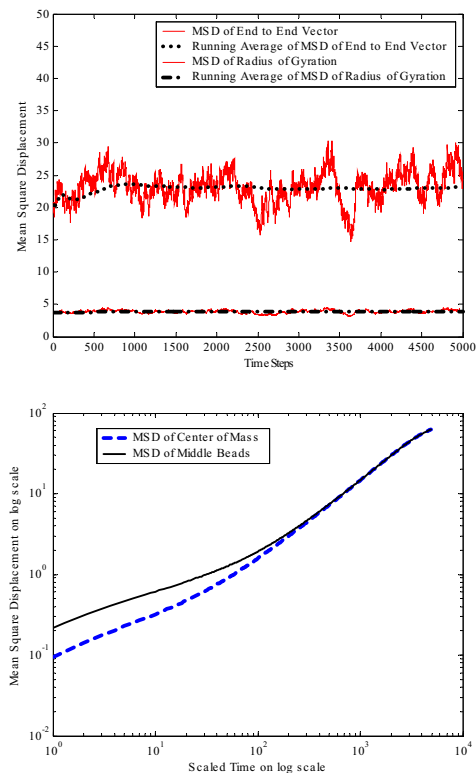


Fig. 6. Structural Properties of Long time Polymer melt Simulations

A few critical structural properties commonly studied in polymer melt simulations are also pre-

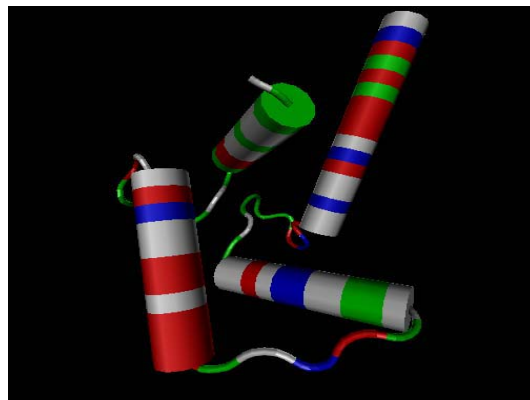


Fig. 7. The C-terminal region calmodulin

sented. Figure (6) shows the plot of the mean-square displacement of the center of mass of the system ($g\beta$) and that of the middle beads in the chain ($g1$) on a log-log scale. The $g\beta$ 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 as shown in Figure (6). As expected, both these values are varying with time but the moving average 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 ensemble. 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 is an example of a protein that undergoes changes in conformation as reported in [50] - [52]. Calmodulin (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. Calmodulin is also known to have interactions with a wide variety of proteins including kinases and cyclases.

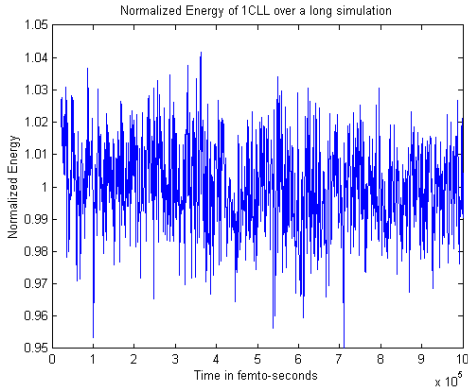


Fig. 8. Energy Plot of Long Time Simulation of Coarse-grained Calmodulin Protein

A representation of the protein is shown in figure (7). 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.

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 α helices and β 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 CHARMM [53] force field. The implicit solvent version of the CHARMM force field was used in lieu of explicit water. A time-step 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 nanosecond simulation of the C-terminal of the protein in NVE ensemble are presented here.

The 1 nanosecond simulation evidenced a good energy conservation with a time-step of 10fs. Figure (8) 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 simu-

lation. 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 ϕ - ψ torsional angles. Figure (9) presents the Ramachandran plot. The plot shows the variations of the ϕ - ψ angles during the entire simulations with the ϕ and ψ angles running along the horizontal and vertical axes, respectively, from 180 to -180 degrees. The Ramachandran plot shows three distinct domains of motion shown by the blue smears. These correspond to the relative motion between the α helices, between the β strands and in the loop domains.

The RMSD variations of the ϕ and ψ angles at each residue are shown in figure (10). 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 α -helix 1, the first β strand, helices 2 and 3, the second β strand and the α -helix 4 respectively. Though the plots show large localized variations in conformation with as much as 70 degree 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 Calmodulin obtained from the simulation and the theoretical values are plotted in Figure (11). The B factors were calculated using the relationship [54]

$$B = \frac{8\pi^2}{3} \langle (\Delta r)^2 \rangle \quad (23)$$

where $\langle (\Delta r)^2 \rangle$ is the mean square of the displacement of C_α 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.

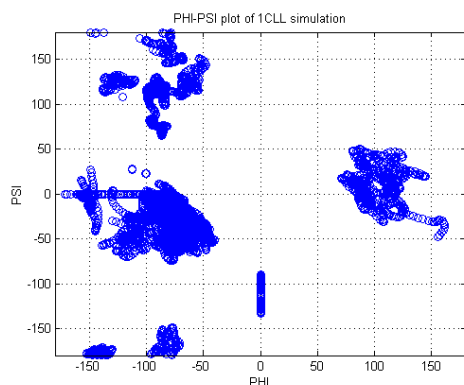


Fig. 9. ϕ - ψ Plot of Long Time Simulation of Coarse-grained Calmodulin Protein

Figure (12) shows the contact map of the protein while figure (13) the root mean squared displacement (RMSD) of the protein during the simulation. The contact map shows the average C_α 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 \AA distance indicating near neighbors or at contact distance, white squares correspond to distances greater than 10 \AA , while the grey ones indicate distance between 0 and 10 \AA . The plot shows a clear backbone of the protein shown along the diagonal, and also the interactions between different residues shown in grey 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, figure (14) 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 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, includ-

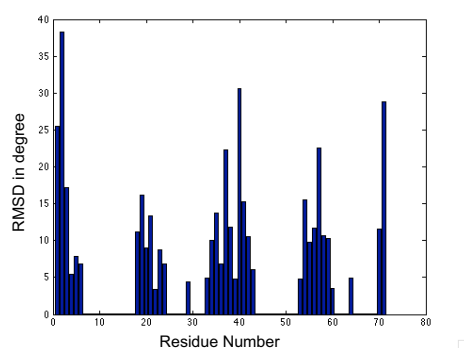
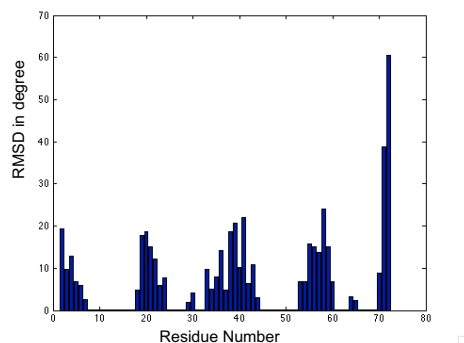


Fig. 10. Per Residue ϕ and ψ Angle Plot of Long Time Simulation of Coarse-grained Calmodulin Protein

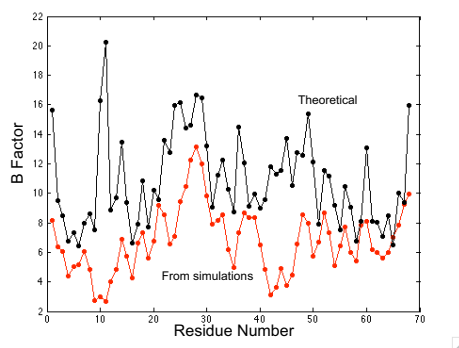


Fig. 11. Comparison of B-factor calculated for Calmodulin

ing 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.

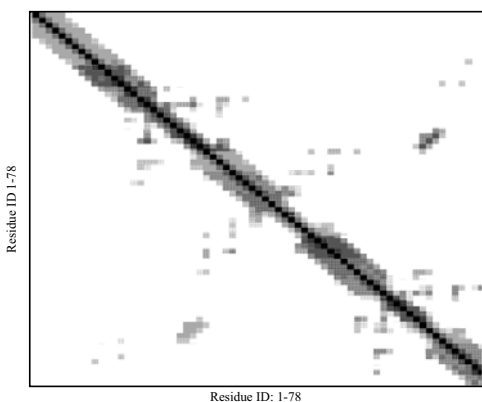


Fig. 12. Inter-Residue Contact Map of Long Time Simulation of Coarse-grained Calmodulin Protein

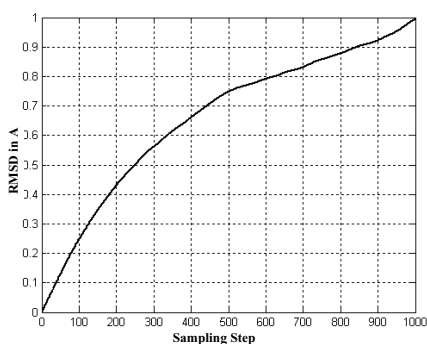


Fig. 13. RMSD of Long Time Simulation of Coarse-grained Calmodulin Protein

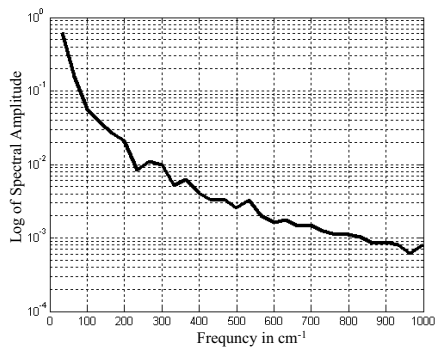


Fig. 14. Spectral Plot of Long Time Simulation of Coarse-grained Calmodulin Protein

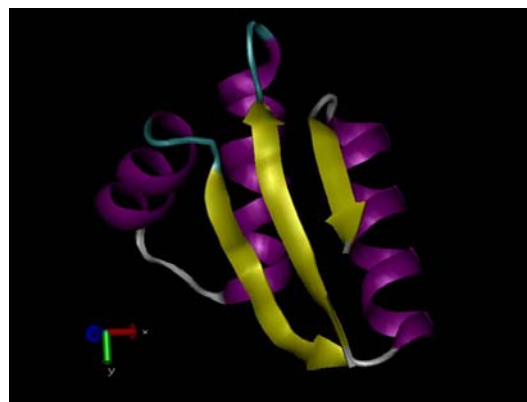


Fig. 15. Coarse-grained Ribosomal Protein L7/L12

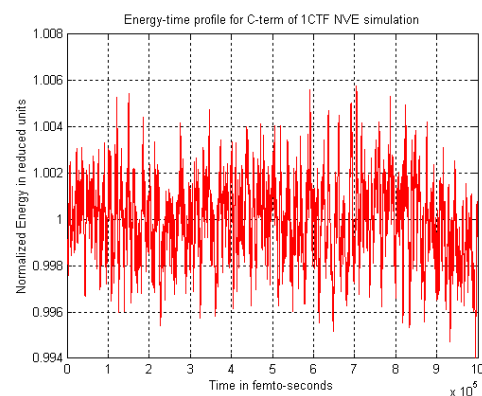


Fig. 16. Energy Plot of Long Time Simulation of Coarse-grained Ribosomal Protein L7/L12

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 α helices and two β strand domains. Figure (15) shows the topology of the protein fragment in the cartoon representation where the topology consists of β strand 1, α helices 1 and 2, β strand 2 and finally the α 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 ϕ or ψ angles. The simulation is monitored for energy conservation and preservation of essential dynamics.

Figure (16) shows the fluctuations in the total energy of the system for a 1 nanosecond simulation carried out with a 10 femto-second integration time step. 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.

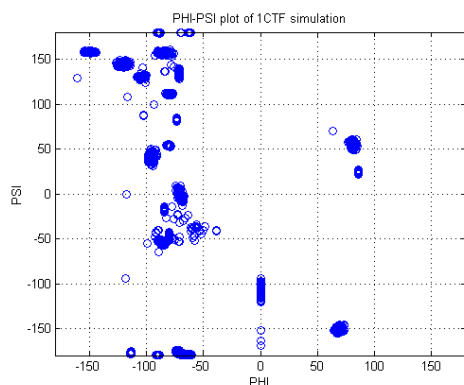


Fig. 17. ϕ - ψ Plot of Long Time Simulation of Coarse-grained Ribosomal Protein

As with the previous test case, the conformational variations of the protein are studied using the Ramachandran plot and the per residue ϕ and ψ angle plots. Figure (17) 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 figure (18) which shows the RMSD variations in the ϕ and ψ angles at each residue in the system. The plots clearly separate the regions corresponding to the secondary structures. In contrast with the corresponding figure (10) for the Calmodulin 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 show variations of about 20 degrees which indicate localized motion. There are no discontinuous changes in the angle values indicating that the simulation does not suffer from abrupt conformational changes.

Figure (19) 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 com-

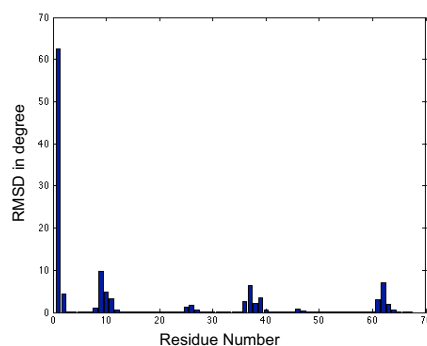
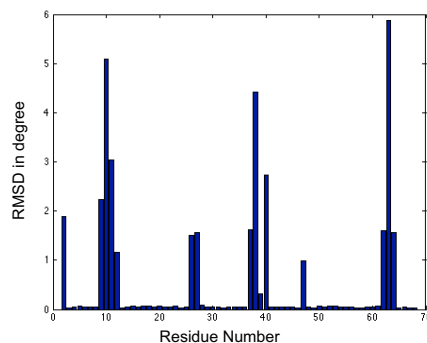


Fig. 18. Per Residue ϕ and ψ Angle Plot of Long Time Simulation of Coarse-grained Ribosomal Protein

ponents. 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. These angular variations are of low amplitude and correspond to the higher frequency motion. The spectral plot presented here clearly shows the dominant low frequency mode of motion and is in agreement with results in [46].

Figure (20) 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 a model of the *RuBisCO* (Ribulose-1,5-Bisphosphate Carboxylase/Oxygenase)

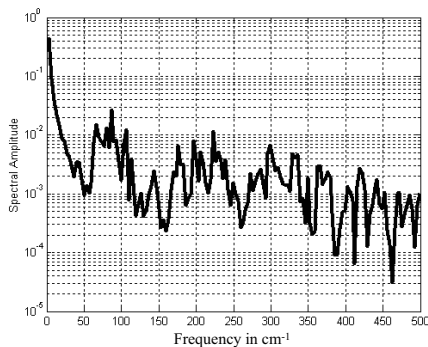


Fig. 19. Spectral Plot of Long Time Simulation of Coarse-grained Ribosomal Protein

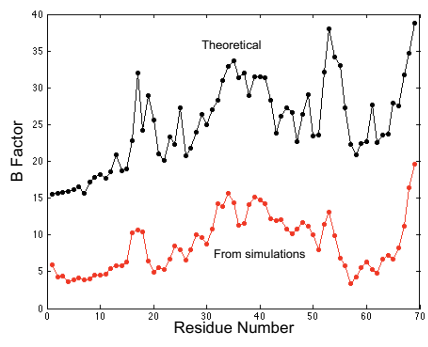


Fig. 20. Comparison of B-factor calculated for Ribosomal

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.

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 bodies connected together to form an articulated serial chain topology. All atom explicit force calculations are supported by this model.

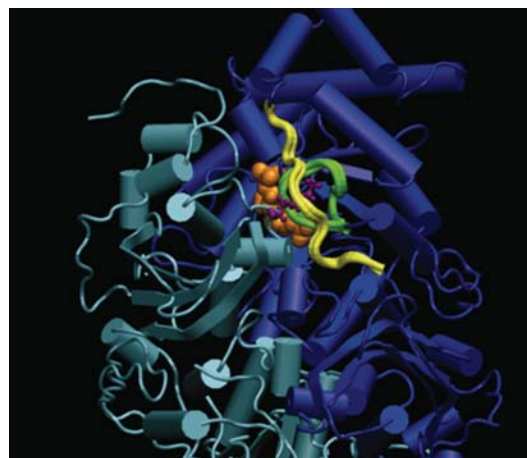


Fig. 21. Coarse-grained RuBisCO Protein

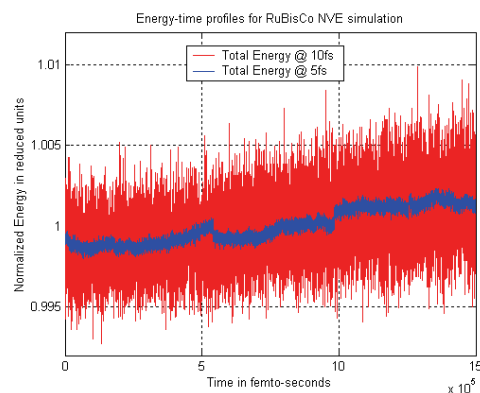


Fig. 22. Energy drift plot of long RuBisCO Protein Simulation

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 validation of the modelling approach. Using this model, NVE simulations were run for 1.5 nano-seconds. Different time steps were used to determine the drift in energy as a function of time-step. The following figure (22) shows the fluctuations in the normalized energy of the system. For the 5 fs time-step, the energy conservation is excellent and even for the 10 fs time-step, 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.

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.

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 one nanosecond 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 15000 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. Figure (24) 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 nanosecond simulation at 10 femto-second integration time step. 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 molecular dynamics simulations have been discussed. Two different types of coarse-grained models have been

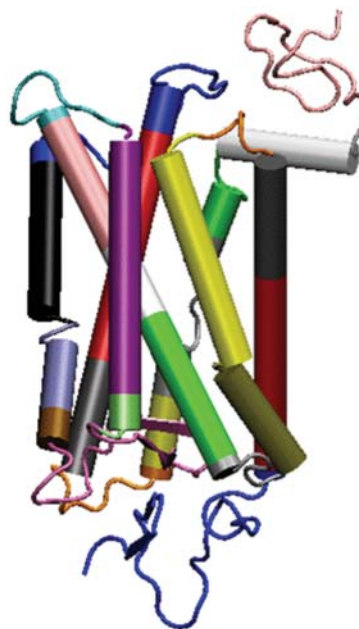


Fig. 23. Coarse-grained Rhodopsin Protein

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 of model is pursued where only the highest frequency content of the system is constrained. Time steps of several femtoseconds are achieved with good energy conservation and retention of structural properties. The atomistic simulations were stable only with sub-femtosecond integration time step. 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 model 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 time steps an order of magnitude larger than the required time step for atomistic simulations. The results reported here are for simulations with a conservative 10 femtosecond time step, an order of magnitude increase in comparison with atomistic simulations. Only 2-3% violation in energy conservation

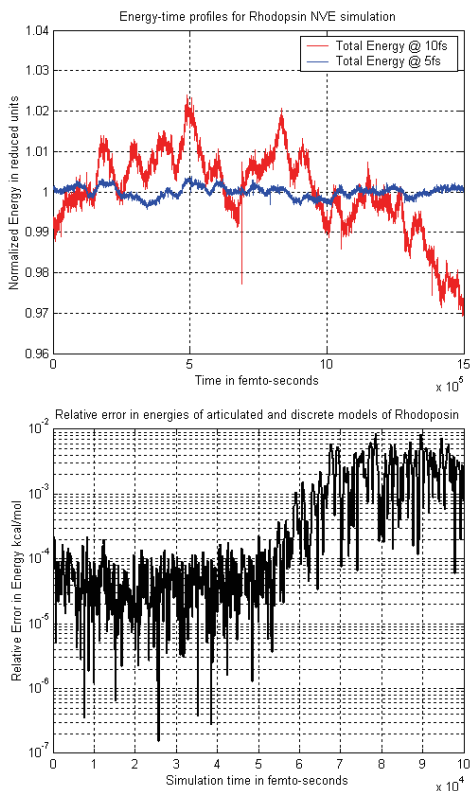


Fig. 24. Energy Results for Long Time Coarse-grained Rhodopsin Protein Simulation

is observed for nanosecond simulations. The conformational statistical results are in agreement with reference studies cited in literature.

While more aggressive time steps have been used in the studies and can definitely be used to get stable simulations, these results are not shown here. This is because, 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 time steps in the range of 20-40 femtoseconds can be achieved. In these models, computational savings are also achieved from reduction in the number of interactions sites for force computations. Figure (25) shows a plot summarizing variations in error in energy conservation with increase in integration time steps 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 time step integration schemes where all degrees of freedom of the system are temporally propagated using the same time step.

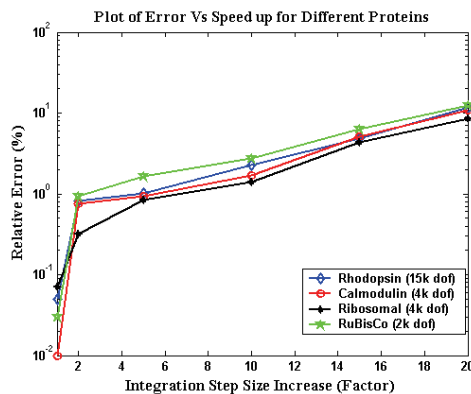


Fig. 25. Variations in computational gains with error

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 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 Calmodulin and Ribosomal were obtained from [60].

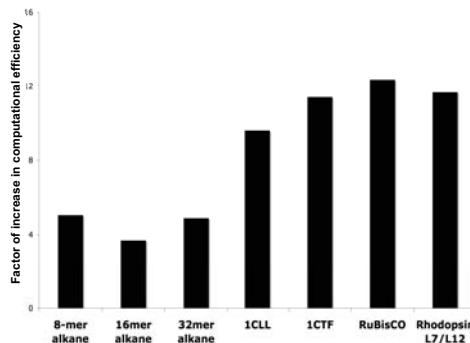


Fig. 26. Speedup obtained in simulating various systems studied

Figure (26) shows the factor of increase in computational speedup obtained by comparing the average simulation times obtained from the method discussed here and classical molecular dynamics 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 nanosecond each.

The time steps for the different simulations were chosen to result in comparable bounds in energy drift. Consequently, the MD simulations were run with a smaller time step as compared to the coarse-grained simulations. For example, the Rhodopsin simulation was run with a time step of 1 femtosecond with classical molecular dynamics, while a time step of 10 femtosecond 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 proteins 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 Multi Time Scale (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 α -helices, several residues could potentially be coarse-grained into a single rigid body. However, using a fixed time step integrator, the simulation of the whole system proceeds with a time step small enough to capture the frequency content of the peptide plane motion. With a MST scheme, it could be possible to propagate different segments of the system at different time scales, resulting in improved simulation times.

7. Software Development

The two research codes used in this work were the classical molecular dynamics package LAMMPS (Large-scale Atomic/Molecular Massively Parallel

Simulator) [35] and the multibody dynamics software POEMS (Parallelizable Open-source Efficient Multibody Software) [36]. 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. Although it can be used for small systems, such as those in some of the example problems, it was designed to efficiently simulate systems with large numbers of coupled 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. Commonly used kinematic and dynamic identities, organization of multibody topologies, and data structures with matrix manipulations which are generic to most multibody algorithms are built into the software and can be called by different modules.

Currently POEMS has three algorithms for solving equations of motion of articulated multi-rigid body systems:

- *KaneSolver()* : The $O(n^3)$ complexity solver based on Kane’s method [37].
- *OnSolver()* : The $O(n)$ complexity recursive solver based [43].
- *DCASolver()* : The Divide and Conquer method of $O(\log(n))$ complexity [64].

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 in the following manner. 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 molecular dynamics timestep. This means 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.

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 time step, POEMS solves the equations of motion of the articulated multi-rigid 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 a 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.

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 WWW site [35].

8. 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, Ru-BisCo and Rhodopsin proteins. The results indicate excellent energy conservation with an integration time step 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.

9. Acknowledgment

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, *Journal of Chemical Physics* 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., *Journal of Chemical Physics* 109 (1998) 7694–7701.
- [3] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computational Physics* 73 (1987) 325–348.
- [4] D. Reith, M. Pütz, F. Müller-Plathe, Deriving effective mesoscale potentials from atomistic simulation, *Journal of Computational Chemistry* 24 (13) (2003) 1624–1636.
- [5] S. Ivzegov, G. A. Voth, A multiscale coarse-graining method for biomolecular systems, *Journal of Physical Chemistry B* 109 (2005) 2469–2473.
- [6] T. D. Hone, S. Ivzegov, G. A. Voth, Fast centroidal molecular dynamics: A force-matching approach for the predetermination of the effective centroidal forces, *Journal of Physical Chemistry* 122 (2005) 054105–1–7.
- [7] H. C. Anderson, Rattle: A ‘Velocity’ version Shake algorithm for molecular dynamics calculations, *Journal of Computational Physics* 54 (1983) 24–34.
- [8] J. P. Rychaert, G. Cicotti, H. J. C. Berendsen, Numerical integration of the Cartesian equations of motion of a system of with constraints: Molecular dynamics of n-alkanes, *Journal of Computational Physics* 23 (1977) 327–341.
- [9] E. Barth, K. Kuczera, B. Leimkuhler, R. D. Skeel, Algorithms for constrained molecular dynamics, *Journal of Computational Chemistry* 16.
- [10] P. J. Ludovice, U. W. Suter, Molecular dynamics of geometrically constrained polymer systems in generalized coordinates: Basic formalism, *Computational Polymer Science* 1 (1995) 69–79.
- [11] R. A. J. Abagyan, A. Mazur, New methodology for computer-aided modelling of biomolecular structure and dynamics, *Journal of Biomolecular Structure* 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, *Journal of Computational Physics* 92 (2) (1991) 261–272.
- [13] L. M. Rice, A. T. Brünger, Torsion angle dynamics: Reduced variable conformational sampling enhanced crystallographic structure refinement, *Proteins: Structure, Function and Genetics* 19 (1994) 277–290.
- [14] J. P. Rychaert, A. Bellemans, Molecular dynamics of of liquid n-butane near the boiling pointics of n-alkanes, *Chemical Physics Letters* 30 (1975) 123–125.
- [15] A. Jain, N. Vaidehi, G. Rodriguez, A fast recursive algorithm for molecular dynamics simulation, *Journal of Computational Physics* 106 (2) (1993) 258–268.
- [16] J. D. Turner, P. Weiner, B. Robson, R. Venugopal, H. S. III, R. Singh, Reduced variable molecular dynamics, *Journal of Computational chemistry* 16 (1995) 1271–1290.
- [17] N. Vaidehi, A. Jain, G. W. A., Constant temperature constrained molecular dynamics: The Newton-Euler inverse matrix operator method, *Journal of Physical Chemistry* 100 (1996) 10508–10517.
- [18] H. M. Chun, C. E. Padilla, D. N. Chin, M. Watanabe, 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, *Journal of Computational Chemistry* 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, *Journal of Chemical Physics* 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, *Journal of Computational Chemistry* 95 (1992) 8362–8364.

- [21] J. L. Scully, J. Hermans, Multiple time steps: Limits on the speedup of molecular dynamics simulation of aqueous systems, *Molecular Simulation* 11 (1993) 67–77.
- [22] M. Watanabe, M. Karplus, Dynamics of molecules with internal degrees of freedom by multiple time-step methods, *Journal of Chemical Physics* 99 (1993) 8063–8074.
- [23] M. Watanabe, M. Karplus, Simulations of macromolecules by multiple time-step methods, *Journal of Physical Chemistry* 103 (1995) 3718–3732.
- [24] T. Forester, T. Smith, On multiple time-step methods and the ewald sum, *Molecular Simulation* 13 (1994) 195–204.
- [25] W. B. Streett, D. J. Tildesley, G. Saville, Multiple time step methods in molecular dynamics, *Molecular Physics* 35 (1978) 639–648.
- [26] G. Zhang, T. Schlick, LIN: A new algorithm combining implicit integration and normal mode techniques for molecular dynamics, *Journal of Computational Chemistry*.
- [27] K. Hinsin, Analysis of domain motions by approximate normal mode calculations, *Proteins: Structure, Function, and Genetics* 33 (1998) 417–429.
- [28] D. Janežič, M. Prapotnik, Symplectic molecular dynamics integration using normal mode analysis, *International Journal of Quantum Chemistry* 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, *Journal of Structural Biology* 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, in: Q. Cui, I. Bahar (Eds.), *Normal Mode Analysis: Theory and Application to Biological Chemical Systems*, CRC Press, 2005, submitted.
- [31] P. Petrone, V. S. Pande, Can conformational change be described by only a few normal modes, *Biophysical Journal* In Press.
- [32] C. S. Peskin, T. Schlick, Molecular dynamics by the backward Euler’s method, *Communications in Pure and Applied Math* 42 (1989) 1001–1031, in Press.
- [33] G. Zhang, T. Schlick, The Langevin/implicit-Euler/Normal-mode scheme (LIN) for molecular dynamics at large timesteps, *Journal of Chemical Physics*.
- [34] G. Zhang, T. Schlick, Implicit discretization schemes for langevin dynamics, *Molecular Physics*.
- [35] <http://lammps.sandia.gov/>.
- [36] K. Anderson, R. Mukherjee, J. Critchley, J. L. Ziegler, S. Lipton, Poems: Paralleizable open-source efficient multibody software, *Engineering with Computers* 23 (1) (2007) 11–23.
- [37] T. R. Kane, D. A. Levinson, *Dynamics: Theory and Application*, McGraw-Hill, NY, 1985.
- [38] R. Featherstone, The calculation of robotic dynamics using articulated body inertias, *International Journal of Robotics Research* 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, *Mechanisms, Structures, and Machines* 15 (3) (1987) 359–382.
- [40] D. Rosenthal, An order n formulation for robotic systems, *The Journal of the Astronautical Sciences* 38 (4) (1990) 511–529.
- [41] K. S. Anderson, An order-n formulation for motion simulation of general constrained multi-rigid-body systems, *Computers and Structures* 43 (3) (1992) 565–572.
- [42] A. Jain, Unified formulation of dynamics for serial rigid multibody systems, *Journal of Guidance, Control, and Dynamics* 14 (3) (1991) 531–542.
- [43] K. S. Anderson, An order-n formulation for the motion simulation of general multi-rigid-body tree systems, *Computers and Structures* 46 (3) (1993) 547–559.
- [44] A. R. Leach, *Molecular Modelling Principles and Applications*, Prentice Hall.
- [45] R. M. Mukherjee, K. S. Anderson, Efficient methodology for multibody simulations with discontinuous changes in system definition, *Multibody System Dynamics ISSN* 1384-5640.
- [46] H. Chun, C. Padilla, D. Chin, Masakatsuwatanabe, K. Valeri, 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, *The Journal of Computational Chemistry* 21 (3) (2000) 159–184.
- [47] J. Norberg, L. Nilsson, Advances in biomolecular simulations: Methodology and recent applications, *Quarterly Reviews of Biophysics* 36 (3) (2003) 257–306.
- [48] D. Wolf, W. G. Rudd, Tabulated potentials in molecular dynamics simulations, *Computer Physics Communication* 120 (1999) 200–232.
- [49] E. von Meerwalla, S. Beckman, Diffusion of liquid n-alkanes: Free-volume and density effects, *Journal of Chemical Physics* 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, *Physical Review E* 66 (031908) (2002) 1–8.
- [51] C. Yang, K. Kuczera, Molecular dynamics simulations of calcium-free calmodulin in solution, *Journal of Biomolecular Structure and Dynamics*, 19 (2002) 801–819.
- [52] W. Wriggers, E. Mehler, F. Pitici, H. Weinstein, K. Schulten, Structure and dynamics of calmodulin in solution, *Biophysical Journal* 74 (1994) 1622–1639.
- [53] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus., Charrm: A program for macromolecular energy, minimization, and dynamics calculations, *Journal of Computational Chemistry* 4 (1983) 187–217.
- [54] D. Kosztin, S. Izraileve, K. Schulten, Unbinding of retinoic acid from its receptor studied by steered molecular dynamics, *Biophysical Journal* 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, *Biophysical Journal* 87 (2004) 780–791.
- [56] Y.-H. Sanejouand, O. Tappin, Low-frequency domain motions in the carboxy terminal fragment of 17/112 ribosomal protein studied with molecular dynamics techniques: Are these movements model independent?, *Journal of Physical Chemistry* 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: Structure, Function, and Genetics* 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, *Journal of Molecular Biology* 333 (2003) 493–514.
- [59] W. Humphrey, A. Dalke, K. Schulten, VMD – Visual Molecular Dynamics, *Journal of Molecular Graphics* 14 (1996) 33–38.
- [60] <http://ignm.cccb.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, *Journal of Chemistry* 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. C. 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, *Computer Physics Communications* 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, *International Journal of Robotics Research* 18 (9) (1999) 867–875.