

A Fast Algorithm for Massively Parallel, Long-Term, Simulation of Complex Molecular Dynamics Systems

Amir Fijany¹, Tahir Cagin², Andres Jaramillo-Botero², and William A. Goddard, 111²

¹Corresponding Author: MS 525-3660, Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109, USA

Tel: (818) 306-6491; Fax: (818) 306-6912; Email: Amir.Fijany@jpl.nasa.gov

²Material and Process Simulation Center, Beckman Institute, California Institute of Technology

EXTENDED ABSTRACT

The advances in theory and computing technology over the last decade have led to enormous progress in applying atomistic molecular dynamics (MD) methods to the characterization, prediction, and design of chemical, biological, and material systems. MD simulation of very large systems are currently being performed by using massively parallel architectures. However, there is a fundamental difficulty with time and length scales that obstructs these MD methodologies for application to some of the most important problems. In fact, with the current technologies it is routine to consider the MD of large scale systems for a time scale of nanosecond. The problem is that many important applications require a time scale of a microsecond or more. In order to achieve such a significant improvement in the computational efficiency, in addition to massively parallel platforms, new algorithms are needed to allow efficient long-term MD simulation.

There are two major computational steps in the MD simulation: calculation of the interaction forces and integration of equations of motion (EOM). There has been a significant research effort on the development of fast algorithms for calculation of interaction forces. A key advance in the calculation of these forces is the development of the Cell Multipole Method (CMM) with an optimal cost of $O(n)$ for a system with n atoms [1]. In addition to the optimal computational cost, the CMM has been also shown to be very efficient for massively parallel implementation [2].

The simplest and most widely used MD methods employ Cartesian coordinates, so that the EOM for a system with n atoms is written as

$$M_C \ddot{\mathbf{X}} = \mathbf{F} \quad (1)$$

where \mathbf{X} and $\mathbf{F} \in \mathbb{R}^{3n}$ are the vectors of Cartesian coordinates and the force acting on the atoms, and M_C is a $3n \times 3n$ diagonal mass matrix. Although Eq. (1) is simple, the use of Cartesian coordinates introduces limitations into the MD simulation due to the inclusion of all atomic motions. However, in polymer and biological systems, the main concerns are conformational dynamics, involving the low frequency, collective motions of the system. Conformational motions may involve time scales up to microseconds and longer. In contrast, atomistic MD simulations, must use integration timestep of 1 to 2 femtoseconds (to maintain stability). Consequently, to simulate a microsecond of conformational collective motions in a large molecular system requires a billion timesteps, which is generally quite impractical.

The requirement for a small integration timestep stems directly from the high frequencies of the vibrating bonds in the atomistic simulation. For conformational dynamics such motions are of little interest, and we would like to remove these degrees of freedom from the system. One approach (implemented in the SHAKE algorithm [3]) uses hard constraints on bonds and angles to freeze all fast internal vibrations. With the inclusion of constraints, the EOM are no longer

ODES, as given by Eq. (1), but more complex differential algebraic equations. However, even with this additional complexity, only slightly longer integration timesteps (2 to 4 femtoseconds) can be achieved.

In addition to freezing bonds and angles, often it is reasonable to partition subsets of atoms in the system into completely rigid bodies. Reducing a set of atoms to a single body with just six degrees of freedom lowers the overall computational cost and eliminates the need to apply constraint satisfaction algorithms to the atoms internal to the rigid body. Constraints are used only to tie the linked bodies to each other. A natural and efficient scheme for inclusion of rigid bodies and linking constraints is to write the EOM using internal coordinates [4-7]. For a molecular system with m constraints and $N = 3n - m$ total degrees of freedom, the EOM are given by

$$M\ddot{Q} = \mathcal{F}(Q, \dot{Q}) \quad (2)$$

where $M \in \mathbb{R}^{N \times N}$ is the mass matrix, $Q \in \mathbb{R}^N$ is the vector of internal coordinates, and $\mathcal{F}(Q, \dot{Q}) \in \mathbb{R}^N$ is the vector of nonlinear terms and interaction forces. Note that, the matrix M , unlike the matrix M_C , is a dense matrix. In recent approaches, e.g. [4,5], the Lagrangian method is used for solving (2). In this approach, first the matrix M is *explicitly* computed and then the linear system in (2) is solved, leading to an overall computational complexity of $O(N^3)$ which is a major limiting factor for simulation of large molecular systems. Most recently, an $O(N)$ algorithm has been proposed for solving (2) which avoids the explicit computation and inversion of matrix M [6]. The implementation of this algorithm at the Caltech's Material and Process Simulation Center (MPSC) has shown that a much larger integration timestep (by a factor of up to 30) over the conventional atomistic can be achieved [7]. However, the main drawback of this $O(N)$ algorithm is that it is *strictly sequential*, i.e., regardless of the number of processors employed, only a very limited speedup in its computation can be achieved. Since the calculation of interaction forces can be efficiently parallelized, it follows that for a successful simulation of large MD systems in a *massively-parallel environment*, efficient parallel solution of Eq. (2) is the key enabling factor.

Motivated by this analysis, we have recently developed the Constraint Force (CF) algorithm which differs from the previous $O(N)$ algorithms in that it is based on a rather unconventional strategy for solving (2). In the CF algorithm, which was originally developed for dynamic simulation of robotics systems and spacecraft [8], a new factorization of the inverse of the mass matrix M in form of *Schur Complements* is derived as

$$M^{-1} = C - B^t A^{-1} B \quad (3)$$

where t denotes the transpose. For a molecular system with a serial chain topology, $A \in \mathbb{R}^{KN \times KN}$, $B \in \mathbb{R}^{KN \times HN}$, and $C \in \mathbb{R}^{HN \times HN}$ are block tridiagonal matrices, H is the number of degrees of freedom of each body, and $H + K = 6$. Further, for systems not containing close-loop, the matrix A is Symmetric, Positive-Definite (SPD). From (3) and (2), we then have

$$\ddot{Q} = (C - B^t A^{-1} B) \mathcal{F}(Q, \dot{Q}) \quad (4)$$

A sequential implementation of the CF algorithm, i.e., Eq. (4), involves a optimal cost of $O(N)$. However, the main advantage of the CF algorithm is that it can be *fully parallelized*, resulting in a both time- and processor-optimal parallel algorithm for solution of (1). That is, an $O(\text{Log } N)$ algorithm by using $O(N)$ processors. In addition to its theoretical significance by achieving for the first time such optimal bounds in solving (2), the CF algorithm is also highly efficient for practical implementation on massively parallel MIMD architectures due to its coarse grain size and simple communication structure.

In this paper we discuss the theoretical foundation of the CF algorithm and its application to large scale MD simulation. We first consider a MD system with a serial chain topology, i.e, the Polyethylene (PE). However, for practical implementation for systems with $N = 10^3 - 10^5$, even the biggest massively parallel architectures do not provide enough processors to fully exploit

parallelism in the CF algorithm. We show that with $P < N$ processors, the CF algorithm can be optimally implemented with a cost of $O(N/P + \text{Log } P)$. We present the practical implementation of the CF and CMM algorithms on an MIMD parallel architecture, Cray T3D, for the Polyethylene.

Many applications of interest involve systems with more complex topology. An example is the PAMAM-Dendrimer polymer which represents a hyperbranched topology. For such systems, the matrices A , B , and C are no longer tridiagonal but rather highly sparse matrices. We show that, by using $O(N)$ processors, the multiplication of these matrices by a vector can be performed in a fully parallel fashion with a cost of $O(1)$ and involving only a nearest neighbor communication. This implies the multiplication of a vector by the matrix A^{-1} , which corresponds to a sparse SPD linear system solution, can be efficiently performed by using the iterative methods, in particular, the conjugate gradient method. We also develop a more efficient variant of the CF algorithm for systems with branched topology. For example, considering a branched topology system with N rigid bodies, each having one or more branches with L bodies (thus forming a system with a total of NL bodies), this variant of the CF algorithm achieves an optimal computation time of $O(\text{Log } N + P)$ by using $O(N)$ processors.

References

1. Ding, H. Q., Karasawa, N. and Goddard, III, W. A. "The Reduced Cell Multipole Method for Coulomb Interactions in Periodic Systems with Million-Atom Unit Cells" Chem. Phys. Lett. 1966 (1992)
2. Lim, K.-T., Brunett, S., Iotov, M., McClurg, R. B., Vaidehi, N., Dasgupta, S., Tayler, S. and Goddard, III, W.A. "Molecular Dynamics for Very Large Systems on Massively Parallel Computers" submitted to J. Comp. Chem.
3. Ryckaert, J. P., Ciccotti, G. and Berendsen, H. J. C. "Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes" J. Comput. Phys. 23327 (1977).
4. Mazur, A. K., and Abagyan, R. A. "New methodology for computer-aided modeling of bimolecular structures and dynamics. I: Non-cyclic structures" J. Biomol. Struct. Dyn. 6 815 (1989)
5. Mazur, A. K., Dorofeev, V. E. and R. A. Abagyan, "Derivation and testing of explicit equations of motion for polymers described by internal coordinates," J. Comput. Phys. 92 261 (1991)
6. Jain, A., Vaidehi, N. and Rodriguez, G. "A fast recursive algorithm for molecular dynamics simulations" J. Comput. Phys. 106258 (1993)
7. Mathiowetz, A. M., Jain, A. and Goddard, III, W. A. "Protein Simulations using Techniques Suitable for Very Large Systems: The Cell Multipole Method for Nonbond Interactions and the Newton-Euler Inverse Mass Operator Method for Internal Coordinate Dynamics" PROTEINS: Structure, Function, and Genetics 20227 (1994).
8. Fijany, A., Sharf, I. and D'Eleuterio, G. M. T. "Parallel $O(\text{Log } N)$ algorithms for computation of manipulator forward dynamics" IEEE Trans. Robotics & Automation 11389 (1995).