An $O(Log\,N)$ Algorithm for Massively Parallel Molecular Dynamics Simulations

Amir Fijany[1], Muhammad Sahimi[2], and John K. Salmon[3]

[1] Jet Propulsion Laboratory, California Institute of Technology
[2] Department of Chemical Engineering, University of Southern California
[3] Center for Advanced Computing Research (CACR), California Institute of Technology

EXTENDED ABSTRACT

A major ingredient of a successful large scale Molecular Dynamics (MD) simulation is an efficient computational strategy for integrating the equations of motion (EOM). The simplest and most widely used MD methods employ Cartesian coordinates, so that the EOM for a system with n atoms is written as $M_c\ddot{X} = F$, where $\mathbf{X}$ and $F \in \Re^{3n}$ are the vectors of Cartesian coordinates and the forces acting on the atoms, and M_c is a $3n \times 3n$ diagonal mass matrix. This approach limits the size of the time step in the integration of the EOM, thus increasing the computation time. An alternative approach is to solve the EOM by imposing explicit constraints on the Cartesian coordinates of all or certain atoms. To this end, an efficient and natural scheme is to write the EOM in *internal coordinates*. For a molecular system with m constraints and $N = n - m$ total degrees of freedom, the EOM is then given by

$$\mathcal{M}\ddot{Q} = \mathcal{F}(Q, \dot{Q}) \tag{1}$$

where $\mathcal{M} \in \Re^{N\times N}$ is the mass matrix, $Q \in \Re^{N}$ is the vector of internal coordinates, and $\mathcal{F}(Q, \dot{Q}) \in \Re^N$ is the vector of nonlinear terms and interaction potential. In many recent approaches the Lagrangian method is used for solving (1) in which first the matrix $\mathcal{M}$ is *explicitly* computed and then the linear system in (1) is solved, leading to an overall computational complexity of $O(N^3)$. Although these approaches enable the use of a much larger time step, their $O(N^3)$ cost is a major limiting factor for simulation of large molecular systems. Most recently, leveraging advances in multibody dynamics, an $O(N)$ algorithm has been proposed for solving (1) which avoids the explicit computation and inversion of matrix $\mathcal{M}$. 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. However, it is clear that for simulation of large MD systems in a *massively-parallel environment*, an efficient parallel solution of (1) is the key.

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 (1). In our algorithm a new factorization of the inverse of the mass matrix $\mathcal{M}$ in the form of *Schur Complements* is derived as

$$\mathcal{M}^{-1} = \mathcal{C} - \mathcal{B}^t\mathcal{A}^{-1}\mathcal{B}, \tag{2}$$

where t denotes the transpose, $\mathcal{A} \in \Re^{KN\times KN}$, $\mathcal{B} \in \Re^{KN\times HN}$, and $\mathcal{C} \in \Re^{HN\times HN}$ are block tridiagonal matrices, H is the number of degrees of freedom of each body, and $H + K = 6$. From (2) and (1), we then have

$$\ddot{Q} = (\mathcal{C} - \mathcal{B}^t\mathcal{A}^{-1}\mathcal{B})\mathcal{F}(Q, \dot{Q}) \tag{3}$$

A sequential implementation of the CF algorithm, i.e. Eq. (3), involves a 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 $O(Log\,N)$ algorithm by using $O(N)$ processors. In addition to its theoretical significance by achieving for the first time such optimal bounds in solving (1), the parallel CF algorithm is also highly efficient for implementation on emerging 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 also discuss the extension of the CF algorithm to systems with different and more complex topologies. The results of practical implementation of the parallel CF algorithm on an MIMD (32 nodes) Hypercube architecture are also presented. We also discuss more recent implementation of the algorithm on MIMD architectures with a much greater number of processors.

We will also briefly discuss the potential for coupling the CF algorithm with the so-called treecodes algorithms for the evaluation of the force term $\mathcal{F}(Q, \dot{Q})$. One of us (JKS) has demonstrated highly efficient parallel implementation of $O(N)$ and $O(N\,Log\,N)$ algorithms for computing the forces between all pairs of bodies in astrophysical systems. The same mathematical and computational techniques may be used in a molecular dynamics setting, resulting in greatly decreased evaluation time for the right-hand side of Eq. (1).