

Algorithmic Challenges in Computational Molecular Biophysics

Tamar Schlick,* Robert D. Skeel,†,‡ Axel T. Brunger,§ Laxmikant V. Kalé,†,¶
John A. Board, Jr.,|| Jan Hermans,** and Klaus Schulten†,††

**Department of Chemistry and Courant Institute of Mathematical Sciences, New York University and The Howard Hughes Medical Institute, 251 Mercer Street, New York, New York 10012; †Beckman Institute, University of Illinois at Urbana-Champaign, 405 North Mathews, Urbana, Illinois 61801-2987; §Department of Molecular Biophysics and Biochemistry, Yale University and The Howard Hughes Medical Institute, 266 Whitney Avenue, Bass Center, Room 434, New Haven, Connecticut 06520; ||Department of Electrical and Computer Engineering, Duke University, School of Engineering, Durham, North Carolina 27708; and **Department of Biochemistry and Biophysics, University of North Carolina, Chapel Hill, North Carolina 27599-7260*

E-mail: *schlick@nyu.edu, †skeel@cs.uiuc.edu, §brunger@laplace.csb.yale.edu, ¶kale@cs.uiuc.edu, ||jab@ee.duke.edu, **hermans@med.unc.edu, ††kschulte@uiuc.edu

Received August 20, 1998; revised December 11, 1998

A perspective of biomolecular simulations today is given, with illustrative applications and an emphasis on algorithmic challenges, as reflected by the work of a multidisciplinary team of investigators from five institutions. Included are overviews and recent descriptions of algorithmic work in long-time integration for molecular dynamics; fast electrostatic evaluation; crystallographic refinement approaches; and implementation of large, computation-intensive programs on modern architectures. Expected future developments of the field are also discussed. © 1999 Academic Press

Key Words: biomolecular simulations; molecular dynamics; long-time integration; fast electrostatics; crystallographic refinement; high-performance platforms.

1. INTRODUCTION: CHALLENGES, PROGRESS, AND ILLUSTRATIVE APPLICATIONS

Computational techniques for modeling large biological molecules have emerged rapidly in recent years as an important complement to experiment. Computer-generated models and simulation data are essential for analyzing structural and kinetic details that are difficult to capture experimentally for large, floppy macromolecules in solution. Modeling approaches also permit systematic studies of the dependence of equilibrium and kinetic properties on internal (e.g., the amino acid sequence) and external (e.g., the salt concentration in the

TABLE 1
The Steady Advances of Molecular Dynamics Simulations: Representative Biomolecular Systems, Simulation Times, and Computers

	1978	1988	1998
System	Oligopeptides in vacuum, with unified atom representation and simplified force treatment	12–24 base pair DNA in vacuum, with long-range interactions truncated	Fully solvated protein models with each atom represented and with counterions and long-range interactions considered
Size (atoms):	250	2,500	12,500
MD time:	1 ps	10 ps	1 ns
Desktop machine:	Hand calculator 0.000001 MFlop	Sun 3 0.4 MFlop	Sun Ultrasparc II 461 MFlop
Laboratory computer:	DEC Vax 11/780 (1 processor) 0.1 MFlop	DEC Vax 6000 (6 processors) 8.4 MFlop	SGI Origin 2000 (32 processors) 3.1 GFlop
Supercomputer:	Cray 1 (1 processor) 110 MFlop	Cray Y-MP (8 processors) 2.1 GFlop	Cray T3E (1,080 processors) 0.9 TFlop

environment) factors. Such studies can enhance our understanding of biological function through the structure/function connection. New lines of experimentation can be proposed on this basis, and many important practical applications, from medicine to technology, follow.

Though the field of computational molecular biophysics can be considered relatively young, it is developing rapidly with the increasing availability of faster computers, larger memory capacities, and parallel architectures, as well as with new algorithmic developments in allied fields (see Table 1). Algorithmic progress is essential for tackling basic obstacles that complex biomolecular systems pose: (1) large system sizes in the natural environment of biopolymers (solvent, salt, other biomolecules); (2) wide range of motion timescales, from fast electronic rearrangements (femtosecond timescale) to global deformations (milliseconds and longer); (3) overall structural diversity and complexity of the systems investigated.

As Table 1 shows, impressive progress has already been achieved in lengthening simulation times and increasing system sizes. Typical simulation times and system sizes are shown in the table, but there are extreme recent examples of unusually long or large simulations (e.g., 1 μ s of a small protein [38], or nearly a million atoms). Thus, the development of more efficient simulation algorithms and computational strategies is continuously motivated by the remaining temporal gap needed to describe protein folding or ligand binding—more than three orders of magnitude—and by a similar size gap, for simulating nanometric biomolecular aggregates.

In addition to larger systems and longer simulation times, further progress in the field must rely on new physical concepts. Examples include approaches for coarse graining over time and length scales, merging macroscopic and microscopic models, and incorporating quantum descriptions of biomolecular systems. Multiscale approaches, successful in other scientific applications with a disparity of timescales, are difficult to implement for biomolecules since structural and dynamic features are so intimately coupled; yet the

potential for success by such techniques exists [114]. Quantum extensions of classical Newtonian mechanics are essential for a variety of properties that involve electronic rearrangements, such as energy transfer and photoprocesses in proteins.

This article describes some accomplishments of a collaborative group funded for 5 years by the National Science Foundation, commencing in 1993, under the High-Performance Computing and Communication initiative. The work described herein also provides a perspective on activities in the field in general. The algorithmic areas covered in this article are long-time integration (Section 2), rapid evaluation of electrostatic potentials (Section 3), high-performance implementations of large simulation packages (Section 5), and experimental-data refinement (Section 6). The remainder of Section 1 describes four examples to illustrate the above computational challenges in terms of specific biomolecular applications. Section 4 presents a case study in the merging of accelerated timestepping and electrostatic protocols in a protein dynamics application. Future perspectives are described in the final section. Readers are referred to general introductions to molecular and biomolecular modeling found in [2, 24, 26, 43, 52, 82, 95].

1.1. Interpretation of Experiments Regarding Unfolding of Immunoglobulin Domains of a Muscle Protein

Molecular recognition and specific ligand/receptor interactions are central to many biochemical processes. Recent experiments have analyzed certain ligand/protein interactions by direct application of an external pulling force on the ligand. These experiments record, as a function of time, the extension (ligand position) in response to the applied force. Together, these measurements yield important structural information about the structure/function relationships of the ligand/receptor complex, binding pathways, and mechanisms underlying the selectivity of enzymes. The elastic properties of biopolymers—long DNA molecules [90] and the muscle protein titin, for example—can also be investigated by such techniques.

The muscle protein titin contains about 30,000 amino acids. Its long filament structure plays an important role in muscle contraction and elasticity. The I-band region of titin, largely composed of immunoglobulin-like (Ig) domains, is responsible for the molecule's extensibility and passive elasticity.¹ In atomic force microscopy experiments, sections of titin composed of adjacent I-band Ig domains were stretched. Measuring the applied forces and the resulting extension revealed a force/extension profile. The profile displays a pattern of sawtooth-shaped peaks, with each force peak corresponding to the unfolding of a single Ig domain. This pattern implies that the Ig domains unfold one by one under the influence of an applied external force. Details of these structural changes, however, were not known.

Simulations using *steered molecular dynamics* (SMD) mimicked these experiments by applying an external force to a protein/ligand complex in which the ligand is restrained to a point in space by a harmonic potential. The restraint point is then shifted in a chosen direction, forcing the ligand to move from its initial position in the protein and allowing the ligand to explore new contacts along its unbinding path. SMD has already been successfully applied to studies of the dissociation of biotin from avidin [48, 66], the unbinding of retinal from bacteriorhodopsin [64] and of retinoic acid from its nuclear hormone receptor [77], and the extraction of lipids from membranes [91, 122]. A review of the method is provided in [67].

¹ The term *passive elasticity* used in the muscle field implies that part of the elastic muscle tissue which does not generate forces.

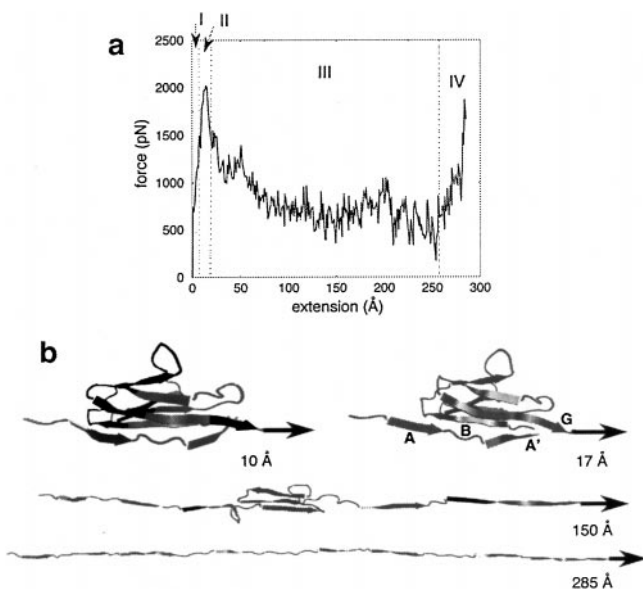


FIG. 1. (a) Force/extension profile from SMD simulations of the titin I27 domain with a pulling velocity $v = 0.5 \text{ \AA/ps}$. The extension domain is divided into four sections: I, pre-burst; II, major burst; III, post-burst; IV, pulling of fully extended chain. (b) Intermediate stages of the force-induced unfolding. All I27 domains are drawn in the cartoon representation of the folded domain; surrounding water molecules are not shown. The four figures at extensions 10, 17, 150, and 285 Å correspond, respectively, to regions I to IV defined in (a). Figure created using [61].

In the application of SMD to the force-induced unfolding of the immunoglobulin domains of titin [88], the starting structure for the SMD simulations was constructed on the basis of a nuclear magnetic resonance (NMR) structure of the Ig domain I27 of the cardiac titin I-band [63]. The 300-ps simulation of a single Ig domain solvated in a bubble of water, shown in Fig. 1, was performed with the XPLOR 3.1 program. Twelve days of computer time was needed on an SGI Onyx2 195-MHz MIPS R10000 processor.

The Ig domains consist of two β -sheets packed against each other, with each sheet containing four strands, as shown in Fig. 1b. After I27 was solvated and equilibrated, SMD simulations fixed the N-terminus and applied a force to the C-terminus in the direction of the vector connecting the two ends. The force/extension profile from the SMD trajectory shows a single force peak (see unraveling of chains in Fig. 1) at short extension, a feature which agrees with the sawtooth-shaped force profile exhibited in atomic force microscopy experiments. The simulation also explained in atomic detail how the force profile in Fig. 1a arises, a sample trajectory being shown in Fig. 1b [88].

Still, these molecular views help interpret only part of these unfolding experiments: The maximum force required for the experimental unfolding of an Ig domain (2000 pN) exceeds the simulated force (250 pN) by a factor of about 10. This discrepancy is mainly due to the associated 10^6 -fold time gap between simulation and experiment. Naturally, algorithmic work, such as that described in this article, e.g., the Langevin trajectory method, should extend such calculations to significantly longer times. Such advances may also permit study of the refolding process of the Ig domains. Presently, a simulation of 10 ns on a cluster of eight PC workstations using the simulation program NAMD2 [72] requires 3 weeks.

An alternative solution to overcome the problem described is suggested in another article in this issue [55].

1.2. Protein/DNA Interactions

In addition to the spatial and temporal limitations, full account of the long-range electrostatic interactions in biomolecules is another computational challenge, especially for polyelectrolyte DNA. Investigations of DNA and of DNA/protein structures are important for understanding basic biological processes such as replication, transcription, and recombination. Advances in crystallography and NMR have elucidated the structures of many protein/DNA complexes, but explanations of the physical mechanisms underlying the function of proteins binding to DNA require further explorations by means of simulations.

A recent simulation of the binding of an estrogen receptor to DNA [76] sought to explain the mechanism underlying DNA sequence recognition by the protein. The estrogen receptor recognizes and binds to specific DNA sequences for the purpose of controlling expression of specific genes. The simulations complexed a dimer of the protein's DNA binding domain to an asymmetric segment of DNA to help identify by comparison the binding interactions and structural features important for sequence recognition: one side of the DNA was identical to the target sequence and the other side was altered, containing two mutations in the sequence (see Fig. 2).

The simulations were performed using the program NAMD [97] on the solvated protein/DNA complex involving approximately 36,600 atoms (Fig. 2). The 100-ps simulation required 22 days in 1996 on eight HP 735 workstations. The full electrostatic interactions are evaluated through the use of a multipole expansion algorithm, namely the program DPMTA [105] (see Section 3). This proper electrostatic treatment was crucial for the stability of the simulated DNA and for proper description of the local water environment. The simulations revealed that water molecules bridging the protein–DNA contacts are instrumental in the recognition of the target DNA sequence [76].

1.3. High-Density Lipoprotein Aggregates

Molecular aggregates, large networks of several biomolecular systems, pose special challenges to modelers due to the system complexity. Not only are some complexes, such as lipid/proteins, extremely difficult to crystallize; they may exhibit multiple conformational states. For example, the protein apolipoprotein A-I (apoA-I), a component of high-density lipoproteins (HDL), is known to have different conformations in the states with bound and unbound lipids. In addition to the 243-residue protein apoA-I, HDL particles consist of phospholipids and several smaller proteins. Experiments show [69] that typical nascent HDL particles are discoidal in shape, consisting of two apoA-I proteins surrounding a circular patch of lipid bilayer.

To explore this complex system, computer modeling has been used to predict and test a three-dimensional description of this system [101]. In the absence of an atomic-resolution model, the geometry was constructed on the basis of the regularity in the apoA-I sequence as well as experimentally observed properties of reconstituted HDL particles. The resulting model, shown in Fig. 3, was tested for stability via simulated annealing [101].

The modeled HDL particle consists of 2 proteins, 160 lipid molecules, and 6000 water molecules (46,000 atoms total). The large size of the system required the use of truncated

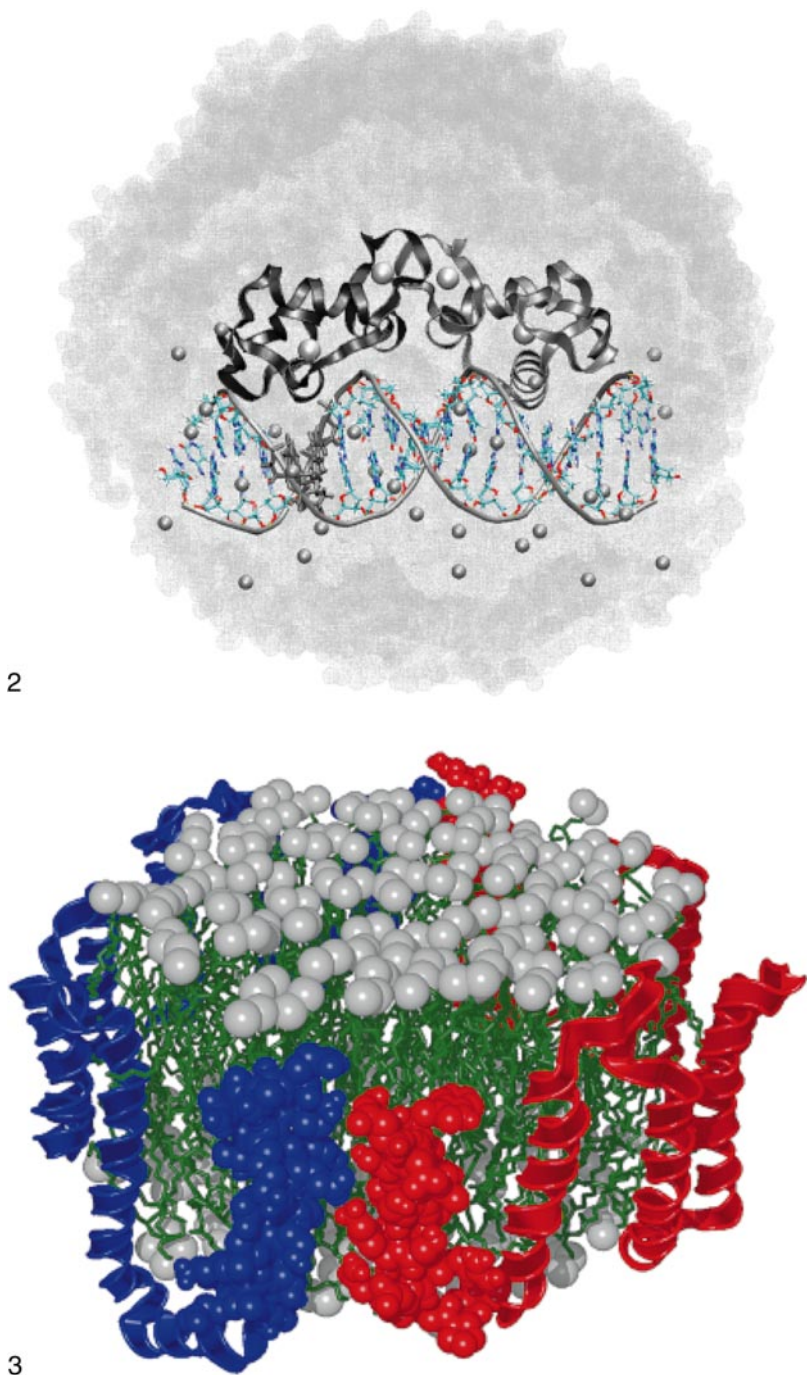


FIG. 2 (TOP). Simulated complex of an estrogen receptor (ER) homo dimer complexed to an asymmetric DNA segment. The structure is oriented such that the viewer is looking along the so-called recognition helices lying in adjacent major grooves of the DNA. The segment contains on one half-site the target sequence of ER and on the other half-site the target sequence with two base pairs mutated; the latter base pairs are emphasized graphically. Shown are also the sodium ions (as van der Waals spheres) and the sphere of water molecules included in the simulations. For a better view of the protein and the DNA, half of the sphere of water was cut. Figure created using VMD [61].

long-range interactions (implemented via a switching function in the range 10–12 Å). The 250-ps simulation was run by the program NAMD2 described in Section 5 employing a two-timestep protocol (see Section 2). On a four-processor HP K460 the simulation required 13 days. Recent performance improvements in NAMD2 [72] can halve the required computing time.

The predicted structure for this system (Fig. 3) awaits experimental verification and further theoretical studies. A future goal is to treat the long-range electrostatic interactions of HDL particles more accurately by avoiding truncation and relying on fast Ewald schemes (see Section 3).

1.4. Quantum Dynamics of Bacteriorhodopsin

An example of a system that demands a quantum mechanical treatment is the proton transport associated with light absorption in photosynthesis. Bacteriorhodopsin (bR) is a protein that exhibits the simplest known form of biological photosynthetic energy storage. It absorbs light and converts its energy into a proton gradient across the cellular membrane of archaeobacteria, transported through vectorial proton translocation [87]. To understand how proton transport is coupled to light absorption, the photodynamics of the retinal chromophore that intercepts the proton conduction pathway must be explored. Namely, upon the absorption of light, the chromophore undergoes a subpicosecond phototransformation in the dihedral orientations around a double bond. The main reaction product triggers later events in the protein that induce pumping of a proton across the bacterial membrane. The structural details of bR, as shown in Fig. 4, are consistent with this pump mechanism.

The primary phototransformation of bR has been suggested to proceed on three potential surfaces associated with different electronic states [117]. Accordingly, it cannot be described via a straightforward solution of Newton’s classical equations of motion. Instead, the quantum mechanical nature of the nuclear degrees of freedom must be confronted in order to account properly for jumps between the three surfaces. To this end, a formally exact quantum-mechanical procedure reported in [14, 92] has been adopted and the motion of bR described quantum mechanically [15]. The procedure expanded the wave function for the nuclear motion of bR on each electronic state in terms of linear combinations of multidimensional Gaussians. Each such Gaussian is represented as a product of “frozen” Gaussians, one factor for each of the 11,286 degrees of freedom of the protein. The Gaussians are characterized by mean positions and momenta that obey classical equations of motion. For the solution of these equations, the classical MD program NAMD [72] had been adapted for the quantum simulations. The phase factors of the bR wave function associated with each of the multidimensional Gaussians are propagated using the Lagrangian which is also supplied by NAMD.

A 1-ps quantum trajectory, sufficient to account for the femtosecond photoprocess in bR, required 1 week of CPU time on a 195-MHz MIPS R10000 processor of an SGI-Onyx2. Several trajectories were run and resulted in retinal’s all-*trans* → 13-*cis* photoisomerization

FIG. 3 (BOTTOM, Previous Page). Predicted structure of an rHDL particle. Two apoA-I molecules (light and medium gray) surround a bilayer of 160 POPC lipids (dark gray). The hydrophilic lipid head groups (top and bottom) are exposed to the solvent while the hydrophobic tails (center) are shielded by the amphipathic protein helices. Figure created using VMD [61].

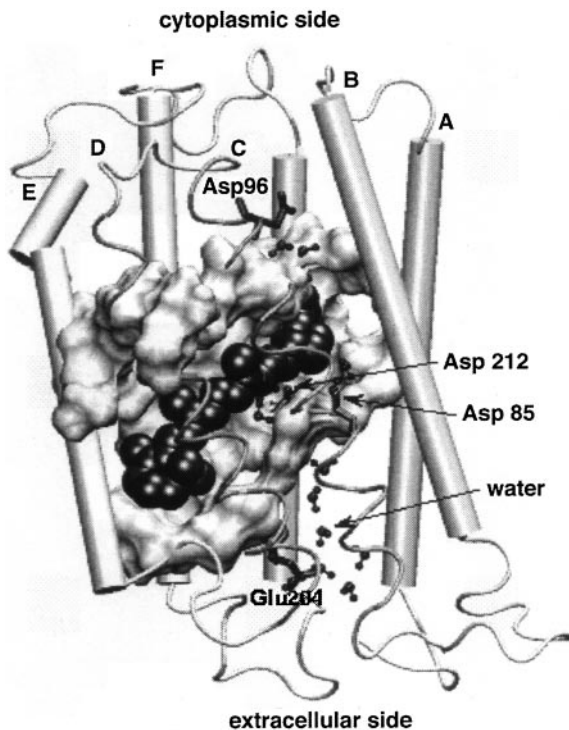


FIG. 4. Bacteriorhodopsin and retinal binding site. Retinal is shown in van der Waals sphere representation, and part of nearby residues are shown in surface representation. Transmembrane helices A, B, E, F, G are shown as cylinders, and helices C, D are shown as thin helical tubes to reveal the retinal binding site. Figure created with VMD [61].

process proceeding on a subpicosecond time scale in agreement with observations. The results revealed that during the isomerization the protonated Schiff base moiety of retinal in bR can switch from ligation with one water molecule to ligation with another water molecule and leave retinal in a conformation suitable for subsequent proton pumping [117].

Naturally, one seeks to extend quantum simulations to other systems and over longer simulation times. It is also desirable to combine quantum simulations of nuclear motion with a synchronous quantum-chemical evaluation of electronic potential surfaces and their non-adiabatic coupling. Respective simulations will require computer resources which exceed presently available hardware speeds by a factor of 100.

2. MOLECULAR DYNAMICS INTEGRATION: LONG-TIMESTEP APPROACHES

2.1. Overview

In molecular dynamics (MD) simulations, the Newtonian equations of motion are solved numerically to generate atomic configurations of a molecular system subject to an empirical force field. Although far from perfect and from agreement with one another, biomolecular force fields are reasonably successful today; they incorporate various experimental and quantum-mechanical information, with parameters chosen to reproduce many thermodynamic and dynamic properties of small molecules (the building blocks of macromolecules).

The typical composition of potential energy functions is

$$\begin{aligned}
E(X) = & \sum_{\text{bonds } i} S_i^b (b_i - \bar{b}_i)^2 + \sum_{\text{bond angles } i} S_i^\theta (\theta_i - \bar{\theta}_i)^2 \\
& + \sum_{\text{dihedral angles } i} \sum_n \left(\frac{V_n^\tau}{2} [1 \pm \cos(n\tau_i)] \right) + \sum_{\text{atoms } i < j} (-A_{ij}/r_{ij}^6 + B_{ij}/r_{ij}^{12}) \\
& + \sum_{\text{atoms } i < j} (Q_i Q_j / D(r_{ij}) r_{ij}) + \dots
\end{aligned} \tag{2.1}$$

In these expressions, the symbols b , θ , τ , and r represent, respectively, bond lengths, bond angles, dihedral angles, and interatomic distances. All are functions of the collective Cartesian positions X . The bar symbols represent equilibrium, or target values. Atomic-sequence-dependent force constants are associated with each interaction. In the third term, each dihedral angle can be associated with one or more rotational index (n). In the Lennard–Jones term (fourth), A and B denote the attractive and repulsive coefficients, respectively. In the Coulomb term (last), the values Q denote the atomic partial charges, and the function $D(r)$ represents a distance-dependent dielectric function, which sometimes is used. Other similar terms are often added to account for hydrogen bonds, planar moieties (via improper dihedral terms), and more. See [112], for example, for a general introduction into the construction, usage, and implementation of these force fields.

Even assuming the perfect force field, we face the difficulty of simulating long-time processes of these complex and inherently chaotic systems. Such numerical propagation in time and space of large systems with widely varying timescales is well appreciated in the computational physics community. Simulation of planetary orbits forms a notable example. Although in biomolecules, the target processes are not measured in years but rather in seconds (putting aside evolutionary processes), the timescale disparity is just as computationally limiting. The fastest high-frequency modes have characteristic oscillations of period $P = 10$ fs, more than 10 orders of magnitude smaller than the slow and large-amplitude processes of major biological interest. Since each timestep is very costly—requiring a force evaluation for a large system, a calculation dominated by long-range nonbonded interactions—computational biophysicists have sought every trick in the book for reducing this cost and, at the same time, increasing the feasible timestep.

2.2. The Equations of Motion

The classical Newtonian equations of motion solved at every timestep of MD are

$$\mathbf{M}\dot{V}(t) = -\nabla E(X(t)), \quad \dot{X}(t) = V(t), \tag{2.2}$$

where X and V are the collective position and velocity vectors, respectively; \mathbf{M} is the diagonal mass matrix; $\nabla E(X(t))$ is the collective gradient vector of the potential energy E ; and the dot superscripts denote differentiation with respect to time, t . Often, frictional and random-force terms are added to the systematic force to mimic molecular collisions through the phenomenological Langevin equation, which in its simplest form is

$$\mathbf{M}\dot{V}(t) = -\nabla E(X(t)) - \gamma \mathbf{M}V(t) + R(t), \quad \dot{X}(t) = V(t), \tag{2.3}$$

$$\langle R(t) \rangle = 0, \quad \langle R(t)R(t')^T \rangle = 2\gamma k_B T \mathbf{M} \delta(t - t'). \tag{2.4}$$

Here γ is the damping constant (in reciprocal units of time), k_B is Boltzmann's constant, T is the temperature, δ is the usual Dirac δ symbol, and R is the random Gaussian force with zero mean and specified autocovariance matrix. The Roman superscript T denotes a matrix or vector transpose. As we will show below, the Langevin heat bath can prevent systematic drifts of energy that might result from the numerical discretization of the Newtonian system. The trajectories generated by Newtonian and Langevin models are different.

2.3. The Leap-frog/Verlet Method and Its Limitations

The typical integrator used for MD is the leap-frog/Verlet method [127]. Its generalization to the Langevin equation above is given by the iteration sweep

$$\begin{aligned} V^{n+1/2} &= V^n + \mathbf{M}^{-1} \frac{\Delta t}{2} [-\nabla E(X^n) - \gamma \mathbf{M} V^n + R^n] \\ X^{n+1} &= X^n + \Delta t V^{n+1/2} \\ V^{n+1} &= V^{n+1/2} + \mathbf{M}^{-1} \frac{\Delta t}{2} [-\nabla E(X^{n+1}) - \gamma \mathbf{M} V^{n+1} + R^{n+1}]. \end{aligned} \quad (2.5)$$

The name ‘‘leap-frog’’ arises from an alternative formulation in terms of positions at integral steps but velocities at half-steps (i.e., $X^0, V^{1/2}, X^1, V^{3/2}, \dots$). Here, the superscripts n refer to the difference-equation approximation to the solution at time $n \Delta t$. The superscripts used for R are not important, as R is chosen independently at each step; when the Dirac δ function of Eq. (2.4) is discretized, $\delta(t - t')$ is replaced by $\delta_{nm}/\Delta t$. The popular Verlet method is recovered by setting γ and R^n to zero in the above propagation formulas.

Classic linear stability dictates an upper bound on the timestep for the Verlet method of $2/\omega$ or P/π , where ω is the oscillator's natural frequency and P is the associated period. Recent studies of nonlinear stability [89, 115] indicate a reduced upper bound of $\sqrt{2}/\omega = P/(\sqrt{2}\pi)$. Given the approximately 10-fs period for the fastest period in biomolecules, the Verlet bound on the timestep is $\Delta t < 2.2$ fs. For the Langevin generalization of Verlet only a linear stability bound applies, and this bound is $(2\omega - \gamma)/\omega^2 = (P/\pi) - [(\gamma P^2)/(4\pi^2)]$ (we assume that $\gamma < 2\omega$ for an underdamped oscillator). Of course, adequate resolution of the fast processes and good numerical behavior may require much smaller timesteps than those dictated above, on the order of 0.01 fs or even smaller [35, 113]. Still, typical values used today in single-timestep, unconstrained biomolecular simulations are 0.5 or 1 fs.

2.4. Constrained Dynamics

Constraining the fastest degrees of freedom by augmenting the equations of motion via Lagrange multipliers has made possible an increase from 0.5 or 1 fs to around 2 fs, sometimes slightly higher, with modest added cost. The algorithm known widely as SHAKE [109] and its variant RATTLE [3] are implemented in many molecular packages. The Verlet-based constrained MD algorithm becomes the iterative sequence

$$\begin{aligned} V^{n+1/2} &= V^n - \mathbf{M}^{-1} \frac{\Delta t}{2} \nabla E(X^n) + \mathbf{G}(X^n)^T \lambda^n \\ X^{n+1} &= X^n + \Delta t V^{n+1/2} \\ g(X^{n+1}) &= 0 \\ V^{n+1} &= V^{n+1/2} - \mathbf{M}^{-1} \frac{\Delta t}{2} \nabla E(X^{n+1}), \end{aligned} \quad (2.6)$$

where the vectors λ^n are determined from $g(X^{n+1}) = 0$. Here the $m \times n$ matrix \mathbf{G} is the Jacobian of the vector g , which specifies the m constraints (i.e., component i of g is $b_i^2 - \bar{b}_i^2 = 0$), and λ is the vector of m constraint multipliers. The added cost of the constrained formulation is small because efficient iterative schemes are used to adjust the coordinates after each timestep so that the constraints are satisfied within a specified tolerance (e.g., $\epsilon = 10^{-4}$) [109]. The process can be recast in the framework of iterative nonlinear-system solvers such as SOR, an analogy which helps explain convergence properties, as well as improve convergence via introduction of an additional relaxation parameter [9].

Unfortunately, the SHAKE approach cannot be extended to freeze the next fastest vibrational mode, heavy-atom bond angles, because the overall motion is altered significantly [10, 53]. This strong vibrational coupling is also the reason for failure of highly damping (but stable) methods for stiff differential equations, like the implicit-Euler, for general biomolecular dynamics [116, 135]. Other approaches for integrating the equations of motion exactly in the constrained space [6, 7, 68, 107] are described in the context of crystallography and NMR refinement (Section 6).

2.5. Other Ways to Achieve Speedup in MD

Additional ways to reap speedups from MD simulations are simplified treatments of the long-range forces. The number of bonded forces—arising from bond length, bond angles, and dihedral angle terms—only grows linearly with the number of atoms. In contrast, the number of nonbonded forces—van der Waals and electrostatics—has a quadratic dependence on the system size. For computational manageability, these nonbonded forces have been considered in the past only within a certain pairwise distance, e.g., 8 or 12 Å. However, this procedure clearly misrepresents long-range interactions in highly charged systems, such as nucleic acids, and can also prevent formation of appropriate dispersive networks between the surrounding water and the biopolymer as the simulation evolves and the structure becomes more compact. More recently, the long-range electrostatic forces have been approximated by suitable expansions that reduce the computational complexity to nearly linear, such as multipoles and Ewald summations (see Section 3). The rapidly decaying van der Waals forces can still be treated by cutoff techniques, though more sophisticated schemes than straightforward truncation—smooth switching and shifting functions [121]—are needed to tame the artifacts that might otherwise result from abrupt changes in energy.

Another practical procedure for cutting the work per MD timestep is to use a nonbonded pairlist and update it only infrequently (see Table 3 later for an example). Such pairlists must be constructed for energy evaluations to record the pair interactions considered in the simulation protocol (e.g., long-range force class). All bonded interactions that atom i forms are excluded from the nonbonded pairlist associated with atom i , or $\{j_i\}$, as well as atoms beyond a spherical cutoff range that might be employed. This list determination is especially important in force-splitting approaches, which update forces in different classes at varying, appropriately chosen frequencies. Such multiple-timestep (MTS) schemes were suggested two decades ago for MD [123] and have recently become popular in biomolecular dynamics simulations with the introduction of symplectic variants.

2.6. Symplectic Integrators

Important mathematical contributions regarding symplectic integrators and MTS schemes have made possible further speedup factors in biomolecular simulations. In fact, the good

observed performance of leap-frog/Verlet has been attributed to the method's symplecticness, an area-preserving property of Liouville's theorem associated with certain conservative integrators of Hamiltonian systems [111]. Backward analysis shows that symplectic integrators lead to trajectories that are very nearly the exact trajectories of slightly different Hamiltonian systems; in practice, this also implies no systematic drifts in energy if the timestep is sufficiently small. The use of the symplectic integrators developed in the physics and mathematics communities spread quickly to the computational chemistry community. It was shown that the popular Verlet method is symplectic, as are the RATTLE scheme for constrained dynamics [85] and the impulse-MTS method [18].

2.7. MTS Schemes and Resonance Problems

Symplectic and time-reversible MTS methods for biomolecular dynamics have been described in the early 1990s [47, 126]. A natural splitting for MD simulations is the following three-class hierarchy: bonded forces are considered "fast" and resolved at an inner timestep $\Delta\tau$; forces within a certain cutoff (e.g., 6 Å) are denoted as "medium" and recalculated at the middle timestep $\Delta t_m = k_1 \Delta\tau$; the remaining, slow forces are only computed every $\Delta t = k_2 \Delta t_m = k_1 k_2 \Delta\tau$, the outermost timestep. The ratio $r = k_1 k_2 = \Delta t / \Delta\tau$ determines the overall speedup, since the majority of the work in the total force evaluation stems from the slow component [11]. Typical timestep regimes are $\Delta\tau = 0.5$ fs, Δt_m around 1 or 2 fs, and $\Delta t = 4$ fs. The requirement of symplecticness unfortunately necessitates the merging of the slow forces with remaining terms via periodic "impulses" rather than extrapolation. The earlier, extrapolative variants exhibited systematic energy drifts and were largely abandoned.

To illustrate, a Verlet-based impulse-MTS scheme for a three-class division is described by the following algorithm:

IMPULSE-MTS ALGORITHM.

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta t}{2} \nabla E_{\text{slow}}(X)$$

For $j = 0, k_2 - 1$

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta t_m}{2} \nabla E_{\text{mid}}(X)$$

For $i = 0, k_1 - 1$

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta\tau}{2} \nabla E_{\text{fast}}(X)$$

$$X \leftarrow X + \Delta\tau V$$

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta\tau}{2} \nabla E_{\text{fast}}(X)$$

end

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta t_m}{2} \nabla E_{\text{mid}}(X)$$

end

$$V \leftarrow V + \mathbf{M}^{-1} \frac{\Delta t}{2} \nabla E_{\text{slow}}(X).$$

Note that the application of the slow force components (∇E_{slow}) modifies velocities by a term proportional to $k_1 k_2 \Delta\tau$ — r times larger than the changes made to X and V in the inner loop—only outside of the inner loop (i.e., at the onset and at the end of a sweep covering Δt).

(2.7)

This impulse yields undesirable resonance effects when the outermost timestep is nearly equal to a period of a fast oscillation. This resonance problem was anticipated from the beginning [47] and first reported for simple oscillator systems [18]. The detailed analysis of resonance in Ref. [89] first established a predictive formula for how these general artifacts are related to the timestep and to the integrator used. In particular, it was shown how the most severe, third-order, resonance occurs for the implicit-midpoint scheme when the timestep is about half the fastest period (see Fig. 1 of [89]). The resonant timesteps are method dependent [115]. The instability problem for Δt nearly equal to the half-period was later reported for the impulse–MTS [12, 46]. Detailed analyses of resonance artifacts in extrapolative versus impulse–MTS schemes are presented in [12, 110], where hybrid schemes are proposed to balance stability with accuracy. Resonance instabilities can in general be avoided only by restricting Δt , though certain implicit symplectic schemes can be devised which remove low-order resonances for model systems [115]. Extensions of the impulse–MTS method can delay these resonances (see subsection 2.8), and stochastic extrapolative force-splitting approaches like LN [11] strongly alleviate them [110] (see also subsection 2.9).

In biomolecular systems, the first such resonance for Verlet-based impulse–MTS methods occurs at about $\Delta t = 5$ fs—half the period of the fastest oscillation. Although not explained by resonance in the papers which first applied these MTS variants to biomolecules [60, 130], large energy growth has been reported beyond this threshold; see also recent illustrations in [19]. Barth and Schlick found that the introduction of stochasticity into MTS schemes delays resonance artifacts of impulse–MTS to an outer timestep near the period (10 fs) rather than half-period; see vivid illustrations for two proteins in Ref. [12], a solvated protein in [110], and also Fig. 7 here. Thus, with this severe limit on the outermost timestep (and hence the ratio r), speedup factors are at most 5 with respect to single-timestep simulations at the inner timestep $\Delta \tau$ for Newtonian dynamics (larger for Langevin).

The speedup can be estimated if the ratios of times spent on evaluating the fast and medium forces, with respect to the total force, are known (i.e., $r_f = T_{\nabla E_{\text{fast}}}/T_{\nabla E}$ and $r_m = T_{\nabla E_{\text{mid}}}/T_{\nabla E}$). As derived in [11], the speedup of the triple-timestep impulse–MTS method over Verlet (or the Langevin analog) can be estimated from the formula

$$\text{impulse–MTS speedup} = k_1 / [(r_m + k_1 r_f) + (1/k_2)]. \quad (2.8)$$

Using $k_1 = 4$, $k_2 = 2$, $r_f = 0.02$, and $r_m = 0.15$, we obtain the observed speedup of approximately 5. Smaller values of r_m and r_f will increase the speedup. Of course, a larger k_2 value would also increase the speedup, but this is possibly limited due to resonance artifacts.

Current work focuses on alleviating these severe resonance artifacts, or overcoming this timestep barrier so that larger computational gains can be realized from MTS methods. The next two subsections describe such strategies.

2.8. The Mollified Impulse Method

A modification to the impulse scheme has been proposed by Skeel and co-workers that is designed to compensate for the inaccuracies due to the use of impulses to approximate a slowly varying force [46]. This *mollified* impulse–MTS scheme retains symplecticity by using a conservative “mollified” force. This is accomplished by substituting $E_{\text{slow}}(\mathcal{A}(X))$ for the slow energy function $E_{\text{slow}}(X)$ where $\mathcal{A}(X)$ is a time-averaging function. More

specifically, the averaged positions $\mathcal{A}(X)$ are to be the result of averaging the given positions X over vibrations due to the fast force $\nabla E_{\text{fast}}(X)$. The effect is to

$$\text{replace } \nabla E_{\text{slow}}(X) \quad \text{with} \quad \mathcal{A}_X(X)^T \nabla E_{\text{slow}}(\mathcal{A}(X)), \quad (2.9)$$

where $\mathcal{A}_X(X)$ is a sparse Jacobian matrix. As an example of averaging, let $X_a(t)$ be the solution of an *auxiliary* problem $M\ddot{X}_a = -\nabla E_{\text{fast}}(X_a)$ with $X_a(0) = X$, $\dot{X}_a(0) = 0$, where the integration is done by the same Verlet integrator used for E_{fast} , and let the average be calculated from the *LongLinearAverage* formula [120]

$$\mathcal{A}(X) = \frac{1}{\Delta t} \int_0^{2\Delta t} \left(1 - \frac{t}{2\Delta t}\right) X_a(t) dt,$$

where the integral is evaluated using the composite trapezoid rule [46].

This method has been implemented in an experimental version of the program NAMD [97] and tested on a 20-Å-diameter sphere of flexible TIP3P water at 370° K. A switching function [60] was used with a cutoff of 6.5 Å to separate long-range electrostatic forces from the other forces. Shown in Fig. 5 are graphs of total pseudo-energy as a function of t for the impulse method, for *LongLinearAverage*, and for another mollified impulse method *Equilibrium* that defines $\mathcal{A}(X)$ to be a projection onto the equilibrium value of the bond lengths and bond angles [65]. (The pseudo-energy is the actual energy with $E_{\text{slow}}(X)$ replaced with $E_{\text{slow}}(\mathcal{A}(X))$.) The energy rises rapidly for the impulse method but is nearly flat for the *Equilibrium* version of the mollified impulse method. The maximum achievable timestep with this approach is around this value (e.g., 6.25 fs) if a drift of a few percent per nanosecond is tolerated.

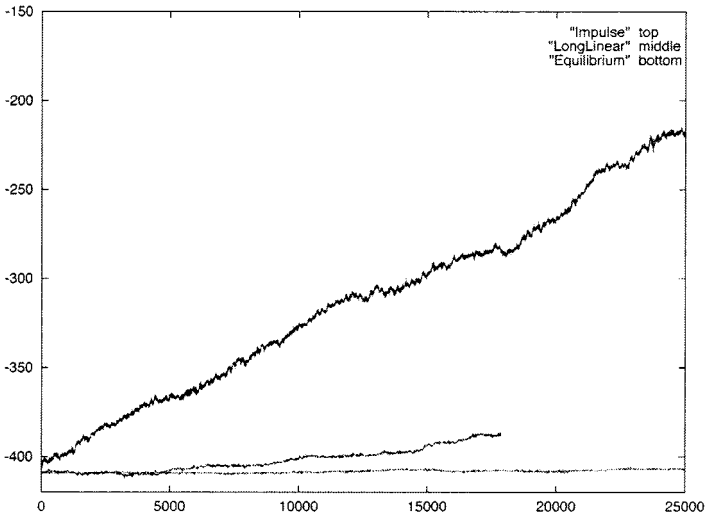


FIG. 5. Total pseudo-energy (in kcal/mol) vs simulation time (in fs) for the time-averaging and *Equilibrium* mollified impulse variants versus the standard impulse–MTS method. The outer timestep Δt for all methods is 6.25 fs.

2.9. The LN Method for Langevin Dynamics

A different, nonsymplectic approach that can extend Δt to the 100-fs range and produce large overall speedups (e.g., an order of magnitude) is the stochastic LN approach [10, 11] (so called for its origin in a Langevin/normal-mode scheme) [134, 135]. In contrast to the MTS methods above, extrapolation of slow forces rather than impulse is used, but the Langevin heat bath prevents systematic energy drifts. The stochastic nature of the governing equations makes the method appropriate for thermodynamic and configurational sampling, a crucial objective in biomolecular simulations; the coupling strength to the heat bath can be minimized by choosing the damping constant as small as possible, just sufficient to guarantee numerical stability [11, 12, 110]. Choosing γ as small as possible is important for minimizing the altering of the Newtonian modes by the Langevin treatment, an altering which affects more significantly the lower-frequency modes.

The LN algorithm can be described as the following sequence:

LN ALGORITHM (with Force Splitting via Extrapolation).

$$X_r = X$$

$$F_s = -\nabla E_{\text{slow}}(X_r)$$

For $j = 1$ to k_2

$$X_r = X + \frac{\Delta t_m}{2} V$$

$$F_m = -\nabla E_{\text{mid}}(X_r)$$

$$F = F_m + F_s$$

For $i = 1$ to k_1

evaluate the Gaussian force R

$$X \leftarrow X + \frac{\Delta \tau}{2} V$$

$$V \leftarrow (V + \mathbf{M}^{-1} \Delta \tau (F + F_{\text{fast}}(X) + R)) / (1 + \gamma \Delta \tau) \quad (2.10)$$

$$X \leftarrow X + \frac{\Delta \tau}{2} V$$

end

end

In this version, the fast forces are calculated directly. It is also possible to propagate them via a linearization of the equations of motion at the reference point X_r [110]. A derivation and historical perspective of the LN method is given in [113].

A detailed comparison of LN trajectories to explicit Langevin trajectories showed excellent agreement for the proteins bovine pancreatic trypsin inhibitor (BPTI) and lysozyme, and a large water system, in terms of energetic, geometric, and dynamic behavior [11]. Similar results have been obtained for solvated BPTI [110] and solvated DNA dodecamers [D. Strahs, X. Qian, and T. Schlick, work in progress]. Significantly, the outer LN timestep can be extended to 50 fs or more, without notable corruption of the trajectories with respect to single-timestep Langevin trajectories. Furthermore, dynamic behavior, as reflected by spectral densities, computed from the trajectories by cosine Fourier transforms of the velocity autocorrelation functions (procedure described in [110]), shows a good agreement between LN and the explicit Langevin trajectories [11]. As γ is decreased, the Langevin and Newtonian ($\gamma = 0$) trajectories resemble each other more closely (see Fig. 6). This suggests that the parameter γ should be kept as small as possible, just sufficient to guarantee numerical stability of the method. An estimate for such a γ is given in [12] and refined in

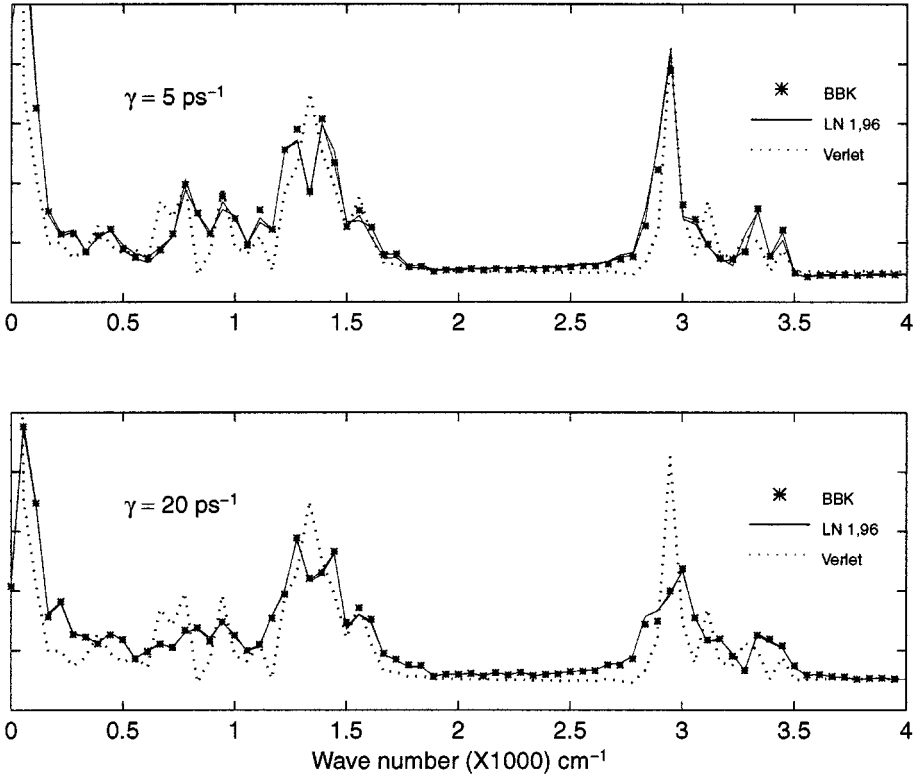


FIG. 6. BPTI spectral densities derived from the cosine Fourier transforms of the velocity autocorrelation function for all atoms for LN versus the Verlet generalization to Langevin dynamics (BBK) as calculated from 6-ps Langevin simulations for two choices of collision frequency $\gamma = 5$ and 20 ps^{-1} . Here $\Delta\tau = 0.5 \text{ fs}$, $\Delta t_m = 3.0 \text{ fs}$, and $\Delta t = \Delta t_m$ for LN 1 and $\Delta t = 96\Delta t_m = 288 \text{ fs}$ for LN 96. Spectral densities for the Verlet method ($\gamma = 0$) are also shown (dots).

[110]. The estimated value based on linear analysis is found to be much larger than that used in practice, $5 \leq \gamma \leq 50 \text{ ps}^{-1}$, which was determined heuristically.

The asymptotic LN speedup can be estimated (in the limit of large k_2) from the following formula, which assumes that the fast forces are computed directly rather than by linearization (see a related derivation in [11]):

$$\text{LN speedup} = k_1 / (r_m + k_1 r_f). \quad (2.11)$$

Using $k_1 = 4$, $r_f = 0.02$, and $r_m = 0.15$, as in Subsection 2.7 we obtain the value of 17, which agrees with our findings [11]. Smaller values of r_m and r_f will increase the speedup. It was shown that moderate values of k_2 can yield nearly the asymptotic speedup, since the medium-force calculations become quite a significant percentage of the total work as k_2 increases and thus limit the realized speedup. Though it might be tempting to increase the medium timestep to reduce the time of these calculations, stability analysis for a model problem shows that the best compromise is to use $\Delta t_m < P/2$ ($P = \text{fastest period}$) and to propagate the medium forces by a midpoint extrapolation approach (via position Verlet), as done in the LN method; the slow forces are best treated by constant extrapolation with a moderate $\Delta t > \Delta t_m$ that yields near-asymptotic speedup. This combination yields good accuracy for the fast motions within the stochastic model and efficient performance overall [110].

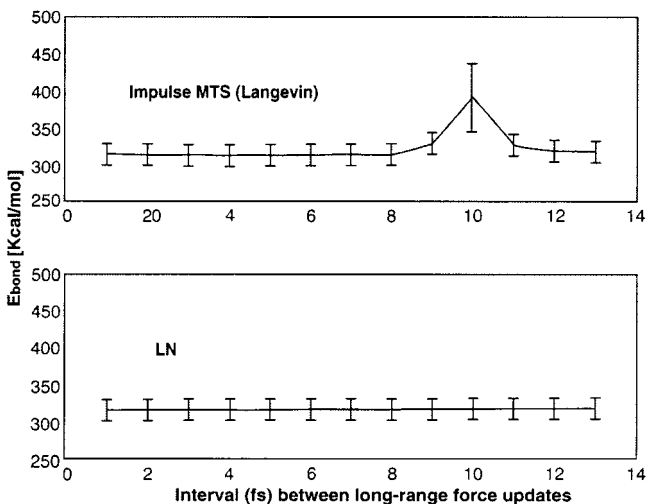


FIG. 7. Average bond energy and variance as calculated for impulse-MTS vs LN for Langevin simulations of BPTI over 2 ps ($\gamma = 20 \text{ ps}^{-1}$) as a function of the outer timestep.

Though results of a linear analysis of resonance cannot be directly applied to complex nonlinear systems (see [110] for specific examples), useful guidelines can be produced [110, 115]. Theoretical analyses of resonances [12, 110] show that impulse methods are generally stable except at integer multiples of half the period of the fastest motion, with the severity of the instability worsening with the timestep; extrapolation methods are generally unstable for the Newtonian model problem, but the instability is bounded for increasing timesteps. This boundedness ensures good long-timestep behavior of extrapolation methods for Langevin dynamics with moderate values of the collision parameter. Analyses of various hybrid impulse/extrapolation MTS schemes [110] and numerical results for linear and nonlinear test problems support the notion that a valuable combination for long-time dynamics simulations of biomolecules is a force splitting scheme using extrapolation, rather than impulses, in combination with a weak coupling to a heat bath to alleviate the intrinsic instability of extrapolative approaches. This combination works especially well for problems with a large disparity of timescales and with class-adapted splitting schemes that yield the best balance between accuracy and stability.

Figure 7 compares the bond energy and associated mean for the protein BPTI (simulated via the complex CHARMM potential) between Langevin LN and impulse-MTS trajectories. We note that the presence of stochasticity alleviates the resonance near half the fastest period, but notable resonance artifacts appear for outer timesteps near the period. This resonance can be avoided by combining the stochastic model with extrapolation-based MTS. The Langevin trajectories are not the same as the Newtonian analogs, and the difference depends on the γ used, but for many thermodynamic and sampling applications they are appropriate. See also Section 4 for a discussion on the pragmatic view that must be taken in biomolecular simulations to balance accuracy with long-time behavior.

3. ELECTROSTATIC INTERACTIONS: FAST EVALUATION METHODS

3.1. Overview

Accurate evaluation of the long-range electrostatic interactions for a biomolecular system remains the most computationally intensive aspect of MD simulations. Historically,

the quadratic ($\mathcal{O}(N^2)$) computational complexity of the all-pairs N -body problem of electrostatics was reduced by using truncated Coulomb potentials, where all interactions past a certain distance were simply ignored. Though acceptable for some simulation experiments, this cutoff approximation leads to unphysical behavior in many situations [131]. An alternative “brute force” quadratic method tailored to specialized parallel hardware [25, 62] increased the size of systems that could be simulated “exactly,” but the quadratic complexity eventually overwhelmed such systems.

New algorithms (such as the *fast multipole algorithm*, FMA), and new implementations of old algorithms (such as the *particle-mesh Ewald* method, PME) have dramatically reduced the cost of performing accurate evaluations of the electrostatic interactions. These fast methods have linear ($\mathcal{O}(N)$) or $\mathcal{O}(N \log(N))$ computational complexity for the N -body problem. For instance, the all-pairs electrostatic interactions in a biomolecular simulation with infinite periodic boundary conditions can now be computed to reasonably high accuracy for the same cost as a traditional 10- to 12-Å cutoff simulation through use of PME [21, 22]. Coupled with modern, inexpensive parallel processing hardware, these new methods enable accurate simulations of large biomolecular systems (10^4 – 10^6 atoms) on machines readily available to biochemists and biophysicists.

This section describes both the multipole-accelerated and the Ewald-derived fast algorithms now used for MD simulations and their implementation on parallel machines.

3.2. Multipole-Accelerated Methods

Multipole-accelerated algorithms first appeared in the astrophysical literature for solving the gravitational N -body problem [5, 8]; the FMA of Greengard and Rokhlin appeared in 1987, placing this class of algorithms on a more rigorous theoretical footing [50]. The FMA was soon adapted to the needs of electrostatic particle simulation for MD [23, 118]. The multipole-accelerated algorithms overcome the $\mathcal{O}(N^2)$ complexity of the N -body problem by using a hierarchy of approximations to represent the effect of increasingly distant groups of particles on a particle of interest (Fig. 8). A convenient power-series

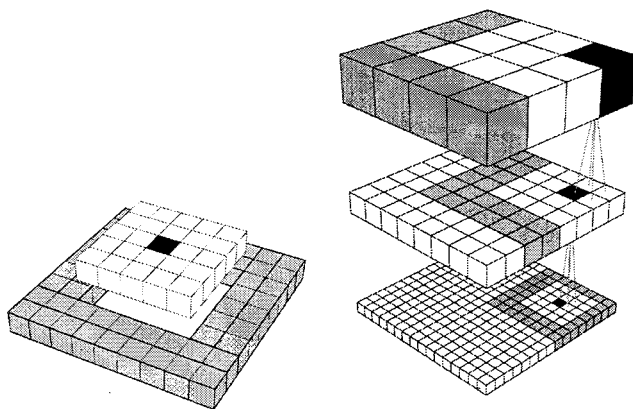


FIG. 8. (Left) Multipole accelerated algorithms all exploit an approximate representation of the effect of groups of distant particles (shaded boxes) on particles of interest (solid box). The interactions of the particles of interest with nearby particles must be computed explicitly since the approximations used for distant particles do not converge at close range. (Right) A hierarchy of these approximations allows increasingly larger groups of particles to be represented by a single power series as the distance from the particles of interest increases.

representation of the Coulomb interaction is used to describe the interaction between groups of charged bodies. The multipole expansion is one suitable power series representation which converges quite rapidly when groups of particles are well separated. This rapid convergence permits the series to be truncated at a relatively small number of terms and still provide an accurate (within known error bounds) representation of the exact potential. For close interparticle separations, the power series representation typically converges very slowly or not at all; thus, Coulomb interactions involving nearby particles must be computed directly via the $1/r$ law (Fig. 8).

Though the multipole expansion has desirable properties, other series expansions can also be used [4, 37, 49, 128]. The basic idea of the FMA and similar algorithms is not restricted to the Coulomb or Newton $1/r$ potentials; the formalism should be applicable to any interaction function which can be approximated by a converging power-series expansion. Similar algorithms have been constructed for $1/r^n$ [39, 80] and Gaussian [32, 80] potentials; a general prescription for handling various types of potentials is given in [80].

The hierarchy of approximations in the multipole-accelerated algorithms leads naturally to the use of hierarchical data structures such as the oct-tree for representing the simulation space [129]. The variants of multipole-accelerated algorithms [20] differ primarily in details of how this tree is traversed and which rules are used to decide when two sets of particles are sufficiently distant from each other to invoke the approximations; details of the power-series representation of the Coulomb potential also vary.

Our multipole code D-PMTA, the distributed parallel multipole tree algorithm, is a message passing code which runs both on workstation clusters and on tightly coupled parallel machines such as the Cray T3D/T3E [20, 21, 104]. Reporting performance numbers for any of these algorithms is always complicated by the fact that there are many variable parameters (simulation and computer dependent) which govern both the accuracy and the efficiency of the simulation. For instance, the optimal number of levels of spatial decomposition for the oct-tree representation can vary for the same simulated biomolecule depending on the workstation used (largely a reflection of whether the square root operation on a particular machine is relatively fast or slow compared to other floating point operations). It is also difficult to compare two different simulations for the same level of accuracy, since the accuracy depends on the size of the test system, the number of terms retained in the multipole expansions, and the number of levels of spatial decomposition employed. Nonetheless, we have developed some standard test cases for judging the quality and efficiency of our codes.

Figure 9 shows the parallel performance of D-PMTA on a moderately large simulation on the Cray T3E for 71,496 atoms (equivalently 23,832 are water molecules). The “low-accuracy” curves reflect 3–4 decimal digits of accuracy in the electrostatic potential and 2–3 decimal digits in the force. The “moderate-accuracy” curves provide 4–5 decimal digits in potential and 3–4 in force. Higher accuracy is possible at the expense of longer runtimes. Accuracy was determined by comparing D-PMTA results to those of an exact quadratic runtime code. In these figures, memory limitations on T3E nodes prevent us from running the complete simulation on one processor, but the scaling behavior from four to eight processors suggests that computing times for one processor are nearly four times longer than those for four processors; thus, the 72 s required on one processor for moderate accuracy is reduced to about 4 s on thirty-two processors.

On a cluster of 300-MHz Intel Pentium II machines with a 100-megabit switched ethernet network and a variant of the Unix operating system, we find serial runtimes slightly faster

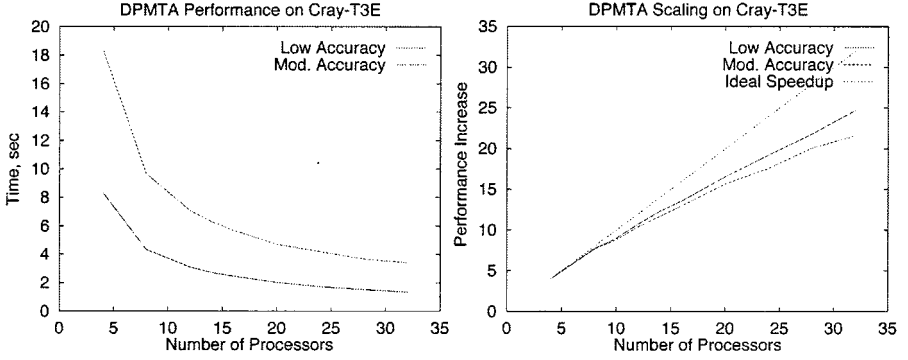


FIG. 9. Performance (left) and scaling behavior (right) of D-PMTA on the Cray T3E for simulating 70,000 particles (see text for details).

than T3E nodes for the above test system. Good performance is still achieved on 16 nodes (speedup of about 12), but the more limited ethernet networking restricts the scalability somewhat compared with the T3E.

3.3. Periodic Systems

Many simulations are best suited for periodic boundary conditions, for example, to represent the properties of bulk media and to reduce surface effects. Infinite periodic boundary conditions are constructed by replicating the original simulation cell out to infinity in all directions. The potential on particle i in the original simulation cell due to all other particles j in the cell and all periodic images of all particles can be written as

$$\Phi_i = \sum_{\substack{j=1 \\ j \neq i \text{ if } \mathbf{n}=(0,0,0)}}^N \sum_{\mathbf{n}} \frac{Q_i Q_j}{r_{i,j_{\mathbf{n}}}}, \quad (3.1)$$

where \mathbf{n} is an integer triple identifying a particular image cell; $\mathbf{n} = (0, 0, 0)$ indicates the “real” simulation cell, and all other cells are identified by their displacement from the central cell, i.e., $\mathbf{n} = (1, 0, 0)$.

Ewald methods consider the periodic boundary case by construction, but it is also possible to extend the multipole-accelerated algorithms to periodic boundary conditions. A single power series can represent the effect of many aggregated copies of the original simulation space on specific particles of interest [81], as shown in Fig. 10. An essentially infinite periodic region (say, a cubic meter) can be simulated for a computational cost of just a few percent over the cost of the multipole methods applied to an original simulation cell of size of order 10 \AA [81].

3.4. Ewald Methods

The Ewald summation was invented in 1921 [42] to permit the efficient computation of lattice sums arising in solid state physics. The desirability of using periodic boundary conditions in MD simulations leads to use of Ewald methods where the entire simulation region is the “unit cell” of an infinite “crystal” lattice of copies of this unit cell.

Ewald recognized that the slowly and conditionally convergent sum of Eq. (3.1) can be recast as two rapidly converging sums, one in *real space* and one in *reciprocal space*. One

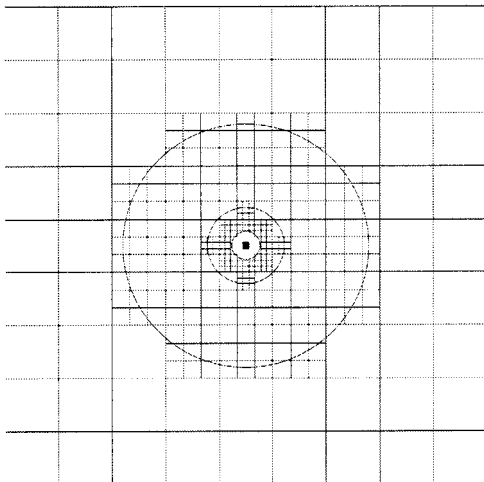


FIG. 10. The periodic multipole algorithm creates exponentially larger aggregates of the original unit cell (small solid box in center) to rapidly construct a large but finite periodic system.

physical explanation of Ewald’s observation is that an auxiliary Gaussian charge distribution can be both added to and subtracted from the original charge distribution. The real space sum is now in terms of the rapidly converging complementary error function $\text{erfc}(r)$ rather than the slowly converging $1/r$ thanks to the Gaussian screening of the charges; the other sum of Gaussian countercharges can be Fourier transformed into a rapidly converging form as well.

Ewald’s formalism reduces the infinite lattice sum to a serial complexity of $\mathcal{O}(N^2)$ in a naive implementation. More efficient $\mathcal{O}(N^{3/2})$ implementations have been known for some time [99]. This complexity has been further reduced to $\mathcal{O}(N \log N)$ in more recent formulations. A review of variants on Ewald summation methods which includes a more complete derivation of the basic method is given in [124].

3.4.1. Particle–Mesh Ewald

One of the most efficient algorithms known for evaluating the Ewald sum is the PME method of Darden *et al.* [33, 34]. The use of Ewald’s “trick” of splitting the Coulomb sum into real space and Fourier space parts yields two distinct computational problems. The relative amount of work performed in real space vs Fourier space can be adjusted within certain limits via a free parameter in the method, but two distinct calculations remain. PME performs the real-space calculation in the conventional manner, evaluating the erfc terms within a cutoff radius. To speed up the Fourier space calculation, PME borrows from the particle–particle, particle–mesh (P^3M) method of Hockney and Eastwood [59] to interpolate all the randomly spaced particles onto a regular mesh. The Fourier sum is then computed via FFT techniques on the regular mesh and the results are interpolated back to the actual particle sites. Another recently reported technique [13] is also derived from the P^3M method and avoids the Fourier sum entirely by using an iterative Poisson solver instead; this method offers the possibility of linear complexity in the number of particles N and is also worthy of further study.

PME originally employed Lagrange interpolation [33]; we have implemented a revised PME formulation which uses B-spline interpolation functions [34]. The smoothness of

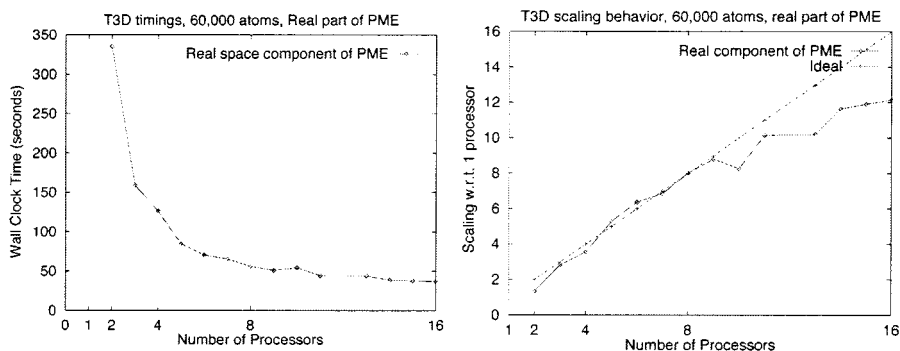


FIG. 11. Performance (left) and scaling behavior (right) of the real space part of PME on the Cray T3D.

B-splines allows the force expressions to be evaluated analytically, with high accuracy, by differentiating the real and reciprocal energy equations rather than using finite differencing techniques.

Our interest has been to parallelize PME; some of our efforts are described in [125]. The three-dimensional (3D) FFT needed by PME is challenging to parallelize effectively on distributed memory parallel computers due to the interprocessor communications requirements of the algorithm and the relatively small size of the FFTs required even for what are considered large molecular simulations in the late 1990s. This contrasts with the direct part of Ewald methods which parallelize extremely well. We first bias the work as much as possible to the real-space term (essentially by increasing the cutoff radius for the error function, which reduces the number of lattice vectors needed for good accuracy in the Fourier term). The real-space sum is easily parallelized with a simple spatial decomposition and appropriate attention to the overlap between adjacent regions due to the cutoff radius. Performance for the real-space contribution alone is given in Fig. 11.

The Fourier sum, involving the 3D FFT, currently can run effectively on perhaps 8 processors in a loosely coupled network-of-workstations environment, and better implementations will probably permit productive use of 16–32 workstations without exotic interconnection hardware. On a more tightly coupled machine such as the Cray T3D/T3E, we obtained fair efficiency on 16 processors, with a measured speedup of about 9, but efficiency falls off fairly rapidly beyond 32 processors. Recently improved FFT libraries from the vendor are expected to have better performance. Our initial production implementation was targeted for a small workstation cluster, so we only parallelized the real-space part, relegating the Fourier component to serial evaluation on the “master” processor. The 16% of the serial work attributable to the Fourier sum limits our potential speedup on 8 processors to 6.25, a value we are able to approach quite closely.

3.5. Performance Comparison

The results in the prior subsections represent isolated multipole and PME timings. A complete MD simulation involves much more. In addition to computing the short-range interactions from bonding forces, etc., the particle positions and velocities need to be updated each timestep. Additionally, efficient MD programs use force splitting, as described in the previous section, so it is not necessary to compute the Coulomb interactions (and thus

TABLE 2
Serial Run Times for MD Simulations with Various Electrostatic Solvers

Solver	Runtime (s)	Accuracy (see text)	Code type
8 Å cutoff	100		
12 Å cutoff	286		
16 Å cutoff	645		
PME-custom	289	Low	Custom PME, tightly integrated
PME-custom	375	Moderate	Custom PME, tightly integrated
PME-library	390	Low	Library PME, loosely integrated
PME-library	408	Moderate	Library PME, loosely integrated
DPMTA	472	Low	Periodic multipole, 4 terms in the multipole expansions
DPMTA	688	Moderate	Periodic multipole, 8 terms in the multipole expansions
DPMTA	1248	High	Periodic multipole, 12 terms in the multipole expansions

multipole or Ewald) at each small timestep. Within a long-time interval, the long-range interactions are often interpolated from prior estimates. Still, the long-range force evaluation remains the dominant computational cost of MD simulations.

Table 2 compares the running time of complete MD simulations (here a box containing 23,832 water molecules, or equivalently 71,496 atoms) performed with the different periodic solvers, serial versions, as well as traditional methods utilizing cutoff radii. The simulation was run for 1000 timesteps of 2 fs each, with a full Coulomb evaluation performed every 6 inner timesteps (except for the cutoff cases). The MD program used is SigmaX, developed by Hermans and co-workers [57], which has been modified to accept both PME and our multipole codes. “Low accuracy” represents 3–4 significant decimal digits in potential, or 2–3 figures in force. “Moderate” is 4–5 digits in potential and 3–4 digits in force. “High” is 5–6 digits in potential and 4–5 decimal digits in force. Accuracy cannot be determined in a comparable way for the cutoff cases and so is omitted.

The salient comparisons are between the low-accuracy PME library case (390 s) and the low-accuracy DPMTA case (472 s). The multipole code uses the periodic option for a fair comparison with Ewald methods (the cutoff code implements a nearest image convention approximation of a periodic system). PME is clearly faster, running in only two-thirds the time of DPMTA. This is in agreement with the findings of Petersen [100] and Esselink [41], who find Ewald methods to be faster than multipole methods for systems of fewer than about 100,000 particles.

The very fast PME custom times resulted from rewriting our PME code to share data structures with SigmaX, thus avoiding the overhead involved in calling the original portable library version of PME. This custom version also used a table lookup for erfc which reduced accuracy somewhat but significantly improved speed. These optimizations combined for a 20% performance improvement at the cost of portability and extra development time. The multipole codes would also benefit from a tighter integration with SigmaX, but this would be more difficult to implement for the multipole than for the PME codes. These tightly integrated custom PME codes compare well with cutoff calculations, giving runtime comparable to a 12-Å cutoff simulation with much more thorough treatment of the electrostatic

interactions. On different machines, we have examples of PME time equivalence to 10-Å cutoff simulations. A more detailed discussion of these results can be found in [21, 22].

The serial speed advantage of PME must be traded against its poorer parallel performance; with 32 or more processors DPMTA is a much better choice. Work continues on improving the performance of both the multipole and Ewald solvers, as well as including such effects as polarizability in the Coulomb potentials.

4. COMBINING MTS WITH EWALD OR ACCELERATED MULTIPOLES: A CASE STUDY

4.1. Efficiency versus Accuracy

The MTS and Ewald methods described in the previous two sections can be combined for simulations of biomolecules in aqueous solution. The underlying aim is to achieve the most accurate simulations possible at a minimum cost of computer time. The desired accuracy depends on the questions being addressed. For example, for applications which seek to determine distributions of structural parameters or potentials of mean force, adequate sampling of configuration space is important and, hence, a highly accurate integration scheme is not required. On the other hand, simulations that are used to calculate response functions, relaxation times, or rates of conformational transitions must employ accurate integration schemes. A pragmatic view also recognizes that the molecular mechanics force field itself is approximate and, hence, the systematic errors introduced by the discretization or the Ewald representation of electrostatic interactions need not be much smaller than these errors if dynamic exploration of large systems is the objective.

4.2. Implementation

The impulse-MTS [47, 126] and LN [11] methods have been implemented in the SigmaX program [57]. Following other such approaches [44, 137], five separate integration timesteps can be selected for calculation of the following terms:

1. bond-stretching forces,
2. angle-bending and out-of-plane twisting forces,
3. nonbonded interactions for pairs $\{i, j\}$ separated by a distance $d_{i,j} \leq R_{cs}$,
4. nonbonded interactions for pairs $\{i, j\}$ separated by a distance $R_{cs} < d_{i,j} \leq R_{cl}$, and
5. Ewald or multipole-derived forces due to pairs separated by $d_{i,j} > R_{cl}$.

The fifth term is computed as the *difference* of the complete Ewald or multipole-derived forces and the electrostatic forces for the third and fourth terms. This protocol employs “distance classes” [47] and avoids using peculiar features of either method [102], so that the two methods (Ewald and fast-multipole) are exchangeable within the same program. Each of these forces successively changes more slowly with time and thus can be updated less often; as calculation of each takes successively longer, the MTS scheme can reduce the total processor time. A common practice in MD is to calculate terms 3 and 4 above using precomputed atom pairlists. The expensive pairlist calculation can be performed less frequently than the Ewald-sum calculations (see Table 3, where $R_{cs} = 6 \text{ \AA}$, $R_{cl} = 10 \text{ \AA}$, and the pairlists are computed every two PME evaluations).

TABLE 3
Practical MD Protocol Used in a Simulation of T4 Lysozyme in Water

Interval (fs)	Force or task	Atom pairs	CPU time (s) ^a	Time/2 fs
2	Angles, bonds ^b		0.23	0.23
2	Short: $d \leq 6 \text{ \AA}^c$	896,000	0.59	0.59
6	Medium: $6 < d \leq 10 \text{ \AA}^c$	3,240,000	2.21	0.73
12	Long: PME ^d		3.38	0.56
24	Pairlist		8.50	0.72
	Sum		[14.91]	2.83

^a On a single SGI R10000 processor.

^b Bond lengths are constrained with Shake.

^c The interparticle separation distance is denoted by d .

^d Particle mesh Ewald sum, computed with a mesh size of approximately 1 \AA , and parameters as for “modest” accuracy, cf. Table 2.

4.3. Application to Lysozyme

Performance is illustrated with a simulation of a mutant of the protein T4-lysozyme in aqueous solution. The mutant has one or two xenon atoms bound in a hydrophobic cavity [40, 96]. The protein contains 162 amino acid residues corresponding to 2605 protein atoms and one or two Xe atoms. With 5835 water molecules, there are 20,116 atoms in total. The system is contained in a rectangular box with dimensions $62.4 \times 56.6 \times 56.6 \text{ \AA}^3$ and simulated using periodic boundary conditions. The 6-ns simulation had two main goals: (1) sample the distribution of the bound Xe atoms, in order to compute a simulated electron density for comparison with X-ray diffraction studies of the complex [103]; (2) observe conformational changes, in particular the slow motions of the active site cleft, which varies significantly with crystal form and mutation [136]; related conformational changes had already been captured in simulations [51].

4.4. Integration Protocol and Performance

The protocol adopted for this simulation used the impulse–MTS scheme [47, 126], and details are shown in Table III. Bond-length forces were not calculated, as the bonds were kept at fixed lengths using the SHAKE method. (This is necessary because the empirical SPC water force field had been developed for rigid bonds [17, 56].)

The used MTS scheme results in a roughly equal distribution of the load over four components repeated at successively longer intervals: 2, 6, 12, and 24 fs. The MTS protocol results in a saving of processor time by a factor of over 5. The net figure of 2.8 s/step (equivalent to 62 ps/1 day) on a single SGI R10000 processor is also recovered by dividing the total reported processor time by the number of steps. Thus, the work per step beyond force calculations is negligible.

This protocol leads to a slow increase of the total energy with time at a rate of $10^{-3} \text{ kcal/mol/ps/degree of freedom}$ [137]. In contrast, a simulation that omits the Ewald sum shows a 50 times faster rise of the energy. The energy drift can be further reduced by choosing smaller time steps ($\leq 4 \text{ fs}$) [44]. This presents a difficult choice between accurate integration on one side and extension of the simulation to longer times (or to multiple trial trajectories) on the other. In particular, long-term trajectories exhibit global motions that can be observed

repeatedly. Simulations with a less accurate protocol can be considered as exploratory; if interesting global motions are observed, they can be confirmed by more accurate, but slower, simulations. In fact, such calculations in progress for the lysozyme system described here have progressed to where the conformational change described below has begun but not completed (after 1.5 ns).

4.5. Lysozyme Dynamics Results

The 6-ns simulation performed isothermally, with coupling to a heat bath following the Berendsen method [16], reveals that the protein has locally stable domains connected flexibly [Geoff Mann, University of North Carolina]. Lysozyme consists of three principal domains: an N-terminal lobe, a C-terminal lobe, and a connecting helix. During the course of the simulation, these domains retain their overall conformations as reflected in hydrogen bonds, backbone dihedral angles, and covariances of atomic positions. However, the separation between the opposing tips of the two lobes changes repeatedly by up to 8 Å during the simulation. Interconversions are captured between “open” and “closed” states of the protein, schematically shown in Fig. 12. The change is reversible and occurs on a nanosecond timescale. The opening/closing motion is required for the enzymatic function of cleaving polysaccharide chains, and the range of conformations corresponds to that found in crystal structures [136].

In earlier calculations without the Ewald sum and a single cutoff of 8 Å, the conformation of this protein had proved unstable, with unfolding of helices becoming apparent after approximately 1 ns of simulation time. By demonstrating stability of local structure combined with large-scale global motions over long timescales, this result extends the findings of the improved accuracy that results from inclusion of all electrostatic forces with Ewald or multipole-accelerated code, reported by Pedersen and Darden [33, 83, 84, 131–133]. With use of the MTS scheme, the additional computational cost of adding the Ewald calculation is only a small fraction of the total cost.



FIG. 12. (Left) Open state of the T4 lysozyme molecule observed after approximately 3.6 ns of simulation. (Right) Closed state observed after approximately 4.4 ns. The protein backbone is shown, as well as the side chains of Thr21 and 142, and the distance between the C α atoms of these two residues is indicated. The contact between the side chains in the closed state at 4.4 ns is very similar to that found in the crystal structure that was used as a starting point for these simulations.

5. BIOMOLECULAR SIMULATION PROGRAMS: PERFORMANCE ENHANCEMENTS VIA PARALLELIZATION

5.1. Challenges in MD Codes

The large computational power required to model biomolecules of current interest poses a challenge. Even when the raw computational abilities provided by parallel supercomputers enable such simulations, efficient exploitation by parallel software is often difficult. As an example, consider the DNA-estrogen receptor (ER) complex, described in Subsection 1.2, that consists of about 36,600 atoms. To understand the effect of the ER binding on the conformation of the target DNA, simulation lengths of several nanoseconds are required. Each simulation step of 1 fs takes about 6 s on an HP workstation. (Note that this figure uses a newer version of NAMD than that used to obtain the results of Section 1.2.) However, the million steps required to simulate a nanosecond imply more than 80 days of computation on a single workstation. Employing multiple processors working in parallel can speed up the simulations. Yet, the dependency of the results at each step on the previous one confines parallelism to a single timestep. Parallelizing a 6-s computation in a scalable manner is difficult because of the large communication-to-computation ratio it typically entails. The structural complexity of biomolecular simulations, caused in part by the combination of bonded and nonbonded forces, and the irregular spatial distribution of atoms, further exacerbates the problem. Electrostatic force computations, in particular, constitute a large fraction of the overall computation. Simulating constrained dynamics is a formidable challenge for parallelization and is not implemented in NAMD except for the option to constrain the lengths of covalent bonds to hydrogens.

5.2. Spatial Decomposition

Several decomposition strategies have been investigated for this problem. Pure force decomposition strategies store all the atom's information at the central processor and only distribute the sets of atom pairlists to multiple processors. Although reasonably effective for a small number of processors, this strategy fails to scale up to larger parallel machines. Distributing atoms to the various processors, say based on the index of each atom, improves scalability but complicates communication requirements, as each processor may need to communicate to all the others during each timestep. The strategy used in the NAMD program developed at UIUC [72] is that of spatial decomposition: atoms are dynamically partitioned into cubical regions of space. By choosing the dimensions of the cubes to be larger than the interaction (cutoff) radius, we ensure that each cube communicates only with its neighboring cubes. Though this strategy requires a small amount of extra communication to migrate atoms to different cubes as they move through space, the overall saving in communication is worthwhile due to the increase in locality.

5.3. Load Balance

In general, multiple cubes may be assigned to individual processors, and load balance is achieved by selecting the set of cubes that reside on each processor (i.e., selecting the mapping of cubes to processors). A technique used in the most recent version of NAMD is to

separate the mapping of electrostatic force computations from that of the cubes themselves, such that each pair of neighboring cubes (called *patches* in NAMD) gives rise to a force computation object. These objects are mapped to the processors, under the control of the dynamic load balancing strategy.

5.4. Relation between Numerical Algorithm and Parallel Code

A parallel molecular code must deal with two sets of issues: those arising from the numerical algorithm, and those dealing with management of parallelism. In earlier versions of NAMD, the intertwining of the parallel code and the numerical algorithm made the program somewhat difficult to understand and modify. A solution to this problem involves using user-level *threads*. Each *patch* has a single control thread associated with it. The function associated with this thread, called the *sequencer*, essentially describes the life cycle of a patch. As a result, it succinctly specifies the numerical algorithm used. At various points during a timestep, this function waits for other computations to finish. Therefore, this thread and the code associated with it are the only functions that need to be understood and possibly recoded for any modification of the numerical algorithm of the MD program. As a result, for a researcher interested in modifying the functional behavior of NAMD, parallelism must be considered only to the extent of realizing that whenever the thread waits, it might be waiting for a combination of local as well as remote data. Only researchers interested in improving the *parallel performance* of NAMD need to be concerned with the rest of the code, that deals with interprocessor communication.

5.5. Implementation and Performance

NAMD is implemented in C++, a modern programming language with excellent support for modularity. Though Fortran is the common language for scientific applications, and C++ is thought to be slower than Fortran, C++ compilers have improved in recent years, approaching the efficiency of Fortran. Use of a careful programming style improves the efficiency further. This involves avoiding inefficient features of C++ such as virtual method invocations in inner loops, and using in-line functions whenever possible. The use of C++ also appears to encourage programmers to use more sophisticated and efficient data structures. As a result of all these factors, NAMD performs faster than some of the Fortran-based programs for equivalent tasks.

Most of the parallel modules in NAMD are implemented using Charm++ [70], a C++-based parallel programming library. The parallelization strategy employed in NAMD, as described above, leads to having multiple patches and force objects on each processor. Each of these objects must wait until the dependent data have arrived from remote processors. Data-driven execution supported in Charm++ allows such dependencies to be implemented without making the processor stop to await the data. Instead of allowing individual computations to issue a *receive* instruction, which might block the computations on the processor entirely, data-driven execution employs a *scheduler* on each processor. The scheduler works with a pool of messages that have already arrived on the processor (or have been locally generated). Repeatedly, this scheduler selects a message from the pool and activates the particular object which can process it. In addition to ensuring that no object or computation can block the entire processor, this technique also helps to modularize parallel codes.

However, not all modules in NAMD were implemented in Charm++. DMPTA, the fast multipole library discussed in Section 3, has been implemented independently in PVM.

TABLE 4
NAMD2 Performance on the Cray T3E for ER-ERE
(36,576 atoms), 12 Å Cutoff

Number of processors	Time per step (s)	Speedup
2	6.12	(1.97)
4	3.10	3.89
8	1.60	7.54
16	0.810	14.9
32	0.397	30.3
64	0.212	56.8
128	0.123	97.9

The sequencers associated with each patch were implemented using threads, while much of the initialization code was implemented using a simple message passing style. Integrating such disparate modules in a single program was a challenge. The Converse framework [71] developed at the University of Illinois is used to support multiple languages and libraries in a single application. Languages and libraries such as Charm++ and PVM are implemented on top of Converse. Converse allows programmers to use the most appropriate parallel programming model (embodied in a parallel coordination language such as Charm++ or libraries such as PVM or MPI) for each module of the program, independently. In addition, Converse also supports portability across a wide variety of parallel machines, with both shared and distributed memory. Converse includes library modules that implement its basic parallel operations using each of the supported parallel machine. Thus, programs implemented using Converse run unchanged on all such machines. As a result, NAMD currently runs on the CRAY T3D and T3E, IBM SP2-3, SGI Origin 2000, the ASCI Red machine, and networks of workstations.

Table 4 shows performance of NAMD on the estrogen receptor DNA complex on the Cray T3E. Speedup is excellent even for the largest number of processors (128), with a speedup factor of 97.9. Reducing an 80-day computation to less than a day is an enormous practical benefit.

6. CRYSTALLOGRAPHIC REFINEMENT: ACCELERATED TECHNIQUES

6.1. Overview

The analysis of X-ray diffraction data generally requires sophisticated computational procedures culminating in structure refinement and validation. These procedures can be formulated as constrained or restrained nonlinear optimization of a target function expressed in terms of positional coordinates and B-factors of covalently (e.g., protein) and noncovalently (e.g., water molecules) linked atoms. The ultimate goal is to simultaneously optimize the agreement of this atomic model with the observed diffraction data and with *a priori* chemical information. In this section, we focus on enhancement of crystallographic refinement by molecular dynamics-based simulated annealing.

The target function used for optimization typically depends on several atomic parameters, including atomic coordinates. The form is complex because a large number of adjustable

parameters (typically at least three times the number of atoms in the system) are involved. The target function thus contains many local minima, and finding the global minimum cannot be assured by any current method. The existence of many local minima confounds local gradient optimization techniques such as conjugate gradient or least-squares methods [54]. That is, such gradient descent schemes find a minimum near the starting point and therefore cannot sample molecular conformation space well to find the most optimal solution. Simulated annealing [73] as an optimization technique is thus a viable alternative. Unlike gradient descent methods, simulated annealing can cross barriers between minima and thus explore a greater volume of parameter space to find better models (deeper and/or wider minima). Following its introduction to crystallographic refinement [30], there have been major improvements of the original method in four principal areas: the measure of model quality, the search of the parameter space, the target function, and the modeling of conformational variability [29].

6.2. The Target Function

In crystallographic refinement, the global minimum of the target function

$$E_{\text{refinement}}(X) = E_{\text{chem}}(X) + w_{\text{xray}} E_{\text{xray}}(X) \quad (6.1)$$

is sought, where X represents the parameters of an atomic model, such as atomic coordinates. E_{chem} contains empirical information about chemical interactions, as shown in Eq. (2.1); it is a function of all atomic positions, describing covalent (bond lengths, bond angles, torsion angles, chiral centers, and planarity of aromatic rings) and nonbonded (intramolecular as well as intermolecular and symmetry-related) interactions [58]. $E_{\text{xray}}(X)$ describes the difference between observed and calculated data, and w_{xray} is a weight appropriately chosen to balance the gradients (with respect to atomic parameters) arising from the two terms.

The conventional form of E_{xray} consists of the crystallographic residual E_{LSQ} , defined as the sum over the squared differences between the observed ($|F_o|$) and calculated ($|F_c|$) structure factor amplitudes for a particular atomic model,

$$E_{\text{xray}}(X) = E_{\text{LSQ}}(X) = \sum_{hkl} (|F_o| - k|F_c|)^2, \quad (6.2)$$

where hkl denotes the indices of the reciprocal lattice points of the crystal, and k is a relative scale factor.

Reduction of E_{LSQ} can result from improvement in the atomic model, but also from an accumulation of systematic errors in the model or from fitting noise in the data [119]. It has been suggested that this problem is partly attributed to the fact that the least-squares residual is poorly justified when the model is far away from the correct one or is incomplete [98]. An improved target for macromolecular refinement can be obtained using the more general maximum-likelihood formulation [1, 31, 98, 106]. The goal of the maximum-likelihood method is to use a statistically correct treatment of errors in the model and in the observed diffraction data for the refinement procedure. The effects of model errors (incorrectly placed and missing atoms) on the calculated structure factors are first quantified with σ_A values, which correspond roughly to the fraction of each structure factor that is expected to be correct. However, overfitting of the diffraction data causes the model bias to be underestimated and undercorrected in the σ_A values. The effect of this overfitting can be

reduced by “cross validating” σ_A values, i.e., by computing them from a randomly selected test set which is excluded from the summation on the right hand-side of Eqs. (6.2) and (6.3) [28, 74]. The expected values of $\langle |F_o| \rangle$ and the corresponding variance (σ_{ML}^2) are derived from σ_A , the observed $|F_o|$, and calculated $|F_c|$ [98]. These quantities can be used to derive a Gaussian approximation of the general maximum-likelihood target function

$$E_{x\text{-ray}}(X) = E_{ML} = \sum_{hkl \in \text{working set}} \frac{1}{\sigma_{ML}^2} (|F_o| - \langle |F_o| \rangle)^2. \quad (6.3)$$

In order to achieve an improvement over the least-squares residual (Eq. (6.2)), cross validation was found to be essential [1] for the computation of σ_A and its derived quantities in Eq. (6.3).

6.3. Simulated Annealing for Conformational Searching

Annealing denotes a physical process wherein a solid is heated until all particles randomly arrange themselves in a liquid phase and then is cooled slowly so that all particles arrange themselves in the lowest energy state. By formally defining the target E (Eq. (6.1)) to be the equivalent of the potential energy of the system, we simulate the annealing process [73]. There is no guarantee that simulated annealing will find the global minimum (except in the case of an infinitely long search) [79]. Still, compared to conjugate-gradient minimization, simulated annealing achieves more optimal solutions by allowing motion against the gradient. The likelihood of *uphill* motion is determined by a control parameter referred to as *temperature*. The higher the temperature, the more likely it is that simulated annealing will overcome barriers. The input temperature typically has no physical meaning and is merely a tool for overcoming barriers of the target function.

The simulated annealing algorithm requires a generation mechanism to create a Boltzmann distribution at a given temperature T . Simulated annealing also requires an annealing schedule, that is, a sequence of temperatures $T_1 \geq T_2 \geq \dots \geq T_l$ at which the Boltzmann distribution is computed. Implementations of the generation mechanism differ in the way they generate a transition, or move, from one set of parameters to another so as to be consistent with the Boltzmann distribution at the given temperature. The two most widely used generation mechanisms are Metropolis Monte Carlo [94] and molecular dynamics simulations. For X-ray crystallographic refinement, molecular dynamics has proved to be extremely successful [30].

6.4. Torsion-Angle Molecular Dynamics for Refinement

A suitably chosen set of atomic parameters can be viewed as generalized coordinates that are propagated numerically in time by the classical (Hamilton) equations of motion, as described in the second section of this article (see Eq. (2.2)). Initial velocities are usually assigned randomly from a Maxwell distribution at the appropriate temperature. Assignment of different initial velocities will produce a somewhat different structure after simulated annealing. By performing several refinements with different initial velocities one can therefore improve the likelihood of success of simulated-annealing refinement. Furthermore, this improved sampling can be used to determine discrete disorder and conformational variability.

Although Cartesian MD (i.e., employing flexible bond lengths and bond angles) places restraints on bond lengths and bond angles (through E_{chem} , Eq. (6.1)), an alternative constrained formulation may be preferred [36]. This is supported by the observation that the deviations from ideal bond lengths and bond angles are usually small in X-ray crystal structures. Indeed, fixed-length constraints have been applied to crystallographic refinement by least-squares minimization [36]. It is only recently, however, that efficient and robust algorithms have become available for MD in torsion-angle space [6, 7, 68, 93, 107].

Rice and Brunger [107] have applied an approach that retains the Cartesian-coordinate formulation of the target function and its derivatives with respect to atomic coordinates so that calculation remains relatively straightforward and topology independent. In this formulation, however, the expression for the acceleration becomes a function of positions and velocities. Iterative equations of motion for constrained dynamics in this formulation can be derived and solved numerically [6, 7, 68, 107]. This approach is distinct from SHAKE and RATTLE procedures outlined in the second section in that it integrates the equations exactly in reduced variable space. Practical experience [A.T.B., unpublished] suggests that the approximate SHAKE method is not suitable for crystallographic or solution NMR refinement. The alternative approach, in contrast, appears to be robust and to have a significantly increased radius of convergence in crystallographic refinement at high temperatures compared to standard Cartesian MD [107].

6.5. Temperature Control

Simulated annealing requires the control of the temperature during molecular dynamics. The current temperature (i.e., at step n or time $n\Delta t$) of the simulation (T^n) is computed from the instantaneous kinetic energy

$$E_{\text{kin}}^n = \frac{1}{2}(V^n)^T \mathbf{M}(V^n) \quad (6.4)$$

of the MD simulation. (As before, we use V for the collective velocity vector, \mathbf{M} for the corresponding mass matrix, and T for vector transposes.) The corresponding kinetic temperature is then

$$T^n = 2E_{\text{kin}}^n / (N_F k_B), \quad (6.5)$$

where N_F is the number of degrees of freedom (e.g., three times the number of atoms). One commonly used approach to control the temperature of the simulation consists of coupling the equations of motion to a heat bath. A friction-like term can be added to the right-hand side of Eq. (2.2) as for Langevin dynamics in the form [16]

$$-\gamma \mathbf{M}V^n (1 - T/T^n) \quad (6.6)$$

to adjust the temperature. Unlike standard Langevin dynamics, the friction coefficient γ may be negative, as well as positive, in this context, depending on whether the current temperature is lower or higher than the target temperature. The magnitude of γ is determined by the ratio T/T^n .

6.6. Why Does Simulated Annealing Refinement Work?

In crystallographic refinement, the global minimum of the specified target function is sought by searching for the conformation or conformations of the molecule that best fit the diffraction data while also maintaining reasonable bonded and nonbonded interactions. Paradoxically, the very reasons that make simulated annealing such a powerful refinement technique—the ability to overcome barriers in the target energy function—might seem to prevent it from working at all; if it crosses barriers so easily, what allows it to stay in the vicinity of the global minimum? Certainly, the system's inertia at the specified temperature allows it to cross energy barriers of the corresponding target function. This target temperature must be large enough to overcome smaller barriers but low enough to ensure that the system does not escape from the global minimum if reached.

While the temperature itself is a global parameter of the system, temperature fluctuations arise principally from local conformational transitions—for example, from an amino acid sidechain falling into the correct orientation. These local changes tend to lower the value of the target function, thus increasing the kinetic energy, and hence the effective temperature, of the system. Once the temperature coupling (Eq. (6.6)) has removed this excess kinetic energy through “heat dissipation,” the reverse transition is unlikely, since it would require a localized increase in kinetic energy where the conformational change occurred in the first place. Temperature coupling retains a sufficient amount of kinetic energy to allow local conformational rearrangements but does not supply enough energy to allow escape from the global minimum. This explains the observation that on average the agreement with the experimental data improves rather than worsens with simulated annealing.

6.7. Refinement Examples

Many examples have shown that simulated annealing refinement starting from reasonable initial models (obtained by standard crystallographic techniques) produces significantly better final solutions compared to those produced by least-squares or conjugate-gradient minimization [27, 30, 45, 78, 107]. In recent tests [1, 107], arbitrarily “scrambled” models were generated from an initial model of α -amylase inhibitor built using experimental phase information from multiple isomorphous replacement. The scrambling was achieved by generating MD simulations at 600 K on the basis of the chemical potential energy alone (i.e., without reference to the X-ray diffraction data). These scrambled structures were then energy minimized to yield chemically reasonable starting models.

We have shown that these models represent realistic test cases for phasing by molecular replacement: the scrambled structures contained sufficient information to solve the crystal structure through an extensive molecular replacement search [1]. For a given model, simulated annealing always outperforms gradient minimization. We also showed that simulated annealing with the least-squares target has a larger radius of convergence than minimization with maximum likelihood and that this convergence is further improved by the use of the maximum-likelihood target. For the most scrambled model an average phase improvement—compared to the least-squares residual—of more than 15° is obtained. The resulting structures are very close to the published crystal structure. Similar results were obtained for a test case at lower (2.8 Å) resolution and for two real cases involving new crystal structures starting with an initial model built into a poor electron density map [1, 75].

7. FUTURE PERSPECTIVES

It is impressive to look back 20 years and appreciate how rapidly the field of biomolecular modeling has progressed from the first dynamic simulation of a small protein in vacuum in the late 1970s. Developments in theory, experiment, and computer technology have made simulations of larger systems over longer times with greater accuracy possible. Animated atomic models of enzymes in their solvent milieu can now be visualized on desktop workstations to discern key motional roles; DNA/protein complexes can be scrutinized to identify local structural recognition elements; glimpses into complex biomolecular networks or reactions, such as lipid bilayers, motor proteins, and photoprocesses, are emerging; and further insights into experimental observations related to elastic polymer deformations, transcription regulation, or mutagenic tendencies of carcinogenic aromatic–DNA compounds are possible. For many processes that require consideration of electronic rearrangements, the development of improved quantum/classical and fast quantum modeling approaches is likely to lead to significant advances on the horizon. Coarse-graining models and the combining of atomic-level and polymer-level models are also promising directions for bridging the gap between experimental and theoretical temporal and spatial frames.

Given these and other exciting algorithmic and application activities in the field, it is not unrealistic to expect that computer modeling of biomolecules will become a general, accurate tool for studying macromolecular structure and function in the near future. On a basic level, modeling programs will likely become more accessible to the general scientific community by requiring less specialized skills in molecular manipulation and simulation protocol. Impressive, user-friendly graphical interfaces, in particular, aid modelers, significantly. Still, the success of modeling on a fine level will likely require expert experience and, perhaps, some art as well as science, so as to determine the molecular and cellular complexities that are essential to consider versus those that can be neglected or approximated.

The growing number of high-resolution solved structures, coupled with the huge amount of incoming genomic sequence data, provide a strong impetus to improve modeling reliability and to develop new application areas. This unprecedented wealth of biological information is driving the development of knowledge-based tools and new fields of scientific challenges. Bioinformatics, structural genomics, and functional genomics are prominent examples of these new areas. Predicting structure from sequence has been a long-time appreciated quest; now it is also natural to ask modelers about the likely *function* associated with an experimentally determined structure. A combination of database-rooted analysis and modeling is likely to be a particularly powerful duo for these tasks. Information-driven modeling might even suggest to experimentalists which sequences to study in the hope of producing new fold motifs.

In addition, theoretical approaches are proving invaluable in cataloging structural information, and in the near future this is likely to extend to functional-based libraries. These databases in turn are providing structures for validating current structure-prediction tools, improving fold classification, and using structures for rational drug design. Indeed, the pharmaceutical industries are benefiting tremendously from chemical libraries, both for organic molecules and for biological macromolecules; enhanced tools for similarity and diversity sampling and for bioactivity prediction are accelerating the rate at which new potential drugs enter the phase of clinical trials. Together, knowledge-based science and biomolecular simulations might ultimately prove to be reliable for making structural and

functional predictions ahead of experiment, an important goal of the field. Such predictions will be particularly valuable for systems that are not amenable to crystallization and other experimental techniques.

ACKNOWLEDGMENTS

We thank the National Science Foundation for its support (NSF BIR 94-23827EQ) and acknowledge the contribution of many young people in all of the authors' groups involved in this research. We are grateful to Wei Xu for technical assistance in preparation of this paper.

REFERENCES

1. P. D. Adams, N. S. Pannu, R. J. Read, and A. T. Brunger, Cross-validated maximum likelihood enhances crystallographic simulated annealing refinement, submitted for publication.
2. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Univ. Press, New York, 1990).
3. H. C. Andersen, Rattle: A 'velocity' version of the SHAKE algorithm for molecular dynamics calculations, *J. Comput. Phys.* **52**, 24 (1983).
4. C. R. Anderson, An implementation of the fast multipole algorithm without multipoles, *SIAM J. Sci. Stat. Comput.* **13**, 923 (1992).
5. A. W. Appel, An efficient program for many-body simulation, *SIAM J. Sci. Stat. Comput.* **6**, 85 (1985).
6. D.-S. Bae and E. J. Haug, A recursive formulation for constrained mechanical system dynamics. Part I. Open loop systems, *Mech. Struct. Mach.* **15**, 359 (1987).
7. D.-S. Bae and E. J. Haug, A recursive formulation for constrained mechanical system dynamics. Part II. Closed loop systems, *Mech. Struct. Mach.* **15**, 481 (1988).
8. J. Barnes, and P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, *Nature* **324**, 446 (1986).
9. E. Barth, K. Kuczera, B. Leimkuhler, and R. D. Skeel, Algorithms for constrained molecular dynamics, *J. Comput. Chem.* **16**, 1192 (1995).
10. E. Barth, M. Mandziuk, and T. Schlick, A separating framework for increasing the timestep in molecular dynamics, in *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, edited by W. F. van Gunsteren, P. K. Weiner, and A. J. Wilkinson (ESCOM, Leiden, The Netherlands, 1997), Vol. III, Chap. 4, p. 97.
11. E. Barth and T. Schlick, Overcoming stability limitations in biomolecular dynamics I. Combining force splitting via extrapolation with Langevin dynamics in LN, *J. Chem. Phys.* **109**, 1617 (1998).
12. E. Barth and T. Schlick, Extrapolation versus impulse in multiple-timestepping schemes. II. Linear analysis and applications to Newtonian and Langevin dynamics, *J. Chem. Phys.* **109**, 1633 (1998).
13. J. V. L. Beckers, C. P. Lowe, and S. W. deLeeuw, An iterative PPPM method for simulating Coulombic systems on distributed memory parallel computers, *Mol. Sim.* **20**, 283 (1998).
14. M. Ben-Nun and T. J. Martínez, Nonadiabatic molecular dynamics: Validation of the multiple spawning method for a multi-dimensional problem, *J. Chem. Phys.* **108**, 7244 (1998).
15. M. Ben-Nun, F. Molnar, H. Lu, J. C. Phillips, T. J. Martínez, and K. Schulten, Quantum dynamics of retinal's femtosecond photoisomerization in bacteriorhodopsin, *Faraday Discuss.* **110**, 447 (1998).
16. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **81**, 3684 (1984).
17. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans, Interaction models for water in relation to protein hydration, in *Intermolecular Forces*, edited by B. Pullman (Reidel, Dordrecht, The Netherlands, 1981), p. 331.
18. J. J. Biesiadecki and R. D. Skeel, Dangers of multiple-time-step methods, *J. Comput. Phys.* **109**, 318 (1993).
19. T. Bishop, R. D. Skeel, and K. Schulten, Difficulties with multiple timestepping and the fast multipole algorithm in molecular dynamics, *J. Comput. Chem.* **18**, 1785 (1997).

20. J. A. Board, Jr., A. John, Z. S. Hakura, W. D. Elliott, and W. T. Rankin, Scalable variants of multipole-accelerated algorithms for molecular dynamics applications, in *Proceedings, Seventh SIAM Conference on Parallel Processing for Scientific Computing* (SIAM, Philadelphia, 1995), p. 295.
21. J. A. Board, Jr., A. John, C. W. Humphries, C. G. Lambert, W. T. Rankin, and A. Y. Toukmaji, Ewald, and multipole methods for periodic N -body problems, in *Proceedings, Eighth SIAM Conference on Parallel Processing for Scientific Computing* (SIAM, Philadelphia, 1997). [CD-ROM]
22. J. A. Board, Jr., C. W. Humphries, C. G. Lambert, W. T. Rankin, and A. Y. Toukmaji, Ewald and multipole methods for periodic N -body problems, in *Computational Molecular Dynamics: Challenges, Methods, Ideas—Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, edited by P. Deuffhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, and R. D. Skeel, Lecture Notes in Computational Science and Engineering (Springer-Verlag, Berlin/New York, 1988), Vol. 4, p. 159.
23. J. A. Board, Jr., A. John, J. W. Causey, J. F. Leathrum, A. Windemuth, and K. Schulten, Accelerated molecular dynamics simulation with the fast multipole algorithm, *Chem. Phys. Lett.* **198**, 89 (1992).
24. J. A. Board, Jr., L. V. Kalé, K. Schulten, R. D. Skeel, and T. Schlick, Modeling biomolecules: Larger scales, longer durations, *IEEE Comput. Sci. Eng.* **1**, 19 (1994).
25. K. Boehncke, H. Heller, H. Grubmüller, and K. Schulten, Molecular dynamics simulation on a systolic ring of transputers, in *Transputer Research and Applications 3* (IOS Press, Washington, DC, 1990), p. 83.
26. C. L. Brooks III, M. Karplus, and B. M. Pettitt, *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics, Advances in Chemical Physics* (Wiley, New York, 1988), Vol. LXXI.
27. A. T. Brunger, Crystallographic refinement by simulated annealing: Application to a 2.8 Å resolution structure of aspartate aminotransferase, *J. Mol. Biol.* **203**, 803 (1988).
28. A. T. Brunger, The free R value: A novel statistical quantity for assessing the accuracy of crystal structures, *Nature* **355**, 472 (1992).
29. A. T. Brunger, P. D. Adams, and L. M. Rice, New applications of simulated annealing in X-ray crystallography and solution NMR, *Structure* **5**, 325 (1997).
30. A. T. Brunger, J. Kuriyan, and M. Karplus, Crystallographic R factor refinement by molecular dynamics, *Science* **235**, 458 (1987).
31. G. Bricogne, A multisolution method of phase determination by combined maximization of entropy and likelihood. III. Extension to powder diffraction data, *Acta Crystallogr. A* **47**, 803 (1991).
32. J. C. Burant, M. C. Strain, G. E. Scuseria, and M. J. Firsich, Kohn–Sham analytic energy second derivatives with the Gaussian very fast multipole method (GvFMM), *Chem. Phys. Lett.* **258**, 45 (1996).
33. T. Darden, D. York, and L. Pedersen, Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems, *J. Chem. Phys.* **98**, 10089 (1993).
34. T. Darden, U. Essmann, H. Lee, L. Perera, M. Berkowitz, and L. Pedersen, A smooth particle mesh Ewald method, *J. Chem. Phys.* **103**, 8577 (1995).
35. P. Deuffhard, M. Dellnitz, O. Junge, and Ch. Schütte, *Computation of Essential Molecular Dynamics by Subdivision Techniques. I. Basic Concepts*, Technical Report SC 96-45, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1996.
36. R. Diamond, A real-space refinement procedure for proteins, *Acta. Crystallogr. A* **27**, 436 (1971).
37. H. Q. Ding, N. Karasawa, and W. A. Goddard III, The reduced cell multipole method for Coulomb interactions in periodic systems with million-atom unit cells, *Chem. Phys. Lett.* **196**, 6 (1992).
38. Y. Duan and P. A. Kollman, Pathways to a protein folding intermediate observed in 1-microsecond simulation in aqueous solution, *Science* **282**, 740 (1998).
39. W. D. Elliott, *Multipole Algorithms for Molecular Dynamics Simulation on High Performance Computers*, Ph.D. dissertation, Department of Electrical Engineering, Duke University, Durham, NC, 1995.
40. A. E. Eriksson, W. A. Baase, J. A. Wozniak, and B. W. Matthews, A cavity-containing mutant of T4 lysozyme is stabilized by buried benzene, *Nature* **355**, 371 (1992).
41. K. Esselink, A comparison of algorithms for long-range interactions, *Comput. Phys. Commun.* **87**, 375 (1995).
42. P. P. Ewald, Die Berechnung optischer und elektrostatischer Gitterpotentiale, *Ann. Phys.* **64**, 253 (1921).

43. D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, San Diego, 1996).
44. F. Figuerido, R. M. Levy, R. Zhou, and B. J. Berne, Large scale simulation of macromolecules in solution: Combining the periodic fast multipole method with multiple time step integration, *J. Chem. Phys.* **106**, 9835 (1997).
45. M. Fujinaga, P. Gros, and W. F. van Gunsteren, Testing the method of crystallographic refinement using molecular dynamics, *J. Appl. Crystallogr.* **22**, 1 (1989).
46. B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel, Long-time-step methods for oscillatory differential equations, *SIAM J. Sci. Comput.* **20**, 930 (1998).
47. H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions, *Mol. Sim.* **6**, 121 (1991).
48. H. Grubmüller, B. Heymann, and P. Tavan, Ligand binding and molecular mechanics calculation of the streptavidin–biotin rupture force, *Science* **271**, 997 (1996).
49. P. M. W. Gill, A new expansion of the Coulomb interaction, *Chem. Phys. Lett.* **270**, 193 (1997).
50. L. Greengard and V. Rokhlin, A fast algorithm for particle simulation, *J. Comput. Phys.* **73**, 325 (1987).
51. B. L. de Groot, S. Hayward, D. M. F. van Aalten, A. Amadei, and H. J. C. Berendsen, Domain motions in bacteriophage T4 lysozyme: A comparison between molecular dynamics and crystallographic data, *Proteins Struct. Funct. Genet.* **31**, 116 (1998).
52. W. F. van Gunsteren, P. K. Weiner, and A. J. Wilkinson, Eds., *Computer Simulation of Biomolecular Systems* (ESCOM, Leiden, The Netherlands, 1996), Vol. 3.
53. W. F. van Gunsteren, Constrained dynamics of flexible molecules, *Mol. Phys.* **40**, 1015 (1980).
54. P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization* (Academic Press, London, 1983).
55. J. Gullingsrud, R. Braun, and K. Schulten, Reconstructing potentials of mean force through time series analysis of steered molecular dynamics simulations, *J. Comput. Phys.* **151**, 190 (1999).
56. J. Hermans, H. J. C. Berendsen, W. F. van Gunsteren, and J. P. M. Postma, A consistent empirical potential for water–protein interactions, *Biopolymers* **23**, 1513 (1984).
57. J. Hermans, R. H. Yun, J. Leech, and D. Cavanaugh, Sigma documentation, available at <http://femto.med.unc.edu/SIGMA/>.
58. W. A. Hendrickson, Stereochemically restrained refinement of macromolecular structures, *Meth. Enzymol.* **115**, 252 (1985).
59. R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
60. D. E. Humphreys, R. A. Friesner, and B. J. Berne, A multiple-time-step molecular dynamics algorithm for macromolecules, *J. Phys. Chem.* **98**, 6885 (1994).
61. W. F. Humphrey, A. Dalke, and K. Schulten, VMD—Visual molecular dynamics, *J. Mol. Graphics* **14**, 33 (1996).
62. T. Ito, T. Ebisuzaki, J. Makino, and D. Sugimoto, A special-purpose computer for gravitational many-body systems: GRAPE-2, *Publ. Astron. Soc. Jpn.* **43**, 547 (1991).
63. S. Improta, A. Politou, and A. Pastore, Immunoglobulin-like modules from titin I-band: Extensible components of muscle elasticity, *Structure* **4**, 323 (1996).
64. B. Israilewitz, S. Izrailev, and K. Schulten, Binding pathway of retinal to bacteriodopsin: A prediction by molecular dynamics simulations, *Biophys. J.* **73**, 2972 (1997).
65. J. Izaguirre, S. Reich, and R. D. Skeel, Longer time steps for molecular dynamics, *J. Chem. Phys.*, in press.
66. S. Izrailev, S. Stepaniants, M. Balsera, Y. Oono, and K. Schulten, Molecular dynamics study of unbinding of the avidin–biotin complex, *Biophys. J.* **72**, 1568 (1997).
67. S. Izrailev, S. Stepaniants, B. Israilewitz, D. Kosztin, H. Lu, F. Molnar, W. Wriggers, and K. Schulten, Steered molecular dynamics, in *Computational Molecular Dynamics: Challenges, Methods, Ideas—Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, edited by P. Deuffhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, and R. D. Skeel, Lecture Notes in Computational Science and Engineering (Springer-Verlag, Berlin/New York, 1998) Vol. 4, p. 36.

68. A. Jain, N. Vaidehi, and G. Rodriguez, A fast recursive algorithm for molecular dynamics simulations, *J. Comput. Phys.* **106**, 258 (1993).
69. A. Jonas, Reconstitution of high-density lipoproteins, *Meth. Enzymol.* **128**, 553 (1986).
70. L. V. Kalé and S. Krishnan, Charm++: Parallel programming with message-driven objects, in *Parallel Programming using C++*, edited by G. V. Wilson and P. Lu (MIT Press, Cambridge, MA, 1996), p. 175.
71. L. V. Kalé, M. Bhandarkar, N. Jagathesan, S. Krishnan, and J. Yelon, Converse: An interoperable framework for parallel programming, in *Proceedings of the 10th International Parallel Processing Symposium Honolulu, Hawaii, April 1996*, p. 212.
72. L. V. Kalé, R. Skeel, R. Brunner, M. Bhandarkar, A. Gursoy, N. Krawetz, J. Phillips, A. Shinzaki, K. Varadarajan, and K. Schulten, NAMD2: Greater scalability for parallel molecular dynamics, *J. Comput. Phys.* **151**, 283 (1999).
73. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671 (1983).
74. G. J. Kleywegt and A. T. Brunger, Cross-validation in crystallography: Practice and applications, *Structure* **4**, 897 (1996).
75. J. Koepke, X. Hu, C. Münke, K. Schulten, and H. Michel, The crystal structure of the light harvesting complex II (B800-850) from *Rhodospirillum rubrum*, *Structure* **4**, 581 (1996).
76. D. Kosztin, T. C. Bishop, and K. Schulten, Binding of the estrogen receptor to DNA: The role of waters, *Biophys. J.* **73**, 557 (1997).
77. D. Kosztin, S. Izrailev, and K. Schulten, Unbinding of retinoic acid from its receptor studied by steered molecular dynamics, *Biophys. J.* **76**, 188 (1999).
78. J. Kuriyan, A. T. Brunger, M. Karplus, and W. A. Hendrickson, X-ray refinement of protein structures by simulated annealing: Test of the method on myohemerythrin, *Acta. Crystallogr. A* **45**, 396 (1989).
79. P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications* (Reidel, Dordrecht, Holland, 1987).
80. C. G. Lambert, Multipole-Based Algorithms in *Molecular Biophysics and Non-parametric Statistics*, Ph.D. dissertation, Department of Computer Science, Duke University, Durham, NC, 1997.
81. C. G. Lambert, T. A. Darden, and J. Board, A multipole-based algorithm for efficient calculation of forces and potentials in macroscopic periodic assemblies of particles, *J. Comput. Phys.* **126**, 274 (1996).
82. A. R. Leach, *Molecular Modelling Principles and Applications* (Longman, Singapore, 1996).
83. H. Lee, T. A. Darden, and L. G. Pedersen, Accurate crystal molecular dynamics simulations using particle mesh Ewald method, *Chem. Phys. Lett.* **243**, 229 (1995).
84. H. Lee, T. A. Darden, and L. G. Pedersen, Molecular dynamics simulation studies of a high-resolution Z-DNA crystal, *J. Chem. Phys.* **103**, 3830 (1995).
85. B. Leimkuhler and R. D. Skeel, Symplectic numerical integrators in constrained Hamiltonian systems, *J. Comput. Phys.* **112**, 117 (1994).
86. R. L. Loncharich and B. R. Brooks, The effects of truncating long-range forces on protein dynamics, *Proteins Struct. Funct. Genet.* **6**, 32 (1989).
87. R. H. Lozier, R. A. Bogomolni, and W. Stoerkenius, Bacteriorhodopsin: A light-driven proton pump in *Halobacterium halobium*, *Biophys. J.* **15**, 955 (1975).
88. H. Lu, B. Israilewitz, A. Krammer, V. Vogel, and K. Schulten, Unfolding of titin immunoglobulin domains by steered molecular dynamics simulation, *Biophys. J.* **75**, 662 (1998).
89. M. Mandziuk and T. Schlick, Resonance in the dynamics of chemical systems simulated by the implicit-midpoint scheme, *Chem. Phys. Lett.* **237**, 525 (1995).
90. J. F. Marko, Twist and shout (and pull): Molecular chiropractors undo DNA, *Proc. Natl. Acad. Sci. USA* **94**, 11770, 1997.
91. S.-J. Marrink, O. Berger, P. Tieleman, and F. Jähnig, Adhesion forces of lipids in a phospholipid membrane studied by molecular dynamics simulations, *Biophys. J.* **74**, 931 (1998).
92. T. J. Martínez, M. Ben-Nun, and R. D. Levine, Molecular collision dynamics on several electronic states, *J. Phys. Chem. A* **101**, 6389 (1997).

93. A. M. Mathiowetz, A. Jain, N. Karasawa, and W. A. Goddard, Protein simulations using techniques suitable for very large systems: The cell multipole method for nonbonded interactions and the Newton–Euler inverse mass operator method for internal coordinate dynamics, *Proteins Struct. Funct. Genet.* **20**, 227 (1994).
94. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
95. J. A. McCammon and S. C. Harvey, *Dynamics of Proteins and Nucleic Acids* (Cambridge Univ. Press, Cambridge, MA, 1987).
96. A. Morton, W. A. Baase, and B. W. Matthews, Energetic origins of ligand binding in an interior nonpolar cavity of T4 lysozyme, *Biochemistry* **34**, 8564 (1995).
97. M. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. Kalé, R. D. Skeel, and K. Schulten, NAMD—A parallel, object-oriented molecular dynamics program, *J. Supercomputing Appl.* **10**, 251 (1996).
98. N. S. Pannu and R. J. Read, Improved structure refinement through maximum likelihood, *Acta. Crystallogr. A* **52**, 659 (1996).
99. J. W. Perram, H. G. Petersen, and S. W. deLeeuw, An algorithm for the simulation of condensed matter which grows as the $\frac{3}{2}$ power of the number of particles, *Mol. Phys.* **65**, 875 (1988).
100. H. G. Petersen, Accuracy and efficiency of the particle mesh Ewald method, *J. Chem. Phys.* **103**, 3668 (1995).
101. J. C. Phillips, W. Wriggers, Z. Li, A. Jonas, and K. Schulten, Predicting the structure of apolipoprotein A-I in reconstituted high density lipoprotein disks, *Biophys. J.* **73**, 2337 (1997).
102. P. Procacci and M. Marchi, Taming the Ewald sum in molecular dynamics simulations of solvated proteins via a multiple time step algorithm, *J. Chem. Phys.* **104**, 3003 (1996).
103. M. L. Quillin, W. A. Baase, and B. W. Matthews, Binding of small electron-dense ligands in large protein cavities, in *International Union of Crystallography XII Congress and General Assembly* (American Crystallographic Association, Seattle, 1996), p. C215.
104. W. T. Rankin and J. Board, A portable distributed implementation of the parallel multipole tree algorithm, in *Proceedings, Fourth IEEE International Symposium on High Performance Distributed Computing*, (IEEE Computer Society Press, Los Alamitos, CA, 1995), p. 17.
105. W. Rankin and J. Board, A portable distributed implementation of the parallel multipole tree algorithm, in *IEEE Symposium on High Performance Distributed Computing*, in press; Duke University Technical Report 95-002.
106. R. J. Read, Structure-factor probabilities for related structures, *Acta Crystallogr. Ser. A* **46**, 900 (1990).
107. L. M. Rice and A. T. Brunger, Torsion angle dynamics: Reduced variable conformational sampling enhances crystallographic structure refinement, *Proteins Struct. Funct. Gen.* **19**, 277 (1994).
108. M. Rief, M. Gautel, F. Oesterhelt, J. M. Fernandez, and H. E. Gaub, Reversible unfolding of individual titin immunoglobulin domains by AFM, *Science* **276**, 1109 (1997).
109. J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes, *J. Comput. Phys.* **23**, 327 (1977).
110. A. Sandu and T. Schlick, Masking resonance artifacts of force-splitting methods for biomolecular simulations by extrapolative Langevin dynamics, *J. Comput. Phys.* **151**, 74 (1999).
111. J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman & Hall, London, 1994).
112. T. Schlick, *Modeling and Minimization Techniques for Predicting Three-Dimensional Structures of Large Biological Molecules*, Ph.D. thesis, New York University, Courant Institute of Mathematical Sciences, New York, Oct. 1987.
113. T. Schlick, Some failures and successes of long-timestep approaches for biomolecular simulations, in *Computational Molecular Dynamics: Challenges, Methods, Ideas—Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, edited by P. Deuffhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, and R. D. Skeel, Lecture Notes in Computational Science and Engineering (Springer-Verlag, Berlin/New York, 1998), Vol. 4, p. 227.
114. T. Schlick and A. Brandt, A multigrid tutorial with applications to molecular dynamics, *IEEE Comput. Sci. Eng.* **3**, 78 (1996).
115. T. Schlick, M. Mandziuk, R. D. Skeel, and K. Srinivas, Nonlinear resonance artifacts in molecular dynamics simulations, *J. Comput. Phys.* **139**, 1 (1998).

116. T. Schlick and C. S. Peskin, Comment on: The evaluation of LI and LIN for dynamics simulations, *J. Chem. Phys.* **103**, 9888 (1995).
117. K. Schulten, W. Humphrey, I. Logunov, M. Sheves, and D. Xu, Molecular dynamics studies of bacteriorhodopsin's photocycles, *Israel J. Chem.* **35**, 447 (1995).
118. J. Shimada, H. Kaneko, and T. Takada, Performance of fast multipole methods for calculating electrostatic interactions in biomacromolecular simulations, *J. Comput. Chem.* **15**, 28 (1994).
119. A. M. Silva and M. G. Rossmann, The refinement of southern bean mosaic virus in reciprocal space, *Acta Crystallogr. Ser. B* **41**, 147 (1985).
120. R. D. Skeel and J. Izaguirre, The five femtosecond time step barrier, in *Computational Molecular Dynamics: Challenges, Methods, Ideas—Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, edited by P. Deuffhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, and R. D. Skeel, Lecture Notes in Computational Science and Engineering (Springer-Verlag, Berlin/New York, 1998), Vol. 4, p. 318.
121. P. J. Steinbach and B. R. Brooks, New spherical-cutoff methods for long-range forces in macromolecular simulation, *J. Comput. Chem.* **15**, 667 (1994).
122. S. Stepaniants, S. Izrailev, and K. Schulten, Extraction of lipids from phospholipid membranes by steered molecular dynamics, *J. Mol. Model.* **3**, 473 (1997).
123. W. B. Streett, D. J. Tildesley, and G. Saville, Multiple time step methods in molecular dynamics, *Mol. Phys.* **35**, 639 (1978).
124. A. Y. Toukmaji and J. Board, Ewald sum techniques in perspective: A survey, *Comput. Phys. Commun.* **95**, 73 (1996).
125. A. Y. Toukmaji, D. Paul, and J. A. Board, Jr., Distributed particle-mesh Ewald: A parallel Ewald summation method, in *Proceedings, International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'96)* (CSREA Press, 1996), p. 33.
126. M. E. Tuckerman, B. J. Berne, and G. J. Martyna, Reversible multiple time scale molecular dynamics, *J. Chem. Phys.* **97**, 1990 (1992).
127. L. Verlet, Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* **159**, 98 (1967).
128. H. Y. Wang and R. LeSar, An efficient fast-multipole algorithm based on an expansion in the solid harmonics, *J. Chem. Phys.* **104**, 4173 (1996).
129. M. S. Warren and J. K. Salmon, A parallel hashed oct-tree N-body algorithm, in *Supercomputing '93 Proceedings* (IEEE Computer Society Press, Washington, DC, 1993), p. 12.
130. M. Watanabe and M. Karplus, Simulations of macromolecules by multiple time-step methods, *J. Phys. Chem.* **99**, 5680 (1995).
131. D. M. York, T. A. Darden, and L. G. Pedersen, The effect of long-ranged electrostatic interactions in simulations of macromolecular crystals: A comparison of Ewald and truncated list methods, *J. Chem. Phys.* **99**, 8345 (1994).
132. D. York, A. Wlodawer, L. Pedersen, and T. Darden, Atomic level accuracy in simulations of large protein crystals, *Proc. Natl. Acad. Sci. USA* **91**, 8715 (1994).
133. D. York, W. Yang, H. Lee, T. Darden, and L. Pedersen, Toward the accurate modeling of DNA: The importance of long-range electrostatics, *J. Am. Chem. Soc.* **117**, 5001 (1995).
134. G. Zhang and T. Schlick, LIN: A new algorithm combining implicit integration and normal mode techniques for molecular dynamics, *J. Comput. Chem.* **14**, 1212 (1993).
135. G. Zhang and T. Schlick, The Langevin/implicit-Euler/normal-mode scheme (LIN) for molecular dynamics at large time steps, *J. Chem. Phys.* **101**, 4995 (1994).
136. X. J. Zhang, J. A. Wozniak, and B. W. Matthews, Protein flexibility and adaptability seen in 25 crystal forms of T4 lysozyme, *J. Mol. Biol.* **36**, 307 (1995).
137. R. Zhou and B. J. Berne, A new molecular dynamics method combining the reference system propagator algorithm with a fast multipole method for simulating proteins and other complex systems, *J. Chem. Phys.* **103**, 9444 (1995).