Simulated Annealing of a Protein in a Continuum Solvent by Multiple-Time-Step Molecular Dynamics

Darryl D. Humphreys,[†,‡] Richard A. Friesner,* and Bruce J. Berne

Department of Chemistry and the Center for Biomolecular Simulation, Columbia University, New York, New York, 10027

Received: January 24, 1995[⊗]

In this paper we present a computationally efficient means of performing simulated annealing on atomic level protein structures. The method which is based upon a Trotter factorization of the classical Liouville propagator is employed in a series of simulated annealing studies of a small protein using a standard molecular mechanics type potential as well as a continuum approximation to include the effects of solvation. Preliminary results are presented for the performance of the potential using standard force field parametrizations in attempts to distinguish native-like structures from those that are distinctly non-native. Low-energy structures with large rms deviations from the minimized X-ray structure were found, thereby suggesting the possibility that the potentials employed in this work may not be of sufficient accuracy to distinguish the native protein structure.

1. Introduction

It is now commonly believed that the native state of a protein is given by the state corresponding to the global minimum of the molecules free energy, although it has alternatively been suggested that there may exist many equivalent accessible "native-like" states. Assuming, for the moment, that a unique native state exists, formally the problem may then be viewed as one of global optimization, whereby one attempts to find the minimum of a given "cost function". In the present instance this cost function would correspond to the proteins free energy or, as is sometimes the case in practice, taken to be its potential energy. In attempting to solve this problem from a theoretical point of view, one is then faced with two principal difficulties. The first difficulty concerns the accuracy of the potential energy function employed. Often, in the study of biomolecular systems, a molecular mechanics type of potential is used. In these models one typically treats the atomic nuclei as classical point particles moving on an approximate Born–Oppenheimer potential energy surface. Although these models have been shown to give reasonable results when applied to the case of small molecules, it remains to be seen as to what degree they provide a valid description for significantly larger systems, such as those of proteins. The second principal obstruction in seeking a theoretical solution to the global optimization problem pertains to the difficulty of efficaciously sampling configuration space in a realistic amount of computer time. This is primarily due to the fact that standard methods of sampling configuration space become inefficient for systems containing multiple minima as well as for those that include disparate time and length scales. In order to address these concerns, in this paper, we investigate the application of efficient numerical integration techniques we have recently developed for molecular dynamics simulations to the problem of simulated annealing of atomic level protein structures. This was done in an attempt to reduce the required CPU time for these simulations and, thus, to enhance our ability to explore the use of slower annealing schedules. The method is employed in simulated annealing studies of a small protein using standard molecular mechanics type potentials which include a continuum approximation to represent the effects of solvation. Preliminary results are presented for the performance of the potential using standard force field parametrizations in attempts to distinguish native-like structures from those that are distinctly non-native. Interestingly, low-energy structures with large rms deviations from the minimized X-ray structure were found, thereby suggesting the possibility that the potentials employed in this work may not be of sufficient accuracy to distinguish the native protein structure.

2. Algorithm for Simulated Annealing by Multiple-Time-Step Molecular Dynamics

The method of simulated annealing was originally introduced by Kirkpatrick, Gelatte, and Vecchi[1] as a practical approach to the problem of global optimization. Essentially, the algorithm seeks to minimize an appropriate cost function by solving an analogous problem in statistical mechanics. Here the cost function may be thought of as a rough potential energy surface with numerous barriers and local minima. Since one is seeking the global minimum of this surface, one requires an algorithm that is able to ergodically explore the configuration space without becoming trapped in one of the many local minima. In simulated annealing, this is achieved by setting the initial temperature to a high value so that potential energy barriers are easily crossed and the system is allowed to explore the available configuration space. If the temperature is then decreased slowly enough, the system should eventually relax into the optimal configuration corresponding to the global minimum of the potential energy surface. With regards to the protein-folding problem, we seek the configuration that is associated with the global minimum of the molecules' free energy or, as is commonly used in practice, potential energy surface. An obstacle arises, however, if the ultimate goal is to find the optimal protein configuration at an atomic level of resolution. For a system the size of a protein, the computational expense of determining the global minimum of the potential energy surface becomes prohibitive for typical atomic level molecular mechanics type potentials using standard methods.

Recently, we have developed efficient numerical integration techniques for molecular dynamics simulations of macromolecular systems based upon the Trotter factorization of the

[†] In partial fulfillment of the Ph.D. in the Department of Chemistry, Columbia University.

[‡] Present address: Department of Chemistry, University of Pennsylvania, Philadelphia, Pennsylvania 19104–6323

[⊗] Abstract published in *Advance ACS Abstracts,* May 1, 1995.

0022-3654/95/2099-10674$09.00/0

Liouville propagator.[2] It was found that this method was considerably more efficient than that of the standard velocity Verlet algorithm when applied to the problem of simulating a small protein in the microcanonical ensemble. Here we explore the question of whether similar ideas might be beneficial when applied to the simulated annealing of a protein as a means to determine the configuration corresponding to the global minimum of its potential energy. In order to do this, we have incorporated the algorithm into the MACROMODEL molecular modeling package.[3] By default this package allows the user to perform simulated annealing by one of two methods, including both molecular dynamics and stochastic dynamics, each of which is based upon the leap-frog numerical integration algorithm. Thus, for the purpose of convenience, we choose to implement our RESPA (reference system propagation algorithm)-based method using a leap-frog integrator, as opposed to the previously adopted velocity Verlet form.

2.1. Leap-Frog Factorization of the Liouville Propagator. Recall that the Liouville operator for a system of N degrees of freedom in Cartesian coordinates may be expressed as

$$iL = [..., H] = \sum_{i=1}^{N}\left[\dot{q}_i\frac{\partial}{\partial q_i} + F_i(q)\frac{\partial}{\partial p_i}\right] \tag{1}$$

where $[..., ...]$ is the Poisson bracket, and H is the Hamiltonian. The state of the system at any time t may then be obtained by

$$\Gamma(t) = U(t)\,\Gamma(0) \tag{2}$$

where $U(t)$ is the classical time evolution operator

$$U(t) = e^{itL} \tag{3}$$

and where

$$\Gamma(t) = \{q(t), p(t)\} \tag{4}$$

$\{q,p\}$ are the set of generalized coordinates and associated conjugate momenta. Expressing the Liouville operator as a sum of two terms,

$$iL = iL_1 + iL_2 \tag{5}$$

allows us to make use of the Trotter theorem,[4]

$$e^{it(L_1+L_2)} = [e^{i(t/P)(L_1+L_2)}]^P = [e^{i(\Delta t/2)L_2}e^{i\Delta tL_1}e^{i(\Delta t/2)L_2}]^P + O(\Delta t^3) \tag{6}$$

where $\Delta t \equiv t/P$. Let us then define a discrete time propagator $G(\Delta t)$,

$$G(\Delta t) = e^{i(\Delta t/2)L_2}e^{i\Delta tL_1}e^{i(\Delta t/2)L_2} \tag{7}$$

such that

$$\Gamma(\Delta t) = G(\Delta t)\,\Gamma(0) \tag{8}$$

It has been shown[5] that the choice

$$iL_1 \equiv \dot{q}\frac{\partial}{\partial q} \tag{9}$$

$$iL_2 \equiv F(q)\frac{\partial}{\partial p} \tag{10}$$

is equivalent to the velocity Verlet integration algorithm. Consider now inserting an identity operator I of the form

$$I = e^{+i(\Delta t/2)L_2}e^{-i(\Delta t/2)L_2} = e^0 = 1 \tag{11}$$

into eq 8,

$$\Gamma(\Delta t) = \{q(\Delta t), p(\Delta t)\} = G(\Delta t)I\{q(0), p(0)\}$$
$$[e^{i(\Delta t/2)L_2}e^{i\Delta tL_1}e^{i(\Delta t/2)L_2}]e^{+i(\Delta t/2)L_2}e^{-i(\Delta t/2)L_2}\{q(0), p(0)\}$$

Upon rearranging, we have

$$= e^{i(\Delta t/2)L_2}[e^{i\Delta tL_1}e^{i\Delta tL_2}]e^{-i(\Delta t/2)L_2}\{q(0), p(0)\} \tag{12}$$

We may further simplify the expression by defining the state $\{q(\Delta t),p(\Delta t/2)\}$ such that

$$\{q(\Delta t), p(\Delta t)\} = e^{i(\Delta t/2)L_2}\{q(\Delta t), p(\Delta t/2)\} \tag{13}$$

where

$$\{q(\Delta t), p(\Delta t/2)\} \equiv [e^{i\Delta tL_1}e^{i\Delta tL_2}]e^{-i(\Delta t/2)L_2}\{q(0), p(0)\} \tag{14}$$

By substituting eq 9 and eq 10 for L_1 and L_2, and making use of the fact that

$$e^{\epsilon(\partial/\partial x)}f(x) = f(x + \epsilon) \tag{15}$$

upon acting to the right, we have

$$p(\Delta t/2) = [p(0) - (\Delta t/2)F(0)] + \Delta tF(0) \tag{16}$$

The leap-frog form[6] of the Verlet algorithm can be shown to generate exactly the same trajectory as that given by the velocity Verlet method provided one uses the initial condition[7]

$$p(-\Delta t/2) = p(0) - (\Delta t/2)F(x(0)) \tag{17}$$

Substituting eq 17 into eq 16 gives

$$p(\Delta t/2) = p(-\Delta t/2) + \Delta tF(0) \tag{18}$$

and

$$q(\Delta t) = q(0) + (\Delta t/m)p(\Delta t/2) \tag{19}$$

where m is the mass of the particle. Equations 18 and 19 are easily recognized as being equivalent to the leap-frog form of the Verlet algorithm. Having now illustrated how one may obtain the leap-frog algorithm from the Trotter factorization of the Liouville propagator, let us now turn to the problem of multiple time scales.

Consider a simple case where the total force on a particle may be expressed as a sum of two terms:

$$F(q) = F_1(q) + F_2(q) \tag{20}$$

One could then decompose the associated Liouville operator, such that

$$iL_1 \equiv \dot{q}\frac{\partial}{\partial q} + F_1(q)\frac{\partial}{\partial p}$$
$$iL_2 \equiv F_2(q)\frac{\partial}{\partial p} \tag{21}$$

Applying the Trotter theorem allows us to write

$$\{q(t),p(t)\} \cong [e^{i(\Delta t/2)L_2}e^{i\Delta tL_1}e^{i(\Delta t/2)L_2}]^P\{q(0),p(0)\} \tag{22}$$

where $t = P\Delta t$. We can further factorize the inner propagator as follows:

$$\epsilon^{i\Delta tL_1} = [\mathrm{e}^{i\delta\tau L_1}]^n \tag{23}$$

where $\Delta t = n\delta\tau$, giving

$$\{q(t), p(t)\} \cong [\mathrm{e}^{i(\Delta t/2)L_2}[\mathrm{e}^{i\delta\tau L_1}]^n \mathrm{e}^{i(\Delta t/2)L_2}]^P \{q(0), p(0)\} \tag{24}$$

Inserting an identity operator of the form

$$I = \mathrm{e}^{+i(\Delta t/2)L_2}\mathrm{e}^{-i(\Delta t/2)L_2} \tag{25}$$

into the previous expression and rearranging gives

$$\{q(t), p(t)\} \cong [\mathrm{e}^{i(\Delta t/2)L_2}[\mathrm{e}^{i\delta\tau L_1}]^n \mathrm{e}^{i(\Delta t/2)L_2}]^P I\{q(0), p(0)\} = \mathrm{e}^{i(\Delta t/2)L_2}[(\mathrm{e}^{i\delta\tau L_1})^n \mathrm{e}^{i\Delta tL_2}]^P \mathrm{e}^{-i(\Delta t/2)L_2}\{q(0), p(0)\} \tag{26}$$

Let us now define

$$L_1 \equiv L_A + L_B \tag{27}$$

where

$$iL_A \equiv \dot{q}\frac{\partial}{\partial q}$$
$$iL_B \equiv F_1(q)\frac{\partial}{\partial p} \tag{28}$$

Interesting an identity operator of the form

$$I = \mathrm{e}^{+i(\delta\tau/2)L_B}\mathrm{e}^{-i(\delta\tau/2)L_B} \tag{29}$$

into eq 26, applying the Trotter factorization, and rearranging gives

$$= \mathrm{e}^{i(\Delta t/2)L_2}\mathrm{e}^{i(\delta\tau/2)L_B}[(\mathrm{e}^{i\delta\tau L_A}\mathrm{e}^{i\delta\tau L_B})^n \mathrm{e}^{i\Delta tL_2}]^P \mathrm{e}^{-i(\delta\tau/2)L_B}\mathrm{e}^{-i(\Delta t/2)L_2} \times \{q(0), p(0)\} \equiv \mathrm{e}^{i(\Delta t/2)L_2}\mathrm{e}^{i(\delta\tau/2)L_B}\left\{q(t), p\left(t - \frac{\delta\tau}{2}\right)\right\} \tag{30}$$

where

$$\left\{q(t), p\left(t - \frac{\delta\tau}{2}\right)\right\} \equiv [(\mathrm{e}^{i\delta\tau L_A}\mathrm{e}^{i\delta\tau L_B})^n \epsilon^{i\Delta tL_2}]^P \mathrm{e}^{-i(\delta\tau/2)L_B}\mathrm{e}^{-i(\Delta t/2)L_2}\{q(0), p(0)\} \tag{31}$$

If we now consider propagation of the system from a time t to $t + \Delta t$, we note

$$\left\{q(t + \Delta t), p\left(t + \Delta t - \frac{\delta\tau}{2}\right)\right\} = [(\mathrm{e}^{i\delta\tau L_A}\mathrm{e}^{i\delta\tau L_B})^n \mathrm{e}^{i\Delta tL_2}]\left\{q(t), p\left(t - \frac{\delta\tau}{2}\right)\right\} \tag{32}$$

$$= \mathcal{J}^n(\delta\tau)\left\{q(t), p'\left(t - \frac{\delta\tau}{2}\right)\right\} \tag{33}$$

where we have defined a single-step leap-frog propagator $J(\delta\tau)$ as

$$J(\delta\tau) \equiv \mathrm{e}^{\delta\tau\dot{q}(\partial/\partial q)}\mathrm{e}^{\delta\tau F_1(q)(\partial/\partial p)} \tag{34}$$

and initial momentum state $p'(t - \delta\tau/2)$ such that

$$p'\left(t - \frac{\partial\tau}{2}\right) \equiv p\left(t - \frac{\delta\tau}{2}\right) + n\delta\tau F_2[q(t)] \tag{35}$$

and where

$$p\left(-\frac{\delta\tau}{2}\right) \equiv p(0) - \frac{\delta\tau}{2}F_1[q(0)] - \frac{n\delta\tau}{2}F_2[q(0)] \tag{36}$$

Thus, we have reduced the multiple-time-scale problem to an algorithm whereby the system is propagated from an initial time t to a time $t + \Delta t$ by integrating with the standard leap-frog method for n little time steps $\delta\tau$, starting from the initial condition $\{q(t), p'(t - \delta\tau/2)\}$. Note that the force $F_2[q(t)]$ is evaluated only once every n time-steps. The generalization to N time scales would follow in an analogous manner.

2.2. Multiple-Time-Step Molecular Dynamics Algorithm for a System with Holonomic Constraints Coupled to an External Bath. Returning to the problem of simulated annealing, we now seek a method of coupling the system to an external heat bath. In order to achieve this, we have adopted the algorithm introduced by Berendsen et al.,[8] out of convenience, as it is one of the methods employed for performing simulated annealing in MACROMODEL. In this method, coupling of the system to a bath at temperature T_0 is accomplished by scaling the velocities v at each time-step to λv, where

$$\lambda \equiv \left[1 + \frac{\delta\tau}{\tau_T}\left(\frac{T_0}{T} - 1\right)\right]^{1/2} \tag{37}$$

and τ_T is an associated time constant. Numerical implementation of this algorithm was originally accomplished by use of the leap-frog integrator through the following prescription.

(1) Evaluate the temperature T from the kinetic energy at time $t - \delta\tau/2$. (2) Calculate the scaling factor λ using

$$\lambda = \left[1 + \frac{\delta\tau}{\tau_T}\left(\frac{T_0}{T\left(t - \frac{\partial\tau}{2}\right)} - 1\right)\right]^{1/2} \tag{38}$$

(3) Compute the new velocities using leap-frog:

$$v\left(t + \frac{\delta\tau}{2}\right) = v\left(t - \frac{\delta\tau}{2}\right) + (\delta\tau/m)F(t) \tag{39}$$

(4) Scale the velocities:

$$v'\left(t + \frac{\delta\tau}{2}\right) = \lambda v\left(t + \frac{\delta\tau}{2}\right) \tag{40}$$

(5) Compute the new positions via leap-frog:

$$x(t + \delta\tau) = x(t) + \delta\tau v'\left(t + \frac{\delta\tau}{2}\right) \tag{41}$$

This method may also be generalized to the case where the system is subjected to internal constraints. One common constraint used frequently in molecular dynamics simulations is to treat the bond lengths as fixed to their equilibrium values using either the SHAKE[9] or RATTLE[10] algorithms, thereby enabling one to use a slightly larger time-step. In order to achieve this, the above prescription may be modified slightly be adding the following steps.

(6) Apply SHAKE to the unconstrained positions $x(t + \delta\tau)$ from step 5 to obtain the constrained positions $x'(t + \delta\tau)$:

$$x'(t + \delta\tau) = \mathrm{SHAKE}[x(t + \delta\tau)]$$

(7) Compute the constrained velocities $v''(t + \delta\tau/2)$:

$$v''\left(t + \frac{\delta\tau}{2}\right) \equiv [x'(t + \delta\tau) - x'(t)]/\delta\tau \tag{42}$$

Given that the above approach is based upon the leap-frog form of the Verlet integration algorithm, incorporation of the leap-frog RESPA method described in the previous section is straightforward. In order to generate the state $\{q(t + \Delta t), p(t + \Delta t - \delta\tau / 2)\}$, where $\Delta t = n\delta\tau$, one need only perform steps 1−7 for n little time-steps $\delta\tau$, starting from the initial condition $\{q(t), p'(t - \delta\tau/2)\}$ given by eq 35. This procedure is then repeated N times for a total simulation time of length $t = N\Delta t$.

2.3. Computational Implementation. In this section we describe the computational implementation of the multiple-time-step methodology introduced above into the MACROMODEL[3] molecular modeling package. This was done by adopting a typical functional form (e.g. AMBER,[11] OPLS[12]) to represent the solute−solute interactions, as well as the generalized Born (GB) continuum solvation model[13−15] to approximate the solute−solvent and solvent−solvent interactions. We take the potential energy Φ of the system to be

$$\Phi \equiv \Phi_{\text{self}} + \Phi_{\text{sol}} \tag{43}$$

where Φ_{self} includes the terms representing the protein−protein interactions,

$$\Phi_{\text{self}} \equiv \Phi_{\text{stretch}} + \Phi_{\text{bend}} + \Phi_{\text{dihedral}} + \Phi_{\text{vdW}} + \Phi_{\text{electrostatic}} \tag{44}$$

and Φ_{sol}, an approximate potential of mean force, to represent the effects of the solvent, such that

$$\Phi_{\text{sol}} \equiv \Phi_{\text{cav}} + \Phi_{\text{vdW}} + \Phi_{\text{pol}} \tag{45}$$

which includes Φ_{cav}, a solvent−solvent cavity term, Φ_{vdW} a solute−solvent van der Waals term, and Φ_{pol}, a solute−solvent electrostatic polarization term. The sum of the first two terms is given by

$$\Phi_{\text{cav}} + \Phi_{\text{vdW}} \equiv \sum_k \sigma_k \text{SA}_k \tag{46}$$

where SA_k is the total solvent accessible surface area of atoms of a given type k, and σ_k is an empirical solvation parameter. The solute−solvent electrostatic polarization term Φ_{pol} is approximated by a modified generalized Born equation of the form (in units of kcal/mol)

$$\Phi_{\text{pol}} \equiv -166\left(1 - \frac{1}{\epsilon}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{Q_iQ_j}{\sqrt{r_{ij}^{2} + \alpha_{ij}^{2}e^{-D}}} \tag{47}$$

for a system containing n atoms, where

$$\alpha_{ij} \equiv \sqrt{\alpha_i\alpha_j} \tag{48}$$

and

$$D \equiv \frac{r_{ij}^{2}}{(2\alpha_{ij})^2} \tag{49}$$

Here, α_i is defined to be the effective Born radius of atom i with charge Q_i surrounded by a dielectric medium with a dielectric constant ϵ, and r_{ij} is the radial distance between atom i and atom j. Evaluation of eq 47 requires a method for determining the α_i, which in some sense, may be thought of as the average distance from an atomic charge site to the dielectric (solute−solvent) boundary. In the GB model, this is accomplished by solving the equation

$$\Phi_{\text{pol},i} = -166\left(1 - \frac{1}{\epsilon}\right)\frac{Q_i^{2}}{\alpha_i} \tag{50}$$

for α_i, by assuming that all of the other atoms ($j \neq i$) are electrically neutral, but serve to displace the dielectric medium. Recently, an analytical approximation to $\Phi_{\text{pol},i}$ was introduced[15] which is based upon the simple pairwise expression

$$\Phi_{\text{pol},i} \cong \Phi_{\text{pol},i}^{\text{self}} + \sum_{j\neq i}\frac{g_{ij}V_jQ_i^{2}}{r_{ij}^{4}} \tag{51}$$

The term $\Phi_{\text{pol},i}^{\text{self}}$ is given by

$$\Phi_{\text{pol},i}^{\text{self}} \equiv \frac{-166Q_i^{2}}{R_i + \phi + \gamma_{\text{self}}} \tag{52}$$

where R_i is the van der Waals radius of atom i, ϕ is the dielectric offset, and γ_{self} is an empirically determined single-atom scaling factor. The second term is included in order to approximate the effect of the other atoms ($j \neq i$) displacing the dielectric medium. The physical basis for the assumed functional form is based upon the loss of a favorable charge/induced-dipole interaction of atom i with the dielectric, due to displacement of the medium by an atom j with volume V_j, resulting in the overall proportionality of V_j/r_{ij}^4. The scaling factors g_{ij} are empirically determined parameters which have been fit to correspond with experiment, according to the particular type of pair interaction, such that

$$g_{ij} = \begin{cases} g_{\text{s}} & \text{if 1−2 (stretch) interaction} \\ g_{\text{b}} & \text{if 1−3 (bend) interaction} \\ g_{nb}S_{ij} & \text{if 1, ≥ 4 (nonbonded) interaction} \end{cases} \tag{53}$$

where the nonbonded close-contact function S_{ij} is defined to be

$$S_{ij} \equiv \begin{cases} 1.0 & \text{if } \left(\dfrac{r_{ij}}{R_i + R_j}\right) > 1/\delta \\ \left\{\dfrac{1}{2}\left[1.0 - \cos\left\{\left(\dfrac{r_{ij}}{R_i + R_j}\right)^{2}\delta\pi\right\}\right]\right\}^{2} & \text{otherwise} \end{cases} \tag{54}$$

where δ is a soft cutoff parameter. In the MACROMODEL implementation, evaluation of eq 51 is further simplified by assuming that the bond lengths and angles will not vary greatly between different molecular conformations and are thereby treated as constants (set equal to their equilibrium values) within the expression. Let us then combine the constant terms by defining $\Phi_{\text{pol},i}^{0}$ and $\Phi_{\text{pol},i}^{\text{nb}}$, such that

$$\Phi_{\text{pol},i}^{0} \equiv \Phi_{\text{pol},i}^{\text{self}} + \sum_{\text{stretch}}\frac{g_{\text{s}}V_jQ_i^{2}}{r_{ij}^{4}} + \sum_{\text{bend}}\frac{g_{\text{b}}V_jQ_i^{2}}{r_{ij}^{4}} \tag{55}$$

and

$$\Phi_{\text{pol},i}^{\text{nb}} \equiv \sum_{\text{nonbonded}}\frac{g_{\text{nb}}S_{ij}V_jQ_i^{2}}{r_{ij}^{4}} \tag{56}$$

where

$$\Phi_{\text{pol},i} = \Phi_{\text{pol},i}^{0} + \Phi_{\text{pol},i}^{\text{nb}} \tag{57}$$

Substituting the previous equation for $\Phi_{\mathrm{pol},i}$ into eq 50 now affords us with an analytical method for determining α_i.

Having now arrived at a complete expression for the potential energy of the system, the Liouville operator for a solvated protein containing N atoms in Cartesian coordinates is then given by

$$iL = \sum_{i=1}^{N}\left\{\dot{x}_i\frac{\partial}{\partial x_i} + \dot{y}_i\frac{\partial}{\partial y_i} + \dot{z}_i\frac{\partial}{\partial z_i} + F_{x_i}(x)\frac{\partial}{\partial p_{x_i}} + F_{y_i}(x)\frac{\partial}{\partial p_{y_i}} + F_{z_i}(x)\frac{\partial}{\partial p_{z_i}}\right\} \quad (58)$$

where the qth component of the force on an atom i is

$$F_{q_i}(x) = -\frac{\partial\Phi}{\partial q_i} = -\frac{\partial\Phi_{\mathrm{self}}}{\partial q_i} - \frac{\partial\Phi_{\mathrm{sol}}}{\partial q_i} \equiv F_{\mathrm{self},q_i}(x) + F_{\mathrm{sol},q_i}(x) \quad (59)$$

Making the appropriate substitutions for Φ_{self} and Φ_{sol}, we have

$$F_{\mathrm{self},q_i}(x) \equiv F_{\mathrm{s},q_i}(x) + F_{\mathrm{b},q_i}(x) + F_{\mathrm{d},q_i}(x) + F_{\mathrm{nb},q_i}(x) \quad (60)$$

such that

$$F_{\mathrm{s},q_i}(x) \equiv -\frac{\partial\Phi_{\mathrm{stretch}}}{\partial q_i} \quad (61)$$

$$F_{\mathrm{b},q_i}(x) \equiv -\frac{\partial\Phi_{\mathrm{bend}}}{\partial q_i}$$

$$F_{\mathrm{d},q_i}(x) \equiv -\frac{\partial\Phi_{\mathrm{dihedral}}}{\partial q_i}$$

$$F_{\mathrm{nb},q_i}(x) \equiv -\frac{\partial\Phi_{\mathrm{nb}}}{\partial q_i} \equiv -\frac{\partial(\Phi_{\mathrm{vdW}} + \Phi_{\mathrm{electrostatic}})}{\partial q_i}$$

and

$$F_{\mathrm{sol},q_i}(x) \equiv F_{\mathrm{SA},q_i}(x) + F_{\mathrm{pol},q_i}(x) \quad (62)$$

where

$$F_{\mathrm{SA},q_i}(x) \equiv -\frac{\partial(\Phi_{\mathrm{cav}} + \Phi_{\mathrm{vdW}})}{\partial q_i} \quad (63)$$

$$F_{\mathrm{pol},q_i}(x) \equiv -\frac{\partial\Phi_{\mathrm{pol}}}{\partial q_i} \quad (64)$$

Seeking to take advantage of the RESPA formalism, let us first consider separating the protein–protein nonbonded interactions into those of a short-range and a long-range contribution. One way of achieving this is to express Φ_{nb} as a sum of two terms (in units of kcal/mol), such that

$$\Phi_{\mathrm{nb}} = \Phi_{\mathrm{nb}}^{\mathrm{short}} + \Phi_{\mathrm{nb}}^{\mathrm{long}} \quad (65)$$

where

$$\Phi_{\mathrm{nb}}^{\mathrm{short}} \equiv \sum_{\{ij|r_{ij}<r_c\}}\left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + \frac{332Q_iQ_j}{r_{ij}}\right) \quad (66)$$

and

$$\Phi_{\mathrm{nb}}^{\mathrm{long}} \equiv \sum_{\{ij|r_{ij}\geq r_c\}}\left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + \frac{332Q_iQ_j}{r_{ij}}\right) \quad (67)$$

r_c being a short-range cutoff radius. This separation may be implemented through the use of short- and long-range neighbor lists, which can be periodically updated during the course of the simulation. We may then define the associated short- and long-range nonbonded forces $F_{\mathrm{nb}}^{\mathrm{short}}(x)$ and $F_{\mathrm{nb}}^{\mathrm{long}}(x)$ as

$$F_{\mathrm{nb},q_i}^{\mathrm{short}}(x) \equiv -\frac{\partial\Phi_{\mathrm{nb}}^{\mathrm{short}}}{\partial q_i} \quad (68)$$

$$F_{\mathrm{nb},q_i}^{\mathrm{long}}(x) \equiv -\frac{\partial\Phi_{\mathrm{nb}}^{\mathrm{long}}}{\partial q_i} \quad (69)$$

Let us now consider the solute–solvent polarization term Φ_{pol}. The associated force $F_{\mathrm{pol},q_i}(x)$ is given explicitly by

$$F_{\mathrm{pol},q_i}(x) = -\frac{\partial\Phi_{\mathrm{pol}}}{\partial q_i} = -\frac{\partial}{\partial q_i}\sum_j\sum_k \phi_{jk}^{\mathrm{pol}} \quad (70)$$

where we have defined

$$\phi_{jk}^{\mathrm{pol}} \equiv -166\left(1 - \frac{1}{\epsilon}\right)\frac{Q_jQ_k}{\sqrt{r_{jk}^{2} + \alpha_{jk}^{2}\mathrm{e}^{-D}}} \quad (71)$$

Expanding the right-hand side of eq 70 gives

$$F_{\mathrm{pol},q_i}(x) = -\sum_j\sum_k\left[\left(\frac{\partial\phi_{jk}^{\mathrm{pol}}}{\partial r_{jk}}\right)\frac{\partial r_{jk}}{\partial q_i} + \left(\frac{\partial\phi_{jk}^{\mathrm{pol}}}{\partial \alpha_{jk}^{2}}\right)\frac{\partial \alpha_{jk}^{2}}{\partial q_i}\right]$$
$$= -2\left[\sum_{j\neq i}\left(\frac{\partial\phi_{ij}^{\mathrm{pol}}}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial q_i} + \sum_j\left(\sum_k\frac{\partial\phi_{jk}^{\mathrm{pol}}}{\partial\alpha_{jk}^{2}}\alpha_k\right)\frac{\partial\alpha_j}{\partial q_i}\right] \equiv F_{\mathrm{pol}_r,q_i}(x) + F_{\mathrm{pol}_\alpha,q_i}(x) \quad (72)$$

where we have defined

$$F_{\mathrm{pol}_r,q_i}(x) \equiv -2\sum_{j\neq i}\left(\frac{\partial\phi_{ij}^{\mathrm{pol}}}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial q_i} \quad (73)$$

and

$$F_{\mathrm{pol}_\alpha,q_i}(x) \equiv -2\sum_j\left(\sum_k\frac{\partial\phi_{jk}^{\mathrm{pol}}}{\partial\alpha_{jk}^{2}}\alpha_k\right)\frac{\partial\alpha_j}{\partial q_i} \quad (74)$$

The term $F_{\mathrm{pol}_r,q_i}(x)$ may be partitioned into a short- and long-range contribution, such that

$$F^{\text{short}}_{\text{pol}_r,q_i}(x) \equiv -2 \sum_{\{j\neq i|r_{ij}<r_c\}} \left(\frac{\partial\phi^{\text{pol}}_{ij}}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial q_i} \tag{75}$$

$$F^{\text{long}}_{\text{pol}_r,q_i}(x) \equiv -2 \sum_{\{j\neq i|r_{ij}\geq r_c\}} \left(\frac{\partial\phi^{\text{pol}}_{ij}}{\partial r_{ij}}\right)\frac{\partial r_{ij}}{\partial q_i} \tag{76}$$

Considering now the term $F_{\text{pol}_\alpha,q_i}(x)$, we have

$$F_{\text{pol}_\alpha,q_i}(x) = -2\sum_j\left(\sum_k \frac{\partial\phi^{\text{pol}}_{jk}}{\partial\alpha^2_{jk}}\alpha_k\right)\frac{\partial\alpha_j}{\partial q_i} = -2\sum_j\left(\sum_k \frac{\partial\phi^{\text{pol}}_{jk}}{\partial{\alpha_{jk}}^2}\alpha_k\right)\frac{\partial\alpha_j}{\partial\Phi_{\text{pol},j}}\frac{\partial\Phi_{\text{pol},j}}{\partial q_i} \tag{77}$$

Substituting eq 57 for $\Phi_{\text{pol},j}$ we have

$$\frac{\partial\Phi_{\text{pol},j}}{\partial q_i} = \frac{\partial(\Phi^0_{\text{pol},j} + \Phi^{\text{nb}}_{\text{pol},j})}{\partial q_i} \cong \frac{\partial\Phi^{\text{nb}}_{\text{pol},j}}{\partial q_i} \tag{78}$$

Choosing to separate $\Phi^{\text{nb}}_{\text{pol},j}$ itself into short- and long-range components,

$$\Phi^{\text{nb,short}}_{\text{pol},j} \equiv \sum_{\{k\neq j|r_{jk}<r_c\}} \frac{g_{\text{nb}}S_{jk}V_kQ_j^2}{{r_{jk}}^4} \tag{79}$$

$$\Phi^{\text{nb,long}}_{\text{pol},j} \equiv \sum_{\{k\neq j|r_{jk}\geq r_c\}} \frac{g_{\text{nb}}S_{jk}V_kQ_j^2}{{r_{jk}}^4} \tag{80}$$

and substituting into eq 78, we have

$$\frac{\partial\Phi^{\text{nb}}_{\text{pol},j}}{\partial q_i} = \frac{\partial\Phi^{\text{nb,short}}_{\text{pol},j}}{\partial q_i} + \frac{\partial\Phi^{\text{nb,long}}_{\text{pol},j}}{\partial q_i} \tag{81}$$

Furthermore, if we assume that $\Phi^{\text{nb,long}}_{\text{pol},j}$ does not vary rapidly in time, it may be treated approximately as a constant, such that

$$\frac{\partial\Phi^{\text{nb}}_{\text{pol},j}}{\partial q_i} \cong \frac{\partial\Phi^{\text{nb,short}}_{\text{pol},j}}{\partial q_i} \tag{82}$$

and

$$\Phi^{\text{nb,long}}_{\text{pol},j} \cong \Phi^{\text{nb,long}}_{\text{pol},j}(x_0) = \text{const} \tag{83}$$

where x_0 is a reference state which may be updated with the short- and long-range pair lists. Substituting this result into eq 77 gives

$$F_{\text{pol}_\alpha,q_i}(x) \cong -2\sum_j\left(\sum_k \frac{\partial\phi^{\text{pol}}_{jk}}{\partial{\alpha_{jk}}^2}\alpha_k\right)\frac{\partial\alpha_j}{\partial\Phi_{\text{pol},j}}\frac{\partial\Phi^{\text{nb,short}}_{\text{pol},j}}{\partial q_i} = -2\sum_j\gamma_j\frac{\partial\alpha_j}{\partial\Phi_{\text{pol},j}}\frac{\partial\Phi^{\text{nb,short}}_{\text{pol},j}}{\partial q_i} \tag{84}$$

where

$$\gamma_j \equiv \sum_k \frac{\partial\phi^{\text{pol}}_{jk}}{\partial{\alpha_{jk}}^2}\alpha_k \tag{85}$$

Finally, we may partition the γ_j into a short- and long-range term,

$$\gamma^{\text{short}}_j \equiv \sum_{\{k|r_{jk}<r_c\}} \frac{\partial\phi^{\text{pol}}_{jk}}{\partial{\alpha_{jk}}^2}\alpha_k \tag{86}$$

and

$$\gamma^{\text{long}}_j \equiv \sum_{\{k|r_{jk}\geq r_c\}} \frac{\partial\phi^{\text{pol}}_{jk}}{\partial{\alpha_{jk}}^2}\alpha_k \cong \gamma_j(x_0) = \text{const} \tag{87}$$

where we have followed the same line of reasoning as in the case of $\Phi^{\text{nb,long}}_{\text{pol},j}$. The resulting expressions for $F_{\text{pol}_\alpha,q_i}(x)$ and α_i become

$$F_{\text{pol}_\alpha,q_i}(x) \cong -2\sum_j[\gamma^{\text{short}}_j + \gamma^{\text{long}}_j(x_0)]\frac{\partial\alpha_j}{\partial\Phi_{\text{pol},j}}\frac{\partial\Phi^{\text{nb,short}}_{\text{pol},j}}{\partial q_i} \tag{88}$$

and

$$\alpha_i \cong -166\left(1-\frac{1}{\epsilon}\right)\left\{\frac{Q_i^2}{\Phi^0_{\text{pol},i} + \Phi^{\text{nb,short}}_{\text{pol},i} + \Phi^{\text{nb,long}}_{\text{pol},i}(x_0)}\right\} \tag{89}$$

Following the previous, and somewhat arduous, task of separating the various components of the total force into short- and long-range contributions, let us now define

$$F_{1,q_i}(x) \equiv F_{\text{s},q_i}(x) + F_{\text{b},q_i}(x) \tag{90}$$

$$F_{2,q_i}(x) \equiv F_{\text{d},q_i}(x) + F^{\text{short}}_{\text{nb},q_i}(x) \tag{91}$$

$$F_{3,q_i}(x) \equiv F^{\text{short}}_{\text{pol}_r,q_i}(x) + F_{\text{pol}_\alpha,q_i}(x) \tag{92}$$

$$F_{4,q_i}(x) \equiv F^{\text{long}}_{\text{nb},q_i}(x) + F^{\text{long}}_{\text{pol}_r,q_i}(x) + F_{\text{SA},q_i}(x) \tag{93}$$

In order to impose the addition of holonomic constraints on the bond-length degrees of freedom, the above equation for $F_{1,q_i}(x)$ may be modified, such that

$$F_{1,q_i}(x) \equiv F_{\text{b},q_i}(x) + C_{q_i}(x) \tag{94}$$

where $C_{q_i}(x)$ is the associated constraint force, which formally may be included through the use of a Lagrange multiplier. In practice, however, the bond-length constraints are implemented by employing the SHAKE algorithm as described above.

Secondly, since it is our supposition that the atomic degrees of freedom are coupled more strongly to the rapidly varying short-range interactions, let us make the further assumption that the interactions given by $F_{4,q_i}(x)$ may be adequately described by a quadratic approximation to the potential. This is equivalent to making a linear approximation to $F_{4,q_i}(x)$ such that

$$F_{4,q_i}(x) \cong F_{4,q_i}(x_0) + \sum_j A_{ij}(x_0)(q - q_0)_j \tag{95}$$

where

$$A_{ij}(x_0) \equiv \left(\frac{\partial F_{4,q_i}(x)}{\partial q_j}\right)_{x=x_0} \tag{96}$$

x_0 being some reference state. Numerically then, evaluation of $F_{4,q_i}(x)$ is accomplished in an approximate manner by a simple and efficient matrix–vector multiplication. The second derivative matrix $A_{ij}(x_0)$ is evaluated infrequently, along with regeneration of the constant long-range terms $F_{4,q_i}(x_0)$, $\Phi_{\text{pol},i}^{\text{nb,long}}(x_0)$, and $\gamma_i^{\text{long}}(x_0)$ defined above and the short- and long-range neighbor lists.

Returning now to the integration of the classical equations of motion, consider for the moment the time evolution of a single atom i in the microcanonical ensemble. We may define the Liouville operator

$$iL_{q_i} \equiv \dot{q}_i \frac{\partial}{\partial q_i} + F_{q_i}(x)\frac{\partial}{\partial p_{q_i}} \tag{97}$$

such that

$$\left\{q_i(t + \Delta t), p_{q_i}\!\left(t + \Delta t - \frac{\delta\tau}{2}\right)\right\} = G_{q_i}(\Delta t)\left\{q_i(t), p_{q_i}\!\left(t - \frac{\delta\tau}{2}\right)\right\} \tag{98}$$

where

$$G_{q_i}(\Delta t) \equiv e^{i\Delta t L_{q_i}} \tag{99}$$

Using the above definitions for $F_{n,q_i}(x)$, where $n \in \{1, 2, 3, 4\}$, one may then express iL_{q_i} as a sum of four terms,

$$iL_{q_i} = iL_{1,q_i} + iL_{2,q_i} + iL_{3,q_i} + iL_{4,q_i} \tag{100}$$

where

$$iL_{1,q_i} \equiv \dot{q}_i \frac{\partial}{\partial q_i} + F_{1,q_i}(x)\frac{\partial}{\partial p_{q_i}} \tag{101}$$

$$iL_{2,q_i} \equiv F_{2,q_i}(x)\frac{\partial}{\partial p_{q_i}}$$

$$iL_{3,q_i} \equiv F_{3,q_i}(x)\frac{\partial}{\partial p_{q_i}}$$

$$iL_{4,q_i} \equiv F_{4,q_i}(x)\frac{\partial}{\partial p_{q_i}}$$

Having done this, we are now able to make use of the multiple-time-step methodology discussed previously in order to generate a numerical integration algorithm for the system. In this case, we choose a factorization of the form

$$G_{q_i}(\Delta t) \cong \{[((e^{i\delta\tau L_{1,q_i}})^{n_1} e^{in_1\delta\tau L_{2,q_i}})^{n_2} e^{in_1n_2\delta\tau L_{3,q_i}}]^{n_3} e^{in_1n_2n_3\delta\tau L_{4,q_i}}\} \tag{102}$$

which would follow by induction from the form of eq 32. Substituting this into eq 98 gives the final result,

$$\left\{q_i(t + \Delta t), p_{q_i}\!\left(t + \Delta t - \frac{\delta\tau}{2}\right)\right\} \cong \tag{103}$$

$$\{[((e^{i\delta\tau_1 L_{1,q_i}})^{n_1} e^{i\delta\tau_2 L_{2,q_i}})^{n_2} e^{i\delta\tau_3 L_{3,q_i}}]^{n_3} e^{i\delta\tau_4 L_{4,q_i}}\}\left\{q_i(t), p_{q_i}\!\left(t - \frac{\delta\tau}{2}\right)\right\}$$

where it is implicitly understood that the initial state at ($t = 0$) is taken to be

$$\left\{q_i(0), p_{q_i}\!\left(-\frac{\delta\tau}{2}\right)\right\} \equiv \tag{104}$$

$\{q_i(0), p_{q_i}(0) - (\delta\tau_1/2)F_{1,q_i}(0) - (\delta\tau_2/2)F_{2,q_i}(0) - (\delta\tau_3/2)F_{3,q_i}(0) - (\delta\tau_4/2)F_{4,q_i}(0)\}$ and where

$$\begin{aligned} \delta\tau_4 &\equiv n_1 n_2 n_3 \delta\tau \\ \delta\tau_3 &\equiv n_1 n_2 \delta\tau \\ \delta\tau_2 &\equiv n_1 \delta\tau \\ \delta\tau_1 &\equiv \delta\tau \end{aligned} \tag{105}$$

Evolution of the system, within the microcanonical ensemble, is then given by the numerical implementation of eq 103. This is accomplished by noting that the action of the various operators upon a state to the right may be expressed as

$$e^{\delta\tau_n F_{n,q_i}(x)(\partial/\partial p_{q_i})}\{q_i, p_{q_i}\} = \{q_i, p_{q_i} + \delta\tau_n F_{n,q_i}(x)\} \tag{106}$$

and

$$e^{\delta\tau_n \dot{q}_i(\partial/\partial q_i)}\{q_i, p_{q_i}\} = \left\{q_i + \frac{\delta\tau_n p_{q_i}}{m_i}, p_{q_i}\right\} \tag{107}$$

In order to make use of our multiple-time-step numerical integration technique within the context of simulated annealing, the above propagation scheme must be modified to include the effect of an external heat bath. This is done to provide us with a means of gradually removing energy from the system over time, hopefully allowing it to relax into a state at or near the global minimum of the potential energy surface. Although there exist numerous approaches to this problem, we have adopted the method described in section 3.2.2, as it is one of the default methods for performing simulated annealing in the MACROMODEL molecular modeling package, and seemed to be adequate for our purposes here. Although the RESPA implementation of this method is straightforward, we have included a schematic FORTRAN example in the Appendix for the sake of clarity. We mention, however, that one could, in principle, use any standard method for coupling the system to an external bath (e.g. Nosé–Hoover thermostat, Langevin dynamics) within the multiple-time-step formalism.

Having coupled the system to an external bath with a temperature T_0, the system may then be annealed by slowly decreasing T_0 according to some prescribed schedule. In this

TABLE 1: Average Final Energies and C^{α} rmsd values for CTF

method	⟨Φ⟩ (kJ/mol)	⟨rmsd⟩ (Å)
1	−14 086.55 ± 18.58	0.99 ± 0.07
2	−14 089.04 ± 25.26	1.24 ± 0.19
3	−14 109.89 ± 46.46	1.37 ± 0.16
4	−14 100.20 ± 25.75	1.25 ± 0.25

TABLE 2: Comparison of Simulated Annealing CPU Times

method	CPU time (s)
1	71682.3
2	18345.9
3	4418.0
4	3635.3

work, we have adopted a linear cooling schedule such that

$$T_0(t) = T_0^{\text{initial}} - \left(\frac{T_0^{\text{initial}} - T_0^{\text{final}}}{N\Delta t}\right)t \qquad (108)$$

where T_0^{initial} and T_0^{final} are the predetermined initial and final bath temperatures, and $N\Delta t$ is the total time of the simulation.

In the following section we apply the above multiple-time-step-based simulated annealing algorithm to the minimization of the C-terminal fragment of the L7/L12 ribosomal protein (CTF) in solution.

3. Simulated Annealing of a Protein in Solution by Multiple-Time-Step Molecular Dynamics

As an initial test of the multiple-time-step methodology described above, we wish to compare its performance against an algorithm that is based upon the standard leap-frog integration scheme, as well as to the default MACROMODEL implementation. In order to accomplish this, we consider the relative performances of these methods in simulated annealing computer simulations of the C-terminal fragment of the L7/L12 ribosomal protein (PDB code 1CTF), a small mixed α/β protein that was one of the subjects of a previous study. In comparing these methods, we first choose to follow a protocol similar to that given in ref 16 and adopt the same united-atom model using the AMBER*[11,17] and GB continuum solvation approximation. It is also noted that both here in this initial comparison and in ref 16 there is no explicit term included for hydrogen bonds in the potential, using instead the approximation given by ref 18. Additionally, we choose to constrain all bond lengths, rather than just to hydrogens, and adopt a time-step of 2.0 fs for the standard methods. Nonbonded cutoffs were set at 12.0 Å for both the van der Waals and electrostatic interactions. The initial protein configuration itself was obtained by performing conjugate gradient minimization within the continuum solvent starting from the X-ray structure. In order to allow for comparisons with the results in ref 16, we adopt the same annealing schedule which includes 10 ps of equilibration at 300 K followed by linear cooling of the bath temperature from 300 to 50 K over the next 40 ps. The structures are then minimized in solution by conjugate gradient using infinite nonbonded cutoffs. Five simulations were performed for each method using different random number seeds. In our first investigation, we compare four different methods: (1) the constant temperature method of Berendsen et al. discussed in section 2.1.2, which is based upon the standard leap-frog integration algorithm, (2) the default MACROMODEL implementation of (1) which uses the standard leap-frog integrator but updates the long-range forces only every 10 time-steps, (3) the multiple-time-step method described in the previous section, where updating the second-derivative matrix is done periodically every 500 time-steps, and (4) the same multiple-time-step method but using a variable updating scheme for the second-derivative matrix based upon the monitoring of atomic displacements. For methods (3) and (4) we choose $n_1 = 2$, $n_2 = 2$, $n_3 = 2$, the associated time-steps of which are given by eq 105, and a short-range cutoff of $r_c = 8.0$ Å.

In Table 1 the results for the average final total energies and average C^{α} rmsd (root mean square deviation) values from the minimized X-ray structure are presented. For the average final energies the agreement is generally good, although the standard deviation for method (3) is somewhat higher. The differences between the average rmsd values is somewhat larger, although curiously there does not seem to be a correlation between lower rmsd values and lower average energies. In Table 2 the total CPU times for a typical simulation are presented. Here it is clear that the multiple-time-step methods are considerably more efficient, being nearly 20 times faster than that of the leap-frog-based method and on the order of 4−5 times faster than that of the default implementation.

Having now established that one may obtain similar results more efficiently using the RESPA-based simulated annealing method, we now wish to address the question of whether lower energies can be found by the use of slower annealing schedules. In this case we choose to decrease the cooling schedule by nearly an order of magnitude. Again, as a first test, we begin with the minimized X-ray structure as our initial configuration. The system is then equilibrated for 50 ps at a somewhat higher temperature of 350 K, followed by a linear cooling from 350 to 50 K over the next 500 ps. The final structures were then minimized by conjugate gradient as before. The RESPA parameter set adopted was the same as above with the exception of r_c, which was increased to 10 Å due to the higher temperature. The results for five such simulations are presented in Table 3.

The average final total energy for the these simulations is −14269.59 ± 35.02 kJ/mol, which is considerably lower than for the cases employing the more rapid cooling schedule. However, the most striking result of this study is that there does not seem to be a correlation between low final energies and low rmsd values. In fact, the structure with the lowest energy has the highest C^{α} rmsd deviation from the minimized X-ray structure. In order to better understand this result, we first

TABLE 3: Simulation Results for RESPA Method Using Slower Annealing Schedule[a]

ID	rmsd	R_G	Φ_{SA}	Φ_{vdW}	Φ_{esc}	Φ_{pol}	Φ_{tes}	Φ
1	2.62	9.89	81.99	−1512.66	−11 777.58	−3120.82	−14 898.40	−14 228.03
2	3.63	9.58	77.78	−1539.22	−11 972.85	−2989.26	−14 962.11	−14 295.35
3	2.92	9.81	82.28	−1515.17	−12 038.43	−2929.07	−14 967.50	−14 235.84
4	1.68	9.79	78.70	−1567.17	−12 123.13	−2850.66	−14 973.79	−14 268.17
5	5.04	9.76	85.99	−1486.10	−12 345.58	−2609.28	−14 954.86	−14 320.58

[a] Here ID identifies the structure, rmsd is the C^{α} rmsd from the minimized X-ray structure, R_G is the radius of gyration in angstroms, Φ_{SA} is the surface area dependent solvation energy, Φ_{vdW} is the van der Waals energy, Φ_{esc} is the electrostatic Coulombic energy, Φ_{pol} is the electrostatic solute−solvent polarization energy, Φ_{tes} is the total electrostatic energy (including solvation), and Φ is the total AMBER*/GB potential energy. All energies are in kJ/mol.

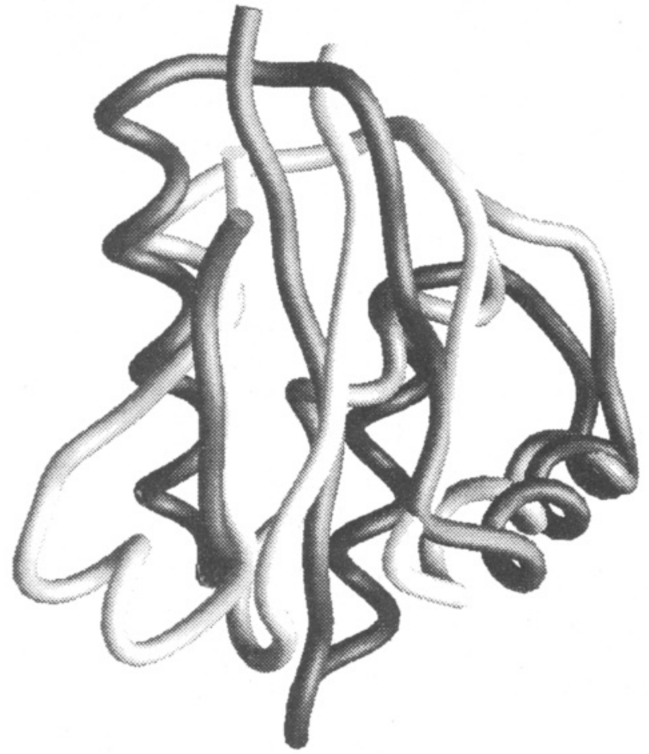

Figure 1. C^α worms for minimized X-ray structure (dark) and minimum energy structure (structure 5) (light).

TABLE 4: Results Using Numerical Born Radii[a]

ID	rmsd	Φ_{esc}	Φ_{pol}	Φ_{tes}	Φ
1	2.62	−11 777.58	−3222.40	−14 999.98	−14 329.72
2	3.63	−11 972.85	−3086.99	−15 059.84	−14 393.18
3	2.92	−12 038.43	−3021.43	−15 059.86	−14 328.32
4	1.68	−12 123.13	−2934.75	−15 057.88	−14 352.38
5	5.04	−12 345.58	−2692.44	−15 038.02	−14 403.86

[a] ID identifies the structure, rmsd is the C^α rmsd from the minimized X-ray structure, Φ_{esc} is the electrostatic Coulombic energy, Φ_{pol} is the electrostatic solute−solvent polarization energy, Φ_{tes} is the total electrostatic energy (including solvation), and Φ is the total potential energy. All energies are in kJ/mol.

TABLE 5: Results Using AMBER* Explicit 10−12 Hydrogen-Bonding Term[a]

ID	rmsd	Φ_{HB}	Φ
1	2.62	−56.27	−14 281.44
2	3.63	−73.67	−14 344.17
3	2.92	−45.12	−14 264.39
4	1.68	−64.64	−14 298.69
5	5.04	−53.86	−14 348.63

[a] ID identifies the structure, rmsd is the C^α rmsd from the minimized X-ray structure, Φ_{HB} is the explicit 10−12 hydrogen-bonding energy, and Φ is the total potential energy. All energies are in kJ/mol.

recalculate the electrostatic polarization component of the solvation energies using a more accurate numerical method for determining the Born radii, as opposed to the analytic approximation given by eq 51. The results are given in Table 4.

Although the results are somewhat affected, it is still the case that the structure with the lowest energy has the highest rmsd. In Figure 1 we compare this minimum energy structure with that of the minimized X-ray structure. In Figure 2 we include the structure with the lowest rmsd for comparison. It is interesting to note the distortion in the secondary structure of the minimum energy configuration, suggesting perhaps a problem with the hydrogen-bonding treatment. In order to investigate this further, we recalculated the energies using the

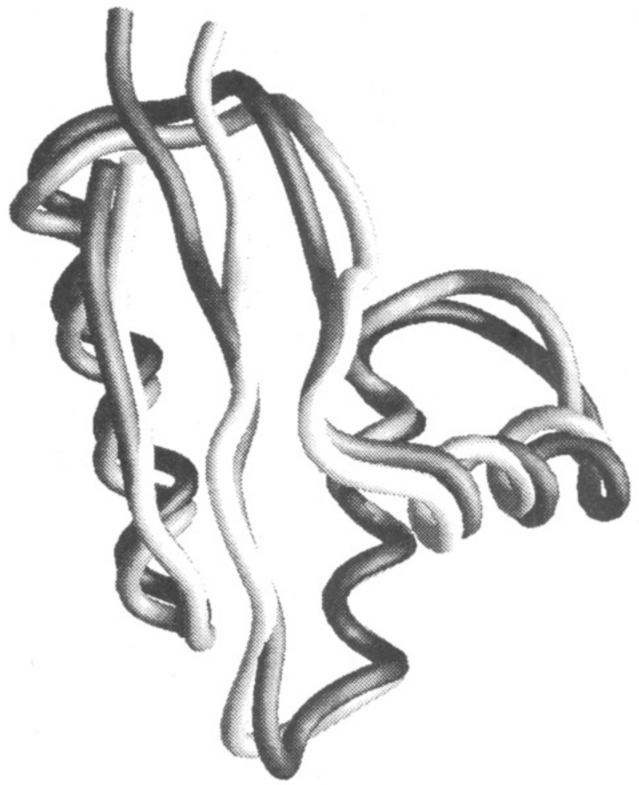

Figure 2. C^α worms for minimized X-ray structure (dark) and minimum rmsd structure (structure 4) (light).

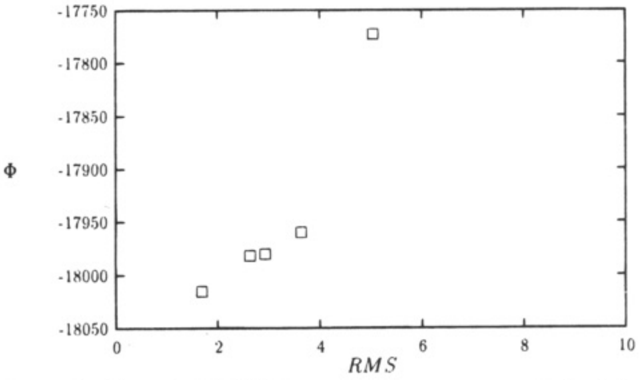

Figure 3. Plot of OPLS*/GB energies for structures obtained by simulated annealing on AMBER*/GB potential surface vs C^α rmsd from the minimized X-ray structure, where Φ is in kJ/mol and rmsd is in angstroms.

AMBER* explicit 10−12 hydrogen-bonding treatment, as opposed to that of ref 18. Table 5 gives the resulting hydrogen-bonding energies and total energies. However, upon examining the results, we see that here too we find that the structure with the highest rmsd has the lowest energy.

We next consider calculating the energies of the same structures using the OPLS*[3,12] force field. The results are presented in Table 6. It is interesting to see that in this case the structure with the lowest energy also has the lowest rmsd to the minimized X-ray structure. Additionally, we plot the total energy vs rmsd in Figure 3. It would appear that, at least for these five structures considered, there does exist a correlation between lower energies and lower rmsd values. Finding this a somewhat peculiar result, that the structures are ranked so differently by the two force fields, in Table 7 we plot the relative energy differences, for each term in the potential, between structure 4 and structure 5 calculated using both the OPLS* and the AMBER* parametrization. In the AMBER* case, the implicit hydrogen-bonding potential was used. Numerically

TABLE 6: Results Using OPLS*/GB Force Field to Calculate Energies of Structures That Were Obtained by Simulated Annealing on AMBER*/GB Potential Surface[a]

ID	rmsd	R_G	Φ_{SA}	Φ_{vdW}	Φ_{esc}	Φ_{pol}	Φ_{tes}	Φ
1	2.62	9.89	81.99	−618.52	−14 872.55	−3611.60	−18 484.15	−17 982.32
2	3.63	9.58	77.78	−570.44	−15 117.56	−3423.05	−18 540.61	−17 960.55
3	2.92	9.81	82.28	−603.17	−15 141.53	−3393.87	−18 535.40	−17 981.03
4	1.68	9.79	78.70	−621.73	−15 300.50	−3267.60	−18 568.10	−18 015.80
5	5.04	9.76	85.99	−460.03	−15 460.00	−3024.81	−18 484.81	−17 773.41

[a] ID identifies the structure, rmsd is the C^{α} rmsd from the minimized X-ray structure, R_G is the radius of gyration in angstroms, Φ_{SA} is the surface area dependent solvation energy, Φ_{vdW} is the van der Waals energy, Φ_{esc} is the electrostatic Coulombic energy, Φ_{pol} is the electrostatic solute−solvent polarization energy, Φ_{tes} is the total electrostatic energy (including solvation), and Φ is the total potential energy. All energies are in kJ/mol.

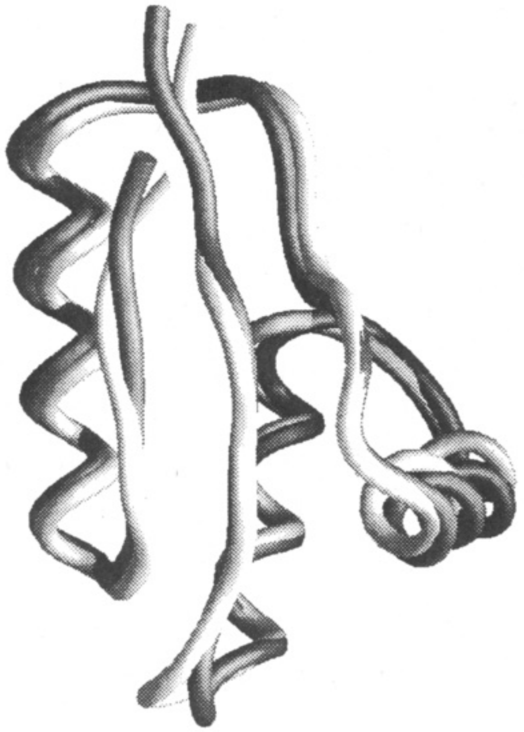

Figure 4. C^{α} worms for minimized X-ray structure (dark) and minimum rmsd structure (structure 6) (light).

TABLE 7: Relative Energy Differences between Structure 4 and Structure 5 Using Both the OPLS*/GB and AMBER*/GB Force Fields[a]

energy term	$\Delta\Phi_{45}^{\mathrm{OPLS*}}$	$\Delta\Phi_{45}^{\mathrm{AMBER*}}$
total	−242.39	+51.48
stretch	−3.62	−3.63
bend	−21.88	−21.93
proper torsion	+33.39	+183.86
improper torsion	+2.00	+1.41
electrostatic	+159.51	+222.45
van der Waals	−161.70	−81.08
solvation (surface area)	−7.29	−7.29
solvation (polarization)	−242.79	−242.30

[a] $\Delta\Phi_{45} \equiv \Phi_4 - \Phi_5$ and energies are in kJ/mol.

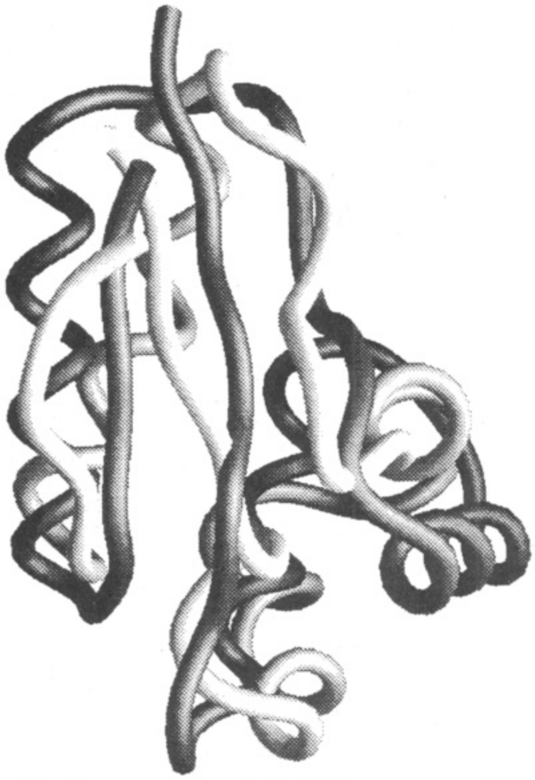

Figure 5. C^{α} worms for minimized X-ray structure (dark) and minimum energy structure (structure 7) (light).

determined Born radii were used for both calculations. The results indicate that the largest difference between the energy components calculated using the two force fields is in the proper torsion term, followed by the van der Waals and electrostatic terms.

Having found such a large discrepancy between the predictions of the two force fields we decided to perform simulated annealing experiments using the OPLS* force field parameters. In Table 8 results are presented for the minimum rmsd structure (structure 6) and the minimum energy structure (structure 7) obtained. Structure 6 (Figure 4) was found using the same annealing schedule as above, whereas structure 7 (Figure 5) was obtained by annealing from an initial temperature of 500 K to a final temperature of 50 K over 1000 ps. The results indicate that the OPLS*/GB force field also leads to low-energy structures with high rmsd values. In Table 9 we compare the relative energy differences between structure 6 and structure 7

TABLE 8: Results Using OPLS*/GB Force Field To Calculate Energies of Structures That Were Obtained by Simulated Annealing on OPLS*/GB Potential Surface[a]

ID	rmsd	R_G	Φ_{pt}	Φ_{SA}	Φ_{vdW}	Φ_{esc}	Φ_{pol}	Φ
6	1.19	10.04	883.08	81.15	−1228.60	−14 948.67	−3894.01	−18 812.50
7	5.19	10.02	816.46	93.29	−1071.44	−16 017.18	−3095.80	−18 917.23

[a] ID identifies the structure, rmsd is the C^{α} rmsd from the minimized X-ray structure, R_G is the radius of gyration in angstroms, Φ_{pt} is the proper torsional energy, Φ_{SA} is the surface area dependent solvation energy, Φ_{vdW} is the van der Waals energy, Φ_{esc} is the electrostatic Coulombic energy, Φ_{pol} is the electrostatic solute−solvent polarization energy, and Φ is the total potential energy. All energies are in kJ/mol.

CHART 1

```
 CALL FORCES_F1
 CALL FORCES_F2
 CALL FORCES_F3
 CALL FORCES_F4
 DO M=1,NATOMS
   VX(M)=VX(M)

$       -0.5*               DELTAT*F1_X(M)/AMASS(M)
$       -0.5*DBLE(N1)*      DELTAT*F2_X(M)/AMASS(M)
$       -0.5*DBLE(N1*N2)*   DELTAT*F3_X(M)/AMASS(M)
$       -0.5*DBLE(N1*N2*N3)*DELTAT*F4_X(M)/AMASS(M)
   VY(M)=VY(M)
$       -0.5*               DELTAT*F1_Y(M)/AMASS(M)
$       -0.5*DBLE(N1)*      DELTAT*F2_Y(M)/AMASS(M)
$       -0.5*DBLE(N1*N2)*   DELTAT*F3_Y(M)/AMASS(M)
$       -0.5*DBLE(N1*N2*N3)*DELTAT*F4_Y(M)/AMASS(M)
   VZ(M)=VZ(M)
$       -0.5*               DELTAT*F1_Z(M)/AMASS(M)
$       -0.5*DBLE(N1)*      DELTAT*F2_Z(M)/AMASS(M)
$       -0.5*DBLE(N1*N2)*   DELTAT*F3_Z(M)/AMASS(M)
$       -0.5*DBLE(N1*N2*N3)*DELTAT*F4_Z(M)/AMASS(M)

 ENDDO
 DO N=1,NSTEPS
   CALL FORCES_F4
   DO M=1,NATOMS

     VX(M)=VX(M)
$         +DBLE(N1*N2*N3)*DELTAT*F4_X(M)/AMASS(M)
     VY(M)=VY(M)
$         +DBLE(N1*N2*N3)*DELTAT*F4_Y(M)/AMASS(M)
     VZ(M)=VZ(M)
$         +DBLE(N1*N2*N3)*DELTAT*F4_Z(M)/AMASS(M)
   ENDDO
   DO I=1,N3
     CALL FORCES_F3
     DO M=1,NATOMS
       VX(M)=VX(M)
$           +DBLE(N1*N2)*DELTAT*F3_X(M)/AMASS(M)
       VY(M)=VY(M)
$           +DBLE(N1*N2)*DELTAT*F3_Y(M)/AMASS(M)
       VZ(M)=VZ(M)
$           +DBLE(N1*N2)*DELTAT*F3_Z(M)/AMASS(M)
     ENDDO
     DO J=1,N2
       CALL FORCES_F2
       DO M=1,NATOMS
         VX(M)=VX(M)
$             +DBLE(N1)*DELTAT*F2_X(M)/AMASS(M)
         VY(M)=VY(M)
$             +DBLE(N1)*DELTAT*F2_Y(M)/AMASS(M)
         VZ(M)=VZ(M)
$             +DBLE(N1)*DELTAT*F2_Z(M)/AMASS(M)
       ENDDO

       DO K=1,N1
         CALL FORCES_F1
```

```
C CALCULATE THE TEMPERATURE FROM CURRENT KINETIC ENERGIES
              CALL TEMPERATURE(T)
C COMPUTE NEW UNCONSTRAINED VELOCITIES
            DO M=1,NATOMS
              VX(M)=VX(M)
   $               +DELTAT*F1_X(M)/AMASS(M)
              VY(M)=VY(M)
   $               +DELTAT*F1_Y(M)/AMASS(M)
              VZ(M)=VZ(M)
   $               +DELTAT*F1_Z(M)/AMASS(M)
            ENDDO

C CALCULATE VELOCITY SCALING FACTOR
              LAMBDA=SQRT(1.0 + (DELTAT/TAU_BATH)*
      $                          ( (TBATH/T)-1)   )

C SCALE THE VELOCITIES
              DO M=1,NATOMS
                VX(M)=LAMBDA*VX(M)
                VY(M)=LAMBDA*VY(M)
                VZ(M)=LAMBDA*VZ(M)
              ENDDO
              DO M=1,NATOMS
                X_OLD(M)=X(M)
                Y_OLD(M)=Y(M)
                Z_OLD(M)=Z(M)
              ENDDO
C COMPUTE NEW UNCONSTRAINED POSITIONS
              DO M=1,NATOMS
                X(M)=X(M)
     $               +DELTAT*VX(M)
                Y(M)=Y(M)
     $               +DELTAT*VY(M)
                Z(M)=Z(M)
     $               +DELTAT*VZ(M)
              ENDDO
C ITERATE TO SATISFY BOND LENGTH CONSTRAINTS USING SHAKE
              CALL SHAKE(X_PRIME,Y_PRIME,Z_PRIME,X,Y,Z)
C COMPUTE CONSTRAINED VELOCITIES
              DO M=1,NATOMS
                VX_PRIME(M)=(XPRIME(M)-X_OLD(M))/DELTAT
                VY_PRIME(M)=(YPRIME(M)-Y_OLD(M))/DELTAT
                VZ_PRIME(M)=(ZPRIME(M)-Z_OLD(M))/DELTAT
              ENDDO
              DO M=1,NATOMS
                X(M)=X_PRIME(M)
                Y(M)=Y_PRIME(M)
                Z(M)=Z_PRIME(M)
                VX(M)=VX_PRIME(M)
                VY(M)=VY_PRIME(M)
                VZ(M)=VZ_PRIME(M)
              ENDDO
            ENDDO
          ENDDO
        ENDDO
      ENDDO
```

using both the OPLS* and the AMBER* force fields. In this case we again see that the two structures are ranked oppositely by the two force field parametrizations. Here the largest differences can be found in the electrostatic, solvent polarization, and van der Waals terms.

Although the previous results certainly merit further investigation, they do draw attention to the importance of developing reliable potentials and to how sensitively the results of a given simulation may be tied to the particular model employed. Furthermore, the results suggest it is possible that the potentials used in this study are not of sufficient accuracy to distinguish the native protein structure. It should, however, be pointed out that the structures obtained are most likely those corresponding to local energy minima, and it is possible that the global minimum structure is indeed more native-like. It is also noted that, ideally, comparisons between structures should be done on the basis of free energy differences as opposed to potential energy differences, and it is not yet known how this would affect the results in this work. Lastly, we mention that although the generalized Born solvation model was used in these studies, in future work we plan to use a potentially more accurate Poisson–Boltzman–based continuum model. This should enable us to determine the accuracy of the generalized Born model and to explore the question of whether the ranking of the structures changes upon the use of the more accurate solvation model.

Given that our main objective in these studies was simply to test the multiple-time-step algorithm's performance against those of other standard methods and to determine whether it might

TABLE 9: Relative Energy Differences between Structure 6 and Structure 7 Using Both the OPLS*/GB and AMBER*/GB Force Fields[a]

energy term	$\Delta\Phi_{67}^{\mathrm{OPLS*}}$	$\Delta\Phi_{67}^{\mathrm{AMBER*}}$
total	+104.72	−126.81
stretch	−10.21	−9.99
bend	−51.70	−51.71
proper torsion	+66.62	+35.72
improper torsion	−0.99	−8.62
electrostatic	+1068.51	+802.57
van der Waals	−157.16	−213.07
solvation (surface area)	−12.14	−12.14
solvation (polarization)	−798.21	−669.58

[a] $\Delta\Phi_{67} \equiv \Phi_6 - \Phi_7$ and energies are in kJ/mol.

be useful as a tool in the investigation of protein structures by enhancing our ability to perform longer simulations, we find these initial studies to be encouraging. We mention that some of the above simulations took as long as 3−4 days of real time on a reasonably loaded IBM RS6000 workstation. The time needed to perform the same simulations with the standard leap-frog-based algorithm is estimated to be on the order of 60−80 days, thereby demonstrating what can be gained by using such a method. However, in order to more fully establish whether the method is truly useful, one should perform many more such tests as those above, and on a variety of systems.

4. Conclusion

It is now generally believed that the three-dimensional structure or native state of a protein is given by the configuration that minimizes the free energy of the system. If we are to assume that a unique native state exists, the problem is then reduced to one of global optimization, the free energy being the associated cost function. However, due to the complexity of the free energy problem, in practice one often resorts to attempting to minimize the systems potential energy. In this paper we have presented an efficient method for performing simulated annealing on atomic level protein structures as a means of generating low-energy configurations. This method, which is based upon a Trotter factorization of the Liouville propagator, was found to be significantly more efficient than that of standard methods when applied to simulated annealing computer experiments upon CTF in a continuum solvent. Additionally, we explored the use of slower annealing schedules using the RESPA-based method and found that lower energies were obtained. Low-energy structures with large rms deviations from the minimized X-ray structure were found, thereby suggesting the possibility that the potentials employed in this work may not be of sufficient accuracy to distinguish the native protein structure. In particular, the unraveling of secondary structure that we observe suggests that the treatment of hydrogen bonding in both the AMBER* and OPLS* potentials may be problematic. Because there are a large number of terms in the potential functions, including the GB solvation model, which is not strictly a part of the AMBER* or OPLS* models, it is not possible to definitely identify any one term in the AMBER* and OPLS* potentials as incorrect. However, the large discrepancies in the relative conformational energies of virtually all important terms imply that at least one of these potentials (and quite possibly both) must be quantitatively incorrect. In future work, we intend to study additional potentials (e.g. CHARMm), solvation models, and proteins to examine the generality of the preliminary results described above. Ultimately, this type of calculation may be used to validate a new generation of protein force fields.

Acknowledgment. This work was supported by grants from the National Institutes of Health (NIH GM40526; PI RAF) and (GM 43340; PI BJB), and by the NIH Division of Research Resources (SP41RR06892). Molecular graphics were generated using the program GRASP developed by Anthony Nicholls.

Appendix

FORTRAN Example of a Multiple-Time-Step Molecular Dynamics Algorithm for a System with Holonomic Constraints Coupled to an External Bath. Chart 1 is a schematic FORTRAN implementation of the molecular dynamics algorithm discussed in section 3 using the propagator given by eq 102 and the heat bath coupling scheme described in section 3.2.2. The Cartesian positions and velocities of an atom I, with mass AMASS(I), are given by X(I), Y(I), Z(I), and VX(I), VY(I), VZ(I). X_PRIME(I), Y_PRIME(I), Z_PRIME(I) and VX_PRIME(I), VY_PRIME(I), VZ_PRIME(I) are the constrained positions and velocities. The atomic forces are contained in the arrays F1_X(I) − F4_Z(I). Here DELTAT corresponds to $\delta\tau_1$, T is the temperature calculated from the kinetic energies, TBATH is the temperature of the external bath, LAMBDA is the velocity scaling factor, and TAU_BATH is the bath time constant.

References and Notes

(1) Kirkpatrick, S., Jr.; Gelatt, C. D.; Vecchi, M. P. *Science* **1983**, *220*, 671.

(2) Humphreys, D. D.; Friesner, R. A.; Berne, B. J. *J. Phys. Chem.* **1994**, *98*, 6885.

(3) Mohamadi, F.; Richards, N. G. J.; Guida, W. C.; Liskamp, R.; Lipton, M.; Caufield, C.; Chang, G.; Hendrickson, T.; Still, W. C. *J. Comput. Chem.* **1990**, *11*, 440.

(4) Trotter, H. F. *Proc. Am. Math Soc.* **1959**, *10*, 545.

(5) Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990.

(6) Hockney, R. W. *Methods Comput. Phys.* **1970**, *9*, 136.

(7) Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1993**, *99*, 2278.

(8) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Haak, J. R. *J. Chem. Phys.* **1984**, *81*, 3684.

(9) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. *J. Comput. Phys.* **1977**, *23*, 327.

(10) Anderson, H. C. *J. Comput. Phys.* **1983**, *52*, 24.

(11) Weiner, S. J.; Kollman, P. A.; NGuyen, D. T.; Case, D. A. *J. Comput. Chem.* **1986**, *7*, 230.

(12) Jorgensen, W. L.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1988**, *110*, 1657.

(13) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*, 6127.

(14) Hasel, W.; Hendrickson, T.; Still, W. C. *Tetrahedron Comput. Method* **1988**, *1*, 103.

(15) Hollinger, F.; Still, W. C. Unpublished work, 1993.

(16) Monge, A.; Lathrop, E. J. P.; Gunn, J. R.; Shenkin, P. S.; Friesner, R. A. *J. Mol. Biol.* **1995**, *247*, 995.

(17) McDonald, D. Q.; Still, W. C. *Tetrahedron Lett.* **1992**, *33*, 7743.

(18) Ferguson, D. M.; Kollman, P. A. *J. Comput. Chem.* **1991**, *12*, 620.

JP9502464