Implementation and Testing of a Frozen Density Matrix−Divide and Conquer Algorithm

Maria D. Ermolaeva, Arjan van der Vaart, and Kenneth M. Merz, Jr.*

Department of Chemistry, 152 Davey Laboratory, The Pennsylvania State University, University Park, Pennsylvania 16802

Received: November 5, 1998

We have implemented and tested a frozen density matrix (FDM) approximation to the basic divide and conquer (DC) semiempirical algorithm. Molecular dynamics and Monte Carlo simulations were performed to estimate the advantages of the method. Results were compared to those obtained from the original DC method and the combined quantum mechanical/molecular mechanical (QM/MM) method. We found that the FDM approximation speeds DC calculations up significantly, while only introducing small errors. We also found that the FDM DC scheme performs better that the standard QM/MM approach in terms of defining the electronic and structural properties of the systems studied herein.

Introduction

Quantum mechanical methodologies have proven to be extremely powerful in furthering our understanding of molecular systems ranging from small organic molecules to moderately sized biologically relevant molecules.[1−3] However, with current generation methodologies it has been very difficult to study larger systems using "fully" quantum mechanical approaches due to computational bottlenecks that scale with the third power of the number of orbitals or even higher.[1−3] This realization has led to the rapid development of linear-scaling quantum mechanical methodologies and their application to semiempirical, density functional or Hartree−Fock based Hamiltonians.[4−8]

Recently a semiempirical molecular orbital method that scales linearly with the number of orbitals was developed in our lab.[9,10] This density matrix based divide and conquer (DC) method[4] divides the system into subsystems which overlap, thereby allowing for electronic information to be exchanged between the subsystems. This particular division brings the density matrix, describing the electron distribution of the system, into a blocklike form with information for a given subsystem concentrated around the diagonal. Physically, this corresponds to a neglect of bonding between pairs of atoms that never appear in the same subsystem. Division of the system into subsystems has two computational advantages. Instead of one large matrix, a number of small matrices need to be diagonalized, decreasing the overall order of the calculation from N^3 to $\sim N$, with N being the number of orbitals. The second advantage is the inherent parallel nature of the DC method. Parallelization of our DC program has been successfully performed, increasing the computation speed significantly.[11]

Our DC method has been tested on a large number of systems giving results virtually within the same accuracy of standard semiempirical methods at only a fraction of the computational expense.[9−11] Since the DC method scales only linearly with the number of orbitals, QM calculations of large systems have become feasible. Still, the expense of a DC calculation is too great to do dynamical studies of large systems within a reasonable time period.

In this work we present a method which combines the DC method with a frozen density matrix (FDM) approximation[12−15] in order to make calculations on large molecules more computationally efficient, while still retaining a full QM representation. The FDM approximation exploits the 'fact that in most molecular simulations high accuracy or a high update frequency is only needed for a small "active" part of the system.[16] For example, this could be the active site of a protein or regions of relatively fast dynamics within a system. The Fock matrix of this part of the system is built and diagonalized at every step, thereby generating a new density matrix, while the rest of the system can be handled with less accuracy or with a lower update frequency using a previously evaluated density matrix. The FDM-DC approach highlights an advantage DC has over other linear-scaling semiempirical approaches:[7,17] since DC divides the system into a series of subsystems, the FDM approach is relatively easy to implement into the DC framework, while with other approaches it might not be as easy to implement in a general way.

An alternative to this FDM-DC method is the combined quantum mechanical molecular mechanical (QM/MM) approach[18−23] in which the fast or high-accuracy part of the system is treated with a QM method, while the remaining part is treated with classical mechanics. Recently, this method has been successfully applied by a number of research groups worldwide;[18−23] however, the quantum mechanical nature of the FDM-DC method has significant advantages over the classical representation used for the MM region in QM/MM simulations. Important phenomena like charge fluctuations, polarization, and charge transfer, which are not dealt with easily by classical methods, are included in a natural manner in our FDM-DC calculations. Moreover, difficulties with defining the quantum/classical boundary through the use with "link" atoms[21] can be either eliminated or partially mitigated by moving the link atom(s) further away from the region where high accuracy is desired. The downside, of course, is the memory requirements (i.e., one must store the coefficient matrix and eigenvalues for each frozen subsystem) of the FDM-DC approach. However, with ever larger capacity machines becoming generally available, this is only a minor consideration in the long run.

In this work we have examined the FDM-DC algorithm for two common types of simulations. We have performed short molecular dynamics simulations on l-*N*-acetyloctaalanine-*N*-methylamide in the α-helical conformation, and we have

10.1021/jp984312o CCC: $18.00
Published on Web 03/06/1999

performed Monte Carlo simulations on a "box" of 64 water molecules. In the next sections we briefly outline the formalism of the FDM-DC approach and we then describe the results of the MD and MC simulations.

Theoretical Background

The DC method[4,9,10] divides a molecular system into overlapping subsystems. For each of these subsystems, α, the localized Roothaan−Hall equation (here given as the semiempirical formulation, i.e., the overlap matrix is the identity matrix)

$$F^{\alpha}C^{\alpha} = C^{\alpha}E^{\alpha} \tag{1}$$

is solved; where C^{α} is the subsystem coefficient matrix, F^{α} the subsystem Fock matrix, and E^{α} the diagonal matrix of orbital energies for subsystem α. The local density matrix for subsystem α is built from the local coefficient vector:

$$P^{\alpha}_{\mu\nu} = \sum_{N^{\alpha}}^{i} n_i^{\alpha}(c^{\alpha}_{\mu i})^* c^{\alpha}_{\nu i} \tag{2}$$

where N^{α} is the number of orbitals in subsystem α and the factors n_i^{α} are occupation numbers which depend on the molecular orbital energy ϵ_i^{α} and the Fermi energy ϵ_F which is determined iteratively subject to the constraint that the total number of electrons in the system is conserved.

$$n_i^{\alpha} = \frac{2}{1 + \exp[(\epsilon_i^{\alpha} - \epsilon_F)/kT]} \tag{3}$$

The local Fock matrix is built from the global Fock matrix according to

$$F^{\alpha}_{\mu\nu} = \begin{cases} F_{\mu\nu} & \text{if the basis functions } X_\mu \text{ and } X_\nu \text{ are in the same subsystem} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The global Fock matrix is constructed from the one-electron matrix $H_{\mu\nu}$, the two-electron integrals $(\mu\nu/\lambda\sigma)$ and $(\mu\sigma/\lambda\nu)$ and the global density matrix $P_{\mu\nu}$:

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda=1}^{N}\sum_{\sigma=1}^{N}[(\mu\nu|\lambda\sigma) - {}^1/_2(\mu\sigma|\lambda\nu)]P_{\lambda\sigma} \tag{5}$$

Every subsystem consists of a core surrounded by two buffer layers.[10] The function of these buffer regions is to determine what information from the local density matrices should be used to build the global density matrix:

$$P_{\mu\nu} = \sum_{\alpha=1}^{n_{\text{sub}}} D^{\alpha}_{\mu\nu} P^{\alpha}_{\mu\nu} \tag{6}$$

$$D^{\alpha}_{\mu\nu} = \begin{cases} 1/n_{\mu\nu} & \text{if } \chi_\mu \text{ is in the core and } \chi_\nu \text{ is in the core or inner buffer region} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The most expensive step in this process is solution of the localized Roothaan−Hall equations (see eq 1) for each of the subsystems contained in the system of interest. This set of equations is solved in an iterative way until self-consistency is reached and whenever the system is altered in any way (i.e., the molecular geometry is changed). The coefficient matrix will change during this calculation, which reflects the adaptation of the wave function to the change in molecular coordinates.

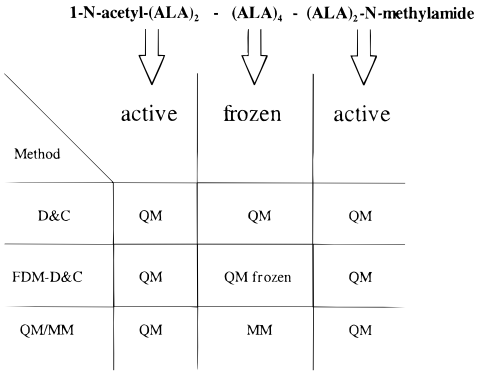

Figure 1. 1-*N*-Acetyloctaalanine-*N*-methylamide molecule, divided into frozen and active parts. The active part is treated with the PM3 semiempirical quantum mechanical method. The way in which the energy and forces of the frozen part are calculated depends on the method used. They are treated fully quantum mechanically, quantum mechanically but the density matrix is updated every few MD steps, or as a molecular mechanical region entirely in a QM/MM calculation. Moreover, the coordinates of the frozen region are either fixed or allowed to move. See the text for further details.

In the FDM approximation it is assumed that the wave function is constant in those regions of the system that were not affected by the perturbation. This means that the coefficient matrix of those "frozen" subsystems can be held constant during the perturbation, making the computationally expensive reevaluation of eq 1 superfluous for these subsystems. Since the coefficient vectors of the frozen subsystems need to be stored, the FDM approximation increases memory usage significantly. However, storage of the eigenvectors also allows us to determine the Fermi energy directly at every iteration step.[9,10] Another important feature of the redetermination of the Fermi energy (and, hence, the subsystem molecular orbital occupation numbers, see eq 3) at every step of a FDM-DC calculation is that it allows electron density to flow from a frozen to a nonfrozen region (or vice versa). The effect of this is clearly illustrated in Figures 4 and 5 discussed in detail below. This nice feature allows for a much more realistic representation of the boundary region between frozen and nonfrozen subsystems as opposed to how this region is handled in the QM/MM approach, for example.

Computational Procedure

First, we tested the FDM-DC method with short molecular dynamics (MD) simulations of α-helical 1-*N*-acetyloctaalanine-*N*-methylamide. We divided this molecule into three parts (see Figure 1): The middle region of the molecule was treated as frozen, while the ends of the helix were considered to be the active part of the system. The Fock matrix of the active part was updated at every step of the MD simulation. The Fock matrix of the frozen part of the molecule was updated only once every 10−30 steps. All calculations were performed with the PM3 semiempirical Hamiltonian.[24−26] In contrast with other implementations of the FDM approximation,[12,13] the frozen region in our calculations does not necessarily have to have fixed coordinates. We performed two different series of runs: In one case the coordinates of the frozen part were fixed and in the other the whole molecule was allowed to move.

All simulations started from the PM3 minimized-structure and the length of each MD run was 0.1 ps, using a time step of 0.1 fs. A short time step was used to ensure that the trajectories

TABLE 1: Number of Cycles and Translation/Rotation Steps Used in the Non-FDM and FDM MC Simulation To Obtain the Speed-Ups

	no. of cycles[a]		no. of steps	
n/*s*[b]	non-FDM	FDM	non-FDM	FDM
$n = 1$	2	19	128	1216
$n = 2$	2	39	64	1248
$n = 4$	4	59	64	944
$n = 6$	5	79	55	869
$n = 8$	8	99	64	792
$n = 16$	17	119	68	476
$n = 32$	35	139	70	278
$s = 1$	2	29	66	957
$s = 2$	4	59	68	1003
$s = 3$	5	29	55	319
$s = 4$	8	39	72	321
$s = 8$	17	39	85	185
$s = 16$	35	39	105	87

[a] The number of cycles with omission of the first, initialization cycle. One (1) cycle is when every molecule has been perturbed once. [b] The number of molecules perturbed per step in the molecule-wise method (*n*), or the number of subsystems perturbed per step in the subsystem-wise method (*s*).

were completely stable so that we could accurately compare the simulations among one another. Longer time steps (0.5 and 1 fs) produced trajectories that were not stable enough to allow for a detailed analysis of the trajectories for both the DC and FDM-DC simulations. Constant temperature (300 K) was maintained with the Nosé-Hoover chain algorithm.[27] For our simulations we coupled our DC program (DivCon)[9,10] with the molecular dynamics package ROAR 1.0.[28] This program will be released in the future as ROAR 2.0.[29] We carried out fully QM (DC) and FDM-DC MD simulations and the same set of simulations were performed using the QM/MM approach.[30] At every time step, gradients and energies of the QM subsystem were calculated with the DC method (DivCon) using the PM3 Hamiltonian. Gradients and energies of the MM subsystem were calculated with the AMBER force field.[31] Interactions between QM and MM atoms consist of the attraction of the MM atom "cores" with the electron cloud of the QM atoms and repulsion between the MM and QM cores. As usual a Lennard-Jones term was also added to obtain better accuracy.[30] Here we treated the middle region of the molecule, which was frozen in our FDM-DC calculations, as the MM system, while the rest of the molecule was treated quantum mechanically. The link atom approach[21] was used to connect the QM and MM regions to one another as is typical practice.[30,32] These link atoms were located along QM/MM bonds and were treated as QM hydrogen atoms. They are invisible to the MM atoms and no interactions between the link atoms and the MM atoms were calculated. Both the FDM-DC and QM/MM results were compared to the fully QM (DC) calculations.

As an another test of the FDM-DC method we performed short DC Monte Carlo (MC) simulations on a box of water molecules (64 water molecules total). In these simulations the system is evolved by the Metropolis scheme[33,34] with freedom of translation and rotation. Since our main goal here was to establish how a FDM approximation affects the computation time and accuracy of a DC Monte Carlo simulation, only short simulations were performed (Table 1). Sampling was clearly not enough to assess thermodynamic or other system properties. In a forthcoming publication we will describe converged DC Monte Carlo runs of liquid water using a modified semiempirical Hamiltonian.[35]

Our system consisted of 64 water molecules fixed at the PM3-optimized gas phase geometry in a cubic box of 13.481 × 13.481 × 13.481 Å. Periodic boundary conditions were in effect, and the particle mesh Ewald method[36] was used to treat the long-range electrostatic interactions. In our implementation, this long-range contribution was obtained as a sum of the direct and reciprocal energy, minus the classical Coulomb interactions of the CM1 charges.[37] The NPT ensemble with a temperature of 298.15 K and pressure of 1 bar was simulated and the subsetting (i.e., obtaining new subsystems that reflect the new coordinates of the box of waters) was redone every 10th cycle. The maximum allowed translation was 0.05 Å, the maximum allowed rotation was 7.5 deg, and the maximum change in the box dimensions was 0.05 Å. All calculations were done with the PM3 Hamiltonian.

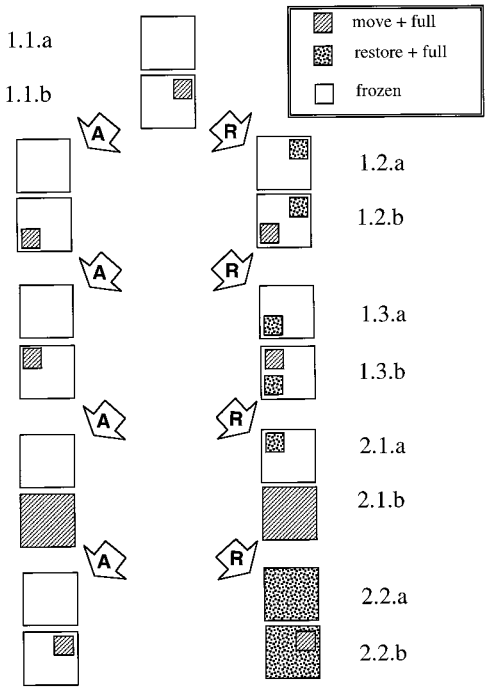

Figure 2. At every step in a FDM-DC Monte Carlo simulation a full diagonalization is only performed on the subsystems that change. Step l.l.a represents the system after a full diagonalization (initial configuration or after volume perturbation). In step 1.l.b, a number of random residues are selected that will be rotated/translated. The subsystem(s) in which these residues are located are indicated by the small gray box. These systems will be fully diagonalized, and all the others will be kept constant (frozen). If this configuration is rejected (indicated by R) in step 1.2.a, the molecules in these subsystems will be reset and a full diagonalization of these subsystems is required in step 1.2.b, since the old eigenvectors and eigenvalues were discarded to save on memory. This rediagonalization is not necessary if the configuration was accepted (indicated by A). In step 1.2.b a new group of random residues is selected, and the process is repeated. In step 2.1.b a volume perturbation is performed, after which every subsystem needs to be diagonalized. Finally, in step 2.2 the whole process is repeated. Further details are given in the text.

In our simulations, a change of the box volume and the corresponding scaling of the coordinates was performed after all the molecules had been rotated and translated. These rotations and translations were done in a number of steps, each step only perturbing a part of the system. Since only a part of the system changes at these rotation/translation steps, the calculation can be sped up by the FDM approximation, as illustrated in Figure 2. At every step only the perturbed subsystems are rediagonalized. If step n is rejected, the system is reset to the $n - 1$ configuration before the $n + 1$ configuration is generated. In this case the eigenvectors and eigenvalues of the subsystems that were perturbed in step n also have to be recovered in step

$n + 1$ since they were discarded to form step $n + 1$ (i.e., the old eigenvectors and eigenvalues were discarded to save on memory). We rediagonalize these subsystems in the same step, in order to save computation time. This means that the number of frozen subsystems decreases after a configuration has been rejected.

We tested two different FDM-MC-DC strategies. In the first, "moleculewise" method, the perturbations were performed on randomly selected molecules. Only subsystems with the perturbed molecules in the core of the subsystem were rediagonalized. We also tested a "subsystemwise" strategy in which a perturbation was not performed on random molecules but rather on the core molecules of randomly selected subsystems.

Speedups were obtained by performing short FDM-DC and non-FDM-DC MC simulations starting from the same initial configuration By the very nature of MC, these FDM-DC and non-FDM-DC trajectories may be very different. Since the FDM-DC calculation will give a (slightly) different electronic energy than the non-FDM-DC calculation, a configuration that is rejected in the non-FDM-DC run might be accepted in the FDM-DC run, and vice versa. Since subsequent configurations were generated by random translations/rotations, completely different trajectories may arise even though the differences between the FDM-DC and non-FDM-DC energies at the initial configuration was very small. We therefore performed another set of simulations to estimate the accuracy of the FDM approximation in our MC runs.

To estimate the accuracy of the FDM approximation in our MC simulations, we calculated the differences in electronic energy between FDM-DC and non-FDM-DC calculations at every rotation/translation step of sample simulations. For every configuration *n*, we first performed a FDM-DC calculation and then a non-FDM-DC calculation on the same configuration to obtain the difference in electronic energy. The energy difference between the non-FDM-DC energies of configuration *n* and $n - 1$ was subsequently used to accept or reject configuration *n*. We saved either all the non-FDM-DC eigenvectors and eigenvalues of configuration *n* if configuration *n* was accepted, or we recovered the non-FDM-DC eigenvectors and eigenvalues of configuration $n - 1$ if configuration *n* was rejected. These saved eigenvalues and eigenvectors were subsequently used for the frozen subsystems of the FDM-DC calculation of the next configuration $n + 1$. In this way, FDM-DC energies were obtained in a consistent manner, i.e., not depending on either acceptance or rejection of the previously generated configuration. Other than in the speedup runs, the percentage of frozen subsystems will remain constant during the energy-comparison runs and will not decrease when a configuration is rejected. Our error analysis, therefore, identifies the *maximum* error.

Results and Discussion

Molecular Dynamics of α-Helical 1-*N*-Acetyloctaalanine-*N*-methylamide. We performed two different sets of simulations. In the first case the coordinates of the frozen part of the molecule were fixed. Figure 3 shows the potential energy during the simulation for the DC (bold line), FDM-DC (solid line), and QM/MM (dashed line) methods. The Fock matrix of the frozen region was updated every 10 steps. The potential energy given by the FDM-DC approach tracks closely to the energy obtained by the DC calculations. The FDM-DC method also allows us to observe charge fluctuations between different regions (i.e., subsystems) of the molecule. Figures 4 and 5 show typical charge (Coulson) fluctuations for a region of the molecule when using DC (bold line), FDM-DC (solid line) and QM/MM (dashed line) methods. The FDM-DC approach realistically allows for charge changes in the frozen part of the molecule (Figure 4) and it also makes the charge distribution in the active part (Figure 5) more accurate, compared to the standard QM/MM method with a link atom.

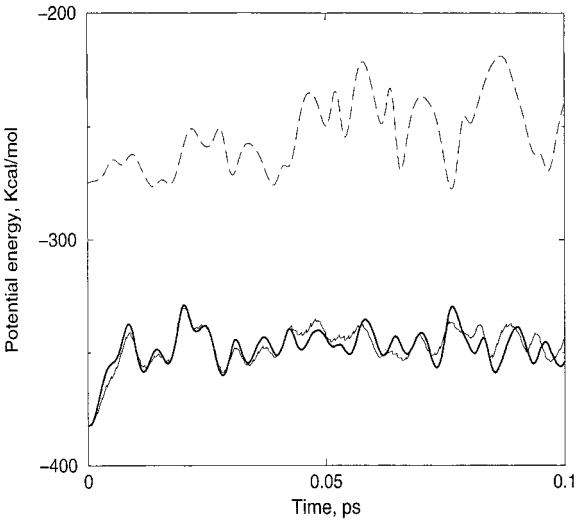

Figure 3. Potential energy of the 1-*N*-acetyloctaalanine-*N*-methylamide α-helix during molecular dynamics simulations. The atomic coordinates of the frozen part were fixed. The simulations were done using DC (bold line), FDM-DC with a density matrix update frequency of 10 steps (solid line), and QM/MM (dashed line) methods.

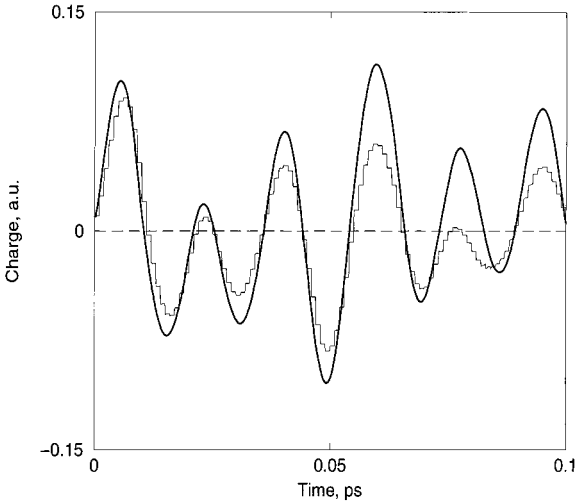

Figure 4. Charge fluctuations on the sixth Ala residue during a molecular dynamics simulation of the 1-*N*-acetyloctaalanine-*N*-methylamide α-helix. This amino acid is located in the frozen region of the molecule. The simulations were done using DC (bold line), FDM-DC (solid line), and QM/MM (dashed line) methods. The atomic coordinates of the frozen part were fixed.

We performed the same FDM-DC simulation where we updated the Fock matrix of the frozen region only once every 30 steps. Figure 6 (dashed line) shows the potential energy versus time for this case and the potential energy obtained from the DC simulation is shown by the bold line. The error introduced by the FDM approximation can be observed as sharp regular peaks on the plot with amplitudes close in magnitude to the amplitudes of the natural energy fluctuations observed at the full QM level. This means that the updating for this system is too infrequent and should not be used for simulations in which better accuracy is desired. Moreover, updating every 30 time steps only marginally speeds up the FDM-DC calculation (see Table 2 below).

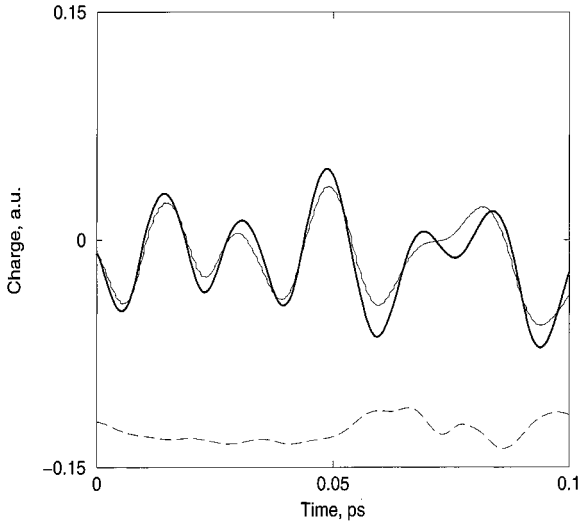

Figure 5. Charge fluctuations on the seventh Ala residue during a molecular dynamics simulation of the 1-*N*-acetyloctaalanine-*N*-methylamide α-helix. This amino acid is located in the active region of the molecule. The simulations were done using DC (bold line), FDM-DC (solid line) and QM/MM (dashed line) methods. The atomic coordinates of the frozen part were fixed.

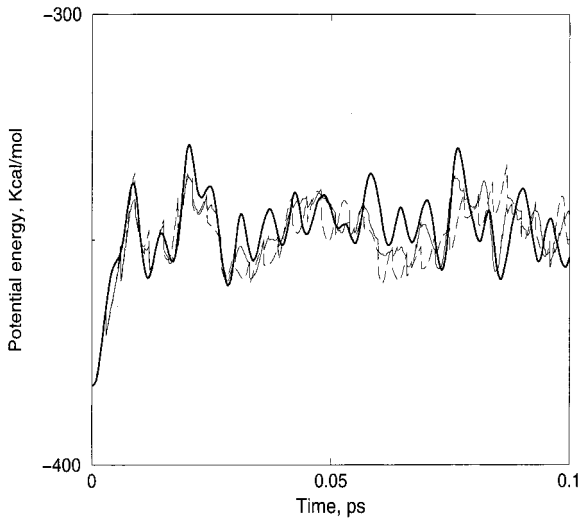

Figure 6. Potential energy of the 1-*N*-acetyloctaalanine-*N*-methylamide α-helix during a molecular dynamics simulation. The atomic coordinates of the frozen part were fixed. The simulations were done using FDM-DC with a density matrix update frequency of 30 steps (dashed line) and FDM-DC with dual updating (30 steps for Ala 4 and *5* and 10 steps for Ala 3 and 6, solid line) methods. The potential energy−time dependence from the DC simulation is indicated by the bold line.

We also performed a simulation in which the update frequency was a function of the distance of the frozen region from the active region. The results of such a dual updating scheme are shown by the solid line in Figure 6. We updated the density matrix for the two central alanine amino acids every 30 steps and the rest of the frozen subsystem every 10 steps. The results are better than updating every 30 steps for all of the frozen subsystems, which suggests that this is a potentially useful way in which to proceed. However, the small size of this test system does not allow us to assess how well it might work in much larger systems where its use might be quite advantageous. Nonetheless, we expect that a multiple frequency update strategy will prove to give significant advantages for big systems such as proteins.

Results for the set of simulations in which all atoms were allowed to move were similar to the ones in which the coordinates of the frozen region were fixed. Surprisingly, though, the results obtained by the DC and FDM-DC methods in this case are in even better accord. Moreover, the update frequency of the density matrix for the frozen subsystem could be lowered without severely affecting the accuracy. For example, updating the frozen density matrix every 30 steps (Figure 7, solid line) results in the potential energy being rather close to that obtained from the full DC calculations (Figure 7, bold line).

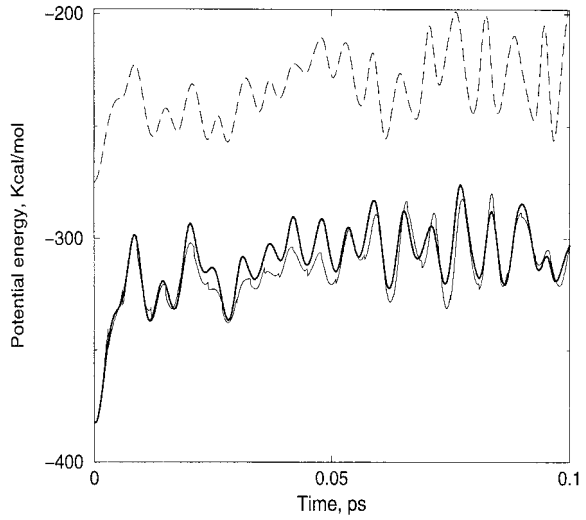

Figure 7. Potential energy of the 1-*N*-acetyloctaalanine-*N*-methylamide α-helix during a molecular dynamics simulation. All atoms of the molecule were allowed to move. The simulations were done using DC (bold line), FDM-DC with a density matrix update frequency of 30 steps (solid line), and QM/MM (dashed line) methods.

TABLE 2: CPU Times for MD Runs Using DC and FDM-DC Methods

method	frozen coordinates	update frequency	CPU (h)	theoretical[a] CPU (h)
DC	fixed	1	6.76	
FDM-DC	fixed	10	3.20	3.14
FDM-DC	fixed	20	2.94	2.95
FDM-DC	fixed	30	2.89	2.88
DC	updated	1	6.72	
FDM-DC	updated	10	3.16	3.13
FDM-DC	updated	20	2.98	2.93
FDM-DC	updated	30	2.93	2.87

[a] Estimation of the theoretical CPU times was done by determining that (1) for this system the Fock matrix build step takes 11% of the total CPU time and (2) the Fock matrix diagonalizations for the middle four subsystems takes 67% of the remaining CPU time. We determined from timings of the subsystem diagonalization steps that the central four residues take 67% of the diagonalization time due to the fact that the central subsystems Fock matrices are larger than the end ones. All timings were done on a SGI Origin 200.

The CPU times of our simulations are shown in Table 2. We expected that a frozen density update frequency of once every *n* steps should give a speed-up of *n* in the part of the calculation involving the frozen subsystems. Our observed speed-ups are quite close to those calculated theoretically based on this criteria. The reason for a slightly lower speed-up (up to 2%) is a slight increase in the number of diagonalization steps needed in the FDM calculation in order to reach convergence. Another important aspect of the data in Table 2 is the observation that the speed-up on going from updating every 10 steps to 30 steps is quite minimal. This suggests that updating every 10 steps is the best compromise in terms of accuracy and in terms of overall speed up of the calculation.

From this study we also observe that the QM/MM method is deficient in terms of accurately defining the electronic charac-

TABLE 3: Errors in the Electronic Energy (kcal/mol) for the FDM Simulations (Errors Are Reported as Total Errors for the 64-Water System)

n/s[a]	n_c[b]	no. of config	% frozen	av error[c]	max error[c]	% < kT[d]	90%<[e]
$n = 1$	64	64	96.9	0.0600	0.7240	98.4	0.1244
$n = 2$	32	64	93.8	0.1157	0.8056	96.9	0.2325
$n = 4$	16	64	87.7	0.1836	0.8184	96.9	0.3998
$n = 6$	11	66	82.5	0.2164	1.0432	92.4	0.4785
$n = 8$	8	64	76.4	0.2912	1.5992	90.6	0.5904
$n = 16$	4	64	54.2	0.3054	1.8539	88.9	0.6234
$n = 32$	2	64	23.0	0.2966	1.2241	96.7	0.5488
$s = 1$	32	66	97.0	0.1234	1.8197	98.5	0.2075
$s = 2$	16	64	93.9	0.2579	2.3775	95.3	0.4294
$s = 3$	11	66	90.9	0.2482	1.3478	95.5	0.4695
$s = 4$	8	64	87.9	0.4829	2.4968	75.0	1.1257
$s = 8$	4	63	75.8	0.7636	3.2731	61.9	1.8117
$s = 16$	2	61	51.5	1.1141	2.9776	21.3	2.1684

[a] The number of molecules perturbed per step in the molecule-wise method (n), or the number of subsystems perturbed per step in the subsystem-wise method (s). [b] The rounded number of rotation/translation steps per cycle. [c] Only unsigned errors are reported. [d] Percentage of configurations with errors less than kT. [e] Maximum error for the 90% lowest error configurations.

teristics of this model system. For example, the charge fluctuation between residues as a function of time is not captured and we also find that charge is built up at the QM/MM boundary (see Figures 4 and 5). Thus, we conclude that when one carries out a QM/MM simulation the boundary between the QM and MM region should be as far away as possible from the regions where the highest accuracy is required. Moreover, the FDM-DC approach could further alleviate this link-atom problem by allowing one to move the QM-MM linking region further away from the region where the highest accuracy is required.

Monte Carlo Simulations of 64 Water Molecules. We performed all Monte Carlo simulations using an automated subsetting algorithm that generated subsystems which had a core that consisted of two neighboring water molecules and one buffer region of 5.5 Å. A single buffer region was used in this case (instead of our usual dual buffering approach[10]) because this approach gave the best energies and performance in test calculations than all other schemes we tried. Note that all buffer-core cross terms were included in this scheme (i.e., the buffer region here is like our standard buffer region 1[10]). For all configurations, this resulted in a total of 33 subsystems. Energy evaluations of a number of randomly selected snap shots showed that the DC energies of these 64 water molecule configurations were within 0.2 kcal of non-DC (i.e., standard diagonalization scheme) calculations. Since all molecules were perturbed when the volume was adjusted, the FDM approximation could not be used for these steps. Thus, no speed-up can be achieved for these steps.

Errors in the electronic energy as introduced by the FDM approximation are listed in Table 3. For the Monte Carlo method in which randomly selected molecules are perturbed, average unsigned errors are smaller than 0.31 kcal/mol and the maximum unsigned errors are smaller than 1.9 kcal/mol for the 64-water system. Moreover, for all simulations the error in the electronic energy is below kT for ∼90% of the configurations. Errors for the subsystem-wise Monte Carlo simulations are also listed in Table 3. For perturbations on a small number of subsystems per step (1−3 subsystems), these errors are only slightly larger than the errors in the molecule-wise runs in which a small number of molecules (1−6) were perturbed. Errors increased when more subsystems were perturbed per step. For simulations in which 8 subsystems were perturbed, the maximum unsigned error is 3.3 kcal/mol with 90% of the configurations having an error under 1.8 kcal/mol. For the run in which 16 subsystems were perturbed per step, these values are 3.0 and 2.2 kcal/mol, respectively.

Errors in the FDM-DC approximation are a complicated function of the number of molecules that were perturbed and the percentage of subsystems that were treated frozen. Since the chance is high that randomly picked molecules do not all belong to the same subsystems, the percentage of frozen subsystems is higher in the subsystem-wise method than in the molecule-wise method at the same number of rotation/translation steps per cycle (Table 3). Because the error in the FDM approximation will decrease when less subsystems are treated frozen, the error in the molecule-wise method is smaller than in the subsystem-wise method.

Another important error factor in the FDM approximation are buffer atoms. Note that in our implementation subsystems are only rediagonalized when core atoms are perturbed. Since subsystems overlap, there may be subsystems in which only buffer atoms are perturbed. For the molecule-wise method these subsystems will predominantly occur for simulations in which a small number of molecules were perturbed per step. Luckily, the resulting error was small, since only a small perturbation was applied to a very small fraction of these subsystems (the chance that a subsystem contains multiple perturbed atoms is quite small). For a larger number of perturbed molecules per step, the relative occurrence of these subsystems decreases, since the chance that a subsystem contains multiple perturbed atoms, including core atoms, increases.

For the subsystem-wise method, all perturbations were performed on the core molecules of randomly picked subsystems. Since these cores are made up from molecules that are geometrically close together, the chance that neighboring subsystems will have multiple perturbed buffer atoms becomes relatively large. This means that the error introduced by subsystems in which only buffer atoms were perturbed is larger than in the molecule-wise method. Moreover, the relative number of these subsystems does not decrease with an increasing number of perturbations per step, since a subsystem has either all core atoms perturbed, or none. To minimize this error, we implemented a strategy in which subsystems were also rediagonalized when buffer atoms were perturbed. Although the error was decreased to ∼0, we also noted that this strategy essentially eliminated the key benefit of the FDM approximation: speed-up. No speed-ups were observed when using this algorithm and further investigation of its use was stopped as a result.

In contrast with the error, speed-up is a simple function of the percentage of frozen subsystems. Since the subsystem-wise method perturbs all the core molecules of randomly picked subsystems at once, every subsystem will only be diagonalized once per cycle. In the molecule-wise method, some subsystems will be diagonalized more often, since the core molecules may be perturbed in more than one step. This is the reason why the subsystem-wise method is usually faster than the molecule-wise method.

Table 4 shows the observed speedups for the rotation/translation steps for the two different Monte Carlo methods (molecule-wise and subsystem-wise). Speed-ups vary between 1.2 and 12.4, depending on the number of frozen subsystems. Note that the percentage of frozen subsystems decreased in these production runs compared to the error analysis runs, since more diagonalizations were performed after a configuration was rejected. Figure 8 shows a plot of the speed-up against the

TABLE 4: Speed-up for the Rotation/Translation Steps of the FDM-DC MC Simulations

n/s^a	speed-up	% frozen[b]	n/s^a	speed-up	% frozen[b]
$n = 1$	12.4	95.2	$s = 1$	11.8	92.1
$n = 2$	8.3	90.9	$s = 2$	7.3	84.0
$n = 4$	5.0	81.2	$s = 3$	5.1	76.5
$n = 6$	3.8	71.8	$s = 4$	4.1	69.0
$n = 8$	3.0	63.8	$s = 8$	2.3	49.0
$n = 16$	2.0	39.4	$s = 16$	1.2	17.5
$n = 32$	1.7	15.8			

[a] The number of molecules perturbed per step in the molecule-wise method (n), or the number of subsystems perturbed per step in the subsystem-wise method (s). [b] The percentage of subsystems that were treated frozen in the FDM simulation.

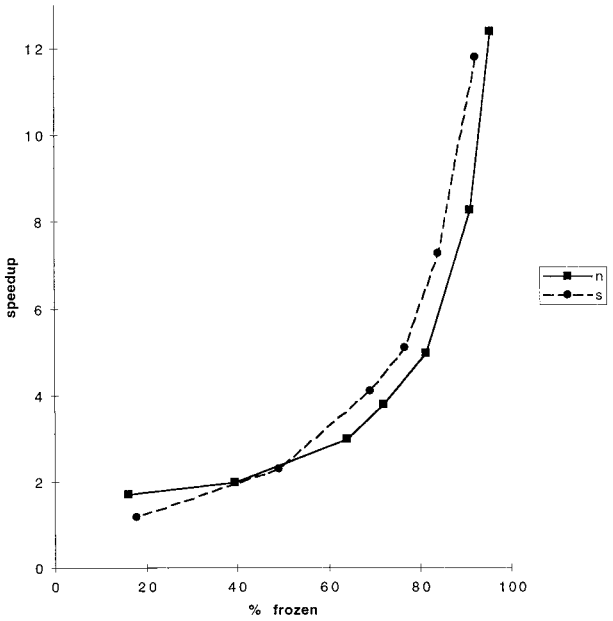

Figure 8. Speed-ups for the FDM-DC-MC simulations compared to the non-FDM-DC-MC simulations plotted against the percentage of subsystems that were treated frozen. Solid line marked with n is the speedup for the molecule-wise MC, and the dotted line marked with s is the speed-up for the subsystem-wise MC.

percentage of subsystems that were frozen. From this figure it can be seen that the speedup is nonlinear. This nonlinearity is caused by the formal cubic expense of the diagonalization routine. Speedups for small Monte Carlo moves (perturbation of 1−6 molecules or 1−3 subsystems per step) are limited by the time spent in building the Fock matrix (see Table 5). Construction of this matrix cannot be sped up by the FDM approximation and contributes a constant factor to the SCF time. For larger Monte Carlo moves, the percentage of the SCF time spent in building the Fock matrix reaches the non-FDM percentage.

Table 6 compares the two FDM Monte Carlo methods by combining timings and results of the speed-up and error analyses. For a large number of rotation/translation steps per cycle, i.e., the perturbation of a small number of molecules (2−6) or subsystems (1−3), it is best to perform a subsystem-wise Monte Carlo simulation. Speed-ups and timings were almost a factor of 1.5 better while the errors were similar for both methods. For a smaller number of rotation/translation steps per cycle, the molecule-wise Monte Carlo is preferred. The speed-up difference was smaller for these cases, but the error in the subsystem-wise Monte Carlo was larger than that observed for the molecule-wise Monte Carlo simulation.

TABLE 5: Percentage of SCF Time Spent in Building the Fock Matrix

n/s^a	% Fock, non-FDM[b]	% Fock, FDM[c]	n/s^a	% Fock, non-FDM[b]	% Fock, FDM[c]
$n = 1$	2.2	26.0	$s = 1$	2.2	25.3
$n = 2$	2.2	17.9	$s = 2$	2.1	15.2
$n = 4$	2.1	10.6	$s = 3$	2.1	10.9
$n = 6$	2.1	7.8	$s = 4$	2.1	8.6
$n = 8$	2.1	6.3	$s = 8$	2.1	4.8
$n = 16$	2.1	4.2	$s = 16$	2.2	2.8
$n = 32$	2.2	3.5			

[a] The number of molecules perturbed per step in the molecule-wise method (n), or the number of subsystems perturbed per step in the subsystem-wise method (s). [b] The percentage of SCF time spent in building the Fock matrix for the rotation/translation steps in a non-FDM DC MC simulation. [c] The percentage of SCF time spent in building the Fock matrix for the rotation/translation steps in a FDM DC MC simulation.

TABLE 6: Comparison of the Molecule-Wise and Subsystem-Wise FDM-MC Methods[a]

	molecule-wise					subsystem-wise				
n_c	n	t_c	s_S	e_{av}	e_{kT}	s	t_c	s_s	e_{av}	e_{kT}
32	2	248.9	8.3	0.1157	96.9	1	180.9	11.8	0.1234	98.5
16	4	218.5	5.0	0.1836	96.9	2	159.7	7.3	0.2579	95.3
11	6	205.3	3.8	0.2164	92.4	3	150.9	5.1	0.2482	95.5
8	8	195.4	3.0	0.2912	90.6	4	145.1	4.1	0.4829	75.0
4	16	153.2	2.0	0.3054	88.9	8	157.4	2.3	0.7636	61.9
2	32	94.3	1.7	0.2966	96.7	16	137.9	1.2	1.1141	21.3

[a] The definitions of the abbreviations used in this table are as follows: n_c = the rounded number of rotation/translation steps per cycle; n = the number of randomly picked molecules per step; t_c = the average time per cycle in sec. for the rotation/translation steps; s_s = speed-up per rotation/translation step compared to a non-FDM simulation; e_{av} = the average total error in electronic energy in kcal/mol; e_{kT} = the percentage of configurations with total unsigned error in electronic energy less than kT in kcal/mol; s = the number of randomly picked subsystems per step.

Conclusions

The FDM-DC method is a powerful approach that significantly decreases the computation time required in a DC calculation. The method keeps part of the system frozen; i.e., the Fock matrix and density matrix describing this subsystem are updated less frequently than for the rest of the system. In this paper, we have demonstrated the performance of this approach in terms of reducing CPU times in DC calculations, but it does have one drawback. The amount of memory required to store the eigenvectors and eigenvalues increases greatly with system size. For the peptide systems studied herein we were able to run them on a machine with only 128 MB of RAM (peptide system uses ∼15 MB), and the 64-water simulations were performed on a large memory machine (2 GB) and used ∼500 MB. For much larger systems (e.g., ∼60 residues surrounding an active site of a protein), much larger amounts of RAM will be required (∼2 GB for the example system). A disk-based procedure (i.e., saving the coefficients and eigenvalues on disk and reading them in) could be used but in core storage is clearly the preferred approach.

We have found that atoms in frozen subsystems may be either fixed or allowed to change their coordinates. We tested both strategies in our MD simulations of an α-helical 1-*N*-acetyl-octaalanine-*N*-methylamide. Both methods resulted in significant speed-ups, while introducing only a minor error. However, physically it was more advantageous to update the coordinates of all atoms at every step in a molecular dynamics simulation. Moving the frozen atoms without updating the Fock matrix

every step introduces some error, but, again, our calculations show that this error was rather small. In particular, the errors incurred were a lot smaller than the errors given by the QM/MM method.

Updating the coordinates of all atoms, at every time step, becomes even more important when this FDM-DC is used, for example, in molecular dynamics simulations of enzymes and other macromolecules in aqueous solution. In such simulations the active site of the enzyme could be treated with a "full" DC quantum mechanical calculation, while the area surrounding the active site region could be treated with the help of FDM-DC approach. Making the update frequency of the frozen density matrix region a function of distance from the active site will allow one to speed up the calculation without significantly affecting the active site dynamics and energetics. Finally, atoms that are far enough from the active site could then be treated with classical molecular mechanics methods as in a standard QM/MM simulation. Using this strategy will remove the perturbative effect of the MM region away from the critical active site region.

A method in which the coordinates of frozen subsystems remain fixed is the FDM-DC Monte Carlo method. Here the computational advantage of the FDM approximation is based on the fact that only a part of the system is changed during a Monte Carlo step. The FDM approximation speeds up the computation time of a DC Monte Carlo simulation significantly, introducing an error under kT for a system of 64 waters. It is computationally advantageous to perturb the system subsystem-wise when a large number of rotation/translation steps per cycle is desired. Simulations where a small number of rotation/translation steps per cycle are used requires the use of the molecule-wise approach in order to reduce the error in electronic energy.

In summary, the FDM-DC approach gives a significant speedup in both the DC-MD and the DC-MC simulations. Speed-up is limited by a couple of factors. First, in going from non-FDM to FDM calculations, the number of diagonalization steps increases slightly in order to reach convergence. Second, the FDM approach does not speed up the Fock matrix build step which accounts 11% of the SCF time for the peptide system, and 3−25% of the SCF time for the water system. At the same time, building the Fock matrix does not dramatically limit the advantages of the FDM method because it can be sped up with parallelization.

Acknowledgment. We thank the DOE (DE-FGO2-96ER62270) and Oxford Molecular for supporting this research. We also thank the Pittsburgh Supercomputer Center, the San Diego Supercomputer Center, the National Center for Supercomputer Applications, and the Cornell Theory Center for generous allocations of supercomputer time.

References and Notes

(1))Hehre, W. J.; Radom, L.; Schleyer, P. v. R.; Pople, J. A. *Ab Initio Molecular Orbital Theory*; John Wiley & Sons: New York, 1986.

(2) Parr, R. G. *Annu. Rev. Phys. Chem.* **1983**, *34*, 631−656.

(3) Pople, J. A.; Beveridge, D. L. *Approximate Molecular Orbital Theory;* McGraw-Hill: New York, 1970.

(4) Yang, W.; Lee, T.-S. *J. Chem. Phys.* **1995**, *103*, 5674−5678.

(5) White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. *Chem. Phys. Lett.* **1994**, *230*, 8−16.

(6) Strain, M. C.; Scuseria, G. E.; Frisch, M. J. *Science (Washington)* **1996**, *271*, 51−53.

(7) Daniels, A. D.; Millam, J. M.; Scuseria, G. E. *J. Chem. Phys.* **1997**, *107*, 425−431.

(8) Lee, T.-S.; York, D. M.; Yang, W. *J. Chem. Phys.* **1996**, *105*, 2744−2750.

(9) Dixon, S. L.; Merz, K. M., Jr. *J. Chem. Phys.* **1996**, *104*, 6643−6649.

(10) Dixon, S. L.; Merz, K. M., Jr. *J. Chem. Phys.* **1997**, *107*, 879−893.

(11) Vincent, J. J.; Merz, K. M., Jr. *Theor. Chem. Acc.* **1998**, *99*, 220−223.

(12) Wesolowski, T.; Warshel, A. *J. Phys. Chem.* **1994**, *98*, 5183−5187.

(13) Wesolowski, T. A.; Warshel, A. *J. Phys. Chem.* **1993**, *97*, 8050−8053.

(14) Cortona, P. *Phys. Rev. B* **1991**, *44*, 8454−8466.

(15) Lee, T. S.; Yang, W. A., to be published.

(16) Mazur, A. K. *J. Phys. Chem. B* **1998**, *102*, 473−479.

(17) Stewart, J. J. P. *Int. J. Quantum Chem.* **1996**, *58*, 133−146.

(18) Field, M. J.; Bash, P. A.; Karplus, M. *J. Comput. Chem.* **1990**, *11*, 700−733.

(19) Stanton, R. V.; Hartsough, D. S.; Merz, K. M., Jr. *J. Comput. Chem.* **1994,** *16*, 113−128.

(20) Gao, J. *Methods and Applications of Combined Quantum Mechanical and Molecular Mechanical Potentials*; Gao, J., Ed.; VCH Press: New York, 1995; Vol. 6.

(21) Warshel, A.; Levitt, M. *J. Mol. Biol.* **1976**, *103*, 227−249.

(22) Luzhkov, V.; Warshel, A. *J. Comput. Chem.* **1992,** *13*, 199−213.

(23) Théry, V.; Rinaldi, D.; Rivail, J. L.; Maigret, B.; Ferenczy, G. G. *J. Comput. Chem.* **1994**, *15*, 269−282.

(24) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 221−264.

(25) Stewart, J. J. P. *J. Comput. Chem.* **1991**, *12*, 320−341.

(26) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 209−220.

(27) Cheng, A.; Merz, K. M., Jr. *J. Phys. Chem.* **1996**, *100*, 1927−1937.

(28) Cheng, A.; Stanton, R. S.; Vincent, J. J.; Damodaran, K. V.; Dixon, S. L.; Hartsough, D. S.; Best, S. A.; Merz, K. M. J. *ROAR*, 1.0 ed.; Cheng, A., Stanton, R. S., Vincent, J. J., Damodaran, K. V., Dixon, S. L., Hartsough, D. S., Best, S. A., Merz, K. M. J., Eds.: The Pennsylvania State University: University Park, PA, 1997.

(29) Ermolaeva, M.; van der Vaart, A.; Cheng, A.; Stanton, R. S.; Vincent, J. J.; Damodaran, K. V.; Dixon, S. L.; Hartsough, D. S.; Best, S. A.; Merz, K. M. J. *ROAR* 2.0 ed.; Ermolaeva, M., van der Vaart, A., Cheng, A., Stanton, R. S., Vincent, J. J., Damodaran, K. V., Dixon, S. L., Hartsough, D. S., Best, S. A., Merz, K. M. J., Eds.: The Pennsylvania State University: University Park, PA, 1998.

(30) Hartsough, D. S.; Merz, K. M., Jr. *J. Phys. Chem.* **1995**, *99*, 11266−11275.

(31) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M., Jr.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*, 5179−5197.

(32) Bash, P. A.; Field, M. J.; Davenport, R. C.; Petsko, G. A.; Ringe, D.; Karplus, M. *Biochemistry* **1991**, *30*, 5826−5832.

(33) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: Oxford, UK, 1987.

(34) Metropolis, N.; Ulam, S. *J. Am. Stat. Ass.* **1949**, *44*, 335−341.

(35) van der Vaart, A.; Brothers, E.; Merz, K. M., Jr., to be published.

(36) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A *J. Chem. Phys.* **1995**, *103*, 8577−8594.

(37) Storer, J. W.; Giesen, D. J.; Cramer, C. J.; Truhlar, D. G. *J. Comput.-Aided Mol. Des.* **1995**, 9, 87−110.