
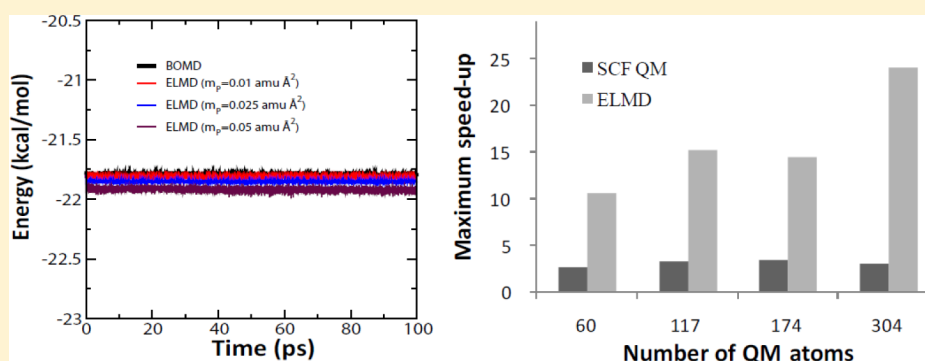


Acceleration of Semiempirical Quantum Mechanical Calculations by Extended Lagrangian Molecular Dynamics Approach

Kwangho Nam*

Department of Chemistry and Computational Life Science Cluster (CLiC), Umeå University, 901 87, Umeå, Sweden

 Supporting Information



ABSTRACT: The implementation and performance of the atom-centered density matrix propagation (ADMP) [*J. Chem. Phys.* **2001**, *114*, 9758] and the curvy-steps (CURV) methods [*J. Chem. Phys.* **2004**, *121*, 1152] are described. These methods solve the electronic Schrödinger equation approximately by propagating the electronic degrees of freedom using the extended Lagrangian molecular dynamics (ELMD) simulation approach. The ADMP and CURV methods are implemented and parallelized to accelerate semiempirical quantum mechanical (QM) methods (such as the MNDO, AM1, PM3, MNDO/d, and AM1/d methods). Test calculations show that both the ADMP and the CURV methods are 2~4 times faster than the Born–Oppenheimer molecular dynamics (BOMD) method and conserve the total energy well. The accuracy of the ADMP and CURV simulations is comparable to the BOMD simulations. The parallel implementation accelerates the MD simulation by up to 28 fold for the ADMP method and 25 fold for the CURV method, respectively, relative to the speed of the single core BOMD. In addition, a multiple time scale (MTS) approach is introduced to further speed up the semiempirical QM and QM/MM ELMD simulations. Since a larger integration time step is used for the propagation of the nuclear coordinates than that for the electronic degrees of freedom, the MTS approach allows the ELMD simulation to be carried out with a time step that is larger than the time step accessible by the original ADMP and CURV methods. It renders MD simulation to be carried out about 20 times faster than the BOMD simulation, and yields results that are comparable to the single time scale simulation results. The use of the methods introduced in the present work provides an efficient way to extend the length of the QM and QM/MM molecular dynamics simulations beyond the length accessible by BOMD simulation.

1. INTRODUCTION

Electronic structure methods including ab initio quantum mechanical (QM) and density functional theory (DFT) methods have been successfully applied to study chemical processes found in many biological systems.^{1–3} However, the large expense of evaluating electron–electron integrals makes even the most efficient ab initio QM or DFT methods impractical for long molecular dynamics (MD) simulations. Many semiempirical QM methods are more than 3 orders of magnitude faster than ab initio QM and DFT methods. They include the MNDO,⁴ AM1,⁵ PM3,⁶ MNDO/d,⁷ and AM1/d^{8,9} methods, which are based on the neglect of diatomic differential overlap (NDDO) approximation, and the SCC-DFTB method¹⁰ based on the tight-binding approximation. Although they are generally considered less accurate than ab initio QM and DFT methods, they yield reasonable results in many biologically relevant reactions.^{11–15} Even in the case where they

fail drastically, the Hamiltonian and the parameters of the semiempirical methods can be reformulated to produce accurate results.^{9,16–20}

The semiempirical QM methods have been an important tool in theoretical studies of many chemical problems, such as charge transfer, bond formation and cleavage, and electronic excitation.^{12–15} In addition, they are the only practical choice for long hybrid quantum mechanical and molecular mechanical (QM/MM) simulations for the study of complex biological systems, due to their high efficiency.^{21–25} However, the semiempirical QM methods are too slow compared to the speed of the molecular mechanical force fields. Recently, two groups^{26,27} independently explored the possibility to use the graphics processing units (GPUs) together with the multi-

Received: February 14, 2013

Published: June 20, 2013

threaded CPUs to speed up the MNDO99²⁸ and the MOPAC2009²⁹ programs. Both groups were able to achieve about 1 order of magnitude speed-up using the hybrid multicore CPU/GPU implementations. However, this speed-up is only observed with molecular systems containing more than 1000 basis functions. For molecular systems with fewer than 500 basis functions (or roughly 200 atoms), which are often the size of the QM region in QM/MM simulation, the speed-up is small and the overhead for the communication between CPU and GPU is not negligible.

In this work, different approaches are taken to accelerate the semiempirical QM and QM/MM simulations. We apply the methods developed by Schlegel et al.^{30,31} and by Herbert and Head-Gordon.³² These methods solve the Hartree–Fock quantum mechanical equation approximately by treating the electron density matrix elements (or the parametric version of the electron density) as dynamic variables and propagating them along with the nuclear coordinates, using extended Lagrangian molecular dynamics (ELMD). In Schlegel et al.’s method,³⁰ which is called the “atom-centered density matrix propagation” (ADMP) method, the one-electron density matrix is directly propagated, instead of propagating the wave function as in the Car–Parrinello molecular dynamics (CPMD) simulation method,^{33,34} and a constraint is applied to preserve the idempotency of the electron density. In the Herbert and Head-Gordon’s method, the generalized electronic coordinates are introduced as the dynamic variables that parametrically update the electron density at each MD step.³² This method is called the “curvy-steps” (CURV) method. Here, these two methods are implemented at the NDDO-based semiempirical QM level and parallelized using the message passing interface (MPI). The MPI protocol is chosen for compatibility with existing MD simulation programs, such as CHARMM,³⁵ for hybrid QM/MM simulation.

The paper is organized as follows. In section 2, theoretical background is briefly reviewed for the semiempirical QM methods and the ADMP and CURV methods, respectively. We also describe the multiple time scale (MTS) extension of the ELMD approach, which is important in the hybrid QM/MM simulations. In section 3, we provide implementation details. In section 4, the results and discussion are presented. The paper is concluded in section 5.

2. THEORY

2.1. Semiempirical Self-Consistent Field (SCF) Molecular Orbital (MO) Methods. In the NDDO-based semiempirical QM method, the total electronic energy is determined by solving the Roothaan–Hall equation³⁶ (eq 1)

$$FC = CE \quad (1)$$

where F and C denote the Fock and eigenvector matrices and E is the diagonal matrix of molecular orbital energies. For a closed-shell system, the Fock matrix element is written as

$$F_{\mu_A \nu_B} = h_{\mu_A \nu_B} + \sum_{\lambda_C \sigma_C} P_{\lambda_C \sigma_C} (\mu_A \nu_B | \lambda_C \sigma_C) - \frac{1}{2} \sum_{\lambda_B \sigma_B} P_{\lambda_B \sigma_B} (\mu_A \sigma_A | \lambda_B \nu_B) \quad (2)$$

where $h_{\mu\nu}$ is the one-electron matrix element and $(\alpha\beta|\gamma\delta)$ represents the two-electron repulsion integral of orthonormal atomic orbitals. In the equation, the first summation runs over

all QM atoms, and λ_X and σ_Y run over all orbitals of atoms X and Y , respectively. The density matrix is defined by

$$P = 2CC^T, \quad (3)$$

and the electronic energy is expressed as

$$E_{\text{el}} = \frac{1}{2} \sum_{\mu\nu} P_{\mu\nu} (h_{\mu\nu} + F_{\mu\nu}) \quad (4)$$

In the self-consistent field (SCF) MO calculation, which is used to derive the energy and nuclear forces in Born–Oppenheimer molecular dynamics (BOMD) simulations,^{37–40} the calculation (eqs 1–4) is iterated until the total electronic energy is converged to a certain threshold level. In the semiempirical QM calculation, eq 1 is commonly solved by diagonalizing the Fock matrix, which constitutes the main computational bottleneck of the entire QM calculation.^{26,27} The extension of the SCF MO method (as well as the ADMP and CURV methods presented below) to an open-shell system is straightforward and not discussed any further in this work.

2.2. Atom-Centered Density Matrix Propagation (ADMP) Method. In the atom-centered density matrix propagation (ADMP) method,³⁰ the extended Lagrangian formula of the coupled electron density and nuclei system is given by

$$\mathcal{L} = \frac{1}{2} \sum_A M_A \left(\frac{d\mathbf{R}_A}{dt} \right)^2 + \frac{1}{2} \sum_{i,j} m_{p_i} \left(\frac{dp_{ij}}{dt} \right)^2 - E(\mathbf{R}, \mathbf{P}) - \text{Tr}[\mathbf{\Lambda}(\mathbf{P}\mathbf{P} - \mathbf{P})] \quad (5)$$

where \mathbf{R} and \mathbf{P} denote the nuclei coordinates and the one-electron density matrix, and M and m_p are the corresponding masses, respectively. The energy of the semiempirical QM methods is expressed as the sum of the electronic energy E_{el} and the core–core repulsion energy E^{core} of the molecule,

$$E(\mathbf{R}, \mathbf{P}) = E_{\text{el}} + \sum_{A < B} E_{AB}^{\text{core}} \quad (6)$$

The last term in eq 5 emerges by imposing a constraint on the idempotency of the electron density matrix (i.e., $\mathbf{P}\mathbf{P} = \mathbf{P}$), and $\mathbf{\Lambda}$ is the Lagrangian multiplier matrix.

The equations of motion of the ADMP method are

$$M \frac{d^2 \mathbf{R}}{dt^2} = - \frac{\partial E(\mathbf{R}, \mathbf{P})}{\partial \mathbf{R}} \quad (7)$$

for nuclei and

$$m_p \frac{d^2 \mathbf{P}}{dt^2} = - \left[\frac{\partial E(\mathbf{R}, \mathbf{P})}{\partial \mathbf{P}} + \mathbf{\Lambda} \mathbf{P} + \mathbf{P} \mathbf{\Lambda} - \mathbf{\Lambda} \right] \quad (8)$$

for electron density, respectively. Equation 7 is the same as the equations of motion of BOMD and can be solved by applying any MD algorithm. The electron density matrix can also be propagated by any MD algorithm. In this work, the Verlet algorithm⁴¹ is chosen for its simplicity for the propagation of electron density with the time step δt ,

$$\mathbf{P}(t + \delta t) = 2\mathbf{P}(t) - \mathbf{P}(t - \delta t) - \frac{\delta t^2}{m_p} \left[\frac{\partial E(\mathbf{R}(t), \mathbf{P}(t))}{\partial \mathbf{P}} + \mathbf{\Lambda}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{\Lambda}(t) - \mathbf{\Lambda}(t) \right] \quad (9)$$

and the leapfrog Verlet is used for the propagation of the nuclear coordinates, respectively. Equation 9 is solved by following the procedure described by Schlegel et al.,^{30,31} the electron density \mathbf{P} is first transformed to the McWeeny-purified form $(\tilde{\mathbf{P}} = 3\mathbf{P}^2 - 2\mathbf{P}^3)^{42}$ and then subjected to the iterative minimizations of $\text{Tr}[(\mathbf{P}\mathbf{P} - \mathbf{P})^2]$ until it converges below a certain threshold level (details can be found in refs 30 and 31). In this work, we use $\{\text{Tr}[(\mathbf{P}\mathbf{P} - \mathbf{P})^2]\}^{1/2}/N < 10^{-12}$ and $|\text{Tr}[\mathbf{P}] - N_e| < 10^{-12}$ as the convergence criteria, in which N and N_e are the number of atomic orbitals and the number of electrons of the molecule, respectively. Both trace terms converge rapidly and the convergence is reached within 2 or 3 iteration cycles in most cases.

Nuclear gradients require some discussion. In general, nuclear gradients involve two terms, the so-called Hellmann–Feynman term (or the integral derivative) and the wave function derivative term.⁴³ As has been discussed in detail by Schlegel et al.,³⁰ the second term arises because the transformation of the nonorthogonal atom-based basis functions, which are commonly used in the ab initio QM and DFT calculations, to the orthogonal basis is dependent on molecular geometry, and is not zero in the ab initio QM and DFT methods if the electron density is not converged to a variational minimum as in the ADMP and CURV methods. On the other hand, this term is zero by definition in the NDDO-based semiempirical QM methods, because the (unitary) transformation matrix is a unit matrix by the NDDO-orthonormality condition and thus does not depend on atomic positions.⁴⁴ This makes the evaluation of nuclear gradients simpler than the ab initio QM/DFT version of the ADMP and CURV methods.

2.3. Curvy-Steps ELMD Method. In the curvy-steps (CURV) approach proposed by Herbert and Head-Gordon,³² the density matrix is updated by the unitary-update scheme

$$\mathbf{P}(t + \delta t) = e^{\Delta(t+\delta t)} \mathbf{P}(t) e^{-\Delta(t+\delta t)} \quad (10)$$

where Δ is the dynamic variable matrix that is propagated by ELMD. Equation 10 is solved by means of the truncated Baker–Campbell–Hausdorff (BCH) commutator expansion scheme.⁴⁵ In the present work, the expansion is carried out until all matrix elements converge below 10^{-15} . This criterion is conservative enough to achieve good conservation of total energy during the MD simulation. The advantage of the CURV method over ADMP is that the idempotency of the electron density matrix is preserved automatically with the unitary-update scheme (eq 10). Then, the last term in eq 5 vanishes and the extended Lagrangian equation becomes

$$\mathcal{L} = \frac{1}{2} \sum_A M_A \left(\frac{d\mathbf{R}_A}{dt} \right)^2 + \frac{1}{2} \sum_{ij} m_{\Delta_{ij}} \left(\frac{d\Delta_{ij}}{dt} \right)^2 - E(\mathbf{R}, \mathbf{P}) \quad (11)$$

where m_{Δ} is the fictitious mass of the dynamic variable Δ . We have observed that $\{\text{Tr}[(\mathbf{P}\mathbf{P} - \mathbf{P})^2]\}^{1/2}/N < 10^{-12}$ and $|\text{Tr}[\mathbf{P}] - N_e| < 10^{-12}$ are satisfied in all simulations carried out in this study.

With the exponential update scheme of eq 10, the equations of motion of the electronic variables are

$$m_{\Delta} \frac{d^2 \Delta}{dt^2} = - \left. \frac{\partial E(\mathbf{R}, \mathbf{P})}{\partial \Delta} \right|_{\Delta=0} = -(\mathbf{F}\mathbf{P} - \mathbf{P}\mathbf{F}) \quad (12)$$

Since eq 12 is valid only when $\Delta = 0$ as indicated, the equation is solved by setting $\Delta = 0$ at each time step.³² Then, $\Delta(t + \delta t)$ is determined by applying the leapfrog Verlet algorithm,⁴¹

$$\frac{d\Delta(t + \delta t/2)}{dt} = \frac{d\Delta(t - \delta t/2)}{dt} - \frac{\delta t}{m_{\Delta}} \frac{\partial E(\mathbf{R}(t), \mathbf{P}(t))}{\partial \Delta} \quad (13)$$

and

$$\Delta(t + \delta t) = \delta t \frac{d\Delta(t + \delta t/2)}{dt} \quad (14)$$

with the initial condition that $\Delta(t) = 0$ at each time step and $d\Delta(t - \delta t/2)/dt = 0$ at $t = 0$. The nuclear coordinates are integrated using the leapfrog Verlet algorithm.⁴¹

2.4. Multiple Time Scale (MTS) Simulation. Two conditions must be met for the ELMD methods to be useful. The first condition is the adiabatic decoupling between the dynamics of nuclei and electronic degrees of freedom. The decoupling can be achieved by choosing the fictitious mass of the electronic variables carefully. With well-chosen fictitious mass, the electronic degrees of freedom stay at a temperature that is close to 0 K.^{31,32,34,46} Then, the electron density can equilibrate quickly around each nucleus and yield well-approximated solution to the SCF MO energy, driving the molecular system close to the correct BO potential energy surface throughout the entire simulation. The second condition is the computational efficiency; that is, the ELMD methods should be computationally more efficient than BOMD. Since the time step for BOMD can be 0.5 fs or larger for molecules carrying hydrogen atoms, the ELMD methods, if carried out with $\delta t = 0.1$ fs, should be at least 5 times faster than BOMD to compete with BOMD. However, the present implementation of the ADMP and the CURV methods is only 2~4 times faster than BOMD with the same time step (see the Results and Discussion section),⁴⁷ and the overall performance of the ELMD methods is slower than BOMD.

In this work, we take two approaches to achieve further speed-up of the ELMD methods. The first approach is the parallelization of the ELMD methods. Since the solution of the ADMP and CURV methods only involves simple matrix multiplications, they are well suited for parallelization. The results are presented in the Results and Discussion section. The second approach is the development of the multiple time scale (MTS) method. In the MTS approach, the nuclear degrees of freedom are solved with a large time step, for example, 0.5 fs, and the electronic degrees of freedom are solved with a small time step, for example, 0.1 fs. Since the QM/MM simulation spends most of the computation time evaluating the QM/MM interaction energies and gradients, including the periodic Ewald summation components,^{48,49} a substantial speed-up, which is roughly proportional to the ratio between the two time steps, is expected in the hybrid QM/MM simulation by evaluating the electron–electron integrals and gradients for nuclei with a large time step.

The MTS approach proposed in this work is illustrated as follows. First, consider that, at time t , the positions of nuclei of time t and $t - \Delta t$, $\mathbf{R}(t)$ and $\mathbf{R}(t - \Delta t)$, and the electron density, $\mathbf{P}(t - \Delta t)$, are known, and that the nuclear degrees of freedom are advanced with the time step Δt . Then, the main task in carrying out the MD simulation is the determination of $\mathbf{P}(t)$ to derive energy and gradients for the propagation of nuclear coordinates to $\mathbf{R}(t + \Delta t)$. If $\mathbf{P}(t)$ is determined by the SCF

procedure, this becomes BOMD. In the MTS approach, $P(t)$ is determined by numerically integrating eq 8 for ADMP and eqs 12 and 10 for the CURV method, respectively, with the small time step δt from $t - \Delta t$ to t . This can be solved exactly if $F(R(t'), P(t'))$ is known for $t - \Delta t \ll t' \ll t$ and nuclei positions are integrated with the δt time step. However, since R is known only at time t and $t - \Delta t$, either $R(t')$ or $F(R(t'), P(t'))$ has to be approximated at each δt time step. In this work, $F(R(t'), P(t'))$ is approximated by the linear interpolation between $F(R(t - \Delta t), P(t'))$ and $F(R(t), P(t'))$ as

$$F(R(t'), P(t')) \approx (1 - \lambda)F(R(t - \Delta t), P(t')) + \lambda F(R(t), P(t')) \quad (15)$$

where $\lambda = |(t' - (t - \Delta t))/(\Delta t)|$, and P is integrated over $N\delta t$ steps from $t - \Delta t$ to t ($\Delta t = N\delta t$).

In eq 15, $R(t - \Delta t)$ and $R(t)$ are used together with $P(t')$ to determine $F(R(t'), P(t'))$ approximately, in which $R(t')$ is implicitly defined. Although $R(t')$ can be determined explicitly, for example, by a linear interpolation between $R(t - \Delta t)$ and $R(t)$, this is not desirable because all interatomic distances and electron–electron integrals would have to be computed at every time step when the nuclear positions are updated, which are computationally demanding. On the other hand, eq 15 provides an efficient way to construct the Fock matrix ($F(R(t'), P(t'))$). Since this equation does not require the explicit determination of $R(t')$ at each δt step, the one-electron matrix elements and the two electron repulsion integrals are computed only once at every Δt time step (and saved for the next time step).

2.5. Comparison to Other Multiple Time Scale Approaches. Previously, several different multiple time scale approaches have been proposed. One example is the approach proposed by Gibson and Carter.⁵⁰ Here, we convert the Gibson and Carter's approach from the wave function based version to an electron density-based one for a consistent comparison with our approach. Gibson and Carter used the midtime point update scheme; that is, P is first propagated with the nuclear positions $R(t)$ over $N/2$ small time steps from t to $t + \Delta t/2$, and at $t + \Delta t/2$, the nuclear positions are updated to produce $R(t + \Delta t)$. Then, it resumes the propagation of P for $N/2$ time steps with $R(t + \Delta t)$ until it reaches $t + \Delta t$. In this method, P experiences a large change of nuclear positions around the $t + \Delta t/2$ time point. This sudden change of nuclear coordinates can lead to instability in long MD simulation, unless Δt is small enough to guarantee quick equilibration of P around each nucleus position.

Another example is the three time-step integration method proposed by Li et al.⁵¹ In their approach, they first propagate the nuclear velocity and position by using velocity Verlet algorithm⁴¹ with a large integration time step. The large time step is then subdivided into multiple intermediate time steps. At each intermediate time step, the nuclear position is interpolated to a midtime point, for example, $t + \Delta t/2$. Finally, the electron density is propagated, for example, from t to $t + \Delta t$, over N small time steps using the midtime point nuclear positions and the time-dependent Hartree–Fock theory.

Although these two approaches are similar to some extent to our MTS approach that instantaneous electron density is used to construct the Fock matrix for the propagation of electron density, details are different. In their approaches, the nuclear position is fixed during the propagation of electron density at a midtime point, thus providing a mean-field potential for the

update of electron density. By contrast, the time-interpolated Fock matrix (eq 15) varies smoothly from $t - \Delta t$ to t , and P is constantly equilibrated around each effective $R(t')$. This interpolation procedure mimics effectively the situation where P and R are integrated simultaneously with the δt time step. Alternatively, the method by Li et al.⁵¹ can be simplified if the small time step is taken to be the same as the intermediate time step. In this case, the nuclear position R that is needed to construct the instantaneous Fock matrix is determined by a linear interpolation between $R(t - \Delta t)$ and $R(t)$. This still differs from our MTS method, because the position is not determined explicitly at each δt step in the proposed MTS approach.

3. IMPLEMENTATION AND TEST CALCULATIONS

Both the ADMP and the CURV methods are implemented to the MNDO97⁵² module of the developmental version of the CHARMM program (version c38a1).³⁵ Although the MNDO97 program has been interfaced previously to the CHARMM program for the hybrid QM/MM calculations,⁴⁸ we have modified the program extensively in the present work. At the same time, we made efforts to implement a parallel version of the MNDO97 QM module, which was not parallel, and optimized it for efficient QM and QM/MM calculations. The semiempirical QM and QM/MM routines of the present rewritten MNDO97 module perform slightly better (by about 20–25% depending on the size of QM region and on the options for the QM-MM nonbonded interactions) than the original MNDO97 routines in carrying out the hybrid QM/MM calculations. The ADMP and CURV methods are then implemented on “top” of these changes and the results of these new methods are presented in the paper. The BOMD results presented here are based on the new MNDO97 module and not the old MNDO97 module, for consistent comparison with the ELMD results.

Two different computer systems are used in the test calculations. The timing calculations are carried out using the 12-core AMD 2.1-GHz processors. Each compute node contains 2 processors (thus 24 cores/node) and 32 GB DDR3 1333-MHz memory and is connected to neighboring nodes by the Cray Gemini Network. The CHARMM program is compiled using the Portland Group compiler (version 12.5) and the PGI programming environment (PrgEng-PGI Version 4.0.46) available on the Cray XE6 system. All other calculations are carried out using the computer system that contains 4 12-core AMD Opteron 2.6-GHz processors and 128 GB DDR3-1600 memory. The Intel FORTRAN compiler (version 13.0.0) and the Open MPI library (version 1.6.1) are used to compile the program on this computer system.

4. RESULTS AND DISCUSSION

4.1. Energy Conservation. The implementation and performance of the ADMP³⁰ and CURV methods³² are tested by carrying out NVE MD simulations for a butane molecule in the gas phase. All simulations are initiated at 300 K and run for 100 ps using 0.1 fs integration time step. The AM1 QM method⁵ is used in the simulation. The ADMP results show excellent energy conservation for the systems with the fictitious mass ≤ 0.05 amu Å² (Figure 1A). In the inset of Figure 1A, the total energy (E_{total}), which is defined as the sum of the kinetic energy of nuclei (KE_N), the fictitious kinetic energy of electronic variables (KE_p), the electronic energy (E_{el}), and

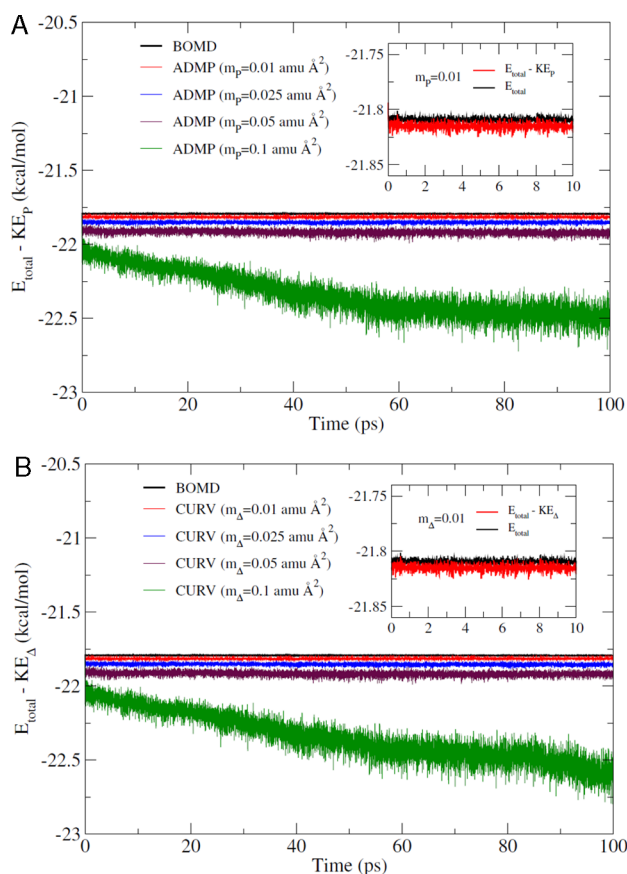


Figure 1. Energy fluctuations of butane molecule in the gas phase for (A) the ADMP method and (B) the CURV method. The NVE MD simulations are initiated at 300 K and run for 100 ps for each method. The integration time step is 0.1 fs. In the inset, the total energy (E_{total}) is compared together with the energy without the fictitious kinetic energy of electronic variables ($E_{\text{total}} - KE_p$) for the initial 10 ps. The fictitious mass has the unit of $\text{amu } \text{\AA}^2$.

the core–core repulsion energy (E^{core}), is shown together with $E_{\text{total}} - KE_p$ for the $m_p = 0.01 \text{ amu } \text{\AA}^2$ simulation. In this work, we denote the $E_{\text{total}} - KE_p$ term as the “real” molecular energy to distinguish it from the total energy (E_{total}), which includes the fictitious kinetic energy. The results suggest that for the fictitious mass $\leq 0.05 \text{ amu } \text{\AA}^2$, the dynamics of the nuclei and the electronic degrees of freedom are well separated from each other and both degrees of freedom behave quasi-adiabatically. For the simulation with $0.1 \text{ amu } \text{\AA}^2$ mass, the energy drifts away significantly from its initial value. We have made more detailed comparison by carrying out the simulations with various integration time steps and fictitious masses. The energy drifts of these simulations are compared in Table 1 and the average energies and standard deviations are presented in Figure 2A. The results show good energy conservation for masses $\leq 0.05 \text{ amu } \text{\AA}^2$ and integration time steps $\leq 0.25 \text{ fs}$. In particular, the energy conservation of the $0.025 \text{ amu } \text{\AA}^2$ mass system is excellent for various integration time steps.

The same comparison is made for the CURV method, and the results are summarized in Table 2. The results show that the energy conservation of the CURV method is very similar to that of the ADMP method over the range of the fictitious mass between $0.01 \text{ amu } \text{\AA}^2$ and $0.05 \text{ amu } \text{\AA}^2$. The fluctuation of the molecular energy (i.e., $E_{\text{total}} - KE_{\Delta}$) of the CURV method is also very similar to that of the ADMP method. In Figure S1A

Table 1. Conservation of the Real Molecular Energy ($E_{\text{total}} - KE_p$) of the Atom-Centered Density Matrix Propagation (ADMP) Method for a Butane Molecule^a

| m_p ($\text{amu } \text{\AA}^2$) | single time scale | | | | multiple time scale ($\Delta t = 0.5 \text{ fs}$) | |
|---|-------------------|--------|---------|--------------|--|------------------------------|
| | 0.05 fs | 0.1 fs | 0.25 fs | 0.5 fs | $\delta t = 0.1 \text{ fs}$ | $\delta t = 0.25 \text{ fs}$ |
| 0.01 | 0.003 | −0.005 | −0.01 | ^b | −0.03 | ^c |
| 0.025 | −0.028 | −0.015 | −0.017 | 0.002 | −0.011 | −0.025 |
| 0.05 | −0.27 | −0.13 | −0.25 | −0.27 | −0.24 | −0.23 |
| 0.1 | −5.6 | −6.7 | −6.3 | −5.7 | −8.0 | −7.4 |

^aAll MD simulations are carried out for 100 ps at the NVE ensemble. The initial velocities are generated at 300 K, and the AM1 model is used to represent the butane molecule. The drift of the energy is shown in $\text{kcal mol}^{-1} \text{ ns}^{-1}$. ^bThe idempotency iteration to minimize $\text{Tr}[(PP - P)^2]$ does not converge during the simulation, likely due to the large change of the density matrix P . This might occur because the time step is too large for this choice of mass to achieve stable dynamics. ^cThe energy deviates too much from the initial energy after about 50 ps.

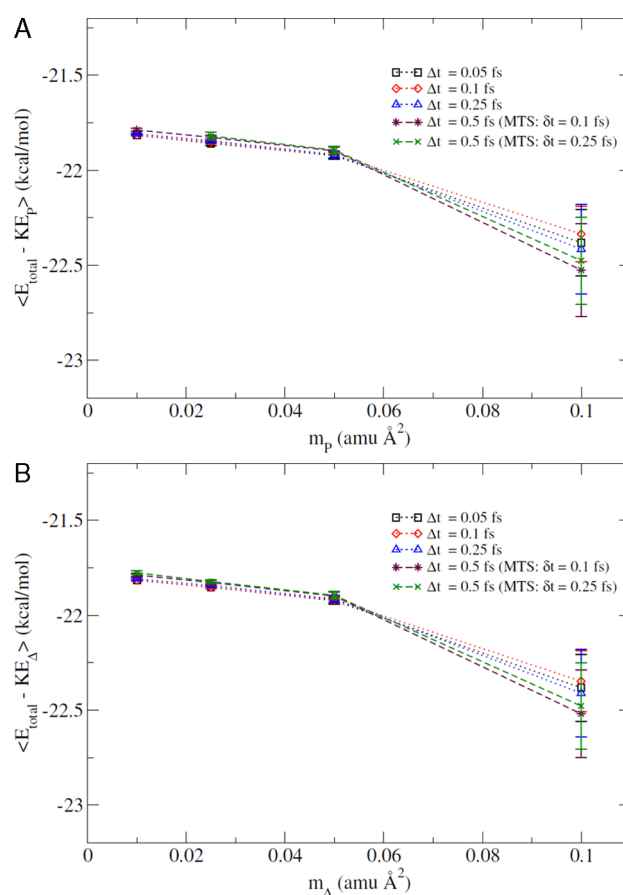


Figure 2. Average real molecular energies, $\langle E_{\text{total}} - KE_p \rangle$, and their standard deviations for various fictitious masses and integrations time steps: (A) the ADMP method and (B) the CURV method. The single time scale results are compared for 0.05 fs, 0.1 fs, and 0.25 fs, respectively, and the multiple time scale results are obtained from the simulations with the 0.5 fs integration time step for nuclear coordinates and 0.1 and 0.25 fs integration time steps for the electronic degrees of freedom, respectively.

(Supporting Information), the energy conservation of the CURV method is compared to that of the ADMP method with the same mass for the electronic variables. The two simulations

Table 2. Conservation of the Real Molecular Energy ($E_{\text{total}} - KE_{\Delta}$) of the Curvy-Steps (CURV) Extended Lagrangian MD Method for a Butane Molecule^a

| m_{Δ} (amu Å ²) | single time scale | | | | multiple time scale ($\Delta t = 0.5$ fs) | |
|---------------------------------------|-------------------|--------|---------|--------------|---|----------------------|
| | 0.05 fs | 0.1 fs | 0.25 fs | 0.5 fs | $\delta t = 0.1$ fs | $\delta t = 0.25$ fs |
| 0.01 | 0.008 | 0.011 | 0.007 | ^b | −0.041 | 0.047 |
| 0.025 | 0.004 | −0.061 | 0.003 | −0.057 | −0.14 | −0.009 |
| 0.05 | −0.27 | −0.11 | −0.22 | −0.25 | −0.27 | −0.25 |
| 0.1 | −5.7 | −5.2 | −6.1 | −5.5 | −7.5 | −7.5 |

^aAll MD simulations are carried out for 100 ps in the NVE ensemble. The initial velocities are generated at 300 K, and the AM1 model is used to represent the butane molecule. The drift of the energy is shown in kcal mol^{−1} ns^{−1}. ^bEnergy drift is too large from the beginning of the MD simulation.

show that the energy conservation of the two methods is almost identical to each other. In addition, we have tested the application of the McWeeny purification scheme (i.e., $\tilde{P} = 3P^2 - 2P^3$ as the input density for the energy and gradient evaluations)⁵³ to the CURV method. Whereas both simulations (simulations with and without the purification) are stable over the 100 ps simulation, the energy drift of the simulations with the McWeeny purification is relatively smaller than the simulation without the purification (Figure S2, Supporting Information). Nevertheless, the two simulations are almost indistinguishable during the initial 20 ps.

To characterize how well the ELMD methods reproduce the BOMD results, we compare the rates of density displacement (in ps^{−1}) for each MD simulation methods in Figure S3 (Supporting Information). In this analysis, the rate is used as a measure of how fast each density element changes during the simulation. Figure S3A (Supporting Information) shows that whereas the elements with large displacements are well reproduced by the ELMD methods, many elements with relatively small displacements in BOMD appear to be systematically up-shifted in both ELMD methods. On the other hand, there appears to be no difference between the ADMP and CURV methods and the effects of fictitious mass are small. In Figure S3B (Supporting Information), we present a two-dimensional map that plots the difference of the rates of density displacement between the ADMP (lower-left of Figure S3B, Supporting Information) and the CURV (upper-right) simulations and BOMD. The figure shows a number of

systematic differences between the ELMD and BOMD results. First, all diagonal elements are significantly different from the BOMD results. In particular, all hydrogen s-orbitals and carbon p-orbitals (2–4 diagonal elements of each carbon) show relatively large differences. Second, the difference is small for pairs involving carbon s-orbital and large for pairs between p-orbitals within each intra-atom block. Last, for the interatomic blocks, the difference appears to be dependent on the interatomic separation. Starting from atom pairs that are directly bonded to each other, for example C₁ and H₁ and C₁ and C₂ pairs, showing relatively small difference from the BOMD result, the deviation from the BOMD results increases as the interatomic distance increases between two atoms pairs. The results suggest that the use of nonuniform fictitious masses, which has been suggested for the ADMP method³¹ and for the Car–Parrinello density matrix search method,⁵⁴ could reproduce better the behavior of the electron density in BOMD. However, they need to be chosen based on the type of orbital pairs and their separation in molecule. This requires detailed analysis of the behavior of the electron density in BOMD.

The BOMD results presented in Figure 1 is from the simulation that is carried out with the electron density determined from previous MD step as the initial guess and with the SCF convergence criteria of 1.0×10^{-6} eV for the SCF energy and 1.0×10^{-6} for the diagonal of the electron density, respectively. In Figure S1B (Supporting Information), we also compare the energy conservations of BOMD with different SCF convergence criteria. As observed for the ab initio methods and the semiempirical QM methods,^{38,49} the energy starts drifting slowly with the SCF criteria lower than the values used above (Figure S1B). This suggests that the use of tight SCF convergence criteria is important to achieve stable BOMD simulations.

4.2. Performance. The computational efficiencies of the ADMP and the CURV methods are compared for the (AAQAA)_N α -helices with $N = 1, 2$, and 3 and the Trp-cage protein.⁵⁵ The number of QM atoms of these molecules range between 60 for (AAQAA)₁ and 304 for Trp-cage. For each method, 100 steps of MD simulation are carried out with 0.1 fs integration time step. The 0.025 amu Å² fictitious mass is used for both ELMD methods. The BOMD simulations are also performed with the same time step to provide the reference data. The AM1 model is used in all test calculations. The timings are presented in Table 3. The results show that the

Table 3. Computation Times (in seconds)^a for the 100 Steps of Molecular Dynamics Simulation with the Single Time Step 0.1 fs

| CPU | (AAQAA) _N | | | | | | | | | | | |
|-----|----------------------|-------|-------|-------------------|-------|-------|-------------------|-------|-------|----------------------|-------|--------|
| | N = 1 (60 atoms) | | | N = 2 (117 atoms) | | | N = 3 (174 atoms) | | | Trp cage (304 atoms) | | |
| | BOMD | ADMP | CURV | BOMD | ADMP | CURV | BOMD | ADMP | CURV | BOMD | ADMP | CURV |
| 1 | 22.6 | 6.6 | 11.1 | 159.5 | 45.6 | 73.6 | 496.8 | 154.3 | 232.4 | 3635.2 | 888.5 | 1357.8 |
| 16 | 8.8 | 1.8 | 2.1 | 54.1 | 9.5 | 10.5 | 176.6 | 27.3 | 38.7 | 1377.0 | 164.3 | 207.3 |
| | (2.6) | (3.7) | (5.3) | (2.9) | (4.8) | (7.0) | (2.8) | (5.7) | (6.0) | (2.6) | (5.4) | (6.5) |
| 32 | 11.5 | 2.4 | 2.4 | 61.3 | 10.9 | 11.3 | 169.7 | 40.0 | 33.4 | 1256.4 | 130.5 | 156.7 |
| | (2.0) | (2.8) | (4.6) | (2.6) | (4.2) | (6.5) | (2.9) | (3.9) | (7.0) | (2.9) | (6.8) | (8.7) |
| 64 | | | | | | | | | | 1458.8 | 147.7 | 146.8 |
| | | | | | | | | | | (2.5) | (6.0) | (9.2) |

^aThe number in parentheses is the speed-up with respect to the single CPU core result of each method (BOMD, ADMP, and CURV). All calculations are carried with the AM1 method as the quantum method. The mass of the electronic variables of the ADMP and CURV calculations is 0.025 amu Å².

ADMP method is 3.2~4.1 times and the CURV method is 2.0~2.7 times faster than BOMD, if compared to the BOMD results with the same time step (i.e., 0.1 fs).

Both ELMD methods can be accelerated further by parallelization. In Figure 3, the parallel scaling is compared

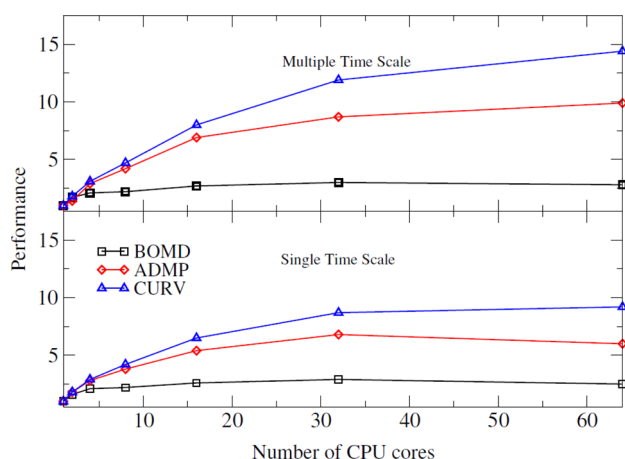


Figure 3. Parallel scaling of the Trp-cage protein. The performance of each method is defined as the parallel speed-up relative to the single core result of the corresponding method. The single time scale simulation results are presented in the lower panel, and the multiple time scale results are presented in the upper panel. The single time scale simulations are run with the 0.1 fs as the integration time step. The multiple time scale simulations are run with the 0.5 fs integration time step for the nuclear coordinates and the 0.1 fs integration time step for the electronic degrees of freedom, respectively. The fictitious mass used is 0.025 amu \AA^2 for both the ADMP and CURV methods. The Born–Oppenheimer MD results are from 0.1 fs integration time step for the lower panel and 0.5 fs integration time step for the upper panel, respectively.

for both ELMD methods as a function of the number of CPU cores for Trp cage. The figure shows that both the ADMP and the CURV methods scale up to 32 CPUs, before they are saturated. The saturation of the scaling occurs when the gain of performance by parallelization is compensated by the interprocessor communication cost. For the (AAQAA)_N systems, the saturation is found around 16 CPUs for both ELMD methods. Table 3 compares the computation times of different test molecules for each method. The performance is compared up to 32 CPUs for the (AAQAA)_N systems and up to 64 CPU cores for Trp cage, respectively. The maximum speed-up by parallelization is 3.7~6.8 for ADMP and 5.3~9.2 for the CURV method, respectively, when compared to the single core results of the corresponding ELMD method. If they are compared to the single core BOMD results, the maximum speed-up is much larger: 12.6~27.9 fold increase for ADMP and 10.8~24.8 fold increase for the CURV method.

The parallel scaling of the BOMD method is much worse than the two ELMD methods (Figure 3 and Table 2). This is partly due to the poor parallel scaling of the matrix diagonalization routines of the SCF MO method. In particular, we do not make much effort to improve the parallel efficiency of the matrix diagonalization routines in the present implementation, because it is expected that the SCF MO method scales poorly for the medium-sized molecule (<200 QM atoms) by the parallelization.⁵⁶

4.3. Multiple Time Scale Simulations. The multiple time scale (MTS) approach is tested by comparing the stability of

the NVE MD simulations. The simulations are carried out with the butane molecule in the gas phase and run for 100 ps by integrating the nuclear coordinates with 0.5 fs time step (large time step) and the electronic variables with 0.1 fs time step (small time step), respectively. Therefore, one large step integration of nuclear coordinates is accompanied by five small time step integrations of electronic variables to complete one MD step. The BOMD simulation is carried out with the 0.5 fs integration time step to provide reference data.

Figure 4A shows that the ADMP method conserves the energy well for the mass ≤ 0.05 amu \AA^2 . However, the

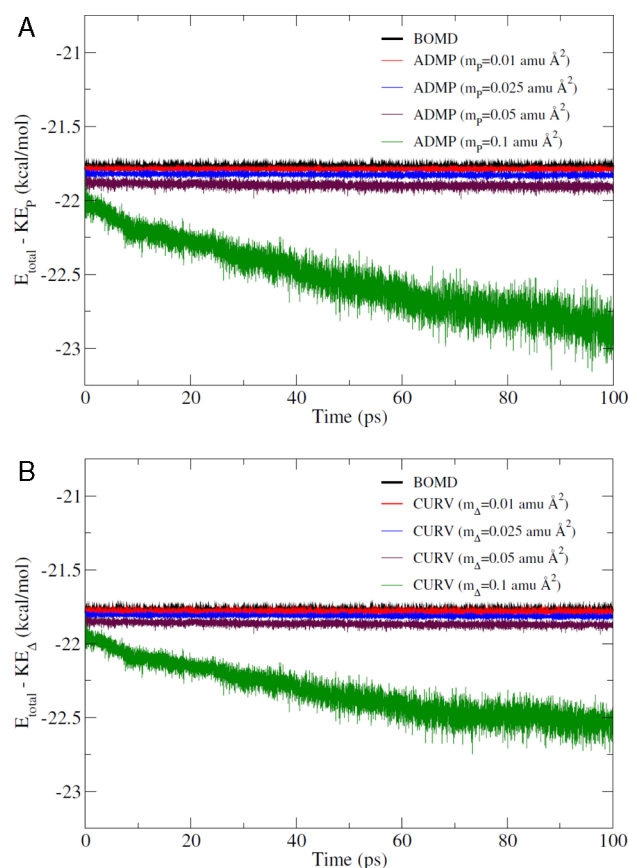


Figure 4. Energy fluctuations of butane molecule in the gas phase for the multiple times scale ELMD simulations: (A) the ADMP method and (B) the CURV method. The NVE MD simulations are initiated at 300 K and run for 100 ps for each method. The integration time step for the nuclear coordinates is 0.5 fs (including BOMD), and that for the electronic degrees of freedom is 0.1 fs.

fluctuation of the energy appears to be slightly increased for the MTS simulation relative to the corresponding single time step ADMP result (Figure 2A). Similar increase of the energy fluctuation is also observed for BOMD as the time step increases from 0.1 to 0.5 fs (compare Figures 1A and 4A). To test if the large energy fluctuation is due to the large integration time step used for the nuclear coordinates, a comparison simulation is carried out with the 0.25 fs time step for the electronic variables while keeping the 0.5 fs time step for nuclear degrees of freedom. The results are presented in Figure 2A and Table 1 and show that both MTS simulations, one with 0.1 fs and the other with 0.25 fs time step, have similar energy conservation and fluctuation. This implies that the increase of energy fluctuation is mainly due to the use of large integration

Table 4. Computation Times (in seconds)^a for the 100 Steps of Molecular Dynamics Simulation with the Multiple Time Scale Approach ($\Delta t = 0.5$ fs and $\delta t = 0.1$ fs).

| (AAQAA) _N | | | | | | | | | | | | |
|----------------------|------------------|-------|-------|-------------------|-------|-------|-------------------|-------|--------|----------------------|--------|--------|
| | N = 1 (60 atoms) | | | N = 2 (117 atoms) | | | N = 3 (174 atoms) | | | Trp cage (304 atoms) | | |
| CPU | BOMD | ADMP | CURV | BOMD | ADMP | CURV | BOMD | ADMP | CURV | BOMD | ADMP | CURV |
| 1 | 26.2 | 25.3 | 44.7 | 189.0 | 178.7 | 317.2 | 603.6 | 530.7 | 1055.3 | 4243.0 | 3635.9 | 6421.1 |
| 16 | 10.2 | 4.8 | 6.2 | 59.6 | 25.3 | 35.0 | 180.7 | 79.1 | 141.4 | 1565.3 | 529.4 | 802.6 |
| | (2.6) | (5.3) | (7.2) | (3.2) | (7.1) | (9.1) | (3.3) | (6.7) | (7.5) | (2.7) | (6.9) | (8.0) |
| 32 | 12.5 | 6.4 | 6.8 | 65.7 | 27.7 | 34.1 | 199.1 | 82.2 | 113.9 | 1416.4 | 415.8 | 538.6 |
| | (2.1) | (4.0) | (6.6) | (2.9) | (6.5) | (9.3) | (3.0) | (6.5) | (9.3) | (3.0) | (8.7) | (11.9) |
| 64 | | | | | | | | | | 1513.9 | 365.7 | 445.4 |
| | | | | | | | | | | (2.8) | (9.9) | (14.4) |

^aThe number in parentheses is the speed-up with respect to the single core result of each method (BOMD, ADMP, and CURV). All calculations are carried with the AM1 method as the quantum method. The mass of the electronic variables of the ADMP and CURV calculations is 0.025 amu Å².

time step for the nuclear coordinates and not the MTS approach. The energy conservation of the CURV method is qualitatively the same as the ADMP method (Figure 4B and Table 2). Taken together, the present results show that the MTS approach approximates reliably the single time scale ELMD simulation.

In Table 4, the computation times of the MTS ELMD simulation methods are compared for various test molecules, in which the ADMP and CURV simulations are performed with 0.5 fs time step for the nuclear coordinates and 0.1 fs time step for the electronic variables. In general, the BOMD with 0.5 fs time step takes a little longer time than that with 0.1 fs. Since the electron density that is determined from previous MD step is used as the initial guess at each time step, it is a poor initial guess for the simulations with a larger time step, and it takes more SCF iterations to reach the convergence. When the speeds of the ADMP and CURV MTS methods are compared to the speed of the 0.5 fs BOMD speed, the ADMP method takes roughly the same amount of computer time as the BOMD MD and the CURV method takes about 60% more (for the single CPU results; see Table 4). If the parallel performances are considered, both ELMD methods are much faster than the SCF QM method (BOMD). The maximum speed-up is about 5.5~11.6 for ADMP and 4.2~9.5 for the CURV method relative to the single core BOMD results, respectively. We expect another factor of 2 speed-ups for both ELMD methods, if 0.2 or 0.25 fs integration time step is used for the propagation of the electronic variables. The timing of the MTS simulations with 0.25 fs integration of the electronic variables are presented in Table S1 (Supporting Information), which shows the speed-up of 9.0~21.9 for ADMP and 7.9~17.8 for the CURV method, respectively. For medium-sized molecules (<200 QM atoms), the largest speed-up is achieved with 16~32 CPUs, and for the larger system (304 atom Trp cage), the largest speed-up is with 64 CPUs (also see Figure 3). It will be worthwhile in the future work to explore the hybrid openMP/MPI or the hybrid GPU/MPI parallelization for further speed-up of these methods.

4.4. Reaction: Potential of Mean Force. The umbrella sampling free energy simulations⁵⁷ are carried out for the SN₂ reaction between Cl[−] and CH₃Cl in the gas phase and the reaction free energies and barrier heights are compared to demonstrate the accuracy of the ELMD methods for studying chemical reaction. The AM1 model is used to represent the reacting molecules (Cl[−] and CH₃Cl). The reaction coordinate is defined as the difference of the distance between the central

C atom and the leaving Cl atom and the distance between the central C atom and the attacking Cl atom. A total of 61 simulation windows cover the entire range of the reaction coordinate from −3.0 Å to 3.0 Å. The simulations are carried out for 200 ps for each window with 0.1 fs integration time step. The temperature is maintained at 300 K using the Langevin thermostat. The BOMD umbrella sampling simulations are performed at 300 K with 0.1 and 0.5 fs as the time step, respectively. The potential of mean force (PMF) values are computed using the WHAM method.^{58,59}

The PMF results are presented in Figure 5A and summarized in Table S2 (Supporting Information). All three methods yield very similar PMF profiles. For example, the barrier height is 12.4 kcal/mol for both ELMD methods with 0.01 amu Å² as the fictitious mass, which is the same as the BOMD result with the same integration time step (Table S2, Supporting Information). We observe a slight mass dependent decrease of the barrier height for both ELMD methods. The barrier height decreases about 0.12 kcal/mol for ADMP and 0.08 kcal/mol for CURV, respectively, as the fictitious mass increases from 0.01 amu Å² to 0.05 amu Å². Although the effects of the fictitious mass on the PMF profile appear to be systematic, they are relatively small and are within the range of the simulation error (see the reaction free energies in Table S2 (Supporting Information), which by definition are zero for the studied SN₂ reaction but are on the order of 0.1 kcal/mol).

The PMF simulations are also carried out using the MTS approach with the fictitious mass of 0.025 amu Å². For each umbrella sampling window, the simulations are carried out for 250 ps by integrating the nuclear coordinates with 0.5 fs time step. Two different time steps, 0.1 and 0.25 fs, are used for the electronic degrees of freedom to examine the effects of the time step on the results. The results are compared to the single time scale ELMD results in Figure 5B. The figure shows that the MTS simulations with the 0.1 fs integration yield essentially the same results as the single time scale results for both ELMD methods. On the other hand, the CURV MTS with 0.25 fs integration produces the barrier height that is slightly lower (by 0.4 kcal/mol) than the single scale results, whereas the ADMP result is not significantly different from the single scale results. These results suggest that the MTS approach produces very reliable results and that the MTS methods, in particular, the ADMP method, are relatively insensitive to the choice of the integration time step of the electronic variables as well as the fictitious mass.

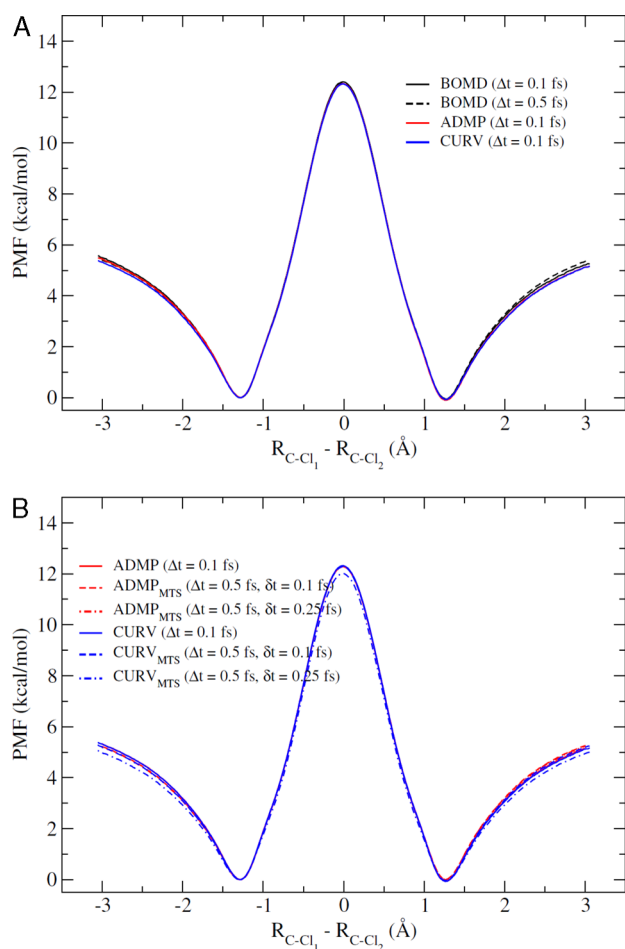


Figure 5. Potentials of mean force (PMFs) for the SN_2 reaction between Cl^- and CH_3Cl in the gas phase. The reaction coordinate is defined as the difference of the distances of the central C atom from the leaving Cl atom (Cl_1) and to the attacking Cl atom (Cl_2). The BOMD results are computed using the 0.1 and 0.5 fs integration time step, respectively. For the ADMP and CURV MD simulations, the fictitious mass of $0.025 \text{ amu } \text{\AA}^2$ is used. In part A, the ADMP and the CURV methods are carried out with 0.1 fs as the integration time step for both the nuclear coordinates and the electronic degrees of freedom. In part B, the multiple time scale (MTS) results are obtained with 0.5 fs as the integration time step for the nuclear coordinates and 0.1 and 0.25 fs integration for the electronic variables, respectively.

4.5. QM/MM Simulation of Trp Cage with Periodic Boundaries. The present ELMD methods provide an efficient way to perform QM/MM MD simulations. To demonstrate this capability, QM/MM MD simulations are carried out using the Trp cage protein.⁵⁵ The system has been set up by fully solvating the protein with 2800 waters, in which the entire Trp cage protein is treated by the AM1 quantum model⁵ and the water molecules by the TIP3P water model,⁶⁰ respectively. The rhombic dodecahedron periodic boundary condition is used with the 50 \AA lattice length. The simulation is carried out for 5 ps by using the ADMP method and the MTS approach with 0.5 fs as the time step for nuclear coordinates and 0.25 fs for the electron density. The fictitious mass for the electronic variables is $0.025 \text{ amu } \text{\AA}^2$. The entire simulation takes about 5 h using 48 CPU cores on the computer system that is used for the timing calculations. The temperature of the system is maintained at 300 K using the Langevin thermostat. Although the long-range electrostatic interactions can be evaluated by the QM/MM-

Ewald summation method,⁴⁸ the QM/MM electrostatics and the van der Waals interactions are evaluated with the cutoff method. The 10.0 \AA cutoff distance is used in the present QM/MM simulation.

In Figure 6, we show the deviation of the Mulliken charge⁶¹ of each QM atom from the corresponding MM charge of the

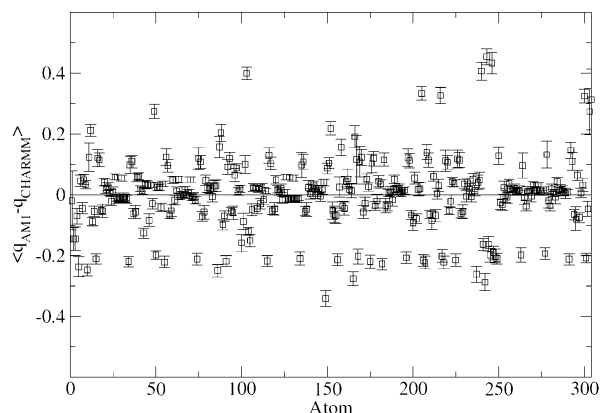


Figure 6. Deviation of the average atomic Mulliken charges from the CHARMM22 force field⁶² values of the Trp cage protein. The average and the standard deviation are computed for each atom, based on the 5 ps QM/MM ADMP simulation. The ADMP simulation is carried out using the multiple time scale approach with 0.5 fs integration for nuclear coordinates and 0.25 fs integration of electron density, respectively. The fictitious mass is $0.025 \text{ amu } \text{\AA}^2$.

CHARMM22 force fields.⁶² Although the majority of the atomic charges fall within 0.1 e from the reference force field values, many atoms have charges that are more than 0.1 e away from the reference values. For example, there is a collection of atoms that have charges differ by about -0.2 e from its force field value (Figure 6). These atoms are in fact the backbone carbonyl carbon atoms. Therefore, the QM/MM Mulliken charges of the backbone carbonyl carbons are less polar than the reference force field value (0.51 e). The carbonyl oxygen is also underpolarized by about 0.1 e from the value of the force field (-0.51 e). Although alternative procedures, such as charge partitioning,^{63–65} mapping,^{66–68} and fitting⁶⁹ schemes, could be used to produce better atomic charges than the Mulliken partitioning,⁶¹ the Mulliken charges are used in the present analysis, because of their simplicity. Therefore, the results should only be considered qualitative and not quantitative.

The QM/MM simulation has its utility to provide benchmark data for the development of accurate force fields. To show the possibility, we present in Figure S4 (Supporting Information) the average atomic charges obtained from the present AM1 QM/MM simulation. The figure shows that the atomic charges of the Trp cage protein are not randomly distributed but grouped together. Most of them are gathered around -0.4 e , -0.2 e , 0.0 e , 0.1 e , and 0.3 e values, and some are distributed between them. Only a small number of atoms are outside of $\pm 0.4 \text{ e}$. This grouping is to some extent consistent with the atom-typing of force fields and, thus, has the potential to be used to fine-tune the force field charges, possibly with better charge partitioning/fitting schemes. In addition, the present ELMD QM/MM simulation shows the possibility that this approach can be used as an alternative to force fields to carry out the MD simulations, because the static charge model used in force fields cannot capture exactly the effects of charge fluctuations during the simulation. Although several groups

have developed linear scaling approaches to treat entire protein quantum mechanically,^{70–74} our approach has the potential to be one of the most efficient approaches. In addition, the present methods can benefit from future work to combine with the linear scaling QM method^{70,74} and the long-range electrostatics models.⁴⁸

5. CONCLUSION

In this work, we have described the implementation and performance of (1) the atom-centered density matrix propagation (ADMP) method and (2) the curvy-steps (CURV) method. The two methods are implemented to the CHARMM program³⁵ using the semiempirical quantum mechanical methods, such as the MNDO,⁴ AM1,⁵ PM3,⁶ MNDO/d,⁷ and AM1/d^{8,9} methods as the QM methods. Although the two methods differ in the details of how the electronic variables are propagated, they yield simulations that are very stable over long period of time for rather broad range of fictitious masses and integration time steps. We found a very weak mass-dependence of the computed potentials of mean force (PMFs). However, the errors are within the range of the simulation error, and the resulting PMFs are fairly comparable to the BOMD results. In addition, both ADMP and CURV methods are computationally more efficient than BOMD, by a factor of 2~4. Additional speed-up is achieved by the MPI parallelization. The overall speed-up after the parallelization is about 11~28 times relative to the speed of the single core BOMD calculations. This large speed-up is possible because the ADMP and CURV methods only involve simple matrix multiplications in the propagation of the electron density, which are better suited for efficient parallelization.

We also show that a much larger integration time step than that normally used in the ADMP and CURV MD methods can be accessed by applying the multiple time scale approach. This approach is not only as accurate as the single time scale approach, but also scales efficiently with increasing number of CPUs, up to 64 CPUs for the large system and up to 16~32 CPUs for the small to medium sized systems. Since the ADMP and CURV methods presented in this work are implemented mainly to speed up the semiempirical QM and QM/MM molecular dynamics simulations using parallelized computer architecture, they have a high potential for the study of large biological systems. The hybrid QM/MM simulations can benefit from the combined use of the present ADMP and CURV methods and the multiple time scale approach.

■ ASSOCIATED CONTENT

■ Supporting Information

Figures showing the effects of the fictitious mass on the extended Lagrangian molecular dynamics simulations and the effects of the SCF convergence criteria for BOMD, the comparison of the density purification scheme for CURV MD and the density change during MD, and the Mulliken charge distributions, table showing computation time of the multiple time scale approach. This material is available free of charge via the Internet at <http://pubs.acs.org>.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: kwangho.nam@chem.umu.se.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the grant from the Umeå University, Umeå, Sweden. We acknowledge Dr. Victor Ovchinnikov and Professor Dan T. Major for many helpful comments on the manuscript and Professor John Herbert for helpful suggestions about the curvy-steps method. Computational resources were provided by the National Energy Research Scientific Computing Center (NERSC) and the High Performance Computing Center North (HPC2N) at Sweden.

■ REFERENCES

- (1) Friesner, R. A. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6648–6653.
- (2) Hu, H.; Yang, W. *Annu. Rev. Phys. Chem.* **2008**, *59*, 573–601.
- (3) Kirchner, B.; Wennmohs, F.; Ye, S.; Neese, F. *Curr. Opin. Chem. Biol.* **2007**, *11*, 134–141.
- (4) Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4899–4907.
- (5) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902–3909.
- (6) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 209–220.
- (7) Thiel, W.; Voityuk, A. A. *Theor. Chim. Acta* **1992**, *81*, 391–404.
- (8) Voityuk, A. A.; Rösch, N. *J. Phys. Chem. A* **2000**, *104*, 4089–4094.
- (9) Nam, K.; Cui, Q.; Gao, J.; York, D. M. *J. Chem. Theory Comput.* **2007**, *3*, 486–504.
- (10) Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, Th.; Suhai, S.; Seifert, G. *Phys. Rev. B* **1998**, *58*, 7260–7268.
- (11) Stewart, J. J. P. *Rev. Comput. Chem.* **1990**, *1*, 45–81.
- (12) Frauenheim, Th.; Seifert, G.; Elstner, M.; Hajnal, Z.; Jungnickel, G.; Porezag, D.; Suhai, S.; Scholz, R. *Phys. Status Solidi B* **2000**, *217*, 41–62.
- (13) Bredow, T.; Jug, K. *Theor. Chem. Acc.* **2005**, *113*, 1–14.
- (14) Thiel, W. In *Theory and Applications of Computational Chemistry: The First Forty Years*; Dykstra, C. E., Frenking, G., Kim, K. S., Scuseria, G. E., Eds.; Elsevier B. V.: Amsterdam, 2005; pp 559–580.
- (15) Gao, J.; Ma, S.; Major, D. T.; Nam, K.; Pu, J.; Truhlar, D. G. *Chem. Rev.* **2006**, *106*, 3188–3209.
- (16) Rossi, I.; Truhlar, D. G. *Chem. Phys. Lett.* **1995**, *233*, 231–236.
- (17) Repasky, M. P.; Chandrasekhar, J.; Jorgensen, W. L. *J. Comput. Chem.* **2002**, *23*, 1601–1622.
- (18) Yang, Y.; Yu, H.; York, D.; Elstner, M.; Cui, Q. *J. Chem. Theory Comput.* **2008**, *4*, 2067–2084.
- (19) Giese, T. J.; York, D. M. *J. Chem. Phys.* **2005**, *123*, 164108.
- (20) Major, D. T.; Gao, J. *J. Am. Chem. Soc.* **2006**, *128*, 16345–16357.
- (21) Warshel, A.; Levitt, M. *J. Mol. Biol.* **1976**, *103*, 227–249.
- (22) Field, M. J.; Bash, P. A.; Karplus, M. *J. Comput. Chem.* **1990**, *11*, 700–733.
- (23) Thiel, W. *THEOCHEM* **1997**, 398–399, 1–6.
- (24) Cui, Q.; Elstner, M.; Kaxiras, E.; Frauenheim, T.; Karplus, M. *J. Phys. Chem. B* **2001**, *105*, 569–585.
- (25) Warshel, A. *Ann. Rev. Biophys. Biomol. Struct.* **2003**, *32*, 425–443.
- (26) Wu, X.; Koslowski, A.; Thiel, W. *J. Chem. Theory Comput.* **2012**, *8*, 2272–2281.
- (27) Maia, J. D. C.; Carvalho, G. A. U.; Manguiera, C. P., Jr.; Santana, S. R.; Cabral, L. A. F.; Rocha, G. B. *J. Chem. Theory Comput.* **2012**, *8*, 3072–3081.
- (28) Thiel, W. *MNDO99, CVS Development Version*; Max-Planck-Institut für Kohlenforschung: Mülheim an der Ruhr, 2012.
- (29) Stewart, J. J. P. *MOPAC2009*; Stewart Computational Chemistry: Colorado Springs, 2012.
- (30) Schlegel, H. B.; Millam, J. M.; Iyengar, S. S.; Voth, G. A.; Daniels, A. D.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **2001**, *114*, 9758–9763.

- (31) Iyengar, S. S.; Schlegel, H. B.; Millam, J. M.; Voth, G. A.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **2001**, *115*, 10291–10302.
- (32) Herbert, J. M.; Head-Gordon, M. *J. Chem. Phys.* **2004**, *121*, 11542–11556.
- (33) Car, R.; Parrinello, M. *Phys. Rev. Lett.* **1985**, *55*, 2471–2474.
- (34) Hutter, J.; Parrinello, M.; Vogel, S. *J. Chem. Phys.* **1994**, *101*, 3862–3865.
- (35) Brooks, B. R.; Brooks, C. L., III; MacKerell, A. D., Jr.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoseck, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. *J. Comput. Chem.* **2009**, *30*, 1545–1614.
- (36) Roothaan, C. C. J. *Rev. Mod. Phys.* **1951**, *23*, 69–89.
- (37) Pulay, P.; Fogarasi, G. *Chem. Phys. Lett.* **2004**, *386*, 272–278.
- (38) Herbert, J. M.; Head-Gordon, M. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3269–3275.
- (39) Niklasson, A. M. N.; Tymczak, C. J.; Challacombe, M. *J. Chem. Phys.* **2007**, *126*, 144103.
- (40) Payne, M. C.; Teter, M. P.; Allan, D. C.; Arias, T. A.; Joannopoulos, J. D. *Rev. Mod. Phys.* **1992**, *64*, 1045–1097.
- (41) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, U.K., 1987.
- (42) McWeeny, R. *Rev. Mod. Phys.* **1960**, *32*, 335–369.
- (43) Pulay, P. *Mol. Phys.* **1969**, *17*, 197–204.
- (44) Field, M. J. *Chem. Phys. Lett.* **1990**, *172*, 83–88.
- (45) Shao, Y.; Saravanan, C.; Head-Gordon, M.; White, C. A. *J. Chem. Phys.* **2003**, *118*, 6144–6151.
- (46) Tangney, P. *J. Chem. Phys.* **2006**, *124*, 044111.
- (47) Schlegel, H. B.; Iyengar, S. S.; Li, X.; Millam, J. M.; Voth, G. A.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **2002**, *117*, 8694–8704.
- (48) Nam, K.; Gao, J.; York, D. M. *J. Chem. Theory Comput.* **2005**, *1*, 2–13.
- (49) Walker, R. C.; Crowley, M. F.; Case, D. A. *J. Comput. Chem.* **2008**, *29*, 1019–1031.
- (50) Gibson, D. A.; Carter, E. A. *J. Phys. Chem.* **1993**, *97*, 13429–13434.
- (51) Li, X.; Tully, J. C.; Schlegel, H. B.; Frisch, M. J. *J. Chem. Phys.* **2005**, *123*, 084106.
- (52) Thiel, W. *MNDO97*, Version 5.0; University of Zurich: Zurich, Switzerland, 1998.
- (53) Li, X.-P.; Nunes, R. W.; Vanderbilt, D. *Phys. Rev. B* **1993**, *47*, 10891–10894.
- (54) Li, X.; Moss, C. L.; Liang, W.; Feng, Y. *J. Chem. Phys.* **2009**, *130*, 234115.
- (55) Neidigh, J. W.; Fesinmeyer, R. M.; Andersen, N. H. *Nat. Struct. Biol.* **2002**, *9*, 425–430.
- (56) Sunderland, A. G. In *Parallel Scientific Computing and Optimization*; Čiegis, R., Henty, D., Kågström, B., Žilinskas, J., Eds.; Springer: New York, 2009; pp 57–66.
- (57) Torrie, G. M.; Valleau, J. P. *J. Comput. Phys.* **1977**, *23*, 187–199.
- (58) Kumar, S.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; Rosenberg, J. M. *J. Comput. Chem.* **1992**, *13*, 1011–1021.
- (59) Rajamani, R.; Naidoo, K. J.; Gao, J. *J. Comput. Chem.* **2003**, *24*, 1775–1781.
- (60) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926–935.
- (61) Mulliken, R. S. *J. Chem. Phys.* **1955**, *23*, 1833–1840.
- (62) MacKerell, A. D., Jr.; Bashford, D.; Bellott, M.; Dunbrack, R. L., Jr.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E., III; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. *J. Phys. Chem. B* **1998**, *102*, 3586–3616.
- (63) Löwdin, P. O. *J. Chem. Phys.* **1950**, *18*, 365–375.
- (64) Reed, A. E.; Weinstock, R. B.; Weinhold, F. *J. Chem. Phys.* **1985**, *83*, 735–746.
- (65) Bader, R. F. W. *Acc. Chem. Res.* **1985**, *18*, 9–15.
- (66) Storer, J. W.; Giesen, D. J.; Cramer, C. J.; Truhlar, D. G. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 87–110.
- (67) Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. *J. Comput. Chem.* **2000**, *21*, 132–146.
- (68) Thompson, J. D.; Cramer, C. J.; Truhlar, D. G. *J. Comput. Chem.* **2003**, *24*, 1291–1304.
- (69) Francl, M. M.; Carey, C.; Chirlian, L. E.; Gange, D. M. *J. Comput. Chem.* **1996**, *17*, 367–383.
- (70) Lee, T.-S.; York, D. M.; Yang, W. *J. Chem. Phys.* **1996**, *105*, 2744–2750.
- (71) Scuseria, G. E. *J. Phys. Chem. A* **1999**, *103*, 4782–4790.
- (72) Pan, W.; Lee, T.-S.; Yang, W. *J. Comput. Chem.* **1998**, *19*, 1101–1109.
- (73) Goedecker, S.; Scuseria, G. E. *IEEE Comput. Sci. Eng.* **2003**, *5*, 14–21.
- (74) Xie, W.; Orozco, M.; Truhlar, D. G.; Gao, J. *J. Chem. Theory Comput.* **2009**, *5*, 459–467.