The Implementation of a Fast and Accurate QM/MM Potential Method in Amber

ROSS C. WALKER,* MICHAEL F. CROWLEY, DAVID A. CASE
Department of Molecular Biology, The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, California 92037

Received 22 May 2007; Revised 1 August 2007; Accepted 16 September 2007
DOI 10.1002/jcc.20857
Published online 10 December 2007 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Version 9 of the Amber simulation programs includes a new semi-empirical hybrid QM/MM functionality. This includes support for implicit solvent (generalized Born) and for periodic explicit solvent simulations using a newly developed QM/MM implementation of the particle mesh Ewald (PME) method. The code provides sufficiently accurate gradients to run constant energy QM/MM MD simulations for many nanoseconds. The link atom approach used for treating the QM/MM boundary shows improved performance, and the user interface has been rewritten to bring the format into line with classical MD simulations. Support is provided for the PM3, PDDG/PM3, PM3CARB1, AM1, MNDO, and PDDG/MNDO semi-empirical Hamiltonians as well as the self-consistent charge density functional tight binding (SCC-DFTB) method. Performance has been improved to the point where using QM/MM, for a QM system of 71 atoms within an explicitly solvated protein using periodic boundaries and PME requires less than twice the cpu time of the corresponding classical simulation.

 J Comput Chem 29: 1019–1031, 2008

Key words: molecular dynamics; Amber; QM/MM; quantum mechanics; MD; PME; particle mesh Ewald; Ewald; molecular mechanics

Introduction

The idea of combining quantum mechanical (QM) and molecular mechanical (MM) potentials into a hybrid QM/MM potential was first realized by Warshel and Levitt (1976) in their paper on the catalytic mechanism of lysozyme.[1] Typically in a QM/MM approach the parts of the protein and substrate that are directly involved in the enzyme reaction are calculated using a QM potential function, and the remaining atoms in the system use a classical MM potential. This avoids a major deficiency with classical MM calculations, where chemical bonds cannot be made or broken during a simulation. The coupling of a QM and MM potential allows just the reaction center to be studied quantum mechanically while keeping the calculation complexity low by using a more approximate MM potential elsewhere. This partitioning of the system allows calculations on systems significantly larger than would have been possible with a pure QM approach, and at the same time enables calculations such as reactions to be studied for which classical MM potentials are not appropriate. A number of different hybrid potentials have been developed with QM methods ranging from empirical valence bond approaches[2,3] to semi-empirical molecular orbital.[4,5] density functional[6] and *ab initio* methods. Such approaches have been used in the study of a wide range of problems including studying solvation effects on reactions in proteins and in solution, and of solvent-induced spectral shifts.[3,7,8]

Interest in hybrid QM/MM potentials has lead to a number of popular implementations, including CHARMM,[9] Dynamo,[10] BOSS,[11] and Amber,[12] that are designed for biomolecular simulations. A QM/MM potential for running MD was first introduced into the Amber software package by Singh and Kollman[13] in 1986, which used a modified version of Gaussian 80.[14,15] Amber v4.0 was later used as the basis for the program ROAR,[16] which provided semi-empirical QM/MM support via a modified version of MOPAC.[17] With version 8 of Amber, semi-empirical support was provided, on a functionally limited basis via an interface to the program DivCon Lite.[18]

Here we discuss the implementation of QM/MM in version 9 of Amber. This new implementation is tightly integrated within the regular *sander* MD program and provides seamless support for QM/MM calculations without the need for any extra configuration files, special executables or modified scripts. We give details of tests of energy and gradient accuracy, a new link atom approach,

This article contains supplementary material available via the Internet at http://www.interscience.wiley.com/jpages/0192-8651/suppmat

Correspondence to: D. A. Case; e-mail: case@scripps.edu

*Present address: San Diego Supercomputer Center, University of California, San Diego, 9500 Gilman Drive #0505, La Jolla, CA, 92093-0505, USA

Contract/grant sponsor: NIH; contract/grant number: GM57513

Contract/grant sponsor: ONR; contract/grant number: N00014-05-1-0457

how we have modified the classical Particle Mesh Ewald (PME)[19] approach for treating long range electrostatics to make it compatible with QM/MM calculations, and new functionality such as support for the generalized Born implicit solvation model.[20,21] We discuss the performance of the new code with comparison to other commonly used MD packages such as CHARMM and Dynamo.

Capabilities of the Code

Our primary motivation for rewriting the QM/MM code in Amber was to make its use as orthogonal as possible to other choices made in classical simulations, so that choosing to have a QM active site places as few restrictions as possible on what else can be done in the simulation. This implies that the accuracy of the energy and gradients (and hence energy conservation in NVE simulations) should be comparable with that achieved for empirical potentials, and commonly used options for MM simulations (such as periodic boundary conditions with PME,[19] or generalized Born implicit[22] solvent models) should be supported in the QM/MM code. Since biochemical simulations require extensive exploration of configuration space, we have limited the code (for now) to semiempirical and DFTB methods that are fast enough to be used for simulations of tens or hundreds of nanoseconds that are commonly used for MM simulations. The capabilities of the new QM/MM implementation include the following:

1. Support for minimization and molecular dynamics with either pure QM or QM/MM energy and gradient evaluation using either the PM3,[23,24] PDDG/PM3,[25] PM3CARB1,[26] AM1,[27] MNDO,[28,29] or PDDG/MNDO[25] semi-empirical Hamiltonians as well as support for the density functional theory-based-tight- binding (DFTB) Hamiltonian[30,31] and its self-consistent-charge version, SCC-DFTB.[32,33] Details of the DFTB implementation are discussed elsewhere.[34]
2. The ability to apply the standard SHAKE[35] algorithm of *sander* to QM as well as MM bonds involving hydrogen atoms. This allows stable QM/MM simulations, where bonds involving hydrogen are not expected to be broken, to be run using a 2 fs time step.
3. Support for gas phase, solvent cap and periodic (PME[36]) QM/MM simulations.
4. Support for all of the generalized Born solvation models[22] that are present in *sander*, using the Born radii developed for classical simulations and dynamically calculated Mulliken charges.
5. Application of restraints on the QM or MM regions using any of the methods currently implemented in *sander* allowing for targeted MD, Nudged Elastic Band, and umbrella sampling.
6. Support for QM/MM Replica Exchange simulations.
7. Support for Path Integral QM/MM MD simulations.
8. Support for QM/MM based thermodynamic integration calculations.
9. Simulation in parallel over either shared memory or distributed memory computer clusters using the message passing interface (MPI).

The Hybrid QM/MM Potential

The system is partitioned into two regions, a QM region consisting of the atoms defined by either the *qmmask* or *iqmatoms* keywords, and an MM region consisting of all remaining atoms. For a typical protein simulation in explicit solvent, the number of MM atoms is much greater than the number of QM atoms. Either region can contain zero atoms, giving either a pure QM simulation or a standard classical simulation. The effective Hamiltonian, $\hat{H}_{\text{eff}}$, operates on the system's wavefunction Ψ, and is dependent on the electron positions, $\bar{r}_e$, the positions of the MM nuclei, $\bar{r}_m$, and the QM nuclei, $\bar{r}_q$:

$$\hat{H}_{\text{eff}}\Psi(\bar{r}_e,\bar{r}_q,\bar{r}_m) = E_{\text{eff}}(\bar{r}_e,\bar{r}_q,\bar{r}_m)\Psi(\bar{r}_e,\bar{r}_q,\bar{r}_m) \quad (1)$$

The effective Hamiltonian consists of three components—one for the QM region, one for the MM region and a term that describes the interaction of the QM and MM:

$$\begin{aligned}\hat{H}_{\text{eff}} &= \hat{H}_{\text{QM}} + \hat{H}_{\text{MM}} + \hat{H}_{\text{QM/MM}}\\ E_{\text{eff}} &= E_{\text{QM}} + E_{\text{MM}} + E_{\text{QM/MM}}\end{aligned} \quad (2)$$

The MM term can be removed from the integral since it is independent of the distribution of the electrons:

$$E_{\text{eff}} = \langle\Psi|\hat{H}_{\text{QM}} + \hat{H}_{\text{QM/MM}}|\Psi\rangle + E_{\text{MM}} \quad (3)$$

The interaction term $\hat{H}_{\text{QM/MM}}$ represents the interaction of the MM point charges with the electron cloud of the QM atoms and the interaction between the MM point charges and the QM atomic cores; here a "core" represents the nuclear charge plus any core electrons that are not treated explicitly. For the case where there are no covalent bonds between the QM and MM regions, this term is the sum of an electrostatic term and a Lennard–Jones (VDW) term:

$$\hat{H}_{\text{QM/MM}} = -\sum_e\sum_m q_m\hat{h}_{\text{electron}}(\bar{r}_e,\bar{r}_m) + \sum_q\sum_m z_q q_m\hat{h}_{\text{core}}(\bar{r}_q,\bar{r}_m) + \sum_m\sum_q\left(\frac{A_{mq}}{r_{mq}^{12}} - \frac{B_{mq}}{r_{mq}^{6}}\right) \quad (4)$$

where e refers to electrons, m to MM atoms and q to the QM cores (nuclei and core electrons). q_m and z_q are the charges on MM atom m and core of the QM atom q, $\bar{r}$ is the coordinate vector, r_{mq} the distance between atoms m and q and A and B are Lennard–Jones interaction parameters. There has been discussion concerning whether or not the Lennard–Jones (LJ) coefficients should be explicitly parameterized for QM-MM interactions.[37,38] Amber allows the user to specify which parameters will be used, but does not yet have a default set of modified LJ parameters. The operators $\hat{h}$ specify the interaction between the QM electrons and MM point charges. For semiempirical Hamiltonians we follow the MOPAC programs, and use the full electrostatic interactions between the QM charge density (expanded in a STO-6G minimal basis set) and the point charges on the MM atoms. For DFTB the QM/MM interactions are represented by a point-charge Coulomb interaction between the Mulliken charges and MM partial charges.

Once an effective Hamiltonian for the system has been defined the wavefunction and energy can be evaluated by minimizing E_{eff} in eq. (3) with respect to the molecular orbital coefficients of the ground state wavefunction using a self-consistent (SCF) procedure to solve a restricted Hartree–Fock approximation of the electronic wave function of the system at each step.

QM/MM Long Range Electrostatics

Short-range electrostatics (for atoms within a cutoff) are straightforward to implement using eq. (4). However, it is well understood that long range electrostatic interactions and solvent effects play an important role in the behavior of solvated systems. Long range interactions of periodic classical systems in *sander* are traditionally included via the application of a Particle Mesh Ewald (PME) method.[19] This is an adaptation of the *regular* Ewald sum method[39] for calculating the full electrostatic energy of a unit cell in a macroscopic lattice of repeating images. The PME method is fast since the reciprocal space Ewald sums are B-spline interpolated onto a grid and the convolutions necessary to calculate the sums are evaluated in Fourier space using fast Fourier transforms.

Recently Nam et al. have described an implementation of the Ewald summation method[40] compatible with semi-empirical QM methods, and Laino et al. have published a similar implementation for DFTB based on the multigrid approach.[41] This Ewald QM/MM method provides a correct implementation of long-range electrostatics in periodic systems, but it is very slow for any sizeable simulation system. Here we describe how the faster PME alternative can be adapted for use with the QM/MM model.

If we denote the electrostatic energy of a periodic QM/MM system as E^{Periodic} then

$$E^{\text{Periodic}} = E^{\text{Periodic}}[\rho,\rho] + E^{\text{Periodic}}[\rho,q] + E^{\text{Periodic}}[q,q] \quad (5)$$

where q represents the partial charges of the MM atoms, ρ represents the electron density and core charges of the quantum region and the $[\rho,q]$ notation represents the interaction of ρ with q.

To use existing nonperiodic treatments of QM/MM systems, we follow Nam et al. to define the complete periodic energy as a sum of the nonperiodic energy (as determined by conventional cutoff techniques), E^{RS}, plus a periodic boundary correction term, ΔE^{PBC},

$$E^{\text{Periodic}} = E^{\text{RS}} + \Delta E^{\text{PBC}} \quad (6)$$

Each of the terms of eq. (5) can be redefined in this way. The key approximation is that the full charge density, ρ, in $\Delta E^{\text{PBC}}[\rho,\rho]$ and $\Delta E^{\text{PBC}}[\rho,q]$, can be approximated by interactions with Mulliken charges, denoted Q, which are used as surrogates for the electron densities of the wave functions on the QM atoms,

$$\begin{aligned}\Delta E^{\text{PBC}}[\rho,\rho] &= E^{\text{Periodic}}[\rho,\rho] - E^{\text{RS}}[\rho,\rho] \\ &\approx E^{\text{Periodic}}[Q,Q] - E^{\text{RS}}[Q,Q] = \Delta E^{\text{PBC}}[Q,Q]\end{aligned} \quad (7)$$

where,

$$E^{\text{Periodic}}[Q,Q] = \frac{1}{2}\sum_{j,i}^{\text{QM}}\sum_{n=0}^{\infty}{}' \frac{Q_iQ_j}{|\bar{r}_{ij}+n|} \quad (8)$$

in which the first sum is over all atom pairs in the unit cell and the second sum is over all unit cell translations. The prime in the second sum is to indicate that the $i = j$ term is not included when $n = 0$. The Ewald method splits the infinite sum into three convergent sums:

$$\begin{aligned}E^{\text{Periodic}}[Q,Q] = \frac{1}{2\pi V}\sum_{i}^{\text{QM}}\sum_{j}^{\text{QM}}\sum_{\bar{k}\neq 0} Q_iQ_j\left(\frac{4\pi^2}{k^2}\right) \\ \times \exp\left(-\frac{k^2}{4\kappa^2}\right)\cos\left(\bar{k}\cdot\bar{r}_{ij}\right) \\ + \frac{1}{2}\sum_{i}^{\text{QM}}\sum_{j\neq i}^{\text{QM}} Q_iQ_j\frac{\text{erfc}\left(\kappa|\bar{r}_{ij}|\right)}{|\bar{r}_{ij}|} + \sum_{i}^{\text{QM}} Q_i^2\frac{\kappa}{\sqrt{\pi}}\end{aligned} \quad (9)$$

where V is the volume of the unit cell, $\bar{k}$ are the reciprocal unit cell vectors, erfc is the complimentary error function, and κ is a parameter, which is chosen such that the erfc function goes to zero within a preferred distance. The three terms are called the reciprocal sum, the direct sum, and the self term, $E^{\text{recip}}[Q,Q]$, $E^{\text{direct}}[Q,Q]$, and $E^{\text{self}}[Q,Q]$, respectively,

$$\begin{aligned}\Delta E^{\text{PBC}}[Q,Q] &= E^{\text{recip}}[Q,Q] + E^{\text{direct}}[Q,Q] + E^{\text{self}}[Q,Q] \\ &\quad - E^{\text{RS}}[Q,Q] \\ &= E^{\text{recip}}[Q,Q] + E^{\text{self}}[Q,Q] \\ &\quad + \frac{1}{2}\sum_{i}^{\text{QM}}\sum_{j\neq i}^{\text{QM}} Q_iQ_j\frac{\text{erfc}(\kappa r_{ij})}{r_{ij}} - \frac{1}{2}\sum_{i}^{\text{QM}}\sum_{j\neq i}^{\text{QM}} Q_iQ_j\frac{1}{r_{ij}} \\ &= E^{\text{recip}}[Q,Q] + E^{\text{self}}[Q,Q] \\ &\quad - \frac{1}{2}\sum_{i}^{\text{QM}}\sum_{j\neq i}^{\text{QM}} Q_iQ_j\frac{\text{erf}\left(\kappa r_{ij}\right)}{r_{ij}} \\ &= E^{\text{recip}}[Q,Q] + E^{\text{self}}[Q,Q] + \Delta E^{\text{PBC}}_{\text{cutoff}}[Q,Q]\end{aligned} \quad (10)$$

where erf is the error function.

An equivalent treatment is also used for $\Delta E^{\text{PBC}}[Q,q]$; since an atom cannot be both a QM and a MM atom there is no self term; this simplifies to,

$$\Delta E^{\text{PBC}}[Q,q] = E^{\text{recip}}[Q,q] + \sum_{i}^{\text{QM}} Q_i\sum_{j}^{\text{MM}} q_j\frac{\text{erf}\left(\kappa r_{ij}\right)}{r_{ij}} \quad (11)$$

Introducing the ΔE^{PBC} Mulliken charge approximation allows us to write the periodic energy from eq. (5) as,

$$\begin{aligned}E^{\text{Periodic}} &\approx E^{\text{RS}}[\rho,\rho] + \Delta E^{\text{PBC}}[Q,Q] \\ &\quad + E^{\text{RS}}[\rho,q] + \Delta E^{\text{PBC}}[Q,q] \\ &\quad + E^{\text{recip}}[q,q] + E^{\text{direct}}[q,q] + E^{\text{self}}[q,q]\end{aligned} \quad (12)$$

The first three terms are treated in the same way as that described by Nam et al.[40] in which the second term is evaluated using a regular Ewald method. The fourth and fifth terms are the

same as Nam et al.[40] except that the reciprocal-space K-sum is replaced with the PME method which closely parallels the regular Ewald method but shows significantly improved performance for macromolecular systems. In particular the fourth term has the form,

$$\begin{aligned}\Delta E^{\mathrm{PBC}}[Q,q] &= E^{\mathrm{Periodic}}[Q,q] - E^{\mathrm{RS}}[Q,q] \\ &= \sum_{i}^{\mathrm{QM}} Q_i \sum_{j}^{\mathrm{MM}} q_j \left[\Psi^{\mathrm{Periodic}}(\bar{r}_i,\bar{r}_j) - \Psi^{\mathrm{RS}}\left(|\bar{r}_{ij}|\right)\right] \\ &= \sum_{i}^{\mathrm{QM}} Q_i \sum_{j}^{\mathrm{MM}} q_j \left[\Delta\Psi^{\mathrm{PBC}}(\bar{r}_i,\bar{r}_j)\right] \end{aligned} \tag{13}$$

where Ψ^{Periodic} is the full pair potential for all pairs including periodic images to infinity, and Ψ^{RS} the full Culomb pair potential within a cutoff. Applying the Ewald method yields the following correction potential,

$$\begin{aligned}\Delta\Psi^{\mathrm{PBC}}(R_{i,j}) &= \Psi^{\mathrm{recip}}(\bar{r}_i,\bar{r}_j) + \Psi^{\mathrm{direct}}\left(|\bar{r}_{ij}|\right) - \Psi^{\mathrm{RS}}\left(|\bar{r}_{ij}|\right) \\ &= \Psi^{\mathrm{recip}}(\bar{r}_i,\bar{r}_j) + \Delta\Psi^{\mathrm{PBC}}_{\mathrm{cutoff}}\left(|\bar{r}_{ij}|\right) \end{aligned} \tag{14}$$

where $\Psi^{\mathrm{recip}}(\bar{r}_i,\bar{r}_j)$ is the reciprocal part of the Ewald pair potential due to all MM atoms interacting with QM atoms represented as Mulliken point charges, and $\Delta\Psi^{\mathrm{PBC}}_{\mathrm{cutoff}}(|\bar{r}_{ij}|)$ is the correction to the cutoff based real-space potential and has the form, $\mathrm{erf}\left(\kappa|\bar{r}_{ij}|\right)/|\bar{r}_{ij}|$ by analogy to the treatment in eq. (10). The reciprocal term of eq. (14) is conveniently calculated with a regular Ewald reciprocal sum but, since there are a large number of MM atoms and the sum scales as the number of MM atoms times the number of QM atoms, the sum becomes computationally expensive. Writing the Ewald sum in the following way with $E^{\mathrm{recip}}_{\mathrm{PME}}(q^*)$ representing the total reciprocal sum energy of all atoms calculated with PME and all atoms represented by point charges, where q^* represents a combination of the static MM point charges and the QM Mulliken charges, we can isolate the term in question:

$$\begin{aligned}E^{\mathrm{recip}}_{\mathrm{PME}}(q^*) &= \frac{1}{2}\sum_{i=1}^{\mathrm{allatoms}} q_i^* \sum_{j=1}^{\mathrm{allatoms}} q_j^* \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) \\ &= \frac{1}{2}\sum_{i=1}^{\mathrm{QM}} Q_i \sum_{j=1}^{\mathrm{QM}} Q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) + \sum_{i=1}^{\mathrm{QM}} Q_i \sum_{j=1}^{\mathrm{MM}} q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) \\ &\quad + \frac{1}{2}\sum_{i=1}^{\mathrm{MM}} q_i \sum_{j=1}^{\mathrm{MM}} q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) \end{aligned} \tag{15}$$

and rearranging for the terms we need,

$$\begin{aligned}&\sum_{i=1}^{\mathrm{QM}} Q_i \sum_{j=1}^{\mathrm{MM}} q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) + \frac{1}{2}\sum_{i=1}^{\mathrm{MM}} q_i \sum_{j=1}^{\mathrm{MM}} q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) \\ &\qquad = E^{\mathrm{recip}}_{\mathrm{PME}}(q^*) - \frac{1}{2}\sum_{i=1}^{\mathrm{QM}} Q_i \sum_{j=1}^{\mathrm{QM}} Q_j \left(\Psi^{\mathrm{recip}}_{\mathrm{PME}}(\bar{r}_i,\bar{r}_j)\right) \end{aligned} \tag{16}$$

Using the relationship in (16), the energies and forces from the reciprocal sums on the left can be determined from the difference of the results of a PME reciprocal sum of all the charges and a PME reciprocal sum of just the Mulliken charges. Thus we can rewrite eq. (12) as,

$$\begin{aligned}E^{\mathrm{Periodic}} \approx\ & E^{\mathrm{RS}}[\rho,\rho] + E^{\mathrm{RS}}[\rho,q] + E^{\mathrm{recip}}_{\mathrm{Ewald}}[Q,Q] + \Delta E^{\mathrm{PBC}}_{\mathrm{cutoff}}[Q,Q] \\ &+ \Delta E^{\mathrm{PBC}}_{\mathrm{cutoff}}[Q,q] + \left(E^{\mathrm{recip}}_{\mathrm{PME}}[q^*] - E^{\mathrm{recip}}_{\mathrm{PME}}[Q,Q]\right) + E^{\mathrm{direct}}_{\mathrm{PME}}[q,q] \end{aligned} \tag{17}$$

The calculation is organized so that the first four terms of eq. (17) come from the QM part of the calculation where the first term is the direct space QM-QM interaction, the second term the interaction of QM atoms with MM atoms that are within the cutoff distance of any QM atom, and the third and fourth terms are the interaction of direct space QM atoms, represented by a Mulliken charge approximation, with the periodic QM images. Since the number of QM atoms is generally small compared with the number of MM atoms, the most efficient method for the evaluation of terms three and four is a regular Ewald approach. The fifth and sixth terms are calculated by the two PME calculations described earlier after the Mulliken charges have been determined and the final term is the classical direct space calculation over static MM point charges. Figure 1 shows a flow chart of the implementation of this method.

Fock Matrix Corrections

The wave function is determined by a self-consistent field method to minimize the total energy as a set of molecular orbital coefficients. This minimization procedure requires the diagonalization of a Fock matrix.

$$F_{\mu\nu} = \frac{\partial E[\rho]}{\partial \rho_{\mu\nu}} \tag{18}$$

is an element of the Fock matrix, and $\rho_{\mu\nu}$ is an element of the single particle density matrix.

The Fock matrix can be split into "real space" and "correction" terms in the same way as the treatment of the potential energy above,

$$F^{\mathrm{Periodic}}_{\mu\nu} = F^{\mathrm{RS}}_{\mu\nu} + \Delta F^{\mathrm{PBC}}_{\mu\nu} \tag{19}$$

The real-space part of the Fock matrix is constructed as in the nonperiodic case in which the MM atoms within the cutoff contribute as static partial charges. The periodic-boundary correction has the form,

$$\Delta F^{\mathrm{PBC}}_{\mu\nu} = \frac{\partial}{\partial \rho_{\mu\nu}}\left\{\Delta E^{\mathrm{PBC}}[Q,Q] + \Delta E^{\mathrm{PBC}}[Q,q]\right\} \tag{20}$$

Both derivatives will be nonzero only when $\mu = \nu$ because the Mulliken charges, Q, depend only on $\rho_{\mu\mu}$, and therefore will only add corrections to the diagonal elements of the Fock matrix.

Since the definition of Mulliken charge in NDDO-based semiempirical methods is,

$$Q_i = Z_i - \sum_{\mu\in i} \rho_{\mu\mu} \tag{21}$$

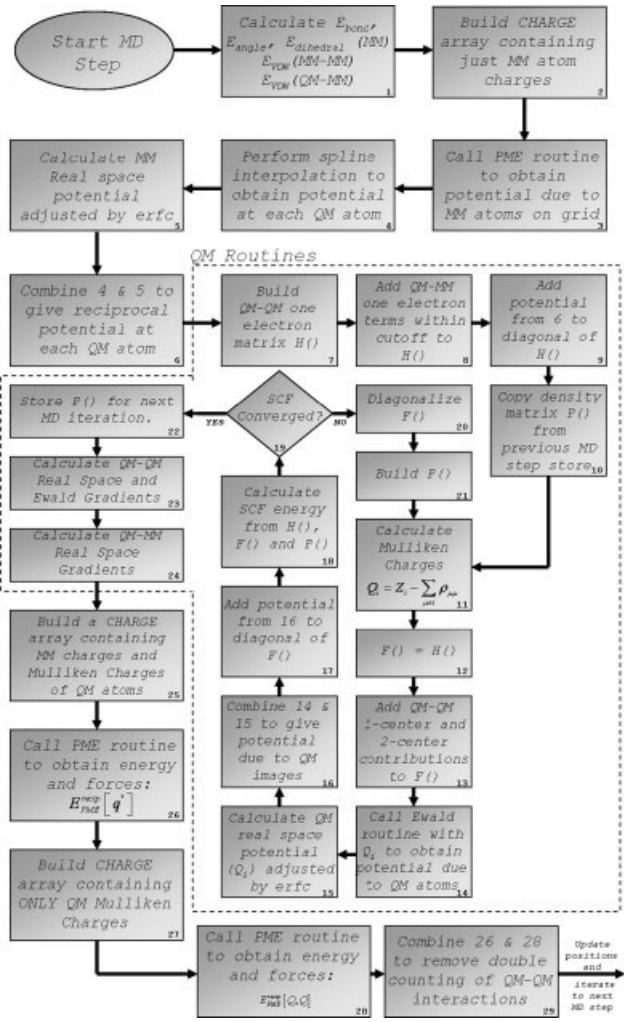

Figure 1. Flowchart illustrating the steps involved in the QM/MM PME calculation process for a single molecular dynamics step.

the derivative of the last line of eq. (13) with respect to $\rho_{\mu\mu}$ is,

$$\frac{\partial \Delta E^{\mathrm{PBC}}[Q,q]}{\partial \rho_{\mu\mu}} = -\sum_{j}^{\mathrm{MM}} q_j \left(\Delta \Psi^{\mathrm{PBC}}\left(\left|\bar{r}_{ij}\right|\right)\right) \tag{22}$$

which is simply the electrostatic potential at the position of QM atom i due to all MM charges and their images that are not within the direct space cutoff. Equation (22) is not a function of the Mulliken charges so the $\Delta F^{\mathrm{PBC}}[Q,q]$ correction term can be calculated once before the SCF procedure and simply added to the one electron matrix. In practice the reciprocal part of this term is calculated using a PME method where the charges on the quantum atoms are zeroed, while the cutoff correction term is calculated by $\sum_j^{\mathrm{MM}} q_j(\mathrm{erf}(\kappa|\bar{r}_{ij}|)/|\bar{r}_{ij}|)$ derived from the second term of eq. (11). In the case of the ΔF^{PBC} $[Q,Q]$ correction term, the dependence on Q does not vanish when taking the derivative with respect to $\rho_{\mu\mu}$, so this correction term must be evaluated on every step of the SCF procedure. If there are relatively few quantum atoms compared with the number of MM atoms, a regular Ewald method is the most efficient for this part of the calculation. Here we use an Ewald procedure that calculates the electrostatic potential at each QM atom position due to all the QM images. This consists of a reciprocal calculation and a cutoff correction term corresponding to terms two and three of the third line of eq. (10).

Once the SCF procedure converges to within the chosen criteria the energy is calculated as described earlier and the forces are simply the derivative of the energy with respect to position. We can write the force along a coordinate on a single atom as,

$$f_{x_i} = \left(\frac{dE}{dx_i}\right) \tag{23}$$

where E depends on the atom positions and has the added dependence on the Mulliken charges for the periodic corrections. Since Mulliken charges are computed from diagonal elements of the Fock matrix, they depend only on the orbital coefficients. The total energy can then be considered as a function of the atomic positions, the orbital coefficients, and the Lagrange multipliers used to maintain the orthonormality constraints.[42,43] Because the SCF equations minimize the energy as a function of the orbital coefficients, we can compute the gradient without needing to know how the orbital coefficients (or the Mulliken charges) change with geometry. Gradients are calculated in the usual way for a closed-shell Hartree–Fock wavefunction (details are in section 30.5 of ref. 42 and eq. (22) of ref. 43), with the simplification that the overlap matrix is independent of geometry for these NDO models.

QM/MM Generalized Born

Support for QM/MM implicit solvent simulations is provided by a generalized-born surface-area (GBSA) approach. Our implementation is based on the QM/MM compatible GBSA approach developed by Pellegrini and Field.[44] The QM energy, E_{QM}, of an implicitly solvated closed shell molecule being treated with a semiempirical QM method of a type supported by Amber is given by

$$E_{\mathrm{QM}} = \frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}\left(H_{\mu\nu} + F_{\mu\nu}\right) + E_{\mathrm{nuc}} + G_{\mathrm{pol}} + G_{\mathrm{SA}} \tag{24}$$

where $P_{\mu\nu}$, $H_{\mu\nu}$, and $F_{\mu\nu}$ are the density, one-electron and Fock matrices, respectively. μ and ν refer to the basis functions used to expand the molecular orbitals, E_{nuc} is the nuclear repulsion energy between QM nuclei and G_{pol} and G_{SA} are the polarization and surface area energies[21]

$$G_{\mathrm{SA}} = \sum_{i=1}^{N} \sigma_i A_i \tag{25a}$$

$$G_{\mathrm{pol}} = -\frac{1}{2}\left(\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_0}\right)\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{f_{ij}(r_{ij})} \tag{25b}$$

where N is the total number of atoms, A_i is the solvent accessible surface of atom i, σ_i is an empirical solvation parameter, ε_i

and ε_0 are the solute and solvent dielectric constants respectively, q_i is the charge on atom i, and f_{ij} is a function dependent on the distance, r_{ij}, between atoms i and j. f_{ij} does not have a unique functional form. In our implementation we use the form proposed by Still et al.[21]

$$f_{ij}(r_{ij}) = \sqrt{r_{ij}^2 + \alpha_{ij}^2 \exp\left(-\frac{r_{ij}^2}{4\alpha_i\alpha_j}\right)} \tag{26}$$

where α_i is the effective Born radius of atom i. The calculation of accurate Born radii is crucial to the effectiveness of the GBSA method. Amber contains a number of Born radii parameter sets that have been developed for classical simulations. All of these approaches are supported in our QM/MM implementation. While some have noted that QM GBSA results can be improved by specifically optimizing the radii for each QM Hamiltonian[44] we do not provide any QM optimized radii sets with Amber, however since these are defined during the construction of the input file containing the molecule's topology it is a simple matter for the user to substitute the QM atom radii with radii of their choice.

Utilizing eq. (25) for a QM calculation requires the determination of effective atomic charges to be used in place of q_i and q_j which for MM calculations would typically be fixed partial charges consistent with the force field being used. The simplest approach is to use a Mulliken population analysis in the same way as was employed for the PME calculations described above eq. (21). Since for an accurate description of the solvation effect on the electron density of the QM region and to be able to differentiate eq. (25b) with respect to just r_{ij} it is necessary to modify the vacuum Fock matrix in a fashion equivalent to that described earlier. This is achieved by adding terms of the following form to the diagonal elements of the vacuum Fock matrix

$$\Delta F_{\mu\mu} = \left(\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_0}\right) \sum_{\mathrm{j\in QM}}^{\mathrm{N}} \frac{Q_j}{f_{ij}} \quad \mu \in i \tag{27}$$

Minimization of E_{eff} in eq. (3) with the additional term in eq. (27) proceeds in exactly the same way as the vacuum case using an iterative SCF approach. This yields a density matrix that is self consistent with having been perturbed by the GB solvent potential. The GB contributions to the derivatives can then be calculated in exactly the same way as would be done for a classical simulation using the Mulliken charges from the self-consistent density matrix in place of classical partial charges for the QM atoms.

Extending the earlier description to a QM/MM system is conceptually simple adding two extra terms to eq. (2) to give

$$E_{\mathrm{eff}} = E_{\mathrm{QM}} + E_{\mathrm{MM}} + E_{\mathrm{QM/MM}} + G_{\mathrm{pol}} + G_{\mathrm{SA}} \tag{28}$$

Since the surface area term G_{SA} is independent of whether the atom is a QM or MM atom so it can be evaluated independently of the rest of the calculation. The polarization term is a function of the QM charges but is pairwise additive and so can be rewritten as

$$G_{\mathrm{pol}} = -\frac{1}{2}\left(\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_0}\right)\left(\sum_{i\in\mathrm{QM}}^{N}\sum_{j\in\mathrm{QM}}^{N}\frac{Q_iQ_j}{f_{ij}} + \sum_{i\in\mathrm{QM}}^{N}\sum_{j\in\mathrm{MM}}^{N}\frac{Q_iq_j}{f_{ij}} + \sum_{i\in\mathrm{MM}}^{N}\sum_{j\in\mathrm{MM}}^{N}\frac{q_iq_j}{f_{ij}}\right) \tag{29}$$

The second term in brackets on the right hand side of the earlier equation contains the interaction between QM atoms described by Mulliken charges and MM atoms described by classical partial charges. This expression means that the expression for modifying the Fock matrix terms to account for the polarization energy, eq. (28), must be extended to include the sum over all atoms, both QM and MM. Since the MM charges remain fixed during the SCF procedure so the contributions to the Fock matrix due to the MM atoms can be evaluated once before entering the SCF procedure while the contributions due to the QM atoms must be reevaluated on each step of the SCF.

The QM/MM Boundary

The way in which nonbonded interactions between the QM and MM parts of a system are handled in *sander* v9's hybrid QM/MM potential was described in the previous section. This, however, only deals with situations where there are no covalent bonds between the QM and MM regions. In many simulations it is necessary to have the QM/MM boundary cut covalent bonds. In this situation a number of additional approximations have to be made. Many methods have been proposed for dealing with this problem, that generally fall into three classes. These classes include capping potential, or pseudo bond methods[45] which use an element of fictitious type to "cap" each bond between the QM and MM regions; hybrid-orbital approaches, which employ either hybrid or localised frozen orbitals on the QM atom of the QM-MM covalent pair[46,47] and the link atom approach.

In *sander* v9, we use the link atom approach. First introduced by Singh and Kollman[13] this method has found widespread use in QM/MM calculations with a number of variations being developed, including those by Bersuker et al.[48] and Morokuma and Maseras.[49] In this approach a link atom, which is typically, but not always, a hydrogen, is placed along the bond between the QM and MM region at a suitable distance ($\sim$1 Å) to satisfy valence requirements. The link atom is included in the QM part of the calculation as a regular QM atom. It shares the same pair list for QM/MM interactions as real QM atoms. Such an approach does nothing to maintain the bond between the QM and MM regions and so this must be dealt with classically in the MM part of the calculation.

There are a number of ways to implement a link atom approach that deal with both the way the link atom is positioned, the way the forces on the link atom are propagated, and the way nonbonding interactions around the link atom are treated. We have implemented a link atom approach that is similar to that used by Dynamo[10] where the link atom is treated as part of the

covalent bond between the QM and MM atoms bonded across the interface. Each time an energy or gradient calculation is to be done, the link atom coordinates are automatically generated from the coordinates of the atoms making up the QM-MM covalent pair. The link atom is placed along the bond vector joining the QM and MM atoms using the formula:

$$\bar{r}_{\mathrm{L}} = \bar{r}_{\mathrm{QM}} + d_{\mathrm{L-QM}} \frac{\bar{r}_{\mathrm{MM}} - \bar{r}_{\mathrm{QM}}}{|\bar{r}_{\mathrm{MM}} - \bar{r}_{\mathrm{QM}}|} \tag{30}$$

where $\bar{r}_{\mathrm{L}}$, $\bar{r}_{\mathrm{QM}}$, and $\bar{r}_{\mathrm{MM}}$ are the position vectors of the link atom, QM atom and MM atom respectively and $d_{\mathrm{L-QM}}$ is a user defined constant specifying the QM to link atom bond length. The default link atom is a hydrogen atom but this can be changed by the user.

Once the QM gradient has been calculated the force on each link atom is redistributed between the QM and MM link pair by application of eq. (31) and the corresponding y and z component forms.

$$\begin{aligned} F'_{x\,\mathrm{QM}} &= -\frac{\partial E(\bar{r}_L)}{\partial x_{\mathrm{QM}}} = -\nabla_L E \cdot \frac{\partial \bar{r}_{\mathrm{L}}}{\partial x_{\mathrm{QM}}} = \bar{F}_L \cdot \frac{\partial \bar{r}_{\mathrm{L}}}{\partial x_{\mathrm{QM}}} \\ F'_{x_{\mathrm{MM}}} &= \bar{F}_L \cdot \frac{\partial \bar{r}_{\mathrm{L}}}{\partial x_{\mathrm{MM}}} \end{aligned} \tag{31}$$

where $F'_{x_{\mathrm{QM}}}$ and $F'_{x_{\mathrm{MM}}}$ are the x components of the force on the QM and MM atom respectively due to the link atom, $\bar{F}_{\mathrm{L}}$ is the force vector on the link atom due to the QM potential, and the partial derivatives can be expressed as:

$$\begin{aligned} \frac{\partial \bar{r}_{\mathrm{L}}}{\partial x_{\mathrm{QM}}} &= \left(1 - \frac{d_{\mathrm{L-QM}}}{d_{\mathrm{MM-QM}}}\right)\bar{i} + \frac{d_{\mathrm{L-QM}}\left(x_{\mathrm{MM}} - x_{\mathrm{QM}}\right)}{d_{\mathrm{MM}} - \mathrm{QM}^3}\left(\bar{r}_{\mathrm{MM}} - \bar{r}_{\mathrm{QM}}\right) \\ \frac{\partial \bar{r}_{\mathrm{L}}}{\partial x_{\mathrm{MM}}} &= \frac{d_{\mathrm{L-QM}}}{d_{\mathrm{MM-QM}}}\bar{i} - \frac{d_{\mathrm{L-QM}}\left(x_{\mathrm{MM}} - x_{\mathrm{QM}}\right)}{d_{\mathrm{MM-QM}^3}}\left(\bar{r}_{\mathrm{MM}} - \bar{r}_{\mathrm{QM}}\right) \end{aligned} \tag{32}$$

where $\bar{i}$ is the Cartesian x unit vector and $d_{\mathrm{MM\ -\ QM}} = |\bar{r}_{\mathrm{MM}} - \bar{r}_{\mathrm{QM}}|$.

There are a number of advantages to this link atom approach. The first is that constraining the link atom position to the QM-MM link pair bond vector does not introduce extra degrees of freedom into the calculation. This makes temperature and pressure control easier, and also means that statistical averages and fluctuations can be directly compared between pure MM and hybrid QM/MM simulations. The second is that the entire link atom procedure is transparent to the user. The user simply selects which atoms are to be treated quantum mechanically and *sander* then determines what adjustments need to be made to ensure the total system charge is conserved, which bonds are to be broken, how many link atoms are needed, and where they are to be placed. The third is that the link atom position need only be known by the QM part of the code and as such there is no need for special restart file formats or extension of the coordinate, force or velocity arrays. This makes the implementation significantly easier and reduces the potential for coding errors. A fourth advantage is that the definition of the link atom position (30) ensures that the link atom is always in the correct position each time the QM potential is calculated. Our experience shows that this greatly improves the convergence behavior and stability of QM/MM MD simulations and allows time steps of the same magnitude as are typically used in classical MD simulations.

The remaining details of how the QM-MM boundary is treated are as follows: the MM bond terms between QM and MM atoms are calculated classically using the Amber force field parameters, as are any angle or dihedral term that include at least one MM atom. The Lennard–Jones interactions between QM-MM atom pairs are calculated in the same way as described in the section above with exclusion of 1–2 and 1–3 interactions and scaling of 1–4 interactions as in the Amber force field.[50] What remains are the electrostatic interactions between QM and MM atoms around the region of the link atom. A number of schemes have been proposed including simply neglecting all charges that are within three bonds of a QM atom or alternatively scaling the charges. However, we have found that all of these approaches are unsatisfactory and so in Amber we have chosen to use the method advocated by Field et al.,[4] where all electrostatic interactions between all MM atoms (excluding MM atoms directly bonded to a QM atom) within the user specified cutoff distance of any QM atom are calculated for all QM atoms, including the link atom, without exclusion or scaling. The electrostatic interactions of the MM link pair atom are replaced by those of the link atom, whereas the VDW interactions remain with the MM link pair atom. The link atom is treated in the same way as the "real" QM atoms, sharing the same nonbond list as the other QM atoms. Test calculations have shown that this approach gives a significantly better distribution of the charge on QM atoms around the QM-MM interface than is observed if the QM link atom interacts only with other QM atoms as was the case with the Amber v8 implementation. Figure 2 illustrates several alternatives for a tryptophan dipeptide in a sphere of water. Here it can be seen that for case 3 the link atom charge remains stable and close to that of the other CB hydrogens. In the alternative cases the link atom charge fluctuates significantly because of an inbalance of the MM charge field experienced by the link and non-link QM atoms.

When adding link atoms to a QM/MM system it is essential that the total charge of the system be maintained. Charge conservation with link atoms in our implementation is achieved by one of two methods. Any difference in charge between the QM region and the parameterized (RESP) charges of the MM atoms that are replaced by QM atoms can either be distributed to the MM atoms surrounding the MM link atom pair or can be distributed evenly across all the remaining MM atoms, the later option being the default behavior. Since the integer charge of the QM region is determined at runtime so this correction can be done as part of the initial simulation setup.

To highlight the importance of rigorous charge conservation in link atom schemes, and to facilitate a comparison between our link atom scheme and the various QM/MM frontier approaches used in other QM/MM implementations we studied the gas phase deprotonation energies of a series of aliphatic alcohols and carbonic acids as was conducted by König et al. in their evaluation of various QM/MM frontier treatments with SCC-DFTB.[51] We used the exact methodology they used to generate Table 3 in their paper. To facilitate a direct comparison with their results we used CHARMM parameters and charges for the MM region. As described earlier Amber 9 offers two dif-

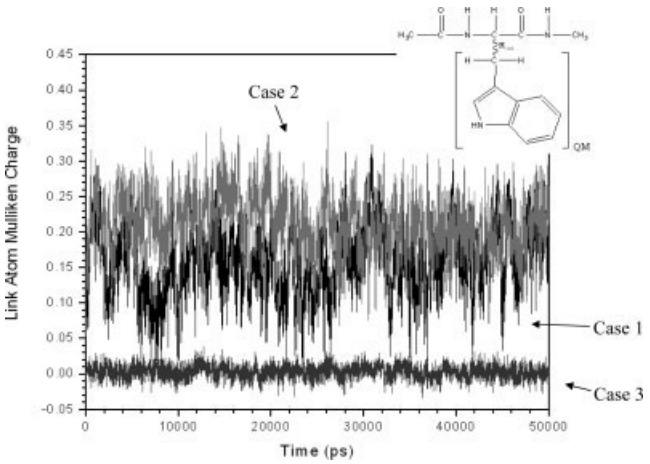

Figure 2. Plot showing the link atom Mulliken charge as a function of time for a MD simulation of tryptophan dipeptide in a 20 Å radius TIP3P solvent cap with no cutoff. Inset shows the QM atoms and the bond vector along which the single link atom was placed. The charge for three different link atom methods is shown. Case 1 is where the link atom interacts only with QM atoms and the MM link pair atom (Cα) interacts with all QM atoms except the link atom. Case 2 is similar to case 1 except that the interaction between the MM link pair atom and the non-link QM atoms is also excluded. Case 3 is the method currently implemented in Amber 9 where by the link atom interacts with both QM and MM atoms and MM link pair atoms are excluded from all electrostatic interactions.

ferent options for dealing with charge conservation. We tested the ability of both methods to reproduce the pure QM gas phase deprotonation energy as well as testing the effects of not rigorously conserving charge. Table 1 shows the results for SCC-DFTB and PM3-PDDG. Failure to ensure that the total system charge is conserved, somewhat analogous to the single link atom (SLA) approach in König et al.'s[51] paper, leads to significant errors in the predicted deprotonation energies. However, our link atom method, when coupled with charge conservation performs very well with errors that are comparable with those observed for the divided frontier charge (DIV) approach that König et al.[51] found to give the best overall performance in their study.

Energy Conservation and Testing

Single Point Accuracy

One problem we encountered when attempting to verify the accuracy of our QM/MM implementation was the inconsistency in energies reported from various semi-empirical QM implementations. For a system as simple as N-methyl acetamide (NMA) in gas phase, MOPAC v6.0,[17] MOPAC 2007,[52] Gaussian 2003,[53] Dynamo v2.0,[10] and CHARMM c31b2[9] all yield different values for the predicted heat of formation (Table 2). These differences have been tracked down to inconsistencies in values of certain constants, sometimes even within a single code. Unlike *ab initio* methods, which can be expressed in atomic units that are independent of conversion factors, semi-empirical matrix elements are defined through fitted parameters that have experimental units (usually in electron Volt). In our QM implementation we have (somewhat arbitrarily) elected to use the same constants and conversion factors that are used in Dynamo. Our code will therefore exactly reproduce the energies reported by Dynamo. The supplemental material shows how to modify CHARMM, Mopac 6.0 and Gaussian so that they will reproduce the Amber or Dynamo QM energies to 10 significant figures.

Gradient Accuracy

People running classical force field simulations can generally expect that the forces used in MD are accurately the gradients of the potential energy so that a NVE simulation should yield constant total energy with respect to time. However, this has often not been true for QM/MM simulations since the QM portion of a QM/MM calculation uses an iterative SCF procedure, the forces are only sufficiently accurate to conserve energy if the difference in energy between subsequent SCF steps is converged to the order of around 1×10^{-8} to 1×10^{-9} kcal/mol.

We tested the ability of *sander* v9, *sander* v8, ROAR v2.1, CHARMM v31b2, and Dynamo v2 to simulate the NVE ensemble for a simple QM system undergoing dynamics by running a 2.5 ns QM MD simulation of NMA in gas phase. We constructed the same simulation in all the codes. Starting from a pre-equilibrated structure of NMA at 300 K we ran 5,000,000 steps of molecular dynamics in a single run in gas phase with no thermostat and an integration time step of 0.5 fs, yielding 2.5 ns of simulation. Since ROAR would not run without a thermostat we selected the Nose Hoover Chain Thermostat with 1 chain and 1 thermostat per chain and a thermostat mass of 1.0×10^{16}. Such a thermostat mass is sufficiently large that it should approximate the NVE ensemble over the timescale of the simulation. The energy and forces at each step of MD were calculated using the PM3 semi-empirical Hamiltonian and the requested SCF convergence criteria was set at 1.0×10^{-9} kcal/mol. No cutoffs were used in the calculation and no atoms had their motions restricted or damped in any way.

Figure 3 shows the total system energy (SCF energy + kinetic energy) for each of the simulations as a function of time. Both *sander* v9 and DYNAMO v2 give the same answer and conserve energy to within 0.02 kcal/mol over the full 2.5 ns simulation. ROAR v2.1 and CHARMM v31b2 show similar behavior, with a gradual cooling of the system as kinetic energy is slowly lost due to inaccuracies in the gradients. *sander* v8 initially looses energy at a very rapid rate and then shows an abrupt jump in the energy after 136,000 steps of MD. This is followed by continued loss of energy and sudden, seemingly random, jumps in energy for the rest of the simulation; (we still do not understand the origins of this odd behavior). The CHARMM simulation on the other hand simply cooled down to 0.1 K over 1.5 ns before crashing with a segmentation fault.

The *sander* v8, ROAR v2.1 and CHARMM v31b2 results are very worrying since they imply that the gradients are inaccurate even with a stringent SCF convergence criteria. It would appear that of the five codes tested only *sander* v9 and DYNAMO v2 have accurate gradients. The analytical gradients in our *sander* v9 implementation agree well with numerical gradients: for an

Table 1. Predicted Gas Phase Deprotonation Energies, as a Function of the Charge Conservation Approach, for QM Fragments Embedded in an MM Environment Using Amber 9's Link Atom Approach (in kcal/mol).[a]

	SCC-DFTB				PM3-PDDG			
Model	Pure QM[b]	NA[c]	NN[c]	ALL[c]	Pure QM[b]	NA[c]	NN[c]	ALL[c]
A. **CH_3−OH**	397.1				387.9			
B. CH_3−**CH_2**−OH	397.0	368.2	397.7	401.2	383.9	355.5	384.6	388.6
dev		−28.8	0.7	4.2		−28.4	0.7	4.7
C. CH_3−CH_2−**CH_2**−OH	395.5	382.2	404.4	400.8	382.9	369.5	391.6	387.8
dev		−13.3	8.8	5.2		−13.4	8.7	4.9
D. CH_3−CH_2−CH_2−**CH_2**−OH		381.7	403.9	397.9	384.3	369.1	391.2	385.1
dev		−13.1	9.0	3.0		−15.2	6.9	0.8
E. **CH_3−CH_2**−OH	394.9				383.9			
F. CH_3−**CH_2−CH_2**−OH	395.5	370.3	397.9	397.1	382.9	358.5	384.5	383.9
dev		−25.2	2.3	1.6		−24.4	1.6	1.0
G. CH_3−CH_2−**CH_2−CH_2**−OH	394.9	380.8	399.3	395.8	384.3	370.3	384.6	385.0
dev		−14.1	4.4	1.0		−14.0	0.3	0.7
H. CH_3−CH_2−CH_2−**CH_2−CH_2**−OH	394.6	380.6	399.1	393.9	384.0	370.1	387.8	383.2
dev		−14.0	4.5	−0.7		−13.9	3.8	−0.8
I. **CH_3−COOH**	366.3				349.2			
J. CH_3−**CH_2−COOH**	365.5	338.4	363.5	366.2	348.8	323.5	345.5	349.0
dev		−17.1	−2.0	0.7		−25.3	−3.3	0.2
K. CH_3−CH_2−**CH_2−COOH**	364.9	349.4	369.5	366.4	348.7	333.2	352.1	349.4
dev		−15.5	4.5	1.5		−15.5	3.4	0.8
L. CH_3−CH_2−CH_2−**CH_2−COOH**	364.2	349.2	368.0	362.8	348.1	332.9	350.6	345.9
dev		−15.1	3.7	−1.4		−15.2	2.5	−2.2
M. CH_3−CH_2−CH_2−CH_2−**CH_2−COOH**	364.4	349.1	368.2	361.4	348.0	332.8	350.6	344.6
dev		−15.3	3.8	−3.0		−15.2	2.6	−3.4

[a]The QM region is shown in the bold faced type. A single link atom is added at the bond crossing the QM/MM boundary. Deviations are shown relative to the entire molecule computed using a pure QM treatment. For SCC-DFTB we used a value of 141.8 kcal/mol for the energy of an isolated proton[51] and for PM3/PDDG we used a value of 367.2 kcal/mol[57].

[b]The Pure QM column indicates a calculation encompassing the entire molecule.

[c]The different link atom charge conservation algorithms, with the corresponding Amber 9 control keyword, are as follows:

NA (*adjust_q = 0*), No adjustment is made to preserve charge.

NN (*adjust_q = 1*), Nearest Neighbor. The charge correction is applied to the nearest *nlink* MM atoms to the MM atoms that are replaced electrostatically by link atoms.

ALL (*adjust_q = 2*), Default behavior. The charge correction is divided equally amongst all MM atoms (except for those adjacent to link atoms).

SCF and density matrix element convergence of 10^{-9} kcal/mol we see an RMS gradient error, for the pure QM gas phase NMA system discussed above, of approximately 1.1×10^{-8} kcal/mol/Å. This compares favorably with the error in the classical MM force field gradients of approximately 0.5×10^{-9} kcal/mol/Å. The remaining error reflects the limitations of the finite difference approach, not actual errors in the MM gradients.

The increased accuracy of the gradients in our new QM/MM implementation is due in part to our careful rewriting of the MOPAC code. This included centralizing all of the constants and conversion factors and ensuring, unlike in MOPAC v6, as illustrated in the supplementary material, that only one value of each constant was used throughout the code and that all constants and conversion factors are self consistent. Additionally there are some conditions where exponentials are vanishingly small, and so can be skipped for performance reasons, and also conditions when rotating from local frames to molecular frames in which certain vectors can be assumed to be parallel to an axis when they are very close to that axis. We have ensured that the checks used in both the energy and derivative sections of the code are consistent. We have also ensured that pseudo diagonalizations[54] of the Fock matrix are only done during the middle part of the SCF. By switching to full diagonalisations for the last few SCF cycles we have ensured that our gradients are accurate. Failure to do this gives analytical gradients that at best have an RMSD of 1×10^{-4} kcal/mol/Å from the numerical gradients.

Our link atom approach, as described earlier, also conserves energy when run in the NVE ensemble, as illustrated in Figure 4. This shows the total energy, classical potential energy + scf energy + kinetic energy, for a 10 ns simulation of alanine dipeptide in which the central part of the molecule was treated quantum mechanically using the PM3 Hamiltonian, and the remainder of the molecule was treated using the classical FF99 force field.[55] Two hydrogen link atoms were added at a distance

Table 2. SCF Energies Reported by Different Semi-Empirical Packages for Identical Single Point Energy Evaluations of *N*-Methyl Acetamide.

Program	AM1 ESCF (Kcal/mol)	PM3 ESCF (Kcal/mol)
AMBER v9.0	−34.92134208	−44.35423579
CHARMM c33b1	−34.97754796 (−34.92134208*)	−44.40827788 (−44.35423579*)
MOPAC v6.0	−34.82651486 (−34.92134209*)	−44.27659532 (−44.35423579*)
MOPAC 2007[a]	−34.90320	−44.39969
Dynamo v2.0	−34.92134208	−44.35423579
Gaussian 03 Rev C.01	−34.97477068[b] (−34.92134208*)	−44.40566821[b] (−44.35423579*)

[a]Source code not available so higher precision printing was not possible.
[b]With iop(4/22 = 100) set.
*Values in brackets show energy reported after code changes described in the supplemental material.
In all cases the SCF convergence was set to 1.0×10^{-8} Kcal/mol and the input structure was identical.

of 1.00 Å, one along each of the C—N bonds that crossed the QM/MM boundary. All other aspects of the simulation were the same as the NMA calculation described above. This is not an ideal choice of QM/MM boundary, but is used here to test the accuracy of the gradients in our implementation. It can be seen from Figure 4 that energy is successfully converged with a change of only 0.03 kcal/mol of energy over the entire 10 ns simulation.

Performance

A common problem that is encountered by researchers carrying out QM/MM simulations is the excessive computational resources such calculations can demand. Our trials with *sander* 8 showed that performance was indeed an issue as illustrated by Tables 3 and 4. The *sander* 8 QM/MM implementation will also not run in parallel and so it is not possible to increase the calculation speed by going to large numbers of processors.

Inspection of the code revealed that the poor performance was due to outdated programming practices such as the use of computed GOTO's as well as inefficient programming techniques such as poor use of memory, branching within loops and reading and writing integral data to disk at every step. To address the performance issues in *sander* 9 we have completely rewritten the code using current programming paradigms as well as more modern machine resource expectations.

Tables 3 and 4 give performance comparisons between the five main codes we have discussed in this article. Table 3 gives timings for malachite green (50 atoms) in gas phase for 1000 steps of 1 fs each and a SCF convergence criteria for all pro-

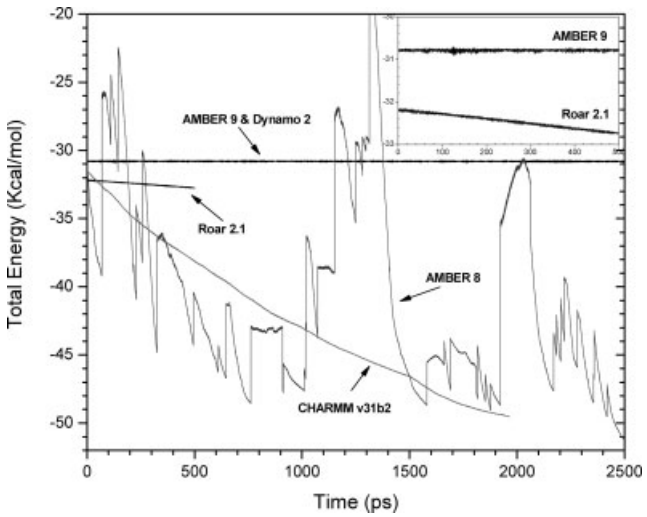

Figure 3. Total energy vs. time for a pure QM MD simulation (PM3) of NMA at 300 K in the NVE ensemble. The results from each program are labelled. The Amber v9 and Dynamo v2 results were the same and so overlap on the plot. (Inset shows zoom of Amber v9 and Roar 2.1 results over the first 500 ps.)

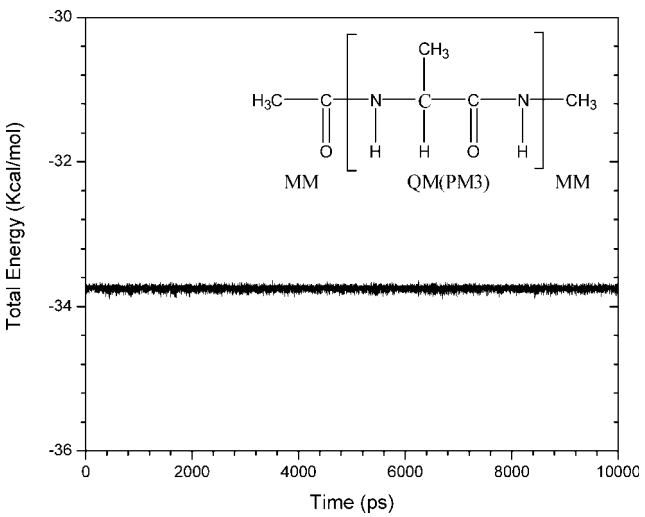

Figure 4. Total energy vs. time for a combined QM/MM simulation with link atoms of Alanine Dipeptide at 300 K in the NVE ensemble using Amber v9. Inset shows the QM and MM partitioning. Hydrogen link atoms were placed at a distance of 1 Å from each nitrogen along the bond vector that was broken.

Table 3. Wallclock Time in Seconds for a 1000 Step Pure QM/MD Simulation of Malachite Green (50 atoms) in Gas Phase on a Single Processor of an Intel Pentium-D 3.2 Ghz Machine with 4 GB of 667 MHz DDR2 Memory.

Program	Time (s)	PS/Day
sander v9	120.4	717.8
sander v8	473.2	182.6
ROAR v2.1	548.3	157.6
DYNAMO v2.0	355.9	242.8
CHARMM v31b2	497.0	173.8

Times are averages over 10 runs. Simulation details: No cutoff, PM3, 1 fs time step, no shake, no coordinate file, SCF convergence = 1.0×10^{-8} Kcal/mol. Intel ifort v9.1.039 was used to compile all codes.

grams of 1×10^{-8} kcal/mol. As can be seen from this data our newly written QM implementation is significantly faster than the other codes. Table 4 shows a similar trend with timings for a QM/MM simulation of malachite green in a solvent cap of TIP3P water. Here the total system size was 5985 atoms, with 50 treated quantum mechanically and 5935 treated classically. Classical-only simulation times (using the gaff force field[56] for the solute) are also shown for comparison. Here the calculation was run with a time step of 1 fs for 1000 steps. The SCF convergence was set to 1×10^{-8} kcal/mol in all codes and a total of two simulations were run; a classical only simulation with a non-bonded cutoff of 14 Å and the equivalent QM/MM simulation. Again our new implementation is significantly faster achieving over 218 ps/day (437 ps/day if shake is used). This performance can also be somewhat improved by running simulations in parallel across several processors.

This increase in performance has been achieved through a number of routes, but has not been achieved by sacrificing accuracy or introducing any new approximations. One way we have done this is by dynamically adjusting the amount of memory used to the available memory on the machine. When the MOPAC code, from which all of the programs we have compared have their origins, was written memory was very sparse and so the atom-atom distances, one and two electron repulsion integrals, etc, were typically recalculated on the fly during the SCF procedure and the gradient evaluation. This is no longer the case. Even a basic workstation now has in excess of 1 GB of ram. With this in mind we have introduced a number of performance enhancing storage options in *sander* 9. By default the code will store as much as possible in memory. This includes storing the one electron integrals, both QM-QM and QM-MM in memory. *sander* v8 and ROAR write these to disk and then read them back in when needed whereas DYNAMO recalculates them on the fly. We also store many other calculation elements in memory whenever possible, including the two electron integrals, basis set expansion data, precomputed parameters and as many equations as possible that depend on the distance between atom pairs. In this way we do each calculation only once. All of these memory storage options are controllable so that the user may choose to save memory at the expense of performance.

We have also significantly improved performance by modernizing the code. We have modified our array storage to try to ensure that memory accesses are almost always linear. Combining several arrays into one ensures that the computer can access the next element that is required for a calculation by simply incrementing a pointer rather than fetching the value from a distant memory location. This maximizes the number of cache hits that are achieved and also allows the cpu to pre-fetch large blocks of memory efficiently.

Finally we have streamlined the code as much as possible. All branching is done outside of loops whenever possible and loops have been split so that we do not mix integer and floating point arithmetic within the same loop. This allows inner loops to be unrolled by the compiler and also allows for easy vectorization to take advantage of special hardware such as the SSE2 registers on Intel P4 chips.

We have also improved performance by writing the code for the purpose of doing QM/MM MD. In this way we have included only those options that are needed for doing MD simulations. This assumption also allows other enhancements to be made. For example the difference in structure between subsequent MD steps will generally be small allowing the density matrix from the previous step to be used as the starting point for the next step's SCF procedure. This greatly accelerates convergence of the SCF.

For explicit solvent periodic boundary simulations Table 5 shows a comparison of our new QM/MM PME implementation and the original QM/MM Ewald implementation based on the work of Nam et al.[40] In both cases the MM/MM interactions were calculated using PME. It can be seen that our PME implementation is significantly faster than the Ewald implementation. At 12,612 atoms it is over 2.3 times faster and this ratio grows as the total number of atoms in the simulation increases. Timings for the equivalent purely classical calculation are shown for comparison. The QM/MM PME implementation is also significantly more efficient in terms of memory usage. The LADH simulation with the QM/MM Ewald method requires more than 1100 MB of memory while the PME version requires only 17.6 MB.

The major remaining bottleneck is the matrix diagonalization step which while it has been optimized is currently not parallelized. This ultimately limits parallel scalability to approximate 8 cpus depending on the size of the QM and MM regions. We are

Table 4. Wallclock Time in Seconds for 1000 Steps QM/MM/MD and Classical Simulations of Malachite Green (QM, 50 atoms) in 1978 Molecule TIP3P Water Sphere (MM, 5935 atoms) on a Single Processor of an Intel Pentium-D 3.2 Ghz Machine with 4GB of 667 MHz DDR2 Memory.

Program	MM Only (s)	QMMM (s)	Ratio QMMM/MM Only
sander v9	189.5	395.4	2.09
sander v8	222.0	2349.8	10.58
ROAR v2.0	213.8	4123.6	19.29
DYNAMO v2.0	333.2	845.0	2.54
CHARMM v31b2	296.4	1366.9	4.61

Times are averages over 10 runs. Simulation details: 14 Å cutoff, 1 fs time step, constant temperature (300 K), no shake, SCF convergence = 1.0×10^{-8} Kcal/mol. Intel ifort v9.1.039 was used to compile all codes.

Table 5. Wallclock Time in Seconds for 1000 Steps QM/MM/MD and Classical Simulations of Various Systems in TIP3P Water with Full Treatment of Periodic Electrostatics Using a QM/MM Compatible PME Implementation and Sander v9.0.

System	NATOM (NQM)	QMMM (PME) (s)	QMMM (Ewald)[a] (s)	MM Only (s)	Ratio QMMM/MM Only (PME/Ewald)
Malachite Green	12,612 (50 QM)	514	1183	210	2.45/5.63
Dihydrofolate Reductase	23,558 (66 QM)[b]	919	2189	420	2.19/5.22
Liver Alcohol Dehydrogenase	76,723 (71 QM)	2818	7032	1505	1.87/4.67

[a]Timings here are for QM-MM and QM-QM interactions calculated using Ewald and MM-MM interactions calculated using PME. An 8 Å cutoff was used for the direct space sum of both the electrostatic and van der Waals interactions.
[b]Consists of 65 MM atoms treated quantum mechanically and 1 link atom.
Times are for a single processor of an Intel Pentium-D 3.2 Ghz machine with 4 GB of 667 MHz DDR2 memory and are averages over 10 runs.

working to remove this serial bottleneck in future versions of the software.

Overall our new QM/MM implementation is sufficiently fast that a portion of the system can be treated quantum mechanically without an excessive overhead in computing time. For example a periodic boundary simulation of LADH in TIP3P water involves a total of 76,723 atoms. Choosing to treat the coenzyme, NADH, in this system using the PM3 Hamiltonian (71 atoms) requires only 1.87 times more cpu time with respect to doing the calculation purely classically (Table 5).

Conclusions and Planned Development

In this article we have discussed our new semi-empirical hybrid QM/MM routine that forms part of the *sander* v9 module in Amber v9. This implementation is a significant improvement over previous code. We have greatly simplified the way in which QM/MM simulations are setup and run such that the user now simply adds a few extra options to their normal input file. If a simulation is setup to run in *sander* using a classical MM force field then it is a trivial matter to have part of this system treated quantum mechanically. A rewrite of the QM portion of the code has significantly improved the calculation speed and at the same time corrected deficiencies in the gradient accuracies of previous codes. Our new QM/MM implementation can successfully conserve energy and so does not need to rely on a thermostat to correct these deficiencies. The gradient accuracy and the enhanced computation speed has been achieved simply by careful coding, we have not implemented any extra approximations or accelerated convergence options. As such there is no reason why these improvements could not be added to other semi-empirical QM/MM codes with minimal effort and indeed work is currently underway to integrate this code into CHARMM.

Our implementation supports either pure QM or QM/MM simulations using either the PM3, PDDG/PM3, PM3CARB1, AM1, MNDO, PDDG/MNDO, and DFT/B Hamiltonians. We have also implemented a complete treatment of long range electrostatics using a QM/MM modified PME method derived in this article as well as a QM/MM compatible Generalized Born model based on the approach described by Pellegrini and Field.[44]

Acknowledgments

RCW and DAC thank the San Diego Supercomputer Center for their continued support of this project and future developments in Amber 10 through their Strategic Applications Collaboration program. The authors also thank Adrian Roitberg and Gustavo Seabra for their support and helpful discussions; the authors of the MOPAC code, which although rewritten in our *sander* v9 implementation, provided the basis for our semi-empirical QM routines; and also the authors of the ROAR code, which although not part of our new implementation, provided a useful template to work from.

References

1. Warshel, A.; Levitt, M. J Mol Biol 1976, 103, 227.
2. Aqvist, J.; Warshel, A. J Mol Biol 1992, 224, 7.
3. Warshel, A. Computer Modelling of Chemical Reactions in Enzymes and Solutions; Wiley: New York, 1991.
4. Field, M. J.; Bash, P. A.; Karplus, M. J Comput Chem 1990, 11, 700.
5. Luzhkov, V.; Warshel, A. J Comput Chem 1992, 13, 199.
6. Stanton, R. V.; Hartsough, D. S.; Merz, K. M. J Phys Chem 1993, 97, 11868.
7. Gao, J. L. In Reviews in Computational Chemistry, Vol. 7, Lipkowitz, K. B.; Boyd, D. B., Eds.; VCH: New York, 1995; pp. p119.
8. Amara, P.; Field, M. J. In The Encyclopedia of Computational Chemistry, Vol. 1, Schleyer, P. V.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P.; Schaefer, H. F.; Schreiner, P. R., Eds.; Wiley: Chichester, 1998; p. 31.
9. Mackerell, A. D.; Brooks, C. L. III; Nilsson, L.; Roux, B.; Won, Y.; Karplus, M. CHARMM: The Energy Function and Its Parameterization with an Overview of the Program, in The Encyclopedia of Computational Chemistry, Vol. 1, Wiley: Chichester, 1998; pp. 271–277.
10. Field, M. J.; Albe, M.; Bret, C.; Proust-De Martin, F.; Thomas, A. J Comput Chem 2000, 21, 1088.
11. Jorgensen, W. L.; Tirado-Rives, J. J Comput Chem 2005, 26, 1689.
12. Case, D. A.; Darden, T. A.; Cheatham, T. E.; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M., Jr.; Pearlman, D. A.; Crowley, M.; Walker, R. C.; Zhang, W.; Wang, B.; Hayik, S.; Roitberg, A. E.; Seabra, G.; Wong, K. F.; Paesani, F.; Wu, X.; Brozell, S.; Tsui, V.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.;

Cui, G.; Beroza, P.; Mathews, D. H.; Schafmeister, C. E. A. F.; Ross, W. S.; Kollman, P. A. AMBER 9; University of California: San Francisco, 2006.
13. Singh, U. C.; Kollman, P. A. J Comput Chem 1986, 7, 718.
14. Singh, U. C.; Kollman, P. QCPE Bull 1982, 2, 17.
15. Weiner, P. K.; Kollman, P. A. J Comput Chem 1981, 2, 287.
16. Cheng, A.; Stanton, R. S.; Vincent, J. J.; van der Vaart, A.; Damodaran, K. V.; Dixon, S. L.; Hartsough, D. S.; Mori, M.; Best, S. A.; Monard, G.; Garcia, M.; Van Zant, L.; Merz, K. M., Jr.; Roar, 2.1 ed.; The Pennsylvania State University, 2002.
17. Stewart, J. J. P. Quantum Chem Program Exchange 1993, 455, 42.
18. Dixon, S. L.; van der Vaart, A.; Wang, B.; Gogonea, V.; Vincent, J. J.; Brothers, E. N.; Suarez, D.; Westerhoff, L. M.; Merz, K. M., Jr., DivCon Lite; The Pennsylvania State University: University Park, PA, 16802, 2004.
19. Darden, T.; York, D.; Pedersen, L. J Chem Phys 1993, 98, 10089.
20. Tsui, V.; Case, D. A. Biopolymers 2000, 56, 275.
21. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. J Am Chem Soc 1990, 112, 6127.
22. Onufriev, A.; Bashford, D.; Case, D. A. Proteins-Structure Function and Bioinformatics 2004, 55, 383.
23. Stewart, J. J. P. J Comput Chem 1989, 10, 209.
24. Stewart, J. J. P. J Comput Chem 1989, 10, 221.
25. Repasky, M. P.; Chandrasekhar, J.; Jorgensen, W. L. J Comput Chem 2002, 23, 1601.
26. McNamara, J. P.; Muslim, A. M.; Abdel-Aal, H.; Wang, H.; Mohr, M.; Hillier, I. H.; Bryce, R. A. Chem Phys Lett 2004, 394, 429.
27. Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. J Am Chem Soc 1985, 107, 3902.
28. Dewar, M. J. S.; Thiel, W. J Am Chem Soc 1977, 99, 4899.
29. Dewar, M. J. S.; Thiel, W. J Am Chem Soc 1977, 99, 4907.
30. Seifert, G.; Porezag, D.; Frauenheim, T. Int J Quantum Chem 1996, 58, 185.
31. Porezag, D.; Frauenheim, T.; Kohler, T.; Seifert, G.; Kaschner, R. Phys Rev B 1995, 51, 12947.
32. Elstner, M.; Hobza, P.; Frauenheim, T.; Suhai, S.; Kaxiras, E. J Chem Phys 2001, 114, 5149.
33. Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T.; Suhai, S.; Seifert, G. Phys Rev B 1998, 58, 7260.
34. Gustavo, M. S.; Walker, R. C.; Elstner, M.; Case, D. A.; Roitberg, A. E. J Phys Chem A 2007, 111, 5655.
35. Ryckaert, J. P.; Cicotti, G.; Berendsen, H. J. J Comput Phys 1977, 23, 327.
36. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. J Chem Phys 1995, 103, 8577.
37. Freindorf, M.; Shao, Y. H.; Furlani, T. R.; Kong, J. J Comput Chem 2005, 26, 1270.
38. Riccardi, D.; Li, G. H.; Cui, Q. J Phys Chem B 2004, 108, 6467.
39. Ewald, P. P. Annal Der Phys 1921, 64, 253.
40. Nam, K.; Gao, J. L.; York, D. J Chem Theory Comput 2005, 1, 2.
41. Laino, T.; Mohamed, F.; Laio, A.; Parrinello, M. J Chem Theory Comput 2006, 2, 1370.
42. Cook, D. B. Handbook of Computational Quantum Chemistry; Dover: New York, 2005.
43. Gauss, J. Molecular properties. In Modern Methods and Algorithms of Quantum Chemistry; Grotendorst, J., Ed.; John von Neumann Institute for Computing: Jülich, 2000; p. 509.
44. Pellegrini, E.; Field, M. J. J Phys Chem A 2002, 106, 1316.
45. Zhang, Y. K.; Lee, T. S.; Yang, W. T. J Chem Phys 1999, 110, 46.
46. Gao, J. L.; Amara, P.; Alhambra, C.; Field, M. J. J Phys Chem A 1998, 102, 4714.
47. Philipp, D. M.; Friesner, R. A. J Comput Chem 1999, 20, 1468.
48. Bersuker, I. B.; Leong, M. K.; Boggs, J. E.; Pearlman, R. S. Int J Quantum Chem 1997, 63, 1051.
49. Maseras, F.; Morokuma, K. J Comput Chem 1995, 16, 1170.
50. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. J Am Chem Soc 1995, 117, 5179.
51. Konig, P. H.; Hoffmann, M.; Frauenheim, T.; Cui, Q. J Phys Chem B 2005, 109, 9082.
52. Stewart, J. J. P.Mopac 2007 v1.0; Stewart Computational Chemistry, 2007.
53. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Montgomery, J. A., Jr.; Vreven, T.; Kudin, K. N.; Burant, J. C.; Millam, J. M.; Iyengar, S. S.; Tomasi, J.; Barone, V.; Mennucci, B.; Cossi, M.; Scalmani, G.; Rega, N.; Petersson, G. A.; Nakatsuji, H.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Klene, M.; Li, X.; Knox, J. E.; Hratchian, H. P.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Ayala, P. Y.; Morokuma, K.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Zakrzewski, V. G.; Dapprich, S.; Daniels, A. D.; Strain, M. C.; Farkas, O.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Ortiz, J. V.; Cui, Q.; Baboul, A. G.; Clifford, S.; Cioslowski, J.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Gonzalez, C.; Pople, J. A. Gaussian 2003, Revision C01; Wallingford CT, 2004.
54. Stewart, J. J. P.; Csaszar, P.; Pulay, P. J Comput Chem 1982, 3, 227.
55. Wang, J. M.; Cieplak, P.; Kollman, P. A. J Comput Chem 2000, 21, 1049.
56. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. J Comput Chem 2004, 25, 1157.
57. Stull, D. R. JANAF Thermochemical Tables NSRDS-NBS37, 1971.