JCTC Journal of Chemical Theory and Computation

Exploiting QM/MM Capabilities in Geometry Optimization: A Microiterative Approach Using Electrostatic Embedding

Johannes Kästner,*[,†,‡] Stephan Thiel,[†] Hans Martin Senn,[†] Paul Sherwood,[‡] and Walter Thiel[†]

Max-Planck-Institut für Kohlenforschung, Kaiser-Wilhelm-Platz 1, D-45470 Mülheim an der Ruhr, Germany, and Computational Science and Engineering Department, CCLRC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, United Kingdom

Received November 28, 2006

Abstract: We present a microiterative adiabatic scheme for quantum mechanical/molecular mechanical (QM/MM) energy minimization that fully optimizes the MM part in each QM macroiteration. This scheme is applicable not only to mechanical embedding but also to electrostatic and polarized embedding. The electrostatic QM/MM interactions in the microiterations are calculated from electrostatic potential charges fitted on the fly to the QM density. Corrections to the energy and gradient expressions ensure that macro- and microiterations are performed on the same energy surface. This results in excellent convergence properties and no loss of accuracy compared to standard optimization. We test our implementation on water clusters and on two enzymes using electrostatic embedding, as well as on a surface example using polarized embedding with a shell model. Our scheme is especially well-suited for systems containing large MM regions, since the computational effort for the optimization is almost independent of the MM system size. The microiterations reduce the number of required QM calculations typically by a factor of 2−10, depending on the system.

I. Introduction

Combined quantum mechanical/molecular mechanical (QM/MM) methods are increasingly applied to model large systems, from solid state[1,2] over surface catalysis[3] to biological systems.[4−6] In this approach, the chemically reactive part is described with an accurate, but computationally demanding, quantum mechanical method, while the environment is treated by force fields. These are computationally highly efficient but unable to describe chemical reactions, such as the breaking or forming of bonds. The systems frequently considered nowadays contain several tens to a few hundred QM atoms and many thousands of MM atoms.

When higher-level QM methods [ab initio or density functional theory (DFT)] are used in QM/MM treatments, the QM calculation generally takes much longer than the MM calculation. For a given size of the QM region, an extension of the MM region does not noticeably increase the computation time for a given configuration. Geometry optimizations, however, need more and more energy and gradient evaluations the larger the system becomes. As one QM calculation is in principle required for each such evaluation, the computational effort of QM/MM structure optimizations increases steeply with system size.

Microiterative QM/MM optimization schemes have been proposed[7−15] that exploit the computational advantage of treating a major part of the system with MM methods: the system is divided into an inner region including, but not limited to, the QM part and an outer region covering the rest of the system, as illustrated in Figure 1. In microiterative schemes, the regions are optimized separately. One can either use an adiabatic scheme,[7−9,11,15] that is, fully optimize the outer region after each optimization step of the inner region, or use an alternating scheme,[10,12] in which each region is completely optimized with the other one frozen. Both

* Corresponding author e-mail: J.Kaestner@dl.ac.uk.
[†] Max-Planck-Institut für Kohlenforschung.
[‡] CCLRC Daresbury Laboratory.

10.1021/ct600346p CCC: $37.00
Published on Web 03/10/2007

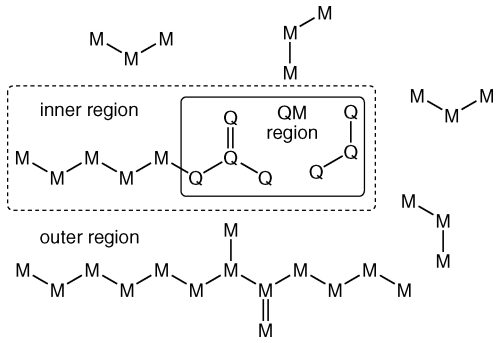

Figure 1. Partitioning of the system: Atoms in the QM region are described quantum mechanically, the remainder by molecular mechanics. Macroiterations are performed in the inner region, which has to include all QM atoms. Microiterations are performed in the outer region.

schemes are iterated until both regions are converged. This work uses the adiabatic scheme to minimize the number of required QM calculations.

Computation time is saved because the number of costly optimization steps in the inner region (macroiterations) is reduced, at the expense of the number of optimization steps in the outer region (microiterations). In mechanical QM/MM embedding, the interaction between the QM and the MM parts is calculated at the force-field level. Thus a microiterative scheme can be applied straightforwardly.[7,13,15]

Using electrostatic QM/MM embedding, the electrostatic interaction between the QM and the MM parts is evaluated by the QM code. This in principle requires a QM calculation even if only MM atoms are moved during the microiterations. Thus, some approximations are necessary in order to reduce the CPU demand. The density of the QM part may be frozen during the microiterations, which almost retains the convergence of the optimizer compared with full QM calculations[14] and saves some computation time as only one-electron integrals have to be evaluated. Alternatively, charges that represent the electrostatic potential of the QM part (ESP charges) may be used to calculate the electrostatic QM/MM interaction in the microiterations. Without corrections, this leads to different potential-energy surfaces for the micro- and macroiterations, which hampers convergence.[14] If ESP charges are used for the electrostatic QM/MM interaction in both the macro- and microiterations, convergence is good,[10,12] but the QM/MM Hamiltonian is altered. To correct the gradient obtained from ESP charges, one can use the exact QM/MM forces at the initial geometry of the microiterations plus the change in the gradient provided by ESP charges during the microiterations.[11] This latter approach is adopted in our work.

Besides the computational cost of the QM calculations, the time required for the optimizer itself becomes an issue for systems with several thousand atoms if the optimizer scales worse than linearly with system size.

Two types of optimizers are frequently employed in atomistic simulations. First-order methods like steepest descent or conjugate gradient methods take only the gradient of the total energy with respect to the nuclear coordinates into account. The computation time for evaluating the search direction is small and generally scales as $\mathcal{O}(N)$, with N being the number of degrees of freedom to be optimized. However, in general, many energy and gradient evaluations are required to achieve convergence. First-order methods are suited for large systems if the energy and gradient evaluations are very fast (as, e.g., in MM). By contrast, second-order methods, including Newton−Raphson, partitioned rational function optimization,[16] or the Broyden−Fletcher−Goldfarb−Shanno[17−20] (BFGS) algorithm, make use of the Hessian. Therefore, they require fewer steps until convergence, but the time to evaluate each search step scales as $\mathcal{O}(N^3)$. The limited-memory BFGS[21,22] (L-BFGS) variant utilizes Hessian information accumulated from the gradients over the history of the optimization and thus scales as $\mathcal{O}(N)$ in both CPU time and memory requirements. It is particularly suited for large-scale optimizations and is therefore employed in this work.

The choice of coordinates also has major effects on the scaling and convergence behavior of the optimizer. Cartesians are easy and efficient to handle, but highly coupled, which impedes convergence. Internal coordinates, such as Z-matrix coordinates, are less coupled, but they are biased. Redundant internal coordinates can be constructed in an unbiased manner, but their redundancy may spoil convergence, as changes in the coordinates do not necessarily lead to changes in the energy. For all variants of internal coordinates covering the whole system, the conversion between internals and Cartesians (which are needed to calculate the energy) scales as $\mathcal{O}(N^3)$, and internals are thus impractical for large systems. We use hybrid delocalized internal coordinates[9] (HDLC), where both the time and memory requirements scale as $\mathcal{O}(N)$. This scaling behavior is achieved by dividing the system into fragments and using internals only within each fragment.

In section II, we present a microiterative scheme for geometry optimization in QM/MM calculations which has been implemented in the general-purpose QM/MM software package ChemShell[23,24] as an extension to the HDLC optimizer.[9] In section III, we describe its performance in a variety of QM/MM calculations using semiempirical and DFT methods as QM components and standard as well as polarizable force fields (shell models) as MM components. These calculations were carried out using the ChemShell software interfaced to MNDO,[25] TURBOMOLE,[26−30] and GAMESS-UK[31,32] as QM codes and GROMOS[33] and GULP[34] as MM codes; the DL_POLY code included in ChemShell was used for CHARMM.[35] We show that the microiterative scheme reduces the computation times for QM/MM geometry optimizations with electrostatic embedding typically by a factor of 2−10.

II. Theory

II.A. Definitions and Notation. We apply an additive QM/MM scheme with link atoms where the total QM/MM energy is defined as

$$E = E_{\mathrm{qm}}(x_{\mathrm{qm}}) + E_{\mathrm{qm/mm}}(x_{\mathrm{qm}},x_{\mathrm{mm}}) + E_{\mathrm{mm}}(x_{\mathrm{mm}}) \qquad (1)$$

with E_{qm} depending on the coordinates of the QM atoms x_{qm} and E_{mm} depending on the coordinates of the MM atoms x_{mm}. The term $E_{\mathrm{qm/mm}}$ includes all energy contributions coupling the QM and the MM parts:

$$E_{qm/mm} = E_Q + E_{vdW} + E_{FF} \quad (2)$$

that is, the electrostatic QM/MM interaction E_Q, the van der Waals interaction E_{vdW}, and the force field terms E_{FF} at the boundary. The latter stem from covalent bonds between QM and MM atoms. Within the electrostatic embedding scheme, the QM code calculates $E_{qm} + E_Q$, while the MM code computes the remaining terms $E_{mm} + E_{vdW} + E_{FF}$. To avoid overpolarization of the QM density at the boundary, we use a charge-shift scheme.[23] The charge of the MM atom of the link is redistributed onto the MM atoms bonded to it. Thus, the overall charge is conserved. To conserve also the dipole moment, pairs of point charges of opposite sign are added to these neighboring MM atoms.

During the microiterations, the electrostatic interaction E_Q is approximated by the point-charge interaction energy E_{Q-ESP} between the ESP charges on the QM atoms and the force-field charges on the MM atoms. We thus need to calculate the approximate energy

$$E_{ESP} = E_{qm} + E_{Q-ESP} + E_{vdW} + E_{FF} + E_{mm} \quad (3)$$

The QM energy E_{qm} is calculated at each macroiteration and remains constant during the microiterations. The second term E_{Q-ESP} takes into account any modifications of MM charges arising from the applied link-atom treatment (i.e., from the charge-shift scheme, see above). In the modular structure of ChemShell, it is computed by a separate dedicated routine, while the remaining three terms in eq 3 are again provided by the MM code.

We adopt the convention to use superscripts 0 and 1 to label quantities that refer to macroiterations and microiterations, respectively. The total QM/MM energy E^0 and its gradient $G^0 = \partial E^0/\partial x^0$ are evaluated only during the macroiterations (coordinates x^0), while the approximate energy E^1_{ESP} and the associate gradient are computed also during the microiterations (coordinates x^1). The energy E^1 and the gradient G^1 that are actually used in the microiterations need to coincide with E^0 and G^0, respectively, when the geometries are identical ($x^0 = x^1$). Their proper choice will be discussed below.

II.B. Microiterative Optimization Algorithm. As already mentioned in the Introduction, we adopt the L-BFGS optimization algorithm in HDLCs, both in the inner and outer regions, as it scales linearly with system size and shows comparable convergence to other second-order optimizers. A trust-radius approach (for details, see ref 9) is used to determine the step size. This allows the optimizer to reject steps.

The general flow of the microiterative geometry optimizer is outlined in Chart 1. All energies E correspond to total energies, including QM and MM contributions. A standard L-BFGS optimization, without microiterations, would consist of cycles between steps 1, 3−5, and 7 only.

Overall convergence requires converged macro- and microiterations. As full QM/MM calculations define the convergence of the macroiterations, the accuracy of standard optimization is retained even when ESP charges are used in the microiterations.

In Chart 1, the ESP charges are calculated in step 2, prior to the macroiterative step 5 that moves the QM atoms. They are then used in the subsequent microiterations at a QM geometry that is different from the one for which they have been determined. This approach, denoted as Q1, introduces some error, but requires only one QM calculation in each macroiteration. Alternatively, one could include a second QM calculation and ESP charge evaluation in analogy to steps 1 and 2 at step 6, in what we call scheme Q2. We tested both approaches and found Q1 to be superior. This comparison is presented in sections 3.2 and 3.3.

II.C. Energy and Gradient Corrections in the Microiterations. The crucial point for fast convergence is the matching of the energy surfaces of the macroiterations and the microiterations. In the former, the energy is calculated with a full QM treatment. In the microiterations, however, no QM calculation is done, and the energy is calculated by replacing the QM density with ESP charges. These provide sufficiently accurate electrostatic forces on MM atoms far from the QM part but do not account for the shielding experienced by MM atoms within the QM density. Following a previous proposal,[11] we use a corrected gradient during the microiterations, that is, in step 6a of Chart 1:

$$G^1 = \frac{\partial E^1}{\partial x_{outer}} = \frac{\partial E^1_{ESP}}{\partial x_{outer}} + G^0_{corr} \quad (4)$$

with

$$G^0_{corr} = \left.\frac{\partial E^0}{\partial x_{outer}}\right|_{x^0_{outer}} - \left.\frac{\partial E^0_{ESP}}{\partial x_{outer}}\right|_{x^0_{outer}} \quad (5)$$

where the correction G^0_{corr} is calculated using the geometry from the last full macroiterative QM calculation, that is, step 1. Hence, we only assume that the ESP charges can faithfully reproduce the *change* in the gradient contribution from the electrostatic QM/MM interaction during the microiterations. The derivatives in eqs 4 and 5 are restricted to the coordinates x_{outer} of the outer region since the coordinates of the inner region remain fixed.

To achieve optimal convergence behavior, the energy expression has to match the gradient expression; that is, the gradient has to be the derivative of the energy. The approximate energy E^1_{ESP} does not correspond to the gradient in eq 4 due to the appearance of G^0_{corr} and can thus not be used as a criterion for accepting or rejecting steps in step 6d of Chart 1 where the current energy is compared to that from the preceding microiteration (or from the preceding macroiteration in the case of the first microiteration). The energy expression for step 6d is

$$E^1 = E^1_{ESP} + E^0 - E^0_{ESP} + G^0_{corr}(x^1_{outer} - x^0_{outer}) \quad (6)$$

where x^1_{outer} are the current coordinates of the outer region and x^0_{outer} are the respective coordinates from the last macroiterative QM calculation. As can be verified easily, the gradient in eq 4 is the derivative of eq 6 with respect to x^1_{outer}. The expressions for the energy E^1 and the gradient G^1 ensure a smooth connection between microiterations and macroiterations: in the case of identical geometries ($x^0 =$

Chart 1. Program Flow of the Microiterative Optimization Algorithm Q1

Step 1: Perform a QM/MM calculation with a full QM calculation to obtain the QM/MM energy E^0 and the gradient G^0 at geometry x^0.

Step 2: Calculate ESP charges fitted on the fly to the actual electron density. Recalculate the approximate energy E^0_{ESP} using ESP charges instead of the QM density for the electrostatic QM/MM interaction. Calculate the gradient correction G^0_{corr} from Eq. (5).

Step 3: If this is not the first macroiteration, test the acceptance of the step. Reject it if E^0 has increased compared with E^0 of the previous macroiteration. In this case reset the coordinates of the whole system to those used in the previous macroiteration, and halve the step size.

Step 4: If the step is accepted, terminate if the convergence criteria are met.

Step 5: Make one L-BFGS step in the inner region using the gradient obtained in step 1.

Step 6: Enter the microiterations

Step 6.a: Make an L-BFGS step in the outer region using the gradient from Eq. (4).

Step 6.b: If the convergence criteria are met, exit the microiterative loop and continue at step 7.

Step 6.c: Calculate the energy (E^1_{ESP}) and the associated gradient using ESP charges.

Step 6.d: If the energy E^1 (see Eq. (6)) has increased, reject the step and reduce the step size.

Step 6.e: Continue at step 6.a.

Step 7: Continue with macroiterations at step 1.

x^1), eqs 6 and 4 recover E^0 and G^0, respectively. Close to convergence, the geometry will change only very slightly in the microiterations and the approximations through ESP charges become exact.

We generally find good convergence of the overall optimization scheme when using the above energy and gradient expressions during the microiterations. While the gradient correction has been proposed before,[11] we are not aware of any previous discussion of a matching energy expression, eq 6, which is crucial for proper convergence. The use of ESP charges in the microiterations will be justified further in section 3.2 through comparisons with optimizations employing frozen QM densities and full self-consistent field (SCF) calculations for the QM region.

The L-BFGS memory, that is, the number of steps used to accumulate Hessian information, is retained during the macroiterations. Whenever a new cycle of microiterations is started, however, the L-BFGS optimizer for the microiterations is reset. The trust radius from the last microiteration cycle is retained as a starting value. This turned out to lead to faster convergence than resetting the trust radius to a fixed value.

II.D. Convergence Criteria. The adiabatic microiterative scheme requires in principle a fully converged outer region in each macroiteration. As this is difficult to achieve in practice, we include the coordinate space of both the inner and outer region into the active space for the L-BFGS algorithm of the macroiterations. In this case, partial convergence of the microiterations is sufficient to reach overall convergence.[15] This can be achieved by using a tighter (smaller) convergence criterion for the microiterations than for the macroiterations. The larger the ratio of the convergence criteria of the macroiterations and the microiterations is, the more MM steps, and accordingly less QM steps, are required to achieve convergence. While this ratio is user-definable in our implementation, we adopt a value of 10 throughout this work.

The convergence criteria for standard optimizations and macroiterations are taken from Gaussian 94.[36] An optimization is considered converged when all of the following four criteria are fulfilled (all values in atomic units, au). (i) The maximum gradient component has to be lower than $c = 0.00\,045$; this value can be specified by the user; (ii) $^2/_3c$ is used as a criterion for the root-mean-square (RMS) of the

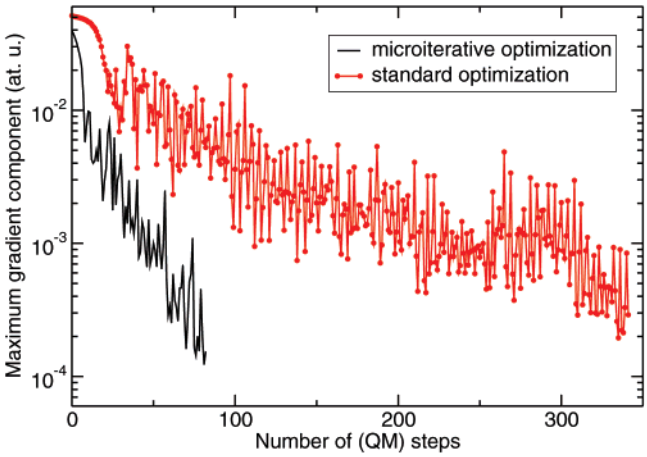

Figure 2. Maximum gradient component (one of the four convergence criteria) during microiterative optimization (only macroiterations shown) and standard optimization. This example refers to PHBH; the microiterative run was done with the parameters of the first line in Table 3.

gradient, (iii) $4c$ for the maximum step size, and (iv) $^8/_3c$ for the RMS of the step vector.

The convergence of the proposed scheme is illustrated in Figure 2. One of the four convergence criteria is plotted against the number of QM calculations. The microiterative optimization converges significantly faster, that is, requires fewer QM steps, than the standard optimization.

III. Results and Discussion

III.A. Scaling Behavior of the Optimizer. To show that the CPU time of an optimization is roughly independent of the MM system size, we optimized the geometry of water clusters in spheres of increasing size. Three water molecules in the center of the sphere served as the QM part and inner region. They were described by the semiempirical AM1 Hamiltonian.[37]

The MM water molecules were represented by the TIP3P[38] model, which constrains all internal coordinates. Only the relative orientations were optimized, leaving six degrees of freedom per water molecule. All 4404 water molecules with at least one atom within 25 Å of the center were cut out of equilibrated bulk water. Different active regions were defined, with radii of 5, 10, 15, and 20 Å. All atoms within the active region were optimized, while those outside were kept frozen and thus provided some kind of boundary potential. The microiterative optimizations were done according to Chart 1 using the standard Q1 approach.

Table 1 shows that the overall effort, measured by the number of QM steps in a microiterative optimization, changes with system size, but in a non-monotonic way, for a given size of the QM region. This is a well-known behavior of optimization algorithms in complex systems, which has often been encountered in the present work: unproductive search directions are tried and rejected in one case, while they may be directly avoided in the other case. In Table 1, the system with 9588 degrees of freedom apparently was especially easy to optimize. It needed less steps than the next smaller system (4644 degrees of freedom) in the microiterative scheme as well as in standard optimization.

Table 1. Optimization of Spherical TIP3P Water Clusters with a Radius of 25 Å (See Text)[a]

active region	DOF	microit. QM	microit. MM	std. QM	ΔE
5 Å	1662	62	2292	259	+23
10 Å	4644	62	2621	517	−57
15 Å	9588	56	3328	471	−19
20 Å	17 424	90	3640	801	−72

[a] The radius of the active region is given in the first column. DOF is the number of degrees of freedom being optimized. The following columns give the number of QM and MM calculations in microiterative optimization and the number of QM calculations in a standard QM/MM optimization. ΔE denotes the energy difference between the result of the microiterative and the standard optimizations in kilojoules per mole (positive if the former leads to a higher energy, and negative otherwise).

Generally speaking, optimizations of the kind considered currently only reach local minima. Since many close-lying minima are present in complex systems, different optimizers take different paths and may converge to different local minima. This happened in all four cases in Table 1. When standard and microiterative optimizations converge to the same minimum, the energy from the latter is generally slightly lower, because of the tighter convergence criteria used for the outer region of the system.

A noticeable difference between the systems with active regions of different size is the number of microiterations preceding the first macroiteration: 146, 654, 627, and 1031 initial microiterations are needed for the four systems in Table 1, respectively. These steps cover the major part of the optimization of the outer region. The following macro- and microiterations take more or less the same number of steps for all active regions considered.

We have also tested the systems with just one water molecule in the QM part and inner region. A total of 85, 48, 28, and 61 QM calculations are required to converge the different active regions, respectively. Thus, in this case, the smallest system needed the most steps to converge, again a result of the non-monotonic behavior discussed above.

The number of QM calculations required in a standard QM/MM optimization is shown in Table 1 as a reference. It is obvious that this number is greatly reduced in the microiterative optimizations, by a factor of 4−9. This indicates the speedup for QM/MM optimizations with high-level QM methods (ab initio or DFT) where the computational effort is dominated by the QM calculation. The speedup is somewhat smaller for semiempirical QM/MM optimizations since the computation time for the MM part is not completely negligible in this case.

III.B. ESP Charges versus Frozen Density and Full SCF Calculations. To check the approximate treatment of electrostatics in the microiterations, we performed test calculations on the example introduced in the previous section (a spherical TIP3P water cluster with a 5 Å active region; see first line of Table 1). The exact evaluation of the electrostatic QM/MM interaction energy would employ full SCF calculations. Successive approximations involve the use of frozen QM densities and ESP charges; the latter are consistent with the frozen QM densities in scheme Q2,

Table 2. Comparison of Different Methods for Evaluating the Electrostatic QM/MM Interaction Energy in the Microiterations (See Text) during Optimization of a Spherical TIP3P Water Cluster (See First Line of Table 1)[a]

electrostatics	macroit.	microit.	ΔE
SCF calculations Q2	44 (88)	1443	+23
frozen density Q2	47 (94)	1359	+16
ESP charges Q2	80 (160)	2528	+47
ESP charges Q1	62 (62)	2292	+23

[a] The second column lists the number of macroiterations until convergence (in parentheses: total number of QM calculations during macroiterations). The third column gives the number of microiterations (QM calculations in the SCF case). ΔE denotes the energy difference between the result of the microiterative and the standard optimization in kilojoules per mole.

whereas those from scheme Q1 are determined at a slightly different geometry and are therefore even more approximate (see section 2.2).

Table 2 shows that full SCF calculations during the microiterations lead to the fastest convergence (44 macroiterations), closely followed by the calculations with the frozen QM density (47 macroiterations). Representing the QM density by ESP charges in scheme Q2 slows down convergence (80 macroiterations), which—surprisingly and probably fortuitously—becomes faster again with scheme Q1 (62 macroiterations). The final energies are comparable in all four cases and virtually identical for the best and the most approximate treatment applied (see Table 2, first and last lines). The present results confirm that the overall convergence behavior of our microiterative optimizer is not very sensitive to the treatment of electrostatics in the microiterations. Under these circumstances, the use of ESP charges is clearly justified considering the huge savings in computation time compared with the use of frozen QM densities (or even full SCF calculations).

III.C. Charge-Fitting Procedures. The approximation of the density by point charges can be done in various ways. We only consider charges derived from the electrostatic potential. One can fit the potential at different positions in space, for example, on a shell around the QM part or at the positions of the nearest MM atoms. The fitting itself can be done without constraints, resulting in ESP charges, or with restrictions keeping the fitted charges at low values, yielding so-called restrained electrostatic potential (RESP) charges.[39]

We tested the effect of different charge-fitting schemes on the optimization process using the enzyme *p*-hydroxybenzoate hydroxylase (PHBH). The QM part contained 49 atoms: *p*-hydroxybenzoate and the isoalloxazine part of the hydroperoxyflavin cofactor of PHBH. Polar groups, such as carboxylates, and atoms buried within the QM part, for which ESP charges are not well-defined, pose significant challenges to the charge fitting. Thus, this example should provide a stringent test for charge-fitting schemes. Again, AM1 was used as the QM method, while the GROMOS force field was chosen to describe the MM environment. Details of the system setup are available elsewhere.[40,41] We optimized an active region with 8847 degrees of freedom, composed of the QM part and all residues with at least one atom within 15 Å of the substrate. The optimizations were started from a MD snapshot taken from umbrella sampling simulations equilibrated at $T = 300$ K.[42]

Table 3. Comparison of Different Charge-Fitting Procedures in the Microiterative Optimization of PHBH (See Text)[a]

	# pt.		QM	MM	ΔE (kJ mol^{-1})
MM atoms	98	ESP	88	4851	−22.9
shell	98	ESP	100	4373	−23.1
MM atoms	98	RESP	91	3854	−24.2
shell	98	RESP	94	4354	−23.9
shell	400	ESP	104	4404	−23.0
standard optimization			363	0	0

[a] The first three columns specify the location of the fitting points (MM atoms vs points on a shell), their number ($2N_{qm} = 98$), and the type of fitting (ESP vs RESP). The next two columns give the number of required QM and MM calculations. The last column lists the final energy relative to that obtained from a standard optimization. All microiterative optimizations are significantly better converged than the standard optimization due to the tight convergence criterion in the outer region.

This system exhibits one covalent link between the QM part and the MM part. The MM atom of the link and the MM atoms bonded to it were included in the inner region. With this choice, all atoms affected by the charge-shift scheme[23] are treated by the macroiterations.

Table 3 summarizes the results of optimizations with different charge-fitting schemes applied. Overall, the efficiency of the optimizer depends only slightly on the charge-fitting scheme. For N_{qm} (number of QM atoms) charges to be fitted, $2N_{qm}$ fitting points are sufficient (i.e., 98 in this case). Using the positions of the MM atoms with the smallest distance to their nearest QM atom seems slightly more efficient (88 QM steps) than using points on a shell of 1.5 times the van der Waals distance of the QM atoms (100 steps). ESP and RESP charges lead to a similar convergence behavior (100 vs 94 and 88 vs 91 steps, respectively), but a slightly lower energy is obtained in both cases when using RESP charges, and fewer MM steps are required. Finally, using 400 points to fit the potential rather than 98 has little effect on the convergence (104 vs 100 steps).

In our implementation, the choice of the fitting method is left to the user. However, for an arbitrary system, a shell of fitting points may be better than the MM positions, as the latter may be distributed unevenly, preventing an accurate fit.

We have also tested on this system the alternative scheme Q2 introduced in section 2.2. With all other parameters set as in the first line of Table 3, 178 QM calculations (i.e., 89 macroiterations) were needed to reach convergence for Q2, compared to 88 when using Q1, and the same optimized energy of −22.9 kJ mol^{-1} was obtained (see Table 3). These results again show the validity of the approximate scheme Q1.

Finally, we note that a standard optimization requires typically almost 4 times more QM calculations than the microiterative approach (see Table 3) and yields a minimum that is 23−24 kJ mol^{-1} higher in energy. This is caused by the relatively loose convergence criteria applied to the whole system: 10 times tighter criteria are used for the outer region in the microiterative optimizations. If 3 times tighter criteria

(i.e., $c = 0.00\ 015$ au) are used in the standard optimization, 557 iterations are needed (i.e., 5−6 times more than with the microiterative scheme), but the resulting energy of −23.8 kJ mol^{-1} is close to the microiterative results. This indicates that the standard optimization does reach the same local minimum as the microiterative scheme, when the convergence criteria are sufficiently tightened.

III.D. Energy Minimization in Enzymes. We chose the enzyme fluorinase to compare actual CPU times rather than the number of QM steps. The DFT calculations used for the QM system take substantially more computation time than the force-field calculations. The system was set up in analogy to our previous work.[43] The fluorinase catalyzes the S_N2 attack of fluoride on *S*-adenosyl-L-methionine (SAM). The QM part consisted of the 39 atoms of fluoride and the 5′-(*S*,*S*-dimethylsulfonio)-5′-deoxyadenosine part of SAM. The CHARMM22[35] force field was used for the MM calculations with the DL_POLY force-field engine included in ChemShell. A group-based cutoff of 20 au was applied for the MM interactions, while no cutoff was used for the QM/MM interactions. The QM calculations were done with TURBOMOLE interfaced to ChemShell at the DFT (BP86) level with the TZVP basis[44] augmented by diffuse functions (1005 basis functions in total). They were message-passing-interface-parallelized over four AMD Opteron 848 processors with a clock rate of 2.2 GHz. A trimer unit of the enzyme with 17 596 atoms, including 4272 water molecules, was used, of which only one active site was solvated. All residues or water molecules within 8 Å of SAM or F^- were included in the optimization. Water was again described by the TIP3P model, with all three internal degrees of freedom constrained. This resulted in 3960 degrees of freedom to be optimized. Fluoride and the whole SAM group (153 degrees of freedom) served as the inner region in the microiterative optimizations.

In contrast to all cases discussed up to now, we started from a structure already optimized under a constraint that keeps the system close to the transition state, rather than from an MD snapshot. This is a severe test for the microiterative scheme, as the structural changes occur predominantly within the QM region rather than uniformly throughout the system.

Nevertheless, the microiterative scheme required only 51% of the time for a standard optimization which took 139 CPU hours. A total of 53 QM steps were required until convergence was reached, rather than 119 steps in the standard calculation. The fact that relatively less CPU time (49%) is saved compared to the change in the number of QM calculations (55%) is due to two factors: (1) The number of MM energy evaluations increased from 119 to 4610. Thus, the MM calculations accounted for less than 0.1% of the overall time in the standard optimization but for 2% in the microiterative optimization. (2) Each QM step in the microiterative scheme takes some 2% longer than in the standard optimization, as the ESP charges have to be evaluated in addition. The tighter convergence criterion for the MM part resulted in a tiny energy lowering by 0.04 kJ mol^{-1}. We did not attempt to converge an MD snapshot due to the potentially high computational cost of a standard optimization.

III.E. Polarizable Force Fields for Surface Catalysis. The applicability of the microiterative optimizer to solid-state embedding was tested by optimizing the structure of a defect at the MgO (001) surface. We modeled the MgO surface as a hemispherical cluster in analogy to previous work.[3,45] GAMESS-UK[31,32] was used as the QM code and GULP[34] with a shell model[46] (i.e., a polarizable force field) as the MM code. The polarization of the MM part adds another layer of iterations to the calculations. Whenever a full QM/MM energy is evaluated, the QM and MM calculations are iterated until the mutual polarization of the QM and MM parts has converged.

This example is intended as a proof of principle, showing that the microiterative optimization works and converges well for a shell model. The system needed 30 QM steps and 459 MM steps in the microiterative optimization, compared to 34 optimization steps in the standard optimization, and it reached the same minimum, the final energy being slightly lower (by 3.5 kJ mol^{-1}) in the microiterative case. With only four QM calculations saved, the benefits of the microiterative scheme are less pronounced in this example, presumably because the optimization started from a well-ordered rather than a disordered structure (such as an MD snapshot).

IV. Conclusion

We have described an optimization scheme for the QM/MM approach with electrostatic embedding. In each QM optimization step (macroiteration), the MM atoms in the outer region are fully relaxed in microiterations, during which the QM density is approximated by ESP charges to allow for an efficient computation of the electrostatic QM/MM interactions. Suitable energy and gradient expressions for the microiterations ensure an excellent convergence behavior of the overall algorithm. The inner region treated by macroiterations must contain all QM atoms but may be extended to include additional MM atoms. The algorithm retains the accuracy of a standard QM/MM optimization scheme, while saving typically a factor of 2−10 in CPU time for the examples discussed. It has been implemented in the modular QM/MM program package ChemShell.

We have considered two variants, Q1 and Q2, which evaluate the ESP charges at different stages of the algorithm. In Q1, they are calculated prior to each macroiterative step and then used at a geometry of the inner region that is slightly different, whereas they are redetermined at this geometry in Q2. Variant Q1 is preferred over Q2 because it requires only half the computational effort and performs well in the test cases studied. We have also compared several charge-fitting procedures (such as ESP vs RESP) and found that the convergence behavior is largely independent of these choices.

The force-field calculations in the microiterations are fast compared to full QM/MM calculations, and they relax the MM part efficiently. The overall computational cost of such an optimization is thus essentially independent of the MM system size, for a given size of the QM part. The microiterative optimization algorithm is particularly beneficial for systems in which the MM part as well as the QM part are far from the minimum, as is often the case for MD snapshots as starting points. In systems where the MM part is almost

relaxed but the QM part is not, the microiterative scheme will show only a minor, if any, advantage over the conventional optimization.

Using the ChemShell implementation, we have demonstrated the applicability of the microiterative optimization scheme for a number of different systems: a water cluster, fully or partially solvated proteins, and a surface defect. We believe that the proposed microiterative optimizer will increase the range of applicability of QM/MM methods because it enables large-scale QM/MM optimizations with electrostatic embedding.

Acknowledgment. S.T. thanks Salomon Billeter for helpful discussions. This work was supported by the Deutsche Forschungsgemeinschaft, Grant SFB 663/C4.

References

(1) Brändle, M.; Sauer, J. *J. Am. Chem. Soc.* **1998**, *120*, 1556.

(2) Sierka, M.; Sauer, J. *Faraday Discuss.* **1997**, *106*, 41.

(3) Sushko, P. V.; Shluger, A. L.; Catlow, C. R. A. *Surf. Sci.* **2000**, *450*, 153.

(4) Åqvist, J.; Warshel, A. *Chem. Rev.* **1993**, *93*, 2523.

(5) Gao, J.; Ma, S.; Major, D. T.; Nam, K.; Pu, J.; Truhlar, D. G. *Chem. Rev.* **2006**, *106*, 3188.

(6) Senn, H. M.; Thiel, W. In *Atomistic Approaches in Modern Biology − From Quantum Chemistry to Molecular Simulations*; Reiher, M., Ed.; Springer: Berlin, Germany, 2007; Topics in Current Chemistry, Vol. 268, pp 173−290.

(7) Maseras, F.; Morokuma, K. *J. Comput. Chem.* **1995**, *16*, 1170.

(8) Turner, A. J.; Moliner, V.; Williams, I. H. *Phys. Chem. Chem. Phys.* **1999**, *1*, 1323.

(9) Billeter, S. R.; Turner, A. J.; Thiel, W. *Phys. Chem. Chem. Phys.* **2000**, *2*, 2177.

(10) Zhang, Y.; Liu, H.; Yang, W. *J. Chem. Phys.* **2000**, *112*, 3483.

(11) Murphy, R. B.; Philipp, D. M.; Friesner, R. A. *J. Comput. Chem.* **2000**, *21*, 1442.

(12) Hall, R. J.; Hindle, S. A.; Burton, N. A.; Hillier, I. H. *J. Comput. Chem.* **2000**, *21*, 1433.

(13) Vreven, T.; Morokuma, K.; Farkas, Ö.; Schlegel, H. B.; Frisch, M. J. *J. Comput. Chem.* **2003**, *24*, 760.

(14) Prat-Resina, X.; Bofill, J. M.; González-Lafont, A.; Lluch, J. M. *Int. J. Quantum Chem.* **2004**, *98*, 367.

(15) Vreven, T.; Frisch, M. J.; Kudin, K. N.; Schlegel, H. B.; Morokuma, K. *Mol. Phys.* **2006**, *104*, 701.

(16) Banerjee, A.; Adams, N.; Simons, J.; Shepard, R. *J. Phys. Chem.* **1985**, *89*, 52.

(17) Broyden, C. G. *J. Inst. Math. Appl.* **1970**, *6*, 76.

(18) Fletcher, R. *Comput. J.* **1970**, *13*, 317.

(19) Goldfarb, D. *Math. Comp.* **1970**, *24*, 23.

(20) Shanno, D. F. *Math. Comp.* **1970**, *24*, 647.

(21) Liu, D. C.; Nocedal, J. *Math. Prog. B* **1989**, *45*, 503.

(22) Nocedal, J. *Math. Comp.* **1980**, *35*, 773.

(23) Sherwood, P.; de Vries, A. H.; Guest, M. F.; Schreckenbach, G.; Catlow, C. R. A.; French, S. A.; Sokol, A. A.; Bromley, S. T.; Thiel, W.; Turner, A. J.; Billeter, S.; Terstegen, F.; Thiel, S.; Kendrick, J.; Rogers, S. C.; Casci, J.; Watson, M.; King, F.; Karlsen, E.; Sjøvoll, M.; Fahmi, A.; Schäfer, A.; Lennartz, C. *THEOCHEM* **2003**, *632*, 1.

(24) ChemShell, a Computational Chemistry Shell. http://www.chemshell.org (accessed Nov 28, 2006).

(25) Thiel, W. *MNDO99*, version 6.1; Max-Planck-Institut für Kohlenforschung: Mülheim an der Ruhr, Germany, 2004.

(26) Ahlrichs, R.; Bär, M.; Häser, M.; Horn, H.; Kölmel, C. *Chem. Phys. Lett.* **1989**, *162*, 165.

(27) Treutler, O.; Ahlrichs, R. *J. Chem. Phys.* **1995**, *102*, 346.

(28) Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. *Chem. Phys. Lett.* **1995**, *242*, 652.

(29) Eichkorn, K.; Weigend, F.; Treutler, O.; Ahlrichs, R. *Theor. Chem. Acc.* **1997**, *97*, 119.

(30) Sierka, M.; Hogekamp, A.; Ahlrichs, R. *J. Chem. Phys.* **2003**, 118, 9136.

(31) GAMESS-UK is a package of ab initio programs. http://www.cfs.dl.ac.uk/gamess-uk/index.shtml (accessed Nov 28, 2006).

(32) Guest, M. F.; Bush, I. J.; van Dam, H. J. J.; Sherwood, P.; Thomas, J. M. H.; van Lenthe, J. H.; Havenith, R. W. A; Kendrick, J. *Mol. Phys.* **2005**, *103*, 719.

(33) van Gunsteren, W. F.; Billeter, S. R.; Eising, A. A.; Hünenberger, P. H.; Krüger, P.; Mark, A. E.; Scott, W. R. P.; Tironi, I. G. *Biomolecular Simulation: The GROMOS96 Manual and User Guide*; vdf and BIOMOS b.v.: Zürich, Switzerland, 1996.

(34) Gale, J. D. *J. Chem. Soc., Faraday Trans.* **1997**, *93*, 629.

(35) MacKerell, A. D., Jr.; Bashford, D.; Bellott, R. L.; 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.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. *J. Phys. Chem. B* **1998**, *102*, 3586.

(36) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Gill, P. M. W.; Johnson, B. G.; Robb, M. A.; Cheeseman, J. R.; Keith, T.; Petersson, G. A.; Montgomery, J. A.; Raghavachari, K.; Al-Laham, M. A.; Zakrzewski, V. G.; Ortiz, J. V.; Foresman, J. B.; Cioslowski, J.; Stefanov, B. B.; Nanayakkara, A.; Challacombe, M.; Peng, C. Y.; Ayala, P. Y.; Chen, W.; Wong, M. W.; Andres, J. L.; Replogle, E. S.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Binkley, J. S.; Defrees, D. J.; Baker, J.; Stewart, J. P.; Head-Gordon, M.; Gonzalez, C.; Pople, J. A. *Gaussian 94*, rev. c.3; Gaussian, Inc.: Pittsburg, PA, 1995.

(37) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902.

(38) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926.

(39) Bayly, C. I.; Cieplak, P.; Cornell, W.; Kollman, P. A. *J. Phys. Chem.* **1993**, *97*, 10269.

(40) Senn, H. M.; Thiel, S.; Thiel, W. *J. Chem. Theory Comput.* **2005**, *1*, 494.

(41) Kästner, J.; Senn, H. M.; Thiel, S.; Otte, N.; Thiel, W. *J. Chem. Theory Comput.* **2006**, *2*, 452.

(42) Kästner, J.; Thiel, W. *J. Chem. Phys.* **2005**, *123*, 144104.

(43) Senn, H. M.; O'Hagan, D.; Thiel, W. *J. Am. Chem. Soc.* **2005**, 127, 13643.

(44) Schäfer, A.; Huber, C.; Ahlrichs, R. *J. Chem. Phys.* **1994**, 100, 5829.

(45) Sokol, A. A.; Bromley, S. T.; French, S. A.; Catlow, C. R. A.; Sherwood, P. *Int. J. Quantum Chem.* **2004**, *99*, 695.

(46) Dick, B. G.; Overhauser, A. W. *Phys. Rev.* **1958**, *112*, 90.

CT600346P