Geometry Optimization with QM/MM, ONIOM, and Other Combined Methods. I. Microiterations and Constraints

THOM VREVEN,[1,2] KEIJI MOROKUMA,[1] ÖDÖN FARKAS,[3,4] H. BERNHARD SCHLEGEL,[4]
MICHAEL J. FRISCH[2]

[1]Cherry L. Emerson Center for Scientific Computation, and Department of Chemistry,
Emory University, Atlanta, Georgia 30322
[2]Gaussian, Inc., 140 Washington Ave., North Haven, Connecticut 06473
[3]Department of Organic Chemistry, Eötvös Loránd University, 112 P.O. Box 32,
H-1518 Budapest, Hungary
[4]Department of Chemistry, Wayne State University, Detroit, Michigan 48202

Received 16 October 2001; Accepted 25 June 2002

Abstract: Hybrid energy methods such as QM/MM and ONIOM, that combine different levels of theory into one calculation, have been very successful in describing large systems. Geometry optimization methods can take advantage of the partitioning of these calculations into a region treated at a quantum mechanical (QM) level of theory and the larger, remaining region treated by an inexpensive method such as molecular mechanics (MM). A series of microiterations can be employed to fully optimize the MM region for each optimization step in the QM region. Cartesian coordinates are used for the MM region and are chosen so that the internal coordinates of the QM region remain constant during the microiterations. The coordinates of the MM region are augmented to permit rigid body translation and rotation of the QM region. This is essential if any atoms in the MM region are constrained, but it also improves the efficiency of unconstrained optimizations. Because of the microiterations, special care is needed for the optimization step in the QM region so that the system remains in the same local valley during the course of the optimization. The optimization methodology with microiterations, constraints, and step-size control are illustrated by calculations on bacteriorhodopsin and other systems.

© 2003 Wiley Periodicals, Inc. J Comput Chem 24: 760–769, 2003

Key words: geometry optimization; QM/MM; ONIOM; microiterations; hybrid methods; combined methods

Introduction

Hybrid techniques that combine two or more computational methods in one calculation allow the accurate exploration of the chemistry of very large systems (for reviews, see refs. 1–3). Examples of such techniques are the QM/MM methods[3–7] (for some recent studies see refs. 8–12), which combine a quantum mechanical (QM) method with a molecular mechanics (MM) method, and the more general ONIOM scheme,[13–19] which can combine any number of molecular orbital methods, as well as molecular mechanics methods. The region of the system where the chemical process takes place, for example bond breaking and formation, is treated with an appropriately accurate method, while the remainder of the system is treated at the lower level. QM/MM schemes in particular have been successful for the study of enzyme reactions, treating the active site by a high level method, often DFT, and the protein environment by molecular mechanics.

In a two layer ONIOM calculation, the total energy of the system is obtained from three independent calculations:

$$E^{\mathrm{ONIOM(QM:MM)}} = E^{\mathrm{QM}}_{\mathrm{model}} + E^{\mathrm{MM}}_{\mathrm{real}} - E^{\mathrm{MM}}_{\mathrm{model}} = E^{\mathrm{high}}_{\mathrm{model}} + E^{\mathrm{low}}_{\mathrm{real}} - E^{\mathrm{low}}_{\mathrm{model}} \tag{1}$$

As described in a number of previous articles,[13–19] the *real* system contains all the atoms, and is calculated only at the MM level. The

Correspondence to: H. B. Schlegel

Contract/grant sponsor: Gaussian, Inc.

Contract/grant sponsor: National Science Foundation; contract/grant numbers: CHE-9627775 (K.M.) and CHE-9874005 (H.B.S.)

Contract/grant sponsor: Hungarian Research Foundation; contract/grant number: OTKA D-29446

model system contains the part of the system that is treated at the QM level, along with the link atoms that are used to cap dangling bonds resulting from cutting covalent bonds between the QM and the MM regions (see Fig. 1). To evaluate the ONIOM energy, both QM and MM calculations need to be carried out for the model system. Because the positions of the link atoms are defined in terms of the atoms in the real system, the potential energy surface (PES), and therefore geometry optimization, is well defined.[13–19] The ONIOM gradient is obtained from

$$\frac{\partial E^{\text{ONIOM}}}{\partial q} = \frac{\partial E^{\text{high}}_{\text{model}}}{\partial q} \cdot \mathbf{J} + \frac{\partial E^{\text{low}}_{\text{real}}}{\partial q} - \frac{\partial E^{\text{low}}_{\text{model}}}{\partial q} \cdot \mathbf{J} \qquad (2)$$

where **J** is the Jacobian, which is needed to convert the coordinate system for the model system to the coordinate system for the real system.[13–19] The Hessian and other properties can be expressed in a similar fashion.

In this article we discuss some aspects of geometry optimization techniques that are specifically designed for use with hybrid methods. Initial applications of the QM/MM approach carried out the geometry optimization of the MM and QM regions in separate programs. Because the MM and QM regions are coupled, the optimization must cycle between the regions until both are converged. Because the MM region is often large but inexpensive to calculate, while the QM region is small but expensive, the separate optimization of the MM and QM regions can be more efficient than applying a regular optimization method to the entire system. However, completely separating optimizers for the QM and the MM regions (i.e., no exchange of information between the two optimizers) causes several immediate problems. First, a displacement in the QM region might bring the combined system to a different (not necessarily lower) region of the PES. Second, when geometrical constraints are applied to the MM region, computational efficiency issues arise that can be dealt with by some integration of the QM and MM optimizers. We address these topics in the present article. In order to do so we will first introduce and illustrate the "standard" microiteration scheme as implemented in a number of QM/MM packages. This will be followed by a detailed discussion of the solutions to the aforementioned problems, including a number of illustrative examples. Although the methods we present are generally applicable to any hybrid method, for clarity we will discuss our work within the 2-layer ONIOM(QM:MM) framework.

Method

The goal of an efficient geometry optimization scheme is to find the optimized geometry with the least expenditure of computational effort. Usually, the QM calculation on the model system is much more expensive than the MM calculation on the real system. A heuristic approach is to minimize the number of expensive QM energy and gradient calculations, even if this results in an increased number of MM calculations on the real system. Due to the large difference in cost for the two calculations, the overall computational expense will often be lower. In practice, this goal is accomplished through a microiteration scheme for optimizing the MM region.

The microiteration can take advantage of the form of the ONIOM energy and gradient [eqs. (1) and (2)]. If the coordinates of the atoms that determine the energies of the model system are frozen, $E^{\text{high}}_{\text{model}}$ and $E^{\text{low}}_{\text{model}}$ are constant, and the ONIOM energy becomes a function of $E^{\text{low}}_{\text{real}}$ only, which can be determined by a low level calculation on the real system. Thus we can fully minimize the energy with respect to the coordinates in the MM region using only real system MM calculations. This is followed by one geometry optimization step that involves only the coordinates that determine the model system, using the forces obtained with eq. (2). Because $E^{\text{low}}_{\text{real}}$ depends on the coordinates of the QM region, the MM forces may no longer be converged after the QM step. The process of minimizing the energy for the low level by microiteration, followed by an optimization step for model system, is repeated until the ONIOM energy has converged to a minimum with respect to all the coordinates.

Because the MM optimization can involve many coordinates but the energy and gradient computations are inexpensive, the memory usage, robustness, and generality of the optimization method are more important than its speed. While there are promising developments in using internal coordinates for very large systems,[20–27] geometry optimization schemes in Cartesian coordinates may still be the best choice for the MM region at this time. On the other hand, for the high level calculations on the model system, the goal is to reach the minimum in the fewest steps. Because the microiterations ensure that the energy is a minimum with respect to the remaining coordinates, relatively few coordinates are needed to treat the QM region. For this phase of the optimization, we can use more sophisticated techniques such as full matrix quasi-Newton methods[28,29] in redundant internal coordinates.[30–35]

Initially such an approach was widely used out of necessity, because the MM and QM calculations were often carried out in separate programs. Nevertheless, the computational advantages can be large. First, the number of QM energy and gradient calculations can be reduced dramatically. Second, because the majority of the system is treated with Cartesian coordinates, the expensive transformations between Cartesian and redundant internal coordinate systems can be avoided for the MM region. However, some special considerations arise for optimizations involving a series of microiterations, and we address these in the next sections.

Microiterations without Constraints

Let the coordinates $\mathbf{q}$ be divided into two sets, $\mathbf{q}^{\text{m}}$ for the QM region and $\mathbf{q}^{\text{l}}$ for the MM region. The $\mathbf{q}^{\text{l}}$ coordinates are chosen so that they do not involve any coordinates that define the energy of the model system, that is, $\partial E^{\text{high}}_{\text{model}}/\partial \mathbf{q}^{\text{l}} = \mathbf{0}$ and $\partial E^{\text{low}}_{\text{model}}/\partial \mathbf{q}^{\text{l}} = \mathbf{0}$. All the remaining coordinates are included in $\mathbf{q}^{\text{m}}$. This separation ensures that the microiteration procedure handles the division between the model and the real systems correctly. Note that the atoms at the QM/MM boundary, which are replaced by link atoms in the model system (see Fig. 1), must be treated in $\mathbf{q}^{\text{m}}$. Under some circumstances, it may be desirable to include coordinates in $\mathbf{q}^{\text{m}}$ for additional atoms. Thus, $\mathbf{q}^{\text{m}}$ may be somewhat larger than the minimal set of coordinates needed to define the model system.

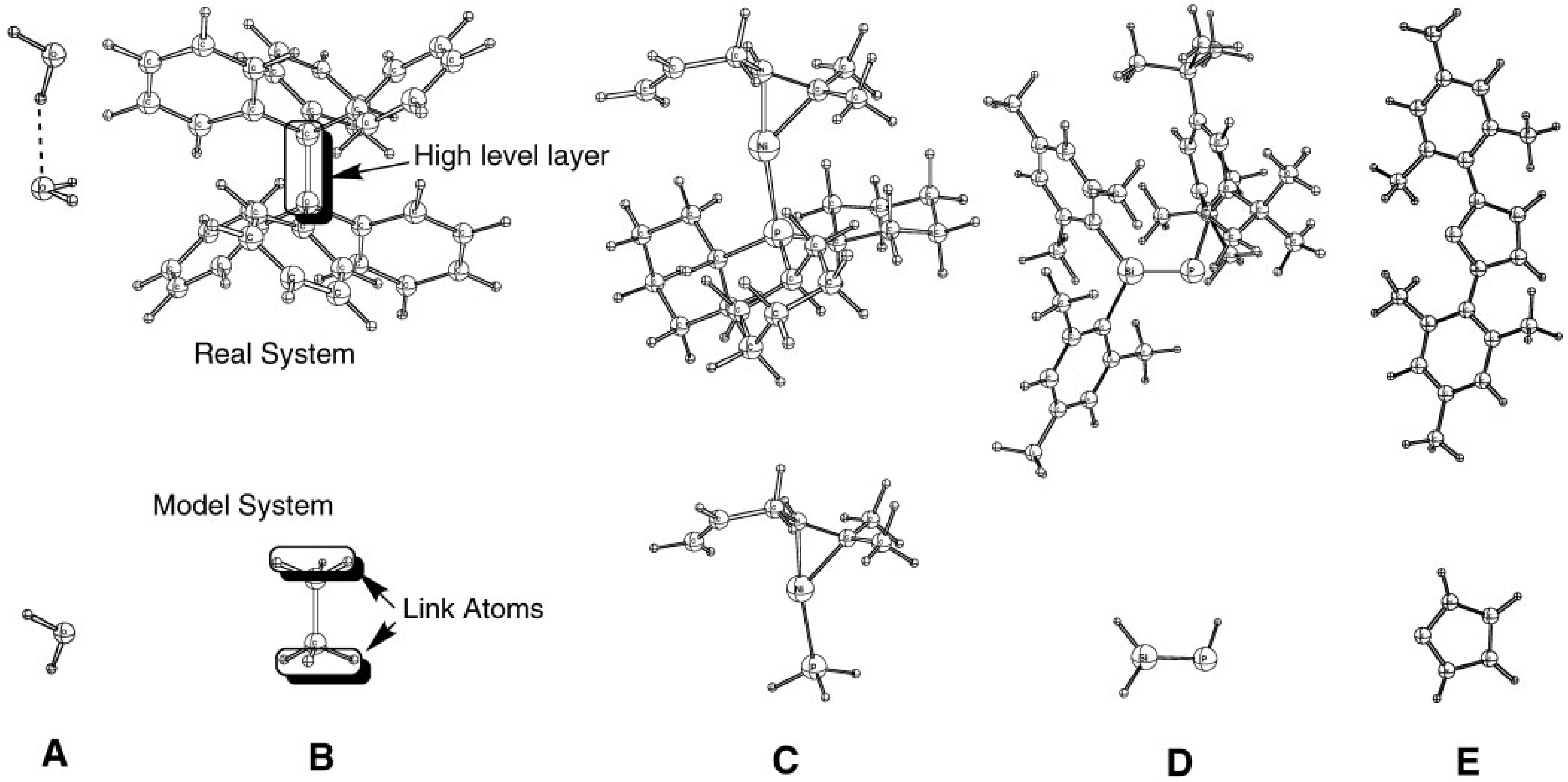

Figure 1. Small systems used in Table 1 to assess the microiteration scheme: (a) water dimer, (b) hexaphenylethane, (c) $Ni(PCy_3)$ complexed with $[CH_2{=\!\!=}CHCH_2HN{=\!\!=}C(CH_3)_2]^+$, (d) a sterically hindered phosphasilene, and (e) a stabilized singlet carbene (real systems are shown in the top row, corresponding model systems in the bottom row).

During the i^{th} outer cycle of the optimization process, the microiterations are used to optimize the $\mathbf{q}^{\text{l}}$ coordinates while the $\mathbf{q}_i^{\text{m}}$ coordinates are frozen, leading to $\mathbf{q}_i^{\text{l}}$, a local minimum in $\mathbf{q}^{\text{l}}$. Conjugate gradient (CG) (see ref. 36 and other standard texts on optimization) and GDIIS[37–39] (geometry optimization using direct inversion in the iterative subspace) based methods in Cartesian coordinates are good choices for optimizing the $\mathbf{q}^{\text{l}}$ coordinates. These techniques have low memory usage and are robust, especially if combined with a good line search algorithm. Because gradients are usually evaluated in Cartesian coordinates, no transformations between coordinate systems are required.

In the CG method, the next point in an optimization is obtained by a linear search, $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{s}_{i+1}$, with the search direction given by

$$\mathbf{s}_{i+1} = -\mathbf{g}_{i+1} + \gamma_i \mathbf{s}_i,\ \gamma_i = (\mathbf{g}_{i+1} - \mathbf{g}_i)\cdot \mathbf{g}_i / \mathbf{g}_i \cdot \mathbf{g}_i \qquad (3)$$

where $\mathbf{g}_{i+1}$ is the gradient at the current point and $\mathbf{s}_0 = \mathbf{0}$. Note that this algorithm requires the storage of only three vectors. In our implementation, the length of the step in the search direction is chosen to be equal to the step length to the minimum in the previous linear search. A constrained quartic polynomial fitted to the energies and gradients is used for the line search, similar to that used in our regular gradient-based geometry optimization.[40]

Once two or more suitable structures have been obtained in the microiterations, a GDIIS step[37–39] can also be used to determine the next structure:

$$\mathbf{x}_{i+1} = \sum_{j=i-k}^{i} c_j(\mathbf{x}_j - \mathbf{H}^{-1}\mathbf{g}_j) \qquad (4)$$

where $\mathbf{H}$ is an approximate Hessian and the coefficients c_j are chosen to minimize $(\Sigma_j c_j(\mathbf{H}^{-1}\mathbf{g}_j))^2$ subject to $\Sigma_j c_j = 1$. For Cartesian coordinates, it is sufficient to use $\mathbf{H} = b\mathbf{I}$; we have found $b = 0.7$ a.u. to be satisfactory. For internal coordinates, an empirical estimate of a diagonal Hessian may be beneficial.[41,42] The GDIIS method requires the storage of only $2k + 2$ vectors if k points are retained and a diagonal Hessian is used. In practice we retain no more than 50 points. We note that the original implementation of the GDIIS[37] method tends to converge to any nearby critical point, and some extra care[38,39] is required to ensure that it converges to a local minimum.

Both the CG and GDIIS methods are suitable for very large systems, because they require only a limited number of vectors and scale as O(N) in memory and CPU time per step. With the recent advances towards efficient, O(N) scaling internal coordinate based optimization methods,[20–27] it may be feasible to use redundant internal coordinates instead, especially if they substantially reduce the number of steps in the microiterations. However, the relative time spent for the $\mathbf{q}^{\text{l}}$ optimization via microiteration is nearly always small, even with CG or GDIIS in Cartesian coordinates.

The microiteration approach has been implemented in a number of programs.[9,10,13,43] As will be described in the Results and Discussion section, we find that this method reduces the number of QM calculations by a factor of two when compared to full optimizations in redundant internal coordinates. However, the microiteration scheme with the coordinate separation as described in this section is directly applicable only for mechanical embedding and only when there are no constraints in the MM region. Extensions to this scheme are discussed below.

Microiterations with Electronic Embedding

The separation of coordinates into $\mathbf{q}^{\text{m}}$ and $\mathbf{q}^{\text{l}}$ sets is most effective for the *mechanical embedding* used in the ONIOM scheme (i.e., when the interactions between the two layers are included only via the MM calculations). With *electronic embedding,* the electrostatic interactions between the layers are included in the QM calculations by adding the Coulombic interactions between the QM electrons and nuclei with the partial charges in the MM region explicitly in the quantum mechanical energy expression. This is the case in most QM/MM implementations, allowing the QM wave function to be polarized and providing a more accurate description of the electrostatic interaction between the two layers. Although there are small differences in the electronic embedding formalism in the various QM/MM implementations, the coordinates cannot be rigorously separated because $\partial E_{\text{model}}^{\text{high}}/\partial \mathbf{q}^{\text{l}} \neq \mathbf{0}$ and $\partial E_{\text{model}}^{\text{low}}/\partial \mathbf{q}^{\text{l}} \neq \mathbf{0}$, and consequently the microiteration scheme as outlined above is no longer valid. To address this problem, several modified schemes have been proposed that allow microiteration optimizations to be used with these types of QM/MM methods.[9,10] Yang et al.[9] approximated the QM charge distribution by electrostatic potential (ESP) charges, which were then used unchanged to describe the Coulombic interaction between the two layers during the optimization of the MM region. After the microiterations, the QM region is fully optimized and the ESP charges are re-evaluated. This sequence is repeated until convergence. However, it is clear that in this scheme the PES used for the geometry optimization, which includes the interaction of the QM ESP charges with the MM partial charges for the electrostatic interaction between the layers, is not the same as that used for the energy calculation, which includes the interaction of the QM nuclei and electrons with the MM partial charges. Friesner et al.[10] presented a similar scheme, but used the ESP charges for the QM region only to describe the *perturbation* to the exact gradient during the microiterations. After these microiterations with essentially a frozen wave function, the optimization step in the QM region is taken, and the process is repeated. Because the perturbation goes to zero at convergence, the optimized structure is a true critical point on the surface for the energy calculation. However, in this approach the MM region is not fully optimized when the QM optimization step is taken, because formally the wave function has to be re-evaluated after the microiterations. It is not clear how this will affect the performance of the QM optimizer, in particular the Hessian update mechanism. Therefore, we propose two modifications to Friesner's scheme. First, after the microiterations, the wave function should be re-evaluated, followed by a new series of microiterations. Thus, the QM wave function and MM geometry optimization are iteratively repeated until self-consistency. Only then do we take a QM optimization step. Second, it is not necessary to use ESP (or other) localized charges to represent the QM charge distribution. With fast multipole methods,[44] the electrostatic interaction with the exact QM charge density can be incorporated efficiently. We will detail with the investigation of these issues in a forthcoming article that discusses the various types of embedding applied to the ONIOM scheme.[45] Preliminary results are very promising, with the average time per QM step only doubled with respect to optimization with mechanical embedding.

Microiterations with Constraints

In the microiteration-based optimization scheme without constraints described above, the Cartesian coordinates of the QM region are frozen during the microiteration phase, and the MM region positions itself with respect to the QM region. Therefore, the coordinate system employed for the optimization of the QM region does not need to contain coordinates that define its absolute position or orientation in space. If we place constraints on the internal coordinates of the model system, the same scheme still can be used, because the $\mathbf{q}^{\mathrm{m}}$ coordinates are held fixed during the microiterations.

The situation is quite different if constraints are placed on the Cartesian coordinates of some of the atoms in the MM region (e.g., freezing the outer layer of a protein according to a crystal structure). Now the QM region needs to move relative to the (partially frozen) MM region, and additional coordinates are required to facilitate this translation and orientation. One straightforward option would be to use Cartesian coordinates instead of internals for the optimization of the QM region. However, this may reduce the efficiency of the QM optimization significantly. A second option is to use internal coordinates for the entire real system. If the constraints can be expressed in terms of mixed Cartesian and internal coordinates, then no additional coordinates are required, but in general it may be rather difficult to construct such a coordinate system to fulfill the requirement that $\mathbf{q}^{\mathrm{l}}$ can be varied independently while $\mathbf{q}^{\mathrm{m}}$ is fixed. A third option is to remove the frozen Cartesian coordinates from the MM region and add them to the internal coordinate system of the QM region, resulting in a different $\mathbf{q}^{\mathrm{m}} - \mathbf{q}^{\mathrm{l}}$ partitioning. However, none of these schemes are very satisfactory, because the translation and orientation of the QM region with respect to the MM region would be handled during the optimization of the QM region. This may result in many extra steps for the QM optimization, involving expensive QM energy and gradient calculations, just to position the QM region in space.

A more efficient approach is to permit the translation and rotation of the QM region during the microiteration phase, which involves only inexpensive MM calculations. Because the internal coordinates of the QM region, $\mathbf{q}^{\mathrm{m}}$, need to held fixed during the microiterations, only rigid body motions of the QM region are allowed. Thus, $\mathbf{q}^{\mathrm{l}}$ must be augmented with coordinates describing the overall translation and rotation of the QM region within the real system. We note that such an approach can also improve the efficiency of an unconstrained optimization by allowing rigid body motion of the QM region during the microiterations, rather than forcing the entire MM region to reorient itself relative to the QM region.

Step Size Control for the QM Region

The optimization of the QM region can be carried out with any of the usual optimization methods for small molecules. Quasi-Newton methods with redundant internal coordinates are particularly efficient for such optimizations.[24,28–35] The size of the optimization steps is usually controlled with a trust radius or similar procedure. Because of the microiterations, however, some caution is necessary. In areas of the PES where the geometry changes rapidly, a step forward in the QM region may go into a new valley after the microiterations are converged. Stepping the QM region back from this new point might not lead back to the original valley. The result would be a type of chemical hysteresis (see Fig. 2). To follow a valley continuously to a minimum, one needs a more general trust region for the QM region, rather than a simple trust radius.

We wish to test whether a displacement, $\Delta\mathbf{q}^{\mathrm{m}}$, yields a structure such that the local minimum found by the next microiteration remains in the same valley. For $\mathbf{q}^{\mathrm{m}}_i$, $\mathbf{q}^{\mathrm{l}}_i$ is the local minimum in $\mathbf{q}^{\mathrm{l}}$ found by the microiteration. After a step from $\mathbf{q}^{\mathrm{m}}_i$ to $\mathbf{q}^{\mathrm{m}}_{i+1} = \mathbf{q}^{\mathrm{m}}_i + \Delta\mathbf{q}^{\mathrm{m}}$, let $\mathbf{q}^{\mathrm{l}}_{i+1}$ be the corresponding local minimum obtained by the next microiteration, starting from $\mathbf{q}^{\mathrm{l}}_i$. We can consider $\mathbf{q}^{\mathrm{l}}_{i+1}$ to be in the same valley as $\mathbf{q}^{\mathrm{l}}_i$, if we step back to $\mathbf{q}^{\mathrm{m}}_i$ and the microiteration starting from $\mathbf{q}^{\mathrm{l}}_{i+1}$ yields $\mathbf{q}^{\mathrm{l}}_i$ (see Fig. 2a). To be more precise, this needs to be true for $\mathbf{q}^{\mathrm{m}}_{i+1} = \mathbf{q}^{\mathrm{m}}_i + \alpha\Delta\mathbf{q}^{\mathrm{m}}$ for all $0 \le \alpha \le 1$. Figure 2b illustrates a situation when the reversed step does not yield the previous point, causing discontinuity in the PES with respect to $\mathbf{q}^{\mathrm{m}}$.

It would be rather time consuming to test $\Delta\mathbf{q}^{\mathrm{m}}$ for every optimization step in the QM region. In practice, we test the displacement only when the change in geometry produced by the microiterations is larger than a trust radius. If $|\mathbf{q}^{\mathrm{l}}_{i+1} - \mathbf{q}^{\mathrm{l}}_i| > \tau^{\mathrm{l}}$, we check whether $\mathbf{q}^{\mathrm{l}}_{i+1}$ obtained in the step forward to $\mathbf{q}^{\mathrm{m}}_{i+1} = \mathbf{q}^{\mathrm{m}}_i + \Delta\mathbf{q}^{\mathrm{m}}$ yields the previous $\mathbf{q}^{\mathrm{l}}_i$ when we step back to $\mathbf{q}^{\mathrm{m}}_i$. If not, then a sequence of microiterations minimizing $\mathbf{q}^{\mathrm{l}}$ is carried out for $\mathbf{q}^{\mathrm{m}}_{i+1} = \mathbf{q}^{\mathrm{m}}_i + \alpha\Delta\mathbf{q}^{\mathrm{m}}$ incrementing α from 0 to 1. This is shown in Figure 2c and demonstrates that the new structure is in the same valley as the old structure. However, when two valleys merge in the course of an optimization, it is possible that stepping back may not lead to the original valley, even with careful step size control, as shown in Figure 2d. This situation does not affect the result of the optimization but may cause unnecessary checking for surface discontinuity.

Results and Discussion

Our unconstrained and constrained microiteration methods have been implemented in the development version of the GAUSSIAN package.[43] For the QM region, we use the regular Berny optimization method, while for the MM region both the CG and GDIIS methods, on the basis of Cartesian coordinates, are employed. The universal force field (UFF),[46] Dreiding,[47] and AMBER[48] force fields are available in GAUSSIAN, as well as the QEq charge equilibration scheme to obtain partial charges for use in the MM calculations.[49]

Microiterations Versus Full Optimization

For the small to moderate size systems shown in Figure 1, we have compared full optimizations in redundant internals with optimizations employing microiterations. The results are collected in Table 1. All of the structures are preoptimized, so that the starting

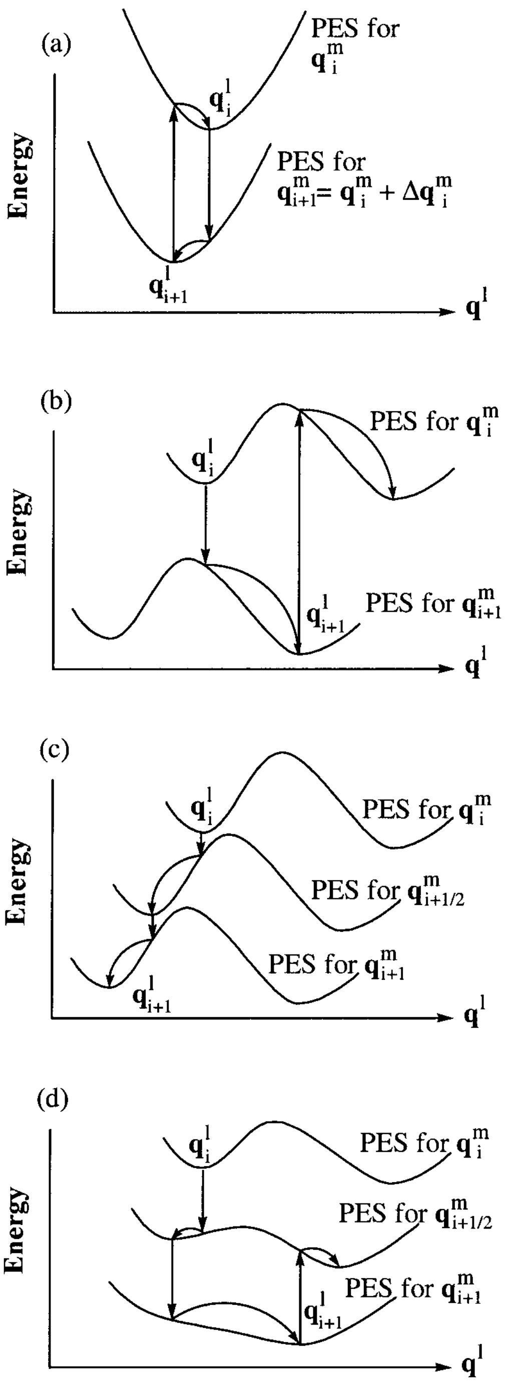

Figure 2. Behavior of microiteration-based optimizations; $\mathbf{q}_i^l$ is the local minimum in the MM region obtained by the microiterations on the potential energy surface for $\mathbf{q}_i^m$ in the QM region. (a) A step of $\Delta\mathbf{q}^m$ yields $\mathbf{q}_{i+1}^l$ in the same valley as $\mathbf{q}_i^l$. (b) A step of $\Delta\mathbf{q}^m$ yields $\mathbf{q}_{i+1}^l$ in a different valley; stepping back to $\mathbf{q}_i^l$ yields a different minimum. (c) When $\Delta\mathbf{q}^m$ is divided into a series of smaller steps, $\mathbf{q}_{i+1}^l$ remains in the same valley as $\mathbf{q}_i^l$. (d) If two valleys merge, it may not be possible to step back into the same valley, even if smaller steps are used.

structures are fairly close to the final equilibrium geometry. On average, the number of QM energy and gradient calculations is reduced by a factor of two as a result of the microiterations. If poorer starting geometries were used, much larger differences in the number of QM steps would be seen. Water dimer is rather flexible, and the regular optimizer has difficulties. However, for the optimization with microiterations, these flexible degrees of freedom are handled in the inexpensive microiterations. The difference in the number of QM calculations required by the two optimization schemes is quite large. Similar efficiencies can be expected in the optimization of more general systems involving solutes calculated by QM methods surrounded by solvent molecules treated by MM. Hexaphenylethane has a very strained C—C bond, but recent calculations using the ONIOM method yield a bond dissociation energy of 16.6 kcal/mol, suggesting that it may be stable enough to synthesize.[50] Because hexaphenylethane is fairly rigid, it is handled well by the full redundant internal coordinate optimizer. Many of the ligands used in organometallic chemistry have bulky substituents, but often these do not affect the electronic structure of the complex. The selectivity of bond activation in allyl-ammonium and iminium salts has been investigated in a complex of Ni(PCy$_3$) with [CH$_2$=CHCH$_2$HN=C(CH$_3$)$_2$]$^+$.[51] Here the initial structure and optimized structure are not very different, and the typical convergence problems associated with transition metals still exist in the model system. Therefore, the use of microiterations only improves the efficiency moderately. Silicon-phosphorus double bonds are very reactive, but can be observed and characterized if surrounded by sufficiently large substituents.[52] Microiterations improve the efficiency of the geometry optimization significantly because of the flexibility of the system. The room temperature isolation of singlet carbenes is an important milestone in the study of these important reactive intermediates.[53] This has been achieved through a combination of electronic and steric effects, as illustrated in the final example in Table 1. Again, microiterations improve the performance of the optimizations. It should be kept in mind, however, that all of the structures in the table were started close to the minimum and are small enough that microiterations are not essential. Nevertheless, the performance on these simple examples implies that much more is to be gained for large QM/MM systems.

Peptide Fragment

To probe the capabilities of microiterations with constraints, we examined a small fragment (Ala39-Lys40-Lys41-Phe42) of the bacteriorhodopsin system,[54,55] which we optimized with Lys41 both protonated and unprotonated, shown in Figure 3. Crystal structures do not contain hydrogens, and the protonation states of titratable residues are not always clear. Because protonation can significantly affect the geometry and relative orientation of the side chain of a residue, this serves as a suitable test for the microiteration procedure. To simulate the effect of the protein environment, we fixed the three terminal non-H atoms at each end of the backbone. Although these particular calculations are only constructed as a test for our method, optimizations in which part of the protein is kept fixed as a simple boundary condition are quite common in biochemical studies. The protonation site of lysine (~CH$_2$CH$_2$NH$_2$, ~CH$_2$CH$_2$NH$_3^+$) was calculated at the B3LYP/

Table 1. Comparison of Regular and Microiteration Optimizations.[a]

System	Water dimer	Hexaphenylethane	Organometallic	Phosphasilene	Carbene
Model system	Water monomer	Ethane	See Figure 1	$H_2Si{=}PH$	See Figure 1
High level	HF/6-31G(d)	AM1	B3LYP/LANL2DZ	AM1	AM1
Low level	UFF[b]	UFF[b]	UFF[b]	UFF[b]	UFF[b]
Without microiterations	41[c]	16[c]	23[c]	34	16
With microiterations	8[c]	10[c]	18[c]	9	10

[a] See Figure 1 for structures; water dimer started from the B3LYP/6-31G(d) geometry, hexaphenylethane from the AM1 geometry, and the rest from the level indicated but without QEq charges.

[b] Charges obtained using QEq scheme.

[c] Tight convergence criteria.

6-31G(d) level, while the remainder of the system was treated using the AMBER force field.[48]

We first optimized the unprotonated system, starting from the X-ray coordinates. This structure was then protonated and reoptimized. In Figure 3 the initial and optimized structures are displayed in tube and wire-frame, respectively, which shows that the QM region has moved significantly because of changes in the interaction between the phenyl ring and the protonation site. As described earlier, one method for performing the constrained optimization is to add Cartesian coordinates for the atoms in the QM region to the redundant internal coordinates in the $\mathbf{q}^m$ set, without augmenting the coordinates in the microiterations phase. In this case, the QM optimization takes care of the translation and rotation, and a total of 233 QM energy + gradient calculations were required. When we use our new approach and let the microiterations handle the position of the QM region, only 9 QM energy + gradient calculations were required.

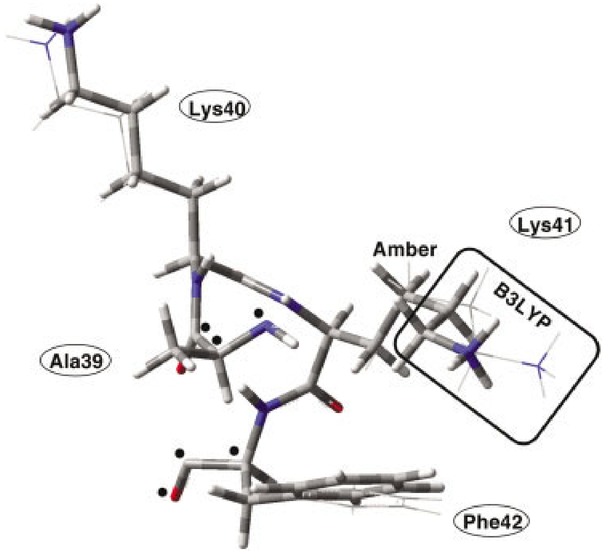

Figure 3. ONIOM(B3LYP/6-31G(d):Amber) initial (tube, based on unprotonated system) and optimized (wire-frame, protonated) geometry of Ala39-Phe42 fragment of bacteriorhodopsin. The six atoms frozen according to the crystal structure are indicated with dots.

Bacteriorhodopsin

The second example involves the full bacteriorhodopsin system. This system is too large to be treated fully with redundant internal coordinates, and the use of microiterations is essential. The protein is a barrel-like structure, around 3500 atoms, with a retinal chromophore in the center, as shown in Figure 4. We treated the chromophore at the PM3 level, and the rest of the protein with AMBER. However, in our calculations we did not include a small fragment (residues 157 to 161) that is missing in the crystal structure,[54] nor the membrane environment. Therefore, full optimization of this system may not be appropriate for the study of bacteriorhodopsin. To address this question, we performed both the full optimization and the optimization in which the four terminal peptide residues are constrained. For details on the system and preparation, see ref. 55.

In Figure 5 we show the deviations of both optimized structures with respect to the crystal structure, which were obtained by overlapping the structures by minimizing the RMS difference. We only show the values for the Cα backbone atoms in order to minimize the noise resulting from the floppy amino acid side chains. The peaks in the figure correspond to the loops between the α-helices that make up the barrel, while the valleys correspond to the center sections of the barrel components. The loops are less tightly bound than the helices, and move more during the optimization. It is clear that the overall structure of the protein stays intact in both optimizations. Looking closer at the plot, we see that the loose ends of the protein move significantly in the case of the unconstrained optimization, but this only occurs in a very localized region. In addition, the chromophore is far away from the loops, and we can conclude that for future studies constrained optimizations are not required.

Additional Applications

A number of studies are currently in progress that use the methods described in the current article. In a recent communication, we used the constraint method to study the activation of two related enzymes, methane monooxygenase and ribonucleotide reductase.[56] Optimization of the active sites alone, and the active sites with a part of the protein environment, showed that the conformation of the reaction center does depend on the environment. In addition, our results suggest that the protein environment drives

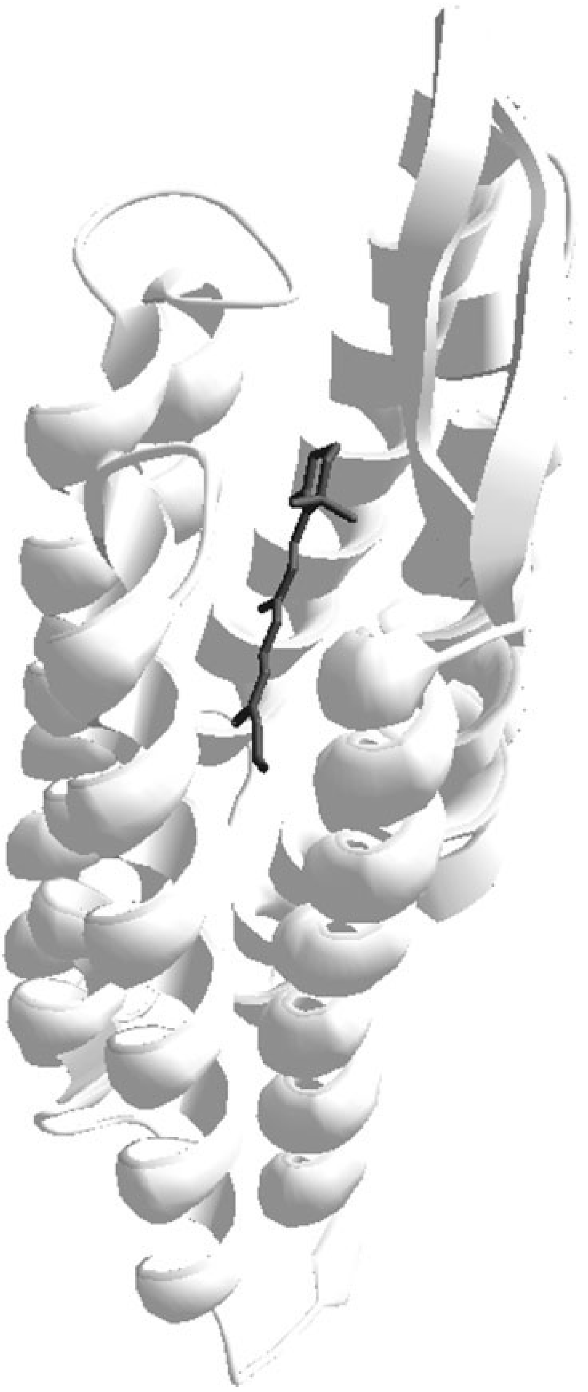

Figure 4. Structure of bacteriorhodopsin showing a series of α helices (in ribbon, treated with the Amber force field) surrounding the retinal chromophore (in tube, treated at the PM3 level).

the reaction in the same direction for both enzymes. This work is being extended to examine the effects of protein environments on the activation of dioxygen and methane with electronic embedding and to investigate different models and QM/MM partitionings. We are also continuing the study of the effects of protein environments on bacteriorhodopsin,[55] studying the mechanism of photoisomerization with most of the protein taken into account in the QM/MM calculation. In collaboration with S. Mobashery, we are looking at the covalent S_N2 inhibition of carboxypeptidase. A (CPA)[57,58] and the carbamylation of lysine in the active site of OXA-10 β-lactamase.[59–62] To assure reliable energetics, the protein environments in these systems are constrained to the X-ray structures (but care must be taken that this does not bias the model). For the S_N2 reaction between CPA and the inhibitor, the calculations show that the protein has a substantial effect on the energy along the reaction path, lowering the barrier and shifting the position of the transition state significantly.[63] In OXA-10, the effect of the environment on height and position of the barrier is small, but the energetics of the carbamylation reaction change from endothermic in the small molecule model system to exothermic in the ONIOM calculations of the enzyme.[64]

Conclusions

For hybrid techniques such as the ONIOM method, geometry optimization can be carried out in a cost-effective manner by taking advantage of the division of the system into a small but expensive QM region and a large but inexpensive MM region. A series of microiterations is used to minimize the MM region for each step in the QM region. We have implemented an efficient geometry optimization scheme for ONIOM calculations in the following manner:

1. Coordinates are separated into two sets, $\mathbf{q}^{\mathrm{m}}$ for the QM region and $\mathbf{q}^{\mathrm{l}}$ for the MM region so that $\partial E^{\mathrm{high}}_{\mathrm{model}}/\partial \mathbf{q}^{\mathrm{l}} = \mathbf{0}$ and $\partial E^{\mathrm{low}}_{\mathrm{model}}/\partial \mathbf{q}^{\mathrm{l}} = \mathbf{0}$. Thus, only the energy and gradient of the real system at the low level need to be calculated during the microiterations.
2. Cartesian coordinates are used for $\mathbf{q}^{\mathrm{l}}$ to avoid costly coordinate transformations for the large MM region, and redundant internal coordinates are used for the QM region to minimize the number of expensive QM steps needed for the optimization.
3. The MM region is augmented with coordinates to permit the rigid body translation and rotation of the QM region. This is essential if any of the atoms in the MM region are constrained, but also improves the efficiency of unconstrained optimizations.
4. The step size in the QM region needs to be controlled so that the optimization stays in the same valley during the next microiteration.

In the present work, we chose to illustrate the microiteration approach within the ONIOM(QM:MM) framework, but the method can be applied to any 2- or 3-layer ONIOM combination. For other combinations, the difference in the computational cost of the real and model calculations is typically not as large as for QM/MM, and the advantage may not be as substantial. For a 3- (or more) layer ONIOM optimization, a triply nested microiteration scheme could be employed, though it is unlikely that this would significantly increase the efficiency beyond the simple microiteration approach. For the optimization of large QM/MM systems, it is clear that microiteration optimization schemes are very useful, and

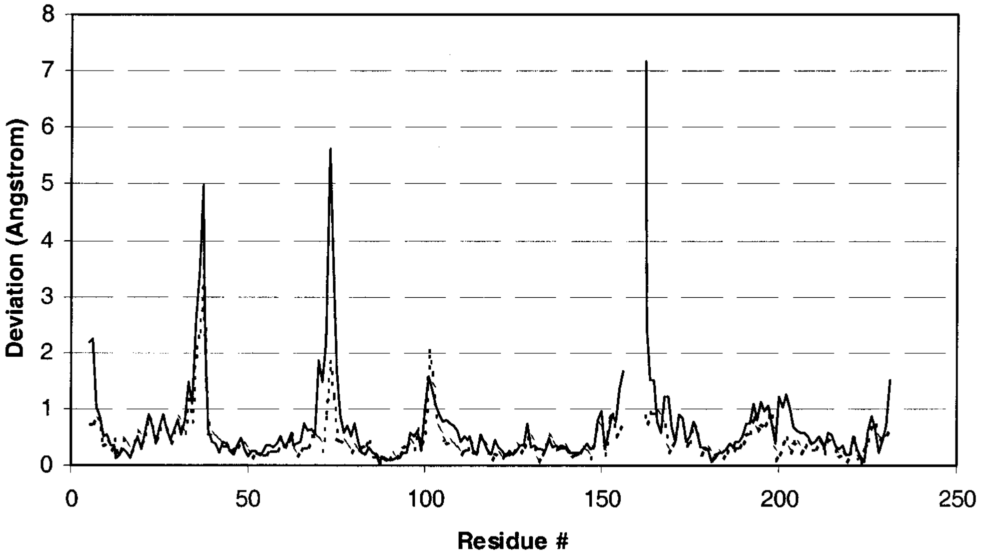

Figure 5. Deviation of the bacteriorhodopsin Cα backbone atoms during unconstrained (solid line) and constrained (broken line) optimization. See text for details.

our new extension makes the constrained and unconstrained optimization of QM/MM systems equally efficient.

Acknowledgments

Ö. F. thanks the Hungarian Academy of Sciences for a János Bolyai Research Scholarship. The authors thank Dr. Maricel Torrent and Jason B. Cross for help in assessing the methods.

References

1. Froese, R. D. J.; Morokuma, K. In Encyclopedia of Computational Chemistry, Vol. 2; Schleyer, P. v. R., Allinger, N. L., Kollman, P. A., Clark, T., Schaefer III, H. F., Gasteiger, J., Schreiner, P. R., Eds.; Wiley: Chichester, 1998; p 1244.
2. Gao, J. In Encyclopedia of Computational Chemistry, Vol. 2; Schleyer, P. v. R., Allinger, N. L., Kollman, P. A., Clark, T., Schaefer III, H. F., Gasteiger, J., Schreiner, P. R., Eds.; Wiley: Chichester, 1998; p 1257.
3. Gao, J. In Reviews in Computational Chemistry, Vol. 7; Lipkowitz, K. B., Boyd, D. B., Eds.; VCH: New York, 1996, p. 119.
4. Warshel, A.; Levitt, M. J Mol Biol 1976, 103, 227.
5. Singh, U. C.; Kollman, P. A. J Comput Chem 1986, 7, 718.
6. Field, M. J.; Bash, P. A.; Karplus, M. J Comput Chem 1990, 11, 700.
7. Mordasini, T. Z.; Thiel, W. Chimia 1998, 52, 288.
8. Turner, A. J.; Moliner, V.; Williams, I. H. PCCP Phys Chem Chem Phys 1999, 1, 1323.
9. Zhang, Y. K.; Liu, H. Y.; Yang, W. T. J Chem Phys 2000, 112, 3483.
10. Murphy, R. B.; Philipp, D. M.; Friesner, R. A. J Comput Chem 2000, 21, 1442.
11. Hayashi, S.; Ohmine, I.; Schulten, K. Biophys J 2001, 80, 2726.
12. Hayashi, S.; Ohmine, I. J Phys Chem B 2000, 104, 10678.
13. Maseras, F.; Morokuma, K. J Comput Chem 1995, 16, 1170.
14. Humbel, S.; Sieber, S.; Morokuma, K. J Chem Phys 1996, 105, 1959.
15. Svensson, M.; Humbel, S.; Froese, R. D. J.; Matsubara, T.; Sieber, S.; Morokuma, K. J Phys Chem 1996, 100, 19357.
16. Svensson, M.; Humbel, S.; Morokuma, K. J Chem Phys 1996, 105, 3654.
17. Dapprich, S.; Komaromi, I.; Byun, K. S.; Morokuma, K.; Frisch, M. J. Theochem-J Mol Struct 1999, 462, 1.
18. Vreven, T.; Morokuma, K. J Comput Chem 2000, 21, 1419.
19. Vreven, T.; Mennucci, B.; da Silva, C. O.; Morokuma, K.; Tomasi, J. J Chem Phys 2001, 115, 62.
20. Farkas, Ö.; Schlegel, H. B. J Chem Phys 1998, 109, 7100.
21. Farkas, Ö.; Schlegel, H. B. J Chem Phys 1999, 111, 10806.
22. Paizs, B.; Fogarasi, G.; Pulay, P. J Chem Phys 1998, 109, 6571.
23. Paizs, B.; Baker, J.; Suhai, S.; Pulay, P. J Chem Phys 2000, 113, 6566.
24. Baker, J.; Kinghorn, D.; Pulay, P. J Chem Phys 1999, 110, 4986.
25. Billeter, S. R.; Turner, A. J.; Thiel, W. PCCP Phys Chem Chem Phys 2000, 2, 2177.
26. Nemeth, K.; Coulaud, O.; Monard, G.; Angyan, J. G. J Chem Phys 2000, 113, 5598.
27. Nemeth, K.; Coulaud, O.; Monard, G.; Angyan, J. G. J Chem Phys 2001, 114, 9747.
28. Schlegel, H. B. In Modern Electronic Structure Theory; Yarkony, D. R., Ed.; World Scientific Publishing: Singapore, 1995; p 459.
29. Schlegel, H. B. In Encyclopedia of Computational Chemistry, Vol. 2; Schleyer, P. v. R., Allinger, N. L., Kollman, P. A., Clark, T., Schaefer III, H. F., Gasteiger, J., Schreiner, P. R., Eds.; Wiley: Chichester, 1998; p 1136.
30. Pulay, P.; Fogarasi, G.; Pang, F.; Boggs, J. E. J Am Chem Soc 1979, 101, 2550.
31. Fogarasi, G.; Zhou, X. F.; Taylor, P. W.; Pulay, P. J Am Chem Soc 1992, 114, 8191.
32. Pulay, P.; Fogarasi, G. J Chem Phys 1992, 96, 2856.
33. Peng, C. Y.; Ayala, P. Y.; Schlegel, H. B.; Frisch, M. J. J Comput Chem 1996, 17, 49.
34. Baker, J.; Kessi, A.; Delley, B. J Chem Phys 1996, 105, 192.
35. von Arnim, M.; Ahlrichs, R. J Chem Phys 1999, 111, 9183.
36. Fletcher, R. Practical Methods of Optimization; Wiley: Chichester, 1981.

37. Csaszar, P.; Pulay, P. Theochem-J Mol Struct 1984, 114, 31.
38. Farkas, Ö. Ph.D. (Csc) Thesis, Eötvös Loránd University and Hungarian Academy of Sciences, Budapest, 1995.
39. Farkas, Ö.; Schlegel, H. B. PCCP Phys Chem Chem Phys 2002, 4, 11.
40. Schlegel, H. B. J Comput Chem 1982, 3, 214.
41. Schlegel, H. B. Theor Chim Acta 1984, 1984, 66, 333.
42. Wittbrodt, J. M.; Schlegel, H. B. Theochem-J Mol Struct 1997, 398, 55.
43. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Montgomery, J. A.; Vreven, T.; Kudin, K. N.; Burant, J. C.; Millam, J. M.; Iyengar, S.; Tomasi, J.; Barone, V.; Mennucci, B.; Cossi, M.; Scalmani, G.; Rega, N.; Petersson, G. A.; Ehara, M.; Toyota, K.; Hada, M.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Kitao, O.; Nakai, H.; Honda, Y.; Nakatsuji, H.; Li, X.; Knox, J. E.; Hratchian, H. P.; Cross, J. B.; Adamo, C.; Jaramillo, J.; Cammi, R.; Pomelli, C.; Gomperts, R.; Stratmann, R. E.; Ochterski, J.; Ayala, P. Y.; Morokuma, K.; 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.; Andres, J. L.; Gonzalez, C.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A. GAUSSIAN 99, Development Version; Gaussian, Inc.: Pittsburgh, PA, 2002.
44. Greengard, L.; Rokhlin, V. J Comput Phys 1987, 73, 325.
45. Vreven, T.; Komaromi, I.; Dapprich, S.; Byun, K. S.; Morokuma, K.; Frisch, M. J. In preparation.
46. Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. J Am Chem Soc 1992, 114, 10024.
47. Mayo, S. L.; Olafson, B. D.; Goddard, W. A. J Phys Chem 1990, 94, 8897.
48. 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.
49. Rappe, A. K.; Goddard, W. A. J Phys Chem 1991, 95, 3358.
50. Vreven, T.; Morokuma, K. J Phys Chem A, to appear.
51. Torrent, M.; Musaev, D. G.; Morokuma, K. Organometallics 2000, 19, 4402.
52. Armitage, D. A. In The Silicon-Heteroatom Bond; Patai, S., Rappoport, Z., Eds.; Wiley: New York, 1991; pp 151, 211, and references therein.
53. Arduengo, A. J Accounts Chem Res 1999, 32, 913, and references therein.
54. Luecke, H.; Schobert, B.; Richter, H. T.; Cartailler, J. P.; Lanyi, J. K. J Mol Biol 1999, 291, 899.
55. Vreven, T.; Morokuma, K. Theor Chem Acc, to appear.
56. Torrent, M.; Vreven, T.; Musaev, D. G.; Morokuma, K.; Farkas, Ö.; Schlegel, H. B. J Am Chem Soc 2001, 124, 192.
57. Tanaka, Y.; Grapsas, I.; Dakoji, S.; Cho, Y. J.; Mobashery, S. J Am Chem Soc 1994, 116, 7475.
58. Massova, I.; Martin, P.; deMel, S.; Tanaka, Y.; Edwards, B.; Mobashery, S. J Am Chem Soc 1996, 118, 12479.
59. Golemi, D.; Maveyraud, L.; Vakulenko, S.; Tranier, S.; Ishiwata, A.; Kotra, L. P.; Samama, J. P.; Mobashery, S. J Am Chem Soc 2000, 122, 6132.
60. Golemi, D.; Maveyraud, L.; Vakulenko, S.; Samama, J. P.; Mobashery, S. Proc Natl Acad Sci USA 2001, 98, 14280.
61. Maveyraud, L.; Golemi, D.; Kotra, L. P.; Tranier, S.; Vakulenko, S.; Mobashery, S.; Samama, J. P. Structure 2000, 8, 1289.
62. Maveyraud, L.; Golemi-Kotra, D.; Ishiwata, A.; Meroueh, O.; Mobashery, S.; Samama, J. P. J Am Chem Soc 2002, 124, 2461.
63. Cross, J. B.; Vreven, T.; Meroueh, O.; Mobashery, S.; Schlegel, H. B. J Am Chem Soc, to appear.
64. Cross, J. B.; Li, J.; Vreven, T.; Meroueh, O.; Mobashery, S.; Schlegel, H. B. In preparation.