Curvature Correction for Microiterative Optimizations with QM/MM Electronic Embedding

Tibor András Rokob*[a] and Lubomír Rulíšek[a]

One of the most common methods to treat the electrostatic effect of the environment in QM/MM calculations is to include the MM atoms as point charges in the QM Hamiltonian. In this case, a microiterative geometry optimization ignoring the QM contributions to the forces in the relaxation of the environment cannot yield exact stationary points. One solution that has been suggested in the literature is based on using a constant additive correction to the MM gradient during the microiterations, determined in the preceding macroiteration. Here, we analyze the convergence properties of the gradient correction method and point out that a smooth relaxation is not ensured if the curvature of the approximate, MM-based description of the potential energy surface of the environment is too small in comparison with the exact one. We suggest a computationally cheap second-order correction that uses an estimated Hessian from the Davidon–Fletcher–Powell method to tackle the problems caused by the too small curvature. Test calculations on four metalloenzymatic systems (~100 QM atoms, ~2000 relaxed MM atoms, ~20,000 atoms in total) show that our approach efficiently restores the convergence where gradient correction alone would lead to oscillations. © 2012 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.22951

Introduction

Computational modeling represents nowadays almost an indispensable tool in our arsenal of methods to study the structure and function of numerous extended systems. Substrate binding processes, reaction mechanisms, and spectroscopic properties of explicitly solvated complexes, enzymes, or heterogeneous catalysts, as well as mechanical behavior of solids with a crack or dislocation can nowadays be addressed by calculations, and the results broaden and deepen our understanding mostly by providing a unique mapping between the structural and energetic space. As such, they may eventually lead to advancements in the field of applications as well, such as drug design or catalyst development. However, a full quantum chemical description of these systems represents a formidable task, and multilevel methods are typically applied to render the problems tractable within acceptable time. As first proposed in the 1970s,[1,2] a quantum mechanical description of the active center and a molecular mechanics force field for the environment (QM/MM) are often adopted as a reasonable compromise.[3–16] In this way, bond breaking or bond forming events, as well as spectroscopic features in the region of interest can be described, and the steric or electronic effects of the environment are also efficiently included to a certain level of accuracy. Besides the choice concerning the QM and MM level of theory, the QM/MM approaches reported in the literature differ in the treatment of the interactions between the QM and MM parts. When there are covalent bonds between these subsystems, which is the usual case, the corresponding short-range bonded-type interactions across the boundary must be treated. Such bonds create dangling valences in the QM part, which are capped by hydrogen atoms, or more sophisticated approaches, like special frozen orbitals, are used to deal with this problem.[3–16] Cross-boundary bonded interaction terms can be described simply by the MM force field or by extensive specific parameterization procedures.[3–16] Among the long-range interactions, electrostatics often plays the dominant role, and several levels of its treatment can be envisioned.[17] The simplest method is to describe it at the MM level, assigning some set of atomic charges to the QM atoms or using its true charge density without polarization. This approach, termed mechanical embedding, has the disadvantage of neglecting the changes in the QM wavefunction in response to the electrostatic field produced by the environment. A more sophisticated procedure, called electronic embedding, takes this into account, typically by including all MM atoms as partial point charges in the QM Hamiltonian. An alternative approach using an electrostatic interaction operator based on fitted atomic charges[18,19] or multipoles[20] has also been suggested. Further levels of accuracy can be achieved, e.g., by taking into account the polarization in the MM region as well.[17] Comparisons of various methods have also been published.[21,22]

Although molecular dynamics can sometimes play a significant role in the understanding of the studied phenomenon by providing adequate sampling of the configuration space of the molecule (enzyme) of interest, it requires significant computational effort, and despite the possible pitfalls,[23] useful information can often be obtained by simple geometry optimizations yielding minima and transition states.[24,25] The efficiency of this approach crucially depends on the

[a] T. A. Rokob, L. Rulíšek
Institute of Organic Chemistry and Biochemistry, Academy of Sciences of the Czech Republic, Flemingovo náměstí 2, 16610 Prague, Czech Republic
E-mail: rokob@uochb.cas.cz

Contract/grant sponsor: Grant Agency of the Academy of Sciences of the Czech Republic; Contract/grant number: IAA400040802.

optimization algorithm. QM calculations traditionally use Hessian-based methods (like the variants of quasi-Newton[26]) as the cost of an energy or gradient calculation is large, and the number of iterations is best kept at the minimum. In contrast, MM optimizations often use simpler methods like conjugate gradient,[27] because the number of degrees of freedom is typically very high, and the storage and computational costs of the classic second-order methods are higher than that of additional iterations. Clearly, for QM/MM, we need to combine the best of the two worlds. We can benefit from the significant developments in this area, yielding low-memory versions of Hessian update methods,[28–31] efficient reaction pathway following techniques,[32] and favorably scaling coordinate systems[33] or coordinate manipulation algorithms,[34–39] which open up the possibility of dealing with large systems. Specifically for QM/MM mechanical embedding, a linearly scaling procedure implicitly using a full Hessian with analytic MM contributions has also been proposed.[40]

An established method used in QM/MM optimizations is the microiterative approach.[18,33,40–53] During a microiterative optimization, two optimizers work separately and alternately: relaxation of the core (QM) region using a Hessian-based method and relaxation of the environment (MM) region using some inexpensive method. One possibility is to do a full relaxation of the environment (in several "microiterations") after every single step done in the core region ("macroiteration")[33,40–43,45,47–52]; in this way, the whole optimization can be considered to be driven[42] by the coordinates in the core region. Another possibility is to alternate between full optimization of environment with fixed core and full optimization of the core with fixed environment.[18,44,46,53] From a comparison of the two approaches, the latter has been suggested to be superior, especially for the location of transition states.[44] For mechanical embedding, explicit coupling of the two optimizations has also been implemented.[40]

One advantage of the microiterative approach is that the number of degrees of freedom is small in the core region, which allows fine-tuned, robust methods to be applied for its optimization. This has been particularly useful in locating transition states; notably, in this case, the core region of the geometry optimization need not coincide with the quantum mechanically treated part.[33,42,44,53] The other merit of the microiterative method stems from the low cost of MM calculations, which allows the fast relaxation of the large MM part, decreasing the number of costly QM calculations. The straightforward applicability of this latter technique hinges on the fact that the QM contributions to the energy and the gradients do not depend on the coordinates of the atoms in the MM region. Unfortunately, this is only true in the case of mechanical embedding. With point-charge-based electronic embedding, MM atoms directly enter the QM Hamiltonian, and thus, in principle, a full QM calculation would be necessary in each microiteration to get the accurate forces acting on the MM atoms, rendering the microiterative approach useless at first sight. Nevertheless, several solutions have been proposed to this problem. The simplest workaround is to treat the interactions between the QM and MM parts at the MM level during the microiterations, while keeping the MM atoms (represented by the point charges) in the QM Hamiltonian during the QM calculations.[43,46,50,51] While this combination was found to be adequate for QM/MM free energy perturbation methods,[54] the location of stationary points is hampered by the QM and MM optimizers working on different potential energy surfaces (PESs). For this reason, the MM atoms are not fully relaxed in the microiterations, and therefore the overall convergence is deteriorated.[44] Moreover, this method does not locate the true stationary points on the electronic embedding QM/MM PES.[44] A similar but less inaccurate approach uses the frozen QM density to treat the electrostatics.[18,22,54] This idea can be considered as a first-order perturbation treatment, and it has also been adopted for Monte Carlo simulations,[56–59] together with a related second-order perturbation theory version.[60] An alternative solution is based on the observation that upon moving the MM atoms, only one-electron integrals change; a useful approximate SCF solution can be obtained quickly by doing a single SCF iteration instead of a full QM calculation.[44] A theoretically more rigorous approximation is to use two QM levels: a more expensive one to describe the active center in vacuum and a cheap, e.g., semiempirical, one to account for its interaction with the point charges environment.[61,62] During the microiterations, only the low level QM calculations have to be done; yet the accuracy will be close to the high level and a single, well-defined PES is used.[62] Finally, the problem can be circumvented by using the electronic embedding treatment with the multipole fitting operator instead of point charges and carrying out the microiterative MM optimization in each SCF iteration.[63]

Staying with the point-charge-based embedding, there is a further approach in addition to the above strategies, pioneered by Friesner et al.[47,48] and later adopted by Kästner and coworkers[49] as well as Schlegel and coworkers.[45] These authors suggested to compute exact forces on the MM atoms in the macroiteration, as part of the QM + point charges calculation, and then use the atomic charge based (i.e., MM) description of the active center only for estimating the changes in the forces as the atoms in the MM region move during the microiterations. In other words, the forces coming from the MM description are corrected by the difference of the exact ones and the MM ones computed at a nearby point. In principle, as the optimization proceeds, the steps done in the QM region become gradually smaller, the MM region has to move less during the relaxation, and this approximation converges to the exact forces, yielding exact QM/MM stationary points.[48] Indeed, the authors reported generally good convergence of this computationally very cheap scheme.[47–49]

Notwithstanding the general experience, we found that in certain cases, the microiterative optimizations do not converge rapidly and smoothly, and they furthermore show oscillatory behavior when the exact forces are recomputed in the succeeding macroiterations. Even with small QM steps, the MM relaxation does not tend to be quick; hence, it became apparent that the convergence of this method is not always guaranteed. However, as this procedure is in general inexpensive, easy to implement, and most often generates good results, a closer study of the necessary conditions for convergence and possible solutions for the problematic cases seemed of interest.

In this article, we report the results of our efforts directed toward a better understanding and improvement of the force correction method. We start by describing the software packages and test systems, and after that we summarize the necessary theoretical background. We begin the description of the actual results with a short analysis of a failed example calculation and then present an analytic treatment of the convergence issue for a specific case. On the basis of the general observations made there, we propose an additional correction, involving the curvature of the PES, which represents an insignificant computational overhead but can correct the behavior, as exemplified by a couple of test cases.

Computational Details

All the development work and calculations were carried out within the frames of the ComQum package,[50,51] which combines Turbomole 6.3.1[64] and Amber 8[65] for QM/MM optimizations. It implements electronic embedding, the hydrogen link atom scheme, and a microiterative approach; the MM system is fully relaxed after each optimization step in the QM region. This relaxation in the original version proceeds using an atomic-charge-based (MM) description of the quantum system. The used charges are Mulliken charges, corrected by an additive constant on each atom. These constants are determined at some intermediate point during the optimization, and they are taken as the difference of charges fitted to reproduce the electrostatic potential and the Mulliken ones.[51] The QM/MM coupling is implemented using a collection of small programs; the QM and MM software packages are used without modifications, apart from a couple of lines in Amber adding the capability to write the computed MM forces to file.

The *statpt* module of Turbomole is utilized to relax the geometry of the QM region, which implements geometric direct inversion in the iterative subspace[66] for small gradients and a restricted-step second-order optimization for large gradients, where a level shift parameter in the Hessian is used to control the step length to correspond to the actual trust radius.[67] The trust radius is updated according to the accuracy of the prediction of the energy change, whereas for the Hessian, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update algorithm[68–71] is used. A diagonal matrix was used as the initial Hessian. The built-in default convergence criteria were used, i.e., the geometry was considered converged when the energy change is below 10^{-6} a.u., the root-mean-square/maximum QM gradient is below $5 \times 10^{-4}/10^{-3}$ a.u., and the root-mean-square/maximum QM displacement is below $5 \times 10^{-4}/10^{-3}$ a.u. Each optimization was allowed to do at least 300 macroiterations; the runs that showed strong oscillations without any appreciable progress toward convergence were terminated there, while the others were allowed as many iterations as necessary to reach the minimum.

For the MM region, the conjugate gradient optimizer of Amber was used. The MM optimizations were run until a convergence criterion of 0.001 kcal mol^{-1} Å^{-1} $\approx 8.4\times10^{-7}$ a.u. on the root-mean-square MM gradient was fulfilled, but at most 1000 microiterations were done. In both the QM and MM optimizations, Cartesian coordinates were used.

During this work, we implemented the force and curvature corrections by adding appropriate code to the force computation routine within Amber as well as by creating a couple of auxiliary programs to prepare the data. In the runs with corrections, the calculations were only accepted as converged if the root-mean-square of the exact forces in the MM region was below 0.0011 kcal mol^{-1} Å^{-1} $\approx 9.3\times10^{-7}$ a.u. To have some tolerance for numerical noise, this threshold has been chosen to be slightly higher than the convergence threshold for the approximate MM forces in the microiterations (see earlier).

The test calculations were carried out for four metalloproteins, previously or currently studied in our group, with two initial structures of the QM system for each case. Specifically, the proteins were the soluble stearoyl acyl carrier protein Δ^9 desaturase,[72] the multicopper oxidase CueO,[73] the human glutamate carboxypeptidase II,[74] and the manganese superoxide dismutase.[75] The QM regions consisted of ∼60 to 170 atoms, the MM regions ∼20,000 to 30,000 atoms; from the latter, ∼2000 to 3000 atoms were actually relaxed in the optimization, with the rest being fixed at the crystallographic coordinates. The BP86/def-SV(P) level of density functional theory[76–78] with the resolution of identity approximation[79] and the Cornell force field[80] were used throughout. The additive constants linking Mulliken and fitted charges were determined in the 1st macroiteration, except for glutamate carboxypeptidase II, where they were set to zero. These specific choices were motivated by the aim to have many calculations in the test set that are problematic for the gradient correction.

Theoretical Background

There are several ways to formulate a QM/MM energy expression. Here, we will use a simplified form of the subtractive formula used in the ComQum implementation,[50,51] but we would like to emphasize that all the subsequent considerations are independent on the specific form chosen. The active center, described at the QM level, is denoted as 'system 1', and the force-field-based environment as 'system 2'. We group the Cartesian coordinates of the full system to the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, corresponding to these subsystems. ComQum uses a hydrogen link atom approach to cap system 1 in the QM calculations, but this only affects the actual form of the dependence of the QM energy on $\mathbf{x}_1$ and is not of particular interest here. As typical in such calculations, ComQum furthermore uses an MM region around system 2 that is frozen at the crystallographic positions. This region, called 'system 3', will be implicit in the subsequent formulae. Then, the QM/MM energy can be written as follows:

$$E(\mathbf{x}_1,\mathbf{x}_2) = E^{\mathrm{QM}}_{1+\mathrm{PC}}(\mathbf{x}_1,\mathbf{x}_2) + E^{\mathrm{MM}}_{12,q_1=0}(\mathbf{x}_1,\mathbf{x}_2) - E^{\mathrm{MM}}_{1,q_1=0}(\mathbf{x}_1). \qquad (1)$$

In this formula, $E^{\mathrm{QM}}_{1+\mathrm{PC}}$ is the QM energy of the (hydrogen-capped) system 1. System 2 atoms are included in this calculation as point charges, but their self-energy is omitted from

$E^{\mathrm{QM}}_{1+\mathrm{PC}}$. The term $E^{\mathrm{MM}}_{12,q_1=0}$ is the MM energy of the whole system with the charges in system 1 zeroed, and $E^{\mathrm{MM}}_{1,q_1=0}$ is the MM energy of (hydrogen-capped) system 1, again, with charges zeroed. During the microiterations, system 2 atoms are moved, and thus the gradient vector

$$\mathbf{g}_2(\mathbf{x}_1,\mathbf{x}_2)=\frac{\partial E(\mathbf{x}_1,\mathbf{x}_2)}{\partial \mathbf{x}_2} \tag{2}$$

should be computed, which clearly involves QM contributions.

As discussed in the Introduction, the simplest way to avoid the costly QM calculations is to use an approximate potential energy surface during the microiterations, based on the MM representation of system 1. We will refer to this representation as the atomic charge (AC) based representation. This is exactly the approach adapted in ComQum; during the microiterations following the *k*th macroiteration, system 2 is relaxed using:

$$E^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2\middle|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)=E^{\mathrm{MM}}_{12,q_1=\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right) \tag{3}$$

$$\mathbf{g}_2^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{k}\right)=\frac{\partial E^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)}{\partial \mathbf{x}_2} \tag{4}$$

where $E^{\mathrm{MM}}_{12,q_1=\mathrm{AC}}$ $(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ is now the MM energy of the full system having coordinates $(\mathbf{x}_1,\mathbf{x}_2)$ and with the QM atoms being assigned a set of atomic charges from the QM calculation in the last macroiteration, which was done at the geometry $(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$.

The basic quantity for the force correction approach is the difference between the exact and the AC-based PESs:

$$E^{\mathrm{corr}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)=E(\mathbf{x}_1,\mathbf{x}_2)-E^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right). \tag{5}$$

We correct the energies[49] and gradients[47,48] using E^{corr} and its derivative with respect to $\mathbf{x}_2$ ($\mathbf{g}_2^{\mathrm{corr}}$), for which the exact values are available at $(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$. The gradient and energy for the microiterations after the *k*th macroiteration need to be consistent with each other,[49] and we will thus have the following expressions (with $\langle \mathbf{a},\mathbf{b}\rangle$ denoting dot product and the superscript G referring to the gradient corrected case):

$$\begin{aligned}\mathbf{g}_2^{\mathrm{G}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)&=\mathbf{g}_2^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)\\&\quad+\underbrace{\left[\mathbf{g}_2\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)-\mathbf{g}_2^{\mathrm{AC}}\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)\right]}_{\mathbf{g}_2^{\mathrm{corr}}\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)}\end{aligned} \tag{6a}$$

$$\begin{aligned}E^{\mathrm{G}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)&=E^{\mathrm{AC}}\left(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)\\&\quad+\underbrace{\left[E\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)-E^{\mathrm{AC}}\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)\right]}_{E^{\mathrm{corr}}\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right)}\\&\quad+\left\langle\mathbf{g}_2^{\mathrm{corr}}\left(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}\right),\mathbf{x}_2-\mathbf{x}_2^{(k)}\right\rangle\end{aligned} \tag{6b}$$

Note that as $(\mathbf{x}_1,\mathbf{x}_2)\to(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$, the corrected values approach the exact ones ($E^{\mathrm{G}}\to E$, $\mathbf{g}_2^{\mathrm{G}}\to\mathbf{g}_2$), which is the basis for the statement about the convergence of this approach in the literature.[48] The background of our work presented in this article was our implementation of this force correction formula to the ComQum package. The used optimization algorithm is summarized in Figure 1 (we did not use the step rejection implemented in Ref. [49]). Note that in this program flow, a geometry relaxation step in system 1 is taken before entering the microiterations. This is emphasized in the above formulae by having $\mathbf{x}_1$ instead of $\mathbf{x}_1^{(k)}$ on the left-hand side, although $\mathbf{x}_1$ is not altered during the microiterations. A fully consistent approach would include a second QM calculation in each macroiteration, but as it was stated in the literature, this does not provide significant improvement.[49] Another step toward consistency would be to iterate between exact force computation and microiterative geometry relaxation of system 2 until the

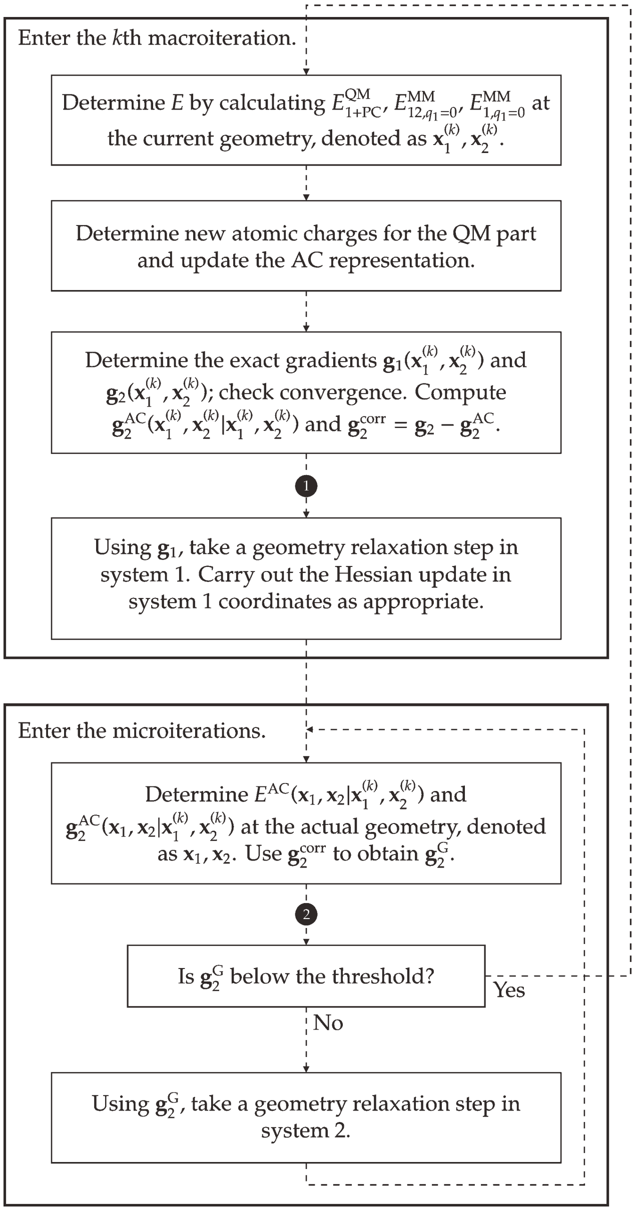
Enter the kth macroiteration.
Determine E by calculating $E^{\mathrm{QM}}_{1+\mathrm{PC}}$, $E^{\mathrm{MM}}_{12,q_1=0}$, $E^{\mathrm{MM}}_{1,q_1=0}$ at the current geometry, denoted as $\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}$.
Determine new atomic charges for the QM part and update the AC representation.
Determine the exact gradients $\mathbf{g}_1(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ and $\mathbf{g}_2(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$; check convergence. Compute $\mathbf{g}_2^{\mathrm{AC}}(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ and $\mathbf{g}_2^{\mathrm{corr}}=\mathbf{g}_2-\mathbf{g}_2^{\mathrm{AC}}$.
1
Using $\mathbf{g}_1$, take a geometry relaxation step in system 1. Carry out the Hessian update in system 1 coordinates as appropriate.
Enter the microiterations.
Determine $E^{\mathrm{AC}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ and $\mathbf{g}_2^{\mathrm{AC}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ at the actual geometry, denoted as $\mathbf{x}_1,\mathbf{x}_2$. Use $\mathbf{g}_2^{\mathrm{corr}}$ to obtain $\mathbf{g}_2^{\mathrm{G}}$.
2
Is $\mathbf{g}_2^{\mathrm{G}}$ below the threshold?
Yes
No
Using $\mathbf{g}_2^{\mathrm{G}}$, take a geometry relaxation step in system 2.

Figure 1. Microiterative geometry optimization scheme with force correction as implemented in the ComQum package. Black circles with numbers 1 and 2 on the arrows show points where the new steps discussed in this article will be added (see later, on Fig. 3).

exact system 2 forces are converged and only then take a geometry step in the QM part[45]; we implemented this method as well but did not use it in this study.

Results and Discussion

Nature of the observed oscillations

In line with the general experience reported in the literature,[47–49] we found that in many cases, the gradient correction works very well. The located minima are significantly lower in energy than the results from the AC-based calculations; the difference in absolute energies can exceed 10 kcal mol^{-1}. A comparison of the root-mean-square values of the exact system 2 gradients also confirms the superiority of the gradient-corrected method. For the AC-based calculations, they were 0.3–0.6 kcal mol^{-1} Å^{-1} throughout; this value can thus be considered as the inherent error of the AC-based force computation. For the gradient correction runs, as expected, these RMS errors showed a decreasing tendency together with the decrease of the geometry changes in system 1; the 0.0011 kcal mol^{-1} Å^{-1} convergence criterion was easily fulfilled here.

However, in the problematic cases, the convergence with gradient correction is slower, or the optimization does not reach the stationary point at all (3 out of our 8 test cases fell into the latter category, but our test set was to some extent biased toward the problematic calculations). The energy from the macroiterations often oscillates at much higher values than the converged final energy from the AC-based calculation. A closer look at one of the failed optimizations reveals several important features. In the oscillatory region, the QM geometry is hardly altered in the macroiterations; nevertheless, the MM relaxation takes unusually high number of iterations to converge (if it does at all). The change of the corrected energy during the MM optimization is dominated by the gradient correction contribution, $\langle \mathbf{g}_2^{\text{corr}}, \mathbf{x}_2 - \mathbf{x}_2^{(k)} \rangle$, suggesting that this constant term in the force erroneously pulls away the MM part in some direction. This finding is corroborated by Figure 2, where geometries and exact forces in the affected part of system 2 are shown from three subsequent macroiterations (referred to here as 1st, 2nd, 3rd), together with the change in the force correction from the 1st to the 2nd macroiteration. It is apparent that at one macroiteration, a large force is computed, and the MM optimization is indeed driven in that direction. However, in the next macroiteration, again a large force is computed, at this time in the opposite direction, which then drives the MM optimization back to the previous point. As apparent in the Figure, the forces in the 2nd macroiteration stem almost completely from the fact that the force correction changed significantly from the 1st macroiteration to the 2nd, meaning that the MM optimization in the 1st iteration was driven well beyond the validity of the corrections.

While this behavior seems to be well localized to a small part of system 2, and its absolute effect in the total energy might not always be deleterious, it efficiently impedes the convergence in the microiterative steps. Furthermore, these local instabilities result in relatively large changes in the total energy that are fully unrelated to changes in $\mathbf{x}_1$, thereby interfering with the Hessian update mechanism and trust radius determination operating in system 1, although this latter problem could be alleviated by using a coupled optimization algorithm in the systems 1 and 2.[40]

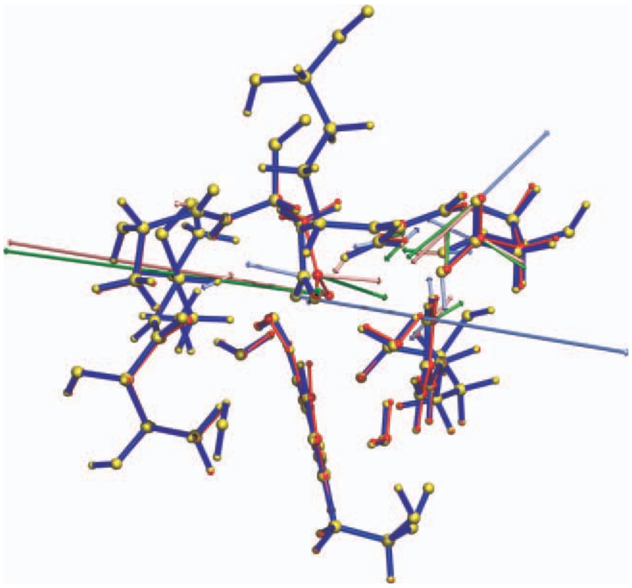

Figure 2. Comparison of geometries of a selected system 2 region of Δ^9 desaturase, obtained from three subsequent macroiterations during an optimization run. The 1st geometry is colored blue, the 2nd is red, and the 3rd is yellow. The ball sizes increase and the stick thicknesses decrease from one iteration to the next to allow better comparison of the structures. Note the difference between the 2nd geometry (red) and the 1st (blue); also note the similarity between the 3rd (yellow) and 1st (blue). Light blue and red arrows show the exact forces computed at the 1st and 2nd macroiteration. Only forces larger than 0.12 kcal mol^{-1} Å^{-1} in absolute value are shown; the two largest ones are scaled by a factor of 0.4 with respect to all others (but are still the largest), to fit within the figure. Green arrows show the difference between the force corrections in the 2nd and 1st iterations (2nd minus 1st). Again, only differences larger than 0.12 kcal mol^{-1} Å^{-1} are shown, and the two largest ones are scaled by 0.4.

Analytical treatment of a one-variable case

To learn more about why the optimization can be driven beyond the validity of the correction in some cases, we will now analyze its behavior analytically in a one-variable case, with the assumption that both the exact QM/MM (E) and the AC-based (E^{AC}) PESs are purely quadratic and have minimum. Let us furthermore assume that E^{AC} does not depend on the geometry where the ACs were determined, i.e., $E^{\text{AC}}(x|x^{(k)}) = E^{\text{AC}}(x)$. Without further loss of generality, we can thus write that

$$E(x) = ax^2, \tag{7a}$$

$$E^{\text{AC}}(x) = b(x-c)^2, \tag{7b}$$

with a, b positive; which yields

$$g(x) = 2ax, \tag{8a}$$

$$g^{\text{AC}}(x) = 2b(x-c) \tag{8b}$$

for the corresponding gradients. Let our starting geometry be $x^{(1)}$. In the first macroiteration, we compute a gradient correction

$$g^{\mathrm{corr}}(x^{(1)}) = g(x^{(1)}) - g^{\mathrm{AC}}(x^{(1)}) = 2ax^{(1)} - 2b(x^{(1)} - c), \tag{9}$$

and we will do the microiterations using the corrected gradient

$$g^{\mathrm{G}}(x) = 2b(x - c) + 2ax^{(1)} - 2b(x^{(1)} - c). \tag{10}$$

The microiterations will now find the minimum $x^{(2)}$ where $g^{\mathrm{G}}(x^{(2)}) = 0$. A short calculation yields that

$$x^{(2)} = (1 - a/b)x^{(1)}, \tag{11}$$

and we can generalize this equation to obtain the result $x^{(n+1)}$ after n macroiterations to be

$$x^{(n+1)} = (1 - a/b)^n x^{(1)}. \tag{12}$$

If this geometric progression is convergent, its limit is 0, which is the location of the minimum on the exact surface $E(x)$. This means that if convergent, our one-variable optimization will correctly end at the right stationary point. However, it is easy to check that convergence holds if and only if $a/2 < b$. This inequality conveys an important message: the AC-based surface must not be too flat. Quantitatively, it must not be more than twice as flat as the exact surface in terms of the second derivative. If it is too flat, the forces estimated by the AC approach will not be able to keep the optimization in the vicinity of the sought minimum against the constant force correction, which will then drive the system to oscillations, in this simple model having ever-increasing magnitude. This is precisely the behavior that we observed above in an actual multidimensional numerical example.

Proposed correction for curvature

In the previous, we have argued that the key reason for the convergence failures of the gradient correction method is the too large difference in the curvature of the exact and approximate PESs. It seems therefore logical to examine a related correction for the Hessian as well and thus use the following energy expression for the microiterations after the kth macroiteration:

$$\begin{aligned} E^{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}) &= E^{\mathrm{AC}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}) + E^{\mathrm{corr}}(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}) \\ &+ \left\langle \mathbf{g}_2^{\mathrm{corr}}(\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}), \mathbf{x}_2 - \mathbf{x}_2^{(k)} \right\rangle \\ &+ \frac{1}{2}\left\langle \mathbf{x}_2 - \mathbf{x}_2^{(k)}, \mathbf{H}_{22}^{\mathrm{corr}}\left[\mathbf{x}_2 - \mathbf{x}_2^{(k)}\right]\right\rangle, \end{aligned} \tag{13}$$

where the matrix $\mathbf{H}_{22}^{\mathrm{corr}}$ could be identified with the difference between the exact and AC-based Hessian matrices, i.e., difference of Hessians for $E(\mathbf{x}_1,\mathbf{x}_2)$ and $E^{\mathrm{AC}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$, evaluated at point $\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)}$. However, there are several things to consider to make this expression useful in practice. First, the number of degrees of freedom in system 2 is large, and therefore the computation of the QM contributions to the exact QM/MM Hessian is prohibitively expensive. Second, for the same reason, even the storage of this full Hessian should be avoided if possible. A third observation is that we still use a truncated power series, and thus $\mathbf{H}_{22}^{\mathrm{corr}}$ should be positive semidefinite. Otherwise, the MM optimization might enter a region where the energy can be lowered significantly by moving in the direction of one of the eigenvectors of $\mathbf{H}_{22}^{\mathrm{corr}}$ corresponding to a negative eigenvalue. In this way, the MM optimization would be again driven beyond the range of validity of the corrections, leading to oscillations.

To fulfill all these requirements, we propose to use a Hessian update method[26] to obtain a cheap estimate of $\mathbf{H}_{22}^{\mathrm{corr}}$ from the coordinates and gradients at the points on the PES already visited at the macroiterations. The exact gradient information for these previous points [$\mathbf{g}_2(\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)})$ for all $i \leq k$] is available from the macroiterations anyway, and the calculation of AC-based gradients for the previous points using the current charges [$\mathbf{g}_2^{\mathrm{AC}}(\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$] is not very expensive. Hessian update methods that guarantee the positive definiteness of the estimated matrix exist, and a low-memory implementation avoids the storage of the full matrix. The number of update steps, corresponding to the number of macroiterations, is typically much smaller than the number of system 2 atoms, which means that curvature information can only be collected for a fraction of the degrees of freedom; however, this should suffice as the problematic behavior seems to only affect localized regions within system 2. We emphasize that this is an untypical usage of these update methods: the resulting estimate of the Hessian is not used to determine which step is to be taken on a PES; instead, it is used to *alter* the approximate PES. Independently, we can still use any algorithm during the microiterations to find the minimum of the resulting E^{H}.

Since taking the difference of Hessians estimated separately for $E(\mathbf{x}_1,\mathbf{x}_2)$ and $E^{\mathrm{AC}}(\mathbf{x}_1,\mathbf{x}_2|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ would not allow to easily keep $\mathbf{H}_{22}^{\mathrm{corr}}$ positive semidefinite, we suggest to estimate the Hessian of the difference surface, $E^{\mathrm{corr}} = E - E^{\mathrm{AC}}$. This matrix is not expected to have any special properties (e.g., is not positive definite in the vicinity of a minimum of E), but the update algorithms simply skip the update steps that would lead to negative eigenvalues in the Hessian and thereby only collect information when E^{AC} is flatter than E, precisely the case where a correction is required. This approach might lead to "positive" curvature information being erroneously retained in the Hessian because even if the optimization proceeds to a different region, it will never be updated by "negative" information. It seems therefore important to drop points that were visited long ago to avoid the accumulation of errors in the Hessian. It is furthermore essential to realize that E^{corr} depends on the coordinates of both system 2 and system 1; therefore, we have to include both $\mathbf{x}_1$ and $\mathbf{x}_2$ as independent variables in the Hessian update algorithm, although during the microiterations, only the "22" block is utilized. The necessary gradients with respect to $\mathbf{x}_1$ ($\mathbf{g}_1$ and $\mathbf{g}_1^{\mathrm{AC}}$) are also available.

Concerning the initial Hessian, the update methods mathematically require positive definite initial matrix and then keep it positive definite. We found that in practice, the Davidon–Fletcher–Powell (DFP) update[81,82] gives good results with a zero initial matrix, which is the most natural choice of initial matrix here, corresponding to the assumption that there is no difference in the curvatures of the exact and AC-based surfaces. We did not attempt to give a formal proof of maintaining the positive semidefiniteness, but this behavior was corroborated empirically by numerical diagonalization of the "22" blocks obtained in our actual test calculations. We note in passing that the more widely used BFGS method is not capable of doing more than a single update step with a zero initial matrix, and therefore it seems to be inappropriate for the present aim.

Formulation and implementation

With respect to a low-memory implementation, a "straightforward" way seemed to be sufficient for our purposes, following Refs. [26] and [29]. To formulate this approach, let us denote by $\mathbf{x}^{(i)}$ the combined $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ coordinates from the *i*th macroiteration, and by $\mathbf{g}^{\mathrm{corr},(i|k)}$ the corresponding combined gradients $\mathbf{g}_1^{\mathrm{corr}}(\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ and $\mathbf{g}_2^{\mathrm{AC}}(\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$. This $\mathbf{g}^{\mathrm{corr},(i|k)}$, as already discussed, is computed with the geometry at the *i*th but using the charges from the current (*k*th) macroiteration. Furthermore, let us introduce

$$\mathbf{s}^{(i)} = \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}, \tag{14a}$$

$$\mathbf{y}^{(i|k)} = \mathbf{g}^{\mathrm{corr},(i|k)} - \mathbf{g}^{\mathrm{corr},(i-1|k)}. \tag{14b}$$

Then, as $\mathbf{H}_{22}^{\mathrm{corr}}$, the Hessian estimate used during the microiterations following the *k*th macroiteration, we will use the "22" subblock of $\mathbf{A}^{(k|k)}$, the final result of *n* DFP update steps using these data, given by the following formulae (T denotes transpose):

$$\mathbf{A}^{(k-n|k)} = 0 \tag{15a}$$

$$\mathbf{A}^{(i|k)} = \left\{ \begin{array}{ll} \mathbf{A}^{(i-1|k)} + \mathbf{v}^{(i|k)}\mathbf{y}^{(i|k)\mathrm{T}} + \mathbf{y}^{(i|k)}\mathbf{v}^{(i|k)\mathrm{T}} & \text{if } \langle \mathbf{y}^{(i|k)}, \mathbf{s}^{(i)} \rangle > 0 \\ \mathbf{A}^{(i-1|k)} & \text{otherwise} \end{array} \right\} \text{ for } i = k-n+1, \ldots, k \tag{15b}$$

with

$$\mathbf{v}^{(i|k)} = \frac{\mathbf{y}^{(i|k)} - \mathbf{A}^{(i-1|k)}\mathbf{s}^{(i)}}{\langle \mathbf{y}^{(i|k)}, \mathbf{s}^{(i)} \rangle} - \frac{\langle \mathbf{y}^{(i|k)} - \mathbf{A}^{(i-1|k)}\mathbf{s}^{(i)}, \mathbf{s}^{(i)} \rangle}{2\langle \mathbf{y}^{(i|k)}, \mathbf{s}^{(i)} \rangle^2}\mathbf{y}^{(i|k)}. \tag{16}$$

In the actual implementation (see Fig. 3), the coordinates $\mathbf{x}^{(i)}$ and exact gradients are stored for the last $n+1$ macroiterations. Before starting the microiterations after the *k*th macroiteration, the AC-based gradients for the previous points $\mathbf{g}^{\mathrm{AC}}(\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)}|\mathbf{x}_1^{(k)},\mathbf{x}_2^{(k)})$ are computed, and the vectors $\mathbf{g}^{\mathrm{corr},(i|k)}$ are evaluated as the difference between the exact and AC-based data. Then, after each macroiteration, *n* update steps are carried out using the above formula, and the resulting $\mathbf{v}^{(i|k)}$ and $\mathbf{y}^{(i|k)}$ vectors are stored. To compute E^{H} or $\mathbf{g}_2^{\mathrm{H}}$, only the product of the "22" block of $\mathbf{A}^{(k|k)}$ with an arbitrary vector is required, which can be obtained in a straightforward way using the "2" blocks of the above determined $\mathbf{v}^{(i|k)}$ and $\mathbf{y}^{(i|k)}$ vectors.

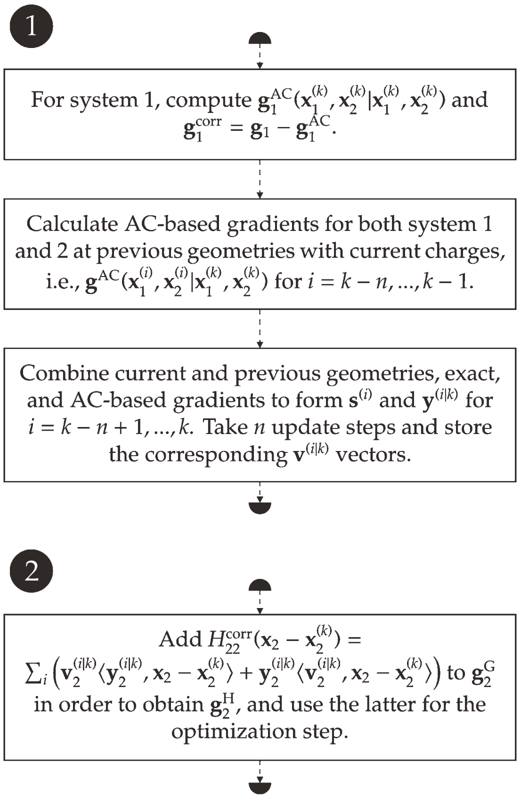
1
For system 1, compute $\mathbf{g}_1^{\mathrm{AC}}(\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}|\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)})$ and $\mathbf{g}_1^{\mathrm{corr}} = \mathbf{g}_1 - \mathbf{g}_1^{\mathrm{AC}}$.
Calculate AC-based gradients for both system 1 and 2 at previous geometries with current charges, i.e., $\mathbf{g}^{\mathrm{AC}}(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}|\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)})$ for $i = k-n, \ldots, k-1$.
Combine current and previous geometries, exact, and AC-based gradients to form $\mathbf{s}^{(i)}$ and $\mathbf{y}^{(i|k)}$ for $i = k-n+1, \ldots, k$. Take n update steps and store the corresponding $\mathbf{v}^{(i|k)}$ vectors.
2
Add $H_{22}^{\mathrm{corr}}(\mathbf{x}_2 - \mathbf{x}_2^{(k)}) = \sum_i \left(\mathbf{v}_2^{(i|k)}\langle\mathbf{y}_2^{(i|k)}, \mathbf{x}_2 - \mathbf{x}_2^{(k)}\rangle + \mathbf{y}_2^{(i|k)}\langle\mathbf{v}_2^{(i|k)}, \mathbf{x}_2 - \mathbf{x}_2^{(k)}\rangle\right)$ to $\mathbf{g}_2^{\mathrm{G}}$ in order to obtain $\mathbf{g}_2^{\mathrm{H}}$, and use the latter for the optimization step.

Figure 3. Additional steps for the curvature correction. These steps are to be inserted into the original program flow at the places marked by black circles with numbers (see Fig. 1); note that $\mathbf{g}_2^{\mathrm{H}}$ is to be used instead of $\mathbf{g}_2^{\mathrm{G}}$ in the microiterative optimization step.

Tests of the new protocol

We implemented the above described scheme into the program system ComQum, by appropriately modifying the force computation routine in Amber and developing a set of separate small programs. Two series of calculations were run, with the maximum number of update steps (*n*) set to 5 and 30, respectively.

We first tested the approach in those cases where the force correction alone provided good results. Here, essentially no effect of the curvature correction was observed on the energies obtained in the macroiterations. A difference was detected in the exact system 2 forces computed in the macroiterations. These forces were in average somewhat larger with the curvature correction scheme; the more so with the larger number of update steps (see Fig. 4). Although in general, we could expect an improvement by the addition of a second-order term in the correction, apparently, the amount of information collected by the update method is not sufficient to yield an overall better estimate of the exact PES. The larger system 2 forces are supposedly the consequence of the inaccuracies of the curvature correction limiting the extent of the relaxation in system 2, which would proceed in these cases appropriately. As a result, with 30 update steps, the necessary number of macroiterations is often higher; nevertheless, convergence is eventually achieved.

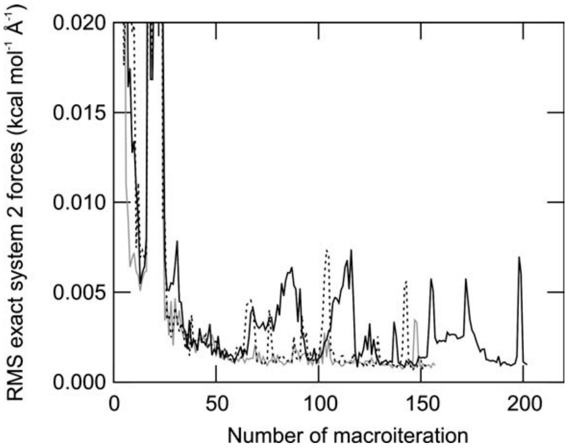

Figure 4. RMS of exact system 2 forces (kcal mol^{-1} Å^{-1}) computed at the macroiterations for a typical optimization run where the force correction itself has no problems. Gray line—only force correction active; black dotted line—curvature correction with five update steps active; black solid line—curvature correction with 30 update steps active. All three calculations converged at the last points shown.

While the above results show that the curvature correction does not provide any advantages for well-behaved optimizations, the situation is opposite for the cases where force correction fails. In all such optimizations we found in our test set, the curvature correction was capable of significantly reducing the exact system 2 forces in the problematic region, and the erroneous oscillatory behavior in the MM region was avoided, using either 5 or 30 update steps. The overall convergence was also restored, and the appropriate minima could always be located (see Fig. 5). These results could be achieved both if the curvature correction was active from the beginning as well as if it was only turned on when the problematic region was already reached. Five update steps typically yielded convergence in somewhat fewer steps. In one case, the optimization with 30 update steps failed in the microiterations following the 323rd macroiteration due to the estimated Hessian developing an exceedingly large eigenvalue; see below the discussion of this issue.

As an additional test, we carried out an optimization in which quadratic restraint forces on some Cartesian coordinates were added for eight system 2 atoms in the exact *E* but not in E^{AC}. This way of implementing restraints is not advantageous from the practical point of restraining system 2 atoms, but it can mimic the effect of E^{AC} being flatter than *E*. We used force constants ranging from 0.1 to 2.0 a.u. ≈ 220–4500 kcal mol^{-1} Å^{-2}. As the largest typical force constants, corresponding to bond stretching modes of light elements, vary around several hundred kcal mol^{-1} Å^{-2}, this test is supposed to represent a worst-case scenario.

From the first attempts, it turned out that the force corrections and oscillations grow exceedingly large here, and it was necessary to implement a damping of the correction to keep the system within its bounds. Namely, eq. (13) was only used unchanged up to a component-wise displacement of 1.0 Å from the geometry at the start of the microiterations; above that displacement, the force and curvature corrections were smoothly damped, reaching zero (meaning no alteration in the gradient and a constant term in the energy) around 2.0 Å. Furthermore, from theoretical considerations, 5 update steps are not sufficient to deal with 8 degrees of freedom, so we tried 15 and 30 steps.

In these test runs, the observed behavior of the energy in the macroiterations was far from being smooth. Apparently, it took several macroiterations for the optimizations to collect the necessary curvature information, but then they were successful in lowering the energy (see Fig. 6). However, the optimization with 30 update steps failed in the microiterations

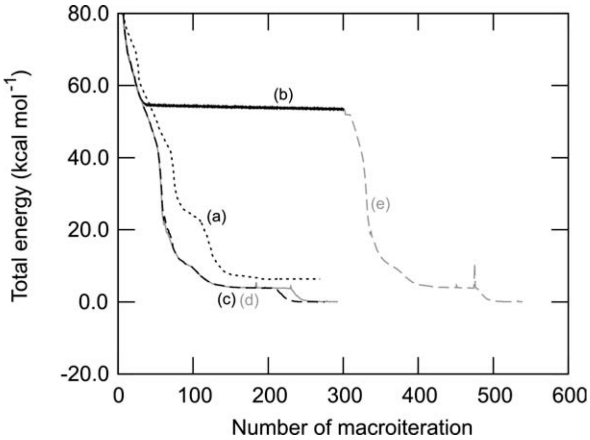

Figure 5. Total energy as a function of the number of the macroiteration for an optimization that is problematic for the force correction. Black dotted line (a), AC-based calculation. Black solid line (b), calculation with force correction (terminated after 300 iterations). Black dashed (c) and gray solid (d) lines, calculations with curvature correction throughout, using 5 and 30 update steps. Grey dashed line (e), calculation with curvature correction with 5 update steps, started at the endpoint of the force-correction-only optimization. The zero value on the energy scale corresponds to the stationary point found in the (c), (d), and (e) runs.

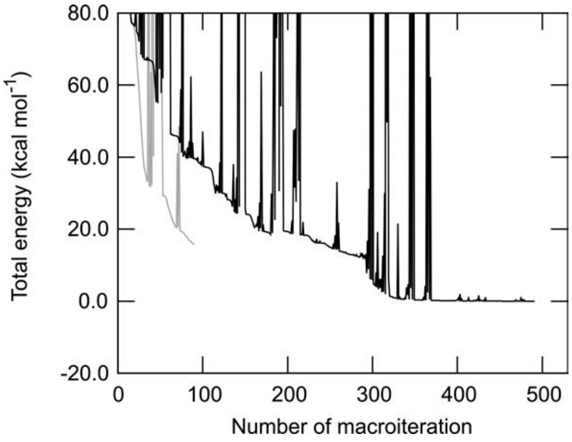

Figure 6. Total energy as a function of the number of the macroiteration for an optimization using restraints for system 2 atoms in the exact *E* but not in E^{AC}. Black and grey lines denote calculations with curvature correction throughout, using 15 and 30 update steps, respectively. The former calculation converged at the last point showed, while the latter was aborted. The zero value on the energy scale corresponds to the stationary point found in the converged calculation.

following the 90th macroiteration, due to the correction Hessian having a very large eigenvalue ($\sim 10^{14}$ kcal mol^{-1} Å^{-2}) and thereby providing exceedingly large forces and displacements during the MM optimization. We found that this large eigenvalue was not produced as a result of numerical problems in a single update step, but it grew into that large value throughout several steps, presumably due to the absence of "negative" updates as discussed earlier. The optimization with 15 update steps achieved the low energy regions slower, but finally it was able to converge to a true stationary point. These findings demonstrate that the method is capable of correcting rather large differences in the curvatures even for eight degrees of freedom. Typical optimizations are expected to perform well with 5 update steps and no damping, while a tuning of these parameters might be necessary for some more difficult cases.

Conclusions

This article is devoted to the analysis and extension of the force correction approach,[47,48] a technique allowing cheap MM microiterations with electronic embedding QM/MM geometry optimizations. The idea lying behind that method is to use a constant additive correction to the MM forces, recomputed exactly at each macroiteration. While very accurate in many cases, convergence problems occasionally arise in this scheme. To tackle this problem, we carried out a detailed numerical and analytical investigation of the convergence behavior and suggested a higher order correction based on an estimation of the Hessian matrix. The results from our work can be summarized as follows:

1. Convergence of the force correction approach is only ensured if the approximate MM potential energy surface is not significantly flatter than the exact one. Roughly speaking and in a purely quadratic model, the corresponding approximate second derivatives should be at least 1/2 of the exact values.

2. If the above condition is not met, the MM region oscillates, apparently due to the correcting forces driving the optimization beyond the range of validity of the correction.

3. A possible solution seems to be the employment of the next higher order correction term; essentially, a curvature correction. The necessary Hessian can be estimated from the information collected in the macroiterations using a Hessian update method (specifically, Davidon-Fletcher-Powell was tested), and then the potential energy surface used in the microiterations can be modified by the addition of a quadratic expression. The additional computational cost of this approach is negligible; only some preparatory MM gradient calculations and simple vector operations are necessary (the new steps were summarized in Fig. 3).

4. The curvature correction was shown to be capable of solving the issues with the investigated problematic optimization runs. A low number of update steps seems to be advantageous to ensure faster convergence and less interference in the problem-free cases. More update steps as well as a damping of the corrections might be necessary in more complicated situations.

In summary, the curvature correction on the basis of a Hessian update method was found to be a viable and cheap approach to improve the convergence properties of the force correction scheme, thereby increasing the stability of the microiterative protocol in electronic embedding QM/MM calculations. Further assessment of the method on a significantly larger number of test cases is, of course, necessary to gain a more complete picture about its effectiveness. Although the implementation was done within ComQum, the presented protocol can be generally applied and easily incorporated into any QM/MM package; the authors are ready to provide programming details or parts of source code that carry out the force and curvature correction procedures.

Acknowledgments

The latest version of the ComQum package was generously provided by Ulf Ryde. The authors thank Jakub Chalupský for sharing his molecular structure visualization program.

Keywords: QM/MM · microiteration · electronic embedding · Hessian update method · convergence

How to cite this article: T. A. Rokob, L. Rulíšek, *J. Comput. Chem.* **2012,** *33,* 1197–1206. DOI: 10.1002/jcc.22951

[1] A. Warshel, M. Karplus, *J. Am. Chem. Soc.* **1972**, *94*, 5612.
[2] A. Warshel, M. Levitt, *J. Mol. Biol.* **1976**, *103*, 227.
[3] R. A. Friesner, V. Guallar, *Annu. Rev. Phys. Chem.* **2005**, *56*, 389.
[4] H. Hu, W. Yang, *Annu. Rev. Phys. Chem.* **2008**, *59*, 573.
[5] H. Lin, D. G. Truhlar, *Theor. Chem. Acc.* **2007**, *117*, 185.
[6] K. E. Ranaghan, A. J. Mulholland, *Int. Rev. Phys. Chem.* **2010**, *29*, 65.
[7] D. Riccardi, P. Schaefer, Y. Yang, H. Yu, N. Ghosh, X. Prat-Resina, P. König, G. Li, D. Xu, H. Guo, M. Elstner, Q. Cui, *J. Phys. Chem. B* **2006**, *110*, 6458.
[8] H. M. Senn, W. Thiel, *Angew. Chem. Int. Ed.* **2009**, *48*, 1198.
[9] H. M. Senn, W. Thiel, *Top. Curr. Chem.* **2007**, *268*, 173.
[10] Y. Tu, A. Laaksonen, *Adv. Quantum Chem.* **2010**, *59*, 1.
[11] R. Zhang, B. Lev, J. E. Cuervo, S. Y. Noskov, D. R. Salahub, *Adv. Quantum Chem.* **2010**, *59*, 353.
[12] G. Monard, X. Prat-Resina, A. González-Lafont, J. M. Lluch, *Int. J. Quantum Chem.* **2003**, *93*, 229.
[13] T. Vreven, K. Morokuma, *Annu. Rep. Comput. Chem.* **2006**, *2*, 35.
[14] N. Bernstein, J. R. Kermode, G. Csányi, *Rep. Prog. Phys.* **2009**, *72*, 026501.
[15] C. R. A. Catlow, S. A. French, A. A. Sokol, J. M. Thomas, *Phil. Trans. R. Soc. A* **2005**, *363*, 913.
[16] M. Sierka, J. Sauer, In Handbook of Materials Modeling; S. Yip, Ed.; Springer: New York, **2005**; p. 241.
[17] D. Bakowies, W. Thiel, *J. Phys. Chem.* **1996**, *100*, 10580.
[18] S. Hayashi, I. Ohmine, *J. Phys. Chem. B* **2000**, *104*, 10678.
[19] S. Ten-no, F. Hirata, S. Kato, *Chem. Phys. Lett.* **1993**, *214*, 391.
[20] N. Ferré, J. G. Ángyán, *Chem. Phys. Lett.* **2002**, *356*, 331.
[21] L. Hu, P. Söderhjelm, U. Ryde, *J. Chem. Theory Comput.* **2011**, *7*, 761.
[22] Y. Hagiwara, T. Ohta, M. Tateno, *J. Phys.: Condens. Matter* **2009**, *21*, 064234.
[23] M. Klähn, S. Braun-Sand, E. Rosta, A. Warshel, *J. Phys. Chem. B* **2005**, *109*, 15645.
[24] H. M. Senn, J. Kästner, J. Breidung, W. Thiel, *Can. J. Chem.* **2009**, *87*, 1322.
[25] H. B. Schlegel, *J. Comput. Chem.* **2003**, *24*, 1514.
[26] J. E. Dennis Jr., J. J. Moré, *SIAM Rev.* **1977**, *19*, 46.
[27] R. Fletcher, C. M. Reeves, *Comput. J.* **1964**, *7*, 149.
[28] J. Nocedal, *Math. Comput.* **1980**, *35*, 773.

[29] R. H. Byrd, J. Nocedal, R. B. Schnabel, *Math Prog.* **1994**, *63,* 129.
[30] J. M. Anglada, E. Besalú, J. M. Bofill, J. Rubio, *J. Math. Chem.* **1999**, *25,* 85.
[31] M. B. Reed, *Int. J. Comput. Math.* **2009**, *86,* 606.
[32] H. P. Hratchian, M. J. Frisch, *J. Chem. Phys.* **2011**, *134,* 204103.
[33] S. R. Billeter, A. J. Turner, W. Thiel, *Phys. Chem. Chem. Phys.* **2000**, *2,* 2177.
[34] Ö. Farkas, H. B. Schlegel, *J. Chem. Phys.* **1998**, *109,* 7100.
[35] Ö. Farkas, H. B. Schlegel, *J. Chem. Phys.* **1999**, *111,* 10806.
[36] K. Németh, O. Coulaud, G. Monard, J. G. Ángyán, *J. Chem. Phys.* **2000**, *113,* 5598.
[37] K. Németh, O. Coulaud, G. Monard, J. G. Ángyán, *J. Chem. Phys.* **2001**, *114,* 9747.
[38] B. Paizs, J. Baker, S. Suhai, P. Pulay, *J. Chem. Phys.* **2000**, *113,* 6566.
[39] J. Baker, D. Kinghorn, P. Pulay, *J. Chem. Phys.* **1999**, *110,* 4986.
[40] T. Vreven, M. J. Frisch, K. N. Kudin, H. B. Schlegel, K. Morokuma, *Mol. Phys.* **2006**, *104,* 701.
[41] F. Maseras, K. Morokuma, *J. Comput. Chem.* **1995**, *16,* 1170.
[42] A. J. Turner, V. Moliner, I. H. Williams, *Phys. Chem. Chem. Phys.* **1999**, *1,* 1323.
[43] R. J. Hall, S. A. Hindle, N. A. Burton, I. H. Hillier, *J. Comput. Chem.* **2000**, *21,* 1433.
[44] X. Prat-Resina, J. M. Bofill, À. González-Lafont, J. M. Lluch, *Int. J. Quantum Chem.* **2004**, *98,* 367.
[45] T. Vreven, K. Morokuma, Ö. Farkas, H. B. Schlegel, M. J. Frisch, *J. Comput. Chem.* **2003**, *24,* 760.
[46] Y. Zhang, H. Liu, W. Yang, *J. Chem. Phys.* **2000**, *112,* 3483.
[47] R. B. Murphy, D. M. Philipp, R. A. Friesner, *Chem. Phys. Lett.* **2000**, *321,* 113.
[48] R. B. Murphy, D. M. Philipp, R. A. Friesner, *J. Comput. Chem.* **2000**, *21,* 1442.
[49] J. Kästner, S. Thiel, H. M. Senn, P. Sherwood, W. Thiel, *J. Chem. Theory Comput.* **2007**, *3,* 1064.
[50] U. Ryde, *J. Comput. Aided Mol. Des.* **1996**, *10,* 153.
[51] U. Ryde, M. H. M. Olsson, *Int. J. Quantum Chem.* **2001**, *81,* 335.
[52] M. Sierka, J. Sauer, *J. Chem. Phys.* **2000**, *112,* 6983.
[53] X. Prat-Resina, À. González-Lafont, J. M. Lluch, *J. Mol. Struct. (Theochem)* **2003**, *632,* 297.
[54] J. Kästner, H. M. Senn, S. Thiel, N. Otte, W. Thiel, *J. Chem. Theory Comput.* **2006**, *2,* 452.
[55] Y. Hagiwara, T. Ohta, M. Tateno, *J. Phys.: Condens. Matter* **2009**, *21,* 064234.
[56] J. Gao, *J. Phys. Chem.* **1992**, *96,* 537.
[57] T. N. Truong, E. V. Stefanovich, *Chem. Phys. Lett.* **1996**, *256,* 348.
[58] I. Tuñón, M. T. C. Martins-Costa, C. Millot, M. F. Ruiz-López, J. L. Rivail, *J. Comput. Chem.* **1996**, *17,* 19.
[59] T. J. Evans, T. N. Truong, *J. Comput. Chem.* **1998**, *19,* 1632.
[60] E. Cubero, F. J. Luque, M. Orozco, J. Gao, *J. Phys. Chem. B* **2003**, *107,* 1664.
[61] S. Martí, V. Moliner, I. Tuñón, I. H. Williams, *J. Phys. Chem. B* **2005**, *109,* 3707.
[62] S. Martí, V. Moliner, I. Tuñón, *J. Chem. Theory Comput.* **2005**, *1,* 1008.
[63] F. Melaccio, M. Olivucci, R. Lindh, N. Ferré, *Int. J. Quantum Chem.* **2011**, *111,* 3339.
[64] Turbomole V6.3.1, University of Karlsruhe and Forschungszentrum Karlsruhe GmbH, 1989–2007, TURBOMOLE GmbH, since 2007; available from http://www.turbomole.com, **2011**.
[65] D. A. Case, T. A. Darden, T. E. Cheatham III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, K. M. Merz, B. Wang, D. A. Pearlman, M. Crowley, S. Brozell, V. Tsui, H. Gohlke, J. Mongan, V. Hornak, G. Cui, P. Beroza, C. Schafmeister, J. W. Caldwell, W. S. Ross, P. A. Kollman, Amber 8, University of California, San Francisco, **2004**.
[66] P. Császár, P. Pulay, *J. Mol. Struct.* **1984**, *114,* 31.
[67] T. Helgaker, *Chem. Phys. Lett.* **1991**, *182,* 503.
[68] C. G. Broyden, *Not. Am. Math. Soc.* **1969**, *16,* 670.
[69] R. Fletcher, *Comput. J.* **1970**, *13,* 317.
[70] D. Goldfarb, *Math. Comput.* **1970**, *24,* 23.
[71] D. F. Shanno, *Math. Comput.* **1970**, *24,* 647.
[72] M. Srnec, T. A. Rokob, J. K. Schwartz, Y. Kwak, L. Rulíšek, E. I. Solomon, Inorg. Chem. 2012, accepted.
[73] M. Srnec, U. Ryde, L. Rulíšek, *Faraday Discuss.* **2011**, *148,* 41.
[74] V. Klusák, C. Bařinka, A. Plechanovová, P. Mlčochová, J. Konvalinka, L. Rulíšek, J. Lubkowski, *Biochemistry* **2009**, *48,* 4126.
[75] M. Srnec, F. Aquilante, U. Ryde, L. Rulíšek, *J. Phys. Chem. B* **2009**, *113,* 6074.
[76] A. D. Becke, *Phys. Rev. A* **1988**, *38,* 3098.
[77] J. P. Perdew, *Phys. Rev. B* **1986**, *33,* 8822.
[78] A. Schäfer, H. Horn, R. Ahlrichs, *J. Chem. Phys.* **1992**, *97,* 2571.
[79] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, R. Ahlrichs, *Chem. Phys. Lett.* **1995**, *240,* 283.
[80] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, P. A. Kollman, *J. Am. Chem. Soc.* **1995**, *117,* 5179.
[81] W. C. Davidon, Variable Metric Method for Minimization, Argonne National Laboratory Report ANL-5990, **1959**.
[82] R. Fletcher, M. J. D. Powell, *Comput. J.* **1963**, *6,* 163.

Received: 2 December 2011
Revised: 11 January 2012
Accepted: 16 January 2012
Published online on 17 February 2012