Microiterative QM/MM Optimisation for Materials Chemistry
A dCSE Project

Thomas W. Keal
STFC Daresbury Laboratory

31st May 2013

Abstract

Microiterative optimisation methods have been implemented in the DL-FIND optimisation library for use in combined quantum mechanical/molecular mechanical (QM/MM) calculations using the ChemShell computational chemistry environment. Microiterative techniques reduce the number of expensive QM optimisation steps required by relaxing the MM environment fully after every QM step. An electrostatic potential fit is used to approximate the MM region during the environmental relaxation. Microiterative optimisation has been implemented for minimisation, transition state optimisation using the P-RFO and dimer methods, and reaction path optimisation using the nudged elastic band method. The microiterative techniques will be used to run large-scale heterogeneous catalysis simulations more efficiently on HECToR.

Contents

1 Introduction

The ChemShell computational chemistry environment [1, 2] is used to perform combined quantum mechanical (QM) and molecular mechanical (MM) calculations. In a QM/MM calculation, accurate (but computationally expensive) QM methods are used to treat the reactive site of a chemical system while a cheaper classical force field is used to describe the surrounding environment, where a description of the electronic structure is not required. In this way the overall computational cost is reduced compared to a full QM calculation without sacrificing overall accuracy.

The QM/MM approach is particularly useful in the study of heterogeneous catalysis. ChemShell supports embedded cluster calculations [2, 3] in which the bulk material or surface is represented by a finite MM cluster model, optionally with additional point charges to mimic the effect of bulk electrostatics. The active site is modelled by a high-level QM calculation with electrostatic embedding, where the cluster environment is represented by point charges in the QM Hamiltonian.

On HECToR the GAMESS-UK code [4, 5] is typically used for the QM calculations and GULP for the MM calculations [6], both of which are linked in directly to ChemShell as libraries for high parallel performance. The DL-FIND geometry optimisation library within ChemShell [7, 8] can be used to characterise reactions taking place on the surface by optimising reactants, products, reaction paths and transition states.

The aim of this distributed Computational Science and Engineering (dCSE) project is to improve the performance of ChemShell for large-scale geometry optimisation by implementing microiterative techniques in DL-FIND. The computational time of a QM/MM optimisation is usually dominated by the quantum mechanical component (100s of atoms), even though the surrounding MM region is typically much larger (1000s of atoms). In a standard geometry optimisation each step requires a QM and MM evaluation at the new geometry. This means that the geometries of the QM and MM regions have to relax at the same rate. In a microiterative optimisation scheme [9], the system is divided into an inner region containing (at a minimum) the QM atoms, and an outer region containing the rest of the system. After each optimisation step of the inner region (the ‘macroiterative’ cycle), the outer region is fully optimised (the ‘microiterative’ cycle). By optimising in this way, the number of QM evaluations is reduced significantly at the cost of increasing the number of MM evaluations of the outer region. As MM evaluations are usually much cheaper, this reduces the overall computational time.

The microiterative QM/MM minimisation scheme implemented in DL-FIND follows an earlier implementation in the HDLCOpt module in ChemShell [9]. HDLCOpt was designed specifically for biological systems and is not suitable for materials chemistry. It has been largely superseded by DL-FIND, but until this project DL-FIND lacked microiterative functionality. This project was therefore both a refactoring exercise (transferring functionality from the HDLCOpt module to DL-FIND, with the intention of deprecating HDLCOpt entirely) and an optimisation of the code (speeding-up DL-FIND calculations for a range of algorithms). The DL-FIND implementation also goes beyond the HDLCOpt version by extending the microiterative approach to a wider range of optimisation methods.

9 months of development effort were deployed on the project over a 12 month period from April 2012 to March 2013. The code development objectives were:

- Implementation of the microiterative optimisation framework in DL-FIND, involving the restructuring of the code to introduce macroiterative and microiterative cycles.
- Modification of the ChemShell/DL-FIND interface to support microiterative optimisation.
- Microiterative energy minimisation, in which the outer region is minimised after each step taken in the inner region.
- Microiterative transition state optimisation using the P-RFO and dimer methods, in which the inner region is optimised to a saddle point while the surrounding environment is relaxed to a minimum.
- Microiterative reaction path optimisation, where the nudged elastic band method (involving multiple geometries along the path) is applied to the inner region, while the environment of each image is relaxed.

All methods have been successfully implemented and tested on HECToR Phase 3 for the case of hydrogen dissociation over Li-doped MgO, as discussed below.

2 Implementation and performance

2.1 Energy minimisation

The microiterative minimisation scheme in DL-FIND follows program logic similar to the original HDLCOpt implementation. The program flow of a microiterative minimisation with electrostatic embedding in DL-FIND is as follows:

1. Perform a full QM/MM calculation (macroiteration).
2. Fit point charges to the electron density in the QM region using the electrostatic potential (ESP). This is used to approximate the QM-MM interaction during the microiterations without requiring a QM calculation.

3. If not the first cycle, test convergence of the macroiteration (rejecting the step if the energy has increased). Terminate if convergence criteria are met.
4. Take a step in the inner region using the gradient from step 1.
5. Enter the microiterations:
    (a) Calculate an ESP-approximated energy and gradient.
    (b) Test convergence of the microiteration (if not the first). Continue to step 6 if converged. Reject the step if the approximated energy has increased and reduce the step size
    (c) Take a step in the outer region using the ESP-approximated gradient
    (d) Continue at step 5a
6. Continue with macroiterations at step 1.

Note that in the original HDLCOpt implementation the macroiterative step energy (i.e. the exact energy before the inner region step is taken) was used as a reference energy to test acceptance of the first microiterative step. This is acceptable for a minimisation because the energy should decrease in both the macroiterations and microiterations. However, the DL-FIND implementation is designed to work for methods where the energy may not decrease after a macroiterative step, so a reference ESP-approximated energy is calculated before any outer region steps are taken.

The main DL-FIND loop cycles once per energy evaluation. A flag (`glob%imicroiter`) is used to switch between macroiterations and microiterations.

At the macroiterative step, a full QM/MM calculation is carried out, as in a standard optimisation, by a call to the ChemShell QM/MM driver, giving an energy E^0 and gradient $\mathbf{g}^0$. In the same call the QM code is then used to calculate the electrostatic potential on a van der Waals surface around the QM region. A least-squares fit algorithm in ChemShell is used to fit point charges at the QM atom positions to reproduce the ESP. A second call to ChemShell is then made to recalculate the QM/MM energy and gradient using the point charge representation for the QM atoms, giving E^0_{ESP} and $\mathbf{g}^0_{\mathrm{ESP}}$.

Following Ref. [9], the corrected gradient $\mathbf{g}^1$ and energy E^1 used in the microiterations are:

$$\mathbf{g}^1 = \frac{\partial E^1_{\mathrm{ESP}}}{\partial \mathbf{x}_{\mathrm{outer}}} + \mathbf{g}^0_{\mathrm{corr}} \tag{1}$$

and

$$E^1 = E^1_{\mathrm{ESP}} + E^0 - E^0_{\mathrm{ESP}} + \mathbf{g}^0_{corr} \cdot (\mathbf{x}^1_{\mathrm{outer}} - \mathbf{x}^0_{\mathrm{outer}}) \tag{2}$$

where

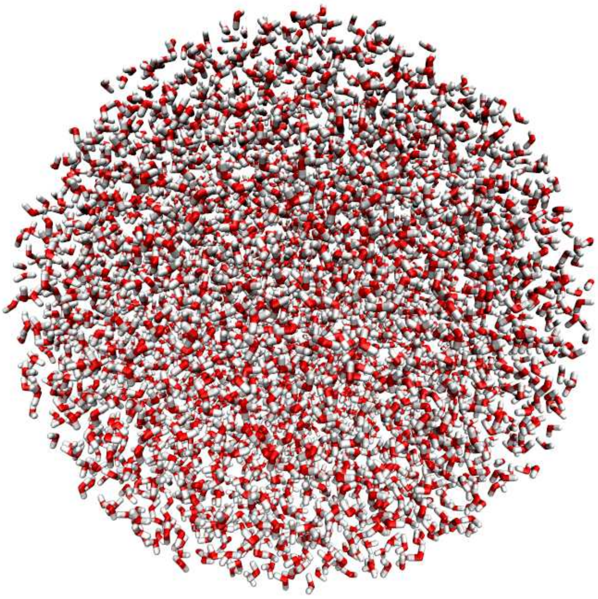
$$\mathbf{g}^0_{\mathrm{corr}} = \left.\frac{\partial E^0}{\partial \mathbf{x}_{\mathrm{outer}}}\right|_{\mathbf{x}^0_{\mathrm{outer}}} - \left.\frac{\partial E^0_{\mathrm{ESP}}}{\partial \mathbf{x}_{\mathrm{outer}}}\right|_{\mathbf{x}^0_{\mathrm{outer}}} \tag{3}$$

where superscript 1 indicates the current coordinates and 0 the respective coordinates from the last macroiterative calculation. The expressions for the corrected energies and gradients ensure a smooth connection between the microiterative and macroiterative cycles, as in the case of identical geometries the equations reduce to the exact answer. In DL-FIND the corrections are actually calculated for the full set of coordinates (not just the outer coordinates), but the energy expression collapses to the outer coordinates because the inner region is fixed, and the gradient correction for the inner coordinates is not used by the microiterative optimiser.

DL-FIND supports multiple instances of the low memory BFGS (L-BFGS) minimisation algorithm, and two instances are used during a microiterative minimisation. The first L-BFGS instance is used for macroiterative steps and covers all coordinates, although only inner region steps are permitted during macroiterations. The reason it covers all coordinates and not just the inner region is because the gradients on all atoms can then be taken into account when calculating the inner region step, and therefore the outer region does not have to be completely converged during the microiterative cycles for the optimisation to succeed. The second L-BFGS instance is initialised at the start of every microiterative loop and covers the outer region coordinates only. The corrected QM/MM energy and gradient is used to relax the environment while the inner region is kept fixed. The second L-BFGS data is kept in a separate set of arrays in the `microiter` module.

Unlike HDLCOpt, DL-FIND supports multiple coordinate systems and therefore special care must be taken during the microiterative cycles that the correct subset of coordinates are used. When the Cartesian coordinates used externally are transformed to the internal list of coordinates used by the optimiser, the new list is ordered into inner region coordinates followed by outer region coordinates. This makes it straightforward to operate over a single region. When Cartesians are used internally, this mapping is trivial. The delocalised coordinate (DLC) and total connection (TC) coordinate systems, however, may not be used, as they are delocalised over the entire system and therefore cannot be decomposed by region. Hybrid delocalised coordinates (HDLCs) and HDLC-TC may be used as they are only delocalised over discrete subsets of coordinates called residues, providing that no residues cross the inner and outer regions. The input is checked to ensure that this does not happen. In the HDLC case the list of internal coordinates is ordered by residue within each region.

Figure 1: Water sphere used to test the DL-FIND implementation of microiterative QM/MM minimisation.



A number of keywords were added to the DL-FIND input to specify that a microiterative optimisation should be carried out, the list of atoms or residues in the inner region, whether to carry out an ESP fit (a full QM calculation can also be used during the microiterations if requested), and the maximum number of microiterative cycles to carry out before taking another macroiterative step.

The microiterative minimisation scheme was tested for correctness on a QM glycine molecule solvated in water described by a classical forcefield. To check that the implementation gave similar performance improvements to the original HDLCOpt implementation, it was further tested on the water sphere system described in Ref. [9]. This consists of 4404 water molecules cut from equilibrated bulk water with a radius of 25 Å, as shown in Figure 1. Three water molecules in the centre of the sphere were defined as the QM region and inner microiterative region. QM calculations were performed with the MNDO semi-empirical QM package using the AM1 Hamiltonian. Active regions of radius 5, 10, 15 and 20 Å were defined, within which the water molecules were described using the TIP3P force field. The atoms outside the active region were frozen during the optimisation.

Results for the optimisations are shown in Table 1. Note that the DL-FIND tests were carried out in Cartesian coordinates (and therefore without bond constraints), as microiterative optimisation with HDLC coordinates

Table 1: Comparison of number of optimisation steps required for convergence of a QM/MM water sphere optimisation using microiterative (QM=macro cycles and MM=micro cycles) and standard minimisation algorithms. DOF = degrees of freedom. ΔE is the difference in converged energy between the microiterative and standard runs.

Active radius/Å	DOF	Microiterative opt. QM	MM	Standard opt. QM	ΔE / kcal/mol
5	2493	114	2018	639	-2
10	6966	176	3230	2038	+17
15	14382	311	4682	1372	-3
20	26136	207	3949	1990	+10

was not implemented at the time. The number of degrees of freedom is therefore greater than in Ref. [9] and so the number of optimisation steps are not directly comparable. However, very similar trends are observed. As in the previous HDLCOpt tests, the optimisation takes place over a complex energy surface with many close lying minima and so different runs may take different paths and reach slightly different converged energies. For the same reason the number of cycles does not necessarily increase monotonically with the number of degrees of freedom. Nevertheless, the improvement in performance from standard to microiterative optimisation is clear. Over all sizes of active region, a reduction in the number of QM evaluations is seen of between 5 and 12 times. An illustration of the significant speed-up of convergence in terms of number of QM steps is shown in Figure 2. With a small, semi-empirical QM region this will not result in an overall reduction of computational time, but it demonstrates the potential for significant savings in typical production QM/MM calculations where the QM calculation dominates.

Microiterative optimisations with HDLC coordinates and HDLC constraints were subsequently tested using the glycine-water system. The converged HDLC minima were in good agreement with those found using Cartesian coordinates.

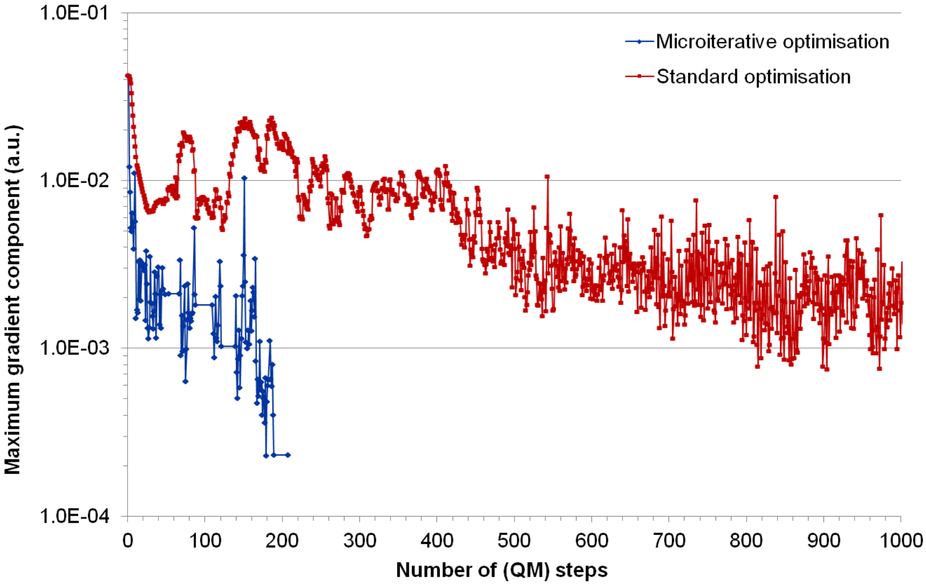
2.2 Transition state optimisation

In the second stage of the work, microiterative optimisation was extended to two transition state optimisation algorithms: partitioned rational function optimisation (P-RFO) and the dimer method.

2.2.1 P-RFO

The standard P-RFO optimisation method in DL-FIND requires a full Hessian calculation and is therefore not appropriate for systems with many degrees of freedom. The HDLCOpt implementation of P-RFO was more

Figure 2: Maximum gradient component (one of four convergence criteria) during microiterative and standard optimisation of a water sphere with an active region of 20 Å.



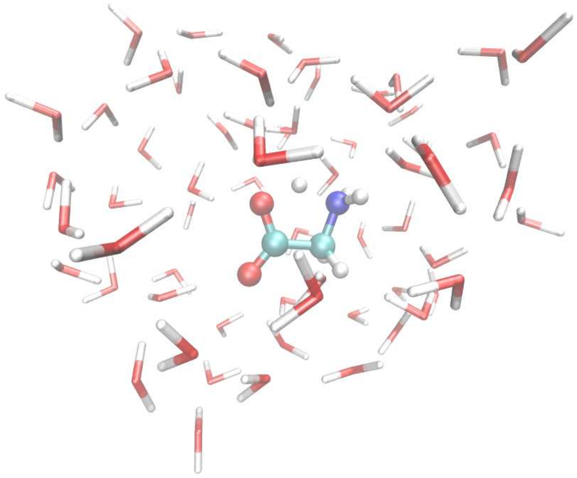
flexible in that a core region could be optimised to a saddle point while the environment was relaxed using the L-BFGS optimisation algorithm. This scheme is similar to microiterative minimisation, as the gradient of the environment must be optimised to zero after every P-RFO step [10].

In DL-FIND, microiterative P-RFO optimisation has been implemented by splitting the system into an inner region optimised by P-RFO and an outer region optimised by L-BFGS. The outer region is optimised using the routines established for microiterative minimisation. As in microiterative minimisation, an ESP fit can be used to relax the environment after each P-RFO step providing that the QM region is contained wholly within the inner region.

The P-RFO and Hessian evaluation routines were modified so that they operated on the subset of coordinates defined in the inner region. This differs from the minimisation scheme, where the macroiterative optimiser operates over the whole system, because in transition state optimisation the inner and outer regions are optimised differently. It is therefore important in the P-RFO case that the environment is fully optimised after each step, particularly towards the end of the optimisation, as the macroiteration steps will not be accurate otherwise [10]. In practice this means that a low maximum for the number of microiterations should not be used.

In standard P-RFO, 'soft' modes are identified corresponding to rota-

Figure 3: Optimised transition state of solvated glycine and its zwitterionic form.



tions and translations of the system which are projected out so they do not hinder convergence. In microiterative optimisation, however, this is not appropriate as rotations or translations of the inner region with respect to the outer region may be necessary for correct convergence. The soft mode algorithms are therefore disabled for microiterative optimisation.

The microiterative P-RFO method was tested for correctness using the same solvated glycine system used for minimisation. In the P-RFO case the transition state between glycine and its zwitterionic form was optimised (Figure 3).

The QM region was consisted of the glycine molecule and was calculated with MNDO using the AM1 Hamiltonian. The inner (macroiterative) region was defined to be equal to the QM region. The outer environmental region consisted of 62 water molecules described with the TIP3P force field, of which 13 were active. ESP fitting was used during the microiterations.

Although standard and microiterative transition state optimisations are not directly comparable (because the environment is part of the saddle point optimisation in the standard case), a well-chosen inner region should result in good agreement between the two, all other things being equal, because transition states are generally fairly localised. In the solvated glycine case there was an excellent agreement in optimised energies between the standard and microiterative runs, and between optimisation in Cartesian and HDLC coordinates, giving confidence that the implementation is correct.

2.2.2 Dimer method

The dimer method [11] is an alternative approach to transition state optimisation in DL-FIND which does not require evaluation of a Hessian and so is more suited to large-scale systems. A dimer optimisation involves the calculation of two close-lying points on the potential energy surface (the ‘dimer’) separated by a fixed distance. In each optimisation cycle, the dimer is first rotated around its midpoint to minimise the sum of the energies of the endpoints, which aligns the dimer along the softest vibrational mode (the mode of the Hessian with the lowest eigenvalue). The dimer is then translated to maximise the energy along the direction of this mode while minimising the energy in all directions perpendicular to it.

In the standard dimer optimisation method implemented in DL-FIND, it is possible to weight individual coordinates when calculating the dimer vector and the rotational force on the dimer. By setting a weight of zero, coordinates can be excluded from the rotation step altogether, so that they are simply minimised during the translation step. The excluded coordinates act like an environment which is relaxed during the optimisation.

It is therefore natural to extend the weighting mechanism to implement a microiterative form of dimer optimisation. As in microiterative P-RFO, the system is divided into an inner region that is optimised to a saddle point and an outer region that is relaxed. The coordinates list is ordered first by image, then by region and finally by residue if applicable.

The dimer weights of the outer region coordinates are set to zero so that they take no part in the transition state optimisation. A macroiterative iteration consists of a rotation of the inner region dimer, followed by an inner region midpoint translation step. The microiterative loop then takes place, which relaxes the environment as in microiterative P-RFO. In total three L-BFGS optimisers are used (for dimer rotation, dimer translation and environment minimisation respectively) compared to two for the standard optimisation. The environment minimisation corresponds to the midpoint of the dimer and so only one set of coordinates is used in the microiterative cycles.

The microiterative dimer method was tested on the same solvated glycine transition state optimisation used for the P-RFO method. The same QM and MM setup was used and an ESP fit was used during the microiterations. Excellent agreement was seen for the optimised transition state energy between standard and microiterative dimer optimisations, between Cartesian and HDLC optimisations, and between microiterative dimer and microiterative P-RFO.

2.3 Reaction path optimisation

The nudged elastic band (NEB) method [12,13] is implemented in DL-FIND to optimise a reaction path between known reactant and product geometries. In a NEB optimisation, an initial path between the endpoints is guessed (either as a linear transit or via extra geometries specified by the user), and multiple geometry images are defined along it. The energies and gradients of each image are calculated independently and then an overall NEB gradient is calculated, consisting of a spring force acting on a local tangent between each of the images, and the true force acting perpendicular to the local tangent. By optimising the NEB gradient the images describe the minimum energy reaction path.

In the DL-FIND implementation of NEB, it is possible as in the dimer method to weight coordinates so that they take a greater or lesser part in the calculation of the spring force. By setting the weight to zero, no spring force is calculated for a given coordinate and so it is simply minimised rather than taking part in the NEB force. In this way, an environmental region can be defined that is relaxed independently of a NEB inner region.

The microiterative NEB optimisation has been implemented as an extension to this weighting procedure. As in the dimer case, the system is split into an inner region where the standard NEB gradient is calculated and an outer region where the weights are set to zero. In a macroiterative step the inner region moves according to the NEB gradient, which is followed by a microiterative loop where the outer region is relaxed fully. The outer region gradient is also defined over all images, but as there is no spring force it is effectively a series of independent minimisations (albeit indirectly linked due to the fixed inner regions). In the DL-FIND implementation however, the set of outer region gradients are collated and treated as one optimisation problem using a single L-BFGS optimiser, as this is more convenient than optimising each independently. In the microiterative loop, the RMS and maximum step and gradient criteria are calculated as normal, while the average of the energies of all the images is used as the energy convergence criterion.

The microiterative scheme must also be able to handle frozen images and the climbing image. Images are frozen after the force on them falls below a certain threshold. In the microiterative case, the outer region coordinates of this image - which should be fully relaxed after the previous step - are also frozen at this point. The spawning of a climbing image introduces one further set of outer region coordinates, which are simply minimised along with all the other outer region images.

ESP fitting is also more complicated with NEB. In order to perform the microiterative cycle correctly, an ESP fit must be carried out for every NEB image during the macroiterative step. These are stored in separate ChemShell fragments indexed by image number, and then retrieved as appropriate during the microiterative gradient calculations.

The microiterative NEB algorithm was tested using solvated glycine, optimising the reaction path between the minimised standard and zwitterionic geometries. The same QM setup was used as for the other algorithms, and multiple image ESP fitting was used as described above. Image freezing and a climbing image were both used in the test. Although, as before, the standard and microiterative optimisations are not directly comparable because the microiterative environment is minimised, the energy and structure of the optimised climbing image indicated that the microiterative optimisation had converged to the same endpoint as the standard optimisation.

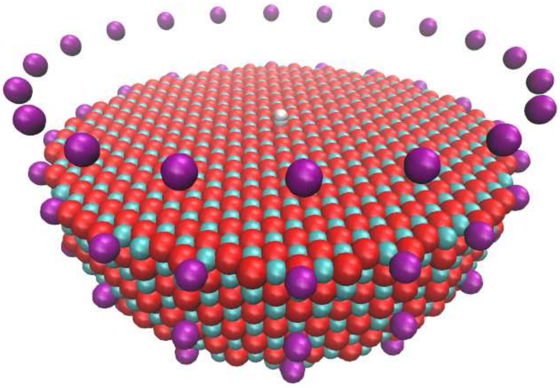
It should be noted that the exclusion of the environment from the NEB spring forces carries the risk that the path taken by the environment from reactant to product will become discontinuous during the optimisation, as the minimisation of each environmental image is independent. One possible solution to this problem is to perform a series of NEB optimisations with restraint terms on the environmental degrees of freedom which are gradually relaxed, as suggested by Xie et al. [14]. However, this would substantially increase the overall cost of the NEB run. In our test cases we have not observed any problems with discontinuities in the environment, so we have not chosen to implement this procedure, but it would be quite straightforward to do so if it became clear over a wider set of test cases that it is necessary.

2.4 Shell model optimisation

In a standard electrostatic embedding QM/MM calculation, the classical MM atoms polarise the QM region by entering the QM Hamiltonian as point charges, but the QM atoms do not in turn polarise the MM region. A shell model force field can be used to enable polarisation of the MM region. Each MM atom consists of a core and a shell, which are connected by a spring force. Some of the MM charge is assigned to the shell, which can move under the influence of the electrostatic potential of the QM region. Because the polarisation of the QM region is in turn influenced by the positions of the shells, each QM/MM energy and gradient evaluation consists of a series of QM and MM calculations which are iterated to self-consistency.

In the context of microiterative optimisation, no special changes need to be made to the algorithms in DL-FIND, as the shells are treated as invisible from the point of view of the optimiser and only the final converged QM/MM energy and gradient is required. However, changes to ChemShell were made to support efficient microiterative shell model optimisation with ESP fitting. Whenever a full QM/MM energy and gradient evaluation is carried out, the polarisation of QM and MM regions is iterated to self-consistency as normal. This is the case for macroiterations, and also microiterations if ESP fitting is not used (although the QM region is fixed if it is in the inner region, the polarisation of the QM region by the MM region will not be fixed). However,

Figure 4: QM/MM model of hydrogen dissociation over Li-doped MgO. Hydrogen shown in white, oxygen in red, magnesium in cyan, and point charges in purple. The lithium atom is below the surface.



if the QM region is approximated by ESP fitted charges, the polarisation of the QM region is also fixed for the duration of the microiterations. The ESP charges are then fed directly to GULP so that the shells can be relaxed in a single GULP calculation (taking into account how the forces on the shells change as they move, rather than feeding in the forces themselves, which would need to be iterated over).

Shell model microiterative minimisation was tested for correctness on the MgO surface defect example used to test the HDLCOpt implementation [9]. The ESP fitting procedure worked well and the optimisation converged to the same minimum as HDLCOpt.

In order to evaluate the performance of all the shell model microiterative optimisation algorithms on HECToR, a test system involving a reaction is required. The reaction we have used is the dissociation of hydrogen over Li-doped MgO, as shown in Figure 4. The test system consists of a cluster of 6349 atoms plus 87 point charges around the edge of the cluster to mimic the effect of the bulk material's electrostatics. The QM region consisted of 33 atoms in the centre of the cluster, including the hydrogen molecule, the lithium dopant, 5 oxygen atoms, and 25 magnesium atoms, of which 4 were treated at a full QM level and 21 were modelled using pseudopotentials to form a boundary around the QM region, which ensures that electron density does not leak into the surroundings. There were 834 active atoms optimised in each run. For the microiterative calculations, the inner region was defined to be the same as the QM region (including boundary atoms).

The QM calculations were performed using GAMESS-UK using the B97-

Table 2: Comparison of energies and number of optimisation steps required for convergence of the hydrogen/Li-doped MgO system for microiterative and standard minimisation algorithms. NEB energies are those of the highest energy image (no climbing image).

Opt. type	E(std)/h	E(mic)/h	Standard opt. Cycles	Microiterative opt. Macro	Micro
H_2 min	-1842.669	-1842.669	27	3	22
HH min	-1842.689	-1842.689	50	4	50
P-RFO	-	-1842.662	-	22	95
dimer	-1842.662	-1842.662	47	35	309
NEB	-1842.664	-1842.666	44	32	31

2 functional with a Def2 TZVP basis (274 spherical harmonic basis functions). The calculations were unrestricted with a multiplicity of 2. The MM calculations were performed in GULP using a shell model potential.

The ESP fitting procedure was found to fail for this system, with unphysical charges being assigned to many of the QM atoms. This is probably due to the indistinct nature of the QM-MM boundary, in which some MM oxygen atoms are within the region enclosed by the QM pseudopotential magnesium atoms. A number of remedies were attempted, including fitting only to the full QM atoms while constraining the charges on the boundary QM atoms, but they did not resolve the problem. Another approach would be to develop pseudopotentials for the oxygen atoms so the boundary can be well-defined, but this would be new research and beyond the scope of the current dCSE project. We therefore performed the tests without ESP fitting, which still gives a good indication of the savings that can be expected in terms of macroiterative cycles when the microiterative algorithms are applied. The tests also confirm that all the algorithms can be used with shell model polarised forcefields.

The tests consisted of minimisation of the physisorbed and dissociated hydrogen reaction endpoints, P-RFO and dimer optimisation of the transition state for the dissociation, and NEB optimisation of the full reaction path. The results are given in Table 2. Unlike the case of the water sphere, the minimisation runs here are much more constrained structurally and so it is not surprising that both the standard and microiterative optimisations result in highly consistent energies. In both the molecular hydrogen and dissociated cases, the number of macroiterative cycles is dramatically lower (roughly ten fold less) than the number of cycles in the standard optimisation. This is likely a function of the good starting geometries and regular nature of the system, and this level of improvement is unlikely to be seen on more complex energy surfaces. Nevertheless it is a very positive result for the microiterative algorithm.

For transition state optimisation, the improvements are more modest. It was not possible to run a full P-RFO optimisation due to the computational cost of calculating the Hessian that would be required. However, the good agreement in energies between microiterative P-RFO and both standard and microiterative dimer optimisations shows that the algorithms are working well. The number of macroiterative cycles for the dimer method is lower than the number of standard cycles, albeit not by as large a number as for the minimisations. In this case P-RFO performed better than the dimer method both in terms of macroiterative and microiterative cycles, but it should be emphasised that this may not be true in general, as there are many factors that influence the number of cycles required to converge. Furthermore, even the microiterative version of P-RFO still requires substantial computational expense to calculate a Hessian.

The excellent agreement in energies between the standard and microiterative dimer methods suggests that the microiterative degrees of freedom are spectators to the reaction, which is consistent with the results of the NEB optimisation of the reaction path. 10 NEB images were used (with no freezing of images or a climbing image), and a relatively loose convergence criterion was employed. This, together with task-farming parallelism (10 workgroups), was necessary to make the NEB calculation tractable on HECToR. The convergence criterion likely explains the small difference in optimised energies between the highest energy standard and microiterative images, rather than relaxation of the environment which would have shown up in the transition state optimisations. The reduction in cycles is similar to that of the dimer method, although much fewer microiterations are required in the case of NEB, which indicated that the initial interpolated reaction path is of very good quality. This is not surprising for a very constrained system with many spectator degrees of freedom.

Overall, the results show that all the microiterative algorithms are functioning correctly for shell model optimisation of materials systems on HECToR, and that microiterative relaxation of the environment does reduce the number of macroiterations required in the test system.

3 Conclusion and outlook

Microiterative QM/MM optimisation methods have been implemented as planned for minimisation, transition state optimisation (P-RFO and dimer), and reaction path optimisation (NEB) in DL-FIND and ChemShell. The algorithms have been tested using a water sphere model, glycine solvated in water, and on HECToR for the case of hydrogen dissociation on an Li-doped MgO cluster, the last using GAMESS-UK for the QM calculations and GULP for the MM calculations with a shell model polarised potential.

In the particular case of the ionic embedding model, where the QM/MM boundary can be indistinct, the ESP fitting procedure was found not to be robust. New research, beyond the scope of this project, is required to improve the ESP fit algorithm or else redefine the boundary so that there is a clear distinction between the QM and MM regions. For all other systems tested, the ESP fitting procedure works well and the microiterative algorithms are fully functional and ready for use. The new methods will be made available to users in ChemShell version 3.6, which is scheduled for release in summer 2013.

4 Acknowledgements

I would like to thank Chris Downing for providing the H_2/LiO/MgO system from which the microiterative shell model optimisation tests were derived. I would also like to thank Richard Catlow, Alexey Sokol, Aron Walsh and Paul Sherwood for their help in preparing the dCSE grant proposal and Phil Ridley and Ian Bush for their suggested improvements. Thanks also to Johannes Kästner for helpful discussions about the details of the microiterative algorithms.

This project was funded under the HECToR Distributed Computational Science and Engineering (CSE) Service operated by NAG Ltd. HECToR A Research Councils UK High End Computing Service - is the UK's national supercomputing service, managed by EPSRC on behalf of the participating Research Councils. Its mission is to support capability science and engineering in UK academia. The HECToR supercomputers are managed by UoE HPCx Ltd and the CSE Support Service is provided by NAG Ltd. http://www.hector.ac.uk

References

[1] ChemShell, a computational chemistry shell. See www.chemshell.org

[2] Sherwood P, de Vries AH, Guest MF, Schreckenbach G, Catlow CRA, French SA, Sokol AA, Bromley ST, Thiel W, Turner AJ, Billeter S, Terstegen F, Thiel S, Kendrick J, Rogers SC, Casci J, Watson M, King F, Karlsen E, Sjøvoll M, Fahmi A, Schäfer A, Lennartz C (2003) J Mol Struct (THEOCHEM) 632:1–28

[3] Sokol AA, Bromley ST, French SA, Catlow CRA, Sherwood P (2004) Int J Quantum Chem 99:695–712

[4] GAMESS-UK, a package of ab initio programs. See http://www.cfs.dl.ac.uk/gamess-uk/

[5] Guest MF, Bush IJ, van Dam HJJ, Sherwood P, Thomas JMH, van Lenthe JH, Havenith RWA, Kendrick J (2005) Mol Phys 103:719–747

[6] Gale JD, Rohl AL (2003) Mol Simul 29:291–341

[7] DL-FIND, a geometry optimiser for quantum chemical and QM/MM calculations. See ccpforge.cse.rl.ac.uk/projects/dl-find/

[8] Kästner J, Carr JM, Keal TW, Thiel W, Wander A, Sherwood P (2009) J Phys Chem A 113:11856–11865

[9] Kästner J, Thiel S, Senn HM, Sherwood P, Thiel W (2007) J Chem Theory Comput 3:1064–1072

[10] Turner AJ, Moliner V, Williams IH (1999) Phys Chem Chem Phys 1:1323–1331

[11] Kästner J, Sherwood P (2008) J Chem Phys 128:014106

[12] Henkelman G, Jónsson H (2000) J Chem Phys 113:9978–9985

[13] Henkelman G, Uberuaga BP, Jónsson H (2000) J Chem Phys 113:9901–9904

[14] Xie L, Liu H, Yang W (2004) J Chem Phys 120:8039–8052