Molecular Physics, Vol. 104, Nos. 5–7, 10 March–10 April 2006, 701–714

Taylor & Francis
Taylor & Francis Group

Geometry optimization with QM/MM methods II: Explicit quadratic coupling

T. VREVEN*†, M. J. FRISCH†, K. N. KUDIN‡, H. B. SCHLEGEL§ and K. MOROKUMA¶
†Gaussian, Inc., USA
‡Princeton University, USA
§Wayne State University, USA
¶Emory University, USA

(*Received 4 May 2005; in final form 10 June 2005*)

Geometry optimization of large QM/MM systems is usually carried out by alternating a second-order optimization of the QM region using internal coordinates ('macro-iterations'), and a first-order optimization of the MM region using Cartesian coordinates ('micro-iterations'), until self-consistency. However, the neglect of explicit coupling between the two regions (the Hessian elements that couple the QM coordinates with the MM coordinates) often interferes with a smooth convergence, while the Hessian update procedure can be unstable due to the presence of multiple minima in the MM region. A new geometry optimization scheme for QM/MM methods is proposed that addresses these problems. This scheme explicitly includes the coupling between the two regions in the QM optimization step, which makes it quadratic in the full space of coordinates. Analytical second derivatives are used for the MM contributions, with O(N) memory and CPU requirements (where N is the total number of atoms) by employing direct and fast multipole methods. The explicit coupling improves the convergence behaviour, while the Hessian update is stable since it no longer involves MM contributions. Examples show that the new procedure performs significantly better than the standard methods.

1. Introduction

Geometry optimization is an integral part of studies employing electronic structure methods, because the location of minima and saddle points is essential for the understanding of reaction mechanisms [1]. The most sophisticated geometry optimization methods have computational cost for each iteration that scales cubically with the size of the system, while the simpler optimization methods scale linearly, but may require many more cycles to converge. This implies that there is a trade-off between the cost of the geometry optimization and the cost of the required energy and derivatives calculations, which means that the choice of geometry optimization method depends on the size of the system and the computational method employed. In practice, the cubic-scaling optimization methods are used for small or medium sized systems treated with expensive computational methods, while the less-expensive methods are used for large systems treated with semi-empirical or molecular mechanics methods.

For QM/MM methods, geometry optimization methods have been developed that do not belong in the categories above [2–8]. These methods exploit particular features of QM/MM schemes, such as the large difference in computational cost between QM and MM methods and the nature of the coupling between the regions. The key concept is to use different types of optimizers for the QM region and for the MM region. First, the MM region is fully minimized, while keeping the QM region fixed, using an optimization scheme appropriate for large systems treated with inexpensive computational methods. This is followed by a single geometry step in only the QM region, using an optimization scheme that is appropriate for small systems treated with expensive methods. These two steps are alternated until both regions are converged.

*Corresponding author. Email: thom@gaussian.com

Molecular Physics
ISSN 0026–8976 print/ISSN 1362–3028 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00268970500417846

The single step in the QM region is referred to as macro-iteration, and the full minimization in the MM region is referred to as micro-iteration.

Despite the apparent success of such QM/MM optimization methods, there are significant drawbacks caused by the particular Hessian update and the neglect of second-order coupling between the regions. This results in a strongly compromised convergence behaviour. In the remainder of this Introduction we will review hybrid methods, conventional geometry optimization schemes, and QM/MM geometry optimization schemes, allowing us to identify problems. In section 2 we present our improved QM/MM geometry optimization scheme, of which the incorporation into the Gaussian suite of programs [9] is discussed in section 3. In section 4 we apply the method to several small systems, as well as a large biochemical system.

1.1. *Hybrid methods*

Hybrid methods combine two or more computational levels in one calculation, and allow the accurate study of the chemistry of very large systems. Examples of such techniques are the QM/MM methods [10–12], which combine a quantum mechanical (QM) method with a molecular mechanics (MM) method, and the more general ONIOM scheme [4, 13–26], which can combine any number of molecular orbital (MO) methods, as well as MM methods. The region of the system where the chemical process takes place, for example bond breaking and formation, is treated with an appropriately accurate method, while the remainder of the system is treated at the lower level. The successes of hybrid methods in the study of chemical problems have been subject of several reviews [27–34]. A variety of QM/MM methods have been reported in recent years, which differ in the details, but are conceptually quite similar. In the present work we will outline our geometry optimization methods using the ONIOM(QM:MM) framework, but it should be noted that they are easily adapted to other QM/MM implementations.

In a two-layer ONIOM(QM:MM) calculation, the total energy of the system is obtained from three independent calculations.

$$E^{ONIOM(QM:MM)} = E^{real,MM} + E^{model,QM} - E^{model,MM} \tag{1}$$

The 'real' system contains all the atoms, and is calculated only at the MM level. The 'model' system contains the part of the system that is treated at the QM level. Both QM and MM calculations need to be carried out for the model system. Unlike most other QM/MM schemes the ONIOM energy is defined as an extrapolation, which allows for combinations of QM with QM, as well as QM with MM.

When there is covalent bonding between the QM and MM regions, we saturate the dangling bonds with link atoms. The model system now includes both the QM layer and the link atoms. The latter are usually hydrogen atoms, but can be any atom that mimics the substituent group. The position of a link atom is obtained by scaling the distance between the centre to which it is connected, and the atom that it substitutes [15]. Because the positions of the atoms in the model system are defined in terms of the atoms in the real system, the potential energy surface, and therefore geometry optimization, is well defined. The ONIOM gradient is obtained from:

$$\begin{aligned} \boldsymbol{g}^{ONIOM} &= \frac{\partial E^{ONIOM}}{\partial \mathbf{x}} = \frac{\partial E^{real,MM}}{\partial \mathbf{x}} \\ &\quad + \frac{\partial E^{model,QM}}{\partial \mathbf{x}^m}\mathbf{J} - \frac{\partial E^{model,MM}}{\partial \mathbf{x}^m}\mathbf{J} \\ &= \frac{\partial E^{real,MM}}{\partial \mathbf{x}} + \frac{\partial E^{model,QM}}{\partial \mathbf{x}} - \frac{\partial E^{model,MM}}{\partial \mathbf{x}} \end{aligned} \tag{2}$$

The Jacobian $\mathbf{J}$ converts the coordinate system for the model system $\mathbf{x}^m$ to the coordinate system for the real system $\mathbf{x}$. Other properties can be expressed in a similar fashion. In the current work the Hessian plays a central role. Because the Jacobian $\mathbf{J}$ does not depend on $\mathbf{x}$, there are no first-order terms in the expression for the ONIOM Hessian.

$$\mathcal{H}^{ONIOM} = \mathcal{H}^{real,MM} + \mathcal{H}^{model,QM} - \mathcal{H}^{model,MM} \tag{3}$$

where

$$\mathcal{H}^{model,level} = \frac{\partial^2 E^{model,level}}{\partial \mathbf{x}\partial \mathbf{x}} = \mathbf{J}^t \frac{\partial^2 E^{model,level}}{\partial \mathbf{x}^m \partial \mathbf{x}^m}\mathbf{J} \tag{4}$$

Besides allowing for QM/QM combinations and three or more layers, there are several differences between ONIOM and other QM/MM methods. First, many QM/MM schemes use frozen orbitals instead of link atoms to treat the boundary [35–37]. Through the required parameterization and the use of *p* and *d* orbitals, frozen orbitals can describe a more accurate charge density than hydrogen link atoms. The number of studies that directly compare link atom methods with frozen orbital methods is limited [37, 38], but it seems generally accepted that the latter can provide a better description of the boundary. However, one of the attractive features of the link atom scheme is exactly that it does not require parameterization, which ensures

that any electronic structure method can be used in the ONIOM scheme without additional work. It is for this generality that we have not implemented frozen orbitals in our scheme. It must be noted that also the 'character' of link atoms can be adjusted by changing its electro-negativity using a shift operator [39], or adjustment of the parameters if a semi-empirical method is used [40]. Besides frozen orbitals and link atoms, which are used by the majority of QM/MM schemes, some implementations use other techniques, such as pseudo potentials [41, 42].

Second, in the standard ONIOM(QM:MM) scheme, the coupling between the two regions is treated completely by molecular mechanics calculations. This is usually referred to as 'mechanical embedding'. Many other QM/MM schemes employ 'electronic embedding', which includes the charge distribution of the MM region in the QM wave function optimization [43]. Electronic embedding allows for polarization of the wave function, and provides a more accurate description of the electrostatic interaction between the two regions. Although we have extended the basic ONIOM scheme to include electronic embedding [18, 23], we will only consider mechanical embedding in the present paper. In the discussion section, we comment on the generalization of our new geometry optimization scheme to other types of QM/MM embedding.

1.2. *First- and second-order geometry optimization methods*

Most geometry optimization schemes are either first order or second order. In first-order schemes, the gradient is used to determine the next point along the optimization path. The simplest example is steepest descent (SD), in which a step is taken in the direction of the negative of the gradient. Other first-order schemes, such as conjugate gradient (CG), also use information obtained along the optimization path. The computational cost to determine the step scales more or less linearly with the size of the system, which makes first-order schemes the method of choice for large systems studied with inexpensive computational methods that evaluate the energy and gradient with linear cost as well, usually empirical or semi-empirical methods.

Second-order schemes use gradient ($\mathbf{g}$) as well as Hessian ($\mathbf{H}$) information to determine the step. Assume a second-order truncation of the potential surface.

$$E(\mathbf{q}+\Delta\mathbf{q}) \approx E(\mathbf{q}) + \mathbf{g}\Delta\mathbf{q} + \frac{1}{2}\Delta\mathbf{q}^t\mathbf{H}\Delta\mathbf{q} \qquad (5)$$

The Newton–Raphson (NR) equations determine the direction and magnitude of the step $\Delta\mathbf{q}$ by requiring the gradient to become zero in the second-order approximation of the surface:

$$\Delta\mathbf{q} = -\mathbf{H}^{-1}\mathbf{g} \qquad (6)$$

The analytical evaluation of the Hessian is expensive for accurate computational methods. Therefore, second-order optimizers usually employ a Hessian update mechanism. This does require an initial guess for the Hessian, which can be obtained analytically, at a lower (less expensive) computational level, or empirically.

However, NR optimization leads to a saddle point if the Hessian is not positive definite. It is also prone to 'overstep' the minimum on flat potential surfaces, or take steps that are too large when the optimization is far from convergence. These problems are addressed by the rational function optimization (RFO) method [44], which controls the step size, and always takes a step downhill, even when the Hessian is not positive definite. Close to convergence, RFO takes steps similar to NR, while far from convergence the steps are similar to those in SD schemes. For minimizations, $\Delta\mathbf{q}$ is obtained from the eigenvector with the lowest eigenvalue of the augmented Hessian:

$$\begin{bmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{bmatrix}\begin{bmatrix} \Delta\mathbf{q} \\ 1 \end{bmatrix} = \lambda\begin{bmatrix} \Delta\mathbf{q} \\ 1 \end{bmatrix} \qquad (7)$$

The second-order methods perform generally much better than first-order methods. However, from equations (6) and (7), it follows that storage and either diagonalization or inversion of the Hessian (or augmented Hessian) is required. This is trivial for small systems, but becomes a bottleneck for systems of thousands or tens of thousands of atoms. A variety of direct methods have been developed that make this part of the scheme scale more favourable, such as Hessian update mechanisms that do not require full storage [45–54], and fast methods for solving the linear equations [55]. Despite these developments, the geometry optimization of very large systems is still usually carried out with first-order methods.

1.3. *Coordinate systems*

In electronic structure methods, the nuclear derivatives of the energy are evaluated in the Cartesian coordinate space. Cartesian coordinates, however, are strongly coupled, and the rate of convergence is improved by carrying out the geometry optimization in redundant internal coordinates [56–62]. In our program we use simple redundant coordinates, which consist of all

the bond stretches, angle bends, and dihedral angles that are present in the system. To carry out an optimization in redundant internals, the derivatives of the energy need to be transformed to the internal coordinate space, while the resulting geometry displacement needs to be transformed back to Cartesian coordinates. The formalism and notation in reference [59] will be followed in the present work.

The internal redundant and Cartesian coordinates are contained in vectors $\mathbf{q}$ and $\mathbf{x}$, respectively. The transformation of the gradient vector in internal coordinates $\mathbf{g} = \partial E/\partial \mathbf{q}$ to the gradient vector in Cartesian coordinates $\mathcal{G} = \partial E/\partial \mathbf{x}$, can be written as

$$g = \mathbf{B}^t \mathbf{g} \tag{8}$$

$\mathbf{B}$ is the Wilson B matrix [63], which contains the relation between the Cartesian and internal displacements. $\mathbf{B}$ is sparse and trivial to compute:

$$B_{ij} = \frac{\partial q_i}{\partial x_j} \tag{9}$$

For the transformation of the Cartesian gradient to internal coordinate gradient, we need $\partial x_i/\partial q_i$, which is obtained using the generalized inverse $\mathbf{G}^- = (\mathbf{B}\mathbf{B}^t)^-$:

$$\mathbf{g} = \mathbf{G}^- \mathbf{B} g \tag{10}$$

The evaluation of $\mathbf{G}^-$ represents the computational bottleneck in this scheme. The computational cost for the inversion scales cubically with the number of variables, which causes the geometry optimization in redundant internals to be impractical for large systems. Although methods have been proposed to alleviate this bottleneck [6, 55, 64–70], optimization in Cartesian coordinates seems still the best choice for very large systems treated with inexpensive computational methods.

The infinitesimal displacement from internal coordinates to Cartesian coordinates can be written as

$$d\mathbf{q} = \mathbf{B} d\mathbf{x} \tag{11}$$

Since the transformation is curvilinear for finite displacements, the geometry displacement needs to be obtained by iteratively solving equation (11).

In second-order schemes that use analytical second derivatives, the Hessian needs to be transformed as well. By differentiating equation (8), we obtain $\mathbf{H} = \mathbf{B}^t \mathbf{H} \mathbf{B} + \mathbf{B}'^t \mathbf{g}$, which we can use for the expression of the Hessian in internal coordinates:

$$\mathbf{H} = \mathbf{G}^- \mathbf{B} (\mathcal{H} - \mathbf{B}'^t \mathbf{g}) (\mathbf{G}^- \mathbf{B})^t \tag{12}$$

$\mathbf{B}'$ denotes the derivative of the $\mathbf{B}$ matrix, with elements defined as $B'_{ijk} = \partial q_i/\partial x_i \partial x_k$, and can be computed analytically. The gradient contribution $\mathbf{B}'^t \mathbf{g}$ contains the second-order relation between the infinitesimal displacement of internal coordinates and Cartesian coordinates.

Finally, we need to modify the gradient and Hessian to account for possible constraints and the redundancies in the coordinate system. The constraints and redundancies are projected from the gradient and Hessian using projector $\mathbf{P}$, and assigned large eigenvalues α. Note that the number of coordinates in the geometry optimization is not reduced. Other implementations may define a new coordinate system that explicitly excludes the redundancies and constrained variables, and use that for the geometry optimization.

$$\tilde{\mathbf{g}} = \mathbf{P}\mathbf{g} \tag{13}$$

$$\tilde{\mathbf{H}} = \mathbf{P}\mathbf{H}\mathbf{P} + \alpha(\mathbf{1} - \mathbf{P}) \tag{14}$$

The second-order RFO optimization in redundant internal coordinates with an updated Hessian is the most commonly used scheme for relatively small systems. However, for large systems typically encountered in fields such as material science and biochemistry, the bottlenecks discussed in the previous paragraphs prohibit the use of these schemes. First, the evaluation and storage of the Hessian, whether analytical or updated, scales at least quadratically with the size of the system. Second, the evaluation of $\mathbf{G}^-$ scales cubically in CPU, while the storage is quadratic. The solution of the RFO or NR equations scales cubically in CPU as well.

1.4. *QM/MM geometry optimization*

For QM/MM optimizations, the bottlenecks related to the coordinate transformation and the solution of the RFO equations can be avoided by employing a hybrid first/second-order scheme [2–8]. As mentioned earlier, this procedure alternates micro-iterations in the MM region with a macro-iteration of the QM region, until both regions are converged. For the micro-iterations, we use a simple, first-order method in Cartesian coordinates, avoiding the need for coordinate transformations and matrix inversions or diagonalizations. Because the QM region is kept fixed, only energy and gradient evaluations at the MM level of theory are needed. Note that during the micro-iterations, we use the QM/MM gradient as defined in equation (2), and therefore optimize on the 'true QM/MM potential surface'. In the subsequent macro-iteration, after the

MM region is fully minimized, we carry out a single geometry optimization step in the QM region. For this step we use a second-order procedure in redundant internal coordinates, and keep the MM region fixed. The alternating macro-step and micro-optimization is repeated until all forces are zero. Separating the optimizations in this manner avoids the bottlenecks associated with each optimization scheme.

Note that the QM step is formally a step in the full QM/MM (mixed internal/Cartesian) coordinate space, with the assumptions that there is no coupling between the regions, and that the forces in the MM region are converged to zero before the macro-step and that after the macro-step the MM coordinates rearrange to the condition of zero forces on these atoms. The NR and RFO equations then hold, and yield zero or undefined displacement for the MM region:

$$\begin{aligned}\begin{bmatrix}\Delta\mathbf{q}_{NR}^{QM}\\ \Delta\mathbf{q}_{NR}^{MM}\end{bmatrix} &= -\begin{bmatrix}\mathbf{H}^{QM} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}^{QM}\\ \mathbf{0}\end{bmatrix}\\ &= -\begin{bmatrix}(\mathbf{H}^{QM})^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}\begin{bmatrix}\mathbf{g}^{QM}\\ \mathbf{0}\end{bmatrix}\end{aligned} \tag{15}$$

$$\Delta\mathbf{q}_{NR}^{QM} = -(\mathbf{H}^{QM})^{-1}\mathbf{g}^{QM};\ \Delta\mathbf{q}_{NR}^{MM} = \mathbf{0} \tag{16}$$

$$\begin{bmatrix}\mathbf{H}^{QM} & \mathbf{0} & \mathbf{g}^{QM}\\ \mathbf{0} & \mathbf{1} & \mathbf{0}\\ (\mathbf{g}^{QM})^t & \mathbf{0}^t & 0\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}_{RFO}^{QM}\\ \Delta\mathbf{q}_{RFO}^{MM}\\ 1\end{bmatrix} = \lambda\begin{bmatrix}\Delta\mathbf{q}_{RFO}^{QM}\\ \Delta\mathbf{q}_{RFO}^{MM}\\ 1\end{bmatrix} \tag{17}$$

$$\begin{bmatrix}\mathbf{H}^{QM} & \mathbf{g}^{QM}\\ (\mathbf{g}^{QM})^t & 0\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}_{RFO}^{QM}\\ 1\end{bmatrix} = \lambda\begin{bmatrix}\Delta\mathbf{q}_{RFO}^{QM}\\ 1\end{bmatrix} \tag{18}$$

However, there are several problems with the standard macro/micro-iteration scheme as outlined above. First, the Hessian that is updated corresponds to the full QM/MM energy, and uses the QM/MM gradients, but involves only the redundant internal variables. Thus the Hessian is well defined, but the standard update procedures are not. As an illustration, consider the following situation: During the optimization process, the MM region (or more precisely, the region that is not described by the internal coordinates), finds a lower minimum. This switch may change forces on an internal coordinate. Now in the Hessian update algorithm, it is assumed that a change in these forces is always the result of a change in the internal coordinates. The consequence is that spurious force constants are introduced. In fact, this problem can corrupt the Hessian so badly that it has a completely wrong curvature, and re-initializing the Hessian is the only way to continue the geometry optimization. One can use update schemes that force the Hessian to be positive definite to avoid this problem to some extent, but this is done at the expense of accuracy in the Hessian and still leads to poor convergence. The second problem in the standard micro-iteration scheme involves the neglect of coupling between the QM region and the MM region. This follows from the structure of the Hessian in equations (15) and (17), and clearly compromises the convergence behaviour, often leading to oscillations in the macro and micro-steps. Third, for equations (15)–(18) to hold, the forces of the MM region must be zero, which requires very tight convergence criteria for the optimization of the MM region. For these reasons, the convergence behaviour of the standard macro/micro-iteration scheme is not as good as one usually experiences with conventional second-order methods. The overall performance is still improved compared to the alternative of using a conventional first-order optimization scheme, but it is clear that there is much room for improvement.

Addressing the problems outlined in the previous paragraphs, we can define a set of requirements for a stable and practical QM/MM macro optimization step. First, the Hessian update must be well defined. Second, we need to include the coupling between the MM and QM regions. This coupling will be completely described by MM contributions, preferably analytically and exact. To make the scheme feasible for systems with very large MM regions, the inclusion of the coupling must scale linearly in both memory and CPU. Third, we want to be able to incorporate all the techniques common in regular second-order optimization schemes, such as trust radius update mechanisms and linear searches, and to apply these to a full step involving all the coordinates.

2. Methods

As outlined in the previous paragraphs, common QM/MM geometry optimization schemes do not include second-order coupling between QM coordinates and MM coordinates, and suffer from a numerically instable Hessian update. In this section we outline the methods that allow us to explicitly include the coupling between the QM region and the MM region, and carry out a numerically stable Hessian update. In our method

we solve the RFO equations for the full QM/MM space of coordinates.

$$\begin{aligned}\begin{bmatrix} \mathbf{H}^{ONIOM} & \mathbf{g}^{ONIOM} \\ (\mathbf{g}^{ONIOM})^t & 0 \end{bmatrix}\begin{bmatrix} \Delta\mathbf{q}^{ONIOM} \\ 1 \end{bmatrix} \\ = \begin{bmatrix} \mathbf{H}^{QM} & \mathbf{H}^{coupling} & \mathbf{g}^{QM} \\ \mathbf{H}^{coupling} & \mathbf{H}^{MM} & \mathbf{g}^{MM} \\ (\mathbf{g}^{QM})^t & (\mathbf{g}^{MM})^t & 0 \end{bmatrix}\begin{bmatrix} \Delta\mathbf{q}^{QM} \\ \Delta\mathbf{q}^{MM} \\ 1 \end{bmatrix} \\ = \lambda\begin{bmatrix} \Delta\mathbf{q}^{QM} \\ \Delta\mathbf{q}^{MM} \\ 1 \end{bmatrix}\end{aligned} \tag{19}$$

Because we want our method to be applicable to very large systems, we cannot assume that the complete Hessian can be stored in memory, nor can we perform operations with cost scaling quadratically or cubically with the size of the system. We therefore employ direct methods. We use the iterative Davidson diagonalization algorithm to solve the RFO equations [71]. The algorithm uses the matrix only as an operator in forming products with vectors, and hence does not require storage of the Hessian, but only of a few vectors. Below we outline how we can evaluate the Hessian vector product with O(N) scaling in both memory and CPU time. For the QM contribution to the QM/MM Hessian (resulting from $\mathbf{H}^{model,QM}$ in equation (3)) we can use either an update mechanism, or we can calculate the contribution analytically. For the MM contributions, resulting from $\mathbf{H}^{real,MM}$ and $\mathbf{H}^{real,MM}$, we always use analytically evaluated second derivatives. This implies that both $\mathbf{H}^{MM}$ and $\mathbf{H}^{coupling}$ in equation (19) are included analytically, because these blocks do not have contributions from $\mathbf{H}^{model,QM}$. As coordinate system, we continue to use redundant internal coordinates for the QM region and Cartesian coordinates for the MM region. In addition, we allow parts of the MM region to be defined as rigid fragments. After we outline the techniques that make solving equation (19) feasible, we give details about our implementation.

2.1. *Coordinate system*

The system is divided up into several parts, which are each treated with a different type of coordinate in the geometry optimization. Region I includes the atoms that are treated with redundant internal coordinates. This region is usually identical to the QM/MM model system (or QM region), but may be larger. In this text, for simplicity, we assume the QM region and the link atoms to be treated with redundant internals. The second region C consists of the MM atoms that are optimized with Cartesian coordinates. Third, we allow groups of MM atoms to maintain their relative position during the optimization, i.e. we treat them as rigid fragments. For clarity we include only two fragments in our equations, K and L. Last, the system can contain a frozen MM region, denoted by F. The total Cartesian coordinate vector can be written in terms of the different regions.

$$\mathbf{x}^t = [\mathbf{x}_I^t \quad \mathbf{x}_C^t \quad \mathbf{x}_K^t \quad \mathbf{x}_L^t \quad \mathbf{x}_F^t] \tag{20}$$

The vector that contains the coordinates for the geometry optimization can be written in terms of the different types of variables.

$$\mathbf{q}^t = [\mathbf{q}_I^t \quad \mathbf{q}_C^t \quad \mathbf{q}_{I'}^t \quad \mathbf{q}_K^t \quad \mathbf{q}_L^t] \tag{21}$$

$\mathbf{q}_I$ are the redundant internal coordinates that describe region I, and $\mathbf{q}_C$ the Cartesian coordinates that describe region C. $\mathbf{q}_K$ describe the rotation and translation of rigid fragment K, and $\mathbf{q}_{I'}$ describe in the same way the rotation and translation of the internal coordinate region. We assume that the rigid fragments have at least three atoms, and are not linear. We need to include $\mathbf{q}_{I'}$ in the geometry optimization because the energy of the region described by internal coordinates is not invariant to translation and rotation if there are frozen atoms in the MM region.

The $\mathbf{q}_I$ coordinates are the same as in conventional geometry optimization discussed in section, and the $\mathbf{q}_C$ coordinates are identical to $\mathbf{x}_C$. We now define the coordinates used for the rigid fragments. The position and orientation of a fragment are described by three translational ($\mathbf{q}_{K^\varphi}$) and three rotational ($\mathbf{q}_{K^\theta}$) coordinates:

$$\mathbf{q}_K^t = [\mathbf{q}_{K^\varphi}^t \quad \mathbf{q}_{K^\theta}^t] = [\varphi_x^K \quad \varphi_y^K \quad \varphi_z^K \quad \theta_x^K \quad \theta_y^K \quad \theta_z^K] \tag{22}$$

The Cartesian coordinates of centre i, $\mathbf{x}_{K,i} = [x_{K,i} \quad y_{K,i} \quad z_{K,i}]^t$ in this rigid fragment are related to the translational and rotational coordinates by

$$\mathbf{x}_{K,i}(\mathbf{q}_K) = \mathbf{O}_K + S_K^\varphi \mathbf{q}_{K^\varphi} + \mathbf{T}^x\mathbf{T}^y\mathbf{T}^z[\mathbf{x}_{K,i}(0) - \mathbf{O}_K] \tag{23}$$

$\mathbf{O}$ is the origin (usually the centre of mass) of the rigid fragment, and $\mathrm{x}_{K_i}(0)$ denotes the initial Cartesian coordinates of the rigid fragment. The rotation matrices T contain the variables $\mathbf{q}_{K^\theta}^t = [\theta_x^K \ \theta_y^K \ \theta_z^K]$:

$$\mathbf{T}^x(\theta_x^K) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(s_\theta^K\theta_x^K) & \sin(s_\theta^K\theta_x^K) \\ 0 & -\sin(s_\theta^K\theta_x^K) & \cos(s_\theta^K\theta_x^K) \end{bmatrix} \tag{24}$$

The translational and rotational variables can be scaled, with factors s_K^{φ} and s_θ^K respectively, to make the gradients of similar order of magnitude as the internal and Cartesian coordinates. The values of the scaling factors depend on the internal geometry of the fragment. In our current implementation we do not scale the translational variables, and use $s_\theta^K = (\sum_i |x_{K,i}(0) - \mathbf{O}_K|)^{-1/2}$ for the rotational variables. This ensures stability for both small and very large rigid fragments.

2.2. *Transformation of the gradient*

For the transformation of the gradient in Cartesian coordinates, $\mathbf{g}$, to the gradient in coordinates for the geometry optimizer, $\mathbf{g}$, we define the matrix $\mathbf{D}$:

$$\mathbf{g} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} g = \mathbf{D} g \tag{25}$$

$\mathbf{D}$ is sparse because only the Cartesian coordinates of a certain region affect the geometry optimization variables for the same region. The non-zero blocks are as follows. A coordinate that appears as a Cartesian coordinate in the geometry optimization does not need to be transformed, thus:

$$\mathbf{D}_{CC} = \frac{\partial \mathbf{x}_C}{\partial \mathbf{q}_C} = \mathbf{1} \tag{26}$$

For the internal coordinate region, we use the same transformation matrices as in the conventional redundant internal optimizer:

$$\mathbf{D}_{II} = \frac{\partial \mathbf{x}_I}{\partial \mathbf{q}_I} = \mathbf{G}^{-}\mathbf{B} \tag{27}$$

For the rigid fragments, the $\mathbf{D}$-matrix elements involving translational coordinates, can be written as:

$$\frac{\partial \mathbf{x}_{K,i,x}}{\partial \varphi_x^K} = \frac{\partial \mathbf{x}_{K,i,y}}{\partial \varphi_y^K} = \frac{\partial \mathbf{x}_{K,i,z}}{\partial \varphi_z^K} = s_\varphi^K \tag{28}$$

The elements that involve rotational elements can be written as

$$\frac{\partial \mathbf{x}_{K_i}}{\partial \theta_y^K} = \mathbf{T}^x \frac{\mathbf{T}^y}{\partial \theta_y^K} \mathbf{T}^z [\mathbf{x}_{K_i}(0) - \mathbf{O}_K] \tag{29}$$

with

$$\frac{\partial \mathbf{T}^y}{\partial \theta_y^K} = \begin{bmatrix} -s_\theta^K \sin(s_\theta^K \theta_y^K) & 0 & s_\theta^K \cos(s_\theta^K \theta_y^K) \\ 0 & 0 & 0 \\ -s_\theta^K \cos(s_\theta^K \theta_y^K) & 0 & -s_\theta^K \sin(s_\theta^K \theta_y^K) \end{bmatrix} \tag{30}$$

The rotation matrices and their derivatives need to be evaluated only once for each rigid fragment. The $\mathbf{D}$-matrix elements for translation and rotation derivatives of the internal redundant region, $\partial \mathbf{x}_I / \partial \mathbf{q}_{I'}$, are obtained in the same way as the rigid fragments. Provided the number of redundant internal coordinates is small, the transformation of the gradient only requires CPU time and memory that scales linearly with the size of the system.

2.3. *Transformation of the Hessian*

We can write the transformation of the Hessian to the optimization coordinate space in two ways.

$$\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{q}\, \partial \mathbf{q}} = \mathbf{D}\mathcal{H}\,\mathbf{D}^t - \mathbf{D}\left[\sum_i^{N^q} \frac{\partial E}{\partial q_i} \frac{\partial^2 q_i}{\partial \mathbf{x} \partial \mathbf{x}}\right]\mathbf{D}^t \tag{31}$$

$$\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{q}\, \partial \mathbf{q}} = \frac{\partial (\mathbf{D} g)}{\partial \mathbf{q}} = \mathbf{D}\mathcal{H}\,\mathbf{D}^t + \sum_i^{N^x} \frac{\partial E}{\partial x_i} \frac{\partial^2 x_i}{\partial \mathbf{q} \partial \mathbf{q}} \tag{32}$$

Equation (31) is equivalent to equation (12), while equation (32) is obtained by differentiating equation (25). $\mathcal{H}$ and $\mathbf{H}$ are the Hessian matrix in Cartesian coordinates and geometry optimization coordinates, respectively, and the $\mathbf{D}\mathcal{H}\mathbf{D}^t$ term in equations (31) and (32) is evaluated with the same $\mathbf{D}$-matrix as the gradient. N^x and N^q are the number of Cartesian coordinates and geometry optimization coordinates, respectively.

For the 'gradient contribution' $\sum_i^{N^x} (\partial E / \partial x_i)(\partial^2 x_i / \partial \mathbf{q}\, \partial \mathbf{q})$, we can either use the second r.h.s. term of equation (31), or the term in equation (32). We use both alternatives, depending on the part of the matrix that is being evaluated. Because $\partial^2 \mathbf{x} / \partial \mathbf{q}\, \partial \mathbf{q}$ is in fact a derivative of the $\mathbf{D}$-matrix, it is sparse, and the non-zero blocks are as follows. The contribution from the internal redundant coordinates is evaluated in the same way as the standard geometry optimizer discussed in the introduction.

$$\sum_i^{N^x} \frac{\partial E}{\partial x_i} \frac{\partial^2 x_i}{\partial \mathbf{q}_I \partial \mathbf{q}_I} = -\mathbf{D}_{II}\left[\sum_i^{N^{q_I}} \frac{\partial E}{\partial q_{I_i}} \frac{\partial^2 q_{I_i}}{\partial \mathbf{x}_I \partial \mathbf{x}_I}\right]\mathbf{D}_{II}^t \tag{33}$$

For the gradient contributions related to the rigid blocks, only

$$\sum_i^{N^x} \frac{\partial E}{\partial x_i} \frac{\partial^2 x_i}{\partial \mathbf{q}_{K^\theta} \partial \mathbf{q}_{K^\theta}}$$

is not zero. These contributions can be efficiently evaluated by identifying the non-zero elements of the derivative **D**-matrix:

$$\frac{\partial \mathbf{x}_{K_i}}{\partial \theta_y^K \partial \theta_y^K} = \mathbf{T}^x \frac{\mathbf{T}^y}{\partial \theta_y^K \partial \theta_y^K} \mathbf{T}^z \left[\mathbf{x}_{K_i}(0) - \mathbf{O}_K \right] \tag{34}$$

$$\frac{\partial \mathbf{x}_{K_i}}{\partial \theta_y^K \partial \theta_z^K} = \mathbf{T}^x \frac{\mathbf{T}^y}{\partial \theta_y^K} \frac{\mathbf{T}^z}{\partial \theta_z^K} \left[\mathbf{x}_{K_i}(0) - \mathbf{O}_K \right] \tag{35}$$

with

$$\frac{\partial^2 \mathbf{T}^{\theta_x^K}}{\partial \theta_x^K \partial \theta_x^K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\left(s_\theta^K\right)^2 \cos(s_\theta^K \theta_x^K) & -\left(s_\theta^K\right)^2 \sin(s_\theta^K \theta_x^K) \\ 0 & \left(s_\theta^K\right)^2 \sin(s_\theta^K \theta_x^K) & -\left(s_\theta^K\right)^2 \cos(s_\theta^K \theta_x^K) \end{bmatrix} \tag{36}$$

The number of non-zero contributions scale linearly with the number of atoms in rigid fragments. Last, the gradient contribution related to the coupling between the rotational coordinates for the internal coordinate region, and redundant internal coordinates, can be written as:

$$\sum_i^{N^x} \frac{\partial E}{\partial x_i} \frac{\partial^2 x_i}{\partial \mathbf{q}_{I^\theta} \partial \mathbf{q}_I} = \sum_i^{N^{x_I}} \frac{\partial E}{\partial x_{I_i}} \frac{\partial}{\partial \mathbf{q}_{I^\theta}} \left[\frac{\partial x_{I_i}}{\partial \mathbf{q}_I} \right] \tag{37}$$

We recognize $\partial x_{I_i}/\partial \mathbf{q}_I$ as the elements of the **D**-matrix that involve internal coordinates (equation (29)). We can take the derivative of those **D**-matrix elements with respect to the rotational angle in the same way as the Cartesian coordinates (expression 29), for example:

$$\frac{\partial^2 \mathbf{x}_{I_i}}{\partial \theta_y^I \partial q_{I_j}} = \mathbf{T}^{\theta_x^I} \frac{\partial \mathbf{T}^{\theta_y^I}}{\partial \theta_y^I} \mathbf{T}^{\theta_z^I} \left[\frac{\partial \mathbf{x}_{I_i}}{\partial q_{I_j}} \right] \tag{38}$$

2.3. *QM/MM Hessian-vector product*

To solve the RFO equations iteratively without storage of the Hessian, we need to evaluate the product of the Hessian with a vector **V**. Here, we explicitly write the Hessian-vector product in terms of the ONIOM contributions.

$$\mathbf{HV} = \mathbf{H}^{real,MM}\mathbf{V} + \mathbf{H}^{model,MM}\mathbf{V} - \mathbf{H}^{model,QM}\mathbf{V} \tag{39}$$

We assume that the model system is relatively small, and that $\mathbf{H}^{model,MM}\mathbf{V}$ and $\mathbf{H}^{model,QM}\mathbf{V}$ can easily be evaluated by forming $\mathbf{H}^{model}$ in memory. For the $\mathbf{H}^{model,QM}\mathbf{V}$ term we can then either use an analytically evaluated Hessian or an approximate (updated) Hessian. If we use the latter, the update is carried out with the gradients from the QM model system, $\mathbf{g}_I = \partial E^{model,QM}/\partial \mathbf{q}_I$, since it is $\mathbf{H}^{model,QM}$ that is being estimated, and not $\mathbf{H}^{ONIOM}$. The MM contributions to $\mathbf{H}^{ONIOM}$ in equation (3) are always evaluated analytically.

It is clear that the only remaining bottleneck in the Hessian vector product, and therefore in our entire optimization scheme, comes from $\mathbf{H}^{real,MM}$. This contribution can be expressed in the Hessian matrix in Cartesian coordinates and the gradient contribution:

$$\mathbf{H}^{real,MM}\mathbf{V} = \left[\mathbf{D}\mathcal{H}^{real,MM}\mathbf{D}^t\right]\mathbf{V} + \left[\sum_i^{N^x} \frac{\partial E^{real,M}}{\partial x_i} \frac{\partial^2 x_i}{\partial \mathbf{q}\partial \mathbf{q}}\right]\mathbf{V} \tag{40}$$

Because of the sparseness of the gradient contribution, the second term of equation (40) can be evaluated without significant CPU or memory requirements. That leaves the contribution of the Hessian matrix in Cartesian coordinates. We can rewrite $[\mathbf{D}\mathcal{H}^{real,MM}\mathbf{D}^t]\mathbf{V}$ to indicate the order in which the multiplications are processed.

$$\left[\mathbf{D}\mathcal{H}^{real,MM}\mathbf{D}^t\right]\mathbf{V} = \mathbf{D}\left[\mathcal{H}^{real,MM}\mathbf{V}'\right] \tag{41}$$

with

$$\mathbf{V}' = \mathbf{D}^t\mathbf{V} \tag{42}$$

The transformation of **V** to **V**′ is trivial due to the sparseness of **D**, and the problem is reduced to the product of the Cartesian Hessian with vector **V**′ (with the product vector, again trivially, multiplied with **D**). In Appendix A we outline the methods that we developed to evaluate $\mathcal{H}^{real,MM}\mathbf{V}'$, which are O($N$) in both memory and CPU time.

Finally, for the optimization we need to account for the redundancies and constraints from equations (13) and (14). The projections do not complicate the scheme outlined in this section. The only non-zero block of **(1-P)** in equations (13) and (14) involves pairs of model system internal coordinates.

3. Implementation

We implemented our methods in a development version of the Gaussian package for electronic structure calculations [9]. As outlined in the beginning of this section, we solve equation (19) using O(N) methods. We can either use an analytical or an updated QM contribution

to the Hessian. When possible, we carried over the features of the conventional (Berny [72]) optimizer to our new quadratically coupled QM/MM optimizer. For example, constraints can be placed on redundant internal coordinates, and the initial Hessian can be improved by numerical differentiation of the forces corresponding to select coordinates. We implemented the standard trust-region update mechanism, and included line searches in the full QM/MM coordinate space to improve the RFO steps [72]. In the calculation of the MM energy and gradient, we use the 'fast multiple method' (FMM) for the electrostatic interaction [73–76]. The evaluation of the bonded interactions is O(N). The Van der Waals interaction can be made O(N) for very large systems using a boxing scheme similar to that in FMM, but for the size of system used in the examples here, a simple quadratic algorithm has modest cost compared to other terms.

After the quadratic macro step, we still carry out micro-iterations. Although this is not strictly necessary for the algorithm, it will reduce the number of macro-steps, and therefore the number of QM energy and gradient evaluations. However, there is no longer a presumption in the macro-step that the forces on the MM atoms are exactly zero, and we can consequently use less tight convergence criteria in the micro-iterations, which reduces the number of micro-iteration steps. (In the examples that follow this section, however, we use always the full convergence criteria, in order to be able to compare only the different macro steps in the standard and new scheme). In fact, the quadratic macro-step includes (quadratic) displacements of the MM atoms, which further reduces the number of cycles in the subsequent micro-iterations. Of course, we could also carry out the micro-iterations in the MM region using the same quadratic scheme as developed for the macro-step, simply by freezing the QM atoms and removing all the contributions involving the internal coordinate space. Although this drastically reduces the number of micro-iterations, each cycle involves the evaluation of many Hessian-vector products, and the result is an increase in overall CPU time compared to the first-order micro-iterations.

Last, we have implemented a simple form of parallelization. The most time consuming parts of the algorithm is the evaluation of the product of the Hessian with a vector. From equation (53) in Appendix A we can see that for each vector, there are three independent contributions that need to be evaluated using FMM. Since there are usually several Hessian vector products required for each cycle of the Davidson diagonalization, in addition to the Van der Waals and non-bonded contributions, we can distribute these tasks over several processors.

4. Results

4.1. *Small systems*

We applied our new optimization scheme to the set of small systems reported in our earlier work on the standard micro-iteration scheme [19]. These systems are small enough to be optimized with a conventional scheme, so that we can assess the performance gain of the standard micro-iterations scheme over conventional optimization schemes, as well as our new scheme with the coupled macro-step. We show the results in table 1. We used the same micro-iterations convergence criteria in the coupled macro-step calculations and the standard micro-iterations calculations, so that the performance gain only reflects the improvement in the macro-step. From the results it is clear that the new coupled scheme always performs better than the standard micro-iteration scheme. Because for small systems like these the standard micro-iteration scheme is already very efficient, the performance gain of the new scheme is only moderate.

Table 1. Number of macro-steps in the different types of optimization schemes.[a]

	Water dimer	Hexaphenylethane	Organometallic	Phosphasilene	Carbene
Model system	monomer	ethane	see ref. [19]	$H_2Si=PH$	see ref. [19]
High level	HF/6-31G(d)	AM1	B3LYP/LANL2DZ	AM1	AM1
Low level	UFF[b]	UFF[b]	UFF[b]	UFF[b]	UFF[b]
Conventional	54[c]	12[c]	11	18	16[c]
Standard macro/micro scheme	7[c]	10[c]	6	6	9[c]
Coupled macro-step with micro-iterations	5[c]	9[c]	4	4	8[c]

[a]See reference [19] for the structures; the water dimer optimizations started from the B3LYP/6-31G(d) geometry, hexaphenylethane from the AM1 geometry, and the other examples from the level indicated but without QEq charges.
[b]Charges obtained using QEq scheme.
[c]Tight convergence criteria.

4.2. *Protein minimizations*

To test our method for a larger system, we minimized the bacteriorhodopsin (bR) protein. The full protein contains about 3500 atoms, and has a retinal chromophore in the centre linked via a protonated Schiff base. The chromophore (37 atoms) was treated with HF/6-31G(d), and the protein environment with the Amber force field [77]. For details on the preparation of the system see [18]. The geometry optimizations were started from the X-ray structure [78], with the MM region pre-optimized. We carried out two sets of optimizations. The first has most of the protein environment frozen, and only the atoms within 6 Å of the chromophore are relaxed (341 atoms in total). In the second set, we allow the full system to relax.

In figure 1, we show the energy profiles along the geometry optimization for the system with most of the protein frozen. The new scheme clearly has a smoother behaviour in the initial part of the minimization than the standard micro-iteration scheme. The erratic steps of the standard scheme are the result of the initial Hessian not including any information about the coupling, although this does develop after a while. The standard scheme cannot explicitly include coupling between the regions, as the new scheme does, but it can include this information implicitly. In the final stages of the optimization, the new scheme converges fast, while the standard scheme needs many more steps to converge the forces and displacements.

Figure 2 shows the energy profiles of the minimizations of the fully relaxed system. The new micro-iteration scheme converges rapidly and smoothly, while the standard scheme again has several erratic steps in the initial phase of the optimization. Towards the end of the profile, the ill-defined update mechanism corrupts the Hessian, which then has the wrong curvature. Several bad steps are taken, and in fact the optimization never converges. Finally, we see that the standard scheme yields a lower energy than the new scheme. This is the result of the new scheme 'carefully' converging to local minimum that is close by. The standard scheme takes several bad steps initially, which shake up the system allowing the MM region to find a different, energetically lower local minimum. This illustrates the general problem of geometry optimization of very large systems. Both the quadratically coupled scheme and the standard scheme find a nearby local minimum, and neither is a method for the search of global minima. If a reaction mechanism is being studied, one must ensure that the stationary points correspond to the same valley on the potential surface. If the global minimum is required, a method such as simulated annealing must be used, in combination of a local optimization method such as the one presented here.

5. Discussion and conclusions

Our examples show that introducing coupling between the QM region and the MM region in the macro-step, significantly improves the convergence behaviour of the micro-iterations optimization scheme. This reduces the number of QM energy and gradient evaluations that are needed, which improves the overall efficiency of the calculation. However, despite the O(N) techniques

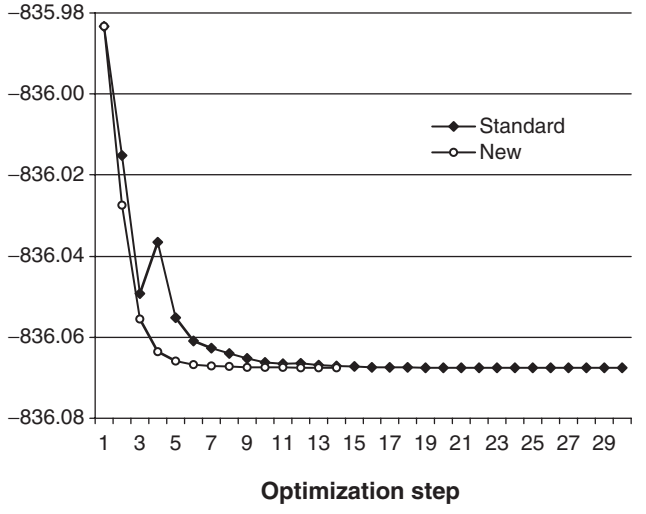

Figure 1. Energy profiles for the minimization of bR with most of the protein frozen, both with the standard and the new micro-iteration schemes.

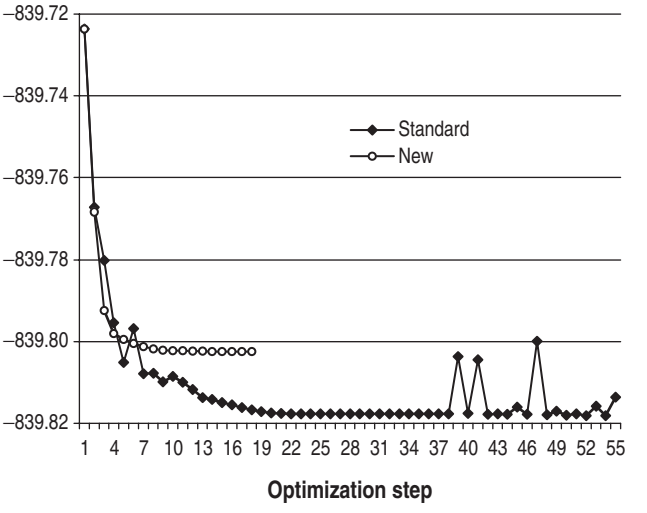

Figure 2. Energy profiles for the minimization of bR with the full protein relaxed, both with the standard and the new micro-iteration schemes.

that we developed, the calculation of the macro-step can be computationally costly, and the method is most suitable for QM/MM schemes with relatively expensive QM methods.

At the core of our new optimization scheme is the O(N) technology to evaluate the product of the ONIOM Hessian with a vector. In the current scheme we used this technology for a RFO macro-step, but it can be used for other optimization schemes as well. For a NR macro-step we would have to use a direct method for the inversion of the ONIOM Hessian, which again requires Hessian vector products [79, 80]. The scheme can also be extended to the search for saddle point structures on the potential surface [81], which will be the topic of a subsequent paper.

Finally, we need to comment on the applicability of our optimization method to other QM/MM schemes. First, we outlined our methods within the ONIOM extrapolation framework, separating the QM/MM Hessian into three components (equation (3)). Although most other QM/MM methods are presented as a summation scheme, they can easily be recast as an extrapolation scheme like ONIOM. Second, our O(N) scheme for the Hessian vector evaluation assumes a charge distribution that interacts through $s_{ij}Q_iQ_j/r_{ij}$ potential functions. Several molecular mechanics force fields, however, employ functions that our FMM implementation currently cannot handle. For example MMFF94 [82] uses buffered interactions of the form $S_{ij}Q_iQ_j/(r_{ij}+\delta)$, and MM3 [83] includes bond dipoles. Either the FMM must be extended for these types of interactions, which would require only minor modifications for the bond dipoles, or the force field must be re-parameterized for $s_{ij}Q_iQ_j/r_{ij}$ potential functions. Third, many QM/MM schemes use electronic embedding, in which the charge distribution of the MM region is incorporated in the QM Hamiltonian [18, 23, 43]. This introduces direct coupling between the coordinates of the MM centres and the wave function, and the $\mathbf{H}^{model,QM}$ Hessian elements that involve 'inactive MM atoms' will no longer be zero. In other words, $\mathbf{H}^{model,QM}$ becomes a full matrix and the evaluation of it requires the solution of the coupled perturbed Hartree–Fock (CPHF) equations for all the QM and MM centers. This is currently not feasible for very large systems, and the scheme as presented in this paper cannot be applied.

Acknowledgements

We gratefully acknowledge support from the National Science Foundation (CHE-0131157 to HBS and CHE-0209660 to KM)

Appendix A. Product of the Cartesian MM Hessian with a vector

For the coupled macro step, we need the product of the MM Hessian in Cartesian coordinates, $\mathcal{H}^{real,MM}$, with a vector. In this section we describe how we obtain this product with O(N) scaling for both memory and CPU. In the most common MM force fields, there are three types of terms that we need to distinguish: the bonded terms, the Van der Waals terms, and the electrostatic terms. The bonded terms usually consist of bond stretches, angle bends, dihedral deformations, and out-of-plane bends. We evaluate the contribution of the bonded terms to the Hessian-vector product on-the-fly, thus without storing the Hessian elements. The computational cost will scale linearly because the number of terms scales linearly with the size of the system and each term involves no more than 12 Cartesian coordinates. The number of Van der Waals terms, on the other hand, scales quadratically with the size of the system. However, the interaction between two particles drops sharply with increasing distance, due to the r^{-6}, r^{-12}, or $\exp(-r)$ factors in the expression, and linear scaling even with high numerical accuracy can be achieved using a distance cut-off for the Hessian vector product. Because the second derivative terms have r^{-8} factors, for a given accuracy significantly smaller cut-off distances and boxes can be used than in the evaluation of the energy and gradient. We evaluate the contributions on-the-fly, and apply the cut-off through a boxing algorithm similar to that used in the 'fast multipole method' [73–76]. The evaluation of the Van der Waals contribution to the Hessian vector product then scales linearly with the size of the system, and requires only O(N) intermediate storage. The last type of contribution, the electrostatic interaction, is long range, and therefore high numerical accuracy cannot be achieved with a simple distance based cut-off. In the remainder of this section we will describe the techniques we developed to evaluate this contribution via extensions to the fast multipole method.

In a MM calculation, the electrostatic interaction is evaluated as a collection of point charges.

$$E^{Q} = \sum_{i}^{N} \sum_{j}^{i-1} \frac{s_{ij} Q_i Q_j}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{43}$$

s_{ij} is a scale factor that reduces the interaction between two charges when they are close together, based on the connectivity. Typically, interactions between centres that are one or two bonds separated are scaled to zero ($s = 0$). Three bond separated interactions are scaled between zero and one ($s = \mathrm{a}$), depending on the force

field, and interactions between charges separated more than three bonds are fully included ($s=1$). We rewrite equation (43)

$$E^{Q} = \sum_{i}^{N}\sum_{j}^{i-1}\frac{Q_iQ_j}{|\mathbf{x}_j-\mathbf{x}_i|} - \sum_{i}^{N}\sum_{j}^{i-1}\frac{(1-s_{ij})Q_iQ_j}{|\mathbf{x}_j-\mathbf{x}_i|} \tag{44}$$

Because s_{ij} differs from 1 only for atoms i and j connected by bonding interactions, the number of non-zero contributions to the second r.h.s. term of equation (44) scales linearly with the size of the system. The corresponding contributions to the Hessian-vector product can be evaluated in a similar manner as the bonded terms, and we are now concerned with the unscaled interaction of a collection of point charges.

$$E^{Q'} = \sum_{i}^{N}\sum_{j}^{i-1}\frac{Q_iQ_j}{|\mathbf{x}_j-\mathbf{x}_i|} \tag{45}$$

We distinguish (block) diagonal and off-diagonal elements of the Hessian in Cartesian coordinates.

$$H^{Q'}_{i,\mu,i',\nu} = \frac{\partial^2 E^{Q'}}{\partial x_{i,\mu}\partial x_{i',\nu}} = Q_iQ_{i'}\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i',\nu}}\frac{1}{|\mathbf{x}_{i'}-\mathbf{x}_i|} \tag{46}$$

$$H^{Q'}_{i,\mu,i,\nu} = \frac{\partial^2 E^{Q'}}{\partial x_{i,\mu}\partial x_{i,\nu}} = Q_i\sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \tag{47}$$

μ and ν denote the axes and i and j the centres. μ and ν can be the same axis, but i and i' denote explicitly different centres. In our algorithm, we need the product $\mathbf{W}$ of the Hessian $H^{Q'}$ with a (displacement) vector $\mathbf{V}$.

$$\mathbf{W} = \mathbf{H}^{Q'}\mathbf{V} \tag{48}$$

We write $\mathbf{W}$ in terms of diagonal and off-diagonal elements of the Hessian.

$$\begin{aligned} W_{i,\mu} &= \left(\mathbf{H}^{Q'}\mathbf{V}\right)_{i,\mu} = \sum_{\nu}^{3}\sum_{j}^{N}H^{Q'}_{i,\mu,j,\nu}V_{j,\nu} \\ &= \sum_{\nu}^{3}\sum_{j}^{N}\frac{\partial^2 E^{Q'}}{\partial x_{i,\mu}\partial x_{j,\nu}}V_{j,\nu} \\ &= \sum_{\nu}^{3}Q_i\sum_{j\neq i}^{N}Q_jV_{j,\nu}\frac{\partial^2}{\partial x_{i,\mu}\partial x_{j,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \\ &\quad + \sum_{\nu}^{3}Q_iV_{i,\nu}\sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \end{aligned} \tag{49}$$

Using the translational invariance of the Hessian we can rewrite $\mathbf{W}$:

$$\begin{aligned} H^{Q'}_{i,\mu,i,\nu} &= -\sum_{j\neq i}^{N}H^{Q'}_{i,\mu,j,\nu}Q_i\sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \\ &= -Q_i\sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{j,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \end{aligned} \tag{50}$$

$$\begin{aligned} W_{i,\mu} &= -\sum_{\nu}^{3}Q_i\sum_{j\neq i}^{N}Q_jV_{j,\nu}\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \\ &\quad + \sum_{\nu}^{3}Q_iV_{i,\nu}\sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \end{aligned} \tag{51}$$

In expression 51, we can recognize terms that resemble sums of electric field gradients:

$$\frac{\partial^2\mathcal{V}_i(Q_1\ldots Q_N)}{\partial\mu\partial\nu} = \sum_{j\neq i}^{N}Q_j\frac{\partial^2}{\partial x_{i,\mu}\partial x_{i,\nu}}\frac{1}{|\mathbf{x}_j-\mathbf{x}_i|} \tag{52}$$

We use this to write $\mathbf{W}$ in terms of electric field gradients (EFG) $\mathcal{V}_i$. This involves the EFG from the original charge distribution $\mathbf{Q}$, and three contributions from the integration of the charges $\mathbf{Q}$ with the elements of $\mathbf{V}$.

$$\begin{aligned} &W_{i,\mu} \\ &= Q_i\sum_{\nu}^{3}\left[V_{i,\nu}\frac{\partial^2\mathcal{V}_i(Q_1\ldots Q_N)}{\partial\mu\partial\nu} - \frac{\partial^2\mathcal{V}_i(Q_1V_{1,\nu}\ldots Q_NV_{N,\nu})}{\partial\mu\partial\nu}\right] \end{aligned} \tag{53}$$

The terms in equation (53) can each be evaluated using the fast multiple method (FMM), which scales linearly with the number of particles in both memory and CPU.

References

[1] H. B. Schlegel, J. Comput. Chem. **24,** 1514 (2003).
[2] Y. K. Zhang, H. Y. Liu, and W. T. Yang, J. Chem. Phys. **112,** 3483 (2000).
[3] R. B. Murphy, D. M. Philipp, and R. A. Friesner, J. Comput. Chem. **21,** 1442 (2000).
[4] F. Maseras and K. Morokuma, J. Comput. Chem. **16,** 1170 (1995).
[5] S. Hayashi and I. Ohmine, J. Phys. Chem. **104,** 10678 (2000).
[6] S. R. Billeter, A. J. Turner, and W. Thiel, Phys. Chem. Chem. Phys. **2,** 2177 (2000).
[7] R. J. Hall, S. A. Hindle, N. A. Burton, and I. H. Hillier, J. Comput. Chem. **21,** 1433 (2000).

[8] A. J. Turner, V. Moliner, and I. H. Williams, Phys. Chem. Chem. Phys. **1,** 1323 (1999).
[9] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, *Gaussian, Revision C.01*, Gaussian Inc., Wallingford, CT (2004).
[10] U. C. Singh and P. A. Kollman, J. Comput. Chem. **7,** 718 (1986).
[11] M. J. Field, P. A. Bash, and M. Karplus, J. Comput. Chem. **11,** 700 (1990).
[12] A. Warshel and M. Levitt, J. Mol. Biol. **103,** 227 (1976).
[13] S. Humbel, S. Sieber, and K. Morokuma, J. Chem. Phys. **105,** 1959 (1996).
[14] M. Svensson, S. Humbel, R. D. J. Froese, T. Matsubara, S. Sieber, and K. Morokuma, J. Phys. Chem. **100,** 19357 (1996).
[15] S. Dapprich, I. Komáromi, K. S. Byun, K. Morokuma, and M. J. Frisch, J. Mol. Struct. (Theochem) **461–462,** 1 (1999).
[16] T. Vreven and K. Morokuma, J. Comput. Chem. **21,** 1419 (2000).
[17] T. Vreven, B. Mennucci, C. O. da Silva, K. Morokuma, and J. Tomasi, J. Chem. Phys. **115,** 62 (2001).
[18] T. Vreven and K. Morokuma, Theor. Chem. Acc. **109,** 125 (2003).
[19] T. Vreven, K. Morokuma, Ö. Farkas, H. B. Schlegel, and M. J. Frisch, J. Comput. Chem. **24,** 760 (2003).
[20] T. Kerdcharoen and K. Morokuma, Chem. Phys. Lett. **355,** 257 (2002).
[21] E. Derat, J. Bouquant, and S. Humbel, J. Mol. Struct. (Theochem) **632,** 61 (2003).
[22] N. Rega, S. S. Iyengar, G. A. Voth, H. B. Schlegel, T. Vreven, and M. J. Frisch, J. Phys. Chem. **108,** 4210 (2004).
[23] T. Vreven, I. Komáromi, S. Dapprich, K.S. Byun, J.A. Montgomery Jr, K. Morokuma, M.J. Frisch, in preparation.
[24] P. B. Karadakov and K. Morokuma, Chem. Phys. Lett. **317,** 589 (2000).
[25] T. Kruger and A. F. Sax, J. Comput. Chem. **23,** 371 (2001).
[26] B. W. Hopkins and G. S. Tschumper, J. Comput. Chem. **24,** 1563 (2003).
[27] A. Warshel, Annu. Rev. Biophys. Biomol. Struct. **32,** 425 (2003).
[28] G. Ujaque and F. Maseras, Struct. Bonding. **112,** 117 (2004).
[29] J. Gao, in *Reviews in Computational Chemistry*, K. Lipkowitz and D. Boyd (Eds.), Vol. 7, p. 119, VCH Publishers, New York (1996).
[30] R. D. J. Froese and K. Morokuma, in *Encyclopedia of Computational Chemistry*, P. Schleyer, N. Allinger, P. Kollman, T. Clark, H. Schaefer III, J. Gasteiger, and P. Schreiner (Eds.), Vol. 2, p. 1244, Wiley, Chichester (1998).
[31] J. Gao, in *Encyclopedia of Computational Chemistry*, P. Schleyer, N. Allinger, P. Kollman, T. Clark, H. Schaefer III, J. Gasteiger, and P. Schreiner (Eds.), Vol. 2, p. 1257, Wiley, Chichester (1998).
[32] K. M. Merz and R. V. Stanton, in *Encyclopedia of Computational Chemistry*, P. Schleyer, N. Allinger, P. Kollman, T. Clark, H. Schaefer III, J. Gasteiger, and P. Schreiner (Eds.), Vol. 4, p. 2330, Wiley, Chichester (1998).
[33] M. F. Ruiz-López and J. L. Rivail, in *Encyclopedia of Computational Chemistry*, P. Schleyer, N. Allinger, P. Kollman, T. Clark, H. Schaefer III, J. Gasteiger, and P. Schreiner (Eds.), Vol. 1, Wiley, Chichester (1998).
[34] J. Tomasi and C. S. Pomelli, in *Encyclopedia of Computational Chemistry*, P. Schleyer, N. Allinger, P. Kollman, T. Clark, H. Schaefer III, J. Gasteiger, and P. Schreiner (Eds.), Vol. 4, p. 2343, Wiley, Chichester (1998).
[35] V. Thery, D. Rinaldi, J. L. Rivail, B. Maigret, and G. G. Ferenczy, J. Comput. Chem. **15,** 269 (1994).
[36] J. Gao, P. Amara, C. Alhambra, and M. J. Field, J. Phys. Chem. A **102,** 4714 (1998).
[37] N. Reuter, A. Dejaegere, B. Maigret, and M. Karplus, J. Phys. Chem. A **104,** 1720 (2000).
[38] R. M. Nicoll, S. A. Hindle, G. MacKenzie, I. H. Hillier, and N. A. Burton, Theor. Chem. Acc. **106,** 105 (2000).
[39] N. Koga and K. Morokuma, Chem. Phys. Lett. **172,** 243 (1990).
[40] I. Antes and W. Thiel, J. Phys. Chem. A **103,** 9290 (1999).
[41] G. A. DiLabio, M. M. Hurley, and P. A. Christiansen, J. Chem. Phys. **116,** 9578 (2002).
[42] Y. K. Zhang, T. S. Lee, and W. T. Yang, J. Chem. Phys. **110,** 46 (1999).
[43] D. Bakowies and W. Thiel, J. Phys. Chem. **100,** 10580 (1996).
[44] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89,** 52 (1985).
[45] T. H. Fischer and J. Almlof, J. Phys. Chem. **96,** (1992).
[46] L. T. Biegler, J. Nocedal, and C. Schmid, SIAM J. Optim. **5,** 314 (1995).
[47] R. H. Byrd, P. H. Lu, J. Nocedal, and C. Y. Zhu, SIAM J. Sci. Comput. **16,** 1190 (1995).
[48] R. H. Byrd, J. Nocedal, and R. B. Schnable, Math. Program. **63,** 129 (1994).
[49] D. C. Liu and J. Nocedal, Math. Program. **45,** 503 (1989).
[50] J. L. Morales and J. Nocedal, SIAM J. Optim. **10,** 1079 (2000).
[51] J. Nocedal, Math. Program. **35,** 773 (1980).
[52] R. Fletcher, SIAM J. Optim. **5,** 192 (1995).
[53] X. Prat-Resina, M. Garcia-Viloca, G. Monard, A. Gonzalez-Lafont, J. M. Lluch, J. M. Bofill, and J. M. Anglada, Theor. Chem. Acc. **107,** 147 (2002).
[54] J. M. Anglada, E. Besalu, J. M. Bofill, and J. Rubio, J. Mater. Chem. **25,** 85 (1999).

[55] O. Farkas and H. B. Schlegel, J. Chem. Phys. **109,** 7100 (1998).
[56] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, J. Am. Chem. Soc. **101,** 2550 (1979).
[57] P. Pulay and G. Fogarasi, J. Chem. Phys. **96,** 2856 (1992).
[58] G. Fogarasi, X. F. Zhou, P. W. Taylor, and P. Pulay, J. Am. Chem. Soc. **114,** 8191 (1992).
[59] C. Y. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, J. Comput. Chem. **17,** 49 (1996).
[60] J. Baker, A. Kessi, and B. Delley, J. Chem. Phys. **105,** 192 (1996).
[61] M. von Arnim and R. Ahlrichs, J. Chem. Phys. **111,** 9183 (1999).
[62] K. N. Kudin, G. E. Scuseria, and H. B. Schlegel, J. Chem. Phys. **114,** 2919 (2001).
[63] E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations*, McGraw-Hill, New York (1955).
[64] O. Farkas and H. B. Schlegel, Phys. Chem. Chem. Phys. **4,** 11 (2002).
[65] O. Farkas and H. B. Schlegel, J. Chem. Phys. **111,** 10806 (1999).
[66] B. Paizs, J. Baker, S. Suhai, and P. Pulay, J. Chem. Phys. **113,** 6566 (2000).
[67] B. Paizs, G. Fogarasi, and P. Pulay, J. Chem. Phys. **109,** 6571 (1998).
[68] K. Nemeth, O. Coulaud, G. Monard, and J. G. Angyan, J. Chem. Phys. **114,** 9747 (2001).
[69] K. Nemeth, O. Coulaud, G. Monard, and J. G. Angyan, J. Chem. Phys. **113,** 5598 (2000).
[70] J. Baker, D. Kinghorn, and P. Pulay, J. Chem. Phys. **110,** 4986 (1999).
[71] E. R. Davidson, J. Computat. Phys. **17,** 87 (1975).
[72] H. B. Schlegel, J. Comput. Chem. **3,** 214 (1982).
[73] K. N. Kudin and G. E. Scuseria, Chem. Phys. Lett. **283,** 61 (1998).
[74] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271,** 51 (1996).
[75] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA (1988).
[76] L. Greengard and V. Rokhlin, J. Computat. Phys. **73,** 325 (1987).
[77] 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, and P. A. Kollman, J. Am. Chem. Soc. **117,** 5179 (1995).
[78] H. Luecke, B. Schobert, H. T. Richter, J. P. Cartailler, and J. K. Lanyi, J. Mol. Biol. **291,** 899 (1999).
[79] P. Pulay, J. Comput. Chem. **3,** 5043 (1982).
[80] C. Lanczos, J. Res. Nat. Bur. Standards **49,** 556 (1952).
[81] J. Baker, J. Comput. Chem. **7,** 385 (1986).
[82] T. A. Halgren, J. Comput. Chem. **17,** 490 (1996).
[83] A. Aped and N. L. Allinger, J. Am. Chem. Soc. **114,** 1 (1992).