Constrained Numerical Gradients and Composite Gradients: Practical Tools for Geometry Optimization and Potential Energy Surface Navigation

Michael Stenrup,[a,b] Roland Lindh,[a,b] and Ignacio Fdez. Galván*[a,b]

A method is proposed to easily reduce the number of energy evaluations required to compute numerical gradients when constraints are imposed on the system, especially in connection with rigid fragment optimization. The method is based on the separation of the coordinate space into a constrained and an unconstrained space, and the numerical differentiation is done exclusively in the unconstrained space. The decrease in the number of energy calculations can be very important if the system is significantly constrained. The performance of the method is tested on systems that can be considered as composed of several rigid groups or molecules, and the results show that the error with respect to conventional optimizations is of the order of the convergence criteria. Comparison with another method designed for rigid fragment optimization proves the present method to be competitive. The proposed method can also be applied to combine numerical and analytical gradients computed at different theory levels, allowing an unconstrained optimization with numerical differentiation restricted to the most significant degrees of freedom. This approach can be a practical alternative when analytical gradients are not available at the desired computational level and full numerical differentiation is not affordable. © 2015 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23987

Introduction

One of the central problems in computational chemistry is that of geometry optimization or, more generally, of locating points on a potential energy surface (PES) with some particular properties (minima, saddle points, crossing points,. . .)[1]. Although there are algorithms that can perform geometry optimization using only the energy computed at selected points of the PES,[2–5] as the number of degrees of freedom increases their performance degrades and their convergence is very slow.[6] The optimization works much better when the gradient information—the derivative of the energy with respect to the geometrical coordinates—is available. The gradient, moreover, is also an essential quantity if one is interested in the dynamics of the system. For many electronic structure methods and software packages, it is possible to obtain the gradient based on an analytical formulation, with a computational cost that is roughly equivalent to that of computing the energy. However, the most sophisticated methods often lack such an analytical implementation, and to obtain the gradient one must resort to numerical differentiation by finite differences, requiring a large number (proportional to the number of atoms in the system) of energy calculations. Undoubtedly, the most desirable solution would be to develop and implement the expressions necessary to compute the analytical gradient for the method to be used, but this tends to be an intimidating task that must, additionally, be performed for every different method. Conversely, the possibility of using numerical differentiation is often discarded because of its high computational cost.

During the last years, there has been a change in the trend of computer technology progress, and the paradigm has changed from more powerful single processors to larger numbers of parallel "cores" or processing units. Computational scientists can no longer assume that their programs will run faster in newer machines, but instead they have to make sure that they can take advantage of the increasingly parallel computing resources. In this scenario, the effort required for an efficient implementation of analytical gradients is even greater, because the code should also be parallelized. Even in the best circumstances, the performance improvement due to parallelism is often only modest. Conversely, an efficient computation of numerical gradients in parallel is almost trivial to implement, and can be applied to any electronic structure method with very little effort. This may make the use of numerical gradients more appealing, where not very long ago it would be out of question.

Another problem faced by geometry optimization methods is the large dimensionality of the systems. As the number of atoms increases, so does the number of coordinates and degrees of freedom, and locating the optimum structure becomes more difficult, not to mention the fact that the number of minima on the PES grows exponentially. A common method to overcome this problem is reducing the

[a] *M. Stenrup, R. Lindh, I. Fdez. Galván*
Department of Chemistry – Ångström, The Theoretical Chemistry Programme, Uppsala University, P.O. Box 518, SE-751 20 Uppsala, Sweden
E-mail: Ignacio.Fernandez@kemi.uu.se

[b] *M. Stenrup, R. Lindh, I. Fdez. Galván*
Uppsala Center for Computational Chemistry – UC$_3$, Uppsala University, P.O. Box 518, SE-751 20 Uppsala, Sweden

Contract grant sponsor: Swedish strategic research programme eSSENCE and Swedish Research Council (grant number 2012-3910)

dimensionality of the system by applying constraints or "freezing" some degrees of freedom that are believed to be of secondary importance for the problem under study. This may be the case of the internal geometries of ligands in metal–ligand complexes or bulky substituents in organic molecules. The structures obtained from such a constrained geometry optimization will be approximations to the true energy minima (i.e., the minima obtained at the unconstrained level), but they can be refined with a final unconstrained optimization if greater accuracy is needed. A number of methods have been proposed to perform constrained geometry optimizations.[7–10] However, to the best of our knowledge, there has been no attempt to apply a similar reduction of dimensionality to the computation of numerical gradients. A recent method published by Vysotskiy et al.[11] takes a similar approach, but instead of explicitly calculating the gradient it fits the PES on a local grid over the reduced dimensions.

In this work, we show how the constraints used to simplify the optimization problem can also be exploited to reduce the number of energy calculations required to compute the numerical gradient. In addition, we propose a method to combine numerical and analytical differentiation, allowing unconstrained optimizations but restricting the number of energy evaluations. Together with the obvious parallelizability of the numerical differentiation, these methods can render optimizations with numerical gradients a viable option for high-level electronic structure calculations.

We have tested the performance of the proposed methods on several sets of systems, composed of more or less independent fragments (individual molecules or ligands), by comparing the optimized geometries and energies with those obtained with a standard optimization method.

Method

As shown in previous works on projected constrained optimization,[9,10] an optimization with a set of nonlinear constraints can be formulated in terms of a unitary matrix $\boldsymbol{T}$ that separates the coordinate space, $\boldsymbol{q}$, into two orthogonal subspaces: one that (to first order) modifies the constraints, and one that does not. Briefly, the optimization problem can be stated as finding a point $\boldsymbol{q}^*$ that minimizes the energy $E(\boldsymbol{q})$, subject to a set of constraints expressed as:

$$r_i(\boldsymbol{q}^*)=0 \tag{1}$$

If the total number of degrees of freedom is N and the number of independent constraints is m, the matrix $\boldsymbol{T}$, of size $N\times N$, can be constructed as a combination of two smaller matrices:

$$\boldsymbol{T}=\overbrace{(\,\boldsymbol{T}_\mathrm{c}\,)}^{m}\overbrace{(\,\boldsymbol{T}_\mathrm{u}\,)}^{N-m}\}N \tag{2}$$

where $\boldsymbol{T}_\mathrm{c}$ is of size $N\times m$ and defines the *constrained* subspace, and $\boldsymbol{T}_\mathrm{u}$ is of size $N\times(N-m)$ and defines the *unconstrained* or free subspace. Note that the operation in (2) is a simple juxtaposition and not a matrix product. The values for $\boldsymbol{T}_\mathrm{c}$ and $\boldsymbol{T}_\mathrm{u}$ can be found considering that:

$$\boldsymbol{T}_\mathrm{c}^t\frac{\partial r_i}{\partial\boldsymbol{q}}\neq 0,\quad \boldsymbol{T}_\mathrm{u}^t\frac{\partial r_i}{\partial\boldsymbol{q}}=0\qquad \forall i\in\{1,\ldots,m\} \tag{3}$$

In other words, $\boldsymbol{T}_\mathrm{c}$ is the subspace spanned by the different $\partial r_i/\partial\boldsymbol{q}$ vectors, and $\boldsymbol{T}_\mathrm{u}$ is orthogonal to $\boldsymbol{T}_\mathrm{c}$. As the constraints are, in general, nonlinear in the coordinates $\boldsymbol{q}$, the values of $\partial r_i/\partial\boldsymbol{q}$, and, therefore, of $\boldsymbol{T}$, depend on $\boldsymbol{q}$ and must be calculated for each point on the PES.

A given optimization step $\Delta\boldsymbol{q}$ can be decomposed in two separate steps in the constrained and unconstrained subspaces, respectively:

$$\Delta\boldsymbol{c}=\boldsymbol{T}_\mathrm{c}^t\Delta\boldsymbol{q},\quad \Delta\boldsymbol{u}=\boldsymbol{T}_\mathrm{u}^t\Delta\boldsymbol{q} \tag{4}$$

such that they can be recombined to form the original step:

$$\Delta\boldsymbol{q}=\boldsymbol{T}_\mathrm{c}\Delta\boldsymbol{c}+\boldsymbol{T}_\mathrm{u}\Delta\boldsymbol{u} \tag{5}$$

The step $\Delta\boldsymbol{c}$ is computed to satisfy the constraints, that is:

$$r_i(\boldsymbol{q}+\boldsymbol{T}_\mathrm{c}\Delta\boldsymbol{c})=0 \tag{6}$$

and if the constraints are fulfilled at the point $\boldsymbol{q}$, then $\Delta\boldsymbol{c}=0$; in the rest of the paper, we assume this is the case (which corresponds to "frozen" degrees of freedom), although it does not change the discussion. In addition, to first order, the constraints are not modified by changes in the unconstrained step $\Delta\boldsymbol{u}$:

$$r_i(\boldsymbol{q}+\boldsymbol{T}_\mathrm{u}\Delta\boldsymbol{u})=r_i(\boldsymbol{q})\qquad \forall i\in\{1,\ldots,m\},\ \Delta\boldsymbol{u} \tag{7}$$

Similarly, the gradient vector can be separated in the two subspaces:

$$\frac{\partial E}{\partial\boldsymbol{q}}=\nabla E(\boldsymbol{q})=\boldsymbol{g}(\boldsymbol{q})=\boldsymbol{T}_\mathrm{c}\boldsymbol{g}_\mathrm{c}+\boldsymbol{T}_\mathrm{u}\boldsymbol{g}_\mathrm{u} \tag{8}$$

$$\boldsymbol{g}_\mathrm{c}=\boldsymbol{T}_\mathrm{c}^t\boldsymbol{g}(\boldsymbol{q}),\quad \boldsymbol{g}_\mathrm{u}=\boldsymbol{T}_\mathrm{u}^t\boldsymbol{g}(\boldsymbol{q}) \tag{9}$$

and a constrained optimization tries to find a stationary point in the *unconstrained* subspace, that is:

$$\boldsymbol{g}_\mathrm{u}^*=\boldsymbol{T}_\mathrm{u}^t\boldsymbol{g}(\boldsymbol{q}^*)=0 \tag{10}$$

This already suggests that only the $(N-m)$-dimensional vector $\boldsymbol{g}_\mathrm{u}$ would need to be known, rather than the N-dimensional vector $\boldsymbol{g}$. In the algorithms proposed,[9,10] the gradient is used as the product $\boldsymbol{T}_\mathrm{u}^t\boldsymbol{g}$, while the full gradient $\boldsymbol{g}$ appears only as an ingredient of the Hessian update in quasi-Newton methods. Thus, ignoring the Hessian update, the optimization process would not change if instead of the full gradient we compute a partial gradient $\tilde{\boldsymbol{g}}$ such that $\boldsymbol{T}_\mathrm{u}^t\tilde{\boldsymbol{g}}=\boldsymbol{T}_\mathrm{u}^t\boldsymbol{g}$. The most trivial way of defining such a partial gradient is simply:

$$\tilde{\boldsymbol{g}}_\mathrm{u}=\boldsymbol{g}_\mathrm{u},\quad \tilde{\boldsymbol{g}}_\mathrm{c}=0\quad\rightarrow\quad \tilde{\boldsymbol{g}}=\boldsymbol{T}_\mathrm{u}\boldsymbol{g}_\mathrm{u} \tag{11}$$

but using (9) to compute $\boldsymbol{g}_\mathrm{u}$ is not practical, as that would require knowing the full gradient $\boldsymbol{g}$.

With numerical differentiation, it is possible to calculate directly the components of $\boldsymbol{g}_\mathrm{u}$, by computing the energy at displaced geometries along the unit vectors of the unconstrained subspace:

$$g_{\mathrm{u},i}(\boldsymbol{q}) \simeq \frac{E(\boldsymbol{q}+h\boldsymbol{T}_{\mathrm{u}}\mathrm{e}_{\mathrm{u},i})-E(\boldsymbol{q}-h\boldsymbol{T}_{\mathrm{u}}\mathrm{e}_{\mathrm{u},i})}{2h} \tag{12}$$

where h is the displacement length and $\mathrm{e}_{\mathrm{u},i}$ is the ith canonical basis vector of $\mathbb{R}^{N-m}$, or, more practically, $\boldsymbol{T}_{\mathrm{u}}\mathrm{e}_{\mathrm{u},i}$ is the ith column of $\boldsymbol{T}_{\mathrm{u}}$.

The method to obtain the partial gradient $\tilde{\boldsymbol{g}}$ can then be stated as:

1. For a given geometry $\boldsymbol{q}$, compute the derivatives of the constraints $\partial \mathrm{r}/\partial \boldsymbol{q}$ (see Ref. 10) and obtain $\boldsymbol{T}_{\mathrm{u}}$ by orthonormalization, according to (3).
2. Generate the displaced geometries $\boldsymbol{q}\pm h\mathrm{t}_i$, where h is the step size used for numerical differentiation and t_i is the ith column of $\boldsymbol{T}_{\mathrm{u}}$. For practical use, an iterative conversion from internal to Cartesian coordinates may be required (see Ref. 12 for details).
3. For each displaced geometry, compute the energy with the chosen method.
4. Obtain $\boldsymbol{g}_{\mathrm{u}}$ with (12), and the partial gradient $\tilde{\boldsymbol{g}}(\boldsymbol{q}) = \boldsymbol{T}_{\mathrm{u}}\boldsymbol{g}_{\mathrm{u}}$.

The partial gradient $\tilde{\boldsymbol{g}}(\boldsymbol{q})$ can then be used instead of the full gradient $\boldsymbol{g}(\boldsymbol{q})$ for a constrained optimization.[9,10] It is worth noting that once the projected constrained optimization is implemented, the implementation of the above method is straightforward, as the $\boldsymbol{T}$ matrix is already computed, and no change at all is needed in the optimization algorithm (or, indeed, any other code that makes use of the gradient).

The numerical differentiation described above requires only $2(N-m)+1$ energy evaluations, instead of the $2N+1$ that would be required for the full gradient. The reduction is evidently greater the more constraints are applied, and it is particularly important if the system is made out of a small number of rigid fragments, where all the internal degrees of freedom within each fragment are constrained. In this latter case, the number of energy evaluations needed can also be expressed as:

$$2n_{\mathrm{u}}+1; \quad n_{\mathrm{u}} = 6n_{\mathrm{f}}-n_{\mathrm{l}}-3n_{\mathrm{a}}-6 \tag{13}$$

where n_{u} is the number of unconstrained degrees of freedom, n_{f} is the total number of fragments, n_{l} is the number of linear fragments, and n_{a} is the number of single-atom fragments; if the complete system is linear, the above equation should have -5 instead of -6. It is important to note that the number of energy evaluations is linear with respect to the number of fragments (it is at most $12n_{\mathrm{f}}-11$), and does not depend on the size of the fragments.

Composite gradients

In the above formulation, the gradient in the constrained subspace, $\boldsymbol{g}_{\mathrm{c}}$, is not evaluated but set to zero in eq. (11). This takes advantage of the fact that in a constrained optimization $\boldsymbol{g}_{\mathrm{c}}$ will be discarded anyway. But one may use the separation into different subspaces more generally, without performing a constrained optimization. One could, for example, compute $\boldsymbol{g}_{\mathrm{c}}$ at some level of theory, $\boldsymbol{g}_{\mathrm{u}}$ at some other level, and build a composite gradient with (8), which would then be used in place of the true gradient. The effect, when using this composite gradient in an optimization, is that different degrees of freedom are optimized at different levels, without explicitly splitting the system in separate subsystems.

For instance, if a system is made out of several molecules, the intramolecular gradients can be computed at a "cheap" level, such as DFT, while the intermolecular gradient is computed at a level more adequate for describing interactions like dispersion forces (say MP2 or CCSD(T)). The real advantage of this approach comes when analytical gradients are not available for the second method. In this case, numerical differentiation can be done only for the intermolecular degrees of freedom, and the gradient for the intramolecular degrees is obtained analytically with the first method.

For simplicity, we keep the same notation from the previous section, although in this case both the "c" and "u" subspaces are unconstrained. The $\boldsymbol{T}_{\mathrm{c}}$ and $\boldsymbol{T}_{\mathrm{u}}$ matrices are defined as in (3), but the constraints in (1) are only pseudoconstraints or "phantom" constraints, in the sense that they are defined, but never enforced. After the $\boldsymbol{T}_{\mathrm{c}}$ and $\boldsymbol{T}_{\mathrm{u}}$ matrices are obtained, the composite gradient $\boldsymbol{g}^{\mathrm{AB}}$ is constructed as:

$$\boldsymbol{g}^{\mathrm{AB}} = \boldsymbol{T}_{\mathrm{c}}\boldsymbol{T}_{\mathrm{c}}^{t}\boldsymbol{g}^{\mathrm{A}}+\boldsymbol{T}_{\mathrm{u}}\boldsymbol{g}_{\mathrm{u}}^{\mathrm{B}} = \boldsymbol{T}\begin{pmatrix}\boldsymbol{T}_{\mathrm{c}}^{t}\boldsymbol{g}^{\mathrm{A}}\\ \boldsymbol{g}_{\mathrm{u}}^{\mathrm{B}}\end{pmatrix} \tag{14}$$

where $\boldsymbol{g}^{\mathrm{A}}$ is the full gradient computed analytically with some method A, while $\boldsymbol{g}_{\mathrm{u}}^{\mathrm{B}}$ is computed numerically with (12) using some method B. For an otherwise unconstrained optimization, once the composite gradient is obtained, the $\boldsymbol{T}$ matrix is not further needed and can be discarded.

Although the composite gradient is well defined as long as the $\boldsymbol{T}_{\mathrm{c}}$ and $\boldsymbol{T}_{\mathrm{u}}$ matrices are, the energy corresponding to this gradient is not. In an actual implementation, the reported energy at each point of the optimization will be the energy obtained either with method A or with method B, but neither of them matches the composite gradient. This may affect the performance of some optimization algorithms that make use of the energy values (e.g., for adaptively modifying the step length). A possible way to circumvent this problem would involve computing the optimization step separately in each subspace, and applying any step length update method independently; however, that would require a more complicated implementation and additional energy calculations. The present implementation simply uses the composite gradient $\boldsymbol{g}^{\mathrm{AB}}$ as if it were the true one, without any modification to the standard optimization algorithms.

Some practical considerations

The constraints defined in (1) are usually derived exclusively from geometrical parameters: they involve distances and angles between atoms, and the derivatives needed in (3) are readily obtained. In the examples presented in this work, we have always considered the system as formed by several "fragments," and defined the constraints as all the intrafragment degrees of freedom. This results in the optimization of rigid fragments (except for composite gradients, see above).

For defining these intrafragment degrees of freedom, we constructed a Z-matrix[13] for each fragment from their A_n atoms in arbitrary order (as defined in the input). The Z-matrix contains A_i–A_{i-1} distances, $A_i-A_{i-1}-A_{i-2}$ angles and $A_i-A_{i-1}-A_{i-2}-A_{i-3}$ dihedrals, and no effort was done to ensure chemical connectivity or optimize the Z-matrix. As long as this simple definition does not result in ill-conditioned constraints, the $\boldsymbol{T}_\text{c}$ and $\boldsymbol{T}_\text{u}$ matrices will correctly separate the intrafragment and interfragment coordinates. Additional numerical robustness and stability in the algorithm can be obtained by optimizing the Z-matrix or manually defining the intrafragment constraints, but for the purpose of this work this was not needed. Note that for (12) only $\boldsymbol{T}_\text{u}$ is needed, which is defined as the subspace orthogonal to the constraints, so the result does not depend on the particular definition of the constraints, as long as they span the same subspace.

It has been mentioned above that the full gradient is only needed for the Hessian update algorithms. This is especially important when using constrained optimizations as a means of approaching a region where the PES has the correct curvature for a saddle point, eventually performing a transition state optimization without constraints. In this case, one wants the force constant along the constrained coordinate (the putative reaction coordinate) to be updated during the optimization, and for this to happen the gradient in this direction must be evaluated, even if it is not directly used in the optimization. There are other situations where the gradient in a constrained direction is desired, for example when computing a minimum energy path as a series of constrained optimizations, where the gradient at one point defines the constraint for the next point.[14] And in other cases, the constraints themselves, or their derivatives, needed for $\boldsymbol{T}$, depend on the full gradient, such as when using the energy difference between two states as a constraint for optimizing a minimum-energy crossing point.[10] This means that not all kinds of constraints should be considered when evaluating $\boldsymbol{g}_\text{u}$ numerically, and sometimes a particular constraint must be applied in the optimization but ignored for the numerical differentiation (note that this is the opposite case from the composite gradients, as implemented above, where constraints are applied for the differentiation but ignored during the optimization).

Computational Details

The present method for constrained geometry optimization using numerical gradients has been implemented into a development version of the MOLCAS 8 quantum chemistry program.[15] To evaluate the performance of the method, we have computed the equilibrium geometries of several sets of molecular complexes. The first of these sets, Set I, is identical to the S22 set of weakly bound dimers,[16] which has been designed with a variety of intermolecular interactions in mind (hydrogen bonding, π–π stacking, mixed electrostatic/dispersion interactions, etc.). The second set, Set II, is composed of water clusters of various sizes (two, three, and four water molecules). This set was chosen to examine the behavior of the present method with an increasing number of fragments. As a third test set, Set III, we have chosen some of the systems used by Vysotskiy et al. in their study[11] on fragment-based geometry optimization. This includes the ethylene molecule, the water dimer, the formic acid dimer, and the formic acid dimer–water cluster. In addition, we have used as a test case the $Fe(CO)_3[P(CH_3)_3]_2$ transition metal complex (Set IV). This system serves as a complement to the other three benchmark sets, whose members are composed of first-row atoms only. An account of all 30 systems considered in this study is given in Table 1; the table shows the number of energy calculations needed for each numerical gradient evaluation when using constraints (n_E^c, as proposed in this work) and without constraints (n_E^u), together with the corresponding reduction in computational effort: for the largest systems in Set I the gain is more than 90%.

The same general strategy was used for all constrained optimizations in this work. First, the geometry of each complex was optimized using the standard, fully unconstrained approach. The resulting equilibrium structures were then deformed by modifying the relative positions/orientations of the molecules of which they are composed. Finally, the intermolecular coordinates were reoptimized using the present constrained approach as well as the approach described in Ref. 11 [hereafter referred to as the Vysotskiy–Boström–Veryazov (VBV) approach]. The ethylene monomer is clearly a special case. For this system, the division into fragments was made in terms of the two CH_2 groups. Thus, the "intermolecular" interaction in this case is the C=C covalent bond. The $Fe(CO)_3[P(CH_3)_3]_2$ transition metal complex deserves commenting as well. For this system, the constrained optimization was performed with the five ligands and the central metal atom treated as separate fragments. It should be noted, though, that the central metal atom, lacking internal coordinates, is not part of the geometrical constraints.

The deformation of the fully optimized structures was done in different ways depending on the benchmark set. In the case of Set I, we used an automated procedure in which one of the dimer fragments was kept fixed, whereas the other was translated a distance 0.5–1.0 Å in a random direction. For Sets II and IV, the deformation was done manually and involved both translations and rotations of the individual fragments. Finally, in the case of Set III, starting structures for the constrained optimization were taken from Ref. 11 to ease comparison with that work. We note that in this case, the deformation is generally larger than for the other benchmark sets. The Cartesian coordinates of all starting structures are available in the Supporting information.

The energy and gradient calculations were performed with the density-fitted MP2 method,[17] which is able to properly treat dispersion interactions at a reasonable computational cost. The auxiliary basis needed in the fitting procedure was generated on-the-fly using the Cholesky decomposition technique.[18] As for the one-electron basis, the cc-pVDZ basis of

Table 1. Overview of benchmark systems considered in this study.

System	n_f	N	n_E^u	n_E^c	c (%)
Set I					
2-pyridoxine–2-aminopyridine	2	25	139	13	9.4
Adenine–thymine (stack)	2	30	169	13	7.7
Adenine–thymine (WC)	2	30	169	13	7.7
Ammonia dimer	2	8	37	13	35.1
Benzene–ammonia	2	16	85	13	15.3
Benzene dimer (C_{2h})	2	24	133	13	9.8
Benzene dimer (C_{2v})	2	24	133	13	9.8
Benzene–HCN	2	15	79	11	13.9
Benzene–methane	2	17	91	13	14.3
Benzene–water	2	15	79	13	16.5
Ethene dimer	2	12	61	13	21.3
Ethene–ethyne	2	10	49	11	22.4
Formamide dimer	2	12	61	13	21.3
Formic acid dimer	2	10	49	13	26.5
Indole–benzene (stack)	2	28	157	13	8.3
Indole–benzene (T-shape)	2	28	157	13	8.3
Methane dimer	2	10	49	13	26.5
Phenol dimer	2	26	145	13	9.0
Pyrazine dimer	2	20	109	13	11.9
Uracil dimer (HB)	2	24	133	13	9.8
Uracil dimer (stack)	2	24	133	13	9.8
Water dimer	2	6	25	13	52.0
Set II					
Water dimer	2	6	25	13	52.0
Water trimer	3	9	43	19	44.2
Water tetramer	4	12	61	25	41.0
Set III					
Ethylene molecule	2	6	25	13	52.0
Water dimer	2	6	25	13	52.0
Formic acid dimer	2	10	49	13	26.5
Formic acid dimer–water cluster	3	13	73	19	26.0
Set IV					
$Fe(CO)_3[P(CH_3)_3]_2$	6	33	187	49	26.2

Abbreviations: WC, Watson–Crick; HB, hydrogen bond; n_f, number of fragments; N, number of atoms; n_E^u, number of energy evaluations per gradient, without constraints; n_E^c, number of energy evaluations per gradient, with constraints; c, relative cost per gradient (n_E^c/n_E^u).

Dunning was used for Sets I, II, and IV, whereas the corresponding augmented version, aug-cc-pVDZ, was used for most of Set III. The only exception is the ethylene monomer, which was optimized using the ANO-L-VDZP basis set. This is in accordance with Ref. 11.

The unconstrained optimizations were largely performed with the default MOLCAS optimization scheme. However, in a few difficult-to-converge cases, Cartesian optimization coordinates had to be adopted instead of the (default) internal coordinates. Conversely, the calculations performed with the constrained method were always done using both internal and Cartesian coordinates, as well as using a number of different convergence criteria. The default convergence criterion in MOLCAS requires that the root mean square or RMS (max) step length is less than $1.2 \cdot 10^{-3}\, a_0$ ($1.8 \cdot 10^{-3}\, a_0$) and the RMS (max) gradient is less than $3.0 \cdot 10^{-4}\, E_h a_0^{-1}$ ($4.5 \cdot 10^{-4}\, E_h a_0^{-1}$). In addition, we used a modified criterion in which the condition on the step length is not enforced provided that the change in energy is below a certain threshold, τ. Three different values of τ were used: $1.0 \cdot 10^{-6}\, E_h$, $5.0 \cdot 10^{-6}\, E_h$ and $1.0 \cdot 10^{-5}\, E_h$. In the calculations performed with the VBV approach, default settings were used throughout. All geometry optimizations were done without symmetry constraints.

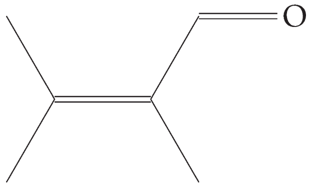

Figure 1. 2,3-dimethyl-2-butenal, “trimethylacrolein.”

The “composite gradient” method (see Composite gradients section) was tested on three different systems: ammonia dimer, stacked adenine–thymine, and trimethylacrolein (2,3-dimethyl-2-butenal, see Fig. 1). For the first two, which were also included in Set I, the cc-pVDZ basis set was used, while the latter used the ANO-RCC basis set, with double-ζ and polarization contraction (ANO-RCC-VDZP).[19] The methods used were MP2 and CCSD(T) for the ammonia dimer, DFT(B3LYP) and MP2 for the adenine–thymine complex, and MP2, CASSCF and CASPT2 for trimethylacrolein. For the multiconfigurational methods (CASSCF and CASPT2), an active space of six electrons in five orbitals was used, including the 4 π and π^* orbitals and the oxygen n orbital perpendicular to the C═O bond.

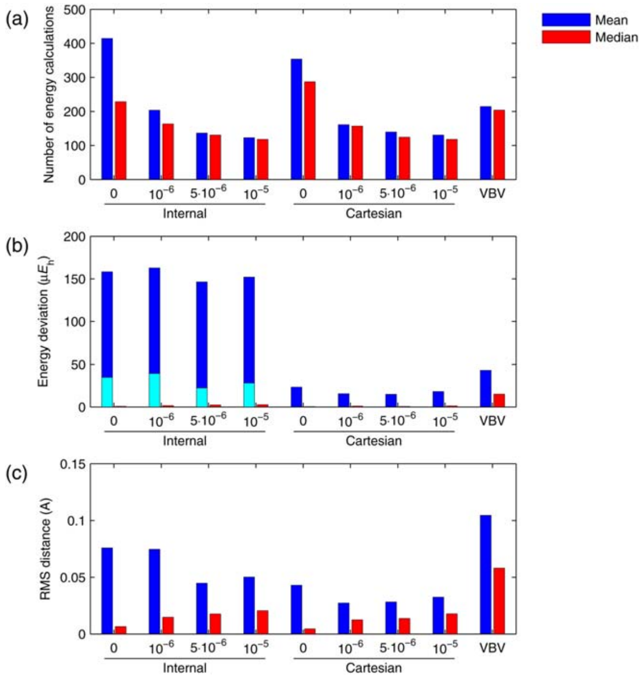

Figure 2. Performance of the present constrained optimization method for the S22 set of weakly bound dimers (Set I). The mean and median number of energy calculations a) is shown together with the mean and median energy deviation b) and RMS distance c) with respect to the fully optimized structures. The numbers appearing on the *x*-axis refer to different values of the energy convergence criterion (in units of E_h). Bars colored in cyan refer to results obtained by removing the formic acid dimer from the benchmark set (see discussion in the main text). Results obtained with the VBV method for a subset of 14 systems are included for comparison. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The CASSCF calculations were done with a state-average of 4 roots, and for CASPT2, the standard IPEA shift of 0.25 E_h was applied, as well as an imaginary shift of 0.1 E_h. Except for the ammonia system, all calculations were performed with the same Cholesky-based density fitting technique indicated above. The optimizations were done using Cartesian coordinates, with an energy threshold τ of $5.0\cdot10^{-6}$ E_h, without symmetry constraints.

Results and Discussion

Constrained optimization using numerical gradients

We begin by examining the performance of the constrained optimization method described in Method section. For this purpose, we use the results of the benchmark calculations outlined in Computational details section. The efficiency of the present method is evaluated based on the number of energy calculations required to optimize the molecular complexes in Sets I–IV, whereas the accuracy is assessed from how much the geometries and energies differ from the full optimization results.

The performance for the weakly bound dimers of Set I is illustrated in Figure 2. Due to the large number of systems in this set, only mean and median values are shown. As can be seen from Figure 2a, the choice of coordinates (internal or Cartesian) has only a moderate effect on the overall efficiency of the optimization procedure. Noteworthy, still, is a tendency for the Cartesian coordinates to give a more uniform convergence behavior across the benchmark set. This is especially so when the default convergence criterion is used, as can be seen by comparing the difference between the mean and median number of energy calculations for the two coordinate types.

An unusually poor convergence was observed when optimizing some of the π–π stacked structures using internal coordinates. Similar although less severe problems were encountered with Cartesian coordinates. In both cases, the gradient was found to converge within only a fraction of the total number of iterations, with the remaining iterations being spent on small geometry updates on an essentially flat PES. Introducing the energy-based convergence criterion described in Computational details section prevents the bulk part of these updates from taking place. As a consequence, the number of energy calculations is significantly reduced.

Incorporating the energy change into the convergence criterion has little effect on the final energies and geometries, as shown by Figures 2b and 2c. In fact, for some complexes such as the π–π stacked structures discussed above, the agreement with the reference may actually improve. Note that the unusually large energy deviation observed when internal coordinates are used is due to one particular complex, the formic acid dimer, alone. The optimization in this case, regardless of the convergence criterion, was found to render the two hydrogen bonds ~0.06 Å too short. The same problem is not present when Cartesian coordinates are used.

Taking into account the full data underlying Figure 2, we find that a good balance between efficiency and accuracy is obtained with Cartesian coordinates and an energy threshold of $1.0\cdot10^{-6}$ E_h to $1.0\cdot10^{-5}$ E_h. With this particular setting, the present method appears to be more cost efficient than the VBV approach. Note, however, that with the latter approach, results were only obtained for 14 of the complexes in Set I. A comparison with the present method restricted to the same subset can be found in Supporting Information Figure S1. As can be seen from that figure, the overall picture is unchanged.

The computational effort of each iteration in the present method scales linearly with the number of fragments, as discussed in Method section. Thus, we expect the real advantage of this method to become clear for systems composed of, say, three fragments or more. That this is indeed the case, and that the linear scaling of the gradient evaluation is not offset by an excess in the number of iterations, is illustrated by the water cluster data in Figure 3. As can be seen from this data, the number of energy calculations increases by a factor of two, roughly, for each step going from the dimer to the trimer to the tetramer. This can be compared with the close to fourfold increase obtained with the VBV method. For the particular case of the tetramer, the present method requires between 600 and 1000 energy calculations depending on which coordinates and convergence criterion are used. The corresponding number required by the VBV method is 6495. We further note

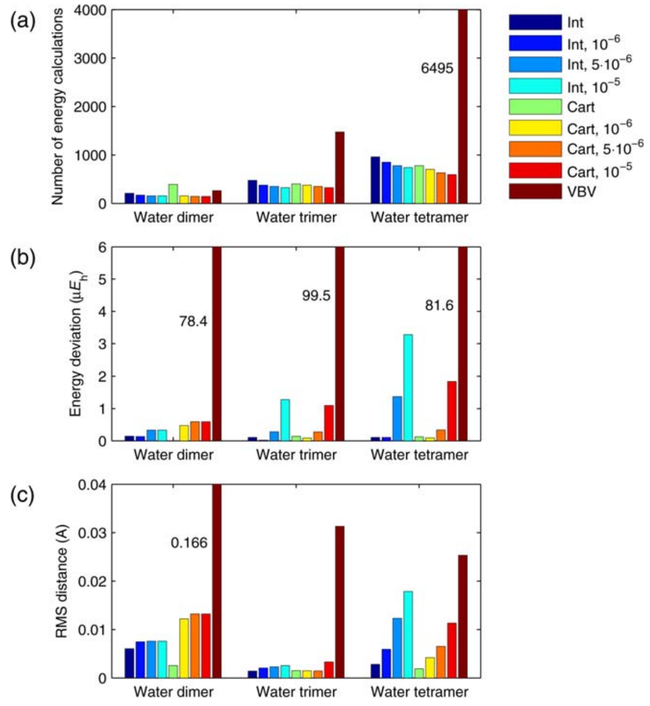

Figure 3. Performance of the present constrained optimization method for water clusters of increasing size (Set II). The number of energy calculations a) is shown together with the energy deviation b) and RMS distance c) with respect to the fully optimized structures. The numbers appearing in the figure legend refer to different values of the energy convergence criterion (in units of E_h). Results obtained with the VBV method are included for comparison.

that the efficiency improvement obtained with the present method does not occur at the cost of a reduced accuracy in the final energies and geometries, as can be seen from Figures 3b and 3c.

Unlike the VBV approach, the present method is designed to be used with a Hessian updating scheme. This is crucial as evaluating the Hessian explicitly would ruin the linear-scaling behavior discussed above. A possible drawback is a less optimal behavior when starting far away from the equilibrium geometry. To get an idea of the extent of this problem, we turn to the data for Set III shown in Figure 4. The starting structures for this set are considerably more deformed than for the other three sets. Comparing with the VBV approach, the present method can be seen to require fewer energy calculations for each system in the test set. This is so regardless of which coordinates and convergence criterion are used. It appears that although an explicit Hessian may well improve convergence in cases like these, the cost of obtaining that Hessian is usually not compensated for by a sufficient gain in the number of iterations.

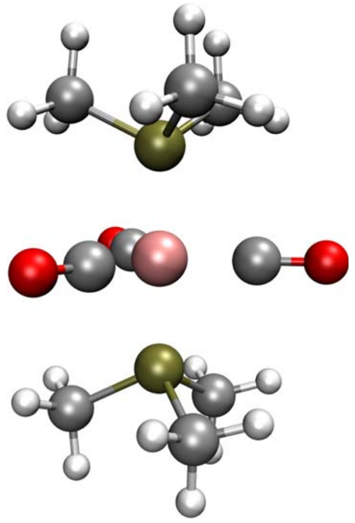

Figure 5. Structure of the $Fe(CO)_3[P(CH_3)_3]_2$ complex. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

As a final illustration of the performance of the present method, we consider the optimization of the $Fe(CO)_3[P(CH_3)_3]_2$ complex displayed in Figure 5. The sheer size of this system, both in terms of the number of electrons and the number of fragments, makes it a challenge to any numerical gradient-based approach. This is true also for the present approach, as can be seen from the number of energy calculations listed in Table 2. We note, however, that without the favorable scaling demonstrated by eq. (13) the optimization of this complex would be an even more daunting task.

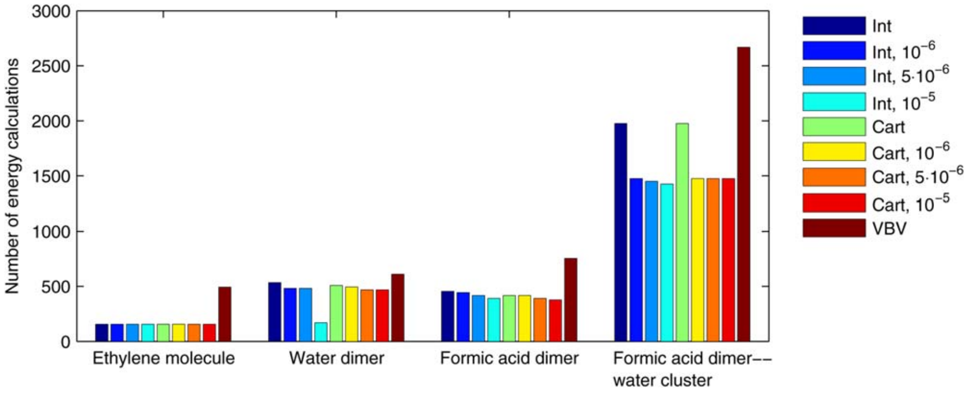

Figure 4. Performance of the present constrained optimization method for various systems taken from Ref. 11 (Set III). The numbers appearing in the figure legend refer to different values of the energy convergence criterion (in units of E_h). Results obtained with the VBV method (taken from Ref. 11) are included for comparison.

Table 2. Performance of the present constrained optimization method (standard convergence criterion) for the $Fe(CO)_3[P(CH_3)_3]_2$ complex (Set IV).

Coordinates	N_E	ΔE	D_{RMS}
Internal	5489	12.9	0.025
Cartesian	4215	41.3	0.044

The number of energy calculations (N_E) is shown together with the energy deviation (ΔE, in μE_h) and RMS distance (D_{RMS}, in Å) with respect to the fully optimized structure.

Table 3. Optimized bond distances (in Å) and angles (in degrees) for the ammonia molecule and ammonia dimer (see Fig. 6), obtained with full optimization at MP2 and CCSD(T) levels.

	NH_3			$(NH_3)_2$	
	MP2	CCSD(T)		MP2	CCSD(T)
N–H	1.023	1.027	N^1–H^2	1.025	1.028
H–N–H	103.9	103.5	N^1–H^3	1.023	1.027
H–N–H–H	108.4	107.8	N^1–H^5	2.444	2.466

Composite gradients

The composite gradient method proposed was tested in three sample systems, as detailed in Computational details section. In this section, the main interest is not to discuss the performance, measured in the number of iterations or energy calculations, but to discuss the effects of using a composite gradient.

The most significant optimized parameters for a single ammonia molecule and the dimer, at MP2 and CCSD(T) levels, are shown in Table 3, the structure of the dimer is displayed in Figure 6. These results correspond to full optimizations using analytical gradients in the case of MP2 and regular numerical differentiation for CCSD(T). It is evident that the main effect of the calculation level is on the intermolecular distance in the dimer, which is ~0.02 Å larger with CCSD(T). If a full optimization at the CCSD(T) level is not affordable or desired (it requires 37 single-point calculations per iteration), a possible strategy is to optimize a single molecule and then optimize the dimer keeping the intramolecular geometries frozen (13 single-point calculations per iteration). This fragment-based optimization can be done using the ammonia geometry optimized at either MP2 or CCSD(T) level and in both cases the resulting N^1–H^5 distance is 2.470 Å, but this approach does not allow the N^1–H^2 distance to respond to the presence of the other molecule.

In contrast, an optimization using composite CCSD(T) and MP2 gradients (analytical MP2 for the intramolecular coordinates, numerical CCSD(T) for the intermolecular) relaxes all coordinates simultaneously. The optimized structure has a N^1–H^2 distance of 1.025 Å and N^1–H^3 of 1.023 Å, which agree with the pure MP2 result, and a N^1–H^5 distance of 2.467 Å, in agreement with the pure CCSD(T) one.

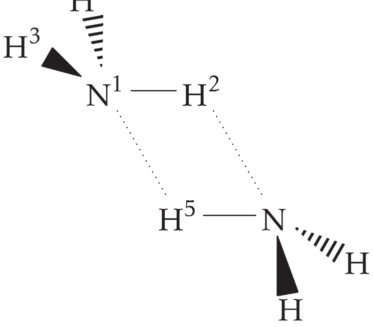

Figure 6. Structure of the ammonia dimer.

Regarding the energies given in Table 4, even though the composite gradient method does not have a well-defined energy, it attains a CCSD(T) energy at the final optimized geometry lower than the CCSD(T) energies at the final geometries of the full MP2 optimization or the optimization with frozen MP2 molecules. The optimization with frozen CCSD(T) molecules reaches a lower energy, but as indicated above it cannot reproduce the changes in the intramolecular geometry when the dimer is formed.

A more interesting situation appears with the stacked adenine–thymine complex. The standard density functionals such as B3LYP do not describe properly the dispersion interactions that are a main factor for the stability of the stacked basis pair complexes and, therefore, a B3LYP optimization results in the two molecules turning and adopting an in-plane conformation instead. With MP2, however, the correct stacked complex can be obtained. Additionally, the formation of the complex can introduce significant changes in the structure of the molecules, especially in the polar groups, so that an optimization of the complex with frozen intramolecular structures obtained in isolation may not be appropriate.

To test the composite gradient method, it was compared with several other optimizations. First, the two molecules were optimized separately at B3LYP and MP2 levels; the stacked complex was then optimized at MP2 level keeping the intramolecular structures frozen at their MP2 or B3LYP geometries; finally, the complex was fully optimized using MP2, and with composite gradients (B3LYP for intramolecular degrees of freedom, MP2 for intermolecular ones). The RMS distance of the different structures are given in Table 5. The complex formation induces a distortion of 0.04–0.06 Å in the bases, according to the MP2 structures. When composite gradients are used and the intramolecular structures are relaxed with B3LYP, the

Table 4. CCSD(T) energies (ΔE, in kJ mol^{-1}) for the ammonia dimer at different geometries.

Inter	Intra	ΔE
Full MP2		0.17
Full CCSD(T)		0.00
CCSD(T)	Frozen MP2	0.21
CCSD(T)	Frozen CCSD(T)	0.10
CCSD(T)	MP2	0.15

The methods used for optimization of the intermolecular and intramolecular degrees of freedom are indicated in the first two columns, the energies are relative to the full CCSD(T) optimization.

Table 5. RMS distance (in Å) of adenine and thymine optimized in different conditions, with respect to the isolated MP2 structures. The last two rows correspond to the molecules in the stacked complex.

Inter	Intra	Adenine	Thymine
–	MP2	0.0000	0.0000
–	B3LYP	0.0270	0.0065
Full MP2		0.0610	0.0403
MP2	B3LYP	0.0897	0.1010

distortion is greater; this is due to the fact that there is no stacked minimum at B3LYP level, so that the distortion increases when the molecules are forced to stay in this conformation.

Table 6 displays the final energies and some representative distances (see Fig. 7) of the different optimized complexes. The distance between the centers of mass, A–T, is quite consistent in all cases, which was to be expected as the intermolecular degrees of freedom are always optimized at MP2 level. Only when composite gradients are used (MP2 and B3LYP) is the distance slightly larger. This is compensated for by the distortion of the molecules mentioned above, which tends to approach the polar substituents between the two molecules, as can be seen in the three interatomic distances given in Table 6. Regarding the energies, it is worth noting that the B3LYP relaxation of the intramolecular geometries is able to decrease the MP2 energy of the complex by 2 kJ mol^{-1}; conversely optimization with frozen MP2 or B3LYP intramolecular geometries results in an energy difference of more than 4 kJ mol^{-1}, although the intermolecular distances are practically the same.

As a final example, trimethylacrolein (Fig. 1) was considered. In this system, the goal is to test the composite gradient method in excited electronic states. The "high level" method is CASPT2, for which analytical gradients are not available in MOLCAS; "low level" methods are CASSCF and MP2. As the active space for CASSCF and CASPT2 calculations spans only the central skeleton ($O{=}C^1{-}C^2{=}C^3$), the multiconfigurational character of these methods will mainly affect this part of the molecule, and the internal degrees of freedom of the CH_3 groups are expected to be relatively unaffected by the electronic state. Therefore, for the calculations using two methods, the methyl groups were defined as "fragments," such that their internal geometries were either frozen or optimized with the low-level method, while all the other degrees of freedom were optimized with CASPT2.

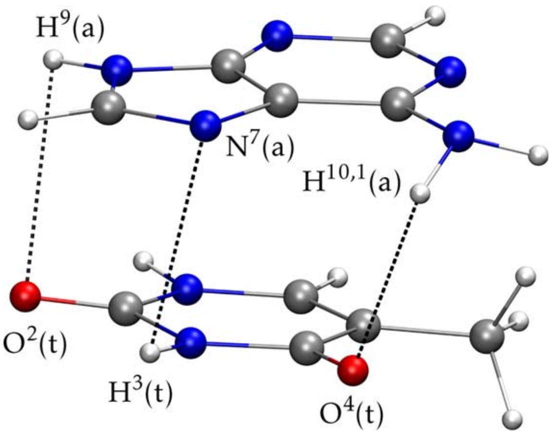

Figure 7. Structure of the adenine–thymine stacked complex. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In the ground state, it is possible to perform a full optimization with MP2 as well as CASSCF and CASPT2. Table 7 shows the energies and geometries for the optimized structures. Compared with the CASPT2 optimization, CASSCF yields better bond lengths and vertical CASPT2 energies than MP2, but the CASPT2 energy at the MP2 geometry is lower than at the CASSCF minimum structure. When the internal CH_3 degrees of freedom are excluded from the CASPT2 optimization, the optimized bond distances and vertical energies remain consistent with the full optimization, validating such an exclusion. The final energies reflect the fact that the MP2 optimization of the methyl groups gives a lower energy than the CASSCF optimization.

When the first excited state is optimized, MP2 is not an option for the full optimization. Table 8 shows the energy and geometry results for different optimizations on the excited state surface, where the "MP2" structures for the methyl groups are either frozen at their ground state values or optimized with their ground state gradients. As in the ground state, the bond lengths and vertical energies of all CASPT2 optimizations are completely stable, and the final energies are lower when combining CASPT2 and MP2 than with CASPT2 and CASSCF.

From the previous results, it would seem that there is no real advantage in optimizing the methyl groups versus keeping them frozen. With CASSCF optimization, the CASPT2 energy of the optimized structure actually increases with respect to the frozen structure, and even with MP2 optimizations the gain is minimal. However, there are other cases where the relaxation of the degrees of freedom not included in the high level optimization may be more important, as shown in the adenine–thymine complex. In any case, excluding the methyl groups from the numerical differentiation at CASPT2 level reduces the required number of energy evaluations per gradient from 91 to 55, with very little effect on the geometries and energies.

In summary, the three examples discussed above show that the proposed composite gradients can be used to optimize simultaneously different degrees of freedom at different computational levels. Obviously, some care should be taken to ensure that the separation of the degrees of freedom and the combination of methods is meaningful for the situation at hand. In some cases, it may be good enough to optimize with frozen fragments; however, the use of composite gradients instead of frozen fragments introduces only a small additional computational effort (the analytical gradient at a low level) and ensures that all the degrees of freedom respond to changes in the rest of the system.

Table 6. MP2 energy (ΔE, in kJ mol^{-1}) and intermolecular distances (in Å) of the adenine–thymine stacked complex at different geometries.

Inter	Intra	ΔE	A–T	H^9–O^2	N^7–H^3	$H^{10,1}$–O^4
Full MP2		0.00	3.105	2.899	2.942	2.586
MP2	Frozen MP2	3.12	3.106	3.073	2.976	2.918
MP2	Frozen B3LYP	7.52	3.103	3.085	2.983	3.010
MP2	B3LYP	5.45	3.138	2.693	2.894	2.606

The methods used for optimization of the intermolecular and intramolecular degrees of freedom are indicated in the first two columns, the energies are relative to the full MP2 optimization. A–T: distance between the centers of mass of the rings, see Figure 7 for other distances.

Table 7. CASPT2 energy (ΔE, in kJ mol^{-1}), vertical absorption energy (T_v, in eV), and bond distances (in Å) of trimethylacrolein at different geometries in the ground state.

Main	CH_3	ΔE	T_v	O–C^1	C^1–C^2	C^2–C^3
Full CASSCF		3.47	3.88	1.209	1.475	1.349
Full MP2		0.91	3.72	1.233	1.469	1.355
Full CASPT2		0.00	3.84	1.219	1.480	1.348
CASPT2	Frozen CASSCF	1.85	3.84	1.219	1.480	1.348
CASPT2	Frozen MP2	0.08	3.84	1.219	1.480	1.348
CASPT2	CASSCF	2.14	3.84	1.219	1.480	1.348
CASPT2	MP2	0.01	3.84	1.219	1.479	1.348

The methods used for optimization of the internal degrees of freedom of the methyl groups and the rest of the molecule is indicated in the first two columns, the energies are relative to the full CASPT2 optimization.

Table 8. CASPT2 energy (ΔE, in kJ mol^{-1}), vertical emission energy (T_v, in eV), and bond distances (in Å) of trimethylacrolein at different geometries in the first excited state.

Main	CH_3	ΔE	T_v	O–C^1	C^1–C^2	C^2–C^3
Full CASSCF		2.92	2.50	1.357	1.369	1.414
Full CASPT2		0.00	2.56	1.352	1.378	1.405
CASPT2	Frozen CASSCF	1.79	2.56	1.352	1.378	1.404
CASPT2	Frozen MP2	0.73	2.56	1.352	1.378	1.404
CASPT2	CASSCF	1.95	2.56	1.352	1.378	1.404
CASPT2	MP2	0.37	2.56	1.352	1.378	1.404

The methods used for optimization of the internal degrees of freedom of the methyl groups and the rest of the molecule is indicated in the first two columns, the energies are relative to the full CASPT2 optimization.

Conclusions

We have implemented and tested a simple way to reduce the number of energy evaluations required to obtain a numerical gradient, whenever there are constraints applied that reduce the number of effective degrees of freedom in the system. The method uses the same linear transformation matrix ***T*** defined for constrained optimizations,[10] such that the additional computations are minimal and no change at all is required in the optimization algorithm. The tests presented in this work show that, by performing this partial numerical differentiation, the accuracy of the final constrained optimization is not significantly affected. Comparison with a method with similar features[11] shows a competitive performance based on the total number of energy evaluations and accuracy of the final result, while the present method is expected to have a better scaling as the number of fragments (or unconstrained degrees of freedom) increases.

The present approach does not intend to be a replacement for the more conventional optimizations with analytical gradients, when they are available. However, the combination of extensive constraints (rigid fragments) and partial numerical differentiation can be attractive for optimizations with electronic structure methods that lack an analytical derivative implementation, especially if large parallel-computing resources can be used, as the multiple energy evaluations required can be trivially performed in parallel. Whether or not adding a particular set of constraints to a system affects the validity of the results is something that must be considered for each case, depending on the system and properties to be studied, and that is beyond the scope of this work.

We have also proposed a method that combines numerical differentiation for a limited set of degrees of freedom and analytical derivatives for the rest, creating a "composite gradient." This allows the same reduction in the number of energy evaluations needed, without the requirement of freezing fragments or coordinates, effectively relaxing different degrees of freedom at different computation levels. The examples studied prove that with an adequate selection of methods and separation of coordinates, meaningful results can be obtained at a reduced cost.

In summary, we believe that the methods presented in this work will help extending the application of state-of-the-art theoretical methods to larger, more realistic chemical systems including bulkier groups and substituents.

Acknowledgment

The computations for this work were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Uppsala University (UPPMAX), Lund University (LUNARC) and Linköping University (NSC).

Keywords: numerical gradients • constrained geometry optimization • MOLCAS

How to cite this article: M. Stenrup, R. Lindh, I. Fdez. Galván. *J. Comput. Chem.* **2015**, *36*, 1698–1708. DOI: 10.1002/jcc.23987

Additional Supporting Information may be found in the online version of this article.

[1] B. H. Schlegel, *WIREs Comput. Mol. Sci*. **2011**, *1*, 790.
[2] M. B. Ruiz, P. Otto, Y. G. Smeyers, *J. Mol. Struct. (Theochem)* **1996**, *365*, 151.
[3] R. M. Lewis, V. Torczon, M. W. Trosset, *J. Comput. Appl. Math*. **2000**, *124*, 191.
[4] U. M. García-Palomares, J. F. Rodríguez, *SIAM J. Opt*. **2002**, *13*, 79.
[5] T. G. Kolda, R. M. Lewis, V. Torczon, *SIAM Rev*. **2003**, *45*, 385.
[6] L. Han, M. Neumann, *Optim. Methods Softw*. **2006**, *21*, 1.
[7] D.-H. Lu, M. Zhao, D. G. Truhlar, *J. Comput. Chem*. **1991**, *12*, 376.
[8] J. Baker, *J. Comput. Chem*. **1992**, *13*, 240.
[9] J. M. Anglada, J. M. Bofill,*J. Comput. Chem*. **1997**, *18*, 992.
[10] L. De Vico, M. Olivucci, R. Lindh, *J. Chem. Theory Comput*. **2005**, *1*, 1029.
[11] V. P. Vysotskiy, J. Boström, V. Veryazov, *J. Comput. Chem*. **2013**, *34*, 2657.
[12] V. Bakken, T. Helgaker, *J. Chem. Phys*. **2002**, *117*, 9160.
[13] F. Jensen, Introduction to Computational Chemistry; Wiley: Chichester, UK, **1999**; pp. 414–421.
[14] C. González, H. B. Schlegel, *J. Chem. Phys*. **1989**, *90*, 2154.
[15] F. Aquilante, L. De Vico, N. Ferré, G. Ghigo, P.-Å. Malmqvist, P. Neogrády, T. B. Pedersen, M. Pitoňák, M. Reiher, B. O. Roos, L. Serrano-Andrés, M. Urban, V. Veryazov, R. Lindh, *J. Comput. Chem*. **2010**, *31*, 224.
[16] P. Jurečka, J. Šponer, J. Černý, P. Hobza, *Phys. Chem. Chem. Phys*. **2006**, *8*, 1985.
[17] J. Boström, V. Veryazov, F. Aquilante, T. B. Pedersen, R. Lindh, *Int. J. Quantum Chem*. **2014**, *114*, 321.
[18] F. Aquilante, R. Lindh, T. B. Pedersen, *J. Chem. Phys*. **2007**, *127*, 114107.
[19] B. O. Roos, R. Lindh, P.-Å. Malmqvist, V. Veryazov, P.-O. Widmark, *J. Phys. Chem. A* **2004**, *108*, 2851.

Received: 25 February 2015
Revised: 26 May 2015
Accepted: 28 May 2015
Published online on 3 July 2015