

Methods for optimizing large molecules

Part III.† An improved algorithm for geometry optimization using direct inversion in the iterative subspace (GDIIS)‡ §

Ödön Farkas*^{ab} and H. Bernhard Schlegel^b

^a Department of Organic Chemistry, Eötvös Loránd University, H-1518 Budapest, 112. PO Box 32, Hungary. E-mail: farkas@para.chem.elte.hu

^b Department of Chemistry, Wayne State University, Detroit, MI, 48202, USA. E-mail: hbs@chem.wayne.edu

Received 24th September 2001, Accepted 1st November 2001

First published as an Advance Article on the web 17th December 2001

The geometry optimization using direct inversion in the iterative subspace (GDIIS) has been implemented in a number of computer programs and is found to be quite efficient in the quadratic vicinity of a minimum. However, far from a minimum, the original method may fail in three typical ways: (a) convergence to a nearby critical point of higher order (*e.g.* transition structure), (b) oscillation around an inflection point on the potential energy surface, (c) numerical instability problems in determining the GDIIS coefficients. An improved algorithm is presented that overcomes these difficulties. The modifications include: (a) a series of tests to control the construction of an acceptable GDIIS step, (b) use of a full Hessian update rather than a fixed Hessian, (c) a more stable method for calculating the DIIS coefficients. For a set of small molecules used to test geometry optimization algorithms, the controlled GDIIS method overcomes all of the problems of the original GDIIS method, and performs as well as a quasi-Newton RFO (rational function optimization) method. For larger molecules and very tight convergence, the controlled GDIIS method shows some improvement over an RFO method. With a properly chosen Hessian update method, the present algorithm can also be used in the same form to optimize higher order critical points.

Introduction

Geometry optimization is an essential part of most computational chemistry studies involving electronic structure methods.¹ It is often a major component of the computational expense of such studies. Therefore, efficient geometry optimization algorithms are very desirable. It is now well established that internal coordinates are more efficient than Cartesian coordinates for geometry optimization of most molecular systems, and that redundant internal coordinates are particularly useful.² Recent advances have overcome the $O(N^3)$ bottleneck in transforming Cartesian coordinates to internals, and these transformations can be carried out in a fashion that asymptotically scales linearly with size for most systems.³ A second well established aspect of geometry optimization is that gradient-based methods are, in general, much more cost effective than algorithms that require only the function values, or that require the calculation of the full matrix of second derivatives.⁴ Of the gradient-based algorithms, quasi-Newton methods have been used most widely.¹ These typically require $O(N^2)$ memory to maintain an updated Hessian. However, there are limited memory quasi-Newton schemes with satisfactory performance but memory requirements that scale as $O(N)$ with the system size.⁵ Conjugate gradient methods⁴ also have memory requirements that scale linearly, but their performance is generally not as good as those of quasi-Newton methods. Geometry optimization using direct inversion of the

iterative subspace^{6,7} is an alternative to quasi-Newton and conjugate gradient methods for large molecules that has received much less attention.

The performance of GDIIS is similar to quasi-Newton methods and in some cases, such as relatively flat regions of the potential energy surface, may even be superior. With an approximate Hessian that is diagonal or sparse, the memory requirements scale linearly; hence, this approach is suitable for large molecules. However, in its original formulation,⁶ GDIIS can have a number of problems. There is a tendency to converge to the nearest critical point, even if it is of different order than desired (*e.g.* a transition structure rather than a minimum). Inflection points (*e.g.* shoulders) on potential energy surfaces may cause endless oscillations. Numerical problems can arise in solving for the coefficients. The present paper outlines a modified GDIIS algorithm⁷ that overcomes these difficulties, and demonstrates its behavior for a number of test cases.

The present algorithm has been extensively used since 1993, when it was implemented into MOPAC 5.0⁸ to carry out not only semiempirical but also *ab initio* calculations through a general driver to external programs; more recently, it has also been implemented in Gaussian 98,⁹ where it is part of the default algorithm to carry out semiempirical optimizations and optional for other electronic structure methods. The present paper serves to document the algorithm.

Method

Regular GDIIS

The GDIIS method is based on a linear interpolation (and extrapolation) of the available structures that minimizes the

† For Part II see ref. 3(b).

‡ Electronic Supplementary Information available. See <http://www.rsc.org/suppdata/cp/b1/b108658h/>

§ In memory of Richárd Hargitai.

length of an error vector.

$$\mathbf{x}^* = \sum c_i \mathbf{x}_i, \quad \text{where } \sum c_i = 1 \quad (1)$$

For each structure \mathbf{x}_i , an error vector is constructed using a quadratic approximation to the potential energy surface. Given an estimated Hessian, \mathbf{H} , a Newton–Raphson optimization step yields \mathbf{x}_i^{SR} .

$$\mathbf{x}_i^{\text{SR}} = \mathbf{x}_i + \mathbf{H}^{-1} \mathbf{f}_i \quad (2)$$

where \mathbf{f}_i is the force (negative of the gradient) at \mathbf{x}_i . The error vector \mathbf{e}_i is chosen as the displacement in this simple relaxation (SR) step:

$$\mathbf{e}_i = \mathbf{x}_i^{\text{SR}} - \mathbf{x}_i = \mathbf{H}^{-1} \mathbf{f}_i \quad (3)$$

In the quadratic approximation, the error or *residuum* vector for \mathbf{x}^* is the linear combination of the individual error vectors,

$$\mathbf{r} = \sum c_i \mathbf{e}_i = \mathbf{e}^* \quad (4)$$

The coefficients c_i are obtained by minimizing $|\mathbf{r}|^2$ with the constraint $\sum c_i = 1$. This least-squares problem leads to the following set of equations:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k,1} & \cdots & a_{k,k} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_k \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (5)$$

where matrix \mathbf{A} is defined as $a_{ij} = \mathbf{e}_i^T \mathbf{e}_j$ and λ is the Lagrangian multiplier arising from the previously mentioned constraint. These coefficients determine the intermediate point \mathbf{x}^* that minimizes the length of the residuum vector. The residuum vector is equal to the simple relaxation step from \mathbf{x}^* , and the next point in the optimization is given by

$$\mathbf{x}_{k+1} = \mathbf{x}^* + \mathbf{r} = \sum c_i (\mathbf{x}_i + \mathbf{e}_i) = \sum c_i \mathbf{x}_i^{\text{SR}} \quad (6)$$

The new point can also be regarded as the same linear combination of the predicted geometries, \mathbf{x}_i^{SR} , produced by simple relaxation from the earlier points.

Difficulties with the regular GDIIS

The regular GDIIS method tends to converge to the closest stationary point. This behavior can be anticipated from the fact that the residuum vector can also be written in terms of an interpolated force $\mathbf{r} = \mathbf{H}^{-1} \sum c_i \mathbf{f}_i$ (for a quadratic surface, $\mathbf{r} = \mathbf{H}^{-1} \mathbf{f}^*$, where \mathbf{f}^* is the force at \mathbf{x}^*). A minimum in \mathbf{r} is achieved when the force vanishes, *i.e.* at any critical point—minimum, transition state, second-order saddle point, *etc.* In this respect, the behavior of the regular GDIIS is similar to the gradient norm method.¹⁰ Certain types of inflection points can also cause problems—points where the gradient is non-zero, but its magnitude is a local minimum. Such a point corresponds to a shoulder along a reaction path, as shown in Fig. 1. If the gradient at this inflection point is larger than the convergence threshold, the optimization will not terminate. In practice, the regular GDIIS algorithm may oscillate about this point.

As the optimization proceeds and more points are collected, the error vectors may become nearly linearly dependent. If there is a linear dependence among the error vectors, then $\mathbf{e}_j = \sum_{i \neq j} b_i \mathbf{e}_i$ for some j . Hence one can choose the coefficients c_i such that $\mathbf{r} = \sum c_i \mathbf{e}_i = 0$ for $\sum c_i = 1$. A value of $\mathbf{r} = 0$ would falsely indicate that the optimization has converged, even though the components of the error vectors perpendicular to the subspace spanned are not zero. Furthermore, linear dependences among the \mathbf{e}_i result in a matrix \mathbf{A} that is singular, *i.e.* $|\mathbf{r}|^2 = (\sum c_i \mathbf{e}_i)^2 = 0 = \sum_{ij} c_i c_j \mathbf{e}_i^T \mathbf{e}_j = \sum_{ij} c_i c_j a_{ij}$. Thus, for numerical stability, one needs to avoid situations

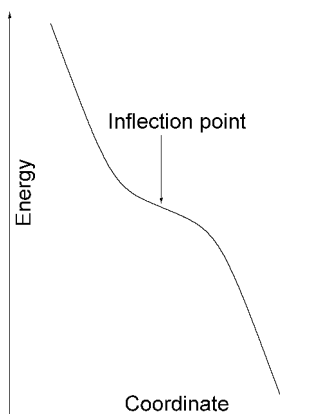


Fig. 1 A potential energy curve with an inflection point, illustrating that the gradient (first derivative) is a local minimum at an inflection point but does not vanish.

where the error vectors are nearly linearly dependent. One remedy is to discard some or all of the previous points when a near-linear dependence is detected and to restart the GDIIS procedure.

Improvements to GDIIS

The GDIIS method is very effective in interpolating within the quadratic region of the desired stationary point. When the current point is outside the quadratic region, a method is needed to control the GDIIS procedure so that it steps reliably toward the stationary point of interest. This is done by comparing the GDIIS step with a reference step.

A simple relaxation step will move toward the desired stationary point, provided that the approximate Hessian, \mathbf{H} , is appropriate for the optimization (for a minimization, no negative eigenvalues; for a transition state search, one negative eigenvalue and a corresponding eigenvector suitable for the reaction, *etc.*). Let this reference step be

$$\Delta \mathbf{x}^{\text{ref}} = \mathbf{x}_k^{\text{SR}} - \mathbf{x}_k = \mathbf{H}^{-1} \mathbf{f}_k \quad (7)$$

where \mathbf{x}_k is the current point. In some cases, the Hessian may have the wrong number of negative eigenvalues, or the resulting step may be too large. In general, it is best to control the step using an RFO¹¹ or trust radius method. In this case, the Hessian in eqn. (7) is replaced by an effective Hessian, $\mathbf{H}^{\text{eff}} = \mathbf{H} + \lambda \mathbf{I}$, where λ is chosen to control the step size and direction. We use this effective Hessian not only for the reference step, but also to compute the error vectors (eqn. (3)).

For a given collection of points, a number of criteria can be developed to determine if a GDIIS step is acceptable. If not, then the number of points used to construct the GDIIS step can be changed, and the step can be tested again. The following four criteria are used to determine whether a particular GDIIS step is acceptable.

(a) The direction of the GDIIS step, $\Delta \mathbf{x}^{\text{GDIIS}} = \mathbf{x}_{k+1} - \mathbf{x}_k$, (eqn. (6)) can be compared to the reference step (eqn. (7)).

$$\cos(\theta) = (\Delta \mathbf{x}^{\text{GDIIS}})^T \cdot \Delta \mathbf{x}^{\text{ref}} / |\Delta \mathbf{x}^{\text{GDIIS}}| |\Delta \mathbf{x}^{\text{ref}}| \quad (8)$$

If the angle θ is greater than a specified value, the GDIIS step is not acceptable. The suitable values for the cut-off depend on the number of points employed in the GDIIS calculation and the expected accuracy of the estimated Hessian. In the present implementation, the cut-offs for $\cos(\theta)$ are 0.97, 0.84, 0.71, 0.67, 0.62, 0.56, 0.49, 0.41 for 2–9 vectors, respectively, and 0.00 for 10 or more vectors.

(b) The length of the GDIIS step is limited to be no more than 10 times the reference step.

(c) The magnitude of the largest GDIIS coefficients, c_i , is an indicator of the quality of the GDIIS step. If one or more of the coefficients are large in absolute value, then the step involves a substantial extrapolation. Also, when the GDIIS step involves many points, each of the coefficients, c_i , may have a reasonable magnitude, yet the net result could be a large extrapolation. The sum of all of the positive coefficients (or of all of the negative coefficients) is a good measure of the extrapolation. The GDIIS step is considered not acceptable if this value exceeds 15.

(d) If \mathbf{A} is nearly singular, $|r|^2$ becomes small and $c/|r|^2$ (see eqn. (9) below) becomes large, signaling numerical stability problems. If the magnitude of $c/|r|^2$ exceeds 10^8 , then the step is assumed to be unacceptable. To avoid the effect of the size of the error vectors, we rescale them for the purpose of solving the DIIS equations so that the smallest error vector has unit length.

In the present implementation, we start with the current point and add the most recent points one at a time until we run out of points or the GDIIS step becomes unacceptable. If the step has become unacceptable, then the last acceptable GDIIS step is used. In general all of the previous points (up to a maximum number) are retained so that in future steps the procedure can make the most use of the information available. However, if the angle between the GDIIS step and the reference step is greater than 90° (eqn. (8)), we permanently discard the offending point and all earlier points in the optimization. If less than three points are used for the current GDIIS step, then the step can be replaced by a linear search (a constrained quartic polynomial fitted to the energies and gradients is employed for the line search, the same as used in regular gradient based geometry optimization¹²). Otherwise, the reference step is used (eqn. (7)).

Solving the GDIIS equations

For the actual computation of the DIIS coefficients, it is convenient to re-arrange eqn. (5). Since it can be easily shown that $\lambda = |r|^2$, eqn. (5) can be re-written as:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,k} \end{pmatrix} \begin{pmatrix} c_1/|r|^2 \\ \vdots \\ c_k/|r|^2 \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad (9)$$

The dimension of \mathbf{A} is relatively small and the solution can be obtained by any of a number of standard methods. However, a sequence of these equations needs to be solved as additional error vectors are incorporated, increasing the dimension of \mathbf{A} by one each time. Hence, iterative methods and Gauss elimination have some advantages since they are able to reuse information from the previous step in the sequence.

Hessian updating

The GDIIS method can perform satisfactorily with a constant Hessian. For an N -dimensional quadratic potential energy surface and $N+1$ points that span the space, GDIIS will predict the exact minimum independent of the value of the Hessian (provided the Hessian is not singular). However, updating the Hessian in a fashion similar to quasi-Newton methods can improve the performance. The BFGS¹³ updating formula

$$\begin{aligned} \mathbf{H}_k &= \mathbf{H}_{k-1} + \Delta\mathbf{H}_k^{\text{BFGS}} \\ \Delta\mathbf{H}_k^{\text{BFGS}} &= -\left(\frac{\Delta\mathbf{f}_k \Delta\mathbf{f}_k^T}{\mathbf{s}_{k-1}^T \Delta\mathbf{f}_k} + \frac{\mathbf{H}_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T \mathbf{H}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{H}_{k-1} \mathbf{s}_{k-1}} \right) \\ \Delta\mathbf{f}_k &= \mathbf{f}_k - \mathbf{f}_{k-1} \end{aligned} \quad (10)$$

is frequently used for minimization; the Powell symmetric-Broyden¹⁴ (PSB)

$$\begin{aligned} \Delta\mathbf{H}_k^{\text{PSB}} &= (\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T \mathbf{s}_{k-1} \frac{\mathbf{s}_{k-1} \mathbf{s}_{k-1}^T}{(\mathbf{s}_{k-1}^T \mathbf{s}_{k-1})^2} \\ &\quad - \frac{(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1}) \mathbf{s}_{k-1}^T + \mathbf{s}_{k-1} (\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T}{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}} \end{aligned} \quad (11)$$

and symmetric rank one¹⁵ (SR1) methods are suitable for transition state searches.

$$\Delta\mathbf{H}_k^{\text{SR1}} = -\frac{(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T}{(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T \mathbf{s}_{k-1}}, \quad (12)$$

Bofill has developed a weighted combination of PSB and SR1 that is superior for transition states.¹⁶

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \varphi^{\text{Bofill}} \Delta\mathbf{H}_k^{\text{SR1}} + (1 - \varphi^{\text{Bofill}}) \Delta\mathbf{H}_k^{\text{Powell}} \quad (13)$$

$$\varphi^{\text{Bofill}} = \frac{((\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T \mathbf{s}_{k-1})^2}{(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T (\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1}) \mathbf{s}_{k-1}^T \mathbf{s}_{k-1}} \quad (14)$$

We have found that a similar weighted combination of BFGS and SR1 provides an improvement over pure BFGS or SR1 for minimization.^{3b} Furthermore, we have also found that the square root of the original Bofill weighting is better.

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \varphi \Delta\mathbf{H}_k^{\text{SR1}} + (1 - \varphi) \Delta\mathbf{H}_k^{\text{BFGS}}, \quad (15)$$

$$\varphi = \sqrt{\varphi^{\text{Bofill}}} = \sqrt{\frac{((\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T \mathbf{s}_{k-1})^2}{(\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1})^T (\Delta\mathbf{f}_k + \mathbf{H}_{k-1} \mathbf{s}_{k-1}) \mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}} \quad (16)$$

For large systems, storage and manipulation of a full Hessian leads to $O(N^2)$ memory and cpu usage. The simplest approach is to use a diagonal or sparse Hessian that is held constant through the optimization. This is appropriate for large molecular mechanics calculations and the micro-iterations in combined QM/MM calculations. Since the energy and gradient evaluations are inexpensive, a small increase in the number of iterations is a good trade-off for linear scaling in memory and cpu. For large semi-empirical optimizations, it may be more economical to use a limited memory BFGS method⁵ that also scales linearly with system size.

Combining GDIIS with other optimization methods

The GDIIS procedure can be readily generalized to encompass a continuous weighting between GDIIS and any other optimization scheme. Let $\mathbf{x}_k^{\text{opt}}$ be the optimized structure predicted by the other optimization scheme. Then we can define another set of error vectors.

$$\mathbf{e}'_i = \mathbf{x}_k^{\text{opt}} - \mathbf{x}_i \quad (17)$$

Note that solving the GDIIS equations with \mathbf{e}'_i will always yield $\mathbf{x}_k^{\text{opt}}$. A new set of error vectors can then be constructed as a linear combination:

$$\mathbf{e}''_i = (1 - \alpha) \mathbf{e}_i + \alpha \mathbf{e}'_i \quad (0 \leq \alpha \leq 1) \quad (18)$$

With these error vectors, the GDIIS procedure will interpolate between regular GDIIS and the other optimization method.

Discussion

For the purpose of validating our changes to the original algorithm we performed several test calculations on a set of 37 molecules ranging in size up to 642 atoms. Most of the test cases were taken from Baker's test set,¹⁷ which has already

been widely used for testing geometry optimization methods.^{2d,3a,b,17} The additional four molecules, taxol, oligopeptides of 10 and 20 alanine residues and crambin, were picked from the test set we used for testing the performance of our $O(N^2)$ scaling optimization algorithm.^{3a,b} While regular optimization criteria are often quite satisfactory for most applications, very tight convergence more readily illustrates the difficulties that optimization algorithms can encounter. Therefore, all optimizations were performed using convergence criteria of 10^{-6} au for the root mean square gradient and 1.5×10^{-6} au for the maximum component of the gradient vector, 4×10^{-6} au for the root mean square displacement in Cartesian coordinates and 6×10^{-6} au for the maximum component of the displacement. All test calculations were performed using redundant internal coordinates with the default diagonal starting Hessian within the large molecular optimization framework of Gaussian as described in ref. 3a and 3b.

We compared three optimization algorithms (see Table 1), (1) regular GDIIS, (2) quasi-Newton RFO with combined BFGS+SR1 Hessian update and line search, and (3) controlled GDIIS using error vectors produced by the RFO based method. The storage and use of a large number of earlier geometries and forces might become a bottleneck for larger molecules. Consequently, we limited all three optimization algorithms to store no more than 10 of the earlier points. For all methods, the optimizations on methylamine, acanil01, benzidine, pterine, mesityl oxide and caffeine molecules from Baker's test set converge to higher order critical points due to the symmetry of the initial structures; therefore, symmetry was turned off for these structures. Nevertheless, except for pterine, the structures still optimized to higher order critical points of the corresponding potential energy surfaces because the high symmetry of the starting structures with vanishing forces pointing toward the minimum.

In any suitable collection of test molecules, there will be some that misbehave. These may be representative of the difficult cases encountered in actual research studies. Therefore, in the remainder of this discussion we will focus on these cases. However, we note that for the rest of the test cases, all tested optimization options, including the simple GDIIS optimization procedure provided similar results. The detailed results are available as Electronic Supplementary information.†

Pterine

With regular convergence criteria or with use of symmetry, all tested optimization techniques converge to a transition struc-

ture. However, with no symmetry and using very tight convergence criteria, the RFO-based optimization and the controlled GDIIS can bypass converging to the transition structure and find the minimum. Regular GDIIS, as can be expected, stays in the vicinity of the transition structure and rapidly converges to it.

Histamine H+ and hydrazobenzene

The regular GDIIS method converges slowly to transition structures for both systems. The two other methods show similar performance and converge to the correct minima.

ACTHCP and taxol

The regular GDIIS method enters into endless oscillation for both molecules due to inflection points on the potential energy surfaces. The RFO-based method and controlled GDIIS converge to the same minima in a similar number of steps.

For-Ala₁₀-NH₂, For-Ala₂₀-NH₂ and crambin

Regular GDIIS cannot achieve convergence because of oscillations. The RFO-based method converges to the minima but needs 10–25% more steps than the controlled GDIIS method. These examples indicate that the controlled GDIIS is an efficient choice for optimizing large molecules.

Conclusions

The present paper discussed the algorithmic details of a controlled GDIIS optimization technique that provides a robust method suitable for optimizing large molecules. The method has been used extensively in a local version of MOPAC and in Gaussian 98, where it is the default geometry optimization scheme for semiempirical methods. A significant advantage of GDIIS is that it can be readily implemented with linear scaling, using a limited amount of memory and cpu, and that it is much less sensitive to the actual Hessian than Newton–Raphson-based optimization methods. The performance of the method depends on a number of control parameters, and may be sensitive to the fine-tuning of these values. The present algorithm is shown to be efficient for optimizing large molecules, suggesting that further development of iterative subspace methods would be advantageous.

Table 1 Comparison of the number of optimization steps taken by regular GDIIS, quasi-Newton RFO and controlled GDIIS algorithms^a

Molecules and theoretical method	Optimization algorithm					
	Regular GDIIS		Quasi-Newton RFO		Controlled GDIIS	
	<i>N</i>	Energy/au	<i>N</i>	Energy/au	<i>N</i>	Energy/au
Pterine (STO-3G)	12 ^b	−569.8488410	36	−569.8538323	36	−569.8538323
Histamine H+ (STO-3G)	85 ^b	−353.9035357	24	−353.9587458	25	−353.9587458
Hydrazobenzene (STO-3G)	41 ^b	−563.2052728	25	−563.2615830	25	−563.2615830
ACTHCP (STO-3G)	— ^c	Oscillation	31	−838.9053214	32	−838.9053214
Taxol (AM1) ^d	— ^c	Oscillation	64	−0.6670634	67	−0.6670634
For-(Ala) ₁₀ -NH ₂ (AM1) ^e	— ^c	Oscillation	67	−0.7331317	59	−0.7331317
For-(Ala) ₂₀ -NH ₂ (AM1) ^f	— ^c	Oscillation	103	−1.4265856	93	−1.4265856
Crambin (MM/UFF)	— ^c	Oscillation	190	0.6556034	150	0.6556034

^a Number of optimization steps (*N*) to reach very tight convergence criteria (see text); starting geometries taken from ref. 17 except as noted. ^b Attempted minimization yielded a transition structure. ^c Attempted minimization resulted in oscillations. ^d Starting geometry from ref. 3(b). ^e Starting geometry “For-Ala₁₀-NH₂ #2” in ref. 3(b). ^f Starting geometry “For-Ala₂₀-NH₂ #1” in ref. 3(b).

Acknowledgements

This work has been supported by Gaussian, Inc., the Hungarian Research Foundation (OTKA D-29446) and the National Science Foundation (CHE 9874005). The authors would also like to thank NSF, MTA and OTKA for an international collaboration grant. Ö. F. thanks the Hungarian Academy of Sciences for a János Bolyai Research Scholarship and also I. G. Csizmadia, P. Császár, G. Fogarasi and P. Pulay for fruitful discussions and for their support.

References

- 1 For reviews see: (a) H. B. Schlegel, in *Modern Electronic Structure Theory*, ed. D. R. Yarkony, World Scientific, Singapore, 1995, pp. 459–500; (b) H. B. Schlegel, in *Encyclopedia of Computational Chemistry*, ed. P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer and P. R. Schreiner, Wiley, New York, 1998, vol. 2, p. 1136.
- 2 (a) P. Pulay, G. Fogarasi, F. Pang and J. E. Boggs, *J. Am. Chem. Soc.*, 1979, **101**, 2550; (b) G. Fogarasi, X. Zhou, P. W. Taylor and P. Pulay, *J. Am. Chem. Soc.*, 1992, **114**, 8191; (c) P. Pulay and G. Fogarasi, *J. Chem. Phys.*, 1992, **96**, 2856; (d) C. Peng, P. Y. Ayala, H. B. Schlegel and M. J. Frisch, *J. Comput. Chem.*, 1996, **17**, 49; (e) J. Baker, A. Kessi and B. Delley, *J. Chem. Phys.*, 1996, **105**, 192; (f) M. v. Arnim and R. Ahlrichs, *J. Chem. Phys.*, 1999, **111**, 9183.
- 3 (a) Ö. Farkas and H. B. Schlegel, *J. Chem. Phys.*, 1998, **109**, 7100; (b) Ö. Farkas and H. B. Schlegel, *J. Chem. Phys.*, 1999, **111**, 10806; (c) B. Paizs, G. Fogarasi and P. Pulay, *J. Chem. Phys.*, 1998, **109**, 6571; (d) J. Baker, D. Kinghorn and P. Pulay, *J. Chem. Phys.*, 1999, **110**, 4986; (e) S. R. Billeter, A. J. Turner and W. Thiel, *Phys. Chem. Chem. Phys.*, 2000, **2**, 2177; (f) B. Paizs, J. Baker, S. Suhai and P. Pulay, *J. Chem. Phys.*, 2000, **113**, 6566; (g) K. Németh, O. Coulaud, G. Monard and J. G. Ángyán, *J. Chem. Phys.*, 2001, **114**, 9747.
- 4 R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, 1981 and similar texts on optimization algorithms.
- 5 (a) D. C. Liu and J. Nocedal, *Math. Prog.*, 1989, **45**, 503; (b) T. H. Fischer and J. Almlöf, *J. Phys. Chem.*, 1992, **96**, 9768.
- 6 (a) P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393; (b) P. Pulay, *J. Comput. Chem.*, 1982, **3**, 556; (c) P. Császár and P. Pulay, *J. Mol. Struct. THEOCHEM*, 1984, **114**, 31.
- 7 Ö. Farkas, PhD (CSc) Thesis, Eötvös Loránd University and Hungarian Academy of Sciences, Budapest, 1995 (in Hungarian).
- 8 (a) *MOPAC, version 5.00*, Frank. J. Seiler Res. Lab., U.S. Air Force Academy, QCPE #455, Colorado Springs, Co 80840; (b) *MOPAC, version 5.50*, modified version by Ö. F. for internal use at the Eötvös Loránd University, Budapest, 1994.
- 9 *Gaussian 98 (Revision A.7)*, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, Ö. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komáromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P. M. W. Gill, B. G. Johnson, W. Chen, M. W. Wong, J. L. Andres, M. Head-Gordon, E. S. Replogle and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1998.
- 10 J. W. McIver, Jr. and A. Komornicki, *J. Am. Chem. Soc.*, 1972, **94**, 2625.
- 11 (a) A. Banerjee, N. Adams, J. Simons and R. Shepard, *J. Phys. Chem.*, 1985, **89**, 52; (b) J. Simons and J. Nichols, *Int. J. Quantum Chem., Quantum Chem. Symp.*, 1990, **24**, 263.
- 12 H. B. Schlegel, *J. Comput. Chem.*, 1982, **3**, 214.
- 13 (a) C. G. Broyden, *J. Inst. Math. Appl.*, 1970, **6**, 76; (b) R. Fletcher, *Comput. J.*, 1970, **13**, 317; (c) D. Goldfarb, *Math. Comput.*, 1970, **24**, 23; (d) D. F. Shanno, *Math. Comput.*, 1970, **24**, 647.
- 14 (a) M. J. D. Powell, in *Nonlinear Programming*, ed. J. B. Rosen, O. L. Mangasarian and K. Ritter, Academic Press, New York, 1970, p. 31; (b) M. J. D. Powell, *Math. Prog.*, 1971, **1**, 26.
- 15 B. Murtagh and R. W. H. Sargent, *Comput. J.*, 1972, **13**, 185.
- 16 J. M. Bofill, *J. Comput. Chem.*, 1994, **15**, 1.
- 17 J. Baker, *J. Comput. Chem.*, 1993, **14**, 1085.