Methods for optimizing large molecules. II. Quadratic search

Ödön Farkas[a)]
Department of Organic Chemistry, Eötvös Loránd University, H-1518 Budapest, 112, P.O. Box 32, Hungary and Department of Chemistry, Wayne State University, Detroit, Michigan 48202

H. Bernhard Schlegel[b)]
Department of Chemistry, Wayne State University, Detroit, Michigan 48202

(Received 28 July 1999; accepted 30 September 1999)

Geometry optimization has become an essential part of quantum-chemical computations, largely because of the availability of analytic first derivatives. Quasi-Newton algorithms use the gradient to update the second derivative matrix (Hessian) and frequently employ corrections to the quadratic approximation such as rational function optimization (RFO) or the trust radius model (TRM). These corrections are typically carried out via diagonalization of the Hessian, which requires $O(N^3)$ operations for N variables. Thus, they can be substantial bottlenecks in the optimization of large molecules with semiempirical, mixed quantum mechanical/molecular mechanical (QM/MM) or linearly scaling electronic structure methods. Our $O(N^2)$ approach for solving the equations for coordinate transformations in optimizations has been extended to evaluate the RFO and TRM steps efficiently in redundant internal coordinates. The regular RFO model has also been modified so that it has the correct size dependence as the molecular systems become larger. Finally, an improved Hessian update for minimizations has been constructed by combining the Broyden–Fletcher–Goldfarb–Shanno (BFGS) and (symmetric rank one) SR1 updates. Together these modifications and new methods form an optimization algorithm for large molecules that scales as $O(N^2)$ and performs similar to or better than the traditional optimization strategies used in quantum chemistry. © *1999 American Institute of Physics.* [S0021-9606(99)30648-6]

I. INTRODUCTION

Developments in quantum chemistry continue to provide more accurate tools for calculating the properties of molecules, studying chemical reactions, and interpreting a wide range of experiments. Improvements in calculational methods and computer hardware allow these calculations to be applied to increasingly larger organic molecules, inorganic complexes and biomolecules. Optimization of the structures of large molecules by quantum chemical methods requires stable, efficient, and therefore, elaborate optimization algorithms,[1] usually in the framework of an over-complete, redundant internal coordinate system.[2,3(c)] Since the derivatives of the energy are calculated with respect to the Cartesian coordinates[4] of the nuclei, they must be transformed into the internal coordinate system[5] to carry out the optimization process. As described in our previous paper,[6] we have developed an $O(N^2)$ method for these coordinate transformations suitable for the optimization of large molecules. This approach is available in the GAUSSIAN 98[7] quantum-chemical program package. Alternative coordinate transformation methods for nonredundant *natural*[3] or *delocalized*[8] internal coordinate systems have also been introduced by Pulay, Paizs, Baker, and co-workers.[9]

Quasi-Newton algorithms are very efficient for finding minima, particularly when inexpensive gradients (first derivatives) are available. These methods employ a quadratic model of the potential-energy surface to take a series of steps that converges to a minimum. The optimization is generally started with an approximate Hessian (second derivative matrix) which is updated at each step using the computed gradients. The stability and rate of convergence of quasi-Newton methods can be improved by controlling the step size, using methods such as rational function optimization[10] (RFO) or the trust radius model[1,11] (TRM). The RFO or TRM equations are normally solved by diagonalization of the Hessian. The $O(N^3)$ computational effort required for this is usually negligible compared to typical quantum-chemical computations. However, it can become a bottleneck in the optimization of larger molecules when semiempirical or linearly scaled electronic structure methods are used.

In this paper we briefly review various aspects of current quasi-Newton optimization methods: The quadratic approximation and the Newton–Raphson step, prevailing methods for updating the Hessian, and control of the step size using RFO and TRM. We then introduce our new contributions: An RFO step independent of size of the system, RFO correction independent of the coordinate system, modification of our $O(N^2)$ equation solver to calculate the combined RFO/TRM step, and an improved Hessian updating scheme for minimizations. Lastly, the various features of the optimization procedure are tested on a range of molecules.

[a)]Electronic mail: farkas@para.chem.elte.hu; URL: http://organ.elte.hu/farkas
[b)]Electronic mail: hbs@chem.wayne.edu; URL: http://chem.wayne.edu/schlegel

A. Quasi-Newton optimization methods

Gradient optimization methods[1,11] are based on a quadratic approximation to the potential-energy surface (PES)

$$E^* = E - \mathbf{f}^T\mathbf{s} + \tfrac{1}{2}\mathbf{s}^T\mathbf{H}\mathbf{s}, \tag{1}$$

where E^* is the predicted energy at a step $\mathbf{s}$ from the current point, E and $\mathbf{f}$ are the energy and force (negative of the gradient) calculated at the current point and $\mathbf{H}$ is the (approximate) Hessian which is updated during the optimization. The related linear approximation to the forces gives the prediction for the force, $\mathbf{f}^*$, at a step $\mathbf{s}$ from the current point:

$$\mathbf{f}^* = \mathbf{f} - \mathbf{H}\mathbf{s}. \tag{2}$$

At a stationary point (e.g., minimum, transition state, etc.), the forces vanish. Within the quadratic approximation, the step to the stationary point is:

$$\mathbf{s}^{\mathrm{NR}} = \mathbf{H}^{-1}\mathbf{f}. \tag{3}$$

Equation (3) is the definition of the well-known Newton–Raphson (NR) or quasi-Newton optimization step, $\mathbf{s}^{\mathrm{NR}}$, which is repeated until the optimization converges. If the quadratic approximation described the PES accurately and the exact Hessian were used, then Eq. (3) would reach the stationary point in one step. However, the PES is not quadratic and the actual Hessian provides a good approximation to the PES only in the vicinity of the current point, which may be far from a stationary point (it may not even have the correct structure for the desired optimization—zero negative eigenvalues for minimization, one negative eigenvalue for transition structure searches).

The calculation of the Hessian can be very expensive, specially for large systems. Consequently, a quasi-Newton optimization usually starts with an approximate Hessian matrix that has the required structure, and then improves the Hessian with an updating procedure during the course of the optimization.[11] The BFGS update[12] is particularly successful for minimization because it ensures that the Hessian remains positive definite. For transition state searches, the update should not force the Hessian to be positive definite. The SR1 update (also known as the Murtagh–Sargent, update)[13] is appropriate, as is the Powell-symmetric-Broyden[14] (PSB) update. Bofill developed an improved update[15] by combining PSB with SR1 in a manner that avoids the problem with the denominator of the latter.

A simple quasi-Newton optimization based on a quadratic model with updated Hessians and Newton–Raphson steps can readily encounter difficulties. Even when the Hessian has the right structure (e.g., positive definite for minimizations), care must be taken not to step too far, since the quadratic model is accurate for only a small region of the potential-energy surface. Both RFO and TRM can be regarded as methods for controlling the step size and have proven to be valuable components of optimization algorithms. TRM and RFO based minimization algorithms can be regarded as Newton–Raphson methods with a corrected Hessian

$$\mathbf{s}^{\lambda} = (\mathbf{H} + \lambda\mathbf{S})^{-1}\mathbf{f}, \tag{4}$$

where λ is a non-negative scalar, $\mathbf{S}$ is usually simply a constant scalar times the unit matrix, $\xi\mathbf{I}$, and $\mathbf{s}^{\lambda}$ is the resulting corrected step.

1. Trust radius model (TRM)

The goal of TRM in minimization is to find the lowest energy point within a suitable trust radius, in the framework of the quadratic approximation. This condition is equivalent to finding a step, $\mathbf{s}^{\lambda}$, such that it is parallel to the predicted force, $\mathbf{f}^{\lambda}$, points in the same direction ($\lambda \geqslant 0$) and has a length no greater than the trust radius, τ

$$\lambda\mathbf{s}^{\lambda} = \mathbf{f}^{\lambda} = \mathbf{f} - \mathbf{H}\mathbf{s}^{\lambda}, \quad |\mathbf{s}^{\lambda}| \leqslant \tau. \tag{5}$$

Note that this is equivalent to Eq. (4) if $\mathbf{S}$ is replaced by the unit matrix. Equation (5) has only one non-negative solution for λ when the Hessian is positive definite (λ is zero if the length of the Newton–Raphson step is less than the trust radius). If the Hessian has one or more negative eigenvalues, the only acceptable solution for minimization is always larger than the negative of the lowest eigenvalue, $\epsilon_{\mathrm{lowest}}$. Combining these conditions yields

$$\lambda \geqslant \mathrm{Max}(0, -\epsilon_{\mathrm{lowest}}). \tag{6}$$

The TRM method usually results in more efficient step size control than a simple scaling of the Newton–Raphson step.

2. Rational function optimization (RFO)

In the RFO approach for minimization, the quadratic model is modified so that a suitably controlled step toward the minimum is obtained. By including a step size dependent scaling denominator, the RFO method contains a self-consistent trust radius

$$E^* = E - \frac{\mathbf{f}^T\mathbf{s} - (1/2)\mathbf{s}^T\mathbf{H}\mathbf{s}}{1 + \mathbf{s}^T\mathbf{S}\mathbf{s}}, \tag{7}$$

where $\mathbf{S}$ is a symmetric, usually diagonal scaling matrix. The expression of the force vector in terms of the RFO model is

$$\mathbf{f}^* = \frac{\mathbf{f} - \mathbf{H}\mathbf{s} + 2(E^* - E)\mathbf{S}\mathbf{s}}{1 + \mathbf{s}^T\mathbf{S}\mathbf{s}} = \frac{\mathbf{f} - \mathbf{H}\mathbf{s}}{1 + \mathbf{s}^T\mathbf{S}\mathbf{s}} + \frac{2\mathbf{f}^T\mathbf{s} - \mathbf{s}^T\mathbf{H}\mathbf{s}}{(1 + \mathbf{s}^T\mathbf{S}\mathbf{s})^2}\mathbf{S}\mathbf{s}. \tag{8}$$

For the RFO step, $\mathbf{s}^{\mathrm{RFO}}$, which satisfies the stationary condition, $\mathbf{f}^* = 0$, the predicted energy lowering is $E^* - E = -\frac{1}{2}\mathbf{f}^T\mathbf{s}^{\mathrm{RFO}}$ (i.e., same form as for the Newton–Raphson step in the quadratic approximation). Thus the RFO step can be calculated by solving the implicit formula

$$\mathbf{s}^{\mathrm{RFO}} = [\mathbf{H} + (\mathbf{f}^T\mathbf{s}^{\mathrm{RFO}})\mathbf{S}]^{-1}\mathbf{f} = (\mathbf{H} + \lambda\mathbf{S})^{-1}\mathbf{f}, \tag{9}$$

where $\lambda = \mathbf{f}^T\mathbf{s}^{\mathrm{RFO}}$. In practice, $\mathbf{S}$ is chosen to be the identity ($\mathbf{I}$), a scalar time the identity ($\xi\mathbf{I}$), or some other diagonal matrix, yielding an RFO correction that is a simple diagonal shift of the Hessian. Equation (9) can be expressed in the eigenvector space of the Hessian, where the $\mathbf{H} + \lambda\mathbf{S}$ matrix is diagonal and its inverse can be given explicitly. Some implementations split the eigenvector space into subspaces for the ''high'' and ''low'' eigenvalues and solve Eq. (9) for them

separately. This approach allows the extension of the RFO method for transition state optimizations and is known as eigenvector following[16] (EF) optimization.

Equation (9) is in implicit form; therefore, the solution can only be obtained by iterating until λ converges. For minimizations, λ should obey condition (6) to ensure the positive definiteness of $\mathbf{H}+\lambda\mathbf{S}$. Constraints on λ for transition state optimizations are discussed elsewhere.[10(b)] Some important features of the RFO approach for minimization are:

(a) smooth convergence to the quadratic approximation in the vicinity of critical points,

(b) automatic scaling down of displacements corresponding to small eigenvalues of a positive definite Hessian,

(c) scaling up of displacement(s) corresponding to negative eigenvalues of the Hessian and directing the step toward a minimum,

(d) avoiding problems when the Hessian is nearly singular, such as in the vicinity of inflection points.

B. Size independent rational function optimization (SIRFO)

The RFO model tends to over-correct the quadratic approximation as the number of dimensions grows. To illustrate the size dependency of the original RFO model, we define a simple quadratic PES with a diagonal Hessian, a positive constant times the identity matrix

$$\mathbf{H}=\beta\mathbf{I}. \tag{10}$$

One of the goals for convergence is to reduce the root-mean-square (rms) force, $\tilde{f}$, below a fixed threshold. The rms force is

$$\tilde{f}=\sqrt{\frac{\mathbf{f}^T\mathbf{f}}{N}}, \tag{11}$$

where N is the number of variables. The expression for calculating the RFO step in this example is

$$\lambda=\mathbf{f}^T\mathbf{s}^{\mathrm{RFO}}=\frac{N\tilde{f}^2}{\beta+\lambda}, \tag{12}$$

where $\mathbf{S}=\mathbf{I}$. The only acceptable, positive solution for λ is

$$\lambda=\frac{-\beta+\sqrt{\beta^2+4N\tilde{f}^2}}{2}. \tag{13}$$

It is clear, that when N approaches infinity λ approaches $\sqrt{N}\tilde{f}$. Therefore, in larger systems and with noticeable forces on the nuclei, the RFO correction can become dominant. The choice of $S=(\xi/\sqrt{N})I$, where ξ is a suitable constant, makes the model size independent. Our implementation in GAUSSIAN 98 uses $\xi=1$. Equation (13) gives a general lower limit for λ when β is not smaller than the largest eigenvalue of an arbitrary positive definite Hessian. The effect of the size independent scaling on the efficiency of the optimization process is shown in Table I.

C. Transformation to avoid problems with RFO and redundant coordinates

Geometry optimizations in a complete set of appropriate internal coordinates usually converge significantly faster than in Cartesian coordinates.[2,17] A suitable internal coordinate system, even if it is based on a local linear combination of individual internal coordinates (like the *natural* internal coordinate system) may contain redundancy. The RFO correction depends on the coordinate system and can be overestimated because of the redundancy. Problems with the redundancy can be avoided by determining the RFO correction in Cartesian coordinates. The relationship between the Cartesian and internal coordinates ($\mathbf{q}_x$ and $\mathbf{q}_{\mathrm{int}}$, respectively) can be expressed in terms of the Wilson B-matrix[18]

$$d\mathbf{q}_{\mathrm{int.}}=\mathbf{B}d\mathbf{q}_x, \tag{14}$$

where $d\mathbf{q}_{\mathrm{int.}}$ and $d\mathbf{q}_x$ are infinitesimal changes in internal and Cartesian coordinates, respectively, and $\mathbf{B}$ contains the derivatives of the internal coordinate values with respect to the Cartesian coordinates. The connection between the internal forces, $\mathbf{f}_{\mathrm{int.}}$, and Cartesian forces, $\mathbf{f}_x$, is

$$\mathbf{f}_x=\mathbf{B}^T\mathbf{f}_{\mathrm{int.}}. \tag{15}$$

The expression for Newton–Raphson step, $\mathbf{s}_{\mathrm{int.}}$, in terms of internal coordinates forces, $\mathbf{f}_{\mathrm{int.}}$, and Hessian, $\mathbf{H}_{\mathrm{int.}}$, is

$$\mathbf{s}_{\mathrm{int.}}=\mathbf{H}_{\mathrm{int.}}^{-1}\mathbf{f}_{\mathrm{int.}}, \tag{16}$$

which can be transformed into the following equation:

$$\mathbf{B}^{-1}\mathbf{s}_{\mathrm{int.}}=(\mathbf{B}^T\mathbf{H}_{\mathrm{int.}}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{f}_{\mathrm{int.}}. \tag{17}$$

The use of the notation $\mathbf{s}_x$ for $\mathbf{B}^{-1}\mathbf{s}_{\mathrm{int.}}$ and $\mathbf{H}_x$ for $(\mathbf{B}^T H_{\mathrm{int.}}\mathbf{B})$ gives

$$\mathbf{s}_x=\mathbf{H}_x^{-1}\mathbf{f}_x, \tag{18}$$

(note that because of the curvilinear nature of the internal coordinates, $\mathbf{H}_x$ is not the Cartesian Hessian when $\mathbf{f}\neq\mathbf{0}$). The RFO correction is applied to Eq. (18) and the displacement is transformed back to internal coordinates

$$\mathbf{s}_{\mathrm{int.}}=\mathbf{B}\mathbf{s}_x. \tag{19}$$

Alternatively, the RFO procedure could be made coordinate system independent by using $\mathbf{S}=\xi\mathbf{G}^{-1}=\xi(\mathbf{B}\mathbf{B}^T)^{-1}$ instead of $\mathbf{S}=\xi\mathbf{I}$ in the denominator in the rational function expression for the energy, Eq. (7).

D. An O(N^2) algorithm for calculating the RFO/TRM step

The RFO/TRM step is usually obtained by diagonalizing the Hessian and solving

$$\mathbf{f}=(\mathbf{H}+\lambda\mathbf{S})\mathbf{s}, \tag{20}$$

in the eigenvector space of the Hessian. Since diagonalization is an O(N^3) process, this can be a serious bottleneck for the optimization of large molecules. Alternatively, our O(N^2) equation solving method that we have employed for the coordinate transformations can be used to obtain the RFO/TRM step. The starting value for λ is taken from the previous optimization step or can be initialized (e.g., $\lambda=\mathbf{f}^T\mathbf{f}$)

TABLE I. Number of cycles required to optimize the test molecules using different sets of options.

	Method options		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	
	SR1-BFGS		−	−	−	−	−	−	−	+	+	+	+	+	+	+	
	BFGS (Ref. 12)		+	+	+	+	+	+	+	−	−	−	−	−	−	−	
	Full history update		+	+	−	+	−	+	−	+	+	−	+	−	+	−	
	RFO (Ref. 10)		+	+	+	−	−	−	−	+	+	+	−	−	−	−	
	SIRFO		−	−	−	+	+	+	+	−	−	−	+	+	+	+	
	TRM (max. component/rms)		−	−	−	+	+	+	+	−	−	−	+	+	+	+	
	Fast equation solving		−	+	+	+	+	+	+	−	+	+	+	+	+	+	
	Other options[a]		+	+	+	+	+	−		+	+	+	+	+	−	−	
	Molecule	N_{atoms}	Number of optimization cycles														Energy (Hartree)
	$\Sigma(1–30)$[b,c]		183	180	186	180	184	181	184	179	176	178	176	177	178	179	
31	ACTHCP	16	28	29	26	29	28	27	26	27	28	26	29	25	26	26	−838.905 321 4
32	histamine H+	18	18	19	19	19	22	20	22	16	17	17	19	20	19	21	−353.958 745 8
33	hydrazobenzene	26		17	16	15	15	19	18		17	17	15	13	19	19	−563.261 583 0
			25							24							−563.263 803 6
	$\Sigma(1–33)$[b]		254	245	247	243	249	247	250	246	238	238	239	235	242	245	
34	C60	60	9	8	8	8	8	8	8	10	8	7	8	8	8	8	1.549 590 7
35	taxol	113	58	55	66	51	56	46	60	55	49	50	49	49	46	45	−0.667 063 4
36	For–Ala_5–NH_2	56	30	36	38	35	37	31	34	31	32	32	35	32	32	31	−0.397 760 0
37	For–Ala_{10}–NH_2 #1	106		66	81	74	78	55	67		60	62	61	58	55	63	−0.733 064 5
										140							−0.733 699 6
			88														−0.733 962 1
38	For–Ala_{10}–NH_2 #2	106		61	68	61	68	51	72		46	59	45	58	45	51	−0.733 131 7
			90							88							−0.734 305 8
39	For–Ala_{20}–NH_2 #1	206								115							−1.423 269 7
			99														−1.425 661 7
								69							59	89	−1.425 829 0
				102	124	105	121		126		78	109	74	108			−1.426 585 5
40	For–Ala_{20}–NH_2 #2	206								104							−1.424 399 3
			104														−1.424 808 2
								72	127						58		−1.425 829 0
						104	122						85	104		90	−1.426 585 5
				101	124						71	103					−1.426 824 4
	$\Sigma(34–40)$		478	429	509	438	490	332	494	543	344	422	357	417	303	377	
	$\Sigma(1–40)$		732	674	756	681	739	579	744	789	582	660	596	652	545	622	
41	RNA+water molecules	368	⋯	⋯		⋯			805	⋯	⋯		⋯				−1.020 335 4
								585									−1.023 630 0
														468			−1.025 104 7
							796										−1.025 887 2
																756	−1.028 741 8
															603		−1.028 750 9
												761					−1.029 757 0
					785												−1.035 861 2
42	crambin[d]	642	⋯	⋯	269	⋯	281	196	475	⋯	⋯	231	⋯	241	129	259	0.755 720 0
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	

[a]Linear search, trust radius update and swapping.
[b]The input structures for molecules 1–33 were taken from Ref. 17.
[c]The details for the first 30 molecules are available in the supplementary material.
[d]The input structure was taken from Ref. 23.

and the equations are solved for $\mathbf{S}$ and λ simultaneously. This task necessitates some minor changes in our original algorithm;[6] therefore, we present a brief summary.

The goal is to solve a system of linear equations

$$\mathbf{y}=(\mathbf{M}+\lambda\mathbf{S})\mathbf{x}=(\mathbf{M}+\mu\mathbf{I})\mathbf{x}=\overline{\mathbf{M}}\mathbf{x}, \tag{21}$$

where $\mathbf{y}$ and $\mathbf{M}$ are known. We assume that solution $\mathbf{x}$ exist and designate the corrected Hessian, $\mathbf{M}+\lambda\mathbf{S}$ by $\overline{\mathbf{M}}$ in the description of the algorithm to simplify the equations. Since the scaling matrix, $\mathbf{S}$, is chosen to be a scalar times the unit matrix, $\lambda\mathbf{S}=\mu\mathbf{I}$ with $\mu=\lambda/\sqrt{N}$ for the size independent RFO and $\mu=1$ otherwise. To solve the RFO/TRM equation, $\mathbf{M}$ is replaced by $\mathbf{H}$, $\mathbf{y}$ by $\mathbf{f}$ and $\mathbf{x}$ by $\mathbf{s}$. The steps of the algorithm are as follows:

(1) Initialize the counter as $k=0$, the solution vector as $\mathbf{x}_0=\mathbf{0}$ and the error in vector $\mathbf{y}$ as $\Delta\mathbf{y}=\mathbf{y}$. Obtain a guess for the inverse, $\overline{\mathbf{M}}_0^{-1}$, or initialize it as the inverse of the diagonal of matrix $\overline{\mathbf{M}}_0$ using a suitable initial value for λ_0.

(2) Form intermediate vectors $\Delta\tilde{\mathbf{x}}=\overline{\mathbf{M}}_k^{-1}\Delta\mathbf{y}$, $\Delta\tilde{\mathbf{y}}=\overline{\mathbf{M}}_k\Delta\tilde{\mathbf{x}}$ and compute the scaling factor $\sigma=\Delta\mathbf{y}^T\Delta\tilde{\mathbf{y}}/\Delta\tilde{\mathbf{y}}^T\Delta\tilde{\mathbf{y}}$. If $|\sigma|$ is smaller than 10^{-3}, replace $\Delta\tilde{\mathbf{x}}$ by $\Delta\mathbf{y}$.

(3) Form the next approximation to the solution vector, $\mathbf{x}_{k+1}=\mathbf{x}_k+\sigma\Delta\tilde{\mathbf{x}}$.

(4) Update the inverse matrix using an SR1 update[13]

$$\overline{\mathbf{M}}_{k+1}^{-1}=\overline{\mathbf{M}}_k^{-1}+\frac{(\Delta\tilde{\mathbf{x}}-\overline{\mathbf{M}}_k^{-1}\Delta\tilde{\mathbf{y}})\cdot(\Delta\tilde{\mathbf{x}}-\overline{\mathbf{M}}_k^{-1}\Delta\tilde{y})^T}{(\Delta\tilde{\mathbf{x}}-\overline{\mathbf{M}}_k^{-1}\Delta\tilde{\mathbf{y}})^T\cdot\Delta\tilde{\mathbf{y}}}. \quad (22)$$

(5) Calculate λ_{k+1} (using the scheme described below) and obtain μ_{k+1} and $\overline{\mathbf{M}}_{k+1}$.

(6) Calculate the new error in $\mathbf{y}$, $\Delta\mathbf{y}=\mathbf{y}-\overline{\mathbf{M}}_{k+1}x_{k+1}$.

(7) Update the counter ($k=k+1$) and go to step (2) until $\Delta\mathbf{y}$ converges to $\mathbf{0}$ within a useful threshold for the norm and/or maximum absolute value.

1. Updating the diagonal correction ($\lambda\mathbf{S}$) to the Hessian

The method of updating λ in step (5) of the algorithm should provide a good balance between stability and fast convergence for most cases. When both RFO and TRM are enabled, the actual method used to control the step size can be determined dynamically. Different updated values for λ^{RFO} and λ^{TRM} are calculated and the larger correction is used in the next iteration.

There are several ways of updating λ^{RFO}. The simplest is to use Eq. (9), e.g., $\lambda_{k+1}^{\mathrm{RFO}}=\mathbf{y}^T\mathbf{x}_{k+1}$; however, this can lead to oscillations resulting from large changes in the solution vector $\mathbf{x}$ near the beginning of the iterative process. When neither $\mathbf{x}$ nor λ^{RFO} are converged Eq. (9) is not satisfied; nevertheless, the current step can be scaled so that both sides of the Eq. (9) give the same scalar product with the current step

$$\mathbf{x}_{k+1}^T(\mathbf{M}+\alpha\mathbf{x}_{k+1}^T\mathbf{y}\mathbf{S})\alpha\mathbf{x}=\mathbf{x}_{k+1}^T\mathbf{y}. \quad (23)$$

Equation (23) can be solved to obtain a new value for λ^{RFO}

$$\lambda_{k+1}^{\mathrm{RFO}}=\alpha\mathbf{x}_{k+1}^T\mathbf{y}. \quad (24)$$

For the SIRFO method, the total diagonal correction to the Hessian is $\lambda^{\mathrm{RFO}}\mathbf{S}=\mu\mathbf{I}=(\lambda^{\mathrm{RFO}}/\sqrt{N})\mathbf{I}$.

The update for λ^{TRM} assumes that a change in λ^{TRM} can be approximated by a simple scaling of the current step. This approach can be used to limit the length of the step or the maximum component of the step. Whichever criterion results in a larger Hessian correction is used for updating λ^{TRM} when both criteria are enabled [i.e., when the original step (RFO or NR) exceeds one of the limits, the TRM correction will provide a step equal to the corresponding limit]. The updating process uses the previous correction, μ_k, resulting in a step $\mathbf{x}_{k+1}$ which has to be scaled by α to fulfill the corresponding limit (maximum step component, rms or total step size). Similar to the updating process for λ^{RFO}, the scalar product of the current step and the right side of Eq. (21) with the previous and the scaled step yields

$$\mathbf{x}_{k+1}^T(\mathbf{H}+\mu_k\mathbf{I})\mathbf{x}_{k+1}=\mathbf{x}_{k+1}^T(\mathbf{H}+\lambda_{k+1}^{\mathrm{TRM}}\mathbf{I})\alpha\mathbf{x}_{k+1}. \quad (25)$$

Solving for $\lambda_{k+1}^{\mathrm{TRM}}$ gives the new TRM correction

$$\lambda_{k+1}^{\mathrm{TRM}}=\frac{(1-\alpha)\mathbf{x}_{k+1}^T\mathbf{H}\mathbf{x}_{k+1}+\mu_k\mathbf{x}_{k+1}^T\mathbf{x}_{k+1}}{\alpha\mathbf{x}_{k+1}^T\mathbf{x}_{k+1}}. \quad (26)$$

If α_{k-1} and α_k are smaller and larger than 1, the TRM correction is updated by a linear interpolation using two previous corrections, μ_{k-1} and μ_k

$$\lambda_{k+1}^{\mathrm{TRM}}=\frac{(1-\alpha_k)\mu_{k-1}+(\alpha_{k-1}-1)\mu_k}{\alpha_{k-1}-\alpha_k}. \quad (27)$$

The TRM correction may cause oscillations when the changes in $\mathbf{x}$ are large; therefore, the largest increase in μ between consecutive cycles should be chosen.

When the lowest eigenvalue of the Hessian is negative, the convergence of λ^{TRM} can be accelerated by assuming that the resulting step has a large component in the direction corresponding to the negative eigenvalue

$$\lambda_{k+1}^{\mathrm{TRM}}=\frac{1}{\alpha_k}(\mu_k+\epsilon_{\mathrm{lowest}})+\epsilon_{\mathrm{lowest}}. \quad (28)$$

A lower limit for the correction is

$$\lambda_{\min}=-\epsilon_{\mathrm{lowest}}+\min\left(\frac{-\epsilon_{\mathrm{lowest}}}{2},\epsilon_{\min}\right), \quad (29)$$

where the $\epsilon_{\min}$ is a suitable limit (e.g., 10^{-4} a.u.) to protect the inverse of the corrected Hessian from approaching singularity. The constraints on λ for transition state optimization are discussed elsewhere.[10(b)]

E. Combined Hessian update method for minimization

For minimizations with simple Newton–Raphson steps, it is necessary to use a Hessian update like BFGS which ensures that the Hessian remains positive definite

$$\mathbf{H}_k=\mathbf{H}_{k-1}+\Delta\mathbf{H}_k^{\mathrm{BFGS}},$$

$$\Delta\mathbf{H}_k^{\mathrm{BFGS}}=-\left(\frac{\Delta\mathbf{f}_k\Delta\mathbf{f}_k^T}{\mathbf{s}_{k-1}^T\Delta\mathbf{f}_k}+\frac{\mathbf{H}_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T\mathbf{H}_{k-1}}{\mathbf{s}_{k-1}^T\mathbf{H}_{k-1}\mathbf{s}_{k-1}}\right),$$

$$\Delta\mathbf{f}_k=\mathbf{f}_k-\mathbf{f}_{k-1}. \quad (30)$$

With step size control methods such as RFO and TRM, an optimization can step toward a minimum even when the Hessian is not positive definite. Bofill developed a succesful update for transition state optimizations

$$\mathbf{H}_k=\mathbf{H}_{k-1}+\varphi^{\mathrm{Bofill}}\Delta\mathbf{H}_k^{\mathrm{SR1}}+(1-\varphi^{\mathrm{Bofill}})\Delta\mathbf{H}_k^{\mathrm{Powell}}, \quad (31)$$

$$\varphi^{\mathrm{Bofill}}=\frac{((\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T\mathbf{s}_{k-1})^2}{(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})\mathbf{s}_{k-1}^T\mathbf{s}_{k-1}}, \quad (32)$$

where the symmetric rank one (SR1) update and the Powell-symmetric-Broyden (PSB) update are given by

$$\Delta\mathbf{H}_k^{\mathrm{SR1}}=-\frac{(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T}{(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T\mathbf{s}_{k-1}}, \quad (33)$$

$$\Delta\mathbf{H}_k^{\mathrm{PSB}}=(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T\mathbf{s}_{k-1}\frac{\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T}{(\mathbf{s}_{k-1}^T\mathbf{s}_{k-1})^2}-\frac{(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})\mathbf{s}_{k-1}^T+\mathbf{s}_{k-1}(\Delta\mathbf{f}_k+\mathbf{H}_{k-1}\mathbf{s}_{k-1})^T}{\mathbf{s}_{k-1}^T\mathbf{s}_{k-1}}. \quad (34)$$

This suggests that a similar combination of SR1 with BFGS could be advantageous for minimizations. However, initial tests showed no noticeable improvement. Closer examination revealed that the weight of the SR1 formula was very small, usually less than 10%. Therefore, we have tested the same combination but with the square root of Bofill's original weighting factor

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \varphi \Delta \mathbf{H}_k^{\mathrm{SR1}} + (1-\varphi)\Delta \mathbf{H}_k^{\mathrm{BFGS}}, \tag{35}$$

$$\varphi = \sqrt{\varphi^{\mathrm{Bofill}}} = \sqrt{\frac{((\Delta \mathbf{f}_k + \mathbf{H}_{k-1}\mathbf{s}_{k-1})^T \mathbf{s}_{k-1})^2}{(\Delta \mathbf{f}_k + \mathbf{H}_{k-1}\mathbf{s}_{k-1})^T (\Delta \mathbf{f}_k + \mathbf{H}_{k-1}\mathbf{s}_{k-1}) \mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}}. \tag{36}$$

This approach is only slightly better for the smaller test cases, but gives noticeably improved results for larger molecules (see Table I). The increased molecular size requires more optimization cycles, which may result in the BFGS update converging to an inaccurate Hessian. This behavior seems to be corrected by our combined update process. Using the square root of the original Bofill weight in the SR1-PSB combination, Eq. (31), also improves transition state optimizations.

F. Test calculations

Our current study contains a number of changes and improvements that need to be validated by comparing them to existing procedures. To test all combinations of the available options would result more than 400 optimizations for each test molecule; therefore, only a few appropriate combinations of options have been selected:

- BFGS or combined SR1-BFGS Hessian update,
- full history update (using all available, up to a maximum of 100, previous points within an acceptable distance) or two vector update (using only the last two points),
- regular or size independent RFO,
- regular step size scaling or TRM on largest component and rms of the step,
- regular or O(N^2) scaling transformation and equation solving methods,
- other features which are implemented in the ''Berny'' optimization algorithm,[19] such as:

—linear search,[11] (fit a quartic polynomial to the latest two points using the energies and forces)

—trust radius update,[11]

—swapping (reorder the stored points, if necessary, so that the linear search uses the lowest energy point).

The optimizations were carried out using the default initial guess[20] Hessian in GAUSSIAN 98. The convergence criteria, optimization parameters, other computational details and the input structures are available in supplementary material.[21]

1. Small molecule tests

A series of 33 molecules employed by Baker[17] form a convenient test set of small molecules that has been used by numerous authors to test geometry optimization methods. For the first 30 molecules, the average of the maximum change in internal coordinates between initial and final structures is ~0.2 bohr or radian,[23] indicating that in general they were started from very good initial geometries. Calculations were carried out at the Hartree–Fock STO-3G level of theory and are summarized in Table I. The differences in performance are very small, less than 5%, indicating that existing minimization methods are already close to optimal for small molecules (further details can be found in the supplementary material). The combined SR1-BFGS update (columns 8–14) is only ~3% better than BFGS (columns 1–7). The full history update option results in a mere 1.5% improvement. The SIRFO/TRM combination gives results nearly identical to the regular RFO for these small tests; however, TRM or step scaling was rarely active. The RFO implementations in the present algorithm is slightly better than the regular RFO, presumably due to the different coordinate systems used in the present method for solving the RFO equations. The other options (trust radius update, linear search, and swapping) show almost no improvement when they are enabled. Provided an appropriate coordinate system is used and the optimization step size is controlled, the tests on these smaller molecules demonstrate that a wide range of options can give almost identical results. By contrast, the medium and larger size molecules discussed below show significant differences for the various options.

2. Medium size molecules

The medium size test set contains buckminsterfullerene C_{60}, taxol, and helical alanine-based oligopeptides built from 5, 10, and 20 amino acids (molecules 34–40). These molecules contain up to 206 atoms, and were optimized at semiempirical AM1 level of theory. The number of steps needed for optimization are listed in Table I, and one can begin to see some significant differences between the various option sets. For the first three molecules, all optimizations converged to the same minimum. The Ala_{10} and Ala_{20} systems, however, are sufficiently flexible and have many local minima; therefore, different options can yield different structures. The performance of different options can be compared directly only if the optimizations reach the same minimum.

A number of factors can cause optimizations to converge to different minima. For the larger oligopeptides, the optimizations encountered negative eigenvalues in the updated Hessian. The RFO method is tends to scale up the step in the direction of the lowest (negative) eigenvalue and can produce very large steps (if allowed by the trust radius), endangering the stability of the optimization. Another potential problem arises if the effective (corrected) Hessian approaches singularity during the equation solving process; this can be controlled with the TRM approach. The trust radius updating varies the trust radius in the interval of 0.05–1.00 a.u. and can also result different optimization path than methods with no trust radius update. The linear search contains a separate step size control; therefore, the resulting total step can be twice as long than the actual trust radius.

For C_{60}, taxol, and Ala_5, the new combined SR1-BFGS Hessian update method shows a noticeable performance improvement compared to BFGS, especially when the full-history Hessian updating is not enabled. Interestingly, the combined SR1-BFGS update shows the worst performance

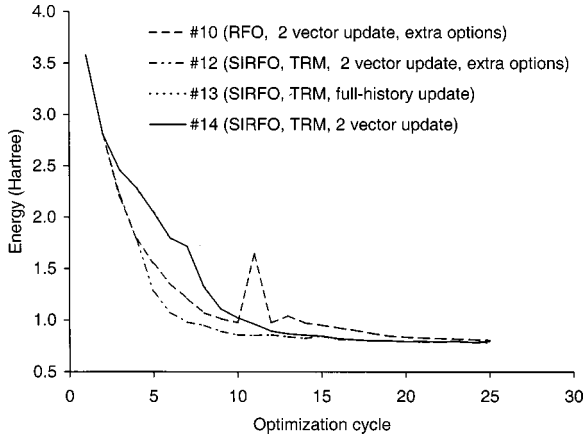

FIG. 1. Energy vs step number for several SR1-BFGS based optimizations on crambin. Cycles 1–25.

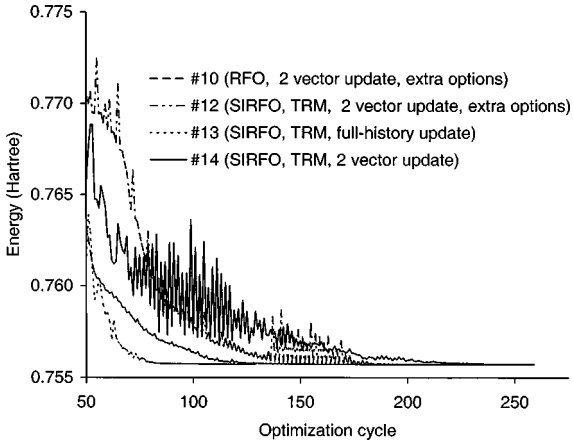

FIG. 2. Energy vs step number for several SR1-BFGS based optimizations on crambin. Cycles from 50 until convergence.

for these medium size molecules when used with the regular optimization scheme (option set 8) but the best performance when used with SIRFO/TRM and full-history Hessian updating. The extra options (linear search, trust radius update, swapping) were tuned to work best for minimization with the BFGS update and show relatively stable performance (options sets 1–5). Surprisingly, BFGS with SIRFO/TRM and full history update works best for these molecules without the extra options (option set 6). The much poorer performance of the two vector update compared to the full history update is unfortunate because the latter scheme cannot be used readily in a limited memory version of our large molecule optimization algorithm.

3. Larger molecules

Crambin and solvated segments of RNA were chosen as our largest test molecules to be optimized at empirical universal force field[22] (UFF) level. The regular optimization algorithm with matrix diagonalizations is very CPU intensive for these systems and thus was not used in the comparisons. The RNA structure contains 8 bases in 4 segments, 4 sodium ions, and 34 water molecules. Due to the relatively weak interaction between the large number of fragments and because the starting structure was far from any minimum, the entire optimization path could not be described with a single set of redundant internal coordinates. The coordinate system was automatically rebuilt 32 times during the course of the optimization [because of the fast coordinate transformation, this did not affect the $O(N^2)$ scaling of the optimization]. Three of the four SR1-BFGS runs resulted in structures with very similar energies, and one of the BFGS runs yielded a lower minimum. However, the ribonucleic acid (RNA) optimizations cannot be used for direct comparisons, since each of the optimizations converged to a different local minimum.

Crambin is a small protein with 642 atoms and can be used to demonstrate the performance of optimization techniques for biologically interesting proteins. The input for crambin was taken form Ref. 23. The structure is not too floppy and the coordinate system needed to be rebuilt only once during the optimization. All the crambin optimizations converged to the same structure; therefore, the performance of the various options can be compared. Table I shows that the full history SR1-BFGS update combined with SIRFO/TRM performs exceptionally well (option set 13). For the two vector update, the RFO based optimization (10) is somewhat better than the SIRFO-TRM option (12 and 14). The SR1-BFGS update is significantly better than BFGS (3 vs 10, 5 vs 12, 6 vs 13, 7 vs 14).

Figures 1–3 show the energy of crambin as a function of the step number for the different optimization procedures. At the beginning (Fig. 1) all methods show a rapid, monotonic decrease in the energy, except RFO (10). The RFO procedure has a very large jump in energy (~0.5 hartree) because of a negative eigenvalue in the updated Hessian, resulting in a bad RFO step along the corresponding eigenvector. The difference in the behavior of the various options is more apparent in the expanded energy scale in Fig. 2. For RFO (10) the energy descends very smoothly, because of the linear search and the larger correction to the Hessian, whereas the SIRFO based optimizations (12–14) start to oscillate. Figure 3 shows the final 50 steps of each optimization. The two vector update SIRFO optimizations (12 and 14) continue to oscillate, with the oscillations dampened somewhat by linear search (12 vs 14). The full history SR1-BFGS update

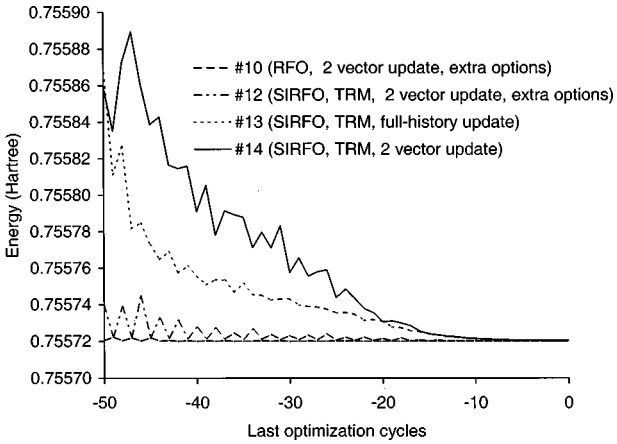

FIG. 3. Energy vs step number for several SR1-BFGS based optimizations on crambin. Last 50 optimization cycles, 0 represents the last cycle.

based scheme (13) converges quickly as the number of vectors used in the update grows. The regular RFO based optimization (10) shows smooth convergence but approaches the minimum very slowly at the end of the optimization (Fig. 3). It appears that toward the end of the optimization, RFO (10) uses too large a correction to the Hessian, slowing convergence, while SIRFO (12) uses too small a correction, causing oscillations. This will be examined in a subsequent paper. Despite the oscillations, both SIRFO based schemes with no extra options (13 and 14) show much faster convergence at the end than the others.

For large molecules, the memory requirements of a full history Hessian update can be a serious limitation. However, the performance penalty for using the two vector update is unacceptably large (e.g., 259 vs 129 steps). Potentially, a limited update using more than two vectors could be better. As a test, crambin was optimized with the SIRFO/TRM/SR1-BFGS combination using a maximum of five vector in the update process. The same minimum was reached and convergence was achieved in 138 steps. This suggests that it may be possible to find a satisfactory compromise between performance and cost. However, considerably more work will be needed to determine the parameters that will be appropriate for a wide range of molecules.

4. CPU timing and scaling

Figure 4 clearly demonstrates that our optimization algorithm has $O(N^2)$ scaling with the size of the system (number of atoms). The break-down of the CPU usage is given in Table II. For semiempirical calculations, the present optimization method uses less than 10% of the total CPU time, compared to 50%–80% for the regular optimization. For the larger systems, the full history Hessian update consumes more than half of the time in the optimization step, but the two vector update takes less than 10%. Solution of the RFO/TRM equations accounts for less than one-third of the CPU time in the optimization step.

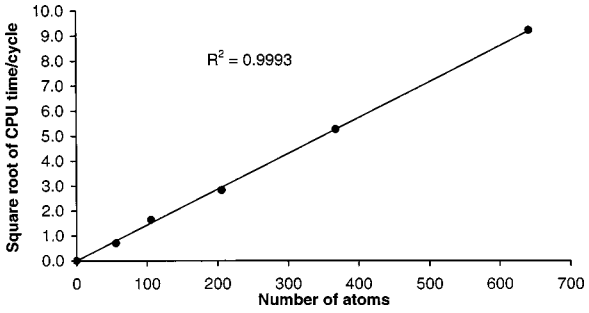

FIG. 4. The square root of the average of the CPU time spent for the optimization algorithm (option set 14 in Table I) in each cycle vs the number of atoms. The data were obtained on an Intel Celeron A/412 MHz processor.

II. CONCLUSIONS

The geometry optimization methods for quantum chemistry have been studied extensively for more than 20 years, ever since gradient calculations became practical. With recent advances in hardware and software, much larger molecules can now be studied by quantum-chemical methods. Optimization techniques need to keep pace with these improvements. The $O(N^2)$ scaling coordinate transformation and RFO/TRM equation solving methods represent significant progress in improving the size dependence of the efforts required to calculate the optimization step. Our combined SR1-BFGS method with full history update and the SIRFO/TRM procedures are very efficient, substantially reducing the number of optimization steps required for larger molecules. These methods were recently used with $O(N)$ scaling PM3 semiempirical method to optimize kringle one of plasminogen[24]—1226 atoms in 362 steps. For even larger systems, the CPU time and memory requirements must be decreased even further. Reduced memory updates and other geometry optimization methods such as GDIIS[25,26] (geometry optimization using direct inversion in the iterative subspace) will be considered for future improvements in optimization methods for large molecules.

TABLE II. Percentage of the total CPU time taken by the optimization step, updating Hessian and solving the RFO/TRM equations. Most details can be found in the supplementary material.

Molecule	For–Ala_5–NH_2	For–Ala_{10}–NH_2	For–Ala_{20}–NH_2	RNA+water	crambin
	(#36)	(#38)	(#39)	(#41)	(#42)
Energy/gradient	AM1	AM1	AM1	UFF (Ref. 22)	UFF
Number of atoms	56	106	206	368	642
Regular optimization (#1)					
Optimization (%)	51.48%	71.58%	79.90%	···	···
Number of cycles	30	90	99	···	···
Combined SR1-BFGS with full-history update and fast equation solving (option set #13)					
Optimization (%)	9.28%	9.78%	7.17%	97.31%	94.02%
Hessian update (%)	4.32%	5.12%	4.34%	72.41%	66.84%
RFO/TRM (%)	0.84%	0.91%	0.66%	7.10%	10.66%
Number of cycles	32	45	59	603	129
Combined SR1-BFGS with two vector update and fast equation solving (option set #14)					
Optimization (%)	5.47%	6.16%	2.76%	80.19%	84.44%
Hessian update (%)	0.26%	0.20%	0.13%	5.00%	5.87%
RFO/TRM (%)	1.57%	1.28%	0.78%	24.08%	25.60%
Number of cycles	31	51	89	756	259

ACKNOWLEDGMENTS

This work has been supported by Gaussian, Inc., the Hungarian Research Foundation (OTKA D-29446), Compaq Hungary, Ltd., and the National Science Foundation (CHE9874005). Ö. F. thanks the Hungarian Academy of Sciences for a János Bolyai Research Scholarship.

[1] (a) H. B. Schlegel, in *Modern Electronic Structure Theory*, edited by D. R. Yarkony (World Scientific, Singapore, 1995); (b) H. B. Schlegel, in *Encyclopedia of Computational Chemistry* (Wiley, New York, 1998).
[2] C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, J. Comput. Chem. **17**, 49 (1996).
[3] (a) P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, J. Am. Chem. Soc. **101**, 2550 (1979); (b) G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay, *ibid.* **114**, 8191 (1992); (c) P. Pulay and G. Fogarasi, J. Chem. Phys. **96**, 2856 (1992).
[4] P. Pulay, Adv. Chem. Phys. **69**, 241 (1987).
[5] P. Pulay, Mol. Phys. **17**, 197 (1969).
[6] Ö. Farkas and H. B. Schlegel, J. Chem. Phys. **109**, 7100 (1998).
[7] 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.
[8] J. Baker, A. Kessi, and B. Delley, J. Chem. Phys. **105**, 11100 (1996).
[9] (a) B. Paizs, G. Fogarasi, and P. Pulay, J. Chem. Phys. **109**, 6571 (1998); (b) J. Baker, Don Kinghorn, and P. Pulay, *ibid.* **110**, 4986 (1999).
[10] (a) A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89**, 52 (1985); (b) J. Simons and J. Nichols, Int. J. Quantum Chem., Quantum Chem. Symp. **24**, 263 (1990).
[11] (a) R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1981); (b) W. Murray and M. H. Wright, *Practical Optimization* (Academic, New York, 1981); (c) *Non-linear Optimization, 1981*, edited by M. J. D. Powell (Academic, New York, 1982); (d) J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Non-Linear Equations* (Prentice Hall, New Jersey, 1983); (e) L. E. Scales, *Introduction to Non-linear Optimization* (Macmillam, Basingstoke, 1985).
[12] (a) C. G. Broyden, J. Inst. Math. Appl. **6**, 76 (1970); (b) R. Fletcher, Comput. J. (Switzerland) **13**, 317 (1970); (c) D. Goldfarb, Math. Comput. **24**, 23 (1970); (d) D. F. Shanno, *ibid.* **24**, 647 (1970).
[13] B. Murtagh and R. W. H. Sargent, Comput. J. (Switzerland) **13**, 185 (1972).
[14] (a) M. J. D. Powell, in *Nonlinear Programing*, edited by J. B. Rosen, O. L. Mangasarian, and K. Ritter (Academic, New York, 1970); (b) M. J. D. Powell, Math. Program. **1**, 26 (1971).
[15] J. M. Bofill, J. Comput. Chem. **15**, 1 (1994).
[16] J. Baker, J. Comput. Chem. **7**, 385 (1986).
[17] J. Baker, J. Comput. Chem. **14**, 1085 (1993).
[18] E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations* (McGraw-Hill, New York, 1955).
[19] H. B. Schlegel, J. Comput. Chem. **3**, 214 (1982).
[20] H. B. Schlegel, Theor. Chim. Acta **66**, 333 (1984).
[21] Supplementary material available from http://chem.wayne.edu/schlegel/supp_mat or http://organ.elte.hu/farkas/suppl_mat
[22] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard, III, and W. M. Skiff, J. Am. Chem. Soc. **114**, 10024 (1992).
[23] C. van Alsenoy, C.-H. Yu, A. Peetrs, J. M. L. Martin, and L. Schäfer, J. Phys. Chem. **102**, 2246 (1998).
[24] A. D. Daniels, G. E. Scuseria, Ö. Farkas, and H. B. Schlegel, Int. J. Quantum Chem. (in press).
[25] (a) P. Pulay, Chem. Phys. Lett. **73**, 393 (1980); (b) P. Pulay, J. Comput. Chem. **3**, 556 (1982); (c) P. Császár and P. Pulay, J. Mol. Struct.: THEOCHEM **114**, 31 (1984).
[26] (a) Ö. Farkas, PhD (CSc) thesis, Eötvös Loránd University and Hungarian Academy of Sciences, Budapest, 1995 (in Hungarian); (b) Ö. Farkas and H. Bernhard Schlegel, J. Chem. Phys. (to be published).