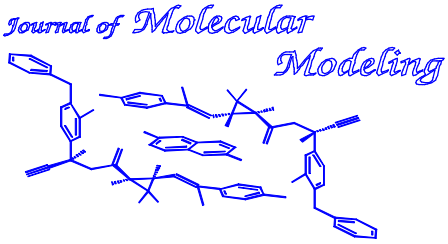
J. Mol. Model. **1995**, *1*, 11 – 17



A Conjugate Gradient Algorithm with a Trust Region for Molecular Geometry Optimization.

Josep Maria Bofill

Departament de Química Orgànica, Universitat de Barcelona, Martí i Franquès 1, E-08028 Barcelona, Catalunya, Spain (jmbofill@canigo.qui.ub.es)

Received: 25 October 1994 / Accepted: 13 December 1994

Abstract

An algorithm is presented for the optimization of molecular geometries and general nonquadratic functions using the nonlinear conjugate gradient method with a restricted step and a restart procedure. The algorithm only requires the evaluation of the energy function and its gradient and less memory storage is needed than for other conjugate gradient algorithms. Some numerical results are also presented and the efficiency and behaviour of the algorithm is compared with the standard conjugate gradient method. On the other hand we present comparisons of both conjugate gradient and variable metric methods with and without the trust region technique. One of the main conclusions of the present work is that a trust region always improves the convergence of an optimitzation method. A sketch of the algorithm is also given.

Keywords: Conjugate gradient, restricted step algorithm, optimization for nonquadratic functions

Introduction

Functions of many variables that are continuously differentiable at least twice represent the central conceptual qualitative as well as quantitative entities of many physical and chemical theories. Generally, the most important points of the functions proposed by theories in chemical physics are their stationary points (e.g., minima, maxima and saddle points). Due to this fact theoreticians need methods for locating stationary points. Using the basic idea of conjugate gradients [1], efficient algorithms for the minimization of quadratic and nonquadratic functions have been proposed. First, we present the theoretical basis and the standard implementation of these algorithms.

Let us consider an arbitrary real-valued continuous function f(x) on an open set S possessing at least continuous first- and second-order partial derivatives on S. At each point x on S we have the Taylor expansion formula

$$f(x+p) = q(x+p) + \frac{1}{2} p^* R(x+p) p \qquad (1)$$

where q(x+p) is the quadratic function approximation

$$q(x+p) = f(x) + p^* g(x) + \frac{1}{2} p^* H(x) p \qquad (2)$$

of f(x+p) on a neighborhood of x. The vector g(x) and the matrix H(x) are the gradient and Hessian of f at x respectively. The remaining term has the feature

$$\lim_{p \to 0} \frac{p^* R(x+p) p}{p^* p} = 0 \qquad (3)$$

and

$$R(x+p) = \int_0^1 2(1-t)\left[H(x+tp) - H(x)\right] dt \qquad (4)$$

The property of equation (3) is the basis of the Newton and conjugate gradient methods since in the neighborhood of a non degenerate stationary point a function behaves like a quadratic function. In others words, the applicability of these methods is based on the principle that an effective algorithm for optimizing a quadratic function q(x+p) can be modified so as to obtain an effective algorithm for optimizing a nonquadratic function f(x+p) [1].

If we are interested in locating a stationary point of the minimum class, starting at an appropriate initial point x the general standard method proceeds as follows [1, 2]:

1. Construct the approximate quadratic function q(x+p) of f(x+p).
2. Use the conjugate gradient technique to locate the minimum point of q(x+p). This can be done in the following way:
   a) Minimize the quadratic function q(x+p) along the line $x_{k+1} = x_k + \alpha_k p_k$ starting with $p_1 = 0$ and $x_1 = x$.
   b) Compute the next direction p_{k+1} using the formulae

$$p_{k+1} = -q'(x_{k+1}) + b_k p_k \tag{5}$$

and

$$b_k = \frac{(q'(x_{k+1}))^* q'(x_{k+1})}{(q'(x_k))^* q'(x_k)} \tag{6}$$

   where q'(x) is the gradient of q at the point x.
   Stop at the k step if $q'(x_{k+1}) = 0$.
3. Replace x by x_{k+1} and stop if $g(x_{k+1})^*g(x_{k+1})$ is small and x_{k+1} is an approximation to the minimum of f, otherwise repeat steps (1) and (2).

The algorithm outlined above was reported by first time by Fletcher and Reeves [3]. Basically Fletcher's algorithm differs from the traditional conjugate gradient method applied to quadratic functions [4, 5] in the explicit minimization of the function along the conjugate descent direction. For nonquadratic functions with many variables this procedure is quite expensive because it normally requires a cubic interpolation in the line search. The origin of the formulas (5) and (6) can be seen as the natural way to build a set of linear independent vectors, say {pi}, from the set of the gradient vectors $\{q'(x_k)\}$, such that they are H(x) conjugate, e.g. $pk^*H(x)p_{k+1} = 0$ [1]. These formulae are exact in the strictly quadratic case.

Sinclair and Fletcher [6] showed that the conjugate gradient technique is still quite efficient in the localization of first order saddle points of nonquadratic functions. In this case one needs the explicit definition of the direction along which the function is maximized and the rest of conjugate basis is built recursively according to Beale's formula [7]. Recently this method was extended by Fischer and Karplus [8], who introduced the so called conjugate peak refinement.

Others methods for the optimization of nonquadratic functions include the variable metric algorithms [2]. This type of method can be seen as an extension of the preconditioned conjugate gradients technique [1, 9]. From the computational point of view the variable metric algorithms are much more efficient than the conjugate gradient one for both quadratic and nonquadratic functions. One of the problems with this class of algorithms is the computer memory storage that it is needed when the function depends of many variables. An attempt to reduce the computer memory was given by Buckley [10] making a combination of conjugate gradient and variable metric algorithms.

In this paper we propose an improved conjugate gradient technique using both the trust region technique [2], and Powell's restart procedure [1, 14]. We will compare the numerical results obtained using the proposed modifications with the standard conjugate gradient algorithm and variable metric methods.

Methodological outline

The modified conjugate gradient method

Writing equation (1) in the following way

$$f(x+p) = f(x) + p^* g(x) + \frac{1}{2} p^* [H(x) + R(x+p)]p \tag{7}$$

we observe that its structure is very close to the quadratic function defined in equation (2), i.e., f(x+p) Å q(x+p) if x+p is near to x. This observation suggests a change in the preceding algorithm in order to make it more efficient [1]. First, we define a vector, say v_k, as

$$v_k = [H(x_k) + R(x_k + p_k)]p_k \tag{8}$$

Taking into account the result of equation (3), the numerical computation of equation (8) is given in the following way

$$v_k = \frac{g\left\{x_k + \varepsilon_k \dfrac{p_k}{(p_k^* p_k)^{1/2}}\right\} - g(x_k)}{\dfrac{\varepsilon_k}{(p_k^* p_k)^{1/2}}} \tag{9}$$

where ε_k is computed according the formula

$$\varepsilon_k = 2\sqrt{\delta}\left[1 + (x_k^* x_k)^{1/2}\right] \tag{10}$$

and where δ is the machine precision [11]. Near to convergence, equation (9) presents numerical problems and should be substituted by the central difference formula, e.g.

(11)

$$\nu_k = \frac{g\left(x_k + \varepsilon_k \dfrac{p_k}{\left(p_k^* p_k\right)^{1/2}}\right) - g\left(x_k - \varepsilon_k \dfrac{p_k}{\left(p_k^* p_k\right)^{1/2}}\right)}{2\dfrac{\varepsilon_k}{\left(p_k^* p_k\right)^{1/2}}}$$

Second, rather than using the gradient of q, q'(x_{k+1}), we force at the end of line search that q'(x_{k+1}) = g(x_{k+1}). This implies both that the gradient is not orthogonal to the previous direction p_k and non-orthogonality between g(x_{k+1}) and g(x_k). In order to minimise these deficiencies locally for nonquadratic functions, we use the trust region technique [2], that has been applied successfully in methods based on the variable metric method [12,13] and the restart procedure recommended by Powell [14]. Briefly the trust region technique consists in modifying the step length in order to maintain the ratio

$$\text{ratio} = \frac{f\left(x_k + \alpha_k p_k\right) - f\left(x_k\right)}{q\left(x_k + \alpha_k p_k\right) - f\left(x_k\right)} \tag{12}$$

close to one at the stationary point of q(x_k+ $\alpha_k p_k$). The way to do this consists of modifying of the current diagonal element of [H(x_k) + R(x_k+p_k)] in the representation of the present conjugate direction. This is the same as that in the minimization of the next restricted quadratic function q^k_r

(13)

$$\begin{aligned} q_r^k\left(x_k + \alpha_k p_k, \nu_k\right) &= f\left(x_k\right) + \alpha_k p_k^* g\left(x_k\right) \\ &\quad + \frac{1}{2}\alpha_k^2 p_k^*\left[H\left(x_k\right) + R\left(x_k + p_k\right) + \nu_k I\right]p_k \\ &\quad - \frac{1}{2}\nu_k R_k^2 \\ &= q\left(x_k + \alpha_k p_k\right) + \frac{1}{2}\nu_k\left(\alpha_k^2 p_k^* p_k - R_k^2\right) \end{aligned}$$

In this expression R_k is the so called trust radius and ν_k is the Lagrangian multiplier. We see that q^k_r is only a Lagrangian function of α_k and ν_k. The optimum value of α_k is

$$\alpha_k^{\text{opt}} = a_k = R_k / \left(p_k^* p_k\right)^{1/2} \tag{14}$$

and ν_k

$$\nu_k = -\left(\frac{p_k^* g\left(x_k\right)}{R_k\left(p_k^* p_k\right)^{1/2}} + \frac{p_k^* \nu_k}{p_k^* p_k}\right) \tag{15}$$

We recall that when the restricted step is not used the optimum value of α_k is [1]

$$\alpha_k^{\text{opt}} = a_k = -\frac{p_k^* g\left(x_k\right)}{\nu_k^* p_k} \tag{16}$$

Let us assume that x_{k+1} is a point of q such that the ratio defined by equation (12) at this point is close to one, then we can write

$$q\left(x_k + a_k p_k\right) \approx f\left(x_k + a_k p_k\right) \tag{17}$$

In this situation we have two cases:

Case a) If $x_{k+1} = x_k + \alpha_k p_k$ is a stationary point of q, then

$$0 = p_k^* q'\left(x_k + a_k p_k\right) \approx p_k^* g\left(x_k + a_k p_k\right) \tag{18}$$

so g(x_{k+1}) and q'(x_{k+1}) are orthogonal to p_k. Imposing the conjugate condition between p_k and p_{k+1} it follows that

$$p_k^*\left[H\left(x_k\right) + R\left(x_k + p_k\right)\right]p_{k+1} = 0 \tag{19}$$

Substituting equations (5) and (8) into equation (19) and replacing q'(x_{k+1}) by g(x_{k+1}) we get

$$-\nu_k^* g\left(x_{x+1}\right) + b_k \nu_k^* p_k = 0 \tag{20}$$

and the formula for the scalar b_k is now

$$b_k = \frac{\nu_k^* g\left(x_{k+1}\right)}{\nu_k^* p_k} \tag{21}$$

Case b) x_{k+1} is not a stationary point of q. In this situation the restricted quadratic model equation (13) has been used, also p_k*g(x_{k+1}) ≠ 0 and g(x_{k+1})*g(x_k) ≠ 0. We redefine the vector ν_k to be the vector

$$\nu_k = \frac{g\left(x_k + a_k p_k\right) - g\left(x_k\right)}{a_k} \tag{22}$$

where ak is evaluated according to equation (14). The formula for the scalar b_k is now

(23)

$$b_k = \frac{g(x_k + a_k p_k)^* g(x_k + a_k p_k) - g(x_k + a_k p_k)^* g(x_k)}{a_k v_k^* p_k}$$

This is the Polak and Ribiere equation [15]. We observe that equation (23) is reduced to equation (21) when $g(x_{k+1})^*g(x_k) = 0$.

With the latter considerations the algorithm now is:

1. Select x_1, $f(x_1)$, $g(x_1)$, $p_1 = -g(x_1)$ and R_1. Set k = 1 and j = 1.
2. Compute v_k using either equations (9) or (11) and the scalar a_k by the equation (16).
3. If $a_k(p_k*p_k)^{1/2} > R_k$ then compute a_k according to equation (14).
4. Compute $f(x_k + a_k p_k)$ and the ratio using equation (12).
If ratio < r_l, then $R_{k+1} = R_k / Sf$.
If ratio > r_u and $a_k(p_k*p_k)1/2 = R_k$, then $R_{k+1} = R_k$ Sf, where Sf is a scaling factor.
Otherwise $R_{k+1} = R_k$.
If ratio ≤ 0, then using the new R_{k+1} set k = k + 1, $x_{k+1} = x_k$ and go to 3
else, update $x_{k+1} = x_k + a_k p_k$ and compute $g(x_{k+1})$.
5. Check the convergence criteria:
$((g(x_{k+1})*g(x_{k+1}))/n)^{1/2} \le \varepsilon_1$ and $|f(x_{k+1}) - f(x_k)| \le \varepsilon_2$, where n is the number of variables. If it is satisfied stop and x_{k+1} is an approximation to the minimum of f.
6. Compute the new direction p_{k+1}. For k = 1 compute p_2 using equation (5).
If k ≥ 2 then test the inequality

$$\frac{\left|g(x_k)^* g(x_{k+1})\right|}{g(x_k)^* g(x_k)} \ge \sigma \qquad (24)$$

if it holds or k – j ≥ n set j = k – 1.
If k > j + 1 then compute p_{k+1} using Beale's formula [7]

$$p_{k+1} = -g(x_{k+1}) + c_k^j p_j + b_k p_k \qquad (25)$$

The scalar b_k is evaluated using equation (23). On the other hand, the scalar c_k^j is

$$c_k^j = \frac{\left(g(x_{j+1}) - g(x_j)\right)^* g(x_{k+1})}{\left(g(x_{j+1}) - g(x_j)\right)^* p_j} \qquad (26)$$

Test that the direction p_{k+1} is sufficiently downhill by using the inequalities

$$\rho_l \le -\frac{p_{k+1}^* g(x_{k+1})}{g(x_{k+1})^* g(x_{k+1})} \le \rho_u \qquad (27)$$

if they are not satisfied set j = k –1 and redefine p_{k+1} using equation (5).
Else evaluate pk+1 using expression (5).
7. If k = 1 or j = k – 1, store $p_j = p_k$, $g_j = g(x_k)$ and $g_{j+1} = g(x_{k+1})$.
Set k = k + 1 and go to 2.

We remember that using equation (5) the scalar bk is evaluated by the expression (23). All the parameters are arbitrary and the algorithm is quite insensitive to their change. Suggested values are $r_l = 0.25$, $r_u = 0.75$, Sf = 2.0, s = 0.2, $r_l = 0.8$, $r_u = 1.2$ and $R_1 = 0.1$. Step 7 is the restart procedure proposed by Powell [14]. Finally, it should be noted that the memory storage required by this algorithm is six vectors.

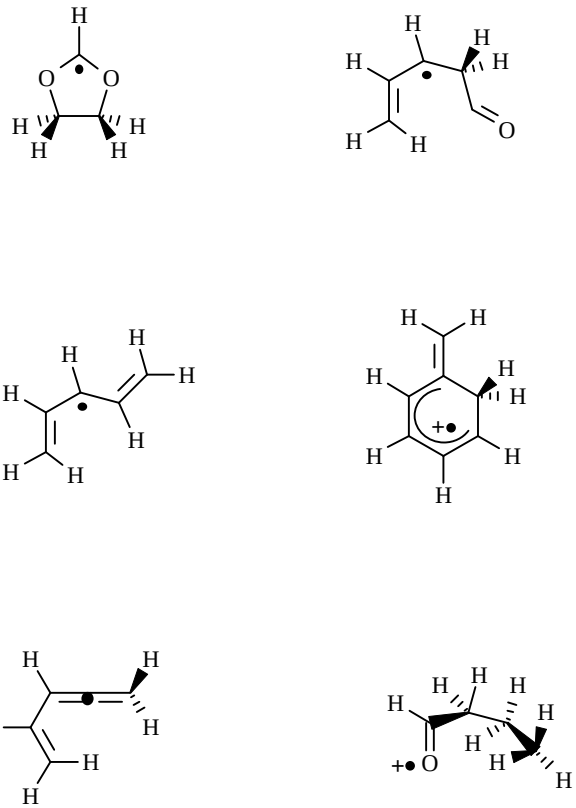


Fig. 1 *Optimized structures contained in this study*

Some numerical results and conclusions.

In order to test the behaviour and performance of the method, the above algorithm has been implemented in the semiempirical program package MOPAC [16]. All molecular geometry optimizations were carried out with the AM1 semiempirical self consistent field (SCF) Hamiltonian [17]. The appropriate wave function, (e.g., RHF or UHF) was taken in each case. The convergence criteria were $\varepsilon_1 = 3x10^{-2}$ kcal mol^{-1} Å^{-1} and $\varepsilon_2 = 5x10^{-5}$ kcal mol^{-1}, which are the units used in MOPAC, and the maximum number of iterations allowed was 100. The results are presented in Table I and the corresponding optimized structures in Figure 1. In the first example the application of the conjugate gradient algorithm whitout restricted step led to a molecular geometry in which the SCF method does not converge. This feature never can occur when the trust region technique is used because the geometrical changes are small from iteration to iteration. The last example presented in the Table I, the non convergence of the standard conjugate gradient algorithm is due to maximum number of iterations. In the column three, last row, we give the value of function difference, $f(x_{100}) - f(x_{101}) = 7x10^{-4}$ kcal/mol, reached after 100 iterations for the example VI. The convergence is achieved at the 140th iteration. This large number of iterations is due to the flatness of the hypersurface near the minimum. When this occurs, the denominator of equation (16) is small because it represents the curvature of the hypersurface and, consequently, α_k^{opt} is big, producing important changes in the molecular geometry even near convergence. In the present case the denominator of equation (16) was around 20 in almost all the iterations, in normal cases the denominator takes values larger than 200. Using the restricted step technique, the effect of the flat hypersurface is corrected at each iteration [2, 13], since the a_k^{opt} does not depend of the curvature (see equation (14)). On the other hand, in the proposed algorithm, the conjugate direction is tested at each iteration in order to check that it is downhill enough. The two important points just mentioned are lacking in the standard conjugate gradient and explain the large difference in the number of iterations encountered in example VI. In examples II and III, both the standard conjugate gradient and the new proposed conjugate gradient algorithm are equally efficient.

Comparing the number of function evaluations (SCF energy calculations) required for both types of methods, e.g. variable metric and conjugate gradient methods, we note that this number is always much smaller when the trust region is used. In the case of variable metric method, the large difference is because no trust region is used and an exact line search minimization is employed. This fact shows numerically the lower efficiency of exact line search as pointed out by Fletcher [2].

Table II shows the behaviour of the conjugate gradient method proposed in this paper for example I. First of all, we see that the restricted step only applies to the first iterations. This fact avoids big changes in the geometry and helps the

Table 1. *Comparison between conjugate gradient and variable metric methods with and without the trust region technique.*

		C.G. [a]		**C.G.T.R. [b]**		**V.M. [c]**		**V.M.T.R. [d]**	
str. [e]	**n.v. [f]**	**n.i. [g]**	**n.f. [h]**	**n.i. [g]**	**n.f. [h]**	**n.i. [g]**	**n.f. [h]**	**n.i. [g]**	**n.f. [h]**
1	8	-	-	20	64	13	61	16	24
2	20	19	57	19	59	15	66	12	32
3	21	12	36	13	40	14	62	8	31
4	26	39	117	24	74	32	139	8	34
5	26	47	141	34	104	15	71	9	35
6	33	7x10E-4	300	79	241	49	228	39	74

a) Conjugate gradient.
b) Conjugate gradient with a trust region and Powell's restart procedure. (See ref. 14)
c) Variable metric method with exact line search.
d) Variable metric method with a trust region. (See ref. 12)
e) Structures, see Figure 1.
f) Number of variables.
g) Number of iterations.
h) Number of SCF energy evaluations.

algorithm to obtain the correct convergence. Second, the column four gives the values of inequality (24). According to the proposed algorithm in the previous section, when the value of the inequality is lower than 0.2 then Powell's restart [14] procedure is applied. This occurs at iterations 7, 10, 12 and from iteration 14 to the end of the process. In column five we present the values of inequality (27). When this inequality is between 1.2 and 0.8, we consider that the conjugate direction obtained from Beale's formula [7] [eq. (25)] is downhill enough. Normally, we see that this occurs in the quadratic region, e.g. in the last iterations of the process.

The important effect of the use of Beale's formula [7] in the last steps of the convergence process can be seen in example VI. The present method needs 79 iterations to reach convergence, whereas using the standard conjugate gradient more than one hundred iterations are needed to reach the convergence.

From the presented results, it can be concluded that in both methods, conjugate gradient and variable metric, the trust region technique improves the efficiency of the algorithms. Furthermore, the restart procedure used in the conjugate gradient augment the efficiency in the last steps of the process and situations where the hypersurface is flat near the minimum.

Recently an optimization technique the so called truncated Newton minimizer [18] has been described, that can be seen as an intermediate between conjugate gradient and variable metric methods. Since this new technique combines the best of both methods and variable metric method is always superior to conjugate gradient, we conclude that for

Table 2. *Behaviour of the conjugate gradient method with a trust region and restart proceudre for the example* ***1*** *of Table 1.*

Iter. k [a]	$\|\|g(x_k)\|\|$ [b]	ΔE [c]	Ineq.(24) [d]	Ineq.(27) [e]	r.s. [f]
1	230.40	-30.36106	-	-	on
2	179.54	-3.27264	0.676	-	on
3	216.03	-0.13212	1.082	-	off
4	199.23	-3.46169	1.551	-	on
5	126.27	-0.17191	1.123	-	off
6	111.29	-2.88722	0.472	-	off
7	62.86	-3.32058	0.068	0.95	off
8	70.99	-3.24341	1.739	-	on
9	35.39	-0.11825	1.328	-	off
10	18.19	-1.65821	0.027	0.91	off
11	21.90	-0.94633	1.743	-	off
12	5.41	-0.00866	0.039	0.95	off
13	5.31	-0.02707	0.382	-	off
14	2.20	-0.01235	0.004	1.00	off
15	2.50	-0.00138	0.003	0.99	off
16	1.41	-0.00451	0.012	0.98	off
17	0.85	-0.00263	0.062	1.03	off
18	0.39	-0.00003	0.126	0.95	off
19	0.22	-0.00015	0.382	-	off
20	0.05	-0.00001	0.003	1.00	off

a) Iteration number.
b) Gradient norm $(g(x_k)^* g(x_k)/n)^{1/2}$ *in kcal mol*$^{-1}$ *Å*$^{-1}$ *, n is the number of variables*
c) Energy difference of $f(x_{k+1}) - f(x_k)$ *in kcal mol*$^{-1}$.
d) Inequality of equation (24).
e) Inequality of equation (27).
f) Restricted step.

large problems, say thousands of variables, the truncated Newton minimizer should be superior to the technique presented in this paper.

I am indebted to Prof. S. Olivella for his suggestions. The research was supported by the Spanish DGICYT (Grant PB92-0796-C02-01).

References.

1. Hestenes, M.R. *Conjugate Direction Methods in Optimization.* Springer-Verlag New York 1980.
2. Fletcher, R. *Practical Methods of Optimization.* Wiley, New York, 1987.
3. Fletcher, R.; Reeves, C.M. *Comput. J.* **1959,** *7,* 149.
4. Hestenes, M.R.; Stiefel, E. *J. Res. Natl. Bur. Stand.* **1952**, *49***,** 409.
5. Lanczos, C. *J. Res. Natl. Bur. Stand.* **1952**, *49***,** 33.
6. Sinclair, J.E.; Fletcher, R. *J. Phys. C.* **1974**, *7***,** 864.
7. Beale, E.M.L. in: *Numerical Methods for Non-Linear Optimization,* F.A. Lootsma (ed) Academic Press, New York, 1972.
8. Fischer, S.; Karplus, M. *Chem. Phys. Letters* **1992,** *194***,** 252.
9. Huang, H.Y. *J. Opt. Th. Applics.* **1970**, *5***,** 405.
10. Buckley, A.G. *Math. Prog.* **1978**, *15***,** 200.
11. Schlick, T.; Fogelson, A. *ACM Trans. Math. Softw.* **1992**, *18*, 46.
12. Culot, P.; Dive, G.; Nguyen, V.H.; Ghuysen, J.M. *Theor. Chim. Acta* **1992**, *82***,** 189.
13. Bofill, J.M. *J. Comp. Chem.* **1994**, *15***,** 1.
14. Powell, M.J.D. *Math. Prog.* **1979**, *12***,** 241.
15. Polak, E.; Ribiere, G. *Rev. Fr. Inr. Rech. Oper.* **1969**, *16***,** 35.
16. MOPAC program. Local version. S. Olivella, *QCPE Bull.* **1984**, ***4,*** 109.
17. Dewar, M.J.S.; Zoebisch, E.G.; Healy, E.F.; Stewart, J.J.P. *J. Am. Chem. Soc.* **1985**, *107***,** 3902.
18. Derreumaux, P.; Zhang, G.; Schlick, T.; Brooks, B. *J. Comp. Chem.* **1994**, *15***,** 532.