An Algorithm for Geometry Optimization Without Analytical Gradients

Jon Baker*

Research School of Chemistry, Australian National University, Canberra, ACT 2601 Australia

Received 27 November 1985; accepted 11 April, 1986

A numerical algorithm for locating both minima and transition states designed for use in the *ab initio* program package GAUSSIAN 82 is presented. It is based on the RFO method of Simons and coworkers and is effectively the numerical version of an analytical algorithm (OPT = EF) previously published in this journal. The algorithm is designed to make maximum use of external second derivative information obtained from prior optimizations at lower levels of theory. It can be used with any wave function for which an energy can be calculated and is about two to three times faster than the default DFP algorithm (OPT = FP) supplied with GAUSSIAN 82.

INTRODUCTION

Geometry optimization to determine the location of minima and transition states on potential energy surfaces has now become standard practice. This is due in no small part to the availability, in recent years, of efficient computational algorithms to obtain analytical gradients.[1-4] Several *ab initio* program packages now incorporate analytical gradients for a wide range of wavefunctions, e.g., RHF, UHF, MP2, CID, and CISD in GAUSSIAN 82[5]; RHF, UHF, and ROHF (restricted open shell SCF) in CADPAC[6] and even CASSCF gradients in GAMESS.[7] However, there are still several types of wavefunction e.g., MP3, MP4, for which analytical gradients are *not* currently available, and, even if they were to become so, are likely to be very expensive to calculate. For these and other types of wavefunction, nonderivative methods for geometry optimization, based on function (energy) evaluations only, must be used.

It is generally accepted that the best nonderivative optimization methods are those based on approximating the gradient by finite differences, and then proceeding in essentially the same way as with an analytical gradient optimization with additional modifications to take into account the fact that the gradient is not "exact". Perhaps the most commonly used algorithm of this type, when searching for minima, is a numerical version of the Davidon-Fletcher-Powell method,[8] first presented over 20 years ago. The nonderivative optimization algorithm in GAUSSIAN 82 (OPT = FP) is a modified version[9] of this.

The standard approach for investigating the structure and energetics of molecules and ions in the gas phase is to locate the relevant stationary points on the potential energy surface at a modest level of theory, e.g., HF/3-21G or HF/6-31G*, determine their exact characteristics — whether minima, transition states or higher-order saddle points — by vibrational analysis, and then perform single-point calculations at higher theoretical levels, e.g., MP3/6-31G** on the optimized structures to give "more accurate" estimates of barrier heights and dissociation energies, the assumption being that the geometries of the stationary points found at the low level are more-or-less what they would be at the higher level. In many cases this assumption is perfectly valid, and a large body of evidence exists[10] suggesting that, for "standard" molecules in their ground state, geometries obtained at the SCF level are similar to those obtained at correlated levels and, indeed, comparable to geometries deter-

*Present Address: Department of Theoretical Chemistry, University Chemical Laboratory, Lensfield Road, Cambridge CB2 1EW, UK.

Journal of Computational Chemistry, Vol. 8, No. 5, 563–574 (1987)

CCC 0192-8651/87/050563-12$04.00

mined experimentally. However, there is always the possibility that reoptimization at the higher level could significantly alter estimated barrier heights, particularly when the barrier is small, and in certain cases inclusion of correlation may change the geometry quite markedly.[11,12]

In general though, geometries should not change too much at higher levels of theory, and if a post-SCF optimization is contemplated in order to improve the energetics it is obviously sensible to start the optimization off from the SCF geometry. In fact, it is usually advisable to utilize as much information as possible from the SCF optimization as input to the post-SCF optimization and, in particular, to use the final SCF hessian as the initial starting hessian for a higher level treatment. This is easy to do in GAUSSIAN 82 since all useful information from a gradient optimization, such as final geometry, hessian and SCF wave function, is stored on a "checkpoint file" at the end of the job and this can be read in as input to start off another (better) optimization.

The purpose of this article is to present the numerical version of a previously presented gradient algorithm[13] (based on the RFO method of Simons and coworkers[14]) which can be used to efficiently locate both minima and transition states on potential energy surfaces for those wavefunctions for which analytical gradients are not available (the algorithm can, of course, be used with *any* wavefunction for which an energy can be calculated). The algorithm has been designed for use in the GAUSSIAN 82 program package and, in line with the comments given above, to make maximum use of second derivative information available from previous lower level optimizations, although it can be used without such information.

Section I discusses some of the problems associated with determining numerical, as opposed to analytical, gradients; Section II gives some details of the algorithm and its implementation within the framework of GAUSSIAN 82; Section III presents numerical results for a variety of systems at various levels of theory with comparisons, where applicable, between the gradient version of the algorithm and the standard DFP algorithm supplied with GAUSSIAN 82, and in Section IV the CH_3F^+ potential energy surface, previously studied at HF/6-31G* by Bouma, Yates, and Radom,[15] is reexamined at MP3/6-31G*.

I. USE OF NUMERICAL GRADIENTS

The Taylor series expansion of a smooth, continuously differentiable function $F(X)$ at the point $X + h$ about X is given by

$$F(X + h) = F(X) + hF'(X) + \frac{h^2}{2}F''(X) + \ldots \quad (1)$$

By ignoring the quadratic and higher terms, (1) can be rearranged to give the well known forward-difference approximation for the gradient at X

$$F'(X) = \frac{F(X + h) - F(X)}{h} + O(h) \quad (2)$$

The major source of error in an approximation such as (2) is due to truncation of the Taylor series; as indicated, this error is of order h, where h is known as the finite-difference interval. The computed function values used to calculate $F(X)$ are also subject to error. This gives rise to cancellation (or condition) error when the two values are subtracted. Such error is proportional to $1/h$, and consequently truncation and cancellation errors oppose each other, i.e., decreasing one means an increase in the other. When choosing a suitable finite-difference interval one has to balance these two errors; h should be small, to reduce the truncation error, but not so small that the function values will be subject to large cancellation error.

By expanding the function at the point X-h

$$F(X - h) = F(X) - hF'(X) + \frac{h^2}{2}F''(X) + \cdots \quad (3)$$

the analogous backward-difference formula can be obtained

$$F'(X) = \frac{F(X) - F(X - h)}{h} + O(h) \quad (4)$$

As in the case of the forward-difference approximation, the truncation error for backward-difference is also of order h.

If the expansions given by (1) and (3) are subtracted and divided by h the central-difference approximation to the gradient is obtained

$$F'(X) = \frac{F(X + h) - F(X - h)}{2h} + O(h^2) \quad (5)$$

The truncation error in the central difference formula is of order h^2, since the terms involving $F''(X)$ cancel. Central-differences are thus inherently more accurate than forward differences, but have the disadvantage that each variable has to be stepped twice in order to estimate its gradient, as opposed to only once with the other two formulae.

From a practical point of view, therefore, gradients should be estimated by a one-step, finite-difference approximation, e.g., forward-differences until this becomes unreliable, and only then should a switch be made to central differences. Forward-differences usually provide approximate gradients of acceptable accuracy unless $\|F'(X)\|$ is small. Since $\|F'(X)\|$ approaches zero at the stationary point, this means that as the optimization converges, the forward-difference approximation becomes less and less accurate and may be inappropriate during the last few iterations. That forward-differences are in error at the stationary point can be clearly seen from Figure 1: A forward-difference approximation to the gradient at X_0 — the bottom of the well — would give a non-zero value and indicate that the minimum should be shifted (incorrectly) towards $X_0 - h$; the central-difference approximation is obviously much more reliable.

In general, forward-differences cease to be accurate when:

(i) The difference in function values (energy) is small relative to the finite-difference interval.

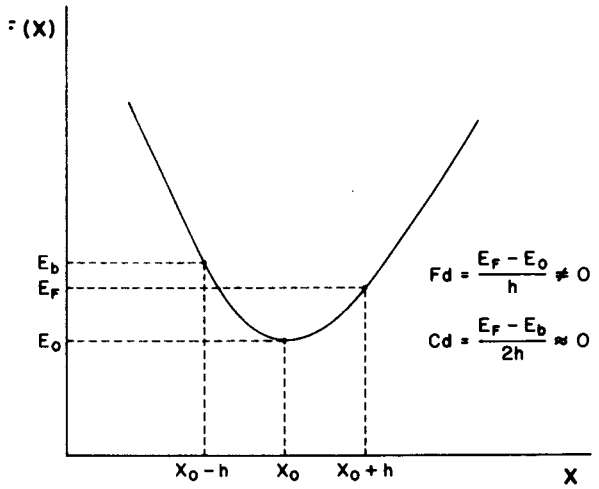


Figure 1. Forward-difference (Fd) versus central-difference (cd) estimate of the gradient at a stationary point (see the text for more details).

(ii) The change in a given variable during an iteration is less than the finite-difference interval.

If either of the above holds, it is probably advisable to switch to central-differences.

If a reliable estimate of the second derivative is available, it is possible to improve the accuracy of the simple forward-difference approximation. Truncating (1) after the quadratic term gives a corrected forward-difference formula

$$F'(X) = \frac{F(X + h) - F(X)}{h} + \frac{h}{2}F''(X) + O(h^2) \quad (6)$$

By including second derivative information in this way it is possible to extend the lifetime of the forward-difference approximation, and, provided the convergence criteria are not too stringent, to use forward-differences throughout the entire optimization.

II. THE ALGORITHM

As mentioned in the Introduction, the gradient version of the algorithm has already been presented[13]; the major changes in the numerical version involve consideration as to what finite-difference approximation to use when estimating the gradient and what the associated finite-difference interval, h, should be. Several of the features incorporated into the algorithm have been taken from ref. 16, which contains useful discussions about the problems associated with numerical evaluation of gradients.

Three options for specifying the finite-difference approximation are supported:

(i) Use the corrected forward-difference formula [eq. (6)] throughout the optimization.

(ii) Use central-differences [eq. (5)] throughout.

(iii) Use corrected forward-differences with a test for accuracy and switch to central-differences when this test fails.

There are also three options for specifying the finite-difference interval:

(i) Use a default interval calculated internally by the program [see eq. (7)].

(ii) Read in a finite-difference interval.

(iii) Estimate the "best" finite-difference interval at the first point before starting the optimization.

In general, the above options can be applied independently to each variable; thus one variable could use the corrected forward-difference formula with a default interval while another uses central-differences with the finite-difference interval read in.

The default finite-difference interval is calculated according to ref. 16.

$$h = 2\rho\sqrt{\frac{\epsilon_A}{1 + |F(X)|}} \qquad (7)$$

where ϵ_A is an estimate of the error in the function (energy) values $F(X)$. ϵ_A is probably the hardest quantity to estimate in the whole algorithm; after some experimentation a value of 10^{-7} was taken, which, with $\rho = 10$, gives finite-difference intervals of $5 \times 10^{-3} - 2 \times 10^{-4}$ a.u. for systems with energies between 1 and 1000 hartrees. All arithmetic was in double precision.

By seeking a balance between truncation and cancellation errors it is possible to estimate a "best" initial finite-difference interval. A procedure that attempts to select h as the smallest value that results in an acceptable cancellation error has been included in the algorithm; this is a slightly modified version of "Algorithm FD" given in section 8.6.2.2 of ref. 16. Estimation of the best h involves several additional function evaluations before the optimization starts however, and in most cases the default interval is sufficient.

The finite-difference interval is allowed to change during the course of the optimization as the variables themselves change. The interval for the jth variable at the kth iteration is given by

$$h_j = \frac{h_j^\circ(1 + |X_j^k|)}{(1 + |X_j^\circ|)} \qquad (8)$$

where the superscript $^\circ$ refers to the initial value of the quantity. In order to start the optimization properly, an initial estimate of the Hessian matrix must be supplied. Ideally, this should be the final Hessian from a previous optimization at a lower level, but if such a Hessian is unavailable then diagonal hessian matrix elements can be estimated according to

$$F''(X) = \frac{F(X + h) - 2F(X) + F(X - h)}{h^2} + O(h) \qquad (9)$$

Equation (9) can be obtained by subtracting (1) and (3) so as to eliminate the term in $F'(X)$. As indicated, the truncation error is of order h.

If no other options are specified, then by default the initial finite-difference interval is calculated according to (7), an initial (diagonal) Hessian formed using (9) and gradients calculated using the corrected forward-difference formula (6) with a switch to central-differences (5) if either of the two criteria given in Section I are satisfied (the energy change is considered to be "small" if it is less than 10% of the square of the finite-difference interval). Since estimating second derivatives involves calculating $F(X + h)$ *and* $F(X - h)$, then the gradient can be calculated using central-differences on the first cycle at no extra cost.

An additional feature of the algorithm is a line search option which can be used when looking for minima. This involves calculating the energy at additional points along the search direction and fitting a quadratic to give an estimate of the minimum, i.e., the greatest energy decrease along the line. The line search is fairly simple: the new step and energy are calculated in the normal manner and then, provided the original step was not scaled, an additional step of length 30% of the original step is taken and the energy found at this new point along the line. If the new energy is higher than the previous energy, a parabola is fitted through the *original* point, the point after the *first* step and the new point after the *additional* step—the minimum of this parabola forms the next estimate of the stationary point (minimum) being sought. If the new energy is lower, then a further step along the line is taken (of length 30% of the original step + total increment) and this process is repeated until the energy rises, with a parabola being fitted through the last three points found. A check is made that the energy at the estimated minimum of the parabola is the lowest energy found along the line; if not, then a new parabola is fitted through this point and two of the others. If the original step was scaled because its length was greater than the maximum allowed steplength

(DMAX) then all subsequent steps along the line are of length DMAX; if the original step resulted in the energy rising, then the step-length is halved repeatedly until a lower energy is found, with, again, a parabola fitted through the last three points. Note that the line search is only attempted if the hessian at the start of the search is positive-definite (since only then is the search direction guaranteed to be a descent direction) and if the RMS gradient is greater than a user supplied tolerance (default 0.003).

The line search as described above requires at least two additional function evaluations per cycle, and sometimes more (although rarely more than three); consequently it is not really worthwhile if the number of variables to be optimized is small, since the gain due to the *decrease* in the number of cycles to achieve convergence is offset by the *increase* in the number of function evaluations per cycle; however, there can be considerable savings with a large number of parameters. The coding in GAUSSIAN 82 is such that the time taken to evaluate the gradient analytically is comparable to the time for a single function evaluation, at least for SCF wavefunctions, so that there is nothing to be gained from a line search, regardless of the number of variables. This is why the *analytical* version of the algorithm does not incorporate a line search.

As well as the features discussed above, most of the options available to the gradient version of the algorithm are also available in the numerical version, including alternative Hessian updates and mode following for transition states (although, as mentioned in the introduction, the idea behind the numerical algorithm is to start an optimization from a previously located stationary point at a lower level and *not* to explore the energy surface in any detail). A full list of options is given in Table I. Again, as with the gradient version, most of these options can be invoked by specifying simple keywords on the route card.

III. APPLICATIONS

In order to test the performance of the algorithm a series of optimizations were carried out at the SCF level. In this way a direct comparison between the numerical and analytical versions of the algorithm could be made. Initially, a series of minimizations at the 3-21G level were done, starting from STO-3G geometries. Two sets of calculations were performed; one in which either an approximate or an exact STO-3G hessian was used to start off the optimization, the other where the initial second derivatives were obtained numerically, and within each set calculations were done with and without a line search and with and without finite-difference accuracy checking. Comparisons were also made with the default GAUSSIAN 82 algorithm (OPT = FP). Results are shown in Tables II and III.

The convergence criteria were the standard default values of 0.0003 on the RMS gradient; 0.00045 on the maximum gradient; 0.0012 on the RMS displacement and 0.0018 on the maximum displacement. These criteria are normally sufficient to give convergence on bond lengths to three decimal places and on angles to one decimal place. In all the examples reported in this section, both for minima and transition states, final structures obtained from the analytical and numerical versions of the algorithm agree with each other to this accuracy.

Table II presents the minimization results with the initial hessian read in. Comparison of the analytical with the numerical algorithm without a line search shows that, in every case, the number of cycles to converge was the same, suggesting that the numerical algorithm emulates the analytical very well; this is further borne out by the convergence pattern, which is essentially the same for both algorithms.

As anticipated, there are more disadvantages than advantages with the line search, even when the number of variables is fairly large. Only for NH_2CH^+ were there any significant savings. Of the smaller systems, only NH_3 benefitted from the line search. Judging from these results — admittedly not really enough to form any general conclusions — for systems with less than about five variables it is probably advisable to switch off the line search.

Switching off the finite-difference accuracy checking generally results in savings of only one or two function evaluations, since forward-differences are normally adequate right up to the last cycle. In fact, as mentioned

Table I. Options available for the numerical RFO algorithm.

```
C  OPTIONS      COMMON/IOP/
C
C  IOP(5)       nature of a required stationary point
C               0    find a TS                                  (default)
C               1    find a minimum
C
C  IOP(6)       maximum number of steps allowed
C               0    min(40,nvar + 20)                          (default)
C                    (where nvar = number of variables)
C               N    N steps
C
C  IOP(7)       convergence criteria on RMS gradient
C               0    0.0003                                     (default)
C               N    0.001/N
C               note: the other convergence criteria are
C               maximum gradient = 1.5 * RMS gradient
C               RMS displacement = 4.0 * RMS gradient
C               max displacement = 1.5 * RMS displacement
C
C  IOP(8)       maximum stepsize allowed during optimization
C               0    DMAX = 0.3                                 (default)
C               N    DMAX = 0.01 * N
C
C  IOP(10)      input of initial second derivative matrix
C               all values must be in atomic units
C               0    estimate diagonal matrix elements numerically
C               1    read in full FRCNST matrix (lower triangle)
C                    format (8F10.6)
C               2    read in selected elements of FRCNST
C                    READ I,J,FRCNST(I,J)      2I3, F20.0
C               3    read FRCNST matrix from the checkpoint file
C               5    read cartesian force constants from checkpoint file
C
C  IOP(13)      type of Hessian update
C               0    Powell update                              (TS default)
C               1    BFGS update                                (MIN default)
C               2    BFGS update with safeguards to ensure retention
C                    of positive definiteness
C
C  IOP(14)      minimum RMS gradient for which a linear search will
C               be attempted (MIN search only)
C               0    FSWTCH = 0.003
C               N    FSWTCH = 0.0001*N
C              -1    switch off linear search
C
C  IOP(16)      maximum allowable magnitude of the Hessian eigenvalues
C               if this magnitude is exceeded, the eigenvalue is replaced
C               0    EIGMAX = 25.0                              (default)
C               N    EIGMAX = 0.1*N
C
C  IOP(17)      minimum allowable magnitude of the Hessian eigenvalues
C               similar to IOP(16)
C               0    EIGMIN = 0.0001                            (default)
C               N    EIGMIN = 1.0/N
C
C  IOP(19)      search selection
C               0    P-RFO or RFO step only                     (default)
C               1    P-RFO or RFO step for "wrong" Hessian
C                    otherwise Newton-Raphson
C               (note: depending on IOP(14) a linear search along the
C                      search direction may also be attempted)
C
C  IOP(21)      expert switch
C               0    normal mode                                (default)
C               1    expert mode
C                    certain cutoffs to control optimization relaxed
C
```

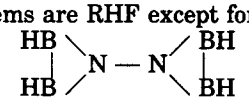
Table I. (*continued*)

```
C  IOP(33)       print option
C                0    on                                          (default)
C                1    off  turns off extra printing
C                (default of "on" by popular request)
C
C  IOP(34)       dump option
C                0    off                                         (default)
C                1    on   turns on debug printing
C
C  IOP(35)       restart option
C                0    normal optimization                         (default)
C                1    first point of a restart
C                     recover geometry etc..from checkpoint file
C
C  IOP(36)       checkpoint option
C                0    checkpointing as normal                     (default)
C                1    suppress checkpointing
C
C  IOP(38)       finite-difference control flag
C                0    forward-difference + accuracy check         (default)
C                     forward-differences unless considered inaccurate;
C                     in which case central differences
C                1    initial finite-difference approximation
C                     maintained throughout the optimization
C                     (see the IC array)
C
C  IOP(39)       stepsize control for finite-differencing
C                0    use internal default or IC array            (default)
C                1    calculate "best" interval for all variables
C                     before starting optimization proper
C
C  MODE FOLLOWING: mode following is turned on via the IC array, which is
C  input with the Z-matrix variables in Link 101. Adding 4 to the IC
C  entry for a particular variable will cause a transition state search
C  to follow the Hessian mode with the largest magnitude component for
C  that variable. Adding 10 to the IC entry for the Kth variable will
C  follow the Kth Hessian mode. ONLY ONE IC ENTRY SHOULD BE MODIFIED IN
C  THIS WAY, I.E., ONLY ONE MODE SHOULD BE FOLLOWED AT A TIME.
```

Table II. Minimization for various systems at 3-21G starting from STO-3G geometry and an approximate or exact Hessian (see the text for more details).[a]

System	No. of Variables	Hessian	No. Cycles to Converge (Function Evaluations in Brackets) with FD Checking No LS	with FD Checking LS	Forward-Differences No LS	Forward-Differences LS	Analytical Gradients
H_2O	2	approximate	4 (13)	3 (13)	4 (12)	3 (12)	4
NH_3	2	exact	5 (19)	3 (15)	5 (15)	3 (14)	5
H_2CO	3	exact	3 (13)	3 (17)	3 (12)	3 (15)	3
$MgCH_3Cl$	4	approximate	5 (25)	5 (35)	5 (25)	5 (32)	5
C_4H_4	4	approximate	5 (29)	5 (37)	5 (25)	4 (26)	5
$(BH)_4N_2$[b]	5	approximate	5 (35)	4 (37)	5 (30)	3 (24)	5
H_2NCH^+	7	exact	4 (39)	3 (29)	4 (32)	3 (28)	4
HNCHOH[c]	9	approximate	4 (47)	4 (51)	4 (40)	4 (43)	4

[a] All systems are RHF except for H_2NCH^+ (UHF).

[b] cyclic 

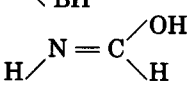
[c] formimidic acid 

Table III. Minimization for various systems at 3-21G starting from STO-3G geometry and a diagonal hessian calculated numerically via equation (6) (see the text for more details).[a]

System	No. of Variables	Hessian	No. Cycles to Converge (Function Evaluations in Brackets) with FD Checking No LS	with FD Checking LS	Forward-Differences No LS	Forward-Differences LS	DFP Algorithm
H_2O	2	numerical	4 (14)	4 (20)	4 (14)	4 (17)	6 (34)
NH_3	2	numerical	5 (18)	4 (21)	5 (17)	4 (20)	4 (22)
H_2CO	3	numerical	3 (15)	3 (18)	3 (15)	3 (17)	5 (40)
$MgCH_3Cl$	4	numerical	5 (32)	4 (32)	5 (29)	4 (30)	8 (70)
C_4H_4	4	numerical	3 (21)	3 (23)	3 (19)	3 (21)	11 (107) F
$(BH)_4N_2$[a]	5	numerical	6 (50)	5 (45)	6 (41)	5 (44)	11 (125+) F
H_2NCH^+	7	numerical	5 (51)	5 (63)	5 (47)	5 (51)	5 (55)
HNCHOH[b]	9	numerical	5 (73)	5 (73)	5 (59)	5 (64)	6 (93)

[a] All systems are RHF except for H_2NCH^+ (UHF); F indicates failure to converge.
[a] See comments under Table II.
[b] See comments under Table II.

at the end of Section I, forward-differences can be used throughout the optimization, even when the displacements are small. However, although there were no problems with any of the systems studied here, numerical instability can easily arise if forward-differences are persisted with when they are really inappropriate, and it is dangerous to switch off the accuracy checking, particularly if the convergence criteria are set any tighter than their default values. Assuming central-differences will be used only on the last or possibly the penultimate cycle, then accuracy checking should result in around N extra function evaluations or less, where N is the number of variables. It is probably worth accepting the extra energy calculations than risking the possibility of numerical instability.

Table III presents the same series of results, this time calculating (diagonal) second derivatives numerically. There is very little difference between the two tables for the smaller systems, the number of cycles to converge being about the same and the total number of function evaluations being only slightly greater with numerical second derivatives than with a full initial Hessian. For the three largest systems, however, there are considerable savings in using an initial hessian. Surprisingly, for C_4H_4, convergence was faster with numerical second derivatives (this was probably due to the approximate STO-3G hessian being a poor starting Hessian in this case). In general, however, it is certainly worthwhile utilizing external derivative information if it is available, since at least N function evaluations (the extra energy calculations required to estimate the second derivatives on the first cycle) can be saved, and often considerably more.

The last column of Table III gives the results for the Davidon-Fletcher-Powell algorithm. Clearly this algorithm is markedly inferior to the one presented here, taking about twice as many function evaluations (and sometimes more) to reach convergence, and in the case of the two cyclic systems, failing to achieve convergence after 107 and more than 125 function evaluations. Only for NH_3 and NH_2CH^+ are the two algorithms comparable. If comparisons are made with Table I, then the DFP algorithm is worse in every case.

The DFP algorithm,[8] as modified for GAUSSIAN 82,[9] involves an initial cycle to calculate second derivatives numerically ($2N + 1$ function evaluations) and then subsequent cycles in which gradients are calculated via a corrected backward-difference formula, similar to (6), with a three step line search ($N + 4$ function evaluations) or, if backward-differences are considered inaccurate, by central-differences with a line search ($2N + 4$ function evaluations). Central-differences are used if the RMS displacement is less than RMSLO (0.001) or greater than RMSHI (0.16), in which event second derivatives need to be recalculated. Additionally, the variables are scaled so that the second derivatives are of order unity.

On the face of it, this seems a fairly reasonable procedure, so why does the DFP algorithm perform so poorly? There are a number of reasons for this, as follows:

(a) The DFP algorithm itself is inherently inferior to the RFO method, as evidenced by the fact that more cycles are required to reach convergence.

(b) Many of the cycles involve at least $2N$ function evaluations either because second derivatives need to be recalculated or because the RMS displacement is too small, and central-differences are needed for the gradient. In this latter case, the situation is exacerbated by the fact that there are no tests on *individual* displacements, i.e., central-differences are either not used or used for *all* the variables.

(c) At least three additional function evaluations are required each cycle for the line search, which in many cases may not be necessary.

(d) The method is poor for variables which are strongly coupled, i.e., have large off-diagonal hessian matrix elements. This is why the two cyclic systems fail to converge.

(e) The finite-difference intervals for OPT = FP are fixed at 0.01 Å for bond lengths and 1° for angles. These intervals are at least an order of magnitude greater than those used in the numerical RFO algorithm, and consequently are likely to lead to numerical instabilities when estimating the gradient by backward-differences much earlier in the optimization, particularly if the second derivatives are not accurate enough.

Table IV presents a similar series of results for transition states, again at the 3-21G level starting from STO-3G geometries. Comparisons are made between the analytical algorithm with an initial hessian, the numerical algorithm with an initial hessian, and the numerical algorithm with diagonal second derivatives calculated internally. No comparisons can be made with the DFP algorithm for these systems since this algorithm cannot locate transition states.

Again, as with the minimizations, the numerical and analytical RFO algorithms follow each other closely, taking identical numbers of cycles to reach convergence. The results starting with a numerical (diagonal) hessian are much worse than those with an initial (STO-3G) hessian read in, the former taking more cycles and about twice as many function evaluations to converge than the latter, and, in one case—FCN—failing to converge at all, although this was due to problems with the SCF rather than with the optimization. Clearly, there is more to be gained by using external second derivative information for transition state searches than for minimization. This is doubtless due to the fact that, in many cases, a diagonal second derivative matrix simply does not have the correct eigenvalue structure (one negative eigenvalue) appropriate to a transition state, since it is often the off-diagonal matrix elements that are "responsible" for the negative eigenvalue, all the diagonal second derivatives being positive. For the systems examined here, only CH_3N had an appropriate diagonal Hessian; H_2CO had two negative eigenvalues initially and all the others had none. The fact that convergence is really fairly rapid indicates that the RFO method is quickly able to recover from the poor starting Hessian.

Finally, Table V presents results for a selection of the systems from Tables II-IV, this time at MP2/3-21G starting from the final 3-21G geometries and approximate hessians obtained from the previous optimizations. All MOs, including the core, were included in the correlation treatment (MP2 = FULL) in

Table IV. Transition state searches for various systems at 3-21G starting from STO-3G geometry and either an approximate/exact hessian or a diagonal hessian calculated numerically via equation (6) (see the text for more details).[a]

System	No. of Variables	Hessian	No. Cycles to Converge (Function Evaluation in Brackets) with Hessian Read In: Analytical	Numerical + FD Check	Numerical Hessian: Numerical + FD Check
HCN <--> HNC	3	approximate	4	4 (17)	5 (27)
FCN <--> FNC	3	approximate	8	8 (34)	6 (27) F
HCCH <--> H_2CC	5	approximate	3	3 (19)	5 (45)
H_2CO <--> H_2 + CO	5	exact	6	6 (40)	10 (71)
H_3CN <--> H_2CNH	6	approximate	3	3 (23)	5 (49)

[a] All systems are RHF except for H_3CN (UHF triplet); F indicates failure to converge

Table V. Optimizations of various systems at MP2/3-21G starting from 3-21G geometry and approximate hessian (see the text for more details).[a]

Systems	No. of Variables	Type	No. of Cycles to Converge (Function Evaluations in Brackets) Analytical	Numerical + FD Check	DFP Algorithm
H_2O	2	MIN	3	3 (11)	3 (15)
NH_3	2	MIN	3	3 (9)	10 (58)
HCN <--> HNC	3	TS	4	4 (18)	n/a
H_2CO	3	MIN	4	4 (16)	7 (55)
$MgCH_3Cl$	4	MIN	3	3 (15)	4 (38)
H_2CO <--> H_2 + CO	5	TS	6	6 (43)	n/a
HCCH <--> H_2CC	5	TS	5	5 (32)	n/a
H_2NCH^+	7	MIN	3	3 (31)	4 (60)

[a] All systems are RHF except for H_2NCH^+ (UHF)

order for a direct comparison with the gradient algorithm to be made. The last column in the table gives results using the DFP algorithm where this is possible (i.e., for the minima). Table V should give a better indication as to how the numerical algorithm is likely to perform in practice, since the geometrical changes on going from 3-21G to MP2/3-21G are probably closer to the kind of changes that will occur between, say, 6-31G* and MP3/6-31G*, which is an example of the type of optimization for which the algorithm was intended, than those between STO-3G and 3-21G, which are often quite significant, particularly for transition states.

As with all the previous optimizations, the numerical and analytical versions of the algorithm perform in essentially an identical manner. The minima converge in three to four cycles; the transition states take slightly more. Assuming these systems are fairly typical, then (for the numerical algorithm) a reasonable estimate of the number of function evaluations required to reach convergence, starting from a suitable initial geometry and hessian, would be around $4(N + 1)$ for minima and, say, $6(N + 1)$ for transition states, where N is the number of variables. The DFP algorithm, if anything, shows up even worse than before: only H_2O is comparable; NH_3 takes ten cycles and over six times as many function evaluations to converge, and H_2CO takes almost four times as many.

IV. THE CH_3F^+ ENERGY SURFACE

This is quite an interesting energy surface, showing at least two distinct *minima*—the fluoromethane ion, CH_3F^+ (1), and the methylene-fluoronium radical cation, CH_2FH^+, (2)—two transition states, and various decomposition pathways. As mentioned in the introduction, this surface has previously been characterized at HF/6-31G* by Bouma, Yates, and Radom,[15] who also optimized the minima at MP2/6-31G*, showing the importance of correlation for determining the structure of CH_3F^+, which has a very long C-F bond length at the SCF level that shortens dramatically on reoptimization at MP2. They were not able to optimize the transition states at correlated levels, however, and their final energies for these structures were based on SCF geometries.

In this section all the stationary points on the CH_3F^+ energy surface as characterized by Bouma, Yates, and Radom[15] were reoptimized at MP3/6-31G* (with frozen core). In all cases the optimizations were started from the HF/6-31G* structures—even for CH_3F^+ (to see how the algorithm handled the very poor initial geometry)—and in most cases an approximate 6-31G* hessian was also available. The exceptions were for the dissociative transition state (4), for which an approximate 3-21G hessian was used, and CH_2^+, for which the hessian was estimated numerically.

The structures of the various species, both at HF/6-31G* and MP3/6-31G*, are shown in Figure 2. Total and relative energies at these levels are given in Table VI. Column 5 of this table gives the "best" relative energies of Bouma, Yates, and Radom[15] (taken from Table II of ref. 15) and column 6 the relative energies from column 4 incorporating Bouma's estimated zero-point vibrational energy corrections. Most of the MP3 optimizations converged in six cycles or less, except for the CH_3F^+ minimum, which took 12 cycles (fairly good, considering the bad starting

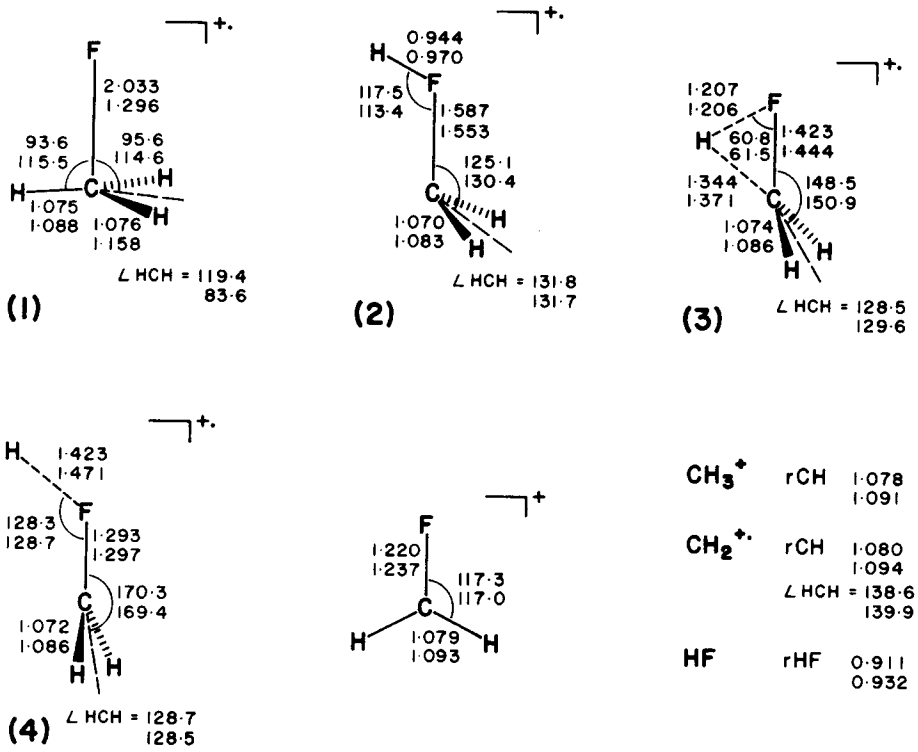


Figure 2. Optimized geometries for CH_3F^+ minima (1,2) transition states (3,4) and dissociation products. Optimized parameters are shown in the order HF/6-31G*, MP3/6-31G*. Lengths in Å, angles in degrees.

geometry), and, surprisingly, the dissociative TS, which took 11 cycles. In this latter case the fact that only a 3-21G hessian was used to start off the optimization may have been a contributory factor to the unexpectedly slow convergence.

Examination of Figure 2 shows that—apart from CH_3F^+ (1), for which the changes are major—there are only really minor changes between the HF/6-31G* and MP3/6-31G* structures. The geometry of CH_3F^+ at the MP3 level is very close to that found by Bouma, Yates, and Radom at MP2. Energetically, the MP3/6-31G* results are in good agreement with those of ref. 15, as can be seen by comparing the last two columns of Table VI. The various stationary points have the same energy ordering in both cases, and, apart from a slight overall lowering throughout column 6, the absolute values for the energy differences also agree very well. Only for the relative energy of the $CH_2F^+ + H^\cdot$ pair is there a significant difference.

Basically, these results support those of Bouma, Yates, and Radom and indicate that, at least for the CH_3F^+ energy surface, single-point calculations based on SCF structures give a reasonable picture of the energetics,

Table VI. Total and relative energies for minima, transition states and decomposition products on the CH_3F^+ energy surface: SCF values from ref. 15; MP3 values this work. Values in column 5 are the "best" values from Table II of ref. 15 (see the text for more details).

System		Total Energies HF/6-31G*	Total Energies MP3/6-31G*	Relative Energies HF	Relative Energies MP3	Relative Energies ref. 15	Relative Energies MP3 + ZPE
CH_3F^+	(1)	−138.60754	−138.89178	36	33	46	41
CH_2FH^+	(2)	−138.62133	−138.90446	0	0	0	0
$CH_2F^+ + H^\cdot$		−138.59202	−138.87305	77	82	96	64
bridge TS	(3)	−138.55361	−138.85439	178	131	129	123
$CH_2^+ + HF$		−138.56910	−138.83356	137	186	162	162
$CH_3^+ + F^\cdot$		−138.59560	−138.83762	68	175	179	169
dissoc TS	(4)	−138.53535	−138.83236	226	189	192	176

except in those cases where the geometry changes significantly with inclusion of correlation (i.e., for CH_3F^+ itself).

SUMMARY AND FINAL DISCUSSION

An efficient numerical algorithm for the location of both minima and transition states has been presented. It is designed for use in the GAUSSIAN 82 program package and, utilizing second derivative information from previous (gradient) optimizations at lower levels of theory, is considerably faster than the standard DFP algorithm supplied with GAUSSIAN 82. With a reasonable starting geometry and initial hessian, minima should be located in about $4(N + 1)$ energy evaluations and transition states in $6(N + 1)$, where N is the number of variables.

Although designed for use with wavefunctions for which analytical gradients are unavailable, the numerical algorithm does have certain advantages over the analytical for post-SCF wavefunctions, e.g., MP2. The current post-SCF gradient coding in GAUSSIAN 82 makes use of the fact that the energy is invarient to rotations amongst the occupied or amongst the virtual orbitals. Consequently, *all* the orbitals used in the SCF *must* be included in the post-SCF wave function, even when those orbitals have negligible effect on structures or relative energies, e.g., core orbitals or the highest virtuals. Such orbitals can easily be left out of a numerical optimization. Another factor is that, unlike SCF gradients, the CPU time required for a post-SCF gradient is often two to three times that for the corresponding energy evaluation; consequently—given the ability of the numerical algorithm to emulate the analytical—you are actually better off performing the optimization numerically when the number of variables is small. A further consideration is file storage; for the numerical algorithm no more disk space beyond that required for a single energy calculation is needed, whereas the additional storage for an analytical gradient calculation is considerable. In terms of CPU time, for a typical MP2 optimization, the numerical algorithm is usually faster than the analytical for three variables or less, particularly for second row systems, when many more "core orbitals" can be frozen. However, the numerical algorithm is heavily influenced by the time taken for the SCF, and if there are SCF convergence problems this will favor use of the analytical algorithm.

AVAILABILITY OF CODE

The FORTRAN code for the above algorithm is available and can be supplied, along with the modifications needed to run in a GAUSSIAN 82 environment, on request. Both VAX and IBM versions exist. In the near future the source and full instructions for use for both the numerical and analytical[13] RFO algorithms will be sent to QCPE for distribution; in the meantime all interested users should contact the author direct.

Thanks to P. M. W. Gill for useful discussions.

References

1. P. Pulay, in *Applications of Electronic Structure Theory,* H. F. Schaefer, Ed., Plenum, New York, 1977.
2. J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, *Int. J. Quantum Chem.,* **S13,** 225 (1979).
3. N. C. Handy and H. F. Schaefer, *J. Chem. Phys.,* **81,** 5031 (1984).
4. P. Jorgensen and J. Simons, *J. Chem. Phys.,* **79,** 334 (1983).
5. J. S. Binkley, M. J. Frisch, D. J. Defrees, K. Raghavachari, R. A. Whiteside, H. B. Schlegel, E. M. Fluder, and J. A. Pople, Carnegie-Mellon University, Pittsburgh, PA 15213, USA
6. R. D. Amos, Publication No. CCP1/84/4, SERC, Daresbury, Warrington, WA4 4AD, United Kingdom.
7. GAMESS (modified version from SERC, Daresbury, UK), original ref., M. Dupois, D. Spangler and J. J. Wendoloski, *Nat. Resour. Comput. Chem.,* Software Cat., Vol. 1, Prog. QG01, 1980.
8. R. Fletcher and M. J. D. Powell, *Comput. J.,* **6,** 163 (1963).
9. J. B. Collins, P. von R. Schleyer, J. S. Binkley, and J. A. Pople, *J. Chem. Phys.,* **64,** 5148 (1976).
10. W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory,* Wiley, New York, in press.
11. K. Raghavachari, R. A. Whiteside, J. A. Pople, and P. von R. Schleyer, *J. Am. Chem. Soc.,* **103,** 5649 (1981). (See C2H7+.)
12. M. J. Frisch, A. C. Scheiner, H. F. Schaefer, and J. S. Binkley, *J. Chem. Phys.,* **82,** 4194 (1985).
13. J. Baker, *J. Comp. Chem.,* **7,** 385 (1986).
14. A. Banerjee, N. Adams, J. Simons, and R. Shepard, *J. Phys. Chem.,* **89,** 52 (1985).
15. W. J. Bouma, B. F. Yates, and L. Radom, *Chem. Phys. Letts.,* **92,** 620 (1982).
16. P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization,* Academic Press, London, 1981.