Theoret. Chim. Acta (Berl.) 50, 299–312 (1979)

THEORETICA CHIMICA ACTA

An Efficient *ab initio* Gradient Program

Péter Pulay

Eötvös L. University, Department of General and Inorganic Chemistry, H-1088 Budapest, Muzeum körut 6–8, Hungary

An *ab initio* Hartree–Fock gradient program is described. It is characterized by (1) efficiency of the gradient evaluation, and (2) capability of handling higher angular momentum (*d* and *f*) basis functions. The latter are constructed from shifted Cartesian Gaussian *p*-type primitives. A satisfactory solution is presented for the problems connected with the neglect of small integrals in a gradient program. Methods for increasing the efficiency of the SCF procedure are discussed.

Key words: Gradient program, *ab initio* ~

1. Introduction

Analytic calculation of the energy gradient, i.e. the forces acting on the atoms, can significantly facilitate the determination of molecular geometries and potential constants by quantum chemical methods [1, 2]. The first such program in existence was MOLPRO [3]; a significant amount of work has been performed by it [4–6]. However, it was conceived to treat small and symmetrical molecules, and its efficiency proved insufficient for systematic calculations on larger molecules. This was due mainly to the fact that integrals in MOLPRO were not labelled and therefore it was not possible to neglect small integrals in a simple way. Recently, three other *ab initio* gradient programs have been reported: two adaptations of the GAUSSIAN 70 program system [7, 8] and a program based on POLYATOM [9]. As to the latter, published timing data show that its efficiency is not yet satisfactory. This objection is probably not true for the other two programs. However, a serious limitation of all these programs, except MOLPRO, is their inability to handle higher angular momentum (*d* and *f*) basis functions.

In view of the above facts, it was decided to write a new *ab initio* SCF gradient

0040-5744/79/0050/0299/$02.80

program, utilizing the accumulated experience. The two main features of the program are: the efficiency of the gradient calculation, and the ability to handle higher angular momentum functions. The requirements for a gradient program differ from that of an ordinary SCF program, and the techniques used in the new program, called TEXAS, are discussed here. In addition to its main parts: integral evaluation, SCF iteration and gradient calculation, TEXAS includes options to transform the forces to internal coordinates [1] and to generate distorted molecular geometries, in addition to a number of expectation values. These features, and problems pertaining to systematic geometry and force constant calculations, will be discussed in a forthcoming paper [10].

2. Basis Functions

All existing Gaussian integral programs use either Gaussian lobes or Cartesian Gaussians. The former have the advantage that the integral derivatives, necessary for gradient evaluation, can be easily programmed. However, a lobe program cannot be made as efficient as a program using Cartesian Gaussians because the shell concept, introduced by Pople and coworkers [11], cannot be utilized. Another disadvantage of the lobe basis set is the decrease in numerical accuracy for integrals involving *d* and *f* functions. Ahlrichs [12] has discussed the relative merits of Gaussian lobe *versus* Cartesian basis sets and the construction of an efficient Gaussian lobe program. His code is perhaps the ultimate in Gaussian lobe programs. Nevertheless, it is still not efficient enough for the present purpose, particularly in non-planar molecules where local symmetries in the lobe construction cannot be used.

In this connection, one may mention the recent suggestion [13, 14] that four Gaussian lobes, arranged at the corners of a tetrahedron, be used instead of the customary Gaussian lobe basis set which requires seven lobes for an sp^3 shell. Such a feature is available in MOLPRO but we abandoned it because of problems with *d*-contamination of the *sp* basis. This manifests itself in a gradient program by non-vanishing net torque of the forces. For a fuller discussion, see [15].

When choosing the basis set, it must be taken into account that the time-consuming part of the gradient evaluation closely resembles integral calculation. However, because of the added complexity, it takes usually 3 to 5 times as much CPU time as the integrals. Therefore, the integral section of the program gains importance in comparison with ordinary SCF programs. This speaks for Cartesian Gaussians which permit faster integral evaluation. A disadvantage of Cartesian Gaussians is that calculation of the necessary integral derivatives over *d* and *f* functions is very involved.

As an alternative to both lobes and Cartesian Gaussians, the following approach was chosen: only *s* and *p* Gaussian primitives are allowed. These are grouped, whenever possible, in shells which may be arbitrary combinations of s, p_x, p_y and p_z functions. The shells most often used are of the type *S*, *XYZ* and *SXYZ*, the latter denoting an *s* and three *p* functions with equal orbital exponents. Higher angular momentum functions are constructed from p-type primitives, in a similar spirit as p and d functions are constructed from lobes. The details of this construction are given in Table 1. This approach preserves the simplicity of an sp integral program, with efficient subroutines for the relatively few types of integrals encountered, and at the same time allows the use of higher angular momentum functions. If the recommendation in Table 1 is followed, contamination by higher spherical harmonics (in the first place contamination of the d's by g functions) is negligible, as shown by the very low net torque in calculations employing d functions. For an sp basis set, the net torque must, of course, vanish exactly.

As in the case of Cartesian Gaussians, the diagonal d functions of Table 1 contain a significant s-type contribution (note, however, that this contribution is not identical in the two cases: our d functions correspond closely to the Hermitte–Gaussian functions [16]). If the basis set is not saturated with respect to s functions, it is necessary to employ six d functions instead of five, to prevent the coupling of s and d functions which might bias the population of the d's. When using Cartesian Gaussians, the angular parts of the d functions are usually chosen as x^2, y^2, z^2, xy, xz, yz. In our case, the special $3s$ combination given in Table 1 can be substituted for one of the diagonal d functions, say d_{z^2}, leading to increased computational efficiency. We have found no way to eliminate the sixth d function completely, unless the basis is sufficiently saturated with respect to s functions.

Table 1. Construction of higher angular momentum functions from s and p type Cartesian Gaussians

Type	Function form[a]	Shells	Exponent	coefficient[b]	δ[c]	n	N[d]
$3d$	$2\eta x^2-1$; xy; xz	XYZ	η	$+1$ -1	$+0.1x$ $-0.1x$	6	2
$3d$	$2\eta y^2-1$; yz	YZ	η	$+1$ -1	$+0.1y$ $-0.1y$	4	2
$3s$[e]	$2\eta r^2-3$	S	0.8517143η 1.1η	0.83390 -1	0 0	2	2
$4f$[f]	$5x^2y-r^2y$; $5x^2z-r^2z$	YZ	1.0125722η 0.9874409η 1.0125722η	$+1$ -1.93874 $+1$	$+0.25x$ 0 $-0.25x$	6	3
$4f$	xyz	Z	η	$+1$ -1 -1 $+1$	$+0.25x$, $+0.25y$ $+0.25x$, $-0.25y$ $-0.25x$, $+0.25y$ $-0.25x$, $-0.25y$	4	4

[a] The common factor $\exp(-\eta r^2)$ has been omitted.

[b] Coefficients refer to normalized primitives.

[c] Magnitude and direction of the displacement of the shell center, expressed in units of the inverse square root of the actual orbital exponent.

[d] N is the number of shells, n the number of primitives used in the construction.

[e] The function d_{z^2}, constructed analogously to $d_{x_2^2}$, may be used instead.

[f] Construction of the functions y^2x, y^2z, and z^2x, z^2y is analogous, using shells XZ, displaced in the y direction, and YZ, displaced in the x direction, respectively.

The situation is somewhat different for *f* functions. When using Cartesian Gaussians, the usual procedure is to employ all ten possible cubic polynomials of *x*, *y*, *z* in the preexponential factor, to prevent the coupling of the *f* functions to *p*'s. This procedure is expensive and can be avoided by using *f* functions which are orthogonal to all *p* functions. The *f* function set of Table 1 was constructed to insure this orthogonality to a high degree of accuracy. However, the orthogonality of the *f* functions to each other had to be abandoned; this does not influence the results but is a minor drawback for the interpretation of the wave function. Using the construction in Table 1, only seven *f* functions need be employed.

A further advantage of the present basis set is that new expectation values can be easily programmed, in contrast to programs using higher Cartesian Gaussians.

3. Integral and Gradient Calculation

As in most modern integral programs, all integrals over four particular shells are calculated simultaneously. Following Dupuis *et al.* [13], integrals over four shells are called a block of integrals. If our method of constructing higher angular momentum functions is adopted then the primitive shells making up a contraction may have different centers. GAUSSIAN 70 [17] makes use of the fact that all primitives in a contraction have the same orbital center. In GAUSSIAN 70, first a small number (70 instead of 256 for four sp^3 shells) of integrals are constructed in a special coordinate system. These integrals can be summed over the contractions in one of the charge distributions, and then transformed back to the original coordinate system. The important point is that the transformation is outside the innermost loop, and is proportional only to the square (and not to the fourth power) of the number of primitives per contraction. This advantage was sacrificed in the current version of TEXAS, and consequently we may expect that for highly contracted functions it will be less efficient than GAUSSIAN 70. However, our experience, as well as that of Schlegel *et al.* [7] shows that highly contracted basis sets, like the STO-3G set [11] are not adequate for systematic force constant calculations. The use of such basis sets is particularly hard to justify if the gradient is to be calculated because their main computational advantage, fast SCF iteration, is much less important in this case.

In order to speed up the integral and force calculation, quantities appearing in the integral formulas are precomputed as far as possible. The total number of these quantities may be very large; in order to keep the storage area limited and eliminate external storage, the algorithm shown in Table 2 is used. Using this algorithm, the core storage needed for integral evaluation is essentially independent of the size of the basis set. The incomplete gamma function is evaluated by the compact and fast rational approximation given by Spellucci and Pulay [18].

Integral blocks in the current version of TEXAS are calculated in the classical canonical order, with *I* running from 1 to *N*, *J* from 1 to *I*, *K* from 1 to *I* and *L* from 1 to min (*J*, *K*). The indices here refer to shells, not to basis functions. Following an idea by Meyer [19], Ahlrichs [12] uses a novel integral order, in which the indices

Table 2. Precomputing algorithm for two-electron integrals and their derivatives

```
   DO 1 I=1, NCSHELL (loop over contracted shells)
      J=1
3:    JBEG=J
4:    precompute all possible data for shell pair I, J; store the data in array PRELIM
      IF (PRELIM is full or J=I) GO TO 5
      J=J+1
      GO TO 4
5:    JEND=J
      DO 2 K, L (loop over contracted shells K and L in lexicographic order; precompute everything for
      shell pair K, L)
         DO 3 JJ=JBEG, JEND
            calculate integrals or integral derivatives using precomputed data
3:       END LOOP JJ
2:    END LOOP K and L
      IF (J.EQ.I) GO TO 1
      J=J+1
      GO TO 3
1: END LOOP I
```

run as follows: $I=1$ to N; $J=1$ to I; $K=1$ to J; $L=1$ to K, and three integrals are computed simultaneously: $(IJ\,|\,KL)$, $(IK\,|\,JL)$ and $(IL\,|\,JK)$. This can make the subsequent SCF iteration significantly (up to three times) more efficient in terms of CPU time, compared to the customary method where contributions from each integral to all Fock matrix elements are calculated independently. (Note that a third method, much inferior to those mentioned but used in some programs, is to generate the integrals in the canonical order and then transform the integral list to the Roothaan supermatrices.) Ahlrichs [12] transforms the original integral list to the P supermatrix, defined as

$$P_{IJ,KL}=[4(IJ\,|\,KL)-(IK\,|\,JL)-(IL\,|\,JK)](1-0.5\delta_{IJ,KL}) \tag{1}$$

This transformation has the unfortunate property that it interferes with the neglect of small integrals, as the probability that an element of the P supermatrix is smaller than a given threshold is much less than the probability that an integral is less than it. For larger molecules, the file containing the P supermatrix may significantly exceed the size of the integral file. This is a decided disadvantage of the Ahlrichs program. A preferable procedure, at least for large molecules, is to calculate the integrals in the Meyer–Ahlrichs order in triplets, but perform the transformation (1) in each SCF iteration immediately preceding the processing of the integrals. Although this procedure consumes a little CPU time, it is more than made up by the decrease in I/O time.

As mentioned above, the current version of TEXAS does not use the Meyer–Ahlrichs integral order, for two reasons: it interferes with the precalculation routine, and it is not particularly helpful in the gradient calculation. However, it was found that SCF time often dominates total calculation times, particularly for smaller basis sets. Presently there is a modification under way which would allow the use of this alternative integral order. In accordance with the shell principle, triplets of blocks are calculated rather than triplets of individual integrals.

The precomputation algorithm, as well as the utilization of symmetry and the neglect of small integrals result in a scrambled integral list. This means that integrals must be labelled. Unlike most other programs which label the integrals individually, only whole blocks are labelled in TEXAS. This shortens the integral file to nearly a half. This feature is considered important, as the length of the integral file and the associated input–output time may be limiting factors in large-scale calculations.

The problem of labelling is closely connected with the neglect of small integrals. A two-electron integral over four primitive Gaussians has the form

$$(ab|cd)=\int a(1)b(1)r_{12}^{-1}c(2)d(2)\,d\tau_1\,d\tau_2=S_{ab}\times S_{cd}\times F \tag{2}$$

where S_{ab} is the overlap integral between the 1*s*-type counterparts of a and b. The magnitude of the integral is mainly determined by the product $S_{ab}\times S_{cd}$, as F is a relatively weak function of orbital exponents, positions and types. Block labelling means that the neglect criterion must be based on the type-independent factor, i.e. on $S_{ab}\times S_{cd}$. This choice has the important advantage of making the total energy strictly invariant with respect to the rotation of the Cartesian axes, in contrast to neglect criteria based on the magnitude of individual integrals. Using this procedure, it is not possible to set an upper limit on the magnitude of the neglected integrals, as F has no upper limit: for coincident functions with equal exponents, it is proportional to the square root of the orbital exponent. At first this may seem a disadvantage. Closer inspection reveals, however, that an integral can significantly exceed the limit set for the overlap factors only if both charge distributions, *ab* and *cd*, have high orbital exponents. In this case, however, the corresponding diagonal Fock matrix elements must be large, and the influence of the neglected integral on the energy and other expectation values will be small. Indeed, the present neglect scheme seems more natural than a flat limit set on the integrals, without reference to the properties of the participating charge densities.

Neglect of small integrals is very important in large molecules. It can, however, lead to discontinuities of the potential surface. These are very annoying in calculating energy derivatives, particularly higher ones. A method of suppressing such discontinuities was suggested by Meyer and programmed in the "*S*" version of MOLPRO [3]. An improved version thereof is used in TEXAS. The essence of this method is to replace the overlap factor $S_{ab}\times S_{cd}$ in the integral expression (2) by $f(S_{ab}\times S_{cd})$ where the function $f(x)$ satisfies

$$\begin{aligned} &f(x)=x \quad \text{if } x\geqslant\varepsilon_1\\ &f(x)=\varphi(x) \quad \text{if } \varepsilon_1\geqslant x\geqslant\varepsilon_2\\ &f(x)=0 \quad \text{if } x\leqslant\varepsilon_2 \end{aligned} \tag{3}$$

the function $\varphi(x)$ being chosen in such a way that $\varphi(\varepsilon_1)=\varepsilon_1$, $\varphi(\varepsilon_2)=0$. Meyer originally suggested a simple linear function for φ. This restores the continuity of the surface but not its smoothness, i.e. the continuity of the gradient. A cubic spline function is used instead in TEXAS, with the additional constraints on the first derivative of φ:

$$\varphi'(\varepsilon_1)=1 \qquad \varphi'(\varepsilon_2)=0 \tag{4}$$

This works very well. Default values are $\varepsilon_1=10^{-6}$, $\varepsilon_2=10^{-7}$, the maximum deviation $|f(x)-x|$ being $2\varepsilon_2=2\times 10^{-7}$.

It is worth while to compare the present approach with the charge conserving integral approximation of Wilhite and Euwema [20]. In their method the overlap function S_{ab} itself is approximated, in the simplest case by truncation. In contrast to the usual neglect scheme, this method is applicable to one-electron integrals, too. One hopes that this results in a more balanced integral approximation: in particular, there are no more excess nuclear attraction integrals which have their compensating electron–electron repulsion integrals deleted.

Without claiming any priority, we may note that, following an idea of Meyer, this approach was implemented in the "*S*" version of MOLPRO [3]. It was not published because we have found it rather inefficient. Even for a fairly large molecule (cyclohexane) it accelerated integral computation only by a factor of 1.6 with a reasonable threshold on the overlap function (10^{-5}). It is possible, however, that for larger systems the charge-conserving integral approximation will be of value. For smaller systems, there are not too many vanishingly small charge distributions, although many two-electron integrals are small because they contain the product of two overlap factors, neither very small in itself. It may be noted that a novel type of integral approximation which is also charge-conserving but much more efficient than the Wilhite–Euwema method [20] is being implemented in TEXAS. This method will be described separately.

We have learned of the program HONDO, developed by King and coworkers [13] when the present project was well under way. Their method of integral evaluation, based on the ingenious use of the properties of orthogonal polynomials [21] seems very promising for higher angular momentum functions, as the computing work increases only linearly with the combined angular momentum of the basis functions in a two-electron integral. E.g. evaluation of an integral of the $(dd\,|\,dd)$ type takes perhaps two times as much computer time as that of a $(pp\,|\,pp)$ integral. In contrast, a $(dd\,|\,dd)$ integral costs about 16 times as much time as a $(pp\,|\,pp)$ one in TEXAS, and the ratio is similar in other programs, too. However, to compare the two methods realistically, one should take into account that, with the usual basis set composition, only a small fraction of the integrals are of the $(dd\,|\,dd)$ type; most integrals contain only one *d* function and in this case the advantage of HONDO is much less. As a typical case, we may assume that about 20% of the basis functions are of *d* type. In this case, 41% of the two-electron integrals contain no *d* functions, 41% contain one *d*, 15.3% two *d*'s, and only 2.7% three or four *d*'s. Integral evaluation time per primitive integral is then only twice larger for the *spd* basis set than for an *sp* basis. However, for a basis set with a significant proportion of *d* functions, HONDO will doubtlessly outperform TEXAS. One disadvantage of the algorithm used in HONDO is that it is not possible to perform part of the integral computation outside the loop over the innermost contraction. Although higher angular momentum functions are rarely contracted, this feature may be still important, as most of the two-electron integrals contain both high and low angular momentum functions.

4. Symmetry and the Storage of Integrals

There are two diametrically opposite philosophies with regard to symmetry. At one extreme, GAUSSIAN 70 [17] does not use symmetry, reasoning probably that large molecules are seldom symmetrical. This is true to a certain extent, particularly in the present context, i.e. for potential surface calculations. However, the systems often have some limited symmetry, e.g. a mirror plane. At the other extreme, there are programs which can use high point group symmetry in a sophisticated manner. Such programs make impressive calculations on large symmetrical molecules possible. Unfortunately, molecules in highly symmetrical nuclear conformations possess disappointingly few interesting properties. Moreover, most of the advantages of symmetry can be utilized by considering only a simple Abelian subgroup of the full molecular point group. To our knowledge, this approach was first followed in our program MOLPRO [3] and has been used since in several programs, including MOLECULE [22] and Ahlrichs' program [12]. It was retained in TEXAS where only point groups isomorphic to C_{2v} or C_2 are allowed. Symmetry operations (inversion, mirror planes or twofold rotations) must be specified in the input. The transformation properties of the basis functions are, on the other hand, determined automatically from the overlap matrix between the original and the transformed basis set.

In the course of integral and force calculation, an integral block is skipped unless all symmetry-related blocks precede it in the canonical integral order. As pointed out by Ahlrichs [12], most of the testing for symmetry equivalence can be done in the outer loops. Of course, further savings are possible for integrals which vanish by symmetry. The latter type is particularly important in planar molecules where all integrals which contain an odd number of antisymmetric (p_z) functions vanish. To take advantage of this, *SXYZ* shells must be split to *SXY* and *Z* shells: if shells are defined in a way that they contain either symmetric or anti-symmetric functions but not both, then integral blocks over an odd number of antisymmetric shells may be omitted.

A special feature of symmetry handling in TEXAS is that it will accept not only perfect but also broken symmetry, i.e. some basis functions may not conform to symmetry. This is useful in force constant calculations because often a single hydrogen atom alone deviates from symmetry. For this method to be useful, at least 80 and preferably 90 per cent of the basis functions should conform to symmetry.

5. SCF Procedure

The SCF procedure in TEXAS has been taken over to a large extent from MOLPRO [3], the SCF package of which was written by W. Meyer. We discuss some of its features because the MOLPRO description [3] did not appear in the open literature.

Construction of the Fock matrix is facilitated by premultiplying the integrals by a permutation number

$$(2-\delta_{IJ})(2-\delta_{KL})(2-\delta_{IJ,KL}).$$

According to the shell structure of TEXAS, the permutation number is applied to blocks rather than to individual integrals. This feature simplifies the construction of the Fock matrix considerably.

A novel feature which emerged from discussions with W. Meyer is the use of a double integral file. The first file contains integrals for which the $(S_{ab} \times S_{cd})$ factor exceeds a certain threshold, say 10^{-4}, while the second contains integrals below this threshold. If the SCF iteration has advanced far enough, the contribution of the integrals on the second file to the Fock matrix does not change significantly for successive iterations. This contribution is then evaluated only in every third iteration. The partial Fock matrix from the small integrals is simply added to the Fock matrix in the following two SCF steps. Some discretion is necessary to use this feature, as too high a threshold may cause the SCF procedure to diverge, while a too low value is inefficient. It was found to work best in molecules above a certain size, say, more than 5 non-hydrogen atoms.

One peculiarity of gradient calculation is that the forces are more sensitive to SCF convergence than the total energy, in accordance with their first-order dependence on the wave function. The worst cases are planar π-systems distorted from planarity. E.g. a calculation on slightly non-planar benzene was started with the wave function at the planar geometry. Total energy was stable to 10^{-7} E_h from the 8'th step on, while the two-electron energy had converged to 8×10^{-7} E_h and the dipole moment to 6×10^{-5} a.u. by step 10. The force obtained at this point was still in error for some components by as much as 2×10^{-5} a.u. $\approx 1.8 \times 10^{-4}$ md ($1 \text{ md} = 10^{-8}$ N; $1\, E_h \approx 4.349814$ aJ), and the total torque of the forces was 5×10^{-4} md Å. Errors of this magnitude are, of course, not important for geometry optimization, neither for the calculation of quadratic force constants. However, they may influence the cubic and higher force constants significantly. In the above example, after ten more SCF iterations the two-electron energy has stabilized to 10^{-8} E_h, and the total torque diminished to 1.6×10^{-6} md Å; we estimate that the force components have converged to better than 10^{-6} md.

In view of the sharp convergence necessary for the gradient, it is important to use a good SCF algorithm. Most methods suggested for improving SCF convergence aim only at eliminating divergence at the beginning of the iteration. Such methods help very little in the later stages of the SCF procedure. TEXAS currently uses a damping procedure which is carried out on the Fock matrix instead of the density matrix; errors arising from the non-idempotency of the interpolated density matrix are avoided this way. To our knowledge, this approach was first used in MOLPRO [3]. The default procedure is to interpolate the Fock matrix according to

$$F'_n = (1-p)F_n + pF_{n-1} \tag{5}$$

with obvious notation. The parameter p is near 0.2 at the beginning of the iteration. Its value is increased if successive changes in the two-electron energy are of opposite sign (signifying that the SCF procedure oscillates) and diminished if the changes are of the same sign. Using starting vectors from minimum-basis calculations, satisfactory convergence was achieved in each case investigated so far. However, as mentioned above, very sharp SCF convergence is sometimes necessary in gradient calculations. Therefore, a simple convergence acceleration technique has been developed; it is described in the Appendix.

Utilization of symmetry in the construction of the Fock matrix depends on the type of the symmetry. In a closed-shell calculation with perfect symmetry, a variation of the time-saving algorithm of Dacre [23] is used. Each block of integrals is multiplied by a symmetry factor which is the number of distinct blocks into which it is transformed under symmetry. Two blocks are not considered distinct if they differ only in the order of the indices. The Fock matrix is constructed using the unique blocks and is subsequently symmetrized according to

$$F_{ij}^{\mathrm{sym}} = F_{ij} + p_{Ri} \times p_{Rj} \times F_{Ri,Rj} \tag{6}$$

where i and j are the indices of the basis functions χ_i and χ_j with transformation properties

$$\hat{R}\chi_i = p_{Ri}\chi_{Ri} \tag{7}$$
$$p_{Ri} = \pm 1$$

The symmetrization operation (6) has to be performed for a full set of generating operators $\hat{R}$ of the group. More sophisticated versions of this algorithm, which are applicable to arbitrary point groups, have been described by Elder [24] and by Dupuis and King [25]. However, as discussion in Sect. 4, the restriction to simple Abelian point groups is rarely a real disadvantage.

The Dacre method is very advantageous because the symmetrization is carried out on the Fock matrix and is therefore a trivial operation. It is not applicable, however, if symmetry is imperfect or if the density matrices lack the total point group symmetry, as it may be the case for unrestricted Hartree–Fock calculations. In these cases symmetry-related integrals must be generated from the unique integral list, and the savings extend only to the calculation, storage and reading of the integrals but not to the construction of the Fock matrix.

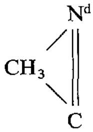
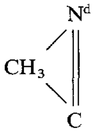
Note that the Fock matrix itself is not factored into symmetry blocks. This is not a significant disadvantage from the computational viewpoint, as symmetry blocking influences only matrix diagonalization and other incidental operations. It makes it, however, more difficult to establish the identity of the orbitals in symmetrical molecules. A simple remedy for this is to transform the Fock matrix to symmetry-adapted basis after it has been constructed, avoiding thus the difficulties associated with the transformation of two-electron integrals.

As it was shown earlier [26], both Roothaan's restricted open-shell method and the unrestricted Hartree–Fock method allow the evaluation of the gradient. MOLPRO did contain both versions beyond the usual closed-shell force routine. Only the closed-shell package has been implemented in TEXAS so far. The general conditions on the wave function for analytic gradient evaluation are discussed in [2] and particularly in [27].

6. Program Organization

Programming techniques are outside the scope of this paper but a few points concerning program organization deserve mentioning. For easy transferability,

Table 3. Some representative computer times[a]

Molecule	Symmetry	Basis	Computer	T_{int}	T_{SCF}	T_{force}	$T_{\text{force}}/T_{\text{int}}$
Ethylene	C_s	4-21	CDC6600[b]	32	8.4	123	3.8
Ethylene	C_{2v}	4-21	CDC6600[b]	24	8.3	96	4.0
Ethylene	C_{2v}	6-31*G**	CDC6600[b]	367	48	1467	3.7
Butadiene	no	4-21	CDC6600[b]	290	111	1026	3.5
Butadiene	C_2	4-21	Harris[c]	481	186	2094	4.4
	no	4-21	Harris[c]	313	112	1236	3.9
	no	6-31*G**	Harris[c]	5400	537	28361	5.2
C_3H_7NO[e]	no	4-21	CDC6600[b]	540	202	2010	3.7
Benzene	C_{2v}	4-21	CDC6600[b]	244	93	1227	5.0
Benzene[f]	C_s	4-21	CDC6600[b]	432	167	1809	4.2
Pyrrole[f]	C_s	4-21	CDC6600[b]	277	91	1254	4·5
ONF	C_s	$7s3p$[g]	Univac1110	65	22	218	3.4
CF_3NO	C_s	$7s3p$[g]	Univac1110	608	302	1515	2.5
$CH_3N{=}NCH_3$	C_2	$7s3p$[g]	Univac1110	462	197	1114	2.4
BH_3CO	C_s	$7s3p$[g]	Univac1110	129	39	449	3.5

[a] In seconds. SCF times refer to a single iteration. Integral threshold is $\varepsilon_2 = 10^{-7}$. The 4-21 basis is described in [10].

[b] TM time, as defined on the CDC6600 computer of the University of Texas at Austin. This is somewhat larger than the CPU time for the forces and integrals, and is about the double of the CPU time for the SCF step. Only the RUN compiler is available in Austin and this was used throughout. If the more efficient FTN compiler is used, all CPU times are approximately halved.

[c] Elapsed time on the Harris Slash 4 minicomputer.

[d] The saddle point geometry on the methyl isocyanide rearrangement potential surface.

[e] *N*-methyl acetamide.

[f] Non-planar geometry.

[g] Integral evaluation is faster by a factor of about 1.5 with the 4-21 basis set than with the $7s3p$ set, and the results are comparable. The $7s3p/3s$ basis was contracted to $4s2p/2s$. The author is indebted to Dr.,E. Flood (Tromsø) for the timing data.

TEXAS has been written fully in standard FORTRAN[1]. Data storage is fully dynamic: all real arrays are stored linearly in a single large area. Integer arrays are similarly stored. Particular emphasis was placed on the input. To facilitate data preparation, integers are read in as real variables. This makes possible to write each datum (keyword, integer or real number) left justified in fields of uniformly 10 characters wide. Administration of external storage goes through special subroutines which keep an account of file contents, current pointer positions, etc. A detailed input manual has been prepared for use with TEXAS. Representative computer times are listed in Table 3.

Acknowledgements. The author is very grateful to Professor J. E. Boggs for his hospitality at the University of Texas at Austin where most of this work has been done. Professor W. Meyer (Kaiserslautern, Germany) contributed a lot of useful ideas. Dr. E. Flood (Tromsø, Norway) has kindly volunteered to implement the localized orbital program from MOLPRO. This research has been supported in part by the U.S. National Science Foundation and the Hungarian Institute of Cultural Relations as part of the program of cooperative research of the University of Texas and the Hungarian Academy of Sciences. It has also been supported in part by a grant from the Robert A. Welch Foundation.

Appendix: A Simple Convergence Acceleration Technique

The essence of this method is to compare the Fock matrices of two consecutive SCF iterations in the intermediate representation, i.e. on the basis of the orbitals used to construct the Fock matrix. SCF convergence means that the Fock matrix becomes diagonal in this representation. Assuming that the orbitals in step k correspond to those in step $k-1$ (i.e. the overlap matrix between the two sets of orbitals is near the unit matrix; see below), an overshoot correction factor may be defined as

$$\xi_{ia} = F_{ia}^{k-1}(F_{ia}^{k-1} - F_{ia}^{k})^{-1} \tag{A1}$$

Here i and a denote occupied and virtual orbitals, respectively. In the special case of a single occupied and a single virtual orbital, the correction factor is approximately

$$\xi_{ia} = (F_{aa} - F_{ii})(F_{aa} - F_{ii} - J_{ia} + 2K_{ia})^{-1} \simeq (\varepsilon_a - \varepsilon_i)(\varepsilon_a - \varepsilon_i - J_{ia} + 2K_{ia}) \tag{A2}$$

where the ε are the orbital energies and J, K denote the usual Coulomb and exchange integrals. This result is easily obtained by comparing the ordinary SCF procedure to a related, quadratically convergent but generally impractical technique: a complete configuration interaction in the space of all singly substituted determinants, followed by rewriting the resulting wave function into single-determinant form to first order accuracy.

The form of (A2) suggests that the correction factor is not strongly dependent on the orbitals. Assuming its constancy, multiplication of the Fock matrix element F_{ia} by ξ_{ia} prior to matrix diagonalization leads to quasi-quadratical SCF convergence in the special case mentioned above. Unfortunately, this is not true generally, due to the neglect of certain coupling terms in the many-dimensional case. Still, the

---

[1] The subroutines for packing and unpacking integers are, however, best transcribed in machine code.

application of correction factors to the occupied-virtual block of the Fock matrix usually results in enhanced convergence.

Practical experience shows that the present method should be used in the following way:
1) The range of the correction factor should be limited: a range between 0.25 and 2 was found adequate.
2) Only those Fock matrix elements should be corrected which are larger, in absolute value, than a specified fraction, say one-tenth, of the largest element in the occupied-virtual block.
3) The method should be used only after initial convergence has been reached. In potential surface calculations this condition is easily met by starting the SCF iteration from the wave function in a neighboring point.

Correct ordering and maximum similarity of the orbitals can be established by subjecting both the occupied and the virtual orbitals of step $k+1$ to unitary transformations U_0 and U_v. These matrices are constructed according to

$$U_0 = A_0(A_0^+ A_0)^{-1/2} \tag{A3}$$

where the positive definite square root is to be taken, and A_0 denotes the overlap matrix between the two sets of occupied orbitals: $(A_0)_{ij} = \langle \varphi_i^{k+1} | \varphi_j^k \rangle$. The transformation of the virtual orbitals is analogous. This transformation gives the maximum overlap sum between the orbitals in step k and those in step $k+1$ [28]. The use of the orbital rotations (A3) is particularly important if there are degenerate orbitals.

The above method can be considered as a generalization of the level-shifting method of Saunders and Hillier [29]. Indeed, their method of increasing the diagonal Fock matrix elements of the virtual orbitals is equivalent, in the sense of perturbation theory, to a correction factor of

$$\xi_{ia} = (\varepsilon_a - \varepsilon_i)(\varepsilon_a + \delta_a - \varepsilon_i)^{-1} \tag{A4}$$

where δ_a is the level shift. As (A4) shows, the level shifting method is less flexible than the correction factor method. Experience shows that the level shifting technique is useful to damp out oscillations in the initial stage of the SCF iteration but it frequently retards convergence in the later stages, at least with positive δ values.

The above technique has been implemented in TEXAS. It was found that it accelerates SCF convergence in most cases, sometimes very significantly. Professor W. Meyer has informed me that he has been using the same method (except the orbital rotation part) since 1973 to converge open-shell (RHF) Hartree–Fock wave functions.

References

1. Pulay, P.: Mol. Phys. **17**, 197 (1969)
2. Pulay, P., in: Modern theoretical chemistry, Vol. 4, p. 153. Schaefer, H. F. III Ed., New York: Plenum Press, 1977

3. Meyer, W., Pulay, P.: MOLPRO Description. München and Stuttgart, Germany 1969
4. Pulay, P.: Mol. Phys. **18**, 473 (1970); **21**, 329 (1971); Pulay, P., Meyer, W.: J. Mol. Spectry. **40**, 59 (1971); Meyer, W., Pulay, P.: J. Chem. Phys. **56**, 2109 (1972); Pulay, P., Meyer, W.: J. Chem. Phys. **57**, 3337 (1972); Meyer, W., Pulay, P.: Theoret. Chim. Acta (Berl.) **32**, 253 (1974); Sawodny, W., Pulay, P.: J. Mol. Spectry. **51**, 135 (1974); Pulay, P., Török, F.: J. Mol. Structure **29**, 239 (1975); Pulay, P., Ruoff, A., Sawodny, W.: Mol. Phys. **30**, 1123 (1975); Molt, K., Sawodny, W., Pulay, P., Fogarasi, G.: Mol. Phys. **32**, 169 (1976); Pulay, P., Meyer, W.: Mol. Phys. **27**, 473 (1974)
5. Botschwina, P.: Chem. Phys. Letters **29**, 98 (1974); **29**, 580 (1974); Bleicher, W., Botschwina, P.: Mol. Phys. **30**, 1029 (1974); Botschwina, P., Pecul, K., Preuss, H.: Z. Naturforsch. **A30**, 1015 (1975); Botschwina, P., Meyer, W., Semkow, A. M.: Chem. Phys. **15**, 25 (1976)
6. Skaarup, S., Boggs, J. E.: J. Mol. Structure **30**, 389 (1976); Schmiedekamp, A., Skaarup, S., Pulay, P., Boggs, J. E.: J. Chem. Phys. **66**, 5769 (1977); Flood, E., Pulay, P., Boggs, J. E.: J. Am. Chem. Soc. **99**, 5570 (1977)
7. Schlegel, H. B., Wolfe, S., Bernardi, F.: J. Chem. Phys. **63**, 3632 (1975); Schlegel, H. B.: Dissertation, Queen's University, Kingston, Ontario, Canada 1975
8. Morokuma, K.: Lecture presented at the Second International Congress on Quantum Chemistry, New Orleans, Louisiana, 1976
9. Huber, H., Čarsky, P., Zahradník, R.: Theoret. Chim. Acta (Berl.) **41**, 217 (1976)
10. Pulay, P., Fogarasi, G., Pang, F., Boggs, J. E.: to be published
11. Hehre, W. J., Stewart, R. F., Pople, J. A.: J. Chem. Phys. **51**, 2657 (1969)
12. Ahlrichs, R.: Theoret. Chim. Acta (Berl.) **33**, 157 (1975)
13. Dupuis, M., Rys, J., King, H. F.: J. Chem. Phys. **65**, 111 (1976)
14. Poshusta, R. D.: Intern. J. Quantum Chem. **12**, 225 (1977)
15. Jerkovich, Gy., Pulay, P.: Intern. J. Quantum Chem., submitted for publication
16. Živković, T., Maksić, Z. B.: J. Chem. Phys. **49**, 3083 (1968)
17. Hehre, W. J., Lathan, W. A., Ditchfield, R., Newton, M. D., Pople, J. A.: GAUSSIAN 70, Program No. 236, Quantum Chemistry Program Exchange, Indiana University, Bloomington, Indiana 1972
18. Spellucci, P., Pulay, P.: Angew. Informatik **17**, 30 (1975)
19. Meyer, W.: Communication at the "First Seminar on Computational Problems in Quantum Chemistry". Strasbourg, France 1969
20. Wilhite, D. L., Euwema, R. N.: J. Chem. Phys. **61**, 375 (1974)
21. King, H. F., Dupuis, M.: J. Comput. Phys. **21**, 144 (1976)
22. Almlöf, J.: University of Stockholm, Institute of Physics Report No. 72–09 (1972)
23. Dacre, P. D.: Chem. Phys. Letters **7**, 47 (1970)
24. Elder, M.: Intern. J. Quantum Chem. **7**, 75 (1973)
25. Dupuis, M., King, H. F.: Intern. J. Quantum Chem. **11**, 613 (1977)
26. Meyer, W., Pulay, P.: Proceedings of the second seminar on computational problems in quantum chemistry, p. 44. Diercksen, G. H., Sutcliffe, B. T., Veillard, A. Eds. München, Germany 1973
27. Pulay, P.: in: The force concept in chemistry, Chapt. 11, Deb, B. M. Ed. Van Nostrand Reinhold Co., New York, to be published
28. Löwdin, P. O.: Advan. Phys. **5**, 1 (1956)
29. Saunders, V. R., Hillier, I. H.: Intern. J. Quantum Chem. **7**, 699 (1973)

Received July 3, 1978