Theoret. Chim. Acta (Berl.) 33, 157—167 (1974)

Methods for Efficient Evaluation of Integrals for Gaussian Type Basis Sets

Reinhart Ahlrichs

Institut für Physikalische Chemie der Universität Karlsruhe, Karlsruhe, Germany

Received November 28, 1973

Methods are described that allow for an efficient evaluation of two-electron integrals over contracted Gaussian lobe functions. The improvement in computational speed is achieved by avoiding the computation of integrals that are: 1. sufficiently small on numerical reasons, 2. zero by symmetry, 3. identical to other integrals by symmetry. Examples of the effectiveness of these techniques are included. We also report the timings for a further processing of two-electron integrals in a Hartree-Fock and correlation energy computation.

Key words: Integrals, evaluation of ~

Introduction

At present most *ab initio* computations on polyatomic molecules are performed in using contracted Gaussians as basis set. The bottleneck of these calculations is usually the evaluation of two-electron integrals. The computer time required for this step can – as is well known – be reduced by the use of symmetry. A further reduction is possible, especially in the case of large molecules, if one takes advantage of the fact that a rather large fraction of the two-electron integrals is negligibly small and needs not be evaluated at all [1]. Clementi [2] has demonstrated that the computer time can be reduced by a factor 100, in the case of the cytosin-guanine complex, if one avoids the evaluation of sufficiently small integrals. This progress has been achieved by the introduction of an "adjoined" basis and an effective testing algorithm which before the actual integral calculation establishes whether a two-electron integral can be approximated by using the adjoined basis or can be neglected entirely [2]. In a more recent publication [3] Clementi notes without further explanation that he has meanwhile abandoned the idea of an adjoined basis, which is, however, used successfully by other workers [4]. We shall comment on this subsequently.

In this paper we give a brief description of some methods developed by the present author which appear to be even more efficient than those described by Clementi [2], at least for medium size basis sets. Our program differs from other existing integral programs mainly in the way in which we avoid the computation of small integrals, whereas other features like the use of symmetry are similar in other existing programs.

In addition to the description of the integral evaluation we point out some problems connected with the use of adjoined basis sets. We also make a few comments on the further processing of the integrals in the SCF-program and in the computation of correlation energies.

Organization

As basis functions φ_I we use linear combinations of Gaussian lobe functions

$$\varphi_I = \sum_i c_i \chi_i, \quad \chi_i(r) = (2\eta_i/\pi)^{3/4} \exp(-\eta_i(r - r_i)^2). \tag{1}$$

The φ_I are also referred to as contracted Gaussians or groups of Gaussians (throughout this paper we lable groups by capitals and lobes by small letters). Basis functions of p-, d-, f-type are then constructed from 2 or more lobes, see e.g. [5, 6]. The use of Gaussian lobes has the advantage that one has to deal with a single type of integral only, which is easily coded in an efficient way.

A two-electron integral

$$(IJ|KL) = \int \varphi_I(r_1)\, \varphi_J(r_1)\, \frac{1}{r_{12}}\, \varphi_K(r_2)\, \varphi_L(r_2)\, d\tau_1\, d\tau_2$$

may be written in the following form:

$$(IJ|KL) = \sum_{kl} t_{kl} \sum_{ij} t_{ij}\, B^{1/2} F(B(r_{ij} - r_{kl})^2), \tag{2}$$

$$t_{ij} = c_i c_j \int \chi_i \chi_j\, d\tau = c_i c_j (4\eta_i \eta_j)^{3/4} (\eta_i + \eta_j)^{-3/2} \exp\left(-\frac{\eta_i \eta_j}{\eta_i + \eta_j}(r_i - r_j)^2\right), \tag{3}$$

$$r_{ij} = \frac{\eta_i r_i + \eta_j r_j}{\eta_i + \eta_j}, \tag{4}$$

$$B = (q_{ij} + q_{kl})^{-1}, \tag{5}$$

$$q_{ij} = (\eta_i + \eta_j)^{-1}, \tag{5a}$$

$$F(x) = x^{-1/2} \operatorname{erf}(x^{1/2}). \tag{6}$$

where erf(x) denotes the error function.

The integrals are computed in an order which to the author's knowledge has first been proposed by Meyer [7]. The loops run over $I \geqq J \geqq K \geqq L$, and for a quadruple $(IJKL)$ one then computes the integrals $(IJ|KL)$, $(IK|JL)$ if $J > K$, $(IL|JK)$ if $I > J$ and $K > L$. If the integrals are obtained in this order it is an easy matter to construct the Roothaan supermatrices [8] $\mathscr{P}$, $\mathscr{J}$, or $\mathscr{K}$, as is shown below. (For the subsequent considerations it is not of too great importance in which order the integrals are obtained.)

It is obvious from Eq. (2) that a fast integral evaluation would be extremely easy if the quantities t_{ij}, r_{ij}, and q_{ij} could be kept in storage for all i and j. The storage requirement for such a procedure is prohibitive, however. Most time-consuming is the evaluation of the t_{ij}. To avoid a permanent recomputation of these quantities, we devide the t_{ij} into blocks of appropriate length and keep only those blocks in storage which are currently needed. The corresponding blocks are recomputed whenever they are needed. (The quantities v_{ij} and V_{IJ} to be defined below are treated in the same way.) The core storage available for the integral routine is provided as an input parameter and a dynamic storage allocation is performed to make efficient use of the core storage.

Let us now consider the quantities r_{ij}, r_{kl}, q_{ij}, and q_{kl} needed to evaluate $(IJ|KL)$. (The computation of $(IK|JL)$ and $(IL|JK)$ is analogous.) After I and J are specified we get and store the r_{ij} and q_{ij} for $i \in I$ and $j \in J$. For the actual integral evaluation we first start the loops over k, l and get the r_{kl}, q_{kl} after k and l are specified. The innermost loop then runs over the combined index (ij). Proceeding this way, we have available all quantities occuring in (2) before the innermost loop (over ij) starts and no redundant computations have to be performed in that loop. The integrals $(IJ|KL)$, $(IK|JL)$, and $(IL|JK)$ are kept on tape together with the lable $(IJKL)$ provided at least one of the integrals is larger than the cut off threshold T, as is outlined below.

Test on the Magnitude of the Integrals

It has already been pointed out by several authors that a rather large fraction of the integrals is essentially zero in the case of large molecules [1, 2]. In order to avoid the unneccessary computation of small integrals $(IJ|KL)$ and the computation of small contributions to the larger integrals an effective testing algorithm is of great importance.

First, we note that $F(x)$ is a slowly varying function of x, for this reason it does not pay to perform a test on x or $F(x)$. Next, we observe that

$$B=(q_{ij}+q_{kl})^{-1} \leqq (4q_{ij}q_{kl})^{-1/2}.$$

If we then introduce the quantities

$$u_{ij}=|t_{ij}| \cdot q_{ij}^{-1/4}, \tag{7}$$

we obtain the following rigorous estimate for the individual contributions to $(IJ|KL)$, see Eq. (2) (note that $F(x) \leqq 2\pi^{-1/2}$):

$$|t_{ij}t_{kl}B^{1/2}F(B(r_{ij}-r_{kl})^2)| \leqq (2/\pi)^{+1/2} u_{ij}u_{kl}. \tag{8}$$

This estimate cannot be improved in general since the equality sign holds if $q_{ij}=q_{kl}$ and $r_{ij}=r_{kl}$.

In our program we avoid the unnecessary computation of small two-electron integrals in neglecting consistently all contributions which are smaller than a certain threshold T, i.e. we neglect all terms on the r.h.s. of Eq. (2) for which $u_{ij}u_{kl} < T$

It is more convenient, however, to consider the logarithms of the corresponding quantities:

$$v_{ij}=\ln(u_{ij}) \tag{9}$$

$$V_{IJ}=\max_{ij} v_{ij} \tag{10}$$

$$VM=\max_{IJ} V_{IJ} \tag{11}$$

$$IT=\ln(T). \tag{12}$$

The quantities defined in (9)–(12) may now be truncated to integers, which can be stored in half words.

With the aid of the quantities v_{ij}, V_{IJ}, and VM it is then possible before the actual integral computations to establish whether or not one has to expect "non zero" contributions at all.

If

$$V_{IJ}+VM<IT \tag{13}$$

one can skip the evaluation of the integrals $(IJ|K'L')$ for arbitrary K', L', since no contribution larger than T occurs in any of these integrals.

If

$$V_{IJ}+V_{KL}<IT \tag{14}$$

the integral $(IJ|KL)$ is set equal to zero.

If

$$v_{kl}+V_{IJ}<IT \tag{15}$$

one can skip for the current kl the whole sum over ij in the actual computation of $(IJ|KL)$, see Eq. (2).

Finally, if

$$v_{kl}+v_{ij}<IT$$

one neglects the corresponding contribution $(ij|kl)$ to $(IJ|KL)$. We thus see that the use of the quantities v_{ij}, V_{IJ} and VM enables us to check at any stage of the integral computation whether or not a subsequent loop has to be entered at all.

The appropriate value for the threshold T has to be determined by computer experiments. The dependence of E_{SCF} on the cut off threshold T is shown in Table 1 for a series of C_2H_4 computations.

These data show, that ΔE_{SCF} – the error of the SCF energy which is due to the neglect of small integrals – is always smaller than $10\cdot T$

Table 1. Dependence of the SCF energy of C_2H_4 (exp geometry) on the integral cut off threshold T. Basis set: C(8,4/4,2), H: (4/2)

$\log T$	-8.1	-7.2	-6.4	-5.5	-4.7
E_{SCF}	-77.994466	-77.994466	-77.994465	-77.994455	-77.994361

Number of Integrals to be Computed in Large Molecules

The basic idea of the above checking algorithm can be used to estimate the total number of two-electron integrals to be actually computed in large molecules. For this purpose we consider a series of computations for molecules A_n with an arbitrary but fixed basis set for the fragments A.

The only requirement on the structure of A_n is that the distance between any two fragments A_i and A_j exceeds an arbitrary but fixed distance r_0.

If we now want to compute only those integrals that are larger than a certain threshold T, we have to consider only the charge distributions $\chi_i \chi_j$ for which

$$v_{ij} + VM > IT. \tag{16}$$

The number of pairs (ij) of Gaussian lobes which fulfill (16) increases only linearly with n (the number of fragments) for sufficiently large n. This follows immediately from the definition of v_{ij}, see (9)–(11), and from the above requirement on the structure of A_n (note that VM does not depend on n, since we keep the basis for A fixed). As the number of pairs of lobes $\chi_i \chi_j$ for which (16) holds increases only linearly with n, we conclude that the number of two-electron integrals larger than T increases only like n^2. We note that the above reasoning holds also for Slater type basis functions.

A similar analysis has recently been published by Dyczmons [9]. He showed (under essentially the same conditions as above) that the computer time increases like $n^2 (\ln n)^2$ if one neglects small integrals in such a way that the sum of absolute values of neglected integrals remains smaller than T. We note without proof that, for the case investigated by Dyczmons, the number of two electron integrals increases in fact only like $n^2 \ln n$ and not like $n^2 (\ln n)^2$.

A Problem Connected with the Use of "Adjoined" Basis Sets

Let us consider an integral $(SaSb|ScSc)$ where Sa, Sb, Sc are identical S groups at the respective centers a, b, c. Let r_{ab} denote the distance between centers a and b. We now want to approximate $(SaSb|ScSc)$ by a corresponding integral over adjoined orbitals $\tilde{S}$.

$$(SaSb|ScSc) \approx (\tilde{S}a\tilde{S}b|\tilde{S}c\tilde{S}c). \tag{17}$$

It is then easily verified by inspection that the correct dependence of $(SaSb|ScSc)$ on r_{ab} can only be reproduced if $\tilde{S}$ contains the term $c_1\chi_1$, where χ_1 is the lobe with smallest orbital exponent η occuring in the group and c_1 the corresponding contraction coefficient.

As the integral contains the charge distribution $(Sc)^2$ one has also to require that S has approximately the same norm as $\tilde{S}$: $\|S\| \approx \|\tilde{S}\|$. In order that (17) is in error by not more than 10% we thus have to put in general

$$\tilde{S} = c_1 \chi_1 + c\chi, \tag{18}$$

where c and χ may be chosen to maximise $|\langle S|\tilde{S}\rangle|$ under the constraint $\|S\| = \|\tilde{S}\|$.

The above consideration shows conclusively that two primitive Gaussians are needed to construct the adjoined orbitals. If one wants to use a single primitive Gaussian only, one must choose $\tilde{S} = c_1\chi_1$, but then – for the case under consideration – one can replace only Sa and Sb by $\tilde{S}a$ and $\tilde{S}$b but not Sc by $\tilde{S}c$, i.e. one can only use

$$(SaSb|ScSc) \approx (\tilde{S}a\tilde{S}b|ScSc).$$

It is probably for the reasons just discussed that Clementi has abandoned the idea of the adjoined basis set and just neglects integrals smaller than a certain threshold [3].

Evaluation of $F(x)$

For a fast evaluation of the function $F(x)$ defined in Eq. (6) we use a rational Tchebycheff approximation in small intervalls

$$F(x) \approx \tilde{F}(x) = (a_i + b_i x + c_i x^2)/(d_i + x), \qquad x_i \leqq x \leqq x_{i+1} . \tag{19}$$

As the accuracy of (19) varies considerably with the location of the intervall (x_i, x_{i+1}), it is advantageous to use a variable intervall length. We have put $x_i = Dj_i$, i.e. $x_{i+1} - x_i = D(j_{i+1} - j_i)$ is always a multiple of a basic intervall D. The parameters j_i are determined in the following way: for a given j_i (starting with $j_1 = 0$) we choose j_{i+1} as large as possible under the provision that (19) has a prescribed accuracy. This procedure is continued until $\tilde{F}(x) = x^{-1/2}$ is sufficiently accurate.

In order to evaluate $F(x)$ we have to store besides the coefficients $a_i - d_i$ an integer vector $i(j)$ which is needed to recover i for the actual $j = [x/D] + 1$.

If one requires an accuracy of $5 \cdot 10^{-13}$ for (19) one needs 1688 basic intervalls of length $D = 0.0148$ and 387 intervalls (x_i, x_{i+1}), reducing the accuracy to $5 \cdot 10^{-10}$ the corresponding numbers are 533 and 127.

Use of Symmetry

a) Parity Check

A set of idempotent symmetry operations (reflections, inversions, 180° rotations etc.) are considered, which are not required to belong to the symmetry group of the molecule. It is then established for each group of Gaussians whether it has even parity, odd parity or no parity at all with respect to each of these symmetry operations. After I, J, K, L are specified in the integral routine we check whether or not $\varphi_I \cdot \varphi_J \cdot \varphi_K \cdot \varphi_L$ has odd parity with respect to one of the symmetry operations. If this is the case all three integrals $(IJ|KL)$, $(IK|JL)$ and $(IL|JK)$ vanish and their computation is skipped.

In order to make efficient use of this program feature it is important to choose the basis set (e.g. the direction of p-orbitals) in an appropriate manner.

b) Redundancy of Integral Values

Symmetry is used further to avoid the redundant computation of integrals over groups which are equal in value (possibly with a factor -1) on symmetry grounds. In the present program version we consider just one two fold symmetry operation. This feature can also be used if a symmetry operation maps just a part of the total basis set into itself.

Let I', J', K', L' denote the lables into which I, J, K, L are mapped by the symmetry operation. The integrals $(IJ|KL)$, $(IK|JL)$ and $(IL|JK)$ are computed only if $(IJKL) > (I'J'K'L')$, i.e. if $IJKL$ succeeds $I'J'K'L'$ in the loops. The three integrals corresponding to I', J', K', L' (which are essentially identical to chose obtained for I, J, K, L) are also kept together with their label $I'J'K'L'$. In principle it is not necessary to keep the redundant integrals, but in the treatment of correlation energies we have to deal with density matrices which are not totally symmetric with respect to molecular symmetry and this case is handled in a more convenient way if all non zero integrals are kept on tape.

The check whether or not $(IJKL) > (I'J'K'L')$ can already be performed partly in the outer loops, i.e. before I, J, K, and L are specified. If $I' > I$, it follows that $(I'J'K'L') > (IJKL)$ and no integrals are computed for the current I. We can thus skip the entire loops over J, K, L etc.

Integral Computation Times

In Table 2 we present some typical computational times for the evaluation of two-electron integrals. The program is written in FORTRAN V for the Univac 1108 and uses double precision floating point arithmetic (72 bit) throughout. A single precision version would require about 60% of the time given in the table.

For a comparison with the timings given by Clementi [2] we make the following comments:

a) The IBM 360/195 is about 12–20 times faster than the Univac 1108. A comparison of different computers is nevertheless rather problematic.

b) The programs compared in Table 2 use different methods to avoid the computation of small integrals. As the integral time may depend crucially on the cut off threshold, a comparison of different programs is meaningfull only if the integrals are obtained with comparable accuracy. For the computations reported in Table 2 we employed a cut off threshold $T = 10^{-7}$, which in all cases inves-

Table 2. Examples of integral computation times

Molecule	Basis	Number of		Storage[a]	Geometry	Time[b]	Other work
		lobes	groups				

C_2H_2	C(7,3,1|4,2,1)
	H(3,1|2,1)
		106	40	50K	exp	6.4	—

C_2H_4	C(7,3,1|4,2,1)
	H(3,1|2,1)
		124	50	50K	exp	22.4	–

$Na^+(H_2O)_2$	Na,O(7,3|2,1)	87	19	50K	linear[c]	1.5	7 (0.66)[d]
$Na^+(H_2O)_4$	Na,O(7,3|2,1)	149	33	50K	tetrah.[c]	9.7	45 (3.75)[d]
$Na^+(H_2O)_4$	Na,O(7,3|2,1)	149	33	50K	planar[c]	7.0	
$Na^+(H_2O)_6$	Na,O(7,3|2,1)	211	47	65K	octah.[c]	24.5	129 (10.75)[d]
C_4NH_5 (Pyrrole)	C,N(7,3|4,2) H(3|2)	140	60	50K	exp	31.0	26 (15.7)[e]
N_2O_4	N, O(8,4|4,2)	192	60	50K	exp	55.0	—

N_2O_4	N(8,4,1|4,2,1)
	O(8,4|4.2)
		230	70	65K	exp	127.0	—

$C_4N_3OH_5$ (Cytosin)	C,N,O(7,3|2,1) H(3|1)	215	45	65K	exp	174	212 (16.9)[d]

[a] In 36 bit words.

[b] UNIVAC 1108 CPU-times, double precision floating point arithmetic is used throughout. The cut off threshold, see text, was set at $T = 10^{-7}$. All times are given in minutes.

[c] An Na–O distance of 5.25 a.u. was used. The terms linear, tetrahedral, planar and octahedral refer to the geometrical arrangement of the oxygen atoms surrounding the Na^+ ion.

[d] Ref. [2], for a better comparison we have multiplied the original IBM 360|195 times, which are given in paranthesis, by a conversion factor 12.

[e] Ref. [10], the original times, given in paranthesis, are multiplied by a single precision – double precision conversion factor 1.66.

tigated guaranteed that E_{SCF} is in error by 2.10^{-6} a.u. at most. Clementi claims the same accuracy for his program [2].

c) In the present program we have used one symmetry operation to avoid the computation of redundant integrals whereas Clementi does not. If one wishes to correct for this fact one should multiply the times in column 7 by an average factor of 1.7, except for the cytosin molecule where no redundancy on symmetry grounds occurs, as far as integrals over groups are concerned. (We do not take advantage of the fact that integrals over lobes are identical by symmetry.)

Our pyrrole computation is compared with that of Almlöf [10]. Almlöf's program combines some features of the program developed by Hehre *et al.* [11] with a sophisticated handling of molecular symmetry and an efficient check on the magnitude of integrals to avoid the computation of small integrals. Almlöf does not discuss the accuracy of his program, but since he uses single precision arithmetic the absolute accuracy is 10^{-6} at most.

As can be seen from Table 2, the integral time is not just a function of the number of Gaussian lobes involved but depends strongly on the interatomic distances, the orbital exponents η and on molecular and local symmetries since all these facts have a marked influence on the number of integrals to be actually evaluated.

Further Processing of Two-Electron Integrals

The further processing of two-electron integrals in the SCF or correlation energy computation is always based on the use of the corresponding Roothaan supermatrix [8]. It is the main advantage of the idea of Meyer (to obtain the integrals in the specific order described above) that the various supermatrices can be computed in an easy way, since the matrix elements with lables (IJ, KL), (IK, JL), and (IL, JK) of any of the supermatrices are just linear combinations of the corresponding three integrals.

For the closed shell SCF computation we first construct the $\mathscr{P}$-matrix. From the three integrals $(IJ|KL)$, $(IK|JL)$ and $(IL|JK)$ we get the corresponding $\mathscr{P}$-matrix elements according to

$$\mathscr{P}_{IJ,KL} = [4(IJ|KL) - (IK|JL) - (IL|JK)]\,(1 - 1/2\,\delta_{IJ,KL}),$$

if $J > K$:

$$\mathscr{P}_{IK,JL} = [4(IK|JL) - (IJ|KL) - (IL|JK)]\,(1 - 1/2\,\delta_{IK,JL}),$$

if $I > J$ and $K > L$:

$$\mathscr{P}_{IL,JK} = [4(IL|JK) - (IJ|KL) - (IK|JL)]\,(1 - 1/2\,\delta_{IL,JK}),$$

where $\delta_{IJ,KL}$ is unity if $I = K$ and $J = L$ and zero otherwise (note that always $I \geqq J \geqq K \geqq L$).

We keep only those elements of the $\mathscr{P}$-matrix which are larger in absolute value than the cut off threshold T introduced above. The combined indices $(IJ) = I \cdot (I-1)/2 + J$ and (KL) are stored on tape or disc together with the corresponding supermatrix element.

Table 3. Typical CPU-times for further processing of integrals

Molecule[c]	Number of	Storage	CPU-time[a] for		Average pair	Comment
	groups	(36 bit words)	$\mathscr{P}$-matrix	one SCF-iteration	correlation[b]	
C_2H_2	40	50K	0.1	0.1	0.9	
C_2H_4	50	50K	0.5	0.4	3.0	
$N_a^+(H_2O)_4$	33	50K	0.08	0.07	—	tetrahedral
$N_a^+(H_2O)_4$	33	50K	0.06	0.06	—	planar
$N_a^+(H_2O)_6$	47	65K	0.26	0.22	—	
N_2O_4	60	50K	1.0	0.75	6.0	
N_2O_4	70	65K	1.6	1.2	11.0	
Cytosin	45	65K	0.34	0.28	–	
Pyrrole	60	50K	1.2	0.9	–	

[a] Times are given in minutes and refer to the UNIVAC 1108, double precesion floating point arithmetic is used.

[b] See text.

[c] See also Table 2.

The construction of the Fock operator F is now extremely easy since any $\mathscr{P}$-matrix element gives just two contributions to the Fock operator F:

$$\mathscr{P}_{IJ,KL} \cdot D(KL) \quad \text{to} \quad F(IJ)$$

and

$$\mathscr{P}_{IJ,KL} \cdot D(IJ) \quad \text{to} \quad F(KL),$$

where D denotes the density matrix stored in triangular form which, in an obvious notation, is defined as

$$D(IJ) = \sum_M C_{IM} C_{JM} (1 - 1/2\, \delta_{I,J}).$$

In Table 3 we collect for some typical examples the CPU time required to construct the $\mathscr{P}$-matrix from the integral tape.

Timings for the construction of the $\mathscr{J}$- or $\mathscr{K}$-matrix are virtually identical, of course. We further give the time per SCF iteration which is essentially the time required to get the F-matrix. It should be noted that we did not use symmetry properties to speed up the computation of the supermatrices or the Fock operator.

The program allows also for the computation of correlation energies within the IEPA-PNO method (IEPA = Independent Electron Pair Approximation, PNO = Pair Natural Orbitals) described in Ref. [12], for a recent review see [13]. The computation of the pair correlation functions and energies is based on the direct calculation of the PNO's of the corresponding two-electron function, which guarantees optimum convergence properties of the CI expansion. The direct computation of the PNO's, i.e. prior to the knowledge of the pair function, can be performed according to a method proposed by Kutzelnigg [14]. Recently even faster and more accurate procedures have been developed for this purpose [15, 16].

In order to perform the IEPA-PNO computation we construct successively the $\mathscr{J}$-, $\mathscr{K}^+$-, and $\mathscr{K}^-$-supermatrices to obtain the corresponding Coulomb and exchange operators required, see [12]. As the details of the integral handling is analogous to the one explained for the $\mathscr{P}$-supermatrix and the SCF-computation a further explanation is not necessary. In Table 3 we give the typical CPU-time required to obtain one pair correlation function and the corresponding pair correlation energy with the present program. The number of PNO's included (10–25) is always chosen large enough to account for at least 98% of the corresponding correlation energy contributions obtainable within the respective basis set. We note that between 7% and 20% of the times reported are used to obtain the PNO's, the remainder is required to set up the pair CI-matrix.

The IEPA does not obey the variational principle. We note, however, that a variational treatment following the IEPA, like e.g. a PNO-CI first proposed by Meyer [15], requires little additional computertime [17], since all tedious CI-matrix elements have already been computed in the IEPA-part.

Discussion

We have shown in the present paper that the evaluation of two-electron integrals over contracted Gaussian lobe functions of medium size basis sets (up to about 300 lobes contracted to up to 80 groups) can be performed within reasonable computer times with rather modest storage requirements (50–65 K 36-bit words for the double precision version of the program) with a rather simple program. We have further demonstrated that the additional steps of an ab initio computation – like SCF iterations, computation of correlation energies – usually require only a rather small fraction of the CPU time necessary to obtain the two-electron integrals.

Finally, let us discuss some features of the present program in connection with some other programs described in the literature. The basic structure of the integral routine is a straight-forward coding of Eq. (2). The performance is increased by a very simple but efficient checking system which determines at every stage whether or not subsequent loops may be skipped.

Three kinds of tests are made for this purpose:

a) parity check to avoid the computation of zeros,

b) symmetry check to avoid the computation of redundant integrals over groups,

c) check on the magnitude of integrals to avoid the evaluation of small contributions to the corresponding integrals.

The realization of the latter point as described above is an essentially novel feature of the present program. Symmetry is exploited in a much more sophisticated way in other programs, like Polyatom [18], Reflect [19] or Almlöf's program [10]. These programs should thus be faster than the present one for highly symmetric molecules.

We do not take advantage of symmetry in the construction of the supermatrices and in the construction of Fock, Coulomb or exchange operators from the supermatrices. The corresponding CPU times are relatively small, however, usually 1–2% of the integral computation time, see Tables 2, 3. The total CPU time can thus not be decreased significantly by further exploitation of symmetry. If the correlation energy computation is based on localized SCF MO's rather than on the canonical ones, one has to deal with density matrices which are not invariant under symmetry operations and symmetry is of little or no help in this case anyway as far as the processing of integrals is concerned.

If the localized valence shell SCF MO's are equivalent on symmetry grounds, as e.g. in CH_4, one can, of course, exploit this equivalency to simplify the computation of correlation energy [17].

Acknowledgements. The author is indebted to F. Driessler and Dr. H. Lischka for their valuable assistence in developing the present computer program. I further thank F. Keil and Dr. B. Zurawski for valuable discussions and for performing some of the computations discussed in the present paper.

The computations were performed at the „Rechenzentrum der Universität Karlsruhe", the assistance of the staff of the Rechenzentrum in performing the computations is greatfully acknowledged.

References

1. Christoffersen, R. E.: Advan. Quantum Chem. **6**, 333 (1972)
2. Clementi, E.: Proc. Nat. Acad. Sci. USA **69**, 2942 (1972)
3. Clementi, E., Kistenmacher, H., Popkie, H.: J. Chem. Phys. **58**, 4699 (1973)
4. Ady, E., Brickmann, J.: private communication
5. Whitten, J. L.: J. Chem. Phys. **39**, 349 (1963)
6. Driessler, F., Ahlrichs, R.: Chem. Phys. Letters (in press)
7. Meyer, W.: Private communication at the "First seminar on computational problems in quantum chemistry", Strasbourg 1969
8. Roothaan, C. C. J.: Rev. Mod. Phys. **32**, 179 (1960)
9. Dyczmons, V.: Theoret. Chim. Acta (Berl.) **28**, 307 (1973)
10. Almlöf, J.: Proceedings of the second seminar on computational problems in quantum chemistry Strasbourg 1972, printed at Max-Planck-Institut für Physik und Astrophysik, München 1973
11. Hehre, W. J., Stewart, R. F., Pople, J. A.: J. Chem. Phys. **51**, 2657 (1969)
12. Jungen, M., Ahlrichs, R.: Theoret. Chim. Acta (Berl.) **17**, 339 (1970)
13. Kutzelnigg, W.: Topics Current Chem. **41**, 31 (1973)
14. Kutzelnigg, W.: Theoret. Chim. Acta (Berl.) **1**, 327 (1963)
15. Meyer, W.: J. Chem. Phys. **58**, 1017 (1973)
16. Ahlrichs, R.: to be published
17. Ahlrichs, R., Kutzelnigg, W., Lischka, H., Staemmler, V.: to be published
18. Csizmadia, I. G., Harrison, M. C., Moskowitz, J. W., Sutcliffe, B. T.: Theoret. Chim. Acta **6**, 191 (1966)
19. Siegbahn, P.: Chem. Phys. Letters **8**, 245 (1971)

Dr. R. Ahlrichs
Institut für Physikalische Chemie
der Universität
D-7500 Karlsruhe
Federal Republic of Germany