Theoret. Chim. Acta (Berl.) 33, 1—6 (1974)
© by Springer-Verlag 1974

Commentationes

Optimized Transformation of Four Center Integrals

Geerd H. F. Diercksen

Max-Planck-Institut für Physik und Astrophysik, 8000 München 40, Germany

Received August 10, 1973

The n^5-order algorithm for the transformation of the quantummechanical four center integrals is analyzed. An optimum implementation of this algorithm for computers with a reasonable amount of direct-access storage is described.

Key words: Four center integrals – Integral transformation

It is a commonly used approach in quantum chemistry, as well as in many other fields of physics, to express mathematical functions $i, j, k, l, \ldots$ as linear combinations of another set of functions $p, q, r, s, \ldots$

$$i = \sum_{p=1}^{n_p} c_{ip}\, p\,, \tag{1}$$

usually in order to allow, or to simplify certain mathematical operations to be performed. In quantum chemistry, examples of such functions i are symmetry orbitals, self-consistent-field orbitals, and correlation orbitals. In this case, the functions r are usually exponential – or gaussian type atomic functions, normally called basis functions.

In many quantum chemical applications the computation of integrals among the functions $a, b, c, d, \ldots$ of one- and two electron operators $O(1)$, $O(1, 2)$ of the following type are necessary

$$(a|b) = \int a(1)\, O(1)\, b(1)\, d\tau_1\,, \tag{2.1}$$

$$(ab|cd) = \int a(1)\, b(1)\, O(1, 2)\, c(2)\, d(2)\, d\tau_1\, d\tau_2\,. \tag{2.2}$$

The integrals over the different sets of functions related by (1) can be calculated according to the following formulas if the appropriate integrals over one set of functions are available:

$$(i|j) = \sum_{p,q} c_{ip}\, c_{jq} (p|q)\,, \tag{3.1}$$

$$(ij|kl) = \sum_{p,q,r,s} c_{ip}\, c_{jq}\, c_{kr}\, c_{ls} (pq|rs)\,. \tag{3.2}$$

Transformations of this form are necessary in all quantum chemical calculations beyond the Hartree-Fock level. The transformation (3.1) of the one electron integrals is trivial for (nearly) any range of indices and will not be discussed here. It involves $3/2\, n^3$ ($n_i = n_j = n_p = n_q = n$) multiplications and additions to compute a lower triangle of the full matrix.

The transformation of the two-electron integrals (3.2) is much more time consuming and has been a stumbling-block for all calculations beyond the Hartree-Fock level which use a reasonable size of basis set. It is obvious that the transformation (3.2) has to be done stepwise by forming four partial sums, and then is of the n^5 order in multiplications and additions. Actually, the big problem of this transformation has never been the unawareness of the n^5-algorithm, although it has only been stated in the literature as late as 1970 [1], but rather that it has not been possible to implement this algorithm on the previously available computers. Most approaches used today [2, 3] are based on an algorithm proposed in 1963 [4] which is of the order n^6 in multiplications and additions and which is especially suitable for tape-oriented (*sequential* external storage) computers. An economic implementation of the n^5-algorithm has only become possible after an efficient reordering algorithm for four-indexed quantities has been proposed, which is based on the use of large size *random access* external storage of the order of 10^6 floating point words [5]. An efficient four-index transformation based on this reordering procedure has implicitly been sketched [6]. An approach based on the n^5 algorithm has recently been described, but seems inapplicable for any set of functions of adequate size because of severe storage problems [7]. In the following an implementation of the n^5 algorithm will be described, which is based on the four indexed quantities reordering algorithm referenced [5], and which has been simultaniously and independently developed to a very similar approach outlined previously [6]. The present algorithm has been programmed and fully tested up to 70 functions and gives encouraging results.

Theory

The transformation (3.2) can be split up into the following four partial sums:

$$(iq|rs) = \sum_p c_{ip}(pq|rs)\,, \tag{4.1}$$

$$(ij|rs) = \sum_q c_{jq}(iq|rs)\,, \tag{4.2}$$

$$(ij|ks) = \sum_r c_{kr}(ij|rs)\,, \tag{4.3}$$

$$(ij|kl) = \sum_s c_{ls}(ij|ks)\,. \tag{4.4}$$

The range of the indices $i, j, k, \ldots$ is assumed to be $1, 2, \ldots n_i$; $1, 2, \ldots n_j$; etc. The above partial sums (4) then will involve the following number of multiplications and additions (*ma*):

$$ma\,11 = n_i \cdot n_p \cdot n_q \cdot n_r \cdot n_s\,, \tag{5.1}$$

$$ma\,12 = n_i \cdot n_j \cdot n_q \cdot n_r \cdot n_s\,, \tag{5.2}$$

$$ma\,13 = n_i \cdot n_j \cdot n_k \cdot n_r \cdot n_s\,, \tag{5.3}$$

$$ma\,14 = n_i \cdot n_j \cdot n_k \cdot n_l \cdot n_s\,. \tag{5.4}$$

Forming these partial sums is obviously an n^5-order process in multiplication and addition. The number of *ma*'s is minimized, if the summation over the different indices is done in such an order, that the following conditions hold:

$$n_i \leqq n_j \leqq n_k \leqq n_l\,, \tag{6.1}$$

$$n_p \geqq n_q \geqq n_r \geqq n_s\,. \tag{6.2}$$

In particular, for the full transformation, defined by the condition

$$n = n_i = n_j = \cdots = n_r = n_s \tag{7}$$

there will be a total of

$$maf = 4\,n^5 \tag{8}$$

multiplications and additions. The number of multiplications and additions can be reduced, if use is made of the symmetry of the integrals $(ab|cd)$

$$\begin{aligned}(ab|cd) &= (ab|dc) = (ba|cd) = (ba|dc)\\ &= (cd|ab) = (cd|ba) = (dc|ab) = (dc|ba)\,.\end{aligned} \tag{9}$$

If the functions a, b, c, and d belong to the same set, and if the indices run over the identical range, n_a, then the list of the integrals which contains only one contribution for each set of integrals identical because of the relation (9), is defined by the following condition on the indices:

$$\begin{aligned} &b \leqq a\,,\\ &c \leqq a\,,\\ &d \leqq c\,, \quad \text{if} \quad c < a\,,\\ &d \leqq b\,, \quad \text{if} \quad c = a\,. \end{aligned} \tag{10}$$

In this case the list contains

$$nl = 1/2\,n_a \cdot (n_a + 1) \cdot 1/2(n_a \cdot (n_a + 1)/2 + 1) \sim 1/8\,n_a^4 \tag{11}$$

two-electron integrals. Making full use of this internal symmetry for the integrals to be *computed*, with the index range n_i, but summing over the full range of indices of the basic integrals, n_p, the partial sums give rise to the following number of multiplications and additions:

$$ma\,21 = n_i \cdot n_p \cdot n_p \cdot 1/2 \cdot n_p \cdot (n_p + 1)\,, \tag{12.1}$$

$$ma\,22 = 1/2\,n_i \cdot (n_i + 1) \cdot n_p \cdot 1/2 \cdot n_p \cdot (n_p + 1)\,, \tag{12.2}$$

$$ma\,23 = 1/2\,n_i \cdot (n_i + 1) \cdot n_i \cdot n_p \cdot n_p\,, \tag{12.3}$$

$$ma\,24 = 1/2\,n_i \cdot (n_i + 1) \cdot 1/2 \cdot (1/2 \cdot n_i \cdot (n_i + 1) + 1) \cdot n_p\,. \tag{12.4}$$

The total number of *ma*'s for step 1 to 4 is then calculated to be

$$\begin{aligned} may = 1/2 \cdot (&n_p^2 \cdot (n_p + 1) \cdot n_i \cdot (n_p + 1/2 \cdot (n_i + 1))\\ &+ n_p \cdot (n_i + 1) \cdot n_i \cdot (n_i \cdot n_p + 1/4 \cdot n_i(n_i + 1) + 1/2))\,. \end{aligned} \tag{13}$$

For the full transformation defined by (7) the number of multiplications and additions is computed to be

$$may f = 11/8\, n^5 + 7/4\, n^4 + 5/8\, n^3 + 1/4\, n^2\,, \tag{14}$$

and for large values of n, the number of multiplications and additions will be approximately

$$may f \sim 3/2\, n^5\,. \tag{14'}$$

If the functions $i, j, \ldots$ belong to different irreducible symmetry representations, say $\alpha, \beta, \ldots$, then only the following types of integrals have to be computed

$$(\alpha\alpha|\alpha\alpha)\,, \quad (\alpha\alpha|\beta\beta)\,, \quad (\alpha\beta|\alpha\beta)\,, \tag{15}$$

all other integrals are identical to zero because of symmetry. For the integrals of type $(\alpha\alpha|\alpha\alpha)$ the number of multiplications and additions is given by formula (13), with $n_\alpha = n_p$. For the integrals of type $(\alpha\alpha|\beta\beta)$, and $(\alpha\beta|\alpha\beta)$ the number of multiplications and additions is easily computed to be

$$\begin{aligned} ma\alpha\alpha\beta\beta = {} & 1/2\, n_\alpha \cdot n_p \cdot (n_p + 1/2 \cdot (n_\alpha + 1)) \cdot n_r(n_r + 1) \\ & + 1/2\, n_\alpha \cdot (n_\alpha + 1)\, n_\beta \cdot n_r \cdot (n_r + 1/2 \cdot (n_\beta + 1))\,, \end{aligned} \tag{16}$$

$$ma\alpha\beta\alpha\beta = n_\alpha \cdot n_p(n_r \cdot n_p(n_r + n_\beta) + n_\beta \cdot n_\alpha(n_r + 1/2\, n_\beta(n_\beta + 1)))\,. \tag{17}$$

In all cases where the set of functions $p, q, r, \ldots$ consist of symmetry functions, n_p and n_r will run only over subranges of n, namely over the range of the appropriate symmetry representation.

By comparing the formula (8) with the formulae (11), (16), and (17) it becomes clear that the number of multiplications and additions can be considerably reduced, if use is made of the symmetry properties of both, or of one of the sets of functions involved. As a rough estimate, the transformation becomes a $(\max)^5$-order algorithm, where max is the largest index range involved.

Implementation

It has been assumed that a transformation matrix C and a list of integrals $(pq|rs)$, in the standard order defined by (10), is given as input, and that a list of integrals $(ij|kl)$ in standard order, has to be generated.

The implementation of the n^5-order algorithm to be developed is aimed at handling transformations involving index ranges of 50 or more ($n \sim 50$). Therefore all effort has been concentrated on finding the algorithm that needs a minimum amount of processor storage and of input/output operations. The algorithm developed consists essentially of a two-step process, applied once to the transformation $(pq|rs) \rightarrow (ij|rs)$, and once to the transformation $(ij|rs) \rightarrow (ij|kl)$. In this section, the present implementation of the algorithm is described. Symmetry is not taken into account explicitly, but the algorithm can be applied to each block of a symmetry partitioned transformation with minor modifications.

In the first, the reordering step of this two-step procedure, the standard integral list $(pq|rs)$ is read into processor storage, one record at a time. This standard list is expanded by forming *all integrals*, that fulfill the condition $p \geqq q$ and $r \geqq s$, by making use of the internal symmetry relation (9) between these integrals. Finally, the integrals are ordered into sublists. A sublist contains *all* integrals for the full index range p, and $q(p \geqq q)$ and for some index range r, and $s(r \geqq s)$. The index range of r, and s is determined such that the full sublist can be held in processor storage in the second, the transformation step. The reordering is performed by dividing the processor storage into as many boxes as there are sublists. The integrals are stored in the appropriate boxes, and are tagged by the sequential position number in the ordered list with $p \geqq q$, and $r \geqq s$, relative to the origin of each sublist. If a box is filled, the contents is written to external direct access storage, together with the direct access address of the contents of this box previously written. In this way "backchained" sublists are created on direct access storage, which can be retrieved by direct addressing record by record, in reverse order. By means of this technique only the direct access address of the contents of each box written last needs be kept in processor storage. – Each integral is uniquely identified by the sublist sequence number, by the sublist size (its index range), and by the relative position number in the sublist. This way of identifying the integrals reduces the number of indices actually to be stored with each integral to one: the relative position number. In addition, a nearly identical index manipulation in both of the two steps is avoided. The relative position number can be used directly in the transformation step.

In the second, the transformation step, the sublists are processed sequentially, one at a time. Each of the sublists is read into the processor storage, record by record, and the integrals are transferred from the input buffer area to their position in the sublist according to their relative position number tag. In this way, a sequence of triangular matrices is built up, each matrix (indirectly) identified by an index pair r and s, and containing the full range of indexes p, and q with $p \geqq q$. For each index pair r, and s, the transformation is carried out over the indexes p, and q generating the semi-transformed integrals $(ij|rs)$ with $i \geqq j$. The summation is carried out stepwise according to the formulae (4.1) and (4.2). First the integrals $(iq|rs)$ are formed for fixed i, r, and s, and for $1 \leqq q \leqq n$, and stored in a buffer area. Then the semi-transformed integrals $(ij|rs)$ are computed for fixed i, r, and s and for $1 \leqq j \leqq i$. Actually, each triangular matrix is expanded into square form, to avoid extra index manipulation in the formation of the partial sums (at the expense of a slightly larger processor storage request). Linear indexing is used throughout the program. The generated semi-transformed integrals $(ij|rs)$ which are nonzero, or *larger than a threshold* are collected in a buffer area, together with the four indices i, j, r, and s and are finally written onto sequential external storage.

In a completely analogous way, this two-step procedure then is applied to the transformation $(ij|rs) \rightarrow (ij|kl)$.

The approach described can practically be adapted to any size of processor- and direct access storage, by forming and processing a limited number of sublists at a time, according to the storage capacities. This makes necessary a

Table 1. Sample timings of the transformation algorithm

Size of		CPU-Time
Basis FCT. SRT	Transf. FCT. SRT	IBM 360/91
35	35	2'40"
41	39	5' 7"
46	42	7'20"
70	68	~40'

repeated scan of the sequential integral lists $(pq|rs)$, and $(ij|rs)$, and increases the number of sequential files by one to a total of two, not including the original integral list $(pq|rs)$.

This method has been implemented in the form described on an IBM 360/91 computer as part of the MUNICH quantum chemical program system [8]. The program has been written in IBM 360 Fortran, except for the direct access I/O routine which is written in IBM 360 Assembler language in order to use direct track addressing. This routine can be substituted by a routine using the Fortran direct write facilities. The program has been fully tested for function sets up to 70 and has been in use for about 6 months. Typical timings for the transformation are given in the following Table 1.

References

1. Tang, K. C., Edmiston, C.: J. Chem. Phys. **52**, 997 (1970)
2. Sutcliffe, B. T.: The POLYATOM system, Part III. Configuration interaction subroutines
3. Veillard, A.: IBMOL/VERSION IV, Special IBM Corp. Technical Report (1968), San Jose
4. Nesbet, R. K.: Rev. Mod. Phys. **35**, 552 (1963)
5. Yoshimine, M.: IMB Corp. Technical Report RJ 555 (1969), San Jose
6. McLean, A. D.: Proceedings of the conference on potential energy surfaces in chemistry. Univ. of Cal., Santa Cruz, Aug. 10–13, 1970. IBM Corp. Technical Report RA–18, San Jose
7. Bender, C. F.: J. Compl. Phys. **9**, 547 (1972)
8. Diercksen, G. H. F., Kraemer, W. P.: Molecular program system, Reference Manual, Special Technical Report, München. Max-Planck-Institut für Physik und Astrophysik (to be published)

Dr. G. H. F. Diercksen
Max-Planck-Institut für Physik und Astrophysik
D-8000 München 40, Föhringer Ring 6
Federal Republic of Germany