Theoret. Chim. Acta (Berl.) 39, 247—253 (1975)

The Two-Electron Integral Transformation and Two-Body Density Matrix Transformation

Christopher N. M. Pounder*
Department of Chemical Engineering, University of Cambridge, England

Received December 10, 1974/May 6, 1975

An n^5 algorithm for the transformation of quantum-mechanical four centre functions is presented in a form best suited for computers having a virtual memory capability.

Key words: Four centre integrals, transformation of ~ – Two-body density matrix

1. The Two-Electron Integral Transformation

A recent advance in the timing and efficiency of the two electron transformation process has recently been published by Diercksen [1], and it is the purpose of this note to propose an improved mechanism by which the number of multiplications and additions can be further reduced. For the sake of brevity, the reader is directed to this paper for recent references and the importance of speed to this process to present-day large scale quantum mechanical calculations.

The process is carried out in four stages, offers full facilities such as partial transformation, ordered restart points, and efficient use of partially transformed integrals, has been used successfully in Newcastle over the last two years.

The method utilizes to the full the well known equalities between integrals

$$(ij|kl)=(ji|kl)=(ij|lk)=(ji|lk)=(lk|ij)=(kl|ij)=(lk|ji)=(kl|ij) \tag{1}$$

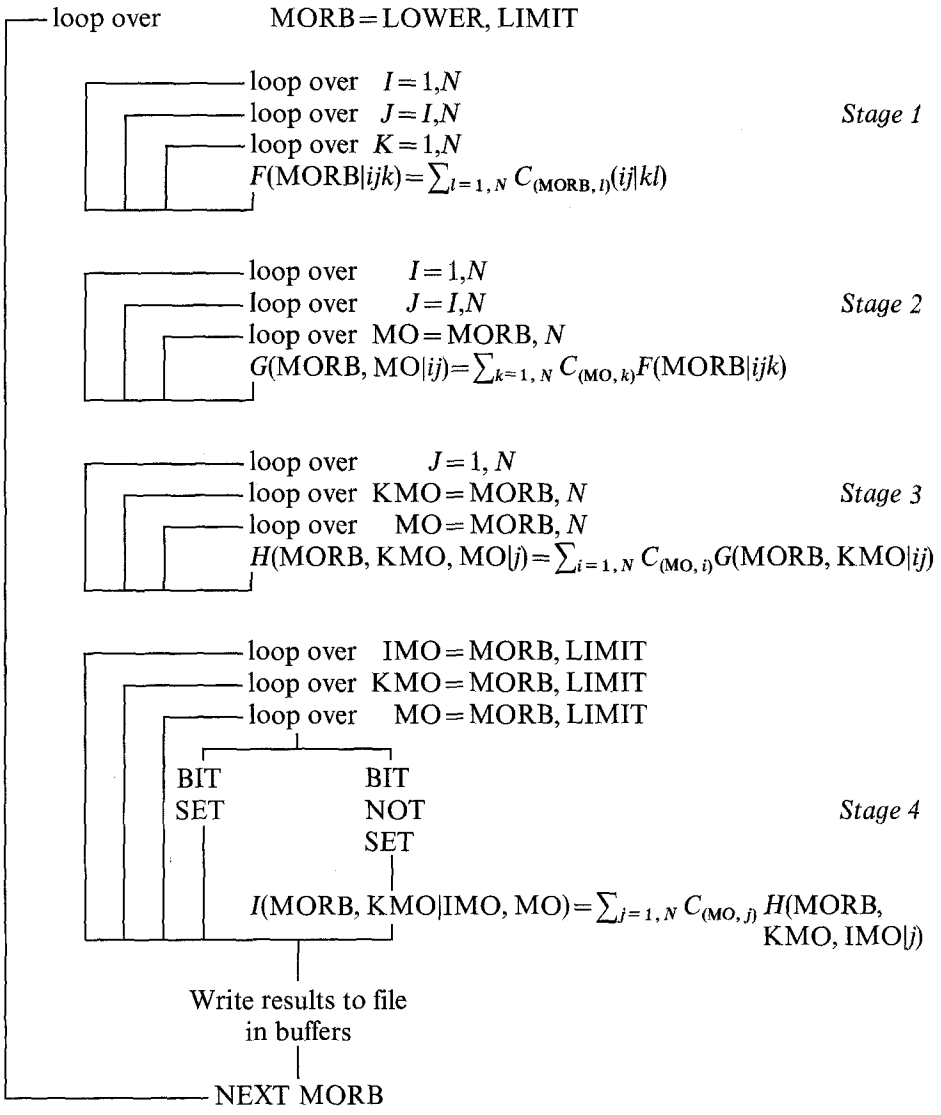
where the integral $(ij|kl)$ is the same charge cloud notation defined in Diercksen's paper.

1.1. Flow Chart of the Transformation

The notational definitions are self-explanatory with F, G, H representing the partially transformed integrals, I being the completely transformed integral. LIMIT is the upper range to which transformation is to proceed whilst LOWER is the starting (or re-starting) label. All labels involving MO are transformed labels, and the matrix of coefficients is held in rows by C. MORB is the current label over which the transformation is occurring and N the total basis set size.

It will be useful to define a term MASTx, the number of Multiplications and Additions for Stage x in measuring the transformation's efficiency.

* Part of the work to be submitted for the degree of Ph. D. in the University of Newcastle-upon-Tyne.



1.1.1. Stage 1

We can use the equality $(ij|kl)$ and $(ji|kl)$ for general k and l to reduce the number of operations by forcing the transformation to start using atomic orbitals only with labels $j \geq i$. From the flow diagram it can be seen that the cost of this stage is

$$\text{MAST}1 = \tfrac{1}{2}N(N+1)N^3 \quad \text{multiplication and additions}.$$

1.1.2. Stage 2

The equality between $(ij|kl)$ and $(ji|kl)$ is again used, as is the ability to start one of the loops from the current value of MORB. This is possible, because when

transformation over MORB = 1 is complete, all integrals involving the label 1 have been transformed. When MORB = 2, the next time round, there is no need to introduce any label of a transformed index involving the label 1, and hence we are able to start transforming from a label = MORB.

The cost of this stage is therefore

$$\tfrac{1}{2}N(N+1)N[N+(N-1)+(N-2)\ldots 1]$$

which can be simplified to

$$\text{MAST}2=\tfrac{1}{4}N^3(N+1)^2\,.$$

At the end of this stage we have transformed over one complete charge cloud.

1.1.3. Stage 3

Two labels now run over the reduced range starting from MORB for the above reason. Care must be taken at this juncture in fetching from core each half transformed integral as it is required twice. This can be seen by example, where using primes to denote transformed labels, the (1′3′|56) must serve for the (1′3′|65) when this latter integral is required. This is the cost of transforming earlier stages over $j \geqslant i$. The number of additions and multiplications is given by

$$\text{MAST}3=N^2[N^2+(N-1)^2+(N-2)^2\ldots 1^2]\,.$$

1.1.4. Stage 4

As well as having three labels running over the reduced range, we can utilize (1) at this point before proceeding with the transformation of a particular integral, as all the labels of this transformed integral are known. From this label, it is possible to generate a unique position of this integral in a one-dimensional array [2], which can be mapped by a bit-array used to test whether this set of labels has occurred before. If the bit is set then the transformation is rejected, whilst if the bit is not set, the bit is then set and transformation of this integral proceeds. Hence the number of multiplications and additions is merely the number of unique integrals multiplied by N.

$$\text{MAST}4=\tfrac{1}{2}N''(N''+1)N \quad \text{where} \quad N''=\tfrac{1}{2}N(N+1)\,.$$

The total number of multiplications and additions is conventionally of the order CN^5 where C is some constant, which can be equated to the components MAST 1–4 for the stages of the transformation.

$$\text{MATOT}=CN^5=\text{MAST}1+\text{MAST}2+\text{MAST}3+\text{MAST}4\,.$$

Dividing both sides of the above by N^5 yields C

N	5	10	20	30	40	50	60	70
C	1.592	1.392	1.299	1.267	1.253	1.243	1.237	1.233

from which it can be said that the process lies between about a $(7/5)n^5$ and $(5/4)n^5$ transformation for most basis sets.

2. Computational Implementation

One of the criticisms which can be raised is similar to that in a previous work [3], that the process requires a work space of N^3 locations thus limiting the range of values N can take. However these criticisms can be invalidated if the computer is attached to Dynamic Address Translation [4] hardware, better known as a Virtual Memory capability, so long as, and the following point has to be stressed, the programmer takes care to minimise page-faults, and as far as possible uses a sequential access of this storage.

The implementation at Newcastle-upon-Tyne does use sequential access for the most expensive stages 1 and 2, whilst for stage 4, access is for elements separated by N. Stage 3 can be regarded as random access because the half transformed integral is required twice.

In further detail, before the transformation can proceed the integrals have to be ordered as shown below for the charge distribution $(ij)_1$ where $j \geq i$ and $i = 1,N$. (The labels have not been ordered to show the structure of the array clearly.)

$$\begin{array}{llll} (11|(ij)_1) & (12|(ij)_1) \ldots & (1k|(ij)_1) \ldots & (1N|(ij)_1) \\ (21|(ij)_1) & (22|(ij)_1) \ldots & (2k|(ij)_1) \ldots & (2N|(ij)_1) \\ \vdots & \vdots & \vdots & \vdots \\ (N1|(ij)_1) & (N2|(ij)_1) \ldots & (Nk|(ij)_1) \ldots & (NN|(ij)_1) . \end{array}$$

The discussion will use primes to denote transformed labels and $(ij)_n$ to represent a generalised charge distribution (ij) where $j \geq i$.

This sort may be achieved by presorting the untransformed integrals (labels ordered) by N blocks of $(ij)_n$ for each i where $j = i, N$; as each block will not contain any integral with all labels greater than the current value of i. The process then could rearrange each of these blocks, placing as many integrals as possible from this block, before searching for the rest in the previous blocks. Although this process is not used (it is inefficient in terms of I/0 operations) it is the easiest to visualise.

To form the quarter transformed integral by

$$F(P'|k(ij)_n) = \sum_l^N C_{(P',l)}(kl|(ij)_n)$$

all that is required is a sequential read through of the files of integrals on disk, placing the result incrementally in the N^3 workspace. The structure of this F matrix will then be ordered as

$$\begin{array}{llll} (P'|1(ij)_1) & (P'|2(ij)_1) \ldots & (P'|k(ij)_1) \ldots & (P'|N(ij)_1) \\ (P'|1(ij)_2) & (P'|2(ij)_2) \ldots & (P'|k(ij)_2) \ldots & (P'|N(ij)_2) \\ (P'|1(ij)_n) & (P'|2(ij)_n) \ldots & (P'|k(ij)_n) \ldots & (P'|N(ij)_n) \end{array}$$

The half transformed integral is formed using

$$G(P'Q'|(ij)_n) = \sum_k^N C_{(Q',k)} F(P'|k(ij)_n) .$$

All this requires is selection of the quarter transformed integrals in blocks of N, which are then transformed over all possible labels and placed back in the matrix in the position from where they came.

Assume that in the previous case $P'=1$, then the structure of the half transformed G matrix would be

$$\begin{matrix}(P'1'|(ij)_1) \ldots (P'Q'|(ij)_1) \ldots (P'N'|(ij)_1)\\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots\\ (P'1'|(ij)_n) \ldots (P'Q'|(ij)_n) \ldots (P'N'|(ij)_n).\end{matrix}$$

Now if $P'>1$, we only form those half transformed integrals $Q' \geq P'$ *but* place them in the structure as if we had formed all the N half transformed integrals.

The transformation over the third index requires careful handling, as values from differing charge distributions $(ij)_n$ have to be obtained.

$$H(P'Q'R'|j)=\sum_i^N C_{(R',j)} G(P'Q'|(ij)).$$

The $H(P'Q'R'|j)$ requires the following half transformed integrals: $(P'Q'|1j)$ $(P'Q'|2j) \ldots (P'Q'|Nj)$, and as we have assumed $i \leq j$, all those integrals where $i>j$ will be found stored as $(P'Q'|(ji))$. A study of the G matrix will show that all those values of $(P'Q'|(ij))$ where $j \geq i$ are seperated by N positions, and those where $i>j$ are separated by $V \times N$ positions where V is the number of charge distributions between (j, i) and $(j+1, i)$. For example the $(P'Q'|14)$ and the $(P'Q'|24)$ are separated by $N(N-1)$ positions as there are $N-1$ charge distributions $(ij)_n$ from (14)(15)... (1N) (22) ... (24) to be by-passed in calculating the address.

The program procedure is that, given the starting position of the first half transformed integral, the position of the rest of the required integrals are calculated in two loops, the first being whilst $i>j$ (when charge distributions are missed out), the other being from $j \geq i$ (when N is added to the previous value to calculate the required address). After transforming over $R' \geq P'$ the resultant three-quarter transformed integral can be stored either in another N^3 workspace inside the virtual machine or incrementally in a buffer that is written to disk periodically, remembering that as before those values $R' \leq P'$ will not be formed, but *must be accounted for* in the buffer or workspace.

Note that having transformed $H(P'Q'R'|j)$, $R'=P', N$ transformation passes to the $H(P'(Q+1)'R'|j)$, $R'=P', N$. A study of the structure of the G matrix will show that the addresses of the half transformed integrals required will be found by adding 1 to the addresses of those required for the $H(P'Q'R'|j)$, and consequently all complicated array addressing takes place in the outermost loop (when j changes).

The final stage utilises the structure of the H matrix shown below for $P'=Q'=1'$.

$$(P'Q'1'|j_1) \ldots (P'Q'R'|j_1) \ldots (P'Q'N'|j_1)$$

$$(P'Q'1'|j_2) \ldots (P'Q'R'|j_2) \ldots (P'Q'N'|j_2)$$

$$(P'Q'1'|j_n) \ldots (P'Q'R'|j_n) \ldots (P'Q'N'|j_n)$$

and

$$I(P'Q'|R'S')=\sum_j^N C_{(S',j)} H(P'Q'R'|j)$$

from which it can be seen that the required three quarter transformed integrals are separated by N places. On collecting these N values, we use the bit map as a test whether to transform the integral thus making the transformation of the same integral twice impossible.

3. Transformation of the Two-body Density functions

The process can be adapted for the transformation of two body density functions quite simply, if we assume a change in notation [5] so that

$$(ij|kl) \equiv \varphi_i(r_1)\varphi_j(r_2)\varphi_k^*(r_1')\varphi_l^*(r_2')$$

where the element now is a member of a two-body density function. It can be seen that the permutations between labels are restricted to

$$(ij|kl) = (ji|lk) = (lk|ji) = (kl|ij)\,.$$

The implication for this is that interchange of i and j cannot now be independent of interchange of k and l (as in the two electron integrals case assumed by the flow chart). All that is required however to compensate for this, is to change all loops in the flow chart from $J = I, N$ to $J = 1, N$ as all other addressing of arrays will be seen to be automatically compensated for by this extension. The problem of selection and addressing of unique density matrix elements is discussed elsewhere [2].

4. Conclusion

With paging devices such as drums and fixed head disks offering the user a transparent expansion of real core by 10 or more megabytes, the major contribution of this note is to suggest that traditional programming methods may have to be re-thought to utilize to the full what can be a powerful aid in manipulating large amounts of data usually associated with present day calculations.

Finally a few brief comments are necessary on the advantage/disadvantage of virtual storage [4] in the normal multi-programming environment.

Firstly, a direct performance comparison with non-virtual machines is impossible because of the 'system overhead' in the extra steps required in address computation, and I/O of demand paging. Secondly, the paging rate, (and hence elapsed time), is not only a function of how the user program is written, but also on how the job-mix user programs are written. Thirdly, the supervisor is likely to monitor paging and hence effect a job's real activity according to system load at a particular time. Fourthly, hardware differences in core storage would mean that running times would vary from installation to installation (even using the release of operating system). Consequently both cpu and elapsed times can vary considerably from run to run and machine to machine, and hence realistic and meaningful comparisons in the timing are of even more questionable value.

However it is the author's belief that as these types of machine are becoming commonplace, large data sets can be manipulated simply and efficiently if, and only if, sequential access of the data is used. The method presented here is not

perfect, especially at stage 3, but as techniques and technology of these types of machines improves, a viable alternative to the few dedicated computers could become available.

Acknowledgements. I would like to thank N.U.M.A.C. for use of the IBM 360/67, the operators who were always helpful, and Dr. I. L. Cooper for many useful suggestions.

References

1. Diercksen, G. H. F.: Theoret. Chim. Acta (Berl.) **33**, 1 (1974)
2. Pounder, C. N. M.: Ph. D. Thesis, Newcastle-upon-Tyne University 1975
3. Bender, C. F.: J. Comp. Phys. **9**, (1972)
4. Introduction to virtual storage in the System 370/GR 20-4260-0 (IBM Publication, 1972)
5. McWeeny, R., Sutcliffe, B. T.: Methods in molecular quantum mechanics. New York: Academic Press 1969

Dr. C. N. M. Pounder
Department of Chemical Engineering
University of Cambridge
Pembroke Street
Cambridge CB2 3RA, England