

A Simple Method to Eliminate Linearly Dependent AO Integrals in ab initio LCAO MO Calculations by Taking Advantage of Shell Structure

TOSHIKAZU TAKADA*

*Department of Chemistry, Faculty of Science,
Hokkaido University, Sapporo 060, Japan*

Received September 19, 1980

A new method to eliminate linearly dependent two electron integrals in an AO basis is proposed. In the conventional method, the number of computer operations to make the list of independent integrals is proportional to the number of two electron integrals. It is proportional to the number of overlap charges in the present method. Thus, the computer time required is negligible in comparison with the conventional method. Results of actual computations using the present method are reported for naphthalene ($C_{10}H_8$), cyclobutane (C_4H_8) and NiF_6 .

1. INTRODUCTION

Recently, ab initio LCAO MO calculations have been carried out on many molecules of medium size using modern computers. However, the treatment of two electron integrals is still a major problem in so-called large scale calculations. The advantage of evaluating only linearly independent integrals is a substantial saving in computer time and storage space. There are a few computer programs for ab initio LCAO SCF MO calculations which take molecular symmetry into account [1-3]. A program package called JAMOL2 for ab initio LCAO SCF MO calculations has been completed by our research group. JAMOL2 uses the full molecular symmetry to reduce the number of two electron integrals that must be treated. The procedure for computing two electron integrals using symmetry consists of the following three steps: (1) creating a list of independent two electron integrals; (2) generating the integrals; and (3) transforming them to a symmetry orbital (SO) basis. In the course of using JAMOL2, it became apparent that for molecules with high symmetry creating the integral list and transforming them to a SO basis were more time consuming processes than generation. For example, in a CoF_6 calculation [4], 20 and 64 % of the computer time was spent in creating the list and transforming to a SO

* Present address: NRCC Lawrence Berkeley Laboratory, University of California, Berkeley, Calif. 94720.

basis, respectively, while only 11% was consumed in generating the integrals.¹ Therefore, for the sake of economy, it is important to increase the efficiency of the two most time consuming steps. An improved algorithm for transforming the integrals has been previously proposed by Takada and Sasaki [5].

A new algorithm to make the list of linearly independent two electron integrals is described in the present paper. Conventionally, the symmetry operations of the point group are applied to each integral to determine the linearly dependent integrals. Therefore, the number of necessary computer operations is proportional to the number of two electron integrals and it becomes very time consuming to make the list in the case of large scale calculations. In the present method, the number of computer operations required to make the list is only proportional to the number of overlap charges. Thus, the computer time required for the list creation is negligible in comparison with the conventional method. Although some dependent integrals are generated in the present method, the total amount of computer time required for computing the two electron integrals is significantly reduced.

Many improvements have been completed on JAMOL2, including implementation of the present algorithm. The new version is called JAMOL3 [6].

In this paper, the algorithm is presented in Section 2 and some results from actual computations are given in Section 3.

2. ALGORITHM

When contracted Gaussian type orbitals (CGTOs) are used as basis functions, the evaluation of two electron integrals can be carried out very efficiently by introducing shell structure [9]. A shell consists of all CGTOs which have the same orbital exponents and are centered on the same atom. The shells are denoted $s_i^A, s_j^A, \dots, s_i^B, s_j^B, \dots, s_n^D$, where the superscripts indicate on which atom each shell is centered and the subscripts indicate the type of shell. For example, in the case of a minimal basis set for a carbon atom, the shells are $s_1^C = \{\chi_{1s}\}$, $s_2^C = \{\chi_{2s}\}$, $s_3^C = \{\chi_{2p_x}, \chi_{2p_y}, \chi_{2p_z}\}$ when different orbital exponents are used for $2s$ and $2p$ orbitals.

Now we show how molecular symmetry can be applied to the shell structure to eliminate most of the dependent two electron integrals. When the molecule has symmetry, equivalent atoms in the molecule can be determined by applying the symmetry operations of the point group. We define new shells S_I, S_J, S_K and S_L , called extended shells, as follows: the extended shell S_I consists of all shells which are equivalent by symmetry. In other words, shells of the same type on equivalent atoms form an extended shell, for example, $S_I = \{s_2^{C1}, s_2^{C2}, s_2^{C3}, s_2^{C4}\}$, where carbon atoms $C1, C2, C3$ and $C4$ can be transformed into one another by appropriate symmetry operations of the point group.

¹ These results were obtained with JAMOL2 using an integral generation program written by Sasaki, which does not take advantage of shell structure [7]. This program has now been revised by Sasaki to include shell structure and is incorporated in JAMOL3 [8]. The percentage of the total computer time required for generation using the new integral program is much less than with the previous version.

The handling of two electron AO integrals is simplified by arranging them in extended shell blocks ($S_I S_J | S_K S_L$). The advantage of this arrangement is that list creation, integral generation and transformation to a SO basis can be carried out for each block independently of the others.

For an integral block ($S_I S_J | S_K S_L$), list creation involves the following two steps: (1) eliminating dependent two electron integrals and (2) determining which integrals are zero by symmetry. This is accomplished by (1) creating two lists of overlap charges, corresponding to each of the ($S_I S_J$) and ($S_K S_L$) sets and (2) forming bit patterns for individual overlap charges, which indicate if an AO integral given by combining the two overlap charges is zero by molecular symmetry.

First, a list of the independent overlap charges included in the ($S_I S_J$) set is made. To make the list, one of the shells, for example, s_i^A , is arbitrarily selected from the extended shell $S_I = \{s_i^A, s_i^{A'}, s_i^{A''}, \dots, s_i^{A''''}\}$ and combinations are formed with all the shells included in the extended shell S_J , i.e., ($s_i^A s_j^B$), ($s_i^A s_j^{B'}$), ..., ($s_i^A s_j^{B''''}$). If some dependent overlap charges are included in the list, they are deleted by applying symmetry operations. Next, a list is made for the ($S_K S_L$) set. This list is constructed

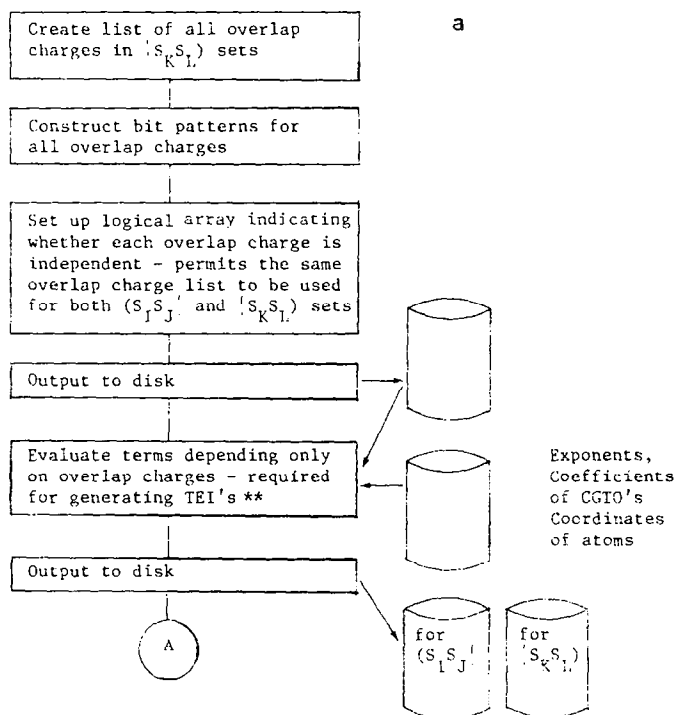


FIG. 1. Flowchart of program. (a) Preparation for integral generation. The integral evaluation program was written by F. Sasaki [8] and is incorporated in JAMOL3 [6]. (**): TEI: two electron integral. (***) An equivalent orbital group consists of the atomic orbitals which are transformed into one another by symmetry operations.

differently from the $(S_I S_J)$ list in that it contains all combinations of pairs of shells including dependent ones. The overlap charge lists are stored on disk and are later combined in the integral generation step to determine which integrals must be evaluated. Thus, with the present algorithm, the necessity of storing an integral list is avoided.

Next, the following procedure can be used to determine which integrals are zero by symmetry so as to avoid computing them. Each symmetry adapted overlap charge corresponds to a particular irreducible representation of the point group and consists of a linear combination of AO overlap charges $\chi_r \chi_s$. Each AO overlap charge will contribute to one or more symmetry adapted overlap charges. The information can be stored in the computer as a bit pattern. For example, in H_2O , an overlap charge of the type $\chi_{1s_H1} \chi_{1s_H1}$ contributes to two symmetry adapted overlap charges which belong

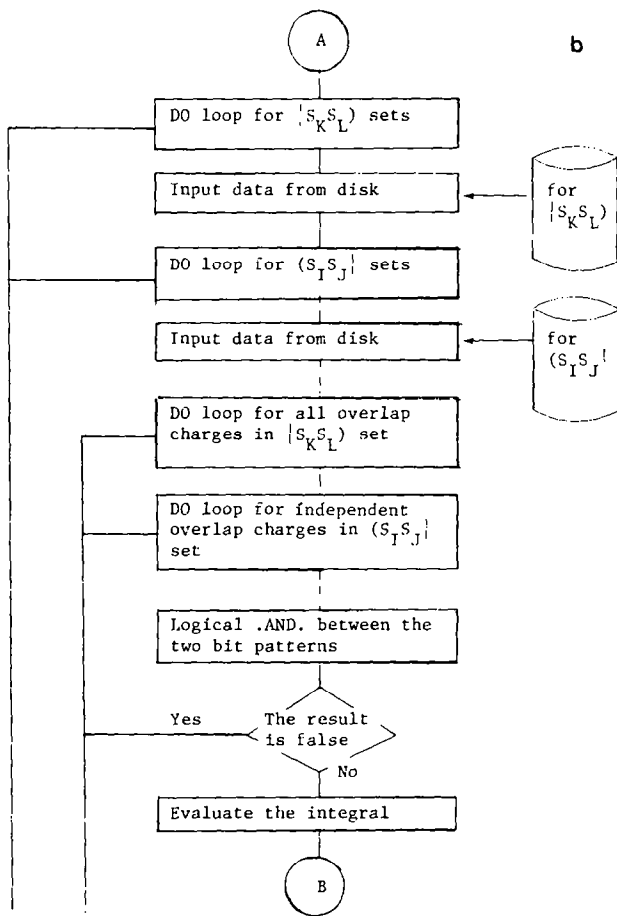


FIG. 1. (b) Integral generation for block $(S_I S_J | S_K S_L)$.

to the A_1 and B_2 irreducible representations of the C_{2v} point group. Therefore, the bit pattern for the overlap charge χ_{1s_H}, χ_{1s_H} is

	A_1	A_2	B_1	B_2
χ_{1s_H}, χ_{1s_H}	1	0	0	1

That is, a bit corresponds to each irreducible representation. If a given bit is on, the AO overlap charge contributes to the symmetry adapted overlap charge belonging to the corresponding irreducible representation. The value of the integral $(\chi_r \chi_s | \chi_t \chi_u)$ is zero unless the bit patterns for the two overlap charges $\chi_r \chi_s$ and $\chi_t \chi_u$ have at least one pair of corresponding bit positions turned on. To check whether the integral vanishes or not, the logical operation ·AND· is taken between the two bit patterns in the integral generation step. If the result of the logical operation is false, the value of the integral is zero and the integral is not calculated.

In Fig. 1, a simplified flowchart of the program implementing the present algorithm is shown. It should be noted that the first three steps depend only on overlap charges and therefore the number of computer operations is proportional to the number of overlap charges, not to the number of two electron integrals. The test using the logical operation ·AND· is repeated as many times as there are numbers of two electron integrals. However, since the logical operation is very fast, the CPU time required is negligible.

The three steps of list creation, generation and transformation are carried out for each integral block $(S_I S_J | S_K S_L)$ for all combinations of I, J, K and L . After completion of the transformation step for a given block, the resulting SO integrals are output to disk or magnetic tape. Therefore, the necessary core is twice the core needed for storing the independent AO integrals, since sorting them into equivalent

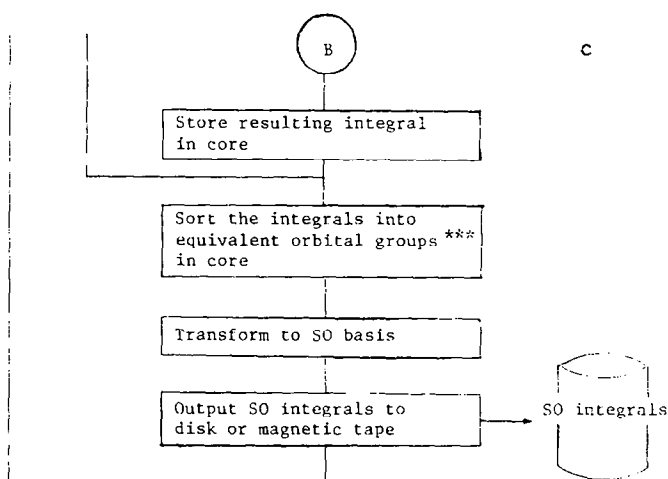


FIG. 1. (c) Transformation to a SO basis.

TABLE I
Number of Two Electron Integrals

Molecules	Symmetry	Number of orbitals	Number of TEIs ^a	Number of independent TEIs ^a	Ratio
CH ₂	<i>C</i> _{2v} (4) ^b	15 ^c	7,260	2,584	0.36
C ₄ H ₈	<i>D</i> _{4h} (16)	56	1,274,406	86,800	0.068
C ₁₀ H ₈	<i>D</i> _{2h} (8)	58	1,464,616	219,484	0.15
				(102,600) ^d	(0.070)
NiF ₆	<i>O</i> _h (48)	61	1,788,886	46,490	0.026
CoF ₆	<i>O</i> _h (48)	128	34,084,896	820,567	0.024
Cu-porphine	<i>D</i> _{4h} (16)	178	126,906,346	9,633,076	0.076
				(904,080)	(0.007)
Co-porphine	<i>D</i> _{4h} (16)	187	154,501,831	10,096,264	0.065
				(2,284,595)	(0.015)

^a TEI; two electron integral.

^b Number of symmetry operations.

^c The basis sets; minimal basis set for C₁₀H₈, double zeta basis set for C₄H₈ and $|4s3p2d/3s1p|$ for NiF₆.

^d An integral approximation based on semiorthogonalized orbitals is adopted. See Refs. [10–12].

orbital groups is needed in the present method as shown in Fig. 1. In the present method, the core memory requirement is largest in the case of the *O_h* point group because there are six equivalent ligand atoms. In this case, the largest extended shell consists of 3*d* orbitals on the ligand atoms and the number of independent overlap charges is 76.² Therefore, the core size for real applications is estimated not to exceed 1.6 Mbyte. Most modern computers have adequate core memory to permit the above procedures to be carried out in core without the necessity of creating temporary files.

3. RESULTS

The number of linearly independent two electron integrals is shown for several molecules in Table I. It is apparent that in each case the number of independent integrals is roughly equal to the number of two electron integrals divided by the number of symmetry operations in the point group. Therefore, even in the case of lower symmetry, the use of molecular symmetry allows a considerable reduction in computer time and storage space.

In Table II, the number of two electron integrals that must be evaluated as

² When 3*s*, 3*p* and 3*d* functions with the same exponents or 4*f* functions are used for ligand atoms, the number of AOs in an extended shell is large. Thus, more core memory will be required to store the integrals. However, in this case, the extended shell can be divided into smaller groups in such a way as not to break up equivalent orbital groups. The efficiency of integral evaluation is slightly reduced by dividing the extended shell into smaller groups, but the core size required is kept to less than 1.6 Mbyte.

TABLE II
Number of Two Electron Integrals

	Molecules		
	$C_{10}H_8^a$	C_4H_8	NiF_6^b
Number of CGTOs	58	56	61
Symmetry	D_{2h}	D_{3h}	O_h
Number of two electron integrals			
1. No consideration of symmetry	1,464,616	1,274,406	1,788,886
2. Full consideration of symmetry	102,600	86,800	46,490
3. Symmetry operations on overlap charges	198,015	140,748	153,418
4. Symmetry operations on overlap charges and zero check	114,095	117,436	142,622
Ratio (4./2.)	1.11	1.35	3.07

^a An integral approximation based on semiorthogonalized orbitals is adopted. See Ref. [10].

^b See Ref. [13].

obtained by the present method is shown for C_4H_8 , $C_{10}H_8$ and NiF_6 . The numbers in the third row were obtained without testing for integrals that are zero by symmetry. The present method reduces the number of integrals to almost that of the independent integrals obtained by full consideration of molecular symmetry. The check for zero integrals leads to a relatively small but significant reduction in the number of integrals to be calculated. Since $C_{10}H_8$ is a planar molecule and integrals of type $(\sigma\sigma|\sigma\pi)$ and $(\sigma\pi|\pi\pi)$ are zero, the test for zero integrals is particularly effective in reducing the number of integrals for this molecule. The reason why the ratio is largest for NiF_6 in spite of its having the highest symmetry can be explained in the following way. The number of equivalent AOs on the Ni atom is only 3. In the NiF_6 calculation, the number of extended shells on Ni and F atoms is 9 and 4, respectively. The probability that either S_j and/or S_l is an extended shell on the Ni atom in all the $(S_j S_l)$ combinations, is 89%. Therefore, the present method does not do as well in reducing the number of dependent integrals in the NiF_6 calculation. Nevertheless, the reduction seems to be adequate because the number of integrals obtained by this method is about 1/13th of the total number. The ratio will be much smaller in the case of complexes such as $Co(CN)_6$, because the number of AOs on the ligand atoms increases in comparison with the number of AOs on the central metal.

The present approach outlined to determine the independent overlap charges in $(S_j S_l)$ is identical to the method proposed by Dupuis and King [3]. In their method, further redundancies related to overlap charges in $(S_k S_l)$ are eliminated, while the present method uses all the overlap charges. As mentioned above, this creates the higher ratio for NiF_6 in Table II. However, from the other two ratios, the $(S_k S_l)$ redundancies are almost eliminated even with this method. The present algorithm was

TABLE III
Computer Time by JAMOL2 and JMAOL3 (sec)

	Molecules	JAMOL2	JAMOL3	Ratio
List creation	C ₄ H ₈	90	—	
	C ₁₀ H ₈	32	—	
	NiF ₆	132	—	
Generation of TEIs in AO basis ^a	C ₄ H ₈	156	38 (1.35) ^b	5.5 ^c
	C ₁₀ H ₈	1573	563 (1.11)	3.1
	NiF ₆	167	231 (3.07)	2.2
Total	C ₄ H ₈	246	38	6.5
	C ₁₀ H ₈	1605	563	2.9
	NiF ₆	299	231	1.3

^a The number of primitive CGTOs is 84, 174 and 128 in C₄H₈, C₁₀H₈ and NiF₆, respectively.

^b Ratio of the number of two electron integrals to be calculated in JAMOL3 to the number of independent integrals. See Table II.

^c Ratio of CPU time for calculating one integral.

designed to make a good interface with a transformation algorithm from an AO basis set to a SO basis set [5], and the Dupuis and King method and the present one are two different approaches in ab initio calculations to take advantage of molecular symmetry.

In Table III, the computer time for these calculations is shown. The integral evaluation program in JAMOL2, which was written by Sasaki, does not take advantage of shell structure [7]. With the use of shell structure and many additional improvements on the program, integral evaluation is much faster with JAMOL3 [8].

The present method has the following advantages.

1. The number of computer operations needed for list creation is only proportional to the number of overlap charges, not to the number of two electron integrals. Therefore, the computer time required is negligible in comparison with the conventional method. Even though some dependent integrals are evaluated, much computer time is saved in large scale calculations.

2. This algorithm can be easily used in conjunction with integral approximations such as those based on semiorthogonalized orbitals [14, 15]. In this approximation scheme, overlap charges are divided into two groups, strongly related pairs and weakly related pairs. If either of the overlap charges $\chi_r\chi_s$ and $\chi_r\chi_u$ in an integral $(\chi_r\chi_s | \chi_t\chi_u)$ is a weakly related pair, the integral is ignored. Thus, any overlap charge which is a weakly related pair can be omitted from the $(S_r S_j)$ and $(S_K S_L)$ lists.

By using the present algorithm and the improved integral transformation algorithm

to a SO basis set, the efficiency gained by exploiting molecular symmetry is significantly increased in large scale calculations especially for metal complexes such as metallo-porphine.

ACKNOWLEDGMENTS

The author wishes to thank Professor K. Ohno, Professor H. Kashiwagi and Dr. G. D. Zeiss for reading and commenting on this paper, and especially Professor F. Sasaki for valuable advice during development of the computer program. The calculations in this paper have been carried out on the FACOM 230-75 at the Hokkaido University Computing Center.

REFERENCES

1. POLYATOM (version 2) QCPE No. 199.
2. J. ALMÖLF, *Int. J. Quantum Chem.* **8** (1974), 915.
3. M. DUPUIS AND H. KING, *Int. J. Quantum Chem.* **11** (1977), 613.
4. E. MIYOSHI, T. TAKADA, S. OBARA, H. KASHIWAGI, AND K. OHNO, *Int. J. Quantum Chem.* **19** (1981), 451.
5. T. TAKADA AND F. SASAKI, *Int. J. Quantum Chem.* **18** (1980), 1157.
6. H. KASHIWAGI, T. TAKADA, S. OBARA, E. MIYOSHI, AND F. SASAKI, to be published. A library program of the Institute for Molecular Science in Okazaki and the Hokkaido University Computing Center in Sapporo, Japan.
7. F. SASAKI, private communication.
8. F. SASAKI, to be published.
9. W. J. HEHRE, R. F. STEWART, AND J. A. POPLI, *J. Chem. Phys.* **51** (1969), 2657.
10. H. KASHIWAGI AND T. TAKADA, *Int. J. Quantum Chem.* **12** (1977), 449.
11. S. OBARA AND H. KASHIWAGI, to be published.
12. H. KASHIWAGI, T. TAKADA, S. OBARA, E. MIYOSHI, AND K. OHNO, *Int. J. Quantum Chem.* **14** (1978), 13.
13. The calculation on NiF₆ by Gladney and Veillard was repeated during debugging of the program. H. M. GLADNEY AND A. VEILLARD, *Phys. Rev.* **180** (1969), 385.
14. H. KASHIWAGI AND F. SASAKI, *Int. J. Quantum Chem. Symp.* **7** (1973), 515.
15. H. KASHIWAGI, *Int. J. Quantum Chem.* **10** (1976), 135.