



Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

RECEIVED

Materials & Molecular Research Division

LIBRARY OF CALIFORNIA

JUN 2 1981

LIBRARY OF
DOCUMENTS

Submitted to the Journal of Computational Chemistry

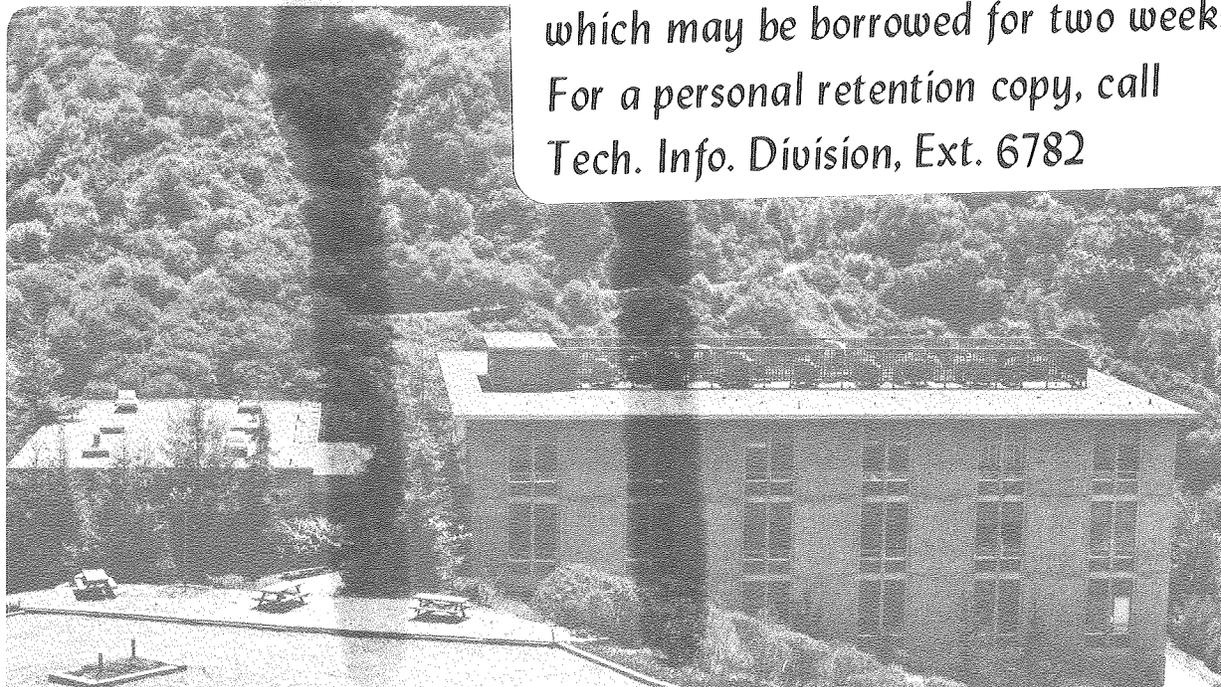
AN ALGORITHM FOR THE GENERATION OF NUCLEAR
SPIN SPECIES AND NUCLEAR SPIN STATISTICAL WEIGHTS

K. Balasubramanian

May 1981

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 6782*



LBL-12644
c.2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

An Algorithm for the Generation of Nuclear Spin Species
and Nuclear Spin Statistical Weights

K. Balasubramanian

Department of Chemistry

and

Lawrence Berkeley Laboratory

University of California, Berkeley, California 94720



Abstract

This paper develops a set of algorithms for the computer generation of nuclear spin species and nuclear spin statistical weights potentially useful in molecular spectroscopy. These algorithms generate the nuclear spin species from group structures known as generalized character cycle indices (GCCIs). Thus the required input for these algorithms is just the set of all GCCIs for the symmetry group of the molecule which can be easily computed from the character table. The algorithms are executed and illustrated with examples.

In recent years chemical applications of non-numerical computational techniques are becoming important in several areas¹⁻⁴. Generation of discrete structures such as graphs and combinatorial structures has been made possible with the advent and rapid growth of algorithms and computational techniques. Randić¹⁻³ has pioneered some such chemical applications of discrete non-numerical computational techniques. In this paper we consider another important application of combinatorial algorithms to a problem in molecular spectroscopy that has been of considerable interest for several decades.

An experiment or theory in molecular spectroscopy typically concerns with possible spectral lines and the intensities of such possible lines. The intensities of allowed inter-rovibronic transitions depend upon the nuclear spin statistical weights of the rovibronic levels.⁵⁻⁶ The conventional technique for obtaining the nuclear spin statistical weights is to find the character of the representation spanned by all the possible nuclear spin functions of the molecule and then break it down into irreducible components. For a molecule containing b_1 nuclei of the same kind with a_1 spin states, b_2 nuclei of the same kind with a_2 spin states etc., there are $a_1^{b_1} a_2^{b_2} \dots$ nuclear spin functions. For ^{12}C triphenylene this number is 4096. Thus the technique mentioned above is quite difficult for polyatomics. The objective of the present paper is to develop very efficient algorithms that would generate nuclear spin species and statistical weights from a minimal input.

The statistical weights of rotational levels in the rotational subgroup have been discussed by Placzek and Teller⁷, Wilson⁸⁻⁹ and Mizushima¹⁰. Hougen¹¹ correlated these to the point groups of molecules.

Galbraith¹² obtained the nuclear spin statistical weights using the unitary group approach and Schur's theorem. Recently, Weber¹³⁻¹⁴ discussed the nuclear spin statistics of spherical top molecules belonging to the point groups D_{nd} or D_{nh} ($n \leq 6$). The present author¹⁵ recently developed a general method for the nuclear spin statistics of molecules belonging to any symmetry. In Sec. 2 we briefly review this method. The present author¹⁶⁻²⁴ has been interested in group theoretical methods for problems in chemistry. Even though this paper is written with the intent of making it self-contained, a more extensive account of preliminaries and definitions can be found in text books.¹⁶⁻¹⁹ This paper uses a theorem of Williamson²⁰ for characters of one dimensional representations which was recently generalized by Merris²¹ to characters of higher dimensional representations.

In Sec. 2 we outline the theoretical methods, and in Sec. 3 we describe the set of algorithms for nuclear spin statistical weights.

2. Theory

Let D be the set of nuclei of the same kind in the molecule and let R be the set of possible spin states of the nuclei in D . In this paper we shall treat each kind of nuclei (such as H, ^{13}C , ^{19}F , etc.) separately, obtain the nuclear spin species of each kind of nuclei and eventually the overall nuclear spin species are obtained as direct products of each kind of nuclear spin species. Let F be the set of all maps from D to R . The image of F is the set of spin functions. For example, for a set of spin $1/2$ nuclei, R would be a set consisting of 2 elements which can be denoted by α (spin up) and β (spin down). A map

f_1 from D to R is shown below.

$$f_1(1) = \underset{\sim}{\beta}$$

$$f_1(2) = \underset{\sim}{\alpha}$$

$$f_1(3) = \underset{\sim}{\beta}$$

$$f_1(4) = \underset{\sim}{\alpha}$$

The spin function thus generated by f_1 is $\underset{\sim}{\beta\alpha\beta\alpha}$. The group G is the point group or more generally, the permutation inversion group of the molecule. G acts on the elements of F in accordance to the following recipe.

$$g(f(i)) = f(g^{-1}i) \text{ for every } i \in D.$$

To illustrate if g is the permutation (12)(34), then $g^{-1} = (12)(34)$.

Consequently, f_1 shown above is transformed as

$$gf_1(1) = f_1(g^{-1}1) = f_1(2) = \underset{\sim}{\alpha}$$

$$gf_1(2) = f_1(g^{-1}2) = f_1(1) = \underset{\sim}{\beta}$$

$$gf_1(3) = f_1(g^{-1}3) = f_1(4) = \underset{\sim}{\alpha}$$

$$gf_1(4) = f_1(g^{-1}4) = f_1(3) = \underset{\sim}{\beta}$$

Thus, the spin function $\underset{\sim}{\beta\alpha\beta\alpha}$ is transformed into $\underset{\sim}{\alpha\beta\alpha\beta}$ by the action of $g = (12)(34)$. Define the cycle representation of a permutation $g \in G$, containing b_1 cycles of length 1, b_2 cycles of length 2, etc, as $x_1^{b_1} x_2^{b_2} \dots$

For example, the cycle representation of the permutation (12)(34), is x_2^2 . The cycle index of G, denoted as P_G is defined as the sum of all cycle representations of $g \in G$ divided by $|G|$, the number of elements in G. In symbols,

$$P_G = \frac{1}{|G|} \sum_{g \in G} x_1^{b_1} x_2^{b_2} \dots$$

In order to book-keep the number of various spin states in a spin function, let us introduce the concept of weight of a spin function. With

each $r \in R$, let us associate a weight $w(r)$, which is just a formal symbol that differentiates the various spin states in the set R . Then the weight of any function $f \in F$ is defined as the product of the weights of the corresponding images. Symbolically,

$$W(f) = \prod w(f(d)).$$

To illustrate, the map f_1 defined above which maps 4 nuclei into the spin function $\beta\alpha\beta\alpha$ would have the weight $\alpha^2\beta^2$ if the weight of the spin states α and β are α and β , respectively.

Let $g \rightarrow \chi(g)$ be the character of an irreducible representation Γ in G . Williamson treated χ as the character of one dimensional representations and Merris generalized to characters of higher dimensional representations. Define the generalized character cycle index (GCCCI) corresponding to the irreducible representation Γ whose character is χ as follows.

$$P_G^\chi = \frac{1}{|G|} \sum_{g \in G} \chi(g) x_1^{b_1} x_2^{b_2} \dots$$

Then using the theorems of Williamson and Merris, it can be shown that the generating function for nuclear spin species which corresponds to the irreducible representation Γ whose character is χ , is given by:

$$\text{Theorem 1: G.F.} = P_G^\chi(x_k \rightarrow \sum_{r \in R} (w(r))^k)$$

The coefficient of a typical term $w_1^{b_1} w_2^{b_2} \dots$ in the G.F. gives the frequency of occurrence of the irreducible representation Γ in the set of spin functions with the weight $w_1^{b_1} w_2^{b_2} \dots$. The beauty of this method is that it did not require the character of all spin functions of dimension $a_1^{b_1} a_2^{b_2} \dots$ to decompose into irreducible components. All that we need is the set of GCCCI's of the group which can be obtained immediately by examining

the character table.

If one is interested only in finding the irreducible representations occurring in Γ^{spin} , the reducible representation spanned by all spin functions, then all one needs to do is to set all the weights to unity.

Then theorem 1 reduces to corollary 1.

Corollary 1: The number of times the irreducible representation Γ occurs in Γ^{spin} is given by

$$P_G^X(x_k \rightarrow |R|)$$

where $|R|$ is the number of elements in the set R.

3. Algorithms

Theorem 1 has simplified the problem of obtaining nuclear spin species with great elegance. Yet one needs to expand several multinomials, combine them, and collect the total coefficients corresponding to all possible weights. For example, if one looks at an aggregate of bosons with spin 2 arranged in O_h symmetry, the GCCI of T_{1g} representation and the corresponding G.F. are shown below.

$$\begin{aligned} \text{GCCI}(T_{1g}) &= \frac{1}{48} (3x_1^6 - 3x_2^3 + 6x_1^2x_4 - 9x_1^2x_2^2 + 6x_2x_4 - 3x_1^4x_2) \\ \text{G.F.} &= \frac{1}{48} [3(\alpha + \beta + \gamma + \delta + \epsilon)^6 - 3(\alpha + \beta + \gamma + \delta + \epsilon)^3 \\ &+ 6(\alpha + \beta + \gamma + \delta + \epsilon)^2 (\alpha + \beta + \gamma + \delta)^4 \\ &- 9(\alpha + \beta + \gamma + \delta + \epsilon)^2 (\alpha + \beta + \gamma + \delta + \epsilon)^2 + 6(\alpha + \beta + \gamma + \delta + \epsilon)^2 \\ &- 3(\alpha + \beta + \gamma + \delta + \epsilon)^4 (\alpha + \beta + \gamma + \delta + \epsilon)^2] \end{aligned}$$

where $\alpha, \beta, \gamma, \delta$ and ϵ are the weights of the 5 spin states. There are 210 terms in the above G.F. One has to collect the coefficients of all the 210 terms in order to obtain the nuclear spin species. Thus, one

can see the necessity of an algorithm for computer generation of G.F.'s and, subsequently, nuclear spin species. Let us describe a set of algorithms which will take GCCI's as inputs and generate nuclear spin species.

A. Composition Algorithm

Let n and k be two positive integers. Then a composition of n into k parts is an ordered sum

$$n = t_1 + t_2 + \dots + t_k$$

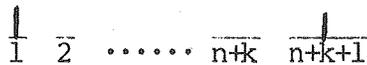
where $t_i \geq 0$.

To illustrate, all the 10 compositions of 3 into 3 parts are shown below.

$$3+0+0, 0+3+0, 0+0+3, 2+1+0, 2+0+1,$$

$$1+2+0, 1+0+2, 0+1+2, 0+2+1, \text{ and } 1+1+1.$$

The algorithm described here is the one given in Nijenhuis and Wilf²² which is based on a method which actually constructs such an arrangement. Suppose one wants to distribute n indistinguishable balls into k labelled cells such that any cell contains 0, 1, 2.... up to n balls, then each such distribution yields a composition. This can be accomplished by constructing the boundaries of cells as follows: Consider a big cell with two walls containing $n+k-1$ available spaces, the two walls occupying the 2 spaces in the extremities as shown below.



The number of ways of distributing n balls in the available $n+k-1$ slots can be seen to be

$$\binom{n+k-1}{n}.$$

After distributing these balls, one can construct the cell boundaries in the rest $k-1$ available spaces. Such an arrangement contains exactly k cells among which we have distributed the n indistinguishable balls.

Thus, we have obtained a composition of n into k parts. Therefore, the number of such compositions $C(n,k)$ is given by

$$C(n,k) = \binom{n+k-1}{n} = \binom{n+k-1}{k-1}$$

The process described above is equivalent to finding all the possible $k-1$ - subsets from a $n+k-1$ subset. The following algorithm describes a way of generating k - subsets from a n - set in a lexicographic order. For example, we list below the 2 - subsets from a 3 - set in a lexicographic order.

1 2

1 3

2 3

Let a_1, a_2, \dots, a_k be a given k - subset. Then all the k - subsets are generated in a lexicographic order by the following algorithm.

Algorithm 1:

Al.1: (Initial set) $a_j \leftarrow j$ ($j=1, k$); go to Al.2.

Al.2: (Subsequent sets) $h \leftarrow \min \{j/a_{k+1-j} \neq n+1-j, j=1, k\}$.

$m_1 \leftarrow a_{k+1-h}; a_{k+1-h} \leftarrow m_1+j$ ($j=1, h$),

if $a_1 = n-k+1$, stop.

Let us now illustrate the above algorithm with example 1.

Example 1: All 3 - subsets from a 5-set.

Al.1: Produces $a_1=1, a_2=2, a_3=3$.

Al.2: step 1: $h = 1; m_1 = a_{3+1-1} = 3; a_3 = 4$.

step 2: $h = 1; m_1 = 4; a_3 = 5$

step 3: $h = 2; m_1 = 2; a_2 = 3, a_3 = 4$, etc.

step 9: $h = 3; m_1 = 2; a_1 = 3; a_2 = 4; a_3 = 5$

since $a_1 = n-k+1 = 3$. stop.

Now let us proceed to generating compositions in a lexicographic order using the algorithm 1 by finding all the possible $k-1$ - subsets from a $n+k-1$ - set. When one translates the language of subsets into the language of composition, A1 modifies to A2.

Algorithm 2:

A2.1: (Initial entry): $t_1 \leftarrow n$; $t_i \leftarrow 0$ (for $i=2, k$); go to A2.2

A2.2: (Subsequent entries): $h \leftarrow \min \{i/t_i \neq 0\}$; $r \leftarrow t_h$; $t_h \leftarrow 0$; $t_1 \leftarrow r-1$;
 $t_{h+1} \leftarrow t_{h+1} + 1$; if $t_k=n$, stop.

The algorithm 2 is illustrated with example 2.

Example 2: Compositions of 3 into 3 parts.

A2.1: $t_1=3, t_2=0, t_3=0$. Produces the ordered vector $(3,0,0)$

A2.2: step 1: $h=1; r=t_1=3; t_1=2, t_2=1$. Produces $(2,1,0)$

step 2: $h=1; r=2, t_1=1, t_2=2$. Produces $(1,2,0)$

step 3: $h=1; r=1, t_1=0, t_2=3$. Produces $(0,3,0)$

step 4: $h=2; r=3, t_2=0, t_1=2, t_3=1$. Produces $(2,0,1)$. etc.,

step 9: $h=2; r=1, t_2=0, t_1=0, t_3=3$. Produces $(0,0,3)$.

The algorithm A2 produces all possible terms in the multinomial expansion

$(x_1 + x_2 + \dots + x_k)^n$. A typical composition vector (a_1, a_2, \dots, a_k) such that $\sum_{i=1}^k a_i = n$, corresponds to the term $x_1^{a_1} x_2^{a_2} \dots x_k^{a_k}$ in this multinomial expansion.

B. VECTOR Algorithm

The algorithm which we call "VECTOR" essentially picks up the compositions generated by A2 and "multiplies" them and collects the coefficients and the overall vector in an expansion of the form

$$(a_1^{b_1} + a_2^{b_1} + \dots + a_k^{b_1})^{n_1} (a_1^{b_2} + a_2^{b_2} + \dots + a_k^{b_2})^{n_2} \dots$$

Let $\max n_1, n_2, \dots$ be m . Then the number of terms in the above expansion

is given by the number of compositions of m into k parts. This algorithm calls A2 for each term in the product with $n=n_i$, obtains the composition vector for each term, the coefficient for each term and the overall coefficient and the vector. Let the number of terms in the above product be t . Algorithm VECTOR generates the overall vector array $vec(i,j)$, $i=1, k$ and $j=1, \prod_{i=1}^t n_i$ and the corresponding coefficients $c(j)$ in the expansion. The vector array is the total vector generated as a result of multiplication of all the terms in the expansion

$$\prod_{i=1}^t \left(\sum_{j=1}^k a_j^{b_i} \right)^{n_i}$$

For example, a typical vector (l_1, l_2, \dots) stands for the term $a_1^{l_1} a_2^{l_2} \dots$ in the above expansion. The resulting coefficient is a product of several multinomial coefficients.

C. Poly Algorithm

Poly is the main algorithm which calls the VECTOR algorithm for each term in the GCCI. This algorithm obtains the vector for each term in the GCCI and matches it with the initial vector generated by VECTOR for the initial term $(a_1 + a_2 + \dots + a_k)^n$. When there is a match it adds the product of the corresponding coefficient in the GCCI and the appropriate coefficient generated by vector to the already existing inventory of coefficients. When Poly exhausts all the terms in the GCCI, the resulting vector and the corresponding coefficients give the generating function with the vector (l_1, l_2, \dots) representing the term $a_1^{l_1} a_2^{l_2} \dots$. Then it calls an algorithm SPIN described below which generates the spin species from the generating function.

D. SPIN Algorithm

SPIN algorithm takes the generating function and the value of the nuclear spin quantum number for the states whose weights are in the

generating function and gives the nuclear spin species. Let the spin values of the states whose weights are a_1, a_2, \dots, a_k be m_1, m_2, \dots, m_k . Then the total spin quantum number of a term $a_1^{b_1} a_2^{b_2} \dots a_k^{b_k}$ is $m_1 b_1 + m_2 b_2 + \dots + m_k b_k$. The corresponding coefficient in the generating function gives the frequency of occurrence of the irreducible representation Γ in the set of spin functions with the total spin quantum number $m_1 b_1 + m_2 b_2 + \dots + m_k b_k$. The spin algorithm first sorts the coefficients in the generating function in accordance to their total spin quantum number. Let C_j be the array of such coefficients sorted in accordance to the total spin quantum number. Let λ_i be the multiplicity of the i^{th} species and let d_i be the corresponding frequency of occurrence. Then the algorithm described below generates the nuclear spin species from the sorted coefficients. Let n be the total number of nuclei in the molecule whose spin species are under consideration.

D1. Algorithm Spin

For Bosons

```

    → for i=1, n, do
        λi ← 2i-1
        j ← k+2-i
        if i=1, dj ← cj; else dj ← cj-cj+1
    → final exit
    
```

for Fermions

if n is odd, $f \leftarrow n+1$.

```

    → for i=2, f, 2, do
        λi ← i
        j ← n+3-i
        if i=2, dj ← cj; else dj ← cj-cj+2
    →
    
```

```
final exit
if n is even, f ← n+1
for i=1, f, 2, do
  λi ← i
  j ← n+2-i
  if i=1, dj ← cj; else dj ← cj-cj+2
final exit.
```

E. A complete example: A₁ proton spin species of rigid butane.

Algorithm 'POLY' with the help of VECTOR and A2 generate the total generating function corresponding to the A₁ representation of the C_{2v} group which is shown below.

<u>Coefficient</u>	<u>Vector</u>
1	(10, 0)
3	(9, 1)
15	(8, 2)
32	(7, 3)
60	(6, 4)
66	(5, 5)
60	(4, 6)
32	(3, 7)
15	(2, 8)
3	(1, 9)
1	(0,10)

where, for example, the vector (7,3) stands for $\alpha^7 \beta^3$ (α and β being the weights of the spin states α and β of protons) and the corresponding coefficients in the generating function is 32. The algorithm spin now sorts the coefficients according to their total spin value. In this

in the G.F. and the array c_j . Now algorithm D1 is called for fermions with $n=6$, D1 sets $\lambda_1=1, \lambda_2=3, \dots, \lambda_6=11$. The corresponding d_j 's are $d_1=6, d_2=28, d_3=17, d_4=12, d_5=2$ and $d_6=1$. Thus the proton species which correspond to A_1 representation are

$${}^1A_1(6), {}^3A_1(28), {}^5A_1(17), {}^7A_1(12), {}^9A_1(2), \text{ and } {}^{11}A_1(1).$$

Acknowledgements

The author thanks Professor Kenneth S. Pitzer for his encouragement. This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division of the U.S. Department of Energy under contract number W-7405-ENG-48.

References

1. M. Randic, *Computers and chemistry*, 3, 5 (1979).
2. M. Randic, *J. Am. Chem. Soc.*, 97, 6609 (1975).
3. M. Randic, *J. Computational Chem.*, 1, 386 (1980)
4. K. Balasubramanian, J.J. Kaufman, W.S. Koski and A.T. Balaban,
J. Computational Chem. 1, 149 (1980).
5. G. Herzberg, "Molecular spectra and molecular structure: Electronic spectra and Electronic structure of polyatomic molecules," vol. 3,
(D. Van Nostrand Co., Inc. 1967).
6. P. Bunker, "Molecular symmetry and spectroscopy" (Academic, N.Y. 1979).
7. G. Placzek and E. Teller, *Z. Physik*, 81, 208 (1933).
8. E.B. Wilson, Jr, *J. Chem. Phys.*, 3, 276 (1935).
9. E.B. Wilson, Jr, *J. Chem. Phys.*, 6, 740 (1938).
10. M. Mizushima, *J. Chem. Phys.*, 21, 1222 (1953).
11. J.T. Hougen, *J. Chem. Phys.*, 37, 1433 (1963).
12. H.W. Galbraith, *J. Chem. Phys.*, 68, 1677 (1978).
13. A. Weber, in "Raman spectroscopy of gases and liquids", edited by A. Weber,
(Springer-Verlag, 1979), pp. 71-121.
14. A. Weber, *J. Chem. Phys.*, 73, 3952 (1980).
15. K. Balasubramanian, *J. Chem. Phys.* (in press).
16. K. Balasubramanian, *J. Chem. Phys.* 72, 665 (1980).
17. K. Balasubramanian, *Theor. Chim. Acta.* 51, 37 (1979).
18. K. Balasubramanian, *Theor. Chim. Acta.* 53, 129 (1979).
19. K. Balasubramanian, *J. Chem. Phys.* 73, 3321 (1980).
20. S.G. Williamson, *J. Combinatorial Theory*, 8, 162 (1970).
21. R. Merris, *Linear Algebra and its application*, 29, 255 (1980).
22. A. Nijenhuis and H.S. Wilf, "Combinatorial Algorithms" Academic,
N.Y. 1975.

