KFKI-76-47

J. MINK
G. KEMÉNY
L.M. MINK

COMPUTER PROGRAMS FOR VIBRATIONAL SPECTROSCOPY IN ALGOL

I. THE CALCULATION OF THE G MATRIX

Hungarian Academy of Sciences

CENTRAL
RESEARCH
INSTITUTE FOR
PHYSICS

BUDAPEST

KFKI-76-47

COMPUTER PROGRAMS FOR VIBRATIONAL SPECTROSCOPY IN ALGOL

I. THE CALCULATION OF THE G MATRIX

J. Mink and G. Kemény
Institute of Isotopes of the Hungarian Academy of Sciences, Budapest
L.M. Mink
INFELOR, Institute of System Engineering, Budapest

ISBN 963 371 162 2

ABSTRACT

An ALGOL program is described, which calculates the so-called G-matrix, /kinetic energy matrix/ necessary to calculate the normal vibrations of molecules. Our aim was to achieve a relatively simple and convenient data input.

With the help of systematic checking of input data, partial results and output matrix elements the possibility of erroneous calculation is reduced.

АННОТАЦИЯ

В настоящей работе приводится программа, разработанная на языке АЛГОЛ для расчёта элементов матрицы кинематических коэффициентов, так наз. матрицы T^{-1}, необходимых при расчёте частот нормальных колебаний молекул.

Основным принципом при разработке алгоритма и написании программы являлись простота и удобство подготовки исходных данных.

С помощью систематического контроля исходных данных, промежуточных результатов и элементов полученной матрицы T^{-1} мы стремились по возможности исключить наличие ошибок в конечных результатах.

KIVONAT

Molekulák normálrezgéseinek elméleti számitásához szükséges kinetikusenergia-mátrix, az un. G-mátrix, számitógéppel történő összeállitására alkalmas, ALGOL nyelvü programot ismertetünk. Az algoritmus és a program összeállitásakor a bemenő adatok egyszerü és kényelmes előkészitésére törekedtünk.

A kiinduló adatok, a részeredmények és a kapott mátrixelemek szisztematikus gépi ellenőrzésével az eredmények hibalehetőségét minimálisra igyekeztünk csökkenteni.

CONTENTS

1. INTRODUCTION

The vibrational frequencies and the form of normal vibrations of a molecule are characteristic of two molecular features:

/i/ the masses of atoms in the molecule together with the geometrical parameters, i.e. interatomic distances and valence angles,

/ii/ the force field, which tends to restore the molecule to equilibrium configuration during vibrational motions.

The atomic masses and equilibrium geometry of the molecule are generally available for molecules of interest. The force field (which arises from changes in the energy of the electrons during vibrational motion) is, in practice, determined by calculation from the observed vibrational frequencies measured in the infrared and Raman spectra.

Force constants are strongly characteristic of the nature of chemical bonds and from the known force field of smaller chemical groups it is possible to predict and explain the vibrational spectra of larger molecules. This can be a great aid in understanding the molecular structure of large molecules.

In the interpretation of absolute intensity studies of vibrational bands it is essential to know the form of the normal coordinate associated with each vibrational frequency - which also implies a detailed knowledge of the force field.

In order to calculate the force constants from experimental data or to calculate vibrational frequencies and normal modes of vibrations or any other data associated with the vibrational motion of the nuclei, the first step is to determine the so-called G matrix - the kinematic matrix - of the given molecule. The G matrix includes information on the molecular structure namely the atomic masses and the geometrical arrangement of the nuclei.

For this reason in our series of papers we wish to discuss first the calculation of the G matrix.

A general determination of the G matrix elements was given by Wilson, Decius and Cross [1].

Nowadays the G matrix is determined almost wholly by computer algorithms. In each of the calculation methods the so-called B matrix is produced, which fulfils the transformation from the Descartes coordinates into the internal coordinates. From the B matrix it is easy to obtain the G matrix

$$B\, M^{-1}\, \tilde{B} = G, \qquad /1/$$

where M^{-1} is the diagonal matrix of the reciprocal atomic masses.

The most widely used algorithm is that of Schachtschneider [2]. The reason for this is that the program is well organized, easily accessible, and adaptable to all computers having a FORTRAN compiler.

Overend and Scherer [3] also show a program which utilises the B matrix as input data. A somewhat more complete program is described in Refs. [4] and [5]. Koster and Freeman [6] describe an ALGOL-60 program based on a similar algorithm to that of Gribov [7,8].

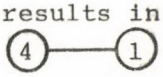
The program to be described here is based mainly on the algorithm presented by Gribov [7,8]. This algorithm was extended for most types of out-of--plane coordinates and its ALGOL-60 version was presented [9]. On the basis of experience gained in the use of these program they have been substantially modified and extended. Extensions mainly concern data preparation, error checking, and control over the reliability of the G matrix values.

2. ALGORITHM

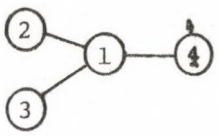
The geometry of a molecule can be given in terms of the Cartesian coordinates of the atoms of the molecule. In the present program, however, it seemed more desirable to characterise molecular geometry by unit vectors along the chemical bonds and the set of atomic distances. The reason for this is partly that the determination of the x,y,z components of the unit vectors are in most cases easier than atomic coordinates. Our program can handle the following types of internal coordinates:

/i/ valence coordinates, which are the displacements of the atoms along the bond direction related to the equilibrium state;

/ii/ bending coordinates, which are the measure of angle deviation during vibration compared to the equilibrium state;

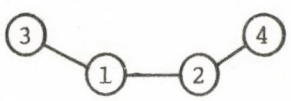
/iii/ out-of-plane coordinates, where an atom is moving relative to the plane defined by three other atoms. This results in the deformation of the angle between the bond 

and the plane of 1-2-3, viz.

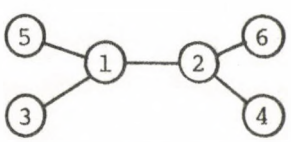


This type of internal coordinate is subsequently called ρ.

/iv/ χ-type torsional coordinates. These are also out-of-plane type coordinates with a different arrangement of the atoms. These express the displacements of atoms 3 and 4 from the planes of 3-1-2 and 1-2-3, respectively:



/v/ χ'type torsional coordinates. These involving six atoms in the following arrangement:



Here we have the deformation of the angle between two planes defined by the right and left side three-three atoms

The last two types, /iv/ and /v/, can be looked upon as the angle deformation of two normal vectors of the corresponding planes. In case /iii/ one normal vector is involved.

The x,y and z components of the unit vectors and normal vectors are included in the EIN matrix*

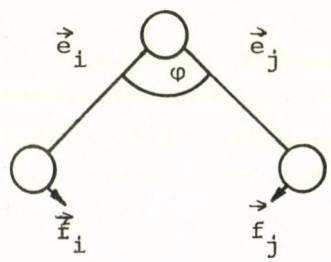
*Notations used in the text are identical to those used in the computer program.

Bending coordinates are described by unit vectors rectangular to the bond and situated in the same plane. These vectors are defined by the following equations in the cases of

$$\vec{f}_i = \frac{1}{\sin\varphi}\,\vec{e}_j - \frac{\cos\varphi}{\sin\varphi}\,\vec{e}_i \qquad /2/$$

$$\vec{f}_j = \frac{1}{\sin\varphi}\,\vec{e}_i - \frac{\cos\varphi}{\sin\varphi}\,\vec{e}_j \qquad /3/$$

Notations are illustrated in the figure below:



The matrix of the f vectors are gained by multiplying the EIN matrix and an appropriate F matrix.

The columns of this F matrix follow the order of valence coordinates and normal vectors of EIN, its rows are according to bending, out-of-plane and torsional coordinates, so that two rows belong to each coordinate. The values in the F matrix are the coefficients expressed by /2/ and /3/ in the proper places so that upon multiplication (F*EIN), they result in $\bar{f}_i$ and $\bar{f}_j$. Coordinates described by normal vectors contain 1 in the columm of the corresponding vector to bring its value unchanged into the product. The coordinates that can be described by one normal vector, however, possess two rows to result in the same number of input data for each coordinate.

The structure of F matrix is thus

	$\bar{e}_1$	$\bar{e}_i$	$\bar{e}_j$		$\bar{n}_1$	$\bar{n}_2$	
α_i	0 ...- 1/sinφ ..	cosφ/sinφ	..		0	0	angle deformations
	0 ... cosφ/sinφ ..	-1/sinφ	..		0	0	
	.	.	.		.	.	
	.	.	.		.	.	
ρ_k	0 ...	0	0	..	1	0	
	0 ...	0	0	..	1	0	
	.	.	.		.	.	
	.	.	.		.	.	
χ_1	0 ...	0	0	..	0	1	torsional and out-of-plane coordinates
	0 ...	0	0	..	0	1	
	.	.	.		.	.	
	.	.	.		.	.	
χ'_m	0 ...	0	0	..	1	0	
	0 ...	0	0	..	0	1	

It is very important to point out that /1/ and /2/ are valid for an angle ($\varphi \neq 180;$) with both bond unit vectors pointing away from the tip of the angle. Thus, if one or both vectors happen to point inwards, the corresponding columns in this coordinate should change their sign.

The result of F~~*~~EIN product is a matrix with columns x,y,z

$$\begin{vmatrix} f_{1x} & f_{1y} & f_{1z} \\ f_{2x} & f_{2y} & f_{2z} \\ . & . & . \\ . & . & . \\ . & . & . \\ f_{ix} & f_{iy} & f_{iz} \\ n_{1x} & n_{1y} & n_{1z} \\ n_{jx} & n_{jy} & n_{jz} \end{vmatrix}$$

On the basis of this matrix the EFK matrices are built by the program.

EFK /K = x,y,z/ matrices are the following:

	r_1	r_2	r_n	α_1			α_n			ρ_n			χ_n			χ'_n	
r_1	e_{1k}	0 ...	0	0	0	...	0	0	...	0	0	...	0	0	...	0	0
r_2	0	e_{2k}	0	0	0		0	0		0	0		0	0		0	0
.	.	.	.	.	.		.	.		.	.		.	.		.	.
.	.	.	.	.	.		.	.		.	.		.	.		.	.
.	.	.	.	.	.		.	.		.	.		.	.		.	.
r_n	0	0 ...	e_{nk}	0	0		0	0	...	0	0	...	0	0	...	0	0
α_1	0	0 ...	0	f_{1k}	f_{2k}	...	0	0	...	0	0	...	0	0	...	0	0
.							0	0		.	.		.	.		.	.
.								.		.	.		.	.		.	.
.										0	0	...	0	0	...	0	0
α_n							$f_{n-1,k}$	$f_{n,k}$		.	.		.	.		.	.
ρ_n										n_{1k}	n_{1k}	...	0	0	...	0	0
.													.	.		.	.
.													.	.		.	.
χ_n													n_{1k}	n_{1k}	...	0	0
.																.	.
.																.	.
χ'_n																n_{1k}	n_{2k}

The atomic distances of the molecule are brought in by a matrix product which in turn results in the mentioned B matrix also having three components corresponding to x,y,z Cartesian coordinates. SIG is the matrix of the inverse atomic distances EFK * SIG = B_k /k = x,y,z/.

The columns of SIG are arranged corresponding to the atoms of the molecule its rows are single at all valence coordinates and double at the bending and torsional coordinates.

In the valence part we find -1 in the column of the atom, where the valence coordinate starts, and +1 under the atom to where the coordinate vector points.

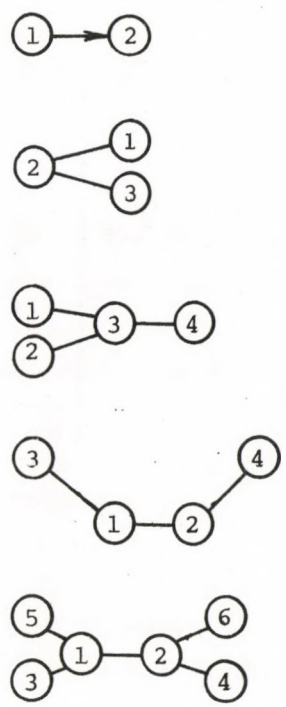
In the pairs of rows corresponding to the bending coordinates the reciprocal values of atomic distances are found with positive sign at the central atoms, and negative sign at the external atoms. Elements of the

torsional and out-of-plane coordinates are more complex and belong to the four or six atom-columms forming that coordinate.

The SIG matrix is presented below.

Number of atoms

	1	2	3	4	5	6
r	-1	1	0	0	0	0
	.	.	.	.	.	.
	.	.	.	.	.	.
α	$-\sigma_{12}$	σ_{12}	0	0	0	0
	0	σ_{23}	$-\sigma_{23}$	0	0	0
	.	.	.	.	.	.
	.	.	.	.	.	.
ρ	$-a_1$	$-b_1$	$a_1 + b_1$	0	0	0
	0	0	σ_{34}	$-\sigma_{34}$	0	0
	.	.	.	.	.	.
	.	.	.	.	.	.
χ	a_2	$-c_2$	c_2	$-a_2$	0	0
	d_2	b_2	$-d_2$	0	$-b_2$	0
	.	.	.	.	.	.
	.	.	.	.	.	.
χ'	a_3	$-b_3$	0	b_3	0	$-a_3$
	0	c_3	$-d_3$	0	d_3	0



where, in the case of $\varphi_{ij} \neq 180^{o}$,

$$a_1 = \sigma_{13} \frac{\sin \varphi_{24}}{\sin \varphi_{12}} \; ; \qquad b_1 = \sigma_{23} \frac{\sin \varphi_{14}}{\sin \varphi_{12}}$$

$$a_2 = \frac{\sigma_{13}}{\sin \varphi_{23}} \qquad b_2 = \frac{\sigma_{24}}{\sin \varphi_{14}}$$

$$c_2 = \sigma_{12} \frac{\cos \varphi_{23}}{\sin \varphi_{23}} \qquad d_2 = \sigma_{12} \frac{\cos \varphi_{14}}{\sin \varphi_{14}}$$

$$a_3 = \sigma_{15} \frac{\cos \varphi_{23}}{\sin \varphi_{35}} \qquad b_3 = \sigma_{13} \frac{\cos \varphi_{25}}{\sin \varphi_{35}}$$

$$c_3 = \sigma_{26} \frac{\cos \varphi_{14}}{\sin \varphi_{46}} \qquad d_3 = \sigma_{24} \frac{\cos \varphi_{18}}{\sin \varphi_{46}}$$

With the help of the thus obtained B matrices relation /4/ produces the G matrix in terms of internal coordinates.

$$B_x M^{-1} \tilde{B}_x + B_y M^{-1} \tilde{B}_y + B_z M^{-1} \tilde{B}_z = G \,, \qquad /4/$$

where M^{-1} is a diagonal matrix containing the reciprocal atomic masses

$$M^{-1} = \begin{vmatrix} 1/m_1 & & & & 0 \\ & 1/m_2 & & & \\ & & 1/m_3 & & \\ & & & \ddots & \\ 0 & & & & 1/m_n \end{vmatrix}$$

If, at the same time, the G matrices of isotopically substituted molecules are to be computed relation /4/ is the first stage of computation of the isotopic cycle with the same B_x, B_y, B_z matrices for all isotope-labelled molecules.

After the symmetry transformation of the G matrix in terms of internal coordinates the G matrix in terms of symmetry coordinates is obtained

$$G_s = \tilde{U} G U. \qquad /5/$$

In the case of isotopically substituted molecules the symmetry - transformation matrix /$\tilde{U}$/ may also differ, consequently this procedure is also carried out within the isotopic cycle.

3. PROGRAM USAGE

1. DESCRIPTION OF DATA INPUT

The following data are in serial order without FORMAT control according to ALGOL input; this fact on the one hand makes data input easy; on the other hand, it increases the hazard that a shift of a number remains unnoticed. Computational checks built into this program virtually eliminate these possible mistakes.

INPUT

VARIABLE	TYPE, DIMENSIONS	DESCRIPTION
NCALC	INTEGER	Number of calculations in one run, Following data contain all parameters except NCALC
KULCS	INTEGER	0: Results of partial calculations are not printed 1: F,EIN*F, B_x, B_y, B_z matrices are also printed
N	INTEGER	Number of atoms
NN	INTEGER	Number of defined normal vectors
NCB	INTEGER	Number of valence coordinates
NYR	INTEGER	Number of bending coordinates
NG	INTEGER	Number of torsional /χ, χ'/ and out-of--plane /ρ/ internal coordinates.
NIZ	INTEGER	Number of isotopically substituted molecules. The group of data, called "isotopic cycle" is NIZ times.
RF	INTEGER	Number of nonzero elements of the F matrix. 4 numbers belong to each coordiante.
EIN	REAL ARRAY	X,Y,Z components of the unit vectors, Dimensions of the EIN matrix /1:(NCB+NN), 1:3/
F	REAL ARRAY /1:RF/	Nonzero elements of the F matrix in serial order. Dimensions of the F matrix are /1:2*(NYR + + NG), 1:(NCB + NN)/

IF	INTEGER ARRAY /1:RF/	Column indices of the nonzero elements of F. Row indices are generated by the program as there are two elements in each row.
RSIG	INTEGER	Number of the nonzero elements of the SIG matrix.
SIG	REAL ARRAY /1:RSIG/	Nonzero elements of SIG matrix. Dimensions of SIG are /1:NCB+2*(NYR+NG), 1:N/ Data should follow the serial order in the matrix.
ISIG	INTEGER ARRAY /1: 2*RSIG/	Row and column indices of SIG nonzero elements.

ISOTOPIC CYCLE

NAME OF ISOTOPE LABELLED COMPOUND

The name can be of any length, ending with the word END.

NRED	INTEGER	Number of redundancies. In this sense this program is able to do a redundancy check on those G matrix elements the sum of which gives zero because of the linear dependence of internal coordinates. This condition is easily obtained at node angles. Thus NRED is the number of such nodes.
NKL	INTEGER	Number of symmetry blocks of the $\tilde{U}$ matrix in this isotopic cycle.
EI	REAL ARRAY /1:N/	Reciprocal values of the masses of the atoms in the given molecule.
INDRED	INTEGER ARRAY /1:2 * NRED/	First and last column indices of the redundancies to be checked.
RC	INTEGER	Number of nonzero elements of the $\tilde{U}$ matrix.
SIZE	INTEGER	Size of the G matrix after the symmetry-transformation, i.e. the number of rows in $\tilde{U}$.
U	REAL ARRAY /1:RC/	Nonzero elements of the $\tilde{U}$ matrix. Dimensions of this matrix are /1:SIZE, 1:NCB + NYR + NG/

INDU	INTEGER ARRAY /1:2 X RC/	Row and column indices of the elements of $\tilde{U}$.
GRKL	INEFER ARRAY /1:NKL/	Row indices of the first rows of the symmetry transposed G matrix block.

2. DESCRIPTION OF OUTPUT

In case of KULCS = 1 partial results are omitted from the output, i.e. F, F ✻ EIN, B matrices are not printed.
If KULCS = 1, the order of output is as follows:

N, NN, NCB, NYR, NG, NIZ*

ERROR MESSAGE 1.**

F ✻ EIN
EFX
EFY
EFZ
ERROR MESSAGE 2.
BX
BY
BZ
NAME OF COMPOUND
VALUES OF REDUNDANCIES
ERROR MESSAGE 3.
G MATRIX
ERROR MESSAGE 4.
ERROR MESSAGE 5.

GS MATRIX

If the symmetry transformation proved to be successful, only the upper triangle, otherwise the whole GS matrix is printed.

It cannot be emphasised enough that during the setting up of the input matrices the order of internal coordinates, normal vectors, and atom numbering must be consistent so that the various matrices remain compatible upon multiplication.

The order within the double rows of F and SIG matrices belonging to two sides of the angles must also follow our convention /the first row must be the smaller number or the left side of the angle, etc.../. This also concerns those torsional coordinates where two different normal vectors are in the two rows.

---

* For description of variables see 3.1

**For interpretation of error messages see 3.3

A very useful feature of the program is that it checks all possible regularities during the calculation, if any is not fulfilled an error message is generated on the line printer, if possible localising the place of error occurrence. This largely eases debugging and also provides a high degree of certainty that data passing all checks are errorfree.

The error messages and possible errors are presented below:

ERROR MESSAGE 1.: "UNIT VECTOR ERROR IN ROW N"

ERROR : Upon multiplication of the F and EIN matrices the product should contain unit vector elements. This property is missing from the N-th row.

ERROR MESSAGE 2.: "BX MATRIX ERROR IN ROW N"
"BY MATRIX ERROR IN ROW N"
"BZ MATRIX ERROR IN ROW N"

ERROR : The sum of N-th row in the B matrix is not equal zero

ERROR MESSAGE 3.: "REDUNDANCE ERROR"

ERROR : One of the redundancy conditions in the G matrix in terms of internal coordinates is not satisfied. The erroneous redundancy can easily be found in the table of all values of redundancies printed before the G matrix.

ERROR MESSAGE 4.: "SYMMETRY ERROR"

ERROR : The G matrix is unsymmetrical.

ERROR MESSAGE 5.: "SYMMETRY TRANSFORMATION ERROR"

ERROR : The symmetry transformed G matrix is not of quasi-diagonal structure, there are nonzero elements outside the blocks. In this case the whole matrix is printed.

The hitherto described G matrix-elements are not weighted, a weighting can be carried out if necessary for example by weighting all atomic distances in the SIG matrix.

It should be mentioned that a STANFORD ALGOL version is also operative at the University, Bristol, and a simplified version on BESM-4, the computer of the Soviet Academy of Sciences, Moscow.

3. DISCUSSION

The program utilises only core memory, however it is able to compute the G matrices of molecules, or molecular systems, up to about 24 atoms.

When designing the input we attempted to simplify it as much as possible because this also decreases the possibility of errors.

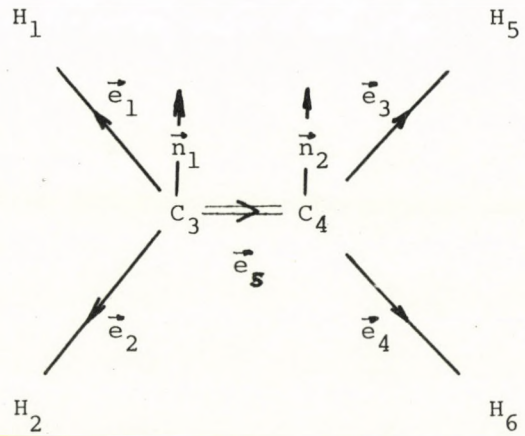
The interpretation of the geometry of a molecule seems to be more feasible in terms of unit vectors than in Descartes coordinates of the atoms, furthermore the unit vectors can easily be determined from Cartesian coordinates of the atoms, but the reverse is far more difficult.

This is an advantage especially with complicated molecules, or when using an exact molecular geometry of distorted tetrahedral or other angles.

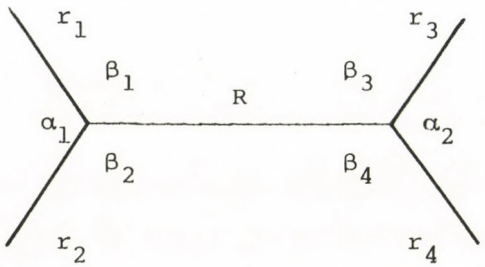
In order to help data preparation, which is justifiable rather by the possibility of mistakes than by the complexities of the calculus, a number of simple programs were written for a minicomputer /TPA 1001i/.It is possible to calculate the EIN matrix from the x,y,z coordinates of the atoms, to produce the F matrix, and to compute the torsional and out-of-plane coordinate components of the SIG matrix.

4. TEST RUN, THE G MATRIX OF ETHYLENE

To test the program we chose a relatively simple molecule, ethylene and its deuterated derivative. The choice of atom numbering and unit vectors are the following:



The in-plane internal coordinates are shown below:



In addition to these, two ρ-type out-of-plane coordinates and a χ′ type were introduced. The ρ_1 and ρ_2 coordinates belong to the out-of-plane movement of the C=C bond from the planes of H_1-C_3-H_2 and H_5-C_4-H_6, respectively. χ′ is an internal coordinate concerning all six atoms in a way described above.

R_{cc} = 135.3pm, r_{CH} = 107.1pm, were used, all angles were taken to be 120°.

The order of the internal coordinates, used for constructing the input matrices:

$$r_1,\ r_2,\ r_3,\ r_4,\ R,\ \alpha_1,\ \beta_1,\ \alpha_2,\ \beta_2,\ \beta_3,\ \beta_4,\ \rho_1,\ \rho_2,\ \chi' .$$

The final G matrix in terms of symmetry coordinates of ethylene are presented below.

A_{1g} $S_1 = 1/2(r_1+r_2+r_3+r_4)$

$S_2 = R$

$S_3 = 1/2(\beta_1+\beta_2+\beta_3+\beta_4)$

B_{1g} $S_4 = 1/2(r_1-r_2-r_3+r_4)$

$S_5 = 1/2(\beta_1-\beta_2-\beta_3+\beta_4)$

B_{2g} $S_6 = 1/\sqrt{2}(\rho_1-\rho_2)$

A_{1u} $S_7 = \chi'$

B_{1u} $S_8 = 1/\sqrt{2}(\rho_1+\rho_2)$

B_{2u} $S_9 = 1/2(r_1-r_2+r_3-r_4)$

$S_{10} = 1/2(\beta_1-\beta_2+\beta_3-\beta_4)$

B_{3u} $S_{11} = 1/2(r_1+r_2-r_3-r_4)$

$S_{12} = 1/2(\beta_1+\beta_2-\beta_3-\beta_4)$

DATA LISTING

```
1
1
6
2
5
6
3
2
36
-0.5,0.866026,0
-0.5,-0.866026,0
0.5,0.866026,0
0.5,-0.866026,0
1,0,0
0,0,1
0,0,-1
```

EIN

```
0.57735,1.1547
1.1547,0.57735
0.57735,1.1547
1.1547,0.57735
0.57735,1.1547
1.1547,0.57735
0.57735,1.1547
1.1547,0.57735
0.57735,-1.1547
1.1547,-0.57735
0.57735,-1.1547
1.1547,-0.57735
0,1
0,1
0,1
0,1
1,0
0,1
```

F

```
1,2  1,2  1,5  1,5  2,5  2,5  3,4  3,4
3,5  3,5  4,5  4,5  5,6  5,6  5,7  5,7
6,7  6,7
52
1,-1
1,-1
-1,1
-1,1
-1,1
-0.93371,0.93371
-0.93371,0.93371
-0.93371,0.93371
0.7391,-0.7391
-0.93371,0.93371
0.7391,-0.7391
0.93371,-0.93371
```

SIG

```
0.93371,-0.93371
0.93371,-0.93371
-0.7391,0.7391
0.93371,-0.93371
-0.7391,0.7391
-0.93371,-0.93371,1.86746
0.7391,-0.7391
1.86742,-0.93375,-0.93375
-0.7391,0.7391
-0.53908,0.53908,0,0
0,0,0.53908,-0.53908
1,1  1,3  2,2  2,3  3,4  3,5  4,4  4,6  5,3  5,4  6,1  6,3  7,2
7,3  8,1  8,3  9,3  9,4  10,2  10,3  11,3  11,4  12,4  12,5  13,4
13,6  14,4  14,5  15,3  15,4  16,4  16,6  17,3  17,4  18,1  18,2
18,3  19,3  19,4  20,4  20,5  20,6  21,3  21,4  22,1  22,2  22,3
22,4  23,3  23,4  23,5  23,6
```

ISOTOPIC CYCLE

```
C2H4 END
2
7
0.992196,0.992196,0.083333,0.083333,0.992196,0.992196
6,8  9,11
38
12
0.5,-0.5,0.5,-0.5
0.5,-0.5,0.5,-0.5
0.5,0.5,-0.5,-0.5
0.5,0.5,-0 5,-0.5
0.707107,0.707107
0.5,0.5,0.5,0.5
1
0.5,0.5,0.5,0.5
1
0.5,-0.5,-0.5,0.5
0.5,-0.5,-0.5,0.5
0.707107,-0.707107
```

E1

U

```
1,1  1,2  1,3  1,4  2,7  2,8  2,10  2,11  3,1  3,2  3,3
3,4  4,7  4,8  4,10  4,11  5,12  5,13  6,1  6,2  6,3  6,4
7,5  8,7  8,8  8,10  8,11  9,14  10,1  10,2  10,3  10,4
11,7  11,8  11,10  11,11  12,12  12,13

1,3,5,6,9,10,12
B2U B3U B1U A1G A1U B1G B2G
```

```
C2D4 END
2
7
0.496499,0.496499,0.083333,0.083333,0.496499,0.496499
6,8  9,11
38
12
0.5,-0.5,0.5,-0.5
0.5,-0 5,0.5,-0.5
0.5,0.5,-0.5,-0.5
0.5,0.5,-0.5,-0.5
0.707107,0.707107
0.5,0.5,0.5,0.5
1
0.5,0.5,0.5,0.5
1
0.5,-0.5,-0.5,0.5
0.5,-0.5,-0.5,0.5
0.707107,-0.707107
```

U

1,1 1,2 1,3 1,4 2,7 2,8 2,10 2,11 3,1 3,2 3,3
3,4 4,7 4,8 4,10 4,11 5,12 5,13 6,1 6,2 6,3 6,4
7,5 8,7 8,8 8,10 8,11 9,14 10,1 10,2 10,3 10,4
11,7 11,8 11,10 11,11 12,12 12,13

1,3,5,6,9,10,12
B2U B3U B1U A1G A1U B1G B2G

THE RESULTS OF THE G MATRIX CALCULATION FOR THE ETHYLENE AND PERDEUTERATED--ETHYLENE

C_2H_4

A_{1g} species

S_1	S_2	S_3
1.033864	-0.083333	0.067384
	0.166666	-0.134769
		0.973986

B_{1g} species

S_4	S_5
1.117197	-0.280744
	1.495548

B_{2g} species

S_6
2.020707

A_{1u} species

S_7
1.153357

B_{1u} species

S_8
2.662869

B_{2u} species

S_9	S_{10}
1.117197	-0.067385
	0.901336

B_{3u} species

S_{11}	S_{12}
1.033864	0.067384
	0.973986

C_2D_4

A_{1g} species

S_1	S_2	S_3
0.538166	-0.083333	0.067384
	0.166666	-0.134769
		0.541831

B_{1g} species

S_4	S_5
0.621499	-0.280744
	1.063392

B_{2g} species

S_6
1.156358

A_{1u} species

S_7
0.577145

B_{1u} species

S_8
1.798520

B_{2u} species

S_9	S_{10}
0.621499	-0.067385
	0.469180

B_{3u} species

S_{11}	S_{12}
0.538166	0.067384
	0.541831

V. PROGRAM LISTING

```
'begin'
'integer'R,N,NN,NCB,P,NYR,NG,I,J,L,M,K,P2;
'integer'NIZ,NRED,NKL,MERET,IZOTOP,S;
'integer'NCALC,COUNT;
'real'SUM; 'real' SZAM;
'boolean' ERROR1, ERROR2;
'integer'KULCS;

'procedure'VEKTKI(VEKTOR,N);
'real''array'VEKTOR;'value'N; 'integer'N;
'begin'
'integer'I,K;
K:=1; newline(1);
'for'I:=1 'step' 1 'until' N 'do'
'begin'
'if' K>10 'then'
'begin'
newline(1); K:=2;
'end''else' K:=K+1;
SZAM:=VEKTOR[I]; print(SZAM,1,6)
'end'I
'end'VEKTKI;

'procedure' TOMBKI(TOMB,N,M);
'real''array'TOMB; 'value'N,M; 'integer'N,M;
'begin'
'integer'I,J,Q;
'for'I:=1 'step' 1 'until' N 'do'
'begin' newline(1); Q:=1;
  'for' J:=1 'step' 1 'until' M 'do'
  'begin'
'if' Q>10 'then'
'begin'
newline(1); Q:=2;
'end''else'Q:=Q+1;
SZAM:=TOMB[I,J]; print(SZAM,1,6);
```

```
'end'J
'end'I
'end' TOMBKI;

'procedure' FELEKI(TOMB,N);
'real''array'TOMB; 'value'N; 'integer'N;
'begin'
'integer'I,J,K;
'for'I:=1 'step' 1 'until' N 'do'
'begin'
K:=1; newline(1);
'for' J:=I 'step' 1 'until' N 'do'
'begin' 'if' K>10 'then'
'begin'
newline(1); K:=2;
'end''else'K:=K+1;
SZAM:=TOMB[I,J]; print(SZAM,1,6);
'end'J
'end'I
'end' FELEKI ;
select input(0); select output(0);

  NCALC:=read; COUNT:=1;
 KEZD: IZOTOP:=1; KULCS:=read; N:=read; NN:=read; NCB:=read;
NYR:=read; NG:=read; NIZ:=read;
writetext('(''('p3cs')'G%
MATRIX%CALCULATION'('3c')'')');
  writetext('(''('4')'N'('6s')'NN'('4s')' NCB
'('4s')'NYR'('4s')'NG'('5s')'NIZ')');
  newline(1);
space(2); print(N,2,0); space(2); print(NN,2,0); space(2);
print(NCB,2,0); space(2); print(NYR,2,0); space(2);
print(NG,2,0); space(2); print(NIZ,2,0);
P2:=NYR+NG;
P:=2*P2;
MERET:=P2+NCB;
'begin'
'real''array'B[1:3,1:MERET,1:N];
'begin'
'real''array'EFI[1:3,1:P+NCB];
'procedure' EFCALC;
'begin'
```

```
'real''array'EIN[1:NCB+NN,1:3];
'real''array'FK[1:P,1:3];
'real''array'F[1:R];
'real''array'STR[1:NCB+NN];
'integer' 'array' IN[1:R];
'real' QQ,ZZ;
'for'I:=1 'step' 1 'until' NCB+NN 'do'
'for'J:=1 'step' 1 'until' 3'do' EIN[I,J]:=read;
'for'I:=1 'step' 1 'until' R 'do' F[I]:=read;
'for'I:=1 'step' 1 'until' R 'do' IN[I]:=read;
L:=1; 'for'K:=1 'step' 1'until' P 'do'
'begin'
'for'I:=1'step' 1 'until' NCB+NN'do' STR[I]:=0;
STR[IN[L]]:=F[L];
STR[IN[L+1]]:=F[L+1]; L:=L+2;
'for'I:=1 'step' 1'until' 3'do'
'begin'
FK[K,I]:=0;
'for'J:=1 'step' 1 'until' NCB+NN 'do' FK[K,I]:=
  FK[K,I]+STR[J]*EIN[J,I]
'end';
'end'K;
'if' KULCS=1 'then'
'begin'
writetext('('''('3c2s')'F%MATRIX'('2c')''')');
TOMBKI(FK,P,3);
newline(3);
'for' K:= 'step' 1 'until' P 'do'
'begin'
QQ:=sqrt(FK[K,1]↑2+FK[K,2]↑2+FK[K,3]↑2);
ZZ:=1.0-QQ;
'if' abs(ZZ)>.0001 'then'
'begin'
writetext('('UNIT%VECTOR%ERROR%IN%ROW%')');
print(K,3,0);
newline(1);
'end';
'end';
  'end';
  'for' K:=1 'step' 1 'until' 3 'do'
'begin'
'for'I:=1 'step' 1 'until' NCB 'do' EFI[K,I]:=EIN[I,K];
```

```
'for'I:=1 'step' 1 'until' P 'do' EFI[K,NCB+I]:= FK[I,K];
'end'K;
'end' EFCALC;

'procedure' BCALC;
'begin'
'real''array'SIG[1:R];
  'real' 'array' STR1,STR2[1:N];
'integer''array'IND[1:2*R];
'for'I:=1 'step' 1 'until' R 'do' SIG[I]:=read;
'for'I:=1 'step' 1 'until' 2*R 'do' IND[I]:=read;
'for'K:=1 'step' 1 'until' 3 'do'
'begin'
L:=1; M:=1;
'for'I:=1 'step' 1 'until' NCB 'do'
'begin'
  'for' J:=1 'step' 1 'until' N 'do' STR1[J]:=0;
MET1: 'if' IND[M]=I 'then'
'begin'
  STR1[IND[M+1]]:= SIG[L];
L:=L+1; M:=M+2; 'goto' MET1
'end';
'for'J:=1 'step' 1 'until' N 'do'
'begin'
B[K,I,J]:=EFI[K,I]*STR1[J]
'end'
'end'I;
'for'I:=1 'step' 1 'until' P2 'do'
'begin'
  'for' J:=1 'step' 1 'until' N 'do' STR1[J]:=STR2[J]:=0;
S:=NCB+2*I;
MET2:'if' IND[M]=S-1'then'
'begin'
STR1[IND[M+1]]:=SIG[L]; M:=M+2; L:=L+1;
'goto' MET2
'end';
MET3: 'if' IND[M]=S 'then'
'begin'
STR2[IND[M+1]:=SIG[L]; M:=M+2; L:=L+1;
'if' M>2*R'then''goto' FENE;
'goto' MET3
'end';
```

```
FENE: 'for'J:=1 'step' 1 'until' N 'do'
B[K,NCB+I]:=EFI[K,S-1]*STR1[J]+
  EFI[K,S]*STR2[J]
'end'I;
'end'K;
'end'BCALC;
R:=read;
EFCALC;
  'if' KULCS 'then'
'begin'
'for'K:=1 'step' 1 'until' 3 'do'
'begin'
newline(2);
'if' K=1 'then' writetext('('EFX%MATRIX')');
'if' K=2 'then' writetext('('EFY%MATRIX')');
'if' K/'then' writetext('('EFZ%MATRIX')');
J:=1; newline(2);
'for'I:=1 'step' 1 'until' NCB+P 'do'
'begin'
'if' J>10 'then'
'begin'
newline(1); J:=2
'end''else' J:=J+1;
  SZAM:=EFI[K,I]; print(SZAM,1,6);
'end'I
'end'K
'end';
R:=read;
BCALC;
'end'EFI;
BCHECK: gor'K:=1 'step'  'until' 3 'do'
'begin'
  ERROR1:='false'; M:=1;
'for'I:=1 'step' 1'until' MERET'do'
'begin'
SUM:=; 'for'J:=1 'step' 1'until' N 'do' SUM:=SUM+B[K,I,J]
'if' abs(SUM)>.0156 'then'
'begin''if' M=1 'then'
'begin'newline(3); ERROR1: true';
'if' K=1 'then' writetext('('BXMATRIX%ERROR')');
'if' K=2 'then' writetext('('BY%MATRIXERROR')');
'if' K=7 'then' writetext('('BZ%MATRIX%ERROR')');
```

```
space(3); writetext('('ROWS:'('3s')'')');
M:=2;
'end'; print(I,2,0);
'end'
'end'I
'end'K;
BOUT: 'if'(ERROR1) 'or' (KULCS=1) 'then'
'begin'
'for'K:=1'step' 'until' 3 'do'
'begin'
 newline(2);
  'if' K=1 'then' writetext('('BX%MATRIX')');
  'if' K=2 'then' writetext('('BY%MATRIX')');
  'if' K=3 'then' writetext('('BZ%MATRIX')');
newline(2);
'for'I:=1'step' 1 'until' MERET 'do'
'begin'
L:=1; newline(1);
'for'J:=1 'step' 1 'until' N 'do'
'begin'
'if' L>10 'then'
'begin'
newline(1); L:=2;
'end''else' L:=L+1;
SZAM:=B[K,I,J]; print(SZAM,1,6);
'end'J
'end'I
'end'K
'end';
'if' ERROR1 'then''goto' KI;
IZCICLE: newline(3); copytext('('END')'); newline(2);
NRED:=read; NKL:=read;
'begin'
'real''array'G[1:MERET,1:MERET];
'real''array'EI[1:N];
'for'I:=1 'step' 1 'until' MERET 'do'
'for'J:=I 'step' 1 'until' MERET 'do' G[J,I]:=G[I,J]:=0;
  'for' I:=1 'step' 1'until' N 'do' EI[I]:=read;
'for'K:1 'step' 1 'until' 3 'do'
'for'I:=1'step' 1 til' MERET 'do'
'for'M:1'step' 1 'until' MERET 'do'
'begin'
```

```
SUM:=0;
'for'J:=1 'step' 1 'until' N'do' SUM:=SUM+B[K,I,J]*EI[J]*
 B[K,M,J]; G[I,M]:=G[I,M]+SUM
'end';
GCHCK1: ERROR1:='false';ERROR2:='false';
'for'I:=1'step' 1'until' MERET 'do'
'for'J:=1 'step' 1 'until' MERET 'do'
'if' abs(G[I,J]-G[J,I])>0.0156 'then'
'begin'
ERROR1:='true'; witetext('(''('2c')'UNSYMMETRICAL%G')');
'goto'GCHCK2
'end';
GCHCK2:'if' NRED=0 'then''goto' GOUTP;
'begin'
'integer''array'INDRED[1:2*NRED]; 'for' I:=1 'step' 1'until' 2*NRED 'do'
INDRED[I]:= read;
newline(2);
writetext('('VALUES%OF%REDUNDANCES')');
newline(1);
'for' K:=1 'step'  'until' MERET 'do'
'begin' newline(1);
'for' I:=1 'step' 1 'until' NRED 'do'
'begin' SUM:=0;
'for' J:=INDRED[2*I-1]'step' 1'until' INDRED[2*I]'do'
SUM:=SUM+G[K,J];
print(SUM,3,6)
'end' I;
'end' K;
newline(2);
'for'I:=1 'step' 1 'until' NRED 'do'
'for'K:=1'step' 1 'until' MERET 'do'
'begin' SUM:=0;
'for'J:=INDRED[2*I-1] 'step' 1 'until' INDRED[2*I]'do'
SUM:=SUM+G[K,J ;
'if' abs(SUM)>.0156 'then'
'begin'
ERROR2:='true';
writetext('(''('2c')'REDUNDANCE%ERROR')');
'goto' GOUTP
'end'
'end'K;
'end'INDRED;
```

```
GOUTP: writetext('(''('3c')'G%MATRIX'('c')'')');
'if' (ERROR1) 'or' (MERET>10) 'then' TOMBKI(G,MERET,MERET)
'else' FELEKI(G,MERET);
R:=read; S:=read;
'begin''integer''array'TEXTS[1:NKL,1:4];
'real''array'GS[1:S,1:S]; 'integer' 'array' GRKL[1:NKL+1];
  'integer' 'array' IND[1:2*R];
'real''array'C[1:R];
'procedure' GSCALC;
'begin'
'real''array' CMAT[1:S,1:MERET];
'real''array' STR[1:MERET];
L:=1; M:=1;
'for'I:=1 'step' 1 'until' S 'do'
'begin'
'for'J:=1'step' 1 'until' MERET 'do' CMAT[I,J]:=0;
MET3: 'if' IND[M]=I 'then'
'begin'
  CMAT[I,IND[M+1]]:=C[L]; L:=L+1; M:=M+2;
'if' M>2*R'then''goto' KESZ; 'goto' MET3
'end'
'end'I;
KESZ:'for'I:=1 'step'1'until' S'do'
'begin'
  'for' J:=1'step' 1 'until' MERET 'do' STR[J]:=0;
  'for' J:=1'step' 1 'until' MERET 'do'
'for'K:=1'step' 1 'until' MERET 'do'
STR[J]:= CMAT[I,K]*G[K,J]+STR[J];
'for'J:=1'step' 1 'until' S 'do'
'begin'
GS[I,J]:=0;
'for'K:=1 'step' 1'until' MERET 'do'
GS[I,J]:=GS[I,J]+STR[K]*CMAT[J,K];
'end'J;
'end'I;
GSCH1: ERROR1:='false'; ERROR2:='false';
'for'I:=1 'step' 1 'until' S 'do'
'for'J:=1 'step' 1 'until' S 'do'
'if' abs(GS[I,J]-GS[J,I])>0.0156 'then'
'begin' ERROR1:='true';
writetext('(''('c')'UNSYMMETRICAL%W')');
  'goto' GSCH2
 'end';
```

```
GSCH2: GRKL[NKL+1]:=S+1;
'for'M:=1 'step' 1 'until' NKL-1 'do'
'for'I:=GRKL[M] 'step' 1 'until' GRKL[M+1]-1'do'
  'for' J:=GRKL[M+1]'step' 1 'until' S 'do'
'if'(abs(GS[I,J])>0.0156) 'or' (abs(GS[J,I])>0.0156)'then'
'begin'
writetext('('''('3C')'SYMMETRY%TRANSFORMATION%
ERROR')'); ERROR2:='true';goto' GSOUTP
'end';
GSOUTP: newline(3);'if' ERROR1 'or' ERROR2 'then'
'begin'
  writetext('('GS%MATRIX'('2C')'')');
  'if' (ERROR1) 'or' (S>10) 'then' TOMBKI (GS,S,S)
'else' FELEKI(GS,S);newline(2);
'goto' VIZSG
'end';
  KLOUT: 'for' M:=1 'step' 1 'until' NKL 'do'
'begin' newline(3);
'for'I:=1'step' 1 'until' 4 'do'
'begin' J:=TEXTS[M,I]; printch(J);
'end';
writetext('('''(2S)'SPECIES'('2C')'')');
'for'I:=GRKL[M]'step' 1'until' GRKL[M+1]-1 'do'
'begin' newline(1); K:=1;
'for'J:=I'step' 1 'until' GRKL[M+1]-1 'do'
'begin'
'if' K>10 'then'
'begin' newline(1); K:=2
'end''else' K:=K+1;
SZAM:=GS[I,J]; print(SZAM,1,6)
'end'J
'end'I
'end'M;
'end' GSCALC;
'for'I:=1'step' 1 'until' R 'do' C[I]:=read;
'for'I:=1 'step' 'until' 2*R 'do' IND[I]:=read;
'for'I:=1'step' 1 'until' NKL 'do' GRKL[I]:=read;
'for'I:=1 'step' 1 'until' NKL 'do'
'for'J:=1'step' 1'until' 4 'do'
```

```
TEXTS[I,J]:=readch;
'if' ERROR1 'or' ERROR2 'then''goto' VIZSG;
GSCALC;
'end'GS;
'end'G;
VIZSG: IZOTOP:=IZOTOP+1;
  'if' IZOTOP 'le' NIZ 'then' 'goto' IZCICLE;
KI:
'end'B;
COUNT:=COUNT+1;
'if' COUNT 'le' NCALC 'then' 'goto' KEZD; free output; free output ;
'end';
```

LITERATURE

[1] Wilson, E. B. jr., Decius, J. C., Cross, P. C., Molecular Vibrations McGraw-Hill New York, 1955

[2] Schachtschneider, J. H., Shell Development Co. Techn. Rep. 1962

[3] Overend, J., Sherer, J. R., Chem. Phys. 32, 1289 /1960/

[4] Papousek, D., Pliva, I. Collection Czechoslov. Chem. Commun., 28, 755 /1963/

[5] Long, D. A., Gravenor, R. B. Spectrochim. Acta, 19, 937 /1963/

[6] Carter, J. H., Freeman, G. L., Hentshall, T., Spectrochim. Acta, 23A 1463 /1967/

[7] Грибов, Л.А.: Введение в теорию и расчет колебательных спектров многоатомных молекул, гл. X. Изд. ЛГУ. 1965

[8] Грибов, Л.А., Жогина, В.В., Архипова, С.Ф., Ж.прикл.спектр. 3, 403 /1966/

[9] Минк, Я., Минк, Л.М., Пентин, Ю.А., Ж. прикл. спектр. 9, 1 /1968/

Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Kósa Somogyi István, a KFKI
Szilárdtestkutatási Tudományos Tanácsának
szekcióelnöke
Szakmai lektor: Jancsó Gábor
Nyelvi lektor: H. Shenker
Példányszám: 205 Törzsszám: 76-668
Készült a KFKI sokszorosító üzemében
Budapest, 1976. július hó