RICE UNIVERSITY

ASSIGNMENT OF MICROWAVE ROTATIONAL SPECTRA BY COMPUTER

by

Dorothy Jean Thomas

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Arts

Thesis Director's signature:

[Signature]

Houston, Texas

June, 1968
ABSTRACT

ASSIGNMENT OF MICROWAVE ROTATIONAL SPECTRA BY COMPUTER

Dorothy Jean Thomas

A computer program was written to assign the rotational quantum number transitions to the lines in a microwave spectrum. Lines in the spectrum of allyl alcohol were assigned using this program.
I would like to express my appreciation to Dr. R. F. Curl for his encouragement and help in this research project. I would also like to thank my husband Val and the U. S. Navy without whom this research would have been impossible.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. BASIC MICROWAVE THEORY</td>
<td>3</td>
</tr>
<tr>
<td>III. ASSIGNMENT OF LINES</td>
<td>8</td>
</tr>
<tr>
<td>IV. ASSIGNMENT OF LINES BY THE COMPUTER</td>
<td>11</td>
</tr>
<tr>
<td>V. DESCRIPTION OF THE PROGRAM</td>
<td>20</td>
</tr>
<tr>
<td>VI. PRACTICAL APPLICATION OF THE PROGRAM TO ALLYL ALCOHOL</td>
<td>22</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>25</td>
</tr>
<tr>
<td>DOCUMENTATION</td>
<td></td>
</tr>
</tbody>
</table>
I. INTRODUCTION

The microwave spectrum of a rigid rotor molecule consists of a number of absorption lines of various frequencies and intensities which correspond to transitions from one rotational energy level to another. The energies of the levels are determined by two or three rotational constants and the levels are labelled by quantum numbers. The molecule must have a dipole moment in order to produce a spectrum and the intensities of the absorption lines are determined by the dipole moment.

A symmetric top or linear molecule with a permanent dipole moment produces a spectrum which is fairly simple to analyze. However, an asymmetric molecule has a much more complex rotational spectrum because the energy levels are no longer degenerate as they are in a symmetric top. Transitions corresponding to greatly different quantum numbers may fall in the same frequency region. This complexity makes it very difficult to assign quantum numbers to observed absorption lines. The assignment of quantum numbers to a sufficient number of observed transitions to determine the rotational constants is called an assignment. Generally only a few lines in a complex spectrum are useful in making an assignment with the methods now in use and these are the lower quantum number transitions. Moreover, these methods are tedious and time-consuming.

Since the assigning of the lines in a microwave spectrum is only an intermediate but essential step in calculating other information about a molecule, such as the rotational constants and the dipole moment, it is important to provide a way to make a more complete
assignment in less time and to enable researchers to tackle the assignment of spectra which previously had been forbiddingly complex. Logically, the computer can be used to do this job.

The purpose of this research was to write a computer program which would assign the correct rotational quantum numbers to the observed lines in a microwave spectrum and to apply this program to actual unassigned spectra. Although there are standard computer programs to predict microwave spectra, to calculate rotational constants, to predict Stark and quadrupole effects, the use of the computer had not been extended to assigning the rotational quantum numbers to the spectra. The computer program on which this research is based is designed to do this assignment. The general logic of the program is to predict a set of frequencies and intensities for the rotational transitions of a molecule and to compare this predicted spectrum with the experimental spectrum of the molecule in all probable combinations. The closest match of the largest number of lines will give the correct assignment.

After testing the program on already known spectra, the assignment of an unknown spectrum was attempted. Allyl alcohol has a quite complex spectrum of which spectroscopists have assigned only a few lines\(^1\); the data was readily available so allyl alcohol was chosen. New lines were assigned and the usefulness of the program was demonstrated.
II. BASIC MICROWAVE THEORY

It has been demonstrated that the energies of molecular systems are quantized and that every different molecule has its own unique energy level structure. Since the mass of the nucleus is so much larger than that of the electron, it is possible to treat the motion of the electrons with respect to the nucleus separately from the nuclear motions relative to one another and the center of mass. The energy of a molecular system is then composed of the electronic energies, the rotation of the system of nuclear masses about the center of mass, and the vibration of the nuclear masses with respect to one another. Then the Schrödinger equation describing the nuclear kinetic and potential energies is

\[ \left[ -\frac{\hbar^2}{2} \sum_{j} \frac{1}{m_j} \nabla_j^2 + V(r_{ji}) \right] \psi = E \psi. \]  

(1)

When the harmonic oscillator function is used as an approximation to the potential energy, \( V(r_{ji}) = \frac{k}{2} (r - r_e)^2 \), the solution of the equation for a two-body problem becomes

\[ E = \frac{J(J+1)\hbar^2}{2I_e} + (v + 1/2)\hbar \nu_e \], \hspace{1cm} v = 0,1,2,..., \hspace{1cm} (2)

where \( \nu_e = 1/(2\pi)^{1/2} k \) and \( \mu \) is the reduced mass. It is obvious that the second term describes the energy levels of a non-rotating harmonic oscillator, \( v \) being the molecular vibration quantum number. The first term describes the energy levels of a rigid freely-rotating diatomic molecule where \( I_e \) is the moment of inertia of the molecule and \( J \) is the angular momentum quantum number. A more accurate solution is
obtained when a better potential function approximation is used and when higher order terms such as those dealing with centrifugal distortion and vibration-rotation interaction are included. Of course the solution of the Schrödinger equation is more complex when it deals with more than two bodies.

Rigid rotor molecules fall into three categories: (1) linear, (2) symmetric top, and (3) asymmetric top molecules. The mass distribution of any molecule can be described completely by three moments of inertia along mutually perpendicular axes through the center of mass of the molecule. These axes are called the principle axes of inertia of the molecule; they are labelled such that the moment of inertia \( I_A \) along axis 'a' is less than the moment of inertia along axis 'b', which in turn is less than the moment along axis 'c', \( I_A < I_B < I_C \). For a linear molecule one of these moments of inertia is almost zero, the other two are equal. A symmetric top also has two equal moments of inertia; an asymmetric top has none equal. The rotational constants \( A, B, C \) are defined in terms of the principle moments of inertia.

\[
A = \frac{\hbar}{4\pi I_A}, \quad B = \frac{\hbar}{4\pi I_B}, \quad C = \frac{\hbar}{4\pi I_C}
\]

For the general case of the energy level transitions of an asymmetric rigid rotor, the solution of the Schrödinger equation is expressed as

\[
E = \frac{1}{2} (A + C)J(J + 1) + \frac{1}{2} (A - C)E_{K-1K+1}^J (\hbar)
\]

where \( E_{K-1K+1}^J (\hbar) \) has been tabulated in the literature as functions of
$K_1$, $K_{+1}$, and $J$. $\chi$ is called the asymmetry parameter and is defined by

$$\chi = \frac{2B - A - C}{A - C}.$$ 

$K_1$ is the quantum number of the component of the total angular momentum along the molecular symmetry axis for a prolate top; for a given $J$, $K$ can be $0, 1, 2, \ldots, J$. Although $K_1$ and $K_{+1}$ have no real significance for an asymmetric top, they are used to label the asymmetric top energy levels.

When realistic numbers are introduced in equation (2) it is found that the energy differences corresponding to changes in $v$ are much larger than those corresponding to changes in $J$. In other words, the rotational energy levels are much more closely spaced than the vibrational energy levels. When electromagnetic radiation corresponding to the difference between a pair of energy levels is made available to the molecule, a transition from one level to the other may occur. Energies corresponding to changes of 0, +1, and -1 in the rotational quantum number $J$ fall in the far infrared and microwave regions of the spectrum (between 3 and 300 GHz). Energies corresponding to changes in the vibrational quantum number fall in the infrared regions of the spectrum. Thus, in general, rotational transitions are studied separately from vibrational transitions.

In an actual determination of the frequencies of rotational transitions a source producing electromagnetic radiation in the microwave region is used. The molecule being studied absorbs certain discrete amounts of energy which correspond to the transition from one rotational
energy level to another. The frequencies at which molecules absorb radiation are specified by the transition energies since $\nu = \frac{\Delta E}{h}$. The absorptions of microwave energy are detected by the microwave spectrometer and a spectrum is a record of absorption versus frequency.

There are certain restrictions on the changes in the quantum numbers in a transition which are called selection rules. For an asymmetric top one selection rule is $\Delta J = 0, \pm 1$. A transition where $\Delta J = 0$ is called a Q-branch transition; transitions where $\Delta J = \pm 1$ are called R and P-branch transitions. The intensities of the absorption lines depend on the permanent dipole moment because the dipole moment interacts with the electric field from the microwave source. The dipole moment can be expressed in terms of its components along the principle axes.

$$\mu^2 = \mu_a^2 + \mu_b^2 + \mu_c^2$$

(7)

Each component of $\mu$ determines the intensity of a certain class of transitions.

The intensities of absorption lines also depend on the population of the levels. When the dependence of population on $J$ is considered, it is found to increase with the increasing $J$ for small $J$, pass through a maximum and then decrease with $J$ increasing for large $J$. Therefore, the strongest lines are often those of intermediate $J$ ($J \sim 10 \rightarrow 30$). The electric field vector from the source defines a preferred direction in space and the total rotational angular momentum $\mathbf{J}$ is confined to $2J + 1$ possible orientations with respect to this field. For each $J$ these possible orientations are all energetically equivalent,
or degenerate.

Normally in the use of a microwave spectrometer an additional electric field which can be controlled by the spectrometrist is applied across the molecule at some angle with respect to the oscillating field produced by the source. This electric field perturbs the energy levels of the molecule and removes this $2J + 1$ degeneracy at least partially. This perturbation is termed the Stark effect. These effects can be calculated by the quantum mechanical perturbation method if the Stark effect is relatively much smaller than the rotational energy level spacing. For low $J$, Stark splittings can be observed in the lines of the microwave spectra. For high $J$ the Stark splittings are indistinct because of the large numbers and the overlap, plus the fact that for high $J$'s the rate at which the Stark lobes move is much slower. The Stark effect is quite a useful tool for the spectroscopist because low $J$ transitions can be easily recognized and the actual assignment of the lines may sometimes be made by the ratio of intensities of the lobes to one another. Also accurate measurement of the Stark splittings will provide the values of the dipole moment components.
III. ASSIGNMENT OF LINES

As has been previously stated, one of the first things to be done with a microwave spectrum is to assign the correct rotational quantum number transitions to the lines. By assigning Q-branch and R-branch lines for a particular vibrational level with the correct rotational quantum numbers, the rotational constants $A$, $B$, and $C$ can be determined. The methods employed by microwave spectroscopists to assign lines for asymmetric top molecules have been improved by the use of the computer in recent years but it is still a fairly difficult job.

The spectroscopist must first construct a model of the molecule, using as an approximation bond distances and angles which are known precisely for other molecules. Computer programs are available for the following steps which are described as for a hand calculation. A graph of a conformation of the molecule is made, choosing an arbitrary origin and set of Cartesian coordinates. Using the estimated bond lengths and angles, the inertial tensor is computed. Then the relation of the center of mass to the arbitrary origin is found, the inertial tensor about the center of mass computed, and the matrix diagonalized. The elements of the diagonal inertial tensor are the principle moments of inertia $I_A$, $I_B$, and $I_C$; the rotational constants $A$, $B$, and $C$ are then given by equation (4). However, the values of the rotational constants obtained in this way are not precise, but only a fairly good initial approximation.

The rotational energy levels are roots of polynomial equations,
the coefficients of which are functions of the rotational constants. For low \( J \) transitions the frequencies calculated from \( A, B, \) and \( C \) are not affected much by centrifugal distortion and are not extremely sensitive to small changes in \( A, B, \) and \( C \). Therefore, when the spectroscopist has finally calculated the rotational constants, he uses these to calculate the frequencies at which he might expect to find low \( J \) lines. In the region of these frequencies he searches for low \( J \) lines with the microwave spectrometer. The low \( J \) lines can usually be identified by their Stark effects. The number of Stark lobes equals the larger value of \( J \) in the transition, and the ratio of the intensities of the Stark lobes distinguishes \( P, R \)-branch transitions from \( Q \)-branch transitions.

There is currently a computer program in use which will take the predicted conformation of the molecule, calculate \( A, B, \) and \( C \) and predict the frequencies of the low \( J \) transitions. This program is very useful since it saves a great deal of time compared to hand calculations. Still there is a problem because the lines cannot always be identified by the Stark effects.

Until now the only method of attacking the assignment problem without Stark identification involved considering first \( Q \)-branch lines. Once \( Q \)-branches have been identified, since \( \Delta J = 0 \), from equation (5),

\[
\gamma = \frac{A - C}{2} \Delta E(\lambda).
\]  

(7)

The values of \( \frac{A - C}{2} \) and \( \lambda \) can be determined by methods which will be discussed later. Then by identifying a \( P \) or \( R \)-branch where \( J = \pm 1 \),
\[ \frac{A + C}{2} \] can be calculated. Then all three rotational constants are determined.

When the Q-branch transitions cannot be identified by the Stark effect, a graphical method may prove valuable. For a particular vibrational state of a rigid molecule \( \frac{A - C}{2} \) can be plotted versus \( \chi \). Usually the uncertainty in \( \chi \) is small. An experimental frequency can be assigned a set of quantum numbers by guesswork. Then varying \( \chi \) between its limits causes \( \frac{A - C}{2} \) to vary because the frequency provides a relationship between them. Thus, for each Q-branch transition a plot of a straight line is obtained for \( \frac{A - C}{2} \) and \( \chi \) over a small range of \( \chi \). The straight lines resulting from making several such guesses should intersect at a point which identified the correct values of \( \chi \) and \( \frac{A - C}{2} \) if the guesses are correct. For an actual molecule there will not be a point of intersection but a very small region of intersection because of experimental error in frequency measurement and centrifugal distortion. Guessed assignment lines that do not intersect at this point either belong to a different vibrational state or have been assigned the wrong J values.
IV. ASSIGNMENT OF LINES BY THE COMPUTER

Since so much human time and effort is involved in assigning the correct rotational quantum number transitions to the lines of a microwave spectrum, it was felt that it would be worthwhile to write a computer program which would do the work of assigning the lines. In order for the program to be profitable, however, it must save time, work, and be more accurate and thorough. The spectrum assignment program written is designed to do the work which previously had been done by human efforts and to do it better.

There are a number of differences in the approach to the problem which the computer must use and the human approach. The method of assigning lines is quite different from the methods used until now, approaching the graph intersection method more closely than any other. To a large extent the previous methods relied upon the Stark effect to get the assignment of the low J lines. It was very difficult to use high J lines in the initial assignment. In a rich asymmetric rotor spectrum the high J lines are most intense and Stark effects are difficult to study because of overlapping. The high J lines are very intense due to the increased population of the higher rotational energy levels. The Stark splittings are not resolved for high J lines, and since Stark splittings move out from the main line at a slower rate than that for low J lines, the high J lines often appear as single intense lines. The frequencies and intensities of the high J lines can be easily obtained from the spectrum and then used as data in the computer program.
The computer program uses a trial-and-error method of assigning the most intense lines, a method which would not be at all practical if one relied on hand calculations, but which is very much suited to the computer. The program is designed to take the frequencies and intensities of the most intense lines in the experimental spectrum and to attempt to match these with the most intense lines in a spectrum predicted by the computer. The assigning is done by trying all probable combinations of experimental and predicted lines, least squaring, calculating the sum of the squares of the deviations, and taking the correct assignment to be the combination which matches the greatest number of lines with the smallest deviation. This method is ideal for a computer since it requires little reasoning; it would take impossible amounts of time by hand.

Basically the computer uses initial estimates of $A$, $B$, $C$, $\mu_a$, $\mu_b$, and $\mu_c$ determined in the manner mentioned previously. A certain frequency region of the spectrum is selected, and the program predicts the lines in the spectrum for that region, giving the assignment for each line. A specified number $s$ of the most intense lines is selected from that predicted spectrum. This information is stored in the computer in the form of a matrix $M$ with $s$ rows and eight columns. The first six columns specify the quantum number transitions, the seventh and eighth contain the frequencies and intensities.

At this point there is an option in the program; one can choose whether to select only the Q-branch lines from $M$ or to deal with the whole set of lines. There are certain advantages to dealing with only the Q-branch lines. In this case, one of the parameters
vanishes (see equation (7)) and also one is dealing with fewer lines. Since the program uses all probable combinations, the fewer the lines used, the less computer-time is taken. Of course, with this route the only assignments are Q-branches.

The whole assignment program is based on the method of least squares. Therefore, a matrix of derivatives is needed. From the matrix "M", whether Q-branches are selected or not, a matrix of the partial derivatives of the predicted frequencies with respect to the rotational constants is formed. This matrix "D" consists of "s" rows (or less if the Q-branch option is used) and three columns, the partials of the frequencies with respect to A, B, and C. From equation (5) the partials can be calculated.

$$\frac{\partial \gamma}{\partial A} = \frac{1}{2} (J + 1)J + \frac{1}{2} \Delta E_{K-1K+1}(\chi) + \frac{A - C}{2} \frac{\partial \Delta E(\chi)}{\partial \chi} \frac{\partial \chi}{\partial A}$$

$$\frac{\partial \gamma}{\partial B} = \frac{A - C}{2} \frac{\partial \Delta E(\chi)}{\partial \chi} \frac{\partial \chi}{\partial B}$$

$$\frac{\partial \gamma}{\partial C} = \frac{1}{2} (J + 1)J - \frac{1}{2} \Delta E_{K-1K+1}(\chi) + \frac{A - C}{2} \frac{\partial \Delta E(\chi)}{\partial \chi} \frac{\partial \chi}{\partial C}$$

And from equation (6), the partials of with respect to the rotational constants can be calculated.

$$\frac{\partial \chi}{\partial A} = \frac{-2(B - C)}{(A - C)^2} , \frac{\partial \chi}{\partial B} = \frac{2}{A - C} , \frac{\partial \chi}{\partial C} = \frac{2(B - A)}{(A - C)^2}$$

Substituting $\frac{\partial \chi}{\partial B}$ into equation (8), it is seen that

$$\frac{\partial \Delta E(\chi)}{\partial \chi} = \frac{\partial \gamma}{\partial B}$$

Thus the matrix can be found in terms of $\frac{\partial \gamma}{\partial B}$, which can be calculated.
by the derivative method.

\[ \frac{df(x)}{dx} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \]

Then

\[ \frac{\partial \nu}{\partial A} = \frac{1}{2} (J + 1)J + \frac{1}{2}\Delta B(\nu) - \frac{(B - C)}{(A - C)} \frac{\partial \nu}{\partial B} \]

\[ \frac{\partial \nu}{\partial C} = \frac{1}{2} (J + 1)J - \frac{1}{2}\Delta B(\nu) + \frac{(B - A)}{(A - C)} \frac{\partial \nu}{\partial B} \]  \hspace{1cm} (11)

The uncertainty of each predicted frequency is given by

\[ (\Delta \nu)^2 = \left( \frac{\partial \nu}{\partial A} \right)^2 \Delta A^2 + \left( \frac{\partial \nu}{\partial B} \right)^2 \Delta B^2 + \left( \frac{\partial \nu}{\partial C} \right)^2 \Delta C^2 \]  \hspace{1cm} (12)

\( \Delta \nu \) can be calculated by forming the vector \( D \) Co  \( \tilde{D} \) (where "Co" is the matrix of covariance) and taking the square roots of the elements.

Thus a matrix of covariance is also needed. This is a diagonal matrix where the elements of the diagonal are the estimated squares of the deviation of the calculated rotational constants from the actual rotational constants. Within the uncertainty for each predicted frequency should lie the actual experimental line that corresponds to the predicted line. Thus choosing from the "Sp" most intense experimental lines within the uncertainty region of the predicted frequency should result in identifying the proper experimental line with the predicted line if the system is successful.

A matrix "T" is formed from all the possible experimental lines that might correspond to the predicted lines. From the region of uncertainty about each predicted frequency (\( \nu - \Delta \nu \) to \( \nu + \Delta \nu \), "Sp"
strongest lines are chosen from the experimental spectrum. Each row of "T" contains the "Sp" possible experimental frequencies that may fit predicted frequency. Therefore, since there are "s" predicted frequencies (unless, of course, the Q-branch option is taken), the matrix consists of "s" rows and "Sp" columns. Then the vector of calculated frequencies (the seventh column of "M") is subtracted from the matrix of observed frequencies which may possibly correspond to the calculated frequencies "T". The new matrix of the difference between the calculated and observed is also called "T" and replaces the former "T".

In order to find which lines in the matrix "T" correspond to an assignment, a method of least-squares is used. In this method the parameters are adjusted to give the closest fit and the sum of the squares of the deviation of calculated frequencies from observed frequencies is computed. When the sum of the squares of the deviation is small, the assignment is plausible, indicating a possible matching of experimental lines with quantum numbers.

The least squaring is done as follows. A Taylor series expansion is made about the guessed rotational constants. In principle the following equation should be satisfied.

\[ \nu^{obs} = \nu^{pre}(A_g, B_g, C_g) + \frac{\partial \nu}{\partial A} (A - A_g) + \frac{\partial \nu}{\partial B} (B - B_g) + \frac{\partial \nu}{\partial C} (C - C_g) \]  

(13)

where \( A_g, B_g, \) and \( C_g \) are the estimated rotational constants and the derivatives are evaluated at the estimated rotational constants. Let \( t \), a column vector, be the difference between the observed frequencies and the predicted frequencies.
Then equation (13) may be expressed as $t = Dx$ where $x$ is a vector consisting of $A - A_g$, etc. In practice there will be residuals $\Delta$ such that $\Delta = t - Dx$. Then $\Sigma$, the sum of the squares of the deviations, is defined by

$$
\Sigma = \sum (t - Dx)(t - Dx)
$$

Minimizing $\Sigma$ with respect to $x$ gives

$$
\frac{\partial \Sigma}{\partial x} = 2D(t - Dx) = 0.
$$

From this it is seen that

$$
Dt = DDx
$$

when $\Sigma$ is a minimum. The vector $x$ gives the shifts in rotational constants. For convenience, let $DD = B$ and $Dt = b$; then

$$
x = B^{-1}b = B^{-1}Dt
$$

and

$$
\begin{align*}
\Sigma &= (t - DB^{-1}b)(t - DB^{-1}b) \\
\Sigma &= c(1 - DB^{-1}b)(1 - DB^{-1}b)t \\
\Sigma &= c(1 - DB^{-1}b)t = cDt
\end{align*}
$$

because $(1 - DB^{-1}b)$ is idempotent. In computing $\Sigma$ one hopes to find it to be very small, indicating an assignment. A test on $\Sigma$ is introduced. When $\Sigma$ is computed and is smaller than the test value, it is assumed that the lines used possibly correspond to an actual assignment.
This is a version of the graphing method; the matching of the lines is like the intersection of lines or planes at the same point.

It is necessary that all possible combinations of the frequency differences be used as "t". The program does this by first using the first column of "T" as "t", then holding all elements of "t" constant except for the last quantity which is varied across the bottom row of the matrix "T". Then the second element of the next to bottom row is used and the process is repeated across the bottom row of "T". This is continued until all possible combinations of "T" have been used to construct "t". In actual practice it was found that the program operated more efficiently if only "r" rows of the "s" possible rows of "T" were used at a time. The rows chosen are also used in every possible combination so that all possibilities are tried. The number of assignments tried, n, is given by

\[ n = \left( \frac{s}{r} \right)(Sp)^r. \]

Each set of rows that passes the test is stored in a matrix "As" which consists of six columns: the first column tells the number of lines which have been matched; the second specifies the \( \Sigma \) computed for this particular match; the third, fourth, and fifth tell the difference in the actual and calculated rotational constants (\( \Delta A, \Delta B, \Delta C \)); the sixth is the number of parameters used. Another matrix is set up in order to indicate with which lines these quantities are dealing; it is important to know which lines match and the assignments of each.

This assignment part of the program takes a great amount of computer-time, mainly because of the matrix multiplying in equation (19).
The time taken by this part of the program was greatly shortened by mathematical analysis. Recognizing that when "C" is an idempotent matrix (i.e., $C^2 = C$) and there are "m" parameters and "n" observations, then "m" elements of the diagonal of "C" are 0, and "n - m" elements are 1. In this case $C = (1 - DB^{-1}D)$ and from equation (19) in bracket form we see that

$$\Sigma = \langle t|\hat{C}|t \rangle$$

$$\Sigma = \sum_{i=1}^{n-m} \langle d_i|\hat{d}_i \rangle \langle d_i|t \rangle$$

$$= \sum_{i=1}^{n-m} \langle d_i|t \rangle^2$$

where $d_i$ are the eigenvectors of "C". When $(n-m) = 1$,

$$\Sigma = \langle 1|t \rangle^2.$$ 

(21)

In this approach "C" is diagonalized and the matrix of eigenvectors corresponding to eigenvalue 1 "Or" is formed.

$$\Sigma = (Or|t)^2$$

The first $n-1$ elements of each row of Or are matrix multiplied with the first $n-1$ rows of $t$. Then the last element of each row of Or is successively multiplied with the last element of $t$ and $\Sigma$ accumulated. This takes much less computer-time than the matrix multiplication.

After the assignment program has tried all possibilities there is a halt in the program in order to allow the operator to look over the assignments and choose further criterions for reevaluating the results. This final portion, termed REFIT, picks out the best
assignments from "As" using the criterions submitted. For these "best" assignments, it recomputes the matrix of derivatives at the new rotational constants and recalculates $\Sigma$, tests with more stringent conditions. If $\Sigma$ found cannot pass the new test then the assignment is thrown away. Finally it tries to fit more lines together. The assignments that are retained after refitting the lines must be tested experimentally. However, presently there are plans to refine further the program to operate more efficiently and sieve more carefully.
V. DESCRIPTION OF THE PROGRAM

The spectrum prediction program is outlined by a basic flow diagram on the next page. For more detailed information concerning the program see the Documentation at the close of this thesis. The spectrum prediction package was not discussed in this thesis because it was written by Dr. R. F. Curl and adapted for use in the program by him.

The remainder of the program was written in the GENIE and AP2 languages of the Rice University Computer. This computer was used for a number of reasons. The languages were not difficult to learn and being able to use the Rice Computer was worth the effort involved. The cost of the Rice Computer is $35.00 per hour, as opposed to the higher IBM rates, and it is available for longer stretches of time. It is ideal for a research computer program because the author of the program is permitted to operate the computer himself, thus enabling him to find mistakes and make corrections in the program. Also there are some things that the Rice computer can do which were particularly helpful in this program, such as convenient use of tables the entries of which are matrices.
BASIC FLOW DIAGRAM OF SPECTRUM ASSIGNMENT PROGRAM

START

READ DATA: A, B, C, μA, μB, μC, νmin, νmax, Co, EXPM, S, Sp, Q, r

OMIT
removes weak lines
from Expm

SPECPRE
predicts a spectrum
in the matrix M

Q = 0? NO
YES

DERIV
takes derivatives
of frequencies with
respect to A, B, C;
forms matrix D.

Δν = √D Co D
along the diagonal

POSS
picks out possible
experimental freq.
that might correspond
to predicted freq.;
forms matrix T

ASSIGN
tries all possible
assignments, r rows
at a time; forms
matrix 'As

REFIT
tests the assign-
ments found and
keeps the best.

FINE
VI. PRACTICAL APPLICATION OF THE PROGRAM TO ALLYL ALCOHOL

The spectrum of allyl alcohol was chosen to test the computer program. For one thing this microwave spectrum is quite complex and very few lines have been assigned by spectroscopists. Some work on the molecule had already been done by Murty and Curl.¹ Also, there was an extensive list of line frequencies and intensities compiled by John Rigden of Eastern Nazarene College. Allyl alcohol can plausibly exist in five conformations, each with different A, B, C's and each with its own set of lines. Unfortunately, all five may be seen in the same spectrum and there are few clues to identify which lines belong to which conformation. The conformation assigned previously predominates at low temperatures but gives relatively weak lines at room temperature. This program seemed as though it might be helpful in assigning other conformations.

The conformation assigned previously is termed the "le" conformation, the "gauche" H "gauche" 2 form of allyl alcohol. The conformation which was investigated by the computer program is termed the "la" conformation, the "cis" H "trans". A table of the rotational constants and dipole moment components for each conformation is given in Table 1. These values were used to predict the spectrum for each conformation.

The lines observed by Rigden in the region from 26,000 MHz to 40,000 MHz were used as experimental data. A 52 by 2 matrix of the experimental frequencies and intensities was typed on paper tape and read into the computer. All lines of relative intensity less than or
equal to 2.0 were eliminated by the program OMIT, leaving 130 lines. From the A, B, C, \( \mu_A \), \( \mu_B \), \( \mu_C \) for "la" allyl alcohol a spectrum of 50 lines was predicted for a temperature of 300° K and a J up to 25. It was stored for later reference. From these 50 lines the 10 most intense lines were selected to work with first.

The Q-branch option was chosen so that only the assignment of Q-branch lines was attempted. The matrix of derivatives was computed and the 15 possible experimental lines that might correspond to each predicted line were selected and arranged in the matrix of possibilities "T".

All of the possible assignments were tried 3 rows at a time. Therefore, the number of assignments tried was

\[
15^3 \cdot \frac{10!}{7! \cdot 3!} = 405,000.
\]

Out of this number 259 assignments were found with \( \Sigma \) less than the test value of 200. Of these, only 14 were 4-line assignments; only 2 were 5-line assignments. The remainder were 2 and 3-line assignments. The test value specified was so large because \( \chi \) is non-linear and the deviation in \( \chi \) is large.

Because of this nonlinearity the matrix of possible assignments "As" was then retested by the program REFIT. All assignments of less than 3 lines were eliminated. For the remaining assignments a new matrix of derivatives was computed. The least squaring was repeated for each assignment until a converged set of rotational constants was obtained. Then a search over the experimental spectrum was made to
assign other lines. A matrix was printed, consisting of the assigned lines and predicted but as yet unobserved lines.

The only 5-line assignment of Q-branches (which corresponded to neither of those found initially) was considered to be the best possibility. The matrix mentioned above is shown in Table 2 for this assignment. This type of matrix is the final output of the system. It consists of the assigned lines (five in this case), the quantum numbers of the transitions, the predicted frequencies and intensities and the observed frequencies and intensities. Note that the predicted lines are also included in the matrix with zeros in the observed frequency and intensity columns. Then the spectroscopist can go back to the microwave spectrometer and see if the predicted lines can be found. If they can be, then the assignment is good. If not, it must be discarded. Thus the next step involves checking the best proposed assignments experimentally.
<table>
<thead>
<tr>
<th>CONFORMATION</th>
<th>A (MHz)</th>
<th>B (MHz)</th>
<th>C (MHz)</th>
<th>(\mu_a) (D)</th>
<th>(\mu_b) (D)</th>
<th>(\mu_c) (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>17364.</td>
<td>5963.</td>
<td>4568.</td>
<td>0.90</td>
<td>1.42</td>
<td>0.00</td>
</tr>
<tr>
<td>1b</td>
<td>16484.</td>
<td>6222.</td>
<td>4697.</td>
<td>1.02</td>
<td>0.44</td>
<td>1.27</td>
</tr>
<tr>
<td>1c</td>
<td>27991.</td>
<td>4282.</td>
<td>4143.</td>
<td>0.17</td>
<td>1.59</td>
<td>0.54</td>
</tr>
<tr>
<td>1d</td>
<td>26716.</td>
<td>4377.</td>
<td>4201.</td>
<td>0.79</td>
<td>0.60</td>
<td>1.36</td>
</tr>
<tr>
<td>1e</td>
<td>25951.</td>
<td>4381.</td>
<td>4267.</td>
<td>1.39</td>
<td>0.06</td>
<td>0.95</td>
</tr>
</tbody>
</table>
\( \text{A} = 1.72338619685 \times 10^4 \), \( \text{B} = 6.03137406185 \times 10^3 \), \( \text{C} = 4.63495501845 \times 10^3 \)

\( \text{NUMB} = 5 \)

\| \) | \( 1 \) | \( 9 \) | \( 3.7 \times 10^4 \) | \( 1.29475594690 \times 10^1 \) | \( \text{N} \) | \( 2 \) | \( 0 \)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 )</td>
<td>( 8 )</td>
<td>( 1 )</td>
<td>( 3.3 \times 10^4 )</td>
<td>( 1.26269864134 \times 10^1 )</td>
<td>( 3.35855000000 \times 10^1 )</td>
<td>( 4.0 \times 10^0 )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( 7 )</td>
<td>( 1 )</td>
<td>( 3.0 \times 10^3 )</td>
<td>( 1.16182415777 \times 10^1 )</td>
<td>( 1.0 \times 10^0 )</td>
<td>( 2 )</td>
</tr>
</tbody>
</table>
| \( 1 \) | \( 6 \) | \( 1 \) | \( 2.97596501909 \times 10^3 \) | \( 9.98059571636 \times 10^1 \) | \( 2.98063800000 \times 10^1 \) | \( 2.0 \times 10^0 \)
| \( 1 \) | \( 5 \) | \( 1 \) | \( 2.98046452030 \times 10^1 \) | \( 7.93722845999 \times 10^1 \) | \( 2.98063800000 \times 10^1 \) | \( 2.0 \times 10^0 \)
| \( 1 \) | \( 6 \) | \( 1 \) | \( 3.1952936586 \times 10^3 \) | \( 6.66218567775 \times 10^1 \) | \( 6.0 \times 10^0 \) | \( 1 \) |
| \( 1 \) | \( 4 \) | \( 1 \) | \( 3.07503976504 \times 10^2 \) | \( 5.79050564722 \times 10^1 \) | \( 3.07512300000 \times 10^1 \) | \( 5.0 \times 10^0 \)
| \( 1 \) | \( 6 \) | \( 1 \) | \( 3.21993072654 \times 10^1 \) | \( 3.78246830326 \times 10^1 \) | \( 3.22054000000 \times 10^2 \) | \( 2.0 \times 10^0 \)
| \( 1 \) | \( 6 \) | \( 1 \) | \( 3.37300456134 \times 10^1 \) | \( 1.9645857291 \times 10^0 \) | \( 1.0 \times 10^0 \) | \( 2 \) |
| \( 1 \) | \( 6 \) | \( 1 \) | \( 3.7796345590 \times 10^1 \) | \( 1.6692172851 \times 10^0 \) | \( 3.77956300000 \times 10^1 \) | \( 6.0 \times 10^0 \)

**Table 2.** Predicted and Assigned Lines for the "1a" Conformation of Allyl Alcohol
BIBLIOGRAPHY


1. **CNTRL** ........................................ D-1
2. **OMIT, NU, ITER** ................................. D-3
3. **SPECPRE** ........................................ D-4
4. **K, KNU** .......................................... D-7
5. **ASYM** .......................................... D-8
6. **COSA** .......................................... D-9
7. **COSB** .......................................... D-10
8. **COSC** .......................................... D-11
9. **TRIDIA** ........................................ D-12
10. **DERIV** .......................................... D-16
11. **POSS** .......................................... D-17
12. **ASSIGN** ......................................... D-19
13. **PUTIN** .......................................... D-25
14. **MPART** .......................................... D-27
15. **REFIT** .......................................... D-29
DEFINE
SCALAR TEST, LEAST, TE, TESTP
SCALAR TEST1, MXSIG
INTEGER MNNUM
MATRIX LINES, EXSV
FUNCTION REFIT
INTEGER JMAX
MATRIX CO, EXPN, M, D, DEL, N, AS, T, P
VECTOR DELNU
FUNCTION OMIT
FUNCTION SPECPRE, KNU, DERIV, POSS, ASSIGN
CNTRL(Z) = SEQ,
INTEGER s, Sp, r, Q, t, 1
INTEGER SW
DATA A, B, C, MUA, MUB, MUC, NUMIN, NUMAX, s, Q, r, Sp, Co
DATA SW
DATA EXPM
EXSV = EXPM
    IF(SLN) SKP /MOO
TRA //NOTOM
EXECUTE OMIT(EXPM)
NOP Z
PRINT A, B, C, MUA, MUB, MUC, NUMIN, NUMAX, s, Q, r, Sp, Co
PRINT TEST, TESTP, TE, LEAST, JMAX
EXECUTE SPECPRE(M, A, B, C, MUA, MUB, MUC, NUMIN, NUMAX, s)
N=M
EXECUTE KNU(N)
PRINT N
P=M
CC=#DER if Q=0
FOR i=s, -1, 1
CC=#BY if M, 1 = M
EXECUTE MINSE(47776, 1, 1, M)
REPEAT
    CLA M, U->B1
    CLA B1
    IF(NUL) TRA NOQ
N=M
EXECUTE KNU(N)
PRINT N
DER
N=M
EXECUTE MINSE(0, SW+1, 1, M)
t=ROW(N)
EXECUTE DERIV(D, M, A, B, C)
PRINT D
DEL=D @ TRAM(D)
EXECUTE VSPACE(DELNU,t)
FOR i=1,1,t
DELNUi=SQR(DEL1,1)
REPEAT
PRINT DELNU
EXECUTE POSS(T,M,DELNU,EXPM,Sp)
PRINT T
EXECUTE ASSIGN(As,LINES,T,D,TEST,r,Co)
PRINT As
HTR #CC
DATA MXSIG,MNNUM,TEST1
EXECUTE REFIT(As,N,EXSV,T,LINES,A,B,C)
CC=#END
NOQ EXECUTE SCRIBE(FNOQ)
M=P
CC=#DER
FNOQ FORMAT
END
NO Q
LEAVE
DEFINE
DEFINE MATRIX EXPM
SCALAR LEAST

OMIT(EXPM) = SEQ
INTEGER i, j
i = ROW(EXPM)
FOR j = 1, -1, 1
CC = #BY if LEAST < EXPM, 2
EXECUTE MINSE(+77776, j, 1, EXPM)
BY
END

LEAVE DEFINE

***

NU

DEFINE FUNCTION ASYM
INTEGER IL, IU
VECTOR EU, EL
NU(A, B, C, JL, PL, IL, JU, PU, IU) = SEQ
EXECUTE ASYM(EU, Z, A, B, C, JU, PU)
EXECUTE ASYM(EL, Z, A, B, C, JL, PL)
RESULT = EU_IU - EL_IU

END

LEAVE DEFINE

***

ITER

DEFINE MATRIX M
FUNCTION NU
SCALAR A, B, C
ITER(M, A, B, C) = SEQ
INTEGER i, j
i = ROW(M)
FOR j = 1, -1, 1
M_j = NU(A, B, C, M_j, 1, M_j, 2, M_j, 3, M_j, 4, M_j, 5, M_j, 6)
REPEAT

END

LEAVE DEFINE

D-3
DEFINE
SCALAR T
MATRIX M,EJ,EJP,TJ,TJP,TW,TWP,C0
MATRIX COSA,COSB,COSC
MATRIX SELA,SELB,SELC
VECTOR EW,EWP
INTEGER JMAX,s
FUNCTION ASYM,COSA,COSB,COSC,PUT,ACOPY
SPECPR(M,A,B,C,MUA,MUB,MUC,NUMIN,NUMAX,s).=SEQ
LET k = 2.0894146*4
INTEGER J,JP,P,PP,1,j,l,a,b,n,m
INTEGER r,t
EXECUTE MSPACE(M,s,8)
J=0
EXECUTE CONTR(1,+0120,+4010,EJ)
EXECUTE CONTR(+77777,+1160,0,EJ)
EXECUTE CONTR(1,+0120,+4010,TJ)
EXECUTE CONTR(+77777,+1160,0,TJ)
EXECUTE VSPACE(EW,1)
EXECUTE MSPACE(TW,1,1)
TW_1_1 = 1.0
EXECUTE ACOPY(EW,Z,Z,EJ,0,Z)
EXECUTE ACOPY(TW',Z,Z,TJ,0,Z)
FOR JP = 1,1,JMAX
n = 3 if JP=1, 4
EXECUTE CONTR(n,+0120,+4010,EJP)
EXECUTE CONTR(n,+0120,+4010,TJP)
EXECUTE CONTR(+77776,+1160,0,EJP)
EXECUTE CONTR(+77776,+1160,0,TJP)
n = n-2
FOR PP = -1,1,n
EXECUTE ASYM(EW,TW,A,B,C,JP,PP)
EXECUTE ACOPY(EW,Z,Z,EJP,PP,Z)
EXECUTE ACOPY(TW,Z,Z,TJP,PP,Z)
REPEAT
m = LENGTH(EJ)
l = 0 if J=0, -1
r = m+l-1
FOR P = 1,1,r
FOR PP = -1,1,n
EXECUTE ACOPY(EJ,P,Z,EW,Z,Z)
EXECUTE ACOPY(EJP,PP,Z,EWP,Z,Z)
EXECUTE ACOPY(TJ,P,Z,TW,Z,Z)
EXECUTE ACOPY(TJP,PP,Z,TWP,Z,Z)
TSR PASS
SPF *END+1
REPEAT
REPEAT
J = JP
r = n-1
D-4
FOR $P = -1, 1, r$

$\mathbf{t} = P + 1$

FOR $PP = t, 1, n$

EXECUTE ACOPY(EJP, $P$, Z, EW, Z)

EXECUTE ACOPY(TJP, $P$, Z, TW, Z)

EXECUTE ACOPY(EJP, $PP$, Z, EWP, Z)

EXECUTE ACOPY(TJP, $PP$, Z, TWP, Z)

TSR PASS SPF #END+1

REPEAT REPEAT

EXECUTE CONTR($n+2$, +0120, +4010, EJ)

EXECUTE CONTR($n+2$, +0120, +4010, TJ)

EXECUTE CONTR(+77776, +1160, 0, EJ)

EXECUTE CONTR(+77776, +1160, 0, TJ)

FOR $P = -1, 1, n$

EXECUTE ACOPY(EJP, $P$, Z, EJ, $P$, Z)

EXECUTE ACOPY(TJP, $P$, Z, TJ, $P$, Z)

REPEAT REPEAT

CC = #END

PASS

PF RPA NC SPF *END+1

SPF

CC = #NA if $MUA = 0$

CC = #NA if $SELA | JP | J | P, PP = 0$

CO = CCSA($J, P, JP, PP$)$J$, P, PP

MU = $MUA$

TSR SORT SPF *END+1

NA CC = #NB if $MUB = 0$

CC = #NB if $SELB | JP | J | P, PP = 0$

CO = CCSB($J, P, JP, PP$)$J$, P, PP

MU = $MUB$

TSR SORT SPF *END+1

NB CC = #NC if $MUC = 0$

CC = #NC if $SELC | JP | J | P, PP = 0$

CO = COSC($J, P, JP, PP$)$J$, P, PP

MU = $MUC$

TSR SORT SPF *END+1

NC SORT

PF RPA NO SPF *END+1

CO = TW CO TRAN(TWP)

a = ROW(CO)

b = COL(CO)

FOR $i = 1, 1, a$

FOR $j = 1, 1, b$

CEL = CO$J$, $J$

D-5
CC = #NOLIN if \(|\text{CEL}| < 0.3\)
\(\text{NU} = \text{EWP}_1 - \text{EWP}_1\)
CC = #NOLIN if \(|\text{NU}| < \text{NUMIN} \text{ or } \text{NUMAX} < |\text{NU}|\)
\(\text{TSR} = \text{LINE} \text{ SPF} \ast \text{END}+1\)

\text{NOLIN REPEAT REPEAT}
\text{NO LINE PF RPA BACK SPF \ast \text{END}+1}
\text{INT} = \text{MU MU CEL CEL EXP} \left(-\text{EWP}_1/(\text{kT})\right) \text{ if } \text{NU} < 0. \text{, MU MU CEL CEL EXP} \left(-\text{EWP}_1/(\text{kT})\right) \text{ if } \text{NU} < 0.
\text{CC = \#BACK if INT} \leq \text{MO}
\text{EXECUTE PUT(M, J, P, i, 3P, PP, j, NU, INT)}
\text{BACK TRA (PF)}
\text{END}
\text{DEFINE}

\text{LEAVE}

D-6
DEFINE
INTEGER J, P, i, L
FUNCTION L
\[ L(J, P, i) = \text{SEQ} \]
INTEGER i, KM, PP, PM
PM = 0 if \((\text{EVEN}(J) \text{ and } P = 0) \text{ or } (\text{not EVEN}(J) \text{ and } P = -1)\),
2 if \((\text{EVEN}(J) \text{ and } P = +1) \text{ or } (\text{not EVEN}(J) \text{ and } P = 2)\), 1
PP = 0 if \((\text{EVEN}(J) \text{ and } P = 0) \text{ or } (\text{not EVEN}(J) \text{ and } P = +1)\),
2 if \((\text{EVEN}(J) \text{ and } P = -1) \text{ or } (\text{not EVEN}(J) \text{ and } P = 2)\), 1
KM = PM + 2(L(J, P) - 1)
i = PP + 2(i - 1)
P = KM

END
LEAVE

*******

KNU

DEFINE
MATRIX M
FUNCTION K
\[ K(M) = \text{SEQ} \]
INTEGER i, j
i = \text{ROW}(M)
FOR j = 1, 1, i
A = M_{j, 2}
B = M_{j, 3}
EXECUTE K(M_{j, 1}, A, B)
M_{j, 2} = A
M_{j, 3} = B
A = M_{j, 5}
B = M_{j, 6}
EXECUTE K(M_{j, 4}, A, B)
M_{j, 5} = A
M_{j, 6} = B
REPEAT

END
LEAVE
DEF
MAT H, HP, T
VEC EV, E
INT J, P
FUC TRIDIA, EVCT, L

ASYM(E, T, A, B, C, J, P) = SEQ
INT n, i, j
BE = (A+C)/2.
AL = B - BE
GA = (A-C)/4.
JF = 1.0(J+1)
n = L(J, P)
EXEC MSPACE(H, 2, n)
K = 1.0|P|
FOR i = 1, 1, n
H_{i,i} = BE JF + AL K K
CC = #BY if i=n
H_{i,i+1} = GA GA ((JF - K(K+1)) (JF - (K+1) (K+2)))
K = K + 2.
BY
REPEAT
CC = #NZ if P<0 or J<2
H_{i,i+1} = 2.H_{i,i}
NZ
CC = #N1 if P+1 1
H_{i,1} = H_{1,1} + GA JF if O<P, H_{1,1} - GA JF
N1
EXEC TRIDIA(HP)
EXEC VSPACE(E, n)
FOR i = 1, 1, n
E_{i,1} = HP
H_{2,i} = SQRT(H_{2,1})
REPEAT
CC = #END if T = 0
EXEC MSPACE(T, n, n)
FOR i = 1, 1, n
E_{i} = EVCT(H, E, i)
FOR j = 1, 1, n
T_{i, j} = EV j
REPEAT
REPEAT
END
DEFINE
LEAVE
DEFINE
MATRIX CA, SELA
INTEGER J, P, JP, PP
INTEGER L
FUNCTION L, THRJ
COSA(J, P, JP, PP) = SEQ
INTEGER n, m, i, j, k, l, K, KP
LET R2 = 1.414213562373
i = |J-JP|
n = L(J, P)
m = L(JP, PP)
EXECUTE MSPACE(CA, n, m)
K = |P|
FOR i = 1, 1, n
KP = |PP|
FOR j = 1, 1, m
CC = //BY if //PLUS if K<KP
CA1, J = (-1)(j+K+KP+1) SQR(0.5(2J+1)(2JP+1)) THRJ(2JP, 2, 2J, -2KP, -2, 2K)
CC = #OUT
CA1, J = (-1)(J+KP) SQR(0.5(2J+1)(2JP+1)) THRJ(2JP, 2, 2J, -2KP, 2, 2K)
OUT
CC = #BY if K=0 and KP=0
CA1, J = R2 CA1, J
BY
KP+J, KP+2
REPEAT
K = K+2
REPEAT
FIN
RESULT = CA
END
DEFINE
LEAVE
DEFINE
MATRIX CA, SELB
INTEGER J, P, JP, PP
INTEGER L
FUNCTION L, THRJ
COSB(J, P, JP, PP) = SEQ
INTEGER n, m, i, j, k, l, K, KP
LET R2 = 1.414213562373
i = |J-JP|
n = L(J, P)
m = L(JP, PP)
EXECUTE MSPACE(CA, n, m)
K = |P|
FOR i = 1, 1, n
KP = |PP|
FOR j = 1, 1, m
CC = #BY if K+KP SQR(1. (2J+1)(2JP+1))THRJ(2JP, 2, 2J, -2KP, 0, 2K)
CA1, J = (-1)(J+KP) SQR(1. (2J+1)(2JP+1))THRJ(2JP, 2, 2J, -2KP, 0, 2K)
BY KP = KP+2
REPEAT
K = K+2
REPEAT
FIN RESULT = CA
END
DEFINE
LEAVE
DEFINE
MATRIX CA, SELC
INTEGER J, P, JP, PP
INTEGER L
FUNCTION L, THRJ
COSC(J, P, JP, PP) = SEQ
INTEGER n, m, i, j, k, l, K, KP
LET R2 = 1.414213562373
I = |J-JP|
N = L(J, P)
M = L(JP, PP)
EXECUTE MSPACE(CA, n, m)
K = |P|
FOR i = 1, 1, n
KP = |PP|
FOR j = 1, 1, m
CC = #BY if |K+KP|+1, #PLUS if K<KP
CA1, j = (-1)(J+KP)SQR(0.5(2J+1)(2JP+1))THRJ(2JP, 2, 2J, -2KP, -2, 2K)
CC = #OUT
CA1, j = (-1)(J+KP)SQR(0.5(2J+1)(2JP+1))THRJ(2JP, 2, 2J, -2KP, 2, 2K)
OUT
CC = #BY if K+0 and KP+0
CA1, j = R2 CA1, j
BY
KP*J, KP+2
REPEAT
K = K+2
REPEAT
FIN
RESULT = CA
END
DEFINE
LEAVE
TRIDIA

ORG

REM BACK-TRANSLATION

-Z

TRA *136, U→R

CLA T7, U→B1

CLA B1, U→B3

CLA B3+1

LUR d39, U→B3

B3 IF(PNZ)SKP a2

T7 TRA L166, U→B4

B3 RPA L220

B3 LUL 17

BLU L235

BAU aL226

TSR *126, U→T7

CLA B1, U→B1

CLA B1+2, U→B5

CLA B3+B5+1

STO L231

CLA B1+1, B5+1

RPA L222, U→B4

CLA B3+B4+1

STO L230

Z LDR→ B5

R STO L232

B5 RPA L223

B6 RPA L227

CLA L226, U→PF

CLA PF+1, U→B6

B3 LDR PF+2, U→B2

R RPA L224, U→PF

Z STO L217, U→B1

B2 IF(PNZ)SKP a1

TRA L125

Z STO L216, U→B3

STO L221, B3+1

LT6 |B1+B3+B5|

TSR *202

SB5 *L223

-L6 FAD B1+B3+B4

STO L216

T6 STQ L221, B3+1

LT6 |B1+B3+B5|

TSR *202

SB5 *L223

-L6 FAD B1+B3+B4

FAD B1+B3+B4

FSB L221, U→T4

IF(PNZ)SKP L216

T4 STG L216

B3 IF(POS)SKP aB2

TRA L233

CLA L233

FAD→ L216, U→T4

D-12
| B4 | FAD→ | L217,R→Z |
| L63 | -T4 | 40023 | a*L220,PF+1 |
| Z | SPF | PF+34001,B3-1 |
| L65 | PAD | *L224,B3+1 |
| B4 | RPA | B1+B3+B4 |
| 40023 | STO | PF+B1+B3 |
| B4 | VDF | B1+B3+B5,U→T4 |
| 40023 | STO | B1+B3+B6,B3+1 |
| B3 | IF(PNZ)SKP | aB2 |
| -T4 | TRA | L65 |
| | SB4 | B2+3 |
| | STX | 77771 |
| L75 | Z | SB3 |
| -U | FAD | B1+B3+B6 |
| | FAD→ | | } | U→T4 |
| B3 | IF(NNZ)SKP | aB2,B3+1 |
| | TRA | L104,B4-1 |
| | VDF | PF+B1+B3 |
| | PMP→ | B1+B3+B6-1,CC+X |
| L104 | B4 | IF(PNZ)TRA | L75 |
| Z | SB4 | *L222,U→B3 |
| | LT4 | PP+B1+1,B3+1 |
| | T4 | PAD | B1+B3+B6 |
| | STO | B1+B3+B4 |
| | CLA | PF+B1+B3+1,U→T4 |
| | PMP | B1+B3+B6 |
| | STO | B1+B3+B5,B3+1 |
| | IF(PNZ)SKP | aB2 |
| | TRA | L107 |
| L116 | CLA | B1+B3+B5 |
| | IF(PNZ)SKP | L234 |
| B3 | TRA | L33,U→B2 |
| B3 | IF(POS)SKP | L220,B3+1 |
| | TRA | L117 |
| | TRA | L33 |
| | CLA | B1+B5 |
| | IF(NEG)SKP | L234 |
| B1 | TRA | L116,B1-1 |
| | IF(POS)SKP | L220,B4+1 |
| | TRA | L125 |
| | LT4 | L217 |
| B4 | RPA | L135 |
| 40022 | STO | B2+B3,0001,B1-1 |
| L135 | T4 | STX | 77775 |
| | SB2 | a1 |
| | 40022 | L220 |
| | T4 | B2+B4,U→B1 |
| | 40022 | L146 |
| B2 | LT7 | B1+B4,B1-1 |
| B1 | IF(ZER)TRA | B1+B4+1 |
| T7 | IF(NEG)JMP | B2+B4,U→T7 |
| T7 | LDR→ | |
| R | STO | |
| D-13 | | |
TRIDIA (con.)

L146   B2  TRA  L141
      IF(ZER)SKP  a2
      TRA  L140, B2-1
B4     SB2  *L220, U->B1
      SB6  *L227
      CLA  L222, U->B3
      CLA  L223, U->B4
      CLA  L230
      STO  B2+B3+1
      CLA  L232
      STO  B4
      CLA  L231
      STO  B2+B4
      CLA  L236
      BAU  aL226
      TSR  *126, U->T7
      TRA  L213
L165   B3  IF(PNZ)SKP  a1, I->B2
L166   PF  TRA  L213, U->B3
      SB1  1
      LT4  *B4, B2+1
      LT5  *B4, B1+1
      LT6  *B4, U->B3
      PF  L237, U->T6
      T4  FMP  T5, B1-1
      T4  FMP  T6
      T4  FMP  T237, U->T6
      FSB  L225
      FMP  U
      FAD  T5
      STO  L225
      FMP  U
      FAO  L225
      FMP  L240
      STO  *B4, B2-1
      T6  L225
      FMP  L240
      STO  *B4, B2+1
      TRA  *137
L213   TRA  PF
L216   OCT  015733623407210671
L217   OCT  020013551072115350
L220   OCT  00000000000000000000616
L221   OCT  770155400541674433
L222   OCT  000000000000034001
L223   OCT  000000000000034021
L224   OCT  000000000000034410
L225   OCT  000000000000033764
L226   OCT  000000000000033765
L227   OCT  000000000000033766
L230   OCT  000000000000000000
L231   OCT  740010000000000000
L232   OCT  000000000000000000
L233   OCT  000000000000000000

D-14
<table>
<thead>
<tr>
<th>TRIDIA (con.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L234</td>
</tr>
<tr>
<td>L235</td>
</tr>
<tr>
<td>L236</td>
</tr>
<tr>
<td>L237</td>
</tr>
<tr>
<td>L240</td>
</tr>
</tbody>
</table>

D-15
DEFINE
MATRIX D, M
FUNCTION NU
DERIV(D, M, A, B, C) = SEQ
INTEGER i, s, j
s = ROW(M)
EXECUTE MSPACE(D, s, 3)
FOR i = 1, 1, s
DELB = 10. (NU(A, B + 0.1, C, M, 1, 1, M, 2, M, 3, M, 4, M, 5, M, 6) - M, 7)
JST = M, 1 if M, 4 < M, 3, 1, M, 3, 1
DELAC = JST(M, 4 + M, 1, 1, M, 3, 1)
DELAC = 2. J
DELAC = 2. J
DELGA = (0.5(A + C) DELAC) 2.0 (A - C) - 1
D, 1 = DELAC(0.5) + DELGA(0.5) - (B - C)(A - C) - 1 DELB
D, 2 = DELB
D, 3 = DELAC(0.5) - DELGA(0.5) + (B - A)(A - C) - 1 DELB
REPEAT
END
DEFINE
LEAVE
DEFINE

MATRX T,M,EXPM,MTRX
VECTOR DELNU
INTEGER Sp,s
POSS(T,M,DELNU,EXPM,Sp).=SEQ
INTEGER i,t,j,k,r,u,v,w
s=ROW(M)
t=ROW(EXPM)
EXECUTE MSPACE(T,s,Sp)
FOR i=1,1,s

LIMIT
EXECUTE MSPACE(MTRX,2,Sp)
FOR j=1,1,t
CC=#CHECK if (M 1y-DELNUj)<EXPM,1
REPEAT

DO
FOR u=1,1,Sp
T 1u=MTRX,1,u
REPEAT

CHECK
CC=#ORDER if EXPM,1<(M 1y+DELNU 1)
CC=#LIMIT
ORDER
FOR r=1,1,Sp
CC=#RECORD if MTRX,2,r<EXPM,1
REPEAT

RECORD
FOR k=Sp,-1,r+1
MTRX 2,k=MTRX,2,k-1
MTRX 1,k=MTRX,1,k-1
REPEAT
MTRX 2,J=EXPM
MTRX 1,J=EXPM,1
CC=#LIMIT
FIN
FOR v=1,1,s
FOR w=1,1,Sp
T 1v,w=MTRX,1,w
REPEAT
REPEAT

END

LEAVE
ASSIGN

DEFINE
MATRIX As, T, D, c, C, B, Bp, Cp
MATRIX C, C0
MATRIX LINES
VECTOR V, PART
INTEGER r
SCALAR TESTP
VECTOR t, x, y, Tp
INTEGER VECT0 j, R, z, w
FUNCTION MPART, PUTIN

ASSIGN(As, LINES, T, D, TEST, r, C0) = SEQ
INTEGER s, Sp, i, k, l, m, n, q, Np, a
INTEGER b, d
EXECUTE MSPACE(As, 1, 6)
EXECUTE CONTR(1, +0120, +4010, LINES)
EXECUTE CONTR(0, +0600, 0, LINES)
EXECUTE CONTR(2, +0120, +4402, +1)
EXECUTE CONTR(0, +0700, 0, 0)
s = ROW(T)
Sp = COL(T)
l = r
EXECUTE VSPACE(R, r)
FOR m = 1, 1, r
Rm = m
REPEAT
EXECUTE VSPACE(t, r)
EXECUTE MPART(n, c, B, D, R)
EXECUTE VSPACE(j, r)
C = C
EXECUTE DIAG(C, C0, 0)
FOR m = r, r-1, 1
CC = #BY if |C - 1, | < 1,.*-3
EXECUTE MINSERT(77776, m, 1, Or)
BY
REPEAT
b = ROW(Or)
EXECUTE VSPACE(V, b)
FOR k = 1, 1, b
Vk = Or, k, r
REPEAT
d = Rr
LOOP
FOR k = 1, 1, r-1
Tk = Tr, j, k + 1
Tr = 0.0
PART = Or t

D-19
ASSIGN (con.)

CLA
B1
ADD
SB2
SB1
L2
Z
NOP
CLA
L1
B1
IF(NEG)TRA
T4
FMP
FAD
FMP
FAD->
TRA
OUT
T5
IF(FNZ)SKP
TRAP
FIT
BACK
B2
IF(FNZ)TRA
PIT
L2

j_r = Sp - 1
k_r = r
PT
CC=#UV if J_k<Sp-1
j_k=0
k=$k-1
CC=#PT if k+0
CC=#FIN if R_1=s-r+1
CC=#G0 if R_1=s
R_1=R_1+1
ITER
CC=#LOOP1 if l=r
l=l+1
R_1=R_1+l+1
GO
l=l-1
R_1=R_1+l+1
CC=#GO if R_1=s-r+l+1
CC=#ITER
FIT
B2
RPA
CLA
PR
x=B t
T4
STO
 PRINT x
IF(SLN)SKP
+40000
TRA
#TRAC2

TRAC2
CC=#PS if 4*Co_1,1<x_1,2
CC=#PS if 4*Co_2,2<x_2,2
CC=#PS if 4*Co_3,3<x_3,2
y=D x
EXECUTE VSPACE(z,s)
EXECUTE VSPACE(w,s)
ASSIGN (con.)

q=0
FOR i=1,1,s
FOR a=1,1,Sp
CC=#ROUND if TESTP<\mid T_{i,a}-y_1 \mid
q=q+1
z=a
w_q=i
CC=#FOUND
ROUND REPEAT
FOUND REPEAT
EXECUTE CONTR(0,+1150,q+1,z)
EXECUTE CONTR(0,+1150,q+1,w)
EXECUTE MPART(Np,Cp,Bp,D,w)
EXECUTE VSPACE(Tp,q)
FOR i=1,1,q
TP_i=T_{w_i,z_i}
REPEAT
SIGMA=Tp Cp Tp
x=Bp Tp
EXECUTE PUTIN(As,LINES,q,SIGMA,x,Np,w,z)
IF(SLN) SKP +20000
TRA #PS
PRINT q,z,w,Tp,Cp,SIGMA,Bp,x,Np,As
CC=#PS
UV j=k+1
CC=#LOOP
PS SB2 *BTWSV
CLA T,U->B1
CLA B1,U->B1
B1 ADD d,U->B3
TRA #BACK
FIN i=ROW(As)
EXECUTE MINSE(+77776,i,+1,As)
EXECUTE MINSE(+77776,i,+1,LINES)
END
DEFINE
LEAVE
PUTIN

DEFINE
VECTOR x,w,z
INTEGER n,q
MATRIX As,LINES
FUNCTION ACOPY

PUTIN(As,LINES,q,SIGMA,x,n,w,z).=SEQ
INTEGER i,j,k
i=ROW(As)
FOR j=1,1,1
FOR k=3,1,5
CC=#OUT if As ,v=x,k-2
REPEAT
CC=#END
OUT
REPEAT
EXECUTE MINSE(+1,0,+1,As)
As,1=q
As,2=SIGMA
As,3=x
As,4=x
As,5=x
As,6=n
EXECUTE CONTR(+1,+150,0,LINES)
EXECUTE CONTR(0,+600,0,LINES)
EXECUTE CONTR(+2,+120,+402,i+1)
EXECUTE CONTR(0,+700,0,Z)
EXECUTE ACOPY(w,Z,Z,LINES,i,1)
EXECUTE ACOPY(z,Z,Z,LINES,i,2)
END
DEFINE

LEAVE
START

\( i = \text{ROW}(A S) \)

FOR \( j = 1, 1, 1 \)

FOR \( k = 3, 1, 5 \)

\( A_s, k = x_{k-2} \) ?

NO

MINSE(\(+1, 0, +1, A S\))

YES

\( A_s, 1 = q \)
\( A_s, 2 = \Sigma \)
\( A_s, 3 = x_1 \)
\( A_s, 4 = x_2 \)
\( A_s, 5 = x_3 \)
\( A_s, 6 = n \)

CONTR(\(+1, +1150, 0, \text{LINES}\))
CONTR(\(0, +0600, 0, \text{LINES}\))
CONTR(\(+2, +0120, +4402, i + 1\))
CONTR(\(0, +0700, 0, Z\))

ACOPY(\(w, Z, Z, \text{LINES, i, 1}\))
ACOPY(\(z, Z, Z, \text{LINES, i, 2}\))

END
DEFINE
MATRIX AA, BB, c, DD, EE
INTEGER FF
VECTOR V
MPART(FF, AA, BB, c, V) = SEQ
INTEGER i, j
j = LENGTH(V)
EXECUTE MSPACE(BB, 1, 3)
FOR j = 1, 1, 1
EXECUTE MPATCH(1, 3, V, 1, c, j, 1, BB)
REPEAT
AA = TRAN(BB) BB
EXECUTE MSPACE/DD, 3, 3/
EXECUTE DIAG(AA, EE, 0)
FF = 0
FOR j = 1, 1, 3
CC = IF NOT IF AA, J, J < 1 * 3
DD, J, J = 1 / AA, J, J
FF/*M*/ = 1
REPEAT
IF (SLN) SKIP 10000
TRA
PRINT BB, AA, DD, EE
TRAC
DD = TRAN(EE) DD EE
EXECUTE MSPACE(AA, 1, 1)
AA = AA + (-1)(BB DD TRAN(BB))
BB = DD TRAN(BB)
END
DEFINE
LEAVE
START

\( i = \text{LENGTH}(V) \)

FOR \( j = 1, 1, 1 \)
EXECUTE MPATCH(V_j)

AA = BB BB

MSPACE(DD)

DIAG(AA, EE, O)

FP = O

NOT FOR \( j = 1, 1, 3 \)

\( AA_{j,j} < 1.0 \ast -3 \)

YES

NO

DD_{j,j} = 1.0 / AA_{j,j}

FP = FP + 1

DD = EE DD EE

AA = 1 - (BB DD BB)

BB = DD BB

END
DEFINE
MATRIX T, A, M, EXPM, LINES, Dp, C1, Bl
FUNCTION ACOPY, ITER, DERIV, MPART, KNU
MATRIX MSV
SCALAR MXSIG, TEST1
INTEGER MN_NUM
VECTOR Tp, X
INTEG VECTOR RW, CL
REFIT(As, M, EXPM, T, LINES, A1, B1, C1) = SEQ
INTEGER i, j, k, l, n, m
INTEGER NUMB
MSV = M
i = ROW(As)
FOR j = 1, 1, 1
k = As
CC = \#H1 if k < MN_NUM or MXSIG < As_j, 2
n = As_j, 1
m = As_j, 6
A = A1 + As_j, 3
B = B1 + As_j, 4
C = C1 + As_j, 5
SIGMA = As_j, 2
PRINT n, SIGMA, m
DA = As_j, 3
DB = As_j, 4
DC = As_j, 5
PRINT DA, DB, DC
EXECUTE ACOPY(LINES, j, 1, RW, Z, Z)
EXECUTE ACOPY(LINES, j, 2, CL, Z, Z)
PRINT RW, CL
EXECUTE VSPACE(Tp, n)
EXECUTE ITER(N, A, B, C)
FOR l = 1, 1, n
TP1 = Tp, CL_1 + MSV RW_1, 7 - M RW_1, 7
REPEAT
EXECUTE DERIV(Dp, M, A, B, C)
EXECUTE MPART(m, C1, Bl, Dp, RW)
SIGMA = Tp C1 Tp
X = Bl Tp
PRINT n, SIGMA, m
PRINT X
A = A + X
B = B + X
CC = \#OUT if 1.1 MXSIG < SIGMA
SHIFT = X X

D-29
REFIT (con.)

CC = #AGAIN if 1.*-4<SHIFT
     PAG
PRINT A, B, C
EXECUTE MINSE(+2, 0, +2, M)
l=ROW(M)
k=ROW(EXPM)
NUMB = 1
FOR m=1,-1,1
NU=M
FOR n=1,1,k
CC=#NO if TEST1<\(NU-EXPM, n, 1\)
M_n, 9=EXPM
M_n, 10=EXPM\_n, 2
CC=#BY2
NO REPEAT
NUMB = NUMB - 1
BY2 REPEAT
EXECUTE KNU(M)
PRINT NUMB
CC = #OUT if NUMB<MNUM
PRINT M
     PAG
M=MSV
BY REPEAT
CC = #END
OUT PRINT BAD
M=MSV
CC= #BY
END DEFINE
LEAVE