MACS M
A PROGRAM FOR COMPUTING THE CRYSTAL DYNAMICAL MATRIX, PHONON
DISPERSION RELATIONS, AND STRUCTURE FACTORS FOR
NEUTRON INELASTIC SCATTERING

C. Noble, A. W. Solbrig, Jr., and H. L. McMurry

MASTER

IDAHO NUCLEAR CORPORATION
NATIONAL REACTOR TESTING STATION
IDAHO FALLS, IDAHO

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

U. S. ATOMIC ENERGY COMMISSION

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

MACS M

A PROGRAM FOR COMPUTING THE CRYSTAL DYNAMICAL MATRIX, PHONON DISPERSION RELATIONS, AND STRUCTURE FACTORS FOR NEUTRON INELASTIC SCATTERING

by

C. Noble
Phillips Petroleum Company

A. W. Solbrig, Jr. and H. L. McMurry
Idaho Nuclear Corporation

IDAHO NUCLEAR CORPORATION

A JOINTLY OWNED SUBSIDIARY OF
AEROJET ALLIED
GENERAL CHEMICAL
CORPORATION CORPORATION

U. S. Atomic Energy Commission Research and Development Report
Issued Under Contract AT(10-1)-1230
Idaho Operations Office

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

MACS M

by

C. Noble
Phillips Petroleum Company

A. W. Solbrig, Jr. and H. L. McMurry
Idaho Nuclear Corporation

ABSTRACT

The FORTRAN IV program MACS M (PPCo 40*0650) computes the dynamical matrix for the harmonic oscillations of a crystal, the eigenvalues and eigenvectors of this matrix, and the corresponding structure factors for coherent single-phonon scattering of neutrons. The calculation of the dynamical matrix is especially adapted to the use of a potential in valence coordinates, but the input allows use of a potential in Cartesian coordinates.

Running times on an IBM 7040 (32K, batch processing) for calculating a dynamical matrix and solving the corresponding secular equation are about one minute for diamond (two atoms per primitive cell) and two minutes to three minutes for beryllium oxide (four atoms per primitive cell).

CONTENTS

	<u>Page No.</u>
ABSTRACT.	iii
I. GENERAL DESCRIPTION OF PROGRAM	1
II. FORMULAS AND CALCULATION METHODS	3
A. Dynamical Matrix.	3
B. Structure Factors	7
C. Subroutines	10
D. Restrictions.	10
III. INPUT-OUTPUT	11
IV. REFERENCES	20
APPENDIX A - SAMPLE INPUT	21
APPENDIX B - FLOW CHART FOR MACS M.	24
APPENDIX C - FLOW CHART FOR STRUCTURE FACTOR CALCULATION.	34

FIGURES

1. Input Cards for MACS M	18
2. Input Cards for Structure Factor Data.	19

MACS M

I. GENERAL DESCRIPTION OF PROGRAM

The FORTRAN IV program MACS M (PPCo 40*0650) computes certain quantities which are of interest in the theory of harmonic oscillations of a crystal and in the theory of inelastic scattering of neutrons from a crystal. These quantities are the dynamical matrix, eigenvalues and eigenvectors of this matrix, and corresponding structure factors for coherent single-phonon scattering of neutrons. The eigenvalues lead directly to the dispersion relations for the crystal. The structure factors are useful in planning and interpreting certain neutron scattering experiments. The structure factors are also an important part of the calculation of cross sections for inelastic scattering of neutrons from crystalline or polycrystalline materials. The calculation of the dynamical matrix is especially adapted to the use of a potential in valence coordinates⁽¹⁾, but the input allows use of a potential in Cartesian coordinates. The formulas for the dynamical matrix are derived (in somewhat different notation) in Ref. 1. The formulas for the structure factor are taken from Ref. 2.

Some features of the program are described briefly here. (More detailed description of most of these features is given in Section II.)

a. The program permits entry of real symmetry coordinates used to factor the secular equation. The only requirement on the symmetry coordinates is linear independence: The program orthonormalizes the coordinates by a Gram-Schmidt procedure beginning with the last coordinate. (Complex symmetry coordinates may be needed to obtain

the maximum factoring possible on the basis of symmetry, but helpful factoring is usually obtainable with real symmetry coordinates.)

b. The program permits entry of a real transformation matrix used to change the representation of eigenvectors from one basis to another basis.

c. An initial set of force constants must be supplied for the calculation. Often it is desired to observe the effects of changes in the force constants. A small amount of additional input causes the machine to perform calculations with changed force constants. The initial constants remain on tape for use as desired.

d. The dynamical matrix is Hermitian, and it often contains complex entries. Since the available subroutines for calculating eigenvalues and eigenvectors were adapted to real symmetric matrices only, we resorted to a trick in order to obtain the eigenvalues and eigenvectors of the dynamical matrix. This trick involves expanding an $N \times N$ complex Hermitian matrix into a $2N \times 2N$ real symmetric matrix. As a result, the program prints $2N$ eigenvalues, $2N$ eigenvectors (each consisting of N complex numbers), and $2N$ structure factors. Not more than N of the eigenvectors may be linearly independent. If the dynamical matrix has a degenerate eigenvalue, then the list of $2N$ eigenvectors may contain fewer than N linearly independent eigenvectors.

e. Each normalized eigenvector of a Hermitian matrix contains an arbitrary factor e^{ia} , where a is real. The program selects this factor to make the first nonzero entry in each eigenvector to be real and positive. The complex entries in the eigenvectors are printed as r and θ , where $0^\circ \leq \theta \leq 180^\circ$ and r may be negative. We say that our eigenvectors in this form are in "pseudopolar" form.

f. The dynamical matrix is Hermitian; but the numerical approximation to this matrix may fail to be exactly Hermitian, and the numerical approximation may have small entries where (according to symmetry arguments) zeros should appear. The program makes the matrix Hermitian and sets small entries equal to zero before the eigenvalue-eigenvector calculation is performed.

g. The results of a trace check on the eigenvalues are printed with every set of eigenvalues.

We have observed that in the case of a 12 x 12 complex Hermitian dynamical matrix with an inappropriate choice of symmetry coordinates, the program may present irregular results (in spite of the precautions mentioned in item f above): Some of the eigenvalues may be slightly in error, their multiplicities may be incorrect, and some of the eigenvectors may be grossly in error. A poor trace check and failure of eigenvalues or eigenvectors to occur in pairs are indications of trouble. We found it possible to obtain good results by factoring the secular equation with better symmetry coordinates into an 8 x 8 and a 4 x 4.

II. FORMULAS AND CALCULATION METHODS

A. Dynamical Matrix

The dynamical matrix is ⁽¹⁾

$$\underline{\underline{M}} = \underline{\underline{A}} \underline{\underline{D}} \underline{\underline{C}}^T \underline{\underline{F}} \underline{\underline{D}} \underline{\underline{A}}^T ;$$

but in the present program only real A matrices are allowed and

$$\underline{\underline{M}} = \underline{\underline{A}} \underline{\underline{D}} \underline{\underline{C}}^T \underline{\underline{F}} \underline{\underline{D}} \underline{\underline{A}}^T .$$

Here

$\underline{A}$ is a square or rectangular matrix with linearly independent symmetry coordinates as rows,

$\underline{A}^*$ is the orthonormalized form computed by the program,

$$\underline{D} = \sum_{\ell} \underline{D}_{0,\ell} e^{-i\kappa \cdot \underline{r}_{\ell}}$$

and

$$\underline{F} = \sum_{\ell} \underline{F}_{0,\ell} e^{-i\kappa \cdot \underline{r}_{\ell}}.$$

The

$$\kappa = 2\pi(f_1 \underline{b}_1 + f_2 \underline{b}_2 + f_3 \underline{b}_3)$$

and

$$\underline{r}_{\ell} = \ell_1 \underline{a}_1 + \ell_2 \underline{a}_2 + \ell_3 \underline{a}_3,$$

where $\underline{a}_1$, $\underline{a}_2$, and $\underline{a}_3$ are the primitive translation vectors of the lattice and $\underline{b}_1$, $\underline{b}_2$, and $\underline{b}_3$ are the associated reciprocal lattice vectors; so $\kappa \cdot \underline{r}_{\ell} = 2\pi(f_1 \ell_1 + f_2 \ell_2 + f_3 \ell_3)$.

The $\underline{D}_{0,\ell}$ matrices express the transformation which relates changes in valence coordinates (bond distance changes, interbond angle changes, etc.) to the mass-adjusted Cartesian displacements of the atoms in the primitive cells near the origin cell. The physical significance and properties of these matrices are discussed further in Ref. 1. A machine program for calculating them has also been prepared⁽³⁾. The $\underline{F}_{0,\ell}$ are force constant matrices. A discussion of how to choose valence force coordinates to fit the needs of crystal calculations is given in Ref. 1. Only the $\underline{F}_{0,\ell}$ matrices with $\ell \geq 0$ are given as input data to the program because $\underline{F}_{0,-\ell} = \underline{F}_{0,\ell}^T$. Only

the diagonal and superdiagonal elements of $\underline{\underline{F}}_{0,0}$ need to be given as input data for the program because $\underline{\underline{F}}_{0,0} = \underline{\underline{F}}_{0,0}^T$. If the force constants are in terms of mass-adjusted Cartesian coordinates, then the user may want $\underline{\underline{D}}_{0,\ell} = 0$ for $\ell \neq 0$ and $\underline{\underline{D}}_{0,0} = \underline{\underline{I}}$ (an identity matrix compatible in size with $\underline{\underline{F}}$)⁽⁴⁾.

In practice, the computed matrix $\underline{\underline{M}}$ may fail to be exactly Hermitian. To form a Hermitian matrix and to allow use of an eigenvalue-eigenvector subroutine for real symmetric matrices, the following operations are performed on $\underline{\underline{M}}$.

$$\text{Re}(m_{ij}) = \frac{1}{2}(\text{Re}(m_{ij}) + \text{Re}(m_{ji}))$$

$$|\text{Im}(m_{ij})| = \frac{1}{2}|\text{Im}(m_{ij})| + \frac{1}{2}|\text{Im}(m_{ji})|$$

The sign of the imaginary part is left unchanged.

Next write $\underline{\underline{M}} = \underline{\underline{M}}_1 + i \underline{\underline{M}}_2$, where $\underline{\underline{M}}_1$ and $\underline{\underline{M}}_2$ are real, and set diagonal elements of $\underline{\underline{M}}_2$ equal to zero. Then form the "expanded" real symmetric matrix

$$\underline{\underline{M}}' = \begin{bmatrix} \underline{\underline{M}}_1 & -\underline{\underline{M}}_2 \\ \underline{\underline{M}}_2 & \underline{\underline{M}}_1 \end{bmatrix}.$$

It may be shown that if λ is an eigenvalue of $\underline{\underline{M}}$ (Hermitian) of multiplicity n , then λ is an eigenvalue of $\underline{\underline{M}}'$ of multiplicity $2n$, and if $(x_1 + iy_1, x_2 + iy_2, \dots, x_p + iy_p)$ is an eigenvector of $\underline{\underline{M}}$ corresponding to λ then $(x_1, x_2, \dots, x_p; y_1, y_2, \dots, y_p)$ is an eigenvector of $\underline{\underline{M}}'$ corresponding to λ ; and conversely.

To eliminate small terms in $\underline{\underline{M}}'$ that occur due to round-off error, the following procedure is applied:

1. Suppose i and j are fixed.

2. Let V represent the element with largest absolute value in row i .
3. Let U represent the element with largest absolute value in column j .
4. Put $W = \text{Max}(V, U)$.
5. If $|m'_{ij}| < \sigma W$, then m'_{ij} is set equal to zero. (In the present program, $\sigma = 10^{-4}$.)

The procedure is applied to all i, j in $\underline{M}'$.

If masses are in atomic mass units and force constants are in millidynes /Å (See Table II of Ref. 1), then the printed "raw eigenvalues" are converted by the program as follows:

$$\text{wave numbers} = 1303.061 \sqrt{\text{raw eigenvalues}}$$

$$\text{electron volts} = \text{wave numbers} / 8066$$

Notice that (raw eigenvalue) $\times (6.02252 \times 10^{28})$ yields (angular frequency)² in (rad/sec)². If a raw eigenvalue is negative, the program prints asterisks for "wave numbers" and "electron volts".

(Negative raw eigenvalues may occur if $\underline{M}$ is not a positive matrix. In this case, some of the parameters entered into the program may be erroneous or not physically meaningful.)

The eigenvectors are returned by the subroutine EIGEN (PPCo 40*0566) in Cartesian form, but MACS $\underline{M}$ prints eigenvectors in "pseudopolar" form. This pseudopolar form is obtained as follows: Let an eigenvector be

$$\underline{V} = (V_1, V_2 \cdots V_n), V_i = x_i + iy_i = (x_i, y_i).$$

Write $V_i = (r_i, \theta_i)$, where $r_i = (x_i^2 + y_i^2)^{\frac{1}{2}}$

$$\theta_i = \tan^{-1} \frac{y_i}{x_i}$$

$$0 \leq \theta_i < 360^\circ.$$

Let r_k be the radial component of the first nonzero entry in V , so

$$e^{-i\theta_k} \underline{y} = ((0,0), \dots, (r_k, 0), (r_{k+1}, \theta_{k+1} - \theta_k), \dots, (r_n, \theta_n - \theta_k)).$$

Let

$$\theta'_{k+l} = \theta_{k+l} - \theta_k \quad l = 1, 2, \dots, n-k.$$

If

$$\theta'_{k+l} < 0 \text{ then } \theta'_{k+l} = \theta'_{k+l} + 360^\circ.$$

If

$$\theta'_{k+l} \geq 180^\circ \text{ then } \theta'_{k+l} = \theta'_{k+l} - 180^\circ$$

and

$$r_{k+l} = -r_{k+l}.$$

Three r, θ' pairs are printed out on each line. "Junk" numbers may appear at the end of an eigenvector if the eigenvector does not fill its last line. (Example: If an eigenvector contains 8 r, θ' pairs, then the printed entry in position 9 is "junk".)

B. Structure Factors

The structure factor is given by⁽²⁾

$$\begin{aligned} g_j^2(\underline{\kappa}, \underline{\tau}) &= \frac{1}{\omega_j(\underline{\kappa})} |Q \cdot \underline{\beta}_j(\underline{\kappa}, \underline{\tau})|^2 \\ &= \frac{1}{\omega_j(\underline{\kappa})} |\underline{Q} \cdot \hat{\underline{\beta}}_j(\underline{\kappa}, \underline{\tau})|^2 [\gamma_j(\underline{\kappa}, \underline{\tau})]^2, \end{aligned}$$

where

$$\begin{aligned}
 \beta_j(\underline{k}, \tau) &= \sum_{k=1}^N \frac{b_k}{\sqrt{m_k}} \varepsilon_{kj}(\underline{k}) e^{i\mathbf{Q} \cdot \mathbf{R}_k} \\
 &= \beta_{j1}\hat{i} + \beta_{j2}\hat{j} + \beta_{j3}\hat{k} \\
 &= \pm |\beta_{j1}| e^{i\theta_{j1}} \hat{i} \pm |\beta_{j2}| e^{i\theta_{j2}} \hat{j} \pm |\beta_{j3}| e^{i\theta_{j3}} \hat{k} \\
 &\quad \theta_{jp} \leq \theta_{jp} < \pi \quad (p = 1, 2, 3) \\
 &= c_{j1} e^{i\theta_{j1}} \hat{i} + c_{j2} e^{i\theta_{j2}} \hat{j} + c_{j3} e^{i\theta_{j3}} \hat{k} \\
 &= \hat{\beta}_j(\underline{k}, \tau) \gamma_j(\underline{k}, \tau) \\
 &= (\hat{c}_{j1} e^{i\theta_{j1}} \hat{i} + \hat{c}_{j2} e^{i\theta_{j2}} \hat{j} + \hat{c}_{j3} e^{i\theta_{j3}} \hat{k}) \cdot \gamma_j
 \end{aligned}$$

$$\hat{c}_{jp} = \frac{c_{jp}}{\gamma_j} \quad (p = 1, 2, 3)$$

$$\gamma_j \geq \sqrt{c_{j1}^2 + c_{j2}^2 + c_{j3}^2} = |\beta_j|$$

N = Number of atoms/unit cell = 1/3 of number of columns in D

b_k = Bound coherent scattering length for atom k

m_k = Mass of atom k

$\mathbf{R}_k$ = Position vector of atom k in origin cell

$$= R_{k1}\hat{i} + R_{k2}\hat{j} + R_{k3}\hat{k}$$

$\xi_{kj}(\kappa)$ = Components of normalized eigenvector j associated with atom k (after basis change by multiplying eigenvector row matrix from right by $\underline{\underline{A}}^*$)

ω_j = eigenvalue j (electron volts)

$\underline{\underline{\tau}}$ = $t_1 \underline{\underline{b}}_1 + t_2 \underline{\underline{b}}_2 + t_3 \underline{\underline{b}}_3$ (a reciprocal lattice vector)

$\underline{\underline{Q}}$ = $2\pi \underline{\underline{\tau}} - \underline{\underline{\kappa}}$

In order to compute dot products it is necessary to consider bases for the vectors involved. Both $\underline{\underline{R}}_k$ and $\xi_{kj}(\kappa)/\sqrt{m_k}$ are represented in the Cartesian $\hat{i}, \hat{j}, \hat{k}$ basis. Write

$$\underline{\underline{b}}_1 = b_{11}\hat{i} + b_{12}\hat{j} + b_{13}\hat{k}$$

$$\underline{\underline{b}}_2 = b_{21}\hat{i} + b_{22}\hat{j} + b_{23}\hat{k}$$

$$\underline{\underline{b}}_3 = b_{31}\hat{i} + b_{32}\hat{j} + b_{33}\hat{k}.$$

If the input data include the matrix

$$\underline{\underline{B}} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

then the dot products may be evaluated by changing $\underline{\underline{Q}}$ from the $\underline{\underline{b}}_1, \underline{\underline{b}}_2, \underline{\underline{b}}_3$ representation to the $\hat{i}, \hat{j}, \hat{k}$ representation.

On the output listing, $|\underline{\underline{Q}} \cdot \underline{\underline{\beta}}_j|^2$ is labeled "polarization factor", γ_j^2 is labeled "modified structure factor", and the product of these two factors is labeled "product". On the line above each modified structure factor, the quantities $\hat{C}_{j1}, \theta_{j1}, \hat{C}_{j2}, \theta_{j2}, \hat{C}_{j3}, \theta_{j3}$ (with the θ 's in degrees) are printed.

C. Subroutines

The following subroutines (in FORTRAN IV) are included in the program:

<u>File Number</u>	<u>Name</u>
40*0566	Eigenvalues and Eigenvectors of a Real Symmetric Matrix (EIGEN)
40*0333	Matrix Read-Write (WRIM)
40*0542	Complex Matrix Read-Write (WRIMC)
40*0342	Angle Function (ANGD)

Decks of MACS M including the above subroutines are available from Phillips Computing Center, P. O. Box 2067, Idaho Falls, Idaho 83401. Writeups of the subroutines are also available from the Computing Center.

D. Restrictions

This program may use as many as 6 tapes. Fortran units 2 and 3 are always used. If the various tape options are used, then additional tapes are needed. The $D_{0,l}$ are on Fortran unit 8, $F_{0,l}$ are on Fortran unit 9, f 's are on Fortran unit 0 and A is on Fortran unit 4. The number of $D_{0,l}$ matrices with positive subscripts must be less than or equal to 40. The largest input matrix the program will now accept is 32 x 32. None of the input matrices can have more than 32 rows or 32 columns. The calculated complex Hermitian dynamical matrix M may not be larger than 16 x 16. (If the user wishes to increase the allowed matrix size, he should increase $N4$ from its present value of 32 and change all dimension statements involving the number 32.)

III. INPUT-OUTPUT

All matrices are entered in the following format (WRIM format): Each card contains six twelve-digit numbers in E format (example: 1.03E+01) or F format (example: 10.3). Card columns 73-74 indicate the row that the values on the card are to occupy. Card columns 75-76 indicate the column of the matrix that the value in word one of the card is to occupy. The remaining values on the card will occupy successive column locations in the given row of the matrix. Card column 77 indicates the total number of values on the card. Card columns 78-80 are not used by this subroutine, so these columns can be used for identification or not as the user sees fit. Before a matrix is called into core, the complete area that it will occupy is set to zero; consequently, it is necessary to enter only nonzero values into the computer.* If a card contains one or more zero matrix elements, such elements may be represented by a 0.0 or by no punches at all. Since each card contains its own input information, values may be entered in any order.* A sample input matrix in WRIM format, along with some other sample input, is given in Appendix A.

The output format for real matrices corresponds to the WRIM input format, except that meaningless "junk" numbers may appear at

* Exception: The loading of a matrix is terminated by the subroutine WRIM storing the final value of the highest numbered row. For example, in a 24×27 matrix, when the element (24,27) is stored then the loading process is terminated and the matrix is assumed to be complete.

the end of matrix rows. (Example: A real 3 x 3 matrix will occupy only the first three positions of each printed line, and the last three positions may contain meaningless "junk".)

No complex matrices are accepted by this program as input, but complex matrices ($\underline{F}$, $\underline{D}$ and $\underline{M}$) may be printed as output in WRIMC format. This format corresponds closely to WRIM format except that each row contains three complex numbers in "x,y" form. Two successive words are used for the real or imaginary parts of each complex number.

Now we list and describe the input cards.

Control Card 1

Column 5	ILTAPE
10	IFTAPE
15	IVTAPE
20	IATAPE
25	KLTAPE
30	KFTAPE
35	KVTAPE
40	KATAPE
41-45	N2
46-50	N3
51-55	N5
56-60	NUMAT
65	ISTACK
66-70	(blank)
75	ISTRU

Column 5	$\underline{D}_{0,l}$ matrices	} 0: quantity read from tape 1: quantity read from cards
10	$\underline{F}_{0,l}$ matrices*	
15	f_1, f_2, f_3	
20	$\underline{A}$ matrix	

* Since $\underline{F}_{0,0} = \underline{F}_{0,0}^T$, only the diagonal and superdiagonal elements of $\underline{F}_{0,0}$ need to be entered as data. The program takes the diagonal and superdiagonal elements and constructs a symmetric $\underline{F}_{0,0}$ from these entries.

25	$\underline{D}_{0,l}$ matrices	} 0: quantity not stored on tape 1: quantity stored on tape
30	$\underline{F}_{0,l}$ matrices	
35	f_1, f_2, f_3	
40	$\underline{A}$ matrix	
41-45	Number of columns in $\underline{A}$ (= number of columns in $\underline{D} = 3N$)	
46-50	Number of rows in $\underline{F}$ (= number of columns in $\underline{F}$ = number of rows in $\underline{D}$)	
51-55	Number of rows in $\underline{A}$	
56-60	Number of $\underline{D}_{0,l}$ matrices with $l > 0$	
65	{ 1 (If columns 5-40 call for no use of tape in a problem, then the program user may save machine time by putting 0 in column 65 for that problem.)	
66-70	(blank)	
75	{ 1: Structure factor input is read from cards and structure factors are computed. 0: Structure factor input is not read from cards and structure factors are not computed.	

Control Card 2

	Column 5	IOP1
	10	IOP2
	15	IOP3
	20	IOP4
	25	IOP5
	30	IOP6
	35	IOP7
	40	IOP8
	45	IOP9
	50	IOP10
	55	IOP11
	60	IOP12
	65	IOP13
	{ 0: quantity is printed 1: quantity is not printed	
Column 5	$\underline{D}_{0,1}, \underline{D}_{0,2}, \dots, \underline{D}_{0,NUMAT}$	
10	$\underline{D}_{0,0}$	
15	$\underline{D}_{0,-1}, \underline{D}_{0,-2}, \dots, \underline{D}_{0,-NUMAT}$	

Column 20	$\underline{D}$	} Only the unchanged $\underline{F}_{0,l}$ are printed (See Control Card 3). The unchanged $\underline{F}_{0,l}$ remain on tape for use as desired.
25	$\underline{F}_{0,1}, \underline{F}_{0,2}, \dots, \underline{F}_{0,NUMAT}$	
30	$\underline{F}_{0,0}$	
35	$\underline{F}$	
40	$\underline{A}$	
45	$\underline{A}$ (normalized)	
50	$\underline{A}^* = \underline{A}$ (orthonormalized)	
55	$\underline{A} \underline{D}^{CT}$	
60	$\underline{A} \underline{D}^{CT} \underline{F}$	
65	$\left\{ \begin{array}{l} \underline{M} \text{ (raw)} \\ \underline{M} \text{ (Hermitian, except that some diagonal elements} \\ \text{may have small nonzero imaginary part)} \\ \underline{M} \text{ (expanded; real, symmetric)} \end{array} \right.$	

Control Card 3

Column 1-5	NOCH (right adjusted) is the number of changes to be made in the $\underline{F}_{0,l}$ matrices. These changes are read from Card Types 4 and 5.
10	KBASIS: If $\neq 0$, a transformation matrix $\underline{A}_1$ is read from cards and $\underline{E} \underline{A}_1$ is computed, where the rows of $\underline{E}$ are the eigenvectors.
11-15	N0 (right adjusted) is the number of columns in $\underline{A}_1$. (If column 10 contains zero, then column 15 should be zero.) Note that number of rows in $\underline{A}_1$ = number of rows in $\underline{A}$ (or $\underline{A}^*$).

If NOCH $\neq 0$, the next two cards indicate the changes to be made in the $\underline{F}_{0,l}$ matrices:

Card Type 4

Columns 1-10 X_1 (F format)

11-20 X_2
 $\vdots$

Card Type 5

Columns 1-10 Y_1 (F format)

11-20 Y_2
 $\vdots$

Each $F_{0,l}$ matrix is scanned and if any element is equal to X_i , then this element is replaced by Y_i . Notice that numbers of X's = number of Y's = NOCH = punch in columns 1-5 of Control Card 3. If NOCH = 0, then Card Types 4 and 5 are omitted. If NOCH > 8, then there are more than one of Card Type 4 and more than one of Card Type 5.

Next, the necessary cards are entered in the following order:

f_1, f_2, f_3 <u>on one card</u>	Columns 1-10	f_1 (F format)
	11-20	f_2
	21-30	f_3

ℓ_1 vector*

$D_{0,1}$ matrix

ℓ_2 vector

* Notes: (a) The ℓ_j vectors are read in one per card,

$\ell_j = (\ell_{1j}, \ell_{2j}, \ell_{3j})$:

Columns 1-5 ℓ_{1j} (integer format)

6-10 ℓ_{2j}

11-15 ℓ_{3j}

(b) In accord with the discussion in Ref. 1, the program uses

$\ell_j = -\ell_j$; so only ℓ 's with positive subscripts are entered as data.

$\underline{D}_{0,2}$ matrix
 $\vdots$
 $\underline{L}_{\text{NUMAT}}$ vector
 $\underline{D}_{0,\text{NUMAT}}$ matrix
 $\underline{D}_{0,0}$ matrix
 $\underline{D}_{0,-1}$ matrix
 $\vdots$
 $\underline{D}_{0,-\text{NUMAT}}$ matrix
 $\underline{F}_{0,1}$ matrix
 $\underline{F}_{0,2}$ matrix
 $\vdots$
 $\underline{F}_{0,\text{NUMAT}}$ matrix
 $\underline{F}_{0,0}$ matrix
 $\underline{A}$ matrix

These cards are followed by the structure factor input (if the punch in column 75 of Control Card 2, ISTRU, is nonzero).

Card Type S1 contains the first $\underline{t}$ -vector.

Columns 1-10	t_1	(F format)
11-20	t_2	
21-30	t_3	

Card Type S2

Columns 1-5	NTAU = number of $\underline{t}$ vectors to be read in (integer format)
-------------	--

Card Type S3

Columns 1-10	m_1	(F format)
11-20	m_2	
$\vdots$	$(m_k, k = 1, \dots, N)$	

Card Type S4

Columns 1-10 b_1 (F format)
 11-20 b_2
 $\vdots$
 ($b_k, k = 1, \dots, N$)

Cards Type S5

$\underline{B}$ matrix (3 x 3)

Cards Type S6

$\underline{R}$ matrix (N rows and 3 columns)

Cards Type S7

Remaining $\underline{t}$ -vectors (if the punch in column 1-5 of Card
Type S3, NTAU, > 1).

The $\underline{A}_1$ matrix is entered next (if KBASIS x N8 $\neq$ 0) as the final set of cards in the input of a given problem. Neither the structure factor input nor the $\underline{A}_1$ matrix is stored on tape: These data must be entered anew for each problem which calls for them.

If there is another problem (ISTACK = 1), then another Control Card 1 comes next in the deck.

Input cards for one problem are drawn schematically in Figs. 1 and 2.

The printed machine output is labeled so as to be self-explanatory, except for the following items: (a) The matrices labeled "F- 1", "F- 2", etc. are $\underline{F}_{0,1}, \underline{F}_{0,2}$, etc. (b) On the line above each structure factor the quantities $\hat{C}_{j1}, \theta_{j1}, \hat{C}_{j2}, \theta_{j2}, \hat{C}_{j3}, \theta_{j3}$ (with the θ 's in degrees) are printed.

Figure 1
Input Cards For MACS M

INC-C-III59

Figure 2
Input Cards For Structure Factor Data

All of these cards (with the possible exception of Cards Type S7) are needed for every problem for which structure factors are desired.

IV. REFERENCES

1. H. L. McMurry, A. W. Solbrig, Jr., J. K. Boyter and C. Noble, "The Use of Valence Force Potentials in Calculating Crystal Vibrations," accepted for publication in Journal of Physics and Chemistry of Solids.
2. B. N. Brockhouse and P. K. Iyengar, "Normal Modes of Germanium by Neutron Spectroscopy," Phys. Rev. 111, 747-754 (1958).
3. J. K. Boyter and H. L. McMurry, "D Matrix: A Computer Program for Calculating Valence Coordinates in Terms of Atom Displacements of a Molecule or Crystal," IN-1076 (May 1967). The punched card output from this program can be used directly as the $\underline{L}_j$ and $\underline{D}_{0,j}$ punched card input to MACS M.
4. M. Born and K. Huang, Dynamical Theory of Crystal Lattices, Oxford University Press, 1954, Equation (38.21).

APPENDIX A

SAMPLE INPUT

These samples are taken from input for two beryllium oxide problems. The only matrix shown is the B matrix,

$$\underline{B} = \begin{bmatrix} 0.370645 & -0.213992 & 0 \\ 0.370645 & 0.213992 & 0 \\ 0 & 0 & 0.228467 \end{bmatrix},$$

in WRIM format. The following additional input (not shown) is needed to run the problems:

$$\begin{array}{l} \underline{l}_1 \\ \underline{D}_{0,1} \text{ (32 x 12)} \\ \vdots \\ \underline{l}_5 \\ \underline{D}_{0,5} \text{ (32 x 12)} \\ \underline{D}_{0,0} \text{ (32 x 12)} \\ \underline{D}_{0,-1} \text{ (32 x 12)} \\ \vdots \\ \underline{D}_{0,-5} \text{ (32 x 12)} \\ \underline{F}_{0,1} \text{ (32 x 32)} \\ \vdots \\ \underline{F}_{0,5} \text{ (32 x 32)} \\ \underline{F}_{0,0} \text{ (32 x 32)} \\ \underline{A} \text{ (8 x 12)} \\ \underline{R} \text{ (4 x 3) } (\underline{2} \text{ needed}) \end{array}$$

[illegible]

FORTRAN WORK SHEET

SIX 12 CHARACTER WORDS

Cards Type S5
(2 needed)

[illegible]

APPENDIX B
FLOW CHARTS FOR MAC'S M

APPENDIX C

FLOW CHARTS FOR STRUCTURE FACTOR CALCULATION