

UNCLASSIFIED

AD NUMBER
AD859289
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; APR 1969. Other requests shall be referred to Naval Air Systems Command, ATTN: AIR-6022, Washington, DC 20360.
AUTHORITY
USNASC ltr dtd 26 Oct 1971

THIS PAGE IS UNCLASSIFIED



Contract No. 00019-68-C-0274 **47**

AD 859289

COLLOCATION FLUTTER ANALYSIS STUDY

This document is subject to special export controls and transmittal to foreign governments or foreign nationals may be made only with prior approval of the Naval Air Systems Command (AIR ~~420014~~).

W. L. ... 6022
22360

VOLUME IV.

COFA - COMPUTER PROGRAM TO PERFORM FLUTTER
ANALYSIS BY THE COLLOCATION METHOD

APRIL 1969



MISSILE SYSTEMS DIVISION

HUGHES

HUGHES AIRCRAFT COMPANY

1015

COFA

COLLOCATION FLUTTER ANALYSIS STUDY

VOLUME IV

COFA - COMPUTER PROGRAM TO
PERFORM FLUTTER ANALYSIS BY THE COLLOCATION METHOD

Prepared by Dynamics & Environments Section Personnel
Hughes Aircraft Company, Missile Systems Division
Contract No. 00019-68-L-0247

APRIL 1969

This document is subject to special export controls and transmittal to foreign governments or foreign nationals may be made only with prior approval of the Naval Air Systems Command (██████████)

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
ABSTRACT.	ii
1 INTRODUCTION	1
2 SIGN CONVENTION	2
3 DERIVATION OF EQUATIONS	3
4 REFERENCES	13
5 DESCRIPTION OF PROGRAM INPUT	14
6 DESCRIPTION OF PROGRAM OUTPUT	40
7 EXAMPLE PROBLEMS	42
8 PROGRAM LISTING	62
9 FLOW DIAGRAM	101
10 NOMENCLATURE	188

ABSTRACT

A collocation solution of the flutter and vibration problems for a multiple component system is presented. The formulation utilizes structural, aerodynamic, and inertial characteristics in the form of matrices of structural and aerodynamic influence coefficients and a mass matrix, respectively, for each component. The use of a rigid-body modal matrix permits a general analysis for a system free in space with up to six rigid-body degrees of freedom.

The computer program provides the flutter or vibration solution for a system composed of as many as 20 flexible components with a maximum total of 49 collocation control points. An option is provided to vary the density as well as the reduced velocity. Another option is provided to yield the modes from a vibration analysis in a punched-card format for use in flutter analysis by modal methods.

SECTION 1

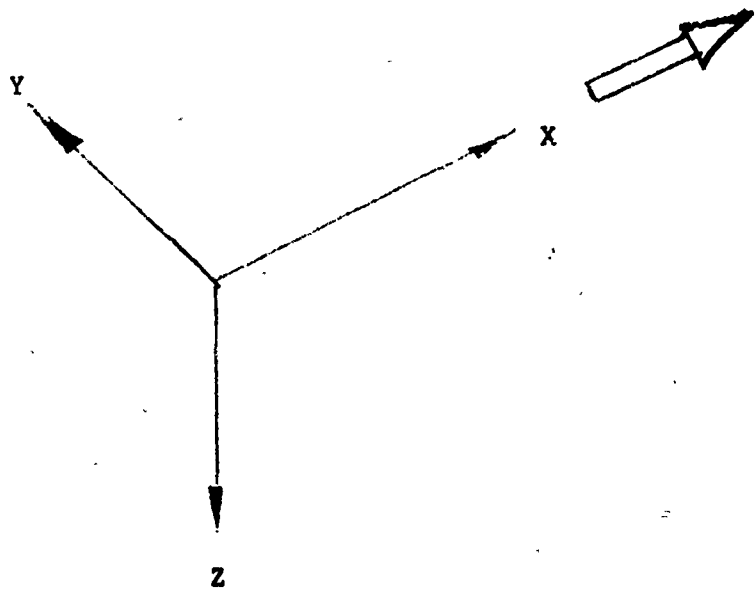
INTRODUCTION

The mathematical formulation of the flutter problem results in a set of integral equations whose closed form solution is impossible to obtain for most practical problems. One of the most useful approximate methods of solving these equations is by direct collocation. A solution by collocation is one in which the equations are satisfied at a finite number of selected points on the structure. These points, known as collocation points, are satisfied simultaneously. The collocation solution results in a matrix formulation which when cast in the canonical form will yield eigenvalues that are directly related to the flutter stability parameters. This manual presents a digital computer program that will perform collocation flutter analyses. The computer program, which is written in Fortran IV, was developed by Rodden, Farkas, and Malcom in Reference 1.

The collocation formulation of the flutter problem has been presented in Reference 2. The equations are presented for analysis of single component systems restrained (cantilevered) in space and for symmetrical systems free in space undergoing either symmetric or antisymmetric flutter. A method of generalizing the matrix equation for free-free flutter to include up to six rigid-body degrees of freedom has been given in Reference 3. The present program extends the formulation of Reference 2 to include an arbitrary combination of rigid-body degrees of freedom (Ref. 3), and to consider more than one flexible component. In addition, an option has been provided to vary the altitude (i. e. density) as well as the reduced velocity. Finally, options have been added to carry out a vibration analysis (which requires no aerodynamic data) and to provide vibration modes in punched-card format for use in a modal flutter or vibration analysis.

SECTION 2

The NASA body axis system with the x , y , and z axes directed forward, starboard, and downward, respectively, is recommended for consistency with the formulation of the static aeroelastic problems in Reference 4. However, the usual flutter convention with the x , y and z axes directed aft, starboard and downward, respectively, may be used instead. In either case, the vertical normal force and deflection are positive downward.



SECTION 3

DERIVATION OF EQUATIONS

The integral equations of aeroelasticity consist of two basic relationships: The first is the relation between the structural deformation, the structural influence function, and the inertial and aerodynamic loadings; the second is the relation between the aerodynamic disturbance (downwash), the aerodynamic influence (kernel) function, and the aerodynamic pressure. A collocation formulation of the deformation integral equation for a vehicle free in space may be written in matrix form by requiring that the integral equation be satisfied at a discrete set of control points. We choose a single type of coordinate, viz., the deflection h , as an adequate measure of both the deformation and the free-stream disturbance, not only for simplicity in the resulting equations but also because deflections have a more general meaning on a cambered vehicle and deflection influence coefficients are more readily obtained from a structural analysis than slope (or twist) influence coefficients. The resulting deformation matrix equation is

$$\{h_1\} - \{h_0\} = K[a] (\{F_i\} + \{F_a\}) \quad (1)$$

where $\{h_1\}$ is the set of components of the absolute deflections of the control points, $\{h_0\}$ is the set of components of the deflections of the control points due to the rigid-body motion of some reference points, $[a]$ is the set of structural influence coefficients (SICs, or flexibility matrix) for the system cantilevered from (or otherwise restrained at) the reference point, $\{F_i\}$ is the set of inertial force components integrated throughout the region adjacent to each control point, $\{F_a\}$ is the set of aerodynamic force components integrated over the vehicle surface adjacent to each control point, and the scalar K has been introduced as a factor to the SICs for convenience in investigating variations in stiffness levels. The inertial forces may be written in terms of a mass matrix $[M]$ and the control point accelerations.

$$\{F_i\} = -(1/386)[M]\{\ddot{h}_i\} \quad (2)$$

where the diagonal elements of the mass matrix are found from integrating the structural mass density throughout the region adjacent to the control points. (N. B., the mass matrix need not be diagonal, and, in general, will not be so if the elements must be derived from a set of weight data previously lumped at a system of control points different from those required in the aeroelastic analysis. The use of a coupled mass matrix permits simulation of given inertial characteristics at a set of control points frequently dictated by more difficult aerodynamic considerations.)

A collocation formulation of the aerodynamic integral equation is more difficult than in the case of the deformation integral equation because of the singularities in the aerodynamic kernel function. The determination of three relationships is necessary to derive a set of aerodynamic influence coefficients (AICs) that relate the control point forces to the deflections. The most basic and difficult is the pressure-downwash relation that is derived from numerical analysis of the aerodynamic integral equation. The simpler relations are the numerical integration of the pressure to obtain the force, and the numerical substantial differentiation of the deflection to obtain the downwash. The effort involved in each step depends on the planform, Mach number regime, and frequency range; a survey of the status of unsteady AICs is given in Ref. 4. For present purposes, it is sufficient to state the definition of the AICs in the oscillatory case. We write the aerodynamic control point forces in terms of the control point deflections as

$$\{F_a\} = (4\pi^2/12)\rho f^2 b_r^2 s [W] [C_h] \{h_i\} \quad (3)$$

where $[C_h]$ is the theoretically derived dimensionless (complex) matrix of oscillatory AICs, f is the frequency of the assumed harmonic motion, ρ is the atmospheric density, b_r is the reference semichord, s is the reference span, and $[W]$ is an empirically derived weighting matrix

for modification of the theoretical AICs. A method for obtaining the elements of the weighting matrix has been suggested in Ref. 4.

The sum of the force components may be written now from Eqs. (2) and (3) for the case of harmonic motion.

$$\{F_i\} + \{F_a\} = (4\pi^2 f^2 / 386)([M] + 32.174 \rho b_r^2 s[W][C_h])\{u_1\} \quad (4a)$$

$$= (4\pi^2 f^2 / 386)[\bar{M}]\{h_1\} \quad (4b)$$

We next discuss the manner of inclusion of the rigid-body degrees of freedom in Eq. (1). The matrix $\{h_o\}$ has been defined as the set of components of the deflections of the control points due to the rigid-body motion of the reference point. Each component of the control point deflections h_o is linearly related to the rigid-body translations and rotations, provided the rotations are small. Therefore, we may define a rigid-body modal matrix $|h_R|$ as the transformation

$$\{h_o\} = [h_R]\{a_R\} \quad (5)$$

where $\{a_R\}$ is the set of amplitudes of rigid-body translations and rotations of the reference point. As an example, if we consider symmetrical vertical motion, $[h_R]$ is composed of two columns: the first is a unit column corresponding to the plunging mode, the second consists of the x-coordinate of each control point corresponding to the pitching mode; $|a_R|$ is composed of two elements: the first is the plunging displacement z_o , the second is the pitching angular displacement θ .

Before proceeding in the derivation, we should review the format of the various matrices in the case of a multiple flexible component system. As an example, we consider a symmetrical flutter analysis of an aircraft whose wing, aft fuselage, and tail are flexible, and whose forward fuselage may be assumed to be rigid. We assume that the reference point (cantilever point) can be located in the vehicle such that

its various components are independent. If we choose a point at the intersection of the wing and fuselage, then the wing is independent of the aft fuselage-tail combination, but the tail and aft fuselage must be considered together. The motion of the rigid forward fuselage is determined by the motion of the reference point, and the forward fuselage will not enter into any flexible considerations but only into the free-free boundary conditions. From the foregoing, it is seen that the various matrices will appear in partitioned form. If we denote the wing and aft fuselage-tail system by the subscripts 1 and 2, respectively, then the flexibility matrix appears as

$$[a] = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \quad (6)$$

the mass matrix as

$$[M] = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \quad (7)$$

the weighting matrix as

$$[W] = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \quad (8)$$

the AICs as

$$[C_h] = \left[\begin{array}{c|c} C_{h1} & 0 \\ \hline 0 & C_{h2} \end{array} \right] \quad (9)$$

and the rigid-body modal matrix as

$$[h_R] = \left[\begin{array}{c} h_{R1} \\ \hline h_{R2} \end{array} \right] \quad (10)$$

Two requirements should be emphasized with regard to the AICs. The first concerns the proper inclusion of the reference geometry associated with the nondimensional AICs. The dimensional form of Eq. (9) may be written

$$b_r^2 s [C_h] = \left[\begin{array}{c|c} b_1^2 s_1 C_{h1} & 0 \\ \hline 0 & b_2^2 s_2 C_{h2} \end{array} \right] \quad (11)$$

where b_r and s are the reference semichord and span of the composite system, b_1 and s_1 are the reference geometry for the first component, and b_2 and s_2 are the reference geometry for the second component. The second requirement is that the AICs for each component must be determined for the same "dimensional" reduced velocity V/ω . If the reference reduced velocity is

$$1/k_r = V/b_r \omega \quad (12)$$

then the reduced velocity for the first component must be

$$1/k_1 = (1/k_r)(b_r/b_1) \quad (13)$$

and, for the second component,

$$1/k_2 = (1/k_r)(b_r/b_2) \quad (14)$$

Both of these requirements can be met in formulating the composite AICs by choosing the same reference geometry in determining the AICs for each component.

The rigid-body modal matrix provides the basis for a general statement of the boundary conditions for the free-free flutter of the composite system. The boundary conditions for harmonic motion may be written as

$$[h_R]^T [\bar{M}] \{h_1\} + [\Delta \bar{m}] \{a_R\} = \{0\} \quad (15)$$

where $[\Delta \bar{m}]$ is an incremental generalized mass matrix, including aerodynamic effects, of any rigid component of the system attached to the reference point (e. g., the forward fuselage that was assumed to be rigid in the foregoing example),* and is not considered in the formulation of the flexible component mass and aerodynamic matrices. The form of the matrix $[\Delta \bar{m}]$ may be illustrated by the previous example with the rigid forward fuselage again in symmetrical motion. We may write

$$[\Delta \bar{m}] = [\Delta m] + [\Delta Q] \quad (16)$$

* N. B.: It is assumed that no dynamic coupling exists between the rigid and flexible components. A suitable distinction can always be made between the rigid and flexible components such that this requirement can be met.

NOTICE TO USERS

Portions of this document have been judged by the Clearinghouse to be of poor reproduction quality and not fully legible. However, in an effort to make as much information as possible available to the public, the Clearinghouse sells this document with the understanding that if the user is not satisfied, the document may be returned for refund.

If you return this document, please include this notice together with the IBM order card (label) to :

Clearinghouse
Attn: 152.12
Springfield, Va. 22151

where the generalized rigid component mass matrix of the forward fuselage is

$$[\Delta m] = \begin{bmatrix} M_o & S_o \\ S_o & I_{yo} \end{bmatrix} \quad (17)$$

in which M_o , S_o , and I_{yo} are the mass, static unbalance about the reference point, and pitching moment of inertia about the reference point, respectively, of the forward fuselage, and the generalized aerodynamic forces on the forward fuselage (if not negligible) are found from

$$[\Delta Q] = 32.174 \rho b_o^2 s_o [h_{Ro}]^T [C_{ho}] [h_{Ro}] \quad (18)$$

where $[h_{Ro}]$ is the rigid-body modal matrix, $[C_{ho}]$ is the set of AICs, and b_o and s_o are the reference geometry for the forward fuselage. Again the AICs must be found for the reduced velocity of the composite system.

We are now in a position to eliminate the rigid-body degrees of freedom and to formulate the eigenvalue problem for the flutter of the flexible free-free system. Substituting Eqs. (4b) and (5) into Eq. (1), and adding the structural damping factor $1/(1 + ig)$ to the flexibility matrix to provide the artificial structural damping necessary to sustain the assumed harmonic motion of the flutter system, we obtain

$$\{h_1\} - [h_R] \{a_R\} = (4\pi^2 K_f^2 / 386(1 + ig)) [a] [\overline{M}] \{h_1\} \quad (19)$$

$$\text{or} \quad \lambda(\{h_1\} - [h_R] \{a_R\}) = [a] [\overline{M}] \{h_1\} \quad (20a)$$

$$= [U] \{h_1\} \quad (20b)$$

where λ denotes the eigenvalue

$$\lambda = \lambda_R + i\lambda_I \quad (21a)$$

$$= 386(1 + ig)/4\pi^2 K_f^2 \quad (21b)$$

Premultiplying Eq. (20b) by $[h_R]^T [\bar{M}]$, and multiplying Eq. (15) by λ and subtracting, permits solution for the amplitudes of the rigid-body motion

$$\lambda \{a_R\} = - [\bar{m}]^{-1} [h_R]^T [\bar{M}] [U] \{h_1\} \quad (22)$$

where

$$[\bar{m}] = [h_R]^T [\bar{M}] [h_R] + [\Delta \bar{m}] \quad (23)$$

Finally, substituting Eq. (22) into Eq. (20b) yields the generalized matrix equation for free-free flutter

$$\lambda \{h_1\} = ([I] - [h_R] [\bar{m}]^{-1} [h_R]^T [\bar{M}]) [U] \{h_1\} \quad (24)$$

The solution of Eq. (24) for the complex eigenvalues leads to the free-free frequency and the required structural damping. From Eq. (21), we obtain the frequency

$$f = (1/2\pi) \sqrt{386/K\lambda_R} \quad (25)$$

and the required structural damping

$$g = \lambda_I/\lambda_R \quad (26)$$

Since the formulation of the AICs requires the assumption of a reduced velocity $1/k_r$, the velocity follows from that and the frequency obtained in Eq. (25)

$$U = 0.5921 (2\pi f b_r)(1/k_r) \quad (27)$$

Equation (24) is seen to be completely general, being applicable from the cantilever case (in which we let $[h_R]$ vanish) to the case of six rigid body degrees of freedom, and for a vibration analysis for which the aerodynamic terms and required structural damping are deleted. We observe that Eq. (24) is a matrix formulation of the algebraic procedures for free-free vibration analysis described in Ref. 4 (Par. 11.2).

From a series of solutions of Eq. (24) for various reduced velocities, the conventional required damping versus velocity stability curve can be constructed for a specific altitude, and the flutter point is determined as the velocity for which the required damping and actual damping are equal. An alternative approach to the flutter analysis is based on a single representative reduced velocity and a series of solutions of Eq. (24), carried out for various densities. The density at which the required damping and actual damping are equal may be used to find a stiffness-altitude similarity parameter for flutter from which the flutter stability may be determined. However, at present, the validity of this latter approach requires further investigation, particularly the sensitivity of the results to the choice of representative reduced velocity.

The generalized mass corresponding to each free vibration mode is of interest in various modal analyses of flying qualities, stability and control characteristics, or transient response of flexible vehicles. If Eq. (24) is solved for the free vibration modes (by deleting the aerodynamic terms) then the n 'th generalized mass is given by

$$m^{(n)} = \{h_1^{(n)}\}^T [M] \{h_1^{(n)}\} + \{a_R^{(n)}\}^T [\Delta m] \{a_R^{(n)}\} \quad (28)$$

where $\{h_1^{(n)}\}$ is the n'th free vibration mode and the corresponding rigid component mode is found from Eq. (22)

$$\{a_R^{(n)}\} = - \frac{4\pi^2 K}{386} f_n^2 [M]^{-1} [h_1]^\top [M] [U] \{h_1^{(n)}\} \quad (29)$$

SECTION 4

REFERENCES

1. W. P. Rodden, E. F. Farkas and H. A. Malcom. "Flutter and Vibration Analysis by a Collocation Method: Analytical Development and Computation Procedure". Aerospace Corporation Report No. TDR-169(3230-11)TN-14, 31 July 1963.
2. W. P. Rodden, "Matrix Approach to Flutter Analysis" Institute of the Aerospace Sciences Fairchild Fund Paper No. FF-23, May 1958; based on North American Aviation, Inc., Report NA-56-1070, 1 May 1956.
3. W. P. Rodden, "On Vibration and Flutter Analysis with Free-Free Boundary Conditions". Journal of the Aerospace Sciences, 28 (1961) 65.
4. W. P. Rodden and J. D. Revell. "The Status of Unsteady Aerodynamic Influence Coefficients". Institute of Aerospace Sciences Fairchild Fund Paper No. FF-33, 23 January 1962; preprinted in Aerospace Corporation Report TDR-930(2230-09) TN-2, 22 November 1961.
5. R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. Aeroelasticity. Reading: Addison-Wesley Publishing Company, Inc., 1955.
6. R. H. Scanlan and Robert Rosenbaum. Introduction to the Study of Aircraft Vibration and Flutter. New York: The MacMillan Company, 1951.

SECTION 5

DESCRIPTION OF PROGRAM INPUT

UNITS

The dimensional data required for each component consist of the mass matrix in pounds, the flexibility matrix in inches per pound, and the reference semichord and semispan in feet. The aerodynamic influence coefficients are dimensionless. In the case of free-free analysis, the rigid-body mass matrix, e. g., in a symmetrical analysis if S_0 and I_{y0} are given in the foot-pound system, the x-coordinates which correspond to the rigid body pitching mode must be measured in feet, whereas if S_0 and I_{y0} are given in the inch-pound system, the x-coordinate must be measured in inches. The density is required in slugs per cubic foot.

CLASSES OF DATA AND PROBLEMS

Five classes of data must be provided: mass, aerodynamic influence coefficients (AICs) and their weighting matrices, structural influence coefficients, the rigid-body motion modal matrix, and the rigid component generalized mass characteristics. The cantilever case does not require either the rigid body modal matrix or the generalized masses. (For a vibration analysis, the aerodynamic input is not required.)

Several classifications of problems may be analyzed using the collocation flutter analysis program. They are cantilever flutter analysis — the structure is restrained from plunging motion. Free-free flutter analysis the structure is free to pitch, plunge, and roll. The free-free cases may have rigid components and flexible components. However, when flexible components are coupled together, the structural attachment between component must be statically determinate. When rigid body components are used, any number up to six rigid body modes may be used. Also vibration analyses may be performed when zero aerodynamic forces are used.

PROGRAM RESTRICTIONS AND OPTIONS

1. The maximum number of control points that can be used for all flexible components of any system is 49. A maximum of 49 control points may also be used for the rigid component for the purpose of deriving the generalized aerodynamic force.
2. The maximum number of flexible components is 20.
3. The maximum number of values used in the reference reduced velocity ($1/K_R$) series is 20.
4. The maximum number of values used in a density series is 20.
5. The program provides for varying the densities with each ($1/K_R$) or for using the same densities with all ($1/K_R$)'s.
6. The maximum number of output modes is 25.
7. The maximum number of rigid-body motion modes is 6.
8. It is possible to reserve a partition in the upper left-hand corner of the flexible components AIC matrices for control points whose aerodynamic forces may be neglected or found from an alternate theory to that used for the primary control points. This partition is termed the external stores region since external stores are an example of a source of additional control points requiring such special consideration. The maximum number of control points that can be reserved on each flexible component for external stores is 48.
9. A weighting matrix is an optional input for each flexible component. The order of this matrix must be identical with the order of the AIC matrices for the particular component.
10. Any number of complete sets (decks) of input data may be stacked and run in one machine pass.

DATA DECK SETUP

Loading Order

The data decks are assembled using cards punched from keypunch forms and/or card that are punched-card output from appropriate computer programs. The data items in each deck have the following order,

with the exception that some of the items may be omitted if indicators used in the control cards specify their absence.

1. Title card
2. Data deck general control card
3. K card (flexibility matrix scale factor)
4. Data card for change in matrix iteration tests
5. Control card(s) for external stores and weighting matrices
6. Reference semichord (b_r) and reference reduced velocities ($1/k_r$)'s
7. Reference semichord (b_{r_i}) and reference semispan (S_i) for rigid and flexible components (surfaces)
8. Density series (if same densities are used for all $(1/k_r)$'s)
9. Generalized mass matrix ($[\Delta m]$) for rigid components
10. Mass matrix ($[M]_i$) for each flexible surface
11. Rigid-body motion modal matrix ($[h_{R_0}]$) for rigid component
12. Rigid-body motion modal matrix ($[h_{R_i}]$) for each flexible surface
13. Flexibility matrix ($[a]_i$)
14. Weighting matrices ($[W]_i$)
15. Aerodynamic input repeated for each $(1/k_r)$
 - a. Density series cards (if densities vary with each $(1/k_r)$)
 - b. AIC matrix ($[C_{h_0}]$) for rigid component (if present)
 - c. AIC matrix ($[C_h]_i$) for each flexible surface

Input Data Description

1. The title card may contain any alphanumeric characters desired in Columns 2 through 72.

2. Data deck general control card (FORMAT 18I4):

Col	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36
Name	NSUR	NAERO	NRIGID	NFUS	NDENS	MODES	NDELM	NPUNCH	NCOM
Field	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

- NSUR:** Number of flexible components (surfaces), ≤ 20 .
NAERO: Number of reference reduced velocities, ≤ 20 ; NAERO = 0 is used for vibration analyses.
NRIGID: Number of rigid-body motion modes to be input (= number of columns in $([h_R])$); NRIGID = 0 for the cantilever case.
NFUS: NFUS must = 1 if AICs $([C_{ho}])_j$, $j = 1$, NAERO are input for the rigid components; NFUS = 0 if $[C_{ho}]$ is not input.
NDENS: If NDENS = 0 the densities are to vary with each $1/k_r$ and are input as part of Item 15; if NDENS > 0 this number of densities must be input as Item 8, and each density value will be used for all $(1/k_r)$'s.
MODES: Number of output modes, ≤ 25 .
NDELM: NDELM = 0 if no rigid component generalized mass matrix $([\Delta m])$ is input; NDELM = 1 if $[\Delta m]$ is input.
NPUNCH: This indicator is used to obtain a printout of the dynamic matrix $([U])$ and to obtain the frequencies and modes in punched-card format; NPUNCH = 0 if no printout of $[U]$ or punched output is desired; NPUNCH = 1 if only punched-card output is desired and NPUNCH = -1 will provide the printout of $[U]$ and the punched output. (The minus sign must be in Column 29 and the 1 (one) in Column 32.)
NCON: This indicator provides for changing five program test numbers used in the matrix iteration subroutine; NCON = 0 if no changes are desired and NCON $\neq$ 0 if any of the tests is to be changed.

3. K card (FORMAT 6E12.8)

Col	1-2
Name	K
Field	(1)

Field 1 contains K, the flexibility matrix normalizing constant; if the [a] matrix has not been normalized, enter K = 1.0. The flexibility matrix calculated by the program FLUENC has not been normalized.

4. Data card for changes in matrix iteration tests (FORMAT 3E12.8 and 2I4). Omit this card when NCON = 0. There are three test numbers and two control numbers that define the convergence criteria for the flutter eigenvalue solution. A suggested set of numbers are built into the program; these, however, may be changed by the program user. To alter any number, all five numbers must be reentered.

Col	1-12	13-24	25-36	37-40	41-44	
Name	EPSP	EPDP	AITKEN	NITRSP	NITRDP	
Field	(1)	(2)	(3)	(4)	(5)	

EPSP = 0.5×10^{-6} or input number; test for eigenvector convergence when the iteration procedure is approaching a single root.

EPDP = 0.5×10^{-7} or input number; test for convergence when the iteration procedure is approaching a pair of close roots.

AITKEN = 0.9 or input number; if this test is met, the program uses a procedure (known as Aitken's δ^2 method) to accelerate convergence.

NITRSP = 40; a maximum of 40 single precision arithmetic iterations will be performed for each eigenvalue if its eigenvectors have not converged in a lesser number.

NITRDP = 100; if the eigenvectors for any one eigenvalue have not converged in NITRSP single precision iterations, the program will then use up to a maximum of 100 double arithmetic iterations.

5. Control card(s) for external stores and weighting matrices (FORMAT 18I4); omit this data when NAERO = 0.

Col	1-4	5-8	9-12	13-16	
Name	ISXT ₁	ISW ₁	ISXT ₂	ISW ₂	
Field	1	2	3	4	

Continue on successive cards to ISW_i = ISW_{NSUR}.

ISXT_i = Number of control points reserved for the external stores on surface i.

ISW_i = 0, no weighting matrix is to be input for surface i.
= 1, weighting matrix is to be input for surface i.

Continue on next card, until i = NSUR.

6. Reference semichord, b_r and reduced velocity ($1/k_r$) series (FORMAT 6E12.8): These reference parameters are used in computing the flutter velocities.

Col	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_r	$(1/k_r)_1$	$(1/k_r)_2$	$(1/k_r)_3$	$(1/k_r)_4$	$(1/k_r)_5$
Field	(1)	(2)	(3)	(4)	(5)	(6)

Continue $(1/k_r)_i$'s on next card(s); $i \leq 20$. The b_r (feet) may be any value; but it is noted that the $(1/k_r)_i$ predetermines the $(1/k_i)$ used when computing the AIC matrices for each surface. The $1/k_i$ for surface i is found from the relationship $(1/k_i) = (1/k_r)(b_r/b_{ri})$ where b_{ri} is the reference semichord for surface i.

7. A reference semichord (b_{r1}) and semispan (s_1) must be input for the rigid component if NFUS = 1, and for each flexible surface if NAERO > 0.

When NFUS = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_{r0}	s_{r0}	b_{r1}	s_{r1}	b_{r2}	s_{r2}
Field	(1)	(2)	(3)	(4)	(5)	(6)

Where b_{r0} and s_{r0} are the reference semichord and semispan for the rigid component. Continue on next card(s) until $b_{ri} = b_{rNSUR}$ and $s_{ri} = s_{rNSUR}$.

When NFUS = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_{r1}	s_{r1}	b_{r2}	s_{r2}	b_{r3}	s_{r3}
Field	(1)	(2)	(3)	(4)	(5)	(6)

Continue on next card(s) until $b_{ri} = b_{rNSUR}$ and $s_{ri} = s_{rNSUR}$.

8. Density series (FORMAT 6E12. 5): Omit this input if NDENS=0 in Item 2. If NDENS>0 begin in field 1 of this card and NDENS densities, ≤ 20 . Continue on successive card(s).

Column	1-12	13-24	25-36	37-48
Name	ρ_1	ρ_2	ρ_3	ρ_4
Field	1	2	3	4

9. Rigid component generalized mass matrix $[\Delta m]$. The $[\Delta m]$ matrix for the rigid component(s) of the system must be compatible with the flexible surfaces product matrix given by $[h_R]^T [M] [h_R]$, i. e., each element in $[\Delta m]$ is based upon the same rigid-body motion generalized coordinate as the respective element in the product matrix. Input by column beginning each column on a new card. Omit this data when NDELM = 0.

Col	1-12	13-24	25-36		
Name	$\Delta m_{1,1}$	$\Delta m_{2,1}$	$\Delta m_{3,1}$	$\Delta m_{(NRIGID-1),1}$	$\Delta m_{NRIGID,1}$
Field	(1)	(2)	(3)	(NRIGID-1)	(NRIGID)

Continue on successive card(s) until

10. Mass matrix $[M]$: The mass matrix is partitioned as shown on page A-10, only the nonzero partitions are input; i. e., a separate mass matrix $([M_i])$ is input for each surface. The sequence for considering the surfaces is the same as that used in Item 5 and 7 if $NAERO > 0$. Repeat the following input for each surface from $i = 1$ to $NSUR$.

Size Control Cards				
Column	1-4	5-8	9-12	
Name	$NSIZE_i$			
Field	(1)	(2)	(3)	

$NSIZE_i$ = the order of $[M]_i$

Often many of the elements in a mass matrix are zero; the following format has been provided so that most of the zero elements will not need to be entered into the program as data.

Control Card(s) for Omitting Zeros (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	17-20	
Name	LOW_1	$LHIGH_1$	LOW_2	$LHIGH_2$	LOW_3	$LHIGH_i$
Field	(1)	(2)	(3)	(4)	(5)	(2i)

LOW_i = The row number in which the first nonzero element appears in Column i.

$LHIGH_i$ = The row number in which the last nonzero element appears in Column i.

If only one nonzero element appears in Column i (i. e., a diagonal mass matrix) the row number in which it appears must be used for both LOW and LHIGH.

Mass Matrix Elements (FORMAT 6E12.8)

Col	1-12	13-24	25-36			
Name	$\Delta m_{1, LOW}$	$\Delta m_{1, LOW+1}$	$\Delta m_{1, LOW+2}$		$\Delta m_{1, LHIGH-1}$	$\Delta m_{1, LHIGH}$
Field	(1)	(2)	(3)			

The elements are entered by column; each column begins on a new card. Any zero elements in rows between LOW and LHIGH must be entered or their respective fields left blank. If external stores are present (ISXT>0) all store control points must be entered before the surface control points; i. e., the elements representing the external stores mass must occupy the upper left-hand corner of the mass matrix.

11. Rigid component inodal matrix, $[h_{R_0}]$ (see page.7). Omit this input when NFUS = 0. The number of rows in $[h_{R_0}]$ must be the same as the number of control points considered when computing the $[C_{h_c}]$ matrices; the number of columns must agree with NRIGID.

Size Control Card(s) (FORMAT 18I4)

Column	1-4	5-8	
Name	NROWS		
Field	(1)	(2)	

NROWS = The number of rows in $[h_{R_0}]$

Matrix $[h_{R_0}]$ Elements (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	$h_{R_{c1,1}}$	$h_{R_{o2,1}}$	$h_{R_{o3,1}}$			$h_{R_{oNROWS,1}}$
Field	(1)	(2)	(3)	(4)	(5)	(6)

The elements are entered by column, with each column beginning on a new card.

12. Rigid-body modal matrix, $[h_R]$ (see page 7). Omit this input if NRIGID = 0. $[h_R]$ is to be input by partitions $[h_R]_i$, each partition is of order (NSIZE x NRIGID) for each surface. The following data is to be repeated for $i = 1, NSUR$.

Size Control Card (FORMAT 18I4)

Column	1-4	5-8	
Name	NSIZE		
Field	(1)	(2)	

NSIZE = Number of control points on each surface

Matrix $[h_R]_i$ Elements (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	$h_{R1,1}$	$h_{R2,1}$	$h_{R3,1}$			$h_{R_{NSIZE,1}}$
Field	(1)	(2)	(3)	(4)	(5)	(6)

The elements are entered by column, with each column beginning on a new card.

13. Flexibility matrices, $[a]$ (see page 6). The flexibility matrix is partitioned, only the nonzero partitions $[a]_i$ corresponding to the

flexible surfaces are entered. The matrix may be formed by any of the well-known procedures using elementary beam theory, force or displacement methods. The program FLUENC will generate this matrix using the displacement method. The punched output from FLUENC may be used as direct input into this program. The following data is repeated for $i = 1, \text{NSUR}$.

Control Card (FORMAT 18I4):

Column	1-4	5-8	9-12	13-16	
Name	m_i	(BLANK)	IFORM	IROW	
Field	(1)	(2)	(3)	(4)	

m_i = The number of rows in $[a]_i$

IFORM = 0 if the elements are to be input using column binary format.

= 1 if the elements are to be input using FORTRAN (FORMAT 6E12.8) or FLUENC output is to be used directly.

IROW = 0 if the matrix elements are to be entered by column.

= 1 if the matrix elements are to be entered by row.

Matrix $[a]_i$ elements (use format specified above):

For IFORM = 1 and IROW = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

For IFORM = 1 and IROW = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	$a_{1,1}$	$a_{2,1}$	$a_{3,1}$	$a_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If IFORM = 0, then IROW must = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + m_1)$; Column 3 starts in location $(1 + 2m_1)$; etc. A TRA* card must end each $[a_i]$ deck. (The column binary format should be used only if the data are available as punched-card output from appropriate computer programs.) The only advantage of the C-B format is the minimum card storage space required.

14. Weighting matrix, $[W]$ (see page 6). The weighting matrix is partitioned, only the nonzero partitions $[W]_i$ corresponding to the flexible surface are entered. No provisions have been made for entering a $[W_o]_i$ matrix for the rigid component; any adjustment to $[C_{h_o}]$ must be made before it is input as data. If $ISW_i = 0$ omit this data. Repeat the following data for $i = 1, NSUR$.

For $(ISW)_i = 0$ and $(ISXT)_i > 0$

Control card for external stores elements (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	$NXST_i$	(BLANK)	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

*The TRA card has a 7 and 9 punch in Column 1, Column 2 through 72 are blank and Column 73 through 80 will contain the characters used for identification and sequencing in the punched card output deck.

$NXST_i = 0$ if no $[W]_i$ matrix is input for the external stores area (the program will use a unit matrix, I)
 $= n$ the number of control points reserved for stores.
 $NFORM = 1$ if the $[W]_i$ matrix elements will be input using FORTRAN (FORMAT 6E12.8)
 $= 0$ if the elements are to be input using column binary format
 $NROW = 0$ if the $[W]_i$ matrix elements are to be input by column
 $= 1$ if the matrix elements are to be input by row

External stores elements $[W]_i$. Format given on control card above.

For $NFORM = 1$ $NROW = 1$

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

For $NFORM = 1$ $NROW = 0$

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If NFORM = 0, then must NROW = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + IXST_1)$; Column 3 starts in location $(1 + 2IXST_1)$; etc. A TRA card must end each $[a_i]$ deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

Control card for flexible surface weighting matrix $[W]_i$

(FORMAT 18I4) The $[W]_i$ matrix is often sparse, sometimes diagonal and may be of large order, ≤ 49 ; for this reason we provide for partitioning of the matrix and entering only the nonzero partitions.

Column	1-4	5-8	9-12	13-16	
Name	NSIZE _i	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = The number of control points used on surface i .

Do not include control points for external stores.

NPART = The number of partitions in the $[W]_i$ surface matrix

NFORM = 1 if the $[W]_i$ will be input using FORTRAN (FORMAT 6E12.8)

= 0 if the elements are to be input using column binary format

NROW = 0 if the $[W]_i$ matrix elements are to be input by column

= 1 if the matrix elements are to be input by row

Repeat the following two data item for each partition $j = 1, \text{NPART}$.

Control card for partition $[W_i]_j$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16	17-21	
Name	N_j					
Field	(1)	(2)	(3)	(4)	(5)	

N_j = The order of partition j of $[W_i]$

Elements in partition $[W_i]_j$. Format given on the control card for flexible surface weighting matrix.

NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

NFORM = 1 NROW = 0

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,2}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If $NFORM = 0$, then must $NROW = 0$: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + IXST_i)$; Column 3 starts in location $(1 + 2IXST_i)$ etc. A TRA card must end each $[a_i]$ deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

For $(ISW)_i = 0$ and $(ISXT)_i = 0$

Control card for flexible surface weighting matrix $[W]_i$ (FORMAT 1814). The $[W]_i$ matrix is often sparse, sometimes diagonal and may be of large order, ≈ 49 ; for this reason we provide for partitioning of the matrix and entering only the nonzero partitions. Repeat the following data for $i = 1, NSUR$.

Column	1-4	5-8	9-12	13-16	17-20	
Name	NSIZE _i	NPART	NFORM	NROW		
Field	(1)	(2)	(3)	(4)	(5)	

NSIZE = The number of control points used on surface i

NPART = The number of partitions in the $[W]_i$ surface matrix

NFORM = 1 if the $[W]_i$ will be input using FORTRAN (FORMAT 6E12.8)

= 0 if the elements are to be input using column binary format

NROW = 0 if the elements are to be input by column

= 1 if the elements are to be input by row

Repeat the following two data items for each partition $j = 1, NPART$.

Control card for partition $[W_i]_j$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16	
Name	N_j				
Field	(1)	(2)	(3)	(4)	

N_j = The order of partition j of $[W_i]$

Elements in partition $[W_i]_j$ format given on the control card for flexible surface weighting matrix.

NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

NFORM = 1 NROW = 0

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If NFORM = 0, then must NROW = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + m_1)$; Column 3 starts in location $(1 + 2m_1)$; etc. A TRA card must end each $[a_i]$ deck. The column binary format should

used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

15. Aerodynamic data: The aerodynamic input consists of NAERO sets of AIC's. Each set of AIC's consists of the AIC's for each surface which have the same reference $1/k_r$ (see Item 6). If NDENS = 0 a density series will precede each set of AIC's. Input the density series (if Item 8 was omitted) and the AIC matrices for each surface with the following input order. Repeat the order for each $(1/k_r)_j, j = 1, \text{NAERO}$.

Control card for density series (FORMAT 18I4). Omit this input if NDENS>0.

Column	1-4	5-8	9-12	13-16	
Name	NRHO				
Field	(1)	(2)	(3)	(4)	

NRHO = The number of densities to be entered for
 $(1/k_r)_j \leq 20$

Density Series (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	ρ_1	ρ_2	ρ_3	ρ_4	
Field	(1)	(2)	(3)	(4)	

Rigid component AIC matrix $[C_{ho}]_j$. Omit this input if NFUS = 0. The $[C_{ho}]_j$ matrix may be sparse; thus, provision has been made to partition the matrix and enter only the nonzero partitions.

Reference $1/k$ for $[C_{ho}]_j$ (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	$1/k$				
Field	(1)	(2)	(3)	(4)	

$1/k$ = The reduced velocity used in computing the rigid component $[C_{ho}]_j$. The AIC's in $[C_{ho}]$ must be computed for a $1/k$ which properly relates them to the j^{th} $1/k_r$.

Control card for $[C_{ho}]_j$ matrix (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = Number of control points on the rigid component

NPART = Number of nonzero partitions in $[C_{ho}]_j$
 = 1 for an unpartitioned matrix

NFORM = 1 if the $[C_{ho}]_j$ matrix is to be input using FORTRAN
 (FORMAT 6E12.8)

= 0 if the $[C_{ho}]_j$ matrix is to be input using column
 binary format

NROW = 1 if the $[C_{ho}]_j$ matrix is to be input by row

= 0 if the $[C_{ho}]_j$ matrix is to be input by column

Repeat the following data for each partition, $K = 1, \text{NPART}$.

Partition Size Card (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	N				
Field	(1)	(2)	(3)	(4)	

N = The order of partition k

Elements in partition K of $[C_{hO}]_j$. Format is given on the control card for $[C_{hO}]_j$ matrix. All the elements in the AIC matrices are complex numbers, but the complexity is considered in the program. Thus, each partition may be input as though it is a real matrix of size $N \times 2N$. The real elements form the odd number columns, and the imaginary elements in the even numbered columns.

For NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$a(\text{Re})_{1,1}$	$a(\text{I})_{1,1}$	$a(\text{Re})_{1,2}$	$a(\text{I})_{1,2}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

For NFORM = 1 and NROW = 0

Column	1-12	13-24	25-36	37-48	49-60	
Name	$a(\text{Re})_{1,1}$	$a(\text{I})_{1,1}$	$a(\text{Re})_{2,1}$	$a(\text{I})_{2,1}$		
Field	(1)	(2)	(3)	(4)	(5)	

Each column starts on a new card.

For NFORM = 0, NROW must = 0. Use column binary format. Column 1 starts in card Origin 1, Column 2 in location (1 + 2N), Column 3 in location (1 + 4N), etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

Flexible component AIC matrix $[C_h]_i$. The AIC matrices are often sparse; thus, a provision is made partitioning the matrix and entering only the nonzero partitions. The following data is repeated for $i = 1, \text{NSUR}$.

For $\text{ISXT}_i > 0$.

Control card for external stores partition of the surface i
AIC matrix $[C_h]_{ij}$ (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	NXST	(BLANK)	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NXST_i = Number of control points reserved for external stores
 $\text{NFORM} = 1$ if the matrix elements are to be input using
 FORTRAN (FORMAT 6E12.8)
 = 0 if the matrix elements are to be entered using
 column binary format
 $\text{NROW} = 1$ if the matrix elements are to be entered by row
 = 0 if the matrix elements are to be entered by column

Elements for external stores partition of AIC matrix $[C_h]_{ij}$

For NFORM = 1 and NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$a(\text{Re})_{1,1}$	$a(I)_{1,1}$	$a(\text{Re})_{1,2}$	$a(I)_{1,2}$	
Field	(1)	(2)	(3)	(4)	

Each row begins on a new card.

For NFORM = 1 and NROW = 0

Column	1-12	13-24	25-36	37-48	
Name	a(Re)i, 1	a(I)1, 1	a(Re)2, 1	a(Re)2, 1	
Field	(1)	(2)	(3)	(4)	

If NFORM = 0 then NROW must = 0 use column binary format. Column 1 starts in card origin 1, Column 2 in Location $(1+2NXT_i)$, Column 3 in Location $(1+4NXST_i)$, etc. A TRA Card must end each deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space requirements.

Reference $1/k_i$ card for control point AIC matrix $[C_h]_{ij}$

FORMAT (6E12.8)

Column	1-12	13-24	25-36	
Name	1/k			
Field	(1)	(2)	(3)	

$1/k_i$ = The reduced velocity used in computing the flexible component $[C_h]_{ij}$. The AIC's must be computed for a $1/k_i$ which properly relates them to the j th $1/k_r$.

Control card for control point AIC matrix $[C_h]_{ij}$ (FORMAT (1814)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

- NSIZE = Order of control point AIC matrix $[C_h]_{ij}$
 NPART = Number of partitions in $[C_h]_{ij}$
 NFORM = 1 matrix elements are to be input using FORTRAN
 (FORMAT 6E12.3)
 = 0 matrix elements are to be input using column binary
 format
 NROW = 1 elements input by row.
 = 0 elements input by column.

Repeat the following data for $j = 2, \text{NPART}$. $j = 1$ corresponds to the external stores partition

Control card for partition size of partition j FORMAT (18J4)

Column	1-4	5-8	9-12	
Name	N			
Field	(1)	(2)	(3)	

N = Order of partition j

Elements of control point AIC matrix $[C_H]_{i,j}$ partition j

For NFORM = 1 and NROW = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(Re)1,1	a(I)1,1	a(Re)1,2	a(I)1,2	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card

For NFORM = 1 and NROW = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(Re)1,1	a(I)1,1	a(Re)2,1	a(I)2,1	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card

For NFORM = 0 then NROW must = 0 use column binary format. Column 1 starts in card origin 1, Column 2 in Location (1 + 2N), Column 3 in Location (1 + 4N), etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space requirements

For $ISXT_i = 0$

Reference $1/k_i$ card for control point AIC matrix $[C_h]_{ij}$

FORMAT (6E12.8)

Column	1-12	13-24	25-36	
Name	$1/k_i$			
Field	(1)	(2)	(3)	

$1/k_i$ = The reduced velocity used in computing the flexible component $[C_h]_{ij}$. The AIC's must be computed for a $1/k_i$ which properly relates them to the j th $1/k_r$.

Control card for control point AIC matrix $[C_h]_{ij}$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = Order of control point AIC matrix $[C_h]_{ij}$
 NPART = Number of partitions in $[C_h]_{ij}$
 NFORM = 1 matrix elements are to be input using FORTRAN (FORMAT 6E12.8)
 = 0 matrix elements are to be input using column binary format.
 NROW = 1 elements input by row
 = 0 elements input by column

Repeat the following data for $j = 1, \text{NPART}$.

Control card for partition size of partition j FORMAT (18I4)

Column	1-4	5-8	9-12	
Name	N			
Field	(1)	(2)	(3)	

N = Order of partition

Elements of control point AIC matrix $[C_h]_{i,j}$ partition j

For $\text{NFORM} = 1$ and $\text{NROW} = 1$ (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(Re)1,1	a(I)1,1	a(Re)1,2	a(I)1,2	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card

For $\text{NFORM} = 1$ and $\text{NROW} = 0$ (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(Re)1,1	a(I)1,1	a(Re)2,1	a(I)2,1	
Field	(1)	(2)	(3)	(4)	

For $\text{NFORM} = 0$ then NROW must = 0 use column binary format. Column 1 start in card origin 1, Column 2 in Location $(1 + 2N)$, Column 3 in Location $(1 + 4N)$, etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage requirements.

SECTION 6

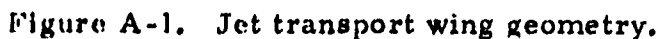
DESCRIPTION OF PROGRAM OUTPUT

A. Printed Output

1. All input data.
2. The dynamic matrix or flutter determinant if NPUNCH = -1 is used in the general control card.
3. For the vibration analysis or for each $1/k_r$ in a flutter analysis
 - a. The eigenvalue for each output mode followed by the number of single- and double-precision iterations and the number of Aitken accelerations.
 - b. The normalized eigenvectors (modes) followed by the check eigenvalues and eigenvectors.
 - c. The frequencies (omegas) in cycles per second followed (in a flutter analysis) by the structural damping coefficient and the velocity (knots) associated with each frequency.
 - d. If NPUNCH = ± 1 , the sequencing numbers used for identifying the punched-card output (frequencies and modes); this will conclude the printout for each $1/k_r$ used in a flutter analysis.
4. In a vibration analysis, the generalized mass corresponding to each output (free vibration) mode will follow the above printed output [see Eq. (28), Section I].
5. A number of different statements may be printed which indicate machine or program detected errors in input data (wrong format or incompatibility in the number of rows input for a matrix and the number designated).
6. If the program or the machine fails in the iteration portion of the program, a note will be printed stating the type or cause of failure.

7. If convergence is not obtained in the allowable number of iterations, a note will be printed and the program will continue. (In this case the eigenvalues and eigenvectors should be compared with the check eigenvalues and eigenvectors to determine if the convergence is sufficiently accurate.)
8. The printed output for the example problem is shown on the following pages.

EXAMPLE PROBLEMS

$$[\Delta m] = \begin{bmatrix} M_o & S_o \\ S_o & I_{yo} \end{bmatrix} = \begin{bmatrix} 17,400 & 1,370,250 \\ 1,370,250 & 4,457,907,200 \end{bmatrix}$$


In the antisymmetric case, the rigid component generalized mass matrix would be $[\Delta m] = [I_{x_0}]$, and in the composite longitudinal-lateral case, the generalized mass matrix would be

$$[\Delta m] = \begin{bmatrix} M_o & S_o & 0 \\ S_o & I_{y_0} & 0 \\ 0 & 0 & I_{x_0} \end{bmatrix}$$

The symmetrical case requires the two rigid-body degrees of freedom of plunging and pitching. The rigid-body modal matrix, therefore, consists of two columns: a unit column, and a column of the x-coordinate of each control point. The rigid-body modal matrix for the wing is

$$[h_R] = [1 \ x] = \begin{bmatrix} 1 & 20.25 \\ 1 & -81.00 \\ 1 & 17.85 \\ 1 & -71.40 \\ 1 & 15.80 \\ 1 & -63.20 \\ 1 & 13.30 \\ 1 & -53.20 \\ 1 & 11.05 \\ 1 & -44.20 \end{bmatrix}$$

The rigid-body modal matrix for the fuselage is

$$[h_{R_0}] = [1 \ x] = \begin{bmatrix} 1 & -373.30 \\ 1 & -248.30 \\ 1 & -123.30 \\ 1 & +1.70 \\ 1 & +126.70 \end{bmatrix}$$

Note that the above matrix is used in computing the incremental generalized mass which results from the aerodynamic loads on one-half of the fuselage and, therefore, the x-coordinates must be given in the

proper order for the control points used in computing the fuselage AICs. [In this problem the fuselage AICs are hypothetical, but the x-coordinates are given in the order necessary for the slender-body theory AICs.]

The reference geometry for the wing is $b_{rw} = 5.468$ ft and $s_w = 37.917$ ft. The reference geometry for the fuselage is $b_{ro} = 5.468$ ft and $s_o = 18.9585$ ft. [We assume that the wing reference geometry was used to nondimensionalize the fuselage AIC matrix, and, because we require only one-half of the fuselage aerodynamic force, it can be obtained by setting $s_o = (1/2)s_w$.] Both example problems are carried out for the single reduced velocity $1/k_r = 16.67$ with the reference semichord for the system $b_r = 5.468$ ft and with sea-level density $\rho = 0.002378$ slugs/ft³.

The output modes, the flutter frequencies and velocities, and the required structural damping can be seen in the example problem printed output (pages A-55 through A-65).

The input keypunch sheets are given, followed by the computer output.

DO-1-A34 23941

FO-2-AMK 20415

017000 REV. 2-64

7094 FORTRAN CODING SHEET

NAME	PHONE	DATE																																																																																																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8																																																																																												

NAME	PHONE	DATE	LOC NO.	IDENTIF. CATEGORY																																																																											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27																																																					

HUGHES AIRCRAFT COMPANY
7094 FORTRAN CODING SHEET

PAGE

NAME _____ PHONE _____ DATE _____

LOG NO. _____ PAGE _____

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
Control Card for [Ch] Matrix Input (FORMAT 1314)																																																																																																			
[Ch] Matrix Elements (Column Binary Format)																																																																																																			
The above control card indicates a 5 x 5 (unpartitioned) matrix to be input using column binary format. The matrix can be seen in the printed output and will not be tabulated.																																																																																																			
(Other input format explained in Section II, Part B, item 15.)																																																																																																			
1/4 for Surface [Ch] Matrix																																																																																																			
Control Card for Surface [Ch] Matrix Input																																																																																																			
[Ch] Matrix by Partitions																																																																																																			
The partitions are identical to those used in the cantilever case.																																																																																																			
HM 2 0 0 0 5 2																																																																																																			
HM 2 0 0 0 6 4																																																																																																			
HM 2 0 0 0 6 5																																																																																																			

JFT TRANSPORT EXAMPLE PROBLEM - CANTILEVER CASE
 FLUTTER ANALYSIS BY A COLLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS

HMD51641

NSUR = 1 NAERO = 1 NRIGID = 0 NFUS = 0 NDENS = 1 MDENS OUT = 4 NDELM = 0 NPUNCH = 0

B (REF) = 0.5488000E 01 K = 0.10000000E-06

SURFACE 8 S EXTERNAL STORES SIZE

1 0.5488000E 01 0.3791700E 02 0

MASS MATRIX

SURFACE 1		CONTROL POINTS		COLUMN		COLUMN		COLUMN		COLUMN	
COLUMN	1	COLUMN	2	COLUMN	3	COLUMN	4	COLUMN	5	COLUMN	6

1	0.5383500E 04	-0.1340000E 03	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	-0.1349000E 03	0.9252000E 03	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	0.	0.	0.2071200E 05	-0.1100500E 05	0.	0.	0.	0.	0.	0.	0.
4	0.	0.	-0.1005000E 05	0.1147800E 05	0.	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.3113900E 04	0.1397000E 03	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.1397000E 03	0.8866000E 03	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

COLUMN

COLUMN 7		COLUMN 8		COLUMN 9		COLUMN 10		COLUMN	
COLUMN	7	COLUMN	8	COLUMN	9	COLUMN	10	COLUMN	
1	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.	0.	0.	0.
7	0.2638800E 04	-0.2100000E 02	0.	0.	0.	0.	0.	0.	0.
8	-0.2100000E 02	0.8033000E 03	0.	0.	0.	0.	0.	0.	0.
9	0.	0.	0.4875000E 03	0.7300000E 01	0.	0.	0.	0.	0.
10	0.	0.	0.7300000E 01	0.1770000E 03	0.	0.	0.	0.	0.

FLEXIBILITY MATRIX SURFACE 1, 10 CONTROL POINTS

COLUMN	1	COLUMN	2	COLUMN	3	COLUMN	4	COLUMN	5	COLUMN	6
1	0.017200E 02	1.136100E 02	0.177700E 03	0.02720000E 02	0.02720000E 02	0.1025100E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03
2	0.136100E 02	0.177700E 03	0.027200E 02	0.0249200E 03	0.0249200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03	0.1049200E 03
3	0.177700E 03	0.027200E 02	0.0249200E 03	0.1572600E 03	0.1572600E 03	0.4825500E 03	0.4825500E 03	0.4825500E 03	0.4825500E 03	0.4825500E 03	0.4825500E 03
4	0.027200E 02	0.0249200E 03	0.1572600E 03	0.0374900E 03	0.0374900E 03	0.3762600E 03	0.3762600E 03	0.3762600E 03	0.3762600E 03	0.3762600E 03	0.3762600E 03
5	0.1025100E 03	0.1049200E 03	0.4825500E 03	0.3762600E 03	0.3762600E 03	0.1275800E 04	0.1275800E 04	0.1275800E 04	0.1275800E 04	0.1275800E 04	0.1275800E 04
6	0.1049200E 03	0.1049200E 03	0.4825500E 03	0.3762600E 03	0.3762600E 03	0.11344300E 04	0.11344300E 04	0.11344300E 04	0.11344300E 04	0.11344300E 04	0.11344300E 04
7	0.2047800E 03	0.1563000E 03	0.7328000E 03	0.6433800E 03	0.6433800E 03	0.1935000E 04	0.1935000E 04	0.1935000E 04	0.1935000E 04	0.1935000E 04	0.1935000E 04
8	0.1563000E 03	0.2047800E 03	0.6433800E 03	0.1001210E 04	0.1001210E 04	0.1816900E 04	0.1816900E 04	0.1816900E 04	0.1816900E 04	0.1816900E 04	0.1816900E 04
9	0.2428800E 03	0.2282500E 03	0.6581000E 03	0.8837000E 03	0.8837000E 03	0.2528300E 04	0.2528300E 04	0.2528300E 04	0.2528300E 04	0.2528300E 04	0.2528300E 04
10	0.2282500E 03	0.2428800E 03	0.8837000E 03	0.1181600E 04	0.1181600E 04	0.2429400E 04	0.2429400E 04	0.2429400E 04	0.2429400E 04	0.2429400E 04	0.2429400E 04

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

COLUMN

WEIGHTING MATRIX SURFACE 1, NO WEIGHTING MATRIX

ALPHADYNAMIC MATRIX

1. K. R. = 0.160/9000E 02

SURFACE 1, 10 CONTROL POINTS

COLUMN 1 COLUMN 2 COLUMN 3 COLUMN 4 COLUMN 5 COLUMN 6

1	0.3009700E 03	-0.6000261E 02	-0.2057901E 03	0.3227809E 02	0.	0.
2	0.1911010E 00	0.1651010E 02	0.2573354E 00	-0.2621412E 02	0.	0.
3	0.	0.	0.	0.	0.2933490E 03	-0.2515277E 02
4	0.	0.	0.	0.	0.1421153E 00	0.1393478E 02
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

COLUMN 7 COLUMN 8 COLUMN 9 COLUMN 10 COLUMN 11 COLUMN 12

1	0.	0.	0.	0.	0.	0.
2	0.2972241E 03	0.3124042E 02	0.	0.	0.	0.
3	0.4634850E 00	-0.1391780E 02	0.	0.	0.	0.
4	0.	0.	0.3048199E 03	-0.5386801E 02	-0.3802750E 03	0.5107155E 02
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.1184775E 00	0.2208791E 02	0.3415432E 00	-0.1260791E 02
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

53

COLUMN 13 COLUMN 14 COLUMN 15 COLUMN 16 COLUMN 17 COLUMN 18

1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.3248151E 03	-0.5227812E 02	-0.14787 E 03	0.3254110E 02	0.	0.
8	0.8214751E 01	0.1107502E 02	0.2520470E 00	-0.1107950E 02	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.3032147E 03	-0.4382780E 02
					0.5323689E 01	0.8430202E 01

COLUMN 19 COLUMN 20

1	0.	0.
2	0.	0.
3	0.	0.
4	0.	0.
5	0.	0.
6	0.	0.
7	0.	0.
8	0.	0.
9	0.	0.
10	0.	0.

FLUTTER ANALYSIS BY A COLLOCATION METHOD. USING AERODYNAMIC INFLUENCE COEFFICIENTS

Adjusted Velocity = 4.16674000E 02

DENSITY = 11530 "237844E-12

WIGID BODY DEONES OF FREEDOM

MODE	FIREVALUE	ITERATIONS	S.P.	D.P.	ATKENS	S.P.	D.P.
1	B.28514254E M7	-0.20187439E M7	18	0	2	3	
2	0.12153621E M7	-0.2851745E M7	18	0	4	0	
3	B.63295750E M7	B.11406000E M7	16	0	4	0	
4	B.28438801E M7	-0.16227640E M7	28	0	4	3	

FIGENVECTORS

COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1 0.4176569E-01	0.1993377E-02	0.5422154E-01	0.27073628E-01	0.11268764E 00	-0.06121170E-01
2 0.3361182E-01	0.5752720E-02	0.0312589E-01	-0.2712133E-02	0.95014571E-01	-0.47670138E-01
3 0.1407006E 00	0.01289036E-02	0.71708658E 00	0.5025255E-01	0.26814691E 00	-0.1977336E-01
4 0.1164470E 00	0.1320853E-01	0.17790778E 00	-0.14084838E-02	0.24426274E 00	-0.50250956E-01
5 0.3610304E 00	0.1749067E-01	0.42532389E 00	0.86469743E-01	0.50779018E 00	-0.44039706E 00
6 0.3453517E 00	0.2550204E-01	0.45030061E 00	0.33808263E-01	0.59176231E 00	-0.17162545E 00
7 0.6771108E 00	0.1475973E-01	0.71753930E 00	0.0829335E-01	0.02465572E 00	-0.01978349E-01
8 0.6506820E 00	0.26070921E-01	0.74431981E 00	0.29082516E-01	0.08081067E 00	-0.13226259E 00
9 0.3100000E 01	0.57600506E-00	0.06564340E 00	0.27935114E-01	0.93078000E 00	0.65759583E-01
10 0.9017352E 00	0.14730217E-01	0.10090000E 01	0.71289627E-10	0.10000000E 01	0.14332015E-00
COLUMN 7 COLUMN 8					
1 0.02558170E-01	-0.11804904E 00				
2 0.1442233E 00	-0.97307966E-01				
3 0.1297208E 00	-1.1712505E 00				
4 0.1619456E 01	-0.157627E 00				
5 0.4691980E 00	-0.1730003E 00				
6 0.4503630E 10	-1.1293572E 00				
7 0.1312656E-01	-0.606310E-01				
8 0.954000E-01	-0.310470E 00				
9 0.100000E 01					
10 0.000000E 01	-0.000000E-01				

CHECK EIGENVALUES AND EIGENVECTORS

COLUMN 1		COLUMN 2		COLUMN 3		COLUMN 4		COLUMN 5		COLUMN 6	
0.2851426E 08 -0.2518184E 05 0.1215363E 08 -0.2858472E 06 0.6329545E 07 0.1186494E 07		0.20439634E 07 -0.1262165E 07									
1	0.41765545E-01	0.39935360E-02	0.57421844E-01	0.27973633E-01	0.11260817E 00	-0.38121185E-01					
2	0.33611786E-01	0.67527245E-02	0.60324983E-01	-0.27113930E-02	0.95835350E-01	-0.47626920E-01					
3	0.14070066E 00	0.8128341E-02	0.17058602E 00	0.50252573E-01	0.26811789E 00	-0.69571621E-01					
4	0.12626694E 00	0.13288569E-01	0.17798768E 00	-0.13998269E-02	0.24426423E 00	-0.90548721E-01					
5	0.36103078E 00	0.17496553E-01	0.42532286E 00	0.88469934E-01	6.58779208E 00	-0.14839379E 00					
6	0.34535149E 00	0.25592861E-01	0.44039804E 00	0.33061094E-01	0.59176783E 00	-0.17164619E 00					
7	0.6771181E 00	0.14759617E-01	0.7153357E 00	0.88293261E-01	0.82485651E 00	0.81976559E-01					
8	0.65068186E 00	0.227876910E-01	0.74431851E 00	0.29903183E-01	0.86821284E 00	-0.13224914E 00					
9	0.10000000E 01	0.10363474E-08	0.96504330E 00	0.27934334E-01	0.93069882E 00	0.65757559E-01					
10	0.00173218E 00	0.14734288E-01	0.16000000E 01	0.19184524E-09	0.10000000E 01	0.48020418E-09					
COLUMN 7		COLUMN 8		COLUMN 9		COLUMN 10		COLUMN 11		COLUMN 12	
1	-0.85493178E-01	-0.11845119E 00									
2	-0.10421509E 00	-0.9738746E-01									
3	-0.15978453E 00	-0.172768E 00									
4	-0.18188963E 00	-0.176879E 00									
5	-0.46087509E 00	-0.1389738E 00									
6	-0.45833512E 00	-0.20537629E 00									
7	-0.18182238E-01	-0.66989833E-01									
8	-0.97884560E-02	-0.12320435E 00									
9	-0.10000000E 01	-0.81157384E-09									
10	-0.95823531E 00	-0.11462694E-01									

MODE FREQ (CPS) DAMPING VELOCITY (KNOTS)

1	0.1851734E 01	-0.88317019E 00	9.62794529E 03
2	0.28363566E 01	-0.23510478E-01	0.96183365E 03
3	0.39303258E 01	0.18747073E 00	0.13328069E 04
4	0.69164967E 01	-0.61758042E 00	0.23454407E 04

JET TRANSPORT EXAMPLE PROBLEM - SYMMETRIC CASE
FLUTTER ANALYSIS BY A COLLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS

PM051699

NSUR = 1 NAFRO = 1 NRIGID = 2 NFUS = 1 MDEMS = 1 MODES OUT = 4 NDELM = 1 NPUNCH = 0

R RIGID COMPONENT = 0.5468000E 01 S RIGID COMPONENT = 0.18958500E 02

H (REF) = 0.5468000E 01 K = 0.1000000E -06

SURFACE 1 0.5468000E 01 0.3791700E 02 0

RIGID COMPONENT MASS MATRIX

COLUMN	1	2	3
1	0.1740000E 05	0.1700000E 07	
2	0.1370220E 07	0.4570072E 10	

MASS MATRIX

COLUMN	1	2	3	4	5	6
--------	---	---	---	---	---	---

1	0.5383600E 04	-0.1490000E 03	0.	0.	0.	0.
2	-0.1349000E 03	0.9252000E 03	0.	0.	0.	0.
3	0.	0.	0.7073000E 05	-0.11005000E 05	0.	0.
4	0.	0.	-0.11005000E 05	0.11478000E 05	0.	0.
5	0.	0.	0.	0.	0.31139000E 04	0.1300000E 03
6	0.	0.	0.	0.	0.13970000E 03	0.000000E 03
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

COLUMN

1 2 3 4 5 6

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.

RIGID COMPONENT MODES.

COLUMN

1 2 3

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

0. 0. 0.

RIGID BODY MODAL MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN
1	0.1100000E 01	0.2025000E 02	
2	0.1100000E 01	0.1850000E 02	
3	0.1100000E 01	0.1785000E 02	
4	0.1100000E 01	0.2147000E 02	
5	0.1100000E 01	0.1780000E 02	
6	0.1100000E 01	0.1632000E 02	
7	0.1100000E 01	0.1336000E 02	
8	0.1100000E 01	0.5220000E 02	
9	0.1100000E 01	0.1105000E 02	
10	0.1100000E 01	0.4420000E 02	

FLEXIBILITY MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.8172000E 02	0.1336000E 02	0.1277000E 03	0.6272000E 02	0.1625100E 03	0.10492200E 03
2	0.1336100E 02	0.3086100E 03	0.6272000E 02	0.5229700E 03	0.10492200E 03	0.13352900E 03
3	0.1277900E 03	0.62721900E 02	0.2772000E 03	0.1572600E 03	0.4825500E 03	0.3762800E 03
4	0.6272000E 02	0.5129700E 03	0.1572600E 03	0.6374900E 03	0.3762800E 03	0.8013600E 03
5	0.1625100E 03	0.1497200E 03	0.4825500E 03	0.3762800E 03	0.1275800E 04	0.11344300E 04
6	0.10492200E 03	0.1336200E 03	0.3762800E 03	0.6433800E 03	0.11344300E 04	0.1699900E 04
7	0.2047800E 03	0.1631000E 03	0.3762800E 03	0.6433800E 03	0.1935000E 04	0.1816000E 04
8	0.5229700E 03	0.3502100E 03	0.6433800E 03	0.1001210E 04	0.1816000E 04	0.2292000E 04
9	0.2428500E 03	0.2025700E 03	0.9581000E 03	0.8637800E 03	0.2528300E 04	0.2429400E 04
10	0.2040300E 03	0.4528500E 03	0.8637800E 03	0.1181060E 04	0.2429400E 04	0.2824900E 04

COLUMN

10

COLUMN

7

COLUMN

8

COLUMN

9

COLUMN

10

COLUMN

11

COLUMN

12

WEIGHTING MATRIX
SURFACE 1, NO WEIGHTING MATRIX

RIGID COMPONENT AERO MATRIX 5 CONTROL POINTS

COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1 -0.1984316E 01	-0.1022417E 01	-0.5178207E 01	0.99733821E 01	0.17128096E 01	-0.24681993E 01
2 0.20347044E 01	-0.10574088E 01	-3.84710978E 01	0.67589525E 00	0.25888789E 01	0.32829872E 01
3 0.	0.	0.25878749E 01	-0.32622072E 01	-0.89710928E 01	-0.67589525E 00
4 0.	0.	0.	0.	0.18608618E 01	-0.23997268E 01
5 0.	0.	0.	0.	0.19459838E 00	0.73566824E 00

COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10
1 0.	0.	0.	0.
2 0.	0.	0.	0.
3 0.20347044E 01	0.30575088E 01	0.	0.175338263E 01
4 -0.5988403E 01	-0.41440609E 00	0.10700520E 01	0.815088698E 00
5 0.4545274E 00	-0.31680204E 01	-0.19732837E 01	0.

AERODYNAMIC MATRIX					
SURFACE 1		CONTROL POINTS			
COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6

1-K R = 0.1667000E 02

1	0.3089070E 03	-0.6090267E 02	-0.3057001E 03	0.3227820E 02	0.
2	0.1931181E 00	0.1623411E 02	0.5713354E 00	-0.1651411E 02	0.
3	0.	0.	0.	0.	0.2933498E 03
4	0.	0.	0.	0.	0.1421153E 00
5	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.

1	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.
3	-0.2728241E 03	0.3124032E 02	0.	0.	0.
4	0.4263405E 00	-0.1393179E 02	0.	0.	0.
5	0.	0.	0.3058399E 03	-0.5086803E 02	0.
6	0.	0.	0.1338477E 00	0.1268799E 02	0.
7	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.

1	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.
7	0.3248325E 03	-0.5227819E 02	-0.5276746E 03	0.3254116E 02	0.
8	0.4421623E-01	0.1197956E 02	0.2526407E 00	-0.1187956E 02	0.
9	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.

1	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.

1	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.
9	-0.1052102E 03	0.2851713E 02	0.	0.	0.
10	0.1507108E 00	-0.0000202E 01	0.	0.	0.

1	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.

OUTPUT DATA

FLUTTER ANALYSIS BY A COLLOCATION METHOD, USING AERODYNAMIC INFLUENCE COEFFICIENTS

DENSITY = 0.2378000E-02 REDUCED VELOCITY = 0.1667000E-02

RIGHT BODY DEGREES OF FREEDOM

MODE EIGENVALUE ITERATIONS S.P. U.P. ATKENS S.P. U.P.

MODE	EIGENVALUE	ITERATIONS	S.P.	U.P.	ATKENS S.P.	U.P.
1	0.22933607E 08	-0.1925949E 08	13	0	3	0
2	0.1301412E 08	-0.15691301E 06	15	0	4	0
3	0.48985028E 07	0.1360431E 07	13	0	2	0
4	0.19962501E 07	-0.12316037E 07	10	0	4	0

EIGENVECTORS

COLUMN	1	COLUMN	2	COLUMN	3	COLUMN	4	COLUMN	5	COLUMN	6
1	-0.22672200E 00	0.07881241E-01	-0.10980637E 00	-0.10249931E 00	-0.11566342E 00	0.10828242E-02					
2	-0.27432261E 00	0.96231452E-01	-0.11757997E 00	-0.13723300E 00	-0.14282341E 00	0.42084159E-03					
3	-0.10896540E 00	0.90128533E-01	0.36834285E-02	-0.72438926E-01	0.39344759E-01	-0.37217868E-01					
4	-0.11140210E 00	0.90914406E-01	0.36912900E-02	-0.12856633E 00	0.62688741E-02	-0.41506400E-01					
5	0.16390737E 00	0.20669461E-01	0.27364991E 00	-0.15872387E-01	0.38962762E 00	-0.14037653E 00					
6	0.13893190E 00	0.93805586E-01	0.27711217E 00	-0.70919944E-01	0.38137447E 00	-0.15814607E 00					
7	0.57352650E 00	0.51462600E-01	0.68613424E 00	0.20940761E-01	0.72148682E 00	-0.93107597E-01					
8	0.54671533E 00	0.64242830E-01	0.64748630E 00	-0.26468319E-01	0.75147159E 00	-0.13747385E 00					
9	0.10000300E 00	0.39255811E-00	0.98694071E 00	0.36698160E-01	0.94384034E 00	0.67121342E-01					
10	0.97166492E 00	0.10200000E-01	0.10000000E 01	0.10248105E-09	0.10000000E 01	0.73126528E-09					
			COLUMN								
			COLUMN								
1	0.55517504E-01	-0.28657167E-01									
2	0.37101370E-01	-0.37456806E-02									
3	-0.61798311E-01	-0.41926120E-01									
4	-0.72267180E-01	-0.58044272E-01									
5	-0.53630950E 00	-0.13743654E 00									
6	-0.51810123E 00	-0.16363449E 00									
7	-0.12079909E 00	-0.60967843E-01									
8	-0.82737129E-01	-0.13077731E 00									
9	0.10000000E 01	0.10000000E 01									
10	0.04100371E 00	-0.00704760E-02									

CHECK EIGENVALUES AND EIGENVECTORS

EIGENVALUE		EIGENVECTOR		EIGENVALUE		EIGENVECTOR		EIGENVALUE		EIGENVECTOR		EIGENVALUE		EIGENVECTOR	
MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)	MODE	FRQCY (GPH)
1	0.2291307E 08	0.1592555E 08	0.1301411E 08	0.1569169E 06	0.4890500E 07	0.1360439E 07									
2	0.1994759E 07	0.1231507E 07													
3	0.2247211E 06	0.981248E-01	0.1125298E 00	0.1372329E 00	0.1156632E 00	0.1087919E-02									
4	0.1089054E 00	0.9623146E-01	0.1683497E-02	0.7243889E-01	0.3934485E-01	0.3721773E-01									
5	0.1314021E 00	0.9591447E-01	0.3669231E-02	0.1286663E 00	0.6260245E-02	0.4150637E-01									
6	0.1639073E 00	0.9466946E-01	0.2738499E 00	0.1587235E-01	0.3896277E 00	0.1403485E 00									
7	0.1089119E 00	0.9380552E-01	0.2771122E 00	0.0519907E-01	0.3813746E 00	0.1581466E 00									
8	0.5735265E 00	0.5146266E-01	0.6380143E 00	0.2094076E-01	0.7214869E 00	0.9310964E-01									
9	0.5457117E 00	0.6424832E-01	0.6474801E 00	0.2646030E-01	0.7514716E 00	0.1374736E 00									
10	0.1060000E 01	0.1825602E-01	0.6694071E 00	0.3669010E-01	0.9438405E 00	0.6712124E-01									
11	0.2910638E 00	0.1122000E-01	0.1000000E 01	0.1926815E-10	0.1000000E 01	0.1159552E-08									
12	0.5351719E-01	0.2863631E-01													
13	0.3750271E-01	0.3145196E-02													
14	0.6129842E-01	0.8192167E-01													
15	0.7276375E-01	0.5864522E-01													
16	0.5363055E 00	0.1137432E 00													
17	0.5161036E 00	0.1630335E 00													
18	0.1202019E 00	0.6962363E-01													
19	0.8736019E-01	0.1189159E 00													
20	0.1000000E 01	0.1274370E-08													
21	0.9430927E 00	0.8078056E-02													

MODE FRQCY (GPH) DAMPING VELOCITY (KNOIS)

MODE	FRQCY (GPH)	DAMPING	VELOCITY (KNOIS)
1	0.2044799E 01	0.6944571E 00	0.7081911E 03
2	0.27499231E 01	0.1205620E-01	0.92947031E 03
3	0.4471336E 01	0.2781806E 00	0.15162683E 04
4	0.69985134E 01	0.61699621E 00	0.23732553E 04

SECTION 8

PROGRAM LISTING

[illegible]

1

1
2
3
4
5
6
7
8
9

HM14007
HM14007
HM14007
HM14008
HM14008
HM14008
HM14008
HM14008
HM14008
HM14008
HM14008
HM14008
HM14009
HM14009
HM14009
HM14009
HM14009
HM14010
HM14010
HM14010

HM14010
HM14010
HM14010
HM14010
HM14010
HM14010

1

1

001
002
003
004

1

1

001
002
003
004
005
006

DIMENSION ISXST(20), ISW(20), HR(21), S(21), RHO(20), UM(6,6),
 1 DU(6,12), DMBAR(6,12), RARMRP(6,12), LOW(50), LHIGH(20)
 2 , IT(218), VELCT(20), NSIZES(20),
 3 A(50,100), F(50,100), U(50,100), HR(50,6), HRT(6,100),
 4 HT(6,100), G(6,100)

C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
 C COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---

COMMON IT

EQUIVALENCE

(IT(1),ISXST), (IT(21),ISW), (IT(41),RHO),
 1 (IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),
 2 (IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),
 3 (IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),
 4 (IT(70),NSUR), (IT(71),NRIGID), (IT(72),BREF),
 5 (IT(73),NAERO), (IT(74),NFUS), (IT(75),NDENS),
 6 (IT(76),MODES), (IT(77),NPOINT), (IT(78),NPUNCH),
 7 (IT(79),MAXR), (IT(80),MAXQ), (IT(81),MAXS),
 8 (IT(82),NC), (IT(83),NSURFS), (IT(84),NR2),
 9 (IT(85),HR), (IT(106),S), (IT(127),VELCTY),
 EQUIVALENCE (IT(147),NTAPEU), (IT(148),NRHO), (IT(149),NITRSP),
 1 (IT(150),NITRDP), (IT(151),FPSP), (IT(152),FPDP),
 2 (IT(153),PICON), (IT(154),NVEL), (IT(155),NCARDS),
 3 (IT(156),AITKEN), (IT(157),NGO), (F,0), (F(601,1),HT)
 4 (U,LOW), (U(51,1),LHIGH), (IT(158),FK), (IT(159),SQR
 5 CON) , (IT(160),NSIZES), (IT(180),UM), (DM,AR,RARMRP)
 6 , (IT(216),NDELH), (IT(217),AITKED), (IT(218),KPART)

1 FORMAT (1814)

2 FORMAT (6E12.8)

3 FORMAT (1H 16X, 41H FLUTTER ANALYSIS BY A COLLOCATION METHOD

1 42H USING AERODYNAMIC INFLUENCE COEFFICIENTS /// 10H NSUR =

2 112, 10H NAERO = 114, 11H NRIGID = 112, 9H NFUS =

3 112, 10H NDENS = 114, 14H MODES OUT = 112

4 , 9H NDELM = 112, 10H NPUNCH = 112)

4 FORMAT (1H0.22X, 11H B (REF) = 1E20.8, 5X, 4HK = 1E20.8 // 1H0 25X,

1 7HSURFACE 18X, 1HR 19X, 1HS 10X, 20HEXTERNAL STORES SIZE //)

5 FORMAT (1H0 10X, 21H B RIGID COMPONENT = 1E18.8, 5Y, 8H S RIGID

1 13H COMPONENT = 1E18.8)

6 FORMAT (1H 1120, 2(5X, 1E20.8), 1112)

10 FORMAT (1H1 48X, 12H MASS MATRIX)

11 FORMAT (41H0 NUMBER OF CONTROL POINTS THIS MATRIX, (114,

1 48H) AND TOTAL NUMBER OF CONTROL POINTS EXPECTED, (114,

2 57H) DO NOT AGREE. PROGRAM CONTINUED....)

12 FORMAT (1H) 42X, 24H RIGID BODY MODAL MATRIX)

13 FORMAT (1H 58X, 8HSURFACE 112, 1H, 116, 15H CONTROL POINTS)

14 FORMAT (1H) 41X, 20H FLEXIBILITY MATRIX)

15 FORMAT (1H) 46X, 18H WEIGHTING MATRIX)

16 FORMAT (1H) 20X, 20H RIGID COMPONENT AERO MATRIX, 119, 8H CONTROL

1 7H POINTS)

17 FORMAT (1H) 31X, 20H AERODYNAMIC MATRIX 8X, 10H 1./K R =

1 1120.8)

18 FORMAT (1H0 40X, 23H RIGID COMPONENT MODES, 1110,

1 17H CONTROL POINTS.)

19 FORMAT (1H0 50X, 24H RIGID COMPONENT MASS MATRIX)

20 FORMAT (47H) ERROR IN INVERSE ROUTINE. PROGRAM TERMINATED)

21 FORMAT (1H 58X, 8HSURFACE 112, 1H, 6X, 13H NO WEIGHTING

1 7H MATRIX)

22 FORMAT (3E12.8, 214)

23 FORMAT (4H0FPSP = 1E16.8, 1X, 8H EPDP = 1E16.8, 1X,

1 10H AITKEN = 1E16.8, 1X, 10H NITRSP = 114, 1X,

2 10H NITRDP = 114)

24 FORMAT (1H0 40X, 20H GENERALIZED MASSES /// (1H 30X,

1 5H MASS 114, 3H = 1E16.8))

25	FORMAT	(1H0 30X, 23H RIGID COMPONENT MODES ///	)	079
26	FORMAT	(214, 62X, 1A6, 114)		080
27	FORMAT	(1H0 33X, 23H PUNCHED CARDS NUMBERS 1A6, 114, 6H THRU		081
1		1A6, 114)		082
28	FORMAT	(1H0 20X, 3H K= 1E16.R	)	083
		DATA Q000CT/0202030440005/		
		RCDZ =Q000CT		085

```

$      OPTION FORTRAN
$      FORTRAN LISTING DECK
C MAIN      FLUTTER OVERLAY
C           JAN 15, 1967
C           COMMON IT(218)
C           CHANGE BCD TAPES TO BINARY TAPES
10 CALL LLINK (6HPAR1TT)
      CALL PART1
      CALL LLINK (6HPAR2TT)

```

```

$      OPTION FORTRAN
$      FORTRAN LST00,DECK
C MAIN      FLUTTER OVERLAY
C            JAN 13,1967

```

```

COMMON IT(218)
C      CHANGE BCD TAPES TO BINARY TAPES
10 CALL LLINK (6HPAR1TT)
   CALL PART1
   CALL LLINK (6HPAR2TT)

   CALL PART2
   GO TO 10
END

```

```

$      FORTRAN LST00,DECK

```

```

C MPRINT

```

```

SUBROUTINE MPRINT (A,M,N,MD,NTAPE)
  DIMENSION A(1), IT(6), C(6)
  EQUIVALENCE (IT,C)
  2 FORMAT (1H , 4X, 6( 6X, 7HCOLUMN 114 ) /// )
  3 FORMAT (1H 114, X, (6E 17.8) )
  N1=N
  N2=6
  N3=6
  N4=1
  4 IF (N3-N1) 6,6,5
  5 N2=N1-N3+6
  N3=N1
  6 K=0
  DO 7 I= N4,N3
    K=K+1
  7 IT(K)=I
  WRITE (NTAPE,2) (IT(I),I=1,N2)
  DO 9 I=1,M
    K=0
    L=MD*(N4-1)+I
    DO 8 J=N4,N3
      K=K+1
      C(K)=A(I)
      L=L+MD
  8 WRITE (NTAPE,3) I,(C(K),K=1,N2)
  IF (N3-N1) 10,11,11
  10 N3=N3+6
  N4=N4+6
  GOTO 4
  11 RETURN
END

```

```

$      FORTRAN LST00,DECK

```

```

C MPUNCH

```

```

SUBROUTINE MPUNCH(A,M,N,IOUT,ITRA,IORG,BCDZ,MAXM,NTAPE,NCARDS)
  DIMENSION A(1)
  RETURN
END

```

```

$      LINK      PAR1TT

```

```

$      FORTRAN LST00,DECK

```

```

C PART1

```

```

SUBROUTINE PART1
C      NSUR=TOTAL NUMBER OF SURFACES ALLOWED.
C      NDENS=TOTAL NUMBER OF DENSITIES ALLOWED.
C      NRIGID=TOTAL NUMBER OF RIGID BODIES ALLOWED
C      NSIZE=TOTAL NUMBER CONTROL POINTS ON ANY ONE SURFACE ALLOWED
C      NMODES=TOTAL NUMBER MODES INPUT ON ANY ONE SURFACE.

```

HM14007

HM140078

HM140079

HM14008

HM14008

HM140084

HM14008

HM14008

HM140087

HM140088

HM14008

HM140090

HM140091

HM14009

HM14009

HM140094

HM140096

HM140097

HM140098

HM140099

HM140100

HM140101

HM140102

HM140104

HM140105

HM140106

HM140107

HM140108

HM140109

0010

0020

0030

0040

0010

0020

0030

0040

0050

0060

DIMENSION	ISXST(20), ISW(20), HR(21), S(21), RHO(20), DM(6,6),	0120
1	DO(6,12), DMBAR(6,12), BARMRR(6,12), LOW(50), HIGH(50)	0130
2	, IT(218), VELCTY(20), NSIZES(20),	0140
3	A(50,100), F(50,100), U(50,100), HR(50,6), RT(6,100),	0150
4	HT(6,100), G(6,100)	0160
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO		0170
C COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---		0180
COMMON	IT	0190
EQUIVALENCE	(IT(1),ISXST), (IT(21),ISW), (IT(41),RHO)	0250
1	, (IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),	0260
2	(IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),	0270
3	(IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),	0280
4	(IT(70),NSUR), (IT(71),NRIGID), (IT(72),HREF),	0290
5	(IT(73),NAERO), (IT(74),NFUS), (IT(75),NDENS),	0300
6	(IT(76),MODES), (IT(77),NPPOINT), (IT(78),NPUNCH),	0310
7	(IT(79),MAXR), (IT(80),MAXU), (IT(81),MAXS),	0320
8	(IT(82),NC), (IT(83),NSURFS), (IT(84),N02),	0330
9	(IT(85),HR), (IT(106),S), (IT(127),VELCTY)	0340
EQUIVALENCE	(IT(147),NTAPE0), (IT(148),NRHO), (IT(149),NITRSP),	0350
1	(IT(150),NITRDP), (IT(151),EPSP), (IT(152),EPDP),	0360
2	(IT(153),PICON), (IT(154),NVEL), (IT(155),NCARDS),	0370
3	(IT(156),AITKEN), (IT(157),NGO), (F,G), (F(601,1),HT)	0380
4	(U,LOW), (U(51,1),LHIGH), (IT(158),FK), (IT(159),SQR	0390
5	CON), (IT(160),NSIZES), (IT(180),DM), (DMBAR,BARMRR)	0400
6	, (IT(216),NDELM), (IT(217),AITKED), (IT(218),KPART)	0410
1	FORMAT (1814)	0430
2	FORMAT (6E12.8)	0440
3	FORMAT (1H 16X, 41H FLUTTER ANALYSIS BY A COLLOCATION METHOD	0450
1	42H USING AERODYNAMIC INFLUENCE COEFFICIENTS ///10H NSUR =	0460
2	112, 10H NAERO = 114, 11H NRIGID = 112, 9H NFUS =	0470
3	112, 10H NDENS = 114, 14H MODES OUT = 112	0480
4	, 9H NDELM = 112, 10H NPUNCH = 112)	0490
4	FORMAT (1H0 22X, 11H H (REF) = 1E20.8, 5X, 4HK = 1E20.8 /1H0 25X,	0500
1	7HSURFACE 18X, 1HB 19X, 1HS 10X, 20HEXTERNAL STORES SIZE ///	0510
5	FORMAT (1H0 10X, 21H R RIGID COMPONENT = 1E18.8, 5V, 8H S RIGID	0520
1	13H COMPONENT = 1E18.8)	0530
6	FORMAT (1H 1E20, 2(5X, 1E20.8), 1112)	0540
10	FORMAT (1H1 48X, 12H MASS MATRIX)	0550
11	FORMAT (41H0 NUMBER OF CONTROL POINTS THIS MATRIX, (114,	0560
1	48H) AND TOTAL NUMBER OF CONTROL POINTS EXPECTED, (114,	0570
2	37H) DO NOT AGREE, PROGRAM CONTINUED....)	0580
12	FORMAT (1H1 42X, 24H RIGID BODY MODAL MATRIX)	0590
13	FORMAT (1H 38X, 8HSURFACE 112, 1H, 116, 15H CONTROL POINTS)	0600
14	FORMAT (1H1 43X, 20H FLEXIBILITY MATRIX)	0610
15	FORMAT (1H1 46X, 18H WEIGHTING MATRIX)	0620
16	FORMAT (1H1 28X, 29H RIGID COMPONENT AERO MATRIX, 119, 8H CONTROL	0630
1	7H POINTS)	0640
17	FORMAT (1H1 34X, 20H AERODYNAMIC MATRIX 8X, 10H 1./K R =	0650
1	1E20.8)	0660
18	FORMAT (1H0 50X, 23H RIGID COMPONENT MODES, 1110,	0670
1	17H CONTROL POINTS.)	0680
19	FORMAT (1H0 50X, 29H RIGID COMPONENT MASS MATRIX)	0690
20	FORMAT (47H1 ERROR IN INVERSE ROUTINE, PROGRAM TERMINATED)	0700
21	FORMAT (1H 38X, 8HSURFACE 112, 1H, 6X, 13H NO WEIGHTING	0710
1	7H MATRIX)	0720
22	FORMAT (3E12.8, 214)	0730
23	FORMAT (8H0EPSP = 1E16.8, 1X, 8H EPDP = 1E16.8, 1X,	0740
1	10H AITKEN = 1E16.8, 1X, 10H NITRSP = 114, 1X,	0750
2	10H NITRDP = 114)	0760
24	FORMAT (1H0 40X, 20H GENERALIZED MASSES //// (1H 30X,	0770
1	5H MASS 114, 3H = 1E16.8))	0780

```

25 FORMAT ( 1H0 3IX, 23H RIGID COMPONENT MODES  /// )
26 FORMAT ( 214, 62X, 1A6, 114 )
27 FORMAT ( 1H0 3IX, 23H PUNCHED CARDS NUMBERS 1A6, 114, 6H THRU
1 1A6, 114 )
28 FORMAT (1H0 20X, 3H K= 1E16.8 )
DATA 0000CT/0202030440005/
RCDZ =0000CT
C NTAPE0 = PUNCH OUTPUT TAPE
C NTAPE2 = INPUT TAPE
C NTAPE3 = OUTPUT PRINT TAPE
C NTAPE4 = /
C NTAPE5 = / ARE UTILITY TAPES
C NTAPE6 = /
C NTAPE7 = /
C NTAPE8 = /
C NTAPE9 = /
IF ( NG0-98765 ) 90,97,99
97 IF ( NAFRO ) 98,320.98
98 NAFRO = NAERO-1
IF ( NAFRO ) 99,99,170
99 NTAPE0 = 8
NTAPE2=5
NTAPE3=6
NTAPE4 = 9
NTAPE5 = 3
NTAPE6 = 4
NTAPE7=11
NTAPE8=1
NTAPE9 =10
MAXR = 50
MAXQ = 6
NCON=0
MAXS=50
991 REWIND NTAPE4
REWIND NTAPE5
REWIND NTAPE6
REWIND NTAPE9
SORCON = SQRT( 386.0 ) / (2.0*3.14159 )
EPSP = .5E-06
FPDP = .5E-07
AITKEN = .9
AITKED = .9
PICON = .5921*2.0*3.14159
NCARDS = 0
NVEI = 0
NC=1
NITRSP=40
NITRDP=100
RHO(1) = 0.0
C.....
C READ IN TITLE, CONTROLS AND CONSTANTS AND PRINT.
100 CALL RDIN (NTAPE2,NTAPE3,1)
READ (NTAPE2,1)NSUR, NAERO, NRIGID, NFUS, NDENS, MODES
INDFIM, NPUNCH, NCON
READ (NTAPE2,2)FK
WRITE (NTAPE3,3)NSUR,NAERO, NRIGID,NFUS, NDENS,MODES
1 , INDFIM, NPUNCH
NR2=NRIGID*NC
NSURFS=NSUR+NFUS
IF ( NCON ) 102,104,102
102 READ (NTAPE2,22)EPSP,FPDP,AITKEN, NITRSP,NITRDP

```

0790
0800
0810
0820
0830
0850
0860
0880
0890
0900
0910
0920
0930
0940
0950
0960
0970
0980
0990
1000

1030
1040
1050

1080
1090
1100

1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390

```

READ (NTAPE2,2)AITKEN
WRITE (NTAPE3,23)EPSP, EPDP, AITKEN, NITRSP, NITROP
104 IF ( NAERO ) 105,103,105
103 WRITE (NTAPE3,28)FK
GOTO 111
105 READ (NTAPE2,1)(ISXST(I),ISW(I),I=1,NSUR)
READ (NTAPE2,2)RREF, (VELCTY(I),I=1,NAERO)
NC=2
NR2= NRIGID *NC
READ (NTAPE2,2)(BR(I),S(I),I=1,NSURFS)
IF ( NDENS ) 106,107,106
106 READ (NTAPE2,2)(RHO(I),I=1,NDENS)
107 IF ( NFUS ) 108,109,108
108 WRITE (NTAPE3,5)BR(1), S(1)
109 WRITE (NTAPE3,4)RREF ,FK
DO 110 I=1,NSUR
J=I+NFUS
110 WRITE (NTAPE3,6) I, BR(J), S(J) ,ISXST(I)
*****
C READ IN FUSE OF MASS CHARACTERISTICS
111 IF ( NRIGID ) 112,117,112
112 IF ( NDELM ) 115,113,115
113 DO 114 I=1,NRIGID
DO 114 J=1,NRIGID
114 DM(I,J)=0.0
GOTO 117
115 DO 116 I=1,NRIGID
116 READ (NTAPE2,2)(DM(J,I),J=1,NRIGID)
WRITE (NTAPE3,19)
CALL MPRINT (DM,NRIGID,NRIGID,MAXQ,NTAPE3)
*****
C READ MASS MATRIX FOR EACH SURFACE, STORE SYSTEM MASS MATRIX ON NTAP4
117 K1=0
WRITE (NTAPE3,10)
DO 120 I=1,NSUR
READ (NTAPE2,1)NSIZE
READ (NTAPE2,1)(LOW(J),LHIGH(J),J=1,NSIZE)
DO 119 J=1,NSIZE
DO 118 K=1,NSIZE
118 A(K,J)=0.0
N1=LOW(J)
N2=LHIGH(J)
119 READ (NTAPE2,2)(A(N,J),N=N1,N2)
WRITE (NTAPE4)NSIZE, NSIZE, ((A(N,J),N=1,NSIZE),J=1,NSIZE)
WRITE (NTAPE3,11)I, NSIZE
CALL MPRINT (A,NSIZE,NSIZE,MAXR,NTAPE4)
NSIZEFS(I)=NSIZE
120 K1=K1+NSIZE
*****
NPOINT = TOTAL NUMBER OF CONTROL POINTS ON ALL SURFACES.
NPOINT=K1
C READ IN RIGID BODY MODAL MATRIX FOR EACH SYSTEM.
IF ( NFUS ) 121,123,121
121 READ (NTAPE2,1)NSIZE
DO 122 J=1,NRIGID
122 READ (NTAPE2,2)(HR(I,J),I=1,NSIZE)
WRITE (NTAPE3,18)NSIZE
CALL MPRINT (HR,NSIZE,NRIGID,MAXS,NTAPE3)
WRITE (NTAPE4)NSIZE, NRIGID, ((HR(I,J),I=1,NSIZE),J=1,NRIGID)
123 IF ( NRIGID ) 124,128,124
124 K1=1

```

```

WRITE (NTAPE3,12)
DO 126 ISUR=1,NSUR
  READ (NTAPE2,1)NSIZE
  K2=K1+NSIZE-1
  DO 125 J=1,NRIGID
125  READ (NTAPE2,2)(HR(I,J),I=K1,K2)
    WRITE (NTAPE3,13)ISUR,NSIZE
    CALL MPRINT (HR(K1,1),NSIZE,NRIGID,MAXS,NTAPE3)
126  K1=K1+NSIZE
  WRITE (NTAPE5)NPOINT, NRIGID, ((HR(N,J),N=1,NPOINT),I=1,
    1 NRIGID )
  IF ( K1-NPOINT-1) 127,128,127
127 WRITE (NTAPE3,11)K1, NPOINT
*****
READ IN FLEXIBILITY MATRIX FOR EACH SURFACE,      STORE ON NTAPES
128 K1=0
  WRITE (NTAPE3,14)
  DO 153 I=1,NSUR
  READ (NTAPE2,1)NSIZE, J, IFORM, IROWS
  CALL MREAD (A,NSIZE,NSIZE,IFORM,IROWS,0,1,F,MAXR,NTAPE2,NTAPE3)
  WRITE (NTAPE3,15)I,NSIZE
  CALL MPRINT (A,NSIZE,NSIZE,MAXR,NTAPE3)
  WRITE (NTAPE5)NSIZE, NSIZE, ((A(J,K),J=1,NSIZE),K=1,NSIZE)
153 K1=K1+NSIZE
  IF ( K1-NPOINT ) 158,164,158
158 WRITE (NTAPE3,11)K1, NPOINT
164 IF ( NAERO ) 166,165,166
165 NRHO=1
  DO 500 I=1,NRIGID
  DO 500 J=1,NRIGID
500  DQ(I,J)=0.0
  WRITE (NTAPE5)((DM(I,J),I=1,NRIGID),J=1,NRIGID)
  NCX=NC
  GOTO 215
C*****
C READ IN WEIGHTING MATRIX FOR EACH SURFACE.....STORE ON NTAPES,
166 WRITE (NTAPE3,16)
  DO 178 I=1,NSUR
  IF ( ISW(I) ) 167,177,167
167 N1=ISYST(I)
  IF ( N1 ) 168,172,168
168 DO 170 J=1,N1
  DO 169 L=1,MAXR
  A(J,L)=0.0
169 A(L,J)=0.0
170 A(J,J)=1.0
  READ (NTAPE2,1)NXST, J, IFORM, IROW
  IF ( NXST ) 171,172,171
171 CALL MREAD (A,NXST,NXST,IFORM,IROW,0,1,F,MAXR,NTAPE2,NTAPE3)
172 K=N1+1
  READ (NTAPE2,1)NSIZE, NPART, IFORM, IROW
  IF ( IFORM ) 174,173,174
173 CALL MREAD (A(K,K),NSIZE,NSIZE,0,0,0,1,F,MAXR,NTAPE2,NTAPE3)
  K=K+NSIZE
  GOTO 176
174 DO 175 J=1,NPART
  READ (NTAPE2,1)NSIZE
  M=K+NC
  DO 1741 L=K,N
  DO 1741 M=1,K
1741 A(M,L)=0.0

```

2030
 2040
 2050
 2060
 2070
 2080
 2090
 2100
 2110
 2120
 2130
 2150
 2160
 2170
 2180
 2190
 2200
 2210
 2220
 2230
 2240
 2250
 2260
 2280
 2290
 2300
 2310
 2320
 2330
 2340
 2350
 2360
 2380
 2390
 2400
 2410
 2420
 2430
 2440
 2450
 2460
 2470
 2480
 2490
 2500
 2510
 2520
 2530
 2540
 2550
 2560
 2570
 2580
 2590
 2600
 2610
 2620
 2630
 2640
 2650
 2660

CALL MREAD (A(K,K),NSIZE,NSIZE,1,IROW,0,0,F,MAXR,NTAPE2,NTAPE3)	2670
175 K=K+NSIZE	2680
176 K=K-1	2690
*****	2700
WRITE (NTAPE3,1)I, K	2710
CALL MPRINT (A,K,K,MAXR,NTAPE3)	2720
WRITE (NTAPE6)K, K, ((A(J,L),J=1,K),L=1,K)	2730
GOTO 178	2740
177 WRITE (NTAPE3,2)I	2750
178 CONTINUE	2760
179 REWIND NTAPE6	2770
REWIND NTAPE7	2780
NCX=NC	2790
NVFL = NVFL+1	2800
*****	2810
IF SERIES OF DENSITIES FOR EACH V/R OMEGA, READ IN THAT SERIES.	2820
IF (NDENS) 180,181,180	2830
180 NRHO=NDENS	2840
GOTO 182	2850
181 READ (NTAPE2,1)NRHO	2860
READ (NTAPE2,2)(RHO(I),I=1,NRHO)	2870
*****	2880
READ IN COMPLEX AERODYNAMIC MATRIX FOR EACH SURFACE	2890
182 DO 210 I=1,NSURFS	2900
K=1	2910
K2=1	2920
IF (NFUS) 183,184,183	2930
183 IF (I-1) 184,188,184	2940
184 L=I-NFUS	2950
IF (ISXST(L)) 185,188,185	2960
185 K=ISXST(L)+1	2970
K2 = K+ISXST(L)	2980
DO 186 J=1,K	2990
DO 186 L=1,K2	3000
186 A(J,L)=0.0	3010
READ (NTAPE2,1)NXST, J, IFORM, IROW	3020
IF (NXST) 187,188,187	3030
187 N= NXST *NC	3040
CALL MREAD (A,NXST,N,IFORM,IROW,0,1,U,MAXR,NTAPE2,NTAPE3)	3050
188 READ (NTAPE2,2)VFLC	3060
READ (NTAPE2,1)NSIZE, NPART, IFORM, IROW	3070
IF (IFORM) 190,189,190	3080
189 N= NSIZE *NC	3090
CALL MREAD (A(K,K2),NSIZE,N,IFORM,IROW,0,1,U,MAXR,NTAPE2,NTAPE3)	3100
NSIZEF=NSIZE+K-1	3110
GOTO 193	3120
190 DO 192 J=1,NPART	3130
READ (NTAPE2,1)NSIZE	3140
N=K2+NC	3150
DO 191 M=K2,N	3160
DO 191 I=1,K	3170
191 A(I,M)=0.0	3180
N= NSIZEF *NC	3190
CALL MREAD (A(K,K2),NSIZE,N,1,IROW,0,0,U,MAXR,NTAPE2,NTAPE3)	3200
K=K+NSIZEF	3210
192 K2=K2+N	3220
NSIZEF=K-1	3230
N=K2-1	3240
193 IF (I-1) 199,194,190	3250
194 IF (NFUS) 195,197,195	3260
195 WRITE (NTAPE3,1)NSIZE	3270

```

      CALL MPRINT (A,NSIZE,N,MAXR,NTAPE3)
C*****
C  COMPUTE (DQ) = (HR)T * (CH) * (HR)
      K1=1
      K2=1
      M=NC-1
      READ (NTAPE4)NSIZE, L, ((U(L,J),L=1,NSIZE),J=1,NRIGID)
      CALL MMULTD (A,M,U,0,F,NSIZE,NSIZE,NRIGID,MAXR,MAXR,MAXR)
      DO 196 L=1,NSIZE
        DO 196 J=1,NRIGID
196      A(J,L)=U(L,J)
      CALL MMULTD (A,0,F,M,DQ,NRIGID,NSIZE,NRIGID,MAXR,MAXR,MAXR)
      GOTO 210
197 DO 198 L=1,NRIGID
      DO 198 J=1,NR2
198      DQ(L,J)=0.0
199 I=1-NFUS
      IF ( ISW(L) )      202,200,202
200 WRITE (NTAPE7)NSIZE, N, ((A(J,M),J=1,NSIZE),M=1,N)
      GOTO 208
202 READ (NTAPE4)I, L, ((F(J,M),J=1,L),M=1,L)
      IF ( NSIZE-L )      204,204,204
204 WRITE (NTAPE3,11)NSIZE,L
206 CALL MMULTD (F,0,A,NC-1,U,L,L,L,MAXR,MAXR,MAXR)
      WRITE (NTAPE7)NSIZE, N, ((U(J,M),J=1,NSIZE),M=1,N)
208 WRITE (NTAPE3,17)VELC
      L=1-NFUS
      WRITE (NTAPE3,13)L, NSIZE
      CALL MPRINT (A,NSIZE,N,MAXR,NTAPE3)
210 CONTINUE
C*****
C  CARRY ON FROM HERE TO END ONCE FOR EACH DENSITY.
215 DO 300 IRHO=1,NRHO
      K=NC*NPOINT
      DO 216 I=1,NPOINT
        DO 216 J=1,K
216      U(I,J)=0.0
      REWIND NTAPE4
      REWIND NTAPE5
      REWIND NTAPE7
      CON=RHO(IRHO)*RR(1)**2*S(1) *32.174
      IF ( NPTIGID ) 218,222,218
218 DO 220 I=1,NRIGID
        DO 220 J=1,NR2*NCX
          DMBAR(I,J+1)=CON*DQ(I,J+1)
          K=J/NC+NC-1
220      DMBAR(I,J)=DM(I,K)+CON*DQ(I,J)
C*****
C  READ ENTIRE HR MATRIX.
      READ (NTAPE4)I, I, ((HR(I,J),I=1,NPOINT),J=1,NRIGID)
222 K1=0
      DO 246 ISUR=1,NSUR
        K=ISUR+NFUS
        CON=RHO(IRHO)*RR(K)**2*S(K) *32.174
C*****
C      READ (M) I
      READ (NTAPE4)NSIZE, NSIZE, ((F(I,J),I=1,NSIZE), I=1,NSIZE)
      IF ( NAFRO )      221,221,221
221 DO 226 I=1,NSIZE
      DO 226 J=1,NSIZE
226      A(I,J)=F(I,J)

```

3280
 3290
 3300
 3310
 3320
 3330
 3340
 3350
 3360
 3370
 3380
 3390
 3400
 3410
 3420
 3430
 3440
 3450
 3460
 3470
 3480
 3490
 3500
 3510
 3520
 3530
 3550
 3560
 3570
 3580
 3590
 3600
 3610
 3620
 3630
 3640
 3650
 3660
 3670
 3680
 3690
 3700
 3710
 3720
 3730
 3740
 3750
 3760
 3770
 3780
 3790
 3800
 3810
 3820
 3830
 3840
 3850
 3860
 3870
 3880
 3890
 3900

```

      I=NSIZE
      GOTO 231
C*****
C      READ (W)*(CH) I
      223 READ (NTAPE7) NSIZE, L, ((A(I,J), I=1, NSIZE), J=1, L)
C*****
C      COMPUTE (M BAR) = ((H)+RHO*RR**2*S*(W)*(CH) ) I
      DO 228 I=J, NSIZE
      DO 228 J=1, L, NCX
      K=J/2+1
      A(I,J)=CON*A(I,J)+F(I,K)
      228 A(I,J+1)=CON*A(I,J+1)
C*****
C READ FIFTH RIGIDITY MATRIX FOR SURFACE
      231 READ (NTAPE5) NSIZE, NSIZE, ((F(I,J), I=1, NSIZE), J=1, NSIZE)
      K=NC*K1+1
      CALL MMULTD (F, 0, A, NC-1, U(K1+1, K), NSIZE, NSIZE, NSIZE, MAXR, MAXR,
      1 MAXR)
      IF (NRIGID) 232, 246, 232
C*****
C FIND (LITTLE M BAR) = (DELTA M) + (H R) TRANSPOSED * (M BAR) * (H R)
      232 DO 240 I=1, NSIZE
      K=K1+1
      DO 240 J=1, NRIGID
      240 G(J,I)=HR(K,J)
      K=NC*K1+1
      CALL MMULTD (G, 0, A, NC-1, HRT(1, K), NRIGID, NSIZE, NSIZE, MAXU, MAXR,
      1 MAXU)
      CALL MMULTD (HRT(1, K), NC-1, HR(K1+1, 1), 0, A, NRIGID, NSIZE, NRIGID,
      1 MAXU, MAXS, MAXR)
      DO 244 I=1, NRIGID
      DO 244 J=1, NRIGID
      244 BARMRR(I,J)=BARMRR(I,J)+A(I,J)
      246 K1=K1+NSIZE
      L=NC*NPOINT
      IF (NRIGID) 247, 268, 247
      247 GOTO (252, 248), NC
      248 DO 250 I=1, NRIGID
      DO 250 J=1, NRIGID, NCX
      K=J/2+1
      G(I,K)=BARMRR(I,J+1)
      250 BARMRR(I,K)=BARMRR(I,J)
C*****
C THEN (LITTLE M BAR) INVERSE AND FINAL U MATRIX STORED ON TAPE9.
      252 CALL MNVRSX (BARMRR, 0, A(1,1), A(1, MAXQ), NRIGID, IR, NC-1)
      IF (IR) 310, 254, 310
      254 GOTO (260, 256), NCX
      256 DO 258 I=1, NRIGID
      DO 258 J=1, NRIGID, NCX
      K=NRIGID-J
      M=K/2+1
      BARMRR(I,K)=BARMRR(I,M)
      258 BARMRR(I,K+1)=G(I,M)
      260 CALL MMULTD (BARMRR, NC-1, HRT, NC-1, F, NRIGID, NRIGID, NPOINT,
      1 MAXQ, MAXQ, MAXR)
      CALL MMULTD (F, NC-1, U, NC-1, HRT, NRIGID, NPOINT, NPOINT, MAXR, MAXR,
      1 MAXQ)
      CALL MMULTD (HR, 1, F, NC-1, A, NPOINT, NRIGID, NPOINT, MAXQ, MAXR, MAXR)
      DO 264 I=1, NPOINT
      DO 262 J=1, I
      F(I,J)=U(I,J)

```

```

262     A(I,J)=-A(I,J)
      K=I*NC-NC+1
264     A(I,K)=1.0+A(I,K)
      CALL MMULTD (A,NC-1,F,NC-1,U,NPOINT,NPOINT,NPOINT,MAXR,MAXR,MAXR)
      IF ( NAERO )      268,266,268
266     READ (NTAPE5)A(1,1)
      WRITE (NTAPE5)((HRT(I,J),I=1,NRIGID),J=1,NPOINT)
268     WRITE (NTAPE9)NPOINT, L, ((U(I,J),I=1,NPOINT),J=1,L)
300     CONTINUE
      REWIND NTAPE5
      REWIND NTAPE6
      REWIND NTAPE7
      REWIND NTAPE8
      REWIND NTAPE9
      KPART = 2
      RETURN
310     WRITE (NTAPE3,20)
      GOTO 991
320     REWIND NTAPE4
      REWIND NTAPE5
      IF ( NRIGID )      330,332,330
330     READ (NTAPE5)I,I, A(1,1)
332     K1=1
      DO 334 I=1,NPOINT
        DO 334 J=1,NPOINT
334       F(I,J)=0.0
        DO 336 ISUR=1,NSUR
          K2=K1+NSIZES(ISUR)-1
          READ (NTAPE5)I,I, A(1,1)
          READ (NTAPE4)NSIZE, NSIZE, ((F(I,J),I=K1,K2),J=K1,K2)
336       K1=K1+NSIZE
          READ (NTAPE4)NSIZE, MODE, ((A(1,J),I=1,NSIZE),J=1,MODE)
          M=MODE/NC
          CALL MMULTD (F,0,A,NC-1,U,NPOINT,NPOINT,M,MAXR,MAXR,MAXR)
          IF ( NDELM )      337,348,337
337       IF ( NRIGID )      338,348,338
338       READ (NTAPE5)((DM(I,J),I=1,NRIGID),J=1,NRIGID)
          READ (NTAPE5)((G(I,J),I=1,NRIGID),J=1,NPOINT)
          DO 339 I=1,NRIGID
            DO 339 J=1,NPOINT
339       G(I,J)=-G(I,J)
          CALL MMULTD (G,0,A,0,HRT,NRIGID,NPOINT,M,MAXQ,MAXR,MAXQ)
          READ (NTAPE4) (HR(I,1),I=1,M)
          DO 340 I=1,M
            DO 340 J=1,NRIGID
340       G(J,I) = HRT(J,I)/HR(I,1)
          WRITE (NTAPE3,25)
          CALL MPRINT (G,NRIGID,M,MAXQ,NTAPE3)
          IF ( NPUNCH )      343,345,343
343       WRITE (NTAPE10,26)NRIGID, M, RCDZ, NCARDS
          NCARD = NCARDS+1
          CALL MPUNCH ( G,NRIGID,M,0,1,1,RCDZ,MAXQ,NTAPE10,NCARD)
          WRITE (NTAPE3,27)RCDZ, NCARDS, RCDZ, NCARD
          NCARDS=NCARD+1
345     CONTINUE
      CALL MMULTD (DM,0,G,0,HRT,NRIGID,NRIGID,M,MAXQ,MAXQ,MAXQ)
      DO 346 I=1,NRIGID
        DO 346 J=1,MODE
346       HR(J,I) = G(I,J)
348     DO 350 I=1,NPOINT
      DO 350 I=1,MODE

```

4500
4510
4520
4530
4540
4550
4560
4580
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4900
4900
4910
4920
4930
4940
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140

350	F(J,I)=A(I,J)	5150
	CALL MMULTD (F,N,U,N,A,M,NPOINT,M,MAXR,MAXR,MAXR)	5160
	IF (NDELM) 352,358,352	5170
352	IF (NRGID) 354,358,354	5180
354	CALL MMULTD (HR,N,HRT,N,F,M,NRGID,M,MAXR,MAXO,MAXR)	5190
	DO 356 I=1,MDEF	5200
	DO 356 J=1,M	5210
356	A(J,I) = A(J,I) + F(J,I)	5220
358	WRITE (NTAPE3,24) (I, A(I,I),I=1,M)	5230
	GOTO 991	5240
	END	5260
	FORTRAN LST00,DECK	
C	MREAD	
	MREAD	M0010
	MATRIX READ SUBROUTINE	M0020
	CALL MREAD (A,M,N,IFORM,IROW,ITRA,IORG,T,MD,NTAPE2,NTAPE3)	M0030
		M0040
	A = MATRIX TO READ IN	M0050
	M = NUMBER OF ROWS	M0060
	N = NUMBER OF COLUMNS	M0070
	IFORM = -1, FORMAT(12A6)	M0080
	= 0, COLUMN BINARY	M0090
	= +1, FORMAT(6E12.8)	M0100
	IROW = .0, MATRIX BY COLUMNS	M0110
	= +1, MATRIX BY ROWS	M0120
	ITRA = 0, TRA CARD AFTER MATRIX	M0130
	= +1, TRA CARD AFTER EACH ROW	M0140
	(OR COLUMN)	M0150
	IORG = ORIGIN OF FIRST C.B. CARD	M0170
	T = MDXN TEMPORARY CELLS	M0180
	MD = DIMENSIONED NUMBER OF ROWS	M0190
	IN A	M0195
	NTAPE2 = INPUT TAPE	M0200
	NTAPE3 = OUTPUT TAPE	M0210
	SUBROUTINE MREAD (A,M,N,IFORM,IROW,ITRA,IORG,T,MD,NTAPE2,NTAPE3)	M0220
	DIMENSION A(1), T(1)	M0230
1	FORMAT (6E12.8)	M0240
2	FORMAT (12A6)	M0250
3	FORMAT (// 26H THATS ALL YOUR DATA.	M0260
4	FORMAT (4E16.8)	M0270
	MN=MD*N	M0280
	DO 5 I=1,MN	M0290
	T(I)=0.0	M0300
5	A(I)=0.0	M0310
	IF (IFORM) 39,15,6	M0320
6	IF (IROW) 8,7,8	M0330
7	K3=1	M0340
	K4=N	M0350
	K5=MD	M0360
	K6=M-1	M0370
	K2=1	M0380
	GOTO 9	M0390
8	K2=MN	M0400
	K3=MD	M0402
	K4=M	M0404
	K5=1	M0406
	K6=0	M0410
9	DO 11 I=1,K4	M0415
	K1=I*K5-K6+1	M0420
	IF (K6) 10,11,10	M0430
10	K2=K1+K6	M0440
11	IF (IFORM = 1) 34,110,109	
109	READ (NTAPE2,4) (A(L),L=K1,K2,K3)	
	GO TO 11	
110	READ (NTAPE2,1)(A(L),L=K1,K2,K3)	
11	CONTINUE	
	GOTO 36	
15	K1=N	
	K2=M	

```

      K3=1
      IF ( IORG-1 ) 16,17,17
16  K3=2
17  IF ( IROW ) 18,19,18
18  K2=N
      K1=M
19  IF ( ITRA ) 22,21,22
21  K1=1
22  K=0
      DO 23 I=1,K1
          K4=K+K3
          K5=1
          CALL RINRD ( T(K4), K5, L, NTAPE2, NTAPE3 )
          GOTO (23,38,23,23),1
23  K=K+K2
      IF ( IROW ) 28,24,28
24  L=0
      IF ( IORG-1 ) 26,26,26
25  L=IORG-1
26  DO 27 I=1,N
          J=I*MD-MD
          DO 27 K=1,M
              J=J+1
          L=L+1
27  A(J)=T(L)
      GOTO 36
28  L=0
      IF ( IORG-1 ) 30,30,29
29  L=IORG-1
30  DO 31 K=1,N
          J= K*MD-MD
          DO 31 I=K,MN,N
              J=J+1
              K1=L+1
31  A(J)=T(K1)
36  RETURN
38  WRITE ( NTAPE3,3 )
      STOP
39  READ ( NTAPE2,2 ) ( A(I), I=1, M )
      GOTO 36
      END

```

5 FORTRAN LSIOU.DECK

C RINRD

```

      SUBROUTINE RINRD ( T,K,L,NTAPE1,NTAPE2 )
      DIMENSION T(1)
      RETURN
      END

```

5 FORTRAN LSIOU.DECK

C RDI N

```

      SUBROUTINE RDI N ( NTAPE2, NTAPE3, I )
1  FORMAT(80H
1
2  FORMAT(1H1)
3  FORMAT ( 1H0 )
      READ ( NTAPE2,1 )
      GOTO (4,5),1
4  WRITE ( NTAPE3,2 )
      GOTO 6
5  WRITE ( NTAPE3,3 )
6  WRITE ( NTAPE3,1 )
      RETURN

```

M0450

M0460

M0470

M0480

M0490

M0500

M0510

M0520

M0530

M0540

M0550

M0560

M0570

M0580

M0590

M0600

M0610

M0620

M0630

M0640

M0650

M0660

M0670

M0680

M0690

M0700

M0710

M0720

M0730

M0740

M0750

M0760

M0770

M0780

M0790

M0800

M0810

M0820

M0830

M0840

M0850

M0860

0010

0020

0030

0040

0010

0020

0030

0040

0050

0060

0070

0080

0090

0100

0110

0120

END
FORTRAN LST00.DECK

0130

C MMULTD

SUBROUTINE MMULTD (A,N1,B,N2,C,M,N,K,MA,MB,MC)

0010

DIMENSION A(1), B(1), C(1)

0020

IC=1

0040

IA=MC*K

0050

IB=MA*N

0060

ID=MA

0070

IH=MC

0080

IJ=MC

0090

IF (N1) 4,5,4

0100

3 IF (N2) 7,8,7

0110

4 IB=2*IB

0120

ID=2*ID

0130

IF (N2) 5,6,5

0140

5 IC=2

0150

GOTO 7

0160

6 IH=2*IH

0170

IC=3

0180

7 IA=2*IA

0190

IJ=2*IJ

0200

8 DO 18 I=1,M

0210

INC=0

0220

DO 11 J=1,IA,IH

0230

C(J)=0.0

0240

IN=INC

0250

DO 10 L=1,IB,ID

0260

IN=IN+1

0270

10 C(J)=C(J)+A(L)*R(IN)

0280

11 INC=INC+MB

0290

INC=0

0300

GOTO (18,12,15),IC

0310

12 DO 14 J=1,IA,IJ

0320

IF=I+MA

0330

IF=J+MC

0340

IN=INC

0350

DO 13 L=IF,IB,ID

0360

IN=IN+1

0370

IG=IN+MB

0380

C(IF)=C(IF)+A(L)*R(IN)

0390

13 C(J)=C(J)-A(L)*R(IG)

0400

14 INC=INC+MB

0410

GOTO 18

0420

15 IF=I+MC

0430

IF=I+MA

0440

DO 17 J=IF,IA,IJ

0450

IN=INC

0460

C(J)=0.0

0470

DO 16 L=IF,IB,ID

0480

IN=IN+1

0490

16 C(J)=C(J)+A(L)*R(IN)

0500

17 INC=INC+MB

0510

18 CONTINUE

0520

RETURN

0530

END

0540

FORTRAN LST00.DECK

C MNVRSX

SUBROUTINE MNVRSX (A,A1,R,C,KSZ,IGOOD,NOP)

0010

DIMENSION A(6,6), A1(6,6), R(6,6), C(6,6)

0020

IGOOD=0

0040

```

      IF (NOP)      10,101,102
101  CALL INVERS (AR,KSZ,IGOSFD)
      GO TO 20
102  CONTINUE
      DO 1 K=1,KSZ
      DO 1 L=1,KSZ
1    R(K,L)=AR(K,L)
      NO=0
      CALL INVERS(R,KSZ,NO)
      IF (NO) 2,3,2
C REAL MATRIX NOT SINGULAR
C MULT R*AI STO. C
3    DO 4 K=1,KSZ
      DO 4 L=1,KSZ
      C(K,L)=0.0
      DO 4 I=1,KSZ
4    C(K,L)=C(K,L)+R(K,I)*AI(I,L)
C MULT. AI*C + AR STO. R
      DO 5 K=1,KSZ
      DO 5 L=1,KSZ
      R(K,L)=AR(K,L)
      DO 5 I=1,KSZ
5    R(K,L)=R(K,L)+AI(K,I)*C(I,L)
      NO=0
      CALL INVERS(R,KSZ,NO)
      IF (NO) 2,7,2
C SECOND MATRIX NOT SINGULAR
C MULT. -C*R STO. AI ALSO SET AR=B
7    DO 8 K=1,KSZ
      DO 8 L=1,KSZ
      AI(K,L)=0.0
      AR(K,L)=B(K,L)
      DO 8 I=1,KSZ
8    AI(K,L)=AI(K,L)-C(K,I)*B(I,L)
      GO TO 20
C REAL MATRIX OR SECOND MATRIX SINGULAR TRY IMAG. ROUTE
9    DO 9 K=1,KSZ
      DO 9 L=1,KSZ
      R(K,L)=AI(K,L)
      NO=0
      CALL INVERS(R,KSZ,NO)
      IF (NO) 10,11,10
C IMAG. NOT SINGULAR
C MULT. R*AR STO. C
11   DO 12 K=1,KSZ
      DO 12 L=1,KSZ
      C(K,L)=0.0
      DO 12 I=1,KSZ
12   C(K,L)=C(K,L)+R(K,I)*AR(I,L)
C MULT. AR*C+AI STO R
      DO 13 K=1,KSZ
      DO 13 L=1,KSZ
      R(K,L)=AI(K,L)
      DO 13 I=1,KSZ
13   R(K,L)=R(K,L)+AR(K,I)*C(I,L)
      NO=0
      CALL INVERS(R,KSZ,NO)
      IF (NO) 10,15,10
C THIRD MATRIX NOT SINGULAR
C MULT -L*B STO AR ALSO SET AI=-R
15   DO 16 K=1,KSZ

```

```

0050
0060
0070
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
0210
0220
0230
0240
0250
0260
0270
0280
0290
0300
0310
0320
0330
0340
0350
0360
0370
0380
0390
0400
0410
0420
0430
0440
0450
0460
0470
0480
0490
0500
0510
0520
0530
0540
0550
0560
0570
0580
0590
0600
0610
0620
0630
0640
0650

```

```

DO 16 I=1,KSZ
AR(K,L)=0
AI(K,L)=-R(K,L)
DO 16 I=1,KSZ
16 AR(K,L)=AR(K,L)+C(K,I)*R(I,L)
GO TO 20
10 I GOOD D=1
20 RETURN
END

```

FORTRAN LISTING DECK

INVERS

SUBROUTINE INVERS (A,N,I GOOD D)

DIMENSION A(6,6), L(6), M(6)

CALL OVERFI(KUNDFX)

GO TO(500,500),KUNDFX

500 CALL OVERFL(KUNDFX)

GO TO(501,501),KUNDFX

501 CALL DVCHK (KUNDFX)

GO TO(502,502),KUNDFX

502 IGOOD=0

SEARCH FOR LARGEST ELEMENT

DO 20 K=1,N

L(K)=K

M(K)=K

RIGA=A(K,K)

DO 20 I=K,N

DO 20 J=K,N

IF(ABS(RIGA)-ABS(A(I,J)))10,20,20

10 RIGA=A(I,J)

L(K)=I

M(K)=J

20 CONTINUE

INTERCHANGE ROWS

JROW=L(K)

IF(L(K)-K)35,35,25

25 DO 30 I=1,N

HOLD=-A(K,I)

A(K,I)=A(JROW,I)

A(JROW,I)=HOLD

30 INTERCHANGE COLUMNS

35 ICOL=M(K)

IF(M(K)-K)45,45,37

37 DO 40 J=1,N

HOLD=-A(J,K)

A(J,K)=A(J,ICOL)

A(J,ICOL)=HOLD

40 DIVIDE COLUMN BY MINUS PIVOT

DO 45 IC=1,N

IF(IC-K)50,55,50

A(IC,K)=A(IC,K)/(-A(K,K))

55 CONTINUE

REDUCE MATRIX

DO 65 I=1,N

DO 65 J=1,N

56 IF(I-K)57,65,57

57 IF(J-K)60,65,60

A(I,J)=A(I,K)*A(K,J)+A(I,J)

65 CONTINUE

DIVIDE ROW BY PIVOT

DO 75 JR=1,N

68 IF(JR-K)70,75,70

0660

0670

0680

0690

0700

0710

0720

0730

0740

0810

0820

0840

0850

0860

0870

0880

0890

0900

0910

0920

0930

0940

0950

0960

0970

0980

0990

1000

1010

1020

1030

1040

1050

1060

1070

1080

1090

1100

1110

1120

1130

1140

1150

1160

1170

1180

1190

1200

1210

1220

1230

1240

1250

1260

1270

1280

1290

1300

1310

1320

1330

```

77 A(K,JR)=A(K,JR)/A(K,K)
75 CONTINUE
C CONTINUED PRODUCT OF PIVOTS
C REPLACE PIVOT BY RECIPROCAL
A(K,K)=1.0/A(K,K)
C CONTINUE COMPLETE OPERATION
80 CONTINUE
CALL DVCHK (K000FX)
GO TO(510,503),K000FX
503 CALL OVERFL(K000FX)
GO TO(510,504),K000FX
504 CALL OVERFL(K000FX)
GO TO(510,505),K000FX
C FINAL ROW AND COLUMN INTERCHANGE
505 K=N
100 K=(K-1)
IF(K)150,150,103
103 I=I(K)
IF(I-K)120,120,105
105 DO 110 J=1,N
HOLD=A(J,K)
A(J,K)=-A(J,I)
110 A(J,I)=HOLD
120 J=M(K)
IF(J-K)100,100,125
125 DO 130 I=1,N
HOLD=A(K,I)
A(K,I)=-A(J,I)
130 A(J,I)=HOLD
GO TO 100
150 RETURN
510 I GOOF D=1
GO TO 150
END

```

```

* LINK PAR2TT,PAR1TT
* FORTRAN LST00,DFCK

```

C PART2

SUBROUTINE PART2

C PART.....2 VIBRATION AND FLUTTER ANALYSIS BY A COLLOCATION METHOD.

```

DIMENSION IT(219),VELCTY(20),NSIZES(20),DM(6,6),RHO(20)
COMMON/11/ U(49,196),GUFSS(49,2),H(49,50),EIG(50),

```

```

1 TEMP(2734), NAKSR(25), NAKDR(25), NITER(75),
2 OMEGA(25), DAMP(25), VELC(25)

```

C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
C COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---

COMMON IT

EQUIVALENCE

```

1 (IT(1),ISXST), (IT(21),ISW), (IT(41),RHO),
2 (IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),
3 (IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),
4 (IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),
5 (IT(70),NSUR), (IT(71),NRIOID), (IT(72),HREF),
6 (IT(73),NAFRO), (IT(74),NFUS), (IT(75),NDENS),
7 (IT(76),MODES), (IT(77),NPOINT), (IT(78),NPUNCH),
8 (IT(79),MAXR), (IT(80),MAXQ), (IT(81),MAXS),
9 (IT(82),NC), (IT(83),NSURFS), (IT(84),Np2),
10 (IT(85),HR), (IT(106),S)

```

EQUIVALENCE

```

1 (IT(127),VELCTY), (IT(147),NTAPE0), (IT(148),NRHO),
2 (IT(149),NITRSP), (IT(150),NITRDP), (IT(151),EPSP),
3 (IT(152),FPDP), (IT(153),PICON), (IT(154),NVEL),
4 (IT(155),NCARDS), (IT(156),AITKEN), (IT(157),NGO),
5 (IT(158),FK), (IT(159),SORCON), (IT(160),NSIZES),

```

0520

0530

0540

0550

0560

0570

0580

0590

0600

0610

0620

0630

0640

0650

0660

0670

0680

0690

0700

0710

0720

0730

0740

0750

0760

0770

0780

0790

0800

0810

0820

0830

0840

0850

0010

0020

0040

0070

0080

0090

0150

0160

0170

0180

0190

0200

0210

0220

0230

0240

0250

0260

0270

0280

0290

```

6          (IT(180),DM),(IT(217),AITKE), (IT(218),KPART) 031
200 FORMAT (1H1 4X, 12H OUTPUT DATA /// 1H 11X, 5H FLUTTER 0320
1          52H ANALYSIS BY A COLLOCATION METHOD, USING AERODYNAMIC 0330
2          24H INFLUENCE COEFFICIENTS // 1H0 14X, 11H DENSITY = 0340
3          1E20.0, 5X, 20H REDUCED VELOCITY = 1E20.0, // 1H0 33X, 0350
4          116, 32H RIGID BODY DEGREES OF FREEDOM /// ) 0360

201 FORMAT (1H1 20X, 5H MODE 7X, 12H OMEGA (CPS) 10X, 6H DAMPING 0370
1          4X, 17H VELOCITY (KNOTS) /// ( 1H 2X, 114, 0380
2          3F20.8) ) 0390
202 FORMAT ( 214, 62X, 1A6, 114 ) 0400
203 FORMAT ( 1H0 33X, 23H PUNCHED CARDS NUMBERS 1A6, 114, 6H THRU 0410
1          1A6, 114 ) 0420
204 FORMAT (1H1 45X, 16H DYNAMIC MATRIX ) 0430
205 FORMAT (31H FLEXIBLE MODE SHAPES, SURFACE 114 ) 0440
206 FORMAT (5H MODE 116, 32H, GIVES AN IMAGINARY FREQUENCY. ) 0450
DATA 0000CT/0202030440005/
RCDZ =0000CT 0470
NVFL=NVEL 0480
MAXP=49 0490
K=NC*NPOINT 0500
DO 290 I=1,NPOINT 0510
DO 290 J=1,K 0520
290 U(I,J)=0.0 0530
REWIND NTAPE0 0540
DO 314 IRHO=1,NRHO 0550
MODE=MODES 0560
READ (NTAPE0)NSIZE, NSIZE2, ((U(I,J),I=1,NSIZE), 0570
J=1,NSIZE2) 0580
IF ( NPUNCH ) 302,304,304 0590
302 WRITE (NTAPE3,204) 0600
CALL MPRINT (U,NPOINT,K,MAXP,NTAPE3) 0610
304 WRITE (NTAPE3,200)RHO(IRHO), VELCTY(NVEL), NRIGID 0620
DO 305 I=1,MODE 0630
NAKDR(I)=0 0640
305 NAKSR(I)=0 0650
CALL MITERS (U,GUESS ,0 ,NPOINT,MODE ,MAXP ,NC ,EPSP , 0660
1 FPOP ,NAKSR ,NAKDR ,NITRSP,NITRDP,AITKE,AITKE, 0670
2 IR ,TEMP ,H ,EIG ,MITER ,NTAPE0,NTAPE3) 0680
IF ( NAERO ) 3055,3054,3055 0690
3054 WRITE (NTAPE4) NPOINT, MODE, ((U(I,J),I=1,NPOINT),J=1,MODE) 0700
WRITE (NTAPE4) (EIG(I),I=1,MODE),NPOINT,MODE,NPOINT,MODE,MODE 0710
3055 CONTINUE 0720
MODES2 = NC*MODE 0730
DO 310 I=1,MODES2,NC 0740
K=I/NC +NC -1 0750
IF (EIG(I)) 3051,3052,3052 0760
3051 WRITE (NTAPE3,206)K 0770
OMEGA(K)=0.0 0780
GOTO 3053 0790
3052 OMEGA(K)= SURCON / ( SORT( FK*FIR(I)) ) 0800
3053 GOTO (306,308),NC 0810
306 DAMP(K)=0.0 0820
VELC(K)=0.0 0830
GOTO 310 0840
308 DAMP(K)= FIR(I+1) / FIR(I) 0850
VELC(K)= PICON*OMEGA(K)*RREF* VELCTY(NVEL) 0860
310 CONTINUE 0870
WRITE (NTAPE3,201)(K, OMEGA(K), DAMP(K), VELC(K) , 0880
1 K=1,MODE ) 0890
IF ( NPUNCH ) 312,314,312 0900

```

```

312 K1=1
WRITE (NTAPE0,202)NSIZES(1), MODES2, BCDZ, NCARDS
NCRDS = NCARDS+1
CALL MPUNCH (OMEGA,MODES2,1,0,1,1,BCDZ,MAXP,NTAPE0,NCRDS)
WRITE (NTAPE0,203)BCDZ, NCARDS, BCDZ, NCRDS
NCARDS=NCARDS+1
DO 313 ISUR =1, NSUR
IF ( ISUR=1 ) 3121,3122,3121
3121 WRITE (NTAPE0,202)NSIZES(ISUR), MODES2,BCDZ, NCARDS
NCRDS=NCARDS+1
GO TO 3123
3122 NCRDS=NCARDS
3123 CALL MPUNCH (H(K1,1),NSIZES(ISUR),MODES2,0,1,1,BCDZ,MAXP,NTAPE0,
1 NCRDS )
WRITE (NTAPE0,205)ISUR
WRITE (NTAPE0,203)BCDZ, NCARDS, BCDZ, NCRDS
NCARDS=NCARDS+1
313 K1=K1+NSIZES(ISUR)
314 CONTINUE
REWIND NTAPE0
NGO = 98765
KPART = 1
RETURN
END

```

* TRAN LST00.DFCK

C NORM7

```

SUBROUTINE NORM (A,H,N,C,INDEX,MAXR,NC,NP)
DIMENSION A(1), H(1), C(1), I(2)
IF ( INDEX ) 100,200,200
100 GO TO (110,140),NP
110 GO TO (500,120),NC
120 RIG = C(1)**2+C(2)**2
GO TO 400
140 CALL DNORM (A,R,N,C,MAXR,NC,T)
GO TO 700
200 INDEX=1
NSTART=NP+1
NSTOP=N*NP
K=NP*MAXR
IF ( NSTART-NSTOP ) 205,205,400
205 GO TO ( 0,300 ),NC
210 RIG=ABS A(1)
DO 230 I=NSTART,NSTOP,NP
IF ( RIG-ABS(A(I)) ) 220,220,230
220 INDEX=I
RIG=ABS(A(I))
230 CONTINUE
GO TO 400
300 RIG = A(1)**2 + A(K+1)**2
DO 310 I=NSTART,NSTOP,NP
J=K+1
IF ( RIG-(A(I)**2+A(J)**2) ) 320,320,330
320 RIG = A(I)**2 + A(J)**2
INDEX = I
330 CONTINUE
400 J=NP*NC
I=INDEX+(NC-1)*K+(NP-1)
C(J)=A(I)
J=J-(NP-1)
I=I-(NP-1)
C(J)=A(I)

```

0910
0920
0930
0940
0950
0960
0970
0980
0990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140

MTR40002
MTR40004
MTR40006
MTR40007
MTR40008
MTR40009
MTR40010
MTR40012
MTR40013
MTR40015
MTR40016
MTR40017
MTR40018
MTR40019
MTR40020
MTR40022
MTR40023
MTR40025
MTR40026
MTR40027
MTR40028
MTR40029
MTR40031
MTR40032
MTR40033
MTR40034
MTR40035
MTR40036
MTR40038
MTR40040
MTR40041
MTR40042
MTR40043
MTR40044
MTR40045

```

      I=INDEX*(NP-1)
      C(NP)=A(I)
      C(1)=A(INDEX)
      GO TO (410,140),NP
410  GO TO (500,600),NC
500  DO 510 I=1,N
510  R(I)=A(I)/C(1)
      GO TO 700
600  BIG = C(1)**2+C(2)**2
      DO 610 I=1,N
          J=I+MAXR
          T= A(I)*C(1)+A(J)*C(2)
          R(J) = ( A(J)*C(1)-A(I)*C(2) ) / BIG
610  R(I)= T/BIG
700  INDEX = INDEX/NP +NP-1
      RETURN
      END

```

MTR40046
 MTR40047
 MTR40048
 MTR40050
 MTR40051
 MTR40053
 MTR40054
 MTR40055
 MTR40057
 MTR40058
 MTR40060
 MTR40061
 MTR40062
 MTR40063
 MTR40064
 MTR40065
 MTR40067

```

      FORTRAN LISTING DECK

```

```

C DNORMZ

```

```

      SUBROUTINE DNORM (A,B,N,C,MAXR,NC,T)
      DOUBLE PRECISION A(1), B(1),C(1),T(1)
410  GO TO (500,600),NC
500  DO 510 I=1,N
510  R(I)=A(I)/C(1)
      GO TO 700
600  BIG=C(1)*C(1) + C(2)*C(2)
      DO 610 I=1,N
          J=I+MAXR
          T= A(I)*C(1)+A(J)*C(2)
          R(J) = ( A(J)*C(1)-A(I)*C(2) ) / BIG
610  R(I)= T/BIG
700  RETURN
      END

```

MTR40069
 MTR40071
 MTR40072
 MTR40074
 MTR40075
 MTR40076
 MTR40078
 MTR40079
 MTR40080
 MTR40081
 MTR40082
 MTR40083
 MTR40084
 MTR40086

```

      FORTRAN LISTING DECK

```

```

C PQHS

```

```

      SUBROUTINE PQH (LMBDN,LMBD1,LMBD2,HN,HN1,H1,H2,N,NC)
      DIMENSION LMBDN(1), LMBD1(1), LMBD2(1), HN(1), HN1(1), H1(1),
1      H2(1), A(2)
      DOUBLE PRECISION LMBDN, LMBD1, LMBD2, HN, HN1, H1, H2, A
      GO TO (200,100),NC
100  I=1,N
      K=I+N
      A(1) = LMBDN(1)*HN(1)-LMBDN(2)*HN(K)
      A(2) = LMBDN(1)*HN(K)+LMBDN(2)*HN(1)
      H1(1) = LMBD2(1)*HN1(1)-LMBD2(2)*HN1(K)-A(1)
      H1(K) = LMBD2(1)*HN1(K)+LMBD2(2)*HN1(1)-A(2)
      H2(1) = A(1)-LMBD1(1)*HN1(1)+LMBD1(2)*HN1(K)
110  H2(K) = A(2)-LMBD1(1)*HN1(K)-LMBD1(2)*HN1(1)
      RETURN
200  DO 210 I=1,N
      A(1)= LMBDN(1)*HN(1)
      H1(1) = LMBD2(1)*HN1(1)-A(1)
210  H2(1) = A(1)-LMBD1(1)*HN1(1)
      RETURN
      END

```

MTR40089
 MTR40091
 MTR40092
 MTR40093
 MTR40095
 MTR40097
 MTR40099
 MTR40101
 MTR40102
 MTR40104
 MTR40105
 MTR40107
 MTR40108
 MTR40110
 MTR40112
 MTR40114
 MTR40116
 MTR40117
 MTR40119
 MTR40121

```

      FORTRAN LISTING DECK

```

```

C AITKNS

```

```

      SUBROUTINE AITKNS (HN,HN1,HN2,HNEW,R,N,MAXR,NC,NP,IR)
      DIMENSION HN(1), HN1(1), HN2(1), HNEW(1), A(2), B(4), C(2), D(2)
      IR=0
      I=NP*N

```

MTR40123
 MTR40125
 MTR40127
 MTR40129

```

C(1) = R**2
DO 110 I=1,I.NP
  D(1) = 0
  D(2) = 0
  DO 100 J=1,NC
    K=(J-1)*MAXR+1
    D(1) = D(1) + (HN1(K)-HN2(K))*2
100   D(2) = D(2) + (HN(K)-HN1(K))*2
    IF ( D(1) ) 105,110,105
105   CONTINUE
    IF (D(2)/D(1) - C(1)) 110,110,800
110   CONTINUE
GO TO (300,200),NP
200 CALL AITKND (HN,HN1,HN2,HNEW,N,MAXR,NC,A,B,C,D)
GO TO 700
300 DO 600 I=1,N
  GO TO (400,500),NC
400 C(1) = HN(1)-2.*HN1(1)+HN2(1)
  IF ( C(1) ) 410,600,410
410 HNEW(1) = HN2(1) - ( (HN1(1)-HN2(1))*2 /C(1) )
  GO TO 600
500 A(1) = 0.
  C(1) = 2.
  D(1) = 0.
  DO 510 J=1,2
    K = 1+(J-1)*MAXR
    R(J) = HN(K)-2.*HN1(K)+HN2(K)
    C = C*(HN1(K)-HN2(K))
    A = -(HN1(K) - HN2(K))*2 - A
510   D = R(J)**2 + D
    IF ( D ) 520,600,520
520 HNEW(1) = HN2(1) - (R(1)*A(1)+R(2)*C(1))/ D(1)
    HNEW(K) = HN2(K) - (R(1)*C(1)-R(2)*A(1))/ D(1)
600 CONTINUE
710 IP = 1
900 RETURN
END

```

* FORTRAN LISTING DECK

C AITKND

```

SUBROUTINE AITKND (HN,HN1,HN2,HNEW,N,MAXR,NC,A,B,C,D)
DIMENSION HN(1), HN1(1), HN2(1),HNEW(1), A(1), B(2) C(1), D(1)
DOUBLE PRECISION HN, HN1, HN2, HNEW, A, H, C, D
300 DO 600 I=1,N
  GO TO (400,500),NC
400 C(1) = HN(1)-2.*HN1(1)+HN2(1)
  IF ( ABS(C(1)) - .0000000000 ) 600,600,410
410 HNEW(1)=HN2(1) - ( (HN1(1)-HN2(1))*(HN1(1)-HN2(1)) /C(1) )
  GO TO 600
500 A(1) = 0.
  C(1) = 2.
  D(1) = 0.
  DO 510 J=1,2
    K = 1+(J-1)*MAXR
    R(J) = HN(K)-2.*HN1(K)+HN2(K)
    C = C*(HN1(K)-HN2(K))
    A = -( (HN1(K)-HN2(K))*(HN1(K)-HN2(K)) ) - A
510   D = R(J)*R(J) + D
    IF ( D ) 520,600,520
520 HNEW(1) = HN2(1) - (R(1)*A(1)+R(2)*C(1))/ D(1)
    HNEW(K) = HN2(K) - (R(1)*C(1)-R(2)*A(1))/ D(1)
600 CONTINUE

```

MTR40130
MTR40131
MTR40133
MTR40134
MTR40136
MTR40138
MTR40140
MTR40141
MTR40143
MTR40144
MTR40145
MTR40146
MTR40148
MTR40149
MTR40150
MTR40152
MTR40154
MTR40155
MTR40156
MTR40158
MTR40159
MTR40161
MTR40162
MTR40163
MTR40164
MTR40166
MTR40167
MTR40168
MTR40169
MTR40170
MTR40172
MTR40173
MTR40174
MTR40176
MTR40178
MTR40180
MTR40182

MTR40184
MTR40186
MTR40187
MTR40189
MTR40191
MTR40192
MTR40193
MTR40195
MTR40196
MTR40198
MTR40199
MTR40200
MTR40201
MTR40203
MTR40204
MTR40205
MTR40206
MTR40207
MTR40209
MTR40210
MTR40211
MTR40212

```

700 IR = 1
800 RETURN
END

```

MTR40215
MTR40217
MTR40219

```

C LEGISS
FORTRAN LSTOU.DECK

```

```

SURROUTINE LEGIS (LAMRDA,POS,E1,E2,NC,IR)
DIMENSION LAMRDA(1), POS(1)
DOUBLE PRECISION LAMRDA, POS, R1, R2A, R2B
IR=0
GO TO (100,200),NC
100 R1 = ( DABS( LAMRDA(1)-LAMRDA(2) ) ) / E1
R2A = DABS( DABS(POS(2)/POS(1))-1. ) / E2
R2A = DABS( DABS(POS(4)/POS(1))-1. ) / E2
CALL MPRINT (R1,2,1,2,6)
CALL MPRINT (R2A,2,1,2,6)
CALL MPRINT (R2B,2,1,2,6)

```

MTR40222
MTR40224
MTR40225
MTR40227
MTR40229
MTR40231
MTR40232
MTR40233
MTR40234
MTR40235
MTR40236

```

110 IF ( R1-R2A ) 120,140,140
120 IF ( R1-R2B ) 130,140,140
130 IR=NC
140 RETURN

```

MTR40238
MTR40240
MTR40242
MTR40244

```

200 R1 = DSORT( (LAMRDA(1)-LAMRDA(3))*(LAMRDA(1)-LAMRDA(3)) + (LAMRDA
1 (2)-LAMRDA(4))*(LAMRDA(2)-LAMRDA(4)) ) / E1
R2A = DABS( DSORT( ((POS(3)*POS(1)+POS(2)*POS(4))*(POS(3)+POS(1) +
1 POS(2)+POS(4)) + (POS(3)+POS(2)-POS(4)*POS(1))*(POS(3)+
2 POS(2)-POS(4)*POS(1)) ) / (POS(1)+POS(1)+POS(2)+POS(2)) )
3 -1. ) / E2
R2B = DABS( DSORT( ( (POS(7)*POS(5)+POS(6)*POS(8)) / (POS(5)+POS
1 (5) + POS(6)*POS(6)) )*( (POS(7)+POS(5)+POS(6)+POS(8)) /
2 (POS(5)+POS(5) + POS(6)+POS(6)) ) + ( (POS(7)+POS(6) - POS
3 (7)+POS(5)) / (POS(5)+POS(5) + POS(6)+POS(6)) ) * ( (POS(7)
4 +POS(6) - POS(8)+POS(5)) / (POS(5)+POS(5) + POS(6)+POS(6))
5 ) ) -1. ) / E2

```

MTR40245
MTR40247
MTR40248
MTR40249
MTR40250
MTR40251
MTR40253
MTR40254
MTR40255
MTR40256
MTR40257
MTR40258

```

GO TO 110
END

```

MTR40260
MTR40262

```

C MAD7
FORTRAN LSTOU.DECK

```

```

SURROUTINE MAD7 (A,B,C,NSIZE,NC)
DOUBLE PRECISION A(1), B(1), C(1)
K=NC*NSIZE
DO 100 I=1,K
100 C(I)=A(I)-B(I)
RETURN
END

```

MTR40264
MTR40265
MTR40267
MTR40269
MTR40271
MTR40272
MTR40273

```

C MULT5
FORTRAN LSTOU.DECK

```

```

SURROUTINE MULT (A,R,C,LIZ,NIZ,MIZ,MAXA,MAXR,MAXC,NC,NP)
DIMENSION A(1), R(1), C(1)
KA=NC*MAXA
KR=NC*MAXR
KC=NC*MAXC
GO TO (200,100),NP
100 CALL DMULT (A,R,C,LIZ,NIZ,MIZ,KA,KR,KC,NC)
GO TO 700
200 DO 600 I=1,IIZ
DO 500 M=1,MIZ
K = (M-1)*KR + 1
J = (I-1)*KA + 1
C(I)=0.
GO TO (300,400),NC
300 DO 310 N=1,NIZ
J=(N-1)*KA+1

```

MTR40275
MTR40277
MTR40279
MTR40280
MTR40281
MTR40283
MTR40285
MTR40286
MTR40288
MTR40290
MTR40292
MTR40293
MTR40295
MTR40297
MTR40299
MTR40301

310	JB = (N-1)*K C(I)=C(I)+A(J)*B(JR)	MTR40302 MTR40303
400	GO TO 500 IC = I+MAXC C(IC) = 0. DO 410 N=1,NIZ J= (N-1)*KA+L JR= (N-1) +K JC= J+MAXA JRC= JR+MAXH C(I)=C(I)+A(J)*H(JR)-A(JC)*H(JRC) 410 C(IC)=C(IC)+A(J)*H(JRC)+A(JC)*H(JR) 500 CONTINUE 600 CONTINUE 700 RETURN END	MTR40304 MTR40306 MTR40307 MTR40309 MTR40311 MTR40312 MTR40313 MTR40314 MTR40316 MTR40317 MTR40319 MTR40321 MTR40323 MTR40325
S FORTRAN LISTING DECK		
C DMULTS		
	SUBROUTINE DMULT (A,R,C,LIZ,NIZ,MIZ,KA,KB,KC,NC) DOUBLE PRECISION A(1), R(1), C(1) MAXA=KA/2 MAXR=KB/2	MTR40327 MTR40329 MTR40330 MTR40331
200	DO 600 L=1,LIZ DO 500 M=1,MIZ K= (M-1)*KR +1 I= (M-1)*KC +L C(I)=0. GO TO (300,400),NC	MTR40332 MTR40334 MTR40336 MTR40337 MTR40339 MTR40341
300	DO 310 N=1,NIZ J=(N-1)*KA+L JR = (N-1)+K	MTR40343 MTR40345 MTR40346
310	C(I)=C(I)+A(J)*R(JR) GO TO 500	MTR40347 MTR40348
400	IC=I+KC/2 C(IC) = 0. DO 410 N=1,NIZ J= (N-1)*KA+L JR= (N-1) +K JC= J+MAXA JRC= JR+MAXH C(I)=C(I)+A(J)*H(JR)-A(JC)*H(JRC) 410 C(IC)=C(IC)+A(J)*B(JRC)+A(JC)*R(JR) 500 CONTINUE 600 CONTINUE 700 RETURN END	MTR40350 MTR40351 MTR40353 MTR40355 MTR40356 MTR40357 MTR40358 MTR40360 MTR40361 MTR40363 MTR40365 MTR40367 MTR40369
S FORTRAN LISTING DECK		
C POLMS		
	SUBROUTINE POLM (PN,PN1,ON,ON1,E2,LMRD1,LMRD2,NC,IR,IG0) DIMENSION PN(1), PN1(1), ON(1), ON1(1), E2(1), LMRD1(1), LMRD2(1) DOUBLE PRECISION PN, PN1, ON, ON1, LMRD1, LMRD2, A(2) IR=0 DO 10 (100,100,100),IG0 100 DO 10 (110,200),NC 110 IF (DAHS((PN-PN1)/(DSORT(DAHS(ON)))) -E2) 12,120,112 112 IF (DAHS(PN-PN1) - E2**2) 120,120,170 120 IF (DAHS((ON/ON1)-1.) - E2) 13,130,170 130 LMRD2 = (PN*PN -4.*ON) LMRD2 = DSORT (DAHS(LMRD2)) LMRD1 =(-PN + LMRD2) /2. LMRD2 =(-PN - LMRD2) /2.	MTR40372 MTR40374 MTR40375 MTR40376 MTR40377 MTR40379 MTR40381 MTR40382 MTR40384 MTR40386 MTR40387 MTR40388 MTR40389

IF (ARS(LMRD1) - ARS(LMRD2))	140,150,160	MTR40391
140 A=LMRD1		MTR40392
LMRD1=LMRD2		MTR40393
LMRD2=A		MTR40394
GO TO 160		MTR40395
150 IF (LMRD1)	140,160,160	MTR40396
160 IR=NC		MTR40398
170 RETURN		MTR40400
180 GO TO (130,220),NC		MTR40401
200 A=DSORT(QN(1)*QN(1) + QN(2)*QN(2))		MTR40403
IF (DSORT(((PN-PN1)*(PN-PN1) + (PN(2)-PN1(2))*(PN(2)-PN1(2)))		MTR40405
1 / A) -E2)	210,210,170	MTR40406
210 A = QN1*QN1 + QN1(2)*QN1(2)		MTR40407
IF (((QN*QN1-QN(2)*QN1(2))/A)*((QN*QN1-QN(2)*QN1(2))/A) +		MTR40409
1 (QN*QN1(2)+QN1*QN(2))/A)*((QN*QN1(2)+QN1*QN(2))/A)		MTR40410
2 -1.) - E2)	220,220,170	MTR40411
220 A(1) = PN*PN - PN(2)*PN(2) -4.*QN		MTR40412
A(2) = 2.*(PN*PN(2)-2.*QN(2))		MTR40413
IF (A(1))	230,230,230	MTR40415
230 A(1) = DSORT((-A(1)+DSORT(A(1)*A(1)+A(2)*A(2))) /2.)		MTR40417
A(2) = A(2) / (2.*A(1))		MTR40418
232 LMRD1(1) = (-PN+A(2))/2.		MTR40420
LMRD1(2) = (-PN(2)+A(1))/2.		MTR40421
LMRD2(1) = (-PN-A(2))/2.		MTR40422
LMRD2(2) = (-PN(2)-A(1))/2.		MTR40423
IF (DSORT(LMRD1*LMRD1+LMRD1(2)*LMRD1(2)) - DSORT(LMRD2*LMRD2		MTR40425
1 LMRD2(2)*LMRD2(2)))	240,160,160	MTR40426
240 A=LMRD1(2)		MTR40427
LMRD1(2)=LMRD2(2)		MTR40428
LMRD2(2)= A		MTR40429
GO TO 140		MTR40430
250 A(1) = DSORT((A(1)+DSORT(A(1)*A(1)+A(2)*A(2))) /2.)		MTR40432
A(2) = A(2) / (2.*A(1))		MTR40433
LMRD1=A(1)		MTR40434
A(1)=A(2)		MTR40435
A(2)=LMRD1		MTR40436
GO TO 232		MTR40437
END		MTR40439
FORTRAN LISTING DECK		
POS		
SUBROUTINE PD (FL, HN, HN1, HN2, HN3, P, Q, NC, MAXR)		MTR40441
DIMENSION FL(1), HN(1), HN1(1), HN2(1), HN3(1), P(1), Q(1),		MTR40442
1 A(3), R(3)		MTR40443
DOUBLE PRECISION FL, HN, HN1, HN2, HN3, P, Q, A, R		MTR40445
MAXR = MAXR/2		MTR40447
GO TO (200,100), NC		MTR40448
100 A(1) = FL(1)*HN - FL(2)*HN(MAXR+1)		MTR40450
A(2) = FL(3)*HN1 - FL(4)*HN1(MAXR+1)		MTR40451
A(3) = FL(5)*HN2 - FL(6)*HN2(MAXR+1)		MTR40452
R(1) = FL(1)*HN(MAXR+1)+FL(2)*HN		MTR40454
R(2) = FL(3)*HN1(MAXR+1)+FL(4)*HN1		MTR40455
R(3) = FL(5)*HN2(MAXR+1)+FL(6)*HN2		MTR40456
P(1) = A(3)*HN1 - R(3)*HN1(MAXR+1)-A(1)*HN3 + B(1)*HN3(MAXR+1)		MTR40458
P(2) = A(3)*HN1(MAXR+1)+R(3)*HN1 - A(1)*HN3(MAXR+1)-R(1)*HN3		MTR40459
Q(1) = A(1)*HN2 - R(1)*HN2(MAXR+1)-A(2)*HN1 + B(2)*HN1(MAXR+1)		MTR40461
Q(2) = A(1)*HN2(MAXR+1)+R(1)*HN2 - A(2)*HN1(MAXR+1)-Q(2)*HN1		MTR40462
A(1) = A(2)*HN1 - R(2)*HN3(MAXR+1)-A(3)*HN2 + B(3)*HN2(MAXR+1)		MTR40464
R(1) = A(2)*HN1(MAXR+1)+R(2)*HN3 - A(3)*HN2(MAXR+1)-R(3)*HN2		MTR40465
A(2) = A(1)*P(1) + R(1)*P(2)		MTR40467
R(2) = A(1)*P(2) - R(1)*P(1)		MTR40468
A(3) = A(1)*Q(1) + R(1)*Q(2)		MTR40469

```

R(3) = A(1)*Q(2) - B(1)*Q(1)
A(1) = A(1)*A(1) + R(1)*R(1)
A(2) = A(2)/A(1)
R(2) = R(2)/A(1)
A(3) = A(3)/A(1)
R(3) = R(3)/A(1)
P(1) = FL(3)*A(2) - FL(4)*R(2)
P(2) = FL(3)*R(2) + FL(4)*A(2)
A(2) = FL(3)*FL(5) - FL(4)*FL(6)
R(2) = FL(4)*FL(5) + FL(3)*FL(6)
Q(1) = A(2)*A(3) - R(2)*R(3)
Q(2) = A(2)*R(3) + A(3)*R(2)
MAXR = 2*MAXR
RETURN
200 A(1) = (FL(2)*HN1*HN3 - FL(3)*HN2*HN2)
P(1) = (FL(3)*HN2(1)*HN1(1) - FL(1)*HN(1)*HN3(1)) / A(1)
Q(1) = (FL(1)*HN(1)*HN2(1) - FL(2)*HN1(1)*HN1(1)) / A(1)
P(1) = P(1)*FL(2)
Q(1) = Q(1)*FL(2)*FL(3)
MAXR = 2*MAXR
RETURN
END

```

MTR40470
MTR40471
MTR40472
MTR40473
MTR40474
MTR40475
MTR40476
MTR40477
MTR40478
MTR40479
MTR40480
MTR40481
MTR40482
MTR40483
MTR40484
MTR40485
MTR40486
MTR40487
MTR40488
MTR40489
MTR40490
MTR40491
MTR40492
MTR40493
MTR40494
MTR40495
MTR40496
MTR40497
MTR40498
MTR40499
MTR40500

§ FORTRAN LSTOU.DECK

§ CLOSES

§ SUBROUTINE CLOSE. COMPUTES 2 CLOSE ROOTS.

```

C     U = MATRIX, DIMENSIONED (MAXR,2*NC*MAXR)
C     H = STARTING GUESS, DIMENSIONED((MAXR,2*NC*4)+2*NC*N)
C     NSIZE = SIZE OF MATRIX
C     MAXR = DIMENSIONED NUMBER OF ROWS OF U AND H
C     MAXTRY = MAXIMUM NUMBER OF DOUBLE PRECISION ITERATIONS.
C     EPS1 = SINGLE ROOT CONVERGENCE CRITERIA
C     EPS2 = DOUBLE ROOT CONVERGENCE CRITERIA
C     R = AITKENS CONVERGENCE CRITERIA
C     IRR = ERROR RETURN INDICATOR.     =1, OVERFLOW
C                                         =2, DIVIDE CHECK
C                                         =3, BOTH OVERFLOW AND DIVIDE
C                                             CHECK.
C     ITERS= NUMBER OF ITERATIONS PERFORMED, - FOR DOUBLE ROOT
C                                         + FOR SINGLE ROOT
C     NC = 1, REAL             2, COMPLEX
C     SUBROUTINE CLOSES (U,H,NSIZE,MAXR,R,EPS1,EPS2,NC,IRR,MAXTRY,ITERS,
1     NAITKN,INDEX1,INDEX2,VALUE,MSIZE)
C     DIMENSION U(1), H(1), VALUE(1)
C     CALL OVERFL ( OVERFL )
C     CALL DVCHK ( INDVCT )
C     IRR=0
C     NX=2*NSIZE
C     N2C=2*NC
C     CALL CHANGE (U,MSIZE,NC*MSIZE,MAXR,1)
C     CALL CHANGE (H,MSIZE,NC,NSIZE,1)
C     I6=N2C*NSIZE
C     I1=1
C     I2=I1+I6
C     I3=I2+I6
C     I4=I3+I6
C     I5=I4+I6
C     K1=I1
C     K4=I2
C     K3=I3
C     K2=I4

```

MTR40501
MTR40502
MTR40503
MTR40504
MTR40505
MTR40506
MTR40507
MTR40508
MTR40509
MTR40510
MTR40511
MTR40512
MTR40513
MTR40514
MTR40515
MTR40516
MTR40517
MTR40518
MTR40519
MTR40520
MTR40521
MTR40522
MTR40523
MTR40524
MTR40525
MTR40526
MTR40527
MTR40528
MTR40529
MTR40530
MTR40531
MTR40532
MTR40533
MTR40534
MTR40535
MTR40536
MTR40537
MTR40538
MTR40539
MTR40540
MTR40541
MTR40542
MTR40543
MTR40544

ITERS = 0		MTR40545
100 K=0		MTR40547
100 = 1		MTR40548
120 ITERS=ITERS+1		MTR40549
IF (ITERS-MAXTRY)	140,140,130	MTR40550
130 IF (K=0)	132,140,132	MTR40551
132 ITERS = ITERS-1		MTR40552
GO TO 800		MTR40553
140 K=K+1		MTR40555
I=K1		MTR40556
K1=K4		MTR40557
K4=K3		MTR40558
K3=K2		MTR40559
K2=1		MTR40560
CALL MULT (I, H(K2), H(K1), NSIZE, NSIZE, 1, MAXR, NSIZE, NSIZE,		MTR40562
1 NC, 2)		MTR40563
INDEX=0		MTR40564
IK=15+(4-K)*N2C		MTR40565
CALL NORM (H(K1), H(K1), NSIZE, H(IK), INDEX, NSIZE, NC, 2)		MTR40566
CALL OVERFL (IOVFLW)		MTR40568
GOTO (150,140) , IOVFLW		MTR40569
150 IRR=IRR+1		MTR40570
160 CALL DVCHK (IDVDCT)		MTR40571
GOTO (170,180) , IDVDCT		MTR40572
170 IRR=IRR+2		MTR40573
180 IF (IRR)	200,200,640	MTR40574
C TEST FOR CONVERGENCE TO A SINGLE ROOT		MTR40576
200 DO 220 I=1,16,2		MTR40578
J2 = K2+I-1		MTR40579
J3 = K1+I-1		MTR40580
IF (ABS(H(J2)-H(J3))-EPS1)	220,220,300	MTR40581
220 CONTINUE		MTR40582
GOTO 750		MTR40583
300 GOTO (120,120,120,320), K		MTR40585
320 J1=15+(160+J)*N2C		MTR40587
J2=15+(160+5)*N2C		MTR40588
J=2*INDEX-2		MTR40590
J3=K1+J		MTR40591
J5=K2+J		MTR40592
J7=K3+J		MTR40593
J9=K4+J		MTR40594
C COMPUTE P N AND Q N.		MTR40596
CALL PD (H(I5), H(J3), H(J5), H(J7), H(J9), H(J1), H(J2), NC, NX)		MTR40597
GOTO (350,340), I60		MTR40600
340 J1= 15+4*N2C		MTR40601
J2= J1+N2C		MTR40602
J3= J2+N2C		MTR40603
J4= J3+N2C		MTR40604
J5= 15+N2C		MTR40605
J6= J5+N2C		MTR40606
C TEST FOR DOUBLE ROOT CONVERGENCE AND IF SO, COMPUTE LAMBDA 1 AND 2.		MTR40608
CALL POLM (H(J2), H(J1), H(J4), H(J3), EPS2, H(J5), H(J6), NC, IR, I60)		MTR40610
GOTO (344,344,400), I60		MTR40611
344 IF (IR)	346,346,400	MTR40612
346 IF (ITERS-MAXTRY)	347,800,800	MTR40613
347 DO 348 I=1,N2C		MTR40614
H(J1)=H(J2)		MTR40615
H(J3)=H(J4)		MTR40616
J1=J1+1		MTR40617
J2=J2+1		MTR40618
J3=J3+1		MTR40619

348	J4=J4+1		MTR40620
350	K=3		MTR40622
	IG0=2		MTR40623
	DO 354 J=1.3		MTR40624
	L=15+(3-J)*N2C		MTR40625
	L1=15+(4-J)*N2C		MTR40626
	DO 354 I=1,NC		MTR40627
	L=L+(I-1)*NC		MTR40628
	L1=L1+(I-1)*NC		MTR40629
	H(L1)=H(L)		MTR40630
354	H(L1+1)=H(L+1)		MTR40631
	CALL AITKNS (H(K1), H(K2), H(K3), H(K4), R, VALUE, NSIZE, NC, 2,		MTR40632
	1 IR)		MTR40633
	IF (IR)	360,120,360	MTR40634
360	I=K1		MTR40635
	K1=K4		MTR40636
	K4=K3		MTR40637
	K3=K2		MTR40638
	K2=1		MTR40639
	NAITKN = NAITKN+1		MTR40640
	GOTO 100		MTR40641
400	CALL POH (H(J5), H(J6), H(K1), H(K2), H(K3), H(K4), NSIZE,		MTR40642
	1 NC)		MTR40643
	GOTO (404,402),NC		MTR40644
402	VALUE(2)=H(J5+2)		MTR40645
	VALUE(4)=H(J6+2)		MTR40646
404	INDEX=0		MTR40647
	VALUE(1)=H(J5)		MTR40648
	VALUE(NC+1)=H(J6)		MTR40649
	CALL NORM (H(K3),H(K3),NSIZE,H(J1),INDEX,NSIZE,NC,2)		MTR40650
	INDEX1 = INDEX		MTR40651
	ITERS = -ITERS		MTR40652
	I=K4+2*INDEX-2		MTR40653
	H(J1)=H(I)		MTR40654
	H(J1+1)=H(I+1)		MTR40655
	GOTO (420,410),NC		MTR40656
410	I = I+NX		MTR40657
	H(J1+2)=H(I)		MTR40658
	H(J1+3)=H(I+1)		MTR40659
420	INDEX=-INDEX		MTR40660
	CALL NORM (H(K4),H(K4),NSIZE,H(J1),INDEX,NSIZE,NC,2)		MTR40661
	IF (K - 2)	440,500,480	MTR40662
440	J3=13		MTR40663
442	J4=14		MTR40664
450	K=1		MTR40665
460	DO 462 J=1.16		MTR40666
	J1=J3+J-1		MTR40667
	J2=J4+J-1		MTR40668
462	H(J1)=H(J2)		MTR40669
	GOTO (520,510,514,600),K		MTR40670
470	J3=11		MTR40671
	GOTO 442		MTR40672
480	IF (K3-13)		MTR40673
490	K=3	490,490,470	MTR40674
	J3=11		MTR40675
	J4=13		MTR40676
	GOTO 460		MTR40677
500	K=2		MTR40678
	J3=13		MTR40679
	J4=11		MTR40680
	GOTO 460		MTR40681
			MTR40682
			MTR40683
			MTR40684
			MTR40685
			MTR40686
			MTR40687

510 J3=11		MTR40688
512 J4=12		MTR40689
514 J3=13		MTR40690
516 J3=13		MTR40691
518 J3=13		MTR40692
520 CALL MANS (H(11),H(13),H(12),NSIZE,NC)		MTR40693
600 CALL CHANGE (H(11),NSIZE,2*NC,NSIZE,2)		MTR40695
604 J=NC*NSIZE+1		MTR40697
INDEX=0		MTR40698
CALL NORM (H(J),H(J),NSIZE,H(15),INDEX,NSIZE,NC,1)		MTR40700
J=J+INDEX-1		MTR40701
J1=J+(NC-1)*NSIZE		MTR40702
H(J)=0.		MTR40703
H(J1)=0.		MTR40704
INDEX2 = INDEX		MTR40705
606 CALL OVERFL (IOVFLW)		MTR40707
GOTO (610,620) ,IOVFLW		MTR40708
610 IRR=IRR+1		MTR40709
620 CALL DVCHK (IDVDCT)		MTR40711
GOTO (630,640) ,IDVDCT		MTR40712
630 IRR=IRR+2		MTR40713
640 CALL CHANGE (0,NSIZE,NC*NSIZE,MAXR,2)		MTR40715
700 RETURN		MTR40717
750 IK=15+(4-K)*N2C		MTR40719
VALUE(1) = H(IK)		MTR40720
IK=IK+2*(NC-1)		MTR40721
VALUE(NC)=H(IK)		MTR40722
INDEX1=INDEX		MTR40723
K=4		MTR40724
J3=11		MTR40725
J4=K1		MTR40726
IF (K1-11)	460,600,460	MTR40727
800 IGO=3		MTR40729
INDEX1=INDEX		MTR40730
J=15		MTR40731
J1 = 15+4*N2C		MTR40732
806 CALL LEGIS (H(J),H(J1),FPS1,EPS2,NC,IR)		MTR40733
IF (IR)	340,340,750	MTR40734
END		MTR40736
* FORTRAN LISTING DECK		
C MITERS		
C A IS STORED IN CORE AT A. (MAXR X NC*NP*NSIZE)		MTR40738
C NTAPUT IS A UTILITY TAPE, FOR CHECK VECTORS IF DESIRED.		MTR40739
C EPSP = EPSILON ONE = SINGLE PRECISION CONVERGENCE TEST NUMBER		MTR40740
C EPDP = EPSILON TWO = DOUBLE PRECISION		MTR40741
C NC = 1, IF REAL NP = 1, IF SINGLE PRECISION		MTR40742
C 2, IF COMPLEX = 2, IF DOUBLE ..		MTR40743
C NGUESS = 0, IF FIRST GUESS IS TO BE A COLUMN OF ONES.		MTR40744
C MODOUT = NO. OF MODES TO BE COMPUTED.		MTR40745
C NAKSR - NO. TIMES AITKENS ACCELERATION WAS USED IN SINGLE PRECISION.		MTR40746
C NAKDR = ... DOUBL ..		MTR40747
C MAXSR = MAXIMUM ITERATIONS ALLOWED IN SINGLE PRECISION.		MTR40748
C MAXDR = ... DOUBLE ..		MTR40749
C IRR = ERROR RETURN = 1, FOR OVERFLOW		MTR40750
C 2, FOR DIVIDE CHECK		MTR40751
C 3, FOR BOTH OVERFLOW AND DIVIDE CHECK		MTR40752
C NSIZE = NO. OF ROWS AND COLUMNS OF A		MTR40753
C RSP = R, AITKENS ACCELERATION CONVERGENCE CONTROL FOR SINGLE PRECIS.		MTR40754
C RDP = R AITKENS ACCELERATION CONVERGENCE CONTROL FOR DOUBLE PRECIS.		MTR40755
C MAXR = DIMENSIONED NUMBER OF ROWS OF A AND GUESS		MTR40756
C SUBROUTINE MITERS(A,GUESS,NGUESS,NSIZE,MODOUT,MAXR,NC,FPS1,EPDP,		MTR40758

[illegible]

```

NDRAK=0
ITERSR=0
ITERDR=0
K1=11
K2=13
K3=12
DO 150 K=1,NC
  J1=(K-1)*NSIZE
  J=(K-1)*MSIZE+K1-1
  DO 150 I=1,MSIZE
    J=J+1
    J1=J1+1
150   H(J)=GUFSS(J1)
152 NAK=0
150 NAK=NAK+1
152 ITERSR=ITERSR+1
    IF ( ITERSR-MAXSR )          1/0,1/0,250
1/0 I=K1
    K1=K3
    K3=K2
    K2=I
C   SET .... NOW MAKE ONE ITERATION.
C
  CALL MULT (A,H(K2),H(K1),MSIZE*MSIZE,1,MAXR,MSIZE,MSIZE,NC,1)
  INDEX=0
  IK= 1+KSIZE+NC*(1-NAK)
  CALL NORM (H(K1),H(K1),MSIZE,H(IK),INDEX,MSIZE,NC,1)
  CALL OVERFL ( IOVFLW )
  CALL DVCHK ( IDVCT )
  GOTO (180,182) ,IOVFLW
180 IRR=IRR+1
182 GOTO (184,186) ,IDVCT
184 IRR=IRR+2
186 IF ( IRR )          190,190,600
190 GOTO (160,200,200),NAK
200 DO 210 I=1,KSIZE
    J1=K1+I-1
    J2=K2+I-1
    IF ( ABS(H(J1))-H(J2))-EPSP )          210,210,220
210 CONTINUE
    GOTO 400
220 GOTO (160,160,230),NAK
230 CALL AITKNS ( H(K1), H(K2), H(K3), H(K3), RSP, MSIZE, MSIZE, NC,
  1          1, IR )          240,232,240
    IF ( IR )
232 I=2*NC
    J=IK+1
    DO 234 I=1,I,NC
      J1=J+NC-1
      J2=J-1+1
      J3=J1-NC
      J4=J2-NC
      H(J1)=H(J3)
234   H(J2)=H(J4)
      GOTO 162
240 I=K1
    K1=K3
    K3=K2
    K2=I
    NSRAK=NSRAK+1
    GOTO 152

```

MTR40836
MTR40837
MTR40838
MTR40840
MTR40841
MTR40842
MTR40844
MTR40845
MTR40846
MTR40847
MTR40848
MTR40849
MTR40850
MTR40852
MTR40853
MTR40855
MTR40857
MTR40859
MTR40860
MTR40861
MTR40862
MTR40864
MTR40865
MTR40866
MTR40868
MTR40869
MTR40870
MTR40872
MTR40873
MTR40875
MTR40876
MTR40877
MTR40878
MTR40880
MTR40882
MTR40884
MTR40885
MTR40886
MTR40887
MTR40888
MTR40889
MTR40891
MTR40893
MTR40894
MTR40895
MTR40896
MTR40897
MTR40898
MTR40899
MTR40900
MTR40901
MTR40902
MTR40903
MTR40904
MTR40905
MTR40907
MTR40908
MTR40909
MTR40910
MTR40911
MTR40912

250	CALL OVERFL (IOVFLW)	MTR40914
	CALL DVCHK (IOVDCT)	MTR40915
	GOTO (260,262) ,IOVFLW	MTR40916
260	IRR=IRR+1	MTR40917
262	GOTO (264,266) ,IOVDCT	MTR40918
264	IRR=IRR+2	MTR40919
266	IF (IRR)	270,270,600
270	IF (K1-I1)	272,280,272
272	DO 274 I=1, KSIZE	MTR40920
	J=K1+I-1	MTR40922
	J1=I1+I-1	MTR40924
274	H(J1) = H(J)	MTR40925
280	J=(MODE-1)*NC+1	MTR40926
	ITERSR=ITERSR-1	MTR40927
	CALL CLOSES(A,H(I1),MSIZE,MAXR,ROP,EPSP,EPDP,NC,IRR,MAXR,ITERDR,	MTR40928
1	NDRAK,INDEX1,INDEX2,VALUE(J), NSIZE)	MTR40929
	IF (IRR)	282,282,610
282	IF (ITERDR)	283,288,288
283	IF (KSIZE-1SIZE)	284,288,288
284	I1 = I1+2*KSIZE	MTR40930
	I2 = I2+KSIZE	MTR40931
	DO 286 I=1, KSIZE	MTR40932
	J1= J1-1	MTR40933
	J2= J2-1	MTR40934
286	H(J2)=H(J1)	MTR40935
288	INDEX = INDEX1	MTR40936
	INDIX = INDEX	MTR40937
290	M1 = NSIZE-1	MTR40938
	J1=INDEX	MTR40939
	IF (J1-M1)	292,292,298
292	DO 296 K=1, ISIZE	MTR40940
	L=(K-1)*MAXR	MTR40941
	L1=L+INDEX	MTR40942
	HOLD= A(L1)	MTR40943
	DO 294 J=J1, M1	MTR40944
	I=L+J	MTR40945
294	A(I) = A(I+1)	MTR40946
296	A(I+1)=HOLD	MTR40947
298	I=NSIZE-MODE+1	MTR40948
	J = (MODE-1) * NC * MAXR +1	MTR40949
	CALL SWAPX (VECTOR(J),A, H, A(L), VALUE, MODE, MSIZE, MAXR, NC,	MTR40950
1	INDIX, EPSP, NSIZE, NITER(MODOUT+1) , IRR)	MTR40951
	CALL OVERFL (IOVFLW)	MTR40952
	CALL DVCHK (IOVDCT)	MTR40953
	GOTO (300,302) ,IOVFLW	MTR40954
300	IRR=IRR+1	MTR40955
302	GOTO (304,306) ,IOVDCT	MTR40956
304	IRR=IRR+2	MTR40957
306	IF (IRR)	310,310,620
310	I=(NC-1)*NSIZE	MTR40958
	DO 312 J=INDEX, NSIZE	MTR40959
	I=I+J	MTR40960
	GUESS(I)=GUESS(I+1)	MTR40961
312	GUESS(J)=GUESS(J+1)	MTR40962
	MSIZE = MSIZE-1	MTR40963
	NITER(MODE) = ITERSR+ITERDR	MTR40964
	NAKSR(MODE)= NSRAK	MTR40965
	NAKDR(MODE)= NDRAK	MTR40966
	IF (ITERDR)	320,360,360
320	MODE=MODE+1	MTR40967
	ITERDR=-ITERDR	MTR40968
		MTR40969
		MTR41000

```

      NITER(MODE-1) = 0.
      INDEX=INDEX2
      IF ( INDEX-NSIZE )
326 IF ( INDEX1-INDEX2 )
330 INDEX=-INDEX
      -INDEX=INDEX-1
      GOTO 342
340 INDEX=-INDEX
342 CONTINUE
      I1=I1+ISIZE
      I2=I2+ISIZE
      I3=I3+ISIZE
      NAKSR(MODE-1) = 0.
      NAKDR(MODE-1) = 0.
      GOTO 290
350 I1=I1+ISIZE
      I2=I1+ISIZE
      I3=I2+ISIZE
      GOTO 130
400 IF ( K1-I1 )
410 DO 412 I=1,KSIZE
      J=K1+I-1
      J1=I1+I-1
412 H(J1)=H(J)
413 DO 414 J=1,NC
      I= NC*(MODE-1)+1
      J1=IK+J-1
414 VALUE(I)= H(J1)
      INDEX=INDEX
      GOTO 290
500 MODE=MODE-1
      IF ( MODE )
502 IF ( NTAPUT )
504 READ ( NTAPUT ) (A(I),I=1,M)
      CALL MULT (A,VECTOR, H,NSIZE,NSIZE,MODE,MAXR,MAXR,MAXR ,NC,1)
      DO 506 I=1,MODE
      J= (I-1)*ISIZE
      J1 = (I-1)*NC+1
      INDEX=0
506 CALL NORM (H(J),H(J1),NSIZE,GUESS(J1),INDEX,MAXR ,NC,1)
510 IF ( NTAPUT )
512 WRITE (NTAPUT,10 )
      DO 522 I=1,MODE
      J1= MAXSR
      J2= NITER(I)-MAXSR
      IF ( J2 )
514 J1=NITER(I)
      J2=0
      GO TO 517
515 IF ( J2-MAXDR )
516 WRITE (NTAPUT,24) 1
517 GO TO (518,520) ,NC
518 WRITE (NTAPUT,11) 1, VALUE(I), J1,J2, NAKSR(I),NAKDR(I)
      GOTO 522
520 J= 2*I
      WRITE (NTAPUT,12) 1, VALUE(J-1), VALUE(J), J1,J2, NAKSR(I),
      NAKDR(I)
522 CONTINUE
      J=NC*MODE
      WRITE (NTAPUT,20)
      CALL MPRINT (VECTOR,NSIZE,J,MAXR,NTAPUT)

```

MTR41001
 MTR41002
 MTR41003
 MTR41004
 MTR41005
 MTR41006
 MTR41007
 MTR41008
 MTR41009
 MTR41010
 MTR41011
 MTR41012
 MTR41013
 MTR41014
 MTR41015
 MTR41017
 MTR41016
 MTR41019
 MTR41020
 MTR41022
 MTR41023
 MTR41024
 MTR41025
 MTR41026
 MTR41027
 MTR41028
 MTR41029
 MTR41030
 MTR41031
 MTR41032
 MTR41034
 MTR41035
 MTR41036
 MTR41037
 MTR41039
 MTR41041
 MTR41043
 MTR41044
 MTR41045
 MTR41046
 MTR41048
 MTR41050
 MTR41051
 MTR41052
 MTR41053
 MTR41054
 MTR41055
 MTR41056
 MTR41057
 MTR41058
 MTR41059
 MTR41060
 MTR41062
 MTR41063
 MTR41065
 MTR41066
 MTR41067
 MTR41068
 MTR41069
 MTR41070
 MTR41071

```

      IF ( NTAPOT )
524 WRITE (NTAPOT,22) ((GUESS(I),I=1,J))
      CALL MPRINT ( H,NSIZE,J,MAXR,NTAPOT)
530 RETURN
540 NITER(MODE) = ITERS+ITEROR
      NAKSR(MODE) = NSRAK
      NAKDR(MODE) = NDRAK
      IF ( NTAPOT )
550,550,524
561 J=1
      J1=4
      GOTO (602,604,606),IRR
562 J1=2
      GOTO 606
564 J1=4
566 WRITE (NTAPOT,26) (ATITLE(I),I=J,J1)
      GOTO 630
568 J2=6
569 NITER(MODE) = ITERS+ITEROR
      NAKSR(MODE) = NSRAK
      NAKDR(MODE) = NDRAK
      IF ( NTAPOT )
580,580,613
581 J=1
      J1=4
      GOTO (614,616,618),IRR
584 J1=2
      GOTO 618
586 J1=4
588 WRITE (NTAPOT,26) (ATITLE(I),I=J,J1), (ATITLE(5)),
      (ATITLE(J2)), (ATITLE(I),I=8,9)
      IF ( J2=6 )
590,590,622
592 J2=7
      GOTO 612
594 WRITE (NTAPOT,18) MODE
      IRR=0
      J= (MODE-1)*NC*MAXR
      DO 624 I=1,ISIZE
          J1=J+1
          J2=J+(NC-1)*MAXR+1
          VECTOR(J2)=0.
596 VECTOR(J1)=0.
      GOTO 310
598 I=MODE-
      WRITE (NTAPOT,16) I
      GOTO 500
      END
*
* FORTRAN LISTING DECK
*
* SWEEPS
*
* S W E E P X S U B R O U T I N E
*
* COMPUTES TRUE MODE AND SWEEPS IT FROM THE MATRIX. (REAL OR COMPLEX)
*
* HTRUE = TRUE MODAL COLUMNS, AS COMPUTED. U = DYNAMIC MATRIX.
* H = SERIES OF MODIFIED MODAL COLUMNS. FL= COLUMN OF EIGENVALUES.
* US = SERIES OF MODIFIED MODAL ROWS OF U.
* MODE = MODE NOW BEING COMPUTED. N = SIZE
* MD = DIMENSIONED NUMBER OF ROWS OF U,US,H,HTRUE
* NX = 1 IF PROBLEM IS REAL.
*      = 2 IF PROBLEM IS COMPLEX.
*
* SUBROUTINE SWEEPX (HTRUE, U, H, US, FL, MODE, N, NC, NC, INDIX,
*      EP, MSIZE, INDIS, IRR)
*
* DIMENSION H(1), US(1), U(1), HTRUE(1), FL(1), G(4) INDIS(1)

```

MTR41071
MTR41072
MTR41073
MTR41074
MTR41075
MTR41076
MTR41077
MTR41078
MTR41079
MTR41080
MTR41081
MTR41082
MTR41083
MTR41084
MTR41085
MTR41086
MTR41087
MTR41088
MTR41089
MTR41090
MTR41091
MTR41092
MTR41093
MTR41094
MTR41095
MTR41096
MTR41097
MTR41098
MTR41099
MTR41100
MTR41101
MTR41102
MTR41103
MTR41104
MTR41105
MTR41106
MTR41107
MTR41108
MTR41109
MTR41110
MTR41111
MTR41112
MTR41113
MTR41114
MTR41115
MTR41116
MTR41117
MTR41118
MTR41119
MTR41120
MTR41121
MTR41122
MTR41123
MTR41124

MTR41126
MTR41127
MTR41128
MTR41129
MTR41130
MTR41131
MTR41132
MTR41133
MTR41134
MTR41135
MTR41136
MTR41137
MTR41138
MTR41139

```

INDEX=IARS(INDEX)
INDIS(MODE) = INDEX
M=MODE-1
K1=M*NC*MD
NN=NC*MD
K2=MSIZE-M
IF ( M ) 70,70,K0
70 GOTO (140,72) ,NC
72 IF ( MSIZE-MD ) 74,140,140
74 L=K1+2*N+1
K=K1+MD+N+1
DO 76 I=1,N
K=K-1
I=L-1
H(K)=H(L)
76 H(L)=0.
GOTO 140
80 DO 99 I=1,M
J1=MODE+I
90 INDIS(J1)=INDIS(I)
100 IF ( INDEX ) 102,104,104
102 K2=K2+1
M=M-1
104 J1=K1+K2*NC+1
J2=K1+MSIZE*NC
IF ( J1-J2 ) 105,105,107
105 DO 106 I=J1,J2
106 H(I)=0.
107 GO TO (114,108),NC
108 J1=K1+K2+MD+1
J2=K1+K2*NC+1
DO 110 I=1,K2
J1=J1-1
J2=J2-1
110 H(J1)=H(J2)
IF ( M ) 118,118,111
111 DO 112 I=1,M
H(J2)=0.
112 J2=J2+1
114 IF ( M ) 118,118,120
120 II=1
DO 120 I=1,M
121 DO 122 J=1,M
J1=MODE+J
IF ( II - INDIS(J1) ) 122,122,122
122 CONTINUE
II=II+1
GOTO 121
123 INDIS(J1)=0
IF ( INDIS(MODE)-II ) 125,124,124
124 INDIS(MODE)=INDIS(MODE)+1
125 I=MSIZE-II
IF ( I ) 129,129,126
126 J1=K1+MSIZE
J2=K1+MSIZE+(NC-1)*MD
DO 128 J=1,I
J1=J1-1
J2=J2-1
H(J2+1) = H(J2)
128 H(J1+1) = H(J1)
129 J1=K1+II

```

```

MTR41156
MTR41157
MTR41158
MTR41159
MTR41160
MTR41161
MTR41162
MTR41163
MTR41164
MTR41165
MTR41166
MTR41167
MTR41168
MTR41169
MTR41170
MTR41171
MTR41172
MTR41173
MTR41174
MTR41175
MTR41176
MTR41177
MTR41178
MTR41179
MTR41180
MTR41181
MTR41182
MTR41183
MTR41184
MTR41185
MTR41186
MTR41187
MTR41188
MTR41189
MTR41190
MTR41191
MTR41192
MTR41193
MTR41194
MTR41195
MTR41196
MTR41197
MTR41198
MTR41199
MTR41200
MTR41201
MTR41202
MTR41203
MTR41204
MTR41205
MTR41206
MTR41207
MTR41208
MTR41209
MTR41210
MTR41211
MTR41212
MTR41213
MTR41214
MTR41215
MTR41216

```

```

      J2=J1+(NC-1)*MD
      H(J1)=0.
      H(J2)=0.
130  CONTINUE
      IF ( INDEX ) 119,131,131
118  M=M+1
131  DO 132 I=1,M
      J1=MODE+I
132  INDIS(J1)=INDIS(I)
      I1=1
      DO 133 I=1,M
133  DO 134 J=1,M
      J1=MODE+J
      IF ( I1-INDIS(J1) ) 1134,1135,1134
1134  CONTINUE
      I1=I1+1
      GOTO 133
1135  L1=(NC*MSIZE-1)*MD+MODE+1
      L3=(MSIZE-1)*NM
      INDIS(J1)=0
      DO 138 J=1,NC
      J2=L1-(J-1)*MD-1
      IF ( L3-NM ) 138,134,134
134  DO 136 I=1,L3,NM
      J1=L1-I
      J2=J1-NM
136  US(J1)=US(J2)
      I1=L1-MD
138  US(J2)=0.
140  DO 142 I=1,NM
      J1=K1+I
142  HTRUE(I)=H(J1)
      IF ( M ) 31,31,R
      R DO 25 I=1,M
      K=MODE-I
      L1=NC*MD*(K-1)
      CALL MULT (US( K ), HTRUE, G, 1, MSIZE, 1, MD, M, 1, NC, 1 )
      IF ( G(1) ) 12,9,12
      9  GO TO (11,10),NC
10  IF ( G(1) ) 12,11,12
11  IRR=IRR+2
12  CONTINUE
      GOTO (14,10),NC
14  IF (ABS(FI(K)/FI(MODE))-1.0) - 1P ) 15,15,16
15  Q=1.0
      GOTO 17
16  Q=(FI(K)-FI(MODE)) / Q
17  DO 18 J=1,MSIZE
      I=L1+J
18  HTRUE(J)=H(I)-G(1)*HTRUE(J)
      GOTO 25
19  K=2*K
      I=2*MODE
      IF ( ABS((FI(K-1)*FI(J-1)+FI(K)*FI(J))/(FI(J-1)**2+FI(J)**2)-1.0)
        - 1P ) 20,20,22
20  IF ( ABS((FI(K)*FI(J-1)-FI(K-1)*FI(J)) / (FI(J-1)**2+FI(J)**2))
        - 1P ) 21,21,22
21  G(1)=1.0
      G(2)=0.0
      GOTO 23
22  G(3)=G(1)**2+G(2)**2

```

MTR41217
 MTR41218
 MTR41219
 MTR41220
 MTR41221
 MTR41222
 MTR41223
 MTR41224
 MTR41225
 MTR41226
 MTR41227
 MTR41228
 MTR41229
 MTR41230
 MTR41231
 MTR41232
 MTR41233
 MTR41234
 MTR41235
 MTR41236
 MTR41237
 MTR41238
 MTR41239
 MTR41240
 MTR41241
 MTR41242
 MTR41243
 MTR41244
 MTR41245
 MTR41246
 MTR41247
 MTR41248
 MTR41249
 MTR41250
 MTR41251
 MTR41252
 MTR41253
 MTR41259
 MTR41260
 MTR41261
 MTR41262
 MTR41263
 MTR41264
 MTR41265
 MTR41266
 MTR41267
 MTR41268
 MTR41269
 MTR41270
 MTR41271
 MTR41272
 MTR41273
 MTR41274
 MTR41275
 MTR41276
 MTR41277
 MTR41278
 MTR41279
 MTR41280
 MTR41281
 MTR41282

```

G(4)=(FL(K)-FL(J))*G(1)-(FL(K-1)-FL(J-1))*G(2)
G(1)=((FL(K-1)-FL(J-1))*G(1)+(FL(K)-FL(J))*G(2)) / G(3)
G(2)= G(4) / G(3)

```

```

23 DO 24 J1=1,MSIZE

```

```

    K2=J1+MD

```

```

    L=L1+J1

```

```

    I2=I+MD

```

```

    G(3)=HTRUE(J1)

```

```

    HTRUE(J1) = H(L) + G(2)*HTRUE(K2) -G(1)*HTRUE(J1)

```

```

    HTRUE(K2)= H(L2)- G(1)*HTRUE(K2)-G(2)*G(3)

```

```

24 CONTINUE

```

```

25 CONTINUE

```

```

    I=0

```

```

    CALL NORM ( HTRUE,HTRUE,MSIZE,G,I,MD,NC,1)

```

```

26 J1 = 1

```

```

    J2 = 1

```

```

    L4 = MODE

```

```

DO 43 I=1,MSIZE

```

```

    I1=J1

```

```

    I2=J2

```

```

    I3=K1+1

```

```

DO 33 MM=1,MODE

```

```

    IF ( I -INDIS(MM) )

```

```

        33,39,33

```

```

27 CONTINUE

```

```

DO 37 J=1,MSIZE

```

```

    DO 35 MM=1,MODE

```

```

        IF ( J-INDIS(MM) )

```

```

            35,37,35

```

```

35 CONTINUE

```

```

    U(L1) = U(I2) - H(I3)*US(L4)

```

```

    GOTO (38,36),NC

```

```

36 J3= L1+MD

```

```

    J4= L2+MD

```

```

    J5= L4+MD

```

```

    J6= L3+MD

```

```

    U(J3)=U(J4)-H(I3)*US(J5) - H(J6)*US(L4)

```

```

    U(L1)= U(I1) + H(J6)*US(J5)

```

```

38 CONTINUE

```

```

    L1=L1+1

```

```

    L2=L2+1

```

```

37 L3=L3+1

```

```

    GOTO 41

```

```

39 IF ( I-INDIS(MODE) )

```

```

        43,42,43

```

```

41 J1=J1+NN

```

```

42 J2=J2+NN

```

```

43 L4=L4+NN

```

```

    L4=(MSIZE-MODE)*NN+1

```

```

DO 52 J=1,NC

```

```

    L4 = L4 + (J-1)*MD

```

```

    L1=L4+(MSIZE-MODE)-1

```

```

DO 52 I=L4,1

```

```

52 U(I)=0.

```

```

    RETURN

```

```

END

```

```

FORTRAN 1 STOU,DECK

```

```

C CHANGES

```

```

SUBROUTINE CHANGE (A,M,N,MAXR,ICHU2)

```

```

DIMENSION A(1)

```

```

MR=2*MAXR

```

```

DO 10 (10,20),ICHU2

```

```

10 DO 12 I=1,N

```

```

    K=(N-1)*MAXR+M+1

```

```

MTR41283

```

```

MTR41284

```

```

MTR41285

```

```

MTR41286

```

```

MTR41287

```

```

MTR41288

```

```

MTR41289

```

```

MTR41290

```

```

MTR41291

```

```

MTR41292

```

```

MTR41293

```

```

MTR41296

```

```

MTR41297

```

```

MTR41298

```

```

MTR41299

```

```

MTR41300

```

```

MTR41301

```

```

MTR41302

```

```

MTR41303

```

```

MTR41304

```

```

MTR41305

```

```

MTR41306

```

```

MTR41307

```

```

MTR41308

```

```

MTR41309

```

```

MTR41310

```

```

MTR41311

```

```

MTR41312

```

```

MTR41313

```

```

MTR41314

```

```

MTR41315

```

```

MTR41316

```

```

MTR41317

```

```

MTR41318

```

```

MTR41319

```

```

MTR41320

```

```

MTR41321

```

```

MTR41322

```

```

MTR41323

```

```

MTR41324

```

```

MTR41325

```

```

MTR41326

```

```

MTR41327

```

```

MTR41328

```

```

MTR41329

```

```

MTR41331

```

```

MTR41332

```

```

MTR41333

```

```

MTR41334

```

```

MTR41335

```

```

MTR41336

```

```

MTR41342

```

```

MTR41343

```

```

MTR41345

```

```

MTR41346

```

```

MTR41347

```

```

MTR41348

```

```

MTR41349

```

```

MTR41350

```

```

      KK=(N-1)*MR+2*M+1
      DO 12 J=1,M
        IGFT=K-J
        IPUT=KK-2*J
        A(IPUT)=A(IGFT)
12     A(IPUT+1)=0.
      GO TO 100
20    K=0
      KK=0
      DO 24 I=1,N
        DO 22 J=1,M
          IGFT = KK+2*J-1
          IPUT=K+J
22     A(IPUT) = A(IGFT)
        K=K+MAXK
24     KK=KK+MR
100   RETURN
      END

```

```

MTR41351
MTR41352
MTR41353
MTR41354
MTR41355
MTR41356
MTR41357
MTR41358
MTR41359
MTR41360
MTR41361
MTR41362
MTR41363
MTR41364
MTR41365
MTR41366
MTR41367
MTR41368

```

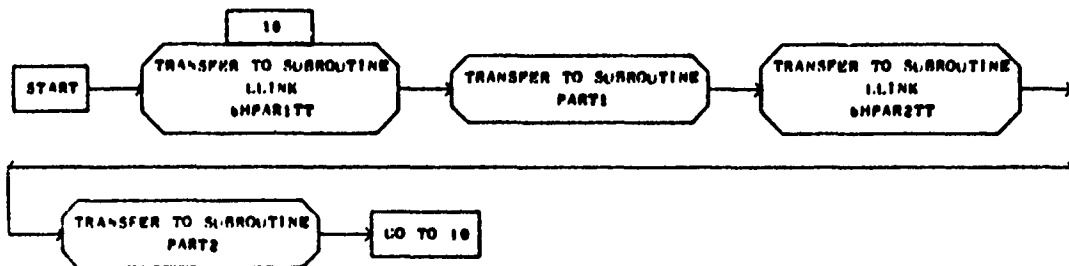
SECTION 9

FLOW DIAGRAMS

MAIN PLOTTER OVERLAY
JAN 13, 1967

COMMON 17(216)

PAGE 1



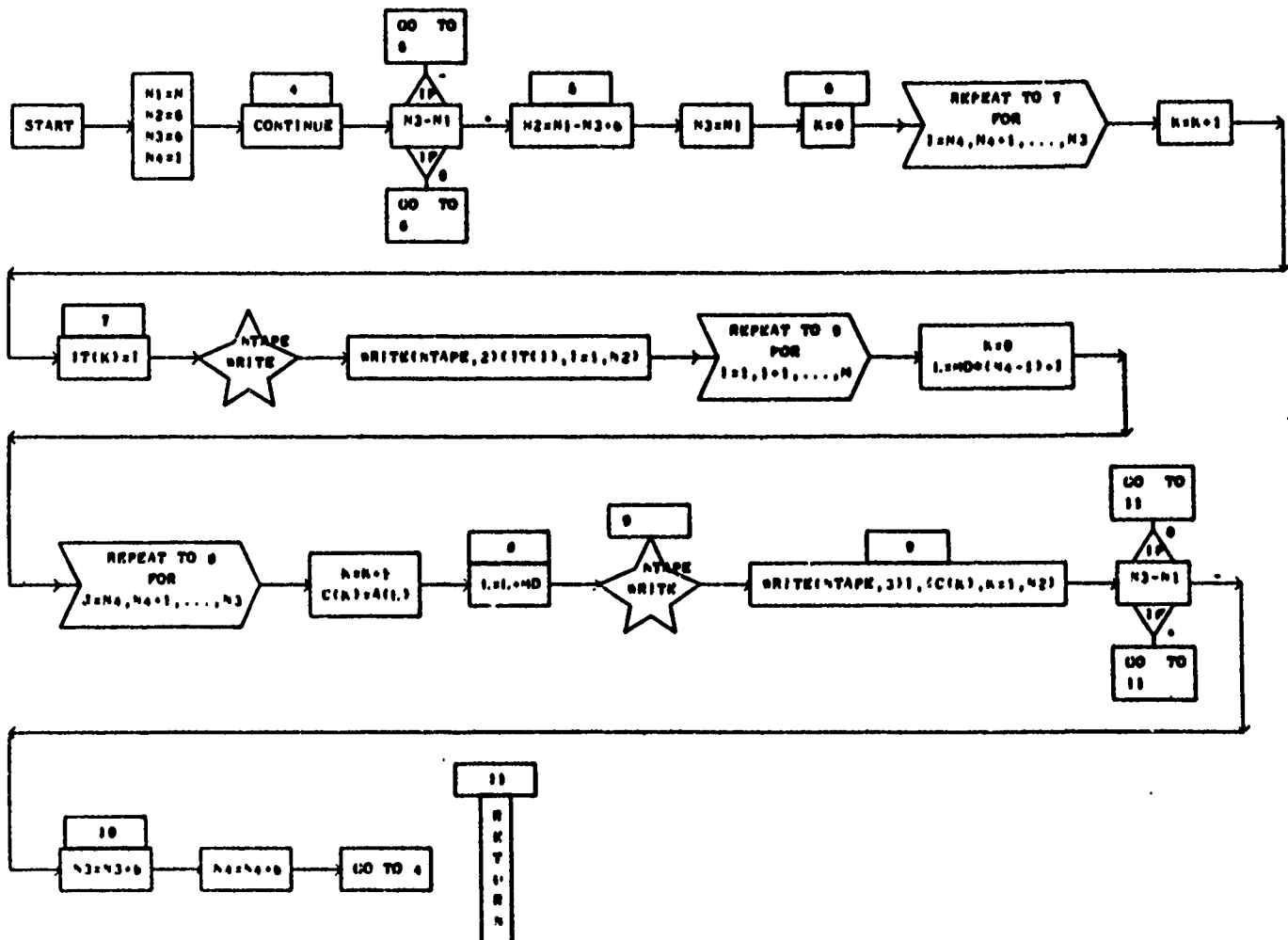
MPRINT

DIMENSIONED VARIABLES

SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
A	1	IT	6	C	6				

SUBROUTINE MPRINT (A,N,M,ND,NTAPE)

PAGE 1



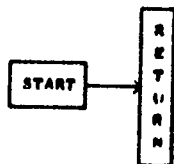
MPUNCH

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
A	1								

SUBROUTINE MPUNCH(A,M,N,IOUT,ITRA,IONG,BCDZ,MAXM,NTAPE,NCARDS)

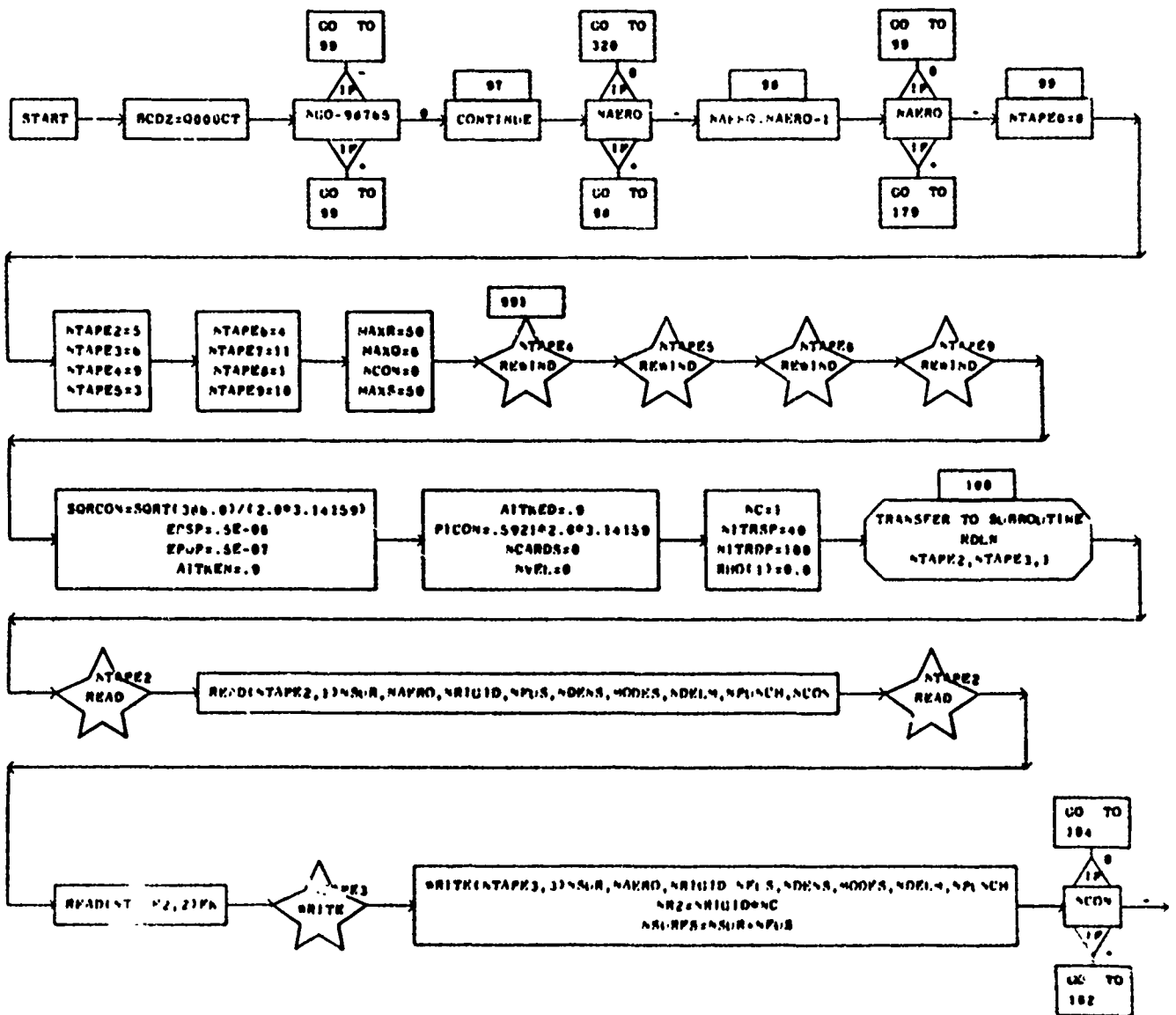
PAGE 1

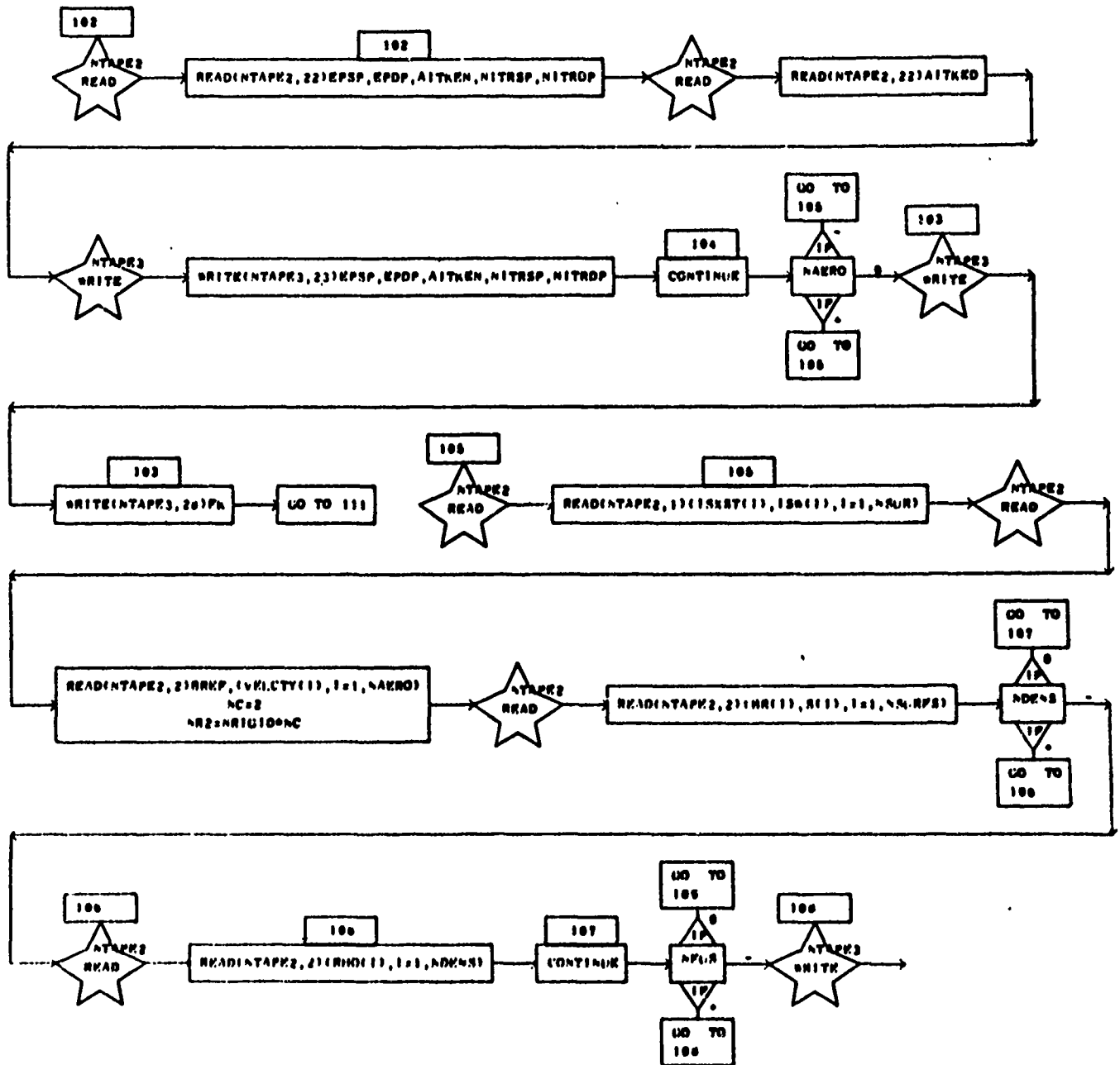


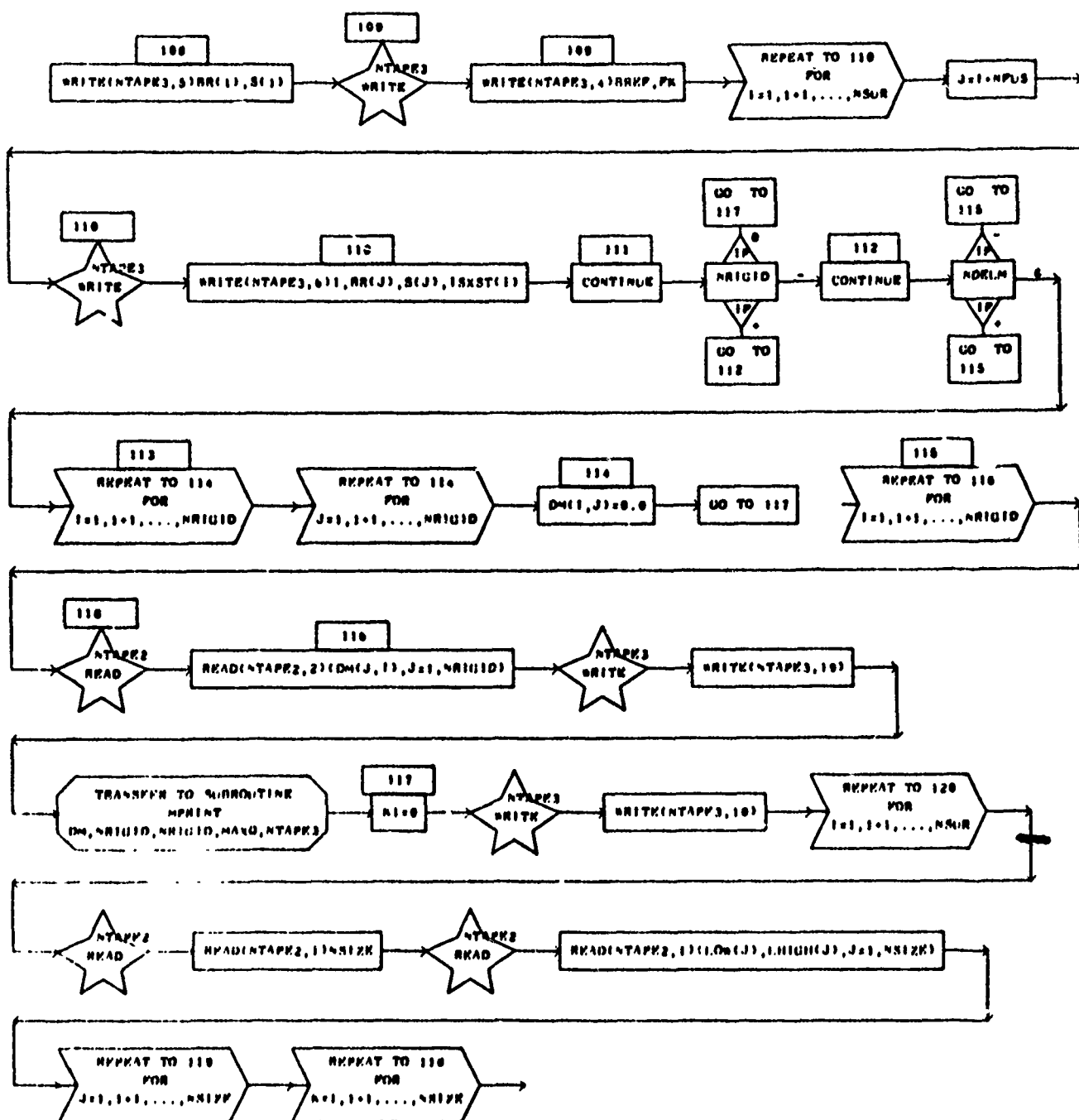
PART1

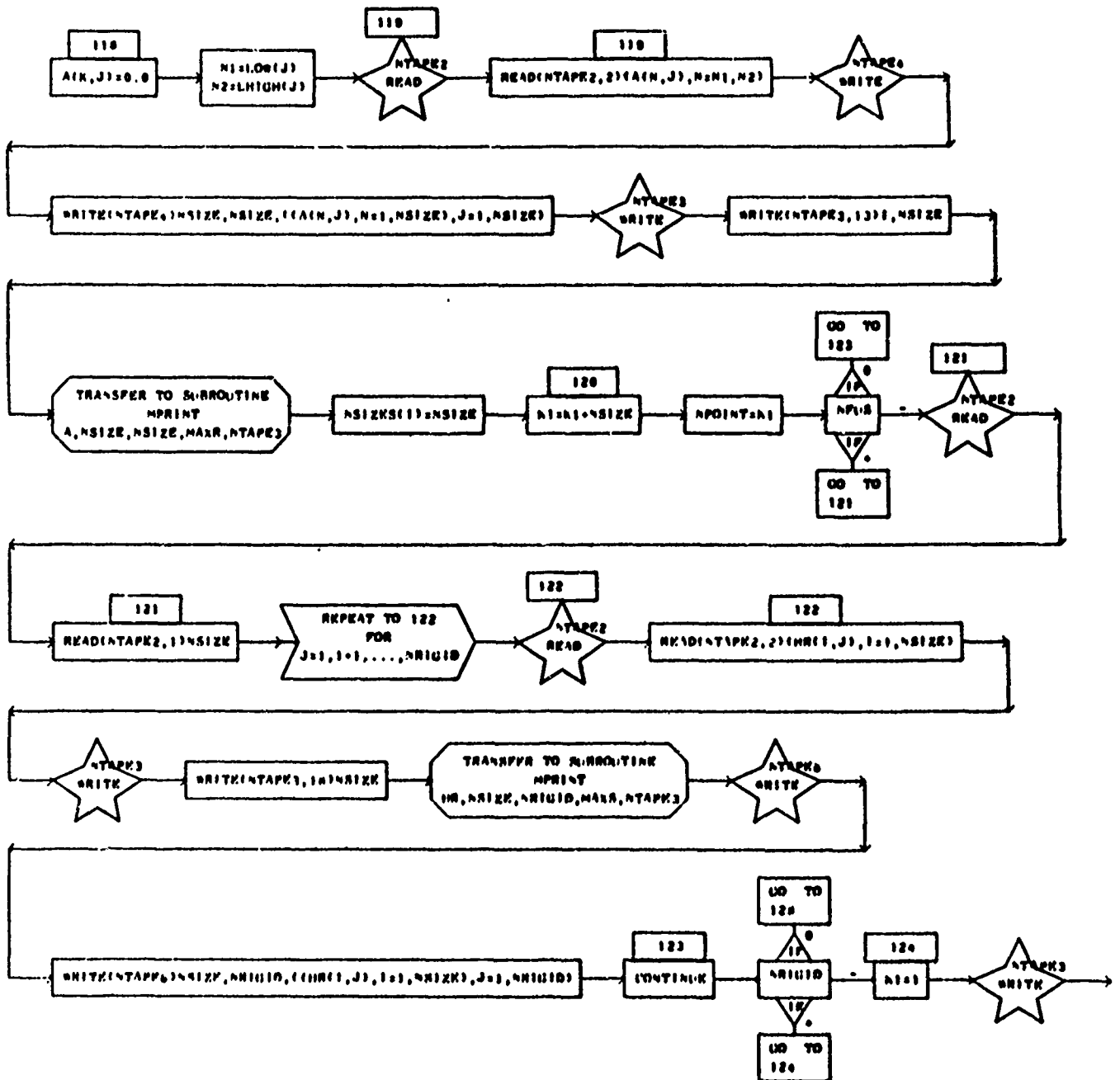
D I M E N S I O N E D V A R I A B L E S

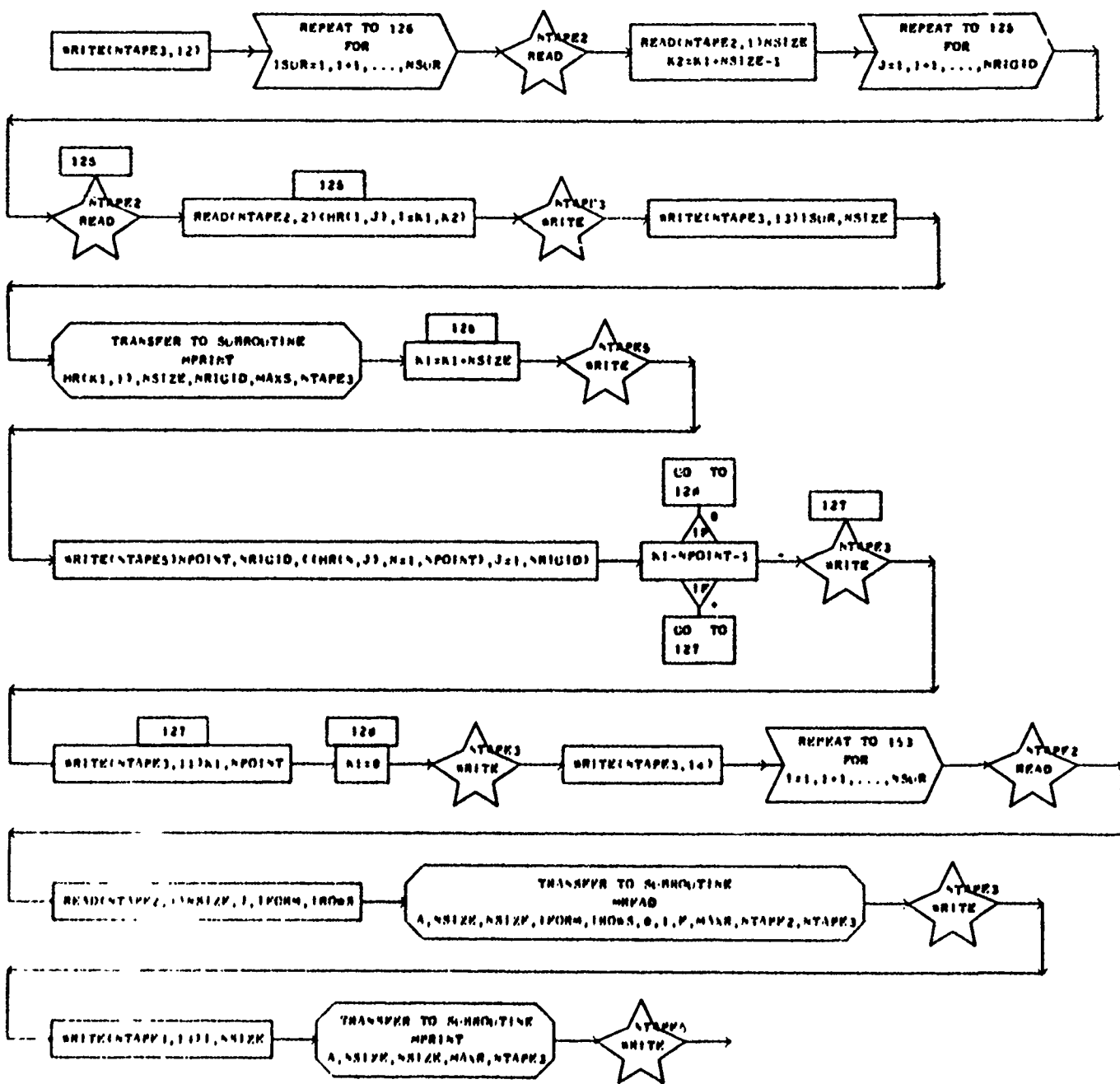
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
ISIST	20	ISO	20	BR	21	S	21	RND	20
DM	0,0	DO	0,12	DMSAR	0,12	RAR+RR	0,12	LOS	50
I.HIGH	50	IT	210	VRCTY	20	NRTZES	20	A	50,100
P	50,100	U	5,100	NR	50,0	HRT	0,100	MY	0,100
G	0,100								

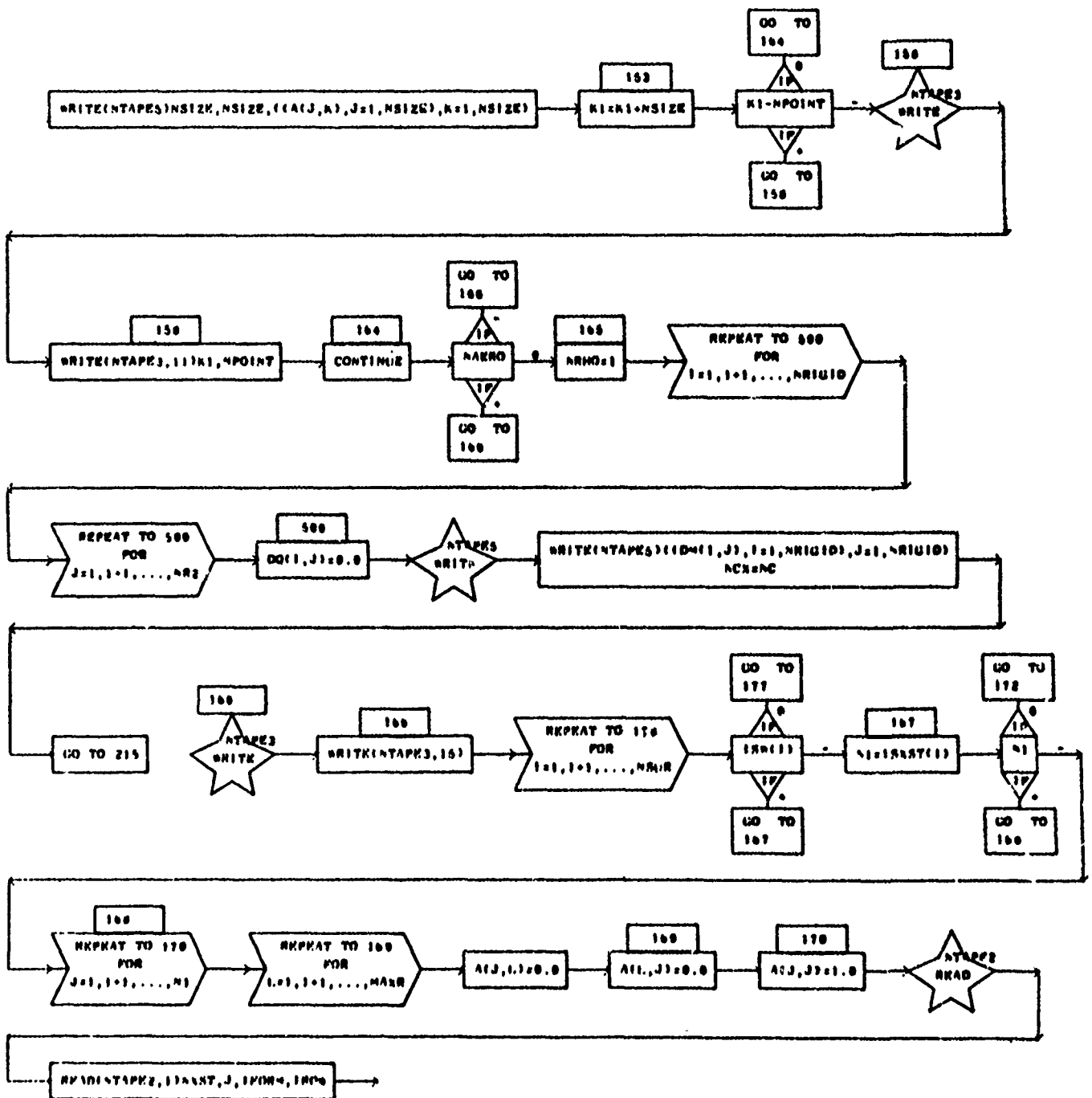




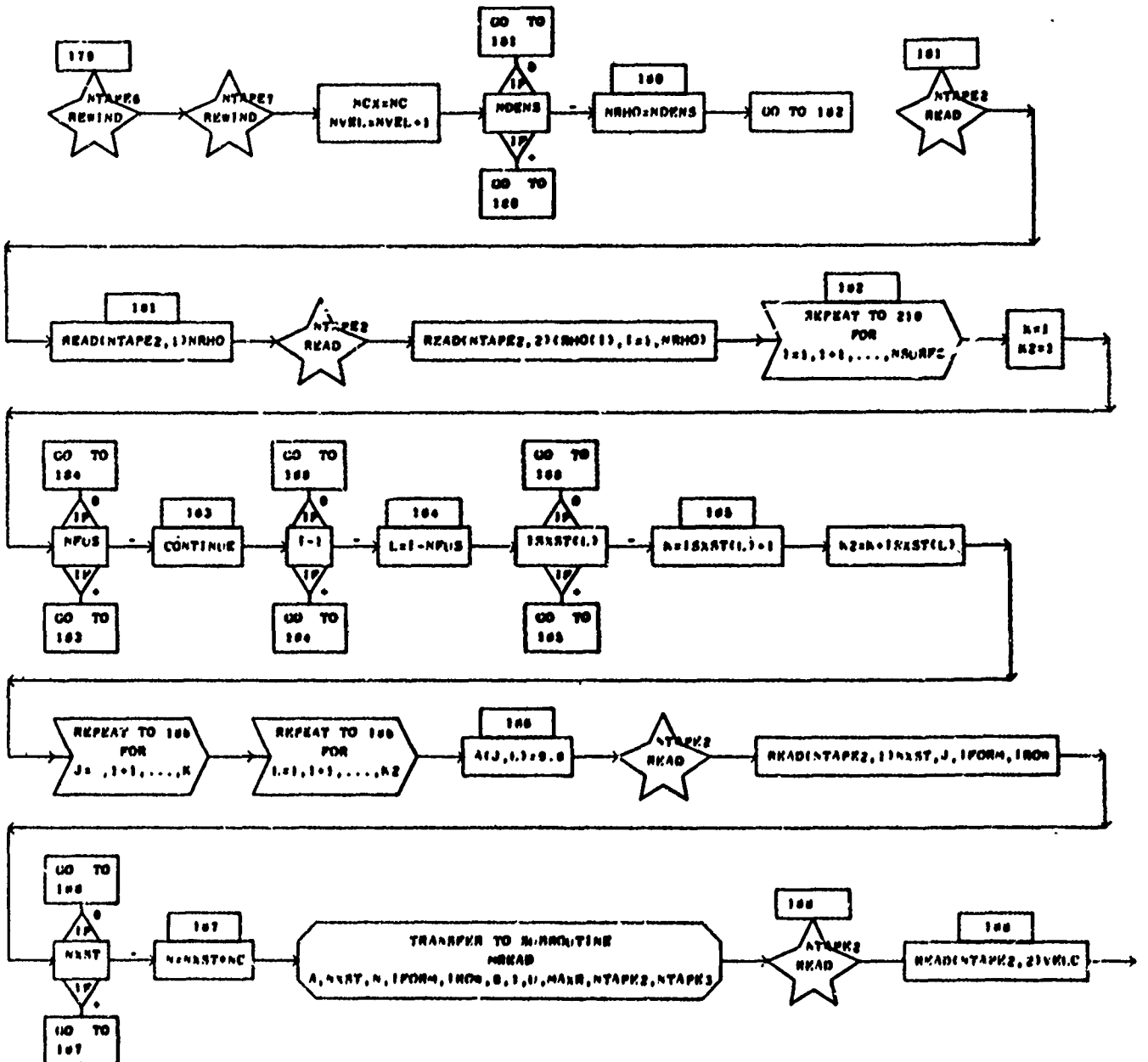




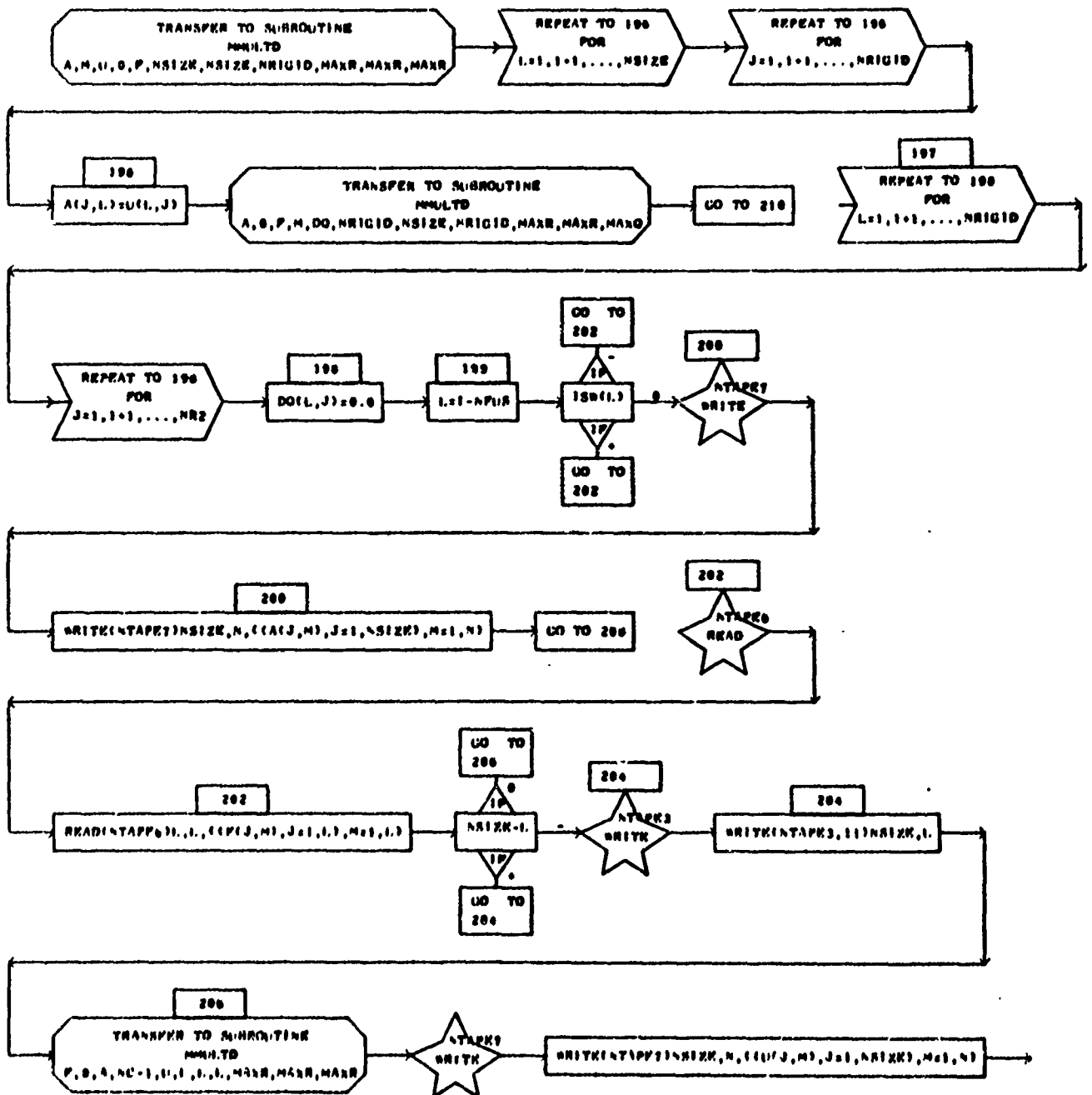


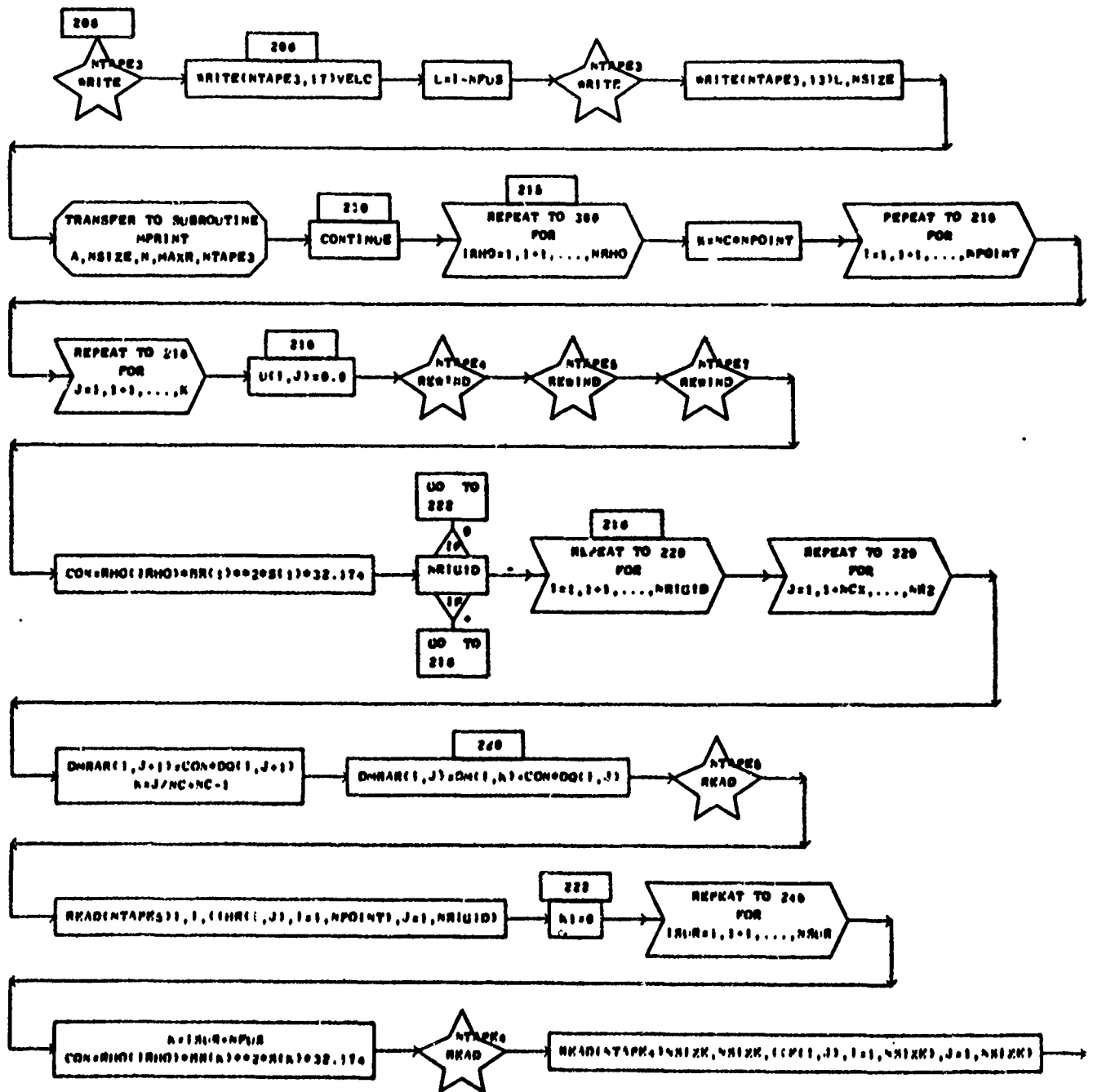


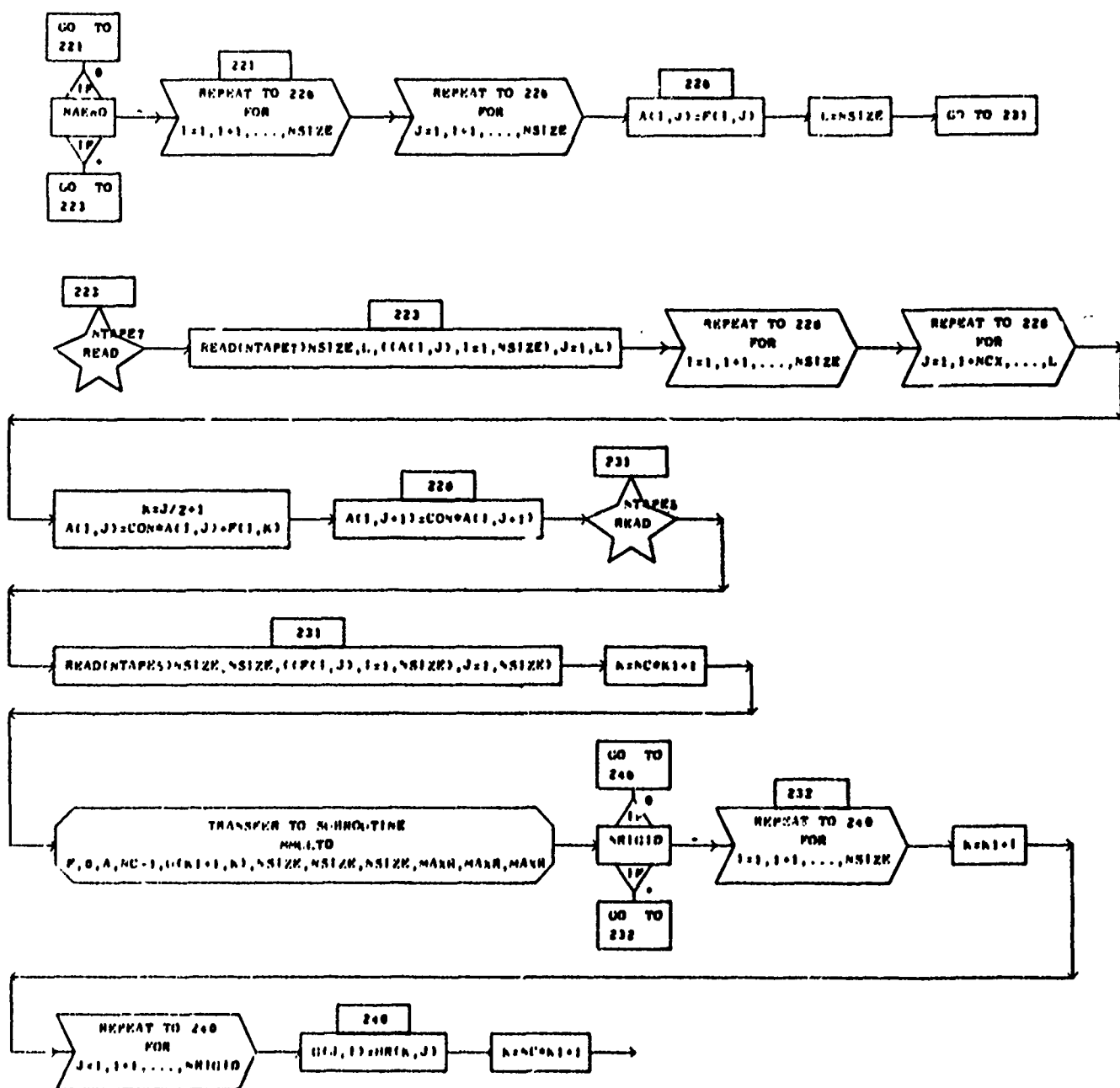


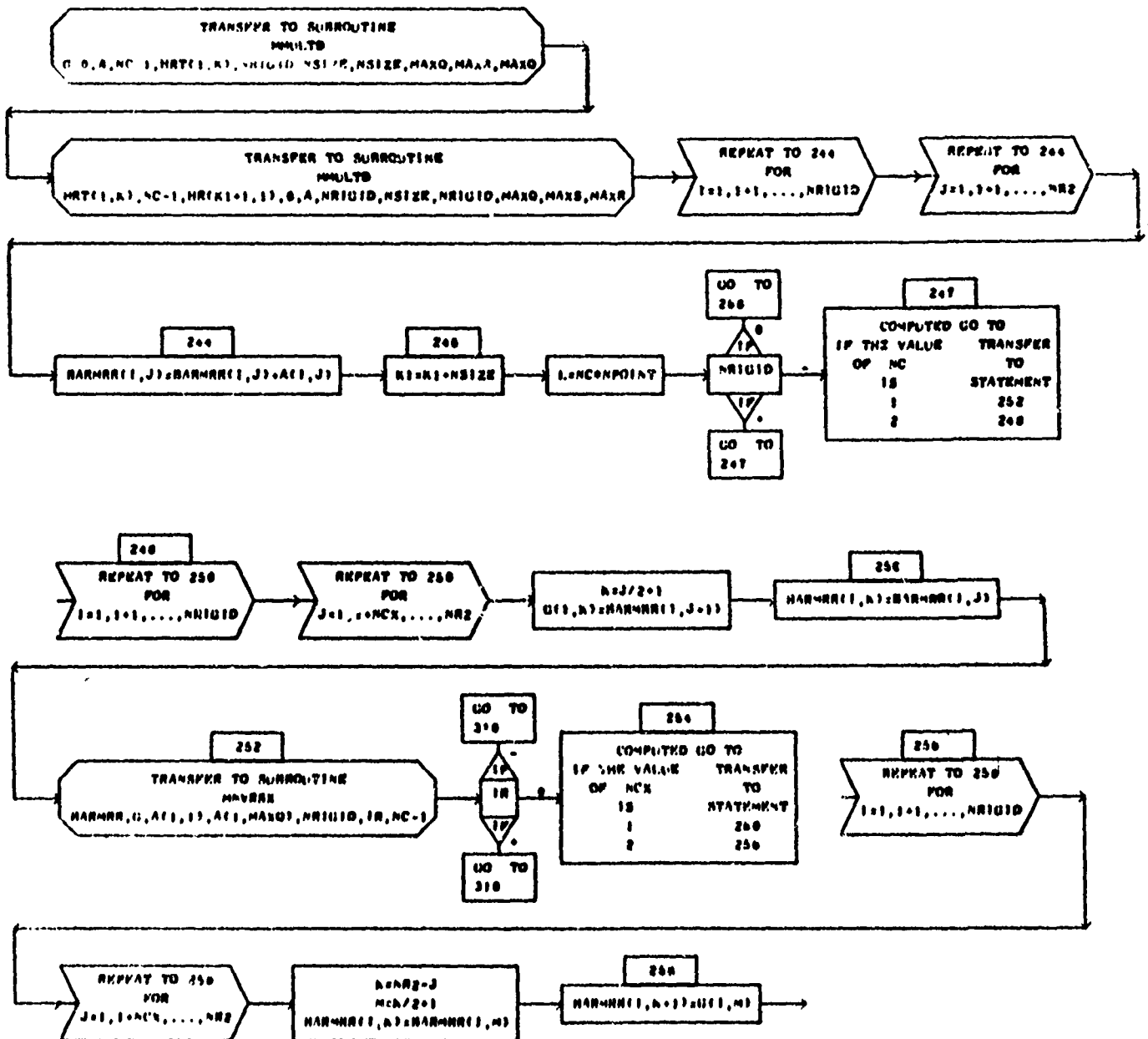




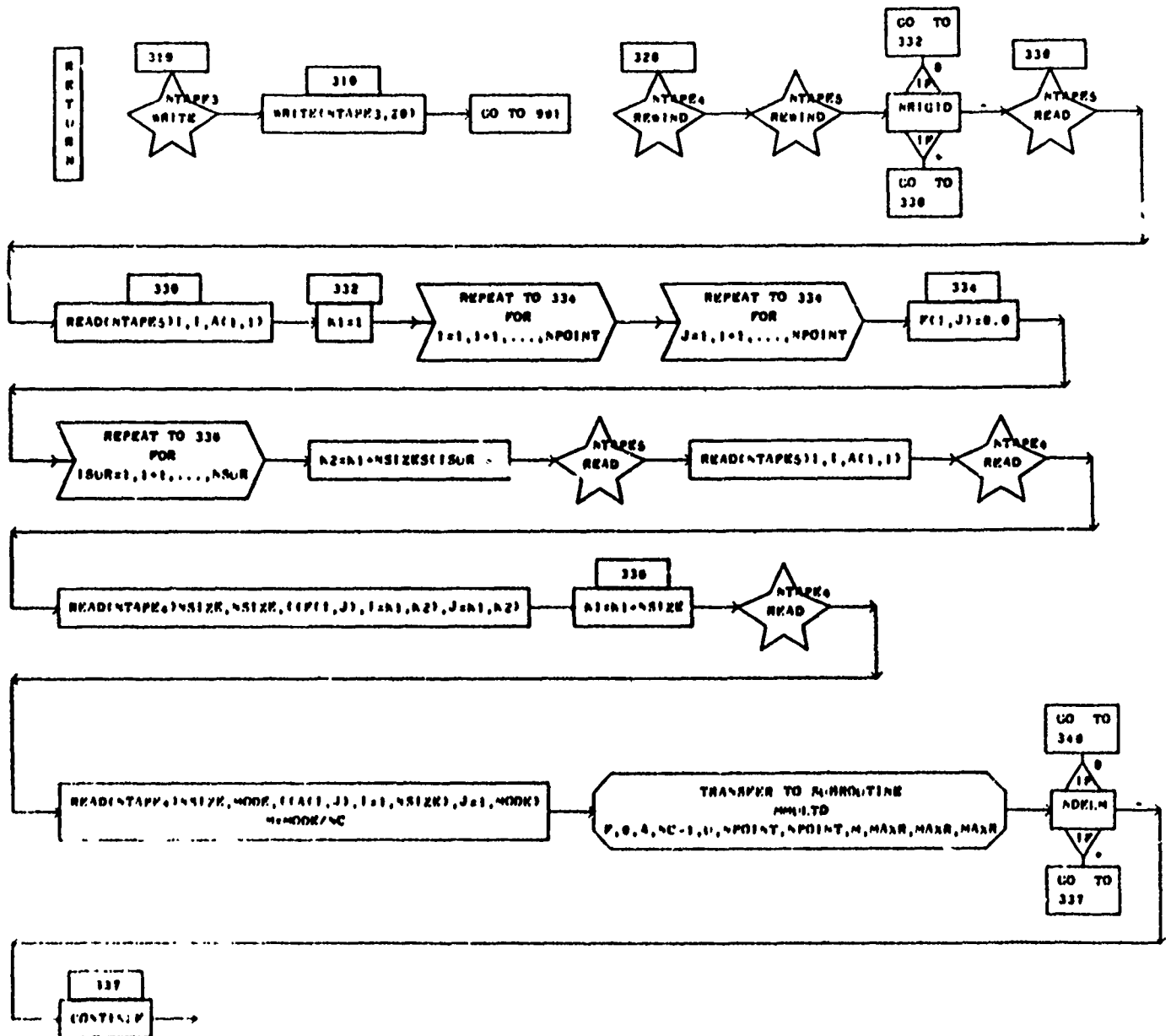


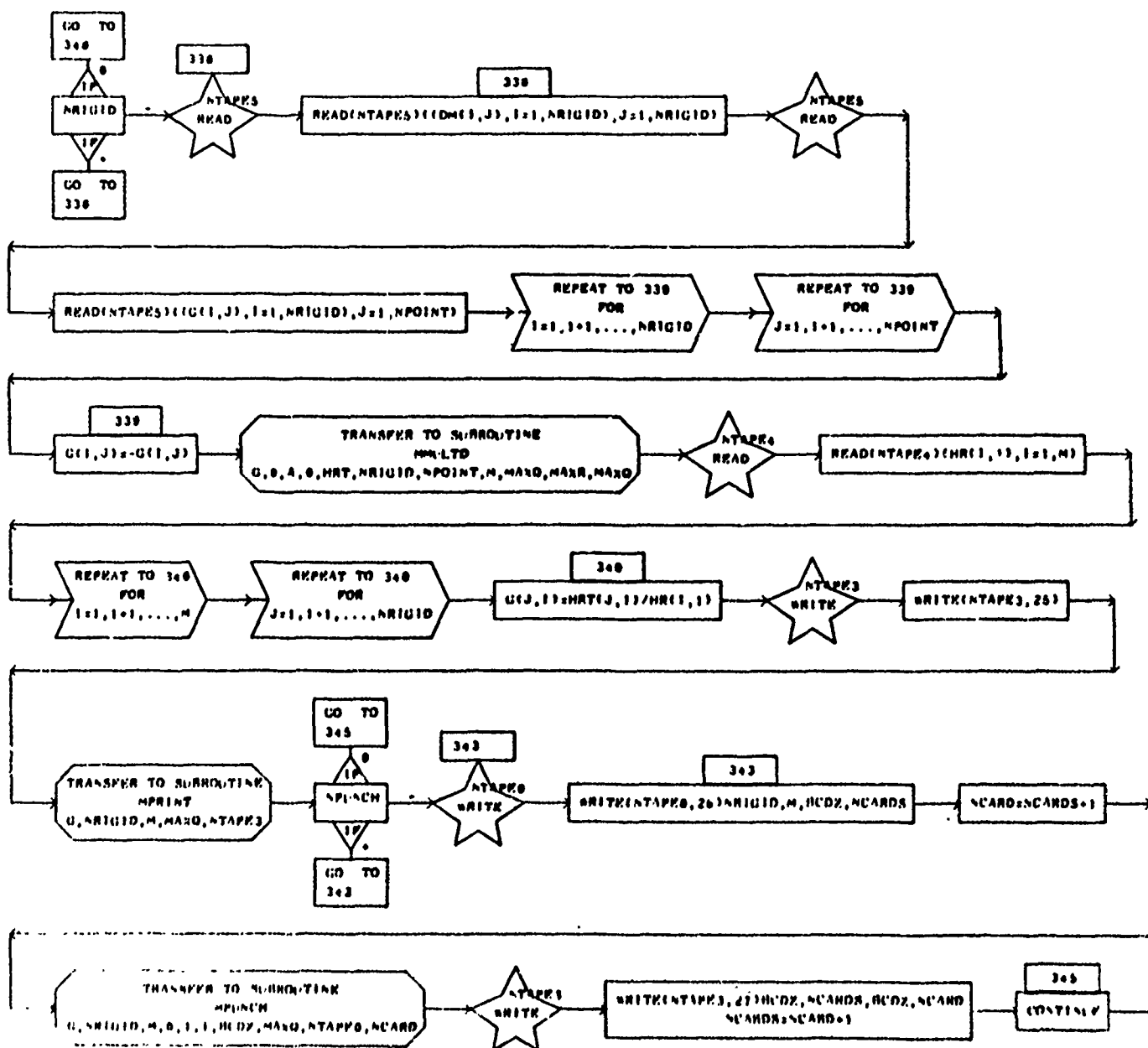


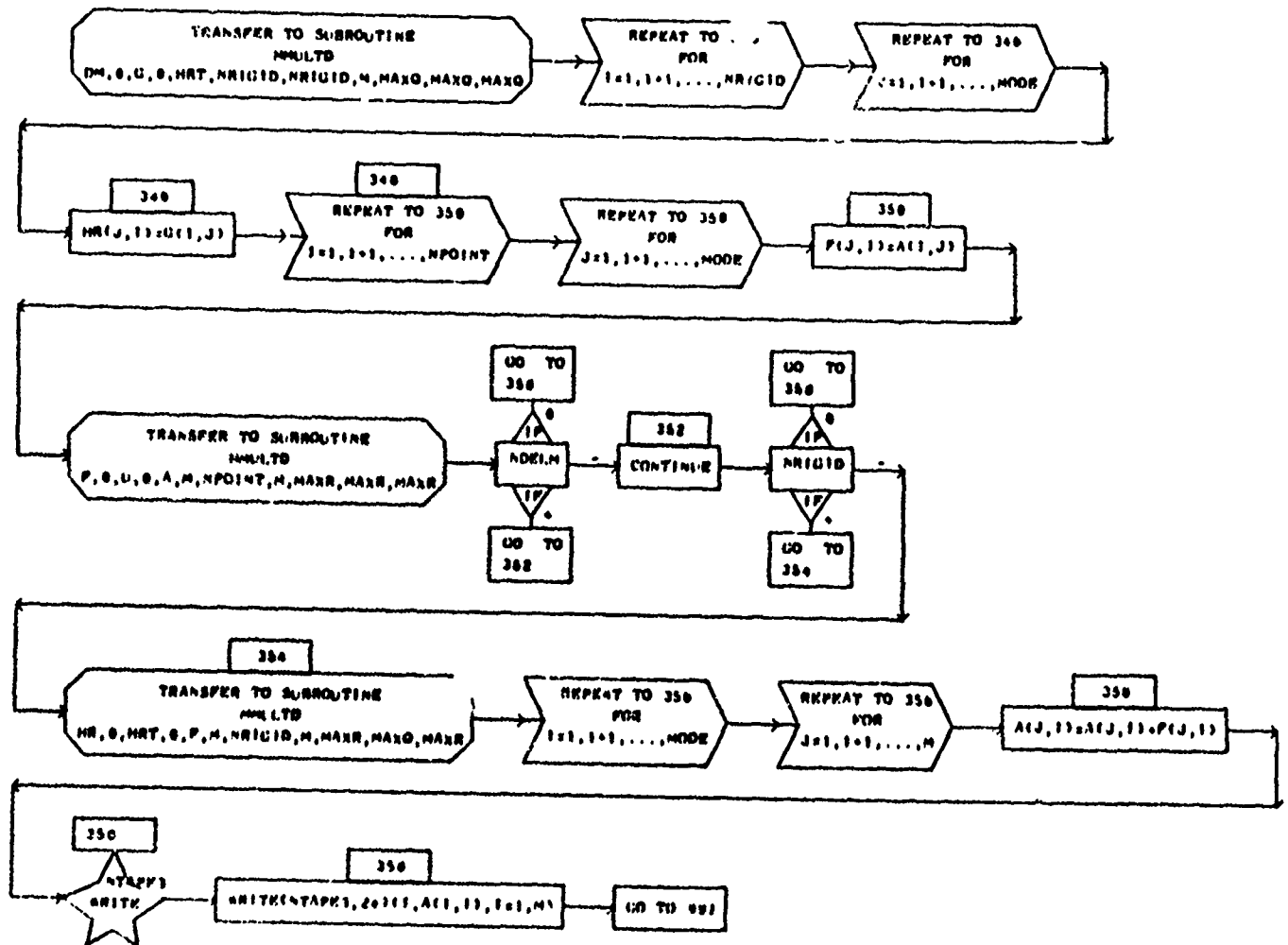












MREAD

MREAD

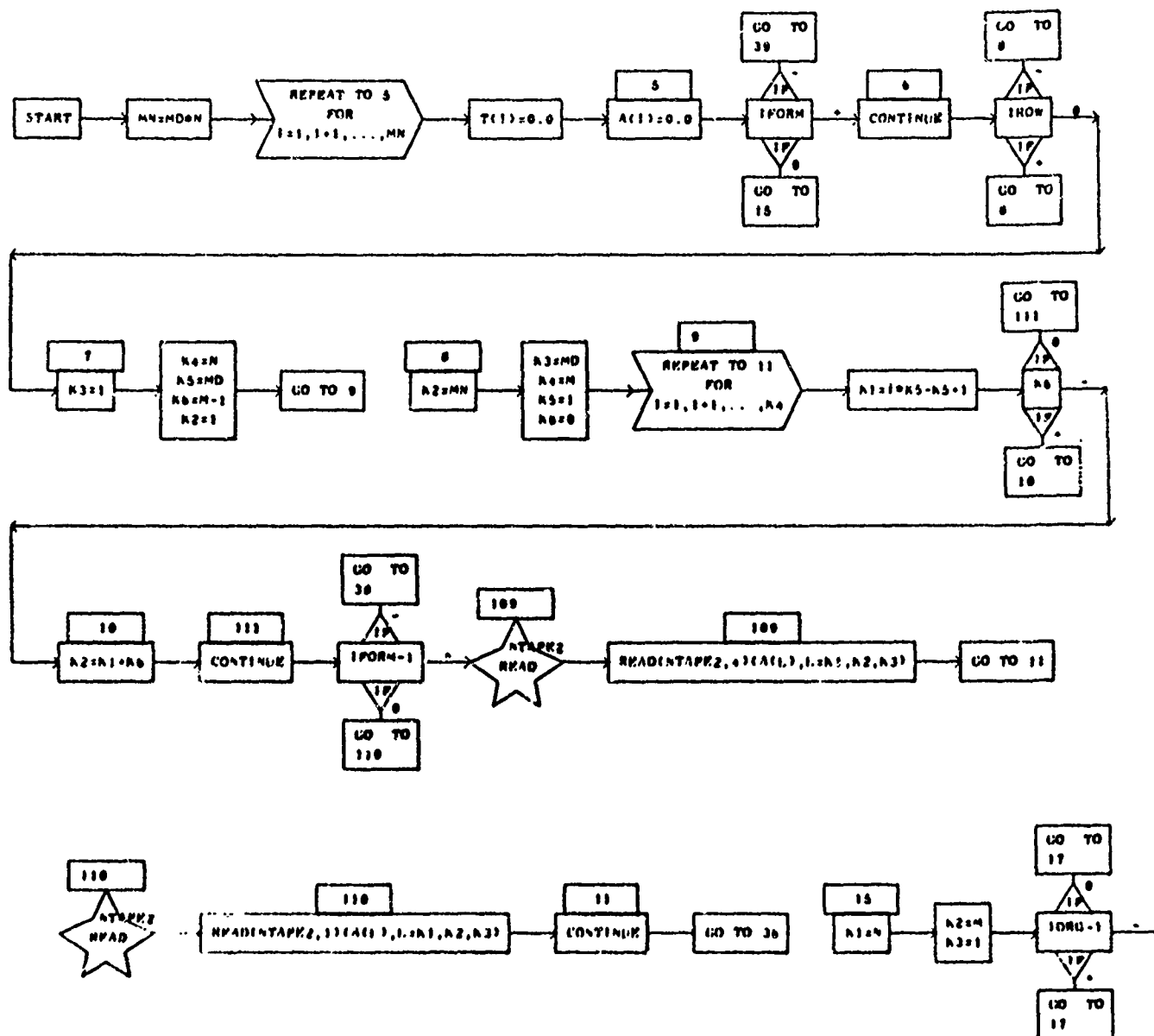
MATRIX READ SUBROUTINE

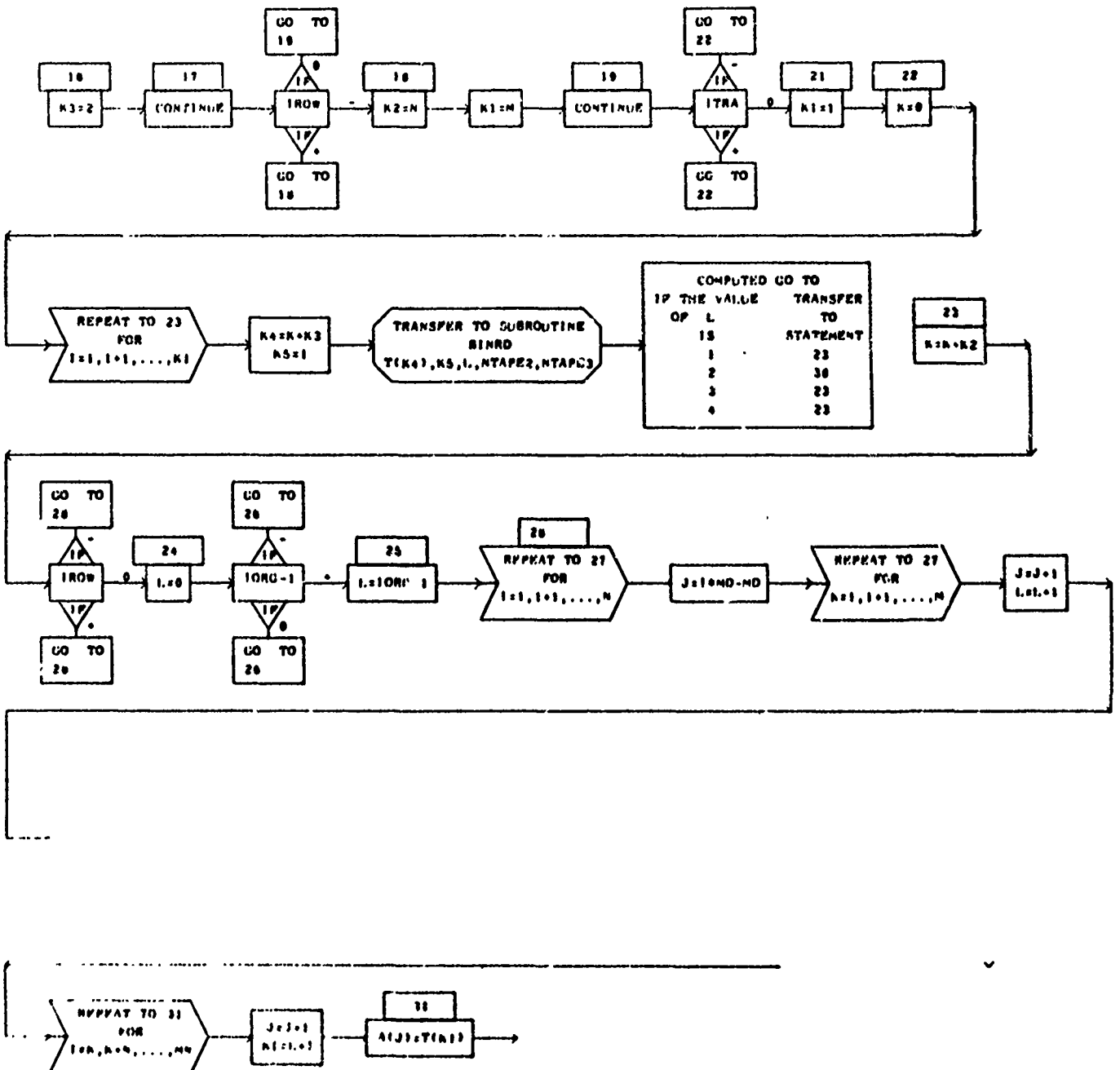
CALL MREAD (A,M,N,IFORM,IROW,ITHA,IORU,T,MO,NTAPE2,NTAPE3)

A = MATRIX TO READ IN ITHA = 0, TRA CARD AFTER MATRIX
M = NUMBER OF ROWS = +1, TRA CARD AFTER EACH ROW
N = NUMBER OF COLUMNS (OR COLUMN)
IFORM = -1, FORMAT(12A6) IORU = ORIGIN OF FIRST C.B. CARD
= 0, COLUMN BINARY T AND/OR TEMPORARY CELLS
= +1, FORMAT(5E12.6) MO = DIMENSIONED NUMBER OF ROWS
IROW = .0, MATRIX BY COLUMNS IN A
= +1, MATRIX BY ROWS NTAPE2 = INPUT TAPE
NTAPE3 = OUTPUT TAPE

D I M E N S I O N E D V A R I A B L E S

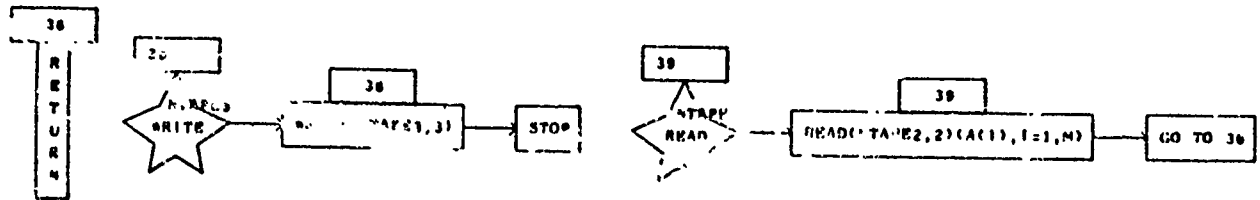
SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
A	1	T	1						





SUBROUTINE XREAD (A,M,N,IFORM,IROW,ITRA,IORU,T,ND,NTAPE2,NTAPE3)

PAGE 3



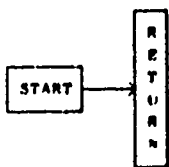
RINRO

D I M E N S I O N E D V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
T	1								

SUBROUTINE RINRO (T,A,L,NTAPE1,NTAPE2)

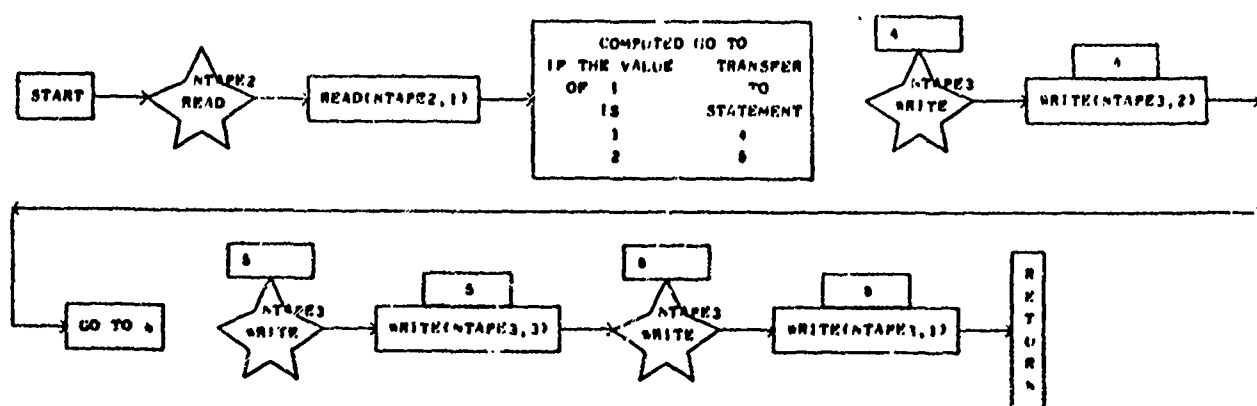
PAGE 1



KOLN

SUBROUTINE KOLN (NTAPE2, NTAPE3, 1)

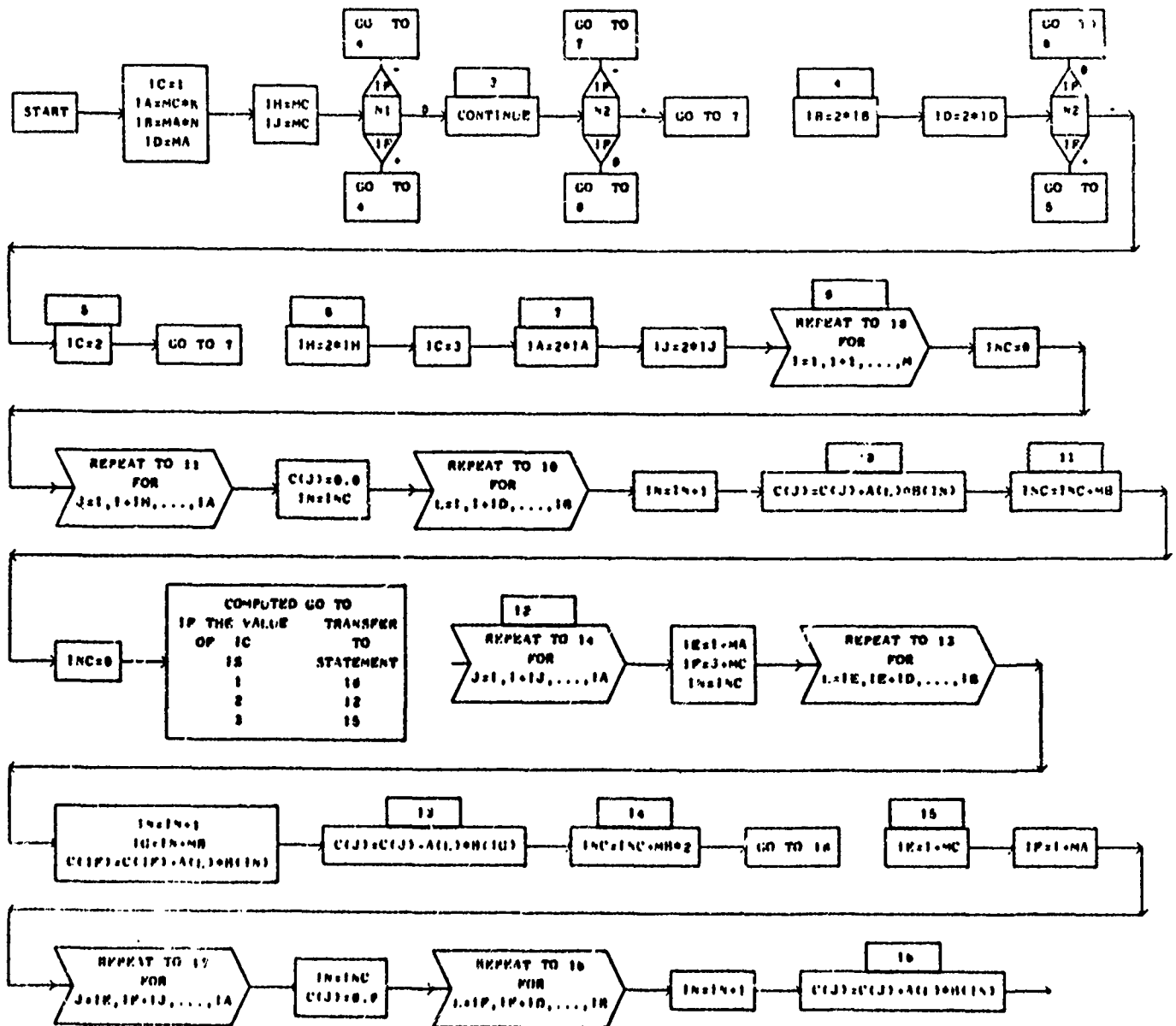
PAGE 1



MM/LTD

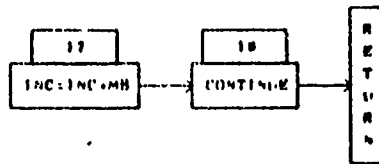
D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
A	1	B	1	C	1				



SUBROUTINE MMULTD (A,N1,R,N2,C,N,N,K,MA,MB,MC)

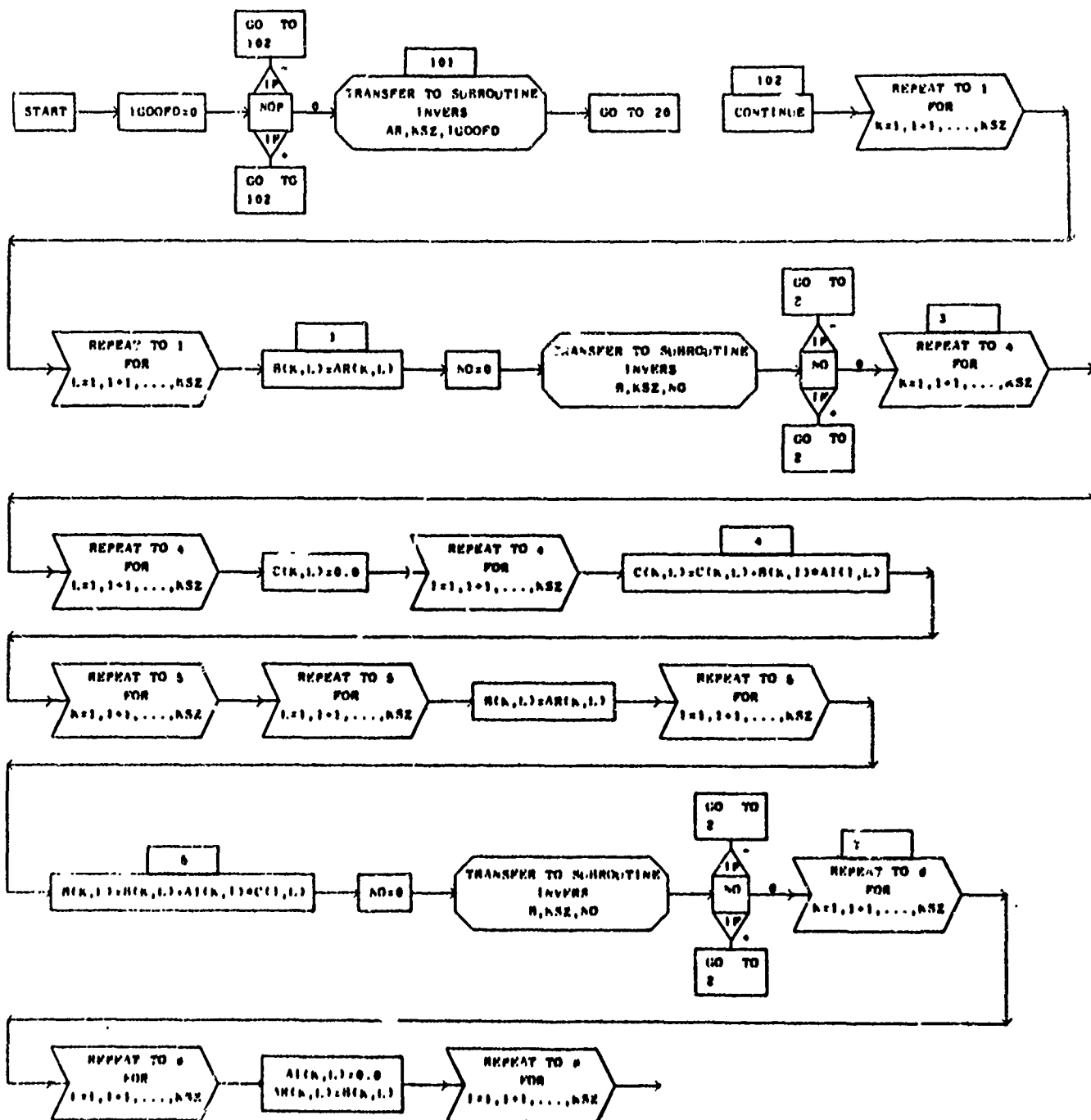
PAGE 2



MMVRSR

D I M E N S I O N E D V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
AR	0,0	AI	0,0	B	0,0	C	0,0		





SUBROUTINE MNVRSX (AR,A1,B,C,KSZ,IGCOFD,NOP)

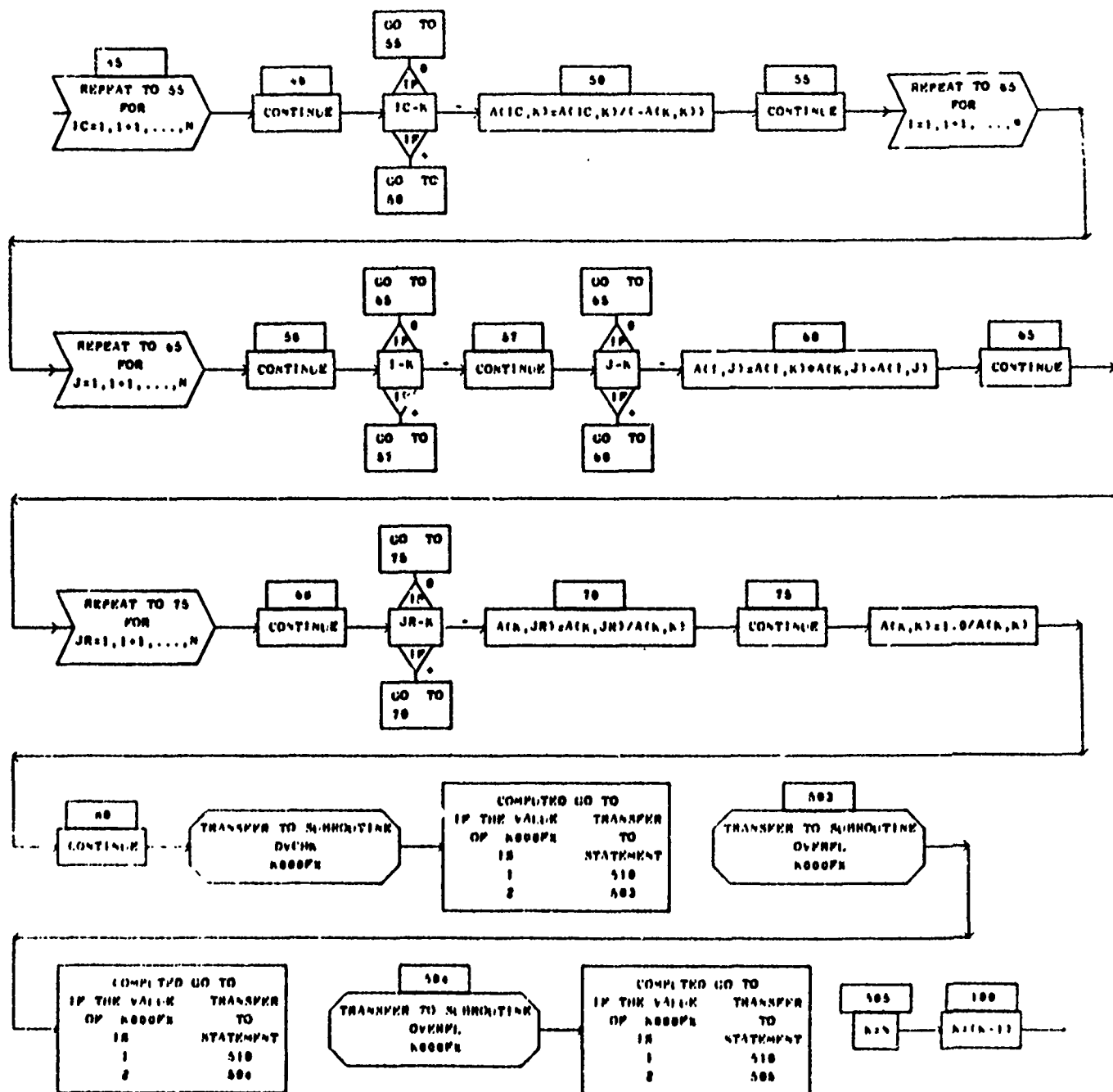
PAGE 3

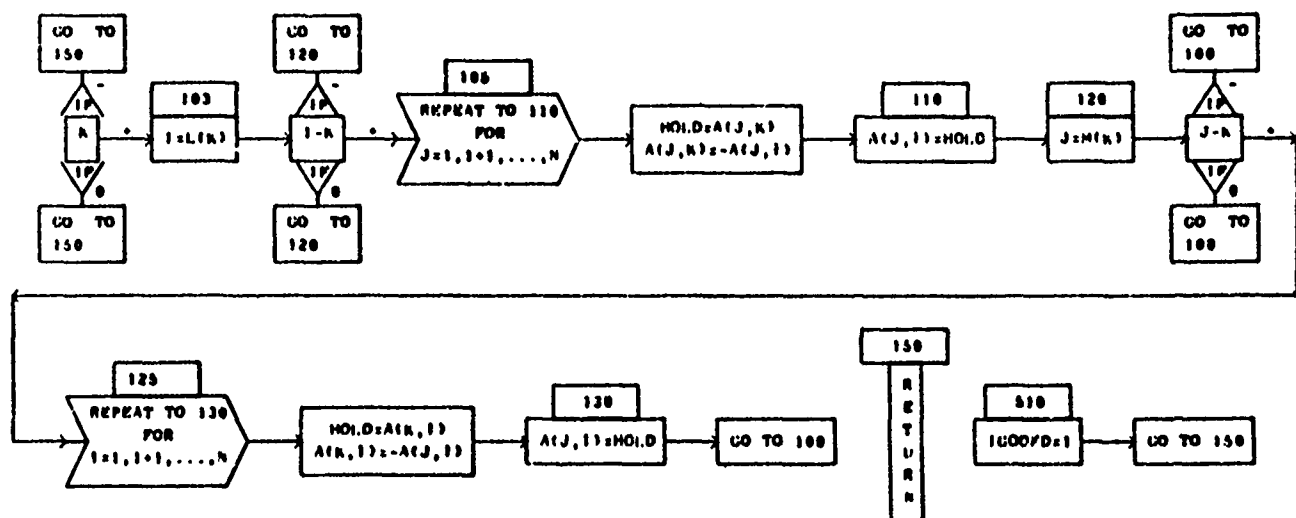
20
R
E
T
U
R
N

INVERS

D I M E N S I O N E D V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
A	0,0	L	0	M	0				

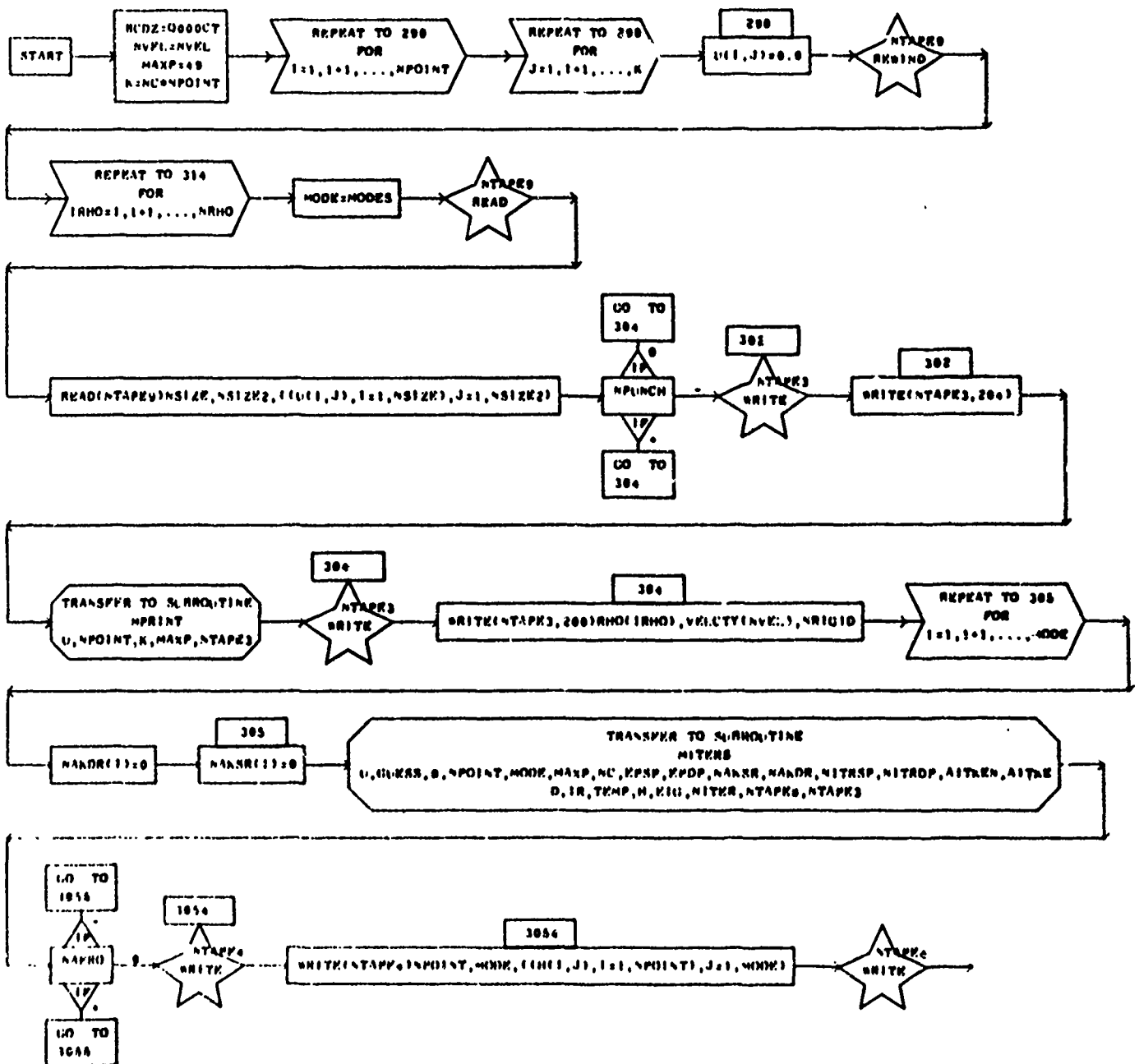




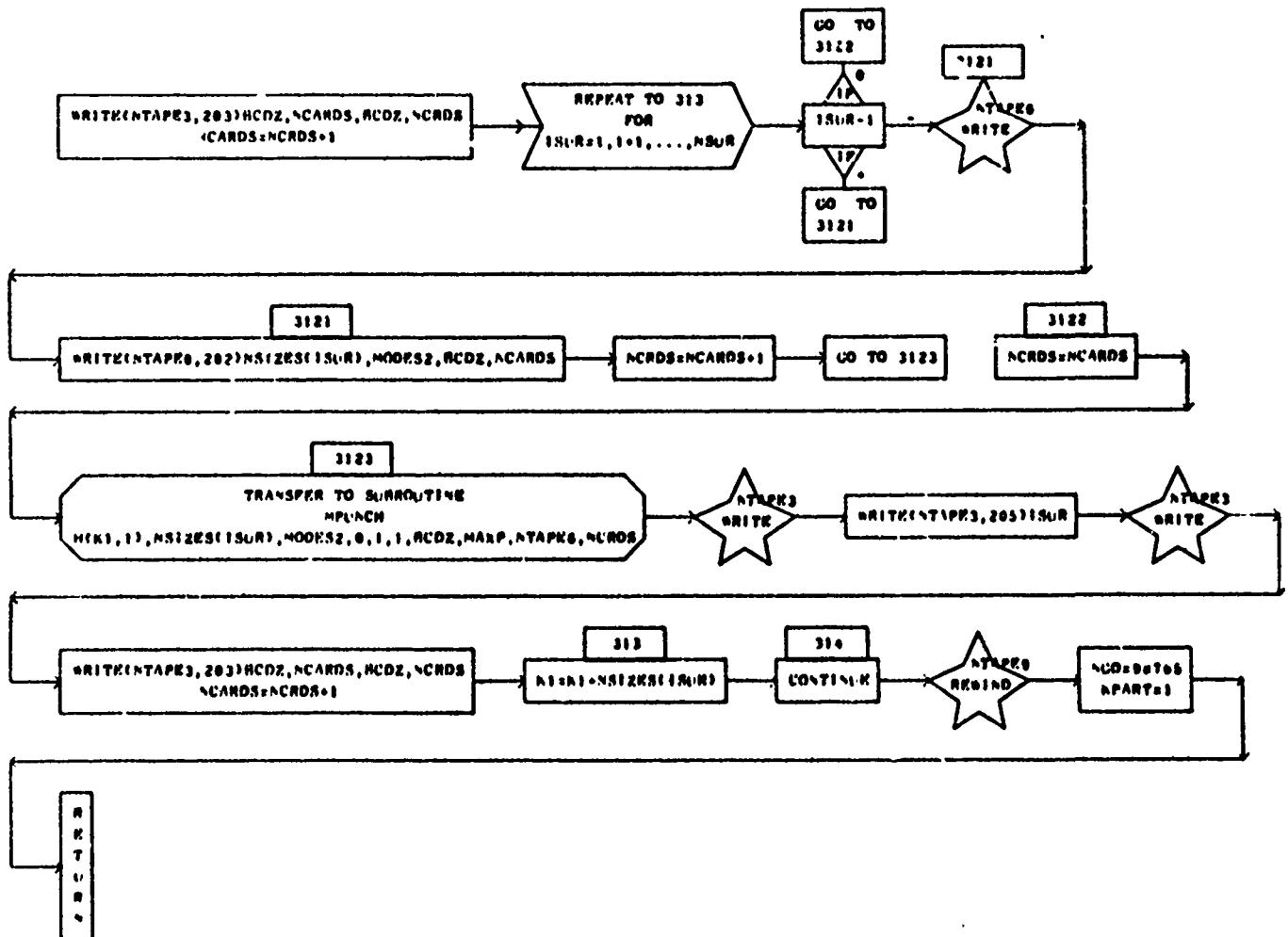
PART2

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
IT	216	VFLCTV	20	NSIZES	20	DM	6,6	RHO	20



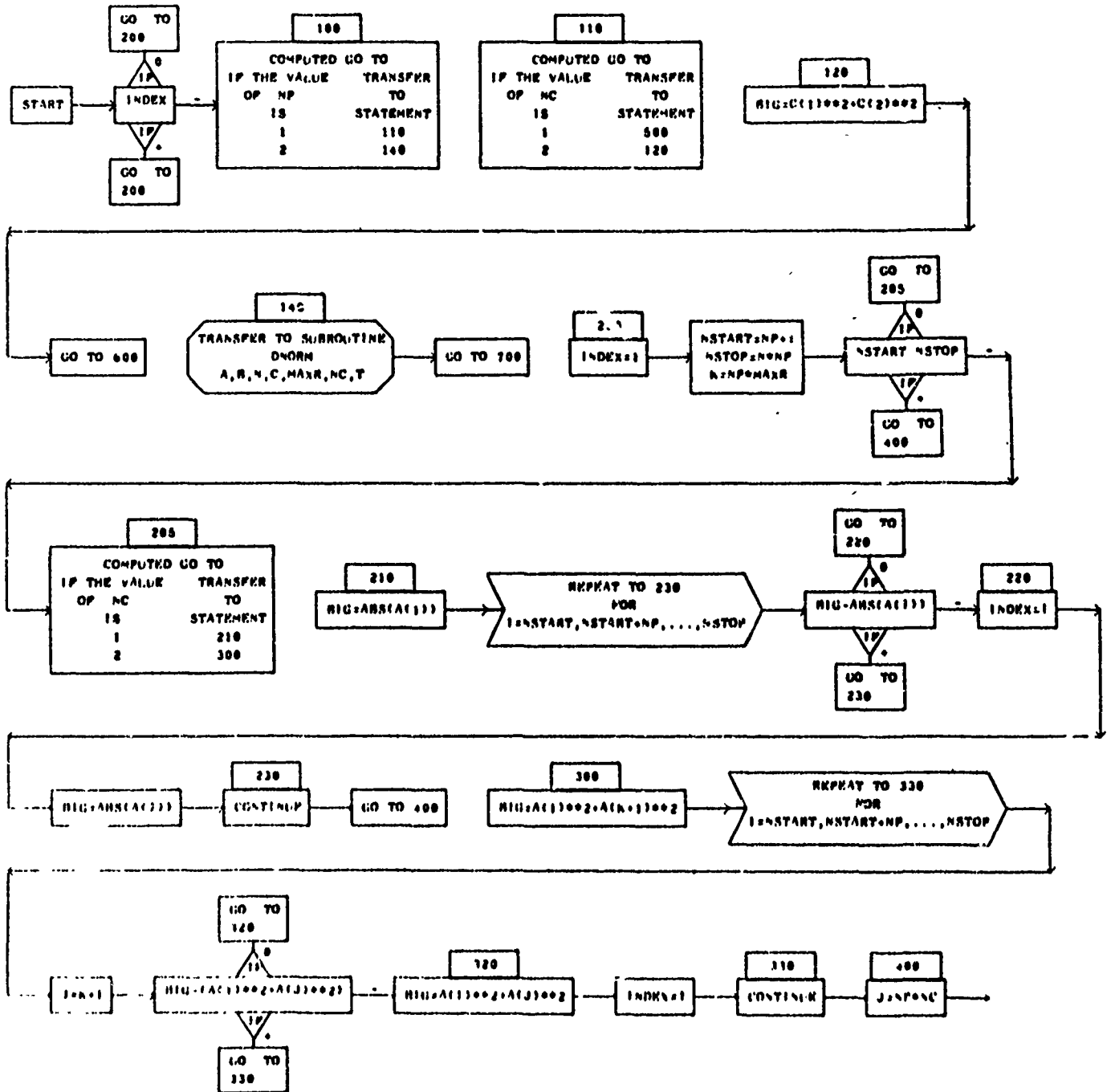


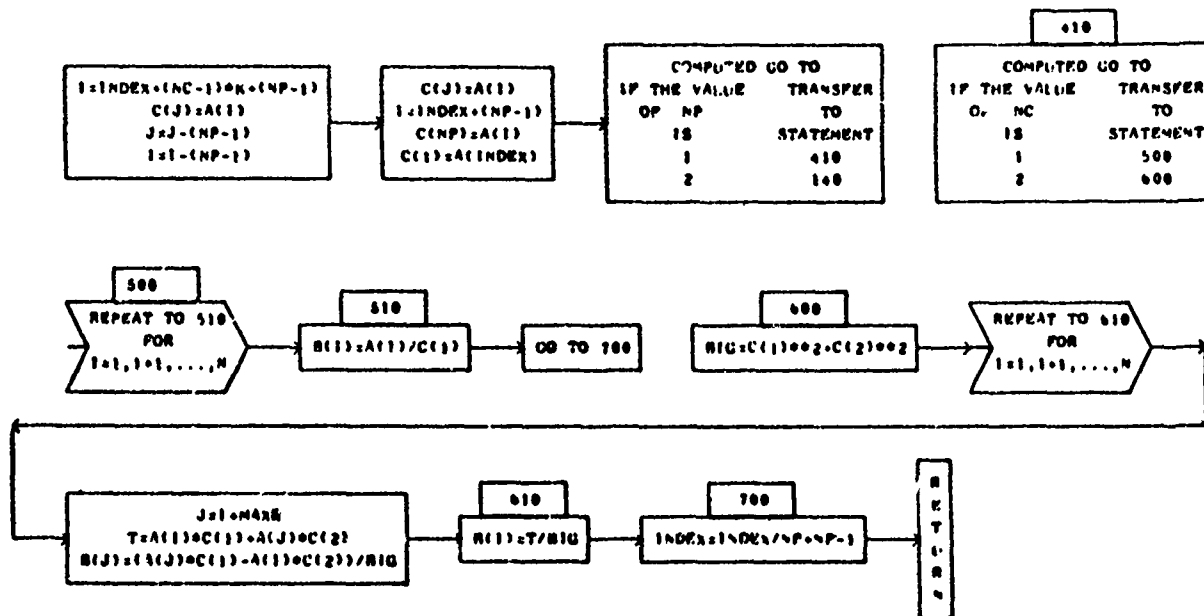


NAME

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	B	1	C	1	T	2		

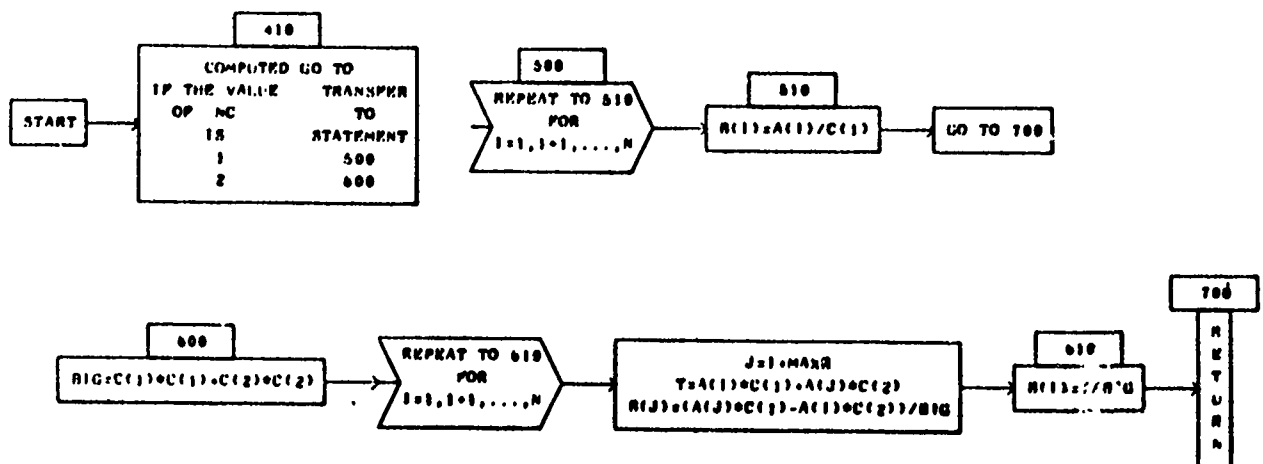




DNORME

SUBROUTINE DNORM (A,B,N,C,MAXR,NC,T)

PAGE 1

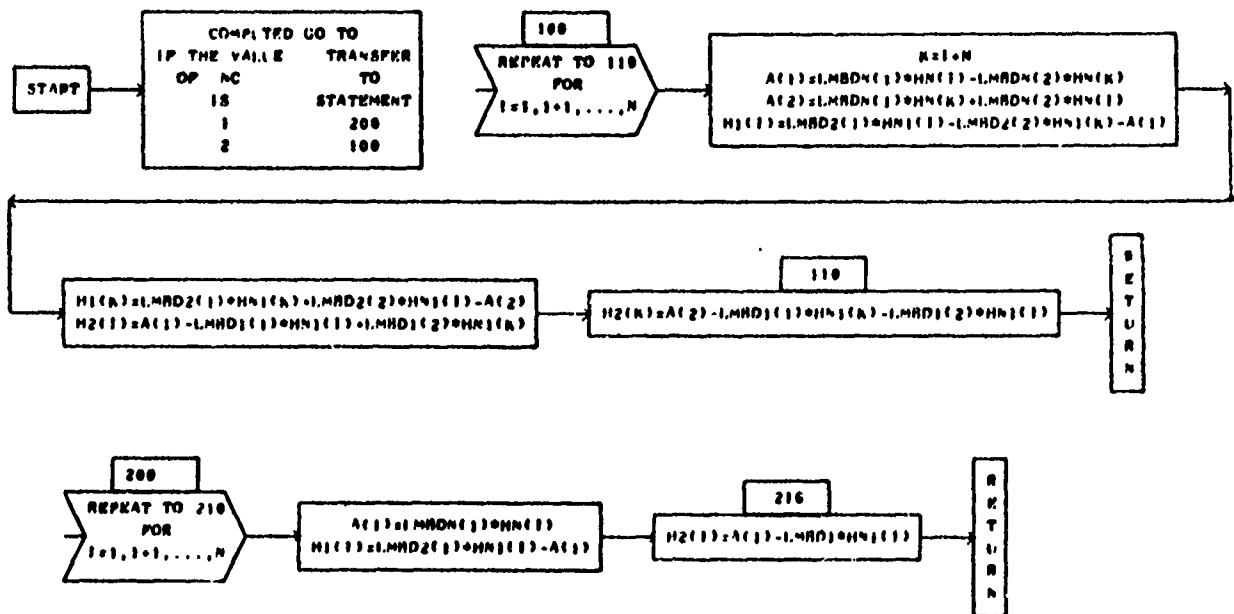


D I M E N S I O N E D V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
LMRDN	1	LMRD1	1	LMRD2	1	MN	1	MN1	1
M1	1	M2	1	A	2				

SUBROUTINE POH (LMRDN,LMRD1,LMRD2,MN,MN1,M1,M2,N,NC)

PAGE 1

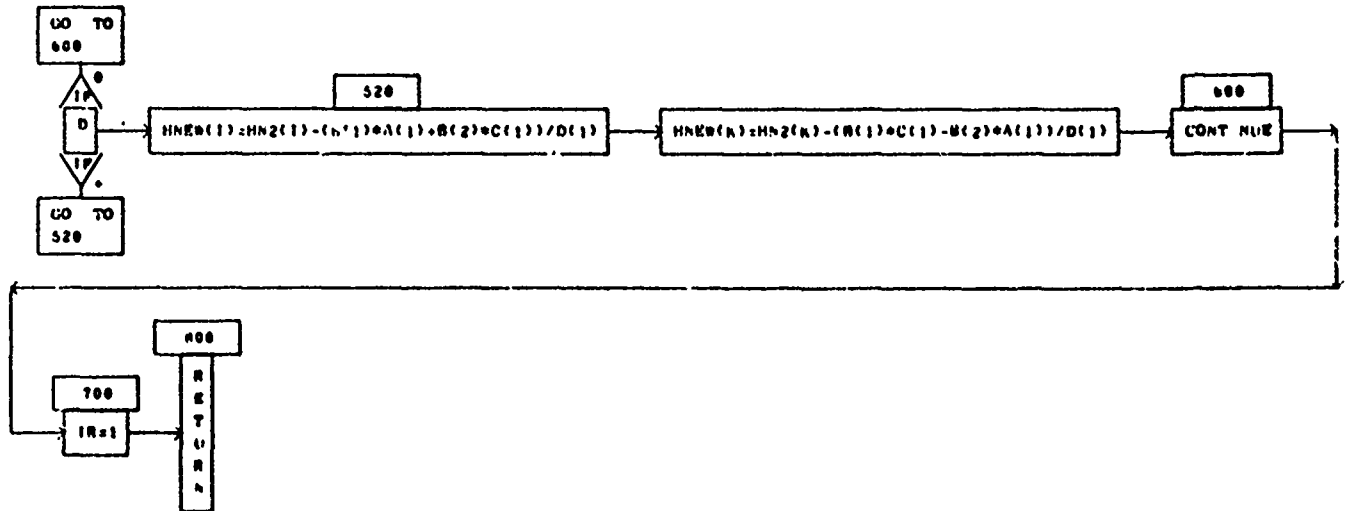


ATKNS

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
MM	1	MM1	1	MM2	1	MM2B	1	A	2
B	4	C	2	D	2				

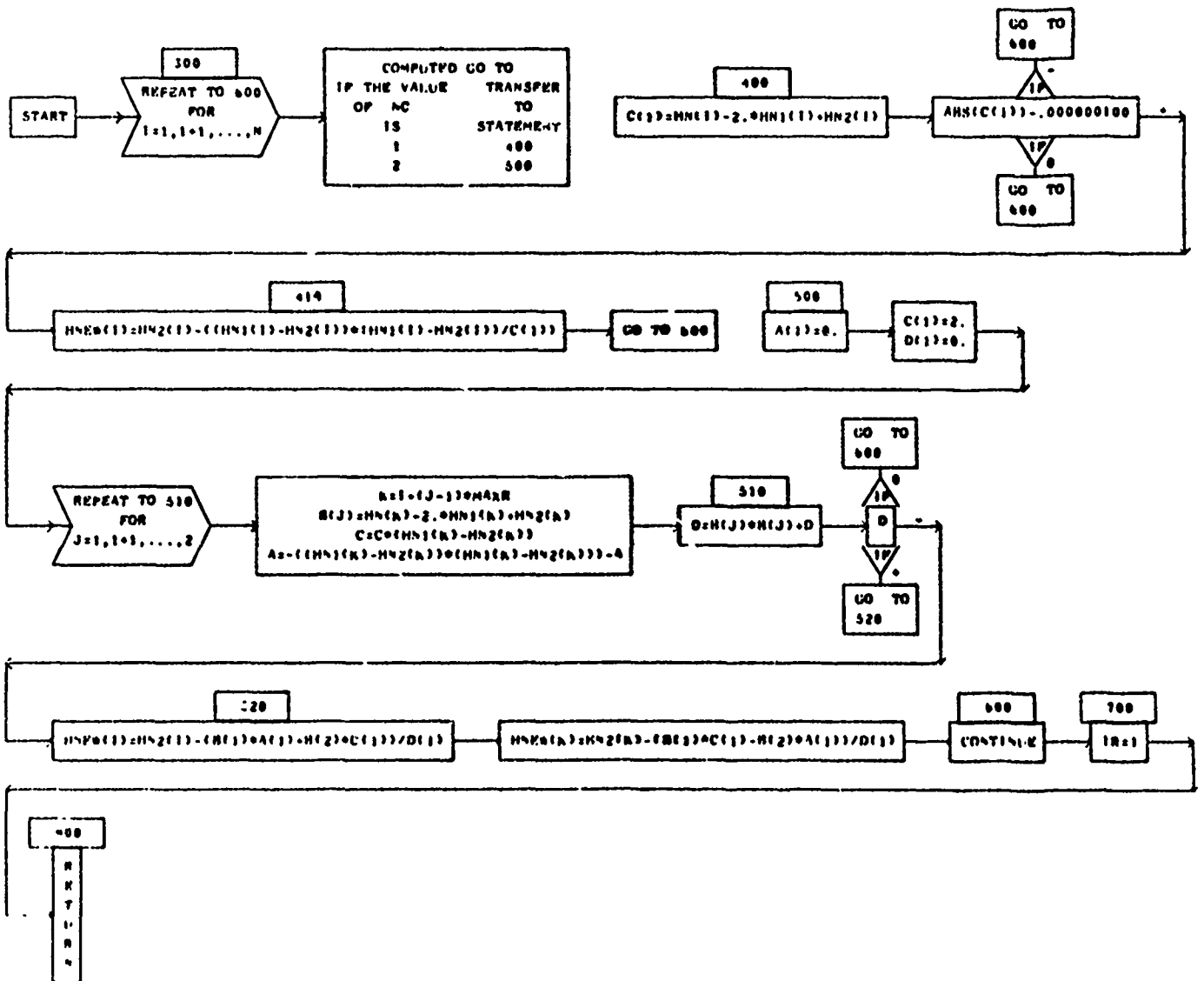




ATEND

D I M E N S I O N / O V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
HN	1	HN1	1	HN2	1	HN2W	1	A	1
B	2	C	1	D	1				



1.20155

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
LAMBDA	1	PDS	1						

MADE

SUBROUTINE MADE (A,B,C,NSIZE,NC)

PAGE 1



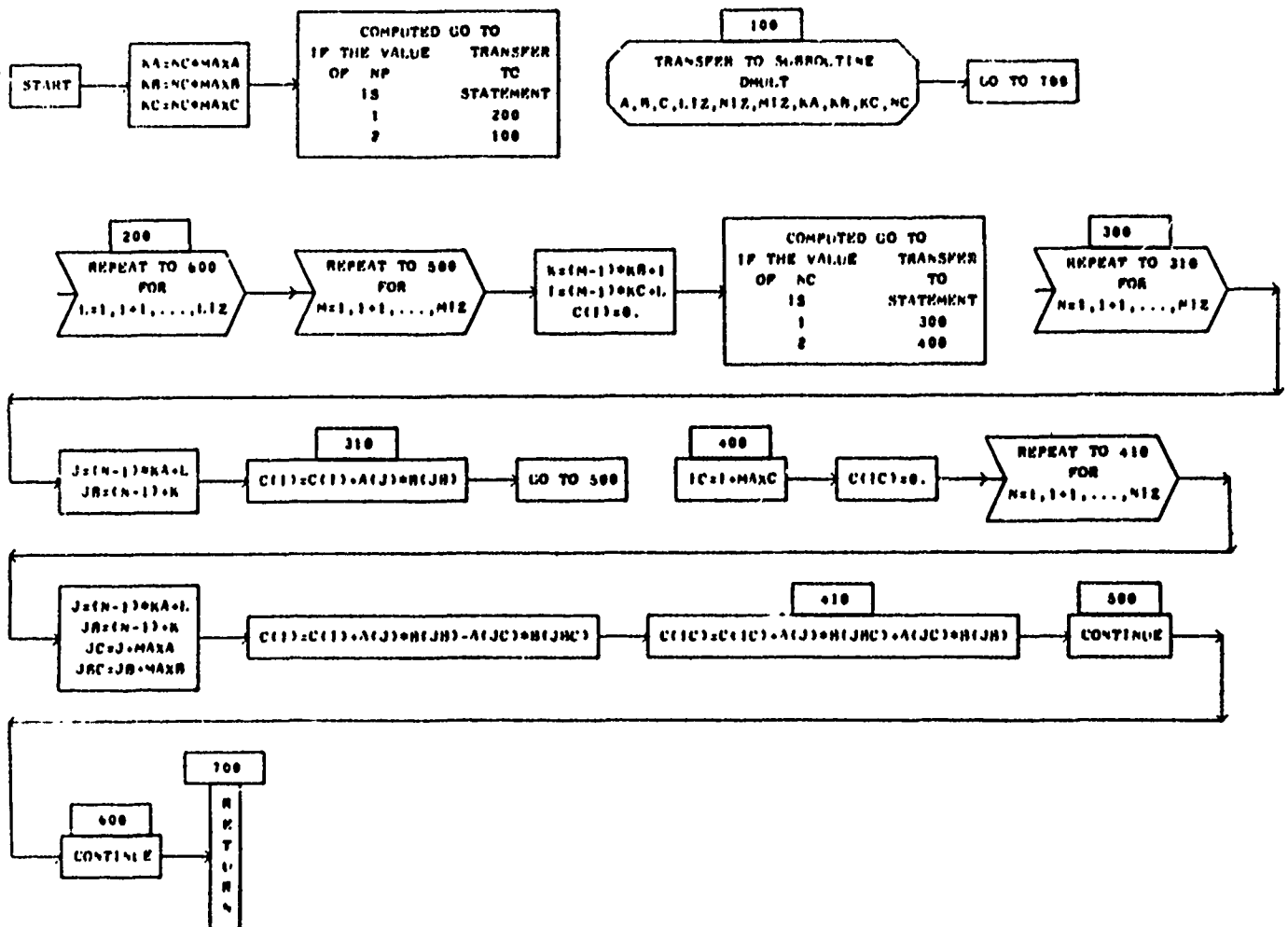
MULTS

D I M E N S I O N E D V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	B	1	C	1				

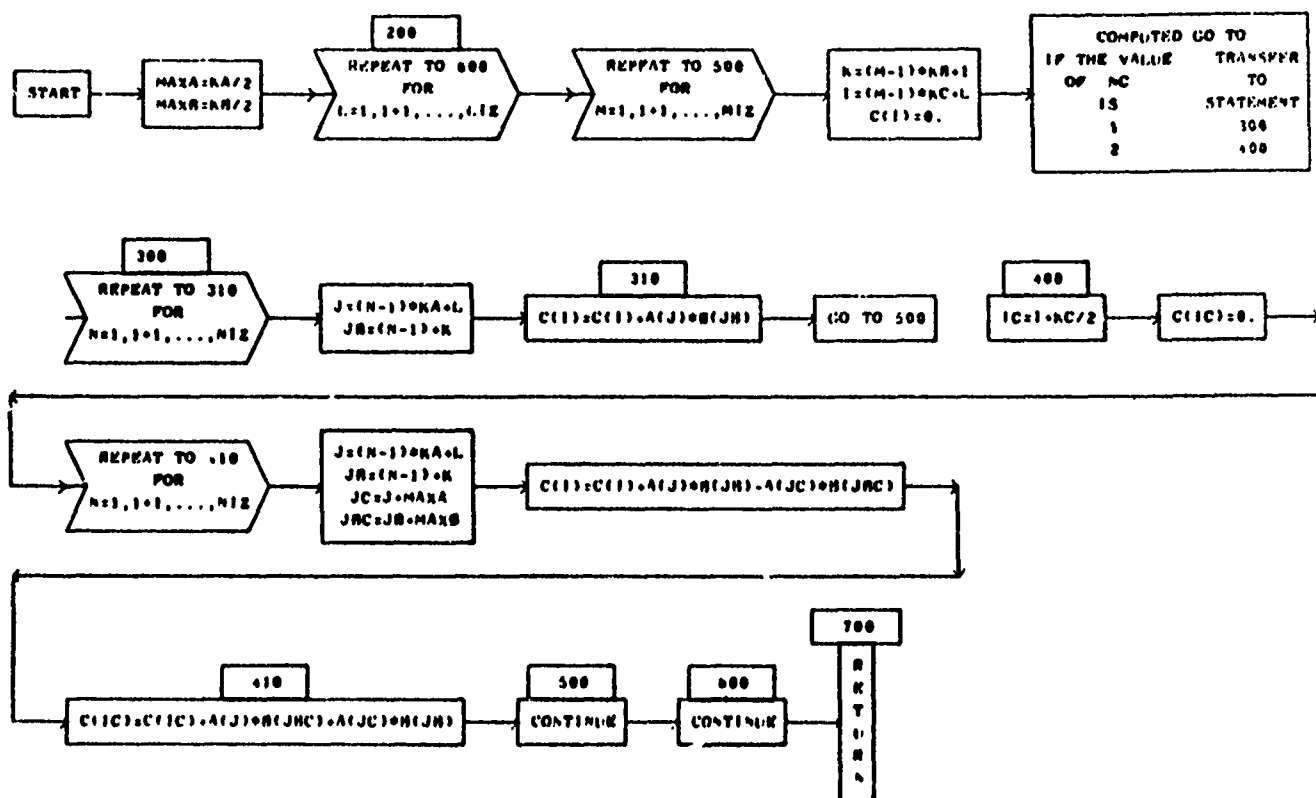
SUBROUTINE MULT (A,B,C,I1Z,M1Z,M1Z,MAXA,MAXB,MAXC,NC,NP)

PAGE 1



1

PAGE 1

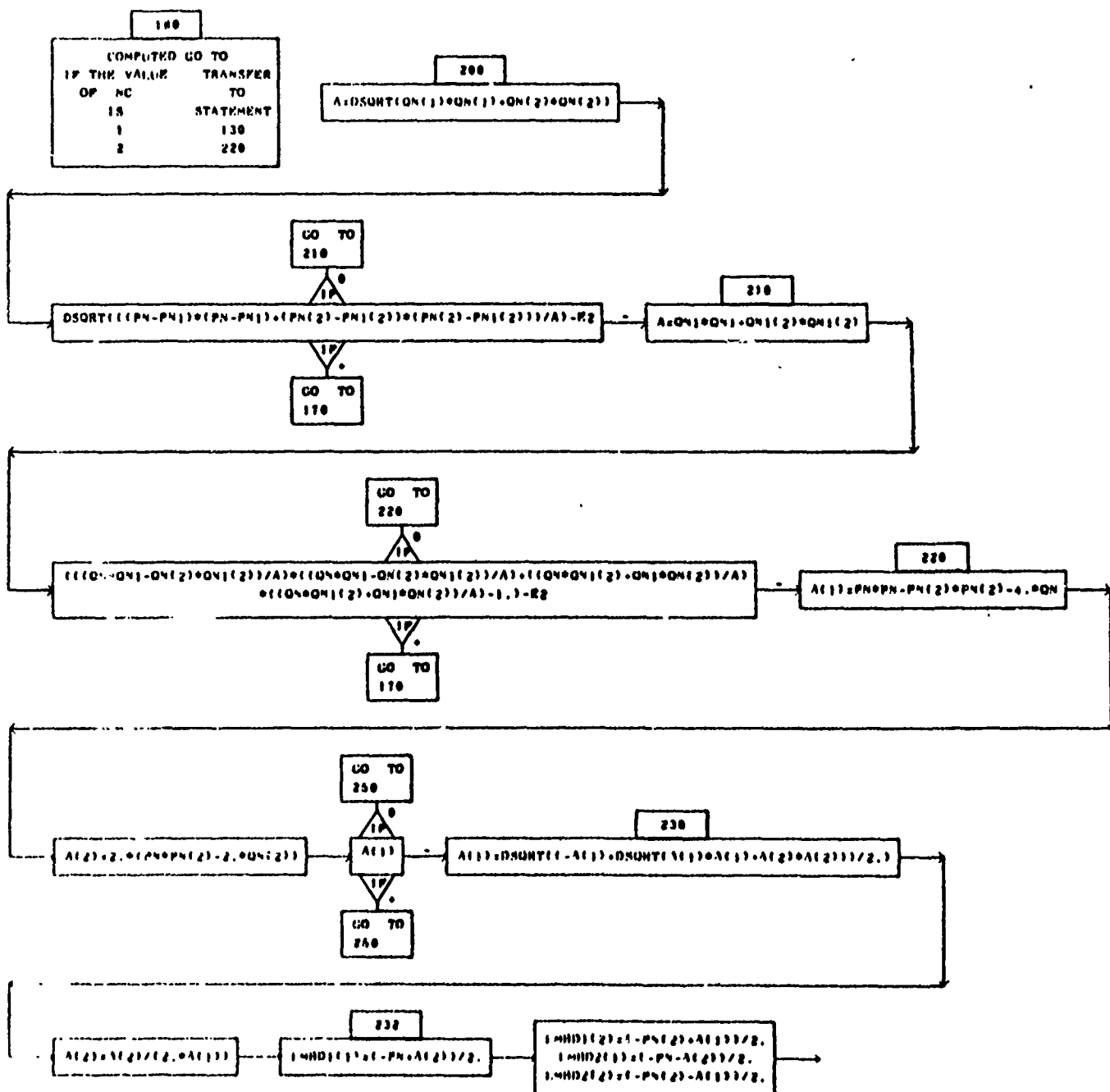


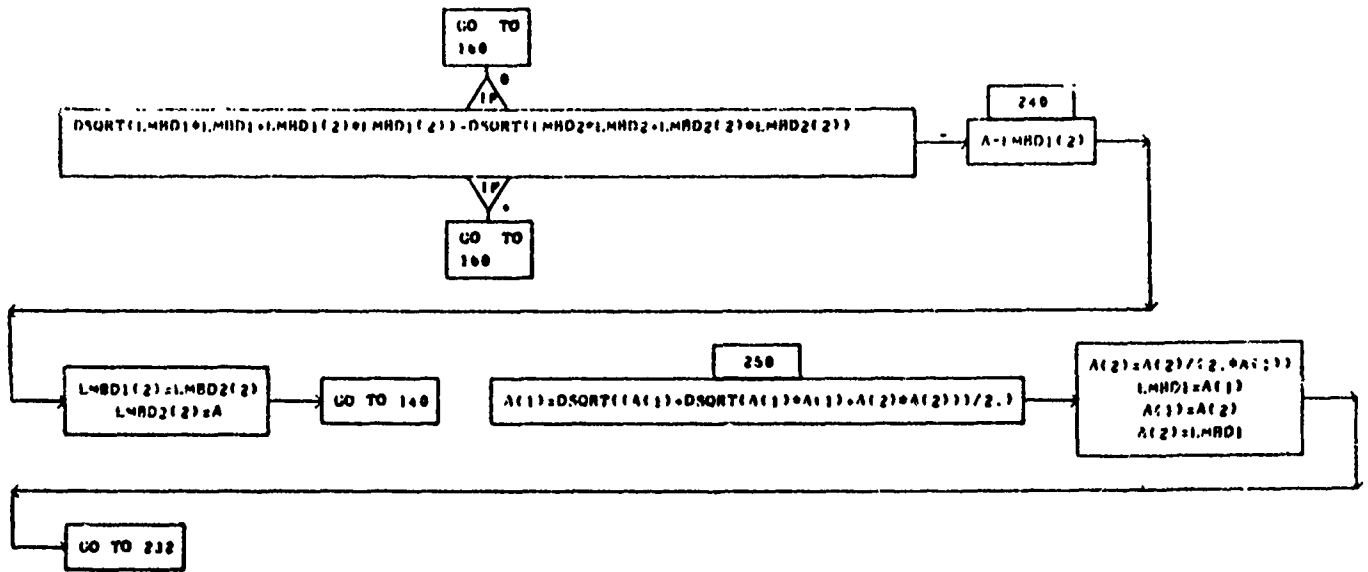
POLMS

D I M E N S I O N E D V A R I A B L E S

SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
PH	1	PH1	1	ON	1	ON1	1	E2	1
IMRD1	1	IMRD2	1						







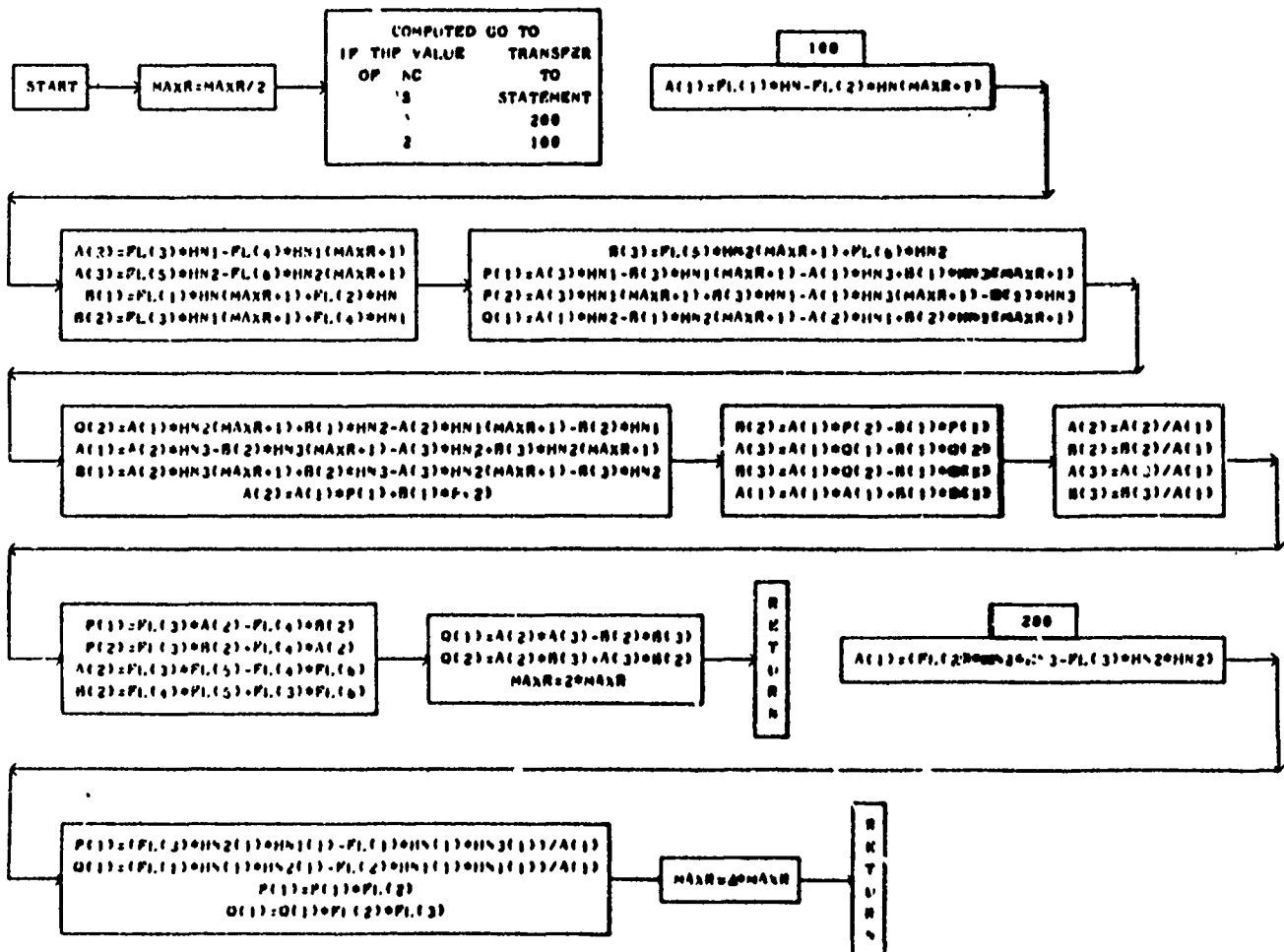
PUS

D I M E N S I O N E D V A R I A B L E S

7/MNOI.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES	SYMBOL.	STORAGES
PL	1	MN	1	MN1	1	MN2	1	MN3	1
P	1	Q	1	A	3	R	3		

SUBROUTINE PG (PL, MN, MN1, MN2, MN3, P, Q, NC, MAXR)

PAGE 1



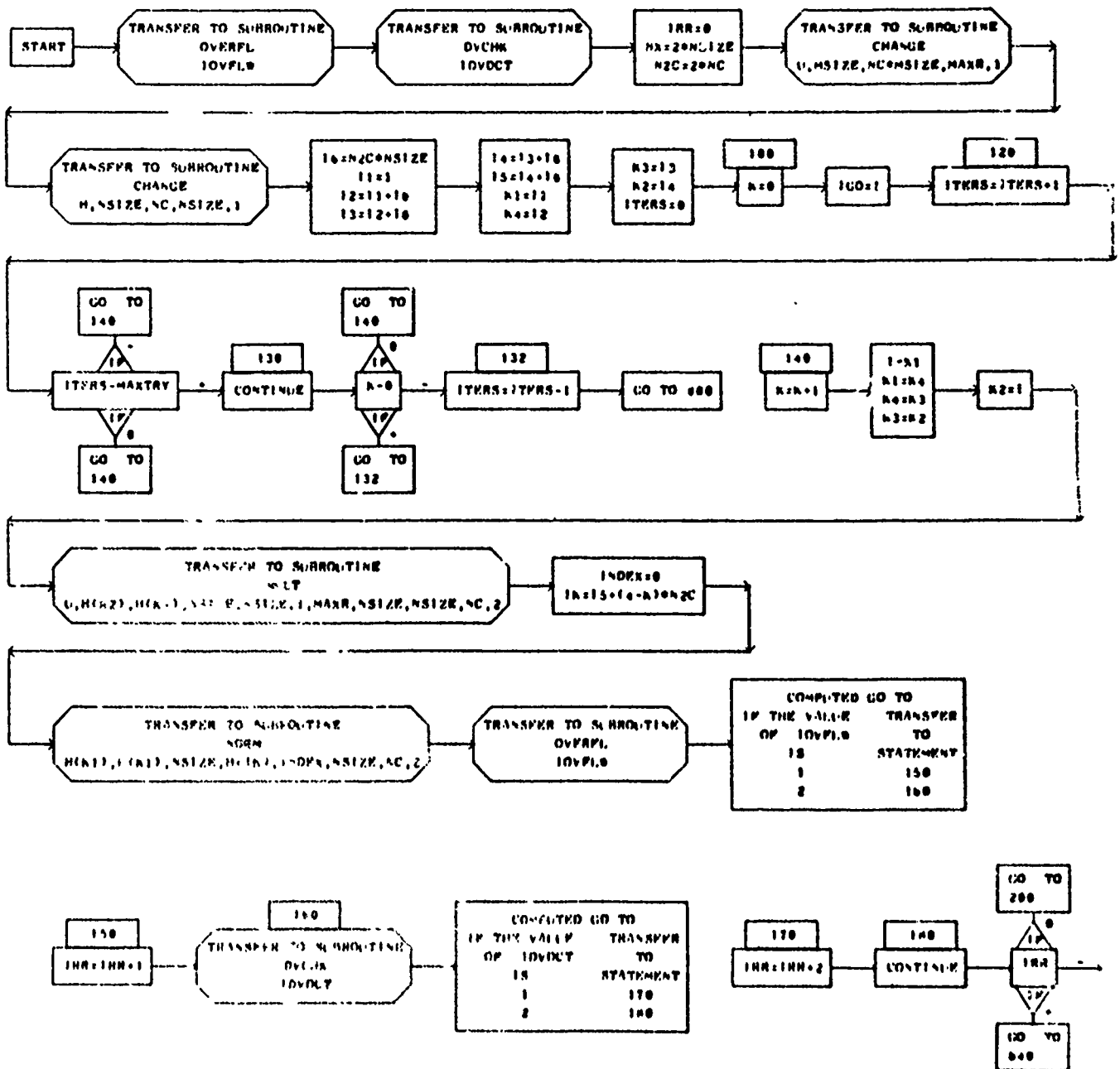
CLOSES

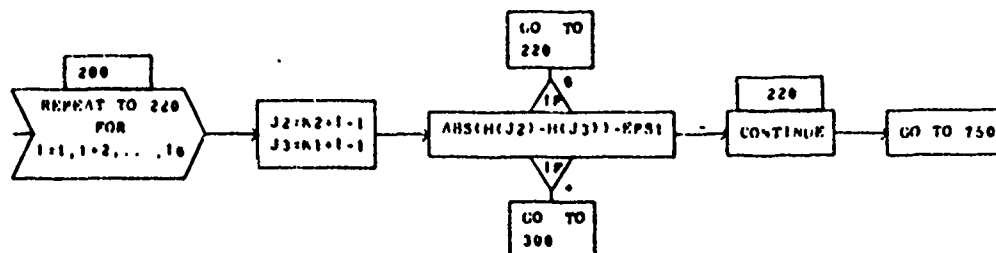
SUBROUTINE CLOSE, COMPUTES 2 CLOSE ROOTS.

U = MATRIX, DIMENSIONED (MAXR,2*NC*MAXR)
M = STARTING GUESS, DIMENSIONED (MAXR,2*NC*4)+2*NC*4)
NSIZE = SIZE OF MATRIX
MAXR = DIMENSIONED NUMBER OF ROWS OF U AND M
MAXTRY = MAXIMUM NUMBER OF DOUBLE PRECISION ITERATIONS.
EPS1 = SINGLE ROOT CONVERGENCE CRITERIA
EPS2 = DOUBLE ROOT CONVERGENCE CRITERIA
R = AITKEN'S CONVERGENCE CRITERIA
IRR = ERROR RETURN INDICATOR. =1, OVERFLOW
=2, DIVIDE CHECK
=3, BOTH OVERFLOW AND DIVIDE
CHECK.
ITERS = NUMBER OF ITERATIONS PERFORMED, = FOR DOUBLE ROOT
= FOR SINGLE ROOT
NC = 1, REAL. 2, COMPLEX

D I M E N S I O N E D V A R I A B L E S

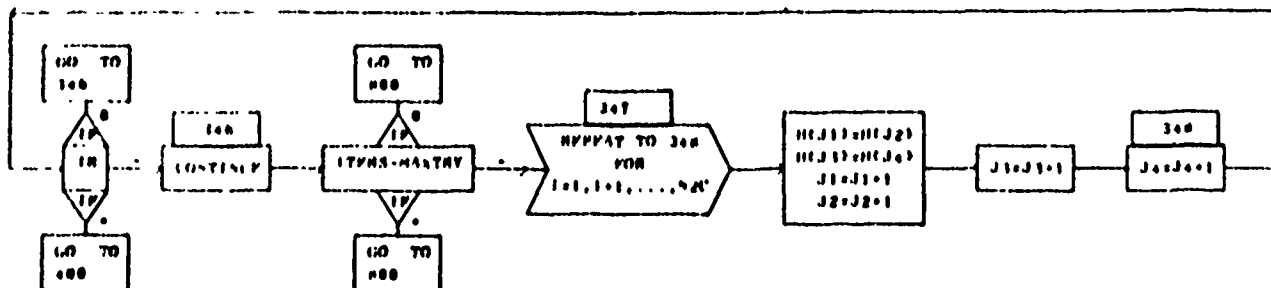
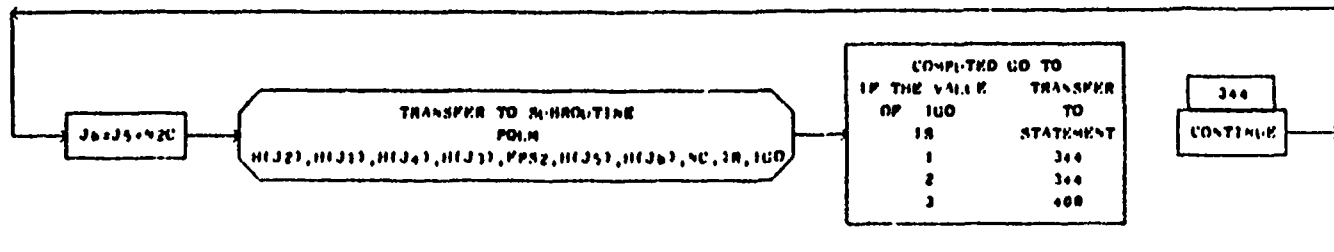
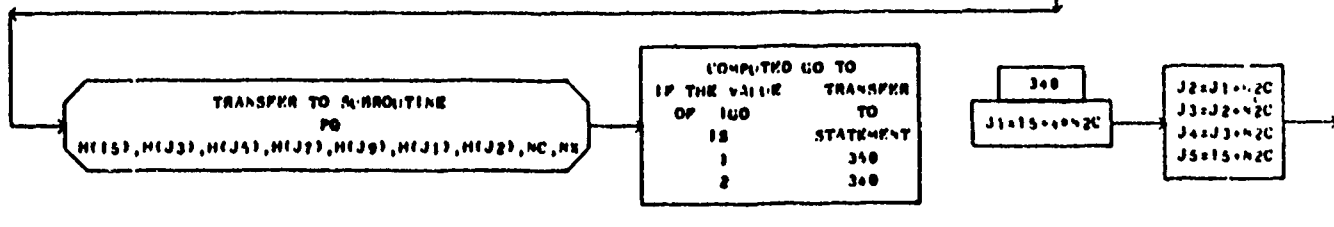
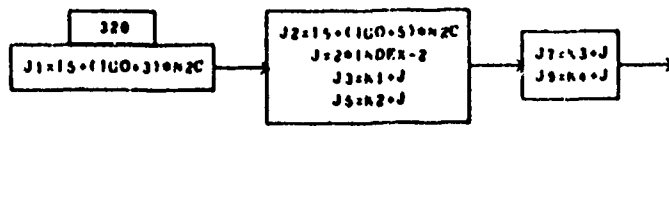
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
U	1	M	1	VALUE	1				

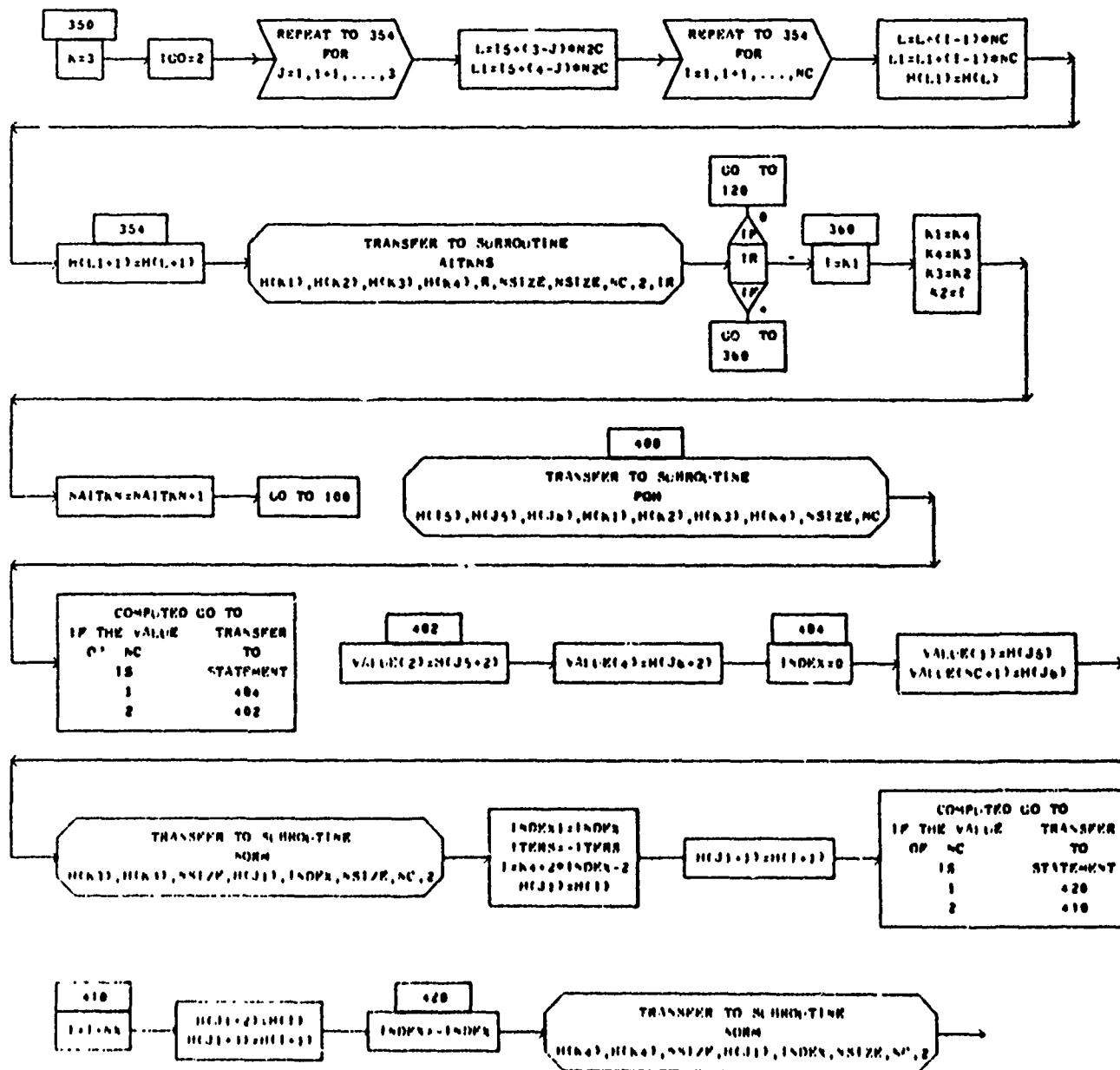




300

COMPUTED GO TO		
IF THE VALUE	TRANSFER	
OF	TO	
IS	STATEMENT	
1	120	
2	120	
3	120	
4	320	

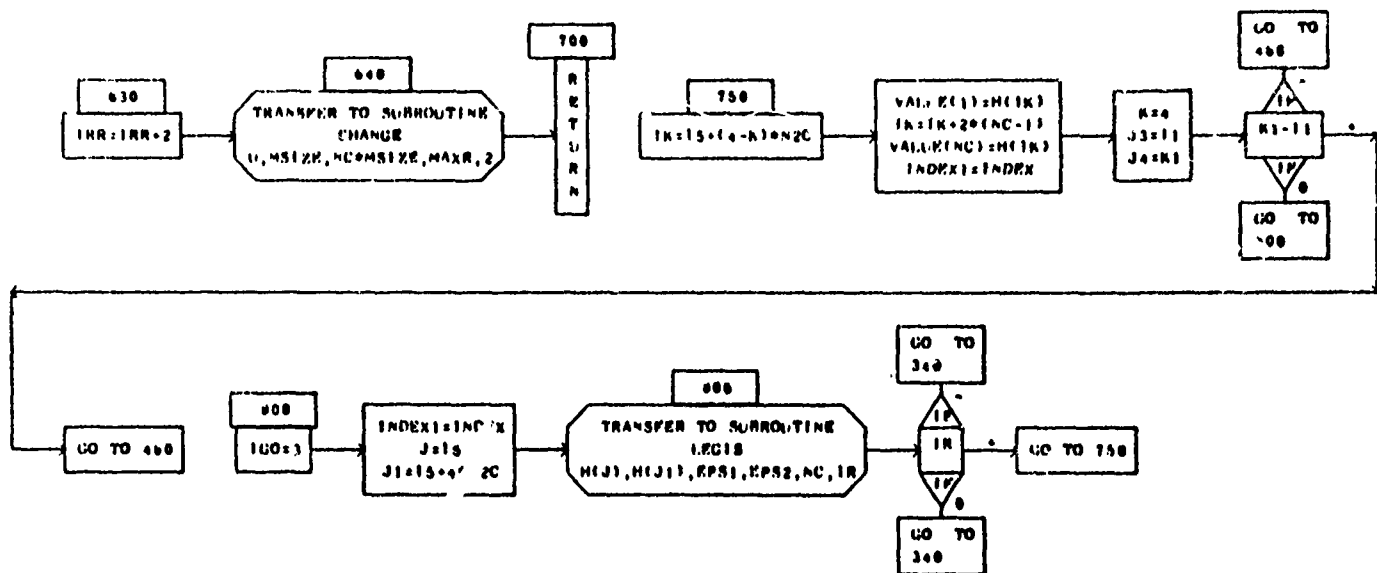






SUBROUTINE CLOSES (U,H,NSIZE,MAXR,R,EPS1,EPS2,NC,IRR,MAXTRY,ITERS,

PAGE 3



METERS

A IS STORED IN CORE AT A. (MAXR X NCNPNNSIZE)

NTAPUT IS A UTILITY TAPE, FOR CHECK VECTORS IF DESIRED.

KPSF = EPSILON ONE = SINGLE PRECISION CONVERGENCE TEST NUMBER

KDPF = EPSILON TWO = DOUBLE PRECISION

NC = 1, IF REAL NP = 1, IF SINGLE PRECISION

2, IF COMPLEX = 2, IF DOUBLE ..

NGUESS = 0, IF FIRST GUESS IS TO BE A COLUMN OF ONES.

MODOUT = NO. OF MODES TO BE COMPUTED.

NAKSR = NO. TIMES AITENS ACCELERATION WAS USED IN SINGLE PRECISION.

NANDR = ... DOUBLE .. .

MAXSR = MAXIMUM ITERATIONS ALLOWED IN SINGLE PRECISION.

NANDR = ... DOUBLE .. .

ERR = ERROR RETURN = 1, FOR OVERFLOW

2, FOR DIVIDE CHECK

3, FOR BOTH OVERFLOW AND DIVIDE CHECK

NSIZE = NO. OF ROWS AND COLUMNS OF A

RSP = R, AITENS ACCELERATION CONVERGENCE CONTROL, FOR SINGLE PRECISION.

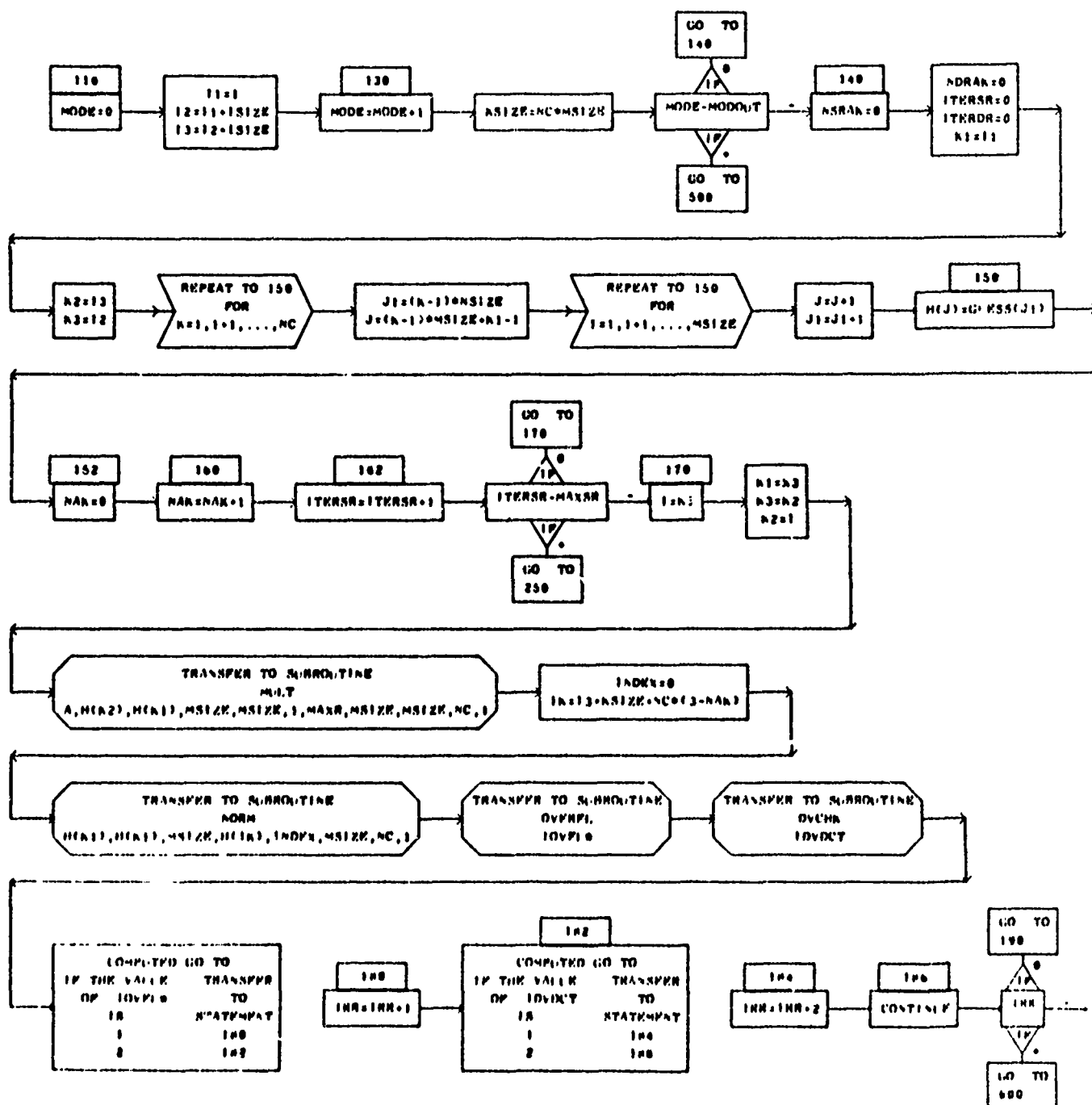
RDP = R AITENS ACCELERATION CONVERGENCE CONTROL, FOR DOUBLE PRECISION.

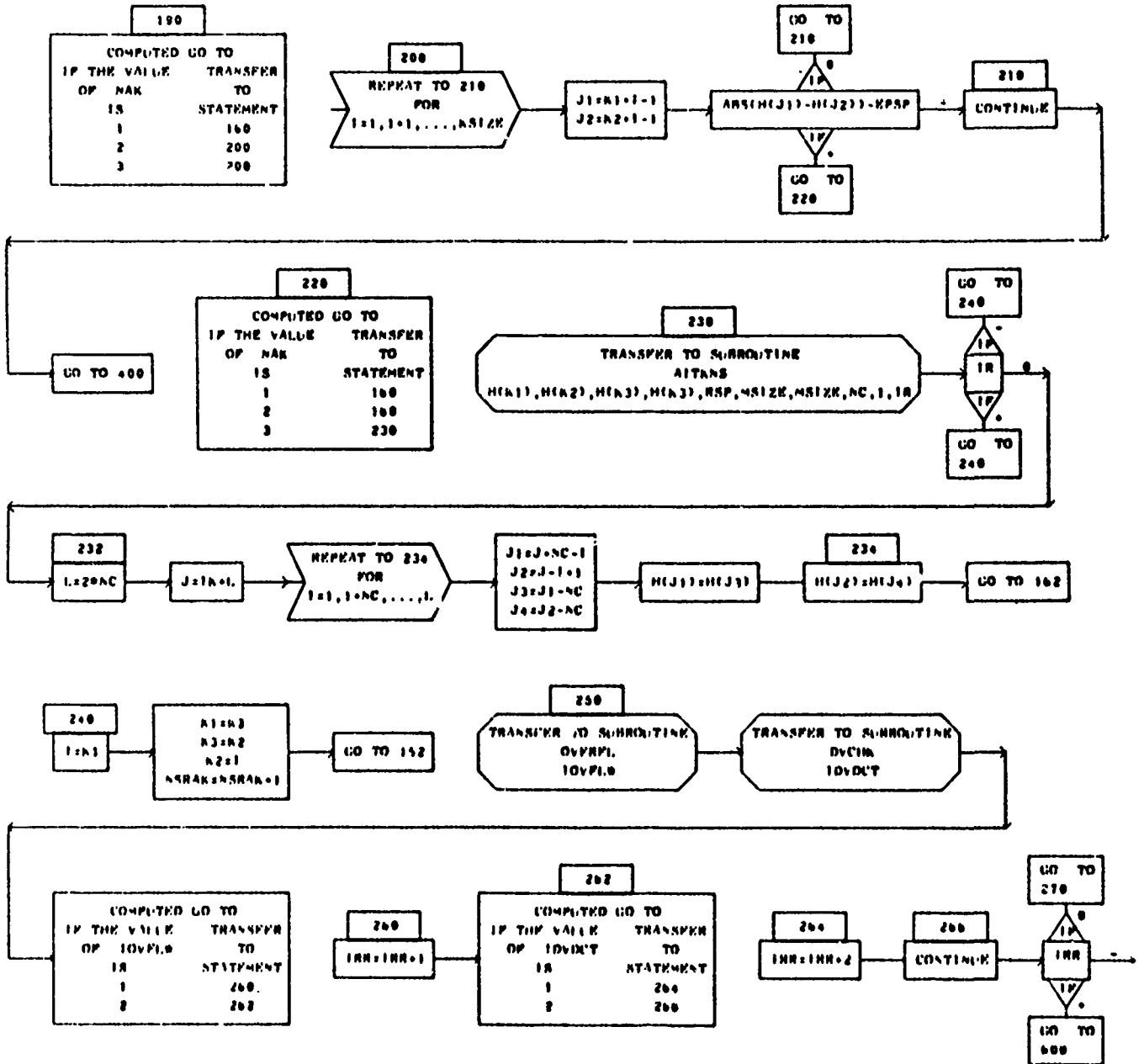
MAXR = DIMENSIONED NUMBER OF ROWS OF A AND GUESS

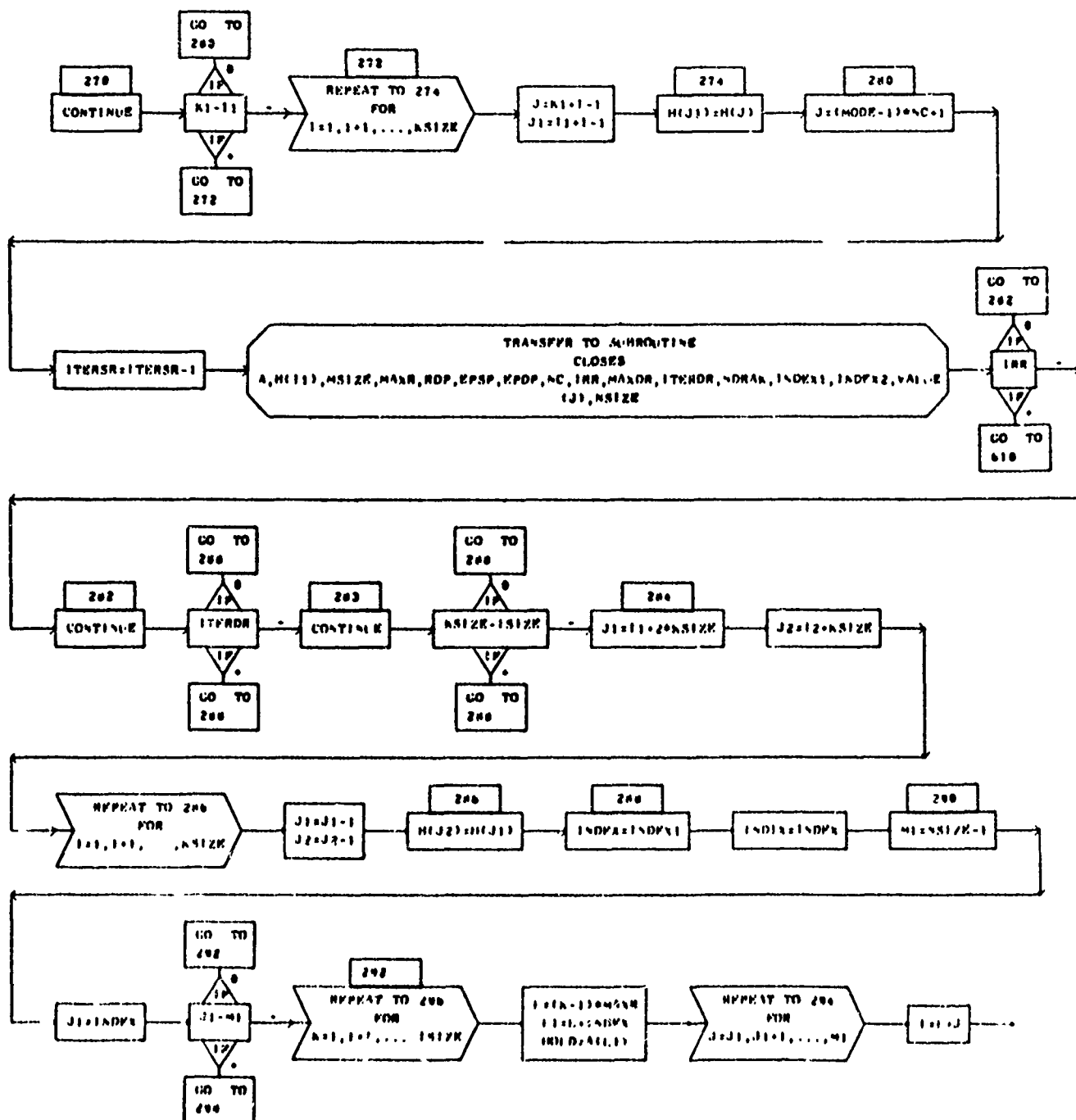
D I M E N S I O N E D V A R I A B L E S

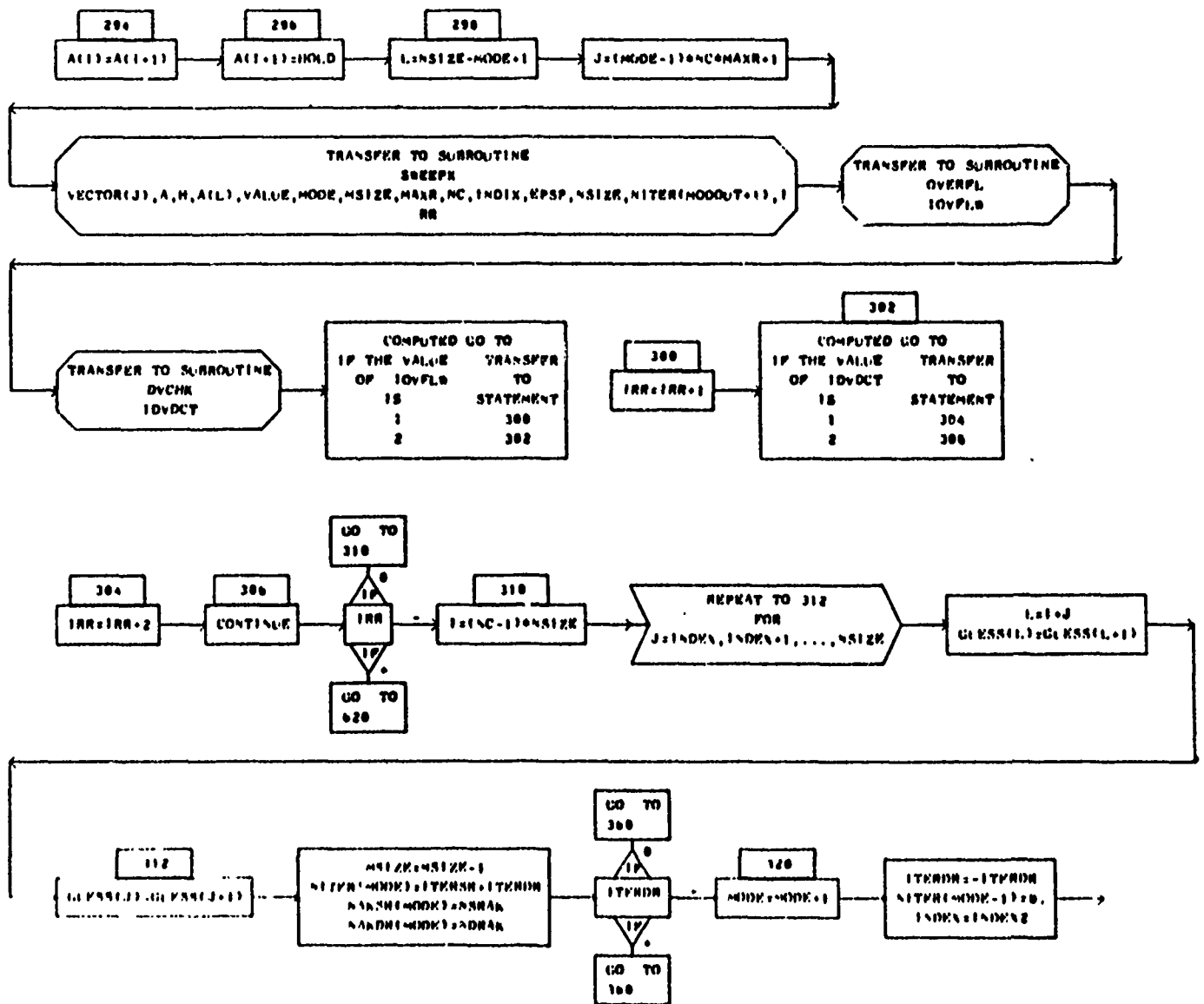
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	GUESS	1	H	1	METER	1	NAKSR	1
NANDR	1	ATITLE	0	VECTOR	1	VALUE	1		

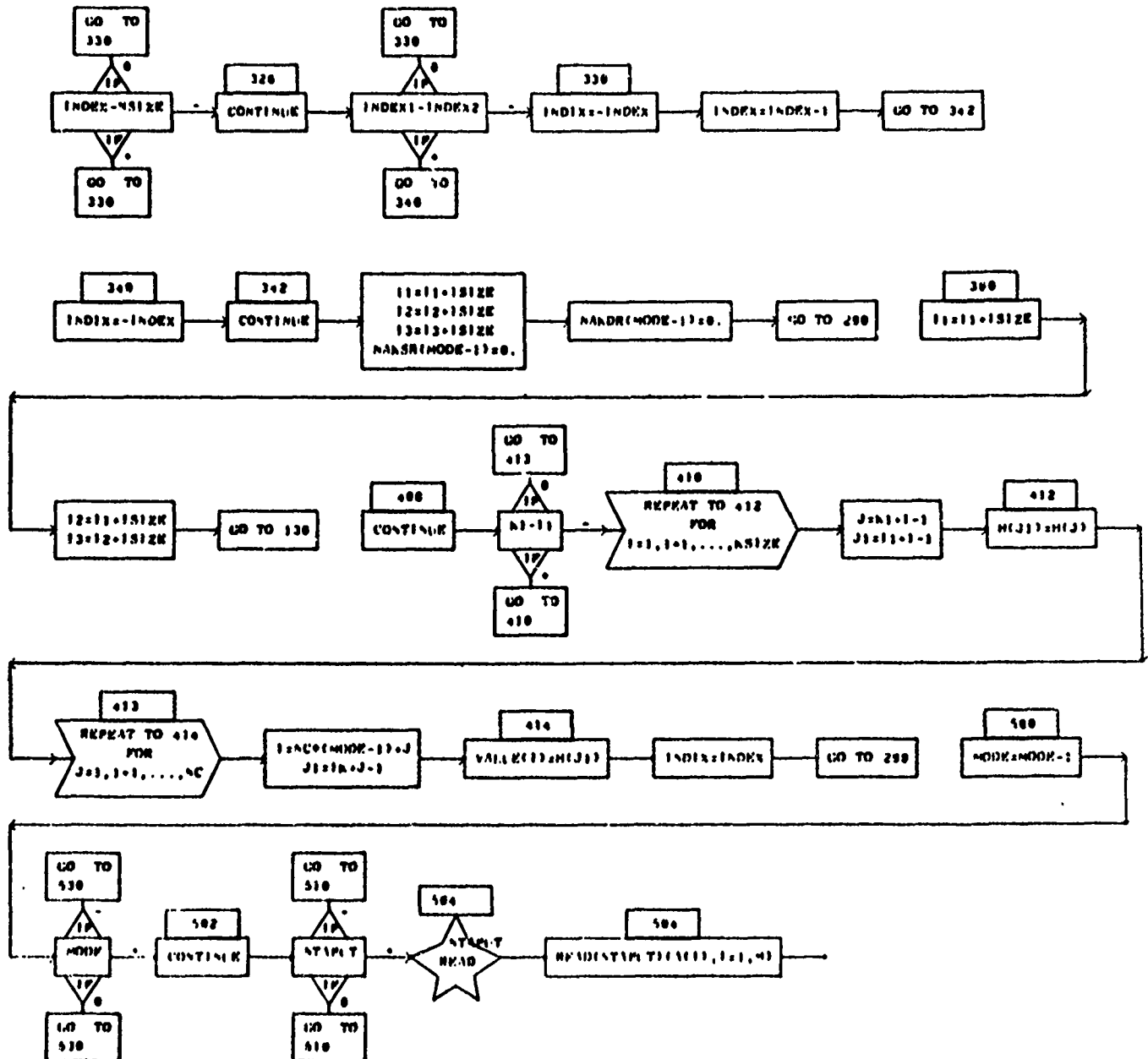


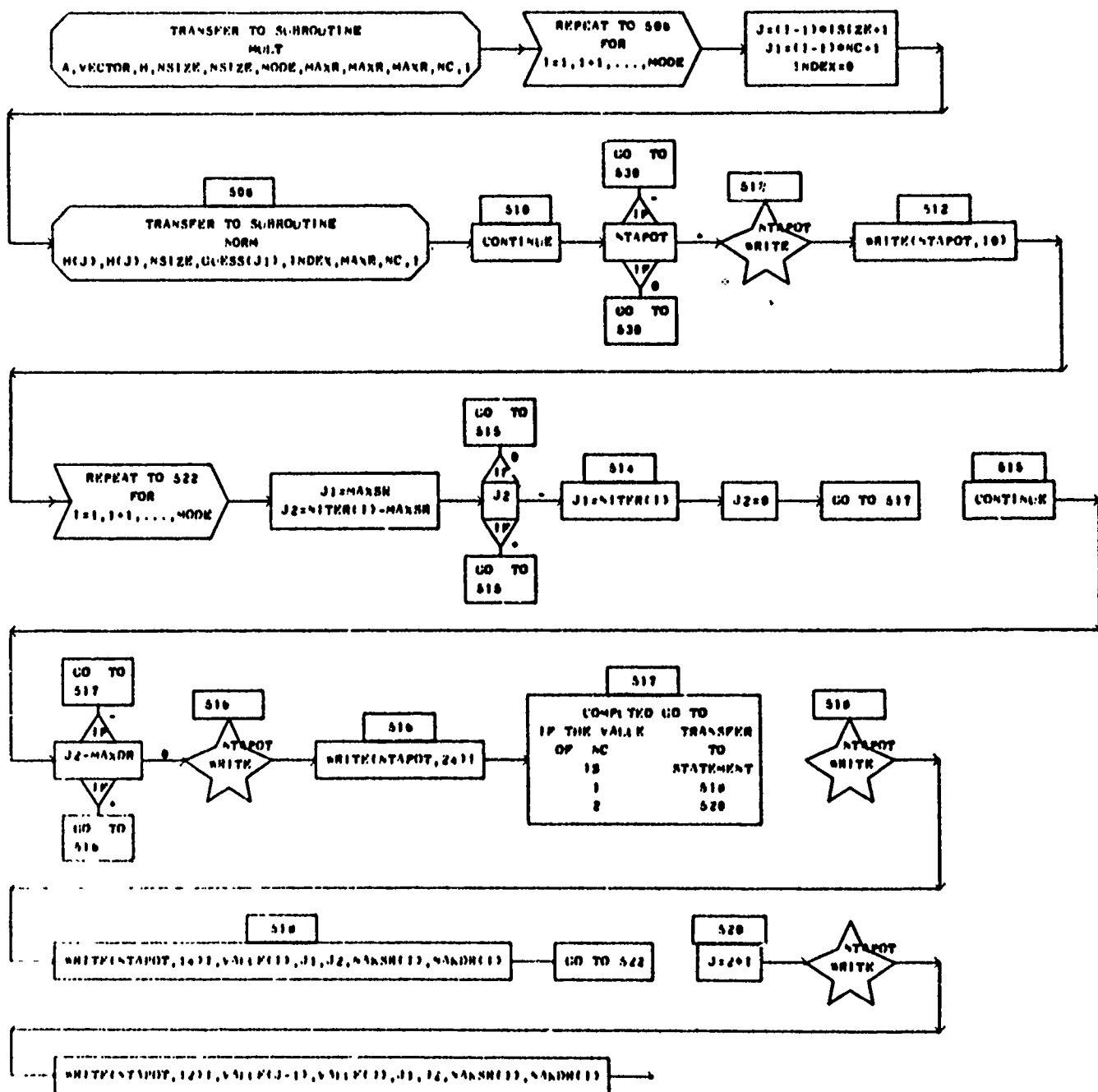


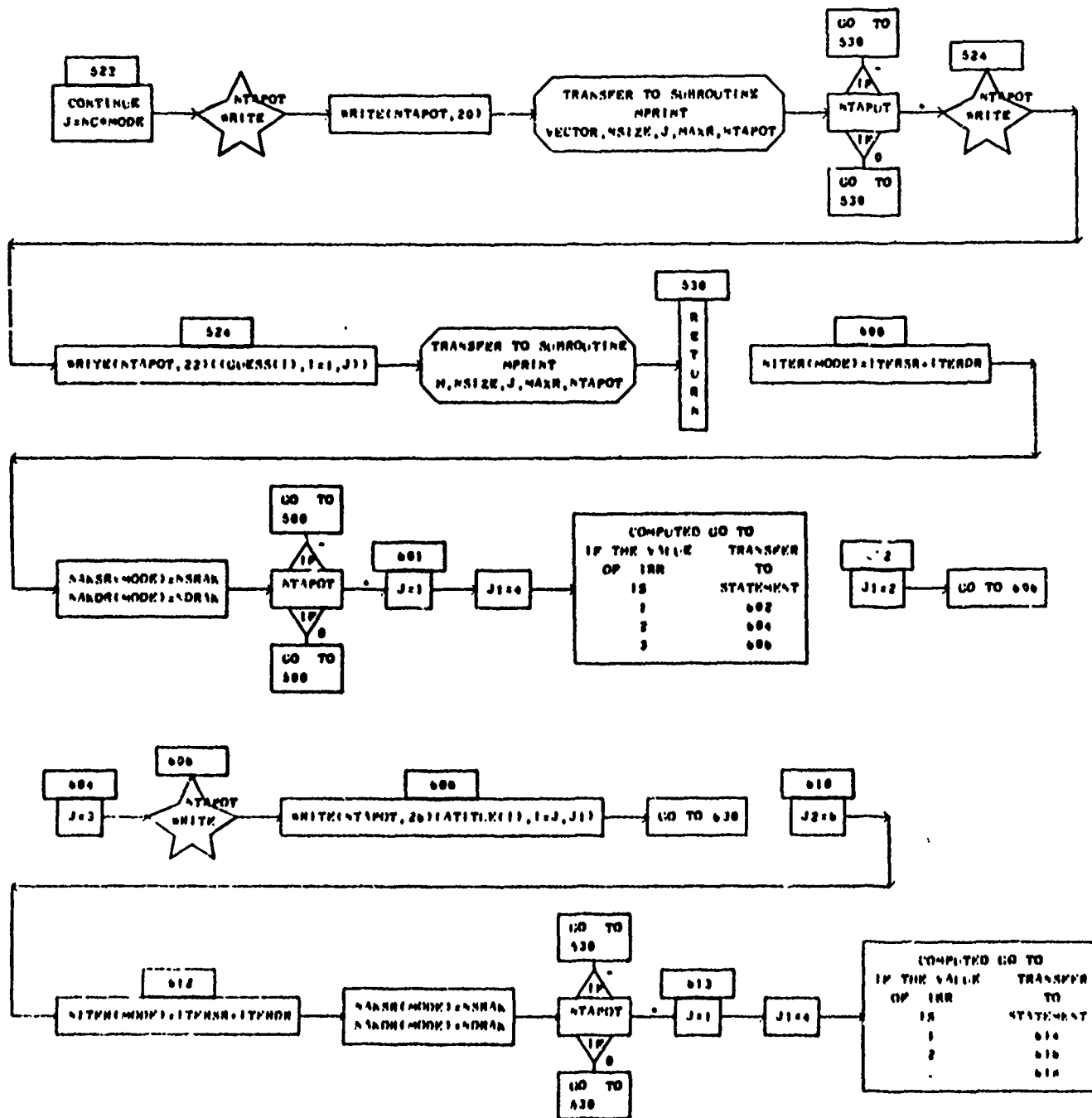


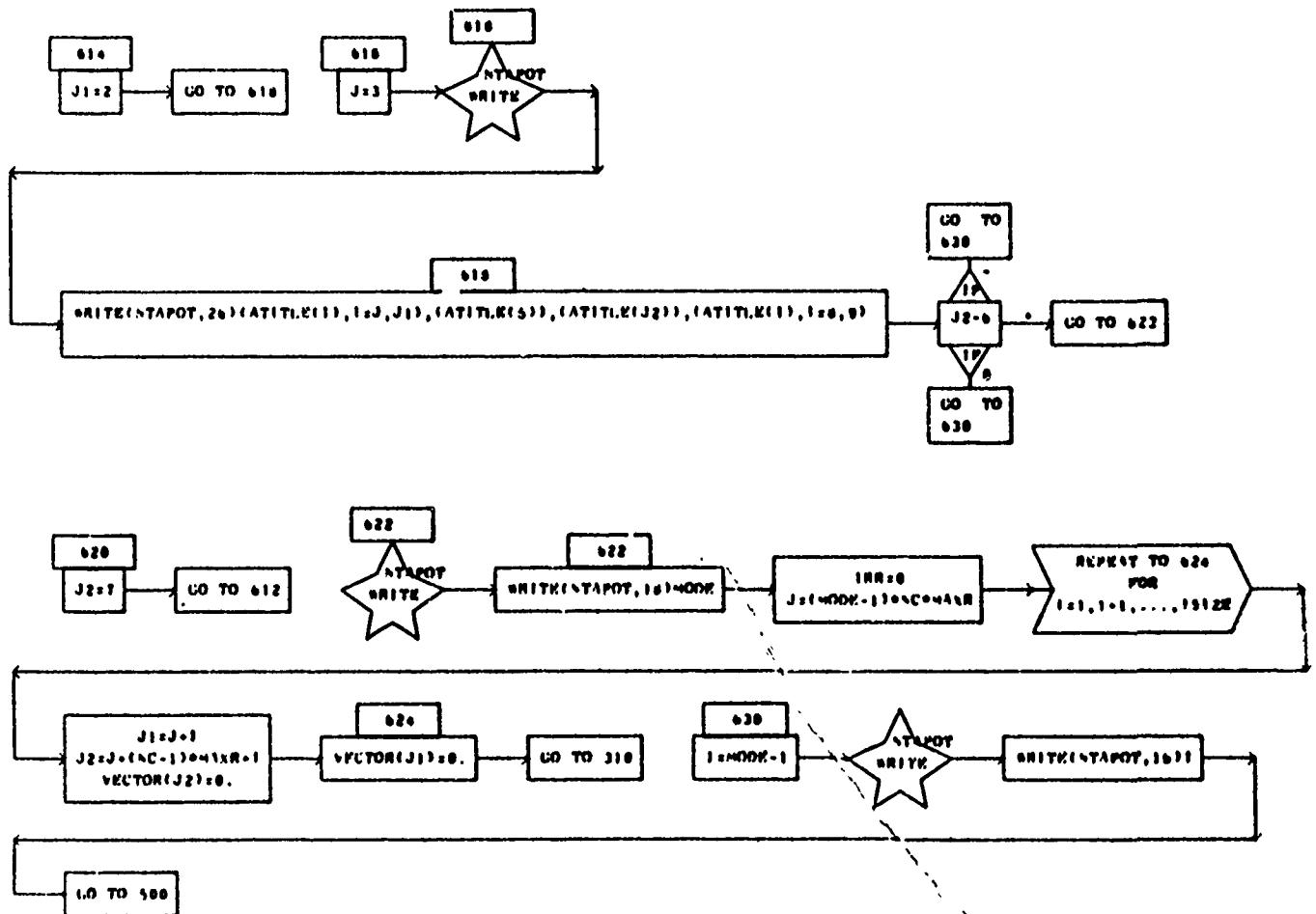












SKEEPS

SKEEPS SUBROUTINE

COMPUTES TRUE MODE AND SKEEPS IT FROM THE MATRIX. (REAL OR COMPLEX)

HTRUE = TRUE MODAL COLUMNS, AS COMPUTED. U = DYNAMIC MATRIX.

H = SERIES OF MODIFIED MODAL COLUMNS. P1 = COLUMN OF EIGENVALUES.

US = SERIES OF MODIFIED MODAL ROWS OF U.

MODE = MODE NOS BEING COMPUTED. N = SIZE

NO = DIMENSIONED NUMBER OF ROWS OF U, H, HTRUE

NA = 1 IF PROBLEM IS REAL.

= 2 IF PROBLEM IS COMPLEX.

DIMENSIONED VARIABLES

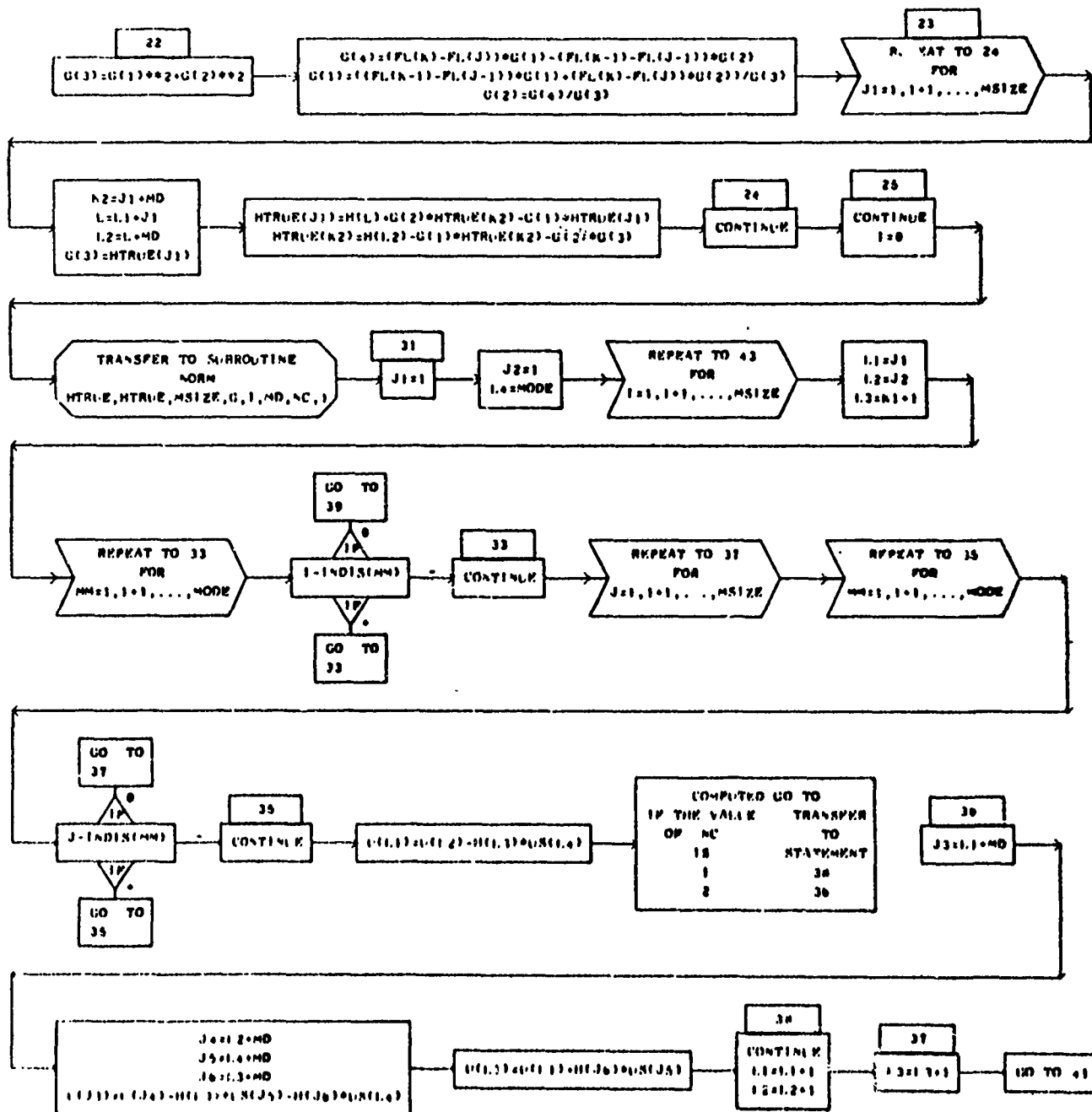
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
H	1	US	1	U	1	HTRUE	1	P	1
G	4	INDIS	1						

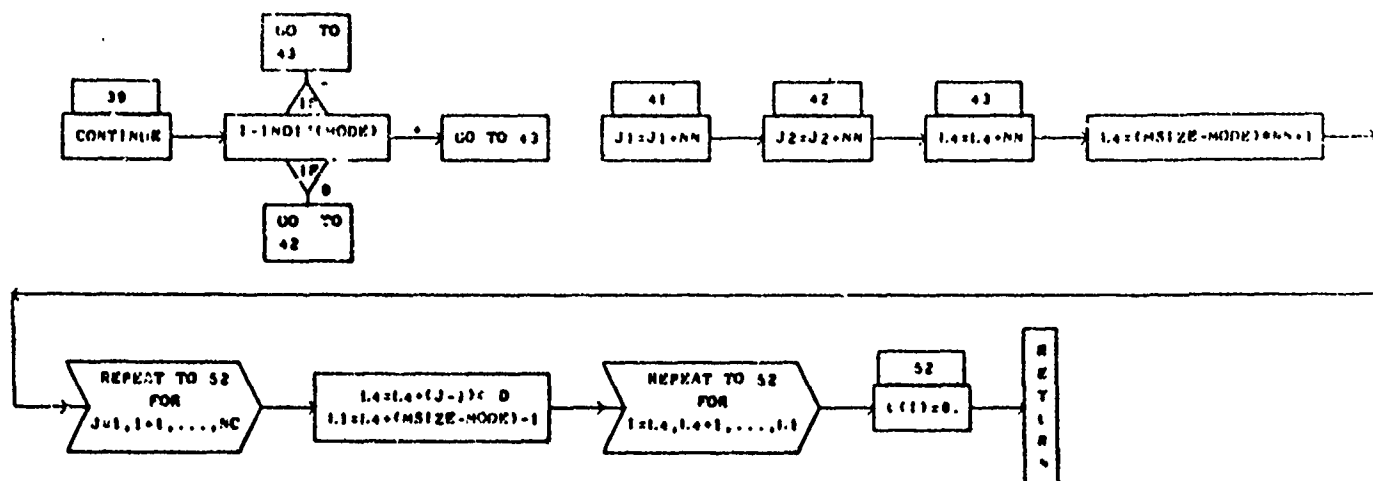


PAGE 3









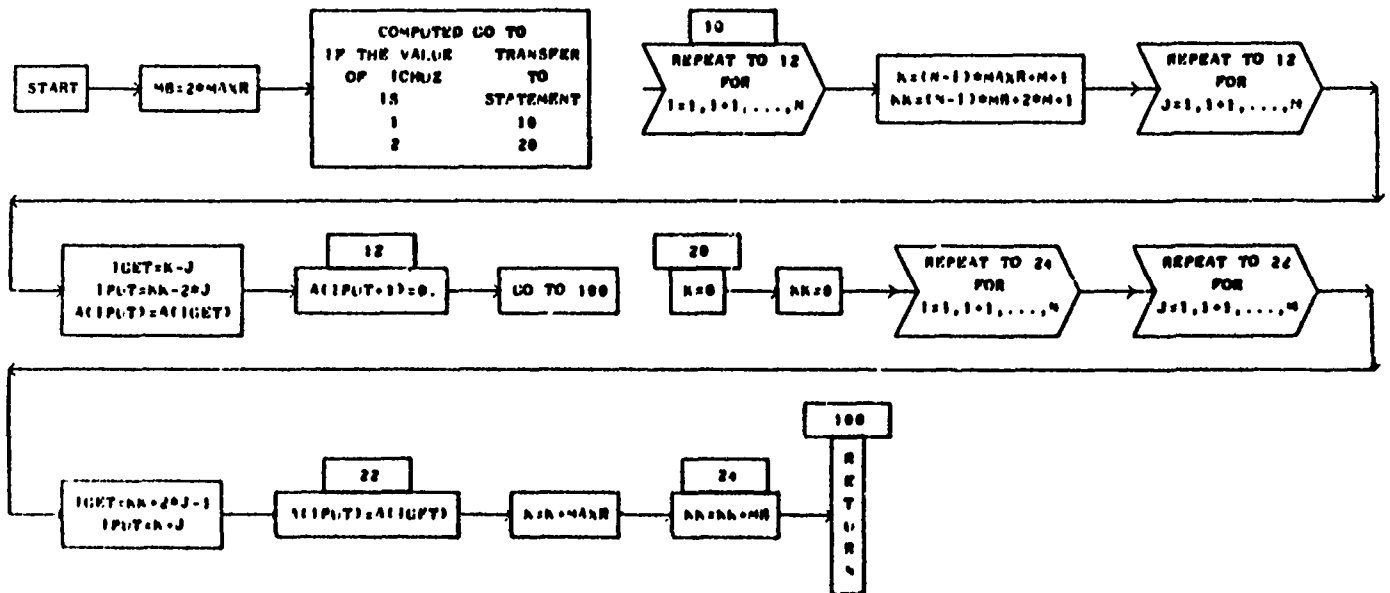
CHANGES

DIMENSIONED VARIABLES

SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE	SYMBOL	STORAGE
A	1								

SUBROUTINE CHANGE (A, N, M, MAXR, ICHU2)

PAGE 1



S E C T I O N 10
NOMENCLATURE

N O M E N C L A T U R E

a	Element of flexibility matrix, in./lb
a_R	Generalized amplitude coefficient of rigid-body modal series, in. or rad
b_r	Reference semichord, ft
C_h	Element of oscillatory aerodynamic influence coefficient matrix, dimensionless
F	Control point force, lb
ζ	Structural damping coefficient, dimensionless
h_o	Control point deflection due to rigid-body motion, in.
h_R	Element in rigid-body modal matrix, in. or dimensionless (see Section II)
h_1	Control point deflection, in.
K	Flexibility matrix normalizing constant, dimensionless
k_r	Reference reduced frequency, dimensionless
M	Element of mass matrix, lb.
$\bar{M}$	Element of complex mass matrix (includes aerodynamic effects), lb
m	Element of generalized mass matrix, lb., in.-lb, or lb-in ² .
$\bar{m}$	Element of sum of generalized mass and aerodynamic matrices, lb, in.-lb, or lb-in ² .
Q	Element of generalized aerodynamic force matrix, lb, in.-lb, or lb-in ² .
R	Number of rigid-body modes
s	Reference semispan, ft (i.e., span measured from root to tip)
U	Element of dynamic matrix, in.
V	Velocity, knots
W	Element of aerodynamic weighting matrix, dimensionless

SYMBOLS (continued)

- λ Eigenvalue, $\lambda = \lambda_R + i\lambda_I$, in.
 ρ Atmospheric density, slugs/ft³
 f Frequency, cps

Matrix Notation

- $\begin{bmatrix} \end{bmatrix}$ Square
 $\begin{Bmatrix} \end{Bmatrix}$ Column
 $\begin{bmatrix} \end{bmatrix}^T$ Transposed
 $\begin{bmatrix} I \end{bmatrix}$ Unit

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when all report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) HUGHES AIRCRAFT COMPANY, MISSILE SYSTEMS DIVISION FALLBROOK AND ROSCOE BLVDS. CANOGA PARK, CALIFORNIA 91304		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
		2b. GROUP
3. REPORT TITLE COLLOCATION FLUTTER ANALYSIS STUDY		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) FINAL REPORT (MARCH 1968 THROUGH MARCH 1969)		
5. AUTHOR(S) (First name, middle initial, last name) DYNAMICS AND ENVIRONMENT SECTION, DONALD R. ULBRICH		
6. REPORT DATE APRIL 4, 1969	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. N00019-68-C-0247	8b. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		
10. DISTRIBUTION STATEMENT IN ADDITION TO SECURITY REQUIREMENTS WHICH APPLY TO THIS DOCUMENT AND MUST BE MET, EACH TRANSMITTAL OF THIS DOCUMENT OUTSIDE THE AGENCIES OF THE U.S. GOVERNMENT MUST HAVE PRIOR APPROVAL OF THE COMMANDER, NASC		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY NAVAL AIR SYSTEMS COMMANDS DEPARTMENT OF THE NAVY WASHINGTON, D.C.	
13. ABSTRACT THIS STUDY COVERS THE DEVELOPMENT OF A SET OF COMPUTER PROGRAM TO PERFORM FLUTTER ANALYSIS BY THE COLLOCATION METHOD. WHILE THIS METHOD HAS BEEN KNOWN FOR SOME TIME, ONLY RECENTLY HAVE ADVANCES IN COMPUTER TECHNOLOGY MADE THE METHOD TECHNICALLY AND FINANCIALLY FEASIBLE. THE INGREDIENTS OF A COLLOCATION FLUTTER ANALYSIS ARE 1) A FLEXIBILITY MATRIX, 2) AERODYNAMIC INFLUENCE COEFFICIENT MATRIX, AND 3) AN EIGENVALUE SOLUTION. THIS STUDY IS PRESENTED IN FOUR VOLUMES. VOLUME I CONTAINS A GENERAL PROGRAM DISCUSSION. VOLUME II CONTAINS THE PROGRAM FLUENC WHICH CALCULATES THE FLEXIBILITY MATRIX. VOLUME III CONTAINS A SET OF THREE PROGRAMS TO CALCULATE AERODYNAMIC INFLUENCE COEFFICIENTS FOR SUBSONIC, TRANSONIC, AND SUPERSONIC FLIGHT REGIMES. VOLUME IV CONTAINS THE PROGRAM COFA WHICH SETS UP AND SOLVES THE FLUTTER EIGENVALUE MATRIX.		

DD FORM 1473
1 NOV 65UNCLASSIFIED
Security Classification

14

KEY WORDS

LINK A

LINK 8

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

FLUTTER

VIBRATION

AERODYNAMIC INFLUENCE COEFFICIENTS

1.5