NASA CR-66121

USER'S MANUAL FOR A FORTRAN IV PROGRAM FOR COMPUTING FLUTTER BOUNDARIES OF FLAT PANEL ARRAYS IN SUPERSONIC FLOW

Contract No. NAS1-4900
8 July 1966

MRI Project No. 2852-P

For

National Aeronautics and Space Administration
Langley Research Center
Langley Station
Hampton, Virginia 23365

MRI
MIDWEST RESEARCH INSTITUTE
425 VOLKER BOULEVARD/KANSAS CITY, MISSOURI 64110/AC 816 LO 1-0202

USER'S MANUAL FOR A FORTRAN IV PROGRAM FOR COMPUTING FLUTTER BOUNDARIES OF FLAT PANEL ARRAYS IN SUPERSONIC FLOW

by

D. R. Kobett
D. I. Sommerville

Contract No. NAS1-4900
8 July 1966

MRI Project No. 2852-P

For

National Aeronautics and Space Administration
Langley Research Center
Langley Station
Hampton, Virginia 23365

MRI

425 VOLKER BOULEVARD/KANSAS CITY, MISSOURI 64110/AC 816 LO 1-0202

PREFACE

This manual was prepared by Midwest Research Institute under Contract No. NAS1-4900, "Research Relating to the Flutter of Flat Panels in a Supersonic Air Stream," for the Langley Research Center of the National Aeronautics and Space Administration.

Approved for:

MIDWEST RESEARCH INSTITUTE

Sheldon L. Levy, Director
Mathematics and Physics Division

8 July 1966

ABSTRACT

The computer program is a by-product of research efforts that have produced NASA TN D-2227, CR-80, CR-538, and AFOSR TN 1952. The report material is arranged for two distinct types of readers, namely, the engineer who wishes to use the program as is, and the programmer who may be required to modify the program for a specific application.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

List of Figures

I. INTRODUCTION

The computer program described in this report is a by-product of panel flutter research efforts conducted under NASA and AFOSR sponsorship. It was developed to facilitate the calculation of flutter characteristics for multi-bay, flat panel arrays exposed on one side to a uniform supersonic airstream. The program is written in FORTRAN IV for running on the IBM 7094 under IBSYS.

The computation technique employed is straightforward in the sense that a computer-run directly obtains, as output, points on the boundary separating stable from unstable regions in a generalized flutter-parameter-plane. (More conventional techniques require cross-plotting or machine interpolation under manual direction to obtain the "flutter points".) Although the program presented here is tailored to the case of a flat panel array, the technique can be conveniently adapted to related applications, for example, the flutter of a cylindrical shell or a wing-body configuration.

The report material is arranged for two distinct types of reader, namely, the engineer who wishes to use the program as is, and the programmer who may be required to modify the program for a specific application. The engineer will want to become familiar with the first six sections through the interpretation of output; the programmer with the main body of the report and those appendices that apply to the intended modification.

The computer program is based on the analytical model developed in [1].* A brief summary description of the model is given in Section II of this report; the reader interested in detail is referred to [1].

* Numbers in brackets refer to the bibliography.

II. BACKGROUND INFORMATION FOR THE PROGRAM USER

This section is divided into two parts, viz., a description of the analytical model employed and a discussion of the parametric options available to the program user.

A. Analytical Model

The computer program is based on the analysis described in [1] , the salient features of which are summarized below.

Four different types of panel array configurations can be analyzed. Each configuration has an arbitrary number of chordwise bays; the distinction between configurations is associated with spanwise features. The first, and most general, array is one with finite span and arbitrary number of spanwise bays typified by Fig. 1.* The second configuration is one in which the array span is divided into equal width bays extending to infinity. The third configuration is an array with one spanwise bay whose side edges are free to deflect. The final configuration consists of the third array above, flanked at a distance by vertical surfaces representing wind tunnel walls.

All configurations have certain features in common. The upper surface of the panels is exposed to a uniform supersonic flow and the lower surface to a constant pressure equal to the static pressure in the undisturbed freestream. Acoustic effects on the lower surface and membrane stresses due to static pressure differential across the panel are not included.

To clarify the differences in the analyses of the different configurations, the case typified by Fig. 1 will first be discussed in some detail. Then those aspects peculiar to the other configurations will be pointed out.

The array is composed of geometrically identical panels; it has L chordwise and N spanwise bays, and is bordered by an inflexible surface. A nondimensional equation of motion for the vertical displacement of the panel surface is obtained using small deflection plate theory and exact, linearized, three-dimensional, potential aerodynamic theory. In formulating boundary and compatibility conditions the array is assumed to be supported at its perimeter and along the interior lines delineating the individual panels by a structure that does not deflect perpendicular to the plane of the array, but that does

* See Appendix I for Figures 1 - 6.

supply torsional restraint. At the leading and trailing edges the moment imposed by the supporting structure is equal to a proportionality factor, ϵ_x, multiplying the local panel slope. At the interior, spanwise directed members, the imposed moment is proportional to 2 ϵ_x times the local panel slope; i.e., the interior members are twice as stiff, torsionally, as the ones at the perimeter.

The above torsional properties hold also for the chordwise directed supporting members, with the exception that the proportionality constant, ϵ_y, can be different from ϵ_x. The slope across the interior panel boundaries is assumed to be continuous.

The equation of motion is in essence a self-excited forced vibration equation where the aerodynamic pressure induced by the panel displacement is the forcing function. It is assumed that conditions for which the equation has harmonic solutions are conditions of neutral stability; this criterion is used in calculating the flutter boundaries for the panel array.

Harmonic solutions are obtained using a Galerkin approach. The chordwise variation of the panel displacement is approximated by a finite summation of natural vibration modes of a beam with L bays and boundary conditions the same as on the spanwise directed panel edges. The spanwise variation is approximated by one natural vibration mode* of a beam with N bays and boundary conditions the same as on the chordwise directed panel edges. Application of the Galerkin procedure requires integration of the aerodynamic pressure over the panel surface. This integration is performed by expanding the spanwise deflection shape in a sine series; then numerically integrating term by term using the unsteady pressure solution derived in [2] for sinusoidal spanwise and arbitrary chordwise deflection shape.

The above procedure leads to a set of simultaneous, complex, algebraic equations (see [1], Eq. (39)) for the amplitude and phasing of the approximating modes. An effect of structural damping is introduced by multiplying the stiffness matrix by the complex damping factor $(1 + jg)$. Solutions to the equations for real values of the basic parameters $1/\mu$ and $Z^{1/3}$

---

* Coupling between spanwise modes is neglected; the mode number is arbitrary.

are obtained by the procedure described in Section III*. The preceding is a brief description of analytical details for the panel array typified by Fig. 1. The analysis for the other three configurations are similar to that described above, with the exceptions noted in the following paragraph.

For the array which extends to infinity in the spanwise direction (second configuration) the mode approximating the spanwise deflection is automatically taken to be the lowest frequency natural vibration mode of a beam with an infinite number of bays. For the array with free side edges (third configuration) the panels are assumed to deflect two-dimensionally, i.e., with no spanwise variation in the deflection shape. The two-dimensionality assumption is incorporated by using the sine series expansion for a rectangular half wave. The fourth configuration, the one with simulated wind tunnel walls, is also handled simply by introducing an appropriate expansion. In this case, the expansion is one which simulates the effect of fictitious image panels, thereby obtaining no cross flow at the walls. Mach wave reflections from the walls are intrinsically included by this technique.

B. Parametric Options Available to the Program User

Within the framework of the four geometrical configurations described earlier, there are options available for what will be called analytical and physical parameters. The analytical parameters are:

1. Number of modes in the approximation of the chordwise deflection shape.
2. Mode numbers of the chordwise approximating terms.
3. Number of terms in the sine series expansion of the spanwise approximating mode.
4. Mode number of the spanwise approximating term (first configuration only).

---

* $\frac{1}{\mu}$ = mass ratio parameter $= \frac{\rho a}{\rho_s h}$

$Z^{1/3}$ = a parameter involving dynamic pressure and bending stiffness

$$= \frac{h}{a}\left[\frac{E}{q(1-\nu^2)}\right]^{1/3}$$

where

ρ = free stream density
ρ_s = panel material density
a = panel chord
h = panel thickness
E = modulus of elasticity
q = freestream dynamic pressure
ν = Poisson's ratio

The physical parameters are:

1. Mach number
2. Aspect ratio
3. Torsional restraint proportionality factors ϵ_x and ϵ_y
4. Number of chordwise panels
5. Number of spanwise panels (first configuration only)
6. Distance from panel edge to wind tunnel wall (fourth configuration only)
7. Structural damping coefficient.

Permissible ranges for the analytical parameters are defined in the following paragraphs.

The number of modes used to approximate the chordwise deflection shape must be at least 2 and at most 10. The lower limit is set by the analytical techniques employed and the upper limit by dimensioned array sizes in the program. Numerical considerations are expected to reduce the permissible upper limit to less than 10. With an earlier version of the program, numerical inaccuracy was encountered in increasing from 4 to 6 modes. The present, improved, version has been used extensively for up to 6 mode analyses [3]; it has not been applied to analyses requiring the use of more than 6 modes. The modes may be any combination of the first 30 natural vibration modes of a beam with the appropriate number of bays.

Up to 20 nonzero terms may be used in the sine series expansion of the approximating spanwise mode. Since the expansion always has zero alternate terms (odd or even numbered depending on mode number and number of spanwise bays), the above condition actually corresponds to a 40 term series expansion.

The spanwise approximating mode (first geometrical configuration only) may be any one of the first 30 natural vibration modes of a beam with N bays, N being the number of spanwise bays in the panel array. The above is a mechanical bound set by the program; practically, the mode number must be selected in cognizance of the number of spanwise bays, N, the spanwise mode number, and the 40 term bound on the sine series expansion.

This completes the definition of permissible ranges for the analytical parameters. Ranges for the physical parameters are discussed in the succeeding paragraphs.

Mach number must be confined to the supersonic regime (i.e., > 1) for compatibility with the analytical model.

The computer program imposes no restriction on aspect ratio. However, a practical bound (a judgment of the user) is imposed by the permissible number of modes in the approximation of the chordwise deflection shape.

The nondimensional torsional restraint proportionality factors ϵ_x and ϵ_y can be independently varied from zero to any maximum desired*. On a practical basis, however, $\epsilon = 0$ corresponds to pinned edge conditions and $\epsilon = 1{,}000$ has been found to very closely approximate the clamped edge case (a judgment made on the bases of both modal frequencies and computed flutter boundaries). Further, it has been observed that flutter boundaries computed using $\epsilon = 100$ and $\epsilon = 1{,}000$ are nearly identical, while $\epsilon = 10$ yields results approximately midway between the pinned and clamped edge cases.

Panel arrays with up to 4 chordwise bays can be analyzed. Again, however, a practical bound must be recognized, associated with the permissible number of modes in the approximation of the chordwise deflection shape.

Up to 6 spanwise bays can be used in analyses of finite span arrays (first geometrical configuration). As noted earlier, however, the number must be selected in cognizance of the spanwise mode number and the 40 term bound on the sine series expansion of the spanwise deflection shape.

For the fourth geometrical configuration, the distance from the panel side edges to the wind tunnel wall must be greater than zero to be compatible with the theory. Practically, it should be equal to or greater than about 10 per cent of the panel span in order to obtain a satisfactory sine series expansion of the spanwise deflection shape.

Finally, there is no program restriction on values for the structural damping coefficient.

This completes the overall description of parametric options available to the program user. Related recommendations for optimum program usage are provided in Section IV.

III. GENERAL PROGRAM DESCRIPTION

The method used to find flutter points is essentially that described on pages 15 - 17 of [1]. A few slight modifications are used, however, in the interest of saving either computer running time or storage.

* See Appendix VIII for description of a mild restriction on admissible values for ϵ_y .

The program makes use of a magnetic tape to store a number of arrays which depend only on the number of spanwise and chordwise bays, the corresponding mode numbers allowed and the parameters ϵ_x and ϵ_y describing edge conditions. If such a tape has not already been generated it can be generated as the first step of any flutter run. If it has been generated on a previous run it may be mounted as an input tape. Since the program may take around 12 min. to produce a tape when 3 chordwise bays are needed, it is advisable to save the tape if it might be used again. Details of the tape format are given in Appendix III.

Because of the size of the program it is necessary to use the overlay feature of the loader during execution. The first overlay generates the tape described in the preceding paragraph, if necessary. The second overlay reads data until a flutter problem has been defined, reads data from the tape generated previously for the specified number of chordwise bays, prints out heading information describing the problem and generates matrices which can be used by a more or less general subroutine, EUCLID, to compute flutter points and vectors. Subroutine EUCLID is loaded as the third overlay.

The flutter matrix used by the program is equivalent to but slightly different from that shown in [1]. Let $E = \left\{ E_{\bar{m},m} \right\}$, $J = \left\{ J_{\bar{m},m} \right\}$ and $C = \left\{ C_{\bar{m},m} \right\}$ to simplify notation. Then since Eq. 39 of Ref. [1] can be premultiplied by any nonsingular matrix, we can premultiply by $\beta(1-jg)E^{-1}$, for E is nonsingular and β is not zero for Mach number greater than 1. The new form of Eq. 40 then becomes

$$\mathrm{Det} \left\{ \frac{Z\beta(1+g^2)I}{24} - \mu\beta k^2(1-jg)E^{-1}J + (1-jg)E^{-1}C \right\} = 0$$

where I is the identity matrix. The argument list for subroutine EUCLID includes g, $-\beta k^2$, $12/\left(\beta(1+g^2)\right)$, and a table of μ-values in addition to the matrices $E^{-1}J$ and $E^{-1}C$. If we let λ denote $\frac{Z\beta(1+g^2)}{24}$ and $\bar{C}(\mu)$ be the matrix $\mu\beta k^2(1-jg)E^{-1}J - (1-jg)E^{-1}C$, the problem to be solved is that of finding real parameters λ and μ such that $\left| \lambda I - \bar{C}(\mu) \right| = 0$.

If μ is assigned a numerical value, then $\left| \lambda I - \bar{C}(\mu) \right|$ is the characteristic equation of the complex matrix $\bar{C}(\mu)$. The condition that the determinant vanish requires that real and imaginary parts of the characteristic equation have a common root. This occurs if and only if the Sylvester resultant [4] between the real and imaginary parts of the characteristic equation vanishes. Thus if for two distinct values of μ, the corresponding Sylvester resultants have opposite signs, there is a μ value between the original two for which the resultant vanishes.

Subroutine EUCLID first tabulates the Sylvester resultant corresponding to each value in the μ - table.* A search is then made for sign changes of the resultant as μ varies. When a sign change is detected, the program interpolates until the flutter value of μ is found to the required degree of accuracy. Then the corresponding value of λ and the flutter vector are found. The results are converted to the parameters of interest, namely $Z^{1/3}$ and $1/\mu$ and printed. The program then repeats the steps above until all sign changes in the resultant table have been processed, whereupon the second overlay is called again to read data for the next problem.

IV. COMPUTATION PROCEDURE

The purpose of a typical application of the computer program is to calculate stability boundaries for a particular geometrical configuration, and fixed values of Mach number, aspect ratio, torsional edge restraint, spanwise mode number (where applicable), and structural damping coefficient. The stability boundaries are obtained via a series of calculations as described below.**

In a discrete calculation, values are assigned to the analytical and physical parameters, a reduced frequency is specified, a table of μ values defined, and the program then computes the flutter points lying within the range of μ defined by the table.*** (For clarity of ensuing discussions a computation case is here defined as a series of discrete calculations for a selected set of reduced frequencies and damping coefficients.) Every computation case yields a number of points on the stability boundaries, each point being characterized by a particular frequency and flutter vector. Distinct stability boundaries are obtained by constructing continuous curves through the computed points, using continuity of reduced frequency and flutter vector as associative criteria.

---

* A table of μ values is defined by input data.

** In general, a number of computer runs will be required to completely define the boundaries.

*** If two flutter points lie within one increment of the μ table neither will be detected. However, experience with the program has shown that this event seldom occurs when a μ table with moderate increment sizes is employed (see Fig. 2 for recommended table). Further, when it does occur it creates little difficulty because the points can be obtained, if desired, by repeating the calculation using a μ table with finer increment sizes. In most instances the repetition is unnecessary because adjacent points on the boundaries are available from calculations made using slightly different frequencies.

The input μ table may be tailored to restrict the range through which the program searches for flutter points. In some instances, this tactic can be profitably employed to reduce computation time. In general, however, a range in $1/\mu$ from 0.01 to 0.20 is of practical interest and the μ table in Fig. 2 has therefore been used extensively in application of the computer program.

It has also been found economical to limit each computation case to between 5 and 10 reduced frequencies, and more or less construct the boundaries step by step.

Computation time can be reduced through judicious use of the intermediate data tape. The data arrays stored on the tape depend on

Number of spanwise bays - N
Spanwise mode number - NGBAR
Torsional edge restraints - ϵ_x and ϵ_y
Number of chordwise bays* - L.

The tape also contains data for 10 chordwise modes which may be any 10 of the first 30 natural vibration modes of a beam with L bays (see Section V on Input Preparation). The data tape, on the other hand, is independent of Mach number, aspect ratio and structural damping coefficient.

Since an appreciable amount of time is required to generate the data arrays stored on the tape (particularly for $L > 1$) it is advisable to reserve the tape for use as an auxiliary input tape in subsequent runs. This is true whether or not Mach number, aspect ratio and structural damping are varied. To emphasize the above point it may be noted that the results reported in [3] were obtained using a tape generated in the first computer run, whereas a total of approximately 15 runs were made in all.

Some final comments are required concerning the case where a computed flutter point(s) is incorrect (indicated by program checks included in the printed output). If this occurs the input data should be double checked first, because improper input can cause erroneous results. (For example, the mode numbers may not be in ascending order.) If the data are correct the specified error bound on μ (TESTR; see Appendix IX) may be too large. Experience with the program has indicated that a value of TESTR equal 10^{-7} is adequate in general for analyses using up to 6 chordwise modes, but numerical peculiarities could occasionally require an even smaller error bound.

* Data for up to 4 chordwise bays can be stored on 1 tape. However, no penalty in computation time is incurred if the user chooses to generate separate tapes for each L value.

Another possible cause of erroneous results is the use of too many chordwise modes in the analysis. As previously noted, with an earlier version of the program numerical inaccuracy was encountered when the number of modes was increased from 4 to 6. Improvement of one of the program routines (FLUT) eliminated this difficulty but it is possible that the use of more than 6 modes may similarly overtax a critical routine.

V. INPUT PREPARATION

A data deck may contain data sets for one or more computation cases, the only restriction being that the cases must all require the same intermediate data tape*, which may be available from a previous run or generated as part of the first case of the current run.

The first card of a data deck refers to the intermediate tape. If the tape is available from a previous run the first card is blank; if the tape is to be generated in the current run the first card is as follows.

Cols.	Format	Symbol	Description
1-2	I2	LMIN	(see LMAX)
3-4	I2	LMAX	Data arrays will be generated and stored for values of L from LMIN to LMAX inclusive (LMIN $\leq$ LMAX).
5-24	10I2	MODE(I)	Array of 10 chordwise mode numbers between 1 and 30 in ascending order. Data arrays will be generated and stored for the 10 modes specified.**
25-26	I2	NSP	Number of spanwise bays.
27-28	I2	NGBAR	Spanwise mode number.
29-38	F10.0	EPY	Torsional restraint proportionality factor for spanwise directed panel edges (ϵ_x in Section II and [1]).

* See discussions of the intermediate tape in Section IV and Appendix III.
** Ten modes are required here even though some are not later used.

Cols.	Format	Symbol	Description
39-48	F10.0	EPX	Torsional restraint proportionality factor for chordwise directed panel edges (ϵ_y in Section II and [1]). This item is irrelevant for options 4 and 5 (see option identification in later description of number 11 data card).

Data for the first case follow the intermediate tape card. This first data set must contain all of the cards described below (and illustrated in Fig. 4 which is provided for user reference) plus a blank terminating card. Subsequent data sets need only include the cards containing input information to be redefined for the new case. Each data set following the first is terminated by a blank card.

The individual data cards each have an identification number (between 1 and 12) in columns 1 and 2. This number is used for reference in the following description of input data details.*

01 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (01 on this card).
11-20	E10.5	TESTR	Upper bound on acceptable decimal per cent error in computed μ (Appendix IX).

02 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (02 on this card).
11-20	E10.5	S	Reciprocal of aspect ratio.

03 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (03 on this card).

* Sample data listings (Figs. 2a, 2b and 3) are discussed at the end of this section.

03 Card (Concluded)

Cols.	Format	Symbol	Description
3-4	I2	MMAX	Number of chordwise modes.
5-6	I2	-	If zero, consecutive modes starting with the first are used; if not zero see next item.
61-80	10I2	MODE(I)	This field is blank if columns 5 - 6 contain zero. If columns 5 - 6 are not zero, this field contains the mode numbers to be used in the analysis, in ascending order. (The modes listed must be ones that are included on the intermediate tape.)

04 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (04 on this card).
11-20	E10.5	EMSQ	Mach number squared.

05 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (05 on this card).
3-4	I2	NG	Number of damping coefficients to be processed (maximum of 5). Blank or zero is taken as 1.
11-60	5E10.5	GT(I)	Damping coefficients.

06 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (06 on this card).

06 Card (Concluded)

Cols.	Format	Symbol	Description
3-4	I2	L	Number of chordwise bays in panel array. (The number specified must be one of those included on the intermediate tape.)
11-20	E10.5	SAVE(1)	Blank or zero causes the inverse of the elastic matrix times the inertia and aerodynamic matrices to be printed. A nonzero value suppresses the printing.

07 Card

A series of 07 cards are used to read in the μ table. The table must contain a minimum of 2 and a maximum of 50 values arranged in monotonic order. Details of a typical 07 card are as follows.

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (07 on this card).
3-4	I2	-	Index of first μ on this card.
5-6	I2	-	Index of last μ on this card.
11-60	5E10.5	ALPHA(I)	Values of μ (up to five values).

08 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (08 on this card).
3-4	I2	MAXAL	Total number of values in the μ table.

09 Card

A series of 09 cards are used to read in the reduced frequencies to be processed. A maximum of 30 frequencies can be read, in any desired order. Details of a typical 09 card are as follows.

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (09 on this card).

09 Card (Concluded)

Cols.	Format	Symbol	Description
3-4	I2	-	Index of first frequency on this card.
5-6	I2	-	Index of last frequency on this card.
11-60	5E10.5	CKAY(I)	Frequencies (up to five values).

10 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (10 on this card).
3-4	I2	NK	Total number of frequencies read.

11 Card

This card identifies which of the 4 geometrical configurations described in Section II is to be analyzed. An option number is used for this purpose. The correlation between option number and geometrical configuration is as follows.

Option 1 - The finite array typified by Fig. 1.

Option 2 - The array with infinite span divided into equal width bays.

Option 4 - The array with one spanwise bay and free side edges.

Option 5 - The option 4 array flanked by vertical surfaces representing wind tunnel walls.

The omission of 3 in the option sequence results from the fact that a fifth geometrical configuration included in the analysis [1] has not been programmed. Details of the 11 card are as follows.

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (11 on this card).
3-4	I2	NCASE	Option number.

11 Card (Concluded)

Cols.	Format	Symbol	Description
11-20	E10.5	A/2	Distance from panel edge to wind tunnel wall, measured in panel spans. (Relevant to Option 5 only.)

12 Card

Cols.	Format	Symbol	Description
1-2	I2	-	Identification number (12 on this card).
3-4	I2	NNN	Number of nonzero terms in sine series expansion of spanwise deflection shape (maximum of 20). For Option 2 and $\epsilon_y = 0$ this number must be 1.

This completes the description of data card details. Sample data deck listings are given in Figs. 2 and 3 to complement the above description. The data listed in Fig. 2a cause two cases to be executed, both using an intermediate tape generated in the current run. The first case is an analysis of a finite array with 2 chordwise bays and 1 spanwise bay, and with

Mach number = 1.35
Aspect ratio = 4
Damping coefficient = 0.01
$\epsilon_x = \epsilon_y = 0$
Spanwise mode number = 1 .

The analysis uses the first 4 chordwise modes, 20 nonzero terms in the sine series expansion, and 2 frequencies, 1.0 and 1.5 are processed.

The second case is an analysis of an Option 2 array with 3 chordwise bays and with

Mach number = 1.2
Aspect ratio = 2
Damping coefficient = 0.015
$\epsilon_x = \epsilon_y = 0$.

The analysis uses chordwise modes 1, 4, 7, and 9,and 2 frequencies, 0.7 and 0.8 are processed.

The data listed in Fig. 2b also cause two cases to be executed, both using an intermediate tape generated in a previous run (assumed to be the run of Fig. 2a). The first case is an analysis of an Option 5 array with 1 chordwise bay and with

Mach number = 1.3
Aspect ratio = 1
Damping coefficient = 0
$\epsilon_y = 0$

The analysis uses the first 4 chordwise modes, and 15 nonzero terms in the sine series expansion. The distance from the panel edge to the wind tunnel wall is one-quarter of the panel span. Three frequencies, 1.0, 1.2 and 1.4, are processed. The second case is identical with the first except Mach number is changed to 1.2.

The data listed in Fig. 3 will cause the sample output of Fig. 5 to be generated.

VI. INTERPRETATION OF OUTPUT

The output listing from a typical computer run illustrated in Fig. 5 is the implied reference for this section.

The output listing includes several checks on the validity of computed results and it is therefore pertinent to begin this discussion with the following brief reiteration of the overall computational technique employed. The purpose of a computer run is to calculate real-valued pairs of the parameters μ and Z which satisfy Eq. (40) of [1], that is, values for which the flutter determinant vanishes. The technique employed is not one of direct iteration of the complex flutter determinant. Instead, the Sylvester determinant composed of the coefficients of the real and imaginary characteristic polynomials of the flutter matrix, is tabulated with respect to the parameter μ *. The μ for which the Sylvester determinant vanishes, together with the corresponding Z obtained by operating on the characteristic polynomials, comprise a real-valued μ - Z pair for which the flutter determinant vanishes. These pairs are listed in the output in the converted form $1/\mu$ and $Z^{1/3}$ together with the flutter vector, program checks and associated identifying information.

---

* The elements of the Sylvester determinant are real-valued functions of μ ; they do not depend on Z .

Referring now to Fig. 5 the first line of output is a self-explanatory identification heading. The next block of output is a summarization of input parameters as follows:

M	- Mach number
S	- Reciprocal of aspect ratio
E (SPAN)	- Torsional restraint proportionality factor for spanwise directed edges (ϵ_x)
E (CHORD)	- Torsional restraint proportionality factor for chordwise directed edges (ϵ_y)
TESTR	- A precision factor imposed on the iteration of the Sylvester determinant (Appendix IX)
L	- Number of chordwise bays
MMAX	- Number of chordwise modes used in the analysis
N	- Number of spanwise bays (relevant only to Option 1 configuration)
UMAX	- Number of nonzero terms in the sine series expansion of the spanwise deflection shape
SPANWISE MODE	- Number of the spanwise approximating mode
CASE	- Geometrical configuration option number
A/2	- Distance, in panel spans, from panel edge to wind tunnel wall (relevant only to Option 5)
1/DPHI	- Number of equal increments (per bay) used in the chordwise numerical integration of the aerodynamic pressure (fixed at 16 in the program)
K	- Reduced frequency
G	- Structural damping coefficient

The next block of output contains intermediate computed results as follows:

GAMMA BAR	- Frequency of spanwise mode
CHORDWISE MODE	- Mode numbers and frequencies of chordwise modes
BN(U)	- Nonzero coefficients of sine-series expansion of spanwise deflection shape; listed in order of decreasing wavelength

INVERSE OF ELASTIC MATRIX TIMES INERTIA MATRIX - The matrix $E^{-1}J$ (see Section III)

INVERSE OF ELASTIC MATRIX TIMES REAL PART AEROD MATRIX - The matrix Real $\{E^{-1}C\}$ (see Section III)

INVERSE OF ELASTIC MATRIX TIMES IMAG PART AEROD MATRIX - The matrix Imag $\{E^{-1}C\}$ (see Section III)

The input μ table is printed next, with the corresponding computed values of the Sylvester determinant. RES denotes the value of the determinant. SFA and SFR are scale factors on RES (see Appendix II). Sign changes in RES indicate the presence of a flutter point.

The preceding output constitutes general information; the remainder of the output is composed of blocks of computed flutter data, repeated for each flutter point that is found. The first item is the number of iterative interpolations of the Sylvester determinant required to obtain the flutter μ. Following this is a three-rowed tabulation containing the computed flutter point in the first row and the first of the program checks in the second and third rows. In the first row, $Z^{1/3}$ and $1/\mu$ are the computed flutter point pair, LAMDA is the negative of the common root of the characteristic polynomials at flutter, and RES (together with SFA and SFR) is the value of the Sylvester determinant for the flutter μ. The next two rows contain the same data as above except the μ's are the last two bracketing μ's in the iterative interpolation of the Sylvester determinant.* The test criterion is that RES in the second and third rows should be of opposite sign.

The next item printed is the flutter vector in polar form, normalized on the maximum component. R denotes the moduli and THETA the relative phase angles in radians. The vector elements are listed in order of ascending mode number.

The tabulation labelled COMPLEX RESIDUALS is a second program check, obtained by premultiplying the flutter vector by the flutter matrix. The test criterion here is that the "COMPLEX RESIDUALS" vector elements should uniformly be small (ideally zero).

The print-out labelled CHARACTERISTIC EQUATION AT FLUTTER are the coefficients of the real and imaginary characteristic polynomials at flutter. The coefficients of the real polynomial are listed first, beginning with the constant term and progressing toward the higher order terms. The real polynomial is of order MMAX and the leading coefficient is always unity. The imaginary polynomial, printed next, is of order MMAX-1.

The next line of output (containing P, Q, P/P1A and Q/Q1A) is a numerical precision check of the computed flutter point, based on the criterion that the real and imaginary characteristic polynomials must have a common root at flutter. The theoretical basis for this check is described in Appendix IX.

* In the print-out these three values of μ (and/or $1/\mu$) will sometimes be identical because only six digits to the right of the decimal are printed.

The last output block (three lines) is a check to verify that the flutter determinant changes sign in the neighborhood of the computed flutter point. The first line contains the flutter LAMDA and the corresponding value of the complex flutter determinant (real part printed first). The next two lines contain, respectively, 1.02λ and 0.98λ with the corresponding values of the flutter determinant. The test criteria are that the determinant must be of opposite sign in the last two lines, and the absolute value of the determinant in the first line should be less than that in the other two lines.

This completes the description of general output and specific output for a computed flutter point. In the general case, more than one flutter point will be obtained for a given reduced frequency, and there will be a corresponding number of output blocks containing the specific information just described. The output illustrated in Fig. 5, for example, contains three flutter points.

VII. PROGRAM ORGANIZATION

The main program (Deck name GHFLUT) is a dummy program used to control program overlay. It calls two subroutines, GHDUMP and CONTRL. The subprograms called by these two routines are shown in Fig. 6a and b.

Subroutine GHDUMP will generate a tape containing a number of arrays depending only on the elastic constraint properties and the number of bays in spanwise and chordwise directions and the corresponding mode shapes to be used in the analysis. There is no direct (in core) transfer of data from this routine to CONTRL; all information is transferred via the intermediate data tape.

Subroutine CONTRL is the control program for the major part of the flutter calculation. It reads input data describing the cases to be run, computes matrices and other parameters that are needed and calls subroutine EUCLID to find the actual flutter points.

Subroutine EUCLID is a specialization of a general purpose routine for finding pairs of real parameters for which a complex matrix vanishes. A slightly different version of the subroutine has been used to determine stability boundaries in a wing-body flutter analysis [5]. Subroutines CBAR, WROUT and VECNRM are written especially for the present panel flutter analysis. The other subroutines called by EUCLID are general purpose and would likely be suitable for use in other types of flutter calculations. The program still contains some of the control parameters which served to differentiate between the two flutter applications, although for all practical purposes these parameters are constants.

Another vestige of a previous application of the program is seen in the symbols used for μ, i.e., ALPHA, ALP, ALS, AP, AN, AL and for λ (formerly α) i.e., U, UN, UP. This, perhaps confusing, interchange should be noted carefully by the programmer.

In an earlier program version the role of the parameters μ, λ were given to α and μ. The present version is to be preferred, however, in spite of an additional matrix inversion required, because it is much easier for the user to choose a suitable range of tabulated μ values than of the old α values.

A list of all routines and a brief description of the purpose of each is given below.

GHFLUT is the main routine. It is a dummy routine used to control program overlay.

GHDUMP reads the mode numbers, number of spanwise bays, ϵ_x, ϵ_y and the range of values of L, the number of chordwise bays. For each L in the range, information is written on logical tape 9. This information is needed by subroutine CONTRL to set up flutter equations.

FREQEQ finds frequencies and generalized coordinates of the chordwise or spanwise modes.

CDFIND transfers the frequencies of modes requested into an array to be written on tape and finds the quantities $C_{m,\ell}$ and $D_{m,\ell}$. For chordwise modes it also generates the matrices $J_{\bar{m},m}$, $K_{\bar{m},m}$ and $R_{\bar{m},m}$.

GMML finds $G_{\bar{m},m,\ell}(\varphi)$. (See [1], page 52.)

HMML finds $H_{\bar{m},m,\ell}(\varphi)$. (See [1], page 53.)

CONTRL is effectually the main program. It has been made into a subroutine to facilitate program overlay. It calls INK to read one set of data. If necessary GPLUSH is called to read required data from logical tape 9. When the mode numbers required are not taken in sequence from the array on tape 9, the matrices read from tape are compressed to delete undesired modes. The routine then calls BFQST and computes $E^{-1}J$ and $E^{-1}R$. Then for each entry in the table of reduced frequencies (k) the matrices AER and AEI are computed, a value of structural damping (g) is chosen from the table and subroutine EUCLID is called to calculate flutter points. After the return from EUCLID, INK is called and the cycle is repeated.

INK reads input cards for the problems being run. Since problems subsequent to the first may be specified by reading only those parameters which change, control parameters are set in this routine which suppress parts of the computation which would be unnecessarily repetitious.

GPLUSH reads from logical tape 9 values for mode numbers, number of spanwise bays, ϵ_x and ϵ_y . Also read are mode frequencies and several matrices which depend on the number of chordwise bays specified.

BFQST calculates the quantities Q, S and T and the array $B_{n,u}$ F(u) for various values of u. (See [1], Appendices D and E.)

UMKEHR inverts a real matrix.

FINDP calculates $P_u(\xi)$. (See [1], page 46.)

FINDI computes the arrays EYER and EYEI where EYER(MBAR,M) = Real $\left\{\sum_u B_{n,u} F(u) I_{\bar{m},m,u}\right\}$ and EYEI(MBAR,M) = Imag $\left\{\sum_u B_{n,u} F(u) I_{\bar{m},m,u}\right\}$. These are also premultiplied by E^{-1} and stored as AER and AEI. (See [1], page 15.) The integrals $I_{\bar{m},m,u}$ are evaluated using the Newton-Coates five point formula (Bode's rule) [6] four times for each bay.

HD1063 prints a heading for each case, prints parameters identifying the case and that portion of output which is common to all flutter points for the given case.

CLOCK interrogates the internal clock and returns date and time coded as four alphameric words.

EUCLID uses values of μ from a table read into the program to construct the complex $\bar{C}$ matrix. For each of these matrices, Sylvester's resultant is found and tabulated versus μ . This table is then searched for sign changes and for each sign change the program interpolates to find a μ value for which Sylvester's resultant vanishes. A corresponding Z value is found and the pair of values μ , Z represent a flutter point. Some checks are performed and the flutter point data along with flutter vector and figures of merit are printed out.

CBAR generates the matrix $\bar{C}$.

EQCHAR finds the characteristic equation of a double precision matrix. The routine handles 1 x 1 and 2 x 2 matrices as special cases. Larger matrices are first reduced to Hessenberg form [7] then subroutine CHAR is called to find the characteristic equation (see Appendix VII).

CHAR uses a recursion scheme to find coefficients of the characteristic equation of a matrix which is in Hessenberg form.

SCALE generates a matrix whose determinant vanishes when real and imaginary parts of the characteristic equation of the flutter matrix have common roots. In the process of generating this matrix two scale factors are removed, one on the roots of the characteristic equation, the other on the determinant of the matrix. This is necessary because of the enormous magnitude fluctuations encountered without scaling. A recursion based on pivotal reduction is used to allow the resultant to be computed as a determinant of order n rather than $2n-1$.

DETERM finds the determinant of a double precision matrix and scales it to prevent underflow. The true value of the determinant is RES x 2^{NSUM}, where RES and NSUM are subroutine arguments.

WRITER prints a table of μ versus Sylvester's resultant from the real and imaginary parts of the characteristic equation. Flutter points are indicated where this resultant changes sign. Two scale factors are also printed which do not affect the sign.

FLUT finds the second flutter parameter, α , corresponding to the flutter value of μ .*

WROUT prints the flutter point data. The same routine also prints surrounding points used in interpolating for the final flutter point.*

VECTOR computes the flutter vector. The vector is premultiplied by the flutter matrix to give the residuals. Polar form of the vector and residuals are printed. Figures of merit are found for the real and imaginary parts of the characteristic equation at flutter. These are printed along with the characteristic equations. Finally, the determinant is computed for the flutter matrix and again for the same matrix where the eigenvalue which was subtracted on the main diagonal is increased and decreased by 2 per cent. These three determinants are printed.

INVCX inverts a complex matrix.

VECNRM normalizes the flutter vector in accordance with a scheme where each mode shape has unit rms amplitude.

---

* Note as mentioned earlier that the program symbols used for μ suggest α rather than μ .

CXDET finds the determinant of a complex matrix. The determinant is $X + iY$ where X and Y are subroutine arguments.

FMM finds $F_{\bar{m},m}(a_1,a_2,\alpha)$. (See [1], pages 51 - 52.)

FMKT finds the k^{th} derivative of $f_m(\upsilon)$ evaluated at $\upsilon = t$. (See page 51, [1]). The derivative of order zero is considered to be the function itself. This function has three arguments, k , m and t .

BFV evaluates the Bessel function $J_\upsilon(x)$ where $\upsilon = 0$ or 1 .

SUMM is called by BFV to evaluate $J_\upsilon(x)$ when x is greater than or equal to 4.

VIII. SUBMITTING A COMPUTER RUN

The program requires that logical tape 9 be available to the program. Hence, the user must either give instructions to mount a previously generated tape, or to mount a tape to be reserved following the current run, or to mount a scratch tape for use as logical tape 9.

Program deck structure, excluding ID or JOB cards for the particular installation, is shown in the following list.

```
$IBJOB        card
     Deck    GHFLUT
     Deck    FMKT.       - subroutine FMKT
$ORIGIN       EINS
     Deck    GHDUM.      - subroutine GHDUMP
     Deck    GMML.       - subroutine GMML
     Deck    HMML.       - subroutine HMML
     Deck    FMM.        - function FMM
     Deck    CDFIN.      - subroutine CDFIND
     Deck    FREQE.      - subroutine FREQEQ
$ORIGIN       EINS
     Deck    CLOCK.      - subroutine CLOCK
     Deck    LANGLY      - subroutine CONTRL
$ORIGIN       ZWEI
     Deck    INK.        - subroutine INK
     Deck    GPLUS.      - subroutine GPLUSH
     Deck    BFQST.      - subroutine BFQST
     Deck    UMKEH.      - subroutine UMKEHR
     Deck    SUMM.       - function SUMM
     Deck    BFV.        - function BFV
```

```
        Deck    PFIND        - subroutine FINDP
        Deck    IFIND        - subroutine FINDI
        Deck    HD1063       - subroutine HD1063
$ORIGIN          ZWEI
        Deck    CBAR.        - subroutine CBAR
        Deck    CHAR.        - subroutine CHAR
        Deck    EQCHA.       - subroutine EQCHAR
        Deck    SCALE.       - subroutine SCALE
        Deck    DETER.       - subroutine DETERM
        Deck    WRITE.       - subroutine WRITER
        Deck    AFLUT        - subroutine FLUT
        Deck    WROUT.       - subroutine WROUT
        Deck    INVCX.       - subroutine INVCX
        Deck    VECNR.       - subroutine VECNRM
        Deck    CXDET.       - subroutine CXDET
        Deck    PVECT        - subroutine VECTOR
        Deck    EUCLI.       - subroutine EUCLID
$ENTRY           GHFLUT
        Intermediate tape data card
        Data for first case
        Blank card
        Data for second case
        Blank card
        .
        .
        .
        Data for last case
        Blank card
        Extra blank card or end of file
```

Computer running time depends on a variety of factors. The following guidelines may be useful in estimating running time until the user has enough experience with his own type of applications to make better estimates.

A four-mode calculation (one frequency-damping combination) takes approximately 30 seconds (based on an expected average of 2 - 3 flutter points). A corresponding six-mode calculation requires about 60 to 75 seconds (4 - 5 flutter points on the average). If logical tape 9 is not available from a previous run, the running time estimate should include, in addition to the above, about L^2 minutes for each value of L for which a set of data will be written on tape 9.

APPENDIX I

FIGURES 1 - 6

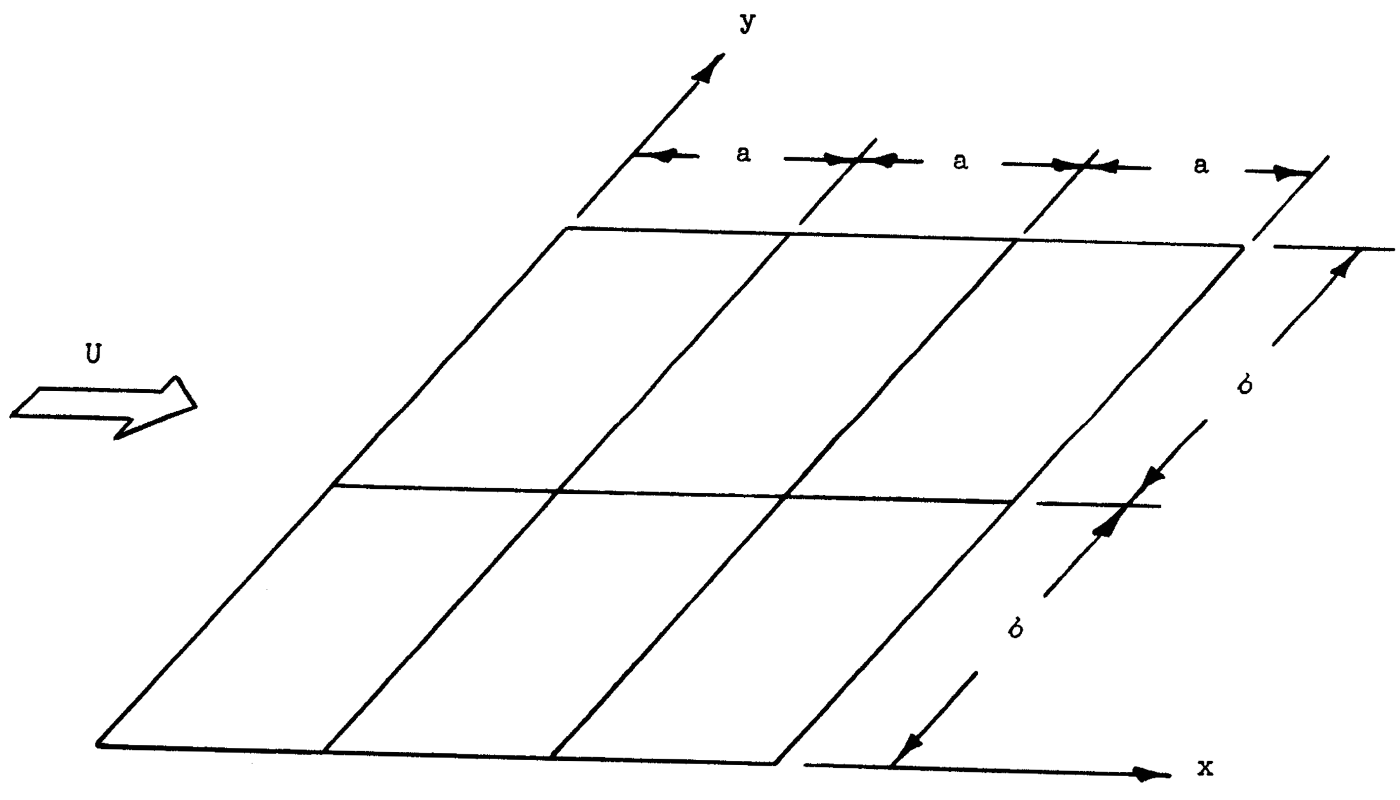


Fig. 1 - Typical Finite Panel Array

```
01030102030407080910111201O1     0.         0.
01          .0000001
02          .25
0304
04         1.8225
05          .01
0602
070105        2.        3.        4.        5.        6.
070610        7.        8.        9.       10.       11.
071115       12.       14.       16.       18.       20.
071620       22.       24.       26.       28.       30.
072125       32.       35.       38.       41.       44.
072630       47.       50.       54.       58.       62.
073135       66.       70.       74.       78.       82.
073640       86.       90.       95.      100.      105.
074144      110.      125.      135.      150.
0844
090102       1.0       1.5
1002
1101
1220
(BLANK)
02          .5
030401                                                        01040709
04         1.44
05          .015
0603
090102        .7        .8
1002
1102
1201
(BLANK)
(BLANK)
```

Fig. 2a - Sample Input Data Set 1

```
(BLANK)
        01         .0000001
        02          1.
        0304
        04          1.69
        05          0.
        0601
        070105       2.        3.        4.        5.        6.
        070610       7.        8.        9.       10.       11.
        071115      12.       14.       16.       18.       20.
        071620      22.       24.       26.       28.       30.
        072125      32.       35.       38.       41.       44.
        072630      47.       50.       54.       58.       62.
        073135      66.       70.       74.       78.       82.
        073640      86.       90.       95.      100.      105.
        074144     110.      125.      135.      150.
        0844
        090103       1.0       1.2       1.4
        1003
        1105        .25
        1215
(BLANK)
        04          1.44
(BLANK)
(BLANK)
```

Fig. 2b - Sample Input Data Set 2

```
0101010203040506070809100102
01               .0000001
02               .25
0304
04               1.8225
05               .01
0601
070105             2.         3.         4.         5.         6.
070610             7.         8.         9.        10.        11.
071115            12.        14.        16.        18.        20.
071620            22.        24.        26.        28.        30.
072125            32.        35.        38.        41.        44.
072630            47.        50.        54.        58.        62.
073135            66.        70.        74.        78.        82.
073640            86.        90.        95.       100.       105.
074144           110.       125.       135.       150.
0844
090101            1.5
1001
1101
1220
(BLANK)
(BLANK)
```

Fig. 3 - Input Data Listing for Sample Output Case (Fig. 5)

INTERMEDIATE TAPE CARD

(Blank if tape is available from a previous run)

COLUMNS						
1-2	3-4	5-24	25-26	27-28	29-38	39-48
LMIN	LMAX	10 chordwise mode numbers in ascending order	Number of spanwise bays	Spanwise mode number	ϵ_x	ϵ_y

NUMBERED DATA CARDS

1-2	3-4	5-6	7-10	11-20	21-30	31-40	41-50	51-60	61-80
01				TESTR					
02				Inverse of aspect ratio					
03	Number of chord-wise modes	If zero or blank, consecutive modes starting with first are used							Chordwise mode numbers in ascending order if cols. 5-6 are not blank or zero
04				Mach number squared					
05	Number of g values (blank taken as 1)			g	g	g	g	g	
06	Number of chord-wise bays			Nonzero will suppress print out of matrices					
07	Index of first μ on this card	Index of last μ on this card		μ	μ	μ	μ	μ	
08	Total number of entries in μ table								
09	Index of first k on this card	Index of last k on this card		k	k	k	k	k	
10	Total number of k's								
11	Option number (must be 1,2,4 or 5)			Distance from panel edge to tunnel wall in panel spans					
12	Number of nonzero terms in spanwise expansion								

Fig. 4 - Data Format for User Reference

```
M.R.I. FLUTTER PROGRAM     DATE=00000000     TIME=0000000                                                      1
   M          S        E(SPAN)  E(CHORD)      TESTR       L        MMAX        N        UMAX   SPANWISE MODE
 1.3500    0.2500     0.         0.         0.10E-06     1          4          1         20          2

CASE     A/2       1/DPHI      K          G
 1     1.0000        16      1.5000     0.0100

GAMMA-BAR=  6.2831853

CHORD-WISE MODE  1            2            3           4
          3.1415924   6.2831853   9.4247772 12.5663706

BN(U)
  1.757877 -0.483447 -0.197721 -0.095521 -0.040572 -0.008510  0.009204  0.017047  0.018142  0.015011
  0.009754  0.004050 -0.000883 -0.004308 -0.005952 -0.005927 -0.004627 -0.002599 -0.000411  0.001449

INVERSE OF ELASTIC MATRIX TIMES INERTIA MATRIX
 6.5702300E-03 -0.             -6.7842637E-10  0.
-0.             5.6835890E-04 -0.             -5.3801797E-13
 6.3923710E-10 -0.              1.1998231E-04  0.
-0.             5.1632369E-13 -0.              3.8877094E-05

INVERSE OF ELASTIC MATRIX TIMES REAL PART AEROD MATRIX
-4.8284101E-03 -1.4912236E-02  3.4849515E-03 -5.4058474E-03
 1.2899855E-03 -1.0846157E-03 -3.2866663E-03 -1.2090123E-04
 6.3639989E-05  6.9382390E-04 -9.2668995E-05 -9.2554983E-04
 3.1987256E-05 -8.2699311E-06  2.9989975E-04 -2.7647084E-06

INVERSE OF ELASTIC MATRIX TIMES IMAG PART AEROD MATRIX
 9.1891894E-03  6.9303840E-03  1.5778516E-03  1.0541559E-03
-5.9951404E-04  8.5195765E-05  6.4234972E-04  3.7153926E-04
 2.8814861E-05 -1.3560132E-04 -1.4406262E-04 -7.9678368E-06
-6.2376079E-06  2.5414165E-05  2.5818019E-06 -2.3953760E-05
    MU        RES      SFA SFR      MU         RES      SFA SFR      MU         RES      SFA SFR      MU          RES      SFA SFR
  2.00E 00  -1.56E 00 -8  -6     1.40E 01   4.45E-01 -6  -9     3.80E 01  -1.20E 00 -4 -25    7.60E 01   1.48E 00 -3 -24
  3.00E 00  -2.10E 00 -7 -19     1.60E 01   1.14E 00 -5 -24     4.10E 01  -1.64E 00 -4 -24    8.20E 01   8.98E-01 -3 -22
  4.00E 00  -1.28E 00 -7 -15     1.80E 01   1.12E 00 -5 -22     4.40E 01  -8.99E-01 -4 -22    8.60E 01   5.18E-01 -3 -20
  5.00E 00  -6.72E-01 -7 -11     2.00E 01   1.73E 00 -5 -21     4.70E 01  -7.22E-01 -4 -21    9.00E 01   1.14E 00 -3 -20
  6.00E 00  -8.77E-01 -7  -9     2.20E 01   5.44E-01 -5 -18     5.00E 01  -8.29E-01 -4 -22    9.50E 01   7.21E-01 -3 -18
  7.00E 00  -6.03E-01 -7  -7     2.40E 01   1.07E 00 -5 -18     5.40E 01   6.55E-01 -4 -16    1.00E 02   6.65E-01 -3 -17
  8.00E 00   8.86E-01 -6 -25     2.60E 01   7.00E-01 -5 -17     5.80E 01   1.11E 00 -4 -16    1.05E 02   9.87E-01 -3 -16
  9.00E 00   7.18E-01 -6 -19     2.80E 01  -8.56F-01 -5 -20     6.20E 01   1.20F 00 -4 -14    1.10E 02   5.40E-01 -3 -14
  1.00E 01   1.09E 00 -6 -17     3.00E 01  -2.23E 00 -5 -16     6.60E 01   2.11E 00 -3 -29    1.25E 02   5.69E-01 -3 -11
  1.10E 01   2.27E 00 -6 -16     3.20E 01  -5.79E-01 -5 -12     7.00E 01   1.63E 00 -3 -27    1.35E 02   2.03E 00 -2 -27
  1.20E 01   9.51E-01 -6 -13     3.50E 01  -1.39E 00 -4 -27     7.40E 01   1.14E 00 -3 -25    1.50E 02   1.44E 00 -2 -24
   48 ITERATIONS

      MU           Z**1/3              LAMDA           RES        SFA SFR     1/MU
   7.923775      0.28041100      8.33269000E-04     5.248E-01     -6 -39    0.126202
   7.923775      0.28041099      8.33268988E-04    -7.836E-01     -6 -42    0.126202
   7.923775      0.28041100      8.33269000E-04     5.248E-01     -6 -39    0.126202

VECTOR R          THETA              COMPLEX RESIDUALS
   0.097796     3.069003           -4.66E-10    7.28E-11
   0.200997     6.236609           -3.82E-11    1.09E-11
   0.658708     3.084337           -2.91E-11    4.55E-13
   1.000000     0.                 -8.37E-11   -1.82E-11

CHARACTERISTIC EQUATION AT FLUTTER
  1.9683872E-09    3.5110776E-06    1.4812592E-03    1.2390968E-01    1.0000000E 00
```

Fig. 5 - Sample Output Listing

```
 -1.9117971E-10  -3.5286575E-07  -1.5675161E-04  -1.0346376E-02

P= 1.4583E-16 Q= 1.1065E-17 P/P1A=-1.3460E-07 Q/Q1A= 1.1732E-07
LAMDA=     0.0008   DET=  1.3125E-16 1.3747E-17
LAMDA=     0.0008   DET= -2.1310E-11 1.8500E-12
LAMDA=     0.0008   DET=  2.1964E-11-1.9227E-12
   43 ITERATIONS

      MU           Z**1/3           LAMDA          RES         SFA SFR    1/MU
   27.883657     0.39191253    2.27492735E-03   -1.292E 00    -5 -34   0.035863
   27.883657     0.39191253    2.27492738E-03   -1.154E 00    -5 -33   0.035863
   27.883655     0.39191251    2.27492720E-03    1.063E 00    -5 -38   0.035863

VECTOR R         THETA            COMPLEX RESIDUALS
   0.017096    3.049019          -2.91E-10   4.37E-11
   0.017926    0.455868           6.37E-12   1.09E-11
   0.196811    3.121018          -1.46E-11  -8.53E-14
   1.000000    0.                 2.48E-10   8.90E-12

CHARACTERISTIC EQUATION AT FLUTTER
  1.9957785E-07   1.2297253E-04   1.6444874E-02   4.2113063E-01   1.0000000E 00
 -8.2005277E-09  -5.1338555E-06  -7.0245779E-04  -1.3318586E-02

P=-1.3527E-14 Q= 9.4869E-18 P/P1A= 1.0882E-07 Q/Q1A= 1.9445E-09
LAMDA=     0.0023   DET= -1.3759E-14 5.4410E-17
LAMDA=     0.0023   DET= -2.4580E-09 9.6310E-11
LAMDA=     0.0022   DET=  2.5143E-09-9.8841E-11
   35 ITERATIONS

      MU           Z**1/3           LAMDA          RES         SFA SFR    1/MU
   50.830359     0.69252619    1.25518787E-02   -7.530E-01    -4 -32   0.019673
   50.830359     0.69252619    1.25518787E-02   -7.530E-01    -4 -32   0.019673
   50.830360     0.69252620    1.25518790E-02    7.288E-01    -4 -33   0.019673

VECTOR R         THETA            COMPLEX RESIDUALS
   0.006969    0.199852          -5.18E-09  -1.30E-09
   0.070176    2.956780           4.40E-09  -5.49E-10
   1.000000    0.                -2.91E-11   1.14E-11
   0.035272   -3.151118           1.56E-07   5.86E-09

CHARACTERISTIC EQUATION AT FLUTTER
  2.1021558E-06   7.2234060E-04   5.3622967E-02   7.6282816E-01   1.0000000E 00
 -8.2196063E-08  -2.5618670E-05  -1.7293702E-03  -1.6735561E-02

P=-1.4753E-13 Q= 2.9833E-15 P/P1A=-4.3347E-08 Q/Q1A=-2.4045E-08
LAMDA=     0.0126   DET= -1.6315E-13 2.3936E-15
LAMDA=     0.0128   DET=  6.9688E-08-2.5505E-09
LAMDA=     0.0123   DET= -6.6431E-08 2.4120E-09
```

Fig. 5 - (Concluded)

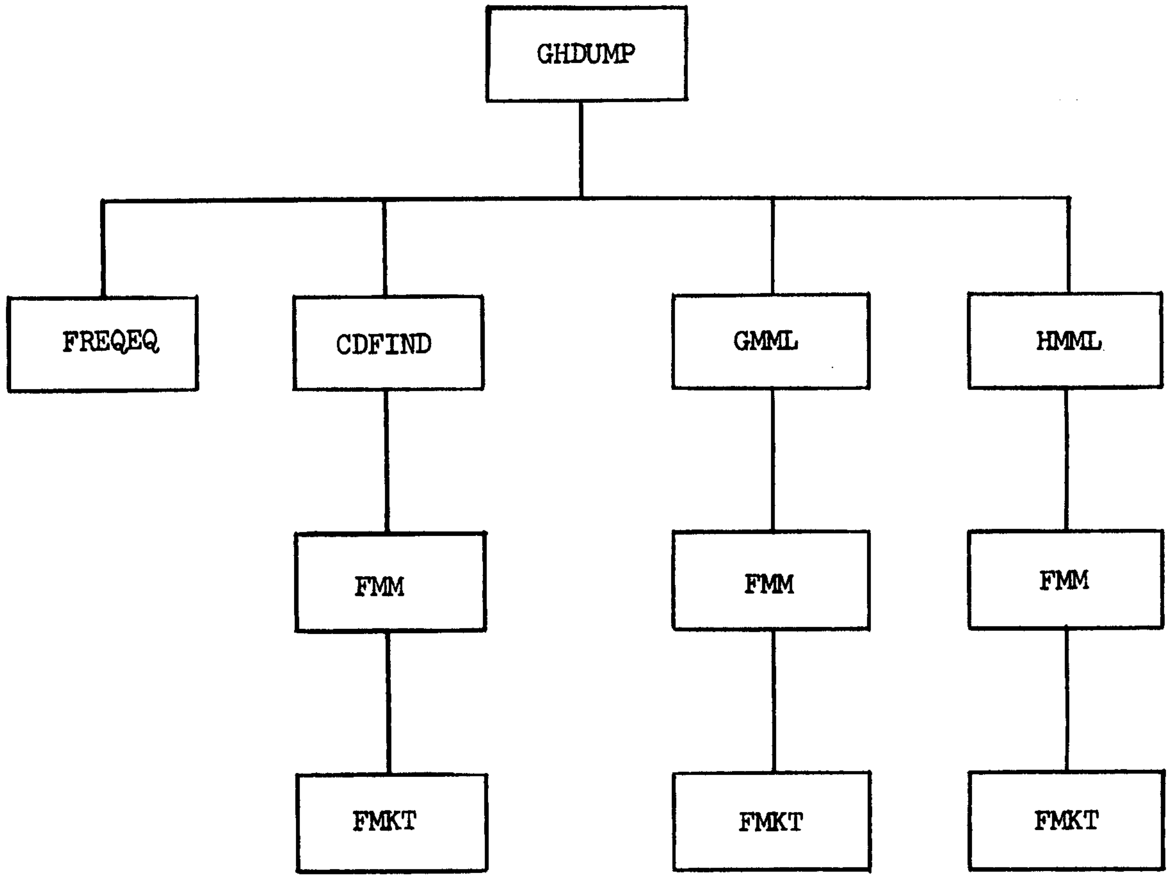


Fig. 6a - Hierarchy of Subroutines

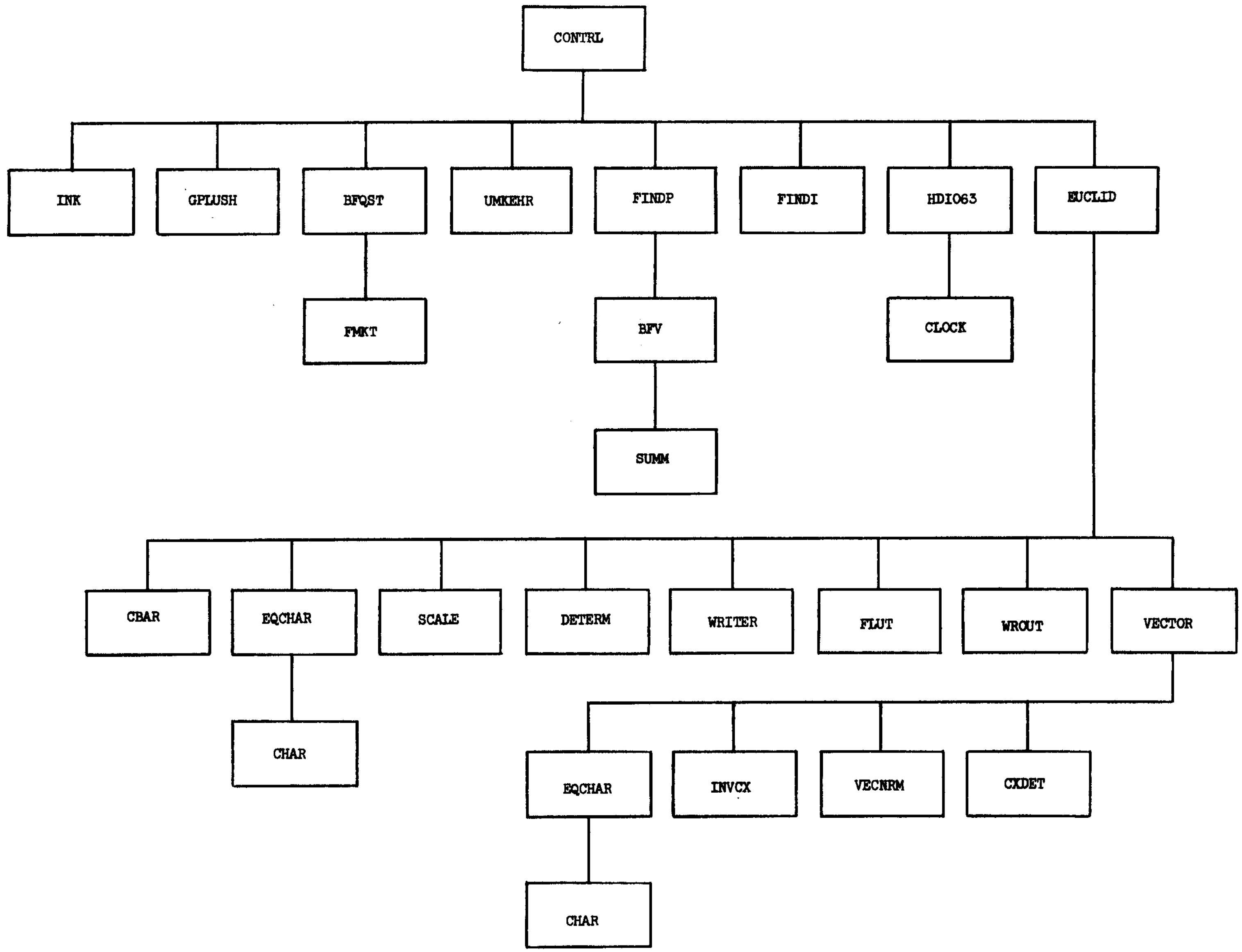


Fig. 6b - Hierarchy of Subroutines (Concluded)

APPENDIX II

DESCRIPTION OF SCALING ROUTINES

Experience with an earlier version of the program showed that the range of magnitudes of Sylvester's resultant of the real and imaginary parts of the characteristic equation can be extremely wide. The object of the program is to find μ values which will cause the resultant to vanish. Yet for the initial values read into the μ table the resultant may be so large as to cause overflow. Hence scaling has been used to keep these numbers in a range that the machine can handle. Binary scaling was used to avoid any loss of accuracy in scaling.

Two kinds of scale factors are used. One of these scales the roots of the characteristic equation; the other scales the reduced determinant. The scale factor applied to the roots of the characteristic equation is based on the real part of the equation and is chosen so that the magnitudes of the nonzero polynomial coefficients cluster around unity. The scale factor applied to the determinant is the product of several scale factors -- one for each row of the determinant except the last -- and is likewise chosen so that the calculated resultant is of the order of magnitude of unity.

The program prints the base 2 logarithms of these two scale factors along with a scaled resultant value. The logarithms of the scale factor for the characteristic equation and the determinant are labeled SFA and SFR , respectively. The true value of the resultant can be obtained from the printed (scaled) resultant from

$$RES_{true} = RES_{scaled} \cdot 2^{n^2 SFA+SFR}$$

where n is the number of modes used in the analysis. No attempt is made to print out the true resultant since much of the time this number is outside the range of magnitudes which the machine can handle.

APPENDIX III

INTERMEDIATE TAPE FORMAT

The program obtains certain essential information from a binary intermediate tape. The program references this tape as logical tape 9. If no previously generated tape containing the required data is mounted, an input card must be punched which will cause the program to generate the tape at the beginning of the current run.

Information on logical tape 9 consists of one or more sets of data, one for each value of L from LMIN to LMAX. The upper bound for LMAX is 4 in the present program. The symbol L represents the number of chordwise bays in the panel array under study. Each set of data on the tape consists of the following sequence of seven records:

Record	Length	List	Description
1	16 words	L	Number of chordwise bays
		(MODE(J),J=1,10)	Chordwise mode numbers
		NSP	Number of spanwise bays
		NGBAR	Spanwise mode number
		ϵ chord	Chordwise stiffness parameter
		ϵ span	Spanwise stiffness parameter
		16	Intervals used in integration
2	77 words	(GAMMA(J),J=1,11)	Frequencies γ_j, and $\bar{\gamma}$
		((C(M,K),M=1,11),K=1,6)	$C_{m,\bar{\ell}}$
3	66 words	((D(M,K),M=1,11),K=1,6)	$D_{m,\bar{\ell}}$
4	100 words	((DJAY(MB,M),MB=1,10), M=1,10)	$J_{\bar{m},m}$
5	100 words	((CAY(MB,M),MB=1,10), M=1,10)	$K_{\bar{m},m}$
6	100 words	((ARE(MB,M),MB=1,10), M=1,10)	$R_{\bar{m},m}$
7	8,000 words	(((GH(MB,M,K),MB=1,10), M=1,10),K=1,80)	$\sum_{\ell=1}^{L}\left[G_{\bar{m},m,\ell}(K\Delta\varphi) + H_{\bar{m},m,\ell-1}(K\Delta\varphi)\right]$

In the arrays GAMMA, C and D the subscript value of 11 refers to the spanwise mode while subscripts 1 - 10 refer to chordwise modes.

In order to form the integrals $I_{\bar{m},m,u}$ (See [1] , page 52) it is necessary to integrate the products of $P_u(\ell-1+\varphi)$ times $G_{\bar{m},m,\ell}(\varphi)$ and $H_{\bar{m},m,\ell}(\varphi)$.

To conserve storage the sum of $G_{\bar{m},m,\ell}$ and $H_{\bar{m},m,\ell}$ is generated and stored. For $\ell = L$, note that $H_{\bar{m},m,\ell}$ is zero. The subscripts of GH(MB,M,K) correspond respectively to $\bar{m}$,m and an index representing the distance from the panel array leading edge measured in terms of the increment used in the numerical integration. Thus since $\Delta\varphi = 1/16$, K ranges from 1 to 16L. Since the program does not presently have the capability of analyzing 5-bay configurations, the dimension of the array GH could be reduced to GH(10,10,64) to gain additional storage space for the program.

Following the data corresponding to $L = L_{max}$, the tape has one 16-word record indicating L = 5. This is presently used to signal end of file.

When data for a flutter case have been read, the program searches logical tape 9 for the data set corresponding to the value of L specified on the 06 data card.

APPENDIX IV

EXPLANATION OF EXCEPTIONAL COMMON STATEMENTS

Routines SUMM, FMKT, FMM, GMML and HMML have short COMMON tables since these routines reference only a few symbols near the beginning of the COMMON list.

In routines GMML, HMML, FREQEQ, CDFIND and GHDUMP an "O" is used in place of "L" in the COMMON table since L is used either in argument lists or as a DO loop index.

In routines FREQEQ and GHDUMP an "ON" is used in place of "BN" since BN is used in FREQEQ as a scratch variable.

Since data generated by GHDUMP are preserved on logical tape 9, and not in COMMON storage, the COMMON table for the overlay link containing GHDUMP does not agree with that of the later overlay links. In particular, the later routines do not contain the arrays CC, CD, DD.

APPENDIX V

LIST OF COMMON AND ARGUMENT LIST QUANTITIES ALTERED BY VARIOUS SUBPROGRAMS

Subroutine GHDUMP and the subprograms it calls communicate with the rest of the program through logical tape 9 and not through COMMON storage. Therefore, the list below is divided to show these routines separate from the other routines.

Subroutines in Overlay Containing GHDUMP

GHDUMP - reads MODE (1 - 10), NSP, NGBAR, EPY, EPX and minimum and maximum values of L . It also defines NODP = 16.

FREQEQ - computes FREQ, QMM arrays.

CDFIND - computes DJAY, CAY, ARE, C, D, CC, CD, DD arrays.

GMML - computes GMM.

HMML - computes HMM.

Subroutines not in Overlay Containing GHDUMP

CONTRL - is effectually the main routine. It does not alter any quantities.

INK - is the input routine. Quantities defined are: EMSQ, EM, BSQ, BETA, BFOR, S, L, MMAX, NNN, A, TESTR, NEWP, NEWGAM, NEWC, NEWEIV, NEWMOD, NNST, NEWGMB, NEWDPH, MAXAL, NK, NG, and the arrays ALPHA, CKAY, GT, MODE, and SAVE.

GPLUSH - reads appropriate data from logical tape 9 into the program. This includes NSP, NGBAR, EPY, EPX, NODP, and the arrays MODE, GAMMA, C, D, DJAY, CAY, ARE and GH.

BFQST - defines Q, SS, T, BN, BUF.

UMKEHR - inverts matrix R in argument list.

PFIND - defines PREAL, PIMAG.

IFIND - defines AER, AEI, EYER, EYEI.

HD1063 - increments NSEQ.

EUCLID - computes array REST in the argument list.

CBAR - defines array CR in the argument list.

EQCHAR - defines array PRE in the argument list.

CHAR - defines array D in the argument list.

SCALE - defines NS, KRUB and array RR in the argument list.

DETERM - redefines NSUM, RES in the argument list.

FLUT - defines U, RES in the argument list.

INVCX - inverts complex matrix R in argument list.

CXDET - computes X, Y in argument list.

VECTOR - redefines array C in argument list.

VECNRM - defines array VM in argument list.

APPENDIX VI

TABLE OF PROGRAM SYMBOLS

Program Symbol	Noncommon Reference	Report Notation	Comment
A		A	Twice the distance from panel edge to wind tunnel wall.
AEI(MB,M)		$Im(E^{-1}C)$	See Section III.
AER(MB,M)		$Re(E^{-1}C)$	See Section III.
ALPHA(J)			Table of μ values.
ARE(MB,M)		$R_{\bar{m},m}$	See page 12, [1]*.
BETA		β	$\sqrt{M^2-1}$
BFOR		β^4	
BN(N)		$B_{n,u}$	Coefficients of spanwise mode expansion. See Appendix E, [1].
BSQ		β^2	
BUF(N)		$B_{n,u}F(u)$	See Appendix E, [1].
C(M,L)		$C_{m,\ell}$	See pages 43 - 44, [1].
CAY(MB,M)		$K_{\bar{m},m}$	See page 4, [1].
CC(M,N)		$\sum C_{m,\ell}C_{n,\ell}$	Summation over chordwise bays.
CD(M,N)		$\sum C_{m,\ell}D_{n,\ell}$	Summation over chordwise bays.
CK		k	Reduced frequency.
CKAY(J)			Table of k values.
CR(MB,M)	EUCLID	$Re(\bar{C})$ and $Im(\bar{C})$	Flutter matrix. See Section III.
D(M,L)		$D_{m,\ell}$	See pages 43 - 44, [1].
DD(M,N)		$\sum D_{m,\ell}D_{n,\ell}$	Summation over chordwise bays.
DJAY(MB,M)		$J_{\bar{m},m}$	See page 4, [1].
EIJ(MB,M)		$E^{-1}J$	Matrix product EINV * DJAY
EINV(MB,M)		$E^{-1}=(E_{\bar{m},m})^{-1}$	See page 15, [1].
EIR(MB,M)		$E^{-1}R$	Matrix product EINV * ARE
EM		M	Mach number.
EMSQ		M^2	
EPX		ϵ_y	Torsional proportionality constant (Section II).
EPY		ϵ_x	Torsional proportionality constant (Section II).
EYEI(MB,M)		$S\cdot B_{\bar{m},m}$	See page 15, [1].
EYER(MB,M)		$S\cdot A_{\bar{m},m}$	See page 15, [1].*

* $R_{\bar{m},m}$ and $A_{\bar{m},m}$ are misprinted in [1]. They should read as follows:

$$R_{\bar{m},m} = \int_0^L \Phi_m^{(x)}(x)\Phi_{\bar{m}}(x)dx$$

$$\left\{A_{\bar{m},m}\right\} = \text{Real}\left\{\sum_u B_{n,u}F(u)I_{\bar{m},m,u}/S\right\}$$

Program Symbol	Noncommon Reference	Report Notation	Comment
FREQ(M)			Frequencies of all modes in range of interest.
G		g	Structural damping coefficient.
GAMMA(M)		γ_m	Frequencies of modes used. See pages 40 - 41, [1].
GH(MB,M,K)			See Appendix III.
JEST(J)	EUCLID		Scale factor table (SFR table).
KRUB	EUCLID		Scale factor SFA.
KIT	EUCLID		Count of iterative interpolations needed to find flutter.
L		L	Number of chordwise bays.
LEST(J)	EUCLID		Scale factor table (SFA table).
MMAX			Number of chordwise modes.
MODE(J)			Chordwise mode numbers.
NCASE			Edge condition option number. See Section V.
NGBAR			Spanwise mode number.
NNN		U_{max}	Number of terms in spanwise mode expansion.
NODP			Constant = 16.
NS	EUCLID		Scale factor SFR.
NSP			Number of spanwise bays.
PIMAG		$Im(P_u(\xi))$	See page 10, [1].
PREAL		$Re(P_u(\xi))$	See page 10, [1].
PRE(J)	EUCLID		Characteristic equation.
Q		Q	See page 4, [1].
QMM(M)		q_m	See pages 39 - 41, [1].
REST(J)	CONTRL		Resultant table.
S		s	Reciprocal of aspect ratio.
SFOR		s^4	
SQK		k^2	
SQS		s^2	
SS		S	See page 4, [1].
T		T	See page 4, [1].
TESTR			Flutter precision control parameter. See Appendix IX.

APPENDIX VII

CHARACTERISTIC EQUATION ROUTINE

To set up the Sylvester determinant it is necessary to calculate the characteristic polynomial of the flutter matrix. An earlier version of the program made use of the Danilewski method [8], which was found to be numerically unstable [7]. The present routine calculates the polynomial in two steps. The flutter matrix is first converted to the almost triangular, or Hessenberg, form via a series of numerically stable operations [7].* The coefficients of the characteristic polynomial are then obtained using a recurrence relation derived from the general induction formula given in [9]. The two-step process proceeds as follows.

The flutter matrix is a full m x m matrix which is represented as

$$F = \begin{Bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1m} \\ A_{21} & \cdots & \cdots & \cdots & A_{2m} \\ \vdots & & & & \vdots \\ A_{m1} & \cdots & \cdots & \cdots & A_{mm} \end{Bmatrix}$$

This matrix is first transformed to the Hessenberg form

$$S = \begin{Bmatrix} s_{11} & s_{12} & s_{13} & \cdots & & s_{1m} \\ s_{21} & s_{22} & \cdots & \cdots & & s_{2m} \\ 0 & s_{32} & s_{33} & \cdots & & \cdot \\ 0 & 0 & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & \cdots & & s_{m,m-1} & s_{mm} \end{Bmatrix}$$

* The operations consist in similarity transformations which do not alter the characteristic polynomial.

Now let $d_{n,p}$ be the coefficient of λ^p in the characteristic polynomial of the n x n submatrix taken from the upper left hand corner of S . The recurrence relation for computing the coefficients is

$$d_{n,p} = d_{n-1,p-1} - \sum_{k=1}^{n-p} v_{n-1,n-k+1}\, s_{n-k+1,n}\, d_{n-k,p} \qquad \text{(VII-1)}$$

where
$$d_{j,j} = 1 \quad \text{(including } j = 0\text{)}$$
$$d_{j,k} = 0 \quad \text{for } j < k$$
$$d_{j,k} = 0 \quad \text{for } k < 0$$
$$d_{j,j-1} = -\sum_{i=1}^{m} s_{i,i}$$
$$v_{j-1,j} = 1$$
$$v_{j,k} = \prod_{i=k}^{j} s_{i+1,i}$$

An mxm+1 d matrix is obtained by application of Eq. VII-1 through the ranges

$$n = 1,2,3 \,\ldots\ldots m$$

$$p = 0,1,2 \,\ldots\ldots m$$

The m + 1 elements of the m^{th} row of the d matrix are the required coefficients of the characteristic polynomial of the m x m matrix F .

APPENDIX VIII

ADMISSIBLE VALUES FOR THE TORSIONAL RESTRAINT PROPORTIONALITY FACTOR ϵ_y

In the case of the finite span array (Option 1) a restriction is imposed on ϵ_y by the sine series expansion of the spanwise deflection shape. The coefficients in the expansion are of the form*

$$B_{n,u} = \frac{2\,B^3}{B^4\bar{\gamma}_n^4 - u^4\pi^4}\,\Phi$$

where
$2B$ = wavelength of expansion = $\lambda + 2N$
$\lambda = sL/2M$
s = inverse of aspect ratio
L = number of chordwise panels
M = Mach number
N = number of spanwise panels
$\bar{\gamma}_n$ = frequency of spanwise mode
n = number of the spanwise mode
u = summation index
Φ = a transcendental function whose form is not relevant to this discussion

The frequency $\bar{\gamma}_n$ depends on ϵ_y and N whereas B depends on N , s , L and M . It is obviously possible to choose values for the above quantities such that the denominator $B^4\bar{\gamma}_n^4 - u^4\pi^4$ vanishes, in which case the computer program will give an erroneous result. An inadvertent choice of such a combination of parameters will be indicated in the print-out of the expansion coefficients.

The restriction on ϵ_y (or more precisely on the combination ϵ_y , N , s , L , M) could be eliminated by causing the program to detect such a situation and artificially set B to a somewhat greater value. It is felt, however, that this modification is unwarranted because of the small probability that the situation will occur.

For the array with span extending to infinity (Option 2) the expansion coefficients contain the term $\bar{\gamma}_n^4 - u^4\pi^4$ in the denominator which vanishes for $\epsilon_y \equiv 0$; $u = n$. This degenerate case is taken care of by program logic. However, it is recommended that values $0 < \epsilon_y \ll 1$ be avoided to avert possible numerical difficulties associated with a vanishingly small (but not precisely zero) value of the denominator.

---

* See [1], Appendix E, Equation (E-7).

APPENDIX IX

DESCRIPTION OF A PROGRAM CHECK AND THE PRECISION FACTOR TESTR

The purpose of this Appendix is to discuss the program check denoted in the output listing by the quantities P , Q , P/P1A and Q/Q1A (following the characteristic equation print out), and the precision factor TESTR.

The basis for the check is the fact that, at flutter, the real and imaginary characteristic polynomials of the flutter matrix must have a common root. Referring to Section III, the flutter matrix may be written as

$$\left\{\lambda I - \mu\beta k^2(1-jg)E^{-1}J + (1-jg)E^{-1}C\right\}$$

where $\lambda = Z\beta(1+g^2)/24$ (as handled in the program). Since μ and λ can be considered independent variables with all other quantities fixed, the characteristic polynomials may be written as

$$\text{Real polynomial} = P(\lambda,\mu) = a_m\lambda^m + \;.....+a_1\lambda + a_o$$

$$\text{Imag. polynomial} = Q(\lambda,\mu) = b_{m-1}\lambda^{m-1} + \;.....\; +b_1\lambda + b_o$$

where the a_i and b_i are real-valued functions of μ only.

Let $\lambda_F = \lambda$ at flutter

$\mu_F = \mu$ at flutter

Then the check criteria stated above may be written as

$$P(\lambda_F,\mu_F) = Q(\lambda_F,\mu_F) = 0$$

The quantities appearing in the program check can now be defined as follows:

$$P = P(\lambda_F,\mu_F)$$

$$Q = Q(\lambda_F,\mu_F)$$

$$\frac{P}{P1A} = \frac{P(\lambda_F,\mu_F)}{\lambda_F \left.\frac{\partial P}{\partial \lambda}\right|_{\lambda_F,\mu_F}}$$

$$\frac{Q}{Q1A} = \frac{Q(\lambda_F,\mu_F)}{\lambda_F \left.\frac{\partial Q}{\partial \lambda}\right|_{\lambda_F,\mu_F}}$$

Now P/P1A and Q/Q1A represent the Newton-Raphson recurrence relation for iteration of the polynomials with respect to λ . For example, if P is iterated holding μ fixed at μ_F , and λ_F is the current estimate for λ , then the Newton-Raphson next approximation for λ is λ_F (1 - P/P1A).

The quantities P/P1A and Q/Q1A cannot be interpreted as a percentage error. However, experience with the computer program has shown that values less than about 10^{-3} imply at least four-digit accuracy in the computed values of $1/\mu$ and $z^{1/3}$. On the other hand, values of the order 10^{-1} imply breakdown of the computational techniques employed.

The input parameter TESTR is an upper bound on the acceptable decimal per cent error in μ . It is used in the program as follows. In searching for a zero of the Sylvester determinant the current bracketing values of μ are used to interpolate for the next approximation for μ . If

μ_1 = current μ value for which determinant is positive

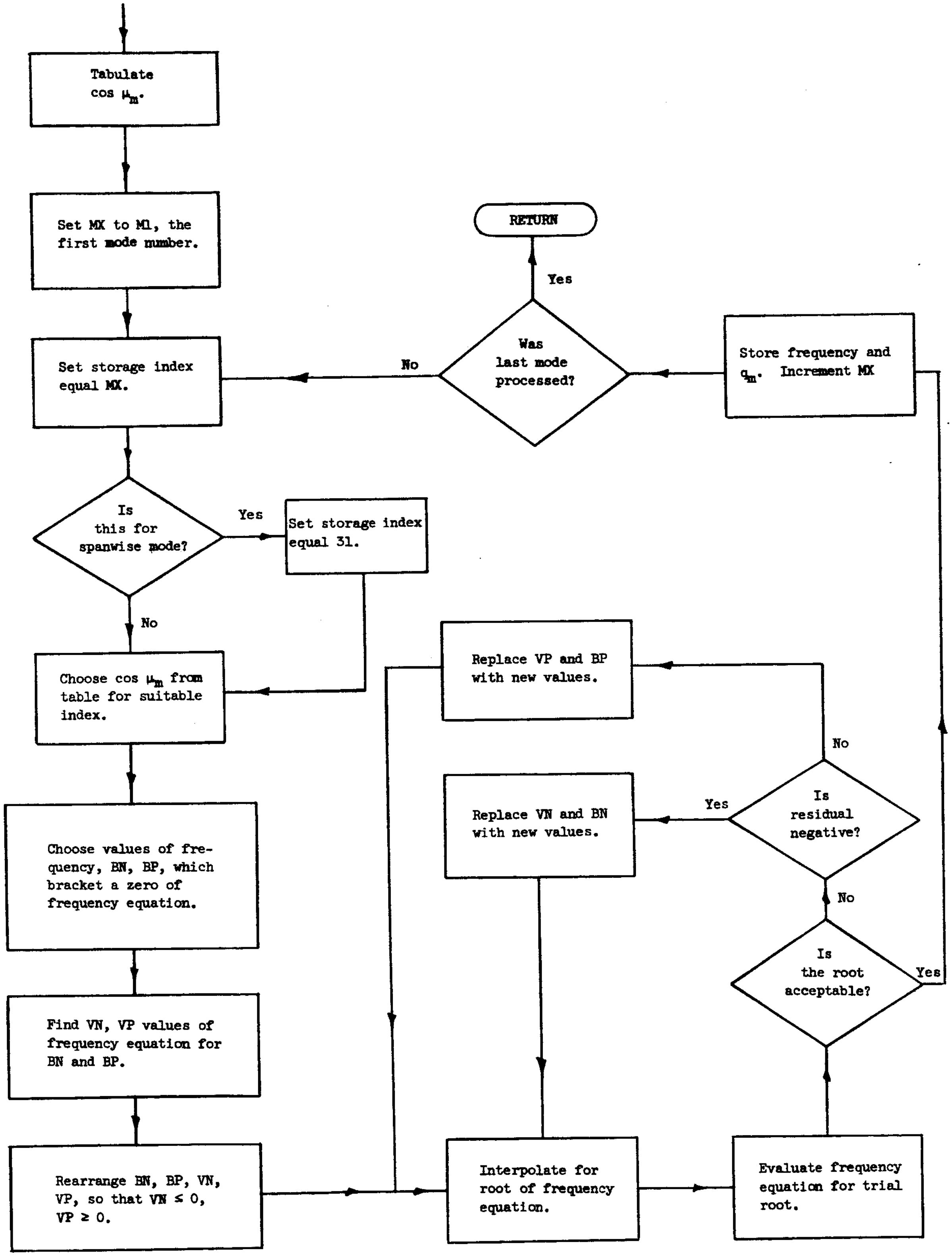
μ_2 = current μ value for which determinant is negative

μ_1 and μ_2 are used to interpolate for the next approximation μ_E . If the absolute value of the range $\mu_1 - \mu_2$ is less than or equal to μ_E times TESTR, μ_E is accepted as the flutter μ .

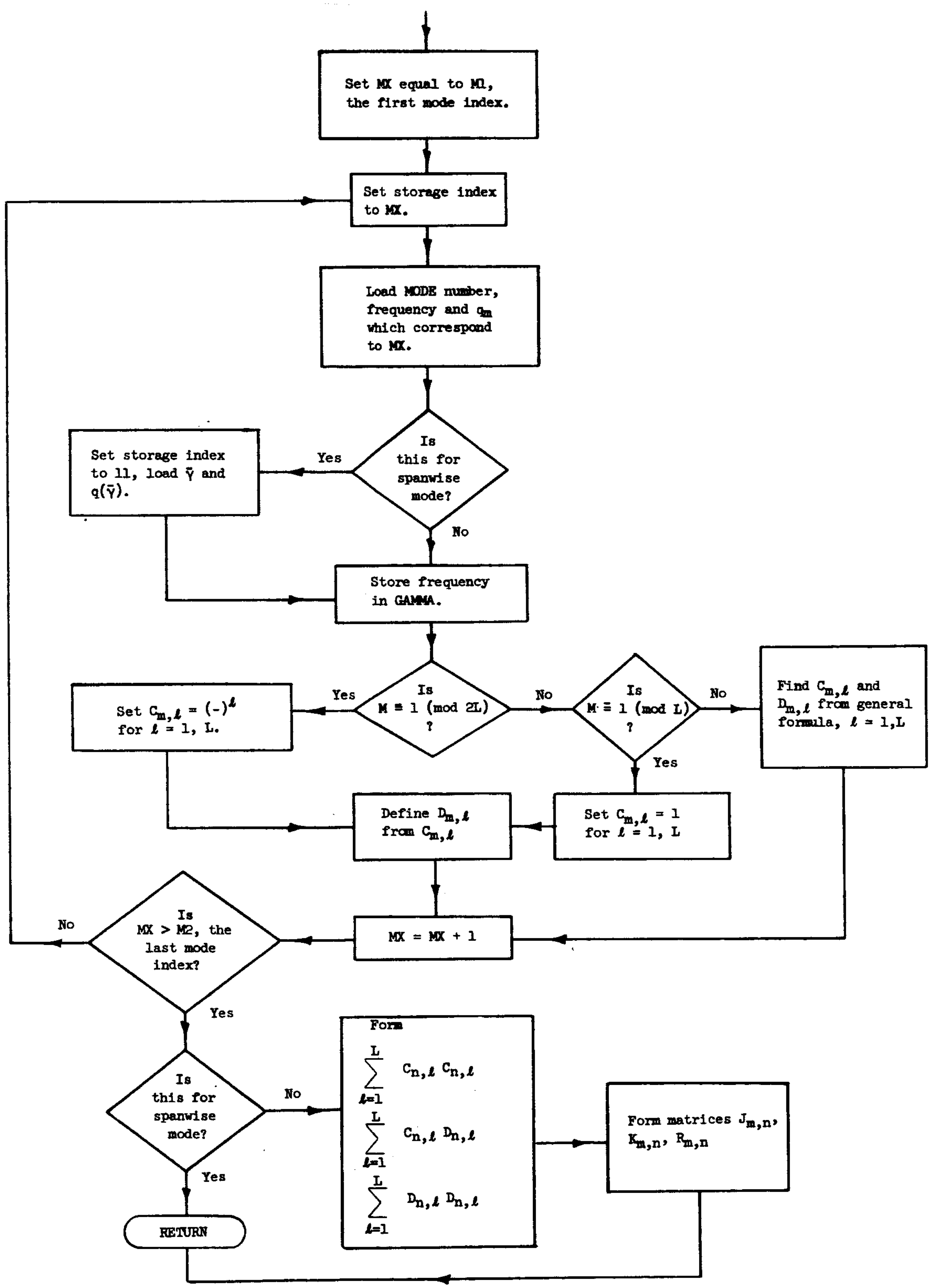
In application of the computer program a value of 10^{-7} has been used for TESTR. The effect of using other values has not been explored.

APPENDIX X

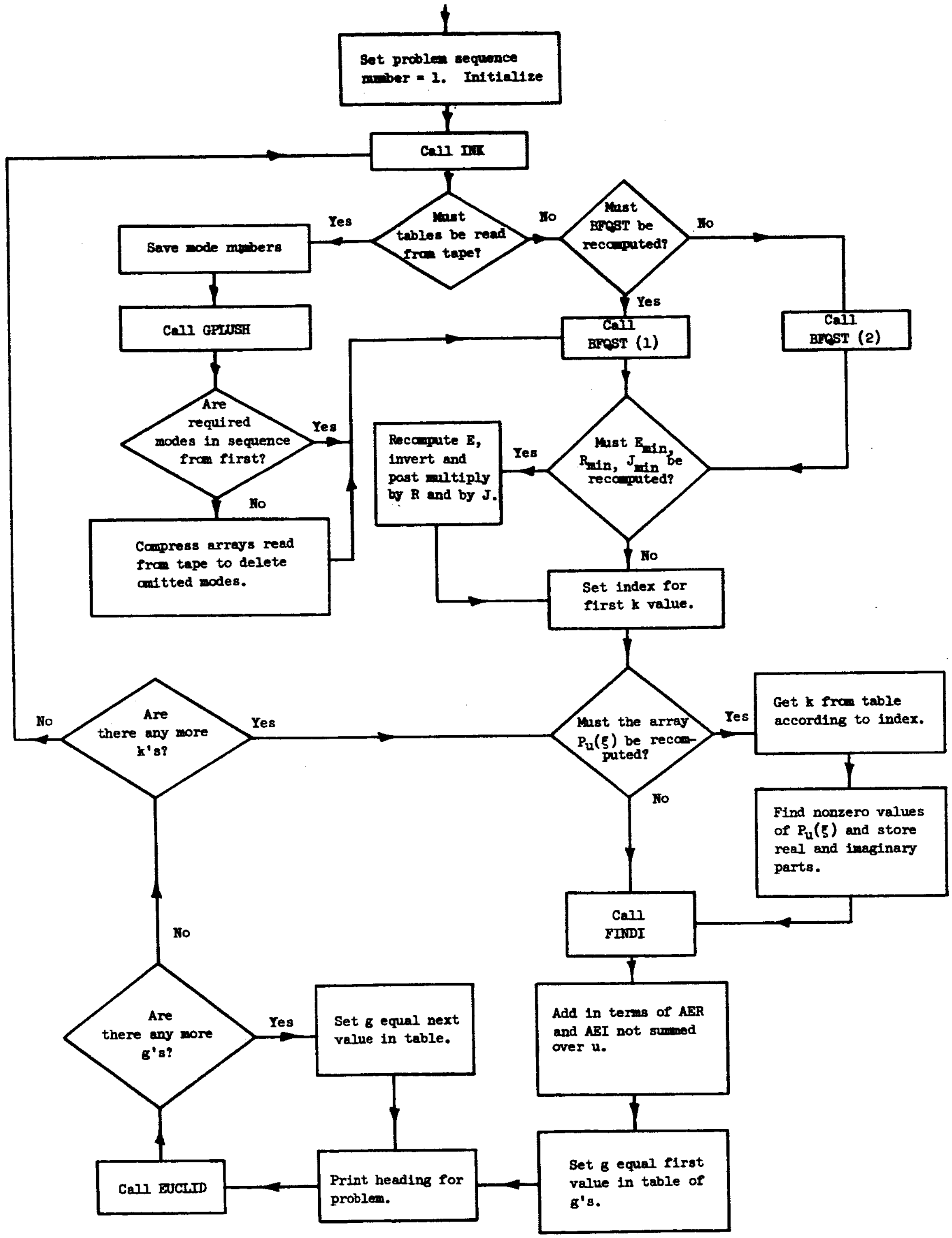
FLOW DIAGRAMS



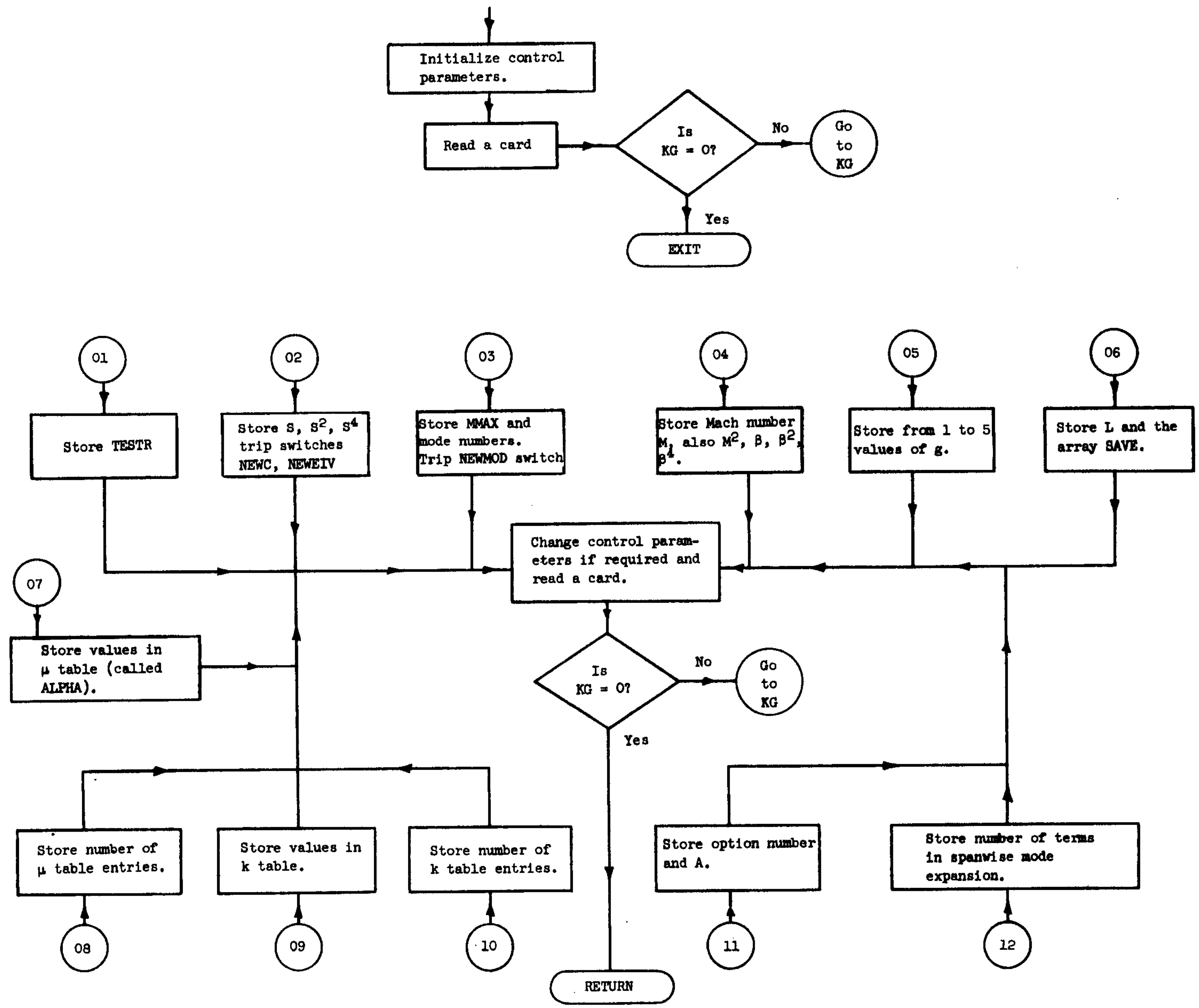
Subroutine FREQEQ



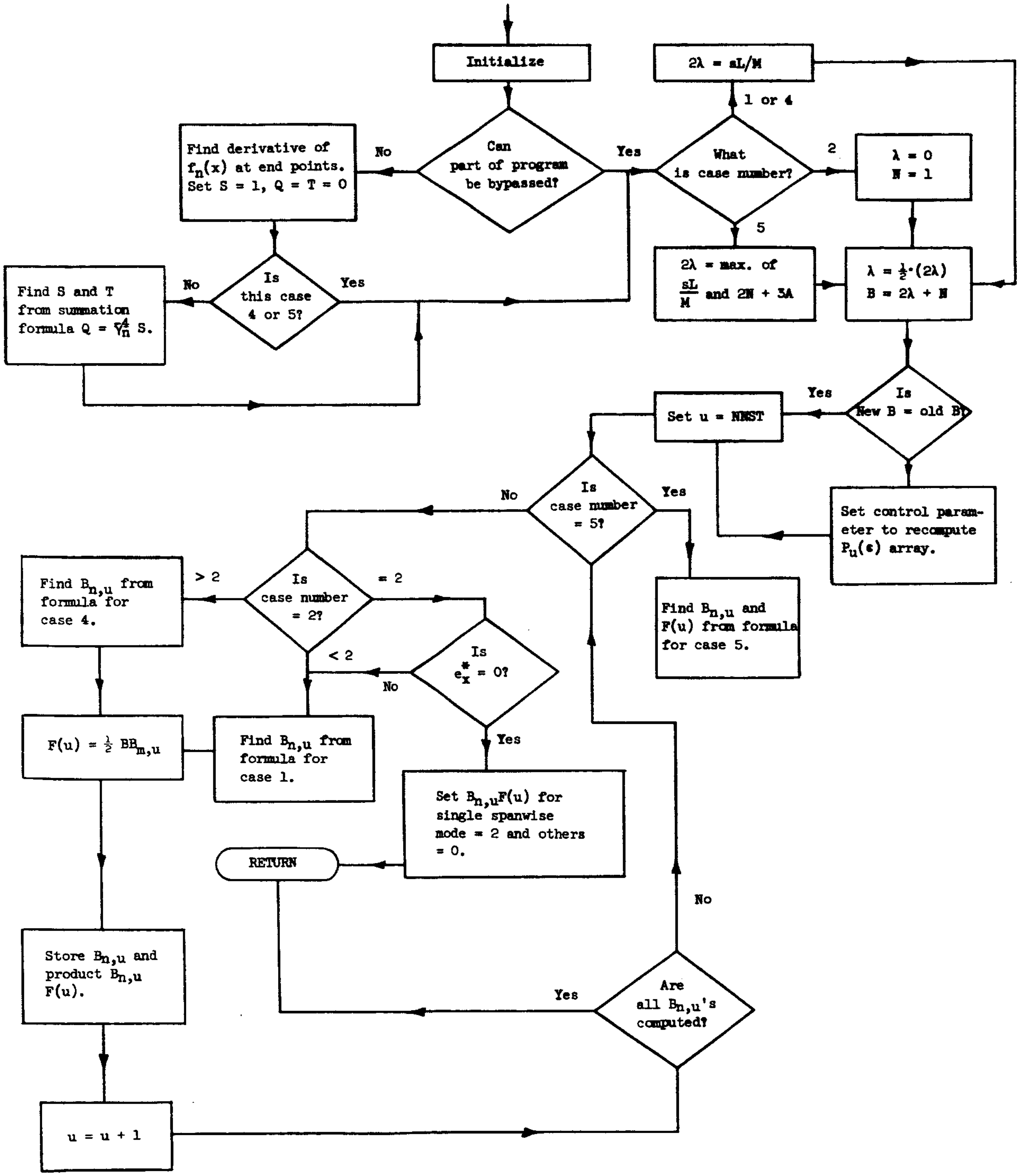
Subroutine CDFIND



Subroutine CONTRL

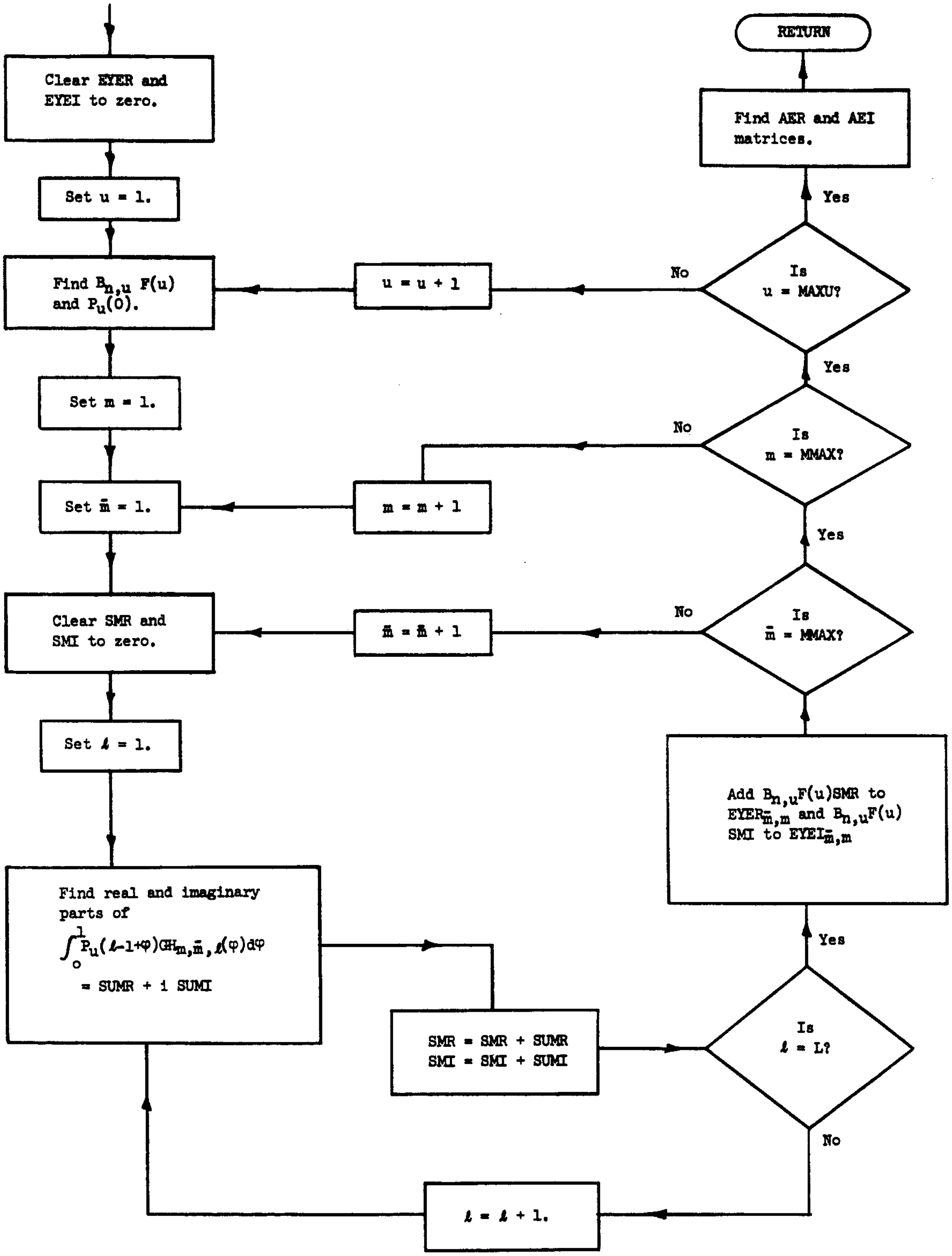


Subroutine INK

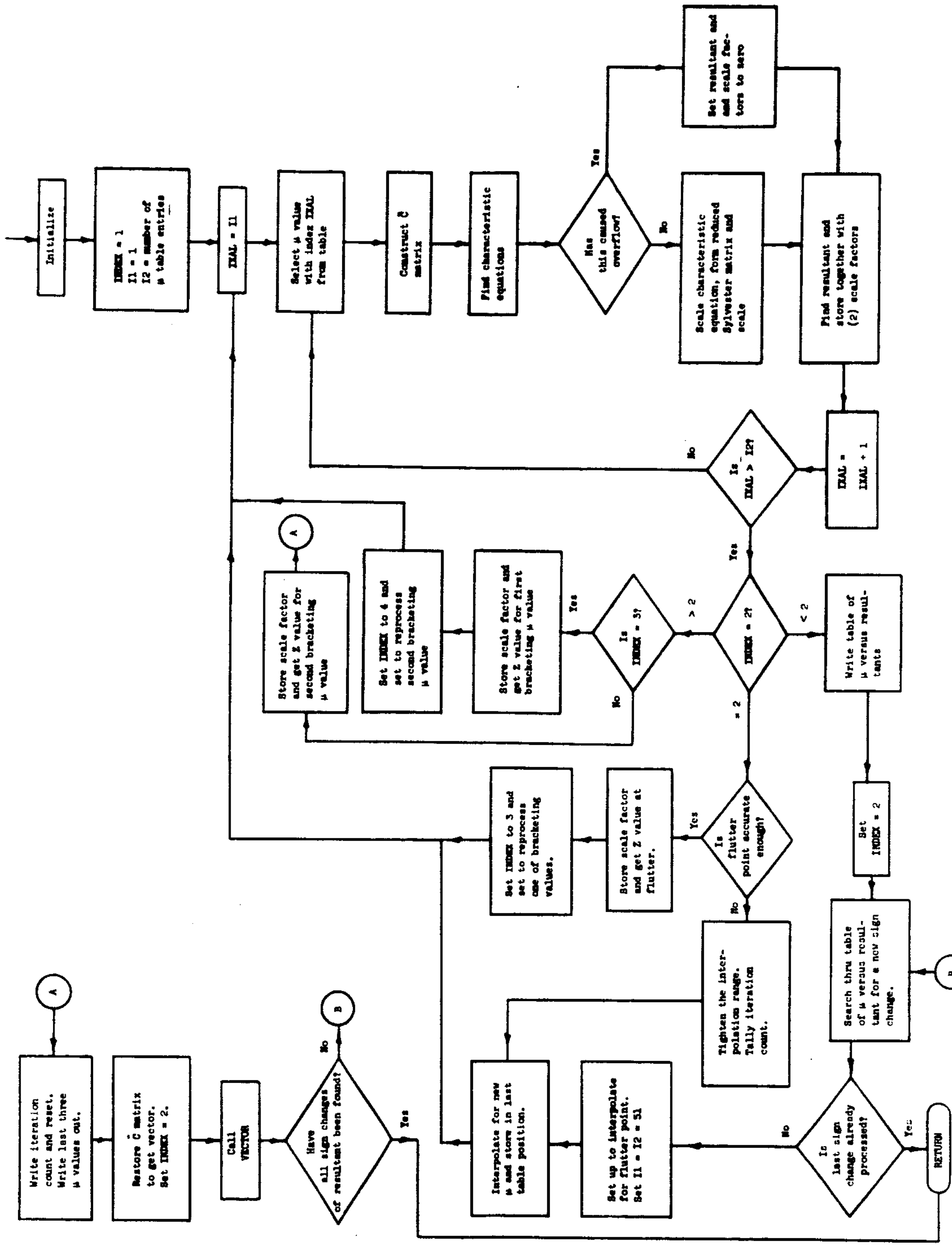


Note: ϵ_x and ϵ_y in the program are interchanged with reference to the notation in NASA CR-80 (Ref (1)).

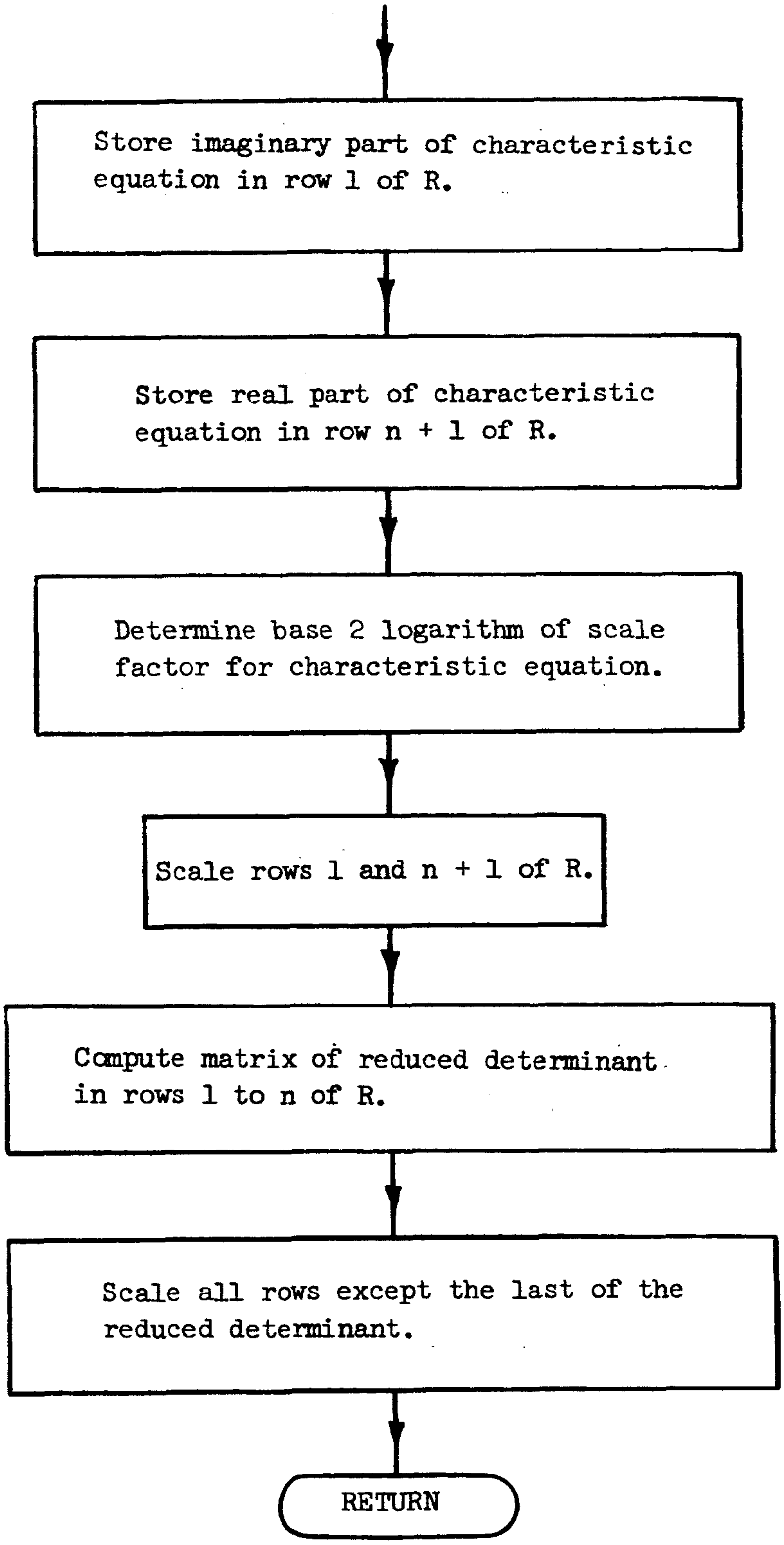
Subroutine BFQST



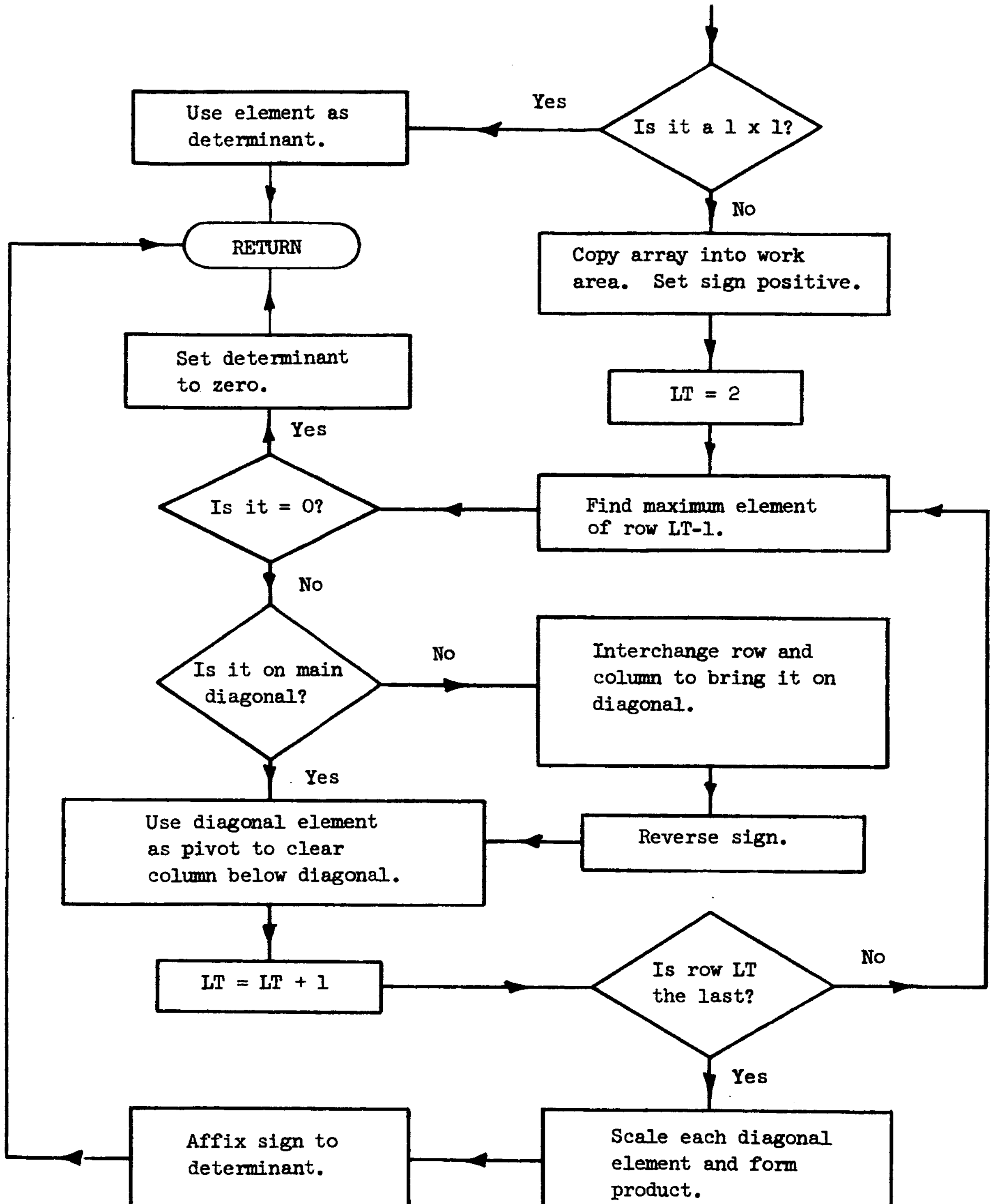
Subroutine FINDI



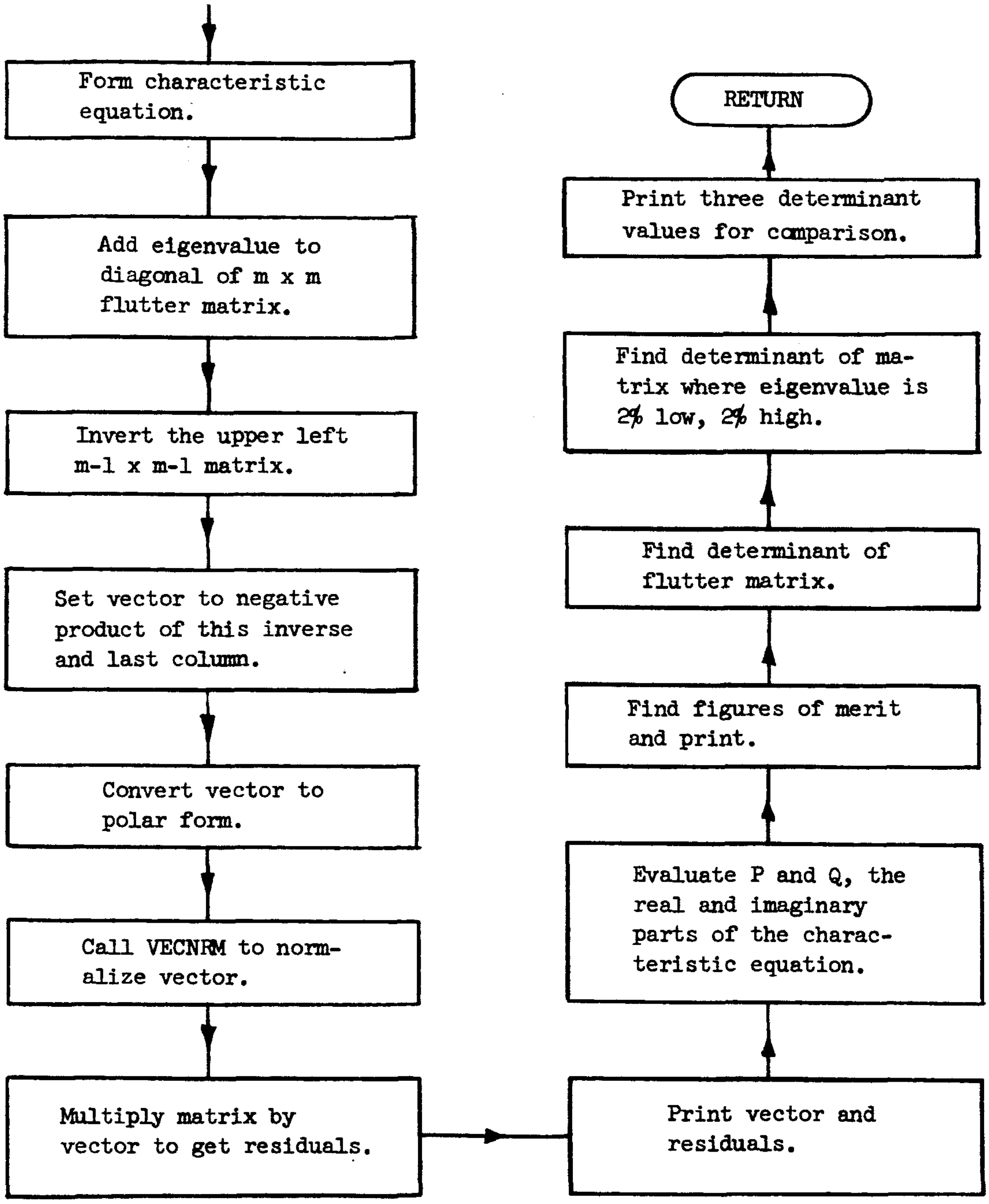
Subroutine EUCLID



Subroutine SCALE



Subroutine DETERM



Subroutine VECTOR

APPENDIX XI

PROGRAM LISTINGS

```
C***  MAIN PROGRAM FOR PROJECT NO. 2852-P

      CALL GHDUMP                                                        GHFLUT20
      CALL CONTRL                                                        GHFLUT30
      RETURN                                                             GHFLUT40
      END                                                                GHFLUT60


      SUBROUTINE GHDUMP                                                  GHDU0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,GHDU0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,O,XL,SMR,SMI,EYER,EYEI,CC,CD,GHDU0040
     2DD,C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  GHDU0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           GHDU0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             GHDU0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,ON           GHDU0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),    GHDU0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), GHDU0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),GHDU0110
     3PUR(20, 80),PUI(20, 80),CC(10,10),CD(10,10),DD(10,10),C(11,6),D(11GHDU0120
     4,6),QMM(31),MODE(10),GT(5)                                         GHDU0130
      DIMENSION SAVE(14)                                                 GHDU0140
      DIMENSION ON(20)                                                   GHDU0150
      NODP=16                                                            GHDU0170
      DPHI=.0625                                                         GHDU0180
      PI = 3.1415927                                                     GHDU0190
      MMAX=10                                                            GHDU0200
      READ(5,600) LMIN,LMAX,MODE,NSP,NGBAR,EPY,EPX                       GHDU0210
  600 FORMAT(14I2,2F10.0)                                                GHDU0220
C***  IF THE CARD IS BLANK, TAPE WAS GENERATED PREVIOUSLY                 GHDU0222
      IF(LMAX.LT.1) RETURN                                               GHDU0225
C***  FIND SPANWISE FREQUENCIES AND MODE SHAPES                          GHDU0227
      CALL FREQEQ(NSP,NGBAR,NGBAR,EPX,FREQ)                              GHDU0230
      CALL CDFIND(1,NSP,NGBAR,NGBAR)                                     GHDU0240
      DO 900 L=LMIN,LMAX                                                 GHDU0250
      LNDP=L*NODP                                                        GHDU0260
C***  FIND CHORDWISE FREQUENCIES AND MODE SHAPES                         GHDU0265
   21 CALL FREQEQ(L,1,MODE(10),EPY,FREQ)                                 GHDU0270
   23 CALL CDFIND(1,L,1,MMAX)                                            GHDU0280
C***  GENERATE G PLUS H MATRIX                                           GHDU0285
   31 DO 46 J=1,LNDP                                                     GHDU0290
      XT=J                                                               GHDU0300
      DARG=XT*DPHI                                                       GHDU0310
   32 IF(DARG-1.001) 35,33,33                                            GHDU0320
   33 DARG=DARG-1.                                                       GHDU0330
      GO TO 32                                                           GHDU0340
   35 JM=LNDP-NODP-J                                                     GHDU0350
      JX=(J-1)/NODP+1                                                    GHDU0360
      DO 46 M=1,MMAX                                                     GHDU0370
      DO 46 MB=1,MMAX                                                    GHDU0380
      CALL GMML(M,MB,JX,L,DARG)                                          GHDU0390
      GH(MB,M,J)=GMM                                                     GHDU0400
      IF(JM) 46,44,44                                                    GHDU0410
   44 CALL HMML(M,MB,JX,L,DARG)                                          GHDU0420
      GH(MB,M,J)=GH(MB,M,J)+HMM                                          GHDU0430
   46 CONTINUE                                                           GHDU0440
      WRITE( 9)L,MODE,NSP,NGBAR,EPY,EPX,NODP                             GHDU0450
      WRITE( 9)GAMMA,C                                                   GHDU0460
      WRITE( 9) D                                                        GHDU0470
      WRITE( 9) DJAY                                                     GHDU0480
      WRITE( 9) CAY                                                      GHDU0490
      WRITE( 9) ARE                                                      GHDU0500
  500 WRITE( 9) GH                                                       GHDU0510
  900 CONTINUE                                                           GHDU0520
C***     WRITE RECORD TO SIGNAL END OF DATA                              GHDU0525
   50 L=5                                                                GHDU0530
      WRITE( 9) L,MODE,NSP,NGBAR,EPY,EPX,NODP                            GHDU0540
      END FILE 9                                                         GHDU0545
      REWIND 9                                                           GHDU0546
      RETURN                                                             GHDU0550
      END                                                                GHDU0560
```

```
      SUBROUTINE FREQEQ(LUL,MM1,MM2,EP,FREQ)                            FREQ0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,FREQ0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,O,XL,SMR,SMI,EYER,EYEI,CC,CD,FREQ0040
     2DD,C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  FREQ0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           FREQ0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             FREQ0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,ON          FREQ0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   FREQ0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), FREQ0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),FREQ0110
     3PUR(20, 80),PUI(20, 80),CC(10,10),CD(10,10),DD(10,10),C(11,6),D(11FREQ0120
     4,6),QMM(31),MODE(10),GT(5)                                        FREQ0130
      DIMENSION SAVE(14)                                                FREQ0140
      DIMENSION ON(20)                                                  FREQ0150
      DIMENSION CST(7)                                                  FREQ0160
      L=LUL                                                             FREQ0170
      M1=MM1                                                            FREQ0180
      M2=MM2                                                            FREQ0190
  670 DO 671 J=2,6                                                      FREQ0200
  671 CST(J)=0.                                                         FREQ0210
      CST(1)=-1.                                                        FREQ0220
  702 GO TO (750,750,703,704,705,706),L                                 FREQ0230
  703 CST(2)=-.5                                                        FREQ0240
      GO TO 750                                                         FREQ0250
  704 CST(2)=-.70710678                                                 FREQ0260
      GO TO 750                                                         FREQ0270
  705 CST(2)=-.80901699                                                 FREQ0280
      CST(3)=-.30901699                                                 FREQ0290
      GO TO 750                                                         FREQ0300
  706 CST(2)=-.86602540                                                 FREQ0310
      CST(3)=-.5                                                        FREQ0320
  750 L2=L+2                                                            FREQ0330
      LH=(L+1)/2                                                        FREQ0340
      DO 753 J=1,LH                                                     FREQ0350
      K=L2-J                                                            FREQ0360
  753 CST(K)=-CST(J)                                                    FREQ0370
C              SOLVE FREQUENCY EQUATION                                 FRFQ0380
  755 DO 760 MX=M1,M2                                                   FREQ0390
      MK=MX                                                             FREQ0400
      M=MK                                                              FREQ0410
      IF(M1-M2) 7553,7551,7551                                          FREQ0420
 7551 MK=31                                                             FREQ0430
 7553 FML=(M-1)/L+1                                                     FREQ0440
      MQ=(M-1)/(2*L)                                                    FREQ0450
      MR=M-2*L*MQ                                                       FREQ0460
      IF(MR-L-1) 752,752,751                                            FREQ0470
  751 MR=L+L+2-MR                                                       FREQ0480
  752 COSMU=CST(MR)                                                     FREQ0490
      BN=3.1415*FML                                                     FREQ0500
      BP= BN+1.6                                                        FREQ0510
  756 BA=BN                                                             FREQ0520
      I=1                                                               FREQ0530
      GO TO 90                                                          FREQ0540
   10 VN=V                                                              FREQ0550
      BA=BP                                                             FREQ0560
      I=2                                                               FREQ0570
      GO TO 90                                                          FREQ0580
   11 VP=V                                                              FREQ0590
      IF(VN) 13,13,12                                                   FREQ0600
   12 V=VN                                                              FREQ0610
      BA=BN                                                             FREQ0620
      VN=VP                                                             FREQ0630
      BN=BP                                                             FREQ0640
      VP=V                                                              FREQ0650
      BP=BA                                                             FREQ0660
   13 GO TO 15                                                          FREQ0670
   15 RAT=VP/(VP-VN)                                                    FREQ0680
      DINT=BN-BP                                                        FREQ0690
      IF(RAT-.9) 152,152,151                                            FREQ0700
```

```
151   RAT=.9
      GO TO 154
152   IF(RAT-.1) 154,154,153
153   RAT=.1
154   BA=BP+RAT*DINT
      I=3
      GO TO 90
150   IF(BN-BA) 180,20,180
180   IF(BP-BA) 18,20,18
18    IF(V) 16,20,17
16    VN=V
      BN=BA
      GO TO 15
17    VP=V
      BP=BA
      GO TO 15
20    FREQ(MK)=BA
      QMM(MK)=EP*(SH-SI)/(2.*BA*SI*SH-EP*(SH*CO-SI*CH))
760   CONTINUE
99    RETURN
90    SI=SIN(BA)
      CO=COS(BA)
      EX=EXP(BA)
      EXM=1./EX
      SH=0.5*(EX-EXM)
      CH=SH+EXM
      V=EP*(1.-CO*CH)+BA*(COSMU*(SH-SI)+SI*CH-CO*SH)
      GO TO (10,11,150),I
      END


      SUBROUTINE COFIND(L1,LUL,MM1,MM2)
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,O,XL,SMR,SMI,EYER,EYEI,CC,CD,
     2DD,C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30),
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),
     3PUR(20, 80),PUI(20, 80),CC(10,10),CD(10,10),DD(10,10),C(11,6),D(11
     4,6),QMM(31),MODE(10),GT(5)
      DIMENSION SAVE(14)
      DIMENSION BN(20)
      L=LUL
      EL=L
      M1=MM1
      M2=MM2
      MM=MMAX
      DO 36 MX=M1,M2
      MK=MX
      M=MODE(MK)
      BA=FREQ(M)
      QM=QMM(M)
      IF(M1-M2) 20,17,17
17    M=MK
      BA=FREQ(31)
      QM=QMM(31)
      MK=11
20    LMOD=  MOD(M-1,2*L)
      GAMMA(MK)=BA
      IF(LMOD)24,22,24
C            M IS CONGRUENT TO 1 MOD 2L
22    C(MK,1)=1.
      IF(L-2) 231,221,221
221   DO 23 J=2,L
23    C(MK,J)=-C(MK,J-1)
231   SIGND=1.
      GO TO 28
24    IF(LMOD-L)30,26,30
```

```
C          M IS CONGRUENT TO 1 MOD L                                    COFI0420
 26   DO 27 J=1,L                                                       COFI0430
 27   C(MK,J)=1.                                                        COFI0440
      SIGND=-1.                                                         COFI0450
 28   DO 29 J=1,L                                                       COFI0460
 29   D(MK,J)=SIGND*C(MK,J)                                             COFI0470
      GO TO 36                                                          COFI0480
C     FORM C AND D                                                      COFI0490
 30   DO 35 J=L1,L                                                      COFI0500
      FJ=J                                                              COFI0510
      ZAHL=L+M-1                                                        COFI0520
      UM=PI*ZAHL/EL                                                     COFI0530
      UML=UM*FJ                                                         COFI0540
      SLUM=SIN(UML-UM)                                                  COFI0550
      SLU=SIN(UML)                                                      COFI0560
      C(MK,J)=SLU-QM*SLUM                                               COFI0570
 35   D(MK,J)=SLUM-QM*SLU                                               COFI0580
 36   CONTINUE                                                          COFI0590
      IF(MK-11) 40,99,99                                                COFI0600
C     FORM SUM OF PRODUCTS OF C AND D                                   COFI0610
 40   DO 5J=1,MM                                                        COFI0620
      DO 5K=1,MM                                                        COFI0630
      CC(J,K)=0.                                                        COFI0640
      CD(J,K)=0.                                                        COFI0650
      DD(J,K)=0.                                                        COFI0660
      DO 5I=1,L                                                         COFI0670
      CC(J,K)=CC(J,K)+C(J,I)*C(K,I)                                     COFI0680
      CD(J,K)=CD(J,K)+C(J,I)*D(K,I)                                     COFI0690
    5 DD(J,K)=DD(J,K)+D(J,I)*D(K,I)                                     COFI0700
C     FORM F TABLE FOR J,K,R                                            COFI0710
      DO 9 J=1,MM                                                       COFI0720
      DO 9 K=1,MM                                                       COFI0730
      F1     =FMM(1.,1.,J,K,0.)-FMM(0.,0.,J,K,0.)                       COFI0740
      F2     =FMM(0.,-1.,J,K,0.)-FMM(1.,0.,J,K,0.)                      COFI0750
      F3     =FMM(1.,1.,J,K,2.)-FMM(0.,0.,J,K,2.)                       COFI0760
      F4     =FMM(0.,-1.,J,K,2.)-FMM(1.,0.,J,K,2.)                      COFI0770
      F5     =FMM(1.,1.,J,K,1.)-FMM(0.,0.,J,K,1.)                       COFI0780
      F6     =FMM(0.,-1.,J,K,1.)-FMM(1.,0.,J,K,1.)                      COFI0790
C                 TRANSPOSE MATRICES TO AGREE WITH REPORT NOTATION      COFI0800
      DJAY(K,J)=F1*(CC(J,K)+DD(J,K))+F2*(CD(J,K)+CD(K,J))               COFI0810
       CAY(K,J)=F3*(CC(J,K)+DD(J,K))+F4*(CD(J,K)+CD(K,J))               COFI0820
  9   ARE(K,J)=F5*(CC(J,K)-DD(J,K))+F6*(CD(J,K)-CD(K,J))                COFI0830
 99   RETURN                                                            COFI0840
      END                                                               COFI0850
```

```
      SUBROUTINE GMML(M,MBAR,LL,L1,PHI)                                 GMML0020
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),SAVE(14)                   GMML0030
      DIMENSION C(11,6) ,D(11,6) ,DD(10,10),CD(10,10),CC(10,10),EYER(10,GMML0040
     110),EYEI(10,10)                                                   GMML0050
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,GMML0060
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,O,XL,SMR,SMI,EYER,EYEI,CC,CD,GMML0070
     2DD,C,D                                                            GMML0080
      L = L1                                                            GMML0090
      SUMCD1=0.0                                                        GMML0100
      SUMCD3=0.0                                                        GMML0110
      SUMCD5=0.0                                                        GMML0120
      SUMCD7=0.C                                                        GMML0130
      DO 10 LBAR=LL,L                                                   GMML0140
      J1=LBAR-LL+1                                                      GMML0150
      SUMCD1=SUMCD1+C(M,J1)*C(MBAR,LBAR)                                GMML0160
      SUMCD3=SUMCD3+C(M,J1)*D(MBAR,LBAR)                                GMML0170
      SUMCD5=SUMCD5+D(M,J1)*C(MBAR,LBAR)                                GMML0180
 10   SUMCD7=SUMCD7+D(M,J1)*D(MBAR,LBAR)                                GMML0190
      F1=FMM(1.0-PHI,1.0,M,MBAR,0.0)                                    GMML0200
      F2=FMM(0.0,PHI,M,MBAR,0.0)                                        GMML0210
      F3=FMM(0.0,1.0-PHI,M,MBAR,0.0)                                    GMML0220
      F4=FMM(PHI-1.0,0.0,M,MBAR,0.0)                                    GMML0230
      F5=FMM(-PHI,1.0,M,MBAR,0.0)                                       GMML0240
      F6=FMM(-1.0,PHI,M,MBAR,0.0)                                       GMML0250
      F7=FMM(1.0,1.0-PHI,M,MBAR,0.0)                                    GMML0260
```

```
      F8=FMM(PHI,0.0,M,MBAR,0.0)                                        GMML0270
      F1=F1-F2                                                          GMML0280
      F3=F3-F4                                                          GMML0290
      F5=F5-F6                                                          GMML0300
      F7=F7-F8                                                          GMML0310
      GMM=F1*SUMCD1-F3*SUMCD3                                           GMML0320
     1-F5*SUMCD5+F7*SUMCD7                                              GMML0330
  999 RETURN                                                            GMML0340
      END                                                               GMML0350


      SUBROUTINE HMML(M,MBAR,LL,L1,PHI)                                 HMML0020
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),SAVE(14)                   HMML0030
      DIMENSION C(11,6) ,D(11,6) ,DD(10,10),CD(10,10),CC(10,10),EYER(10,HMML0040
     110),EYEI(10,10)                                                   HMML0050
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,HMML0060
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,O,XL,SMR,SMI,EYER,EYEI,CC,CD,HMML0070
     2DD,C,D                                                            HMML0080
      L = L1                                                            HMML0090
      SUMCD1 = 0.0                                                      HMML0100
      SUMCD3 = 0.0                                                      HMML0110
      SUMCD5 = 0.0                                                      HMML0120
      SUMCD7 = 0.0                                                      HMML0130
      LL1 = LL+1                                                        HMML0140
      DO 20 LBAR = LL1,L                                                HMML0150
      J1 = LBAR -LL                                                     HMML0160
      SUMCD1=SUMCD1+C(M,J1)*C(MBAR,LBAR)                                HMML0170
      SUMCD3=SUMCD3+C(M,J1)*D(MBAR,LBAR)                                HMML0180
      SUMCD5=SUMCD5+D(M,J1)*C(MBAR,LBAR)                                HMML0190
   20 SUMCD7=SUMCD7+D(M,J1)*D(MBAR,LBAR)                                HMML0200
      F1 = FMM(1.0,PHI,M,MBAR,0.0)                                      HMML0210
      F2 = FMM(1.0-PHI,0.0,M,MBAR,0.0)                                  HMML0220
      F3 = FMM(PHI -1.0,1.0,M,MBAR,0.0)                                 HMML0230
      F4 = FMM(-1.0,1.0-PHI,M,MBAR,0.0)                                 HMML0240
      F5 = FMM(0.0,PHI,M,MBAR,0.0)                                      HMML0250
      F6 = FMM(-PHI,0.0,M,MBAR,0.0)                                     HMML0260
      F7 = FMM(PHI,1.0,M,MBAR,0.0)                                      HMML0270
      F8 = FMM(0.0,1.0-PHI,M,MBAR,0.0)                                  HMML0280
      F1 = F1-F2                                                        HMML0290
      F3 = F3-F4                                                        HMML0300
      F5 = F5-F6                                                        HMML0310
      F7 = F7-F8                                                        HMML0320
      HMM = F1*SUMCD1 - F3*SUMCD3 - F5*SUMCD5 + F7*SUMCD7               HMML0330
  999 RETURN                                                            HMML0340
      END                                                               HMML0350


      SUBROUTINE CONTRL                                                 CONT0020
C           ONE MODE PROBLEMS NOT ALLOWED                               CONT0030
      DIMENSION KODE(10)                                                CONT0040
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,CONT0050
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      CONT0060
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  CONT0070
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           CONT0080
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             CONT0090
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN          CONT0100
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   CONT0110
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), CONT0120
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),CONT0130
     3PUR(20, 80),PUI(20, 80),                           C(11,6),D(11CONT0140
     4,6),QMM(31),MODE(10),GT(5)                                        CONT0150
      DIMENSION SAVE(14)                                                CONT0160
      DIMENSION BN(20)                                                  CONT0170
      DIMENSION REST(51)                                                CONT0180
      PI=3.1415927                                                      CONT0190
      COEFS (1)= .999999997                                             CONT0200
      COEFS (2)=-.004394275                                             CONT0210
      COEFS(3)= .000434725                                              CONT0220
      COEFS (4)=-.000122226                                             CONT0230
      COEFS (5)= .000043506                                             CONT0240
```

```
      COEFS(6)=-.000009285                                              CONT0250
      COEFS(7)=-.031249995                                              CONT0260
      COEFS(8)= .001144106                                              CONT0270
      COEFS(9)=-.000218024                                              CONT0280
      COEFS(10)=.000085844                                              CONT0290
      COEFS(11)=-.000035614                                             CONT0300
      COEFS(12)= .000008099                                             CONT0310
      COEFS(13)=1.000000004                                             CONT0320
      COEFS(14)=.007323931                                              CONT0330
      COEFS(15)=-.000559487                                             CONT0340
      COEFS(16)=.000145575                                              CONT0350
      COEFS(17)=-.000050363                                             CONT0360
      COEFS(18)=.000010632                                              CONT0370
      COEFS(19)=.093749994                                              CONT0380
      COEFS(20)=-.001601836                                             CONT0390
      COEFS(21)=.000266891                                              CONT0400
      COEFS(22)=-.000099941                                             CONT0410
      COEFS(23)=.000040658                                              CONT0420
      COEFS(24)=-.000009173                                             CONT0430
      NSEQ=1                                                            CONT0440
      SAVE(1)=0.                                                        CONT0445
   20 CALL INK                                                          CONT0450
      IF(MMAX.LE.0) RETURN                                              CONT0460
   82 LNDP=L*16                                                         CONT0470
      IF(NEWGAM+NEWMOD+NEWDPH+NEWGMB) 50,50,25                          CONT0480
C     PULL TABLES OF G+H OFF OF TAPE 9.                                 CONT0490
   25 DO 26 J=1,MMAX                                                    CONT0500
   26 KODE(J)=MODE(J)                                                   CONT0510
      CALL GPLUSH                                                       CONT0520
      IF(MODE(MMAX)-KODE(MMAX)) 27,51,51                                CONT0530
C***  IF MODES REQUESTED DISAGREE WITH MODES ON TAPE, COMPRESS ARRAYS    CONT0534
C***  TO GIVE AGREEMENT                                                 CONT0535
   27 DO 35 IM=1,MMAX                                                   CONT0540
      IF(MODE(IM).EQ.KODE(IM)) GO TO 35                                 CONT0542
      DO 272 K=IM,9                                                     CONT0544
      IF(MODE(K+1).NE.KODE(IM)) GO TO 272                               CONT0546
      IK=K+1                                                            CONT0548
      GO TO 275                                                         CONT0550
  272 CONTINUE                                                          CONT0552
      WRITE(6,273) KODE(IM)                                             CONT0554
  273 FORMAT(16HOREQUESTED MODE I3,14H NOT ON TAPE 9)                   CONT0556
      GO TO 20                                                          CONT0558
  275 GAMMA(IM)=GAMMA(IK)                                               CONT0560
      DO 28 J=1,L                                                       CONT0580
      C(IM,J)=C(IK,J)                                                   CONT0590
   28 D(IM,J)=D(IK,J)                                                   CONT0600
      DO 32 LK=1,MMAX                                                   CONT0610
      KM=MODE(LK)                                                       CONT0620
      KK=KODE(LK)                                                       CONT0630
      DJAY(KM,IM)=DJAY(KK,IK)                                           CONT0640
      CAY(KM,IM)=CAY(KK,IK)                                             CONT0650
      ARE(KM,IM)=ARE(KK,IK)                                             CONT0660
      LGH=20*L                                                          CONT0670
      DO 32 JGH=1,LGH                                                   CONT0680
   32 GH(KM,IM,JGH)=GH(KK,IK,JGH)                                       CONT0690
   35 CONTINUE                                                          CONT0700
      GO TO 51                                                          CONT0710
   50 IF(NEWC) 60,60,51                                                 CONT0720
   51 CALL BFQST(1)                                                     CONT0730
      GO TO 70                                                          CONT0740
   60 IF(NNST-1) 70,70,61                                               CONT0750
   61 CALL BFQST(2)                                                     CONT0760
   70 IF(NEWEIV) 80,80,71                                               CONT0770
   71 DO 72 J=1,MMAX                                                    CONT0780
      SG=SS*GAMMA(J)**4                                                 CONT0790
C                  FORM E INVERSE, PREMULTIPLY R AND J                  CONT0800
      DO 72 K=1,MMAX                                                    CONT0810
   72 EINV(J,K)=(DJAY(J,K)*(SG+Q*SFOR)+(T+T)*SQS*CAY(J,K))/SS           CONT0820
   73 GO TO 74                                                          CONT0830
   74 CALL UMKEHR(EINV,MMAX)                                            CONT0840
      DO 75 J=1,MMAX                                                    CONT0850
      DO 75 K=1,MMAX                                                    CONT0860
```

```
      EIJ(J,K)=0.                                                       CONT0870
   75 EIR(J,K)=0.                                                       CONT0880
       DO 76 J=1,MMAX                                                   CONT0890
      DO 76 K=1,MMAX                                                    CONT0900
      DO 76 I=1,MMAX                                                    CONT0910
      EIJ(J,K)=EIJ(J,K)+EINV(J,I)*DJAY(I,K)                             CONT0920
   76 EIR(J,K)=EIR(J,K)+EINV(J,I)*ARE(I,K)                              CONT0930
   20 JK=1                                                              CONT0940
      IF(NNST+NK+NEWP -3) 90,81,81                                      CONT0950
C***  CHOOSE REDUCED FREQUENCY FROM TABLE                                CONT0955
   81  CK=CKAY(JK)                                                      CONT0960
      SQK=CK*CK                                                         CONT0970
C     CALCULATE AND STORE P                                             CONT0980
   83 DO 84 J=1,LNDP                                                    CONT0990
      ARG=J                                                             CONT1000
      ARG=ARG*DPHI                                                      CONT1010
      DO 84 K=NNST,NNN                                                  CONT1020
      U=2*K-  MOD(NGBAR*NSP,2)                                          CONT1030
      CALL FINDP(U,ARG)                                                 CONT1040
      PUR(K,J)=PREAL                                                    CONT1050
   84 PUI(K,J)=PIMAG                                                    CONT1060
   90 CALL FINDI(NNN)                                                   CONT1070
C                    ADD IN TERMS NOT SUMMED OVER U                     CONT1080
      COA=CK-CK/BSQ                                                     CONT1090
      DO 91 J=1,MMAX                                                    CONT1100
      DO 91 K=1,MMAX                                                    CONT1110
      AER(J,K)=AER(J,K)/SS+EIR(J,K)                                     CONT1120
   91 AEI(J,K)=AEI(J,K)/SS+COA*EIJ(J,K)                                 CONT1130
      DO 960 JG=1,NG                                                    CONT1140
C***  CHOOSE STRUCTURAL DAMPING FROM TABLE                               CONT1145
      G=GT(JG)                                                          CONT1150
      CALL HD1063(NSEQ,JK,KODE)                                         CONT1160
   95 FACTR=-SQK*BETA                                                   CONT1170
   96 F12=12./(BETA*(1.+G*G))                                           CONT1180
      CALL EUCLID(MMAX,G,EIJ,AER,AEI,ALPHA,REST,FACTR,MAXAL,TESTR,L,F12,CONT1190
     1DJAY)                                                             CONT1200
  960 CONTINUE                                                          CONT1210
   97 JK=JK+1                                                           CONT1220
      IF(JK-NK) 81,81,20                                                CONT1230
      END                                                               CONT1240
```

```
      SUBROUTINE INK                                                    INK 0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,INK 0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      INK 0040
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  INK 0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           INK 0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             INK 0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD             INK 0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   INK 0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), INK 0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),INK 0110
     3PUR(20, 80),PUI(20, 80),                             C(11,6),D(11INK 0120
     4,6),QMM(31),MODE(10),GT(5)                                        INK 0130
      DIMENSION SAVE(14)                                                INK 0140
      DIMENSION W(5),KW(3),KM(10)                                       INK 0150
      NEWDPH=0                                                          INK 0160
      NEWEIV=0                                                          INK 0170
      NNST=1                                                            INK 0180
      NEWC=0                                                            INK 0190
      NEWP=0                                                            INK 0200
      NEWGAM=0                                                          INK 0210
      NEWGMB=0                                                          INK 0220
      NEWMOD=0                                                          INK 0230
   40 READ          (5,41)KG,KW,W,KM                                    INK 0240
   41 FORMAT(4I2,2X5E10.5,10I2)                                         INK 0250
C***  IF FIRST CARD OF CASE IS BLANK, CALL EXIT                          INK 0255
      IF(KG)42,42,44                                                    INK 0260
   42 CALL EXIT                                                         INK 0270
C***  BRANCH TO STORE AS INDICATED BY CARD CODE                          INK 0275
   44 GO TO (1,2,3,4,5,6,7,8,9,10,11,12),KG                             INK 0280
```

```
 1      TESTR=W(1)                                                      INK 0290
        GO TO 52                                                        INK 0300
 2      S=W(1)                                                          INK 0310
        SQS=S*S                                                         INK 0320
        SFOR=SQS*SQS                                                    INK 0330
        NEWEIV=1                                                        INK 0340
        NEWC=1                                                          INK 0350
        GO TO 93                                                        INK 0360
 3      MMAX=KW(1)                                                      INK 0370
        NEWMOD=1                                                        INK 0380
        IF(KW(2))35,35,30                                               INK 0390
 30     DO 31  J=1,10                                                   INK 0400
 31     MODE(J)=KM(J)                                                   INK 0410
        GO TO 37                                                        INK 0420
 35     DO 36  J=1,MMAX                                                 INK 0430
 36     MODE(J)=J                                                       INK 0440
 37     GO TO 62                                                        INK 0450
 4      EMSQ=W(1)                                                       INK 0460
        BSQ=EMSQ-1.                                                     INK 0470
        EM=SQRT(EMSQ)                                                   INK 0480
        BETA=SQRT(BSQ)                                                  INK 0490
        BFOR=BSQ*BSQ                                                    INK 0500
        GO TO 93                                                        INK 0510
 5      NG=KW(1)                                                        INK 0520
        DO 505  J=1,5                                                   INK 0530
 505    GT(J)=W(J)                                                      INK 0540
        IF(NG) 506,506,52                                               INK 0550
 506    NG=1                                                            INK 0560
        GO TO 52                                                        INK 0570
 6      DO 61  J=1,5                                                    INK 0580
 61     SAVE(J)=W(J)                                                    INK 0590
        L=KW(1)                                                         INK 0595
        GO TO 52                                                        INK 0600
 62     NEWGAM=1                                                        INK 0640
 63     NEWEIV=1                                                        INK 0650
        GO TO 52                                                        INK 0660
 7      NK1=KW(1)                                                       INK 0670
        NK2=KW(2)                                                       INK 0680
        NKL=NK1-1                                                       INK 0690
        DO 71  J=NK1,NK2                                                INK 0700
        K=J-NKL                                                         INK 0710
 71     ALPHA(J)=W(K)                                                   INK 0720
        GO TO 52                                                        INK 0730
 8      MAXAL=KW(1)                                                     INK 0740
        GO TO 52                                                        INK 0750
 9      NK1=KW(1)                                                       INK 0760
        NK2=KW(2)                                                       INK 0770
        NKL=NK1-1                                                       INK 0780
        DO 91  J=NK1,NK2                                                INK 0790
        K=J-NKL                                                         INK 0800
 91     CKAY(J)=W(K)                                                    INK 0810
 93     NEWP=1                                                          INK 0840
        GO TO 52                                                        INK 0850
 10     NK=KW(1)                                                        INK 0860
        GO TO 52                                                        INK 0870
 11     NCASE=KW(1)                                                     INK 0910
C                          A/2 IS PUNCHED ON INPUT CARD                 INK 0920
        A=W(1)+W(1)                                                     INK 0930
        NEWC=1                                                          INK 0940
        GO TO 93                                                        INK 0950
 12     GO TO 124                                                       INK 0960
 124    NEWP=1                                                          INK 0970
 125    NNN=KW(1)                                                       INK 0980
        GO TO 52                                                        INK 0990
 52     READ          (5,41)KG,KW,W,KM                                  INK 1090
        IF(KG)99,99,44                                                  INK 1100
C***    A BLANK CARD SIGNALS END OF DATA FOR THIS CASE                  INK 1105
 99     RETURN                                                          INK 1120
        END                                                             INK 1140
```

```
      SUBROUTINE GPLUSH                                                 GPLU0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,GPLU0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      GPLU0040
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  GPLU0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           GPLU0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             GPLU0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN          GPLU0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   GPLU0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), GPLU0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),GPLU0110
     3PUR(20, 80),PUI(20, 80),                            C(11,6),D(11GPLU0120
     4,6),QMM(31),MODE(10),GT(5)                                        GPLU0130
      DIMENSION SAVE(14)                                                GPLU0140
      DIMENSION BN(20)                                                  GPLU0150
   10 READ      (9)LL,MODE,NSP,NGBAR,EPY,EPX,NODP                       GPLU0160
      IF(LL-L) 80,80,50                                                 GPLU0170
   80 READ      (9)GAMMA,C                                              GPLU0180
      READ (9) D                                                        GPLU0190
      READ (9) DJAY                                                     GPLU0200
      READ(9) CAY                                                       GPLU0210
      READ(9) ARE                                                       GPLU0220
      READ(9) GH                                                        GPLU0230
      IF(LL-L) 10,90,50                                                 GPLU0240
   90 IF(L-3) 999,999,998                                               GPLU0250
  998 REWIND 9                                                          GPLU0260
  999 RETURN                                                            GPLU0270
   50 REWIND 9                                                          GPLU0280
      GO TO 10                                                          GPLU0290
      END                                                               GPLU0300


      SUBROUTINE BFQST(NN)                                              BFQS0040
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,BFQS0050
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      BFQS0060
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,  BFQS0070
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           BFQS0080
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             BFQS0090
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN          BFQS0100
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   BFQS0110
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), BFQS0120
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),BFQS0130
     3PUR(20, 80),PUI(20, 80),                            C(11,6),D(11BFQS0140
     4,6),QMM(31),MODE(10),GT(5)                                        BFQS0150
      DIMENSION SAVE(14)                                                BFQS0160
      DIMENSION BN(20)                                                  BFQS0170
      N=NN                                                              BFQS0180
      JSP=11                                                            BFQS0190
      ENN=NSP                                                           BFQS0200
      EL=L                                                              BFQS0210
      G4=GAMMA(11)**4                                                   BFQS0220
      GO TO (540,620),N                                                 BFQS0230
  540 COES=.25/G4                                                       BFQS0240
      FIN3=FMKT(3,JSP,1.)                                               BFQS0250
      FON3=FMKT(3,JSP,0.)                                               BFQS0260
      FIN1=FMKT(1,JSP,1.)                                               BFQS0270
      FON1=FMKT(1,JSP,0.)                                               BFQS0280
      FIN2=FMKT(2,JSP,1.)                                               BFQS0290
      FON2=FMKT(2,JSP,0.)                                               BFQS0300
      T=0.                                                              BFQS0310
      Q=0.                                                              BFQS0320
      SS=1.                                                             BFQS0330
      IF(NCASE-3) 546,546,620                                           BFQS0340
  546 SS=0.                                                             BFQS0350
      DO 550 K=1,NSP                                                    BFQS0360
      CNKB=C(11,K)                                                      BFQS0370
      DNKB=D(11,K)                                                      BFQS0380
      RUB=(CNKB*FON1-DNKB*FIN1)*(CNKB*FON2+DNKB*FIN2)                   BFQS0390
      RAB=CNKB*FIN1-DNKB*FON1                                           BFQS0400
      ROB=CNKB*FIN2+DNKB*FON2                                           BFQS0410
      RIB=CNKB*FIN3-DNKB*FON3                                           BFQS0420
      SS=SS+COES*(RUB-2.*RIB*RAB+ROB*(ROB-RAB))                         BFQS0430
```

```
      RUB=(CNKB*FON2+DNKB*FIN2)*(CNKB*FON3-DNKB*FIN3)              BFQS0440
 550  T=T-0.25*RAB*RAB+COES*(RIB*(ROB-RIB)-RUB)                    BFQS0450
      Q=SS*G4                                                      BFQS0460
C     FIND LAMDA AND B                                             BFQS0470
 620  GO TO (81,82,83,81,83),NCASE                                 BFQS0480
 81   WHAT=S*EL/EM                                                 BFQS0490
      GO TO 90                                                     BFQS0500
 82   WHAT=0.                                                      BFQS0510
      ENN=1.                                                       BFQS0520
      GO TO 90                                                     BFQS0530
 83   WHAT=S*EL/EM                                                 BFQS0540
      WAS=3.*A+ENN+ENN                                             BFQS0550
      IF(WHAT-WAS)831,90,90                                        BFQS0560
 831  WHAT=WAS                                                     BFQS0570
 90   AMDA=0.5*WHAT                                                BFQS0580
      WAS=WHAT+ENN                                                 BFQS0590
      IF(BEE-WAS) 138,140,138                                      BFQS0600
  138 BEE=WAS                                                      BFQS0610
      NFWP=1                                                       BFQS0620
C     FIND BU AND FU                                               BFQS0630
 140  DO 147 JU=NNST,NNN                                           BFQS0640
      U=2*JU-MOD(NGBAR*NSP,2)                                      BFQS0650
      UPB=U*PI/BEE                                                 BFQS0660
      COEB=2./(G4-UPB**4)/BEE                                      BFQS0670
      GO TO (91,92,93,94,95),NCASE                                 BFQS0680
 91   BNU=0.                                                       BFQS0690
      DO 916 K=1,NSP                                               BFQS0700
      CNKB=C(11,K)                                                 BFQS0710
      DNKB=D(11,K)                                                 BFQS0720
      CPL=K                                                        BFQS0730
      CPL=(CPL+AMDA)*UPB                                           BFQS0740
      CPLM=CPL-UPB                                                 BFQS0750
      SINUK=SIN(CPL)                                               BFQS0760
      COSUK=COS(CPL)                                               BFQS0770
      SINUM=SIN(CPLM)                                              BFQS0780
      COSUM=COS(CPLM)                                              BFQS0790
      RUB=CNKB*FIN3-DNKB*FON3-UPB**2*(CNKB*FIN1-DNKB*FON1)         BFQS0800
      RAB=CNKB*FON3-DNKB*FIN3-UPB**2*(CNKB*FON1-DNKB*FIN1)         BFQS0810
      ROB=(CNKB*FON2+DNKB*FIN2)*COSUM-(CNKB*FIN2+DNKB*FON2)*COSUK  BFQS0820
 916  BNU=BNU+COEB*(RUB*SINUK+ROB*UPB-RAB*SINUM)                   BFQS0830
 918  FU=BNU*BEE*0.5                                               BFQS0840
      GO TO 98                                                     BFQS0850
 92   IF(EPX)921,921,91                                            BFQS0860
 921  DO 922 K=NNST,NNN                                            BFQS0870
      BN(K)=0.                                                     BFQS0875
 922  BUF(K)=0.                                                    BFQS0880
      BN(NGBAR)=2.                                                 BFQS0885
      BUF(NGBAR)=2.                                                BFQS0890
      GO TO 99                                                     BFQS0900
 93   PAUSE                                                        BFQS0910
      GO TO 91                                                     BFQS0920
 94   GO TO 941                                                    BFQS0930
 941  BNU =4./(PI*U)*COS(0.5*UPB*(BEE-1.))                         BFQS0940
 945  GO TO 918                                                    BFQS0950
 95   ABAR=AMDA-ENN-A                                              BFQS0960
      COEB=2./(U*PI)                                               BFQS0970
      RUB=UPB*(1.+A)                                               BFQS0980
      ROB=RUB+UPB*ABAR                                             BFQS0990
      RAB=ROB+UPB                                                  BFQS1000
      BNU=COEB*(1.+2.*COS(RUB))*(COS(ROB)-COS(RAB))                BFQS1010
      FU=(COS(AMDA*UPB)-COS(UPB*(AMDA+1.)))/UPB                    BFQS1020
 98   BUF(JU)=BNU*FU                                               BFQS1030
 147  BN(JU)=BNU                                                   BFQS1040
 99   RETURN                                                       BFQS1050
      END                                                          BFQS1060


      SUBROUTINE UMKEHR(R,IF)                                      UMKE0020
      DIMENSION R(10,10)                                           UMKE0030
      N=IF                                                         UMKE0040
      IF(N-1)31,31,38                                              UMKE0050
```

```
31    R(1,1)=1./R(1,1)                                                  UMKE0060
      GO TO 99                                                          UMKE0070
38    DO 41 K=1,N                                                       UMKE0080
      D=R(K,K)                                                          UMKE0090
      R(K,K)=1.0                                                        UMKE0100
50    DO 42 J=1,N                                                       UMKE0110
42    R(K,J)=R(K,J)/D                                                   UMKE0120
56    IF(K-N)43,44,44                                                   UMKE0130
43    KPLUS=K+1                                                         UMKE0140
51    DO 41 I=KPLUS,N                                                   UMKE0150
      D=R(I,K)                                                          UMKE0160
      R(I,K)=0.0                                                        UMKE0170
52    DO 41 J=1,N                                                       UMKE0180
41    R(I,J)=R(I,J)-D*R(K,J)                                            UMKE0190
44    MINUS=N-1                                                         UMKE0200
53    DO 45 K=1,MINUS                                                   UMKE0210
      KPLUS=K+1                                                         UMKE0220
54    DO 45 I=KPLUS,N                                                   UMKE0230
      D=R(K,I)                                                          UMKE0240
      R(K,I)=0.0                                                        UMKE0250
55    DO 45 J=1,N                                                       UMKE0260
45    R(K,J)=R(K,J)-D*R(I,J)                                            UMKE0270
99    RETURN                                                            UMKE0280
      END                                                               UMKE0290


      SUBROUTINE FINDP (U,ZETA)                                         PFIN0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,PFIN0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      PFIN0040
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,   PFIN0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           PFIN0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE             PFIN0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN          PFIN0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   PFIN0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), PFIN0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),PFIN0110
     3PUR(20, 80),PUI(20, 80),                              C(11,6),D(11PFIN0120
     4,6),QMM(31),MODE(10),GT(5)                                        PFIN0130
      DIMENSION SAVE(14)                                                PFIN0140
      DIMENSION BN(20)                                                  PFIN0150
      BIGK=CK*EM/BSQ                                                    PFIN0160
      GAMUSQ=BIGK*BIGK+(U*PI*S/(BETA*BEE))**2                           PFIN0170
      GAMU=SQRT(GAMUSQ)                                                 PFIN0180
      ARG=GAMU*ZETA                                                     PFIN0190
      CO = 0.5*GAMUSQ + SQK/BFOR                                        PFIN0200
      C1=2.0*CK*GAMU/BSQ                                                PFIN0210
      C2=GAMUSQ*0.5                                                     PFIN0220
      BO=BFV(0,ARG)                                                     PFIN0230
      B1=BFV(1,ARG)                                                     PFIN0240
      IF(B1)10,11,10                                                    PFIN0250
11    B2=0.0                                                            PFIN0260
      GO TO 12                                                          PFIN0270
10    B2=2.0*B1/ARG-BO                                                  PFIN0280
12    EV=C2*B2-CO*BO                                                    PFIN0290
      ODD=C1*B1                                                         PFIN0300
      AA=BIGK*EM*ZETA                                                   PFIN0310
      SINA=SIN(AA)                                                      PFIN0320
      COSA=COS(AA)                                                      PFIN0330
      PREAL=COSA*EV+SINA*ODD                                            PFIN0340
      PIMAG=COSA*ODD-SINA*EV                                            PFIN0350
999   RETURN                                                            PFIN0360
      END                                                               PFIN0370


      SUBROUTINE FINDI(MAXU)                                            IFIN0030
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,IFIN0040
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      IFIN0050
     2   C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,   IFIN0060
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,           IFIN0070
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE(10)         IFIN0080
```

```
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN          IFIN0090
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),   IFIN0100
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30), IFIN0110
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),IFIN0120
     3PUR(20, 80),PUI(20, 80),                             C(11,6),D(11IFIN0130
     4,6),QMM(31)                                                       IFIN0140
      DIMENSION SAVE(14),GT(5)                                          IFIN0150
      DIMENSION X(5),Y(5)                                               IFIN0160
      RUB=-SQK*(0.5*EMSQ+1.)/BFOR                                       IFIN0170
      ROB=0.5*(PI*S/BEE)**2/BSQ                                         IFIN0180
      J=NODP                                                            IFIN0190
      DO 47 K=1,MMAX                                                    IFIN0200
      DO 47 KB=1,MMAX                                                   IFIN0210
      EYER(K,KB)=0.                                                     IFIN0220
   47 EYEI(K,KB)=0.                                                     IFIN0230
C***  LOOP TO SUM OVER INDEX U                                          IFIN0235
      DO 201 JU=1,MAXU                                                  IFIN0240
      BUFJU=BUF(JU)                                                     IFIN0250
      U=2*JU-  MOD(NGBAR,2)                                             IFIN0260
C***  LOOPS TO FIND CONTRIBUTION OF TERM FOR EACH MATRIX ELEMENT        IFIN0265
      PZERO=RUB-U*U*ROB                                                 IFIN0270
      DO 200 M=1,MMAX                                                   IFIN0280
      DO 199 MBAR=1,MMAX                                                IFIN0290
      SMR=0.                                                            IFIN0300
      SMI=0.                                                            IFIN0310
      PASTPI=0.                                                         IFIN0320
      PASTPR=PZERO                                                      IFIN0330
      PASTGH=DJAY(M,MBAR)                                               IFIN0340
C***  LOOP TO SUM OVER ALL CHORDWISE PANELS                             IFIN0345
      DO 56 LL=1,L                                                      IFIN0350
      ILL=(LL-1)*J                                                      IFIN0360
      SUMR=0.                                                           IFIN0370
      SUMI=0.                                                           IFIN0380
C***  NUMERICAL INTEGRATION LOOP                                        IFIN0385
      DO 20 I=1,J,4                                                     IFIN0390
      IND=ILL+I-1                                                       IFIN0400
      DO 17 N=1,4                                                       IFIN0410
      IS=IND+N                                                          IFIN0420
      GMM=GH(M,MBAR,IS)                                                 IFIN0430
      X(N+1)=GMM*PUR(JU,IS)                                             IFIN0440
   17 Y(N+1)=GMM*PUI(JU,IS)                                             IFIN0450
      ANSR=(7.*(PASTPR*PASTGH+X(5))+32.*(X(2)+X(4))+12.*X(3))/22.5      IFIN0460
      ANSI=(7.*(PASTPI*PASTGH+Y(5))+32.*(Y(2)+Y(4))+12.*Y(3))/22.5      IFIN0470
      SUMR=SUMR+DPHI*ANSR                                               IFIN0480
      SUMI=SUMI+DPHI*ANSI                                               IFIN0490
      PASTGH=GMM                                                        IFIN0500
      PASTPI=PUI(JU,IS)                                                 IFIN0510
   20 PASTPR=PUR(JU,IS)                                                 IFIN0520
      SMR=SMR+SUMR                                                      IFIN0530
   56 SMI=SMI+SUMI                                                      IFIN0540
      EYER(M,MBAR)=EYER(M,MBAR)+SMR*BUFJU                               IFIN0550
  199 EYEI(M,MBAR)=EYEI(M,MBAR)+SMI*BUFJU                               IFIN0560
  200 CONTINUE                                                          IFIN0570
  201 CONTINUE                                                          IFIN0580
C              FORM E INVERSE I= AER AND AEI                            IFIN0590
  106 DO 203 J=1,MMAX                                                   IFIN0600
      DO 203 K=1,MMAX                                                   IFIN0610
      AER(J,K)=0.                                                       IFIN0620
  203 AEI(J,K)=0.                                                       IFIN0630
      DO 205 J=1,MMAX                                                   IFIN0640
      DO 205 K=1,MMAX                                                   IFIN0650
      DO 205 I=1,MMAX                                                   IFIN0660
      AER(J,K)=AER(J,K)+EINV(J,I)*EYER(I,K)                             IFIN0670
  205 AEI(J,K)=AEI(J,K)+EINV(J,I)*EYEI(I,K)                             IFIN0680
   99 RETURN                                                            IFIN0690
      END                                                               IFIN0700


      SUBROUTINE HD1063(NSEQ,JK,KODE)                                   HEAD0020
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,HEAD0030
     1EM,CK,S,PI,PREAL,PIMAG,HMM,DPHI,MMAX,L,XL,SMR,SMI,EYER,EYEI,      HEAD0040
```

```
     2    C,D,PUR,PUI,GH,BUF,DJAY,CAY,ARE,NNN,EMSQ,NODP,XNODP,BEE,NEWP,      HEAD0050
     3NEWGH,NEWGAM,NEWGMB,NEWDPH,ALPHA,CKAY,NEWEIV,NEWC,NNST,                HEAD0060
     4DPNO,EINV,SQS,SFOR,EIJ,EIR,AER,AEI,NK,A,LNDP,QMM,MODE                  HEAD0070
      COMMON G,TESTR,MAXAL,MMM,NCASE,NGBAR,NSP,NG,GT,NEWMOD,BN               HEAD0080
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),EYER(10,10),EYEI(10,10),        HEAD0090
     1EIJ(10,10),EIR(10,10),AER(10,10),AEI(10,10),EINV(10,10),CKAY(30),      HEAD0100
     2ALPHA(51),DJAY(10,10),CAY(10,10),ARE(10,10),BUF(20),GH(10,10, 80),     HEAD0110
     3PUR(20, 80),PUI(20, 80),                               C(11,6),D(11HEAD0120
     4,6),QMM(31),MODE(10),GT(5)                                             HEAD0130
      DIMENSION SAVE(14)                                                     HEAD0140
      DIMENSION BN(20),KODE(10)                                              HEAD0150
      A2=A*0.5                                                               HEAD0160
 20   CALL CLOCK(TIME1,TIME2,DATE1,DATE2)                                    HEAD0170
      WRITE           (6,21)DATE1,DATE2,TIME1,TIME2,NSEQ                     HEAD0180
 21   FORMAT(32H1M.R.I. FLUTTER PROGRAM    DATE=,A6,A2,5X5HTIME=A6,A1,50HEAD0190
     1X,I4)                                                                  HEAD0200
      NSEQ=NSEQ+1                                                            HEAD0210
      WRITE           (6,22)EM,S,EPY,EPX,TESTR,L,MMAX,NSP,NNN,NGBAR          HEAD0220
 22   FORMAT(5X1HM9X1HS6X7HE(SPAN)2X8HE(CHORD)4X5HTESTR6X1HL6X4HMMAX9X1HHEAD0230
     1N6X4HUMAX3X13HSPANWISE MODE/4F10.4,E10.2,I6,4I10)                      HEAD0240
      WRITE           (6,23)NCASE,A2,NODP,CK,G                               HEAD0250
 23   FORMAT(5H0CASE5X3HA/25X6H1/DPHI4X1HK9X1HG/I4,F10.4,I8,2F10.4)          HEAD0260
      IF(NEWGAM+NEWGMB)92,92,24                                              HEAD0270
 24   WRITE           (6,25)GAMMA(11)                                        HEAD0280
 25   FORMAT(11H0GAMMA-BAR=F11.7)                                            HEAD0290
      WRITE           (6,26)(KODE(J),J=1,MMAX)                               HEAD0300
 26   FORMAT(16H0CHORD-WISE MODEI3,9I11)                                     HEAD0310
      WRITE           (6,27)(GAMMA(J),J=1,MMAX)                              HEAD0320
 27   FORMAT(8X10F11.7)                                                      HEAD0330
      WRITE           (6,948)(BN(J),J=1,NNN)                                 HEAD0340
 948  FORMAT(6H0BN(U)/(1X10F10.6))                                           HEAD0350
 92   IF(SAVE(1)) 99,93,99                                                   HEAD0360
 93   IF(JK-2) 94,95,95                                                      HEAD0370
 94   WRITE           (6,941)                                                HEAD0380
      DO 942 J=1,MMAX                                                        HEAD0390
 942  WRITE           (6,943)(EIJ(J,K),K=1,MMAX)                             HEAD0400
 95   WRITE           (6,944)                                                HEAD0410
      DO 945 J=1,MMAX                                                        HEAD0420
 945  WRITE           (6,943)(AER(J,K),K=1,MMAX)                             HEAD0430
      WRITE           (6,946)                                                HEAD0440
      DO 947 J=1,MMAX                                                        HEAD0450
 947  WRITE           (6,943)(AEI(J,K),K=1,MMAX)                             HEAD0460
 941  FORMAT(47H0INVERSE OF ELASTIC MATRIX TIMES INERTIA MATRIX)             HEAD0470
 943  FORMAT(1P8E15.7)                                                       HEAD0480
 944  FORMAT(55H0INVERSE OF ELASTIC MATRIX TIMES REAL PART AERCD MATRIX)HEAD0490
 946  FORMAT(55H0INVERSE OF ELASTIC MATRIX TIMES IMAG PART AERCD MATRIX)HEAD0500
 99   RETURN                                                                 HEAD0510
      END                                                                    HEAD0520


      SUBROUTINE CLOCK(TIME1,TIME2,DATE1,DATE2)                              CLOC0020
C     DUMMY ROUTINE TO BE REPLACED BY CLOCK ROUTINE APPROPRIATE TO           CLOC0030
C     PARTICULAR INSTALLATION                                                CLOC0040
      TIME1=0.                                                               CLOC0050
      TIME2=0.                                                               CLOC0060
      DATE1=0.                                                               CLOC0070
      DATE2=0.                                                               CLOC0080
      RETURN                                                                 CLOC0090
      END                                                                    CLOC0100


      SUBROUTINE EUCLID(M,G,E,AR,AI,ALS,REST,FACTR,MAXAL,TESTR,L,F12,DJ)EUC 0010
C     THIS VERSION OF EUCLID CALLS FLUT SUBROUTINE WHICH                     EUC 0020
C     REDUCES CHARACTERISTIC POLYNOMIALS TO LINEAR AND QUADRATIC FACTORSEUC 0030
C     FOR CALCULATION OF ALPHA(EUCLIDIAN ALGORITHM MODIFIED)                 EUC 0040
      DOUBLE PRECISION R(11,10),CR(10,20),PRE(22),ALP(51),AP,AN,AL,DUB,  EUC 0050
     1DET,RANGE                                                              EUC 0060
C     FLUTTER DETERMINATION USING SYLVESTERS RESULTANT                       EUC 0070
      DIMENSION REST(51)          ,LEST(51),JEST(51)                         EUC 0080
      DIMENSION ALS(50)                                                      EUC 0090
```

```
      DIMENSION E(10,10),AR(10,10),AI(10,10),DJ(10,10)                  EUC 0110
      J7=51                                                             EUC 0130
      JPRE=11                                                           EUC 0140
      DO 802 J=1,MAXAL                                                  EUC 0150
  802 ALP(J)=ALS(J)                                                     EUC 0160
      N=M                                                               EUC 0180
      MP=N+1                                                            EUC 0190
  803 INDEX=1                                                           EUC 0200
      I1=1                                                              EUC 0210
      I2=MAXAL                                                          EUC 0220
  804 DO 30 IXAL=I1,I2                                                  EUC 0230
  805 AL=ALP(IXAL)                                                      EUC 0240
C***  SET UP MATRIX CBAR                                                EUC 0245
      CALL CBAR(AL,G,AR,AI,E,N,CR,FACTR)                                EUC 0250
  807 CALL OVERFL(JO)                                                   EUC 0260
C***  FIND CHARACTERISTIC EQUATION                                      EUC 0265
  809 CALL EQCHAR(CR,PRE,N)                                             EUC 0280
C                SYLVESTER RESULTANT                                    EUC 0290
      CALL OVERFL(JO)                                                   EUC 0300
      GO TO (8100,810),JO                                               EUC 0305
  810 IF(N) 8100,8100,811                                               EUC 0310
 8100 DET=0.                                                            EUC 0320
      N=M                                                               EUC 0330
      GO TO 30                                                          EUC 0340
  811 CALL SCALE(R,PRE,N,JPRE,NS,KRUB)                                  EUC 0350
      CALL OVERFL(JO)                                                   EUC 0360
      GO TO (8100,7),JO                                                 EUC 0365
    7 CALL DETERM(N,DET,R,NJL)                                          EUC 0370
C     STORE RESULTANT                                                   EUC 0380
      LEST(IXAL)=NS                                                     EUC 0390
      JEST(IXAL)=KRUB+NJL                                               EUC 0400
      CALL OVERFL(JO)                                                   EUC 0410
      GO TO (8100,30),JO                                                EUC 0415
   30 REST(IXAL)=DET                                                    EUC 0420
      GO TO (31,38,501,506),INDEX                                       EUC 0430
   31 CALL WRITER(ALP,REST,LEST,MAXAL,JEST)                             EUC 0440
C     SEARCH FOR SIGN CHANGE OF REMAINDER                               EUC 0450
      JSUB=2                                                            EUC 0460
      INDEX=2                                                           EUC 0470
   32 DO 33 J=JSUB,MAXAL                                                EUC 0480
      IF( ABS(REST(J))/REST(J)*REST(J-1)) 325,33,33                     EUC 0490
  325 JSUB=J+1                                                          EUC 0500
      GO TO 35                                                          EUC 0510
   33 CONTINUE                                                          EUC 0520
   34 RETURN                                                            EUC 0530
C     INTERPOLATE FOR FLUTTER POINT                                     EUC 0540
   35 J=JSUB-1                                                          EUC 0550
      I1=J7                                                             EUC 0560
      I2=J7                                                             EUC 0570
      KIT=0                                                             EUC 0580
      AP=ALP(J)                                                         EUC 0590
      AN=ALP(J-1)                                                       EUC 0600
      RP=REST(J)                                                        EUC 0610
      RN=REST(J-1)                                                      EUC 0620
      JP=JEST(J)                                                        EUC 0630
      JN=JEST(J-1)                                                      EUC 0640
      LP=LEST(J)                                                        EUC 0650
      LN=LEST(J-1)                                                      EUC 0660
      IF(RP) 36,365,365                                                 EUC 0670
   36 DUB=AP                                                            EUC 0680
      AP=AN                                                             EUC 0690
      AN=DUB                                                            EUC 0700
      RUB=RP                                                            EUC 0710
      RP=RN                                                             EUC 0720
      RN=RUB                                                            EUC 0730
      LUB=LN                                                            EUC 0740
      LN=LP                                                             EUC 0750
      LP=LUB                                                            EUC 0760
      LUB=JN                                                            EUC 0770
      JN=JP                                                             EUC 0780
      JP=LUB .                                                          EUC 0790
  365 LDIF=(LN-LP)*N+JN-JP                                              EUC 0800
```

```
      TWOL=2.**LDIF                                                     EUC 0810
   37 RANGE=AN-AP                                                       EUC 0820
      DUB    =RN/RP*TWOL                                                EUC 0830
      RATIO=1./(1.-DUB)                                                 EUC 0850
      IF(RATIO-.9) 372,372,371                                          EUC 0860
  371 RATIO=.9                                                          EUC 0870
      GO TO 374                                                         EUC 0880
  372 IF(RATIO-.1) 373,374,374                                          EUC 0890
  373 RATIO=.1                                                          EUC 0900
  374 ALP(J7)=AP+RATIO*RANGE                                            EUC 0910
      GO TO 804                                                         EUC 0920
C     TEST FOR ACCEPTANCE OF ALPHA                                      EUC 0930
   38 IF(ABS(RANGE    )-TESTR*AN    )  42,42,39                         EUC 0940
   39 IF(REST(J7)) 40,41,41                                             EUC 0950
   40 RN=REST(J7)                                                       EUC 0960
      KIT=KIT+1                                                         EUC 0970
      LN=LEST(J7)                                                       EUC 0980
      JN=JEST(J7)                                                       EUC 0990
      IF (AN-AL) 401,42,401                                             EUC 1000
  401 AN=AL                                                             EUC 1010
      GO TO 37                                                          EUC 1020
   41 RP=RFST(J7)                                                       EUC 1030
      KIT=KIT+1                                                         EUC 1040
      LP=LEST(J7)                                                       EUC 1050
      JP=JEST(J7)                                                       EUC 1060
      IF(AP-AL) 411,42,411                                              EUC 1070
  411 AP=AL                                                             EUC 1080
      GO TO 37                                                          EUC 1090
C     WRITE FLUTTER DATA AND LOOP                                       EUC 1100
   42 LSC=LEST(J7)                                                      EUC 1110
   43 CALL FLUT(PRE,N,LSC,U,RES,JPRE,1)                                 EUC 1120
   50 INDEX=3                                                           EUC 1130
      AF=AL                                                             EUC 1140
      LF=LEST(J7)                                                       EUC 1150
      JF=JEST(J7)                                                       EUC 1160
      RF=REST(J7)                                                       EUC 1170
      ALP(J7)=AN                                                        EUC 1180
      GO TO 804                                                         EUC 1190
  501 LSC=LEST(J7)                                                      EUC 1200
      CALL FLUT(PRE,N,LSC,UN,RES,JPRE,1)                                EUC 1210
      INDEX=4                                                           EUC 1220
      ALP(J7)=AP                                                        EUC 1230
      GO TO 804                                                         EUC 1240
  506 LSC=LEST(J7)                                                      EUC 1250
      CALL FLUT(PRE,N,LSC,UP,RES,JPRE,1)                                EUC 1260
      WRITE(6,507) KIT                                                  EUC 1270
  507 FORMAT(1XI5,11H ITERATIONS)                                       EUC 1280
      KIT=0                                                             EUC 1290
  508 CALL WROUT(AF, U,RF,LF,0,JF,F12)                                  EUC 1300
      CALL WROUT(AN,UN,RN,LN,1,JN,F12)                                  EUC 1310
      CALL WROUT(AP,UP,RP,LP,1,JP,F12)                                  EUC 1320
   55 CALL CBAR(AL,G,AR,AI,E,N,CR,FACTR)                                EUC 1330
   56 INDEX=2                                                           EUC 1340
   57 CALL VECTOR( U,CR,N,L,DJ)                                         EUC 1350
   58 IF(JSUB-MAXAL) 32,32,34                                           EUC 1370
      END                                                               EUC 1380


      SUBROUTINE CBAR(AL,G,AR,AI,E,N,CR,FACTR)                          CBAR0020
C***  SET UP MATRIX CBAR                                                CBAR0025
      DOUBLE PRECISION CR(10,20)                                        CBAR0030
      DIMENSION E(10,10),AR(10,10),AI(10,10)                            CBAR0040
      P=FACTR                                                           CBAR0050
      ALF=AL*P                                                          CBAR0060
      DO 806 J=1,N                                                      CBAR0070
      DO 806 K=1,N                                                      CBAR0080
      P=E(J,K)                                                          CBAR0090
      Q=AR(J,K)                                                         CBAR0100
      R=AI(J,K)*G                                                       CBAR0110
      S=AI(J,K)                                                         CBAR0120
      T=AR(J,K)*G                                                       CBAR0130
```

```
      U=E(J,K)*G
      CR(J,K)=ALF*P+Q+R
  806 CR(J,K+10)=S-T-ALF*U
   99 RETURN
      END


      SUBROUTINE EQCHAR(AA,PRE,N1)
      DOUBLE PRECISION AA(10,20),A(10,20),D(11,22),X(20),TEMP,TEMPI,DENO
      DOUBLE PRECISION PRE(22)
      N=N1
      M=N
      M1=M-1
      M2=M-2
      PRE(N)=0.
      PRE(N+11)=0.
      IF(M2)140,130,143
  130 PRE(1)=AA(1,1)*AA(2,2)-AA(1,2)*AA(2,1)-AA(1,11)*AA(2,12)+AA(1,12)*
     1AA(2,11)
      PRE(12)=AA(1,1)*AA(2,12)+AA(1,11)*AA(2,2)-AA(1,2)*AA(2,11)-AA(2,1)
     1*AA(1,12)
      PRE(2)=-AA(2,2)
      PRE(13)=-AA(2,12)
  140 PRE(N)=PRE(N)-AA(1,1)
      PRE(N+11)=PRE(N+11)-AA(1,11)
      GO TO 92
  143 DO 144 J=1,N
      DO 144 K=1,N
      A(J,K)=AA(J,K)
  144 A(J,K+10)=AA(J,K+10)
C     INSURE MAXIMUM SUB-DIAGONAL ELEMENT AT PIVOT
  100 REFM=A(M,M1)**2+A(M,M1+10)**2
      I=M1
      DO 104 J=1,M2
      REFT=A(M,J)**2+A(M,J+10)**2
      IF(REFM-REFT)101,104,104
  101 I=J
      REFM=REFT
  104 CONTINUE
      IF(I-M1)210,105,105
  105 IF(REFM)106,106,22
C     DECOUPLED
  106 WRITE(6,107)
      WRITE(6,108) ((A(J,K),K=1,N),J=1,N)
      WRITE(6,107)
      WRITE(6,108) ((A(J,K+10),K=1,N),J=1,N)
  107 FORMAT(10H DECOUPLED)
      N1=0
      GO TO 92
  108 FORMAT(6(D11.3,2X))
C     INTERCHANGE ROWS AND COLUMNS
  210 DO 212 J=1,M
      TEMP=A(J,I)
      TEMPI=A(J,I+10)
      A(J,I)=A(J,M1)
      A(J,I+10)=A(J,M1+10)
      A(J,M1+10)=TEMPI
  212 A(J,M1)=TEMP
      DO 213 J=1,N
      TEMP=A(M1,J)
      TEMPI=A(M1,J+10)
      A(M1,J)=A(I,J)
      A(M1,J+10)=A(I,J+10)
      A(I,J+10)=TEMPI
  213 A(I,J)=TEMP
C     SIMILARITY TRANSFORM
   22 DENO=A(M,M1)**2+A(M,M1+10)**2
      TEMP=A(M,M1)/DENO
      TEMPI=-A(M,M1+10)/DENO
      DO 30 J=1,M2
      X(J+10)=A(M,J)*TEMPI+A(M,J+10)*TEMP
```

```
30    X(J)=A(M,J)*TEMP-A(M,J+10)*TEMPI                                    CHEQ0660
      X(M1)=1.                                                            CHEQ0670
      X(M1+10)=0.                                                         CHEQ0680
      DO 31 J=M,N                                                         CHEQ0690
      X(J+10)=0.                                                          CHEQ0700
31    X(J)=0.                                                             CHEQ0710
      DO 42 K=1,M2                                                        CHEQ0720
      DO 41 J=1,N                                                         CHEQ0730
      A(J,K+10)=A(J,K+10)-X(K)*A(J,M1+10)-X(K+10)*A(J,M1)                 CHEQ0740
41    A(J,K)=A(J,K)-X(K)*A(J,M1)+X(K+10)*A(J,M1+10)                       CHEQ0750
      A(M,K+10)=0.                                                        CHEQ0760
42    A(M,K)=0.                                                           CHEQ0770
      DO 52 J=1,M2                                                        CHEQ0780
      DO 52 K=1,N                                                         CHEQ0790
      A(M1,K+10)=A(M1,K+10)+X(J)*A(J,K+10)+X(J+10)*A(J,K)                 CHEQ0800
52    A(M1,K)=A(M1,K)+X(J)*A(J,K)-X(J+10)*A(J,K+10)                       CHEQ0810
      IF(M2-1)99,99,62                                                    CHEQ0820
62    M=M1                                                                CHEQ0830
      M1=M2                                                               CHEQ0840
      M2=M2-1                                                             CHEQ0850
      GO TO 100                                                           CHEQ0860
99    GO TO 990                                                           CHEQ0870
990   CALL CHAR(A,D,N)                                                    CHEQ0880
90    NP=N+1                                                              CHEQ0890
      DO 91 J=1,NP                                                        CHEQ0900
      PRE(J+11)=D(NP,J+11)                                                CHEQ0910
91    PRE(J)=D(NP,J)                                                      CHEQ0920
92    RETURN                                                              CHEQ0930
      END                                                                 CHEQ0940


      SUBROUTINE CHAR(A,D,L)                                              CHAR0010
      DOUBLE PRECISION A(10,20),D(11,22),SUM,SUMI,FAC,FACI,ZR,ZI,TERM,    CHAR0015
     1TERMI,RUB                                                           CHAR0020
      MAX=L                                                               CHAR0030
      NP=MAX+1                                                            CHAR0040
      DO 1 J=1,NP                                                         CHAR0050
      D(J,J)=1.                                                           CHAR0060
1     D(J,J+11)=0.                                                        CHAR0070
      DO 10 N=1,MAX                                                       CHAR0080
      N10=N+10                                                            CHAR0090
      DO 10 K=1,N                                                         CHAR0100
      NK=N-K                                                              CHAR0110
      NK1=NK+1                                                            CHAR0120
      NK12=NK1+11                                                         CHAR0130
      SUM=0.                                                              CHAR0140
      SUMI=0.                                                             CHAR0150
      FAC=1.                                                              CHAR0160
      FACI=0.                                                             CHAR0170
      DO 5 M=1,K                                                          CHAR0180
      NM=N-M                                                              CHAR0190
      NM1=NM+1                                                            CHAR0200
      NM10=NM+10                                                          CHAR0210
      ZR=A(NM1,N)*D(NM1,NK1)-A(NM1,N10)*D(NM1,NK12)                       CHAR0220
      ZI=A(NM1,N)*D(NM1,NK12)+A(NM1,N10)*D(NM1,NK1)                       CHAR0230
      TERM=FAC*ZR-FACI*ZI                                                 CHAR0240
      TERMI=FAC*ZI+FACI*ZR                                                CHAR0250
      SUM=SUM+TERM                                                        CHAR0260
      SUMI=SUMI+TERMI                                                     CHAR0270
      IF(K-M) 51,51,4                                                     CHAR0280
4     RUB=FAC*A(NM1,NM)-FACI*A(NM1,NM10)                                  CHAR0290
      FACI=FAC*A(NM1,NM10)+FACI*A(NM1,NM)                                 CHAR0300
5     FAC=RUB                                                             CHAR0310
51    IF(NK) 8,8,6                                                        CHAR0320
6     D(N+1,NK1)=D(N,NK)-SUM                                              CHAR0330
      D(N+1,NK12)=D(N,NK12-1)-SUMI                                        CHAR0340
      GO TO 10                                                            CHAR0350
8     D(N+1,1)=-SUM                                                       CHAR0360
      D(N+1,12)=-SUMI                                                     CHAR0370
10    CONTINUE                                                            CHAR0380
99    RETURN                                                              CHAR0390
```

```
      END                                                               CHAR0400

      SUBROUTINE SCALE(RR,P,NN,JPRE,NS,KRUB)                            SCAL2210
      DOUBLE PRECISION RR(11,10),R(11,10),P(22),SF,RUB                  SCAL2230
      N=NN                                                              SCAL2260
  700 EN=N                                                              SCAL2270
      NP=N+1                                                            SCAL2280
      DO 1 J=1,N                                                        SCAL2290
      K=NP-J                                                            SCAL2300
      JPR=JPRE+J                                                        SCAL2310
      R(1,K)=P(JPR)                                                     SCAL2320
    1 R(NP,K)=P(J)                                                      SCAL2330
  701 SUMN2=(N*NP*(N+NP))/6                                             SCAL2340
      SCA=0.                                                            SCAL2350
      DO 410 J=1,N                                                      SCAL2360
      C=J                                                               SCAL2370
      AB=DABS(R(NP,J))                                                  SCAL2380
      IF(AB) 410,410,409                                                SCAL2390
  409 ROG2=C*ALOG(AB)                                                   SCAL2400
      SCA=SCA+ROG2                                                      SCAL2410
  410 CONTINUE                                                          SCAL2420
      NS=ABS (1.44270*SCA/SUMN2)+.5                                     SCAL2430
      IF(SCA) 4100,4101,4101                                            SCAL2440
 4100 NS=-NS                                                            SCAL2450
 4101 FAC=1.0                                                           SCAL2460
      SF=0.5**NS                                                        SCAL2470
      DO 411 K=1,N                                                      SCAL2480
      FAC=FAC*SF                                                        SCAL2490
      R(1,K)=R(1,K)*FAC                                                 SCAL2500
  411 R(NP,K)=R(NP,K)*FAC                                               SCAL2510
      DO 5 K=2,N                                                        SCAL2530
      RUB=-R(K-1,1)                                                     SCAL2540
      R(K,N)=RUB*R(NP,N)                                                SCAL2550
      DO 5 J=2,N                                                        SCAL2560
    5 R(K,J-1)=R(K-1,J)+RUB*R(NP,J-1)                                   SCAL2570
   60 KRUB=0                                                            SCAL2580
      NM=N-1                                                            SCAL2590
C              DO NOT SCALE LAST ROW UNLESS YOU WANT TROUBLE            SCAL2600
      DO 307 J=1,NM                                                     SCAL2610
      RUB=0.                                                            SCAL2620
      DO 304 K=1,N                                                      SCAL2630
      AB=DABS(R(J,K))                                                   SCAL2640
      IF(AB) 301,304,301                                                SCAL2650
  301 ROG2=ALOG(AB)*1.4427                                              SCAL2660
      RUB=RUB+ROG2                                                      SCAL2670
  304 CONTINUE                                                          SCAL2680
      LRUB=RUB/EN+.5                                                    SCAL2690
      KRUB=KRUB+LRUB                                                    SCAL2700
  305 SF=0.5**LRUB                                                      SCAL2710
      DO 306 K=1,N                                                      SCAL2720
  306 RR(J,K)=R(J,K)*SF                                                 SCAL2730
  307 CONTINUE                                                          SCAL2740
      DO 308 K=1,N                                                      SCAL2750
  308 RR(N,K)=R(N,K)                                                    SCAL2760
   99 RETURN                                                            SCAL2770
      END                                                               SCAL2780


      SUBROUTINE DETERM(MM,RES,R,NSUM)                                  DETE0650
C***  DOUBLE PRECISION DETERMINANT                                      DETE0660
      DOUBLE PRECISION R(11,10),A(10,10),DET,RES,DSIGN,AB,RUB,RA        DETE0670
      M=MM                                                              DETE0680
      NSUM=0                                                            DETE0690
      IF(M-1) 26,26,27                                                  DETE0700
   26 DET=R(1,1)                                                        DETE0710
      GO TO 99                                                          DETE0720
   27 DO 28 J=1,M                                                       DETE0730
      DO 28 K=1,M                                                       DETE0740
   28 A(J,K)=R(J,K)                                                     DETE0750
```

```
      DSIGN=1.                                                          DETE0760
C                K LOOP                                                 DETE0770
      DO 30 LT=2,M                                                      DETE0780
      K=LT-1                                                            DETE0790
      L=0                                                               DETE0800
      BIG=0.                                                            DETE0810
      DO 100 I=K,M                                                      DETE0820
      AB=DABS(A(K,I))                                                   DETE0830
      IF(BIG-AB) 150,100,100                                            DETE0840
  150 L=I                                                               DETE0850
      BIG=AB                                                            DETE0860
  100 CONTINUE                                                          DETE0870
      IF(L) 16,16,13                                                    DETE0880
   13 IF(K-L) 14,5,5                                                    DETE0890
C              IF DIAGONAL ELEMENT IS NOT MAX, INTERCHANGE COLUMNS      DETE0900
   14 DO 8 J=K,M                                                        DETE0910
      RUB=A(J,K)                                                        DETE0920
      A(J,K)=A(J,L)                                                     DETE0930
    8 A(J,L)=RUB                                                        DETE0940
      DSIGN=-DSIGN                                                      DETE0950
      IF(A(K,K)) 5,13,5                                                 DETE0960
C                REDUCTION LOOP                                         DETE0970
    5 RA=1./A(K,K)                                                      DETE0980
      DO 3 I=LT,M                                                       DETE0990
       AB=RA*A(I,K)                                                     DETE1000
      DO 3 J=LT,M                                                       DETE1010
    3 A(I,J)=A(I,J)-AB*A(K,J)                                           DETE1020
   30 CONTINUE                                                          DETE1030
C                FORM PRODUCT OF DIAGONAL ELEMENTS                      DETE1040
    4 DO 6 J=2,M                                                        DETE1050
C               SCALE TO PREVENT UNDERFLOW                              DETE1060
      B=DABS(A(J,J))                                                    DETE1070
      IF(B) 16,16,50                                                    DETE1080
   50 N=1.4427*ALOG(B)                                                  DETE1090
      NSUM=NSUM+N                                                       DETE1100
    6 A(J,J)=A(J,J)*(.5**N)*A(J-1,J-1)                                  DETE1110
   98 DET=DSIGN*A(M,M)                                                  DETE1120
   99 RES=DET                                                           DETE1130
      RETURN                                                            DETE1140
C               MATRIX IS SINGULAR, UNREMOVABLE ZERO ON DIAGONAL        DETE1150
   16 DSIGN=0.                                                          DETE1160
      GO TO 98                                                          DETE1170
      END                                                               DETE1180


      SUBROUTINE WRITER(A,R,L,ML,JE)                                    WRIT0020
C***  WRITE MU VERSUS RESULTANT TABLES                                  WRIT0025
      DIMENSION A(51),R(51),L(51),JE(51)                                WRIT0030
      WRITE            (6,600)                                          WRIT0040
      NR=ML/4                                                           WRIT0050
      IF(4*NR-ML)7,8,8                                                  WRIT0060
    7 NR=NR+1                                                           WRIT0070
    8 DO 10 J=1,NR                                                      WRIT0080
   10 WRITE            (6,601)(A(2*K-1),R(K),L(K),JE(K),K=J,ML,NR)      WRIT0090
   99 RETURN                                                            WRIT0100
  600 FORMAT(5X2HMU8X3HRES4X7HSFA SFR5X2HMU8X3HRES4X7HSFA SFR5X2HMU8X3HRWRIT0110
     1ES4X7HSFA SFR5X2HMU8X3HRES4X7HSFA SFR)                            WRIT0120
  601 FORMAT(4(1PE11.2,E11.2,I3,I4))                                    WRIT0130
      END                                                               WRIT0140


      SUBROUTINE FLUT(PRE,NR,NS,U,RES,JPRE,I)                           FLUT0070
      DOUBLE PRECISION PRE(22),A(11),B(11),C(11),DENOM,ROOT,UU,PSUM,    FLUT0080
     1PDSUM,QSUM,QDSUM,REST                                             FLUT0090
      IN=I                                                              FLUT0140
      N=NR                                                              FLUT0150
      JPR=JPRE                                                          FLUT0160
C     COPY THE CHARACTERISTIC POLYNOMIALS                               FLUT0170
      JJ=N+1                                                            FLUT0180
      DO 2 J=1,JJ                                                       FLUT0190
```

```
      A(J)=PRE(J)                                                       FLUT0200
      J11=J+11                                                          FLUT0210
    2 B(J)=PRE(J11)                                                     FLUT0220
      NN=N-1                                                            FLUT0230
      DENOM=B(N)                                                        FLUT0240
      DO 3 K=1,NN                                                       FLUT0250
    3 B(K)=B(K)/DENOM                                                   FLUT0260
      B(N)=1.                                                           FLUT0270
    6 NO=N-1                                                            FLUT0280
      AA=A(1)/B(1)                                                      FLUT0290
      DENOM=A(NO+1)-B(NO)-B(NO+1)*AA                                    FLUT0300
      DO 10 K=1,NO                                                      FLUT0310
   10 C(K)=(A(K+1)-B(K)-B(K+1)*AA)/DENOM                                FLUT0320
      IF(N-3)30,30,14                                                   FLUT0330
   14 A(N)=B(N)                                                         FLUT0340
      DO 15 J=1,NO                                                      FLUT0350
      A(J)=B(J)                                                         FLUT0360
   15 B(J)=C(J)                                                         FLUT0370
      N=N-1                                                             FLUT0380
      GO TO 6                                                           FLUT0390
   30 ROOT=-C(1)                                                        FLUT0400
      IF(IN)25,20,25                                                    FLUT0410
   20 REST=B(2)+B(1)/ROOT+ROOT                                          FLUT0420
      RES=REST                                                          FLUT0430
  100 RETURN                                                            FLUT0440
C     CALCULATE SECOND ORDER ESTIMATE OF ROOT                           FLUT0450
   25 DO 32 J=1,JJ                                                      FLUT0460
      A(J)=PRE(J)                                                       FLUT0470
      J11=J+11                                                          FLUT0480
   32 B(J)=PRE(J11)                                                     FLUT0490
      PSUM=A(NR+1)                                                      FLUT0500
      PDSUM=0.                                                          FLUT0510
      QSUM=B(NR+1)                                                      FLUT0520
      QDSUM=0.                                                          FLUT0530
      DO 35 K=1,NR                                                      FLUT0540
      L=NR-K+1                                                          FLUT0550
      XL=L                                                              FLUT0560
      PSUM=ROOT*PSUM+A(L)                                               FLUT0570
      PDSUM=ROOT*PDSUM+A(L+1)*XL                                        FLUT0580
      QSUM=ROOT*QSUM+B(L)                                               FLUT0590
   35 QDSUM=ROOT*QDSUM+B(L+1)*XL                                        FLUT0600
      UU=(.5*(PSUM/PDSUM+QSUM/QDSUM)-ROOT)                              FLUT0610
      U=UU                                                              FLUT0620
      GO TO 100                                                         FLUT0630
      END                                                               FLUT0640


      SUBROUTINE WROUT(A,U,R,L,N,JL,FAC)                                WROU0020
C                 FLUTTER POINTS FOR PANEL                              WROU0030
      A3=(U*FAC*2.)**.33333333                                          WROU0040
      RU=1./A                                                           WROU0050
      IF(N ) 617,617,619                                                WROU0060
  617 WRITE             (6,618)                                         WROU0070
  618 FORMAT(1H07X2HMU10X6HZ**1/315X5HLAMDA10X3HRES7X7HSFA SFR4X4H1/MU) WROU0080
C            BEWARE-- SYMBOLS FOR MU AND ALPHA ARE INTERCHANGED BACK    WROU0090
  619 WRITE             (6,620)A,A3,U,R,L,JL,RU                         WROU0100
  620 FORMAT(F14.6,F16.8,1PE19.8,E14.3,I6,I4,0PF11.6)                   WROU0110
      RETURN                                                            WROU0120
      END                                                               WROU0130


      SUBROUTINE VECTOR(EW,C,I,L,DJ)                                    VECT1180
      DOUBLE PRECISION P(22),C(10,20)                                   VECT1190
      COMPLEX R(10),CK(10),S(10,10),SS                                  VECT1200
      DIMENSION VM(10),VT(10),DJ(10,10)                                 VECT1210
      M=I                                                               VECT1220
       CALL EQCHAR(C,P,M)                                               VECT1240
      MN=M-1                                                            VECT1250
C             FORM CB=C+EIGENVALUE*I                                    VECT1260
      DO 5 J=1,M                                                        VECT1270
```

```
    5  C(J,J)=C(J,J)+EW                                                  VECT1280
C              COPY CB MATRIX                                           VECT1290
       DO 199 J=1,M                                                     VECT1300
       DO 199 K=1,M                                                     VECT1310
  199  S(J,K)=CMPLX(C(J,K),C(J,K+10))                                   VECT1330
C              INVERT REDUCED CB MATRIX                                 VECT1340
       CALL INVCX(S,MN)                                                 VECT1350
C              MULTIPLY INVERSE*(-LAST COLUMN)                          VECT1360
       DO 10 J=1,MN                                                     VECT1370
       R(J)=0.                                                          VECT1380
       DO 10 K=1,MN                                                     VECT1390
   10  R(J)=R(J)-S(J,K)*S(K,M)                                          VECT1400
       R(M)=1.                                                          VECT1410
       WRITE(6,12)                                                      VECT1420
   12  FORMAT(9H0VECTOR R,9X5HTHETA,11X17HCOMPLEX RESIDUALS)            VECT1430
C              CONVERT TO POLAR FORM                                    VECT1440
       DO 16 J=1,MN                                                     VECT1450
       VM(J)=CABS(R(J))                                                 VECT1460
       VT(J)=ATAN2(AIMAG(R(J)),REAL(R(J)))                              VECT1465
   16  VT(J)=AMOD(VT(J)+6.2831953,6.2831953)                            VECT1480
       CALL VECNRM(VM,L,M,DJ)                                           VECT1530
       VM(M)=1.                                                         VECT1540
       VT(M)=0.                                                         VECT1550
       IMAX=0                                                           VECT1560
       VMAX=1.                                                          VECT1570
       DO 20 J=1,MN                                                     VECT1580
       IF(VMAX-VM(J)) 17,20,20                                          VECT1590
   17  VMAX=VM(J)                                                       VECT1600
       IMAX=J                                                           VECT1610
       VTHET=VT(IMAX)                                                   VECT1620
   20  CONTINUE                                                         VECT1630
       IF(IMAX) 23,23,21                                                VECT1640
   21  DO 22 J=1,M                                                      VECT1650
       VM(J)=VM(J)/VMAX                                                 VECT1660
   22  VT(J)=VT(J)-VTHET                                                VECT1670
C              MULTIPLY VECTOR BY MATRIX FOR CHECK                      VECT1680
   23  DO 24 J=1,M                                                      VECT1690
       CK(J)=0.                                                         VECT1700
       DO 200 K=1,M                                                     VECT1710
       SS =   CMPLX(C(J,K),C(J,K+10))                                   VECT1730
  200  CK(J)=CK(J)+SS*R(K)                                              VECT1740
       DO 201 N=1,M                                                     VECT1750
       DO 201 K=1,M                                                     VECT1760
  201  S(N,K)=CMPLX(C(N,K),C(N,K+10))                                   VECT1780
   24  WRITE            (6,25)VM(J),VT(J),CK(J)                         VECT1790
   25  FORMAT(2F12.6,       1PE18.2,E11.2)                              VECT1800
C***   FIND FIGURES OF MERIT                                            VECT1805
       E=EW                                                             VECT1820
       U=-EW                                                            VECT1830
       PMU=U+P(M)                                                       VECT1840
       PMUP=M                                                           VECT1850
       QMU=P(M+11)                                                      VECT1860
       QMUP=0.                                                          VECT1870
       MM1=M-1                                                          VECT1880
       DO 108 J=1,MM1                                                   VECT1890
       K=M-J                                                            VECT1900
       CAA=K                                                            VECT1910
       PMU=PMU*U+P(K)                                                   VECT1920
       QMU=QMU*U+P(K+11)                                                VECT1930
       PMUP=PMUP*U+P(K+1)*CAA                                           VECT1940
  108  QMUP=QMUP*U+P(K +12)*CAA                                         VECT1950
       R3=PMU/(PMUP*U)                                                  VECT1960
       R4=QMU/(QMUP*U)                                                  VECT1970
       MP=M+1                                                           VECT1980
       WRITE(6,628)                                                     VECT1990
       WRITE(6,629) (P(J),J=1,MP)                                       VECT2000
       WRITE(6,629) (P(J+11),J=1,M)                                     VECT2010
  628  FORMAT(35HOCHARACTERISTIC EQUATION AT FLUTTER)                   VECT2020
  629  FORMAT(1P8E16.7)                                                 VECT2030
       WRITE(6,631) PMU,QMU,R3,R4                                       VECT2070
  630  FORMAT(7H LAMDA=F10.4,8H   DET= 1P2E11.4)                        VECT2080
  631  FORMAT(3HOP=1PE11.4,3H Q=E11.4,7H P/P1A=E11.4,7H Q/Q1A=E11.4)    VECT2090
```

```
C           EVALUATE DETERMINANT FOR FLUTTER ALPHA AND MU*(1,1.02,.98)VECT2095
      DO 28 J=1,3                                                       VECT2100
      FL=J*(7-J-J)-5                                                    VECT2110
      FL=.02*FL                                                         VECT2120
      DO 26 K=1,M                                                       VECT2130
   26 S(K,K)=S(K,K)+FL*EW                                               VECT2140
      E=E*(1.+FL)                                                       VECT2150
      CALL CXDET(M,X,Y,S)                                               VECT2160
      U=ABS(E)                                                          VECT2170
   28 WRITE(6,630) U,X,Y                                                VECT2180
   99 RETURN                                                            VECT2190
      END                                                               VECT2200


      SUBROUTINE INVCX(R,N)                                             INVC0020
C***  COMPLEX MATRIX INVERSION                                           INVC0025
      COMPLEX R(10,10),D                                                INVC0030
   40 IF(N-1) 37,37,38                                                  INVC0040
   37 R(1,1)=1./R(1,1)                                                  INVC0050
      GO TO 99                                                          INVC0060
   38 DO 41 K=1,N                                                       INVC0070
      D=R(K,K)                                                          INVC0080
      R(K,K)=1.                                                         INVC0090
   50 DO 42 J=1,N                                                       INVC0100
   42 R(K,J)=R(K,J)/D                                                   INVC0110
   56 IF (K-N) 43,44,44                                                 INVC0120
   43 KPLUS=K+1                                                         INVC0130
   51 DO 41 I=KPLUS,N                                                   INVC0140
      D=R(I,K)                                                          INVC0150
      R(I,K)=0.                                                         INVC0160
   52 DO 41 J=1,N                                                       INVC0170
   41 R(I,J)=R(I,J)-D*R(K,J)                                            INVC0180
   44 NMINUS=N-1                                                        INVC0190
   53 DO 45 K=1,NMINUS                                                  INVC0200
      KPLUS=K+1                                                         INVC0210
   54 DO 45 I=KPLUS,N                                                   INVC0220
      D=R(K,I)                                                          INVC0230
      R(K,I)=0.                                                         INVC0240
   55 DO 45 J=1,N                                                       INVC0250
   45 R(K,J)=R(K,J)-D*R(I,J)                                            INVC0260
   99 RETURN                                                            INVC0270
      END                                                               INVC0280


      SUBROUTINE VECNRM(VM,L,MMAX,DJAY)                                 VECN0020
C     TO NORMALIZE VECTOR ON BASIS OF EQUAL MODAL RMS                   VECN0030
      DIMENSION VM(10),DJAY(10,10)                                      VECN0040
      EL=L                                                              VECN0050
      VM(MMAX)=1.                                                       VECN0060
      DO 20 J=1,MMAX                                                    VECN0070
      VPM=DJAY(J,J)/EL                                                  VECN0080
      VPM=SQRT(VPM)                                                     VECN0090
   20 VM(J)=VM(J)*VPM                                                   VECN0100
      MINUS=MMAX-1                                                      VECN0110
      DO 30 J=1,MINUS                                                   VECN0120
   30 VM(J)=VM(J)/VM(MMAX)                                              VECN0130
      RETURN                                                            VECN0140
      END                                                               VECN0150


      SUBROUTINE CXDET(MM,X,Y,R)                                        CXDT0401
C***  COMPLEX DETERMINANT                                                CXDT0402
      COMPLEX R(10,10),A(10,10),DET,DSIGN,RUB,RA,AB                     CXDT0402
      M=MM                                                              CXDT0404
      IF(M-1) 26,26,27                                                  CXDT0405
   26 DET=R(1,1)                                                        CXDT0406
      GO TO 99                                                          CXDT0407
   27 DO 28 J=1,M                                                       CXDT0408
      DO 28 K=1,M                                                       CXDT0409
```

```
 28     A(J,K)=R(J,K)                                                CXDT0410
        DSIGN=1.                                                     CXDT0411
C          K LOOP                                                    CXDT0412
        DO 30 LT=2,M                                                 CXDT0413
        K=LT-1                                                       CXDT0414
        IF(CABS(A(K,K)))5,7,5                                        CXDT0415
C          ZERO ON MAIN DIAGONAL,INTERCHANGE ROWS                    CXDT0416
  7     IF(M-K) 16,16,71                                             CXDT0417
 71     L=LT                                                         CXDT0418
 14     DO 8 J=K,M                                                   CXDT0419
        RUB=A(J,K)                                                   CXDT0420
        A(J,K)=A(J,L)                                                CXDT0421
  8     A(J,L)=RUB                                                   CXDT0422
        DSIGN=-DSIGN                                                 CXDT0423
        IF(CABS(A(K,K)))5,13,5                                       CXDT0424
C          ZERO STILL ON DIAGONAL                                    CXDT0425
 13     L=L+1                                                        CXDT0426
        IF(L-M)14,14,16                                              CXDT0427
C           REDUCTION LOOP                                           CXDT0428
  5     RA=    1. /A(K,K)                                            CXDT0429
        DO 3 I=LT,M                                                  CXDT0430
        AB=RA*A(I,K)                                                 CXDT0431
        DO 3 J=LT,M                                                  CXDT0432
  3     A(I,J)=A(I,J)-AB*A(K,J)                                      CXDT0433
 30     CONTINUE                                                     CXDT0434
C          FORM PRODUCT OF DIAGONAL ELEMENTS                         CXDT0435
  4     DO 6 J=2,M                                                   CXDT0436
  6     A(J,J)=A(J,J)*A(J-1,J-1)                                     CXDT0437
 98     DET=DSIGN*A(M,M)                                             CXDT0438
        X=REAL(DET)                                                  CXDT0439
        Y=AIMAG(DET)                                                 CXDT0440
 99     RETURN                                                       CXDT0441
C          MATRIX IS SINGULAR,UNREMOVABLE ZERO ON DIAG               CXDT0442
 16     DSIGN=0.                                                     CXDT0443
        GO TO 98                                                     CXDT0444
        END                                                          CXDT0445
```

```
        FUNCTION FMM(A1,A2,M,MBAR,ALPHA)                             FMM 0020
        DIMENSION GAMMA(11)                                          FMM 0030
        COMMON GAMMA                                                 FMM 0040
        GAM4=GAMMA(M)**4                                             FMM 0050
        FM01=FMKT(0,M,A1)                                            FMM 0060
        FM11=FMKT(1,M,A1)                                            FMM 0070
        FM21=FMKT(2,M,A1)                                            FMM 0080
        FM31=FMKT(3,M,A1)                                            FMM 0090
        IALF=ALPHA+1.                                                FMM 0100
        GO TO (5,3,4),IALF                                           FMM 0110
      5 FMA01=FM01                                                   FMM 0120
        FMA11=FM11                                                   FMM 0130
        FMA21=FM21                                                   FMM 0140
        FMA31=FM31                                                   FMM 0150
        GO TO 2                                                      FMM 0160
    3   FMA01=FM11                                                   FMM 0170
        FMA11=FM21                                                   FMM 0180
        FMA21=FM31                                                   FMM 0190
        FMA31=FMKT(4,M,A1)                                           FMM 0200
        GO TO 2                                                      FMM 0210
    4   FMA01=FM21                                                   FMM 0220
        FMA11=FM31                                                   FMM 0230
        FMA21=FMKT(4,M,A1)                                           FMM 0240
        FMA31=FMKT(5,M,A1)                                           FMM 0250
      2 IF(M-MBAR)10,1,10                                            FMM 0260
  1     FM02=FMKT(0,M,A2)                                            FMM 0270
        FM12=FMKT(1,M,A2)                                            FMM 0280
        FM22=FMKT(2,M,A2)                                            FMM 0290
        FM32=FMKT(3,M,A2)                                            FMM 0300
    6   TERM1=0.25*A2*FMA01*FM02                                     FMM 0310
        TERM2=0.25*A2*(FMA11*FM32+FMA31*FM12)                        FMM 0320
        TERM3=0.125*(FMA21*FM12+FMA11*FM22)                          FMM 0330
        TERM4=0.25*A2*FMA21*FM22                                     FMM 0340
```

```
      TERM5=0.375*(FMA01*FM32+FMA31*FM02)                               FMM 0350
      FMM=TERM1+(TERM4+TERM5-TERM2-TERM3)/GAM4                          FMM 0360
      RETURN                                                            FMM 0370
10    FM02=FMKT(0,MBAR,A2)                                              FMM 0380
      FM12=FMKT(1,MBAR,A2)                                              FMM 0390
      FM22=FMKT(2,MBAR,A2)                                              FMM 0400
      FM32=FMKT(3,MBAR,A2)                                              FMM 0410
      GAMB4=GAMMA(MBAR)**4                                              FMM 0420
      FMM=(FMA31*FM02-FMA01*FM32+FMA11*FM22-FMA21*FM12)/(GAM4-GAMB4)    FMM 0430
999   RETURN                                                            FMM 0440
      END                                                               FMM 0450


      FUNCTION FMKT(K,M,T)                                              FMKT0020
      DIMENSION GAMMA(11)                                               FMKT0030
      COMMON GAMMA                                                      FMKT0040
      SINH(EX)=0.5*(EX-1.0/EX)                                          FMKT0050
      COSH(EX)=0.5*(EX+1.0/EX)                                          FMKT0060
      GAM=GAMMA(M)                                                      FMKT0070
      ARG=T*GAM                                                         FMKT0080
      EX=EXP(GAM)                                                       FMKT0090
      SH=(EX-1.0/EX)*0.5                                                FMKT0100
      EX=EXP(ARG)                                                       FMKT0110
      K1=K+1                                                            FMKT0120
      GO TO (1,2,3,4,1,2),K1                                            FMKT0130
1     FMKT=(SIN(ARG)-SIN(GAM)*SINH(EX)/SH)*GAM**K                       FMKT0140
      GO TO 99                                                          FMKT0150
2     FMKT=(COS(ARG)-SIN(GAM)*COSH(EX)/SH)*GAM**K                       FMKT0160
      GO TO 99                                                          FMKT0170
3     FMKT=(-SIN(ARG)-SIN(GAM)*SINH(EX)/SH)*GAM**K                      FMKT0180
      GO TO 99                                                          FMKT0190
4     FMKT=(-COS(ARG)-SIN(GAM)*COSH(EX)/SH)*GAM**K                      FMKT0200
99    RETURN                                                            FMKT0210
      END                                                               FMKT0220


      FUNCTION BFV(NU,Z)                                                BVF 0020
C***  BESSEL FUNCTION J0,J1                                              BVF 0025
      DIMENSION BF(30)                                                  BVF 0030
      IF(Z- 4.0)200,201,201                                             BVF 0040
201   BFV = SUMM(NU,Z)                                                  BVF 0050
      RETURN                                                            BVF 0060
200   IF(Z)202,205,202                                                  BVF 0070
205   IF(NU)204,204,203                                                 BVF 0080
204   BFV=1.0                                                           BVF 0090
      RETURN                                                            BVF 0100
203   BFV=0.0                                                           BVF 0110
      RETURN                                                            BVF 0120
202   M=1.4*Z+6.5                                                       BVF 0130
      ERR=.000001                                                       BVF 0140
      NCODE=1                                                           BVF 0150
60    BFV=0.0                                                           BVF 0160
      X = 2.0/ Z                                                        BVF 0170
      FLG=ALOG(.5*Z)                                                    BVF 0180
2     C=-1.0                                                            BVF 0190
50    N=NU+M                                                            BVF 0200
      BF(N+1)=1.0                                                       BVF 0210
      BF(N+2)=0.0                                                       BVF 0220
      BF(N+3)=0.0                                                       BVF 0230
      A=1.0                                                             BVF 0240
53    CR=0.0                                                            BVF 0250
      DO 6 J=1,M                                                        BVF 0260
      I=N+1-J                                                           BVF 0270
      F=I                                                               BVF 0280
      BF(I)=F*X*A+C*CR                                                  BVF 0290
      CR=A                                                              BVF 0300
6     A=BF(I)                                                           BVF 0310
      IF (NU) 7,7,8                                                     BVF 0320
7     CM=1.0                                                            BVF 0330
      CN=2.0                                                            BVF 0340
```

```
      GO TO 13
 8    CM=1.0
      DO 9 J=1,NU
      F=J
 9    CM=CM*F
 10   CN=CM*(F+2.0)
 13   GO TO (14,20),NCODE
 14   SUM=CM*BF(NU+1)+CN*BF(NU+3)
      K=NU+5
      J=2
 12   F=J
      G=NU
      CN=CN*(G+2.*F)*(G+F-1.0)/(F*(G+2.*F-2.))
      SUM=SUM+CN*BF(K)
      IF (N-K) 30,16,16
 16   J=J+1
      K=K+2
      GO TO 12
 20   SUM=CM*BF(NU+1)+CN*BF(NU+2)
      J=2
      K=NU+3
 19   F=J
      G=NU
      CN=CN*(G+F)*(2.*G+F-1.0)/(F*(G+F-1.))
      SUM=SUM+CN*BF(K)
      IF (N-K) 30,17,17
 17   J=J+1
      K=K+1
      GO TO 19
 30   F=FLG
      AM=NU
      F=AM*F
      GO TO (32,31),NCODE
 31   G=X
      GO TO 33
 32   G=0.0
 33   AMF=F+G
      AMF=EXP(AMF)
      G = AMF/SUM
      AMR=BF(NU+1)
      BF(NU+1)=AMR*G
      ER=BF(NU+1)-BFV
      ER=ABS(ER)
      BFV=BF(NU+1)
      IF (ERR-ER) 43,34,34
 43   M=M+1
      GO TO 50
 34   RETURN
      END


      FUNCTION SUMM(NU,X)
      DIMENSION GAMMA(11),FREQ(31),COEFS(24),SAVE(14)
      COMMON GAMMA,EPY,EPX,GMM,SS,T,Q,FREQ,SAVE,COEFS,BFOR,BETA,BSQ,SQK,
     1EM,CK,S,PI
      AR=4.0/X
      ARG=AR*AR
      INU=NU+1
      GO TO (1,100),INU
C     JO TO BE COMPUTED.
 1    THETA=X-PI*0.25
      I=1
      J=7
 101  PN=ARG*(ARG*(ARG*(ARG*(ARG*COEFS(I)+COEFS(I+1))+COEFS(I+2))+COEFS(
     1I+3))+COEFS(I+4))+COEFS(I+5)
      QN=AR*(ARG*(ARG*(ARG*(ARG*COEFS(J)+COEFS(J+1))+COEFS(J+2))
     1+COEFS(J+3))+COEFS(J+4))+COEFS(J+5))
      SUMM=SQRT(AR*0.5/PI)*(COS(THETA)*PN-SIN(THETA)*QN)
 999  RETURN
C     J1 TO BE COMPUTED.
 100  THETA=X-0.75*PI
      I=13
      J=19
      GO TO 101
      END
```

BIBLIOGRAPHY

1. Kobett, D. R., "Research on Panel Flutter," NASA CR-80 (1964).

2. Luke, Y. L., and A. D. St. John, "Supersonic Panel Flutter," WADC Technical Report 57-252 (1957).

3. Kobett, D. R., "Flutter of Multiple Streamwise Bay Panels at Low Supersonic Mach Number," NASA CR-538 (1966).

4. Bocher, M., "Introduction to Higher Algebra," The MacMillan Company (1936).

5. Yates, J. E., "A Study of Aeroelastic (Flutter) Behavior of Slender F Flexible Wing-Body Configurations in Hypersonic Flow," Midwest Research Institute Phase Report No. 1, Contract No. NAS1-2620 (1964).

6. "Handbook of Mathematical Functions," U. S. Department of Commerce publication (AMS55), p. 886 (1964).

7. Hansen, E. R., "On the Danilewski Method," J. Assn. Comput. Mach., January 1963.

8. Faddeeva, V. N., "Computational Methods of Linear Algebra," (Translated by C. D. Benster), Dover Publications, Inc., New York (1959).

9. Givens, W., "The Characteristic Value - Vector Problem," J. Assn. Comput. Mach., July, 1957.