2miy

NASA TECHNICAL MEMORANDUM

NASA TM X-2846

NASA TM X-2846

A COMPUTER PROGRAM FOR AUTOMATED FLUTTER SOLUTION AND MATCHED POINT DETERMINATION

by Kumar G. Bhatia

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • DECEMBER 1973

1. Report No. NASA TM X-2846	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A COMPUTER PROGRAM FOR AUTOMATED FLUTTER SOLUTION AND MATCHED-POINT DETERMINATION		5. Report Date December 1973
		6. Performing Organization Code
7. Author(s) Kumar G. Bhatia		8. Performing Organization Report No. L-9128
		10. Work Unit No. 501-22-04-01
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Author is an NRC-NASA Resident Research Associate.

16. Abstract

The use of a digital computer program (MATCH) for automated determination of the flutter velocity and the matched-point flutter density is described. The program is based on the use of the modified Laguerre iteration formula to converge to a flutter crossing or a matched-point density.

A general description of the computer program is included and the purpose of all subroutines used is stated. The input required by the program and various input options are detailed, and the output description is presented. The program can solve flutter equations formulated with up to 12 vibration modes and obtain flutter solutions for up to 10 air densities. The program usage is illustrated by a sample run, and the FORTRAN program listing is included.

17. Key Words (Suggested by Author(s)) Automated flutter solutions Matched-point determination	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	56	Domestic, \$3.50 Foreign, \$6.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

A COMPUTER PROGRAM FOR AUTOMATED FLUTTER SOLUTION AND MATCHED-POINT DETERMINATION

By Kumar G. Bhatia*
Langley Research Center

SUMMARY

The use of a digital computer program (MATCH) for automated determination of the flutter velocity and the matched-point flutter density is described. The program is based on the use of the modified Laguerre iteration formula to converge to a flutter crossing or a matched-point density.

A general description of the computer program is included and the purpose of all subroutines used is stated. The input required by the program and various input options are detailed, and the output description is presented. The program can solve flutter equations formulated with up to 12 vibration modes and obtain flutter solutions for up to 10 air densities. The program usage is illustrated by a sample run, and the FORTRAN program listing is included.

INTRODUCTION

An automated method for determining the flutter velocity and the matched-point flutter density is described in reference 1 which contains the theoretical development of the method and outlines the computational steps necessary to implement the method on a digital computer. However, reference 1 does not contain detailed information about the computer program MATCH that was developed to implement the method. The purpose of this report is to serve as a user's manual for this computer program. The basic equations used in the computer program are repeated from reference 1, and the general program organization is described. The purpose of all the subroutines used is stated, and flow diagrams for the two main subprograms are included. The program input and

---

*NRC-NASA Resident Research Associate.

output are described, and a sample run of the program is included in appendix A. The FORTRAN program listing and the Langley library subprograms used by MATCH are described in appendixes B and C, respectively.

The present report relies on reference 1, but this report contains complete information regarding the use of the computer program. It is, however, recommended that reference 1 be used in conjunction with this report for a complete understanding of the theoretical basis of the procedure implemented.

SYMBOLS

[A] nondimensional aerodynamic matrix (see eq. (4))

$[AI] = 4\pi(BR)^3 \left(\frac{SS}{BR}\right)^2 \left(\frac{1}{k}\right)^2 [A]$

$[AF] = \rho[AI]$

A_S airspeed

BR reference chord length

$F = V_f - A_S$

{G} vector of damping functions (see eq. (5))

{G1} first partial derivative of {G} with respect to $\frac{1}{k}$

{G2} second partial derivative of {G} with respect to $\frac{1}{k}$

[I] identity matrix

IOK current value of $\frac{1}{k}$ (see eq. (6))

k reduced frequency

NM	number of modes
$\{RFI\}$	vector of predicted values of $\frac{1}{k}$ corresponding to flutter crossings
$[SK]$	symmetric structural stiffness matrix
$[SM]$	symmetric structural inertia matrix
SS	semispan
$\{U_m\}$	eigenvector (see eq. (1))
V	velocity
V_f	lowest flutter velocity for an air density
$\{V_m\}$	associated eigenvector (see eq. (2))
v_f	flutter velocity
x,y	Cartesian coordinates
μ	eigenvalue (see eqs. (1) and (2))
$\xi = \sqrt{\rho_o/\rho}$	
ρ	air density
ρ_o	air density at sea level
ω	harmonic frequency

Superscripts:

R,I	denote real and imaginary parts of a complex number, respectively

T denotes a matrix transpose

Subscript:

m denotes the mode number

Subscripts following a parenthesis denote derivatives.

GENERAL DESCRIPTION OF THE COMPUTER PROGRAM

The basic equations used to implement the procedure for the flutter solution and determination of the matched-point flutter condition and the general organization of the computer program MATCH are described in this section. A matched-point flutter condition is obtained when the flutter velocity, air density, and Mach number are consistent for standard atmospheric conditions. The aerodynamic matrices are generated external to the present program and are required as input to MATCH. These matrices are calculated for a given structural configuration and a fixed Mach number. The reduced frequency range of interest is selected, and the aerodynamic matrices are evaluated at discrete values of the reduced frequency within the selected range. The Mach number is held fixed in the program, and therefore the same set of aerodynamic matrices is used.

The program can be used to obtain a flutter solution at one or more air densities or to determine a matched-point density. For a flutter solution at a specified air density, an initial value of the inverse of the reduced frequency is input to start the iteration procedure, and the program will automatically determine the velocities at which the damping becomes zero, if any, within the range of reduced frequency for which the aerodynamic matrices have been input. If a matched-point density is desired, an initial air density and an inverse of the reduced frequency are input into the program. The program determines the lowest flutter velocity for the input density. This flutter velocity will, in general, not be the same as the airspeed corresponding to the input density and the fixed Mach number. A new air density is predicted to yield the matched-point flutter condition, and the lowest flutter velocity for the predicted air density is determined for comparison with the airspeed (at the predicted density). This procedure is repeated until an air density is determined where the lowest flutter velocity is within a specified tolerance of the airspeed.

The program is dimensioned for a maximum of 12 modes and 10 air densities, that is, the structural and aerodynamic matrices can be up to (12 X 12), and flutter solutions for up to 10 air densities may be obtained during one run. The program does not provide a rigid-body mode capability, but it is possible to extend the program to include rigid-body modes. The program requires a field length of about 46 000 octal storage locations plus the field length required by the loader.

Equations Required To Implement the Procedure

The basic equations to implement the flutter solution procedure and to determine the matched-point flutter density are stated in their final form. The derivation of these equations is given in reference 1 and is not repeated here.

The characteristic flutter equation is expressed as an eigenvalue problem in matrix form by

$$\Big[[SK]^{-1}\big[[SM] + [AF]\big] - \mu_m[I]\Big]\{U_m\} = \{0\} \qquad (m = 1, \ldots, NM) \qquad (1)$$

where μ_m and $\{U_m\}$ are the complex eigenvalues and eigenvectors, respectively. The associated eigenvectors $\{V_m\}$ are determined from the following equation:

$$\Big[[SK]^{-1}\big[[SM] + [AF]^T\big] - \mu_m[I]\Big]\{V_m\} = \{0\} \qquad (m = 1, \ldots, NM) \qquad (2)$$

where

$$[AF] = \rho[AI] \qquad (3a)$$

$$[AI] = 4\pi(BR)^3\left(\frac{SS}{BR}\right)^2\left(\frac{1}{k}\right)^2[A] \qquad (3b)$$

and the elements of [A] are nondimensional. Each element A_{ij} of matrix [A] is defined by

$$A_{ij} = \frac{1}{8\pi}\iint_S h_i(x,y)\,\frac{\Delta p_j(x,y)}{(BR)\left(\frac{1}{2}\rho V^2\right)}\,\frac{dx}{(SS/BR)}\,\frac{dy}{(SS/BR)} \qquad (4)$$

where $h_i(x, y)$ is the displacement in the ith vibration mode, and $\Delta p_j(x, y)$ is the aerodynamic pressure over the lifting surface S induced by the downwash associated with simple harmonic motion in the jth vibration mode.

A flutter solution is obtained (for an assumed density ρ) when the imaginary part of one of the eigenvalues of equation (1) (or eq. (2)) is zero. A damping function G(M) is defined for each eignevalue μ_m and is given by

$$G(M) = \frac{\mu_m^I}{\mu_m^R} \qquad (m = 1, \ldots, NM; \quad M = 1, \ldots, NM) \tag{5}$$

where

$$\mu_m = \mu_m^R + \sqrt{-1}\, \mu_m^I$$

Thus, a flutter solution is obtained when one of the damping functions is zero and the corresponding frequency is real $\left(\mu_m^R > 0\right)$. Each G(M) is regarded as a function of the inverse of reduced frequency $\frac{1}{k}$.

A modified Laguerre iteration scheme is used to predict a value of $\frac{1}{k}$ for which the damping function would be zero and the slope of the curve for damping as a function of $\frac{1}{k}$ is positive. The modified Laguerre formula used to predict a zero of G(M) is

$$RFI(M) = IOK - \frac{GM}{\sqrt{[G1(M)]^2 - [G(M)]\,[G2(M)]}} \tag{6}$$

where IOK is the current value of $\frac{1}{k}$ and RFI(M) is the predicted value of $\frac{1}{k}$ corresponding to G(M) = 0. The first derivative (G1(M)) and the second derivative (G2(M)) of G(M) with respect to $\frac{1}{k}$ are evaluated from the following expressions:

$$G1(M) = \frac{\left(\mu_m^I\right)_{\frac{1}{k}} - G(M)\left(\mu_m^R\right)_{\frac{1}{k}}}{\mu_m^R} \tag{7}$$

and

$$G2(M) = \frac{\left(\mu_m^I\right)_{\frac{1}{k}\frac{1}{k}} - G(M)\left(\mu_m^R\right)_{\frac{1}{k}\frac{1}{k}} - 2G1(M)\left(\mu_m^R\right)_{\frac{1}{k}}}{\left(\mu_m^R\right)_{\frac{1}{k}}} \tag{8}$$

The expressions for $\left(\mu_m\right)_{\frac{1}{k}}$ and $\left(\mu_m\right)_{\frac{1}{k}\frac{1}{k}}$ are given by equations (9) and (10) of reference 1.

The flutter solution is determined by using equations (1) to (8), and there may be several flutter crossings where one of the damping functions is zero. The flutter solution consists of the values of $\frac{1}{k}$, and ω^2 and v_f for the flutter mode (eigenvalue for which the damping is zero) at each crossing. The lowest flutter velocity V_f thus obtained will, in general, not be consistent with the airspeed A_s for the assumed Mach number (fixed) and the assumed air density determined from the standard atmosphere (ref. 2). An iterative scheme similar to that used for the flutter solution is used to predict an air density at which the lowest flutter velocity and the relevant airspeed will be nearly the same. A function F is defined as

$$F = F(\xi) = V_f - A_s \tag{9}$$

where $\xi = \sqrt{\frac{\rho_o}{\rho}}$, and V_f and A_s are also regarded as functions of ξ. It is apparent that a zero of F will yield a matched-point density. The predicted zero of F, that is, ξ_p, is determined from the Laguerre formula

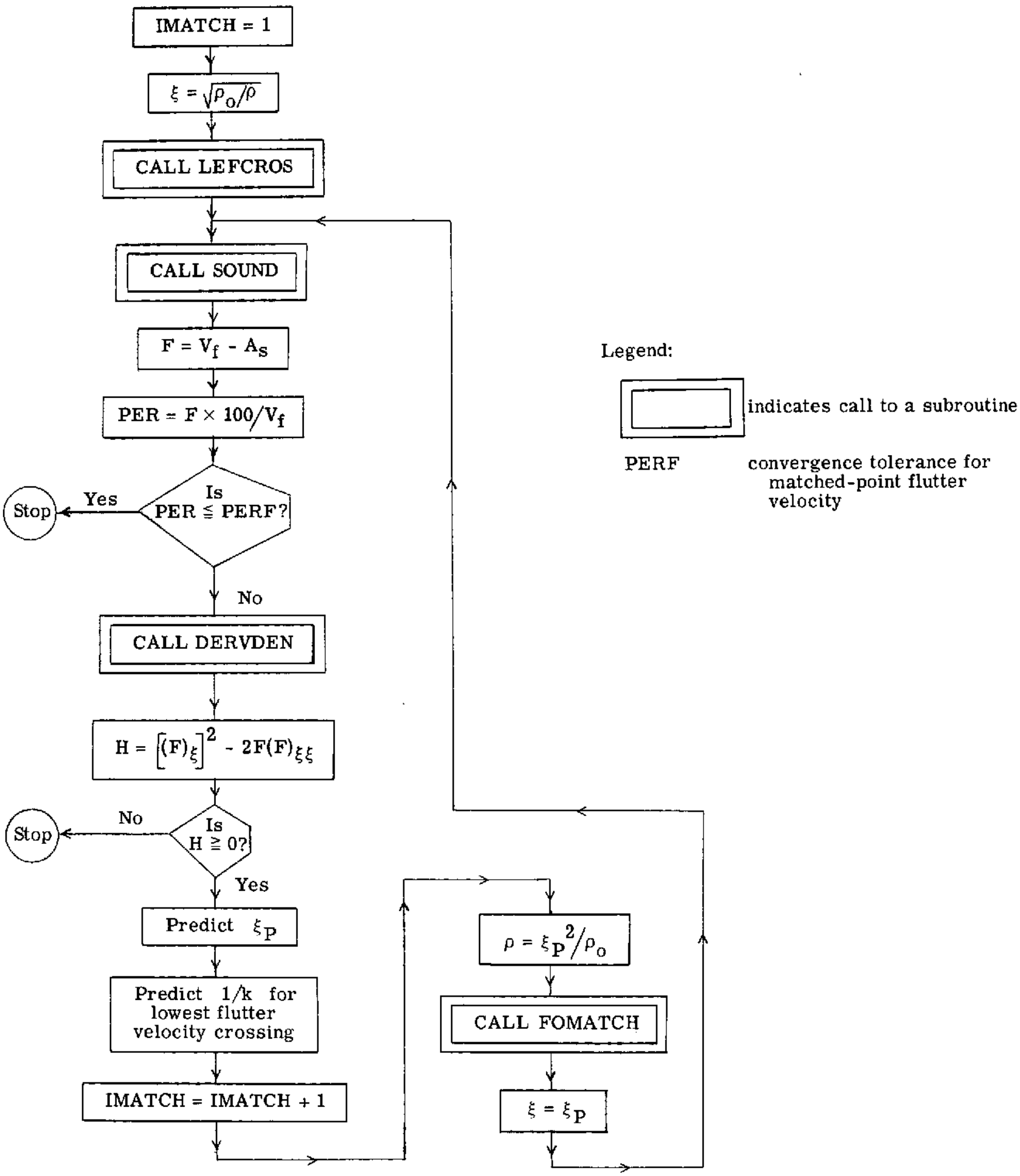
$$\xi_p = \xi - \frac{2F}{(F)_\xi + \operatorname{sgn}\left[(F)_\xi\right]\left\{\left[(F)_\xi\right]^2 - 2F(F)_{\xi\xi}\right\}^{1/2}} \tag{10}$$

where $(F)_\xi$ and $(F)_{\xi\xi}$, respectively, are evaluated by using equations (18a) to (20f) of reference 1. A flutter solution is again obtained for the air density corresponding to ξ_p, and the value of F is determined. If F is within some acceptable tolerance, then the iteration is terminated; if F is not within the tolerance, the whole procedure is repeated.

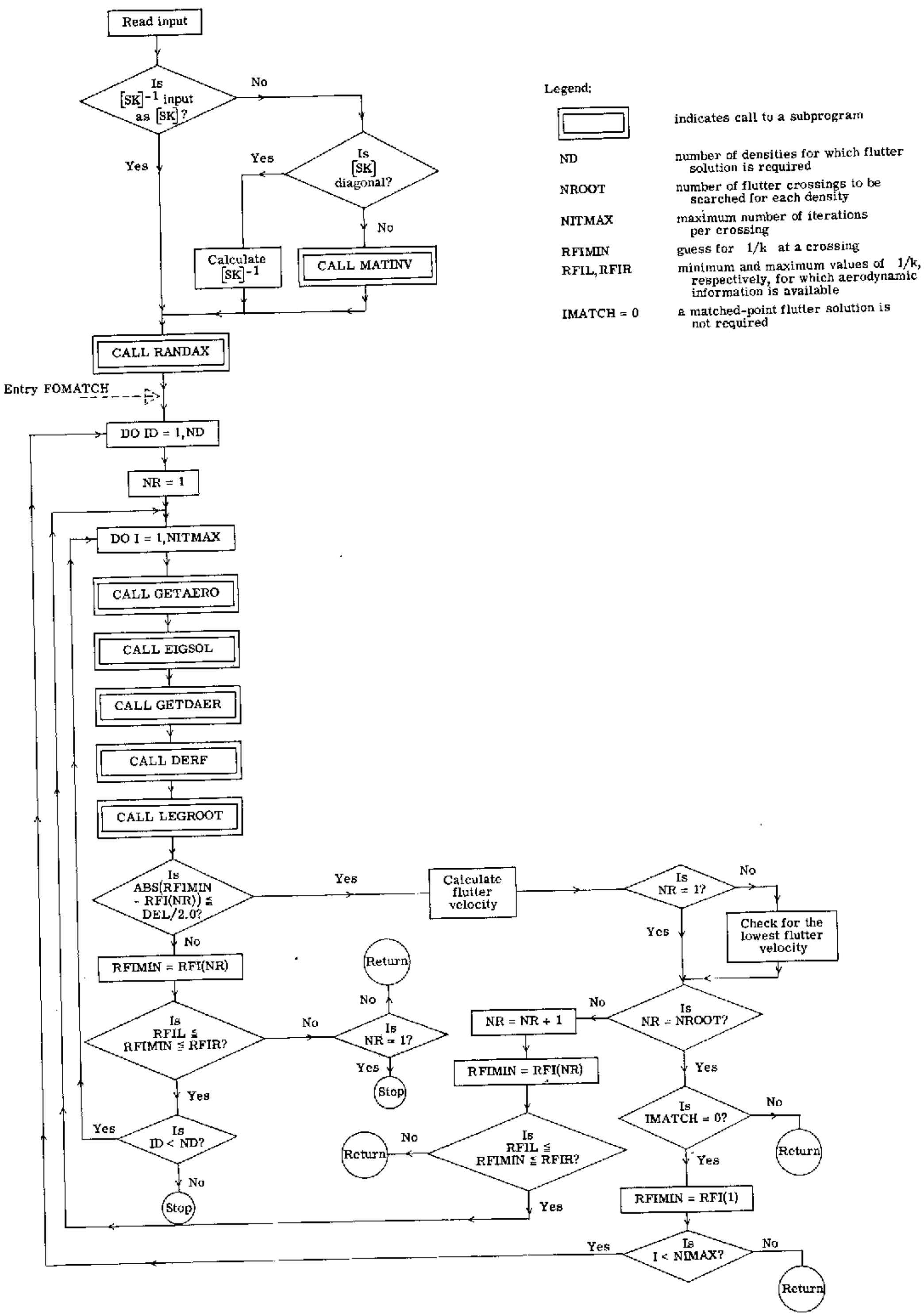
Organization of Program MATCH

The program MATCH is divided into the two major subprograms LEFCROS and CROSMAT. LEFCROS is the subprogram which controls the basic flutter solution capability, and CROSMAT controls the determination of the matched point. Both of these subprograms call various other subprograms, and since flutter solutions are required as a part of the matched-point search CROSMAT calls LEFCROS. Simplified flow diagrams of subprograms CROSMAT and LEFCROS are presented and the various subroutines called by these two subprograms are described subsequently in this section.

Simplified Flow Diagram of Subprogram CROSMAT



Simplified Flow Diagram of Subprogram LEFCROS



The aerodynamic matrices and their derivatives are stored in the program on a random-access file for easy retrieval during program execution. This aerodynamic information is furnished as input to the program by the user. The random-access file is generated in subprogram RANDAX which is called from subprogram LEFCROS. The required aerodynamic matrix and its first two derivatives are retrieved from the random-access file by calls from LEFCROS to subprogram GETAERO and entry point GETDAER. These and other subprograms called from CROSMAT and LEFCROS are briefly described.

Subprogram	Description
SOUND	Determines the airspeed and its first two derivatives with respect to ξ between geometric altitudes of -5000 meters ($\xi = 0.7964651669$) to 20 000 meters ($\xi = 3.711884976$). The airspeed is expressed as a second-order polynomial in ξ for altitudes between -5000 meters and 11 100 meters, and as a constant between 11 100 meters and 20 000 meters. This functional representation is based on data from U.S. Standard Atmosphere, 1962 (ref. 2).
DERVDEN	Evaluates the first two derivatives of the flutter velocity with respect to ξ. It also determines the first two derivatives of the reduced frequency and the flutter frequency squared with respect to ξ. It calls subprogram TMMPROD to evaluate a matrix triple product.
FOMATCH	This is an entry point in LEFCROS, and is called from CROSMAT.
MATINV	Langley library subroutine used for determining the inverse of the stiffness matrix when the matrix is not diagonal. (See appendix B.)
RANDAX	Called only once to read nondimensional aerodynamic matrices from a disk file, convert them to appropriate dimensional form, and write them on a random access file. It uses computer-system-dependent Control Data subroutines OPENMS and WRITMS at Langley Research Center. (See appendix B.)
GETAERO	Called to retrieve the aerodynamic matrices corresponding to a value of the inverse of reduced frequency from the random access file generated by subprogram RANDAX. It uses computer-system-dependent Control Data subroutine READMS. (See appendix B.)
EIGSOL	Called from LEFCROS to determine the eigenvalues, eigenvectors, and associated eigenvectors by solving equations (1) and (2). It calls Langley

Subprogram	Description
	library subroutine EECM to solve these equations (see appendix B), and subprogram TMMPROD to evaluate triple matrix products.
GETDAER	This is an entry point in GETAERO and is called from LEFCROS to retrieve the derivatives of the aerodynamic matrix from the random access file.
DERF	Called from LEFCROS to evaluate the first two derivatives of the inverse of flutter frequency squared with respect to the inverse of reduced frequency. It calls subprogram TMMPROD to evaluate the matrix triple products required.
LEGROOT	Called from LEFCROS to calculate the predicted values of the inverse of reduced frequency corresponding to zero damping crossings by using equation (6), and arranging the predicted values in ascending order. It calls subprogram DAMPAR to calculate damping from each eigenvalue by equation (5).

INPUT AND OUTPUT DESCRIPTION

Input

The input required by the computer program is described. There are two types of input to the program:

(1) Aerodynamic matrices through a disk file (tape 4)

(2) Namelist input

Description of tape 4. - All the aerodynamic matrices are in nondimensional form. These matrices must be generated by the user and provided as input to the program on a disk file (tape 4) in a format and arrangement that is compatible with the program read operations described. Tape 4 is rewound in the program and all information is read in binary.

The first read statement executed is

```
READ(4) NK, MACH, NM
```

where NK is the number of reduced frequencies for which aerodynamic matrices are on tape 4 (NK $\leq$ 1600), MACH is the Mach number at which the aerodynamic matrices have been calculated, and NM is the number of modes defining the size of the aerodynamic matrices. The next 3NK read operations are described by the following three read statements executed NK times:

READ(4) RF, X, ((A^R(L, M), L=1, NM), M=1, NM), ((A^I(L, M), L=1, NM), M=1, NM)

READ(4) ((DA^R(L, M), L=1, NM), M=1, NM), ((DA^I(L, M), L=1, NM), M=1, NM)

READ(4) ((SDA^R(L, M), L=1, NM), M=1, NM), ((SDA^I(L, M), L=1, NM), M=1, NM)

where

NM	number of modes, ≦12
RF	reduced frequency for which the six aerodynamic matrices have been calculated
X	a dummy scalar (real), not used in the program
A^R	real part of (NM X NM) aerodynamic matrix defined by equation (4)
A^I	imaginary part of (NM X NM) aerodynamic matrix defined by equation (4)
DA^R	first partial derivative of A^R with respect to reduced frequency
DA^I	first partial derivative of A^I with respect to reduced frequency
SDA^R	second partial derivative of A^R with respect to reduced frequency
SDA^I	second partial derivative of A^I with respect to reduced frequency

It is required that the aerodynamic matrices be on tape 4 for increasing values of the inverse of reduced frequency $\frac{1}{k}$ and at a constant increment of $\frac{1}{k}$. For example, if the first value of $\frac{1}{k}$ for which the aerodynamic matrices are on tape 4 is RFIL (RF_1 = 1/RFIL), the second value of $\frac{1}{k}$ must be RFI_2 = RFIL + DEL (RF_2 = 1/RFI_2), and the last value of $\frac{1}{k}$ must be RFIR = RFIL + (NK-1) DEL (RF_{Nk} = 1/RFIR) where DEL is the constant increment in $\frac{1}{k}$.

Description of namelist input. - The following two namelists are read from the input file in the order presented.

(1) NAMELIST/NAMATCH/PERF, MAXMAT, MACH, ITROPO, IMATCH REFSLD, UNITL

(2) NAMELIST/NAM1/SK, SM, LSTIFF, SS, BR, NM, RFIL, RFIR, DEL, NROOT, NITMAX, ND, RHO, RFIMIN, IPRT, IOPT

The dimensional parameters in the namelist statements determine the force (for example, newtons, pounds, etc.) and length (meters, feet, etc.) units with which the program operates; the unit of time used is seconds. The user must therefore prepare the namelist input to be consistent with any desired force and length units. The definitions of the various namelist input parameters in NAMATCH are

PERF — Nondimensional convergence tolerance for matched-point flutter solution. The program will terminate when $\left|1 - \frac{\text{Airspeed}}{\text{Lowest flutter velocity}}\right| \times 100 \leq \text{PERF}$. Not required if IMATCH = 0.

MAXMAT — Maximum number of iterations permitted for the matched-point density search. Not required if IMATCH = 0.

MACH — Mach number (real variable) for which the aerodynamic matrices have been calculated. Not required if IMATCH = 0.

ITROPO — Defines the initial air density for the matched-point search if
- = 0, initial density = sea-level density
- = 1, initial density = density at altitude of 11 100 meters
- = -1, initial density = RHO(1) from input for namelist NAM1.

Not required if IMATCH = 0.

IMATCH — If IMATCH = 0, flutter solutions for densities in namelist NAM1 are required. If IMATCH ≠ 0, a matched-point flutter solution is required.

REFSLD — Reference sea-level density in $\frac{\text{Force-sec}^2}{(\text{Length})^4}$ units. Not required if IMATCH = 0.

UNITL — Ratio of the number of length units selected to 1 foot. Not required if the length units selected are feet.

The definitions of the various namelist input parameters in NAM1 are

SK — Symmetric structural stiffness matrix $\left(\frac{\text{Force}}{\text{Length}} \text{ units}\right)$ or symmetric structural flexibility matrix $\left(\frac{\text{Length}}{\text{Force}} \text{ units}\right)$, (NM × NM).

SM — Symmetric structural inertia matrix $\left(\frac{\text{Force-sec}^2}{\text{Length}}\right)$, (NM × NM).

LSTIFF — If = 0, SK is diagonal stiffness matrix.
If = +1, SK is nondiagonal symmetric stiffness matrix.
If = -1, SK is flexibility matrix, and may or may not be diagonal.

SS	Semispan (or reference length) used to generate the aerodynamic matrices (Length units).
BR	Reference chord used to generate the aerodynamic matrices (Length units).
NM	Number of vibration modes used to generate aerodynamic matrices. Maximum value of NM is 12.
RFIL	Inverse of reduced frequency (nondimensional) corresponding to first value of reduced frequency for which aerodynamic matrices are on tape 4.
RFIR	Inverse of reduced frequency (nondimensional) corresponding to last value of reduced frequency for which aerodynamic matrices are on tape 4.
DEL	Constant increment of the inverse of reduced frequency (nondimensional) at which the aerodynamic matrices are on tape 4, for example, the first set of matrices are for $\frac{1}{k}$ = RFIL, the second set for $\frac{1}{k}$ = RFIL + DEL, etc.
NROOT	Number of flutter crossings desired. If the program cannot determine all the NROOT crossings within the selected range of RFIL to RFIR, it will continue with execution of the next task, if any.
NITMAX	Maximum number of iterations per crossing allowed for convergence. If a crossing cannot be determined in NITMAX iteration, the execution will be terminated. Suggested value 5.
ND	Number of densities for which a flutter solution is required ($1 \leq$ ND ≤ 10). If IMATCH $\neq$ 0 in namelist NAMATCH, then ND should be input as 1.
RHO	One-dimensional array of input densities $\left(\frac{\text{Force-(Second)}^2}{\text{(Length)}^4}\text{ units}\right)$ for which flutter solutions are required. If IMATCH $\neq 0$ and ITROPO = -1 in namelist NAMATCH, then RHO(1) is the initial density for the matched-point density search. If IMATCH $\neq$ 0 and ITROPO $\neq$ -1, no input is required for RHO.
RFIMIN	Initial guess for inverse of reduced frequency to start search for first flutter crossing. Experience with the program indicates that the convergence from a value of RFIMIN which is higher than the inverse of reduced frequency for the (actual) first crossing is faster than that from a RFIMIN which is lower.
IPRT	Determines amount of output printed by program. It is nominally set to zero within the program, and if nominal output is required, then it can be omitted

from the namelist input. This procedure will be discussed further when the program output is described.

IOPT Unused parameter, not required.

Output

The program output is described in this section. The program output consists of two categories:

(1) An output summary on a coded (BCD) disk file (tape 7) which may be routed for printing.

(2) Output file containing either a nominal printout (IPRT = 0 specified by input) or a detailed printout (IPRT = 1 or 2).

The first category of the output is described first and is followed by the second category. The output is in all cases in units consistent with those used for the program input.

The output summary on tape 7 includes the following:

(1) Air density and the initial value of $\frac{1}{k}$ for each air density at which a flutter solution is determined.

(2) Root number, flutter velocity, the inverse of reduced frequency, and the total number of iterations required for each flutter crossing determined.

(3) Iteration number, air density, square root of sea-level density/air density, lowest flutter velocity, airspeed, and $\left(1 - \frac{\text{Airspeed}}{\text{Lowest flutter velocity}}\right) \times 100$ for each matched-point iteration, if matched-point density search is executed.

(4) Informative messages:

(a) "FOUND NR ROOTS, RFI FOR THE NEXT ROOT PREDICTED = X, IS BEYOND RANGE." This message is printed when the predicted inverse of reduced frequency (X) for the NRth crossing (NR $\leq$ NROOT) is not within the range RFIL to RFIR.

(b) "RFI PREDICTED FOR THE NEXT ROOT = . . ., DIFFERENCE FROM RFI FOR PREVIOUS ROOT LESS THAN DEL/2.0." This message is printed to inform the user that the next flutter crossing is within DEL/2.0 of $\frac{1}{k}$ for the previous flutter crossing; therefore, the next crossing is taken to be at the same value of $\frac{1}{k}$ as the previous flutter crossing.

(5) Various messages explaining abnormal termination:

(a) "MATCH-POINT ITERATION DID NOT CONVERGE IN MAXMAT ITERATIONS."

(b) "ARGUMENT OF RADICAL IN LAGUERRE = X, ITERATION NO. = . . ., DENSITY = . . ., VEL = . . ., SPEED OF SOUND * MACH = . . ." This message is printed out when X is negative during a matched-point density search.

(c) "ARGUMENT OF RADICAL IN LAGUERRE ITERATE IS NEGATIVE FOR NM MODES." This message is printed out when a real value for predicted inverse of reduced frequency for a flutter crossing (from eq. (6)) cannot be obtained for any of the NM modes.

(d) "RFI = . . ., IS OUTSIDE THE RANGE OF VALUES." This message is printed when the initial value of the inverse of reduced frequency input in the program is outside the range RFIL to RFIR.

(e) "PROGRAM TERMINATED, COULD NOT FIND ROOT NO. NR IN NITMAX ITERATIONS."

(f) "NUMBER OF EIGENVALUES COMPUTED M." This message is printed out from subprogram EIGSOL when during eigensolution, convergence is obtained for only $M < NM$ eigenvalues.

The second category of the output depends on the value for IPRT (0, 1 or 2). In all cases, the printout described for tape 7 is included. The output for IPRT = 0 is described by stating the additional output relative to printout on tape 7, output for IPRT = 1 is described by stating the additional output relative to IPRT = 0, and the output for IPRT = 2 is similarly described.

For IPRT = 0, the following information is printed in addition to the information written on tape 7:

(1) Printout of the two namelists.

(2) Eigenvalues and the predicted values of the inverse of reduced frequency (RFI) in increasing order of magnitude, for flutter crossings at each iteration. RFI = 1000.0000 indicates that a real value for the inverse of reduced frequency corresponding to a flutter crossing could not be predicted for that mode. RFI = 3000.0000 indicates that the real part of the eigenvalue corresponding to that mode was negative.

(3) Flutter eigenvalue number, eigenvalues, correspondence of the predicted crossings (which are arranged in increasing order of magnitude) to eigenvalues (which are obtained (and printed) in the decreasing order of their absolute values from the eigensolution), root number, flutter velocity, the inverse of reduced frequency at the crossing, and the total number of iterations required for convergence for each flutter crossing determined. The number of iterations for convergence include the last iteration for the convergence check.

(4) The inverse of reduced frequency, the first and second derivatives of reduced frequency, flutter frequency squared, and flutter velocity with respect to ξ , for each match-point iteration.

(5) Predicted values for ξ and the inverse of reduced frequency for a matched-point solution for each match-point iteration.

For IPRT = 1, damping, the first and second derivatives of damping with respect to $\frac{1}{k}$, argument of the square root in equation (6), and the predicted crossing are printed for each mode during every iteration for a flutter solution. If IPRT = 2, the eigenvectors and associated eigenvectors, and the eigenvalue derivatives are printed during every iteration for a flutter solution.

CONCLUDING REMARKS

A digital computer program MATCH for automated determination of the flutter velocity and the matched-point flutter density has been described. The program was based on the use of the modified Laguerre iteration formula to converge to a flutter crossing or a matched-point density.

A general description of the computer program and the related subroutines has been included. Detailed descriptions of the output, input, and input options have been presented. The program can solve flutter equations formulated with up to 12 vibration modes and can obtain flutter solutions for up to 10 air densities. Use of the program is illustrated with a sample run and the FORTRAN listing is included.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., September 10, 1973.

APPENDIX A

SAMPLE RUN OF PROGRAM MATCH

The input and output for a sample program run are presented in this appendix in order to illustrate the application of the program. The units used in this sample run are pounds and inches; the program dictates use of second as the unit for time.

This sample run is for the all-movable control surface example of reference 1. The flutter equation is formulated with five vibration modes and the aerodynamic matrices have been calculated for a Mach number of 0.6. A matched-point flutter solution is required, and the initial values of air density and the inverse of reduced frequency are $1.146797839 \times 10^{-7} \frac{\text{lb-sec}^2}{\text{in}^4}$ (sea-level density from ref. 2) and 6.5, respectively. A detailed output is desired, and IPRT = 2 is input. The namelist input for a sample run follows.

NAMELIST INPUT FOR SAMPLE RUN

PROGRAM NO. ________
CODED BY ________
DIVISION ________ SECTION ________

LANGLEY RESEARCH CENTER
FORTRAN - DATA CODING FORM

DATE ________
PAGE ____ OF ____
JOB ORDER ________ TASK NO. ________

STATEMENT NUMBER | CONTINUATION | FORTRAN STATEMENT | IDENTIFICATION AND SEQUENCING

```
 $NAMATCH  PERF=1.0,MAXMAT=3,MACH=0.6,ITRØPØ=0,IMATCH=1,REFSLD=1.146797839E-07,

          UNITL=12.0$


 $NAM1    SK(1)=4.049E01,12*0.0,1.016E03,12*0.0,6.2E03,12*0.0,5.521E03,12*0.0,

             2.165E04,

          SM(1)=7.631E-03,12*0.0,4.741E-03,12*0.0,2.49E-03,12*0.0,7.562E-04,12*0.0,

             1.933E-03,

          LSTIFF=0,SS=16.0,BR=6.5,NM=5,RFIL=1.0,RFIR=20.0,DEL=0.05,NRØØT=3,

          NITMAX=5,ND=1,RHØ(1)=1.146797839E-07,RFIMIN=6.5,IPRT=2$
```

NASA-Langley Form 67 (MAR 69)

NOTE: WRITE NUMBERS 10, LETTERS I Ø U G Z C, SYMBOLS / . , ' Note that namelist NAM1 does require an input for air density since ITROPO = -1 in namelist NAMATCH.

The program output is in two parts: (a) summary on tape 7 and (b) output file. The output obtained from the sample run follows.

(a) Listing of tape 7

```
DENSITY =     1.146798E-07 , RFIMIN =         6.5000

ROOT NUMBER    1 , VELOCITY =  6316.702 , RFI =    10.800 , NO. OF ITERATIONS REQD, =  3

ROOT NUMBER    2 , VELOCITY = 29430.954 , RFI =    11.350 , NO. OF ITERATIONS REQD, =  1
FOUND  2 ROOTS, RFI FOR THE NEXT ROOT PREDICTED =     22.8007 ,IS BEYOND THE RANGE

ITERATION NO.  1 DENSITY =    1.1468E-07 SQRT(SEA LEVEL DENSITY/DENSITY) =    1.0000E+00
 FLUTTER VEL =    6.3167E+03 AIR SPEED =    8.0361E+03 (VEL-AIRSPEED)*100/VEL = -27.2204


DENSITY =     7.806131E-08 , RFIMIN =        13.1467

ROOT NUMBER    1 , VELOCITY =  7650.904 , RFI =    13.150 , NO. OF ITERATIONS REQD, =  1

ROOT NUMBER    2 , VELOCITY = 35709.560 , RFI =    13.800 , NO. OF ITERATIONS REQD, =  1
FOUND  2 ROOTS, RFI FOR THE NEXT ROOT PREDICTED =     27.6403 ,IS BEYOND THE RANGE

ITERATION NO.  2 DENSITY =    7.8061E-08 SQRT(SEA LEVEL DENSITY/DENSITY) =    1.2121E+00
 FLUTTER VEL =    7.6509E+03 AIR SPEED =    7.6504E+03 (VEL-AIRSPEED)*100/VEL =     .0065
```

(b) Listing of output file

```
$NAMATCH

PERF     =  0.1E+01,

MAXMAT   =  3,

MACH     =  0.6E+00,

ITROPO   = -1,

IMATCH   =  1,

REFSLD   =  0.1146797839E-06,

UNITL    =  0.12E+02,

$END




$NAM1

SK       =  0.4049E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.1016E+03,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.62E+03,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.5521E+03,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.2165E+04,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,

SM       =  0.7631E-02,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.4741E-02,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.249E-02,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.7562E-03,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.1933E-02,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
           I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,

LSTIFF   =  1,

SS       =  0.16E+02,

BR       =  0.65E+01,

NM       =  5,

REIL     =  0.1E+01,

REIR     =  0.2E+02,
```

```
DEL     =  0.5E-01,
NROOT   =  3,
NITMAX  =  5,
ND      =  1,
RHO     =  0.1146797839E-06, I, I, I, I, I, I, I, I, I,
RFIMIN  =  0.65E+01,
IPRT    =  2,
IOPT    =  0,
$END
```

```
DENSITY =     1.146798E-07 , REIMIN =        6.5000

EIGENVALUES
     1.814E-03  -1.174E-04     7.226E-05  -2.578E-06     4.607E-06  -1.013E-07     2.170E-06  -2.388E-08     9.083E-07  -3.991E-08


EIGENVECTORS
     9.941E-01   1.068E-01    -2.113E-02  -3.943E-02    -1.858E-02   7.782E-02    -1.012E-01   1.203E-01     5.704E-03  -6.108E-03
     1.757E-02  -2.174E-03     9.979E-01   2.985E-02     4.052E-03   2.246E-01    -5.166E-02   8.887E-02     9.186E-02  -1.200E-01
     1.530E-03   3.048E-04    -3.409E-02  -2.620E-03     1.179E-02   9.657E-01     2.288E-01  -3.272E-01    -1.442E-02   3.318E-02
    -2.434E-04  -1.799E-05     6.669E-03   1.006E-03    -4.244E-03   1.020E-01    -4.899E-01   7.513E-01    -3.822E-04   5.495E-04
    -3.290E-04  -5.137E-05     7.155E-03   2.551E-04    -5.493E-03  -3.169E-02     1.374E-02  -2.593E-02    -5.253E-01   8.365E-01


ASSOCIATED EIGENVECTORS
     1.359E+02  -5.660E+00     6.126E+01   1.476E+00    -7.246E+00   4.719E+01    -1.553E+01  -2.226E+01    -2.164E+01  -4.550E+01
     1.247E-01   2.024E-01     1.355E+02   8.425E-01     4.253E+00  -6.595E+01     2.209E+01   4.470E+01     3.338E+01   6.238E+01
    -5.730E-02  -1.883E-02    -4.600E+00   2.209E-03     1.049E+01  -3.436E+02     4.056E+01   7.124E+01    -3.390E+01  -4.145E+01
    -2.049E-01  -2.819E-02    -5.099E+00   4.345E-02     1.090E+01  -1.698E+02    -4.811E+02  -7.497E+02    -5.459E+01  -7.655E+01
     3.544E-03   1.960E-03     1.017E+00   8.877E-02     5.029E-01   3.141E+00    -3.277E-01  -2.077E-01    -2.535E+02  -4.463E+02


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -2.384E-05   -1.815E-05
SECOND DERIVATIVE
  -3.889E-06  -2.214E-07=     7.335E-06   5.585E-06+   -1.647E-07  -1.500E-07+    -1.106E-05  -5.656E-06


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       8.361E-06   -8.565E-08
SECOND DERIVATIVE
   1.503E-06   2.107E-07=    -2.573E-06   2.635E-08+    1.925E-07   1.493E-07+     3.883E-06   3.510E-08


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       1.682E-07   -1.989E-09
SECOND DERIVATIVE
   3.637E-08   6.828E-09=    -5.175E-08   6.119E-10+    9.853E-09   4.481E-09+     7.826E-08   1.735E-09


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE       2.029E-07   -7.093E-09
SECOND DERIVATIVE
  -1.305E-09  -3.590E-09=    -6.243E-08   2.182E-09+   -3.242E-08  -3.042E-09+     9.355E-08  -2.731E-09


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -9.752E-09   -7.242E-09
SECOND DERIVATIVE
  -6.749E-09  -8.536E-10=     3.001E-09   2.228E-09+   -5.247E-09  -6.799E-10+    -4.503E-09  -2.402E-09

SUBROUTINE LEGROOT
      DAMPING            FIRST DERIV        SECOND DERIV      RADICAL IN LEGUERRE    EIGENVALUE NO.    PROJECTED CROSSING
   -1.1006E-02         -2.2397E-03          -1.2423E-03           -8.6567E-06             4               1.0000E+03
   -2.1978E-02          3.7063E-04           1.6283E-03            3.5923E-05             3               1.0167E+01
   -3.5673E-02          2.9424E-03           2.9769E-03            1.1485E-04             2               9.8287E+00
   -4.3937E-02         -8.4452E-03          -1.4476E-03            7.7175E-06             5               2.2316E+01
   -6.4711E-02         -1.0859E-02          -5.4637E-04            8.2564E-05             1               1.3622E+01


ITERATION  1
     6.5000             9.8287      10.1669      13.6217      22.3160    1000.0003
```

```
EIGENVALUES
     1.711E-03  -1.798E-04     1.093E-04  -1.245E-06     5.524E-06  -5.602E-08     2.698E-06  -7.886E-08     8.356E-07  -6.999E-08


EIGENVECTORS
     9.758E-01   2.142E-01    -6.039E-02  -6.409E-02    -1.923E-02   2.389E-01     2.736E-02   1.873E-01    -4.170E-03  -2.716E-03
    -4.246E-02  -1.257E-02     9.930E-01  -5.561E-02     1.284E-02   4.600E-01     1.566E-02   3.245E-02     1.133E-01  -2.201E-01
     3.682E-03   1.280E-03    -5.386E-02  -2.649E-04     7.485E-02   8.053E-01    -9.596E-02  -7.555E-01     5.776E-02  -6.003E-02
    -5.892E-04  -1.564E-04     1.050E-02   4.330E-04     7.344E-03   2.737E-01     1.199E-01   6.070E-01    -3.436E-02   5.270E-02
    -7.939E-04  -2.520E-04     1.145E-02  -2.518E-04    -7.300E-03  -4.086E-02     5.636E-03  -6.979E-03    -3.687E-01   8.899E-01


ASSOCIATED EIGENVECTORS
     1.432E+02  -1.554E+01     1.012E+02   1.464E+01    -1.070E+01   4.980E+01     1.360E+01  -6.253E+01    -1.807E+01  -8.209E+01
     3.467E-01   3.427E-01     8.890E+01   6.152E+00     5.314E+00  -6.276E+01    -3.273E+01   1.096E+02     2.869E+01   1.188E+02
    -1.484E-01  -2.961E-02    -5.604E+00  -6.233E-01     1.577E+01  -2.510E+02    -5.560E+01   2.195E+02    -3.578E+01  -6.738E+01
    -5.287E-01  -3.823E-02    -8.799E+00  -1.207E+00     4.578E+01  -3.440E+02     1.427E+02  -7.609E+02    -2.619E+01  -1.247E+02
     9.317E-03   3.298E-03     1.079E+00   1.430E-01     9.543E-01  -9.024E-01    -8.435E+00   1.786E+01    -1.740E+02  -5.431E+02


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -3.745E-05    -1.930E-05
SECOND DERIVATIVE
  -4.298E-06  -5.097E-07=    7.605E-06   3.919E-06+   -4.560E-07  -3.804E-07+   -1.145E-05  -4.049E-06


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       1.399E-05     1.042E-06
SECOND DERIVATIVE
   1.907E-06   5.044E-07=   -2.841E-06  -2.116E-07+    4.678E-07   3.775E-07+    4.280E-06   3.385E-07


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       4.228E-07     3.084E-08
SECOND DERIVATIVE
   1.090E-07   9.498E-09=   -8.585E-08  -6.261E-09+    6.570E-08   5.722E-09+    1.291E-07   1.004E-08


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE       7.324E-08    -2.603E-08
SECOND DERIVATIVE
  -6.581E-08  -3.576E-09=   -1.487E-08   5.286E-09+   -7.316E-08  -1.743E-09+    2.222E-08  -7.119E-09


DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -3.419E-08    -1.103E-08
SECOND DERIVATIVE
  -7.834E-09  -1.456E-09=    6.943E-09   2.240E-09+   -4.328E-09  -1.049E-09+   -1.045E-08  -2.647E-09

SUBROUTINE LEGROOT
    DAMPING             FIRST DERIV         SECOND DERIV      RADICAL IN LEGUERRE     EIGENVALUE NO.    PROJECTED CROSSING
  -1.0141E-02          6.3583E-03          9.4621E-04            5.0024E-05                3                1.1284E+01
  -1.1390E-02          1.0989E-02          1.9996E-03            1.4354E-04                2                1.0801E+01
  -2.9233E-02         -8.8569E-03         -1.5577E-03            3.2907E-05                4                1.4946E+01
  -8.3760E-02         -1.6632E-02         -3.8888E-03           -4.9108E-05                5                1.0000E+03
  -1.0509E-01         -1.3580E-02         -1.1563E-03            6.2920E-05                1                2.3098E+01


ITERATION  2
     9.8500             10.8007     11.2838     14.9460     23.0979   1000.0000
```

```
EIGENVALUES
    1.674E-03  -1.984E-04     1.235E-04  -6.779E-09     5.976E-06  -2.273E-08     2.737E-06  -1.047E-07     7.995E-07  -8.117E-08

EIGENVECTORS
    9.718E-01   2.291E-01    -6.531E-02  -8.207E-02     2.867E-01  -8.232E-02     1.605E-01  -8.086E-02    -7.911E-03  -3.673E-03
   -5.209E-02  -1.729E-02     9.908E-01   6.105E-02     4.926E-01  -1.877E-01    -5.950E-03  -9.165E-03     2.411E-01  -1.266E-01
    4.516E-03   1.714E-03    -5.764E-02  -7.158E-03     6.596E-01  -3.135E-01    -7.417E-01   3.443E-01     1.401E-01  -3.883E-02
   -7.234E-04  -2.206E-04     1.117E-02   1.710E-03     2.909E-01  -1.161E-01     4.798E-01  -2.620E-01    -8.150E-02   3.318E-02
   -9.739E-04  -3.422E-04     1.233E-02   1.225E-03    -3.507E-02   1.986E-02    -5.322E-03  -7.259E-03    -8.000E-01   5.069E-01

ASSOCIATED EIGENVECTORS
    1.464E+02  -1.598E+01     1.125E+02   3.794E+00    -4.074E+01  -2.557E+01     6.662E+01   3.822E+01    -6.933E+01  -6.625E+01
    4.349E-01   4.015E-01     7.873E+01  -4.622E+00     5.109E+01   2.238E+01    -1.134E+02  -7.557E+01     1.034E+02   9.590E+01
   -1.860E-01  -3.649E-02    -5.698E+00   4.803E-02     2.046E+02   8.561E+01    -2.355E+02  -1.437E+02    -7.284E+01  -3.611E+01
   -6.634E-01  -5.222E-02    -9.779E+00  -2.072E-01     3.336E+02   1.663E+02     6.607E+02   3.509E+02    -9.850E+01  -1.016E+02
    1.171E-02   3.999E-03     1.050E+00  -2.366E-03     1.098E+00   1.291E+00    -2.345E+01  -2.115E+01    -4.767E+02  -3.732E+02

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -4.162E-05   -1.986E-05
SECOND DERIVATIVE
  -4.470E-06  -6.631E-07=    7.707E-06   3.677E-06+   -5.866E-07  -5.033E-07+   -1.159E-05  -3.837E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       1.588E-05    1.589E-06
SECOND DERIVATIVE
   2.082E-06   6.554E-07=   -2.941E-06  -2.943E-07+    5.927E-07   4.993E-07+    4.431E-06   4.504E-07

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       5.287E-07    3.896E-08
SECOND DERIVATIVE
   1.125E-07   7.736E-09=   -9.790E-08  -7.214E-09+    6.320E-08   3.878E-09+    1.472E-07   1.107E-08

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE       1.090E-08   -2.781E-08
SECOND DERIVATIVE
  -6.387E-08  -2.579E-10=   -2.018E-09   5.150E-09+   -6.482E-08   1.420E-09+    2.969E-09  -6.828E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -4.188E-08   -1.256E-08
SECOND DERIVATIVE
  -8.379E-09  -1.772E-09=    7.755E-09   2.325E-09+   -4.463E-09  -1.298E-09+   -1.167E-08  -2.800E-09

SUBROUTINE LEGROOT
     DAMPING              FIRST DERIV          SECOND DERIV      RADICAL IN LEGUERRE     EIGENVALUE NO.     PROJECTED CROSSING
   -5.4884E-05            1.2874E-02            1.9962E-03            1.6586E-04               2                 1.0804E+01
   -3.8043E-03            6.8553E-03            1.5321E-04            4.7579E-05               3                 1.1352E+01
   -3.8242E-02           -1.0006E-02           -9.0677E-04            6.5447E-05               4                 1.5527E+01
   -1.0153E-01           -2.1024E-02           -5.4834E-03           -1.1472E-04               5                 1.0000E+03
   -1.1855E-01           -1.4812E-02           -1.4495E-03            4.7553E-05               1                 2.7991E+01

ITERATION  3
    10.8000             10.8043      11.3515      15.5271      27.9913   1000.0000

FLUTTER EIGENVALUE NO. =    2, EIGENVALUES
 1.6737E-03 -1.9841E-04  1.2351E-04 -6.7786E-09  5.9759E-06 -2.2734E-08  2.7375E-06 -1.0469E-07  7.9949E-07 -8.1170E-08

REI FOR PREDICTED CROSSINGS CORRESPOND TO EIGENVALUES NUMBERS
   2    3    4    1    5

ROOT NUMBER   1 , VELOCITY =  6316.702 , REI =   10.800 , NO. OF ITERATIONS REQD, =  3
```

```
EIGENVALUES
     1.650E-03  -2.094E-04     1.326E-04   9.718E-07     6.284E-06  -1.679E-10     2.734E-06  -1.199E-07     7.752E-07  -8.836E-08

EIGENVECTORS
     9.657E-01   2.521E-01    -8.335E-02  -8.037E-02     3.068E-01   1.249E-01     1.679E-01   4.338E-02    -1.125E-02  -5.360E-03
    -5.795E-02  -2.148E-02     9.899E-01  -5.282E-02     5.349E-01   1.736E-01    -1.714E-02  -1.575E-02     2.761E-01  -7.365E-02
     5.019E-03   2.092E-03    -6.026E-02  -5.455E-04     6.640E-01   1.531E-01    -8.060E-01  -2.407E-01     1.872E-01  -1.356E-02
    -8.054E-04  -2.781E-04     1.170E-02   3.869E-04     3.153E-01   9.380E-02     4.983E-01   1.157E-01    -1.025E-01   1.726E-02
    -1.083E-03  -4.216E-04     1.288E-02  -1.850E-04    -3.835E-02  -6.209E-03     2.648E-03  -9.817E-03    -8.799E-01   3.130E-01

ASSOCIATED EIGENVECTORS
     1.483E+02  -1.835E+01     1.170E+02   1.803E+01    -4.632E+01   5.042E+00     8.215E+01  -1.725E+01    -8.843E+01  -5.473E+01
     5.000E-01   4.338E-01     7.340E+01   3.652E+00     4.986E+01  -1.371E+01    -1.470E+02   2.124E+01     1.319E+02   7.951E+01
    -2.116E-01  -3.844E-02    -5.675E+00  -5.943E-01     1.985E+02  -5.765E+01    -2.989E+02   5.484E+01    -8.196E+01  -1.899E+01
    -7.539E-01  -5.232E-02    -1.017E+01  -1.389E+00     3.729E+02  -8.327E+01     7.134E+02  -1.739E+02    -1.249E+02  -8.667E+01
     1.336E-02   4.303E-03     1.016E+00   1.044E-01     1.796E+00   1.903E-01    -4.035E+01  -6.975E-02    -5.688E+02  -2.767E+02

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -4.411E-05   -2.025E-05
SECOND DERIVATIVE
  -4.586E-06  -7.767E-07=    7.772E-06   3.569E-06+    -6.755E-07  -5.953E-07+    -1.168E-05  -3.750E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       1.706E-05    3.979E-06
SECOND DERIVATIVE
   2.201E-06   7.671E-07=   -3.006E-06  -3.488E-07+     6.782E-07   5.906E-07+     4.529E-06   5.253E-07

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       5.904E-07    4.306E-08
SECOND DERIVATIVE
   1.117E-07   7.274E-09=   -1.040E-07  -7.588E-09+     5.932E-08   3.427E-09+     1.564E-07   1.143E-08

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE      -2.309E-08   -2.752E-08
SECOND DERIVATIVE
  -5.947E-08   1.222E-09=    4.068E-09   4.850E-09+    -5.739E-08   2.717E-09+    -6.144E-09  -6.345E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -4.660E-08   -1.360E-08
SECOND DERIVATIVE
  -8.827E-09  -2.020E-09=    8.212E-09   2.396E-09+    -4.681E-09  -1.500E-09+    -1.236E-08  -2.916E-09

SUBROUTINE LEGROOT
      DAMPING            FIRST DERIV          SECOND DERIV      RADICAL IN LEGUERRE    EIGENVALUE NO.    PROJECTED CROSSING
     7.3308E-03           1.3989E-02            2.0645E-03            1.8055E-04               2                 1.0804E+01
    -2.6727E-05           6.8553E-03           -1.3013E-04            4.6992E-05               3                 1.1354E+01
    -4.3870E-02          -1.0438E-02           -6.8352E-04            7.8964E-05               4                 1.6287E+01
    -1.1398E-01          -2.4393E-02           -6.8370E-03           -1.8426E-04               5                 1.0000E+03
    -1.2693E-01          -1.5665E-02           -1.6609E-03            3.4592E-05               1                 3.2931E+01

ITERATION  1
    11.3500            10.8044      11.3539      16.2869      32.9306   1000.0000

FLUTTER EIGENVALUE NO. =    3, EIGENVALUES
 1.6501E-03 -2.0944E-04  1.3256E-04  9.7180E-07  6.2836E-06 -1.6794E-10  2.7340E-06 -1.1994E-07  7.7517E-07 -8.8356E-08

RFI FOR PREDICTED CROSSINGS CORRESPOND TO EIGENVALUES NUMBERS
   2    3    4    1    5

ROOT NUMBER   2 , VELOCITY = 29430.954 , RFI =   11.350 , NO. OF ITERATIONS REQD, =  1
```

EIGENVALUES

1.370E-03	-3.269E-04	2.498E-04	2.787E-05	1.054E-05	3.191E-07	2.090E-06	-2.109E-07	4.017E-07	-1.969E-07

EIGENVECTORS

-7.385E-01	-6.544E-01	1.431E-01	2.277E-01	5.162E-01	8.459E-02	-7.224E-02	1.166E-01	4.420E-02	-5.061E-02
9.177E-02	1.334E-01	-9.352E-01	-2.180E-01	7.277E-01	1.068E-01	6.799E-02	-9.898E-02	-1.688E-01	2.886E-01
-7.764E-03	-1.187E-02	6.927E-02	2.070E-02	3.098E-01	1.456E-03	5.167E-01	-7.577E-01	-3.841E-01	5.381E-01
1.291E-03	1.849E-03	-1.318E-02	-3.750E-03	2.977E-01	2.296E-02	-1.884E-01	2.982E-01	1.374E-01	-2.085E-01
1.703E-03	2.515E-03	-1.501E-02	-4.202E-03	-1.728E-02	1.447E-03	3.491E-02	-3.929E-03	2.591E-01	-5.632E-01

ASSOCIATED EIGENVECTORS

-1.640E+02	7.921E+01	-1.736E+02	9.738E+00	-4.605E+01	-6.133E+00	-7.116E+01	-1.052E+02	-4.618E+01	3.568E+02
-1.786E+00	-6.441E-01	-3.822E+01	1.201E+01	3.094E+01	-2.860E+00	1.141E+02	2.003E+02	8.168E+01	-5.696E+02
6.416E-01	-5.128E-02	5.363E+00	-1.144E+00	1.263E+02	-1.566E+01	2.767E+02	5.006E+02	1.210E+02	6.060E+01
2.223E+00	-3.724E-01	1.360E+01	-2.325E+00	4.017E+02	-3.146E+01	-3.802E+02	-6.008E+02	-1.364E+02	5.542E+02
-4.304E-02	-1.884E-03	-7.013E-01	2.056E-01	1.208E+00	9.164E-02	8.601E+01	2.383E+02	1.787E+01	1.622E+02

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE -7.062E-05 -2.984E-05
SECOND DERIVATIVE
-6.194E-06 -4.195E-06= 8.668E-06 3.661E-06+ -1.764E-06 -3.568E-06+ -1.310E-05 -4.289E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE 3.169E-05 1.150E-05
SECOND DERIVATIVE
3.758E-06 4.183E-06= -3.889E-06 -1.411E-06+ 1.736E-06 3.557E-06+ 5.911E-06 2.038E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE 1.132E-06 9.201E-08
SECOND DERIVATIVE
1.147E-07 1.410E-08= -1.389E-07 -1.129E-08+ 4.494E-08 9.646E-09+ 2.086E-07 1.575E-08

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE -1.990E-07 -3.885E-09
SECOND DERIVATIVE
-1.282E-08 8.123E-09= 2.442E-08 4.766E-10+ -6.379E-10 7.562E-09+ -3.660E-08 8.480E-11

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE -1.160E-07 -3.502E-08
SECOND DERIVATIVE
-2.378E-08 -7.645E-09= 1.423E-08 4.297E-09+ -1.655E-08 -6.230E-09+ -2.146E-08 -5.712E-09

SUBROUTINE LEGROOT

DAMPING	FIRST DERIV	SECOND DERIV	RADICAL IN LEGUERRE	EIGENVALUE NO.	PROJECTED CROSSING
1.1156E-01	3.1884E-02	6.9776E-03	2.3816E-04	2	9.0710E+00
3.0272E-02	5.4797E-03	-1.6806E-04	3.5114E-05	3	1.1191E+01
-1.0090E-01	-1.1467E-02	1.0844E-03	2.4090E-04	4	2.2801E+01
-2.3870E-01	-3.4091E-02	-7.6580E-03	-6.6577E-04	1	1.0000E+03
-4.9019E-01	-2.2874E-01	-1.8017E-01	-3.5996E-02	5	1.0000E+03

ITERATION 1
16.3000 9.0710 11.1915 22.8007 1000.0000 1000.0000

FOUND 2 ROOTS, RFI FOR THE NEXT ROOT PREDICTED = 22.8007 ,IS BEYOND THE RANGE

ITERATION NO. 1 DENSITY = 1.1468E-07 SQRT(SEA LEVEL DENSITY/DENSITY) = 1.0000E+00
FLUTTER VEL = 6.3167E+03 AIR SPEED = 8.0361E+03 (VEL-AIRSPEED)*100/VEL = -27.2204

REDUCED FREQ. = 10.800
THE DERIVATIVES ARE W.R.T. SQRT(SEA LEVEL DENSITY/DENSITY)

DERIV. OF REDUCED FREQ. = -9.4800E-02
DERIV. OF FREQ**2 = -4.7296E+02
DERIV. OF VELOCITY = 6.2828E+03

SECOND DERIV. OF RF = 1.9344E-01 , SECOND DERIV. OF FREQ**2 = 1.0175E+03 , SECOND DERIV. OF VEL= 6.0119E+01

DENSITY = 7.806131E-08 , RFIMIN = 13.1467

EIGENVALUES

1.659E-03	-1.643E-04	1.248E-04	-9.995E-10	6.007E-06	-1.774E-08	2.735E-06	-8.727E-08	7.959E-07	-6.752E-08

EIGENVECTORS

9.757E-01	2.120E-01	-7.332E-02	-6.663E-02	2.950E-01	-5.938E-02	1.687E-01	-5.965E-02	-4.900E-03	-6.767E-03
-5.355E-02	-1.582E-02	9.930E-01	2.038E-02	5.126E-01	-1.407E-01	-5.744E-03	-7.547E-03	2.692E-01	4.842E-02
4.662E-03	1.555E-03	-5.833E-02	-4.163E-03	6.855E-01	-2.377E-01	-7.805E-01	2.527E-01	1.398E-01	5.118E-02
-7.412E-04	-2.032E-04	1.127E-02	1.070E-03	3.021E-01	-8.712E-02	5.069E-01	-1.946E-01	-8.650E-02	-2.275E-02
-1.003E-03	-3.119E-04	1.246E-02	6.382E-04	-3.716E-02	1.539E-02	-4.054E-03	-6.419E-03	-9.402E-01	-1.046E-01

ASSOCIATED EIGENVECTORS

1.472E+02	-1.651E+01	1.131E+02	6.579E+00	-4.357E+01	-1.999E+01	7.192E+01	2.903E+01	-9.691E+01	-5.179E+00
4.548E-01	3.260E-01	7.799E+01	-1.407E+00	5.293E+01	1.679E+01	-1.245E+02	-5.839E+01	1.429E+02	5.932E+00
-1.902E-01	-2.646E-02	-5.685E+00	-1.328E-01	2.107E+02	6.380E+01	-2.558E+02	-1.100E+02	-7.946E+01	1.603E+01
-6.771E-01	-2.924E-02	-9.808E+00	-4.672E-01	3.513E+02	1.275E+02	7.006E+02	2.611E+02	-1.432E+02	-1.216E+01
1.203E-02	3.092E-03	1.041E+00	2.941E-02	1.311E+00	1.052E+00	-2.745E+01	-1.716E+01	-6.094E+02	1.684E+01

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE -3.463E-05 -1.356E-05
SECOND DERIVATIVE
-3.073E-06 -3.895E-07= 5.268E-06 2.063E-06+ -4.375E-07 -2.941E-07+ -7.904E-06 -2.158E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE 1.331E-05 1.121E-06
SECOND DERIVATIVE
1.460E-06 3.826E-07= -2.024E-06 -1.705E-07+ 4.413E-07 2.919E-07+ 3.043E-06 2.613E-07

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE 4.414E-07 2.692E-08
SECOND DERIVATIVE
7.665E-08 4.305E-09= -6.713E-08 -4.094E-09+ 4.291E-08 2.120E-09+ 1.009E-07 6.279E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE 6.380E-09 -1.895E-08
SECOND DERIVATIVE
-4.310E-08 -5.573E-11= -9.704E-10 2.882E-09+ -4.357E-08 8.841E-10+ 1.440E-09 -3.822E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE -3.516E-08 -8.626E-09
SECOND DERIVATIVE
-5.776E-09 -1.013E-09= 5.347E-09 1.312E-09+ -3.089E-09 -7.440E-10+ -8.034E-09 -1.581E-09

SUBROUTINE LEGROOT

DAMPING	FIRST DERIV	SECOND DERIV	RADICAL IN LEGUERRE	EIGENVALUE NO.	PROJECTED CROSSING
-8.0085E-06	8.9845E-03	1.1500E-03	8.0731E-05	2	1.3151E+01
-2.9531E-03	4.6981E-03	6.3973E-05	2.2261E-05	3	1.3776E+01
-3.1908E-02	-6.8534E-03	-4.9126E-04	3.1293E-05	4	1.8854E+01
-8.4835E-02	-1.4585E-02	-3.1776E-03	-5.6847E-05	5	1.0000E+03
-9.8396E-02	-1.0165E-02	-8.3629E-04	2.1048E-05	1	3.4597E+01

```
ITERATION  1
    13.1500              13.1509     13.7759     18.8540     34.5970   1000.0000

FLUTTER EIGENVALUE NO. =    2, EIGENVALUES
 1.6694E-03 -1.6426E-04  1.2481E-04 -9.9954E-10  6.0067E-06 -1.7738E-08  2.7351E-06 -8.7272E-08  7.9590E-07 -6.7520E-08

RFI FOR PREDICTED CROSSINGS CORRESPOND TO EIGENVALUES NUMBERS
   2    3    4    1    5

ROOT NUMBER   1 , VELOCITY =  7650.904 , RFI =   13.150 , NO. OF ITERATIONS REQD, =  1

EIGENVALUES
     1.646E-03  -1.732E-04     1.338E-04   8.130E-07     6.310E-06   6.459E-10     2.730E-06  -9.954E-08     7.718E-07  -7.335E-08

EIGENVECTORS
     9.772E-01   2.030E-01    -8.001E-02  -7.497E-02     3.153E-01   1.067E-01     1.674E-01   4.296E-02    -7.202E-03  -9.833E-03
    -6.012E-02  -1.774E-02     9.906E-01   5.135E-02     5.456E-01   1.489E-01    -1.873E-02  -1.420E-02     2.687E-01   9.967E-02
     5.236E-03   1.733E-03    -6.028E-02  -6.222E-03     6.650E-01   1.309E-01    -8.094E-01  -2.346E-01     1.665E-01   9.285E-02
    -8.324E-04  -2.293E-04     1.162E-02   1.433E-03     3.194E-01   8.030E-02     4.957E-01   1.167E-01    -9.538E-02  -4.397E-02
    -1.126E-03  -3.487E-04     1.291E-02   1.082E-03    -3.838E-02  -5.399E-03     1.880E-03  -8.243E-03    -8.945E-01  -2.651E-01

ASSOCIATED EIGENVECTORS
     1.496E+02  -1.424E+01     1.189E+02   3.622E+00    -4.626E+01   4.501E+00     8.254E+01  -1.794E+01    -1.045E+02   1.175E+01
     5.097E-01   3.660E-01     7.269E+01  -3.976E+00     5.023E+01  -1.174E+01    -1.475E+02   2.406E+01     1.549E+02  -1.890E+01
    -2.149E-01  -3.367E-02    -5.690E+00   5.484E-02     1.996E+02  -4.908E+01    -3.001E+02   5.850E+01    -7.756E+01   3.139E+01
    -7.662E-01  -4.927E-02    -1.028E+01  -1.652E-01     3.757E+02  -7.205E+01     7.130E+02  -1.747E+02    -1.537E+02   1.067E+01
     1.359E-02   3.698E-03     1.013E+00  -1.152E-02     1.791E+00   1.358E-01    -4.101E+01   1.729E+00    -6.249E+02   1.237E+02

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -3.666E-05   -1.394E-05
SECOND DERIVATIVE
  -3.168E-06  -4.533E-07=     5.313E-06   2.005E-06+    -5.057E-07  -3.479E-07+    -7.976E-06  -2.111E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       1.428E-05    1.390E-06
SECOND DERIVATIVE
   1.551E-06   4.468E-07=    -2.070E-06  -2.015E-07+     5.072E-07   3.453E-07+     3.113E-06   3.030E-07

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       4.911E-07    2.962E-08
SECOND DERIVATIVE
   7.606E-08   4.068E-09=    -7.117E-08  -4.293E-09+     4.032E-08   1.897E-09+     1.069E-07   6.465E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE      -2.072E-08   -1.871E-08
SECOND DERIVATIVE
  -4.012E-08   7.400E-10=     3.002E-09   2.712E-09+    -3.860E-08   1.575E-09+    -4.517E-09  -3.547E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -3.901E-08   -9.328E-09
SECOND DERIVATIVE
  -6.093E-09  -1.151E-09=     5.654E-09   1.352E-09+    -3.249E-09  -8.569E-10+    -8.497E-09  -1.646E-09

SUBROUTINE LEGROOT
       DAMPING             FIRST DERIV           SECOND DERIV       RADICAL IN LEGUERRE      EIGENVALUE NO.     PROJECTED CROSSING
      6.0773E-03           9.7437E-03            1.1888E-03            8.7714E-05                  2                 1.3151E+01
      1.0237E-04           4.6870E-03           -8.6030E-05            2.1977E-05                  3                 1.3778E+01
     -3.6457E-02          -7.1307E-03           -3.7288E-04            3.7253E-05                  4                 1.9773E+01
     -9.5037E-02          -1.6889E-02           -3.9494E-03           -9.0099E-05                  5                 1.0000E+03
     -1.0519E-01          -1.0747E-02           -9.5653E-04            1.4885E-05                  1                 4.1064E+01
```

```
ITERATION  1
    13.8000           13.1511     13.7782     19.7731     41.0639   1000.0000

FLUTTER EIGENVALUE NO. =   3, EIGENVALUES
 1.6462E-03 -1.7316E-04  1.3377E-04  8.1299E-07  6.3098E-06  6.4591E-10  2.7303E-06 -9.9540E-08  7.7181E-07 -7.3350E-08

RFI FOR PREDICTED CROSSINGS CORRESPOND TO EIGENVALUES NUMBERS
   2    3    4    1    5

ROOT NUMBER   2 , VELOCITY = 35709.560 , RFI =   13.800 , NO. OF ITERATIONS REQD. =  1

EIGENVALUES
     1.365E-03  -2.702E-04     2.533E-04   2.375E-05     1.055E-05   2.630E-07     2.091E-06  -1.736E-07     3.968E-07  -1.626E-07

EIGENVECTORS
    -8.583E-01  -4.858E-01     1.715E-01   1.998E-01     5.171E-01   7.622E-02    -1.339E-01  -2.992E-02     6.131E-02   2.691E-02
     1.239E-01   1.082E-01    -9.418E-01  -1.951E-01     7.295E-01   9.714E-02     1.156E-01   3.068E-02    -3.235E-01  -8.539E-02
    -1.066E-02  -9.686E-03     7.019E-02   1.829E-02     3.094E-01   4.979E-03     8.873E-01   2.323E-01    -6.243E-01  -2.157E-01
     1.734E-03   1.499E-03    -1.327E-02  -3.299E-03     2.977E-01   2.265E-02    -3.438E-01  -7.952E-02     2.382E-01   7.416E-02
     2.310E-03   2.043E-03    -1.519E-02  -3.726E-03    -1.724E-02   9.902E-04     1.717E-02   2.718E-02     6.108E-01   1.094E-01

ASSOCIATED EIGENVECTORS
    -1.782E+02   5.187E+01    -1.751E+02   1.028E+01    -4.607E+01  -4.483E+00    -1.223E+02   3.168E+01     3.576E+02   1.095E+02
    -1.745E+00  -7.132E-01    -3.836E+01   1.045E+01     3.111E+01  -2.756E+00     2.258E+02  -4.339E+01    -5.712E+02  -1.819E+02
     6.625E-01   1.639E-02     5.359E+00  -1.018E+00     1.364E+02  -1.461E+01     5.620E+02  -1.017E+02     7.900E+01  -9.035E+01
     2.321E+00  -1.087E-01     1.358E+01  -2.107E+00     4.015E+02  -3.092E+01    -6.914E+02   1.605E+02     5.462E+02   2.244E+02
    -4.389E-02  -5.645E-03    -7.003E-01   1.790E-01     1.330E+00   5.838E-02     2.522E+02  -1.228E+01     1.658E+03   3.054E+02

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 1
FIRST DERIVATIVE      -5.964E-05   -2.078E-05
SECOND DERIVATIVE
  -4.778E-06  -2.661E-06=    6.039E-06   2.105E-06+   -1.695E-06  -2.294E-05+   -9.122E-06  -2.471E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 2
FIRST DERIVATIVE       2.746E-05    8.317E-06
SECOND DERIVATIVE
   3.108E-06   2.657E-05=   -2.781E-06  -8.422E-07+    1.676E-06   2.288E-06+    4.213E-06   1.211E-06

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 3
FIRST DERIVATIVE       9.350E-07    6.267E-08
SECOND DERIVATIVE
   7.833E-08   7.921E-09=   -9.468E-08  -6.346E-09+    3.085E-08   5.421E-09+    1.422E-07   8.846E-09

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 4
FIRST DERIVATIVE      -1.634E-07   -2.658E-09
SECOND DERIVATIVE
  -8.669E-09   4.528E-09=    1.654E-08   2.692E-10+   -3.926E-10   4.214E-09+   -2.482E-08   4.516E-11

DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER 5
FIRST DERIVATIVE      -9.668E-08   -2.383E-08
SECOND DERIVATIVE
  -1.628E-08  -4.255E-09=    9.791E-09   2.413E-09+   -1.132E-08  -3.462E-09+   -1.475E-08  -3.205E-09
```

```
SUBROUTINE LEGROOT
     DAMPING            FIRST DERIV          SECOND DERIV      RADICAL IN LEGUERRE     EIGENVALUE NO.     PROJECTED CROSSING
    9.3771E-02           2.2666E-02            4.4264E-03               9.8671E-05                  2             1.0310E+01
    2.4933E-02           3.7317E-03           -9.5807E-05               1.6314E-05                  3             1.3577E+01
   -8.3046E-02          -7.7608E-03            6.0868E-04               1.1078E-04                  4             2.7640E+01
   -1.9804E-01          -2.3887E-02           -4.7313E-03              -3.6638E-04                  1             1.0000E+03
   -4.0967E-01          -1.5985E-01           -1.0542E-01              -1.7634E-02                  5             1.0000E+03

ITERATION  1
    19.7500             10.3100      13.5771      27.6403   1000.0000    1000.0000

FOUND  2 ROOTS, REI FOR THE NEXT ROOT PREDICTED =    27.6403 ,IS BEYOND THE RANGE

ITERATION NO.  2 DENSITY =   7.8061E-08 SQRT(SEA LEVEL DENSITY/DENSITY) =   1.2121E+00
 FLUTTER VEL =   7.6509E+03 AIR SPEED =   7.6504E+03 (VEL-AIRSPEED)*100/VEL =    .0065
```

APPENDIX B

FORTRAN PROGRAM LISTING

The FORTRAN program listing for program MATCH and related subroutines are presented in this appendix.

Program MATCH

```
      OVERLAY(MATCH,0,0)
      PROGRAM MATCH(INPUT=1,OUTPUT=1,TAPE4,TAPE7=1,TAPE88,
     1              TAPE5=INPUT,TAPE6=OUTPUT)
C***********************************************************************
C*    KUMAR G.BHATIA, JULY 24,1972                                     *
C*    FINDS FLUTTER MATCH POINT OR CROSSINGS FOR SPECIFIED DENSITIES   *
C*    PERF = MAXIMUM (VELOCITY-SPEED OF SOUND*MACH)*100/VELOCITY       *
C*           TOLERANCE SPECIFIED FOR TERMINATION (IN PERCENT)          *
C*    MAXMAT = MAXIMUM NUMBER OF ITERATIONS ALLOWED                    *
C*    ITROPO DEFINES THE INITIAL DENSITY FOR MATCH POINT SEARCH        *
C*           = 0, INITIAL DENSITY = SEA LEVEL DENSITY                  *
C*           = 1, INITIAL DENSITY = DENSITY FOR GEOMETRIC ALTITUDE  OF *
C*                                  36,200 FEET                        *
C*           =-1, INITIAL DENSITY = RHO(1) IN NAMELIST FOR LEFCROS     *
C*    IMATCH = 0 COMPUTES FLUTTER CROSSINGS AND VELOCITIES FOR SINGLE  *
C*               OR MULTIPLE DENSITIES AS SPECIFIED IN LEFCROS         *
C*           = 1 MATCH POINT IS COMPUTED WITH INITIAL DENSITY SPECIFIED*
C*               BY ITROPO                                             *
C*    REFSLD = REFERENCE SEA LEVEL DENSITY IN APPROPRIATE MASS UNITS   *
C*    UNITL = 1 IF ALL LENGTH UNITS ARE IN FEET, NO INPUT REQUIRED     *
C*            NO. OF LENGTH UNITS / FOOT, MUST BE INPUT FOR MATCH-POINT*
C*                                        SEARCH WHEN THE LENGTH UNITS *
C*                                        SELECTED ARE OTHER THAN FEET *
C***********************************************************************
      REAL MACH
      NAMELIST/NAMATCH/ PERF,MAXMAT,MACH,ITROPO,IMATCH,REFSLD,UNITL
      UNITL = 1.0
      READ(5,NAMATCH)
      WRITE(6,NAMATCH)
      IF ( IMATCH .EQ. 0 ) GO TO 100
      CALL CROSMAT(MACH,PERF,MAXMAT,ITROPO,REFSLD,UNITL)
      GO TO 200
  100 CALL LEFCROS(IMATCH,RHOM,REIMIN,SVEL)
  200 CONTINUE
      END
```

Subroutine CROSMAT

```
      SUBROUTINE CROSMAT(MACH,PERF,MAXMAT,ITROPO,REF,UNITL)
      COMMON/DERIVS/ DRF,DMU,DVEL,SDRF,SDMU,SDVEL
      COMMON NM,NMAX,NEIG,NVEC
      REAL MACH
      DENTROP = 1.836826E82
      IMATCH = 1
      NDER = 2
      NL = 2
      NL1 = NL - 1
      IF (ITROPO) 6,7,8
    6 RHO = -1.0
      GO TO 9
    7 RHO = REF
      GO TO 9
    8 RHO = REF * 2.9639E-01
      DRFI = -RFI*RFI*DRF
      SDRFI = 2.0*DRFI*DRFI/RFI - RFI*RFI*SDRF
      RFI = RFI + DRFI*DEL + 0.5*SDRFI*DEL*DEL
      GO TO 1
  500 RETURN
 1000 FORMAT(//,43H MATCH-POINT ITERATION DID NOT CONVERGE IN ,I3,
     1 11H ITERATIONS)
 2000 FORMAT(/,35H ARGUMENT OF RADICAL IN LAGUERRE = ,F12.3,18H, ITERATI
     1ON NO. = ,I3,12H, DENSITY = ,E12.4,8H, VEL = ,F9.3,/,10X,24H, SPEE
     2D OF SOUND*MACH = ,F9.3)
 2500 FORMAT(1H1)
 3000 FORMAT(//,14H ITERATION NO.,I3,11H DENSITY = ,F12.4, 35H SQRT(SEA
     1LEVEL DENSITY/DENSITY) = ,F12.4,/,16H  FLUTTER VEL = ,F12.4,
     213H AIR SPEED = ,F12.4,26H (VEL-AIRSPEED)*100/VEL = ,F8.4)
      END
```

Subroutine DERVDEN

```
      SUBROUTINE DERVDEN(RHO,NDER,RFI)
C*    RHO = SQRT(REFERENCE DENSITY/DENSITY)
      COMMON/DERIVS/ DRF,DMU,DVEL,SDRF,SDMU,SDVEL
      COMMON NM,NMAX,NEIG,NVEC
      COMMON/BLK1/ AF,DAF,SDAF
      COMMON/BLK2/ EIG,VEC,AVEC,DIFS,SDIFS
      COMMON/BLK3/ TP,TP2
      COMPLEX VSAU,RI1,RI4,RI5,RIT,RI7,RI8
      COMPLEX AF(12,12),DAF(12,12),SDAF(12,12),EIG(12),VEC(12,12),
     1        AVEC(12,12),DIFS(12),SDIFS(12),TP(12,12),TP2(12,12)
C***********************************************************************
C*    KUMAR G. BHATIA, JULY 21,1972.                                   *
C*    COMPUTES DERIVATIVES WITH RESPECT TO SQRT(2.378E-03/DENSITY)     *
C*    DVEL = FIRST DERIV OF VELOCITY, SDVEL = SECOND DERIV OF VELOCITY *
C*    DMU,SDMU ARE THE FIRST AND SECOND DERIV,RESP, OF FLUTTER FREQ**2 *
C*    DRF,SDRF ARE THE FIRST AND SECOND DERIV,RESP, OF REDUCED FREQ.   *
C*    NDER = NUMBER OF DERIVATIVES REQUIRED, 1 OR 2                    *
C***********************************************************************
      REWIND 4
      READ(4) M,RFI,VEL,VSAU,EIG,VEC,AVEC,AF,TP
      RHOS = RHO*RHO
```

```
      EIGM = EIG(M)
      CALL TMMPROD(AVEC,AF,VEC,NM,NVEC,NMAX,NDER,TP2)
      EIGMI = 1.0/EIGM
      RI1 = 2.0*TP2(M,M)/RHO
      R2 = -REAL(TP(M,M))
      A2 = -AIMAG(TP(M,M))
      DRF = -AIMAG(RI1)/A2
      DMU = EIGMI*( REAL(RI1)+DRF*R2)
      COEF = 0.5*EIGM*DMU - RFI*DRF
      DVEL = VEL*COEF
      PRINT 1000, RFI,DRF,DMU,DVEL
      IF (NDER .EQ. 1) RETURN
      RI4 = 2.0*EIGM*DMU*DMU
      RI5 = 0.0
    9 CONTINUE
      CALL LEFCROS(IMATCH,RHO,RFI,VEL)
      DEN = SQRT(REF/RHO)
      DENL = DEN
      DENI = 1.0/DEN
      GO TO 2
    1 CONTINUE
      IMATCH = IMATCH+1
      VELL = VEL
      DENL = DEN
      RHO = REF*DENI*DENI
      CALL FOMATCH(IMATCH,RHO,RFI,VEL)
    2 CONTINUE
      CALL SOUND(MACH,DEN,SOS,DSOS,SDSOS)
      SOS = SOS*UNITL
      DSOS = DSOS*UNITL
      SDSOS = SDSOS*UNITL
      F = VEL-SOS
      PER = F*100.0/VEL
      PRINT 2500
      PRINT 3000, IMATCH,RHO,DEN,VEL,SOS,PER
      WRITE(7,3000) IMATCH,RHO,DEN,VEL,SOS,PER
      IF ( ABS(PER) - PERF ) 500,500,5
    5 IF (IMATCH .NE. MAXMAT) GO TO 10
      PRINT 1000,IMATCH
      WRITE(7,1000) IMATCH
      GO TO 500
   10 CALL DERVDEN(DEN,NDER,RFI)
      IF (DEN .NE. DENTROP) GO TO 200
      IF (F .GT. 0.0) GO TO 200
      DSOS = 0.0
      SDSOS = 0.0
  200 CONTINUE
      DF = DVEL - DSOS
      SDF = SDVEL-SDSOS
      H = NL1*(NL1*DF*DF-NL*F*SDF)
      IF (H .GE. 0.0) GO TO 250
      PRINT 2000, H,IMATCH,RHO,VEL,SOS
      WRITE(7,2000) H,IMATCH,RHO,VEL,SOS
      GO TO 500
  250 CONTINUE
      H = SQRT(H)
      H = SIGN(H,DF)
      DEN = DEN - NL*F/(DF+H)
      IF (DENL - DENTROP) 260,290,270
  260 IF (DEN - DENTROP) 290,290,280
  270 IF (DEN - DENTROP) 280,290,290
  280 DEN = DENTROP
  290 CONTINUE
      DENI = 1.0/DEN
      DEL = DEN - DENL
      DO 100 L=1,NVEC
         IF (L .EQ. M) GO TO 100
      RIT = DRF*DRF*TP(L,M)*TP(M,L) - 2.0/RHO*DRF*(TP2(L,M)*TP(M,L)+
     1      TP2(M,L)*TP(L,M)) + 4.0/RHOS*TP2(L,M)*TP2(M,L)
      RIT = RIT /( 1.0-EIGM/EIG(L) )
```

```
      RI5 = RI5 + RIT
  100 CONTINUE
      RI5 =  2.0*EIGMI*RI5
      RI7 = -EIGMI* ( 6.0*TP2(M,M)/RHOS-4.0*TP(M,M)*DRF/RHO+VSAU*DRF*
     1       DRF )
      RI8 = -EIGMI*TP(M,M)
      SDRF = -AIMAG(RI5+RI4+RI7)/AIMAG(RI8)
      SDMU = REAL(RI5+RI4+RI7) + REAL(RI8)*SDRF
      SDVEL = DVEL*COEF + VEL*( 0.5*EIGM*(SDMU-EIGM*DMU*DMU) + REI*(REI*
     1        DRF*DRF-SDRF) )
      PRINT2000, SDRF,SDMU,SDVEL
      RETURN
 1000 FORMAT(//,17H REDUCED FREQ. = ,F8.3,/,63H **THE DERIVATIVES ARE W.
     1R.T. SQRT(SEA LEVEL DENSITY/DENSITY)**,/,/,28H  DERIV. OF REDUCED
     2FREQ. = ,F13.4,/,22H  DERIV. OF FREQ**2 = ,F13.4,/,
     3  23H  DERIV. OF VELOCITY = ,F13.4)
 2000 FORMAT(/,22H SECOND DERIV. OF RF =,F13.4,29H , SECOND DERIV. OF FR
     1EQ**2 =,  F13.4,24H , SECOND DERIV. OF VEL=,F13.4)
      END
```

Subroutine SOUND

```
      SUBROUTINE SOUND(MACH,RHO,SOS,DSOS,SDSOS)
      REAL MACH
C*    RHO = SQRT(SEA LEVEL DENSITY)/SQRT(DENSITY)
      IF (RHO .GT. 1.836826882) GO TO 10
      A = 1515.639571
      B = -520.6920622
      C = 121.1824916
      SOS = MACH*( A+RHO*(B+RHO*C) )
      DSOS = MACH*( B+2.0*RHO*C)
      SDSOS = MACH * 2.0 * C
      RETURN
   10 SOS = MACH*968.08
      DSOS = 0.0
      SDSOS = 0.0
      RETURN
      END
```

Subroutine LEGROOT

```
      SUBROUTINE LEGROOT(NEIG,ICON,REI,G,IAR,IPRT,NL)
      COMMON/BLK2/ EIG,VEC,AVEC,DIFS,SDIFS
      COMPLEX EIG(12),VEC(12,12),AVEC(12,12),DIFS(12),SDIFS(12)
      DIMENSION REI(1),ICON(1),G(1),IAR(1)
      REAL IOK,MINIOK
C*    KUMAR G. BHATIA, JUNE 12,1972.                                   *
C*    COMPUTES THE ROOTS USING MODIFIED LEGUERRE ITERATION, WHERE THE  *
C*       ROOTS CORRESPOND TO THE IMAGINARY PART OF INVERSE OF THE FREQ.*
C*       SQUARED AS A FUNCTION OF INVERSE OF REDUCED FREQ.(RF).        *
C*    AT INPUT REI(1) CONTAINS 1/RF WHERE FUNCTION+DERIVATIVES ARE KNOWN
C*    AT OUTPUT REI(J) CONTAIN PROJECTED ROOTS                         *
C*    PROJECTED CROSSING = 3000, REAL PART OF EIGENVALUE IS NEGATIVE   *
C*    PROJECTED CROSSING = 2000, ICON(J) .NE. 0                        *
C*    PROJECTED CROSSING = 1000, REAL REI(J) COULD NOT BE PREDICTED    *
C*    *****************************************************************
      IOK = REI(1)
      REI2 = 0.5*IOK
      N = 0
      CALL DAMPAR(NEIG,EIG,G,IAR)
```

```
      IF (IPRT .NE. 0) PRINT 1000
      DO 200 J=1,NEIG
         I = IAR(J)
      II = I
      G1 = 0.0
      G2 = 0.0
      A = 0.0
         IF ( G(I) .EQ. -1000.0 ) GO TO 4
         IF (ICCN(J) .EQ. 0) GO TO 5
      RFI(J) = 2000.0
      RFIJ = RFI(J)
      GO TO 100
    4 RFI(J) = 3000.0
      RFIJ = RFI(J)
         GO TO 100
    5 CONTINUE
         A0 = AIMAG(EIG(I))
         A1 = AIMAG(DIFS(I))
         A2 = AIMAG(SDIFS(I))
      R0=REAL(EIG(I)) $ R1=REAL(DIFS(I)) $ R2=REAL(SDIFS(I))
      G0 = G(I)   $ G1 = (A1 - G0*R1)/R0 $ G2 = (A2-G0*R2-2.*G1*R1)/R0
      A = G1*G1 - G0*G2
      IF (NL .NE. 0) A = (NL-1)*( (NL-1)*G1*G1-NL*G0*G2 )
      IF ( A .GT. 0.0 ) GO TO 10
      IF (G0 .LT. 0.0) GO TO 8
      IF (G1 .GE. 0.0) RFI(J) = IOK - G0/G1
      IF (G1 .LT. 0.0) RFI(J) = IOK + G0/G1
      IF (RFI(J) .LT. RFI2) RFI(J) = RFI2
      GO TO 12
    8 CONTINUE
      N = N + 1
         RFI(J) = 1000.0
      RFIJ = RFI(J)
         GO TO 100
   10 IF (NL .EQ. 0) GO TO 11
      RFI(J) = IOK - NL*G0/(G1+SQRT(A))
      GO TO 12
   11 RFI(J) = IOK - G0/SQRT(A)
   12 CONTINUE
      RFIJ = RFI(J)
      IF ( J .EQ. 1 ) GO TO 100
      IF ( RFIJ .GE. RFI(J-1) ) GO TO 100
      J1 = J - 1
      DO 15 I=1,J1
      IF ( RFIJ .GE. RFI(J-I) ) GO TO 16
      JMIN = J - I
   15 CONTINUE
   16 SAVEI = IAR(J)
      JJMIN = J-JMIN
      DO 20 I=1,JJMIN
      IAR(J-I+1) = IAR(J-I)
   20 RFI(J-I+1) = RFI(J-I)
      RFI(JMIN) = RFIJ
      IAR(JMIN) = SAVEI
  100 CONTINUE
      IF (IPRT .EQ. 0) GO TO 200
      PRINT 1500,G0,G1,G2,A,II,RFIJ
  200 CONTINUE
      IF ( RFI(1) .GT. 0.0 ) GO TO 400
      IF ( RFI(NEIG) .LE. 0.0 ) GO TO 400
      NEIG1 = NEIG - 1
      DO 300 J=1,NEIG1
         JJ = NEIG1 -J + 1
         IF (RFI(JJ)) 250,250,300
  250    JJ1 = JJ + 1
         DO 300 I=1,JJ
```

```
            RFI(I) = RFI(JJ1)
  300 CONTINUE
  400 CONTINUE
      IF ( N .LT. NEIG)  RETURN
      PRINT 2000, N
 1000 FORMAT(/,19H SUBROUTINE LEGROOT,/,6X,8H DAMPING,10X,12H FIRST DERI
     1V,7X,13H SECOND DERIV,4X,20H RADICAL IN LEGUERRE,3X,15H EIGENVALUE
     2 NO.,3X,19H PROJECTED CROSSING)
 1500 FORMAT(4(5X,E11.4,5X),10X,I2,13X,E11.4)
 2000 FORMAT(//,38H ARGUMENT OF LAGUERRE IS NEGATIVE FOR ,I3,7H  MODES)
      END
```

Subroutine DAMPAR

```
      SUBROUTINE DAMPAR(NEIG,EIG,G,IAR)
      COMPLEX EIG(1)
      DIMENSION G(1),IAR(1)
      IAR(1) = 1
      G(1) = AIMAG(EIG(1))/REAL(EIG(1))
      IF (REAL(EIG(1)) .LE. 0.0) G(1) = -1000.0
      DO 5 I=2,NEIG
         IAR(I) = I
         G(I) = AIMAG(EIG(I))/REAL(EIG(I))
         IF (REAL(EIG(I)) .LE. 0.0) G(I) = -1000.0
         IC = I
         I1 = I - 1
         DO 4 J=1,I1
            M = I1 -(J-1)
         ICC = IAR(IC)
         MM = IAR(M)
            IF (G(ICC) .LE. G(MM)) GO TO 5
            IT = IAR(M)
            IAR(M) = IAR(IC)
            IAR(IC) = IT
            IC = IC-1
    4    CONTINUE
    5 CONTINUE
      RETURN
      END
```

Subroutine GETAERO

```
      SUBROUTINE GETAERO(NM,NMAX,RFI,ID,RFIL,RFIR,DEL)
      COMMON/BLK1/ AF,DAF,SDAF
      COMPLEX AF(12,12),DAF(12,12),SDAF(12,12)
C*    GETS AERODYNAMIC FORCES FROM RANDOM ACCESS FILE 88 IF ID=0,ELSE  *
C*         GETS DERIVATIVES TOO. ENTRY GETDAER GETS DERIVS. ONLY.      *
C*    RFI = 1.0/REDUCED FREQ., RFIMIN = FIRST RFI RECORD ON 88.        *
C*    DEL = CONSTANT INCREMENT OF RFI ON 88.                           *
C*    KUMAR G.BHATIA, JUNE 13,1972.                                    *
C*
      IF (RFI .GE. RFIL .AND. RFI .LE. RFIR) GO TO 10
      PRINT 2000,RFI
 2000 FORMAT(/,* RFI = *,F10.3,*, IS OUTSIDE THE RANGE OF VALUES *)
      STOP
```

```
   10 CONTINUE
      NW = 2*NMAX*NMAX
      STEPS = (RFI-RFIL)/DEL
      IK = STEPS
      IR = 2
      IF ((STEPS-IK) .LE. 0.5) IR=1
      RFI = RFIL + (IK+IR-1)*DEL
      IK = (IK+IR-1)*3 + 1
      CALL READMS(88,AF,NW,IK)
      IF (ID .EQ. 0) RETURN
      ENTRY GETDAER
      CALL READMS(88,DAF,NW,IK+1)
      CALL READMS(88,SDAF,NW,IK+2)
      RETURN
      END
```

Subroutine RANDAX

```
      SUBROUTINE RANDAX(NM,NMAX,SS,BR,RHO,NRF)
      DIMENSION NRF(1)
      COMMON/BLK1/AF,DAF,SDAF
      COMPLEX AF(12,12),DAF(12,12),SDAF(12,12)
      DIMENSION R0(2,12,12),R1(2,12,12),R2(2,12,12)
      EQUIVALENCE (AF,R0),(DAF,R1),(SDAF,R2)
C*    THE SUBROUTINE READS FROM TAPE 4 AND TRANSFERS TO RANDOM ACCESS  *
C*       FILE ON TAPE 88. THE AERO FORCE+DERIV MATRICES ARE MULTIPLIED*
C*       BY DENSITY PARAMETER BEFORE TRANSFER TO 88.                  *
C*    KUMAR G.BHATIA, JUNE 13,1972
C
C     COMPUTE THE DENSITY PARAMETER
C     DP IS IN LB.SEC**2/INCH UNITS
C     INPUT  SS=SEMISPAN,BR=REFERENCE SEMICHORD ARE IN INCHES
C            RHO=AIR DENSITY IN SLUGS/FT**3
      PI = 3.14159265358979
      DP = 4.0*PI*BR*SS*SS*RHO
      REWIND 4
      READ(4) NK,MACH,NM
      CALL OPENMS(88,NRF,1600,0)
      NW = 2*NMAX*NMAX
      DO 100 IK=1,NK
      READ(4) RF,X,((R0(1,I,J),I=1,NM),J=1,NM),((R0(2,I,J),I=1,NM),J=1,
     1              NM)
      READ(4) ((R1(1,I,J),I=1,NM),J=1,NM),((R1(2,I,J),I=1,NM),J=1,NM)
      READ(4) ((R2(1,I,J),I=1,NM),J=1,NM),((R2(2,I,J),I=1,NM),J=1,NM)
      RFI = 1.0/RF
      F = DP*RFI*RFI
         DO 10 I=1,NM
         DO 10 J=1,NM
         AF(I,J) = F*AF(I,J)
         DAF(I,J) = F*DAF(I,J) - 2.0*RFI*AF(I,J)
         SDAF(I,J) = F*SDAF(I,J) - 2.0*RFI*(2.0*DAF(I,J)+RFI*AF(I,J))
   10    CONTINUE
         IK3 = (IK-1)*3 + 1
         CALL WRITMS(88,AF,NW,IK3)
         CALL WRITMS(88,DAF,NW,IK3+1)
         CALL WRITMS(88,SDAF,NW,IK3+2)
```

```
  100 CONTINUE
      RETURN
      END
```

Subroutine LEFCROS

```
      SUBROUTINE LEFCROS(IMATCH,RHOM,RFIMIN,SVEL)
      COMPLEX AF(12,12),DAF(12,12),SDAF(12,12),TP(12,12),TP2(12,12),
     1        EIG(12),VEC(12,12),AVEC(12,12),DIFS(12),SDIFS(12)
      DIMENSION INTH(12,2),NRF(1600),ICON(12),SM(12,12),SK(12,12),
     1          C(12,12),RFI(12),G(12),IAR(12),VEL(12),RHO(10)
      EQUIVALENCE (SK,C)
      COMPLEX MUM
      COMMON NM,NMAX,NEIG,NVEC
      COMMON/BLK1/ AF,DAF,SDAF
      COMMON/BLK2/ EIG,VEC,AVEC,DIFS,SDIFS
      COMMON/BLK3/ TP,TP2
      COMMON/BLK4/ SM,C,INTH
      NAMELIST/NAM1/ SK,SM,LSTIFF,SS,BR,NM,RFIL,RFIR,DEL,
     1               NROOT,NITMAX,ND,RHO,RFIMIN,IPRT,IOPT
      NAMELIST/OPTION/ NMAX,NEIG,NEVRED,EIGRAT,NL,ICON
C*    *****************************************************************
C*    COMPUTES FLUTTER CROSSINGS AND VELOCITIES FOR SINGLE OR MULTIPLE*
C*       DENSITIES, IMATCH = 0 AND ND = NO. OF DENSITIES.              *
C*    FOR IMATCH .NE. 0, THE LOWEST FLUTTER VELOCITY AND OTHER INFO IS*
C*       RETURNED TO THE CALLING PROGRAM. THE INITIAL GUESS FOR RFIMIN*
C*       IS PICKED UP FROM THE NAMELIST FOR IMATCH = 0 OR 1, FOR OTHER*
C*       VALUES OF IMATCH THE GUESS SUPPLIED FROM THE PARAMETER LIST. *
C*    KUMAR G. BHATIA, PROGRAM CHECK COMPLETED JULY 20,1972.          *
C*    *****************************************************************
C*    DEFINITION AND ASSIGNEMENT OF THE NAMELIST NAM1 PARAMETERS      *
C*    LSTIFF = 0 DIAGONAL STIFFNESS MATRIX IS INPUT IN SK, DIAGONAL   *
C*          FLEXIBILITY MATRIX IS COMPUTED AND STORED IN SK           *
C*    LSTIFF = +1 FULL STIFFNESS MATRIX IS INPUT IN SK, IS INVERTED AND*
C*          DESTROYED USING CDC MATRIX INVERSION ROUTINE               *
C*    LSTIFF = -1 DIAGONAL OR FULL FLEXIBILITY MATRIX IS INPUT IN SK   *
C*    SK = GENERALISED STIFFNESS OR FLEXIBILITY MATRIX, SEE LSTIFF     *
C*    SM = GENERALISED MASS MATRIX, MAYBE DIAGONAL OR FULL             *
C*    SS = SEMISPAN, BR = REFERENCE CHORD - BOTH MUST BE IN APPROPRIATE*
C*                                     UNITS                           *
C*    DEL = EQUAL INCREMENT ON RFI AT WHICH AERODYNAMIC FORCES ARE ON  *W9
C*       TAPE 4                                                        *
C*    RFIL = MINIMUM VALUE OF RFI FOR WHICH AERO FORCES ARE SUPPLIED   *
C*    RFIR = MAXIMUM VALUE OF RFI FOR WHICH AERO FORCES ARE SUPPLIED   *
C*    IAERO = 0 AERODYNAMIC FORCES AND  FIRST TWO DERIVATIVES ARE     *
C*       SUPPLIED AT EQUAL RFI INTERVAL OF DEL, STARTING WITH RFIL
C*    IAERO .NE. 0 ONLY AERODYNAMIC FORCES ARE SUPPLIED FOR INCREASING *
C*       VALUES OF RFI, STARTING WITH RFIL                             *
C*    NM = NUMBER OF MODES, NMAX = MAXIMUM NO. OF MODES ALLOWED = 12   *
C*    NEIG = NO. OF EIGENVALUES TO BE COMPUTED, NEIG .LE. NM           *
C*    NVEC = NO. OF EIGENVECTORS TO BE COMPUTED, NVEC .LE. NM          *
C*    EIGRAT  REQUIRED ONLY WHEN NEVRED .NE. 0, SEE NEVRED             *
C*    NEVRED = 0 NEIG EIGENVALUES AND NVEC VECTORS ARE COMPUTED FOR    *
C*       FIRST AND SUBSEQUENT EIGENSOLUTIONS                           *
C*    NEVRED .NE. 0  AFTER THE FIRST EIGENSOLUTION ONLY THE  SMALLEST  *
C*       EIGENVALUES (AND VECTORS) ARE COMPUTED SUCH THAT THE SMALLEST *
C*       EIGENVALUE NOT COMPUTED IS AT LEAST EIGRAT TIMES THE FLUTTER  *
C*       EIGENVALUE                                                    *
C*    ICON IS INITIALLY SET TO ZERO, IF ICON(L) IS INPUT AS NONZERO    *
C*       THE L TH LARGEST DAMPING ROOT PROJECTION FOR FINDING RFI, IS  *
C*       NOT COMPUTED                                                  *
```

```
C*    NITMAX = MAXIMUM NUMBER OF ITERATIONS ALLOWED PER ROOT
C*    NROOT = NO. OF ROOTS TO BE SEARCHED                                   *
C*    ND = NO. OF DENSITIES FOR WHICH THE FLUTTER CROSSINGS ARE DESIRED*
C*    RHO IS THE VECTOR OF DENSITIES IN APPROPRIATE MASS UNITS              *
C*    IPRT = 0 SUMMARY PRINTOUT ONLY, =1 ROOT PROJECTIONS AND EIGENVALU*
C*             ES PRINTED AT EACH STEP, = 2 EIGENVALUE DERIVS. AND EIGEN*
C*             VECTORS ALSO PRINTED AT EACH STEP
C*    NL SPECIFIES THE ASSUMED NO. OF ZEROS OF DAMPING AS A FUNCTION OF*
C*       RFI,  NL=0 ASSUMES DAMPING AS A TRANSCEDENTAL FUNCTION, DEFAULT*
C*       VALUE IS 0                                                         *
C*    RFI = RECIPROCAL OF REDUCED FREQUENCY = REDUCED VELOCITY              *
C*    RFIMIN = INITIAL GUESS FOR RFI                                        *
C*********************************************************************
      REWIND 7
      DO 5 I=1,12
    5 ICON(I) = 0.0
      NL = 0
      IOPT = 0
      IPRT = 0
      NEVRED = 0
      NMAX = 12
      READ(5,NAM1)
      WRITE(6,NAM1)
      NEIG = NM
      NVEC = NM
      IF (IOPT .EQ. 0) GO TO 7
      READ(5,OPTION)
      WRITE(6,OPTION)
    7 CONTINUE
      DEL2 = DEL/2.0
      IF (LSTIFF) 14,6,12
    6 DO 10 I=1,NM
      DO 10 J=1,NM
         IF ( J .EQ. I ) C(I,J)=1.0/SK(I,J)
   10 CONTINUE
      GO TO 14
   12 CALL MATINV(SK,NM,DUMMY,0,DET,NRF,INTH,NMAX,ISCALE)
   14 REFRHO = RHO(1)
      IF (RHOM .EQ. -1.0) RHOM = RHO(1)
      IF (IMATCH .NE. 0) REFRHO = RHOM
   15 CALL RANDAX(NM,NMAX,SS,BR,REFRHO,NRF)
      ENTRY FOMATCH
      IF (IMATCH .EQ. 0) GO TO 16
      ID = 1
      RHO(ID) = RHOM
   16 CONTINUE
      DO 500 ID=1,ND
      PRINT 9000
      PRINT 4000, RHO(ID),RFIMIN
      WRITE(7,4000) RHO(ID),RFIMIN
      NR = 1
   20 CONTINUE
      DO 100 I=1,NITMAX
      PRINT 9000
      CALL GETAERO(NM,NMAX,RFIMIN,0,RFIL,RFIR,DEL)
      IF (ID .EQ. 1 .AND. IMATCH .LE. 1) GO TO 40
      DM = RHO(ID)/REFRHO
      DO 30 IA=1,NM
      DO 30 JA=1,NM
   30 AF(IA,JA) = DM * AF(IA,JA)
   40 CALL EIGSOL(NM,NMAX,NEIG,NVEC,IPRT)
      CALL GETDAER(NM,NMAX,RFIMIN,1,RFIL,RFIR,DEL)
      IF (ID .EQ. 1 .AND. IMATCH .LE. 1) GO TO 60
      DO 50 IA=1,NM
      DO 50 JA=1,NM
      DAF(IA,JA) = DM * DAF(IA,JA)
   50 SDAF(IA,JA) = DM * SDAF(IA,JA)
   60 CALL DERF(2,NM,NMAX,NVEC,RFIMIN,IPRT)
         RFI(1) = RFIMIN
      CALL LEGROOT(NEIG,ICON,RFI,G,IAR,IPRT,NL)
```

```
      PRINT 1000,I,RFIMIN,(RFI(J),J=1,NEIG)
      IF (NEVRED .EQ. 0) GO TO 80
      NEVRED = 0
      IM = IAR(NR)
      MUM = EIG(IM)
      IM1 = IM + 1
      DO 70 J=IM1,NM
      D = MUM/EIG(J)
      IF (D .GE. EIGRAT) GO TO 75
   70 CONTINUE
      NEIG = NM
      NVEC = NM
      GO TO 80
   75 NEIG = J-1
      NVEC = J-1
      PRINT 6000,NVEC
      WRITE(7,6000) NVEC
   80 IF ( ABS(RFIMIN-RFI(NR)) .LE. DEL2 ) GO TO 110
      RFIMIN = RFI(NR)
      IF (RFIMIN .GT. RFIL .AND. RFIMIN .LT. RFIR) GO TO 100
      NR1 = NR - 1
      PRINT 7000, NR1,RFIMIN
      WRITE(7,7000) NR1,RFIMIN
      IF (NR1 .GT. 0) GO TO 400
      STOP
  100 CONTINUE
      PRINT 1500, NR,I
      STOP
  110 CONTINUE
      IM = IAR(NR)
      PRINT 5000, IM,(EIG(I),I=1,NEIG)
      PRINT 5500,(IAR(I),I=1,NEIG)
      VEL(NR) = BR*SQRT(1.0/REAL(EIG(IM)) )*RFIMIN
      IF (NR .NE. 1) GO TO 200
  120 CONTINUE
      REWIND 4
      WRITE(4) IM,RFIMIN,VEL(NR),TP2(IM,IM),EIG,VEC,AVEC,AF,TP
      NFROOT = NR
      SVEL = VEL(NR)
  200 IF (VEL(NR) .GE. VEL(NFROOT)) GO TO 300
      GO TO 120
  300 PRINT 3000, NR,VEL(NR),RFIMIN,I
      WRITE(7,3000) NR,VEL(NR),RFIMIN,I
  310 CONTINUE
      IF ( NR .EQ. NROOT ) GO TO 400
      NR = NR+1
      IF ( ABS(RFIMIN-RFI(NR)) .GT. DEL2 ) GO TO 350
      RFI(NR) = RFIMIN
      WRITE(7,8000) RFI(NR),DEL2
      PRINT 8000, RFI(NR),DEL2
      IM = IAR(NR)
      VEL(NR) = BR*SQRT(1.0/REAL(EIG(IM)) )*RFIMIN
      IF (VEL(NR) .GE. VEL(NFROOT)) GO TO 320
      REWIND 4
      WRITE(4) IM,RFIMIN,VEL(NR),TP2(IM,IM),EIG,VEC,AVEC,AF,TP
      NFROOT = NR
      SVEL = VEL(NR)
  320 CONTINUE
      PRINT 3000, NR,VEL(NR),RFIMIN,I
      WRITE(7,3000) NR,VEL(NR),RFIMIN,I
      GO TO 310
  350 CONTINUE
         RFIMIN = RFI(NR)
      IF (RFIMIN .GE. RFIL .AND. RFIMIN .LE. RFIR) GO TO 20
      NR1 = NR - 1
      PRINT 7000, NR1,RFIMIN
      WRITE(7,7000) NR1,RFIMIN
```

```
 400 CONTINUE
     IF (IMATCH .EQ. 0) RFIMIN = RFI(1)
     IF(IMATCH .NE. 0) GO TO 600
 500 CONTINUE
 600 CONTINUE
     RETURN
1000 FORMAT(//10H ITERATION,I3,/,F12.4,8X,5F12.4,(/20X,5F12.4))
1500 FORMAT(/,* PROGRAM TERMINATED, COULD NOT FIND ROOT NO. *,I3, * IN*
    1,I3,* ITERATIONS*)
3000 FORMAT(//,13H ROOT NUMBER ,I3,14H , VELOCITY = ,F9.3,
    1    9H , RFI = ,F8.3,29H , NO. OF ITERATIONS REQD, = ,I2)
4000 FORMAT(///,11H DENSITY = ,F15.6,12H , RFIMIN = ,F12.4)
5000 FORMAT(/,26H FLUTTER EIGENVALUE NO. = ,I3,13H, EIGENVALUES,/,
    1 (10F12.4/))
6000 FORMAT(/,91H OPTION TO REDUCE NO. OF EIGENVALUES AND EIGENVECTORS
    1EXERCISED, NEIG AND NVEC SET EQUAL TO,I3)
5500 FORMAT(/,62H RFI FOR PREDICTED CROSSINGS CORRESPOND TO EIGENVALUES
    1 NUMBERS,(/12I5))
7000 FORMAT(/,6H FOUND,I3,42H ROOTS, RFI FOR THE NEXT ROOT PREDICTED =
    1,F10.4,21H ,IS BEYOND THE RANGE)
8000 FORMAT(/,35H RFI PREDICTED FOR THE NEXT ROOT = ,F10.4,
    1 52H, DIFFERENCE FROM RFI FOR PREVIOUS ROOT IS LESS THAN,F8.4)
9000 FORMAT(1H1)
     END
```

Subroutine DERF

```
      SUBROUTINE DERF(ND,NM,NMAX,NVEC,RFI,IPRT)
      COMMON/BLK1/ AF,DAF,SDAF
      COMMON/BLK2/ EIG,VEC,AVEC,DIFS,SDIFS
      COMMON/BLK3/ TP,TP2
      COMPLEX AF(12,12),DAF(12,12),SDAF(12,12),EIG(12),VEC(12,12),
     1        AVEC(12,12),DIFS(12),SDIFS(12),TP(12,12),TP2(12,12),A,B,C,
     2        MUM
C*********************************************************************
C*                                                                   *
C*    ND = NUMBER OF DERIVATIVES REQUIRED, 1 OR 2.                   *
C*    NM = NUMBER OF MODES.                                          *
C*    NMAX = MAXIMUM NUMBER OF MODES DEFINING SIZE OF VARIOUS ARRAYS. *
C*    NVEC = NUMBER OF FREQUENCIES FOR WHICH THE DERIVATIVES COMPUTED. *
C*    RF = REDUCED FREQUENCY                                         *
C*    EIG(NMAX)  VECTOR OF INVERSE OF FREQUENCIES SQUARED.           *
C*    VEC(NMAX,NMAX) ARRAY OF EIGENVECTORS, ONE PER COLUMN.          *
C*    AVEC(NMAX,NMAX) ARRAY OF ASSOCIATED EIGENVECTORS, ONE PER COLUMN.*
C*    AF(NMAX,NMAX)  AIR FORCE MATRIX.                               *
C*    DAF(NMAX,NMAX) FIRST DERIVATIVE OF AF W.R.T. REDUCED FREQUENCY. *
C*    SDAF(NMAX,NMAX) SECOND DERIV. OF AF W.R.T. REDUCED FREQ.       *
C*    DIFS(NMAX) FIRST DERIV. OF EIG(NMAX) W.R.T. 1/RF.              *
C*    SDIFS(NMAX) SECOND DERIV. OF EIG(NMAX) W.R.T. 1/RF.            *
C*    TP(NMAX,NMAX),TP2(NMAX,NMAX) TEMPORARY STORAGE ARRAYS.         *
C*    COMPUTES DERIVATIVES OF INVERSE OF FREQUENCY W.R.T. 1/REDUCED  *
C*        FREQUENCY WHERE LAMBDA=FREQ**2 IS DEFINED BY THE EQUATION  *
C*        (STIFFNESS - LAMBDA(MASS+AF))VEC = 0                       *
C*    KUMAR G. BHATIA, JUNE  8,1972.                                 *
C*********************************************************************
C*                                                                   *
C     COMPUTE THE REQUIRED ELEMENTS OF TRIPLE PRODUCT MATRIX
C        AVEC TRASPOSED * DAF * VEC.
      CALL TMMPROD(AVEC,DAF,VEC,NM,NVEC,NMAX, 2,TP)
C     COMPUTE THE FIRST DERIVATIVES OF 1./FREQ**2 W.R.T. 1/RF
      RF = 1.0/RFI
      DO 10 M=1,NVEC
   10 DIFS(M) = -RF*RF*EIG(M)*TP(M,M)
      IF ( ND .EQ. 1 ) RETURN
```

```
C     COMPUTE THE SECOND DERIVATIVES OF 1./FREQ**2 W.R.T. 1/RF
      CALL TMMPROD(AVEC,SDAF,VEC,NM,NVEC,NMAX,1,TP2)
      RF4 = RF**4
      DO 100 M=1,NVEC
         MUM = EIG(M)
         A = -2.0*RF*DIFS(M)
         B = 0.0
         DO 20 L=1,NM
            IF (L .EQ. M) GO TO 20
            B = B + TP(L,M)*TP(M,L)/(1.0 - MUM/EIG(L))
   20    CONTINUE
      B = - 2.0*B*RF4*MUM
         C = RF4*MUM*TP2(M,M)
         SDIFS(M) = A + B + C
      IF (IPRT .EQ. 2) PRINT 1000,M,DIFS(M),SDIFS(M),A,B,C
  100 CONTINUE
 1000 FORMAT(// 48H DERIVATIVES OF INVERSE OF FREQ. SQUARED, NUMBER,I2,
     1        /, 17H FIRST DERIVATIVE,E15.3,E13.3,/,18H SECOND DERIVATIVE
     2/E13.3,E12.3,1H=,E13.3,E12.3,1H+,E13.3,E12.3,1H+,E13.3,E12.3)
      RETURN
      END
```

Subroutine TMMPROD

```
      SUBROUTINE TMMPROD(A,D,V,N,NV,NMAX,ND,R)
      COMPLEX A(NMAX,1),D(NMAX,1),V(NMAX,1),R(NMAX,1),TEMP
C               T
C     COMPUTES A *D*V = R . IF ND=1 THEN ONLY DIAGONALS ARE COMPUTED
      DO 100 I=1,NV
      DO 100 J=1,NV
      IF (ND .EQ. 1 .AND.I .NE. J) GO TO 100
      R(I,J) = 0.0
      DO 50 K =1,N
         TEMP = 0.0
         DO 40 L=1,N
   40       TEMP = TEMP + D(K,L)*V(L,J)
   50    R(I,J) = R(I,J) + A(K,I)*TEMP
  100 CONTINUE
      RETURN
      END
```

Subroutine EIGSOL

```
      SUBROUTINE EIGSOL(NM,NMAX,NEIG,NVEC,IPRT)
      COMMON/BLK1/ AF,H,HL
      COMMON/BLK2/ EIG,VEC,AVEC,CNT,COLM
      COMMON/BLK3/ TP,TP2
      COMMON/BLK4/ SM,C,INTH
      COMPLEX AF(12,12),H  (12,12),HL  (12,12),EIG(12),VEC(12,12),
     1        AVEC(12,12),CNT(12),COLM(12),TP(12,12),TP2(12,12),SUM,SUM1
      DIMENSION SM(12,12),C(12,12),INTH(12,2)
C
C                                    T
C     COMPUTE THE PRODUCT C*(SM+AF )
      DO 10 I=1,NM
      DO 10 J=1,NM
         TP(I,J) = 0.0
         DO 5 K=1,NM
    5       TP(I,J) = TP(I,J) + C(I,K)*(SM(K,J)+AF(J,K))
```

```
   10 CONTINUE
      INTH(1,1) = NM
      INTH(2,1) = NVEC
      CALL EECM(TP,EIG,AVEC,HL,H,CNT,COLM,INTH,NMAX)
      IF (INTH(1,1) .EQ. NM) GO TO 15
      PRINT 1000, INTH(1,1)
      STOP
   15 CONTINUE
C     COMPUTE TP2 = SM+AF , AND C*(SM+AF)
      DO 20 I=1,NM
      DO 20 J=1,NM
   20 TP2(I,J) = SM(I,J)+AF(I,J)
      DO 30 I=1,NM
      DO 30 J=1,NM
         TP(I,J) = 0.0
         DO 25 K=1,NM
   25       TP(I,J) = TP(I,J) + C(I,K)*TP2(K,J)
   30 CONTINUE
      INTH(1,1) = NM
      INTH(2,1) = NVEC
      CALL EECM(TP,EIG,VEC,HL,H,CNT,COLM,INTH,NMAX)
      IF (INTH(1,1) .EQ. NM) GO TO 40
      PRINT 1000, INTH(1,1)
      STOP
   40 CONTINUE
C     NORMALIZE VEC
      DO 50 J=1,NVEC
         SUMR = 0.0
         DO 45 I=1,NM
   45    SUMR = SUMR + REAL(VEC(I,J))**2 + AIMAG(VEC(I,J))**2
         SUMR = SQRT(SUMR)
      DO 50 I=1,NM
         VEC(I,J) = VEC(I,J)/SUMR
   50 CONTINUE
C     NORMALIZE AVEC
      DO 70 J=1,NVEC
         SUM = 0.0
         DO 60 I=1,NM  $ SUM1 = 0.0
            DO 55 L=1,NM
   55       SUM1= SUM1+ TP2(I,L)*VEC(L,J)
   60    SUM = SUM + AVEC(I,J)*SUM1
      DO 70 I=1,NM
         AVEC(I,J) = AVEC(I,J)/SUM
   70 CONTINUE
      IF (IPRT .EQ. 0) GO TO 100
      PRINT 2000, (EIG(I),I=1,NEIG)
      IF (IPRT .EQ. 1) GO TO 100
      PRINT 3000,((VEC(I,J),J=1,NVEC),I=1,NM)
      PRINT 4000,((AVEC(I,J),J=1,NVEC),I=1,NM)
  100 CONTINUE
      RETURN
 1000 FORMAT(//,31H NUMBER OF EIGENVALUES COMPUTED,I5)
 2000 FORMAT(//,12H EIGENVALUES/(1X,5(E14.3,E12.3)))
 3000 FORMAT(//,13H EIGENVECTORS,/,(1X,5(E14.3,E12.3)))
 4000 FORMAT(//,24H ASSOCIATED EIGENVECTORS,/,(1X,5(E14.3,E12.3)))
      END
```

APPENDIX C

USAGE DESCRIPTION OF LANGLEY LIBRARY SUBROUTINES USED BY PROGRAM MATCH

Usage descriptions of the Langley library subroutines used by program MATCH are presented in this appendix.

Langley Library Subroutine MATINV

Language: FORTRAN

Purpose: MATINV solves the matrix equation AX = B where A is a square coefficient matrix and B is a matrix of constant vectors. The solution to a set of simultaneous equations, the matrix inverse, and the determinant may be obtained. If the user does not want the inverse, use SIMEQ for savings in time and storage. For the determinant only, use DETEV.

Use: CALL MATINV (A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)

A — A two-dimensional array of the coefficients. On return to the calling program, A^{-1} is stored in A.

N — The order of A; $1 \leq N \leq NMAX$.

B — A two-dimensional array of the constant vectors B. On return to calling program X is stored in B.

M — The number of column vectors in B. M = 0 signals that the subroutine is used solely for inversion, however, in the call statement an entry corresponding to B must still be present.

DETERM — Gives the value of the determinant by the following formula:
$$DET(A) = \left(10^{100}\right)^{ISCALE}(DETERM)$$

IPIVOT A one-dimensional array of temporary storage used by the routine.

INDEX A two-dimensional array of temporary storage used by the routine.

NMAX The maximum order of A as stated in the dimension statement of the calling program.

ISCALE A scale factor computed by the subroutine to keep the results of computation within the floating point word size of the computer.

Restrictions: Arrays A, B, IPIVOT, and INDEX are dimensioned with variable dimensions in the subroutine. The maximum size of these arrays must be specified in a DIMENSION statement of the calling program as: A (NMAX,NMAX), B (NMAX,M), IPIVOT (NMAX), INDEX (NMAX, 2). The orginal matrices, A and B, are destroyed. They must be saved by the user if there is further need for them. The determinant is set to zero for a singular matrix.

Method: Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations: $\ell_n, \ell_{n-1}, \ldots, \ell_1$. A = I. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the results are A^{-1} and X where AX = B. Each transformation is selected so that the largest element is used in the pivotal position.

Accuracy: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is.

Reference: Fox, L.: AN INTRODUCTION TO NUMERICAL LINEAR ALGEBRA

Storage: 542_8 locations.

Subroutine OPENMS

Language: COMPASS

Purpose: To open a random access file.

Use: CALL OPENMS (U,IX,L,P)

where

U	The logical unit number.
IX	The first word address of the index.
L	The length of the index.
P	P = 0 for numbered indexing. P = 1 for named indexing.

Restrictions: OPENMS must be the first operation on a random access file. The file must be a disk file. For n index entries, the length of the index must be at least $2n + 1$ if using named indexing, whereas the index length must be at least $n + 1$ for numbered indexing.

Method: OPENMS sets the first word in the index to a positive number for numbered indexing or to a negative number for named indexing. The random access bit, index address, and index length are set by OPENMS into the FET of the file for system communication. If the file already exists, the master index is read into central memory.

Accuracy: Not applicable.

References: None.

Storage: 103_8 locations.

Subprograms used: GETBA, SIO$, SYSTEM

Error messages: (1) UNASSIGNED MEDIUM FILE XXXXXX

(2) FILE DOES NOT RESIDE ON A RANDOM ACCESS DEVICE, XXXXXX

(3) INDEX BUFFER IS OF INSUFFICIENT LENGTH, XXXXXX

XXXXXX is the file name. Termination is abnormal in each case.

Subroutine WRITMS

Language: COMPASS

Purpose: To write a record on a random access file.

Use: CALL WRITMS (U,FWA,N,I)

where

U	The logical unit number.
FWA	The central memory address of the first word of the record.
N	The number of central memory words to be transferred.
I	The record number or record name depending upon the indexing mode set by the initial call to OPENMS.

Restrictions: The file must have been opened by a call to OPENMS.

Method: The specified record is written on the file and an address entered in the index to reference the record.

Accuracy: Not applicable.

References: None.

Storage: 102_8 locations.

Subprograms used: GETBA, SYSTEM, SIO$

Error messages: (1) UNASSIGNED MEDIUM, FILE XXXXXX

(2) FILE WAS NOT OPENED BY A CALL TO SUBROUTINE OPENMS

(3) INDEX BUFFER IS OF INSUFFICIENT LENGTH.

Subroutine READMS

Language: COMPASS

Purpose: To read a record on a random access file.

Use: CALL READMS (U,FWA,N,I)

where

U	The logical unit number.
FWA	The central memory address of the first word of the record.
N	The number of words of the record to be transferred.
I	The record number or record name depending upon the indexing mode set by the initial call to OPENMS.

Restrictions: The file must have been opened by a call to OPENMS.

Method: The disk address of the record is determined by using the index. If n words are requested to be transferred and there are m words in the record, where $m \leqq n$, m words are transferred. If $m > n$, n words are transferred.

Accuracy: Not applicable.

References: None.

Storage: 131_8 locations.

Subprograms used: GETBA, SYSTEM, SIO$

Error messages: (1) UNASSIGNED MEDIUM, FILE XXXXXX

(2) FILE WAS NOT OPENED BY A CALL TO SUBROUTINE OPENMS

(3) RECORD NAME REFERRED TO IN CALL IS NOT IN THE FILE INDEX

(4) *READ PARITY ERROR*

(5) SPECIFIED INDEX IN THIS MASS STORAGE CALL .GT. MASTER INDEX OR IS ZERO.

Termination is abnormal.

Subroutine EECM

Language: FORTRAN

Purpose: To compute eigenvalues and eigenvectors of a complex N by N matrix.

Use: CALL EECM (A,LAMBDA,VECT,HL,H,CNT,COLM,INTH,MAX)

A — A two-dimensional complex array of the input matrix. It is not destroyed.

LAMBDA — A one-dimensional complex array of eigenvalues. They are arranged in descending order of absolute magnitude.

VECT — A two-dimensional complex array of eigenvectors. Each vector is normalized so that the sum of the squares of the moduli of the components is unity.

HL,H — Two-dimensional complex temporary arrays.

CNT,COLM — One-dimensional complex temporary arrays.

INTH — A two-dimensional integer array.

Upon entry – Before each CALL, set INTH as follows:

INTH(1, 1) = N = order of matrix A.

INTH(2, 1) = NV = number of eigenvectors to be computed.

Upon return

INTH(1, 1) = the actual number of eigenvalues computed.

INTH(2, 1) is destroyed.

MAX — An integer, the maximum order of A.

Restrictions: The calling program must type the following complex arrays and dimension them as follows: A(MAX, MAX), LAMBDA(MAX), VECT(MAX, NV), HL(MAX, MAX), H(MAX, MAX), CNT(MAX), COLM(MAX). The integer array is dimensioned INTH(MAX, 2).

Before each CALL to EECM, N and NV must be stored in the first 2 locations of INTH (see Use).

The column dimension, NV, for VECT may be ≦N. If no vectors are to be computed (INTH(2, 1) = 0), VECT need not be dimensioned, but it must appear as an argument in the call statement.

The eigenvalues are not necessarily calculated in any absolute order, but are arranged in descending order of absolute magnitude prior to the calculation of the eigenvectors. Ten iterations per eigenvalue are allowed. In case of nonconvergence, the subroutine will return a value less than the order of the input matrix in INTH(1, 1). Thus, the user should test INTH(1, 1) upon return. If, then, it is less than the value of the number of vectors asked for, only that number of eigenvalues and eigenvectors is computed. If the number of eigenvalues computed is less than the order of the input matrix, the programer may want to use arbitrary shifts on the input matrix, or add a constant to the diagonal. Either change may eliminate the difficulty. Matrices apt to get nonconvergence are lower triangular with all equal eigenvalues, those with ones on the lower diagonal, and those with one as the Nth component of the first row and zeros elsewhere.

If overflows or underflows occur, scaling the input matrix so that its largest element is in modulus about 1 will probably eliminate the difficulty.

Equal computed eigenvalues return identical corresponding eigenvectors even though linearly independent vectors may exist.

Method: The input matrix A is reduced to an upper Hessenberg matrix H by a sequence of elementary triangular and permutation matrices which make up a matrix P such that $P^{-1}AP = H$. The QR algorithm is made use of in EECM by applying unitary similarity transformations to Hessenberg matrices, H_i: $H_1 = P^{-1}AP$, $H_s = (h_{ij}^{(s)}) = Q_sT_s$, $H_{s+1} = Q_s^H H_s Q_s = Q_s^H Q_s T_s Q_s = T_s Q_s$ where Q_s^H is the product of plane rotations, chosen so that T_s is upper triangular. This process makes $h_{n,n-1}^{(s)}$ converge to zero and therefore $h_{nn}^{(s)}$ converges to an eigenvalue of A. When convergence is met ($h_{n,n-1}^{(s)}$ negligible), the Hessenberg matrix H_s is deflated (i.e., last row and column eliminated) and EECM proceeds with its leading principal submatrix (a new H_1) of order one less. If $h^{(s)}_{n-1,n-2}$ becomes negligible, the eigenvalues of the lower right-hand matrix of order two are calculated and EECM proceeds with the leading principal submatrix of order two less. It can be shown that convergence is accelerated by judiciously subtracting scalar matrices from the H_s matrices. EECM actually replaces H_s by $H_s - k_sI$ so that k_s is one of the eigenvalues p_s or q_s of the lower right-hand 2×2 matrix of H_s. The choice of p_s or q_s is made on the basis of whether $\left|h^{(s)}_{nn} - p_s\right|$ or $\left|h^{(s)}_{nn} - q_s\right|$ is a minimum. The shift technique is applied at each iteration.

Two passes of the Wielandt inverse power method are used to calculate the eigenvectors, Y_i of H. Very little work is required for the second pass since the necessary elementary triangular and permutation matrices are stored in COLM and INTH(col. 2) (both internal storage areas). Finally, the eigenvectors of A, P Y_i are calculated. The matrix P is in INTH (col. 1) and the lower part of H (internal arrays).

The theory and a complete description of the algorithms appear in the first reference.

Accuracy: The accuracy obtainable in computing the eigenvalues of input matrix A is usually related to the spectral radius, rho(A), of matrix A or more generally to some norm of A times the norm of its inverse. Hence, the greater rho(A)/min(abs(LAMBDA(1))), the fewer significant digits the smaller eigenvalues may have. Accuracy also decreases as the order of the matrix increases. Close eigenvalues are usually less accurate than well separated ones.

References: Wilkinson, J. H.: The Algebraic Eigenvalue Problem. Clarendon Press (Oxford), 1965.

Householder, Alston Scott: The Theory of Matrices in Numerical Analysis. First ed., Blaisdell Pub. Co., 1964.

Storage: 2745_8 locations.

Subprograms used: None

Timing: On Control Data 6000 computer, time for the actual solution of all eigenvalues and eigenvectors of a 30 by 30 matrix was 5.2 seconds. This was about 5 times faster than routines presently in the Langley library.

REFERENCES

1. Bhatia, Kumar G.: An Automated Method for Determining the Flutter Velocity and the Matched Point. AIAA Paper No. 73-195, Jan. 1973.

2. Anon.: U.S. Standard Atmosphere, 1962. NASA, U.S. Air Force, and U.S. Weather Bur., Dec. 1962.

NASA-Langley, 1973 —— 32 L-9128