NASA-CR-135295
PWA-5554-5

NASA

METHOD OF FAN SOUND MODE STRUCTURE DETERMINATION COMPUTER PROGRAM USER'S MANUAL MOD/.L CALCULATION PROGRAM

by

G. F. Pickett, R. A. Wells and R. A. Love

August 1977

PRATT & WHITNEY AIRCRAFT GROUP
Commercial Products Division

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA-Lewis Research Center
Contract NAS3-20047

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-135295		
4. Title and Subtitle Method of Fan Sound Mode Structure Determination Computer Program User's Manual - Modal Calculation Program		5. Report Date August 1977
		6. Performing Organization Code
7. Author(s) G. F. Pickett, R. A. Wells and R. A. Love		8. Performing Organization Report No. PWA 5554-5
		10. Work Unit No.
9. Performing Organization Name and Address Pratt & Whitney Aircraft Group Commercial Products Division United Technologies Corporation East Hartford, Connecticut 06108		11. Contract or Grant No. NAS3-20047
		13. Type of Report and Period Covered Contract Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This computer user's manual describes the operation and the essential features of the Modal Calculation Program, the second of two programs developed under the Method of Fan Sound Structure Determination Program, NAS3-20047. Jointly the two programs are used to determine the coherent modal structures of inlet sound fields. The purpose of the Modal Calculation Program is to calculate the amplitude and phase of modal structures by means of acoustic pressure measurements obtained from microphones placed at selected locations within the fan inlet duct. These locations are determined by the first of the two programs. In addition, the Modal Calculation Program also calculates the first-order errors in the modal coefficients that are due to tolerances in microphone location coordinates and inaccuracies in the acoustic pressure measurements.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
fan sound mode structures amplitude phase coherent sound modes extraneous sound modes		Unclassified – Unlimited	
19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	75	

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

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

METHOD OF FAN SOUND MODE STRUCTURE DETERMINATION
COMPUTER PROGRAM USER'S MANUAL
MODAL CALCULATION PROGRAM

by

G. F. Pickett, R. A. Wells and R. A. Love
Pratt & Whitney Aircraft Group

1.0 SUMMARY

This computer user's manual describes the operation and the essential features of the Modal Calculation Program, the second of two programs developed under the Method of Fan Sound Structure Determination Program, NAS3-20047. Jointly the two programs are used to determine the coherent modal structures of inlet sound fields. The purpose of the Modal Calculation Program is to calculate the amplitude and phase of modal structures by means of acoustic pressure measurements obtained from microphoes placed at selected locations within the fan inlet duct. These locations are determined by the first of the two programs. In addition, the Modal Calculation Program also calculates the first-order errors in the modal coefficients that are due to tolerances in microphone location coordinates and inaccuracies in the acoustic pressure measurements.

2.0 INTRODUCTION

New fan designs for modern high bypass ratio commercial engines utilize blade-vane interaction theory to the extent possible for contolling the propagation of interaction noise. Currently, this theory defines the modes that can propagate, but has not been developed to the extent that it can reliably predict the strengths of the propagating modes.

Further noise reduction could be achieved if the propagating modal structure were quantified. Once the modal structure were defined, an analytical system for acoustic-treatment design could be utilized to optimize treatment for a given modal structure, to produce more efficient schemes. In addition, the modal structure could be employed to verify developing theories of fan noise generation. To provide this capability by means of measured data the Method of Fan Sound Mode Structure Determination Program (NAS3-20047) was undertaken. The method would be utilized until a valid fan noise generation model on a model basis becomes available.

The theory upon which fan spinning mode theory is founded was presented in 1961 by Tyler and Sofrin (ref. 1), following extensive analytical and experimental studies. Later, Sofrin and McCann (ref. 2) derived the general form of a coherent acoustic wave in an infinitely long cylindrical duct which extended the theory to include effects of axial flow. This equation expresses the coherent acoustic pressure at locations in the duct as a function of the amplitude and phase of the propagating modes comprising the sound field. These purely coherent signals, which are due to the contributions of the constituent modes, are extracted from the overall signal by enhancement techniques adapted at Pratt & Whitney Aircraft – the advantages of utilizing signal enhancement is discussed by Posey in reference 3.

Both the analytical expression derived for a general coherent acoustic wave and a signal enhancement technique form the basis for developing a method to determine fan sound mode structures. The method, in principle, is capable of determining the amplitude and phase of all modes that can propagate at a given frequency. In practice, the number of modes that can be determined is limited by the storage capacity and the running time of the computer and by measurement and location accuracy.

The method for determining fan sound mode structure (ref. 4) requires two computer programs: a Microphone Location Program (MLP) and a Modal Calculation Program (MCP). This User's Manual describes the MCP; the MLP is presented in a companion Manual.

The MLP identifies microphone locations in the duct for measuring acoustic pressures for input to the MCP that will insure a numerically stable solution. The MCP calculates modal structures from acoustic pressure measurements and calculates coefficients that can be used to determine the sensitivity of the modal calculation procedure to first-order errors in acoustic pressure measurements and microphone placement.

In the following sections, the algorithm for the modal calculations and the program elements – such as subroutines, functional elements, and principal element interrelationships – are discussed. A description of the input parameters is included. The output format is also described and illustrated by a sample case. Finally, a listing of the program code is provided in Appendix B.

3.0 PROGRAM DESCRIPTION

3.1 ALGORITHM

The Modal Calculation Program is an algorithm for calculating the modal structure from input data comprising acoustic pressure measurements and a finite set of modes. The general form of any coherent acoustic wave in an infinitely long cylinderical duct having uniform axial flow can be written as the real part of

$$P(x, r, \theta; t) = \sum_{\substack{\text{Finite Set}\\ \text{of Modes}}} A_{m,\mu} \, E(k^{\sigma}_{m\mu} r) \, e^{i [kx X_n + m\theta_n - \omega t + \Phi_{m,\mu}]}$$

and

$$Kx = \frac{Mx\,(\omega/c) \pm \sqrt{(\omega/c)^2 - (1 - Mx^2)\, k^{\sigma\,2}_{m\mu}}}{1 - M_x^2}$$

where the notation is consistent with reference 1.

Equation 1 can be written in matrix form where the measured pressures are obtained from microphone locations identified by the MLP. The equation system is solved in the usual manner by matrix inversion. The output from this procedure is the amplitude and phase of the coherent acoustic duct modes comprising the inlet sound field.

The input to the program consists of: the sound field in the duct comprising N acoustic duct modes, the geometric parameters (e. g. duct radius, hub-tip ratio), test parameters (e. g. frequency, axial Mach number, speed of sound), and measured acoustic pressure amplitude and phase at locations identified by the MLP. The characteristic numbers that include the eigen value $k'^{\sigma}_{m,\mu}$, the axial wave number kx, and the value of the eigen function $E(k^{\sigma}_{m,\mu} r)$ are calculated.

In addition, this equation requires the input of acoustic pressure measurements, the number of which exactly equal the number of specific modes. A set of equations can then be established with the number of equations equaling the number of acoustic measurements. This set was written in matrix form with the matrix coefficients a function of the particular modes comprising the sound field and the microphone locations.

If the determinant of the equation system is non-zero, a set of independent equations exists. This equation system in principle can be inverted in the usual way to solve for the unknown amplitude and phase of the particular modes comprising the sound field. A Gaussian elimination procedure is used to reduce the equation system to a triangularized matrix for solution of the complex modal coefficients. The overall pressure at any location in the duct can be calculated from the information in the modal structure.

Once the modal structure has been determined, a set of influence coefficients (ref. 4) is calculated. These coefficients can be used to determine the errors in modal amplitudes and phases that are the results of first-order inaccuracies in measured pressures and the tolerances in microphone placement.

As an option, the MCP can also calculate the resultant sound field at any specified duct location based on a given modal structure. This modal structure is supplied by the user either arbitrarily or as output from an analytical prediction deck.

3.2 PROGRAM OVERVIEW

The Modal Calculation Program comprises six major sections which are utilized in part or whole to accomplish the objectives of the two possible modes of operation. These six major sections are:

1) Input – The input of all data is by the NAMELIST specification, and the internal parameters are initiated for program execution.

2) Characteristic Number Calculation – The characteristic numbers $K'^{\sigma}_{m,\mu}$ and $Q^{\sigma}_{m,\mu}$ are calculated using the procedure described in Appendix A.

3) Mode Amplitude and Phase Calculation – The coherent acoustic wave equation system,. e.g. (1), in an infinitely long cylinderical duct with uniform axial flow is solved using a Gaussian elimination procedure for the modal amplitude and phase.

4) Sensitivity Coefficient Calculation – Standard deviations due to the first-order independent errors in the measurement of both the acoustic pressures and the microphone coordinates are obtained for the error in the modal amplitudes and phases.

5) Overall Pressure Calculation – Resultant pressure amplitude and phase are calculated at the desired prediction locations using the amplitude and phase of the constituent modes comprising the sound field.

6) Output – All results from the program calculations are printed.

The interrelationships between the six major sections and their utility for each option is illustrated in Figure 1. As input, both options require a specific mode group, inlet geometry, and test condition to calculate characteristic numbers. One option, "A", requires additional input in the form of acoustic pressure signals at selected duct locations to calculate the modal structure comprising the sound field. Additionally, influence coefficients, which are functions of the modal structure, are calculated. The other option, "B", requires that the modal structure be specified as input. In both options, the amplitude and phase of the constituent modes are utilized to calculate the overall acoustic pressure at any duct location. The results from both options are printed by the output section.

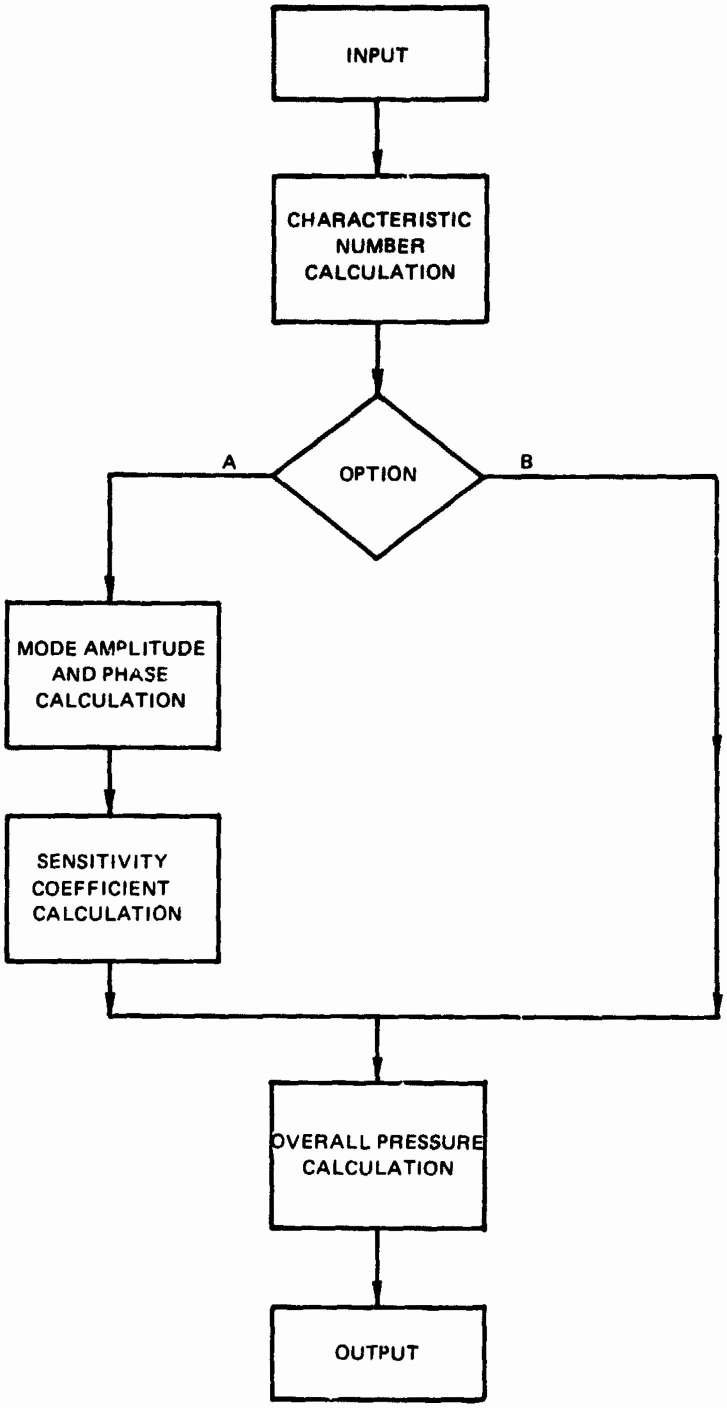

Figure 1 Program Overview

3.3 PROGRAM SUBROUTINES AND FUNCTIONS DESCRIPTION

The subroutines and functions used in the six program sections presented in Section 3.2 are listed below; the purpose of each subroutine or function is described. Also as appropriate, principal-element diagrams of the more complicated sections are presented and discussed.

Input Section

The NAMELIST format is used to input data for execution of the computer program. This form of input is described in Section 3.4.1. The input variable names are listed in Section 3.4.2, including a description of their purpose. All input is read into the program by the following subroutine:

INPUT – This subroutine inputs data for each case and sets up the necessary internal parameters.

Characteristic Number Calculation Section

Expressions are derived in Appendix A for solving two simultaneous equations that define the characteristic numbers $k'^{\sigma}_{m\mu}$ and $Q^{\sigma}_{m\mu}$. A principal-element diagram is presented in Figure 2 to illustrate the functional elements that lead to a determination of these numbers. Initially, the order of the Bessel functions is determined from the circumferential order of a particular mode. The J_m and Y_m Bessel functions are evaluated, as appropriate, depending on the value of the duct hub-tip ratio. Finally, the characteristic numbers are calculated by solving the simultaneous equations comprising the Bessel functions. The subroutines and functions utilized in this section are:

KQCAL – This subroutine calculates the characteristic numbers $K'^{\sigma}_{m\mu}$ and $Q^{\sigma}_{m\mu}$.

KMUCAL – This subroutine is used by KQCAL to calculate the characteristic number $k'^{\sigma}_{m\mu}$.

EMUCAL – This subroutine calculates characteristic E-function values for a particular radial value, $r' = r/b$.

FALZIP – This function solves for a root of a given function using a combination of false position and bisection techniques

BESL1 – This function is used by KMUCAL to calculate values of $K'^{\sigma}_{m\mu}$ for the equation which defines the system of differential equations.

$$\frac{d}{dr'}[J_m(K'^{\sigma}_{m\mu})] + Q^{\sigma}_{m\mu}\frac{d}{dr'}[Y_m(K'^{\sigma}_{m\mu})] = 0$$

$$\frac{d}{dr'}[J_m(\sigma K'^{\sigma}_{m\mu})] + Q^{\sigma}_{m\mu}\frac{d}{dr'}[Y_m(\sigma K'^{\sigma}_{m\mu})] = 0$$

for a hub-tip ratio not equal to zero.

BESL2 – This function is used by KMUCAL to calculate values of $K'^{\sigma}_{m\mu}$ for the equation which defines the above system of differential equations for a hub-tip ratio equal to zero.

BESJ – This subroutine calculates values of the Bessel function of the first kind.

BESY – This subroutine calculates values of the Bessel function of the second kind.

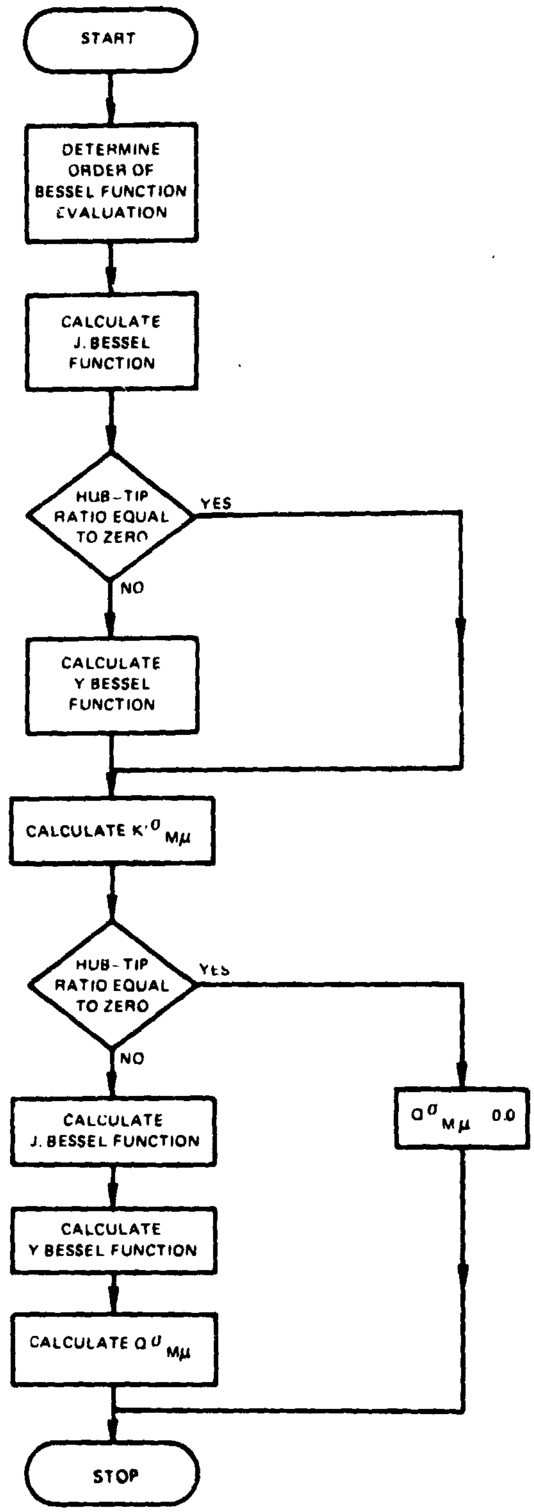

Figure 2 Principal-Element Diagram – Characteristic Number Calculation Section

Mode Amplitude and Phase Calculation

The modal amplitude and phase are solved by matrix inversion techniques from data that includes pressure measurements at selected microphone locations. The equations that define the matrix coefficients and a description of the procedure for fan sound mode determination was presented in Section 3.1 - Algorithm. To illustrate the functional elements that lead to a solution of the modal coefficients, a principal-element diagram is presented in Figure 3. Initially, the matrix coefficients, which are functions of the particular modes comprising the sound field and the microphone locations, are calculated. This equation system is solved by a Gaussian elimination method for the modal coefficients. The mode amplitude and phase are then extracted from these complex pressure vectors. The subroutines used in the calculation procedure are:

SOLVE -- This subroutine set ups and using SIMECQ solves the acoustic wave equation matrix for the modal amplitude and phase.

SIMECQ - This subroutine solves a N x N system of simultaneous equations having complex coefficients, using a Gaussian elimination method.

Sensitivity Coefficient Calculation

The Sensitivity Coefficient Calculation procedure is illustrated in the principal-element diagram presented in Figure 4.

An important element in this procedure is the calculation of influence coefficients, which reflect the sensitivity of mode amplitude and phase calculations to first-order errors in pressure measurements and microphone placement – the derivation of the influence coefficient is provided in reference 4, Section 3.4.

Because the inverse-matrix element is a common term in each expression, the procedure is initiated by calculating the inverse matrix. The influence coefficients are calculated next as a function of the modal structure and pressure measurements. The specific error in the modal amplitude and phase due to one of the five possible measurement errors is calculated from the product of the error in the measured quality and the root-sum-square of the influence coefficients. Finally, the error in a particular mode amplitude and phase is obtained as the combined effect of each measurement error.

The subroutines utilized in the Sensitivity Coefficient Calculation procedure are:

SENSTY – This subroutine calculates the standard deviations of the modal amplitude and phase for errors associated with pressure measurement and microphone location.

INVERT – This subroutine inverts a complex N x N matrix.

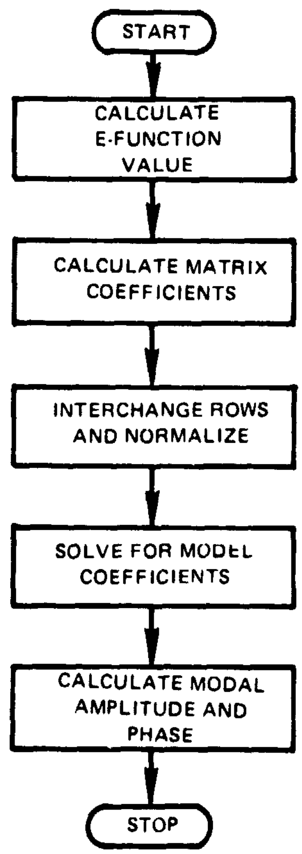

Figure 3 Principal-Element Diagram -- Mode Amplitude and Phase Calculation

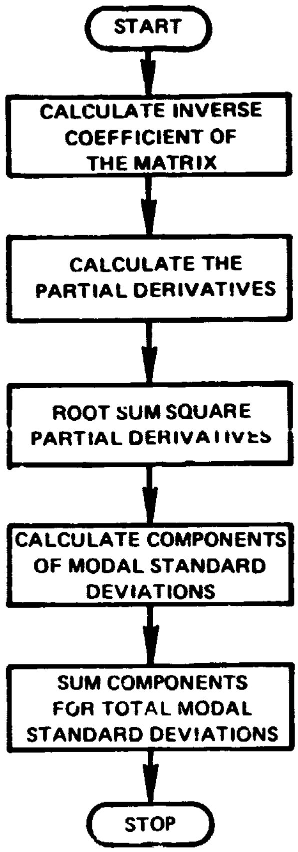

Figure 4 Principal-Element Diagram – Sensitivity Coefficient Calculation

Overall Pressure Calculation

The overall pressure at any location in the duct is obtained from the modal structure. The procedure for overall pressure calculation is illustrated in the principal-element diagram shown in Figure 5. The procedure summarizes the pressure contribution of each mode at a location in a duct defined by the user. The resultant amplitude and phase are then extracted from the complex pressure vector. Since this calculation is performed in the MAIN routine there are no subroutines or functions to list.

Output Section

The output format and the variables from the Modal Calculation Program are discussed in Section 3.5.1 and a sample case for three propagating modes is providing in Section 3.5.2. Both Sections 3.5.1 and 3.5.2 address the two possible modes of operation that can be executed with the program. Results from the computations are printed by the subroutines listed below after all angles are converted to within the range of 0° to 360°.

PRINT – This subroutine prints input and resultant values.

ANGPØS – This subroutine converts negative angles to positive angles in the range 0° to 360° for printing.

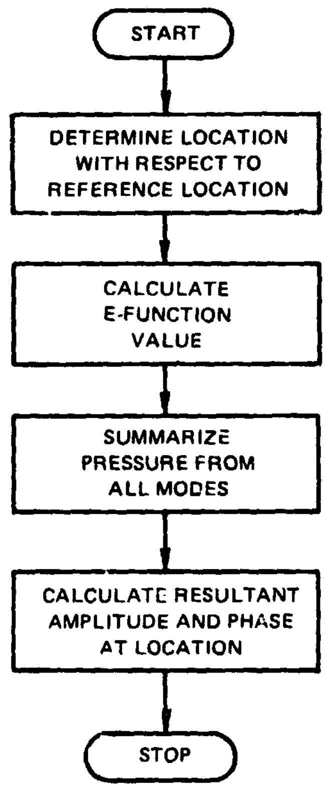

Figure 5 *Principal-Element Diagram – Overall Pressure Calculation*

3.4 INPUT DESCRIPTION

3.4.1 Input Format

The NAMELIST format is used to input data into the Modal Calculation Program and consists of a list of parameter names grouped under an identifying name: &INDATA. The parameter names correspond to variables – single variables and matrix elements – used in the program. These variables are set by specifying both the parameter name and its value. A feature of this type of input is that all associated parameters need not be specified. Any parameter not specified in the input retains its value from the preceeding case or the default value if the input is for the first case.

NAMELIST input for each case is identified by the characters &INDATA in Columns 2-7 of the first input card. Beginning in Column 9, parameters may be set using the format:

Parameter Name = Constant

The constant may be either a real or integer value and must be followed immediately by a comma. Parameter names, assigned values, or necessary commas must not extend beyond Column 72; and names of values cannot be continued on a subsequent card. Embedded blanks are not permitted in either the parameter name or constant value. Parameter names and their associated values may be specified in any order. The characters &END signify the end of the input for a particular case. If additional cards are required, parameters names must begin in Column 2.

A sample of this form of input for three microphones is presented in Figure 6.

3.4.2 Input Parameters

A sign convention was adopted for assigning positive or negative values to the input parameters. Any input parameter not addressed in this discussion is a positive value. The sign convention is formulated with respect to a cylindrical coordinate system that is consistent with the derivation of the coherent acoustic wave propagation model. Its unit vectors are designated by the directions: axial - x, circumferential - θ, radial - r.

A constant radius, annular duct is aligned with respect to this coordinate system in such a way that the positive axial unit vector is in a direction opposite to the flow. Thus, the Mach number of a uniform axial flow is always designated by a negative value to denote the axial flow rate in the negative axial direction. A positive circumferential unit vector projects in the direction that the rotor spins, and a negative vector projects in the counterrotating direction. Finally, the radial axis projects perpendicular to the centerline of the duct; thus, radial values are positive.

Each mode is characterized by three parameters which represent the circumferential and radial pressure distribution and its propagation direction. A specific mode is uniquely defined by the parenthetical notation (M, μ). The M defines a periodic circumferential pressure distribution with M number of lobes. Positive integers represent a corotating M-circumferential lobe pattern with respect to the rotor direction, and negative M integers refer to counterrotating modes. The radial mode index μ corresponds to the radial pressure distribution. These values are always non-negative integer numbers with high integer values indicating large pressure variations with respect to the radius.

The modal propagation direction in an inlet or discharge duct can be either an incident wave propagating from the fan or a reflected wave propagating towards the fan. Wave propagation in a moving medium is similarly effected by the flow rate for modes that are propagating with or against the flow direction. Hence, the input variable IDIR designates wave propagation with respect to the flow direction. Positive values denote waves propagating in the opposite direction with respect to the flow such as incident waves in the inlet duct and reflected waves in the discharge duct. Modes that propagate in the same direction as the flow are designated by a negative value for the input parameter IDIR.

Assigning of values to the input parameters will now be considered.

Since a determinative equation system is required, the number of mode indices, wave direction indicators, microphone coordinates, and measured pressures must be equal. When option B is utilized, the number of mode indices, wave direction indicators, and modal amplitude and phase values must correlate. These input parameters are listed in several tables at the end of this section. Each parameter has a corresponding description that is sufficient for assigning a value to these input parameters. However, assigning a value to the coefficient parameters for the standard deviation in measurement errors is not as straight forward as the previous parameters. The following discussion is provided to assist the user when assigning values to these variables.

The deviation coefficients for microphone location errors are the tolerances in the three coordinates: axial - x, radial - r, and circumferential - θ. These errors are related to the tolerance of a measurement – such as a micromenter – for determining the location of a microphone. Specifically, a user can estimate the microphone location standard deviation by assuming a high confidence level – such as ninety-five percent – to be associated with the number of significant digits used to define the pressure measuring coordinates. The standard deviation coefficients can then be computed from this information. For example, if a 95 percent confidence level is assigned to an axial measurement accuracy of 0.005 centimeter, the standard deviation (68.3 percent confidence level) is about 2.5×10^{-3} centimeter.

The error deviation coefficients for acoustic pressures include the two components amplitude and phase which correspond to the measured resultant pressure at any duct location. Two mechanisms can generate errors that affect the measurement of resultant pressure. One type of error is due to both response characteristics of the measuring device and repeatability of the coherent signal. The second type of error is caused by measuring contributions from modes not included in the calculation for determining the modal structure. A user can estimate the former pressure measurement error in a similar manner as previously presented for microphone location measurement errors.

A standard deviation can be computed by assuming a high confidence level to be associated with the combined inaccuracy of both pressure amplitude calibration errors and an error attributed to the repeatability of enhanced pressure signals during a period of time. In practice, however, this category of errors is small and can be minimized by requiring reasonable experimental procedures.

The second mechanism that can generate pressure measurement errors was not encountered in the previous category of location measurement errors. Ideally, the contribution from modes that are unlikely to control the duct sound field will not hinder the determination of fan sound mode structures. In practice, however, these modes have to be anticipated and their impact quantified if a meaningful standard deviation for the modal coefficients is to be calculated. This mechanism, which can be perceived as a measured pressure error, is difficult to assess prior to an experimental program. A general expression for this standard deviation is presented in Appendix E of reference 4. The actual value for the standard deviation used as input to the modal calculation program should be obtained from that general expression.

A description of the input variables for operating the Modal Calculation Program is provided in Tables I, II, and III: Table I – General Parameters; Table II – Test Geometry and Condition Parameters; Table III–Error Deviation Coefficient Parameters. Under the column heading "Variable Type": the letter "R" indicates that the number is real and contains a decimal point; the letter "I" indicates the number is an integer and does not have a decimal point. "Default Values" are also delineated and indicate the value of the parameter that is internally initialized prior to the program execution. Parameters not specified in the input for the first case retain this value. Although the default values are expressed in units of the English System, the computer program can be executed with data in any consistent system of units.

3.5 OUTPUT DESCRIPTION

3.5.1 Output Format

The output from the Modal Calculation Program is organized into four sections: Input Variables, Modal Amplitude and Phase Calculation, Sensitivity Coefficient Calculation, and Characteristic E-Function Values. All four sections are included as output when either option is requested by the input. The printout for a sample case is provided in Appendix C to illustrate the output format.

The Input Variable Section includes the value of the various parameters supplied by the user. The parameters that define the modal structure – the circumferential and radial order, and the wave-direction indicator – are listed. The reference pressure for converting the modal amplitude and resultant amplitude to decibels is also output in this section. The test geometry and conditions subsection lists various parameters that define the fan duct geometry and operating conditions observed during the experimental program. These parameters include the duct radius, duct hub-tip ratio, axial Mach number, and frequency.

The Modal Amplitude and Phase Calculation section includes both parameters that were provided by the user and the results from the calculation procedure. In this section, the user obtains the modal amplitude in units of pressure and decibels and the modal phase in units

TABLE I

GENERAL INPUT PARAMETERS

Input Name	Variable Type	Default Value	Description
LOCM	I	2	Number of microphones or modes. (Less than or equal to fifty).
LOCP	I	2	Number of prediction locations. (Less than or equal to fifty).
IEMU	I	0	Print indicator for characteristic E-function value. 0 = No print 1 = Print
PREF	R	2.9×10^{-9}	Reference pressure to convert pressure to decibels.
X0[a]	R	0.0	Axial coordinate of the reference location.
TH0[a]	R	0.0	Circumferential coordinate of the reference location. (degrees)
M(1)	I	−2	Circumferential mode index.
M(2)		−2	(Input NLOC values)
M(3)		0	
.		.	
.		.	
.		.	
M(50)		0	
MUS(1)	I	0	Radial mode index.
MUS(2)		1	(Input NLOC values)
MUS(3)		0	
.		.	
.		.	
.		.	
MUS(50)		0	
IDIR(1)	I	1	Mode propagation direction indicator.
IDIR(2)		1	(Input NLOC values)
IDIR(3)		0	
.		.	1 = opposite flow direction
.		.	-1 = with flow direction
.		.	
IDIR(50)		0	

Note a): Final character is a zero.

TABLE II

TEST GEOMETRY AND CONDITION INPUT PARAMETERS

Input Name	Variable Type	Default[a] Value	Description
HTR	R	0.44	Hub-tip ratio.
OR	R	5.0	Outer radius of duct.
EMX	R	0.07	Axial Mach number (always positive).
FRQ	R	3100.	Test frequency (Hertz)
SPEED	R	13566.	Speed of sound
X(1)	R	9.568	Axial coordinates of the measurement
X(2)		6.582	microphone locations. (Input LOC value)
X(3)		0.0	
.		.	
.		.	
.		.	
X(50)		0.0	
R(1)	R	5.0	Radial coordinates of the measurement
R(2)		5.0	microphone locations. (Input NLOC value)
R(3)		0.0	
.		.	
.		.	
.		.	
R(50)		0.0	
THM(1)	R	0.0	Circumferential coordinates of the measurement
THM(2)		0.0	microphone locations (degrees).
THM(3)		0.0	(Input NLOC value)
.		.	
.		.	
.		.	
THM(50)		0.0	
BETAM(1)	R	0.03136	Pressure amplitude at the measurement
BETAM(2)		0.02097	microphone locations. (Input NLOC value)
BETAM(3)		0.0	
.		.	
.		.	
.		.	
BETAM(50)		0.0	

TABLE II (Cont'd.)

Input Name	Variable Type	Default[a] Value	Description
PSIM(1)	R	97.8	Pressure phase at the measurement microphone locations (degrees). (Input NLOC values)
PSIM(2)		215.6	
PSIM(3)		0.0	
.		.	
.		.	
.		.	
PSIM(50)		0.0	
PX(1)	R	5.788	Axial coordinates of the prediction microphone locations. (Input LOCP values)
PX(2)		2.513	
PX(3)		0.0	
.		.	
.		.	
.		.	
PX(50)		0.0	
PR(1)	R	5.0	Radial coordinates of the prediction microphone locations. (Input LOCP values)
PR(2)		5.0	
PR(3)		0.0	
.		.	
.		.	
.		.	
PR(50)		0.0	
THP(1)	R	0.0	Circumferential coordinates of the prediction microphone locations (degrees). (Input LOCP values)
THP(2)		0.0	
THP(3)			
.		.	
.		.	
.		.	
THP(50)		0.0	
ICHK	I	0	Mode amplitude and phase indicator. 0 = Calculated from measured pressure; 1 = Input values
AM(1)	R	0.0	Mode amplitude. (If ICHK = 1, input NLOC values)
AM(2)		0.0	
AM(3)		0.0	
.		.	
.		.	
.		.	
AM(50)		0.0	

TABLE II (Cont'd.)

Input Name	Variable Type	Default[a] Value	Description
PHI(1)	R	0.0	Mode phase (degrees). (If ICHK = 1, input NLOC values)
PHI(2)		0.0	
PHI(3)		0.0	
.		.	
.		.	
.		.	
PHI(50)		0.0	

Note: (a) Default values shown in table are in units of the English System. The program, however, is designed to be executed with data in any consistent system of units.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE III

ERROR DEVIATION COEFFICIENT INPUT PARAMETERS

Input Name	Variable Type	Default Value	Description
SIGX	R	0.0	Standard deviation of the axial coordinate error.
SIGR	R	0.0	Standard deviation of the radial coordinate error.
SIGT	R	0.0	Standard deviation of the circumferential coordinate error (degrees).
SIGB	R	0.0	Standard deviation of the pressure amplitude error.
SIGP	R	0.0	Standard deviation of the pressure phase error (degrees).

of degrees. The corresponding mode indices, axial wave number in units of degrees-per-length, and eigen value $k_{m\mu}{}^{\sigma}$ are delineated.

Additional input parameters listed in this section include the reference location usually corresponding to the fan face where the modal phases are calculated. Coordinates of the input measurement locations and resultant prediction locations are listed adjacent to the respective acoustic pressure values. The input pressure values are supplied by the user in pressure units for the amplitude and degrees for the phase. The resultant pressure is calculated by the program and output is provided in units of decibels for the amplitude and degrees for the phase.

The Sensitivity Coefficient Calculation portion of the output comprises a number of sections, the primary output of which is the total normalized amplitude and the total phase deviation for each mode. These expressions represent the modal amplitude and phase error caused by a specified set of independant errors associated with the measurement of acoustic pressure and the tolerance of pressure measuring coordinates. The amplitude standard deviation of a specific mode is expressed as both the normalized quantity with respect to the mode amplitude and the mode amplitude error in decibels. The total phase deviation is expressed in degrees for each mode.

The contribution to the total amplitude and phase deviation assuming zero errors for the other error sources is provided under the heading "Normalized Standard Deviation Components". The amplitude deviation was normalized with respect to the mode amplitude. The total phase deviation in degrees for each error source is also provided under the heading. When these values are root-sum-squared, the previous expression for the total modal deviation is obtained. A user will benefit from the error deviation components by identifying which of the errors is controlling the total modal error.

The standard deviation components are also normalized with respect to their respective error. These parameters – referred to as the root-sum-square of the influence coefficients – enhance the combined variance of the influence coefficients at each microphone location. Thus, these parameters are the previous standard deviation components with respect to a unit measurement error in pressures or microphone coordinates. The root-sum-square of the influence coefficients is a convenient expression for assessing the probability of successfully tracking modes. A future user could examine these parameters to determine if the accuracy of experimental measurements made during an earlier test is sufficient to provide a desired confidence level in the mode amplitude and phase.

The influence coefficients are the partial derivatives of the mode amplitude and phase with respect to an error at each pressure measurement location that provides input for calculating the modal coefficients. These expressions allow a future user to evaluate the effect of non-uniform errors at the microphone locations. For example, an amplitude measurement error may be known to be significantly larger at one microphone location (e.g., inaccurate calibration) The user could then evaluate the impact of this error on the overall modal structure calculation.

The final section, Characteristic E-Functions, includes the value of E-functions, $E(k^{\sigma}_{m\mu} r)$, at the measurement and prediction locations corresponding to each mode. This final section is provided as output only if it has been requested by the user.

3.5.2 Sample Cases

Two cases are presented in the sample printout, to illustrate the two options: 1) calculating mode amplitude and phase values from acoustic pressure signals and 2) specifying these values either arbitrarily or as output from an analytical prediction deck. These sample cases demonstrate the execution of each option with data listed in Figure 6. The length units in the printout are in centimeters; the time units, in seconds; the force units, in dynes.

The first sample case illustrates the option of calculating the mode amplitude and phase for a situation where three modes are propagating in a half-meter diameter annular duct. Three coherent acoustic pressure amplitude and phase values are specified at three microphone locations on the duct wall. These acoustic signals are at a frequency of 6200-Hertz, and are used to determine the modal structure of the (−4,0), (−4,1) and (−4,2) modes.

The output for this sample case reveals that the amplitudes of the above modes are 137.4, 142.8, and 138.5 decibels, respectively; the modal phases are, respectively, 126.9, 160.0, and 229.2 degrees. Once the modal structure has been determined, the resultant sound field can be calculated at other duct locations. The resultant amplitude and phase – expressed in the same units as the modal coefficients – are requested at three microphone coordinates. The resultant amplitude at these locations are, respectively, 121.1, 115.7, and 115.1 decibels. The resultant phases are, respectively, 90.1, 357.8, and 67.2 degrees.

The sensitivity coefficient calculation portion of the program calculates the accuracy of the mode amplitude and phase values based on inaccuracies in the measured acoustic pressures and the microphone coordinates. Errors in the five measured quantities are expressed as standard deviations with zero mean. For this sample case they are axial - 2x 10^{-3} cm, radial - 2x 10^{-3} cm, circumferential - 2x 10^{-2} degree, amplitude - 25 dynes, and phase - 1.5 degrees. The combined effects of the error source deviations multiplied by the influence coefficients yields the modal amplitude and phase deviation. These calculated values for the (-4, 0), (-4, 1), and (-4, 2) modes are, respectively, 0.89, 0.84, and 0.87 decibel for the modal amplitude and 3.6, 2.5, and 1.7 degrees for the modal phase.

The second sample case illustrates the option to input the amplitudes and phases for the propagating modes to calculate the resultant acoustic pressure at specified locations. This case is similar to the first sample case because the (-4, 0), (-4, 1), and -4, 2) modes are propagating at 6200-Hertz in a half-meter diameter annular duct. The amplitude of all the modes is 121.9 decibels and the phases of these modes are, respectively, 325, 250, and 100 degrees. Output from the Modal Calculation Program comprises the resultant sound field at three microphone locations. The value of the resultant sound field is 115.8, 120.0, and 120.0 decibels for the resultant amplitude and 36.4, 345.0, and 298.2 degrees for the resultant phase.

PRATT & WHITNEY AIRCRAFT Division of UNITED TECHNOLOGIES

UTILITY CODING FORM

PWA FORM 3415 REV 6 72

ANALYST	ADDRESS	PHONE EXT	TITLE	SHEET ___ OF ___ / JOB OR PROGRAM NO

```
&INDATA NLOC=3,LOCP=3,IEMU=1,PREF=2.E-04,XO=0.,THO=0.,
M(1)=-4,-4,-4,
MUS(1)=0,1,2,
IDIR(1)=1,1,1,
HTR=0.44,DR=25.,EMX=-0.1,FRQ=6200.,SPEED=34345.,
X(1)=0.,14.976,29.960,
R(1)=25.,25.,25.,
THI(1)=0.,0.,0.,
RETAM(1)=228.7,128.9,84.10,
PSIM(1)=329.7,135.9,252.2,
PX(1)=0.,13.525,20.300,
PR(1)=25.,25.,25.,
THP(1)=239.9,109.6,15.9,
SIGX=0.002,SIGR=0.002,SIGT=0.02,SIGB=25.,SIGP=1.5 &END
&INDATA ICHK=1,
AM(1)=250.,250.,250.,
PHI(1)=325.,2[illegible]0.,100.,
PX(1)=0.,14.976,29.960,
PR(1)=25.,25.,25.,
THP(1)=0.,0.,0. &END
/*
//
```

Figure 5 Sample Input Form for Three Microphones

ORIGINAL PAGE IS OF POOR QUALITY

3.6 MACHINE REQUIREMENTS

The Modal Calculation Program can be compiled, linkage edited, and executed in 512 bytes of core storage.

The following mathematical functions and procedure are required:

CMPLX	-	Expresses two real arguments in complex form.
CABS	-	Modulus of a complex argument.
CEXP	-	Exponentiation of a complex argument.
AIMAG	-	Obtain imaginary part of a complex argument.
REAL	-	Obtain real part of a complex argument.
FLØAT	-	Conversion from integer to real.
IFIX	-	Conversion from real to integer.
ABS	-	Absolute value of a real number.
IABS	-	Absolute value of an integer.
SQRT	-	Square root of a real value.
MAXO	-	Obtain maximum value of input integers.
ALØG	-	Natural logarithim of a real positive argument.
SIN	-	Sine of a real argument.
CØS	-	Cosine of a real argument.
ATAN2	-	Arc tangent of two real arguments.

3.7 RESOURCE ESTIMATES

The central-processor-unit (CPU) time required to process a particular case depends on the number of modes input wnich determines the size of the matrix to be inverted. The average estimate of CPU time per mode is 0.15 second.

REFERENCES

1. Tyler, J. M. and Sofrin, T. G.: "Axial Flow Compressor Noise Studies," *SAE, Trans.* 70, p. 309, (1962)

2. Sofrin, T. G. and McCann, J. C.: "Pratt & Whitney Aircraft Experience in Compressor-Noise Reduction," For Presentation at the 72nd Meeting of the Acoustical Soc. Amer., Los Angeles, CA, Nov. 1966.

3. Posey, J. W.: "Comparison of Cross-Spectral and Signal Enhancement Methods for Mapping Steady-State Acoustic Fields in Turbomachinery Ducts," NASA TM X-73916, (1976)

4. Pickett, G. F.: "Fan Sound Mode Structure Determination - Final Report," NASA CR-135293, (1977)

5. Subroutines BESJ, BESY, and INVERT were adapted from the IBM Scientific Subroutine Package.

APPENDIX A

Calculation of the Characteristic Numbers

The characteristic numbers $K'^{\sigma}_{m\mu}$ and $Q^{\sigma}_{m\mu}$ are defined to be the paired roots of the simultaneous equations

$$\left[\frac{d}{dr'} J_m (K'^{\sigma}_{m\mu} r') + Q^{\sigma}_{m\mu} \frac{d}{dr'} Y_m (K'^{\sigma}_{m\mu} r')\right]_{r' = 1} = 0 \tag{1}$$

$$\left[\frac{d}{dr'} J_m (\sigma K'^{\sigma}_{m\mu} r') + Q^{\sigma}_{m\mu} \frac{d}{dr'} Y_m (\sigma K'^{\sigma}_{m\mu} r')\right]_{r' = 1} = 0 \tag{2}$$

For a given circumferential mode number, m, radial order, μ, and hub/tip ratio, σ, (where σ is not equal to zero); J_m and Y_m are the Bessel functions of the first and second kinds of order m.

The following relations are used in the formulation of a solution

$$\frac{d}{dr'} J_m (x) = J'_m (x) \frac{dx}{dr'} \tag{3}$$

$$\frac{d}{dr'} Y_m (x) = Y'_m (x) \frac{dx}{dr'} \tag{4}$$

$$J_{m+1} (x) = \frac{2m}{x} J_m (x) - J_{m-1} (x) \tag{5}$$

$$J_m' (x) = \frac{1}{2} [J_{m-1} (x) - J_{m+1} (x)] \tag{6}$$

$$= \frac{1}{2} [J_{m-1} (x) - \frac{2m}{x} J_m (x) + J_{m-1} (x)]$$

$$= J_{m-1} (x) - \frac{m}{x} J_m (x)$$

$$Y'_m(x) = \frac{2}{\pi x J_m(x)} + J'_m(x) \frac{Y_m(x)}{J_m(x)}$$

$$= \frac{2}{\pi x J_m(x)} + [J_{m-1}(x) - \frac{m}{x} J_m(x)] \frac{Y_m(x)}{J_m(x)} \tag{7}$$

Letting $K = K'^{\sigma}_{m\mu}$ and $Q = Q^{\sigma}_{m\mu}$, and evaluating at $r' = 1$; (1) and (2) become

$$J'_m(K) K + Q Y'_m(K) K = 0 \tag{8}$$

$$J'_m(\sigma K) \sigma K + Q Y'_m(\sigma K) \sigma K = 0 \tag{9}$$

From (8), $Q = - \frac{J'_m(K) K}{Y'_m(K) K}$ substituting into (9) yields

$$J'_m(\sigma K) \sigma K - \frac{J'_m(K) K}{Y'_m(K) K} Y'_m(\sigma K) \sigma K = 0 \tag{10}$$

Let $f(K) = J'_m(\sigma K) Y'_m(K) \sigma K^2 - J'_m(K) Y'_m(\sigma K) \sigma K^2 = 0$ (11)

Using the expressions in (5), (6), (7), and (11) then:

$$f(K) = \sigma K^2 [J_{m-1}(\sigma K) - \frac{m}{\sigma K} J_m(\sigma K)] \left\{ \frac{2}{\pi K J_m(K)} + [J_{m-1}(K) - \frac{m}{K} J_m(K)] \frac{Y_m(K)}{J_m(K)} \right\}$$

$$\sigma K^2 [J_{m-1}(K) - \frac{m}{K} J_m(K)] \left\{ \frac{2}{\pi \sigma K J_m(\sigma K)} + [J_{m-1}(\sigma K) - \frac{m}{\sigma K} J_m(\sigma K)] \frac{Y_m(\sigma K)}{J_m(\sigma K)} \right\} = 0 \tag{12}$$

$$f(K) = \sigma K^2 \left\{ \frac{2 [J_{m-1}(\sigma K) - \frac{m}{\sigma K} J_m(\sigma K)]}{\pi K J_m(K)} - \frac{2 [J_{m-1}(K) - \frac{m}{K} J_m(K)]}{\pi \sigma K J_m(\sigma K)} + \right.$$

$$\left. [J_{m-1}(K) - \frac{m}{K} J_m(K)] \; [J_{m-1}(\sigma K) - \frac{m}{\sigma K} J_m(\sigma K)] \; [\frac{Y_m(K)}{J_m(K)} - \frac{Y_m(\sigma K)}{J_m(\sigma K)}] \right\} = 0 \tag{13}$$

Equation (13) is evaluated for values of $\widetilde{K}_i = M + 3(i-1)$; $i = 1, 2, 3, \ldots$ until $f(\widetilde{K}_j)\ f(\widetilde{K}_j - 1) < 0$ for some j. A procedure employing a combination of false position and bisection techniques is then used to obtain a value of $K'^{\sigma}_{m\mu}$ in the interval $[\widetilde{K}_{j-1}, \widetilde{K}_j]$.

Having calculated a value of $K = K'^{\sigma}_{m\mu}$, the corresponding value of $Q = Q^{\sigma}_{m\mu}$ can be calculated. Combining (8) and (9) yields.

$$[J'_m(K) + J'_m(\sigma K)\sigma]\ K + Q\ [Y'_m(K) + Y'_m(\sigma K)\sigma]\ K = 0 \tag{14}$$

from which

$$Q = -\frac{J'_m(K) + J'_m(\sigma K)\sigma}{Y'_m(K) + Y'_m(\sigma K)\sigma} \tag{15}$$

For $\sigma = 0$, $Q^{\sigma}_{m\mu} = 0$ and $K'^{\sigma}_{m\mu} = 0$ is defined to be the root of

$$\left[\frac{d}{dr'} J_m\ (K'^{\sigma}_{m\mu}\ r')\right]_{r'=1} = 0 \tag{16}$$

Letting $K = K'^{\sigma}_{m\mu}$, and evaluating at $r' = 1$, (16) becomes

If $f(K) = J_m'(K)\ K = 0$, then (6) yields (17)

$$f(K) = [J_{m-1}(K) - \frac{m}{K} J_m(K)]\ K = 0 \tag{18}$$

Equation (18) is evaluated for values of $\widetilde{K}_i = m + 3(i-1)$; $i = 1, 2, 3, \ldots$ until $f(\widetilde{K}_j)\ f(\widetilde{K}_{j-1}) < 0$ for some value of j. A procedure employing a combination of false position and bisection techniques is then used to obtain a value of $K'^{\sigma}_{m\mu}$ in the interval $(\widetilde{K}_{j-1}, K_j)$.

APPENDIX B

MODAL CALCULATION PROGRAM

PROGRAM LISTING

```
++WRITE PRINT,T89902
C         DATA SET T89902     AT LEVEL 024 AS OF 06/22/77
C         DATA SET T89902     AT LEVEL 014 AS OF 02/01/77
C
C THIS PROGRAM CALCULATES MODAL AMPLITUDE AND PHASE
C
C
      COMMON /PREDCT/ XP(50), RP(50), THETAP(50)
      COMMON /APHIMU/ AMU(50), PHIMU(50), ICHECK
      COMMON /REFCON/ REFPRS
      COMMON /EMUS/   EMU(50,50), EMUP(50,50), EMUPRM(50,50), IEMPRT
      COMMON /CNSTNT/ NMEAS, NPRED, NMODES, SIGMA, B, MX, FREQ, A,
     1                OMEGA
      COMMON /KQMU/   KMU(50), QMU(50)
      COMMON /MODES/  MODE(50), MU(50), IWAVE(50)
      COMMON /ANGLES/ DEGRAD, RADDEG
      COMMON /OUTPUT/ AMPR(50), PHASER(50)
      COMMON /WAVENO/ KX(50)
      COMMON /REFS/   XREF, RREF, THREF
      REAL KMU, MX
      COMPLEX KX, EXPNT, SUM1, Q(50,50)
      DIMENSION EMUDUM(50)
C
C INPUT DATA FOR THIS CASE
C
   20 CALL INPUT( IEND )
      IF( IEND .GT. 0 )                GO TO 9999
C
C CALCULATE THE CHARACTERISTIC NUMBERS KMU AND QMU FOR EACH SET OF
C  CIRCUMFERENTIAL MODE NUMBER AND RADIAL ORDER
C
      CALL KQCAL
C
C CALCULATE AXIAL WAVE NUMBER
C
      FLOW          = OMEGA / A
      AMACH         = 1. - MX * MX
      DO 40 I=1,NMODES
      RADICL        = FLOW ** 2 - AMACH * ( KMU(I) / B ) ** 2
      IF( RADICL )                     25, 30, 30
   25 KX(I)         = CMPLX( -MX * FLOW / AMACH, IWAVE(I) *
     1                SQRT( ABS( RADICL ) ) / AMACH )
                                       GO TO 40
   30 KX(I)         = CMPLX( ( -MX * FLOW + IWAVE(I) * SQRT( RADICL ) ) /
     1                AMACH, 0.0 )
   40 CONTINUE
C
C IF THIS IS A CHECK RUN, AMU AND PHIMU HAVE BEEN INPUT. THUS THERE IS
C  NO NEED TO CALCULATE THEM.
C
      IF( ICHECK .GT. 0 )              GO TO 60
C
C SET UP AND SOLVE THE EQUATION SYSTEM ASSOCIATED WITH THE MEASUREMENT
```

```
C  LOCATIONS
C
      CALL SOLVE
C
C CALCULATE SENSITIVITY COEFFICIENTS
C
      CALL SENSTY( Q, NMODES )
C
C CALCULATE SUM OF MODAL AMPLITUDES AND PHASES FOR EACH PREDICTION
C  LOCATION
C
   60 DO 120 J=1,NPRED
      RPRIME        = RP(J) / B
C
C CALCULATE CHARACTERISTIC E-FUNCTIONS FOR RPRIME
C
      CALL EMUCAL( RPRIME, EMUP(1,J), EMUDUM, 0 )
C
      DXP           = XP(J) - XREF
      DTHETP        = THETAP(J) - THREF
      SUM1          = CMPLX( 0., 0. )
      DO 100 I=1,NMODES
      EXPNT         = CMPLX( 0.0, REAL( KX(I) ) * DXP + MODE(I) * DTHETP
     1                 * PHIMU(I) )
      SUM1          = AMU(I) * EMUP(I,J) * CEXP( EXPNT ) * EXP( -DXP *
     1                 AIMAG( KX(I) ) ) + SUM1
  100 CONTINUE
C
      AMPR(J)       = CABS( SUM1 )
      PHASER(J)     = ATAN2( AIMAG( SUM1 ), REAL( SUM1 ) )
  120 CONTINUE
C
C PRINT RESULTS OF THIS CASE
C
      CALL PRINT
C
C RECYCLE FOR NEXT CASE
C
                                   GO TO 20
 9999 STOP
      END
      SUBROUTINE ANGPOS( ANGLE, NUMBER )
C
C THIS SUBROUTINE CONVERTS NEGATIVE ANGLES TO CORRESPONDING POSITIVE
C  ANGLES
C
      DIMENSION ANGLE(1)
      DATA DEGREE / 360. /
C
      DO 80 I=1,NUMBER
      IF( ANGLE(I) )                20, 80, 80
   20 DO 40 J=1,10
      DELTA         = J * DEGREE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF( ANGLE(I) + DELTA )             40, 60, 60
   40 CONTINUE
   60 ANGLE(I)       = DELTA + ANGLE(I)
   80 CONTINUE
 9999 RETURN
      END
      FUNCTION BESL1( X )
C
C THIS FUNCTION CALCULATES VALUES OF THE EQUATION DEFINING THE SYSTEM OF
C  DIFFERENTIAL EQUATIONS FOR A NON-ZERO HUB/TIP RATIO
C
      COMMON /BESSL/ ISIGN, JSIGN, DELKMU, TOL, M, PI
      COMMON /CNSTNT/ DUM1(3), SIGMA, DUM2(5)
C
      X1             = X * SIGMA
      CALL BESJ( X1, M-JSIGN, EMJM1, TOL, IER1 )
      CALL BESJ( X1, M , EMJX1, TOL, IER2 )
      CALL BESJ( X, M, EMJ, TOL, IER3 )
      CALL BESJ( X, M-JSIGN, EMJP1, TOL, IER4 )
      CALL BESY( X, M, EMYX, IER5 )
      CALL BESY( X1, M, EMYX1, IER6 )
C
      EMJM1          = JSIGN * ISIGN * EMJM1
      EMJX1          = ISIGN * EMJX1
      EMJ            = ISIGN * EMJ
      EMJP1          = JSIGN * ISIGN * EMJP1
      EMYX           = ISIGN * EMYX
      EMYX1          = ISIGN * EMYX1
C
      A1             = EMJM1 - ( M * JSIGN / X1 ) * EMJX1
      A2             = EMJP1 - ( M * JSIGN / X ) * EMJ
      A3             = 2. * A1 / ( PI * X * EMJ )
      A4             = 2. * A2 / ( PI * X1 * EMJX1 )
      A5             = A1 * A2 * ( EMYX / EMJ - EMYX1 / EMJX1 )
C
      BESL1          = X1 * X * ( A3 - A4 * A5 )
      RETURN
      END
      FUNCTION BESL2( X )
C
C THIS FUNCTION CALCULATES VALUES OF THE EQUATION DEFINING THE SYSTEM OF
C  DIFFERENTIAL EQUATIONS FOR A HUB/TIP RATIO OF ZERO
C
      COMMON /BESSL/ ISIGN, JSIGN, DELKMU, TOL, M, PI
      COMMON /CNSTNT/ DUM1(3), SIGMA, DUM2(5)
C
      CALL BESJ( X, M-JSIGN, EMJM1, TOL, IER1 )
      CALL BESJ( X, M, EMJ, TOL, IER2 )
C
      EMJM1          = JSIGN * ISIGN * EMJM1
      EMJ            = ISIGN * EMJ
      BESL2          = X * EMJM1 - M * JSIGN * EMJ
      RETURN
```

```
      END
      SUBROUTINE BESJ( X, N, BJ, D, IER )
C
C THIS SUBROUTINE CALCULATES THE J BESSEL FUNCTION FOR A GIVEN ARGUMENT,
C  X, AND ORDER N. THIS SUBROUTINE WAS TAKEN FROM THE IBM SCIENTIFIC
C  SUBROUTINE PACKAGE
C
      BJ            = 0.0
      IF( N .GE. 0 )                  GO TO 20
C
C ERROR - NEGATIVE ORDER. SET ERROR INDICATOR TO 1 AND RETURN
C
      IER           = 1
                                       GO TO 9999
   20 IF( X )                          40, 30, 60
   30 IF( N .GT. 0 )                   GO TO 40
      BJ            = 1.0
                                       GO TO 9999
C
C ERROR - ARGUMENT ZERO OF NEGATIVE. SET ERROR INDICATOR TO 2 AND RETURN
C
   40 IER           = 2
                                       GO TO 9999
C
C CALCULATE MAXIMUM ORDER NUMBER THAT CAN BE PROCESSED FOR X.
C   IF X .LE. 15, N MUST BE LESS THAN 20 + 10*X - X**2/3.
C   IF X .GT. 15, N MUST BE LESS THAN 90 + X/2
C
   60 IF( X - 15. )                    80, 80, 100
   80 NTEST         = 20. + 10. * X - X ** 2 / 3.
                                       GO TO 120
  100 NTEST         = 90. + X / 2.
  120 IF( N .LT. NTEST )               GO TO 140
C
C ERROR - ORDER RANGE COMPARED TO X IS NOT CORRECT. SET ERROR INDICATOR
C  TO 4 AND RETURN.
C
      IER           = 4
                                       GO TO 9999
  140 IER           = 0
      N1            = N + 1
      BPREV         = 0.0
C
C COMPUTE STARTING VALUE OF M
C
      IF( X - 5. )                     160, 180, 180
  160 MA            = X + 6.
                                       GO TO 200
  180 MA            = 1.4 * X + 60. / X
  200 MB            = N + IFIX( X ) / 4 + 2
      MZERO         = MAX0( MA, MB )
C
C SET UPPER LIMIT OF M
```

```
C
      MMAX            = NTEST
  220 DO 320 M=MZERO,MMAX,3
C
C SET F(M), F(M-1)
C
      FM1             = 1.0E-28
      FM              = 0.0
      ALPHA           = 0.0
      JT              = 1
      IF( ( M / 2 ) * 2 .EQ. M )        JT = -1
      M2              = M - 2
      DO 280 K=1,M2
      MK              = M - K
      BMK             = 2. * FLOAT( MK ) * FM1 / X - FM
      FM              = FM1
      FM1             = BMK
      IF( MK - N - 1 )                  260, 240, 260
  240 BJ              = BMK
  260 JT              = -JT
      S               = 1 + JT
      ALPHA           = ALPHA + BMK * S
  280 CONTINUE
C
      BMK             = 2. * FM1 / X - FM
      IF( N .EQ. 0 )                    BJ = BMK
      ALPHA           = ALPHA + BMK
      BJ              = BJ / ALPHA
      IF( ABS( BJ - BPREV ) - ABS( D * BJ ) )  9999, 9999, 300
  300 BPREV           = BJ
  320 CONTINUE
C
C ERROR - REQUIRED TOLERANCE NOT OBTAINED. SET ERROR INDICATOR TO 3 AND
C  RETURN
C
      IER             = 3
 9999 RETURN
      END
      SUBROUTINE BESY( X, N, BY, IER )
C
C THIS SUBROUTINE CALCULATES THE Y BESSEL FUNCTION FOR A GIVEN ARGUMENT,
C  X, AND ORDER N. THIS SUBROUTINE WAS TAKEN FROM THE IBM SCIENTIFIC
C  SUBROUTINE PACKAGE
C
      IER             = 0
      IF( N .GE. 0 )                    GO TO 20
C
C ERROR - NEGATIVE ORDER. SET ERROR INDICATOR TO 1 AND RETURN
C
      IER             = 1
                                        GO TO 9999
   20 IF( X )                           40, 40, 60
C
```

```
C ERROR - ARGUMENT ZERO OR NEGATIVE. SET ERROR INDICATOR TO 2 AND RETURN
C
   40 IER           = 2
                                         GO TO 9999
C
C BRANCH IF X IS LESS THAN OR EQUAL TO 4.
C
   60 IF( X - 4. )                       100, 100, 80
C
C CALCULATE Y0 AND Y1 FOR X GREATER THAN 4.
C
   80 T1            = 4. / X
      T2            = T1 * T1
      P0            = ( ( ( ( -.0000037043 * T2 + .0000173565 ) * T2 -
     1                .0000487613 ) * T2 + .00017343 ) * T2 - .001753062
     2                ) * T2 + .3989423
      Q0            = ( ( ( ( .0000032312 * T2 - .0000142078 ) * T2 +
     1                .0000342468 ) * T2 - .0000869791 ) * T2 +
     2                .0004564324 ) * T2 - .01246694
      P1            = ( ( ( ( .0000042414 * T2 - .0000200920 ) * T2 *
     1                .0000580759 ) * T2 - .000223203 ) * T2 +
     2                .002921826 ) * T2 + .3989423
      Q1            = ( ( ( ( -.0000036594 * T2 + .00001622 ) * T2 -
     1                .0000398708 ) * T2 * .0001064741 ) * T2 -
     2                .00063904 ) * T2 + .03740084
      A             = 2. / SQRT( X )
      B             = A * T1
      C             = X - .7853982
      Y0            = A * P0 * SIN( C ) + B * Q0 * COS( C )
      Y1            = -A * P1 * COS( C ) + B * Q1 * SIN( C )
                                         GO TO 160
C
C CALCULATE Y0 AND Y1 FOR X LESS THAN OR EQUAL TO 4
C
  100 XX            = .5 * X
      X2            = XX * XX
      T             = ALOG( XX ) + .5772157
      SUM           = 0.0
      TERM          = T
      Y0            = T
      DO 120 L=1,15
      IF( L .NE. 1 )                     SUM = 1. / FLOAT( L-1) + SUM
      FL            = L
      TS            = T - SUM
      TERM          = ( -X2 * TERM / ( FL ** 2 ) ) * ( 1. - 1. / ( FL *
     1                TS ) )
      Y0            = TERM + Y0
  120 CONTINUE
      TERM          = XX * ( T - .5 )
      SUM           = 0.0
      Y1            = TERM
      DO 140 L=2,16
      SUM           = 1. / FLOAT( L-1 ) + SUM
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      FL              = L
      FL1             = FL - 1.
      TS              = T - SUM
      TERM            = ( -X2 * TERM / ( FL * FL1 ) ) * ( ( TS - .5 / FL )
     1                  / ( TS * .5 / FL1 ) )
      Y1              = TERM + Y1
  140 CONTINUE
      PI2             = .6366198
      Y0              = PI2 * Y0
      Y1              = PI2 * ( Y1 - 1. / X )
C
C CHECK IF ONLY Y0 OR Y1 IS DESIRED
C
  160 IF( N .GT. 1 )                    GO TO 180
C
C RETURN Y0 OR Y1 AS REQUIRED
C
      BY              = Y0
      IF( N .EQ. 1 )                    BY = Y1
                                        GO TO 9999
C
C PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
C
  180 YA              = Y0
      YB              = Y1
      K               = 1
  200 T = FLOAT( 2*K ) / X
      YC              = T * YB - YA
      IF( ABS( YC ) - 1.0E70 )          240, 240, 220
C
C ERROR - BY HAS EXCEEDED MAGNITUDE OF 10**70. SET ERROR INDICATOR TO 3
C  AND RETURN
C
  220 IER             = 3
                                        GO TO 9999
  240 K               = 1 + K
      IF( K .EQ. N )                    GO TO 260
      YA              = YB
      YB              = YC
                                        GO TO 200
  260 BY              = YC
 9999 RETURN
      END
      BLOCK DATA
      COMMON /DFAULT/ LOCM, LOCP, HTR, OR, EMX, FRQ, X0, R0,
     1                THO, X(50), R(50), THM(50), BETAM(50), PSIM(50),
     2                PX(50), PR(50), THP(50), M(50), MUS(50),
     3                IDIR(50), PREF, AM(50), PHI(50), ICHK,
     4                SIGX, SIGR, SIGT, SIGB, SIGP, IEMU, SPEED
C
C CONSTANT DEFAULT VALUES
C
C    LOCM    - NUMBER OF MEASUREMENT LOCATIONS
```

```
C     LOCP    - NUMBER OF PREDICTION LOCATIONS
C     HTR     - HUB / TIP RATIO
C     OR      - OUTER RADIUS
C     EMX     - AXIAL MACH NUMBER
C     FRQ     - TEST FREQUENCY
C     SPEED   - SPEED OF SOUND
C
      DATA LOCM / 2 /, LOCP / 2 /, HTR / 0.44 /, OR / 5.0 /,
     1     EMX / -0.07 /, FRQ / 3100. /, SPEED / 13566.24 /
C
C REFERENCE LOCATION VALUES
C
C     XO      - AXIAL COMPONENT OF REFERENCE LOCATION
C     RO      - RADIAL COMPONENT OF REFERENCE LOCATION
C     THO     - ANGULAR COMPONENT OF REFERENCE LOCATION (DEG)
C
      DATA XO / 0.0 /, RO / 0.0 /, THO / 0.0 /
C
C MEASUREMENT LOCATION VALUES
C
C     X       - AXIAL COMPONENT OF MEASUREMENT LOCATION
C     R       - RADIAL COMPONENT OF MEASUREMENT LOCATION
C     THM     - ANGULAR COMPONENT OF MEASUREMENT LOCATION (DEG)
C     BETAM   - AMPLITUDE OF MEASURED VALUE
C     PSIM    - PHASE ANGLE OF MEASURED VALUE (DEG)
C
      DATA X / 9.568, 6.582, 48*0.0 /, R / 2*5.0, 48*0.0 /,
     1     THM   / 50*0.0 /, BETAM / 0.03136, 0.05097, 48*0.0 /,
     2     PSIM / 97.8, 215.6, 48*0.0 /
C
C PREDICTION LOCATION VALUES
C
C     PX      - AXIAL COMPONENT OF PREDICTION LOCATION
C     PR      - RADIAL COMPONENT OF PREDICTION LOCATION
C     THP     - ANGULAR COMPONENT OF PREDICTION LOCATION (DEG)
C
      DATA PX / 5.788, 2.513, 48*0.0 /, PR / 2*5.0, 48*0.0 /,
     1     THP   / 50*0.0 /
C
C MODE VALUES
C
C     M       - CIRCUMFERENTIAL MODE NUMBER
C     MUS     - RADIAL ORDER
C     IDIR    - INCIDENT OR REFLECTED WAVE INDICATOR
C
      DATA M / -2, -2, 48*0 /, MUS / 0, 1, 48*0 /, IDIR / 1, 1, 48*0 /
C
C REFERENCE CONSTANTS
C
C     PREF    - REFERENCE PRESSURE
C
      DATA PREF / 2.9E-9 /
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C RESULTANT AMPLITUDE AND PHASE VALUES
C
C    AM     - AMPLITUDE
C    PHI    - PHASE ANGLE (DEG)
C    ICHK   - CHECK CASE INDICATOR
C
      DATA AM / 50*0.0 /, PHI / 50*0.0 /, ICHK / 0 /
C
C BESSEL FUNCTION VALUES
C
      COMMON /BESSL/  DUM2(2), DELKMU, TOL, MM, PI
      DATA DELKMU / 3. /, TOL / .0001 /, PI / 3.141593 /
C
C ANGULAR CONVERSION FACTORS
C
C    DEGRAD - DEGREES TO RADIANS
C    RADDEG - RADIANS TO DEGREES
C
      COMMON /ANGLES/ DEGRAD, RADDEG
      DATA DEGRAD / .0174533 /, RADDEG / 57.29578 /
C
C
C ERROR DEVIATION COEFFICIENTS
C
C    SIGX   - AXIAL COEFFICIENT
C    SIGR   - RADIAL COEFFICIENT
C    SIGT   - ANGULAR COEFFICIENT
C    SIGB   - AMPLITUDE COEFFICIENT
C    SIGP   - PHASE COEFFICIENT
C
      DATA SIGX / 0.0 /, SIGR / 0.0 /, SIGT / 0.0 /, SIGP / 0.0 /,
     1     SIGB / 0.0 /
      END
      SUBROUTINE EMUCAL( RPRIME, EMU, EMUPRM, IDERIV )
C
C THIS SUBROUTINE CALCULATES NMODES CHARACTERISTIC E-FUNCTION VALUES FOR
C  A PARTICULAR RADIAL VALUE, RPRIME.
C
      DIMENSION EMU(1), EMUPRM(1)
      COMMON /CNSTNT/ NMEAS, NPRED, NMODES, DUM1(6)
      COMMON /KQMU/   KMU(50), QMU(50)
      COMMON /MODES/  MODE(50), DUM2(100)
      COMMON /BESSL/  ISIGN, JSIGN, DELKMU, TOL, M, PI
      REAL KMU, JPRIME
C
C
      DO 40 I=1,NMODES
      M            = IABS( MODE(I) )
      IF( M .NE. 0 )                 GO TO 10
      ISIGN        = 1
      JSIGN        = -1
                                     GO TO 20
   10 ISIGN        = MODE(I) / M
```

```
      JSIGN          = ISIGN
      IF( ISIGN .GE. 0 )                  GO TO 20
C
C NEGATIVE MODE NUMBER. IF EVEN, SIGN OF BESSEL FUNCTION WILL BE +1. IF
C  ODD, SIGN OF BESSEL FUNCTION WILL BE -1.
C
      IF( ( M / 2 ) * 2 .EQ. M )         ISIGN = 1
   20 CONST          = KMU(I) * RPRIME
C
C CALCULATE BESSEL FUNCTIONS OF FIRST AND SECOND KIND FOR KMU(I)*RPRIME
C
      CALL BESJ( CONST, M, EMJ, TOL, IER1 )
      CALL BESY( CONST, M, EMY, IER2 )
      CALL BESJ( CONST, M-JSIGN, EMM1, TOL, IER3 )
      EMM1           = ISIGN * JSIGN * EMM1
      EMJ            = ISIGN * EMJ
      EMY            = ISIGN * EMY
C
C CALCULATE CHARACTERISTIC E-FUNCTION
C
      EMU(I)         = EMJ + QMU(I) * EMY
      IF( IDERIV .LE. 0 )                 GO TO 40
      IF( KMU(I) )                        30, 25, 30
C
C (0,0) CASE. SET DERIVATIVE TO 0.0
C
   25 EMUPRM(I)      = 0.0
                                          GO TO 40
C
C CALCULATE DERIVATIVE OF CHARACTERISTIC E-FUNCTION
C
   30 JPRIME         = EMM1 - MODE(I) * EMJ / CONST
      YPRIME         = 2. / ( PI * CONST * EMJ ) + JPRIME * EMY / EMJ
      EMUPRM(I)      = KMU(I) * ( JPRIME + QMU(I) * YPRIME )
   40 CONTINUE
 9999 RETURN
      END
      FUNCTION FALZIP (FUNCT, AL , BR , TOL, ROOT, ITER, YY)
C
C CORRESPONDS TO OLD VERSION (FALSIE) ARGUMENT LIST AS FOLLOWS (THIS IS
C  FOR INTERNAL PURPOSES ONLY, IN USE THE TWO ARE INTERCHANGEABLE).
C
C     FUNCTION FALSIE (AXR,    XXL, XXR, TOL, ROOT, ITER, YY)
C
C
C  THIS ROUTINE USES A COMBINATION OF FALSE POSITION AND BISECTION
C    TECHNIQUES TO SOLVE FOR A ROOT ('ROOT') OF A GIVEN FUNCTION
C    ('FUNCT') WHICH HAS ONE ARGUMENT (THE INDEPENDENT VARIABLE).
C
C   'AL,BR' DEFINES THE INTERVAL TO BE SEARCHED.
C
C  THE VALU". RETURNED BY THE FUNCTION IS FALZIP.  FUNCT(FALZIP) = ROOT
C
```

```
C  THE SEARCH CONTINUES UNTIL TWO SUBSEQUENT GUESSES ARE WITHIN 'TOL'
C    OF EACH OTHER, OR UNTIL 'ITER' ITERATIONS HAVE TAKEN PLACE.
C
C  'YY' IS RETURNED AS FUNCT(FALZIP), AND SHOULD BE CLOSE TO 'ROOT'.
C
C  THE TECHNIQUE WAS ADAPTED FROM AN ALGOL SUBROUTINE APPEARING IN THE
C    COMPUTER JOURNAL 12 (1969) --  'EIGENVALUES OF A*X = LAMBDA*B*X
C    WITH BAND SYMMETRIC A AND B' BY G. PETERS + J.H. WILKINSON
C
      EXTERNAL FUNCT
      REAL INTERP
C
C  J IS COUNT OF ITERATIONS.
    1 J = 0
      A = AL
      B = BR
C
C  EVALUATE FUNCTION AT LEFT (A) AND RIGHT (B) BRACKETS.
      AF = FUNCT (A)
      BF = FUNCT (B)
C
C  THE FOLLOWING (THROUGH STATEMENT 3) DETERMINES IF THE FUNCTION IS OF
C    OPPOSITE SIGN AT THE ENDPOINTS GIVEN.
      ISW = 1
      IF (BF - ROOT) 2, 75, 3
    2 ISW = -1
    3 IF ((AF - ROOT) * ISW) 50, 80, 85
C
C  STATEMENT 5 INCREMENTS THE COUNTER J; FIRST TIME THROUGH GO TO 50.
    5 J = J + 1
C
C  IF LEFT BRACKET HAS 'SAME' FUNCTION VALUE AS RIGHT, USE BISECTION.
C    OTHERWISE, SET UP INTERPOLATED POINT FOR POSSIBLE USE.
      IF (ABS((AF - BF)/BF) - 1.E-5) 10, 10, 15
   10 INTERP = BISECT
      GO TO 20
   15 INTERP = (A*BF - B*AF + (B-A)*ROOT) / (BF-AF)
C
C  IF WITHIN A TOLERANCE OF THE BRACKET B, MOVE THE INTERPOLATED POINT
C    ONE TOLERANCE AWAY.
   20 IF ((ABS(INTERP-B)/ABS(INTERP+B)) -2.*TOL) 22,23,23
   22 INTERP = B + (C - B) / ABS (C - B) * TOL
C
C  SET A=B (B IS ALWAYS THE POINT WITH SMALLEST (ABS) VALUE OF FUNCTION.
   23 A = B
      AF = BF
C
C  USE POINT CLOSEST TO B (INTERP OR BISECT) AS NEW B AND EVALUATE BF.
      IF ((INTERP - BISECT) * (B - INTERP)) 30, 25, 25
   25 B = INTERP
      GO TO 35
   30 B = BISECT
   35 BF = FUNCT(B)
```

```
      BFMR = BF - ROOT
C
C  IF CF IS ON THE SAME SIDE OF THE ROOT AS BF, LET POINT C = POINT A.
   40 IF ((CF - ROOT) * BFMR) 55, 75, 50
   50 C = A
      CF = AF
C
C  IF CF IS CLOSER (ABS) TO ROOT THAN BF, SWITCH POINTS B AND C.
C    IN ANY CASE, B AND C ARE THE TWO BRACKETS. ALSO BF IS CLOSER TO THE
C    ROOT THAN CF IS.
   55 IF (ABS(BF - ROOT) - ABS(CF - ROOT)) 60, 60, 57
   57 A  = B
      AF = BF
      B  = C
      BF = CF
      C  = A
      CF = AF
C
C  SET UP BISECTION POINT.  IF CLOSE ENOUGH, FINISH UP.  OTHERWISE GO
C    BACK IF ITERATION COUNT DOESN'T EXCEED MAXIMUM.
   60 BISECT = (B + C) / 2.
      IF ((ABS(BISECT-B)/ABS(BISECT+B)) -2.*TOL ) 75,65,65
   65 IF (J - ITER) 5, 70, 70
   70 WRITE(6,1000)J,B,BF,C,CF
 1000 FORMAT (1H0/// 30X, 'IN FALZIP, AFTER', I4, ' ITERATIONS' //
     1     10X, 'BRACKET 1 = ', G15.8, 5X, 'FUNCTION = ', G15.8/
     2     10X, 'BRACKET 2 = ', G15.8, 5X, 'FUNCTION = ', G15.8/
     3     5X, 'BRACKET 1 WAS RETURNED AS RESULT.')
   75 FALZIP = B
      YY = BF
      RETURN
   80 FALZIP = A
      YY = AF
      RETURN
   85 WRITE(6,1100)ROOT,A,AF,B,BF
 1100 FORMAT ('0***IN FALZIP, ROOT GIVEN (=', G15.8, ') DIDN''T FALL BET
     1WEEN VALUES OF FUNCTION AT BRACKETS GIVEN***'/
     2    10X, 'BRACKET 1 = ', G15.8, 5X, 'FUNCTION = ', G15.8 /
     3    10X, 'BRACKET 2 = ', G15.8, 5X, 'FUNCTION = ', G15.8 /
     4          40X, 'TERMINATING RUN' )
      STOP
      END
      SUBROUTINE INPUT( IEND )
C
C THIS SUBROUTINE INPUTS THE DATA REQUIRED FOR THE EXECUTION OF A CASE
C
      COMMON /DFAULT/ LOCM, LOCP, HTR, OR, EMX, FRQ, XO, RO,
     1                THO, X(50), R(50), THM(50), BETAM(50), PSIM(50),
     2                PX(50), PR(50), THP(50), M(50), MUS(50),
     3                IDIR(50), PREF, AM(50), PHI(50), ICHK,
     4                SIGX, SIGR, SIGT, SIGB, SIGP, IEMU, SPEED
      COMMON /CNSTNT/ NMFAS, NPRED, NMODES, SIGMA, B, MX, FREQ, A,
     1                OMEGA
```

```
      COMMON /REFS/   XREF, RREF, THREF
      COMMON /MEASUR/ XM(50), RM(50), THETAM(50), BETA(50), PSI(50)
      COMMON /PREDCT/ XP(50), RP(50), THETAP(50)
      COMMON /MODES/  MODE(50), MU(50), IWAVE(50)
      COMMON /BESSL/  DUM1(5), PI
      COMMON /REFCON/ REFPRS
      COMMON /ANGLES/  DEGRAD, RADDEG
      COMMON /APHIMU/ AMU(50), PHIMU(50), ICHECK
      COMMON /EMUS/   EMU(50,50), EMUP(50,50), EMUPRM(50,50), IEMPRT
      COMMON /DVIATE/ SIGMAX, SIGMAR, SIGMAT, SIGMAB, SIGMAP, DUM2(150),
     1                IDEV
C
      NAMELIST /INDATA/ LOCM, LOCP, HTR, OR, EMX, FRQ, XO,
     1                  THO, X, R, THM, BETAM, PSIM, PX, PR, THP, M,
     2                  MUS, IDIR, PREF, VREF, AM, PHI, ICHK,
     3                  SIGX, SIGR, SIGT, SIGB, SIGP, IEMU, SPEED
      REAL MX, LREF
C
C
      IEND          = 0
      READ(5,INDATA,END=9998)
C
C SET UP INTERNAL PARAMETERS
C
      NMEAS         = LOCM
      NPRED         = LOCP
      NMODES        = LOCM
      SIGMA         = HTR
      MX            = EMX
      FREQ          = FRQ
      REFPRS        = PREF
      ICHECK        = ICHK
      XREF          = XO
      RREF          = RO
      THREF         = THO * DEGRAD
      B             = OR
      A             = SPEED
      IEMPRT        = IEMU
      SIGMAX        = SIGX
      SIGMAR        = SIGR
      SIGMAT        = SIGT
      SIGMAB        = SIGB
      SIGMAP        = SIGP
C
      DO 20 I=1,NMEAS
      XM(I)         = X(I)
      RM(I)         = R(I)
      THETAM(I)     = THM(I) * DEGRAD
      PSI(I)        = PSIM(I) * DEGRAD
      BETA(I)       = BETAM(I)
   20 CONTINUE
C
      DO 40 I=1,NPRED
```

```
      XP(I)         = PX(I)
      RP(I)         = PR(I)
      THETAP(I)     = THP(I) * DEGRAD
   40 CONTINUE
C
      DO 60 I=1,NMODES
      MODE(I)       = M(I)
      MU(I)         = MUS(I)
      IWAVE(I)      = IDIR(I)
      AMU(I)        = AM(I)
      PHIMU(I)      = PHI(I) * DEGRAD
   60 CONTINUE
C
      DO 80 J=1,20
      DO 80 I=1,20
      EMU(I,J)      = 0.0
      EMUP(I,J)     = 0.0
      EMUPRM(I,J)   = 0.0
   80 CONTINUE
C
C CALCULATE RADIAN FREQUENCY
C
      OMEGA         = 2. * PI * FREQ
C
C
C SET INDICATOR FOR ERROR SOURCE STANDARD DEVIATIONS
C
      IF( SIGMAX )                        200, 100, 200
  100 IF( SIGMAR )                        200, 120, 200
  120 IF( SIGMAT )                        200, 140, 200
  140 IF( SIGMAB )                        200, 160, 200
  160 IF( SIGMAP )                        200, 180, 200
  180 IDEV          = 0
                                          GO TO 9999
  200 IDEV          = 1
                                          GO TO 9999
C
C END OF DATA SET
C
 9998 IEND          = 1
 9999 RETURN
      END
      SUBROUTINE KMUCAL( VALUE, DELTA, KMU, RIGHT )
C
C THIS SUBROUTINE CALCULATES THE CHARACTERISTIC NUMBER, KMU
C
      EXTERNAL BESL1, BESL2
      COMMON /CNSTNT/ DUM1(3), SIGMA, DUM2(5)
      REAL KMU, LEFT
   30 IPLUS         = 0
      IMINUS        = 0
   35 IF( SIGMA )                         50, 40, 50
C
```

```
   40 KMU            = BESL2( VALUE )
                                         GO TO 60
   50 KMU            = BESL1( VALUE )
C
   60 IF( KMU )                          80, 65, 70
   65 RIGHT          = VALUE
                                         GO TO 130
   70 IPLUS          = 1
                                         GO TO 90
   80 IMINUS         = 1
C
C DETERMINE IF LEFT AND RIGHT BRACKETS HAVE BEEN FOUND.
C
   90 IF( IPLUS .EQ. 1 .AND. IMINUS .EQ. 1 )  GO TO 100
C
C BRACKETS NOT FOUND. RECYLE.
C
      VALUSV         = VALUE
      VALUE          = DELTA + VALUE
                                         GO TO 35
C
C BRACKETS FOUND, CALCULATE KMU
C
  100 LEFT           = VALUSV
      RIGHT          = VALUE
      IF( SIGMA )                        110, 120, 110
  110 KMU            = FALZIP( BESL1,  LEFT, RIGHT, .001, 0.0, 50, YY )
                                         GO TO 130
  120 KMU            = FALZIP( BESL2,  LEFT, RIGHT, .001, 0.0, 50, YY )
  130 RETURN
      END
      SUBROUTINE KQCAL
C
C THIS SUBROUTINE CALCULATES THE CHARACTERISTIC NUMBERS KMU AND QMU
C
      COMMON /MODES/  MODE(50), MU(50), IWAVE(50)
      COMMON /KQMU/   KMU(50), QMU(50)
      COMMON /BESSL/  ISIGN, JSIGN, DELKMU, TOL, M, PI
      COMMON /CNSTNT/ DUM1(2), NMODES, SIGMA, DUM2(5)
      REAL KMU, KMUPRM
C
      DO 100 I=1,NMODES
C
C CALCULATE ORDER FOR BESSEL FUNCTION EVALUATION
C
      M              = IABS( MODE(I) )
      IF( M .NE. 0 )                     GO TO 10
       ISIGN         = 1
      JSIGN          = -1
      BRAKTL         = .1
                                         GO TO 20
   10 ISIGN          = MODE(I) / M
      JSIGN          = ISIGN
```

```
      BRAKTL        = M
      IF( ISIGN .GE. 0 )            GO TO 20
C
C NEGATIVE ORDER. IF EVEN, SIGN OF BESSEL FUNCTION WILL BE +1. IF ODD,
C  SIGN OF BESSEL FUNCTION WILL BE -1.
C
      IF( ( M / 2 ) * 2 .EQ. M )    ISIGN = 1
C
   20 NUMMUS        = MU(I) + 1
C
C CALCULATE CHARACTERISTIC NUMBER KMU CORRESPONDING TO MODE(I) AND MU(I)
C  THE VALUE OF KMU WILL BE THE MU(I)+1 ROOT OF THE EQUATION DEFINING
C  THE SYSTEM OF SIMULTANEOUS EQUATIONS
C
      KMUPRM        = 0.0
      DO 40 J=1,NUMMUS
      IF( M .EQ. 0 .AND. J .EQ. 1 )   GO TO 40
      CALL KMUCAL( BRAKTL, DELKMU, KMUPRM, BRAKTR )
      BRAKTL        = BRAKTR
   40 CONTINUE
      KMU(I)        = KMUPRM
C
C CALCULATE CHARACTERISTIC NUMBER QMU CORRESPONDING TO MODE(I) AND MU(I)
C  IF THE HUB/TIP RATIO IS ZERO, SET QMU TO ZERO AND CONTINUE
C
      IF( SIGMA )                   60, 60, 80
   60 QMU(I)        = 0.0
                                    GO TO 100
   80 IF( KMU(I) )                  90, 60, 90
   90 CALL BESJ( KMUPRM, M-JSIGN, EMM1, TOL, IER )
      EMM1          = ISIGN * JSIGN * EMM1
      CALL BESJ( KMUPRM, M, EMJ, TOL, IER2 )
      CALL BESY( KMUPRM, M, EMY, IER3 )
      EMJ           = ISIGN * EMJ
      EMY           = ISIGN * EMY
      CALL BESY( KMUPRM, M-JSIGN, EYM1, IER4 )
      EYM1          = ISIGN * JSIGN * EYM1
      A             = EMM1 - ( M * JSIGN * EMJ ) / KMUPRM
      B             = EYM1 - ( M * JSIGN * EMY ) / KMUPRM
      SIGMAK        = SIGMA * KMUPRM
      CALL BESJ( SIGMAK, M-JSIGN, EMM1, TOL, IER5 )
      CALL BESJ( SIGMAK, M, EMJ, TOL, IER6 )
      CALL BESY( SIGMAK, M, EMY, IER7 )
      CALL BESY( SIGMAK, M-JSIGN, EYM1, IER8 )
      EMM1          = ISIGN * JSIGN * EMM1
      EYM1          = ISIGN * JSIGN * EYM1
      EMJ           = ISIGN * EMJ
      EMY           = ISIGN * EMY
      C             = EMM1 - ( M * JSIGN * EMJ ) / SIGMAK
      D             = EYM1 - ( M * JSIGN * EMY ) / SIGMAK
      QMU(I)        = - ( A + C * SIGMA ) / ( B + D * SIGMA )
  100 CONTINUE
 9999 RETURN
```

```
      END
      SUBROUTINE PRINT
C
C THIS SUBROUTINE PRINTS INPUT AND CALCULATED VALUES
C
      COMMON /CNSTNT/ NMEAS, NPRED, NMODES, SIGMA, B, MX, FREQ, A,
     1                OMEGA
      COMMON /REFS/   XREF, RREF, THREF
      COMMON /MEASUR/ XM(50), RM(50), THETAM(50), BETA(50), PSI(50)
      COMMON /PREDCT/ XP(50), RP(50), THETAP(50)
      COMMON /MODES/  MODE(50), MU(50), IWAVE(50)
      COMMON /EMUS/   EMU(50,50), EMUP(50,50), EMUPRM(50,50), IEMPRT
      COMMON /ANGLES/ DEGRAD, RADDEG
      COMMON /OUTPUT/ AMPR(50), PHASER(50)
      COMMON /REFCON/ REFPRS
      COMMON /WAVENO/ KX(50)
      COMMON /KQMU/   KMU(50), QMU(50)
      COMMON /APHIMU/ AMU(50), PHIMU(50), ICHECK
      COMMON /DVIATE/ SIGMAX, SIGMAR, SIGMAT, SIGMAB, SIGMAP, SIGAM(50),
     1                SIGTM(50), SIGAMC(50), IDEV
      COMMON /DERSUM/ ARNSUM(50), PRNSUM(50), AXNSUM(50), PXNSUM(50),
     1                ATNSUM(50), PTNSUM(50), ABNSUM(50), PBNSUM(50),
     2                APNSUM(50), PPNSUM(50)
      COMMON /DERIVS/ DAMDRN(50,50), DAMRNS(50,50), DPHDRN(50,50),
     1                DPHRNS(50,50), DAMDXN(50,50), DAMXNS(50,50),
     2                DPHDXN(50,50), DPHXNS(50,50), DAMDTN(50,50),
     3                DAMTNS(50,50), DPHDTN(50,50), DPHTNS(50,50),
     4                DAMDBN(50,50), DAMBNS(50,50), DPHDBN(50,50),
     5                DPHBNS(50,50), DAMDPN(50,50), DAMPNS(50,50),
     6                DPHDPN(50,50), DPHPNS(50,50)
      COMMON /MCOMP/  XACOMP(50), RACOMP(50), TACOMP(50), BACOMP(50),
     1                PACOMP(50), XPCOMP(50), RPCOMP(50), TPCOMP(50),
     2                BPCOMP(50), PPCOMP(50)
      DIMENSION AMUDB(50), DEVDB(50)
      COMPLEX KX
      REAL KMU, MX
C
C CONVERT INTERNAL UNITS TO OUTPUT UNITS
C
      THREF         = RADDEG * THREF
C
      DO 20 I=1,NMEAS
      THETAM(I)     = RADDEG * THETAM(I)
      PSI(I)        = RADDEG * PSI(I)
   20 CONTINUE
C
      DO 25 I=1,NPRED
      THETAP(I)     = RADDEG * THETAP(I)
      PHASER(I)     = RADDEG * PHASER(I)
      AMPR(I)       = 20. * ALOG10( AMPR(I) / REFPRS )
   25 CONTINUE
C
      DO 30 I=1,NMODES
```

```
      KX(I)          = RADDEG * KX(I)
   30 CONTINUE
C
      DO 35 I=1,NMODES
      PHIMU(I)       = RADDEG * PHIMU(I)
      AMUDB(I)       = 20. * ALOG10( AMU(I) / REFPRS )
   35 CONTINUE
C
      IF( IDEV .LE. 0 )                 GO TO 50
      DO 40 I=1,NMODES
      DEVDB(I)       = 20. * ALOG10( 1. + SIGAM(I) / AMU(I) )
   40 CONTINUE
C
      DO 45 I=1,NMODES
      DIVSOR         = 1. / AMU(I)
      XACOMP(I)      = DIVSOR * SQRT( XACOMP(I) )
      RACOMP(I)      = DIVSOR * SQRT( RACOMP(I) )
      TACOMP(I)      = DIVSOR * SQRT( TACOMP(I) )
      BACOMP(I)      = DIVSOR * SQRT( BACOMP(I) )
      PACOMP(I)      = DIVSOR * SQRT( PACOMP(I) )
      XPCOMP(I)      = SQRT( XPCOMP(I) )
      RPCOMP(I)      = SQRT( RPCOMP(I) )
      TPCOMP(I)      = SQRT( TPCOMP(I) )
      BPCOMP(I)      = SQRT( BPCOMP(I) )
      PPCOMP(I)      = SQRT( PPCOMP(I) )
   45 CONTINUE
C
   50 DO 55 I=1,NMODES
      DIVSOR         = 1.0 / AMU(I)
      ARNSUM(I)      = DIVSOR * SQRT( ARNSUM(I) )
      AXNSUM(I)      = DIVSOR * SQRT( AXNSUM(I) )
      ATNSUM(I)      = DIVSOR * SQRT( ATNSUM(I) )
      ABNSUM(I)      = DIVSOR * SQRT( ABNSUM(I) )
      APNSUM(I)      = DIVSOR * SQRT( APNSUM(I) )
      PRNSUM(I)      = SQRT( PRNSUM(I) )
      PXNSUM(I)      = SQRT( PXNSUM(I) )
      PTNSUM(I)      = SQRT( PTNSUM(I) )
      PBNSUM(I)      = SQRT( PBNSUM(I) )
      PPNSUM(I)      = SQRT( PPNSUM(I) )
   55 CONTINUE
C
C CONVERT ANY NEGATIVE ANGLES TO POSITIVE ANGLES FOR PRINTING
C
      CALL ANGPOS( PHIMU, NMODES )
      CALL ANGPOS( PHASER, NPRED )
C
C PRINT INPUT VARIABLES
C
      WRITE(6,9000)
 9000 FORMAT( 1H1, T45, '***  MODAL CALCULATION COMPUTER PROGRAM  ****' )
      WRITE(6,9001) NMEAS, NPRED, NMODES
 9001 FORMAT( //, T56, '... INPUT VARIABLES ...', //, T5, 'NUMBER OF MEA
     1SUREMENT LOCATIONS = ', I2, T51, 'NUMBER OF PREDICTION LOCATIONS =
```

```
     2 ', I2, T96, 'NUMBER OF (MODE,MU) SETS = ', I2 )
      WRITE(6,9002)
 9002 FORMAT( //, 1X, '... INPUT MODES ...', //, T5, 'MODE', T14,
     1'CIRCUMFERENTIAL', T34, 'RADIAL', T47, 'WAVE', /, T16, 'MODE NUMBE
     2R', T34, 'ORDER', T45, 'INDICATOR' )
      DO 60 I=1,NMODES
      WRITE(6,9003) I, MODE(I), MU(I), IWAVE(I)
 9003 FORMAT( 5X, I2, 11X, I4, 13X, I2, 11X, I2 )
   60 CONTINUE
C
C PRINT REFERENCE VALUES
C
      WRITE(6,9004) REFPRS
 9004 FORMAT( ///, 1X, '... REFERENCE VALUES ...', //, T5, 'REFERENCE PR
     1ESSURE = ', E9.4 )
C
C PRINT TEST GEOMETRY AND CONDITIONS
C
      WRITE(6,9005) SIGMA, B, MX, FREQ, A, OMEGA
 9005 FORMAT( ///, 1X, '... TEST GEOMETRY AND CONDITIONS ...', //, T5,
     1'HUB / TIP RATIO = ', F8.3, T42, 'OUTER RADIUS OF DUCT = ', F8.2,
     2T84, 'AXIAL MACH NUMBER = ', F8.2, /, T5, 'FREQUENCY = ', F8.2,
     3T42, 'SPEED OF SOUND = ', F8.2, T84, 'RADIAN FREQUENCY = ',
     4F10.2 )
C
C PRINT CALCULATED MODAL AMPLITUDES AND PHASES
C
      WRITE(6,9000)
      WRITE(6,9006)
 9006 FORMAT( //, T45, '... MODAL AMPLITUDE AND PHASE CALCULATION ...',
     1//, 1X, '... CALCULATED MODAL AMPLITUDES AND PHASES ...', //, T5,
     2'MODE', T12, 'CIRCUMFERENTIAL', T30, 'RADIAL', T41, 'WAVE', T47,
     32(6X,'AMPLITUDE'), T84, 'PHASE', T98, 'AXIAL WAVE NUMBER',
     4T125, 'KMU', /, T14, 'MODE NUMBER', T30, 'ORDER', T39, 'INDICATOR'
     5, T53, '(PRESSURE)', T71, '(DB)', T82, '(DEGREES)', T98, 'REAL',
     6T109, 'IMAGINARY', / )
      DO 80 I=1,NMODES
      WRITE(6,9007) I, MODE(I), MU(I), IWAVE(I), AMU(I), AMUDB(I),
     1              PHIMU(I), KX(I), KMU(I)
 9007 FORMAT( 5X, I2, 9X, I4, 11X, I2, 8X, I2, 6X, E12.6, 3X, E12.6,
     14(3X,F10.4) )
   80 CONTINUE
C
C PRINT REFERENCE LOCATION VALUES
C
      WRITE(6,9008) XREF, RREF, THREF
 9008 FORMAT( ///, 1X, '... REFERENCE LOCATION ...', //, T10, 'X', T27,
     1'R', T42, 'THETA', //, 4X, E12.6, 2(5X,E12.6) )
C
C PRINT MEASUREMENT LOCATION VALUES
C
      IF( ICHECK .GT. 0 )              GO TO 110
      WRITE(6,9009)
```

```
 9009 FORMAT( ///, 1X, '... MEASUREMENT LOCATIONS ...', //, T5, 'LOCATIO
     1N', T23, 'X', T40, 'R', T55, 'THETA', T70, 'AMPLITUDE', T89,
     2'PHASE', /, T6, 'NUMBER', T73, '(B)', T89, '(PSI)', / )
      DO 100 I=1,NMEAS
      WRITE(6,9010) I, XM(I), RM(I), THETAM(I), BETA(I), PSI(I)
 9010 FORMAT( 7X, I2, 3X, 3(5X,F12.6), 5X, E12.6, 5X, F12.6 )
  100 CONTINUE
C
C PRINT PREDICTION LOCATION VALUES
C
  110 WRITE(6,9011)
 9011 FORMAT( ///, 1X, '... PREDICTION LOCATIONS ...', //, T5, 'LOCATION
     1', T23, 'X', T40, 'R', T55, 'THETA', T70, 'AMPLITUDE', T89,
     2'PHASE', /, T6, 'NUMBER', T69, '(RESULTANT)', T86, '(RESULTANT)',
     3/ )
      DO 120 I=1,NPRED
      WRITE(6,9010) I, XP(I), RP(I), THETAP(I), AMPR(I), PHASER(I)
  120 CONTINUE
C
C PRINT SENSITIVITY CALCULATION VALUES IF NOT A CHECK CASE
C
      IF( ICHECK .GT. 0 )               GO TO 250
      WRITE(6,9000)
      WRITE(6,9012)
 9012 FORMAT( //, T45, '... SENSITIVITY COEFFICIENT CALCULATION ...' )
      IF( IDEV .LE. 0 )                 GO TO 190
C
C PRINT ERROR SOURCE STANDARD DEVIATION VALUES
C
      WRITE(6,9013) SIGMAX, SIGMAR, SIGMAT, SIGMAB, SIGMAP
 9013 FORMAT( ///, 1X, '... ERROR SOURCE STANDARD DEVIATIONS ...', //,
     1T7, 'SIGMA X', T24, 'SIGMA R', T39, 'SIGMA THETA', T58, 'SIGMA B',
     2T74, 'SIGMA PSI', /, 4X, E12.6, 4(5X,E12.6) )
C
C PRINT MODAL STANDARD DEVIATIONS
C
      WRITE(6,9014)
 9014 FORMAT( ///, 1X, '... NORMALIZED STANDARD DEVIATIONS DUE TO ALL ER
     1ROR SOURCES ...', //, T5,
     1'MODE', T12, 'CIRCUMFERENTIAL', T30, 'RADIAL', T41, 'WAVE', T52,
     2'NORMALIZED AMPLITUDE', T80, 'AMPLITUDE', T105,
     3'PHASE', /, T14, 'MODE NUMBER', T30, 'ORDER', T39,
     4'INDICATOR', T57, 'DEVIATION', T80, 'DEVIATION', T103, 'DEVIATION'
     5, /, T83, '(DB)', T103, '(DEGREES)', / )
      DO 140 I=1,NMODES
      WRITE(6,9015) I, MODE(I), MU(I), IWAVE(I), SIGAMC(I), DEVDB(I),
     1             SIGTM(I)
 9015 FORMAT( 5X, I2, 9X, I4, 11X, I2, 8X, I2, 1X, 3(11X,E12.6) )
  140 CONTINUE
C
C PRINT MODAL STANDARD DEVIATION COMPONENTS
C
      WRITE(6,9016)
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
 9016 FORMAT( ///, 1X, '... NORMALIZED STANDARD DEVIATION COMPONENTS (ER
     1ROR SOURCE DEVIATION TIMES RMS SUM OF NORMALIZED INFLUENCE COEFFIC
     2IENTS) ...' )
      WRITE(6,9017)
 9017 FORMAT( //, T5, 'MODE', T26, 'AMPLITUDE DUE TO ERROR IN', T92,
     1'PHASE DUE TO ERROR IN',  /, T15, 'X', T27, 'R', T37, 'THETA',
     2T51, 'B', T62, 'PSI', T79, 'X', T91, 'R', T101, 'THETA', T115,
     3'B', T126, 'PSI', / )
      DO 160 I=1,NMODES
      WRITE(6,9018) I, XACOMP(I), RACOMP(I), TACOMP(I), BACOMP(I),
     1              PACOMP(I), XPCOMP(I), RPCOMP(I), TPCOMP(I),
     2              BPCOMP(I), PPCOMP(I)
 9018 FORMAT( 5X, I2, 5(1X,E11.4), 4X, 5(1X,E11.4) )
  160 CONTINUE
C
C PRINT INFLUENCE COEFFICIENTS
C
  180 WRITE(6,9019)
 9019 FORMAT( ///, 1X, '... RMS SUM OF NORMALIZED INFLUENCE COEFFICIENTS
     1 ...' )
      WRITE(6,9017)
      DO 200 I=1,NMODES
      WRITE(6,9018) I, AXNSUM(I), ARNSUM(I), ATNSUM(I), ABNSUM(I),
     1              APNSUM(I), PXNSUM(I), PRNSUM(I), PTNSUM(I),
     2              PBNSUM(I), PPNSUM(I)
  200 CONTINUE
C
C PRINT PARTIAL DERIVATIVES
C
      WRITE(6,9020)
 9020 FORMAT( ///, 1X, '... INFLUENCE COEFFICIENTS (PARTIAL DERIVATIVES
     1) ...' )
      DO 240 I=1,NMEAS
      WRITE(6,9021) I
 9021 FORMAT( //, T45, 'INFLUENCE COEFFICIENTS FOR MEASUREMENT LOCATION
     1 ', I2 )
      WRITE(6,9017)
      DO 220 J=1,NMODES
      WRITE(6,9018) J, DAMDXN(I,J), DAMDRN(I,J), DAMDTN(I,J),
     1              DAMDBN(I,J), DAMDPN(I,J), DPHDXN(I,J),
     2              DPHDRN(I,J), DPHDTN(I,J), DPHDBN(I,J),
     3              DPHDPN(I,J)
  220 CONTINUE
  240 CONTINUE
C
C PRINT CHARACTERISTIC E-FUNCTION VALUES IF REQUESTED
C
  250 IF( IEMPRT .LE. 0 )             GO TO 9999
      WRITE(6,9000)
      WRITE(6,9022)
 9022 FORMAT( //, T23, '... CHARACTERISTIC E-FUNCTION VALUES FOR MODAL A
     1MPLITUDE AND PHASE CALCULATIONS ...' )
      WRITE(6,9023) ( I,I=1,NMODES )
```

```
 9023 FORMAT( ///, 1X, '... MEASUREMENT LOCATIONS ...', //, 1X, 'LOCATIO
     1N', T68, 'MODES', //, 8X, 15(6X,I2), / )
      DO 260 J=1,NMEAS
      WRITE(6,9024) J, ( EMU(I,J),I=1,NMODES )
 9024 FORMAT( 4X, I2, 5X, 15(1X,F7.3) )
  260 CONTINUE
      WRITE(6,9025) ( I,I=1,NMODES )
 9025 FORMAT( ///, 1X, '... PREDICTION LOCATIONS ...', //, 1X, 'LOCATION
     1', T69, 'MODES', //, 9X, 15(6X,I2), / )
      DO 280 J=1,NPRED
      WRITE(6,9024) J, ( EMUP(I,J),I=1,NMODES )
  280 CONTINUE
 9999 RETURN
      END
      SUBROUTINE SIMEQC( A, C, NA, NB, SNGUL )
C
C THIS SUBROUTINE SOLVES A NA X NA SYSTEM OF SIMULTANEOUS EQUATIONS
C  HAVING COMPLEX COEFFICIENTS USING GAUSSIAN ELIMINATION METHOD.
C
      COMPLEX A(50,1), C(1), SAVE, ZERO
      DATA ZERO / (0.0,0.0) /
C
      SNGUL         = 0.0
      DO 240 I=1,NA
C
C FIND MAXIMUM ELEMENT IN JTH COLUMN, ROWS I+1 TO NA
C
      JZ            = I + 1
      IF( I - NA )                     20, 100, 20
   20 VALMX         = CABS( A(I,I) )
      MZ            = I
      DO 60 KZ=JZ,NA
      B             = CABS( A(KZ,I) )
      IF( VALMX - B )                  40, 40, 60
   40 VALMX         = B
      MZ            = KZ
   60 CONTINUE
C
C INTERCHANGE ROW CONTAINING MAXIMUM WITH ITH ROW
C
      DO 80 IK=1,NB
      SAVE          = A(I,IK)
      A(I,IK)       = A(MZ,IK)
      A(MZ,IK)      = SAVE
   80 CONTINUE
C
C NORMALIZE ITH ROW
C
  100 IF( REAL( A(I,I) ) )             160, 120, 160
  120 IF( AIMAG( A(I,I) ) )            160, 140, 160
C
C ERROR - COEFFICIENT MATRIX IS SINGULAR
C
```

```
  140 SNGUL         = 1.0
                                              GO TO 9999
  160 DO 220 LZ=JZ,NB
      A(I,LZ)       = A(I,LZ) / A(I,I)
      IF( JZ - NB )                           180, 260, 260
  180 DO 200 NZ=JZ,NA
      A(NZ,LZ)      = A(NZ,LZ) - A(NZ,I) * A(I,LZ)
  200 CONTINUE
  220 CONTINUE
  240 CONTINUE
C
C SOLVE FOR COEFFICIENTS
C
  260 DO 280 MZ=1,NA
      C(MZ)         = ZERO
  280 CONTINUE
      C(NA)         = A(NA,NB)
      NC            = NA - 1
      II            = 1
      DO 320 IZ=1,NC
      KK            = NA
      LZ            = NA - IZ
      C(LZ)         = A(LZ,NB)
      DO 300 N=1,II
      C(LZ)         = C(LZ) - C(KK) * A(LZ,KK)
      KK            = KK - 1
  300 CONTINUE
      II            = II + 1
  320 CONTINUE
 9999 RETURN
      END
      SUBROUTINE SOLVE
C
C THIS SUBROUTINE SETS UP AND SOLVES THE EQUATION SYSTEM ASSOCIATED WITH
C  MEASUREMENT LOCATION PARAMETERS
C
      COMMON /BESSL/ DUM3(5), PI
      COMMON /REFS/   XREF, RREF, THREF
      COMMON /MEASUR/ XM(50), RM(50), THETAM(50), BETA(50), PSI(50)
      COMMON /MODES/  MODE(50), DUM1(100)
      COMMON /EMUS/   EMU(50,50), EMUP(50,50), EMUPRM(50,50), IEMPRT
      COMMON /APHIMU/ AMU(50), PHIMU(50), ICHECK
      COMMON /CNSTNT/ NMEAS, NPRED, NMODES, SIGMA, B, DUM2(4)
      COMMON /WAVENO/ KX(50)
      COMMON /LMATRX/ EQ1(50,51)
      COMMON /DVIATE/ DUM4(155), IDEV
      COMPLEX KX, EXPNT, EQ1, EQ(50,51), ANSWER(50)
C
C SET UP COEFFICIENT MATRIX
C
      DO 40 I=1,NMEAS
      DX            = XM(I) - XREF
      DR            = RM(I) - RREF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DTHETA        = THETAN(I) - THREF
C
C CALCULATE CHARACTERISTIC E-FUNCTION VALUES AND DERIVATIVES FOR RPRIME
C
   10 RPRIME        = DR / B
      CALL EMUCAL( RPRIME, EMU(1,I), EMUPRM(1,I), 1 )
C
   15 DO 20 J=1,NMODES
      EXPNT         = CMPLX( 0.0, REAL( KX(J) ) * DX + MODE(J) * DTHETA )
      EQ(I,J)       = EMU(J,I) * CEXP( EXPNT ) * EXP( -DX *
     1                AIMAG( KX(J) ) )
      EQ1(I,J)      = EQ(I,J)
   20 CONTINUE
C
C SET UP RIGHT HAND SIDE
C
      EQ(I,NMODES+1) = BETA(I) * CEXP( CMPLX( 0.0, PSI(I) ) )
      EQ1(I,NMODES+1) = EQ(I,NMODES+1)
   40 CONTINUE
C
C SOLVE EQUATION SYSTEM
C
      CALL SIMEQC( EQ, ANSWER, NMEAS, NMODES+1, SNGULR )
      IF( SNGULR )                          60, 80, 60
C
C ERROR - SINGULAR MATRIX. TERMINATE EXECUTION
C
   60 NM1           = NMODES + 1
      WRITE(6,1000) ( ( EQ1(I,J),J=1,NM1 ),I=1,NMEAS )
 1000 FORMAT( //, 5X, 'COEFFICIENT MATRIX IS SINGULAR', ( /, 1X,
     110G13.6 ) )
                                            STOP
C
C CALCULATE AMPLITUDE AND PHASE VALUES
C
   80 DO 100 I=1,NMODES
      AMU(I)        = CABS( ANSWER(I) )
      PHIMU(I)      = ATAN2( AIMAG( ANSWER(I) ), REAL( ANSWER(I) ) )
  100 CONTINUE
C
 9999 RETURN
      END
      SUBROUTINE SENSTY( Q, NDIM )
C
C THIS SUBROUTINE CALCULATES THE SENSITIVITY COEFFICIENTS ASSOCIATED
C  WITH THE EQUATION SYSTEM
C
      DIMENSION EMUAVG(50,50), IROW(50), ICOL(50)
      COMPLEX KX, EQ1, TERM, ZERO, SUM, Q(NDIM,NDIM), DET
      REAL KMU
      COMMON /DERIVS/ DAMDRN(50,50), DAMRNS(50,50), DPHDRN(50,50),
     1                DPHRNS(50,50), DAMDXN(50,50), DAMXNS(50,50),
     2                DPHDXN(50,50), DPHXNS(50,50), DAMDTN(50,50),
```

```
     3                DAMTNS(50,50), DPHDTN(50,50), DPHTNS(50,50),
     4                DAMDBN(50,50), DAMBNS(50,50), DPHDBN(50,50),
     5                DPHBNS(50,50), DAMDPN(50,50), DAMPNS(50,50),
     6                DPHDPN(50,50), DPHPNS(50,50)
      COMMON /DERSUM/ ARNSUM(50), PRNSUM(50), AXNSUM(50), PXNSUM(50),
     1                ATNSUM(50), PTNSUM(50), ABNSUM(50), PBNSUM(50),
     2                APNSUM(50), PPNSUM(50)
      COMMON /MCOMP/  XACOMP(50), RACOMP(50), TACOMP(50), BACOMP(50),
     1                PACOMP(50), XPCOMP(50), RPCOMP(50), TPCOMP(50),
     2                BPCOMP(50), PPCOMP(50)
      COMMON /DVIATE/ SIGMAX, SIGMAR, SIGMAT, SIGMAB, SIGMAP, SIGAM(50),
     1                SIGTM(50), SIGAMC(50), IDEV
      COMMON /CNSTNT/ NMEAS, NPRED, NMODES, SIGMA, B, DUM1(4)
      COMMON /MEASUR/ DUM2(150), BETA(50), PSI(50)
      COMMON /APHIMU/ AMU(50), PHIMU(50)
      COMMON /EMUS/   EMU(50,50), EMUP(50,50), EMUPRM(50,50), IEMPRT
      COMMON /REFCON/ REFPRS
      COMMON /ANGLES/ DEGRAD, RADDEG
      COMMON /KQMU/   KMU(50), QMU(50)
      COMMON /WAVENO/ KX(50)
      COMMON /LMATRX/ EQ1(50,51)
      COMMON /MODES/  MODE(50), MU(50)
      DATA ZERO / (0.0,0.0) /
C
C CALCULATE INVERSE OF MEASUREMENT LOCATION MATRIX
C
      DO 10 J=1,NDIM
      DO 10 I=1,NDIM
      Q(I,J)        = EQ1(I,J)
   10 CONTINUE
      CALL INVERT( Q, NDIM, DET, IROW, ICOL )
C
C CALCULATE AVERAGE CHARACTERISTIC E-FUNCTION VALUES
C
      DO 40 J=1,NMODES
      DO 20 I=1,NMEAS
      EMUAVG(I,J)   = KMU(J) * EMUPRM(J,I) / ( EMU(J,I) * B )
   20 CONTINUE
   40 CONTINUE
C
C CALCULATE DERIVATIVES WITH RESPECT TO R
C
      DO 100 K=1,NMEAS
      SUM           = ZERO
      DO 60 J=1,NMODES
      SUM           = EMUAVG(K,J) * EQ1(K,J) * AMU(J) *
     1                CEXP( CMPLX( 0., PHIMU(J) ) ) + SUM
   60 CONTINUE
      DO 80 L=1,NMODES
      TERM          = Q(L,K) * SUM * CEXP( CMPLX( 0., -PHIMU(L) ) )
      DAMDRN(K,L)   = - REAL( TERM )
      DAMRNS(K,L)   = DAMDRN(K,L) ** 2
      DPHDRN(K,L)   = - AIMAG( TERM / AMU(L) ) * RADDEG
```

```
      DPHRNS(K,L)  = DPHDRN(K,L) ** 2
   80 CONTINUE
  100 CONTINUE
C
C CALCULATE DERIVATIVES WITH RESPECT TO X
C
      DO 160 K=1,NMEAS
      SUM            = ZERO
      DO 120 J=1,NMODES
      SUM            = EQ1(K,J) * KX(J) * AMU(J) * CEXP( CMPLX( 0.,
     1                   PHIMU(J) ) ) + SUM
  120 CONTINUE
      DO 140 L=1,NMODES
      TERM           = Q(L,K) * SUM * CEXP( CMPLX( 0., -PHIMU(L) ) )
      DAMDXN(K,L)    = AIMAG( TERM )
      DAMXNS(K,L)    = DAMDXN(K,L) ** 2
      DPHDXN(K,L)    = - REAL( TERM / AMU(L) ) * RADDEG
      DPHXNS(K,L)    = DPHDXN(K,L) ** 2
  140 CONTINUE
  160 CONTINUE
C
C CALCULATE DERIVATIVES WITH RESPECT TO THETA
C
      DO 220 K=1,NMEAS
      SUM            = ZERO
      DO 180 J=1,NMODES
      SUM            = EQ1(K,J) * MODE(J) * AMU(J) *
     1                   CEXP( CMPLX( 0., PHIMU(J) ) ) + SUM
  180 CONTINUE
      DO 200 L=1,NMODES
      TERM           = Q(L,K) * SUM * CEXP( CMPLX( 0., -PHIMU(L) ) )
      DAMDTN(K,L)    = AIMAG( TERM ) / RADDEG
      DAMTNS(K,L)    = DAMDTN(K,L) ** 2
      DPHDTN(K,L)    = -REAL( TERM / AMU(L) )
      DPHTNS(K,L)    = DPHDTN(K,L) ** 2
  200 CONTINUE
  220 CONTINUE
C
C CALCULATE DERIVATIVES WITH RESPECT TO BN
C
      DO 260 L=1,NMODES
      DO 240 K=1,NMEAS
      TERM           = Q(L,K) * CEXP( CMPLX( 0., PSI(K) - PHIMU(L) ) )
      DAMDBN(K,L)    = REAL( TERM )
      DAMBNS(K,L)    = DAMDBN(K,L) ** 2
      DPHDBN(K,L)    = AIMAG( TERM / AMU(L) ) * RADDEG
      DPHBNS(K,L)    = DPHDBN(K,L) ** 2
  240 CONTINUE
  260 CONTINUE
C
C CALCULATE DERIVATIVES WITH RESPECT TO PSI
C
      DO 300 L=1,NMODES
```

```
      DO 280 K=1,NMEAS
      TERM           = Q(L,K) * BFTA(K) * CEXP( CMPLX( O., PSI(K) -
     1                 PHIMU(L) ) )
      DAMUPN(K,L)    = - AIMAG( TERM ) / RADDEG
      DAMPNS(K,L)    = DAMDPN(K,L) ** 2
      DPHDPN(K,L)    = REAL( TERM / AMU(L) )
      DPHPNS(K,L)    = DPHDPN(K,L) ** 2
  280 CONTINUE
  300 CONTINUE
C
C CALCULATE SUMS OF DERIVATIVES
C
      DO 340 J=1,NMODES
      SUMM           = 0.0
      DO 320 I=1,NMEAS
      SUMM           = DAMRNS(I,J) + SUMM
  320 CONTINUE
      ARNSUM(J)      = SUMM
  340 CONTINUE
C
      DO 360 J=1,NMODES
      SUMM           = 0.0
      DO 360 I=1,NMEAS
      SUMM           = DPHRNS(I,J) + SUMM
  360 CONTINUE
      PRNSUM(J)      = SUMM
  380 CONTINUE
C
      DO 420 J=1,NMODES
      SUMM           = 0.0
      DO 400 I=1,NMEAS
      SUMM           = DAMXNS(I,J) + SUMM
  400 CONTINUE
      AXNSUM(J)      = SUMM
  420 CONTINUE
C
      DO 460 J=1,NMODES
      SUMM           = 0.0
      DO 440 I=1,NMEAS
      SUMM           = DPHXNS(I,J) + SUMM
  440 CONTINUE
      PXNSUM(J)      = SUMM
  460 CONTINUE
C
      DO 500 J=1,NMODES
      SUMM           = 0.0
      DO 480 I=1,NMEAS
      SUMM           = DAMINS(I,J) + SUMM
  480 CONTINUE
      AINSUM(J)      = SUMM
  500 CONTINUE
C
      DO 540 J=1,NMODES
```

```
      SUMM           = 0.0
      DO 520 I=1,NMEAS
      SUMM           = DPHTNS(I,J) + SUMM
  520 CONTINUE
      PTNSUM(J)      = SUMM
  540 CONTINUE
C
      DO 580 J=1,NMODES
      SUMM           = 0.0
      DO 560 I=1,NMEAS
      SUMM           = DAMBNS(I,J) + SUMM
  560 CONTINUE
      ABNSUM(J)      = SUMM
  580 CONTINUE
C
      DO 620 J=1,NMODES
      SUMM           = 0.0
      DO 600 I=1,NMEAS
      SUMM           = DPHBNS(I,J) + SUMM
  600 CONTINUE
      PBNSUM(J)      = SUMM
  620 CONTINUE
C
      DO 660 J=1,NMODES
      SUMM           = 0.0
      DO 640 I=1,NMEAS
      SUMM           = DAMPNS(I,J) + SUMM
  640 CONTINUE
      APNSUM(J)      = SUMM
  660 CONTINUE
C
      DO 700 J=1,NMODES
      SUMM           = 0.0
      DO 680 I=1,NMEAS
      SUMM           = DPHPNS(I,J) + SUMM
  680 CONTINUE
      PPNSUM(J)      = SUMM
  700 CONTINUE
C
C CALCULATE COEFFICIENTS OF DEVIATION FOR EACH MODE IF REQUESTED
C
      IF( IDEV .EQ. 0 )                   GO TO 9999
      SIGR           = SIGMAR ** 2
      SIGX           = SIGMAX ** 2
      SIGT           = SIGMAT ** 2
      SIGB           = SIGMAB ** 2
      SIGP           = SIGMAP ** 2
C
C CALCULATE COMPONENTS OF MODAL STANDARD DEVIATIONS
C
      DO 720 I=1,NMODES
      XACOMP(I)      = AXNSUM(I) * SIGX
      RACOMP(I)      = ARNSUM(I) * SIGR
```

```
      TACOMP(I)     = ATNSUM(I) * SIGT
      BACOMP(I)     = ABNSUM(I) * SIGB
      PACOMP(I)     = APNSUM(I) * SIGP
  720 CONTINUE
      DO 740 I=1,NMODES
      XPCOMP(I)     = PXNSUM(I) * SIGX
      RPCOMP(I)     = PRNSUM(I) * SIGR
      TPCOMP(I)     = PTNSUM(I) * SIGT
      BPCOMP(I)     = PBNSUM(I) * SIGB
      PPCOMP(I)     = PPNSUM(I) * SIGP
  740 CONTINUE
      DO 760 I=1,NMODES
      SIGAM(I)      = SQRT( XACOMP(I) + RACOMP(I) + TACOMP(I) +
     1                BACOMP(I) + PACOMP(I) )
      SIGTM(I)      = SQRT( XPCOMP(I) + RPCOMP(I) + TPCOMP(I) +
     1                BPCOMP(I) + PPCOMP(I) )
      SIGAMC(I)     = SIGAM(I) / AMU(I)
  760 CONTINUE
C
 9999 RETURN
      END
      SUBROUTINE INVERT( A, N, D, L, M )
C
C THIS SUBROUTINE INVERTS A COMPLEX MATRIX. THIS PROCEDURE WAS ADAPTED
C  FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE
C
      DIMENSION L(1), M(1)
      COMPLEX A(1), BIGA, HOLD, D, ONE, ZERO
      DATA ZERO / (0.0,0.0) /, ONE / (1.0,0.0) /
C
C SEARCH FOR THE LARGEST ELEMENT
C
      D             = ONE
      NK            = -N
      DO 380 K=1,N
      NK            = N + NK
      L(K)          = K
      M(K)          = K
      KK            = K + NK
      BIGA          = A(KK)
      DO 60 J=K,N
      IZ            = N * ( J - 1 )
      DO 60 I=K,N
      IJ            = IZ + I
   20 IF( CABS( BIGA ) - CABS( A(IJ) ) )  40, 60, 60
   40 BIGA          = A(IJ)
      L(K)          = I
      M(K)          = J
   60 CONTINUE
C
C INTERCHANGE ROWS
C
      J             = L(K)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF( J - K )                   120, 120, 80
   80 KI          = K - N
      DO 100 I=1,N
      KI          = N + KI
      HOLD        = -A(KI)
      JI          = KI - K + J
      A(KI)       = A(JI)
      A(JI)       = HOLD
  100 CONTINUE
C
C INTERCHANGE COLUMNS
C
  120 I           = M(K)
      IF( I - K )                   180, 180, 140
  140 JP          = N * ( I - 1 )
      DO 160 J=1,N
      JK          = NK + J
      JI          = JP + J
      HOLD        = -A(JK)
      A(JK)       = A(JI)
      A(JI)       = HOLD
  160 CONTINUE
C
C DIVIDE COLUMN BY MINUS PIVOT ( VALUE OF PIVOT IS CONTAINED IN BIGA )
C
  180 IF( CABS( BIGA ) )            200, 200, 220
  200 D           = ZERO
                                    GO TO 9999
  220 DO 260 I=1,N
      IF( I - K )                   240, 260, 240
  240 IK          = NK + I
      A(IK)       = A(IK) / ( -BIGA )
  260 CONTINUE
C
C REDUCE MATRIX
C
      DO 320 I=1,N
      IK          = NK + I
      HOLD        = A(IK)
      IJ          = I - N
      DO 320 J=1,N
      IJ          = IJ + N
      IF( I - K )                   280, 320, 280
  280 IF( J - K )                   300, 320, 300
  300 KJ          = IJ - I + K
      A(IJ)       = HOLD * A(KJ) + A(IJ)
  320 CONTINUE
C
C DIVIDE ROW BY PIVOT
C
      KJ          = K - N
      DO 360 J=1,N
      KJ          = N + KJ
```

```
      IF( J - K )                              340, 360, 340
  340 A(KJ)           = A(KJ) / BIGA
  360 CONTINUE
C
C CALCULATE DETERMINANT
C
      D               = D * BIGA
C
C REPLACE PIVOT BY RECIPROCAL
C
      A(KK)           = ONE / BIGA
  380 CONTINUE
C
C FINAL ROW AND COLUMN INTERCHANGE
C
      K               = N
  400 K               = K - 1
      IF( K )                                  9999, 9999, 420
  420 I               = L(K)
      IF( I - K )                              480, 480, 440
  440 JQ              = N * ( K - 1 )
      JR              = N * ( I - 1 )
      DO 460 J=1,N
      JK              = JQ + J
      HOLD            = A(JK)
      JI              = JR + J
      A(JK)           = -A(JI)
      A(JI)           = HOLD
  460 CONTINUE
  480 J               = M(K)
      IF( J - K )                              400, 400, 500
  500 KI              = K - N
      DO 520 I=1,N
      KI              = N + KI
      HOLD            = A(KI)
      JI              = KI - K + J
      A(KI)           = -A(JI)
      A(JI)           = HOLD
  520 CONTINUE
                                               GO TO 400
 9999 RETURN
      END
***** ABOVE ACTION SATISFACTORILY COMPLETED *****
```

APPENDIX C
MODAL CALCULATION PROGRAM
SAMPLE CASE

```
***  MODAL CALCULATION COMPUTER PROGRAM  ***

... INPUT VARIABLES ...

NUMBER OF MEASUREMENT LOCATIONS =  3        NUMBER OF PREDICTION LOCATIONS =  3        NUMBER OF (MODE,MU) SETS =  3

... INPUT MODES ...

 MODE     CIRCUMFERENTIAL      RADIAL      WAVE
            MODE NUMBER        ORDER     INDICATOR
   1            -4               0           1
   2            -4               1           1
   3            -4               2           1

... REFERENCE VALUES ...

 REFERENCE PRESSURE = .2000E-03

... TEST GEOMETRY AND CONDITIONS ...

 HUB / TIP RATIO =     0.440        OUTER RADIUS OF DUCT =     25.00        AXIAL MACH NUMBER =     -0.10
 FREQUENCY =  6200.00               SPEED OF SOUND = 34345.00               RADIAN FREQUENCY =   38955.75
```

*** MODAL CALCULATION COMPUTER PROGRAM ***

... MODAL AMPLITUDE AND PHASE CALCULATION ...

... CALCULATED MODAL AMPLITUDES AND PHASES ...

MODE	CIRCUMFERENTIAL MODE NUMBER	RADIAL ORDER	WAVE INDICATOR	AMPLITUDE (PRESSURE)	AMPLITUDE (DB)	PHASE (DEGREES)	AXIAL WAVE NUMBER REAL	AXIAL WAVE NUMBER IMAGINARY	KMU
1	-4	0	1	0.148735E+04	0.137428E+03	126.9814	71.0845	0.0	5.2510
2	-4	1	1	0.276112E+04	0.142801E+03	160.0302	69.0154	0.0	8.7800
3	-4	2	1	0.169032E+04	0.138539E+03	229.2115	65.0738	0.0	12.9210

... REFERENCE LOCATION ...

X	R	THETA
0.0	0.0	0.0

... MEASUREMENT LOCATIONS ...

LOCATION NUMBER	X	R	THETA	AMPLITUDE (B)	PHASE (PSI)
1	0.0	25.000000	0.0	0.228700E+03	329.699951
2	14.976000	25.000000	0.0	0.128900E+03	135.400009
3	29.960007	25.000000	0.0	0.841000E+02	252.200073

... PREDICTION LOCATIONS ...

LOCATION NUMBER	X	R	THETA	AMPLITUDE (RESULTANT)	PHASE (RESULTANT)
1	0.0	25.000000	239.900040	0.121165E+03	90.099747
2	13.525000	25.000000	100.600037	0.115699E+03	357.780762
3	20.300003	25.000000	15.900006	0.115084E+03	67.249161

*** MODAL CALCULATION COMPUTER PROGRAM ***

... SENSITIVITY COEFFICIENT CALCULATION ...

... ERROR SOURCE STANDARD DEVIATIONS ...

SIGMA X	SIGMA R	SIGMA THETA	SIGMA B	SIGMA PSI
0.200000E-02	0.200000E-02	0.200000E-01	0.250000E+02	0.150000E+01

... NORMALIZED STANDARD DEVIATIONS DUE TO ALL ERROR SOURCES ...

MODE	CIRCUMFERENTIAL MODE NUMBER	RADIAL ORDER	WAVE INDICATOR	NORMALIZED AMPLITUDE DEVIATION	AMPLITUDE DEVIATION (DB)	PHASE DEVIATION (DEGREES)
1	-4	0	1	0.107999E+00	0.890786E+00	0.360487E+01
2	-4	1	1	0.101800E+00	0.842055E+00	0.249215E+01
3	-4	2	1	0.105987E+00	0.874497E+00	0.168806E+01

... NORMALIZED STANDARD DEVIATION COMPONENTS (ERROR SOURCE DEVIATION TIMES RMS SUM OF NORMALIZED INFLUENCE COEFFICIENTS) ...

MODE	AMPLITUDE DUE TO ERROR IN X	R	THETA	B	PSI	PHASE DUE TO ERROR IN X	R	THETA	B	PSI
1	0.7415E-03	0.9[illegible]E-05	0.[illegible]E-03	0.1075E+00	0.1004E-01	0.8676E-01	0.8131E-04	0.4966E-01	0.3481E+01	0.9311E+00
2	0.4612E-03	0.655[illegible]E-05	0.3636E-03	0.1016E+00	0.6818E-02	0.8406E-01	0.1140E-03	0.4898E-01	0.2315E+01	0.9185E+00
3	0.3074E-03	0.[illegible]E-05	0.1573E-03	0.1059E+00	0.2950E-02	0.8200E-01	0.7069E-03	0.4985E-01	0.1402E+01	0.9347E+00

*** MODAL CALCULATION COMPUTEP PROGRAM ***

... SENSITIVITY COEFFICIENT CALCULATION ...

... RMS SUM OF NORMALIZED INFLUENCE COEFFICIENTS ...

MODE	AMPLITUDE DUE TO ERROR IN X	R	THETA	B	PSI	PHASE DUE TO ERROR IN X	R	THETA	B	PSI
1	0.3708E+00	0.4957E-02	0.2678E-01	0.4301E-02	0.6696E-02	0.4338E+02	0.4066E-01	0.2483E+01	0.1392E+00	0.6207E+00
2	0.2306E+00	0.4278E-02	0.1818E-01	0.4063E-02	0.4545E-02	0.4203E+02	0.5702E-01	0.2449E+01	0.9259E-01	0.6123E+00
3	0.1512E+00	0.4098E-02	0.7867E-02	0.4238E-02	0.1967E-02	0.4100E+02	0.1035E+00	0.2492E+01	0.5610E-01	0.6231E+00

... INFLUENCE COEFFICIENTS (PARTIAL DERIVATIVES) ...

INFLUENCE COEFFICIENTS FOR MEASUREMENT LOCATION 1

MODE	AMPLITUDE DUE TO ERROR IN X	R	THETA	B	PSI	PHASE DUE TO ERROR IN X	R	THETA	B	PSI
1	-0.4323E+03	0.2770E+01	0.3207E+02	0.2590E+01	0.8017E+01	-0.2869E+02	0.2251E-01	0.1593E+01	-0.7737E-01	0.3982E+00
2	-0.4557E+03	0.4714E+01	0.3977E+02	0.5591E+01	0.9939E+01	-0.3154E+02	0.4634E-01	0.1823E+01	-0.5167E-01	0.4557E+00

3	0.3793E+02	0.1888E+01	0.4259E+01	0.3065E+01	0.1065E+01	-0.2737E+02	0.6339E-01	0.1659E+01	-0.9041E-02	0.4147E+00

INFLUENCE COEFFICIENTS FOR MEASUREMENT LOCATION 2

MODE	AMPLITUDE DUE TO ERROR IN					PHASE DUE TO ERROR IN				
	X	R	THETA	B	PSI	X	R	THETA	B	PSI
1	0.3253E+03	-0.6132E+01	-0.2075E+02	0.5216E+01	-0.5187E+01	-0.3014E+02	-0.3239E-01	0.1808E+01	0.8881E-01	0.4520E+00
2	0.4445E+03	-0.9217E+01	-0.2852E+02	0.7925E+01	-0.7129E+01	-0.2464E+02	-0.1927E-01	0.1460E+01	0.6576E-01	0.3700E+00
3	0.1588E+03	-0.6414E+01	-0.1076E+02	0.5605E+01	-0.2695E+01	-0.2936E+02	0.1666E-01	0.1771E+01	0.4060E-01	0.4427E+00

INFLUENCE COEFFICIENTS FOR MEASUREMENT LOCATION 3

MODE	AMPLITUDE DUE TO ERROR IN					PHASE DUE TO ERROR IN				
	X	R	THETA	B	PSI	X	R	THETA	B	PSI
1	0.1070E+03	0.3156E+01	-0.1132E+02	0.2649E+01	-0.2630E+01	-0.1226E+02	0.9860E-02	0.5990E+00	0.7426E-01	0.1498E+00
2	0.1116E+02	0.5690E+01	-0.1124E+02	0.5724E+01	-0.2811E+01	-0.1263E+02	-0.2707E-01	0.6974E+00	0.3973E-01	0.1744E+00
3	-0.1967E+03	0.1807E+01	0.6520E+01	0.2867E+01	0.1630E+01	-0.8342E+01	-0.8005E-01	0.5706E+00	-0.3764E-01	0.1427E+00

ORIGINAL PAGE IS
OF POOR QUALITY

*** MODAL CALCULATION COMPUTER PROGRAM ***

... CHARACTERISTIC E-FUNCTION VALUES FOR MODAL AMPLITUDE AND PHASE CALCULATIONS ...

... MEASUREMENT LOCATIONS ...

LOCATION	MODES		
	1	2	3
1	0.404	-0.315	0.233
2	0.404	-0.315	0.233
3	0.404	-0.315	0.233

... PREDICTION LOCATIONS ...

LOCATION	MODES		
	1	2	3
1	0.404	-0.315	0.233
2	0.404	-0.315	0.233
3	0.404	-0.315	0.233

*** MODAL CALCULATION COMPUTER PROGRAM ***

... INPUT VARIABLES ...

NUMBER OF MEASUREMENT LOCATIONS = 3 NUMBER OF PREDICTION LOCATIONS = 3 NUMBER OF (MODE,MU) SETS = 3

... INPUT MODES ...

MODE	CIRCUMFERENTIAL MODE NUMBER	RADIAL ORDER	WAVE INDICATOR
1	-4	0	1
2	-4	1	1
3	-4	2	1

... REFERENCE VALUES ...

REFERENCE PRESSURE = .2000E-03

... TEST GEOMETRY AND CONDITIONS ...

HUB / TIP RATIO = 0.440	OUTER RADIUS OF DUCT = 25.00	AXIAL MACH NUMBER = -0.10
FREQUENCY = 6200.00	SPEED OF SOUND = 34345.00	RADIAN FREQUENCY = 38955.75

*** MODAL CALCULATION COMPUTER PROGRAM ***

... MODAL AMPLITUDE AND PHASE CALCULATION ...

... CALCULATED MODAL AMPLITUDES AND PHASES ...

MODE	CIRCUMFERENTIAL MODE NUMBER	RADIAL ORDER	WAVE INDICATOR	AMPLITUDE (PRESSURE)	AMPLITUDE (DB)	PHASE (DEGREES)	AXIAL WAVE NUMBER REAL	AXIAL WAVE NUMBER IMAGINARY	KMU
1	-4	0	1	0.250000E+03	0.121938E+03	325.0000	71.0845	0.0	5.2510
2	-4	1	1	0.250000E+03	0.121938E+03	250.0001	69.0154	0.0	8.7800
3	-4	2	1	0.250000E+03	0.121938E+03	100.0000	65.0758	0.0	12.9210

... REFERENCE LOCATION ...

X	R	THETA
0.0	0.0	0.0

... PREDICTION LOCATIONS ...

LOCATION NUMBER	X	R	THETA	AMPLITUDE (RESULTANT)	PHASE (RESULTANT)
1	0.0	25.000000	0.0	0.115820E+03	36.437406
2	14.976000	25.000000	0.0	0.120052E+03	345.049316
3	29.960007	25.000000	0.0	0.119999E+03	298.232422

*** MODAL CALCULATION COMPUTER PROGRAM ***

... CHARACTERISTIC E-FUNCTION VALUES FOR MODAL AMPLITUDE AND PHASE CALCULATIONS ...

... MEASUREMENT LOCATIONS ...

LOCATION	MODES 1	2	3
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0

... PREDICTION LOCATIONS ...

LOCATION	MODES 1	2	3
1	0.404	-0.315	0.233
2	0.404	-0.315	0.233
3	0.404	-0.315	0.233