NASA TECHNICAL MEMORANDUM

NASA TM X-3281

NASA TM X-3281

A COMPUTER PROGRAM TO PREDICT ROTOR ROTATIONAL NOISE OF A STATIONARY ROTOR FROM BLADE LOADING COEFFICIENTS

Ramani Ramakrishnan, Donald Randall,
and Robert N. Hosier

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1976

1. Report No.
NASA TM X-3281

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A COMPUTER PROGRAM TO PREDICT ROTOR ROTATIONAL NOISE OF A STATIONARY ROTOR FROM BLADE LOADING COEFFICIENTS

5. Report Date
February 1976

6. Performing Organization Code

7. Author(s)
Ramani Ramakrishnan, Donald Randall, and Robert N. Hosier
(See supplementary notes, block 15)

8. Performing Organization Report No.
L-9796

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23665

10. Work Unit No.
505-10-26-02

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes
Ramani Ramakrishnan: The George Washington University, Joint Institute for Acoustics and Flight Sciences.
Donald Randall: Computer Sciences Corporation.
Robert N. Hosier: Langley Directorate, U.S. Army Air Mobility R&D Laboratory.

16. Abstract

A digital computer program is presented. The program calculates the rotational noise of stationary helicopter rotors based on multiple sets of measured or hypothetical high-frequency blade-loading coefficient data. The programing language used is FORTRAN IV. A description of all main and subprograms is provided so that any user possessing a FORTRAN compiler and random access capability can adapt the program to his facility.

In the proper format, empirical or hypothetical blade surface-pressure spectra are used by the program to calculate: (1) blade station loading spectra, (2) chordwise and/or spanwise integrated blade-loading spectra, and (3) far-field rotational noise spectra. Any of five standard inline functions describing the chordwise distribution of the blade loading can be chosen in order to study parametrically the acoustic predictions.

The program output consists of both printed and graphic descriptions of the blade-loading coefficient spectra and far-field acoustic spectrum. The results may also be written on binary file for future processing.

Examples of the application of the program along with a description of the rotational noise prediction theory on which the program is based are also provided.

17. Key Words (Suggested by Author(s))
Rotor noise prediction
Blade surface pressures
Fluctuating blade loads

18. Distribution Statement
Unclassified – Unlimited

Subject Category 71

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	164	$6.25

*For sale by the National Technical Information Service, Springfield, Virginia 22161

“PAGE MISSING FROM AVAILABLE VERSION”

CONTENTS

Page

A COMPUTER PROGRAM TO PREDICT ROTOR ROTATIONAL NOISE OF A STATIONARY ROTOR FROM BLADE LOADING COEFFICIENTS

Ramani Ramakrishnan
The George Washington University
Joint Institute for Acoustics and Flight Sciences

Donald Randall*

and Robert N. Hosier
Langley Directorate, U.S. Army Air Mobility R&D Laboratory

SUMMARY

A digital computer program is presented. The program calculates the rotational noise of stationary helicopter rotors based on multiple sets of measured or hypothetical high-frequency blade-loading coefficient data. The programing language used is FORTRAN IV. A description of all main and subprograms is provided so that any user possessing a FORTRAN compiler and random access capability can adapt the program to his facility.

In the proper format, empirical or hypothetical blade surface-pressure spectra are used by the program to calculate: (1) blade station loading spectra, (2) chordwise and/or spanwise integrated blade-loading spectra, and (3) far-field rotational noise spectra. Any of five standard inline functions describing the chordwise distribution of the blade loading can be chosen in order to study parametrically the acoustic predictions.

The program output consists of both printed and graphic descriptions of the blade-loading coefficient spectra and far-field acoustic spectrum. The results may also be written on binary file for future processing.

Examples of the application of the program along with a description of the rotational noise prediction theory on which the program is based are also provided.

INTRODUCTION

The studies reported in references 1 and 2 demonstrated that the far-field rotational noise generated by a stationary (no forward speed) rotor can be accurately predicted from

*Computer Sciences Corporation.

measurements of the fluctuating surface pressures on the rotor blade surface. In these references, it was also shown that accurate predictions were achieved only when fluctuating surface pressures up to frequencies of 500 Hz to 1 kHz were considered. This report documents the computer program which was developed as a part of those studies. The program uses the measured spectra of the fluctuating surface pressures in the theory of references 3 and 4 in order to predict the far-field rotational noise spectra. The program is called a Stationary Rotor Rotational Noise Prediction Program, hereafter abbreviated as SRRNPP.

SRRNPP is written in a version of FORTRAN IV currently in use on Control Data Corporation (CDC) 6000 series machines at the Langley Research Center. It is written for the user with a basic familiarity with references 1 and 2. In addition to the rotor geometry and performance parameters, SRRNPP requires as input either measured or hypothetical blade surface-pressure spectra (amplitude and phase). These inputs are used to calculate: (1) blade station loading spectra, (2) chordwise and/or spanwise-integrated blade-loading spectra, and (3) far-field rotational noise spectrum. The station-loading spectra describe the frequency content of the fluctuating blade surface pressures at the measurement points on the rotor blade. The integrated blade-loading spectra describe the summed frequency content of a number of chordwise and/or spanwise measurement stations or points. The far-field rotational spectrum describes the predicted frequency content of the rotational noise at any far-field point. Any of the five inline functions shown in table I may be chosen to study parametrically the acoustic predictions.

The SRRNPP output consists of both printed and Calcomp Graphic descriptions of the station and integrated loading spectra and of the rotational noise spectrum. At the user's option, these results may be written on binary file for future processing. The SRRNPP is designed to run economically and efficiently through job-stepping and rigid formatting.

The format for this paper is to present an overview of the main and subprograms in the main body. The overview includes program application, subroutine usage, input and output guides, program test cases, and concluding remarks. Appendices A to G present details of the program. Such details include the theory and equations used, transducer spectrum-data management, random access file usage, plotting guides, numerical subroutine usage, and program usage and listing.

SYMBOLS

The values are given both in the U.S. Customary Units and in the International System of Units (SI). The measurements and calculations were made in U.S. Customary Units.

a_0	speed of sound, m/sec (ft/sec)
B	number of blades
c	blade chord, m (ft)
D	observer distance from source, m (ft)
D_F	total rotor drag force, $L_T \sin \beta$, N (lb)
$\bar{F}$	force vector, N (lb)
$J_q(Z_0)$	Bessel function of the first kind of order q and argument Z_0
j	complex operator, $\sqrt{-1}$
K_D	drag operating constant, $\frac{ND_F}{Ra_0 M_e}$, N/m^2 (lb/ft^2)
K_L	lift operating constant, $\frac{NL_F}{Ra_0} \sin \sigma'$, N/m^2 (lb/ft^2)
L	load per blade per unit span, N/m (lb/ft)
L_F	total rotor lift force, $L_T \cos \beta$, N (lb)
L_T	total rotor aerodynamic load, N (lb)
M_T	tip Mach number, $\Omega r_t / a_0$
M_e	effective Mach number, $0.8M_T$
mB	acoustic harmonic number
N	rotor shaft frequency, Hz
P	general blade pressure, N/m^2 (lb/ft^2)
P_q, P_s	amplitude of the qth or sth fluctuating blade pressure harmonic, N/m^2 (lb/ft^2)

P_0	amplitude of the zeroth (uniform) blade pressure harmonic, N/m^2 (lb/ft^2)
p	acoustic pressure, N/m^2 (lb/ft^2)
Q	limit of blade-loading coefficient summation (defined in eq. (A13))
q	mode number $(q_{\pm} = mB \pm s)$
R	observer distance from rotor center, m (ft)
r	source point distance from rotor center, m (ft)
r_e	effective rotor radius, $0.8r_t$, m (ft)
r_t	blade tip radius, m (ft)
SP	sound pressure, N/m^2 (lb/ft^2)
SP_{mB}	amplitude of mBth sound pressure harmonic, N/m^2 (lb/ft^2)
S_q	jump function, $S = 0$ for $q < mB$, $S = 1$ for $mB \geqq q$
s	blade-pressure harmonic number
t	time variable, sec
V_T	blade tip speed
$\bar{x}$	observer coordinate vector, m (ft)
$\bar{y}$	source coordinate vector, m (ft)
Z_0	Bessel function argument, $mBM_e \cos \sigma'$
α_q	qth blade-loading coefficient, $\frac{P_q}{P_0}$
α_s	sth blade-loading coefficient, $\frac{P_s}{P_0}$
β	effective blade force angle, radians
γ_q	limit and directivity function, $mBJ_q(Z_0)$

η,ξ	arbitrary phase constants, radians
θ	observer azimuthal angle, radians
σ'	observer elevation angle, radians
τ	time period, sec
ϕ_q,ϕ_s	phase angles of the blade pressure or blade loading harmonics, radians
χ_{mB}	chordwise loading distribution spectrum function
ψ	rotor azimuthal angle, radians
Ω	angular frequency, $2\pi N$

Subscripts:

m,mB	sound pressure harmonic number
q,s	blade-loading or blade-pressure harmonic number

PROGRAM OVERVIEW

A concise explanation and derivation of the theory used in this study are detailed in appendix A. The computational procedure of the SRRNPP uses the final equation (eq. (A12)) of appendix A. It is clear from equation (A12) that in addition to general rotor parameters, the prediction process requires a set of blade-loading coefficients called α's. These α's are the Fourier coefficients of the rotor blade surface pressures normalized by the uniform or static surface pressures. SRRNPP manipulates the given or computed α's to predict the rotational noise efficiently.

A general review of the contents of the program is undertaken here. The function of its main and subprograms becomes apparent in the program application section. The SRRNPP package consists of five major parts; two are computational, the rest are efficient data (input/output) manipulators. Table II shows the program interrelation. A detailed description of the program usage and the various efficient techniques is undertaken separately in appendix F for the benefit of the user. A periodic reference is made to this appendix at various stages of the text and hence the user is advised to familiarize himself with appendix F.

Program RAMANI

This program combines the upper and lower blade surface-pressure coefficients at each location along the chord (or span). RAMANI converts the gage-referenced surface pressures into differential pressures. The conversion takes one of the following three forms: pressure data from upper and lower surfaces can be added; the data can be weighted if surface pressures from only one surface are available; and the datum from a particular location can be skipped if it is in error.

Program SPECPLT

The program plots the differential pressure spectra (both amplitude and phase) against frequency.

Program TRANS

TRANS is a preprocessing program. The program preprocesses the differential spectra output from program RAMANI by reorganizing and storing the output on random access file for efficient use in RNPPE4.

Program RNPPE4

This program calculates the rotor rotational noise using equation (A12). It predicts the sound pressure levels at sound harmonic numbers mB. The sound pressure levels are then modified by one of five chord spectrum functions chosen by the user. (See appendix A.)

Program SPLPLT

This program is similar to program SPECPLT. SPLPLT plots the output from program RNPPE4.

PROGRAM APPLICATIONS

There are three different ways in which the blade-loading data can be provided to the SRRNPP. The program operation for each method is illustrated by the following three examples.

Example 1

In this example, the blade-loading data are obtained from an experimental setup where it is possible to obtain loading information at a representative span location on a blade of a hovering helicopter.

Manipulation of Loading Spectra

Figure 1(a) shows the location of surface-pressure transducers mounted at a representative span location on the blade used in references 1 and 2. These transducers measure the surface pressures for various rotor operating conditions. After digitization, these data are then Fourier analyzed using a program such as the one in reference 5. After the Fourier analysis, each transducer location has an amplitude and phase spectrum associated with it; the associated amplitude and phase spectrum has been stored on output tape, in the format required by SRRNPP. (See section on "SRRNPP Input Guide.")

Thus, after Fourier analysis the user has a set of gage-referenced (amplitude and phase) pressure spectra at each upper and lower surface location along the chord of the blade. These spectra are stored on magnetic tapes in blocks of 1500 points for efficient operation. (See appendix B.) In order to convert the gage pressures into differential pressures, a call to program RAMANI is made.

The conversion is done through complex addition. Let r_f be the amplitude and θ_f be the phase of the blade pressure at a given frequency f. If the data from location n (upper) is to be added to the one at location k (lower), the sum is formed by a point-by-point (i.e., at each frequency) complex addition

$$r_{nkf} \exp^{i\theta_{nkf}} = r_{nf} \exp^{i\theta_{nf}} + r_{kf} \exp^{i\theta_{kf}} \tag{1}$$

where

$$r_{nkf}^2 = r_{nf}^2 + r_{kf}^2 + 2r_n r_k \cos\left(\theta_{nf} - \theta_{kf}\right)$$

$$\theta_{nkf} = \text{arc tan} \left| \frac{r_{nf} \sin \theta_{nf} + r_{kj} \sin \theta_{kf}}{r_{nf} \cos \theta_{nf} + r_{kf} \cos \theta_{kf}} \right|$$

If only one set of the spectrum data is to be used at a location, then the data are scaled. For example, in figure 1(a), the user desires to use the upper surface data only at location 3, after finding the data at location 4 is unusable. The scaling is then

$$\left.\begin{aligned} r_{34f} &= Wr_{3f} \\ \theta_{34f} &= \theta_{3f} \end{aligned}\right\} \tag{2}$$

where W is the amplitude weighting factor provided by the user. (Note that the phase is left unaltered.)

The user may choose to skip a particular chord location entirely. For example, if locations 3 and 4 (fig. 1(a)) contain usable data, the user can omit the addition at locations 3 and 4. The user then would have differential pressure data at 3 chord stations instead of at 4 chord stations.

The user may input surface-pressure spectra in any sequence, keeping in mind, however, that the spectra have both amplitude and phase. He must specify the combining order using the two arrays ISTN and JSTN. (See section on "SRRNPP Input Guide.")

For the present example problem, the user leaves program RAMANI with 4 differential spectra. The spectra consist of both amplitude and phase.

The user then has the option to plot these combined data or to skip to the next subprogram. If the user decides to plot the data, a call to program SPECPLT is made. Program SPECPLT is usually job-stepped (see appendix F) with program RAMANI, although it can be run separately if the data from program RAMANI are stored on magnetic tape. In these plots, the pressure data are normalized by P_0 to provide loading coefficients.

The user controls the plotting of the differential spectra through the array NNPLOT. NNPLOT(I) = 0, 1, 2 for I = 1, 2, . . ., MTRACKS. For NNPLOT(I) = 0, no combined spectrum plots are produced for combined position I. For NNPLOT(I) = 1, only a blade-loading coefficient spectrum plot is produced for combined position I. For NNPLOT(I) = 2, both a blade-loading coefficient spectrum plot and a phase spectrum plot are produced for combined position I. The user can partially control the axis length and can do some of his own scaling; however, he must conform to the restrictions of the Langley Research Center Graphic Output System. (See ref. 6, and appendix D.)

At this stage, the user can immediately proceed to the next part of SRRNPP or can store the data from program RAMANI on a magnetic tape. These choices enable the user to check the output from program RAMANI before proceeding.

Data Preprocessing

The user enters the program TRANS either immediately following program SPECPLT or with the information stored on magnetic tape. This program is an intermediate step. TRANS reorganizes the combined spectrum for efficient use (see appendixes B and F) in program RNPPE4 which immediately follows.

Calculation of Rotational Noise

Program RNPPE4 computes the sound pressure level at various mB harmonics using equation (A12). RNPPE4 is run with program TRANS unless the loading spectra are to be computed internally (as is shown in example 3). Equation (A12) only requires one set of amplitude and phase spectra, which are assumed representative of the entire blade, to predict the rotational noise. Hence, the user may choose one of three options in RNPPE4 to manipulate the spectral data.

These options are best illustrated by the example that has been presented so far. Using the eight sets of spectra (fig. 1(a)), the user left program RAMANI with differential spectra at 4 locations along the chord. He could then use each differential spectrum separately (that is, the amplitudes and phases from each station) in equation (A12), and come out with 4 predicted sound-pressure-level spectra; or, he could use all the 4 sets of differential spectrum data and come out with amplitude and phase spectra representing all the blade loads. This representative spectrum is obtained by integrating the differential spectra along all or part of the chord.

The procedure for integration follows. (See section on "SRRNPP Input Guide.") A loading distribution at each loading frequency is defined by six points; four points are along the chord and two zeros are at the leading and trailing edges. (See fig. 1(b).) If the user wishes to use the entire distribution, a call to subroutine SPLS is made for each loading frequency. The 6 points defining the loading distribution at each frequency are fitted with a cubic spline fit by SPLS. SPLS then evaluates the integral of this curve by using a modified Simpson's rule. This process continues until the amplitude arrays are exhausted for all loading harmonics.

However, if for some reason the user wishes to use only a part of the loading distribution (e.g., the front half in fig. 1(b)), subroutine CSIUNI is called to determine a partial loading distribution. After fitting the points with a cubic spline curve, subroutine CSIUNI interpolates to the point (if the point is not at one of the measurement locations) which defines the trailing limit of the partial distribution. SPLS is called to integrate this curve at each loading harmonic.

Thus, the user obtains representative spectra of loading coefficients α's and phases. If the Fourier analyzed spectra (ref. 5) have different bandwidths, program RNPPE4 automatically chooses the loading spectral points such that the spectra have a bandwidth equal to the blade passage frequency.

Program RNPPE4 then uses this representative loading information in equation (A12) to calculate sound pressure levels at frequencies mB. The user can modify these sound pressures by any one of five chord spectrum functions (see section on "SRRNPP Input Guide") with the input parameter ICHORD. The functions include: point loading, rectangular, half-cosine, triangular, and saw tooth (refs. 1 and 2; table I). These functions affect the sound pressures at each sound harmonic.

The integration procedure is used only for the amplitudes of the blade loads. The phase arrays are averaged to produce a representative phase spectrum. The amplitudes are normalized by the P_0 value to produce blade-loading coefficients (α's).

RNPPE4 is designed to compute multiple sets of sound pressure level (SPL) values for the same general rotor parameters; RNPPE4 can also accept more than one set of rotor data. The general flow of this program is described in the flow chart in table III. If the user calls program SPLPLT for a graphic presentation of his results, SPLPLT plots the SPL values for each set of loading data. SPLPLT also plots the loading data (blade-loading coefficients and, optionally, phases) if computed through integration. Thus, in the example if the SPL's are computed by using the 4 sets of data separately and once by integrating these 4 sets, the user obtains 5 SPL plots, an integrated loading amplitude plot, and, optionally, the corresponding phase plots from RNPPE4.

Program RNPPE4 stores the SPL and integrated blade-loading harmonic (BLH) data temporarily on a disk file. Therefore, it is necessary that SPLPLT be job-stepped with program RNPPE4. The plotting data are written on the disk in blocks as they are computed in RNPPE4. This sequential storing of the SPL and integrated BLH data requires that the order of the plots generated in SPLPLT corresponds to the order of computation in RNPPE4. To do this, successive values of the input parameter NNPLOT in program SPLPLT must correspond to the input parameter NTEGRAT in program RNPPE4.

As in the combined spectrum-plotting program SPECPLT, the scaling and range of the points to be plotted can be partially controlled through user input. The SPL and integrated BLH plotting must conform with the restrictions of the LRC Graphic Output System. (See appendix D.)

The present example has exercised the entire SRRNPP by manipulating detailed experimental data to predict the far-field rotational noise. The sequencing of the operations can be carried out by the user as follows:

1. RAMANI – SPECPLT – TRANS – RNPPE4 – SPLPLT, or
   RAMANI – SPECPLT – tape – TRANS – RNPPE4 – SPLPLT

The efficient use of this sequencing is explained in more detail in appendix F.

Example 2

This example is typical of most experiments with limited facilities. Instead of using an array of spanwise and chordwise blade surface-pressure measurements, the user may only have surface-pressure data from one measurement location available to him. In such a case, the user skips the programs RAMANI and SPECPLT. However, it should be remembered here that, depending on the locations of the measurement stations, if the user has more than one set of loading data, he cannot skip these two programs. After arranging the spectrum data in the required manner (see section on "SRRNPP Input Guide") on a magnetic tape (or cards), the user enters program TRANS directly. Here the data is reorganized and supplied as input to program RNPPE4 to calculate the SPL's.

In this example, there is no need for integration. Thus, the sequence of operations (see appendix F) has been TRANS – RNPPE4 – SPLPLT. Only an acoustic spectrum plot is provided.

Example 3

Hypothetical blade-loading coefficient spectra are computed theoretically by the user who has an a priori knowledge of the fall-off rate of the amplitudes of blade-loading coefficient spectra. The user skips programs RAMANI, SPECPLT, and TRANS, and supplies the loading data directly to program RNPPE4. In addition to the general rotor operating parameters, the user assigns values to the three variables C, X, and NBLHPT. (See section on "SRRNPP Input Guide.") The theoretical set of amplitude coefficients are then:

$$\mathrm{BLH}(i) = \frac{c}{(i)^{x}} \qquad (i = 1, 2, 3, \ldots)$$

where BLH(i) are the amplitude coefficients α's. The corresponding phases are all set to zero. For more details on the negligible effects of phases, see references 1 and 2. These α's are used directly in equation (A12) with a chord spectrum function chosen by the user to predict the SPL's. As in the previous example, the user obtains only one graphic output if a call to program SPLPLT is made. For this example, the sequence of operations has been RNPPE4 – SPLPLT.

A sample test run for example 1 was made using the entire SRRNPP package. This test case is described in the section on "Program Applications."

SUBROUTINE USAGE AND DESCRIPTION

This section provides a general description of the usage of all subroutines used in the five subprograms of the SRRNPP. A detailed documentation of each routine is contained in appendixes C, D, and E.

Random Access Subroutines

Programs RAMANI, SPECPLT, TRANS, and RNPPE4 use random access files to improve program efficiency and to reduce storage requirements. Manipulation of the random access file in each program is accomplished through the three subroutines OPENMS, READMS, and WRITMS. The three FORTRAN callable routines are system resident in the LRC computer complex.

A detailed description of each of these routines is contained in appendix C.

OPENMS is the first routine called and is used initially to open the random access file. The parameters of OPENMS define the type of record indexing and the size of the index table. Numbered indexing is used in all programs and the size of the index table is set to 987 to accommodate 30,000 spectrum data points from 24 separate transducers. OPENMS is called once at the beginning of each program. OPENMS also designates the random access file name.

WRITMS is used to transfer a record from central memory to the random access file. READMS is used to transfer a logical record from the random access file to central memory. The parameters of both of these routines define the record number and the length of the record.

Plotting Subroutines

Programs SPECPLT and SPLPLT use many plotting subroutines of the LRC Graphic Output System. The subroutines are PSEUDO, CALPLT, AXES, NUMBER, NOTATE, and NFRAME. These FORTRAN callable subroutines are system resident.

PSEUDO is called once at the beginning of programs SPECPLT and SPLPLT. PSEUDO initializes the LRC Graphic Output System and indicates the name of the plot vector file.

AXES is called to draw and label the plot axes. One call is required for each axis to be drawn.

NOTATE and NUMBER are used for annotation of the individual plots. NOTATE is used for drawing alphanumeric information. NUMBER converts floating-point numbers to binary coded decimal (BCD) and draws the resulting alphanumeric information.

CALPLT is used to do the actual plotting of the data points. It also terminates the plotting for the plot vector file.

NFRAME is used to indicate the completion of the present plot (frame). It also sets up for the next plot.

Although these subroutines are the actual routines called, many of the individual routines are themselves routine dependent. Appendix D describes in detail each of the required plotting routines.

Numerical Subroutines

Program RNPPE4 requires additional subroutines to compute the integrated BLH data and to evaluate the SPL values using equation (A12). The subroutines SPLS, CSIUNI, and BSSLS are required. SPLS and CSIUNI are two routines presently in the LRC Math Library. BSSLS is a modified form of the math library subroutine BJIR. The three subroutines are documented in appendix D.

If the blade-loading coefficients and phases are to be computed by integration of the loading distribution over a fraction of the chord, the cubic spline interpolater subroutine CSIUNI determines a modified partial loading distribution. The loading distribution (or the interpolated partial steady loading distribution) is integrated by applying the cubic spline integration subroutine SPLS.

CSIUNI and SPLS are used in a similar manner with like parameters. A cubic spline is fit to a supplied set of data points; the spline is then interpolated at a specified point (CSIUNI) or integrated over a specified range (SPLS).

The solution in equation (A12) calls for evaluating Bessel functions of the first kind at various orders for numerous arguments. Subroutine BSSLS evaluates the Bessel function using a backward recursion technique.

SRRNPP INPUT GUIDE

The input parameters necessary to operate the SRRNPP are described in this section. Included is a description of the restrictions, mode, units, and default values for the input parameter to each of the five subprograms in the SRRNPP.

Data may be input to the five subprograms by punched cards, magnetic tape, random access file, and disk file. The input method chosen depends on the subprogram(s) being considered, on the extent to which the program job-stepping is used, and on the values of input control parameters supplied to each subprogram through NAMELIST. Sample inputs for each program for an actual test case are given in the "Program Test Cases" section.

Program RAMANI Input

The input for program RAMANI consists of the transducer spectrum data, the spectrum data combining sequence, and the program control parameters. Magnetic tape is used as the means of input for the transducer spectrum data. One or two tapes can be used depending on the number of transducers. The transducer data input tape is generated by a FAST FOURIER TRANSFORM program (ref. 5). The spectrum data consist of an amplitude and a phase at each specified frequency. The specific spectrum data input parameters are described in table IV, and the tape format is described in appendix B.

The spectrum-combining sequence and program control parameters are input by NAMELIST. The NAMELIST is named INPUT, and its parameters with default conditions and restrictions are contained in table V.

There are two techniques used to skip unneeded transducer data. NSKIP is used to skip spectrum data records at the beginning of the input tape, or the combining sequence arrays ISTN and JSTN are used to select specific transducers for skipping by setting

JSTN = 0. MTRACKS must be increased by one for each transducer skipped through the use of JSTN = 0.

An example of the usage of the spectrum data combining sequence follows. Let ISTN(I) = M and JSTN(I) = N, where I = 1, 2, 3, . . ., 24, 0 ≦ M ≦ 24, 0 ≦ N ≦ 24, (24 being the maximum number of surface locations allowed). Then if:

1. M ≠ N, and N ≠ 0, data from locations M and N are added.

2. M = N, and N, M ≠ 0, the data from location M will be scaled by an amplitude weighting factor W.

3. M ≠ N, and N = 0, data from location M will be skipped.

To conserve storage requirements two restrictions have been placed on the arrays ISTN and JSTN: (1) the ISTN array must be strictly increasing; and (2) ISTN(I) ≧ JSTN(J) for all I ≧ J. The second restriction amounts to choosing the smallest remaining transducer number as the next element in ISTN array.

Program SPECPLT Input

The input for program SPECPLT consists of the combined transducer spectrum data generated by program RAMANI and various plot control parameters.

The combined spectrum data are input to SPECPLT by random access file or by magnetic tape. The temporary storage of the random access is utilized when the combined spectrum plots accompany the combined spectra data. In this case, program SPECPLT should be job-stepped with program RAMANI and the default IOPTN (default is 1) should be used. If the combined spectrum data plots are requested separately, the RAMANI output tape can be used as input and the parameter IOPTN should be set to zero. The program deck setups for the different combinations are illustrated in appendix F. The format of the combined spectrum data is detailed in appendix B, and the combined spectrum data parameters are described in table IV.

The program control parameters are input by a NAMELIST named INPUT. The NAMELIST parameters with specified default conditions and restrictions are described in table VI. The array NNPLOT determines which, if any, of the combined spectrum data plots are to be generated for each position. The parameters XMIN, XMAX, YMIN, YMAX, and YSCALE give the user flexibility in determining variable axis lengths and scale factors. These plot control parameters must correspond to the frame size specified on the plot control card. (See appendix D.)

Program TRANS Input

The input for program TRANS consists of two input control parameters and of the combined transducer data generated by program RAMANI. As in program SPECPLT, the combined transducer data can be input by random access file, by magnetic tape, or with an additional option for card input. The means of input are dependent upon the job-stepping of programs RAMANI and TRANS; the input is again controlled by the parameter IOPTN. The spectrum-data input parameters are listed in table IV. The data format for tape and random access is listed in appendix B. The program TRANS input control parameters are entered through the NAMELIST named INPUT. The parameters with specified default conditions are described in table VII. The optional card format is described in table VIII.

Program RNPPE4 Input

The input for program RNPPE4 consists of the reorganized combined spectrum data which are generated and passed by program TRANS. The input also contains certain job control and rotor parameters controlling the computation of the BLH and SPL values.

Since the program TRANS is job-stepped with program RNPPE4, the combined spectrum data are passed through the random access file. Random access record pointers and counters are passed along with the spectrum data. The combined spectrum-data random access parameters are described in table IX, and the combined spectrum data format is listed in appendix B. If only experimental data generated by the inline BLH function are used, program TRANS can be bypassed along with the reorganized spectrum data.

The job-control parameters and necessary rotor parameters are passed through the NAMELISTS named ROTOR, INPUT, and INBLH. The ROTOR NAMELIST contains numerous helicopter rotor parameters and the number of sets of BLH and SPL data to be determined for the specified rotor. The ROTOR NAMELIST is reinput if another rotor is to be used with the combined spectrum data. For each set of BLH and SPL values, the NAMELIST INPUT or INBLH must be supplied. INPUT is used when the BLH coefficients and phases are to be computed using the combined spectrum data; INBLH is used when the BLH coefficients and phases are to be computed internally. NAMELIST INPUT contains parameters which determine whether or not integration is to be used to compute the BLH data; these parameters also determine which spectrum chord function is to be used to adjust SPL values. NAMELIST INBLH is used primarily to define the inline BLH function. The parameters of NAMELISTS ROTOR, INPUT, and INBLH, together with specified default conditions and restrictions, are contained in table X.

The parameter NTEGRAT of NAMELIST INPUT determines whether the BLH data are computed through the integration of the surface-pressure data and whether this integration is over the entire chord or over a fraction of the chord. NTEGRAT set to zero implies no integration. Full-chord integration is accomplished by setting NTEGRAT to K where K is the number of combined transducers stations with the addition of two stations for the blade edges. Partial integration is accomplished by one or two means. If the upper limit of integration coincides with the ith combined transducer position, it is more efficient to use NTEGRAT = K. If the upper integration limit does not coincide with a combined position, set NTEGRAT to -1 and input the desired value of PRTLINT.

Program SPLPLT Input

The input for program SPLPLT consists of the sets of computed sound pressure levels, the sets of integrated blade-loading coefficients and phases, and various plot-control parameters. If SPL and integrated BLH plots are desired, programs RNPPE4 and SPLPLT are job-stepped. The temporary storage of the disk is used to transfer the sets of integrated BLH data and SPL values to program SPLPLT. The BLH and SPL parameters passed by disk to SPLPLT are described in table XI; the format of the data is described in appendix B.

The SPL and BLH plot-control parameters are input through the NAMELISTS FIXED and INPUT. NAMELIST FIXED inputs the number of SPL plots desired as well as various plotting range and scaling parameters. As in program SPECPLT, the user has specific controls over his plotting. NAMELIST INPUT is input before each SPL plot is generated and dictates whether integrated BLH plots are desired. All parameters of NAMELISTS FIXED and INPUT, with restrictions and default conditions, are described in table XII.

The integrated BLH and SPL data are written to the disk as it is computed in program RNPPE4. This sequential storage of the plotting data necessitates the generation of the plots in the same order as RNPPE4 generates the data. This plotting is done by making values of NTEGRAT in program RNPPE4 equivalent to corresponding values of NNPLOT in program SPLPLT. If NTEGRAT is zero, the corresponding value of NNPLOT must be zero. If NTEGRAT is nonzero and if integrated BLH plots are desired, NNPLOT must be set to 1 or 2 depending on whether an integrated phase plot is desired. If NTEGRAT is nonzero and if integrated BLH plots are not desired, NNPLOT must be set to -1.

SRRNPP OUTPUT GUIDE

The output from the Stationary Rotor Rotational Noise Prediction Program is described in this section. The output from each of the five programs making up the SRRNPP is considered separately.

The programs generate printed, tape, and plotted output as well as temporary random access and disk output. The type of program output depends on the individual program, on the extent to which program job-stepping is being utilized, and the values of various output control parameters supplied to each program through NAMELIST. The section contains sample output for each program including plots for an actual case run.

Program RAMANI Output

Program RAMANI generates printed output, a magnetic tape, and optional random-access output.

The printed output includes: (1) a list of all NAMELIST input parameters including default conditions, and (2) a table of combined amplitudes and phases for user specified NPRINT frequencies for each of the combined transducer spectrum data positions.

The magnetic tape generated consists of the complete combined transducer spectrum data (amplitudes and phases at the various frequencies) tables for each of the combined positions. The format of the combined spectrum data tape is discussed in appendix B.

If program SPECPLT or program TRANS (or both) are to be job-stepped with RAMANI, it is advantageous to generate and to use the optional random access output of the combined spectrum data. The format of the combined spectrum data stored on the random access is discussed in appendix B.

Program SPECPLT Output

The program SPECPLT generates both printed and plotted output. The printed output consists solely of the NAMELIST input control parameters. The random access output from the preprocessor program TRANS consists of the reorganized combined transducer spectrum data. The random access storage format for the reorganized spectrum data is discussed in appendix B. It is reemphasized that the temporary random access storage is utilized only when programs TRANS and RNPPE4 are job-stepped.

Program RNPPE4 Output

Program RNPPE4 generates printed ouput and temporary disk output. The printed output consists of a listing of all NAMELIST input with default conditions, a table of the BLH data for each case, and a table of sound pressure levels for each set of BLH data computed. Initially, a list of all the ROTOR NAMELIST parameters is output. At this point, the following three-step output sequence is repeated until all cases have been run with the given set of ROTOR data. First, a list of the INPUT or INBLH NAMELIST parameters is output. Second, a BLH table is generated consisting of a BLH coefficient, a BLH coefficient divided by the uniform loading coefficient, and the phases (radians) computed at the various harmonics. The size of the BLH table is dependent on the value of the input parameter INCOF. Third, a SPL table is generated. This table consists of the sound pressure levels (decibels) computed at the harmonic frequencies mBN. After all sets of BLH and SPL data have been output for the given set of ROTOR data, the output process terminates if another set of ROTOR data is not supplied. The process will repeat if another set of ROTOR data is supplied.

The temporary disk output consists of the SPL data, the integrated blade-loading coefficients, and integrated phases along with associated plot-control and plot-heading parameters. The SPL and integrated BLH data are output to the disk as they are computed.

Program SPLPLT Output

Program SPLPLT generates printed and plotted output. The printed output consists solely of a listing of all NAMELIST input parameters including specified default conditions.

The plotted output consists of the sound pressure level plots and the optional integrated blade-loading coefficient and phase plots. One plot of SPL (decibels) plotted against frequency (hertz) is generated for each set of sound pressure levels computed in program RNPPE4. The optional integrated BLH plotting is controlled by the input parameter NNPLOT. One plot of integrated blade-loading coefficients (or averaged phases) plotted against frequency is optionally generated if the BLH data are computed through integration in program RNPPE4. The actual plots are obtained through the use of job control cards activating the LRC Graphics Output System. (See appendix D.)

PROGRAM TEST CASES

Specific case by case execution of the entire SRRNPP is illustrated in this section. Appendix G contains a listing of the programs. The test run used for the illustrations consisted of surface-pressure spectral data obtained from the time series analysis program (ref. 5). Seven sets of amplitude and phase data (2 at 15-percent, 1 at 30-percent, 2 at 50-percent, and 2 at 75-percent chord stations (see fig. 3)) were obtained from 7 transducers placed at an 80-percent span on a stationary rotating rotor blade. These pressure data were then used to obtain far-field rotational noise by exercising the SRRNPP.

Program RAMANI

RAMANI adds the upper and lower surface data. Since there was only one set of data available at the 30-percent station, the data were carried over with no weighting. Figures 2(a) and (b) contain a sample input and a printed output, respectively, for program RAMANI.

Program SPECPLT

SPECPLT plots the information from program RAMANI. Since NNPLOT is 2, the user obtains both amplitude and phase plots. There are 4 sets of plots after addition. A sample input and plotted output are given in figures 3(a) and (b), respectively, for program SPECPLT.

Program TRANS

RAMANI provides 4 sets of data after addition; each set represents the 4 positions along the chord. TRANS reorganizes the data to be used by RNPPE4.

Program RNPPE4

The data from TRANS can be used in different ways to produce SPL's at various harmonic numbers (frequencies). A sample input consisting of the general rotor parameters is presented in figure 4. The flexibility of this program is evident from the following options.

Case A.- A single set of data from the 15-percent chord station (ITRACK = 1) is used in equation (A12) with a rectangular chord spectrum function (ICHORD = 1, also see table I). Figure 5 contains a sample input and the printed output.

Case B.- A sample input and printed output are given in figure 6. In this case four (in addition to one leading and one trailing edge) sets of data are integrated (NTEGRAT = 6) and used with saw-tooth spectrum function (ICHORD = 4, also see table I).

Case C.- This example uses integration up to 40 percent of the chord from the leading edge (NTEGRAT = 1). The integrated data are then input to the equations that are coupled with triangular chord spectrum function (ICHORD = 3, also see table I). A sample input and printed output are presented in figure 7.

Program SPLPLT

SPLPLT plots the results obtained from RNPPE4. Figure 8 shows a general parametric input sample. Figure 9 contains sample input and output plots corresponding to Case A, alone. Figure 10 and figure 11 are similar to figure 9, but correspond to Cases B and C, respectively. Since NNPLOT = 2 for Cases B and C, figures 10 and 11 contain integrated amplitude and phase spectra plots in addition to the SPL plots.

CONCLUDING REMARKS

A digital computer program which predicts stationary rotor rotational noise from measured or hypothetical fluctuating blade surface-pressure data has been developed. The program uses the theory which was initially developed by Wright and was later refined by Ramakrishnan and by Hosier and Ramakrishnan.

A complete program documentation including program listings, examples, and test cases has been presented so that the user can exercise the many program options judiciously. This documentation should allow the program to be adapted to any system with FORTRAN IV compiler and random access file capability.

Langley Research Center
National Aeronautics and Space Administration
Hampton, Va. 23665
September 26, 1975

APPENDIX A

THEORY AND EQUATIONS

Rotational noise is defined as the noise a rotor would generate in an inviscid fluid. The origin of this noise is in the periodic forces experienced by the blade as it rotates. Rotational noise has a characteristic frequency spectrum consisting of discrete components which are multiples of a fundamental frequency BN.

The governing unconvected wave equation for a nondissipating medium is given by

$$\frac{1}{a_0^2}\frac{\partial^2 p}{\partial t^2} - \nabla^2 p = -\nabla \cdot \bar{F}(\bar{x},t) \qquad \text{(A1)}$$

where p is the sound pressure, $\bar{F}$ is the fluctuating force, a_0 is the sound speed, and Einstein's tensor convention is used. In solving this wave equation, the following assumptions are made regarding the rotor model: (1) the rotor system as a whole is assumed to be stationary, and (2) nonuniform inflow conditions are assumed. This inflow gives rise to periodic loading on the rotor disk.

The force and rotor coordinate systems are shown in figure 12. The solution for a point force $\bar{f}$ acting at $\bar{y}$ is, then (see also refs. 1 and 2)

$$\therefore p(\bar{x},t) = -\frac{1}{4\pi D^2}(\bar{x} - \bar{y}) \cdot \left[\frac{1}{a_0}\frac{\partial[\bar{f}]}{\partial t} + \frac{[\bar{f}]}{D}\right] \qquad \text{(A2)}$$

where $[\ ]$ implies retarded time $t - D/a_0$ and $D = |\bar{x} - \bar{y}|$. Then

$$\bar{f}(\bar{y}(t),t) = \bar{f}_{0\eta}(r)\exp\left[j(\eta\Omega t - \xi\psi - \phi)\right]$$

where η, ϕ, and ξ are phase constants. The sound pressure from one component η is, then,

$$SP_\eta = -\frac{(\bar{x} - \bar{y})}{4\pi D^2} f_{0\eta}\left(\frac{j\eta\Omega}{a_0} + \frac{1}{D}\right)\exp\left\{j\left[\eta\Omega\left(t - \frac{D}{a_0}\right) - \xi\psi - \phi\right]\right\} \qquad \text{(A3)}$$

Let

$$|\bar{f}_{0\eta}(r)| = P_\eta(r)|d\bar{A}| = P_\eta(r)\Omega\ dr\ d\psi$$

Then the radiation from the entire disk is

$$SP_\eta = \int_0^{2\pi}\int_0^{r_t} -\frac{R}{4\pi D^2}P_\eta(r)\Big[\sin\beta\cos\sigma'\sin(\psi-\theta)$$
$$+\cos\beta\sin\sigma'\Big]\left(\frac{j\eta\Omega}{a_0}+\frac{1}{D}\right)\exp\left\{j\left[\eta\Omega\left(t-\frac{D}{a_0}\right)-\xi\psi-\phi\right]\right\}r\,dr\,d\psi \qquad \text{(A4)}$$

At this stage the following approximations are made: (1) the observer remains in the far field; (2) θ is assumed to be zero for computational simplicity; and (3) the loading is assumed to be concentrated over Δr at an effective radius r_e. With the use of series identities for integration over ψ, equation (A4) reduces to

$$SP_\eta = \frac{\eta\Omega}{2Ra_0}\exp\left\{j\left[\eta\Omega\left(t-\frac{R}{a_0}\right)-\phi\right]\right\}r_eP_\eta(r_e)\Delta r\left(\cos\beta\sin\sigma'\right.$$
$$\left.-\sin\beta\frac{\xi}{\eta M_e}\right)J_\xi\left(\eta M_e\cos\sigma'\right)(-j)^{(\xi+1)} \qquad \text{(A5)}$$

where $J_\xi(\ )$ is the Bessel function of order ξ and M_e is the effective rotational Mach number. This solution is the resulting sound field from a general pressure pattern $P_\eta(r_e)\exp\left[j(\eta\Omega t-\xi\psi-\phi)\right]$. Once the fluctuating pressures on the rotor disk are determined, the far-field radiated discrete noise is then determined.

A general blade pressure pattern and time history of the loading is shown in figure 13. This unsteady periodic loading can be divided into two groups: uniform and nonuniform. The uniform pressures can be modeled as

$$f(t) = \sum_{k=-\infty}^{\infty} C_k \exp j(kB\Omega t - kB\psi) \qquad \text{(A6)}$$

where $C_k = \frac{LB}{\pi r_e}\chi_k$ and χ_k is the chord spectrum function which accounts for the distribution of the pressures over the chord.

For real cases χ_k will not be a simple function. Table I lists simple distributions and their respective chord functions χ. The uniform pressures have a one-one correspondence with discrete noise, that is, the kth pressure harmonic influences only the kth sound harmonic as shown in reference 7. For a particular sound harmonic m the mode of interest is

$$P_{mB} = \underline{P}_{mB} \exp^{j}(mB\Omega t - mB\psi) \tag{A7}$$

where $\underline{P}_{mB} = \frac{LB}{\pi r_e}\chi_{mB}$ and $\underline{P}_{mB}$ is independent of ψ for uniform pressures. If the nonuniform pressures are included, then

$$P_{mB} = \underline{P}_{mB}(\psi) \exp\left[j(mB\Omega t - mB\psi)\right] = \sum_{S=-\infty}^{\infty} \underline{P}_{mB,s} \exp\left[j(s\psi - \phi_s)\right] \tag{A8}$$

Combining equations (A7) and (A8), assuming $\phi_s = -\phi_{-s}$ and $\underline{P}_{mB,-s} = \underline{P}_{mB,s}$, and letting $\alpha_s = \frac{P_s}{P_0}$ = Loading harmonic coefficient, the pressure mode for mB sound harmonic and s loading harmonic is obtained

$$P_{mB} = \underline{P}_{mB} \frac{\alpha_s}{2}\left\{\exp\left[j\left(mB\Omega t - q_{-}\psi - \phi_s\right)\right] + \exp\left[j\left(mB\Omega t - q_{+}\psi + \phi_s\right)\right]\right\} \tag{A9}$$

where

$$q_{\pm} = mB \pm s$$

Comparing equation (A9) with the general pressure pattern P_{η}, we find

$$\eta = mB, \quad \xi = q_{\pm}, \quad P_{\eta}\left(r_e\right) = \frac{LB\alpha_s}{2\pi r_e}\chi_{mB}, \text{ and } \phi = \phi_s$$

Therefore

$$\begin{aligned} \underline{SP}_{mB,s} = \frac{\alpha_s \Omega \chi_{mB}}{2Ra_0} LB\Delta r \Bigg[&\left(\cos\beta \sin\sigma' - \frac{q_{-}}{mBM_e}\sin\beta\right) mBJ_{q_{-}}\left(mBM_e \cos\sigma'\right) \\ &+ \left(\cos\beta \sin\sigma' - \frac{q_{+}}{mBM_e}\sin\beta\right) mBJ_{q_{+}}\left(mBM_e \cos\sigma'\right)\Bigg] \end{aligned} \tag{A10}$$

Now,

$$LB\Delta r = L_T$$

and $L_T \cos\beta = L_F$ is the total rotor lift force and $L_T \sin\beta = D_F$ is the total drag force. Hence,

$$\underline{SP}_{mB,s} = \frac{\alpha_s}{2}\chi_{mB}\left[\left(K_L - \frac{K_D q_{-}}{mB}\right)\gamma_{q_{-}} + \left(K_L - \frac{K_D q_{+}}{mB}\right)\gamma_{q_{+}}\right] \tag{A11}$$

where

$$K_L = \frac{N}{Ra_0} L_F \sin \sigma'$$

$$K_D = \frac{N}{Ra_0}\frac{D_F}{M_e}, \quad N = \frac{\Omega}{2\pi}$$

The values $\gamma_q = mBJ_q\left(mBM_e \cos \sigma'\right)$ and $\alpha_s = \frac{P_s}{P_0}$ loading harmonic coefficients. Equation (A11) is the sound radiation from a particular loading harmonic s. It does not include any phase terms. Now if all the phase terms are included, sum over s and then convert it into a sum over q. Finally, the sound radiation at mB sound harmonic is obtained as

$$\begin{aligned} SP_{mB} = {} & \exp\Big[jmB\Omega\Big(t - \frac{R}{a_0}\Big)\Big]\chi_{mB}\Bigg(\frac{\alpha_0}{2}\Big(K_L - K_D\Big)\gamma_{mB} \exp\Big[j\Big(mB\theta - \phi_0\Big)\Big]\Big(-j^{mB+1}\Big) + \frac{\alpha_{mB}}{2} K_L \gamma_0 \exp\Big(-j\phi_{mB}\Big)(-j) \\ & + \sum_{q=1}^{\infty} \frac{\alpha_{|q-mB|}}{2}\Big(K_L - \frac{K_D q}{mB}\Big)\gamma_q \exp\Big\{j\Big[q\theta + \Big(2S_q - 1\Big)\phi_{|q-mB|}\Big]\Big\} + \frac{\alpha_{|mB+q|}}{2}\Big(K_L + \frac{K_D q}{mB}\Big)\gamma_q \exp\Big[-j\Big(q\theta - \phi_{|mB+q|}\Big)\Big]\Big(-j^{q+1}\Big)\Bigg) \end{aligned} \tag{A12}$$

where

$$S_q = 0 \quad q < mB$$

and

$$S_q = 1 \quad q \geqq mB$$

References 1 and 2 showed that it is not necessary to sum infinite values of q. Based on equation (A12) with the Bessel function values, the upper bound for q is found to be

$$Q = \left[10 + \frac{5}{4} mBM_e \cos \sigma'\right] \tag{A13}$$

where $[\ ]$ implies the integer part of $10 + \frac{5}{4} mBM_e \cos \sigma'$.

APPENDIX B

MANAGEMENT OF THE TRANSDUCER SPECTRUM DATA

The Stationary Rotor Rotational Noise Prediction Package is designed to accept up to 30,000 spectrum data (amplitude and phase) from up to 24 transducers. In order to conserve computer storage needs, only small blocks of spectrum data at any given time need to be considered. The purpose of appendix B is to acquaint the user with spectrum-data management techniques used in each of the five programs comprising the package.

The spectrum data are input to program RAMANI from magnetic tape in transducer blocks of 1500. The RAMANI input tape format is illustrated in table XIII for 3400 spectrum-data points per transducer and with 7 transducers.

The input spectrum data are initially stored on a random access file to facilitate ease of handling, thus conserving central processing unit (CPU) time. The random access format of the spectrum data is similar to the tape (or disk) format of table XIII, except for the following: (1) a parameter denoting the number of blocks of spectrum data per transducer is added to the transducer identification record, and (2) a tape spectrum-data record containing both amplitude and phase becomes two random access records (one containing the amplitudes and the other containing the phases).

Therefore, the 28 records of the input tape illustrated in table XIII become 49 random access records.

The transducer spectra are ready to be combined by addition or by scaling. If addition is to occur, a corresponding amplitude (phase) block is selected from each of the two transducers. If scaling is to occur, only one amplitude (phase) block is selected. The combined spectrum data are now output to disk (tape) and, optionally, to the random access. To conserve random access storage the combined spectrum data are written over the original input spectra. (This overwriting produces the restrictions placed on the RAMANI input arrays ISTN and JSTN.) Table XIV describes the random access format if 4 combined spectrum-data positions are produced from the spectrum data of the previous example. The RAMANI output tape has a format similar to the random access format with the following exceptions: (1) two random access records containing corresponding amplitude and phase blocks are reformed into one tape record; and (2) the steady loading and amplitude average for each position form a record after the last spectrum-data block from that position, instead of in arrays as in the random access format. Therefore, the 30 random access records of table XIV become 20 tape (disk) records.

The combined spectrum data are now fed to the plotting program SPECPLT or to the spectrum-data reorganization program TRANS. If the program RAMANI is job-stepped with SPECPLT or TRANS, the random access records can be used. If the programs are not run simultaneously, the tape must serve as the medium of spectrum-data input because the random access is only temporary storage. If the magnetic tape is used to input the combined spectrum data to either SPECPLT or TRANS, the data are first stored on random access as in table XIV.

In program SPECPLT consider the use of the combined spectrum data in generating the amplitude and phase spectrum plots. The total number of random access records used in storing the combined spectrum data (NRCSUM) is internally computed. The X (frequency) values and Y (amplitude or phase) values are stored in blocks on random access immediately following the combined spectrum data. The plotting data for succeeding plots are restored over these same random access records. If TRANS is to run in the job-step mode with program SPECPLT, the combined spectrum data are already prestored on random access.

Assume that in program TRANS the combined spectrum data have been input or stored on random access file in the format of table XV. TRANS reorganizes the spectrum data and restores the data so that up to 250 amplitude (or phase) points from each combined position comprise a typical random access record. The reorganized spectrum data are positioned on the random access file after the combined spectrum data. Table XV illustrates the reorganization of the spectrum data.

Since program RNPPE4 is run behind TRANS in a job-step mode, the reorganized spectrum is passed from TRANS to RNPPE4 through random access. (Necessary counters and points are passed to RNPPE4 through two random access records, NREC = 1 and NREC = 2.) If the BLH data (coefficients and phases) are determined through integration, the random access is again utilized for temporary storage. The BLH data are stored over the unorganized spectrum data (in records indexed from 3 to NRCSUM) or following the reorganized spectrum data (in records indexed from record NRCSUM + 2*NREAD + 1). The storage location depends on the number of combined positions and the number of points per position. BLH data from successive iterations overstore the original set of integrated BLH data.

The SPL and BLH data are written to a disk by case as the data are computed. The format of the SPL and BLH is illustrated in table XV for the preceding example.

Since program SPLPLT is run behind RNPPE4 in a job-step mode, the SPL and BLH plotting data are passed to SPLPLT by disk. The data to be plotted are read from the disk sequentially unless an integrated phase plot is requested. In this case, the disk is repositioned by the FORTRAN BACKSPACE i statement.

Summary of the management of transducer spectrum data follows:

APPENDIX B

1. The storage of the spectrum data and the corresponding SPL and BLH values involves the utilization of magnetic tape, serial disk, and random access for the purpose of reducing storage requirements and reducing CPU time.

2. Magnetic tape is used when permanent storage of the spectrum data is desired.

3. Serial disk is used when temporary storage of the spectrum data with corresponding BLH and SPL data is desired, and when these data can be processed sequentially.

4. Random access is used as a temporary storage medium when the data are to be processed in a variable user specified order.

APPENDIX C

RANDOM ACCESS FILE USAGE

The transducer spectrum data used by the programs in the SRRNPP are often used nonsequentially. As described in appendix B, the programs in the SRRNPP utilize a random access file to manipulate and to access the spectrum data. This appendix is designed to explain the motivation behind using a random access file and to document the three system resident FORTRAN callable random access subroutines OPENMS, WRITMS, and READMS.

In all four programs (RAMANI, SPECPLT, TRANS, RNPPE4) numbered indexing is used and the size of the index is 987. Therefore, a maximum of 986 logical records is permitted on the random access file and each record is assigned a number between 1 and 986.

The subroutines in this appendix are taken from Volume II of reference 6.

INTRODUCTION

It is often advantageous to use a random access file whenever it is necessary to access records within a file in a nonsequential manner. Since a data cell file or a magnetic tape file can only be accessed sequentially, these files cannot be used as random access files. Therefore, only disk files can be used as random access files.

An index of disk addresses is used in accessing a random access file. The disk addresses are pointers to operating system tables. In processing a random access file, the operating system returns a disk address in the index when a logical record is written. The operating system accepts a disk address from the user whenever a logical record is read.

Using a random access file, where applicable, normally results in a reduction in execution time when the records are accessed in a nonsequential manner. For instance, on a test using 1000 logical records, the use of a random access file took 1/15 as much CPU time and 1/22 as much peripheral processing unit (PPU) time as the use of a sequential file.

The drawbacks of a random access file are that extra central memory core is required for the index and that the number of logical records per file is limited by the size of the index. Also, data cannot be stored permanently on a random access file since the disk is a temporary storage medium.

APPENDIX C

SUBROUTINE OPENMS

Language: COMPASS

Purpose: To open a random access file.

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

where

U is the logical unit number.

IX is the first word address of the index.

L is the length of the index.

P = 0 for numbered indexing.
= 1 for named indexing.

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

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

Accuracy: Not applicable.

References: None.

Storage: 103_8 locations.

Subprograms used: GETBA, SIO$, SYSTEM

APPENDIX C

Error messages: (1) UNASSIGNED MEDIUM FILE XXXXXX

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

(3) INDEX BUFFER IS OF INSUFFICIENT LENGTH. XXXXXX

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

Source: Control Data Corporation.

Responsible person: Mickey G. Rowe, NASA Langley Research Center. D. E. Newell, Computing and Software, Incorporated.

APPENDIX C

SUBROUTINE WRITMS

Language: COMPASS

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

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

where

U is the logical unit number.

FWA is the central memory address of the first word of the record.

N is the number of central memory words to be transferred.

I is the record number or record name depending upon the indexing mode set by the initial call to OPENMS.

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

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

Accuracy: Not applicable.

References: None.

Storage: 102_8 locations.

Subprograms used: GETBA, SYSTEM, SIO$

Error messages: (1) UNASSIGNED MEDIUM, FILE XXXXXXX

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

(3) INDEX BUFFER IS OF INSUFFICIENT LENGTH

Source: Control Data Corporation

APPENDIX C

Responsible person: Mickey G. Rowe, NASA Langley Research Center. D. E. Newell, Computing and Software, Incorporated.

APPENDIX C

SUBROUTINE READMS

Language: COMPASS

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

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

where

U is the logical unit number.

FWA is the central memory address of the first word of the record.

N is the number of words of the record to be transferred.

I is the record number or record name depending upon the indexing mode set by the initial call to OPENMS.

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

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

Accuracy: Not applicable.

References: None.

Storage: 131_8 locations.

Subprograms used: GETBA, SYSTEM, SIO$

Error messages: (1) UNASSIGNED MEDIUM, FILE XXXXXXX

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

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

(4) *READ PARITY ERROR*

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

Termination is abnormal.

Source: Control Data Corporation.

Responsible person: Mickey G. Rowe, NASA Langley Research Center. D. E. Newell, Computing and Software, Incorporated.

APPENDIX D

PLOTTING GUIDE

All plotting in programs SPECPLT and SPLPLT is done on a Calcomp 12-inch drum plotter. The Langley Research Center (LRC) Graphic Output System is the postprocessing system utilized by both programs; therefore, only a discussion of SPECPLT is included herein.

Before the executionof program SPECPLT, the applications graphic software routines discussed later in this appendix are loaded from the system tape onto the file containing the binary version of SPECPLT. At this point, SPECPLT is executed. The basic flow of program SPECPLT follows:

1. CALL PSEUDO initializes the graphics postprocessor and names the "plot vector file" created during execution (CALL PSEUDO (6LSPECTRA)).
2. CALL LEROY is made to set up parameters necessary for plotting with a liquid pen.
3. Calls are made to subroutines AXES, NOTATE, and NUMBER to draw and label the AXES and to annotate each plot.
4. CALL CALPLT creates the plotting arrays for the combined spectrum (or BLH and SPL) data.
5. CALL NFRAME signals the termination of a specific plot.
6. If more plots are requested, steps 3 to 5 are repeated. If all plotting is completed, a terminated call to subroutine CALPLT is made.

The above mentioned subroutines are documented later in this appendix. The plot vector file created during the execution of SPECPLT is itself executed through the loading of the postprocessor program. The user can control the execution of the plot vector file through the use of the plot control card. The plot control card for program SPECPLT follows:

```
PLOT.CALPOST.12 (PVF=SPECTRA.X0=2.0.Y0=0.5.FSH=14.FSV=11)
  //SINGLE PLOT MODE.

LEROY .3MM PEN TYPE       BLACK INK COLOR.

RAG TYPE PAPER.           Paper No. 00.//
```

The postprocessor program creates the separate plot file for the Calcomp plotter, prints messages and statistics on each frame, prints plotting operations controls, and permits certain user options. For program SPECPLT the specific user options (listed on the plot control card) are: (1) naming the plot vector file (SPECTRA); (2) setting

X-origin and Y-origin offsets (2.0 and 0.5 inches); (3) setting horizontal and vertical frame sizes (14 and 11 inches, respectively); (4) specifying plot mode (single); (5) specifying size and type of pen with desired ink color; and (6) specifying paper type and number.

A typical set of frame statistics and printed operations controls for SPECPLT is contained in table XVII. These statistics and controls are supplied to the user at the end of the printed output and to the plotter operator to provide necessary plotting instructions.

This discussion of the plotting techniques and use of the graph postprocessor applies to program SPLPLT as well as to program SPECPLT. The remainder of this appendix deals with documentation of the FORTRAN callable CALCOMP plotting subroutines used by SPECPLT and SPLPLT. The documentation is taken from reference 6 with a rearrangement of figures. Additional information concerning the CALCOMP plotting subroutines and the LRC Graphic Output System is also attainable from reference 6.

APPENDIX D

SUBROUTINE PSEUDO

Language: COMPASS

Purpose: To create and write an appropriately named Plot Vector File. Through linkages set up by an initial call to PSEUDO, all subsequent graphics data generated by the user will be routed through one of the PSEUDO entry points and written on the Plot Vector File. The PSEUDO processor is designed for use with the frame dependent postprocessors described in Section 3.2.3, Volume II of the Computer Programming Manual.

Use: CALL PSEUDO

or

CALL PSEUDO(FN)

FN file name left-justified with zero fill. Default file name is SAVPLT.

Example:

CALL PSEUDO

This will establish a Plot Vector File named SAVPLT.

CALL PSEUDO(6LMYFILE)

This will establish a Plot Vector File named MYFILE.

NOTE: The Plot Vector File (or Files) will usually be written to disk (as opposed to tape) and may be postprocessed following user program termination via appropriate specification of one or more plot control cards. (See Section 3.2.3, Volume II, Computer Programming Manual.)

Restrictions: (1) An initializing call to PSEUDO (with or without a file name argument) must be made prior to any calls to CALPLT or any other graphics output routine.

(2) Every Plot Vector File should be terminated with a 999 pen code, CALL CALPLT (0.0, 0.0, 999). The transmission of the 999 code will cause an EOF write on the Plot Vector File, and the file will temporarily be closed. Thus, any given Plot Vector File will contain only one 999 pen code and/or one EOF.

(3) To continue plotting execution following transmission of a 999 code to a current Plot Vector File, the user program must call the PSEUDO processor to create new Plot Vector File (i.e., CALL PSEUDO (6LMYFIL2)).

Method: In addition to entry PSEUDO, this processor contains two other entry points, namely PLT9999 and PLT9998. An initializing call to PSEUDO will set PLT9999 into the processor switching mechanism (PLOTSW). Subsequent plot data generation will then be routed via CALPLT, PLOTSW, and PLT9999 and written on the Plot Vector File. The entry PLT9998 is used to record special purpose data from routines NFRAME and PLTSTOP.

Accuracy:

References: See Section 3.2.3, Volume II, Computer Programming Manual.

Storage: 2247_8 locations total for direct subprograms.

Subprograms used: NUMARG, PLOTSW

Other coding information:

Source: E. C. Johnson, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE LEROY/BALLPT

Language: FORTRAN

Purpose: To set up the parameters necessary to accommodate plotting with the liquid ink pen. Once set, this mode will remain in effect unless a call to BALLPT is given.

The parameters for plotting with the ballpoint pen are reset by CALL BALLPT. This mode is automatically in effect unless there has been a call to LEROY.

Use: CALL BALLPT

CALL LEROY

Restrictions: The CALL LEROY should only be used with the CalComp. In addition to reducing the speed of the plotter for all plotting movements, the number of plot vectors in any annotation is considerably increased.

The CALL LEROY must be made prior to any plotting calls, but after the CALL PSEUDO or CALL CDC250.

Method:

Accuracy:

References:

Storage: 40_8 locations 6000 Series.

Subprograms used: CALPLT

Other coding information:

Source: E. H. Senn, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE NFRAME

Language: FORTRAN

Purpose: To provide users specific means of executing frame advance movements on any plotter device via an appropriate frame-oriented device postprocessor. Frame advance distances are generally defined to be incremental from current frame origin (i.e., comparable to frame advance executions for the DDI or 252 CRT devices). CALL NFRAME is intended to be used as a frame advance mechanism, not as a plot origin offset.

Use: CALL NFRAME

or

CALL NFRAME(H,V)

where

H and V are Horizontal (parallel to device X) and Vertical (parallel to device Y) distances from the current frame origin. H and/or V must be expressed in floating-point inches.

The short form CALL NFRAME will cause the device postprocessor to execute a frame advance move parallel to the device X (horizontal) axis. The movement will be (FSH + h) inches, where FSH is the horizontal frame size and h will be an increment appropriate to the particular device (0 < h ≦ 2"). (See the Formal Parameters List of the plot control card, Section 3.2.3.2 for a more complete definition of frame size parameters FSH and FSV.)

When H and V parameters are provided on the NFRAME call, only the following values are permissible:

CALL NFRAME(H,0.)	Frame advance H" horizontal
CALL NFRAME(0.,V)	Frame advance V" vertical
CALL NFRAME(H,V)	Frame advance H" by V"
CALL NFRAME(0.,0.)	Return to current frame origin
CALL NFRAME(H,-V)	Frame advance H" by -V"

This should be used to execute a frame advance move parallel to horizontal axis and to establish a new origin for roll paper plotters and a return to origin for flatbed plotters.

Restrictions: 1. This routine is intended for use only in concert with the frame-dependent graphics postprocessors.

2. This routine must be used in any case which may require AUTO modification of Plot Vector File data by a graphics postprocessor.

3. The frame advance distances specified by H and/or V should always be at least slightly greater than the intended usable frame size.

4. The H dimension of a frame advance may not be negative ($H \geqq 0$). For purposes of frame stacking, V may be either negative, zero, or positive. (See Section 3.2.3 for a definition of frames and frame advances.)

Method:

Accuracy:

References: Computer Programming Manual, Volume II, Section 3.2.3.

Storage: 76_8 memory words, CDC 6000 series.

Subprograms used: NUMARG, CALPLT, PLT9998, ABORT

Other coding information:

Source: E. H. Senn, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE CALPLT

Language: FORTRAN

Purpose: To move the plotter pen to a new location with pen up or down and to signal the end of a job segment by incrementing the block address number.

Use: CALL CALPLT(X,Y,IPEN)

where

X,Y are the floating-point values for pen movement.

IPEN = 2 pen down

= 3 pen up

Negative IPEN will assign X = 0, Y = 0 as the location of the pen after moving the X,Y (create a new reference point) and will increase the block number by one. (This number is that which appears in the display at the top of the tape drive on the plotter and identifies the portion of the output tape that is being plotted. The block address 001 is written automatically as a result of the initialization PSEUDO processor call.) Each block address generally implies a separate page or plot.

= 999 Writes a terminating block address of 999 to terminate the Plot Vector File and all further processing is skipped.

CALL CALPLT(0.0,0.0,999)

Restrictions: All X and Y coordinates must be expressed as floating-point inches (actual page dimensions) in deflection from the origin.

A TERMINATING BLOCK ADDRESS (IPEN = 999) MUST BE GIVEN AS THE LAST PLOTTING INSTRUCTION BEFORE ENDING A PROGRAM WHICH USES ANY OF THE PLOTTER SUBROUTINES; THIS IS TO BE SURE THAT ALL PLOTTER INSTRUCTIONS ARE WRITTEN ON THE PLOTTER TAPE.

Method: The main subroutine in the graphics language is the CALPLT subroutine. All other special purpose subroutines eventually call CALPLT either directly or indirectly. This routine moves the pen in a straight line between the present pen position and another pen location to which the programer wishes the pen to be moved.

In order to cause such instructions to be written, the programer specifies the coordinates of the point to which the pen is to be moved and whether the pen is to be moved in a raised or lowered position. This is accomplished by the FORTRAN instruction:

```
CALL CALPLT(X,Y,IPEN)
```

Also, the subroutine provides "sequence numbers" on the tape, making it possible to afford identification of job segments. The block address 001 is written on the first call to CALPLT. Thereafter, if the programer defines a new origin within the frame, he need only set the argument IPEN negative. The CALPLT routine then moves the pen to X,Y; stores this location as (0,0), that is, at a new origin; and increases the block address by one.

The following explanation in table form in conjunction with figures 14 and 15 will illustrate results of various commands to the plotter using the CALPLT subroutine. The figures illustrate the position of the pen after moving to its new position as the movement from in front of the plotter is observed. Each square represents a square inch. The dark circle represents the pen, the dashed lines and letters indicate the plotter directions at each change of origin.

FORTRAN statement	Subroutine action	Pen movement
CALL CALPLT(0.0,0.0,3)	The pen is raised and since X = 0.0 and Y = 0.0, the pen is not moved.	Pen is only raised.
CALL CALPLT(2.,2.,2)	Instructions are written causing the pen to be lowered and moved in the +X, +Y direction.	The pen is moved (note directions indicated by dotted lines) from its original position (fig. 14(a) considered (0,0)) to the point (2,2) (fig. 14(b)).
CALL CALPLT(0.,4.,2)	The pen remains lowered and instructions are written to cause movement in the -X,+Y direction.	Figure 14(c).
CALL CALPLT(0.,0.,2)	Pen remains lowered and instructions are written to cause movement in the -Y direction.	Figure 14(d).

FORTRAN statement	Subroutine action	Pen movement
CALL CALPLT(12.,6.,-3)	The pen is lifted; the routine computes the necessary steps to move the pen in order to approximate closely the line between (0,0) and (12.,6.). These instructions are written on the tape. The stored position of the pen is then reset to (0,0) and the block address 002 is written on the tape.	The pen is moved to the position shown in figure 15(a); this new position is now considered the origin (note directions indicated by dotted lines).
CALL CALPLT(0.,-7.,2)	Instructions are written to lower pen and to move in the -Y direction.	Figure 15(b).
CALL CALPLT(-5.,-7.,2)	Pen remains lowered; instructions are written to move pen in the -X direction.	Figure 15(c).
CALL CALPLT(-5.,0.,2) CALL CALPLT(0.,0.,2)	Instructions are written to move the pen in the +Y direction and then in the +X direction.	Top of figure 15(d).
CALL CALPLT(-7.,-3.,-3)	Instructions are written to lift the pen, the steps necessary to move the pen in order to approximate closely the line between (0,0) and (-7,-3) are computed and written on the tape. The stored position of the pen is set to (0,0) and the block address 003 is written on the tape.	Center of figure 15(d) (note directions in dotted lines). The pen is now raised and sitting over the point indicated in the figure; any further plotting will begin from that point unless the pen is moved manually beforehand.

FORTRAN statement	Subroutine action	Pen movement
CALL CALPLT(12.,6.,-3)	The pen is lifted; the routine computes the necessary steps to move the pen in order to approximate closely the line between (0,0) and (12.,6.). These instructions are written on the tape. The stored position of the pen is then reset to (0,0) and the block address 002 is written on the tape.	The pen is moved to the position shown in figure 15(a); this new position is now considered the origin (note directions indicated by dotted lines).
CALL CALPLT(0.,-7.,2)	Instructions are written to lower pen and to move in the -Y direction.	Figure 15(b).
CALL CALPLT(-5.,-7.,2)	Pen remains lowered; instructions are written to move pen in the -X direction.	Figure 15(c).
CALL CALPLT(-5.,0.,2) CALL CALPLT(0.,0.,2)	Instructions are written to move the pen in the +Y direction and then in the +X direction.	Top of figure 15(d).
CALL CALPLT(-7.,-3.,-3)	Instructions are written to lift the pen, the steps necessary to move the pen in order to approximate closely the line between (0,0) and (-7,-3) are computed and written on the tape. The stored position of the pen is set to (0,0) and the block address 003 is written on the tape.	Center of figure 15(d) (note directions in dotted lines). The pen is now raised and sitting over the point indicated in the figure; any further plotting will begin from that point unless the pen is moved manually beforehand.

APPENDIX D

Accuracy:

References:

Storage: 251_8 locations 6000 series.

Subprograms used: PLOTSW, STRCALL, LOCATE

Other coding information:

Source: George C. Salley, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE NUMBER

Language: FORTRAN

Purpose: To convert a floating-point number to BCD (expressed in F format), and to draw the resulting alphanumeric characters.

Use: CALL NUMBER(X,Y,HEIGHT,FPN,THETA,NODIGIT)

where

X,Y are the coordinates in floating-point inches of the left lower corner of the first digit of output.

HEIGHT is the height of the plotted number in floating-point inches. (See NOTATE routine.)

FPN is the floating-point number to be drawn.

THETA is the angle in floating-point degrees at which the number is to be drawn. (See NOTATE routine.)

NODIGIT is the number of decimal digits to the right of the decimal point for output.

NODIGIT=1 or NODIGIT=0 both specify no decimal places; however, -1 suppresses the decimal point.

Restrictions: The number is restricted to a maximum of 12 significant digits.

Method:

Accuracy: The routine truncates the floating-point number at the required decimal place.

References:

Storage: 271_8 locations 6000 series.

Subprograms used: NOTATE, ROUND, ALOG

Other coding information: Examples:

```
FPN = 12.34567891234
CALL NUMBER(X,Y,S,FPN,0.,-1)
will draw
    starting location → 12
CALL NUMBER(X,Y,S,FPN,0.,0)
will draw
    starting location → 12
CALL NUMBER(X,Y,S,FPN,0.,7)
will draw
    starting location → 12.3456789
```

Source: George C. Salley, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE NOTATE

Language: FORTRAN

Purpose: To draw alphanumeric information for annotation and labeling and to provide special centered symbols for annotation of data points.

Use: CALL NOTATE(X,Y,HEIGHT,BCD,THETA,NOCHAR)

where

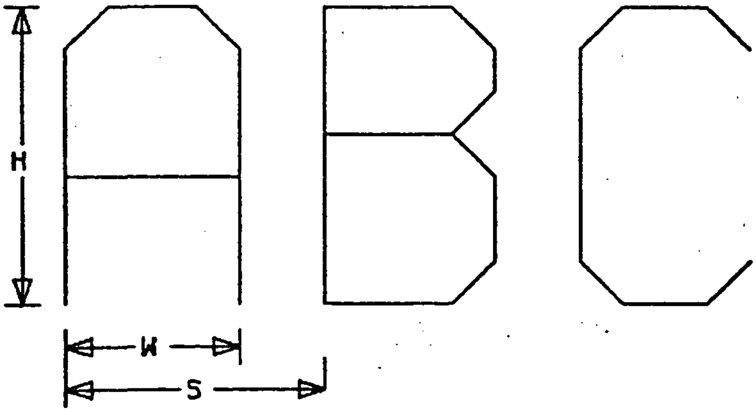
X,Y are the floating-point coordinates of the first character.

For alphanumeric characters, the coordinates of the lower left-hand corner of the characters are specified.

For special symbols 0 to 5, the coordinates of the center of the symbol are specified.

For special symbols above 6, the coordinates of the lower left-hand corner of the character are specified.

HEIGHT specifies character size and spacing in floating-point inches for a full-size character. The smallest possible character is 0.07 inch high. The width of a character will be (4/7)* HEIGHT and the space between characters is (2/7)* HEIGHT. (See sketch (a).)



H = HEIGHT

W = (4/7) * HEIGHT

S = (6/7) * HEIGHT + WADJ

Sketch (a)

The ith character is plotted at:

$$x_i = X + (i-1)\ (6/7)\ (\text{HEIGHT})\ (\cos\theta)$$
$$y_i = Y + (i-1)\ (6/7)\ (\text{HEIGHT})\ (\sin\theta)$$

BCD is the string of characters to be drawn and, is usually written in the form: nHXXXX--- (the same way an alpha message is written using FORTRAN format statements). Instead of specifying alpha information as above, the beginning storage location of an array containing alphanumeric information may be given.

Special symbols will be drawn when BCD is an integer reference and NOCHAR is negative. (See fig. 16.)

Note: A binary zero in the BCD string will cause truncation of plotting at that point, and a normal return to the calling program.

THETA is the angle in floating-point degrees at which the information is to be drawn. Zero degrees will print horizontally reading from left to right; 90^o will print the line vertically reading from bottom to top; 180^o will print the line horizontally reading from right to left (i.e., upside down); and 270^o will print vertically reading from top to bottom.

NOCHAR is the number of characters, including blanks, in the label. A negative NOCHAR will produce a single special symbol from the integer reference table. (See METHOD for further explanation.)

Restrictions: Noted under METHOD.

Method: The character height is a variable entry parameter to the subroutine NOTATE. However, the width-to-height ratio is fixed at 4/7. This is because the characters are defined by a series of bioctal offset pairs for a 4-by-7 matrix as shown by the examples in figure 17. The reference origin for the offset pairs which define each character is the lower left-hand corner of the matrix. The X and Y values, which are entry parameters to NOTATE, define the location of the lower left-hand corner of the first character to be plotted for this entry to NOTATE. Subsequent characters to be plotted are spaced from the previous character origin by 6/7 of the specified character height.

Figure 16 shows the characters available for the CDC 6000 series computers. The figure is divided into two main groups: (1) on the left are 62 symbols related to the 62 possible Hollerith card codes and (2) on the right is a set of special characters which are referred to by integer numbers. The entry parameter NOCHAR tells the subroutine

NOTATE which one of these two main groups is being referred to for this entry to NOTATE. If NOCHAR is a positive integer, this means that the entry parameter BCD is the location of an array of Console Display codes and that NOCHAR characters corresponding to these codes are to be plotted. (See fig. 18 for extended card codes.) If NOCHAR is negative, this means that the entry parameter BCD is the location of an integer number. This integer number refers to one of the special symbols in the special symbol table. Note here that when NOCHAR is positive (normal character pickup), more than one character may be plotted by a single entry to NOTATE. With the special character pickup (NOCHAR negative), only one character is plotted per NOTATE routine entry.

The special symbol table is divided into two groups. The group a symbol is in is based on the value of the integer number referring to the symbol.

Integer Reference Number	Subgroup
0 to 5	A set of six centered symbols. That is, the X,Y entry coordinates are the center of the character.
Above 5	Other special symbols whose reference origin is the lower left-hand corner the character.

For the first group of special symbols, the centering is achieved by the following technique. The symbols are defined by offset values within a 4-by-4 matrix with the first and last offset pair being the center of the symbol. The X and Y coordinates (which are entrance parameters to the subroutine) are then adjusted by

$$XS = X + H/2(\sin(\theta) - \cos(\theta))$$

and

$$YS = Y - H/2(\sin(\theta) + \cos(\theta))$$

where θ is an angle of rotation for the character and is an entry parameter to the subroutine. For example, when $\theta = 0$, the reference origin is shifted left and downward by one-half the character height. This results in the original X,Y location becoming the center of the symbol since it is centered in a 4-by-4 matrix. Of course, all offset values for the character are multiplied by the character height to dimension the symbol properly. To provide annotation at any angle of orientation, each offset pair for all the offset pairs for each character is transformed as follows:

X0 = an X offset value

Y0 = a Y offset value

X = reference origin of the character

Y = reference origin of the character

XS = final plot coordinate for this offset

YS = final plot coordinate for this offset

XS = X + H(X0*COS(θ) - Y0*SIN(θ))

YS = Y + H(X0*SIN(θ) + Y0*COS(θ))

Another option on entering the NOTATE routine is provided. If the X and/or Y values are 999., the plotting of symbols is resumed at the coordinates where it left off on the last entry to NOTATE. For example, if the first entry to NOTATE is

```
CALL NOTATE(X,Y,0.14,3HABS,0.0,3)
```

and the next entry is

```
CALL NOTATE(999.,999.,0.14,10,0.0,-1)
```

the following symbols will be plotted

ABS √‾

The symbol height and the character height were both specified as 0.14.

Another method of determining where the pen left off on the last entry to NOTATE or any other plotting routine is to use the routine WHERE (Section 3.2.4.100). WHERE returns the current value of X,Y and pen code.

```
CALL NOTATE(0.,0.,.20,3H100,0.0,3)

CALL WHERE(X,Y,IPEN)

CALL NOTATE(X,Y,.20,21,0.0,-1)
```

The following symbols will be plotted.

100%

APPENDIX D

Accuracy:

References:

Storage: 1152_8 locations 6000 Series.

Subprograms used: CALPLT, CNTRLN, DECOD1, DECOD2, SIN, COS

Other coding information:

Source: George C. Salley, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE PNTPLT

Language: FORTRAN

Purpose: To draw NASA Standard Plot symbols centered on a given coordinate value.

Use: CALL PNTPLT(X,Y,ISYM,IS)

where

X is the X coordinate for the centered symbol in floating-point inches.

Y is the Y coordinate for the centered symbol in floating-point inches.

ISYM is an integer specifying the symbol to be used. (See figs. 19 and 20.)
= 21 for a point
= 22 for a plus sign +

IS is an integer value specifying the size symbol to be used.
= 1 small
= 2 medium
= 3 large
(See fig. 19.)

Restrictions:

Method:

Accuracy: A positive integer value for ISYM in the calling sequence will produce symbols of the same quality as in figure 19. A negative integer value will produce symbols of less quality but will result in a considerably faster computer run.

References:

Storage: 506_8 locations 6000 Series.

Subprograms used: CALPLT, CIRCLE, CNTRLN

Other coding information:

APPENDIX D

Source: George C. Salley, NASA Langley Research Center.

Responsible person: Nancy L. Taylor.

APPENDIX D

SUBROUTINE AXES

Language: FORTRAN

Purpose: To draw a line, to annotate the value of the variable at specified intervals with or without tic marks, and to provide an axis identification label.

Use: CALL AXES(X,Y,THETA,S,ORG,SFX,TMAJ,TMIN,BCD,HEIGHT,NOCHAR)

where

X,Y are the coordinates in floating-point inches of the starting point of the axis with reference to the plotting area origin as established by CALPLT.

THETA is the angle of rotation measured counterclockwise from the X-axis in floating-point degrees. Normally, THETA is 0° for an X-axis and 90° for a Y-axis.

S is the length of the axis in floating-point inches. Should be a multiple of TMAJ.

+S will generate tic marks.
-S will eliminate tic marks.

ORG is the functional value to be assigned to the origin (i.e., the value of the first scale) in floating point.

SFX is the adjusted scale factor for the array to be plotted (change in value per inch).
NOTE: Values of ORG and SFX which will produce a reasonable scale may be calculated using subroutine ASCALE or BSCALE.

TMAJ is the distance in floating-point inches for major tic marks (0.25 inch high). Numbers are placed on the axis at the major tic marks in accordance with the values of ORG and SFX. The numbers written along the axis are adjusted to be between 1000.00 and 0.01 in magnitude. Immediately after the last number on the axis is placed the caption $\times 10^{exp}$, where exp is the required exponent.

If the values are integer multiples, the decimal point and decimal places are eliminated. A negative TMAJ will cause the actual value to be written instead of the adjusted value.

TMIN is the number of divisions per inch in floating point for minor tic marks (0.125 inch high). To eliminate minor tic marks the following may be used:

TMIN = 0.

BCD is the character label for the axis (see NOTATE routine).

HEIGHT is the height of the full-size characters in the BCD title. Numbers at the tic marks will be (0.75 *HEIGHT) high. HEIGHT is in floating-point inches.

If HEIGHT = 0., all annotation will be eliminated.

NOCHAR is an integer specifying the number of characters in BCD title. A negative NOCHAR places the annotation on the clockwise side of the axis and a positive NOCHAR places the annotation on the counterclockwise side of the axis. NOCHAR = 0 is not allowed. If it is desired to have no label, then the BCD parameter should be 1H, and NOCHAR either +1 or -1.

Restrictions: Only perpendicular axes are recommended.

Method:

Accuracy:

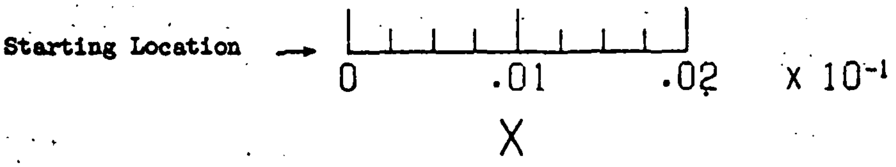
References:

Storage: 1451_8 locations 6000 Series.

Subprograms used: CALPLT, NOTATE, NUMBER, ROUND, SIN, COS, WHERE

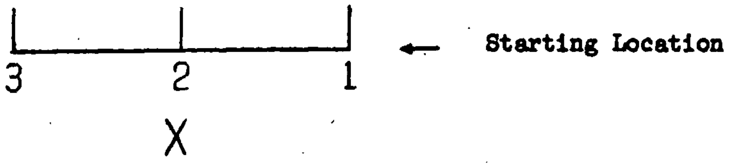
Other coding information: Examples:

CALL AXES(X,Y,0.0,2.0,0.0,.001,+1.,4.0,1HX,0.2,-1)

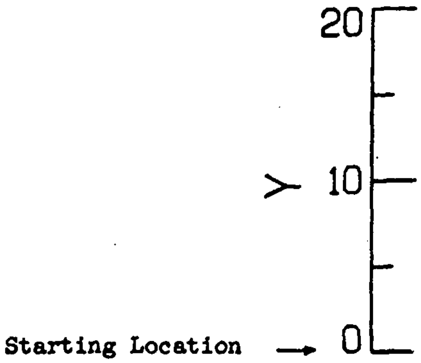


APPENDIX D

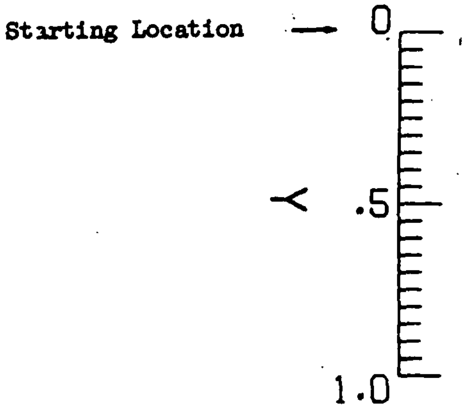
```
CALL AXES(X,Y,180.0,2.0,1.0,1.0,-1.,0.0,1HX,0.2,+1)
```



```
CALL AXES(X,Y,90.0,2.0,0.0,10.0,-1.,2.0,1HY,0.2,+1)
```



```
CALL AXES(X,Y,270.0,2.0,0.0,0.5,-1.,10.0,1HY,0.2,-1)
```



Source: George C. Salley, NASA Langley Research Center.

QUESTIONS ON THE USE OF THIS PROGRAM SHOULD BE DIRECTED TO THE ACD PROGRAMER SUPPORT GROUP, EXTENSION 3548.

APPENDIX E

NUMERICAL SUBROUTINE DOCUMENTATION

Appendix E provides documentation of the numerical subroutines used in program RNPPE4. Subroutines SPLS and CSIUNI are used by computing blade-loading data by integration of the steady loading distribution over all or part of the chord. Subroutine BSSLS is used in evaluating Bessel functions in Wright's solution.

APPENDIX E

SUBROUTINE SPLS

Language: FORTRAN

Purpose: To compute $\int_{X(KI)}^{X(KJ)} f_i(X)\, dX$ by a cubic spline approximation where X(KI) and X(KJ) are the first and last points of the interval and i = 1, 2, . . ., NCVS.

Use: CALL SPLS(MNPTS, N, NCVS, X, Y, KI, KJ, PROXIN, IW, WK, IERR)

MNPTS An input integer specifying the maximum number of values in the independent variable array as stated in the dimension statement of the calling program.

N An input integer specifying the number of points in the independent variable array. $N \leqq MNPTS$.

NCVS An input integer specifying the number of dependent variable tables associated with the independent variable.

X A one-dimensional input array containing the independent variables. X should be dimensioned by at least N in the calling program.

Y A two-dimensional input array containing the dependent variables. The array Y is dimensioned with variable dimension in the subroutine; therefore, Y must be dimensioned in the calling program with first dimension MNPTS and second dimension at least NCVS.

KI An input integer specifying the X array index containing the lower limit of integration.

KJ An input integer specifying the X array index containing the upper limit of integration.

PROXIN A one-dimensional output array in which $\int_{X(KI)}^{X(KJ)} f(X)\, dX$ is stored for each dependent variable array.

PROXIN must be dimensioned by at least NCVS in the calling program.

IW An input integer used for initialization. On first entry into SPLS the user must set IW = -1. The routine will then test the independent variable array to determine if it is increasing, and, also, compute certain values pertaining to the X and Y arrays.

These values will not change unless either the X or Y array is replaced.

WK An array used by SPLS as a work area. WK must be dimensioned at least 2(N × NCVS) + 8N. This array should not be used elsewhere in the program.

IERR An output error integer

= 0 Normal return

= 1 The independent variable array is not increasing. An error message will be printed by SPLS. "INDEPENDENT VARIABLE ARRAY NOT INCREASING IN SPLS AT POSITION IIII X = XXXX,XXXX."

Upon return to the calling program, the parameter IERR should be tested.

Restrictions: All arrays must be floating point. The values in the independent variable array must be increasing.

Method: The method used in SPLS is that of the reference. The reference gives the derivative of a matrix equation relating the second derivative of a univariate spline function at the given values of the independent variable to the values of the function at these values of the independent variable. Values of the second derivative are assumed to be zero at the end points. The matrix equation is tridiagonal and is solved by the Thomas algorithm which is equivalent to Gaussian elimination without pivoting. Expressions are derived for the integral in terms of the spline function and its second derivative at the nodes of the specified interval.

Accuracy: Cubic spline functions yield a best fit to a set of data. When fitting data derived from an analytical function, the cubic spline function yields an approximation of that function to an order of h^4 where h is the interval size of the independent variable of the spline fit. The accuracy of the integral is of an order of h^5. Care should be taken when fitting data where large gradients exist. Large gradients may cause extreme oscillations in the spline function.

References: Greville, T. N. E., "Spline Functions, Interpolation and Numerical Quadrature," Mathematical Methods for Digital Computers, Vol. II, pp. 156-168, John Wiley & Sons, 1967.

Storage: 556_8 locations.

Subprograms used: None.

FORTRAN functions: None.

Other coding information: It is recommended that the independent variable array contain at least four points, N ≧ 4.

EXAMPLE:

To compute the integral from X(KI) to X(KJ) for two dependent variable tables, the coding would be as follows:

```
          DIMENSION X(10), Y(10,2), PROXIN(2), WK(120)
          .
          .
          .
          MNPTS = 10
          N = 10
          NCVS = 2
C
C         TO COMPUTE THE INTEGRAL FROM X(1) TO X(3)
          KI = 1
          KJ = 3
          IW = -1
          CALL SPLS(MNPTS,N,NCVS,X,Y,KI,KJ,PROXIN,IW,WK,IERR)
          IF(IERR.EQ.0) GO TO 10
          .
          .
          .
   10     CONTINUE
C
C         TO COMPUTE THE INTEGRAL FROM X(2) TO X(7)
          KI = 2
          KJ = 7
          CALL SPLS(MNPTS,N,NCVS,X,Y,KI,KJ,PROXIN,IW,WK,IERR)
          IF(IERR.EQ.0) GO TO 11
          .
          .
          .
   11     CONTINUE
```

APPENDIX E

Source: NASA Langley Research Center, Computer Mathematics and Programing Branch.

QUESTIONS ON THE USE OF THIS PROGRAM SHOULD BE DIRECTED TO THE ACD PROGRAMER SUPPORT GROUP, EXT. 3548.

APPENDIX E

SUBROUTINE CSIUNI

Language: FORTRAN

Purpose: To perform a cubic spline interpolation on a univariate function for any number of different dependent variable arrays associated with the independent variable array.

Use: CALL CSIUNI(MNPTS,N,NCVS,MMAX,M,X,Y,T,F,IW,WK,IERR)

MNPTS An input integer specifying the maximum number of values in the independent variable array as stated in the dimension statement of the calling program.

N An input integer specifying the number of values in the independent variable array, $N \leqq MNPTS$.

NCVS An input integer specifying the number of dependent variable tables associated with the independent variable.

MMAX An input integer specifying the maximum number of values at which interpolation is desired as stated in the dimension statement of the calling program.

M An input integer specifying the number of values to be interpolated on this entry into CSIUNI, $M \leqq MMAX$.

X A one-dimensional input array containing the independent variables. The array X should be dimensioned by at least N in the calling program.

Y A two-dimensional input array containing the dependent variables. The array Y is dimensioned with variable dimension in the subroutine; therefore, Y must be dimensioned in the calling program with first dimension MNPTS and second dimension at least NCVS.

T A one-dimensional input array containing the values of the independent variable for which values of the dependent variable are desired. The array T must be dimensioned by at least M in the calling program.

F A two-dimensional output array in which CSIUNI stores the values of the function at the M values of the independent variable. The array F is dimensioned with variable dimension in the subroutine; therefore, F must be dimensioned in the calling program with first dimension MMAX and second dimension at least NCVS.

IW An input-output integer.

INPUT: IW is the initialization integer. On first entry into CSIUNI the user must set IW = 1. This will cause the independent variable array to be tested to determine if it is increasing. Also, certain values pertaining to the X and Y arrays will be computed. These values will not change unless either the X or Y arrays are replaced.

OUTPUT: IW is an index pointer indicating that $X_{IW} \leqq X_0 \leqq X_{IW+1}$. On the next call to CSIUNI, the previous IW is used to begin the search for the interval containing the interpolation point. Linear extrapolation is provided; therefore, IW = 0 indicates lower end extrapolation, and IW = N indicates upper end extrapolation.

WK An array used by CSIUNI as a work area. WK must be dimensioned at least 3(N×NCVS) + 8N.

IERR An output integer error code

= 0 Normal return

= 1 The independent variable array is not increasing. CSIUNI will print "INDEPENDENT VARIABLE ARRAY NOT INCREASING IN CSIUNI AT POSITION IIII,X=XXXX,XXXX."

Upon return to the calling program, the parameter IERR should be tested.

Restrictions: All arrays must be floating point. The values of the independent variable array must be increasing.

Method: The method used in CSIUNI is that of the reference. The reference gives the derivative of a matrix equation relating the second derivative of a univariate spline function at the given values of the independent variable to the values of the function at these values of the independent variable. Values of the second derivative are assumed to be zero at the end points. The matrix equation is tridiagonal and is solved by the Thomas algorithm, which is equivalent to Gaussian elimination without pivoting.

Accuracy: Cubic spline functions yield a best fit to a set of data. When fitting data derived from an analytical function, the cubic spline function yields an approximation of that function to an order of h^4 where h is the interval size of the independent variable of the spline fit. When fitting arbitrary sets of data, care should be taken in interpolating between the nodes to assure that the spline fit is satisfactory to the user. This is especially important when fitting data where large gradients exist. Large gradients may cause extreme oscillations in the spline function.

APPENDIX E

References: Greville, T. N. E., "Spline Functions, Interpolation and Numerical Quadrature," Mathematical Methods for Digital Computers, Vol. II, pp. 156-176, John Wiley & Sons, 1967.

Storage: 1014_8.

Subprograms used: None.

FORTRAN functions: None.

Other coding information: Linear extrapolation is available but should be used with care. It is recommended that the independent variable array contain at least four points, $N \geqq 4$.

EXAMPLE:

To interpolate for two dependent variable tables, the coding would be as follows:

```
      DIMENSION X(10),Y(10,2),T(100),F(100,2),WK(140)
      .
      .
      .
      MNPTS = 10
      N = 10
      NCVS = 2
      MMAX = 100
C
C     TO INTERPOLATE FOR 50 VALUES
C
      M = 50
      IW = -1
      CALL CSIUNI (MNPTS,N,NCVS,MMAX,M,X,Y,T,F,IW,WK,IERR)
      IF(IERR.EQ.0) GO TO 10
      .
      .
      .
10    CONTINUE
C
C     TO USE A NEW X ARRAY OF 5 VALUES
C
```

```
          N = 5
          IW = -1
          CALL CSIUNI(MNPTS,N,NCVS,MMAX,M,X,Y,T,F,IW,WK,IERR)
          IF(IERR.EQ.0) GO TO 100
          .
          .
          .
100       CONTINUE
```

Source: NASA Langley Research Center, Computer Mathematics and Programing Branch.

QUESTIONS ON THE USE OF THIS PROGRAM SHOULD BE DIRECTED TO THE ACD PROGRAMER SUPPORT GROUP, EXT. 3548.

APPENDIX E

SUBROUTINE BSSLS

Language: FORTRAN

Purpose: To compute Bessel functions of the first kind, $J_N(X)$ for a positive real argument, from order zero to order N-1.

Use: CALL BSSLS(X,F,N)

X An input variable specifying the argument for which the Bessel functions are to be evaluated.

N An input integer specifying the maximum order Bessel function desired.

F A one dimensional output array containing the Bessel functions from order zero to order N-1. Because of the recursion technique employed by the routine, the user must reserve at least

$N + 28$ if $N \geqq 1.5*[X]$
$1.5*[X] + 28$ if $N < 1.5*[X]$

locations for F. $[X]$ is the integral part of the argument X.

Restrictions: N is assumed positive and X is assumed non-negative. Labeled COMMON with FIX for a block name may not be used. This identifier is used by BSSLS.

Method: BSSLS is a Langley Research Center (LRC) mathematics library system subroutine BJIR, C3.2 (see program ref. 1) modified to allow computation of Bessel functions of order greater than 30.

A backward recursion technique is used as suggested in program reference 2. An arbitrary value (small) is assigned to a large order Bessel function determined by k, a sufficiently large number. Assuming $J_k(x) \neq 0$ and $J_{k+1}(x) = 0$, the recursion formula

$$J_{k-1}(x) = \frac{2k}{X} J_k(x) - J_{k+1}(x)$$

is used to compute all of the Bessel functions from $J_{k-1}(x)$ down to $J_0(x)$ of the same given argument. Finally the equation

$$J_0(x) + 2J_2(x) + 2J_4(x) + \ldots = 1$$

is used to normalize the resulting numbers to their correct values.

Accuracy: A check of the routine was made with the following arguments: X = 0.0, 0.1, 1.0, 5.0, 10.0, 50.0, and 100.0. Selected values for N were N = 0, 1, 2, 10, 20, 30, 40, 50, and 100. Various combinations of X and N were tried and the results compared with the table entries of program reference 2. The Bessel functions computed maintain an accuracy of at least 9 digits as compared with program reference 2.

It is expected that the routine can handle orders N > 100 and arguments X < 0.1 and X > 100; but available tables did not include these values.

References: (1) Langley Research Center, Computer Programming Manual; Volume I, Section C3.2, pages 2-5.

(2) I. Stegun and M. Abramowitz, Handbook of Mathematical Functions; U.S. Department of Commerce, National Bureau of Standards Applied Mathematics Series 55, June 1964, pp. 392, 394, and 407.

Storage: 117_8 locations.

Other coding information: (1) If high-order Bessel functions are computed for small arguments, an exponent underflow may result.

(2) The results $J_0(x)$, $J_1(x)$, . . ., $J_i(x)$ . . ., $J_{N-1}(x)$ are stored as follows.

BJ(1) = $J_0(x)$
BJ(2) = $J_1(x)$

.
.
.

BJ(I) = $J_{I-1}(x)$

.
.
.

BJ(N) = $J_{N-1}(x)$

Subprograms used: None.

FORTRAN functions: None.

Source: NASA Langley Research Center, Noise Control Branch, Robert Hosier.

APPENDIX F

PROGRAM USAGE

Operational Environment

The five programs making up the Stationary Rotor Rotational Noise Prediction Program are written in FORTRAN. All program testing and case running were done on the CDC 6400 and CDC 6600 computer systems. The CDC 6400 and CDC 6600 machines have 131,900 word memories and use 60 bit words.

The operating system is the Langley Research Center (LRC) SCOPE Operational System versions 3.3. A RUN 2.3 Compiler is employed by the operating system.

Program Job-Stepping

In order to reduce program overhead and to increase program efficiency, the Stationary Rotor Rotational Noise Prediction Package is divided into five programs. The five programs are related, but are independent logical steps in predicting the rotor noise.

The programs are independent in the sense that all five perform a function without "calling" any of the others; the programs are related in the sense that a file may be passed from one program to another. As an example, the output file (the combined transducer spectrum data) of program RAMANI can be used as an input file to program TRANS. This technique of program communication through files is called job-stepping and is accomplished through the use of job control cards.

Running the five program package requires the use of certain peripheral equipment including a card reader, a line printer, a magnetic tape drive, disk storage drives, a data cell drive, and a plotter. The specific types of peripheral equipment utilized are described in table XVIII. The various allowable program job-step combinations together with the corresponding control card and program deck setups are described in the next section.

Job Control and Program Deck Setups

The five programs making up the Stationary Rotor Rotational Noise Prediction Program can be run in a variety of program combinations. Through the correct use of job control cards, the following combinations are possible:

RAMANI

SPECPLT

RAMANI-SPECPLT

RNPPE4

TRANS-RNPPE4

RNPPE4-SPLPLT

TRANS-RNPPE4-SPLPLT

RAMANI-RNPPE4

RAMANI-TRANS-RNPPE4

RAMANI-SPECPLT-RNPPE4

RAMANI-SPECPLT-TRANS-RNPPE4

RAMANI-TRANS-RNPPE4-SPLPLT

RAMANI-SPECPLT-TRANS-RNPPE4-SPLPLT

In choosing the appropriate combination(s) to be run, considerations should be given to the type of data (measured or theoretical), to the reliability of measured data, to specific output needs, and to the necessity of plotting.

The standard sequence is a two-combination process. The first combination involves job-stepping programs RAMANI and SPECPLT in a separate run to check the validity of the combined transducer spectrum data before proceeding to the actual rotor noise prediction. The second combination involves job-stepping programs TRANS-RNPPE4-SPLPLT, a separate run to predict the rotor noise and to generate the sound-pressure-level plots. If the user is relatively confident in the accuracy of the measured transducer data, the one-step process of job-stepping programs RAMANI-SPECPLT-TRANS-RNPPE4-SPLPLT into one run saves time and reduces program overhead.

This section contains job control and program deck setups for the three major program combinations mentioned above. The file descriptions and manipulations are discussed later.

In order to eliminate the handling of card source decks, the five programs have been placed on two data cells. Programs RAMANI and SPECPLT (binary and source versions) are stored on data cell Z4186, and programs TRANS, RNPPE4, and SPLPLT are stored on data cell Z4188. The job control cards and program setups for both card and data cell input for the combinations RAMANI-SPECPLT, TRANS-RNPPE4-SPLPLT, and RAMANI-SPECPLT-TRANS-RNPPE4-SPLPLT are explained in tables XIX to XXIV. Although file descriptions and manipulations are discussed later, certain general observations about the preceding job control card setups should be made.

A program source listing, together with corresponding load and cross reference maps, are generated for all combinations run from the source versions. Neither program listings nor corresponding maps are generated for program combinations run when using binary versions.

If the user desires the source version instead of the binary version when running from data cell, the following changes must be made (see tables XX, XXII, and XXIII):

1. The four control cards

```
REWIND (BN FILE)
FETCH (Z418_,,SOURCE)
RUN (S,,,SFILE, BN FILE)
REWIND (BNFILE)
```

replaced the previous card

```
FETCH (Z418_,,BINARY).
```

2. Following the first 7/8/9 card the user must place one record containing program modifications, if any, for each source program fetched. Program modifications are discussed in reference 6.

File Description and Manipulation

Magnetic tape assignments (input – output files).- Tape 8 is the assigned name of the input tape to RAMANI; the tape contains the transducer spectrum data for a given run. At present, this tape is generated by a fast Fourier Transform Program. Tape 51 is the assigned name of the output tape containing the combined transducer spectrum data. This tape is generated by program RAMANI. This same tape is assigned name TAPE 8 when used as input to program TRANS. The plotting tapes generated by programs SPECPLT and SPLPLT are assigned file names (PLT___) by the plotting postprocessor and not by the user.

Other program files.- The names RAMANI, SPECPLT, TRANS, RNPPE4, SPLPLT are the binary file names of the programs with corresponding names. Tape 20 is the name assigned to the disk file being used as a random access file. Tape 7 and Tape 4 are the disk files corresponding to the magnetic tape input and output files, Tape 8 and Tape 51, respectively. PLOT is the binary version of the postprocessor program necessary in generating a plot tape.

File manipulation.- The scope operating system is a file-oriented system. Because of this fact, the user must have a general knowledge of the file manipulation. All files used in the execution of the SRRNPP are binary files. The file-manipulation process is similar regardless of the program combination being run and regardless of whether the source deck is loaded from cards or whether the binary version is loaded from data cell. The first step is to store the binary versions of all programs being considered on separate files corresponding to their program names. (This step assumes compilation is successful, if source decks are used.) The second step consists of requesting the input file from Tape 8; copying the input file to disk (Tape 7), dropping the magnetic tape, and using Tape 7 as to the actual input file. At this point, the binary version of the program is loaded and executed. A disk output file (Tape 4) is created; if necessary, the output file is copied to a magnetic tape (Tape 51) for future use. The first job-step is now completed.

If another job-step is to be executed, the output file (Tape 4) can be rewound and used as an input file to the next job-step. The random access file (Tape 20) can be used in a similar manner with no rewind being necessary. This program communication through input and output files is the essence of the job-stepping techniques.

If one of the two plotting programs is to be run, the binary version of the graphic software is loaded by the system on the same file as the program binary before execution. Once this move is accomplished, the plotting program is executed. Before the actual plot tape is created, the graphic output postprocessor program (plot) must be loaded and executed. Appendix C describes the plotting techniques in more detail. The process can continue or terminate at any time depending on the number of job-steps to be executed.

The preceding description of file manipulation, although general in nature, describes the technique used in the running and execution of all program combinations of the SRRNPP.

Program Performance and Memory Requirements

Estimates of central processor time, peripheral processor time, and operating system calls for the three major program combinations on a typical problem are approximated in table XXV.

The field length required for the execution of each of the five programs follows:

Program	Required field length (Octal)
RAMANI	43625
SPECPLT	41211
TRANS	44525
RNPPE4	44603
SPLPLT	24651

If program job-stepping is employed, any combination of two or more programs can be run in a single job at a required field length of less than 50 K (Octal). It is important to note that the automatic field-length reduction performed at load time by the operating system must be suppressed by the job control card NORFL if job-stepping is employed.

APPENDIX F

Performance times and statistics vary significantly on the problem being run. The number of transducers, the number of spectrum points per station, the method for determining the blade-loading (BLH) data, and the amount of plotting generated are the major factors determining program performance. The following times and statistics are based on the following problem. Transducer data are input from 7 transducers with 1840 spectrum-data points per transducer. The spectrum-data combining results in 4 combined positions generated by 3 additions and 1 scaling. Four pairs of combined amplitude and phase plots are generated. Seven sets of sound pressure level (SPL) data are computed based on 5 sets of nonintegrated BLH data and 2 sets of integrated BLH data. Seven SPL plots are generated and two pairs of integrated BLH coefficients and phases are generated.

The figures shown in table XXV are based on the binary version of all programs being loaded for a data cell. If the source programs are loaded from card decks, the CPU time slightly increases because of compile time, the PPU time significantly decreases because of the absence of the data cell, and operating system (OS) calls remain nearly constant.

All figures are based on the CDC 6600 computer system.

APPENDIX G

PROGRAM LISTING

```
      PROGRAM RAMANI (INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE7=3001,TAPE4=3001,TAPE20=201)
C***********************************************************************
C*             PURPOSE                                                 *
C*                TO COMBINE (BY ADDITION OR SCALING) THE SPECTRUM     *
C*                DATA OBTAINED FROM THE TRANSDUCERS PLACED AT THE     *
C*                ≠NTRACKS≠ POSITIONS ALONG THE CHORD                  *
C*                                                                     *
C*             NAMELIST INPUT PARAMETERS                               *
C*                NTRACKS - NUMBER OF TRACKS OF TRANSDUCER SPECTRUM DATA*
C*                          (1≤NTRACKS≤24)                             *
C*                DBREF   - AN ARRAY DIMENSIONED 24 (FOR THE MAXIMUM   *
C*                          NUMBER OF TRANSDUCER STATIONS) CONTAINIG THE*
C*                          DECIBEL REFERENCE FREQUENCY CONSTANTS      *
C*                          (DBREF(I)=0.0 IS DEFAULT VALUE  1≤I≤24)    *
C*                NPRINT  - NUMBER OF COMBINED SPECTRUM DATA RECORDS   *
C*                          (FREQUENCY,AMPLITUDE,PHASE) TO BE PRINTED  *
C*                          FOR EACH POSITION                          *
C*                          (NPRINT=50 IS DEFAULT VALUE)               *
C*                NSKIP   - NORMALLY INPUT AS ZERO. IF SPECTRUM DATA   *
C*                          FROM BEGINNING STATIONS IS  TO BE SKIPPED, *
C*                          (THERE ARE 2*((NUMBER OF SPECTRUM DATA POINTS
C*                          PER STATION - 1) / 1500 + 1) + 1 TAPE RECORDS
C*                          PER TRACK)                                 *
C*                          (NSKIP=0 IS DEFAULT VALUE)                 *
C*                MTRACKS - NUMBER OF SPECTRUM DATA STATIONS AFTER     *
C*                          COMBINING (ADDITION OR SCALING) PLUS THE   *
C*                          NUMBER OF STATIONS TO BE SKIPPED.          *
C*                          (1≤MTRACKS≤24)                             *
C*                ISTN,   - TWO INPUT ARRAYS DIMENSIONED 24 (FOR THE   *
C*                JSTN      MAXIMUM NUMBER OF TRANSDUCER STATIONS)     *
C*                          DESCRIBING THE COMBINING SEQUENCE.         *
C*                            (1) ISTN(K)≠JSTN(K) (JSTN(K)≠0)  1≤K≤24  *
C*                            SPECTRUM DATA FROM TRANSDUCER TRACK NUMBERS*
C*                            ISTN(K) AND JSTN(K) WILL BE ADDED.       *
C*                             (2) ISTN(K)=JSTN(K) 1≤K≤24              *
C*                            SPECTRUM DATA FROM TRANSDUCER TRACK NUMBER *
C*                            ISTN(K) WILL BE SCALED BY THE FACTOR WGHT. *
C*                             (3) ISTN(K)≠0 AND JSTN(K)=0  1≤K≤24     *
C*                            SPECTRUM DATA FROM TRANSDUCER TRACK NUMBER *
C*                            ISTN(K) WILL BE SKIPPED.                 *
C*                          RESTRICTIONS: ISTN ARRAY MUST BE IN STRICTLY*
C*                            INCREASING ORDER AND ISTN(I)≥JSTN(J) FOR *
C*                            I≥J.                                     *
C*                          (ISTN(K)=0 IS DEFAULT VALUE 1≤K≤24)        *
C*                          (JSTN(K)=0 IS DEFAULT VALUE 1≤K≤24)        *
C*                WGHT    - A SCALING FACTOR BY WHICH SPECTRUM DATA FROM*
C*                          AN UNADDED STATION WILL BE WEIGHTED        *
C*                          IF WGHT=1.0, THE DATA WILL BE LEFT AS IS   *
C*                          (WGHT=1.0 IS DEFAULT VALUE)                *
C*                IOPTN   -  0 COMBINED SPECTRUM DATA WILL BE OUTPUT   *
C*                              TO MAGNETIC TAPE ONLY.                 *
C*                           1 COMBINED SPECTRUM DATA WILL BE OUTPUT   *
C*                              TO RANDOM ACCESS FOR LATER USE AS      *
C*                              WELL AS MAGNETIC TAPE.                 *
C*                          (IOPTN=1 IS DEFAULT VALUE)                 *
C*                NTAPE   -  1 ONE TAPE IS USED TO INPUT SPECTRUM DATA *
C*                           2 TWO TAPES ARE USED TO INPUT SPECTRUM DATA*
C*                          (NTAPE=1 IS DEFAULT VALUE)                 *
C*                LU      - AN ARRAY DIMENSIONED 24 (FOR THE MAXIMUM   *
C*                          NUMBER OF TRANSDUCER STATIONS) CONTAINING  *
C*                          THE STEADY LOADING FREQUENCY FOR EACH TRANS-*
```

```
C*                         DUCER STATION (PSI)                         *RAMA0630
C*                         (LO(I)=0.0 IS DEFAULT VALUE 1≤I≤24)         *RAMA0640
C*                                                                     *RAMA0650
C*            TAPE INPUT PARAMETERS                                    *RAMA0660
C*               ICH      - TRANSDUCER STATION NUMBER                  *RAMA0670
C*                          (1≤ICH≤24)                                 *RAMA0680
C*               DELTF    - BLADE LOADING FREQUENCY                    *RAMA0690
C*               NSPCT    - NUMBER OF SPECTRUM DATA POINTS PER         *RAMA0700
C*                          TRANSDUCER STATION                         *RAMA0710
C*               AMP      - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-*RAMA0720
C*                          ING THE SPECTRUM DATA AMPLITUDE            *RAMA0730
C*               PHASE    - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-*RAMA0740
C*                          ING THE SPECTRUM DATA PHASE                *RAMA0750
C*                                                                     *RAMA0760
C*            SUBROUTINES USED                                         *RAMA0770
C*               NONE                                                  *RAMA0780
C*                                                                     *RAMA0790
C***********************************************************************RAMA0800
C                                                                       RAMA0810
      COMMON AMP(1500,2),PHASE(1500,2),NUAMP(1500),NUPHASE(1500)        RAMA0820
      DIMENSION DBREF(24),ISTN(24),JSTN(24),INDEX(987),REC1(4),OASPL(24)RAMA0830
     +,LO(24)                                                           RAMA0840
C                                                                       RAMA0850
      REAL IM,NUPHASE,NUAMP,LO                                          RAMA0860
C                                                                       RAMA0870
      NAMELIST /INPUT/ NTRACKS,DBREF,NPRINT,NSKIP,ISTN,JSTN,MTRACKS,WGHTRAMA0880
     +,IOPTN,NTAPE,LO                                                   RAMA0890
      DATA RADIAN/57.2957795/                                           RAMA0900
C                                                                       RAMA0910
C           FUNCTIONS USED IN THE COMPLEX ADDITION OF THE TRANSDUCER    RAMA0920
C           SPECTRUM DATA                                               RAMA0930
      RE(I,J)=AMP(I,J)*COS(PHASE(I,J)/RADIAN)                           RAMA0940
      IM(I,J)=AMP(I,J)*SIN(PHASE(I,J)/RADIAN)                           RAMA0950
      P(I,J,K,L)=(AMP(I,J)**2)+(AMP(K,L)**2)                            RAMA0960
      PP(I,J,K,L)=2.*(AMP(I,J)*AMP(K,L))*                               RAMA0970
     1(COS((PHASE(I,J)-PHASE(K,L))/RADIAN))                             RAMA0980
C                                                                       RAMA0990
C           OPEN RANDOM ACCESS FILE                                     RAMA1000
      CALL OPENMS (20,INDEX,987,0)                                      RAMA1010
C                                                                       RAMA1020
C           INPUT NAMELIST DEFAULT PARAMETER VALUES                     RAMA1030
      NPRINT = 50                                                       RAMA1040
      NSKIP = 0                                                         RAMA1050
      WGHT = 1.0                                                        RAMA1060
      IOPTN = 1                                                         RAMA1070
      NTAPE = 1                                                         RAMA1080
      DO 5 I=1,24                                                       RAMA1090
      DBREF(I) = 0.0                                                    RAMA1100
      LO(I) = 0.0                                                       RAMA1110
      ISTN(I) = 0                                                       RAMA1120
      JSTN(I) = 0                                                       RAMA1130
    5 CONTINUE                                                          RAMA1140
C                                                                       RAMA1150
C           READ NAMELIST INPUT, CHECK FOR END OF FILE, AND OUTPUT      RAMA1160
C           NAMELIST INPUT TO PRINTER                                   RAMA1170
   10 READ (5,INPUT)                                                    RAMA1180
      IF (EOF,5) 999,20                                                 RAMA1190
   20 WRITE (6,INPUT)                                                   RAMA1200
C                                                                       RAMA1210
C           SKIP FIRST NSKIP RECORDS OF INPUT TAPE                      RAMA1220
```

```
      IF (NSKIP .EQ. 0) GO TO 40                                        RAMA1230  12300000
      DO 30 I=1,NSKIP                                                   RAMA1240  12400000
C               MAKE EOF CHECK AND DETERMINE IF INPUT SPECTRUM DATA     RAMA1250  12500000
C               COMES FROM 1 OR 2 TAPES                                 RAMA1260  12600000
      READ (7)                                                          RAMA1270  12700000
      IF (EOF,7) 25,30                                                  RAMA1280  12800000
   25 IF (NTAPE .NE. 1) GO TO 30                                        RAMA1290  12900000
      WRITE (6,2040) NSKIP                                              RAMA1300  13000000
 2040 FORMAT (///,40X,*THE NUMBER OF SPECTRUM DATA RECORDS TO BE SKIPPEDRAMA1310  13100000
     1 *,I5,/,30X,*IS GREATER THAN THE NUMBER OF RECORDS OF SPECTRUM DATRAMA1320  13200000
     2A ON INPUT TAPE*)                                                 RAMA1330  13300000
      GO TO 999                                                         RAMA1340  13400000
   30 CONTINUE                                                          RAMA1350  13500000
C                                                                       RAMA1360  13600000
C               INITIALIZE COUNTERS FOR NUMBER OF TRANSDUCER STATIONS,  RAMA1370  13700000
C               RECORD SIZE OF SPECTRUM DATA BLOCKS, AND THE NUMBER     RAMA1380  13800000
C               OF SPECTRUM DATA BLOCKS PER STATION                     RAMA1390  13900000
   40 NSTN = 1                                                          RAMA1400  14000000
   50 NPOIN = 1500                                                      RAMA1410  14100000
      NR = 1                                                            RAMA1420  14200000
C                                                                       RAMA1430  14300000
C               READ TRANSDUCER STATION IDENTIFICATION RECORD FROM INPUT RAMA1440  14400000
C               TAPE AND USE IT FOR SAME PURPOSE ON RANDOM ACCESS FILE  RAMA1450  14500000
      READ (7) ICH,DELTF,NSPCT                                          RAMA1460  14600000
C                                                                       RAMA1470  14700000
C               MAKE EOF CHECK AND DETERMINE IF INPUT SPECTRUM DATA     RAMA1480  14800000
C               COMES FROM 1 OR 2 TAPES                                 RAMA1490  14900000
      IF (EOF,7) 53,57                                                  RAMA1500  15000000
   53 IF (NTAPE .NE. 1) GO TO 55                                        RAMA1510  15100000
      WRITE (6,2050) NTRACKS                                            RAMA1520  15200000
 2050 FORMAT (///,40X,*THE NUMBER OF TRACKS OF SPECTRUM DATA TO BE COMBIRAMA1530  15300000
     1NED *,I2,/,30X,*IS GREATER THAN THE NUMBER OF TRACKS OF SPECTRUM DRAMA1540  15400000
     2ATA ON INPUT TAPE*)                                               RAMA1550  15500000
      GO TO 999                                                         RAMA1560  15600000
   55 READ (7) ICH,DELTF,NSPCT                                          RAMA1570  15700000
C                                                                       RAMA1580  15800000
   57 REC1(1) = FLOAT(ICH)                                              RAMA1590  15900000
      REC1(2) = DELTF                                                   RAMA1600  16000000
      REC1(3) = FLOAT(NSPCT)                                            RAMA1610  16100000
C                                                                       RAMA1620  16200000
C               BY DETERMINING THE NUMBER OF SPECTRUM DATA BLOCKS PER   RAMA1630  16300000
C               STATION COMPUTE BEGINNING RANDOM ACCESS RECORD LOCATION RAMA1640  16400000
      NREAD = (NSPCT-1)/1500 + 1                                        RAMA1650  16500000
      NRPTS = 2*NREAD + 1                                               RAMA1660  16600000
      NREC = (ICH-1)*NRPTS + 1                                          RAMA1670  16700000
      CALL WRITMS (20,REC1,3,NREC)                                      RAMA1680  16800000
      NREC = NREC+1                                                     RAMA1690  16900000
C                                                                       RAMA1700  17000000
C               DETERMINE RECORD SIZE OF DATA BLOCK NREAD, READ SPECTRUM RAMA1710  17100000
C               DATA BLOCK FROM INPUT TAPE, AND STORE DATA BLOCK ON     RAMA1720  17200000
C               RANDOM ACCESS FILE IN TWO RECORDS -AMPLITUDE + PHASE.   RAMA1730  17300000
   60 NTOTL = NR*1500                                                   RAMA1740  17400000
      IF (NTOTL .GT. NSPCT) NPOIN = NSPCT-NTOTL+1500                    RAMA1750  17500000
      READ (7) (AMP(I,1),PHASE(I,1),I=1,NPOIN)                          RAMA1760  17600000
      CALL WRITMS (20,AMP(1,1),NPOIN,NREC)                              RAMA1770  17700000
      NREC = NREC+1                                                     RAMA1780  17800000
      CALL WRITMS (20,PHASE(1,1),NPOIN,NREC)                            RAMA1790  17900000
      NREC = NREC+1                                                     RAMA1800  18000000
C                                                                       RAMA1810  18100000
C               DETERMINE IF LAST DATA BLOCK FOR THIS STATION HAS BEEN  RAMA1820  18200000
C               READ AND STORED                                         RAMA1830  18300000
```

```
      IF (NPOIN .NE. 1500) GO TO 70
      NR = NR+1
      GO TO 60
C
C               DETERMINE IF SPECTRUM DATA HAS BEEN INPUT AND STORED FOR
C               ALL TRANSDUCER STATIONS
   70 NSTN = NSTN+1
      IF (NSTN .LE. NTRACKS) GO TO 50
C
C               SET POINTER FRO THE RECORD SIZE OF THE LAST SPECTRUM DATA
C               BLOCK FOR EACH STATION
      MPOIN = NPOIN
C               INITIALIZE STEADY LOADING AND OASPL COUNTER
      NNS = 0
C               INITIALIZE COMBINED SPECTRUM DATA RECORD REWRITE COUNTER
      NREC = 1
C
C               LOOP TO REDUCE THE NUMBER OF TRANSDUCER STATIONS (BY
C               ADDITION) FROM NTRACKS TO MTRACKS
C
      DO 160 NS=1,MTRACKS
C
C
C               DETERMINE IF DATA FROM TRANSDUCER TRACK ISTN(NS) IS TO BE
C               IGNORED
      IF (JSTN(NS) .EQ. 0) GO TO 160
C
C               INITIALIZE COUNTERS (IREC IS ORIGINALLY SET TO TRANSDUCER
C               IDENTIFICATION RECORD FOR STATION NS - SIMILAR DEFINITION
C               FOR JREC AND NREC).
      SUM = 0.0
      II = 1
      IREC = (ISTN(NS)-1)*NRPTS + 1
C
C               READ STATION IDENTIFICATION RECORD, RESTORE THIS RECORD
C               ON RANDOM ACCESS IF DESIRED, AND READ SECOND STATION
C               IDENTIFICATION RECORD.
      CALL READMS (20,REC1,3,IREC)
      IF (IOPTN .EQ. 0) GO TO 80
      REC1(4) = FLOAT(NREAD)
      CALL WRITMS(20,REC1,4,NREC)
      NREC = NREC+1
   80 IF (ISTN(NS) .EQ. JSTN(NS)) GO TO 90
      JREC = (JSTN(NS)-1)*NRPTS + 1
      CALL READMS (20,REC1,3,JREC)
C
C               OUTPUT POSITION HEADING INFORMATION
   90 IF (ISTN(NS) .NE. JSTN(NS)) WRITE(6,2000) NS,ISTN(NS),JSTN(NS)
 2000 FORMAT (1H1,//,40X,*COMBINING OF TRANSDUCER STATION SPECTRUM DATA*
     1,//,50X,*FINAL VALUES FOR POSITION *,I2,/,35X,*(SPECTRUM DATA FROM
     2 STATIONS *,I2,* AND *,I2,* HAS BEEN ADDED)*,//)
      IF (ISTN(NS) .EQ. JSTN(NS)) WRITE (6,2030) NS,ISTN(NS),WGHT
 2030 FORMAT (1H1,//,40X,*COMBINING OF TRANSDUCER STATION SPECTRUM DATA*
     1,//,50X,*FINAL VALUES FOR POSITION *,I2,/,35X,*(SPECTRUM DATA FROM
     2 STATION *,I2,* HAS BEEN SCALED BY THE FACTOR*,G16.8,*)*,//)
      WRITE (6,2020)
 2020 FORMAT (//,24X,*FREQUENCY*,10X,*AMPLITUDE*,12X,*PHASE*,/)
C
C               WRITE POSITION IDENTIFICATION RECORD ON OUTPUT TAPE
      WRITE (4) NS,DELTF,NSPCT,NREAD
```

```
C
C
C           LOOP TO REREAD TRANSDUCER SPECTRUM DATA FROM RANDOM ACCESS
C           FILE IN PREPARATION FOR ADDITION, CREATE OUTPUT TAPE, AND
C           TO RESTORE COMBINED SPECTRUM DATA TO RANDOM ACCESS.
C
      DO 150 NR=1,NREAD
C
C
      NPOIN = 1500
      IF (NR .EQ. NREAD) NPOIN=MPOIN
C
      IREC = IREC+1
      CALL READMS (20,AMP(1,1),NPOIN,IREC)
      IREC = IREC+1
      CALL READMS (20,PHASE(1,1),NPOIN,IREC)
      IF (ISTN(NS) .EQ. JSTN(NS)) GO TO 100
      JREC = JREC+1
      CALL READMS (20,AMP(1,2),NPOIN,JREC)
      JREC = JREC+1
      CALL READMS (20,PHASE(1,2),NPOIN,JREC)
C
C
C           LOOP TO ADD SPECTRUM DATA FOR THE APPROPRIATE STATIONS OR
C           TO SCALE UNADDED DATA, COMPUTE FREQUENCY AND TO OUTPUT
C           NPRINT VALUES TO PRINTER FOR EACH POSITION
C
  100 DO 140 I=1,NPOIN
C
C
C           DETERMINE IF TRANSDUCER DATA IS TO BE ADDED OR SCALED
      IF (ISTN(NS) .EQ. JSTN(NS)) GO TO 110
C
C           COMPLEX ADDITION OF SPECTRA (ISTN+JSTN)
      P1=P(I,1,I,2)+PP(I,1,I,2)
      NUAMP(I)=SQRT(P1)
      NUPHASE(I)=ATAN2((IM(I,1)+IM(I,2)),(RE(I,1)+RE(I,2)))
      NUPHASE(I)=NUPHASE(I)*RADIAN
      GO TO 120
C
C           WEIGHT UNCOMBINED SPECTRA
  110 NUAMP(I) = WGHT * AMP(I,1)
      NUPHASE(I) = PHASE(I,1)
C
C           COMPUTE FREQUENCY
  120 F = (I-1) * DELTF
C
C           OUTPUT NPRINT VALUES TO PRINTER
      IF (II .GT. NPRINT) GO TO 130
      WRITE (6,2010) II,F,NUAMP(I),NUPHASE(I)
 2010 FORMAT (10X,I5,3(5X,E14.5))
      II = II+1
C
  130 SUM = SUM + ABS(NUAMP(I))**2
  140 CONTINUE
C
C           WRITE POSITION DATA BLOCKS TO OUTPUT TAPE
      WRITE (4) (NUAMP(I),NUPHASE(I),I=1,NPOIN)
C
C           DETERMINE IF POSITION DATA BLOCKS ARE TO BE RESTORED ON
C           RANDOM ACCESS
      IF (IOPTN .EQ. 0) GO TO 150
      CALL WRITMS (20,NUAMP(1),NPOIN,NREC)
      NREC = NREC+1
      CALL WRITMS (20,NUPHASE(1),NPOIN,NREC)
      NREC = NREC+1
C
  150 CONTINUE
C
C             COMPUTE POSITION STEADY LOADING FACTORS, COMPUTE OASPL
C             ARRAY, AND OUTPUT TO TAPE
      NNS = NNS+1
      LO1 = ISTN(NS)
      LO2 = JSTN(NS)
      IF (LO1 .EQ. LO2) LO(NNS) = ABS (LO(LO1))
      IF (LO1 .NE. LO2) LO(NNS) = ABS (LO(LO1)-LO(LO2))
      RMS = SQRT (SUM*DELTF)
      OASPL(NNS) = 20. * ALOG10(RMS/DBREF(NS))
      WRITE (4) LO(NNS),OASPL(NNS)
C
  160 CONTINUE
C
C             IF RANDOM ACCESS RESTORAGE OF COMBINED SPECTRA IS DESIRED
C             WRITE STEADY LOADING ARRAY AND OASPL ARRAY TO FILE
      IF (IOPTN .EQ. 0) GO TO 333
      CALL WRITMS(20,LO(1),MTRACKS,NREC)
      NREC = NREC+1
      CALL WRITMS(20,OASPL(1),MTRACKS,NREC)
C
  333 GO TO 10
C
  999 CONTINUE
      STOP
      END
      PROGRAM SPECPLT (INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE4=3001,TAPE20=201)
C***********************************************************************
C*                                                                     *
C*            PURPOSE                                                  *
C*               PLOT (CALCOMP PLOT) THE COMBINED TRANSDUCER SPECTRUM  *
C*               DATA GENERATED BY PROGRAM RAMANI.  ONE AMPLITUDE      *
C*               SPECTRUM AND ONE PHASE SPECTRUM IS PRODUCED FOR       *
C*               EACH OF THE COMBINED (ADDED OR SCALED) POSITIONS.     *
C*               THE PLOTS ARE CALCOMP PLOTS OF AMPLITUDE (OR PHASE)   *
C*               .VS. FREQUENCY.                                       *
C*                                                                     *
C*            NAMELIST INPUT PARAMETERS                                *
C*               MTRACKS - NUMBER OF SPECTRUM DATA POSITIONS AFTER     *
C*                         COMBINING (1≤MTRACKS≤14)                    *
C*               THRUST  - TOTAL LIFT OF HELICOPTER (LBS.)             *
C*               RPM     - ROTATIONAL SPEED OF ROTOR (RPM)             *
C*               XMIN    - (XMAX-XMIN) IS ALLOWABLE FREQUENCY RANGE    *
C*                         (XMIN=0.0 IS DEFAULT VALUE)                 *
C*               XMAX      FOR PLOTTING THE SPECTRUM DATA (HZ.)        *
C*                         (XMAX=1000.0 IS DEFAULT VALUE)              *
C*               RUN     - RUN NUMBER                                  *
C*               YMIN    - (YMAX-YMIN) IS ALLOWABLE AMPLITUDE RANGE    *
C*                         (YMIN=-100.0 IS DEFAULT VALUE)              *
C*               YMAX      FOR PLOTTING THE SPECTRUM DATA              *
C*                         (YMAX=-0.0 IS DEFAULT VALUE)                *
C*               YSCALE  - AMPLITUDE SCALE FACTOR                      *
C*                         (YSCALE=10.0 IS DEFAUL VALUE)               *
C*               NNPLOT  - AN ARRAY DIMENSIONED 14 (FOR THE MAXIMUM    *
C*                         NUMBER OF TRANSDUCER STATIONS) DETERMINING  *
C*                         WHETHER A SPECTRUM PLOT IS DESIRED FOR      *
C*                         A GIVEN STATION                             *
C*                         0 NO SPECTRUM DATA PLOTS ARE GENERATED FOR  *
C*                           ITH POSITION  I=1,2,...,MTRACKS           *
C*                         1 AN AMPLITUDE SPECTRUM PLOT IS GENERATED   *
C*                           FOR THE ITH POSITION  I=1,2,...,MTRACKS   *
C*                         2 BOTH AMPLITUDE AND PHASE SPECTRUM PLOTS   *
C*                           WILL BE PRODUCED FOR THE ITH POSITION     *
C*               IOPTN   - 0 THIS PROGRAM IS NOT RUN IMMEDIATELY       *
C*                           FOLLOWING PROGRAM RAMANI. THEREFORE,      *
C*                           COMBINED SPECTRUM DATA IS INPUT FROM      *
C*                           MAGNETIC TAPE.                            *
```

```
C*                     1 THIS PROGRAM IS RUN IMMEDIATELY FOLLOWING *
C*                       (JOB-STEPPED WITH) PROGRAM RAMANI.        *
C*                       THEREFORE, COMBINED SPECTRUM DATA IS INPUT*
C*                       FROM RANDOM ACCESS FILE.                  *
C*                     (IOPTN=1 IS DEFAULT VALUE)                  *
C*                                                                 *
C*           MAG TAPE OR RANDOM ACCESS FILE INPUT                  *
C*              ICH      - SPECTRUM DATA POSITION NUMBER           *
C*                         (1≤ICH≤14)                              *
C*              DELTF    - BLADE LOADING FREQUENCY                 *
C*              NSPCT    - NUMBER OF SPECTRUM DATA POINTS PER TRANS-*
C*                           DUCER (COMBINED) STATION              *
C*              AMP      - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-*
C*                         ING THE COMBINED SPECTRUM DATA AMPLITUDE. *
C*              PHASE    - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-*
C*                         ING THE COMBINED SPECTRUM DATA PHASE    *
C*                                                                 *
C*           SUBROUTINE USED                                       *
C*              NONE                                               *
C*****************************************************************
C
      COMMON NUAMP(1500),NUPHASE(1500),XPLOT(1500),RI(1500)
      DIMENSION DBREF(14),LO(14),DASPL(14),NNPLOT(14),PAR1(2),PAR2(2),
     +PAR3(2),PAR4(2),PAR5(2),PAR6(2),PAR7(2),INDEX(987),REC1(4)
C
      REAL LO,NUAMP,NUPHASE
C
      NAMELIST /INPUT/ MTRACKS,THRUST,RPM,XMIN,XMAX,RUN,YMIN,YMAX,YSCALE
     +,NNPLOT,DBREF,IOPTN
C
C              OPEN RANDOM ACCESS FILE AND CALL CALCOMP PROCESSOR
      CALL OPENMS (20,INDEX,987,0)
      CALL PSEUDO (6LSPCTRA)
      CALL LEROY
C
C              INPUT NAMELIST DEFAULT PARAMETER VALUES
      XMIN = 0.0
      XMAX = 1000.0
      YMIN = -100.0
      YMAX = -0.0
      YSCALE = 10.0
      IOPTN = 1
      DO 5  I=1,14
      NNPLOT(I) = 0
      DBREF(I) = 0.0
    5 CONTINUE
C
C              READ NAMELIST INPUT, CHECK FOR END OF FILE, AND OUTPUT
C              NAMELIST TO PRINTER
   10 READ(5,INPUT)
      IF (EOF,5) 999,30
   30 WRITE(6,INPUT)
C
C              TEST TO DETERMINE IF COMBINED TRANSDUCER SPECTRUM DATA IS
C              TO BE INPUT FROM RANDOM ACCESS FILE OR PROGRAM RAMANI
C              OUTPUT TAPE
      IF (IOPTN .NE. 0) GO TO 70
C
C
C              INITIALIZE RANDOM ACCESS RECORD COUNTER FOR SPECTRUM DATA
      NREC=1
C
C               *LOOP* TO INPUT COMBINED TRANSDUCER SPECTRUM DATA FROM
C              RAMANI OUTPUT TAPE AND RESTORE THIS SPECTRUM DATA ON THE
C              RANDOM ACCESS FILE
      DO 50 NS=1,MTRACKS
C
C              READ TRANSDUCER STATION IDENTIFICATION RECORD (AND RESTORE
C              ID. RECORD ON RANDOM ACCESS)
      READ (4) ICH,DELTF,NSPCT,NREAD
      REC1(1) = FLOAT (ICH)
      REC1(2) = DELTF
      REC1(3) = FLOAT (NSPCT)
      REC1(4) = FLOAT (NREAD)                                           SPEC1150  45300000
      CALL WRITMS (20,REC1,4,NREC)                                      SPEC1160  45400000
      NREC = NREC+1                                                     SPEC1170  45500000
C                                                                       SPEC1180  45600000
C             DETERMINE APPROPRIATE RECORD SIZE OF AMPLITUDE (PHASE)    SPEC1190  45700000
C             DATA BLOCKS                                               SPEC1200  45800000
      NPOIN = 1500                                                      SPEC1210  45900000
      MPOIN = NSPCT - (NREAD-1)*1500                                    SPEC1220  46000000
C                                                                       SPEC1230  46100000
C             READ SPECTRUM DATA BLOCKS FROM TAPE (AND RESTORE DATA     SPEC1240  46200000
C             BLOCKS ON RANDOM ACCESS                                   SPEC1250  46300000
      DO 40 NR=1,NREAD                                                  SPEC1260  46400000
C                                                                       SPEC1270  46500000
      IF (NR .EQ. NREAD) NPOIN=MPOIN                                    SPEC1280  46600000
      READ (4) (NUAMP(I),NUPHASE(I),I=1,NPOIN)                          SPEC1290  46700000
      CALL WRITMS (20,NUAMP(1),NPOIN,NREC)                              SPEC1300  46800000
      NREC = NREC+1                                                     SPEC1310  46900000
      CALL WRITMS (20,NUPHASE(1),NPOIN,NREC)                            SPEC1320  47000000
      NREC = NREC+1                                                     SPEC1330  47100000
C                                                                       SPEC1340  47200000
   40 CONTINUE                                                          SPEC1350  47300000
C                                                                       SPEC1360  47400000
C             READ STEADY LOADING FACTOR AND OASPL TO BE USED IN SPECTRUSPEC1370  47500000
C             PLOT LABELING                                             SPEC1380  47600000
      READ (4) LO(NS),OASPL(NS)                                         SPEC1390  47700000
C                                                                       SPEC1400  47800000
   50 CONTINUE                                                          SPEC1410  47900000
C                                                                       SPEC1420  48000000
C             RESTORE POSITION STEADY LOADING FACTORS AND POSITION      SPEC1430  48100000
C             OASPL FACTORS ON RANDOM ACCESS                            SPEC1440  48200000
      CALL WRITMS (20,LO(1),MTRACKS,NREC)                               SPEC1450  48300000
      NREC  = NREC+1                                                    SPEC1460  48400000
      CALL WRITMS (20,OASPL(1),MTRACKS,NREC)                            SPEC1470  48500000
C                                                                       SPEC1480  48600000
      GO TO 80                                                          SPEC1490  48700000
C                                                                       SPEC1500  48800000
C             SPECTRUM DATA IS TO BE INPUT DIRECTLY FROM RANDOM ACCESS  SPEC1510  48900000
C             (READ STATION IDENTIFICATION RECORD, STEADY LOADING FACTORSPEC1520  49000000
C              RECORD, AND OASPL RECORD)                                SPEC1530  49100000
   70 NREC = 1                                                          SPEC1540  49200000
      CALL READMS (20,REC1,4,NREC)                                      SPEC1550  49300000
      DELTF = REC1(2)                                                   SPEC1560  49400000
      NSPCT = IFIX (REC1(3))                                            SPEC1570  49500000
      NREAD = IFIX (REC1(4))                                            SPEC1580  49600000
      NREC = MTRACKS*(2*NREAD+1) + 1                                    SPEC1590  49700000
      CALL READMS (20,LO(1),MTRACKS,NREC)                               SPEC1600  49800000
      NREC = NREC+1                                                     SPEC1610  49900000
      CALL READMS (20,OASPL(1),MTRACKS,NREC)                            SPEC1620  50000000
      MPOIN = NSPCT - (NREAD-1)*1500                                    SPEC1630  50100000
C                                                                       SPEC1640  50200000
C                                                                       SPEC1650  50300000
C             SECTION TO COMPUTE *BANDWIDTH* FOR TRANSDUCER STATION     SPEC1660  50400000
C             SPECTRUM PLOTS                                            SPEC167C  50500000
   80 DO 90 I=1,NSPCT                                                   SPEC1680  50600000
      IJ=I                                                              SPEC1690  50700000
      IF ((IJ-1)*DELTF .GE. XMIN) GO TO 100                             SPEC1700  50800000
   90 CONTINUE                                                          SPEC1710  50900000
  100 IJ1 = IJ                                                          SPEC1720  51000000
      DO 110 I=IJ1,NSPCT                                                SPEC1730  51100000
      IJ = NSPCT -I +IJ1                                                SPEC1740  51200000
      IF ((IJ-1)*DELTF .LE. XMAX) GO TO 120                             SPEC1750  51300000
  110 CONTINUE                                                          SPEC1760  51400000
  120 IJ2 = IJ                                                          SPEC1770  51500000
C                                                                       SPEC1780  51600000
      IF (IJ1 .LT. IJ2) GO TO 130                                       SPEC1790  51700000
      WRITE (6,2000)                                                    SPEC1800  51800000
 2000 FORMAT (1H1,//,10X,*BANDWIDTH FOR SPECTRUM PLOTS IS TOO NARROW - NSPEC1810  51900000
     +O COMBINED TRANSDUCER STATION SPECTRUM PLOTS ARE GENERATED*)      SPEC1820  52000000
      GO TO 333                                                         SPEC1830  52100000
C                                                                       SPEC1840  52200000
C             COMPUTE THE NUMBER OF POINTS TO BE PLOTTED                SPEC185C  52300000
  130 NPLOT = IJ2-IJ1+1                                                 SPEC1860  52400000
C                                                                       SPEC1870  52500000
```

```
C                                                                        SPEC1880  52600000
C             COMPUTE SCALE FACTORS, AND X-AXIS AND Y-AXIS LENGTHS       SPEC1890  52700000
      FRANGE = XMAX-XMIN                                                 SPEC1900  52800000
      IPOW10=ALOG10(FRANGE)                                              SPEC1910  52900000
      POW10 = 10.**IPOW10                                                SPEC1920  53000000
      IF (POW10 .GE. FRANGE) POW10 = POW10/10.                           SPEC1930  53100000
      IF (FRANGE/POW10 -10.) 190,180,180                                 SPEC1940  53200000
  180 XSCALE = POW10                                                     SPEC1950  53300000
      GO TO 240                                                          SPEC1960  53400000
  190 IF (FRANGE/POW10 - 5.) 210,200,200                                 SPEC1970  53500000
  200 XSCALE = POW10 / 2.                                                SPEC1980  53600000
      GO TO 240                                                          SPEC1990  53700000
  210 IF (FRANGE/POW10 - 2.) 230,220,220                                 SPEC2000  53800000
  220 XSCALE = POW10 / 5.                                                SPEC2010  53900000
      GO TO 240                                                          SPEC2020  54000000
  230 XSCALE = POW10 / 10.                                               SPEC2030  54100000
  240 IMIN = XMIN / XSCALE                                               SPEC2040  54200000
      IF(IMIN*XSCALE.GT.XMIN) IMIN=IMIN-1                                SPEC2050  54300000
      XMIN=IMIN*XSCALE                                                   SPEC2060  54400000
      IMAX=XMAX/XSCALE                                                   SPEC2070  54500000
      IF(IMAX*XSCALE.LT.XMAX) IMAX=IMAX+1                                SPEC2080  54600000
      XL=IMAX-IMIN                                                       SPEC2090  54700000
      NMAX=(YMAX-YMIN)/YSCALE                                            SPEC2100  54800000
      IF(NMAX*YSCALE+YMIN.LT.YMAX) NMAX=NMAX+1                           SPEC2110  54900000
      IF(NMAX.GT.10) NMAX=10                                             SPEC2120  55000000
      YL=NMAX                                                            SPEC2130  55100000
C                                                                        SPEC2140  55200000
C                                                                        SPEC2150  55300000
C             SET MINIMUM AND MAXIMUM PHASE VALUES, AND Y-AXIS LENGTH    SPEC2160  55400000
C             AND SCALE FACTORS FOR THE OPTIONAL PHASE SPECTRUM PLOTS    SPEC2170  55500000
      YPMIN = -180.0                                                     SPEC2180  55600000
      YPMAX = 180.0                                                      SPEC2190  55700000
      YPL = 10.0                                                         SPEC2200  55800000
      YPSCALE = 40.0                                                     SPEC2210  55900000
C                                                                        SPEC2220  56000000
C             DETERMINE TOTAL NUMBER OF SPECTRUM DATA RECORDS STORED     SPEC2230  56100000
C             ON RANDOM ACCESS                                           SPEC2240  56200000
      NRCSUM = MTRACKS * (2*NREAD+1) + 3                                 SPEC2250  56300000
C                                                                        SPEC2260  56400000
C                                                                        SPEC2270  56500000
C              *LOOP* TO CREATE TRANSDUCER SPECTRUM DATA PLOTS           SPEC2280  56600000
      DO 300 NS=1,MTRACKS                                                SPEC2290  56700000
C                                                                        SPEC2300  56800000
C             DETERMINE IF PLOT IS REQUESTED FOR THIS STATION            SPEC2310  56900000
      IF (NNPLOT(NS) .EQ. 0) GO TO 300                                   SPEC2320  57000000
C                                                                        SPEC2330  57100000
C                                                                        SPEC2340  57200000
C             SET LOOP INDEX DEPENDING ON WHETHER PHASE PLOT IS DESIRED  SPEC2350  57300000
      NAP = 1                                                            SPEC2360  57400000
      IF (NNPLOT(NS) .EQ. 2) NAP=2                                       SPEC2370  57500000
C                                                                        SPEC2380  57600000
C             LOOP FOR AMPLITUDE AND OPTIONAL PHASE SPECTRUM PLOTS       SPEC2390  57700000
C             IF IAP=2, ALL REFERENCES TO AMPLITUDE IN THE FOLLOWING     SPEC2400  57800000
C             LOOP ACTUALLY ARE DIRECT TO PHASE.                         SPEC2410  57900000
      DO 295 IAP=1,NAP                                                   SPEC2420  58000000
C                                                                        SPEC2430  58100000
C             INITIALIZE RANDOM ACCESS RECORD COUNTER FOR PLOT DATA      SPEC2440  58200000
      IPREC = NRCSUM                                                     SPEC2450  58300000
C                                                                        SPEC2460  58400000
C             INITIALIZE COUNTERS FOR THE TOTAL NUMBER OF ELEMENTS IN    SPEC2470  58500000
C             PLOTTING ARRAYS, THE PLOTTING ARRAY BLOCK SIZE, THE        SPEC2480  58600000
C             SPECTRUM ARRAY BLOCK SIZE, THE TOTAL NUMBER OF AMPLITUDE   SPEC2490  58700000
C             VALUES READ, AND THE RANDOM ACCESS RECORD COUNTER FOR      SPEC2500  58800000
C             SPECTRUM DATA                                              SPEC2510  58900000
      MTOTL = 0                                                          SPEC2520  59000000
      KPNTS = 0                                                          SPEC2530  59100000
      NTOTL = 0                                                          SPEC2540  59200000
      NPOIN = 1500                                                       SPEC2550  59300000
      NREC = (NS-1) * (2*NREAD+1) + 2                                    SPEC2560  59400000
      IF (IAP .EQ. 2) NREC=NREC+1                                        SPEC2570  59500000
C                                                                        SPEC2580  59600000
C                                                                        SPEC2590  59700000
C              *LOOP* TO READ AMPLITUDE DATA FROM RANDOM ACCESS AND      SPEC2600  59800000
C             TO DETERMINE THE SIZE  + THE ELEMENTS OF THE PLOTTING      SPEC2610  59900000
```

```
C              ARRAYS
      DO 260 NR=1,NREAD
C
C              COMPUTE TOTAL SIZE OF PLOTTING ARRAYS AND TEST IF PLOTTING
C              ARRAYS ARE COMPLETE
      MTOTL = MTOTL+KPNTS
      IF (MTOTL .EQ. NPLOT) GO TO 270
C
C              DETERMINE RECORD SIZE, READ AMPLITUDE DATA FROM RANDOM
C              ACCESS, AND INCREMENT RECORD COUNTER
      IF (NR .EQ. NREAD) NPOIN=MPOIN
      CALL READMS (20,NUAMP(1),NPOIN,NREC)
      NREC = NREC+2
C
C              COMPUTE TOTAL NUMBER OF AMPLITUDE VALUES READ, AND TEST
C              IF AMPLITUDE VALUES HAVE REACHED THE LOWER FREQUENCY BAND
C              LIMIT
      NTOTL = NTOTL + NPOIN
      IF (IJ1 .GT. NTOTL) GO TO 260
C
C              COMPUTE POINTER USED IN DETERMINING SIZE OF PLOTTING ARRAY
C              BLOCKS AND STARTING AMPLITUDE VALUE USED IN COMPUTING *RI*
      IJ = NTOTL-IJ1+1
C
C              DETERMINE PLOTTING ARRAY RECORD BLOCK SIZE AND SAVE THE
C              SIZE OF THE FIRST RECORD
      KPNTS = IJ
      IF (IJ .GT. NPLOT) KPNTS = NPLOT-MTOTL
      IF (KPNTS .GT. NPOIN) KPNTS = NPOIN
      IF (MTOTL .EQ. 0) IPNTS=KPNTS
C
C              DETERMINE INITIAL STARTING AMPLITUDE USED IN FORMATION OF
C              PLOTTING ARRAYS
      IF (IJ .LT. NPOIN) IJ = NPOIN-IJ
      IF (IJ .GE. NPOIN) IJ = 1
C
C              INITIAL DETERMINATION OF PLOT ARRAY
      DO 250 I=1,KPNTS
      XPLOT(I) = (IJ-1)*DELTF
      IF (IAP .EQ. 2) GO TO 245
      RI(I) = 20. * ALOG10(NUAMP(IJ)/DBREF(NS))
      IF (RI(I) .LT. YMIN) RI(I)=YMIN
      IF (RI(I) .GT. YMAX) RI(I)=YMAX
      IJ = IJ+1
      GO TO 250
  245 RI(I) = NUAMP(IJ)
      IF (RI(I) .LT. YPMIN) RI(I)=YPMIN
      IF (RI(I) .GT. YPMAX) RI(I)=YPMAX
      IJ=IJ+1
  250 CONTINUE
C
C              STORE PLOTTING ARRAY BLOCKS ON RANDOM ACCESS FILE
      CALL WRITMS (20,XPLOT(1),KPNTS,IPREC)
      IPREC = IPREC+1
      CALL WRITMS (20,RI(1),KPNTS,IPREC)
      IPREC = IPREC+1
C
  260 CONTINUE
C
C
C              SET COUNTERS FOR NUMBER OF PLOTTING ARRAY DATA BLOCKS PER
C              STATION, BEGINNING RECORD LOCATION, RECORD SIZE FOR LAST
C              DATA BLOCK
  270 NBLOCK = (IPREC-NRCSUM) / 2
      IPREC = NRCSUM
      JPNTS = KPNTS
C
C              DETERMINE SPECTRUM PLOT LABELING PARAMETERS
      TAPETK = NS
      ENCODE(10,700,PAR1) RUN
  700 FORMAT(*RUN =*,F5.0)
      ENCODE (10,710,PAR2) LO(NS)
  710 FORMAT(*LO =*,F6.3)
      ENCODE (19,720,PAR3) OASPL(NS)
  720 FORMAT(*OASPL =*,G10.4,*DB*)                                      SPEC3360   67400000
      ENCODE (18,730,PAR4) DBREF(NS)                                    SPEC3370   67500000
  730 FORMAT(*DBREF =*,G11.3)                                           SPEC3380   67600000
      ENCODE(17,740,PAR5) THRUST                                        SPEC3390   67700000
  740 FORMAT(*THRUST =*,F7.0,*LB*)                                      SPEC3400   67800000
      ENCODE(12,750,PAR6) RPM                                           SPEC3410   67900000
  750 FORMAT(*RPM =*,F7.0)                                              SPEC3420   68000000
      ENCODE(13,760,PAR7) TAPETK                                        SPEC3430   68100000
  760 FORMAT(*TAPE TK =*,F4.0)                                          SPEC3440   68200000
C                                                                       SPEC3450   68300000
C              DRAW AND LABEL BOTH X-AXIS AND Y-AXIS                    SPEC3460   68400000
      CALL AXES(0.,0.,0.,XL,XMIN,XSCALE,-1.,10.,13HFREQUENCY, HZ,.15,-13SPEC3470   68500000
     1)                                                                 SPEC3480   68600000
      IF (IAP .EQ. 1) CALL AXES (0.,0.,90.,YL,YMIN,YSCALE,1.,10.,44HBLADSPEC3490   68700000
     +E LOADING COEFFICIENTS (20LOG(PS/DBREF)),.21,44)                  SPEC3500   68800000
      IF (IAP .EQ. 2) CALL AXES (0.,0.,90.,YPL,-200.0,YPSCALE,1.,10.,24HSPEC3510   68900000
     +PHASE SPECTRUM (DEGREES),.21,24)                                  SPEC3520   69000000
C                                                                       SPEC3530   69100000
C              LABEL STATION SPECTRUM PLOT                              SPEC3540   69200000
      CALL NOTATE(.5,9.7,.2,PAR1,0.,10)                                 SPEC3550   69300000
      CALL NOTATE(3.,9.7,.2,PAR4,0.,18)                                 SPEC3560   69400000
      CALL NOTATE(.5,9.4,.2,PAR7,0.,13)                                 SPEC3570   69500000
      IF (IAP .EQ. 1) CALL NOTATE (3.9,9.4,.2,PAR3,0.,19)               SPEC3580   69600000
      CALL NOTATE(.5,9.1,.2,PAR6,0.,12)                                 SPEC3590   69700000
      CALL NOTATE(3.,9.1,.2,PAR2,0.,10)                                 SPEC3600   69800000
      CALL NOTATE(5.,9.1,.2,PAR5,0.,17)                                 SPEC3610   69900000
C                                                                       SPEC3620   70000000
C              MAKE INITIAL CALL TO CALPLT                              SPEC3630   70100000
      CALL CALPLT (0.,0.,3)                                             SPEC3640   70200000
C                                                                       SPEC3650   70300000
C                                                                       SPEC3660   70400000
C               *LOOP* TO READ PLOT DATA FROM RANDOM ACCESS, SCALE PLOT SPEC3670   70500000
C              ARRAYS, AND PLOT                                         SPEC3680   70600000
      DO 290 NB=1,NBLOCK                                                SPEC3690   70700000
C                                                                       SPEC3700   70800000
C              DETERMINE SIZE OF PLOT DATA BLOCK                        SPEC3710   70900000
      KPNTS = 1500                                                      SPEC3720   71000000
      IF (NB .EQ. 1) KPNTS = IPNTS                                      SPEC3730   71100000
      IF (NB .EQ. NBLOCK) KPNTS=JPNTS                                   SPEC3740   71200000
C                                                                       SPEC3750   71300000
C              READ PLOTTING ARRAYS FROM RANDOM ACCESS                  SPEC3760   71400000
      CALL READMS (20,XPLOT(1),KPNTS,IPREC)                             SPEC3770   71500000
      IPREC = IPREC+1                                                   SPEC3780   71600000
      CALL READMS (20,RI(1),KPNTS,IPREC)                                SPEC3790   71700000
      IPREC = IPREC+1                                                   SPEC3800   71800000
C                                                                       SPEC3810   71900000
      DO 280 K=1,KPNTS                                                  SPEC3820   72000000
C                                                                       SPEC3830   72100000
C              SCALE PLOTTING ARRAYS                                    SPEC3840   72200000
      XPLOT(K) = (XPLOT(K)-XMIN) / XSCALE                               SPEC3850   72300000
      IF (IAP .EQ. 1) RI(K) = (RI(K)-YMIN)/YSCALE                       SPEC3860   72400000
      IF (IAP .EQ. 2) RI(K) = (RI(K)+200.0)/YPSCALE                     SPEC3870   72500000
C                                                                       SPEC3880   72600000
C              PLOT POINTS                                              SPEC3890   72700000
      CALL CALPLT (XPLOT(K),RI(K),2)                                    SPEC3900   72800000
C                                                                       SPEC3910   72900000
  280 CONTINUE                                                          SPEC3920   73000000
  290 CONTINUE                                                          SPEC3930   73100000
C                                                                       SPEC3940   73200000
C                                                                       SPEC3950   73300000
C              MOVE PEN TO ORIGIN OF NEXT PLOT                          SPEC3960   73400000
      CALL NFRAME                                                       SPEC3970   73500000
C                                                                       SPEC3980   73600000
  295 CONTINUE                                                          SPEC3990   73700000
  300 CONTINUE                                                          SPEC4000   73800000
C                                                                       SPEC4010   73900000
  333 GO TO 10                                                          SPEC4020   74000000
C                                                                       SPEC4030   74100000
C              TERMINATING CALL TO CALPLT                               SPEC4040   74200000
  999 CALL CALPLT (0,0,999)                                             SPEC4050   74300000
C                                                                       SPEC4060   74400000
      STOP                                                              SPEC4070   74500000
      END                                                               SPEC4080   74600000
```

APPENDIX G

```
      PROGRAM TRANS (INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,      TRAN0010    100000
     +TAPE20=201,TAPE7=3001)                                             TRAN0020    200000
C***********************************************************************TRAN0030    300000
C*          PURPOSE                                                     *TRAN0040    400000
C*             PREPROCESSES THE COMBINED TRANSDUCER SPECTRUM DATA       *TRAN0050    500000
C*             OUTPUT FROM PROGRAM RAMANI.  THE REORGANIZED SPECTRUM    *TRAN0060    600000
C*             DATA (PRODUCED BY PROGRAM TRANS) IS THEN INPUT TO THE    *TRAN0070    700000
C*             ROTOR NOISE PREDICTION PROGRAM (RNPPE4) BY RANDOM        *TRAN0080    800000
C*             ACCESS FILE.                                             *TRAN0090    900000
C*                                                                      *TRAN0100   1000000
C*          NAMELIST INPUT PARAMETERS                                   *TRAN0110   1100000
C*             IOPTN   - 0 COMBINED TRANSDUCER SPECTRUM DATA WILL       *TRAN0120   1200000
C*                           BE INPUT BY MAG. TAPE                      *TRAN0130   1300000
C*                         1 COMBINED TRANSDUCER SPECTRUM DATA WILL     *TRAN0140   1400000
C*                           BE INPUT BY RANDOM ACCESS. (PROGRAM TRANS  *TRAN0150   1500000
C*                           IS JOB-STEPPED WITH PROGRAM RAMANI)        *TRAN0160   1600000
C*                        -1 COMBINED TRANSDUCER SPECTRUM DATA WILL BE  *TRAN0170   1700000
C*                           INPUT BY DATA CARD                         *TRAN0180   1800000
C*                         (IOPTN=0 IS DEFAULT VALUE)                   *TRAN0190   1900000
C*             MTRACKS - NUMBER OF POSITIONS OF COMBINED TRANSDUCER     *TRAN0200   2000000
C*                         SPECTRUM DATA. (1≤MTRACKS≤14)                *TRAN0210   2100000
C*                                                                      *TRAN0220   2200000
C*          OPTINAL INPUT (RANDOM ACCESS, TAPE, OR DATA CARD)           *TRAN0230   2300000
C*             ICH     - COMBINED SPECTRUM DATA POSITION NUMBER         *TRAN0240   2400000
C*                         (1≤ICH≤14)                                   *TRAN0250   2500000
C*             DELTF   - BLADE LOADING FREQUENCY (HZ.)                  *TRAN0260   2600000
C*             NSPCT   - NUMBER OF SPECTRUM DATA POINTS PER POSITION    *TRAN0270   2700000
C*             NREAD   - NUMBER OF BLOCKS (OF SIZE 1500) OF SPECTRUM    *TRAN0280   2800000
C*                         DATA POINTS PER STATION                      *TRAN0290   2900000
C*             NUAMP   - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-    *TRAN0300   3000000
C*                         ING THE SPECTRUM DATA AMPLITUDES             *TRAN0310   3100000
C*             NUPHASE - AN ARRAY DIMENSIONED AT LEAST 1500 CONTAIN-    *TRAN0320   3200000
C*                         ING THE SPECTRUM DATA PHASES                 *TRAN0330   3300000
C*                                                                      *TRAN0340   3400000
C*          SUBROUTINES USED                                            *TRAN0350   3500000
C*             NONE                                                     *TRAN0360   3600000
C*                                                                      *TRAN0370   3700000
C*                                                                      *TRAN0380   3800000
C*          REMARK                                                      *TRAN0390   3900000
C*             IF SPECTRUM DATA IS INPUT BY DATA CARD, FOLLOW THE       *TRAN0400   4000000
C*             SPECIFIED FORMAT FOR EACH POSITION                       *TRAN0410   4100000
C*             CARD 1  -  ICH  (INTEGER FORMAT  COLUMNS 3,4)            *TRAN0420   4200000
C*                          NSPCT  (INTEGER FORMAT  COLUMNS 6-10)       *TRAN0430   4300000
C*                          DELTF (G FORMAT  COLUMNS 11-30)             *TRAN0440   4400000
C*             CARD 2  -  NUAMP (G FORMAT  COLUMNS 1-20)                *TRAN0450   4500000
C*                          NUPHASE (G FORMAT  COLUMNS 21-40)           *TRAN0460   4600000
C*                                                                      *TRAN0470   4700000
C***********************************************************************TRAN0480   4800000
C                                                                        TRAN0490   4900000
      COMMON STORE(750,14),NUAMP(1500),INDEX(987),REC1(4),ID(6)          TRAN0500   5000000
      DIMENSION NUPHASE(1500)                                            TRAN0510   5100000
      EQUIVALENCE (STORE(1,1),NUPHASE(1))                                TRAN0520   5200000
C                                                                        TRAN0530   5300000
      REAL NUAMP,NUPHASE                                                 TRAN0540   5400000
C                                                                        TRAN0550   5500000
      NAMELIST /INPUT/ IOPTN,MTRACKS                                     TRAN0560   5600000
C                                                                        TRAN0570   5700000
C           OPEN RANDOM ACCESS FILE.                                     TRAN0580   5800000
      CALL OPENMS (20,INDEX,987,0)                                       TRAN0590   5900000
C                                                                        TRAN0600   6000000
C           INPUT NAMELIST DEFAULT PARAMETER VALUES                      TRAN0610   6100000
      IOPTN = 0                                                          TRAN0620   6200000
      READ (5,INPUT)                                                     TRAN0630   6300000
      WRITE (6,INPUT)                                                    TRAN0640   6400000
C                                                                        TRAN0650   6500000
C           TEST FOR RANDOM ACCESS INPUT                                 TRAN0660   6600000
      IF (IOPTN .GT. 0) GO TO 50                                         TRAN0670   6700000
C                                                                        TRAN0680   6800000
C ............SPECTRUM DATA IS INPUT BY MAG TAPE.  .................... TRAN0690   6900000
C           OR DATA CARD.  TRANSFER DATA TO RANDOM ACCESS.               TRAN0700   7000000
C                                                                        TRAN0710   7100000
      NREC=1                                                             TRAN0720   7200000
C                                                                        TRAN0730   7300000
C        .....LOOP FOR THE NUMBER OF SPECTRUM DATA POSITIONS        .....TRAN0740   7400000
      DO 40 NS=1,MTRACKS                                                 TRAN0750   7500000
```

APPENDIX G

```
C
C             READ STATION IDENTIFICATION RECORD AND RESTORE ID RECORD
C             ON RANDOM ACCESS FILE
C             (ID RECORD MAY COME FROM MAG TAPE OR CARDS)
      IF (IOPTN .EQ. 0) GO TO 10
      READ (5,1000) ICH,NSPCT,DELTF
 1000 FORMAT (2I5,G20.10)
      NREAD = (NSPCT-1)/1500 + 1
      GO TO 15
   10 READ (7) ICH,DELTF,NSPCT,NREAD
   15 REC1(1) = FLOAT(ICH)
      REC1(2) = DELTF
      REC1(3) = FLOAT(NSPCT)
      REC1(4) = FLOAT(NREAD)
      CALL WRITMS (20,REC1(1),4,NREC)
      NREC = NREC+1
C
C             SET NUMBER OF POINTS PER AMPLITUDE (OR PHASE) BLOCK
C             INCLUDING SIZE OF FINAL BLOCK
      NPOIN = 1500
      MPOIN = NSPCT -(NREAD-1)*1500
C
C       .....LOOP FOR THE NUMBER OF DATA BLOCKS PER STATION          .....
      DO 30 NR=1,NREAD
C
C             CHECK FOR LAST BLOCK
      IF (NR .EQ. NREAD) NPOIN=MPOIN
C             TEST TO DETERMINE TYPE OF SPECTRUM DATA INPUT
C                          (MAG TAPE OR DATA CARD)
      IF (IOPTN .EQ. 0) GO TO 20
C             READ SPECTRUM DATA BLOCK
      READ (5,1010) (NUAMP(I),NUPHASE(I),I=1,NPOIN)
 1010 FORMAT (2G20.10)
      GO TO 25
   20 READ (7) (NUAMP(I),NUPHASE(I),I=1,NPOIN)
C             TRANSFER BLOCK INTO TWO RANDOM ACCESS RECORDS
C             (PHASE AND AMPLITUDE)
   25 CALL WRITMS (20,NUAMP(1),NPOIN,NREC)
      NREC = NREC+1
      CALL WRITMS (20,NUPHASE(1),NPOIN,NREC)
      NREC = NREC+1
   30 CONTINUE
C
C             SKIP RECORD WITH STEADY LOADING AND OASPL
      IF (IOPTN .LT. 0) GO TO 40
      READ (7)
C
   40 CONTINUE
C
      GO TO 80
C ..............................................................
C
C ............SPECTRUM DATA IS INPUT BY RANDOM ACCESS.................
C              READ STATION IDENTIFICATION RECORD
   50 NREC = 1
      CALL READMS (20,REC1(1),4,NREC)
      DELTF = REC1(2)
      NSPCT = IFIX (REC1(3))
      NREAD = IFIX (REC1(4))
C ..............................................................
C
C
C        ..... INITIALIZE NECESSARY COUNTERS AND POINTERS.....
C
C             INITIAL RECORD NUMBER FOR REVISED SPECTRUM DATA
   80 NRCSUM = MTRACKS * (2*NREAD+1) + 1
      IREC = NRCSUM
C             USING RECORD SIZE OF LAST DATA BLOCK, DETERMINE NUMBER
C             OF POINTS IN SECOND HALF OF LAST DATA BLOCK
      MPOIN = NSPCT - (NREAD-1)*1500
      J = (MPOIN-1)/750
      JPNTS = MPOIN - J*750
C             DETERMINE SIZE OF THE LAST RECORD IN REVISED SPECTRUM DATA
      L = (JPNTS-1)/250
      LMEMB = JPNTS - L*250
      MWORDS = MTRACKS*LMEMB
C          ..................................................
C
C
C  ..............SECTION TO REORGANIZE COMBINED SPECTRUM DATA.........
C            NEW RANDOM ACCESS RECORDS ARE FORMED BY GROUPING
C             250 AMPLITUDE (OR PHASE) VALUES FROM EACH POSITION.
C
C        .....OUTER LOOP FOR EITHER AMPLITUDE OR PHASE DATA.      .....
      DO 300 I=1,2
C
C        .....INNER LOOR FOR THE NUMBER OF SPECTRUM DATA BLOCKS
C             OF SIZE 1500.
      DO 300 NR=1,NREAD
C
C            SET RECORD LENGTH FOR UNREVISED SPECTRUM DATA BLOCK -
C            INCLUDING CHECK FOR LAST RECCRD
      NPOIN = 1500
      IF (NR .EQ. NREAD) NPOIN=MPOIN
C
C
C       .....INNER LOUP TO DIVIDE UNREVISED SPECTRUM DATA BLOCKS  .....
C            IN HALF (I.E. TWO BLOCKS OF 750).
      DO 250 II=1,2
C
C            SET RECORD LENGTH FOR REVISED SPECTRUM DATA BLOCKS
      NWORDS = 250*MTRACKS
C
C       .....INNER LOOP FOR THE NUMBER OF COMBINED SPECTRUM DATA ......
C            POSITIONS.
      DO 200 NS=1,MTRACKS
C            SET RECORD NUMBER POINTER.  ALSO TEST IF AMPLITUDE OR
C            PHASE DATA IS TO BE READ.
      NREC = (NS-1) * (2*NREAD+1) + NR+NR
      IF (I .EQ. 2) NREC = NREC+1
C
C            READ BLOCK OF UNREVISED AMPLITUDE OR PHASE DATA.
      CALL READMS (20,NUAMP(1),NPLIN,NREC)
C
C            DETERMINE THE NUMBER OF POINTS IN FIRST (OR SECOND) HALF
C            OF UNREVISED AMPLITUDE (OR PHASE) DATA BLOCK.
      IPNTS = 750
      IF (NR .NE. NREAD) GO TO 100
      IF (II .EQ. 1 .AND. J .EQ. 1) GO TO 100
      IF (II .EQ. 2 .AND. J .EQ. 0) GO TO 300
      IPNTS = JPNTS
C
C            SET COUNTER FOR EITHER TFE FIRST OR SECOND HALF OF
C            UNREVISED DATA BLOCK
  100 IPP = 1
      IF (II .EQ. 2) IPP= 751
C
C            SET COUNTER FOR NUMBER OF POINTS PER STATION IN A REVISED
C            RECORD.  ALSO TEST TO DETERMINE HOW MANY REVISED RECORDS
C            WILL BE MADE FROM FIRST (OR SECOND) HALF OF UNREVISED DATA
      KMEMB = 250 ,
      LL = 3
      IF (NR .EQ. NREAD) LL=L+1
C
C        .....INNER LOOP FOR CREATING LL AMPLITUDE (OR PHASE)      .....
C             RECORDS FOR FIRST (OR SECOND) HALF OF UNREVISED RECORD
      DO 160 N=1,LL
C
C            TEST FOR LAST RECORD
      IF (NR.NE.NREAD .OR. N.NE.LL) GO TO 130
      IF (J.EQ.0 .OR. II.EQ.2) KMEMB=LMEMB
C
C            SET POINTER FOR INITIAL LOCATION IN STORE ARRAY WHERE
C            AMPLITUDE (OR PHASE) VALUES SHOULD BEGIN TO BE STORED
  130 NSTRT = (N-1)*NWORDS + (NS-1)*KMEMB
C
C          ...INNER TO STORE AMPLITUDE (OR PHASE) VALUES
      DO 150 IP=1,KMEMB
      NSTRT = NSTRT+1
      STORE(NSTRT) = NUAMP(IPP)
      IPP = IPP+1
  150 CONTINUE
C
  160 CONTINUE
C
  200 CONTINUE
C
C                RESET NUMBER OF POINTS PER STATION PER RECORD INDICATOR
      KMEMB = 250
C
C          ..... INNER LOOP TO WRITE REVISED RECORD TO .....
C                         RANDOM  ACCESS  FILE
      DO 250 N=1,LL
C
C                SET POINTER FOR INITIAL LOCATION OF DATA TO BE WRITTEN
C                TO RANDOM ACCESS
      NN = (N-1)*NWORDS + 1
C                TESTS FOR LAST RECORD
      IF (NR.NE.NREAD .OR. N.NE.LL) GO TO 230
      IF (II.NE.2 .AND. J.NE.0) GO TO 230
      KMEMB = LMEMB
      NWORDS = MWORDS
  230 CALL WRITMS (20,STORE(NN),NWORDS,IREC)
      IREC = IREC+1
C
  250 CONTINUE
C
  300 CONTINUE
C     ...................................................................
C
C                 CREATE ARRAY TO PASS NEEDED PARAMETERS TO PROGRAM
C                 RNPPE4 BY MEANS OF RANDOM ACCESS
C
      NREC = 1
      ID(1) = MTRACKS
      ID(2) = NSPCT
      ID(3) = NRCSUM
      ID(4) = IREC
      ID(5) = LMEMB
      ID(6) = MWORDS
      CALL WRITMS (20,ID(1),6,NREC)
      NREC = 2
      CALL WRITMS (20,DELTF,1,NREC)
C
      STOP
      END
      PROGRAM RNPPE4 (INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE20=201,TAPE4=1001)
C***********************************************************************
C*               PURPOSE                                               *
C*                TO COMPUTE THE ROTOR ROTATIONAL NOISE FOR A HOVERING *
C*                HELICOPTER.  SOUND PRESSURE LEVELS (SPL) ARE COMPUTED *
C*                AT EACH HARMONIC USING WRIGHTS SOLUTION.  THE BLADE  *
C*                LOADING HARMONICS (BLH) CAN BE OBTAINED BY INTEGRATING*
C*                OVER ALL OR PART OF THE STEADY LOADING DISTRIBUTION  *
C*                THE SPL VALUES GENERATED AT THE VARIOUS HARMONICS    *
C*                CAN BE ADJUSTED BY APPLYING ONE OF FIVE SPECTRUM     *
C*                CHORD DISTRIBUTIONS                                  *
C*                                                                     *
C*               NAMELIST INPUT PARAMETERS                             *
C*                                                                     *
C*                  ROTOR NAMELIST   (FIXED PARAMETERS)                *
C*                NSPL     - NUMBER OF SOUND PRESSURE LEVELS PLOTS TO  *
C*                           BE DETERMINED WITH THESE ROTOR PARAMETERS *
C*                           (USUALLY NSPL = (NUMBER OF COMBINED SPECTRUM*
C*                            DATA POSITIONS)+(NUMBER OF INTEGRATION   *
C*                            DESIRED))                                *
C*                           (NSPL=6 IS DEFAULT VALUE)                 *
C*                EFMACH   - EFFECTIVE RADIAL MACH NUMBER              *
C*                OBSELV   - OBSERVER ELEVATION ANGLE (RADIANS)        *
C*                EFPTCH   - EFFECTIVE BLADE PITCH (RADIANS)           *
C*                OBSDIS   - OBSERVER DISTANCE FROM ROTOR CENTER (FT.) *
C*                NB       - NUMBER OF BLADES                          *
```

```
C*                  (NB=4 IS DEFAULT VALUE)                      *RNPP0280  29900000
C*        THRUST  - TOTAL LIFT (WEIGHT OF HELICOPTER) (LBS.)     *RNPP0290  30000000
C*        TORQUE  - TOTAL ROTOR DRAG FORCE (LBS.)                *RNPP0295  30100000
C*        C       - SPEED OF SOUND (FT./SEC.)                    *RNPP0300  30200000
C*                  (C=1084.8 IS DEFAULT VALUE)                  *RNPP0310  30300000
C*        RS      - ROTOR ROTATIONAL SPEED (RPM)                 *RNPP0320  30400000
C*        F       - BLADE PASSAGE FREQUENCY (RPS)                *RNPP0330  30500000
C*        EFRAD   - EFFECTIVE BLADE RADIUS (FT.)                 *RNPP0340  30600000
C*        EFCORD  - EFFECTIVE BLADE CHORD (FT.)                  *RNPP0350  30700000
C*        XMAX    - MAXIMUN FREQUENCY OF AVAILABLE FLIGHT POINT  *RNPP0360  30800000
C*                  LOADING DATA (RPS)                           *RNPP0370  30900000
C*                  (XMAX=1000.0 IS DEFAULT VALUE)               *RNPP0380  31000000
C*        OBSAZI  - OBSERVER AZIMUTH ANGLE (RADIANS)             *RNPP0390  31100000
C*        STDYLD  - AN ARRAY DIMENSION 14 (FOR THE MAXIMUN       *RNPP0400  31200000
C*                  NUMBER OF COMBINED SPECTRUM DATA POSITIONS)  *RNPP0410  31300000
C*                  CONTAINING THE STEADY LOADING COEFFICIENTS   *RNPP0420  31400000
C*                  FOR EACH OF THE SPECTRUM DATA POSITIONS      *RNPP0430  31500000
C*                  (STDYLD(I)=0.0 IS DEFAULT VALUE  1≤I≤14)     *RNPP0440  31600000
C*        RP      - AN ARRAY DIMENSION 14 (FOR THE MAXIMUN       *RNPP0450  31700000
C*                  NUMBER OF COMBINED SPECTRUM DATA POSITIONS)  *RNPP0460  31800000
C*                  CONTAINING THE RELATIVE SPECTRUM DATA        *RNPP0470  31900000
C*                  POSITIONS. THE RP ARRAY MUST BE STRICTLY     *RNPP0480  32000000
C*                  INCREASING AND 0≤RP(I)≤1 FOR ALL I.          *RNPP0490  32100000
C*                  (RP(I)=0.0 IS DEFAULT VALUE 1≤I≤14)          *RNPP0500  32200000
C*        IBLHOPT - 0 COMBINED SPECTRUM DATA IS TO BE INPUT AND  *RNPP0510  32300000
C*                    USED TO COMPUTE BLADE LOADING HARMONIC     *RNPP0520  32400000
C*                    COEFFICIENTS (BLH)                         *RNPP0530  32500000
C*                - 1 BLH COEFFICIENTS AND PHASE ANGLES ARE TO   *RNPP0540  32600000
C*                    BE COMPUTED INTERNALLY BY MEANS OF AN      *RNPP0550  32700000
C*                    INLINE FUNCTION                            *RNPP0560  32800000
C*                  (IBLHOPT=0 IS DEFAULT VALUE)                 *RNPP0570  32900000
C*                                                               *RNPP0580  33000000
C*             OPTIONAL INPUT                                    *RNPP0590  33100000
C*                                                               *RNPP0600  33200000
C*          INPUT NAMELIST   (TO BE USED WITH SPECTRUM DATA      *RNPP0610  33300000
C*             INPUT.  THE PARAMETERS IN THIS NAMELIST MUST      *RNPP0620  33400000
C*             BE REDEFINED FOR EACH SET OF BLH COEFFICIENTS     *RNPP0630  33500000
C*             AND PHASE ANGLES BEING COMPUTED.)                 *RNPP0640  33600000
C*        INTEGRAT - 0 NO INTEGRATION OF THE STEADY LOADING DIST.*RNPP0650  33700000
C*                    IS TO BE PERFORMED                         *RNPP0660  33800000
C*                  K STEADY LOADING DIST. IS INTEGRATED FROM    *RNPP0670  33900000
C*                    ZERO TO THE KTH COMBINED SPECTRUM DATA     *RNPP0680  34000000
C*                    POSITION (0<K≤MTRACKS)                     *RNPP0690  34100000
C*                - -1 STEADY LOADING DIST. IS TO INTEGRATED     *RNPP0700  34200000
C*                     ALONG A FRACTION OF THE CHORD BUT NOT AT  *RNPP0710  34300000
C*                     ONE OF THE COMBINED SPECTRUM DATA POSITIONRNPP0720  34400000
C*                  (INTEGRAT=0 IS DEFAULT VALUE)                *RNPP0730  34500000
C*        ITRACK  - NUMBER OF THE SPECTRUM DATA POSITION         *RNPP0740  34600000
C*                  (IF INTEGRATION IS DESERIRED SET ITRACK=1)   *RNPP0750  34700000
C*                  (ITRACK=1 IS DEFAULT VALUE)                  *RNPP0760  34800000
C*        PRTLINT - FRACTION OF CHORD ALONG WHICH THE STEADY     *RNPP0770  34900000
C*                  LOADING DIST. IS TO BE INTEGRATED.           *RNPP0780  35000000
C*                  (0<PRTLINT<1)                                *RNPP0790  35100000
C*                  (PRTLINT=1.0 IS DEFAULT VALUE)               *RNPP0800  35200000
C*        LO      - STEADY LOADING COEFFICIENT FOR THE ITRACK    *RNPP0810  35300000
C*                  (TH) SPECTRUM DATA POSITION                  *RNPP0820  35400000
C*                  (LO=0.0 IS DEFAULT VALUE)                    *RNPP0830  35500000
C*        ICHORD  - DETERMINES WHICH CHORD DISTRIBUTION FUNCTION*RNPP0840  35600000
C*                  IS USED TO CORRECT SPL VALUES                *RNPP0850  35700000
C*                  0 POINT LOADING DATA IS USED UNCORRECTED     *RNPP0860  35800000
C*                  1 RECTANGULAR SPECTRUM FUNCTION              *RNPP0870  35900000
C*                  2 HALF-COSINE SPECTRUM FUNCTION              *RNPP0880  36000000
C*                  3 TRIANGULAR SPECTRUM FUNCTION               *RNPP0890  36100000
C*                  4 SAWTOOTH SPECTRUM FUNCTION                 *RNPP0900  36200000
C*                  (ICHORD=0 IS DEFAULT VALUE)                  *RNPP0910  36300000
C*        INCOF   - BLH COEFFICIENT OUTPUT CONTROL PARAMETER     *RNPP0920  36400000
C*                = K EVERY K-1(TH) BLH COEFFICIENT AND PHASE    *RNPP0930  36500000
C*                      ANGLE WILL BE OUTPUT (IF K=1 ALL BLH     *RNPP0940  36600000
C*                      COEFFICIENTS WILL BE OUTPUT)             *RNPP0950  36700000
C*                  (INCOF=5 IS DEFAULT VALUE)                   *RNPP0960  36800000
C*                                                               *RNPP0970  36900000
C*          INBLH NAMELIST (TO BE USED WHEN BLH COEFFICIENTS     *RNPP0980  37000000
C*             AND PHASE ANGLES ARE INTERNALLY COMPUTED.  THE    *RNPP0990  37100000
C*             PARAMETERS IN THIS NAMELIST MUST BE REDEFINED     *RNPP1000  37200000
C*                    FOR EACH SET OF BLH COEFFS. THAT ARE DESIRED)  *RNPP1010  37300000
C*             NBLHPT  - NUMBER OF BLH COEFFIENTS AND PHASE ANGLES    *RNPP1020  37400000
C*                       TO BE COMPUTED. (1≤NBLHPT≤2000)              *RNPP1030  37500000
C*             X,C     - TWO INPUT CONSTANTS USED IN DEFINING THE     *RNPP1040  37600000
C*                       BLH COEFFICIENT INLINE FUNCTION.             *RNPP1050  37700000
C*                              BLH(Y) = C / Y**X                     *RNPP1060  37800000
C*             ICHORD  - SAME AS DESCRIBED ABOVE                      *RNPP1070  37900000
C*             INCOF   - SAME AS DESCRIBED ABOVE                      *RNPP1080  38000000
C*                                                                    *RNPP1090  38100000
C*          RANDOM ACCESS INPUT PARAMETERS                            *RNPP1100  38200000
C*             MTRACKS - NUMBER OF COMBINED SPECTRUM DATA POSITIONS   *RNPP1110  38300000
C*                       (1<MTRACKS< 14)                              *RNPP1120  38400000
C*             NSPCT   - NUMBER OF SPECTRUM DATA POINTS PER POSITION  *RNPP1130  38500000
C*             NRCSUM  - BEGINNING AND ENDING RANDOM ACCESS RECORD    *RNPP1140  38600000
C*             KREC      LOCATIONS FOR THE REVISED SPECTRUM DATA      *RNPP1150  38700000
C*             MPOIN   - RECORD SIZE OF THE LAST SPECTRUM DATA RANDOM*RNPP1160  38800000
C*             MWORDS  - ACCESS RECORD (WITH AND WITHOUT INTEGRATION)*RNPP1170  38900000
C*             SPECTRA - AN ARRAY DIMENSION 250X16 CONTAINING THE     *RNPP1180  39000000
C*                       REVISED SPECTRUM DATA BLOCKS                 *RNPP1190  39100000
C*                                                                    *RNPP1200  39200000
C*          SUBROUTINUE USED                                          *RNPP1210  39300000
C*             SPLS                                                   *RNPP1220  39400000
C*             CSIONI                                                 *RNPP1230  39500000
C*             BSSLS                                                  *RNPP1240  39600000
C*                                                                    *RNPP1250  39700000
C*********************************************************************RNPP1260  39800000
C                                                                      RNPP1270  39900000
      COMMON SPECTRA(250,16),PRESSUR(250),PHASE(250),REPRESS(2000),    RNPP1280  40000000
     +REPHASE(2000),ALPHA(2000)                                        RNPP1290  40100000
      DIMENSION INDEX(987),ID(6)                                       RNPP1300  40200000
      DIMENSION STDYLD(14),CP(16),T(16),XPP(17),YPP(17),RP(16),WK(500), RNPP1310  40300000
     +XINT(51),YINT(51)                                                RNPP1315  40400000
      DIMENSION BJ(2000),SPL(500),ALIMIT(500),AMBN(500)                RNPP1320  40500000
C                                                                      RNPP1330  40600000
      EQUIVALENCE (BJ(1),SPECTRA(1,1))                                 RNPP1340  40700000
      EQUIVALENCE (SPL(1),SPECTRA(1,9))                                RNPP1350  40800000
      EQUIVALENCE (ALIMIT(1),SPECTRA(1,11))                            RNPP1360  40900000
      EQUIVALENCE (AMBN(1),SPECTRA(1,13))                              RNPP1370  41000000
      EQUIVALENCE (XINT(1),ALPHA(1))                                   RNPP1372  41100000
      EQUIVALENCE (YINT(1),ALPHA(1001))                                RNPP1374  41200000
      EQUIVALENCE (WK(1),REPHASE(1))                                   RNPP1375  41300000
C                                                                      RNPP1380  41400000
      DATA P0/.000000417/, RADIAN/57.29577913/, PI/3.14159265359/,     RNPP1390  41500000
     +SQRT2/1.4142135623/                                              RNPP1400  41600000
C                                                                      RNPP1410  41700000
C                                                                      RNPP1420  41800000
      INTEGER Q,ASKFOR,STEP,SUB                                        RNPP1430  41900000
      REAL KT,KQ,KQQMB,MBBJ,LO                                         RNPP1440  42000000
      COMPLEX EI,TERM0,TERM1,TERM2,TERM3,FIRST,ZERO,SECOND,THIRD,EIQ1  RNPP1450  42100000
C                                                                      RNPP1460  42200000
      NAMELIST /ROTOR/ NSPL,EFMACH,OBSELV,EFPTCH,OBSDIS,NB,THRUST,     RNPP1470  42300000
     1TORQUE,C,RS,F,EFRAD,EFCORD,XMAX,OBSAZI,STDYLD,RP,IBLHOPT         RNPP1480  42400000
     2 /INPUT/ NTEGRAT,ITRACK,PRTLINT,LO,ICHORD,INCOF                  RNPP1490  42500000
     3 /INBLH/ NBLHPT,C,X,ICHORD,INCOF                                 RNPP1500  42600000
C                                                                      RNPP1510  42700000
C           DEFINE THE RECTANGULAR DISTRIBUTION FUNCTION               RNPP1520  42800000
      CHORD1(M) = SIN(M*BPIFTO) / (M*BPIFTO)                           RNPP1530  42900000
C           DEFINE HALF COSINE SPECTRUM FUNCTION                       RNPP1540  43000000
      CHORD2(M) = COS(M*BPIFTO) / (1.0-M*M*SQ2BFTO)                    RNPP1550  43100000
C           DEFINE TRIANGULAR SPECTRUM FUNCTION                        RNPP1560  43200000
      CHORD3(M) = ((SIN(M*BPIFTO2)/(M*BPIFTO2))**2)                    RNPP1570  43300000
C           DEFINE SAWTOOTH SPECTRUM FUNCTION                          RNPP1580  43400000
      CHORD4(M) = CABS ( (1.0,0.0) - CEXP(CMPLX(0.0,M*C2BPIFT)) + CMPLX(RNPP1590  43500000
     +             0.0,M*C2BPIFT) ) / (M*M*SQBPIFT)                    RNPP1600  43600000
C           DEFINE OPTIONALLY USED BLH COEFFICIENT FUNCTION            RNPP1610  43700000
      BLH(Y)      = C / Y**X                                           RNPP1620  43800000
C                                                                      RNPP1630  43900000
C           OPEN THE RANDOM ACCESS FILE                                RNPP1640  44000000
      CALL OPENMS (20,INDEX,987,0)                                     RNPP1650  44100000
C                                                                      RNPP1660  44200000
C                                                                      RNPP1670  44300000
C           ROTOR NAMELIST DEFAULT PARAMETER VALUES                    RNPP1680  44400000
      NSPL = 6                                                         RNPP1690  44500000
      NB = 4                                                           RNPP1700  44600000
```

APPENDIX G

```
      C = 0.10848E04                                                    RNPP1710   44700000
      XMAX = 1.0E03                                                     RNPP1720   44800000
      DO 5 I=1,16                                                       RNPP1730   44900000
      IF (I .LE. 14) STDYLD(I)=0.0                                      RNPP1740   45000000
      RP(I) = 0.0                                                       RNPP1750   45100000
    5 CONTINUE                                                          RNPP1760   45200000
      IBLHOPT = 0                                                       RNPP1770   45300000
C                                                                       RNPP1780   45400000
C               READ AND WRITE FIXED PROGRAM PARAMETERS AND CHECK       RNPP1790   45500000
C               FOR EOF                                                 RNPP1800   45600000
   10 READ (5,ROTOR)                                                    RNPP1810   45700000
      IF (EOF,5) 999,20                                                 RNPP1820   45800000
   20 WRITE (6,ROTOR)                                                   RNPP1830   45900000
C                                                                       RNPP1840   46000000
C     ..................................................................RNPP1850   46100000
C                                                                       RNPP1860   46200000
C               COMPUTE MAXIMUM NUMBER OF BLH COEFFICIENTS DESIRED      RNPP1870   46300000
      MMAX=XMAX/F                                                       RNPP1880   46400000
      NN=MMAX+20                                                        RNPP1890   46500000
C               DEFINE COMPLEX NUMBER I                                 RNPP1900   46600000
      EI=CMPLX(0.,1.)                                                   RNPP1910   46700000
C               COMPUTE SPECTRUM DISTRIBUTION FUNCTION PARAMETERS       RNPP1920   46800000
      TWOPI = PI+PI                                                     RNPP1930   46900000
      TO = EFCORD / (TWOPI*EFRAD*F)                                     RNPP1940   47000000
      SQ2BFTO = 4.0 * (NB*F*TO)**2                                      RNPP1950   47100000
      BPIFTO = NB*PI*F*TO                                               RNPP1960   47200000
      BPIFTO2 = BPIFTO / 2.0                                            RNPP1970   47300000
      SQBPIFT = BPIFTO * BPIFTO * 2.0                                   RNPP1980   47400000
      C2BPIFT = -2.0 * BPIFTO                                           RNPP1990   47500000
C               COMPUTE MAXIMUM NUMBER OF HARMONICS                     RNPP2000   47600000
      S = EFMACH*NB*COS(OBSELV)                                         RNPP2010   47700000
      MAXHAR = (MMAX-10) / (NB+1.25*S)                                  RNPP2020   47800000
C               COMPUTE THRUST OPERATING AND TORQUE OPERATING CONSTANTS RNPP2030   47900000
      FTHRUST = F*THRUST                                                RNPP2040   48000000
      COBSDIS = C*OBSDIS                                                RNPP2050   48100000
      EFDRAG = TORQUE/EFRAD                                             RNPP2055   48200000
      KT = FTHRUST/COBSDIS * SIN(OBSELV)                                RNPP2060   48300000
      KQ = F*EFDRAG / (COBSDIS*EFMACH)                                  RNPP2070   48400000
C                                                                       RNPP2080   48500000
C                                                                       RNPP2090   48600000
C               TEST TO DETERMINE IF BLADE LOADING HARMONIC             RNPP2100   48700000
C               (BLH) COEFFICIENTS ARE TC BE INTERNALLY COMPUTED.       RNPP2110   48800000
      IF (IBLHOPT .NE. 0) GO TO 22                                      RNPP2120   48900000
C                                                                       RNPP2130   49000000
C               READ IDENTIFICATICN RECORD AND BLADE PASSAGE FREQUENCY  RNPP2140   49100000
      NREC = 1                                                          RNPP2150   49200000
      CALL READMS (20,ID(1),6,NREC)                                     RNPP2160   49300000
      NREC = 2                                                          RNPP2170   49400000
      CALL READMS (20,DELTF,1,NREC)                                     RNPP2180   49500000
      DELTF2 = DELTF/2.                                                 RNPP2190   49600000
C               DETERMINE NUMBER CF TRACKS (INCLUDING ENDPOINTS FOR     RNPP2200   49700000
C               INTEGRATION)                                            RNPP2210   49800000
      MTRACKS = ID(1)+2                                                 RNPP2220   49900000
C               SET TOTAL NUMBER CF SPECTRUM POINTS (PER STATION)       RNPP2230   50000000
      NSPCT = ID(2)                                                     RNPP2240   50100000
C               SET COUNTER FOR NUMBER OF RAMDOM ACCESS READS           RNPP2250   50200000
      NRCSUM = ID(3)                                                    RNPP2260   50300000
      KREC = ID(4)                                                      RNPP2270   50400000
      NDIFF = KREC-NRCSUM                                               RNPP2280   50500000
      NREAD = NDIFF/2                                                   RNPP2290   50600000
      NREAD2 = NREAD*2                                                  RNPP2300   50700000
      NRSUM3 = NRCSUM-3                                                 RNPP2310   50800000
C               SET RECORD SIZE FOR LAST RANDOM ACCESS READ (WITH AND   RNPP2320   50900000
C               WITHOUT INTEGRATICN)                                    RNPP2330   51000000
      MPOIN = ID(5)                                                     RNPP2340   51100000
      MWORDS = ID(6)                                                    RNPP2350   51200000
C     ..................................................................RNPP2360   51300000
C                                                                       RNPP2370   51400000
C..............LOOP FOR NUMBER CF SCUND PRESSURE LEVEL PLOTS............RNPP2380   51500000
C                                                                       RNPP2390   51600000
   22 DO 777 NS=1,NSPL                                                  RNPP2400   51700000
C                                                                       RNPP2410   51800000
C                                                                       RNPP2420   51900000
C               INPUT OR INBLH NAMELIST DEFAULT PARAMETER VALUE         RNPP2430   52000000
```

```
      NTEGRAT = 0
      ITRACK = 1
      PRTLINT = 1.0
      LU = 0.0
      ICHORD = 0
      INCUF = 5
C
C
C              TEST TO DETERMINE OPTIONAL BLH INPUT
      IF (IBLHOPT .EQ. 0) GO TO 28
      READ (5,INBLH)
      IF (EOF,5) 999,25
   25 WRITE (6,INBLH)
      GO TO 402
C
C
C              INTIALIZE RANDOM ACCESS RECORD COUNTERS AND RECORD
C              SIZES (WITH AND WITHOUT INTEGRATION)
   28 IREC = NRCSUM
      JREC = IREC+NREAD
      NRVSUM = 3
      IF (NRSUM3 .LT. NREAD2) NRVSUM = NRCSUM+NREAD2+1
      NREC = NRVSUM
      MREC = NREC+NREAD
      NPOIN = 250
      NWORDS = NPOIN*ID(1)
C
C              READ AND WRITE VARIABLE INPUT PARAMETERS AND CHECK
C              FOR EOF.
      READ (5,INPUT)
      IF (EOF,5) 999,30
   30 WRITE (6,INPUT)
C
C              TEST FOR CHORD INTEGRATICN
      IF (INTEGRAT .EQ. 0) GO TO 200
C
C   **************************************************************
C                        INTEGRATION OVER CHORD
C
C              SET INDEPENDENT VARIABLE ARRAY (INCLUDING ENDPOINTS)
      IPRTL = 0
      CP(1) = 0.0
      DO 40 I=2,MTRACKS
C              TEST TO DETERMINE POSITICN OF FRACTIONAL CHORD
C              INTEGRATION CONSTANT
      IF (INTEGRAT .GT. 0) GO TC 35
      IF (IPRTL .EQ. 0 .AND. RP(I) .GT. PRTLINT) IPRTL=I
   35 CP(I) = RP(I)*EFCORD
   40 CONTINUE
C
C              DETERMINE PERCENTAGE OF CHORD INTEGRATED
      PRCNTGR = 100.0*PRTLINT
C
C              TEST FOR PARTIAL CHORD INTEGRATION
      IF (INTEGRAT .GT. 0) GO TC 70
C              SET UP INDEPENDENT VARIABLE ARRAY (INCLUDING ENDPOINTS)
      MTP1 = MTRACKS+1
      II = 1
      DO 60 I=1,MTP1
      IF (I .NE. IPRTL) GO TO 50
      XPP(I) = PRTLINT*EFCORD
      PRTLINT = XPP(I)
      GO TO 60
   50 XPP(I) = CP(II)
      II = II+1
   60 CONTINUE
C
C              ZERO THE DEPENDENT VARIABLE AT THE ENDPOINTS
   70 DO 80 I=1,250
      SPECTRA(I,1) = 0.0
      SPECTRA(I,MTRACKS) = 0.0
   80 CONTINUE
C
C              CREATE INDEPENDENT VARIABLE ARRAY NECESSARY IN
```

```
C              COMPUTING INTEGRATED (AVERAGED) PHASE DATA.                  RNPP3162  59500000
      INTP = IFIX(51.0 * PRTLINT)                                           RNPP3163  59600000
      IF (INTP .LE. 0)  INTP=1                                              RNPP3164  59700000
C                                                                           RNPP3165  59800000
      XINT(1) = 0.0                                                         RNPP3166  59900000
      DXINT = EFCORD / FLOAT(INTP)                                          RNPP3167  60000000
      DO 85 I=2,INTP                                                        RNPP3168  60100000
      XINT(I) = XINT(I-1) + DXINT                                           RNPP3169  60200000
   85 CONTINUE                                                              RNPP316A  60300000
      XINT(INTP) = PRTLINT * EFCORD                                         RNPP316B  60400000
C                                                                           RNPP316C  60500000
C                                                                           RNPP3170  60600000
C     ......... LOOP FOR THE NUMBER OF RANDOM ACCESS READS.........         RNPP3180  60700000
      DO 190 NR=1,NREAD                                                     RNPP3190  60800000
C                                                                           RNPP3200  60900000
C              CHECK FOR LAST READ (AND ADJUST RECORD SIZE)                 RNPP3210  61000000
      IF (NR .NE. NREAD) GO TO 90                                           RNPP3220  61100000
      NPOIN = MPOIN                                                         RNPP3230  61200000
      NWORDS = MWORDS                                                       RNPP3240  61300000
C                                                                           RNPP3250  61400000
C              READ SPECTRUM DATA (AND INCREMENT COUNTER)                   RNPP3260  61500000
   90 CALL READMS (20,SPECTRA(1,2),NWORDS,IREC)                             RNPP3270  61600000
      IREC = IREC+1                                                         RNPP3280  61700000
C                                                                           RNPP3290  61800000
C              USE STEADY LOADING VALUES FOR FIRST *SET* OF KNOWN           RNPP3300  61900000
C              SPECTRUM DATA POINTS                                         RNPP3310  62000000
      IF (NR .NE. 1) GO TO 110                                              RNPP3320  62100000
      MT = MTRACKS-2                                                        RNPP3330  62200000
      DO 100 I=1,MT                                                         RNPP3340  62300000
      IP1 = I+1                                                             RNPP3350  62400000
      SPECTRA(1,IP1) = STDYLO(I)                                            RNPP3360  62500000
  100 CONTINUE                                                              RNPP3370  62600000
C                                                                           RNPP3380  62700000
C          .....LOOP TO INTERPOLATE AND INTEGRATE PRESSURE DATA.....        RNPP3390  62800000
  110 DO 130 I=1,NPOIN                                                      RNPP3400  62900000
      IW = -1                                                               RNPP3410  63000000
C              USE SPECTRUM DATA TO CONSTRUCT INTEGRATION                   RNPP3420  63100000
C              (OR INTERPOLATION) NODE POINTS                               RNPP3430  63200000
      DO 120 K=1,MTRACKS                                                    RNPP3440  63300000
      T(K) = SPECTRA(I,K)                                                   RNPP3450  63400000
  120 CONTINUE                                                              RNPP3460  63500000
C              TEST FOR PARTIAL CHORD INTEGRATION                           RNPP3470  63600000
      IF (NTEGRAT .GT. 0) GO TO 125                                         RNPP3480  63700000
C                                                                           RNPP3490  63800000
C              INTERPOLATE FOR PARTIAL CHORD INTEGRATION                    RNPP3500  63900000
      CALL CSIUNI (16,MTRACKS,1,1,1,CP,T,PRTLINT,FX,IW,WK,IERR)             RNPP3510  64000000
C              CONSTRUCT PARTIAL CHORD INTEGRATION NODE POINTS              RNPP3520  64100000
      I1 = 1                                                                RNPP3530  64200000
      DO 122 K=1,MTP1                                                       RNPP3540  64300000
      IF (K .NE. IPRTL) GO TO 121                                           RNPP3550  64400000
      YPP(K) = FX                                                           RNPP3560  64500000
      GO TO 122                                                             RNPP3570  64600000
  121 YPP(K) = T(I1)                                                        RNPP3580  64700000
      I1 = I1+1                                                             RNPP3590  64800000
  122 CONTINUE                                                              RNPP3600  64900000
C              INTEGRATE OVER PARTIAL CHORD                                 RNPP3610  65000000
      IW = -1                                                               RNPP3620  65100000
      CALL SPLS (17,MTP1,1,XPP,YPP,1,IPRTL,PROXIN,IW,WK,IERR)               RNPP3630  65200000
      GO TO 128                                                             RNPP3640  65300000
C                                                                           RNPP3650  65400000
C              INTEGRATE OVER ENTIRE CHORD OR OVER PARTIAL CHORD            RNPP3660  65500000
C              IF CHORD FRACTION IS ONE OF SPECTRUM DATA POSITIONS          RNPP3670  65600000
  125 CALL SPLS (16,MTRACKS,1,CP,T,1,NTEGRAT,PROXIN,IW,WK,IERR)             RNPP3680  65700000
  128 PRESSUR(I) = PROXIN                                                   RNPP3690  65800000
  130 CONTINUE                                                              RNPP3700  65900000
C                                                                           RNPP3710  66000000
      IF (NR .EQ.1) LO = PRESSUR(1)                                         RNPP3720  66100000
C                                                                           RNPP3730  66200000
C                                                                           RNPP3740  66300000
C              WRITE INTEGRATED VALUES TO RANDOM ACCESS                     RNPP3750  66400000
      CALL WRITMS (20,PRESSUR(1),NPOIN,NREC)                                RNPP3760  66500000
      NREC = NREC+1                                                         RNPP3770  66600000
C         .............................................................     RNPP3780  66700000
C                                                                           RNPP3790  66800000
```

```
      CALL READMS (20,SPECTRA(1,2),NWORDS,JREC)
      JREC = JREC+1
C
C         .....LOOP TO INTEGRATE AND INTERPOLATE PHASE DATA  .....
      DO 160 J=1,NPOIN
      IW = -1
C             USE SPECTRUM DATA TO CONSTRUCT INTEGRATION
C             (OR INTERPOLATION) NODE POINTS
      DO 140 K=1,MTRACKS
      T(K) = SPECTRA(J,K)
  140 CONTINUE
C
C             INTERPOLATE FOR INTP EQUALLY SPACED POINTS OVER THE
C             PHASE DISTRIBUTION
      CALL CSIUNI (16,MTRACKS,1,51,INTP,CP,T,XINT,YINT,IW,WK,IERR)
C
C             AVERAGE INTERPOLATED VALUES
      SUMINT = 0.0
      DO 145 I=1,INTP
      SUMINT = SUMINT + YINT(I)
  145 CONTINUE
      PHASE(J) = SUMINT / FLOAT(INTP)
C
  160 CONTINUE
C
C             WRITE INTEGRATED VALUES TO RANDOM ACCESS
      CALL WRITMS (20,PHASE(1),NPOIN,MREC)
      MREC = MREC+1
C         .........................................................
C
C             WRITE INTEGRATED BLADE LOADING HARMONIC AND
C             INTEGRATED PHASE SPECTRUM TO DISC FOR LATER USE
C             BY THE PLOTTING PROGRAM SPLPLT
      IF (NR .EQ. 1) WRITE (4) NREAD,MPOIN,DELTF,LO,PRCNTGR,RS,THRUST
      WRITE (4) (PRESSUR(I),PHASE(I),I=1,NPOIN)
C
  190 CONTINUE
C .....................................................................
C
C
C             SET TRACK NUMBER (WITH AND WITHOUT INTEGRATION)
  200 J = ITRACK
      IF (NTEGRAT .NE. 0) J=1
C
C             WRITE STEADY LOADING AND HEADING INFORMATION
      WRITE (6,2010) LO
 2010 FORMAT (1H1,//,35X,*LOADING HARMONIC COEFFICIENTS AND PHASE ANGLES
     +*,//,20X,*THE STEADY LOADING IS = *,F12.5,* PSI*,//)
      WRITE (6,2020)
 2020 FORMAT (/,10X,*HARMONIC NUMBER*,7X,*PRESSURE*,15X,*ALPHA*,17X,
     +*PHASE*,//)
C
C             SET RANDOM ACCESS RECORD LOCATION AND SIZE COUNTERS
C             DEPENDING ON WHETHER INTEGRATION IS PERFORMED
      IF (NTEGRAT .NE. 0) GO TO 250
      IREC = NRCSUM
      JREC = IREC+NREAD
      GO TO 270
  250 IREC = NRVSUM
      JREC = IREC+NREAD
      NPOIN = 250
      NWORDS = 250
      MWORDS = MPOIN
C
C .....................................................................
C                     COMPUTE BLADE LOADING HARMONICS
C
C             INITIALIZE DATA POINT COUNTER, BLH COUNTER, AND TEST
C             PARAMETER
  270 II = 0
      I = 0
      NPP = 0
      IITEST = 0
C
```

```
C
C     ..........LOOP FOR THE NUMBER OF RANDOM ACCESS READS..........
      DO 400 NR=1,NREAD
C
C               SET RECORD SIZE FOR LAST READ (WITH AND WITHOUT
C               INTEGRATION
      IF (NR .EQ. NREAD) NPOIN=MPOIN
      IF (NR .EQ. NREAD) NWORDS=MWORDS
C
C               READ PRESSURE DATA FROM RANDOM ACCESS
      CALL READMS (20,SPECTRA(1,1),NWORDS,IREC)
      IREC = IREC+1
C               USING THE DESIRED TRACK, DETERMINE PRESSURES
      DO 290 NP=1,NPOIN
      PRESSUR(NP) = SPECTRA(NP,J)
  290 CONTINUE
C               SET FIRST PRESSURE TO STEADY LOADING
      IF (NR .EQ. 1) PRESSUR(1) = ABS(LO)
C
C               READ PHASE DATA FROM RANDOM ACCESS
      CALL READMS (20,SPECTRA(1,1),NWORDS,JREC)
      JREC = JREC+1
C               USING THE DESIRED TRACK DETERMINE PHASES
      DO 310 NP=1,NPOIN
      PHASE(NP) = SPECTRA(NP,J)
  310 CONTINUE
C               SET FIRST PHASE TO ZERO
      IF (NR .EQ. 1) PHASE(1)=0.0
C
C
      NP = 0
C
C               TEST TO DETERMINE IF NEW R.A. RECORD IS NEEDED
  320 IF ((NP+1) .GT. NPOIN) GO TO 400
C               TEST TO DETERMINE IF NEW FREQUENCY IS NEEDED
      IF (NP .EQ. 0 .AND. IITEST .EQ. NR) GO TO 330
      I = I+1
C               COMPUTE MULTPLE OF BLADE PASSAGE FREQUENCY
      FREQ = (I-1)*F
C
C               TEST TO DETERMINE IF NEW R.A. RECORD IS NEEDED
  330 IF ((NP+1) .LE. NPOIN) GO TO 350
C               SET NEW R.A. RECORD TEST PARAMETER
      IITEST = NR+1
      GO TO 400
C
C               INCREMENT COUNTERS
  350 NP = NP+1
      II = II+1
C               TEST FOR COMPLETION OF BLH COMPUTATIONS
      IF (I .GT. NN .OR. II .GT. NSPCT) GO TO 410
C
C               SELECT DATA AT A MULTIPE OF BLADE PASSAGE FREQUENCY
C               IF IT IS NOT THE SAME AS THE BAND-WIDTH OF FOURIER
C               TRANSFORM ANALYSIS (BLH COEFFICIENTS)
      DIFF = FREQ - (II-1)*DELTF
      ADIFF = ABS(DIFF)
      IF (ADIFF .GT. DELTF2) GO TO 330
C
      NPP = NPP+1
      REPRESS(NPP) = PRESSUR(NP)
      IF (NR .EQ. 1 .AND. NPP .EQ. 1) REPRESS(1)=LO
      REPHASE(NPP) = PHASE(NP)/RADIAN
      ALPHA(NPP) = ABS (REPRESS(NPP)/LO)
C
      LASPNT = NPP
      GO TO 320
C
  400 CONTINUE
C
      GO TO 410
C ...................................................................
C
C               TEST FOR OPTIONAL BLH COEFFICIENT AND PHASE
```

APPENDIX G

```
C               ANGLE INPUT
  402 WRITE (6,2070)
 2070 FORMAT (1H1,//,35X,*LOADING HARMONIC COEFFICIENTS AND PHASE ANGLES
     +*,/,24X,*(COMPUTED USING THE INLINE FUNCTION BLH(Y) = C/Y..X)*,//)
      WRITE (6,2020)
      DO 405 M=1,NBLHPT
      FM = FLOAT(M)
      REPRESS(M) = BLH(FM)
      REPHASE(M) = 0.0
      ALPHA(M) = REPRESS(M)
  405 CONTINUE
      LASPNT = NBLHPT
C
C               WRITE BLADE LOADING HARMONIC COEFFICIENTS
  410 WRITE (6,2060) (I,REPRESS(I),ALPHA(I),REPHASE(I),I=1,LASPNT,INCOF)
 2060 FORMAT (12X,I5,3(10X,F12.5))
C
C
C               WRITE SPL HEADINGS
  420 WRITE (6,2030) OBSELV,OBSAZI
 2030 FORMAT (1H1,//,40X,*THE SOUND PRESSURE LEVELS AT VARIOUS MB NUMBER
     1S*,///25X,*OBSERVER ELEVATION ANGLE = *,F6.3,15X,*OBSERVER AZIMUTH
     2 ANGLE = *,F6.3,//,24X,*MB*,14X,*SPL*,18X,*LIMIT*,/)
C
C   ..................................................................
C                         COMPUTE SOUND PRESSURE LEVELS
C
C               INITIALIZE *OASPL* PARAMETER
      OVERALL = 0.0
C
C     .................LOOP FOR THE NUMBER OF HARMONICS.............
      DO 580 M=1,MAXHAR
C
C               COMPUTE MB NUMBERS
      MB = M*NB
      AMBN(M) = FLOAT(MB)
C               SELECT DESIRED ARGUMENT AND ORDER OF BESSEL
C               FUNCTION
      ARG = M*S
      ASKFOR=ARG+100.
C
      CALL BSSLS(ARG,BJ,ASKFOR)
C
C          ...................................................
C                    COMPUTE SPL USING WRIGHT SUMMATION
C               (REAL AND IMAGINARY PARTS ARE ADDED SEPERATELY
C                AND MAGNITUDE YIELDS SPL)
C
      LMIT = 10. + 1.25*ARG
      ALIMIT(M) = FLOAT(LMIT)
C
C
      TERM1 = CEXP (-EI*REPHASE(MB+1))
      FIRST=.5*ALPHA(MB+1)*KT*BJ(1)*MB*TERM1*(-EI)
      FIRST1=REAL(FIRST)
      FIRST2=AIMAG(FIRST)
      THETA = OBSAZI
      TERM0 = CEXP (EI * (MB*THETA-REPHASE(1)) )
      ZERO=.5*ALPHA(1)*(KT-KQ)*BJ(MB+1)*MB*TERM0*(-EI**(MB+1))
      ZERO1=REAL(ZERO)
      ZERO2=AIMAG(ZERO)
      TOTAL1 = 0.
      TOTAL2=0.
C
      DO 500 Q=1,LMIT
C
      QTHETA = Q*THETA
      KQQMB  = KQ*Q/MB
      MBBJ = MB*BJ(Q+1)
      EIQ1 = -EI**(Q+1)
      MBQ1 = MB+Q+1
C
      TERM2 = CEXP (-EI * (QTHETA-REPHASE(MBQ1)) )
      SECOND = .5*ALPHA(MBQ1) * (KT+KQQMB)*MBBJ*TERM2*EIQ1
      SECO1=REAL(SECOND)
      SECO2=AIMAG(SECOND)
C
      STEP = 0
      IF (Q .GE. MB) STEP=2
C
      SUB=IABS(Q-MB)+1
C
      TERM3 = CEXP (EI * (QTHETA + REPHASE(SUB)*(STEP-1.)) )
      THIRD = .5*ALPHA(SUB) * (KT-KQQMB)*MBBJ*TERM3*EIQ1
      THIRD1=REAL(THIRD)
      THIRD2=AIMAG(THIRD)
      TOTAL1=TOTAL1+SECO1+THIRD1
      TOTAL2=TOTAL2+SECO2+THIRD2
C
  500 CONTINUE
C
      NSPLPT = M
C
C              TOTAL REAL AND IMAGINARY PARTS
      TOTALR=TOTAL1+ZERO1+FIRST1
      TOTALI=TOTAL2+ZERO2+FIRST2
C
C              TAKE MAGNITUDE
      TOTAL = SQRT ((TOTALR*TOTALR)+(TOTALI*TOTALI))
      TOTAL = TOTAL/SQRT2
C
C              APPLY APPROPRIATE CHORD DISTRIBUTION FUNCTION TO SPL
      IF (ICHORD .EQ. 0) GO TO 520
      IF (ICHORD .EQ. 1) TOTAL = TOTAL*CHORD1(M)
      IF (ICHORD .EQ. 2) TOTAL = TOTAL*CHORD2(M)
      IF (ICHORD .EQ. 3) TOTAL = TOTAL*CHORD3(M)
      IF (ICHORD .EQ. 4) TOTAL = TOTAL*CHORD4(M)
C
  520 IF (TOTAL .NE. 0.0) GO TO 550
      SPL(M) =  0.0
      GO TO 590
C
C              COMPUTE SOUND PRESSURE LEVELS
  550 SPL(M) = 20.*ALOG10 (ABS(TOTAL)/PO)
C              TOTAL OASPL PARAMETER
      OVERALL = OVERALL + TOTAL*TOTAL
C
  580 CONTINUE
C     ...........................................................
C
C              WRITE SOUND PRESSURE LEVELS
  590 WRITE (6,2040) (AMBN(I),SPL(I),ALIMIT(I),I=1,NSPLPT)
 2040 FORMAT (19X,F6.1,10X,F10.3,10X,F12.1)
C
C              TAKE ROOT MEAN SQUARE
      RMS = SQRT(OVERALL)
C              EVALUATE OASPL
      OASPL=20.*ALOG10(RMS/PO)
C
C              WRITE OASPL
      WRITE (6,2050) OASPL
 2050 FORMAT (//,20X,*OASPL = *,F6.1,* DB*)
C
      TRACK = FLOAT(ITRACK)
C              WRITE  SOUND PRESSURE LEVEL  PLOTTING PARAMETERS
C              TO DISC FOR LATER USE BY THE PLOTTING PROGRAM SPLPLT
      WRITE (4) NB,F,RS,OASPL,THRUST,OBSELV,OBSAZI,TRACK,PRCNTGR,ICHORD
      WRITE (4) MAXHAR,(SPL(M),M=1,MAXHAR)
C
  777 CONTINUE
C.......................................................................
C
C              RETURN TO DETERMINE IF ANOTHER CASE IS TO BE RUN
      GO TO 10
C
  999 CONTINUE
      STOP
      END
      SUBROUTINE CSIUNI(MNPTS,N,NCVS,MMAX,M,X,Y,T,F,IW,WK,IERR)
      DIMENSION F(MMAX,NCVS)
      DIMENSION WK(1)
      DIMENSION X(N),Y(MNPTS,NCVS),T(M)
      JK=N*NCVS
      IERR=0
      IEXP=0
      IT=0
      IS=0
      IP=0
      ID=0
      IM=1
      IN=1
C
C     WK(1 TO N)          H
C
C     WK(N+1 TO JK+N)           IDLY
C
C     WK(JK+N+1 TO JK+2N)       ICC
C
C     WK(JK+2N+1 TO JK+3N)      IH2
C
C     WK(JK+3N+1 TO JK+4N)      IDSQ
C
C     WK(JK+4N+1 TO JK+5N)      IDD
C
C     WK(JK+5N+1 TO 2JK+5N)     IS2
C
C     WK(2JK+5N+1 TO 3JK+5N)  IS3
C     WK(3JK+5N+1 TO 3JK+6N)  IWW
C
C     WK(3JK+6N+1 TO 3JK+7N)  IGG
C
C     WK(3JK+7N+1 TO 3JK+8N)  ISV
C
C     DIMENSION WK (3JK+8N)
C
      IDLY=N+1
      ICC=JK+N+1
      IH2=JK+2*N+1
      IDSQ=JK+3*N+1
      IDD=JK+4*N+1
      IS2=JK+5*N+1
      IS3=2*JK+5*N+1
      IWW=3*JK+5*N+1
      IGG=3*JK+6*N+1
      ISV=3*JK+7*N+1
      IF(IW-(-1)) 15,2,15
    2 N1=N-1
C
C     IS INDEPENDENT VARIABLE ARRAY INCREASING
C
      DO 118 I=1,N1
      WK(I)=X(I+1)-X(I)
      IF(WK(I)) 119,119,118
  119 WRITE(6,122) I,X(I),X(I+1)
  122 FORMAT(1H06X64HINDEPENDENT VARIABLE ARRAY NOT INCREASING IN CSIUNI
     1 AT POSITION I4 ,2X2HX=2F10.4/)
      IERR=1
      RETURN
  118 CONTINUE
      IW=1
      DO 101 L=1,NCVS
    3 DO 51 I=1,N1
      I1=I+1
      IM1=I-1
      WK(IDLY+IP)=(Y(I1,L)-Y(I,L))/WK(I)
      IP=IP+1
      WK(ICC+IM1)=WK(I)
   51 CONTINUE
      DO 65 I=2,N1
      IM1=I-1
      WK(IH2+IM1)=(WK(IM1)+WK(I))*2.
      WK(IDSQ+IM1)=(WK(ID LY+IP)-WK(ID LY+IM-1))*6
      IM=IM+1                                                           CSIU0750 103900000
   65 CONTINUE                                                          CSIU0760 104000000
  222 CONTINUE                                                          CSIU0770 104100000
      WK(IH2)=1.                                                        CSIU0780 104200000
      WK(IDSQ-1)=1.                                                     CSIU0790 104300000
      WK(ICC)=0.                                                        CSIU0800 104400000
      WK(N1)=0.                                                         CSIU0810 104500000
      WK(IDSQ)=0.                                                       CSIU0820 104600000
      WK(IDD-1)=0.                                                      CSIU0830 104700000
  223 CONTINUE                                                          CSIU0840 104800000
C                                                                       CSIU0850 104900000
C THIS ROUTINE SOLVES THE TRIDIAGONAL (EXCEPT TWO ELEMENTS)   MATRIX    CSIU0860 105000000
C                                                                       CSIU0870 105100000
      IIP=ISV-1                                                         CSIU0880 105200000
      WK(IWW)=WK(IH2)                                                   CSIU0890 105300000
      WK(ISV)=WK(ICC)/WK(IH2)                                           CSIU0900 105400000
      WK(IGG)=WK(IDSQ)/WK(IWW)                                          CSIU0910 105500000
      DO 100 K=2,N                                                      CSIU0920 105600000
      KM2=K-2                                                           CSIU0930 105700000
      KM1 = K-1                                                         CSIU0940 105800000
      WK(IWW+KM1)=WK(IH2+KM1)-WK(KM1)*WK(ISV+KM2)                       CSIU0950 105900000
      IF (K.EQ.N)    GO TO 5                                            CSIU0960 106000000
    4 WK(ISV+KM1)=WK(ICC+KM1)/WK(IWW+KM1)                               CSIU0970 106100000
    5 WK(IGG+KM1)=(WK(IDSQ+KM1)-WK(KM1)*WK(IGG+KM2))/WK(IWW+KM1)        CSIU0980 106200000
  100 CONTINUE                                                          CSIU0990 106300000
      WK(IS2-1)=WK(ISV-1)                                               CSIU1000 106400000
      IBW=IS2-1                                                         CSIU1010 106500000
      DO 200 K=1,N1                                                     CSIU1020 106600000
      IBC=IBW-K                                                         CSIU1030 106700000
      KK= N-K                                                           CSIU1040 106800000
      KKM1=KK-1                                                         CSIU1050 106900000
      WK(IBC)=WK(IGG+KKM1)-WK(ISV+KKM1)*WK(IBC+1)                       CSIU1060 107000000
  200 CONTINUE                                                          CSIU1070 107100000
C                                                                       CSIU1080 107200000
C                                                                       CSIU1090 107300000
      DO 66 I=1,N                                                       CSIU1100 107400000
      IM1=I-1                                                           CSIU1110 107500000
      WK(IS2+ID)=WK(IDD+IM1)                                            CSIU1120 107600000
      ID=ID+1                                                           CSIU1130 107700000
   66 CONTINUE                                                          CSIU1140 107800000
      WK(N1)=WK(IH2-2)                                                  CSIU1150 107900000
   14 DO 53 I=1,N1                                                      CSIU1160 108000000
      II=I+1                                                            CSIU1170 108100000
      WK(IS3+IS)=(WK(IS2+IN)-WK(IS2+IN-1))/WK(I)                        CSIU1180 108200000
      IS=IS+1                                                           CSIU1190 108300000
      IN=IN+1                                                           CSIU1200 108400000
   53 CONTINUE                                                          CSIU1210 108500000
      IM=IM+2                                                           CSIU1220 108600000
      IP=IP+1                                                           CSIU1230 108700000
      IN=IN+1                                                           CSIU1240 108800000
      IS=IS+1                                                           CSIU1250 108900000
  101 CONTINUE                                                          CSIU1260 109000000
   15 CONTINUE                                                          CSIU1270 109100000
      IF(IW .EQ.0) IW=1                                                 CSIU1280 109200000
  104 J=0                                                               CSIU1290 109300000
  105 J=J+1                                                             CSIU1300 109400000
   16 I=IW                                                              CSIU1310 109500000
   54 IF(T(J)-X(1)) 58,117,55                                           CSIU1320 109600000
  117 I=1                                                               CSIU1330 109700000
      IW=1                                                              CSIU1340 109800000
      GO TO 17                                                          CSIU1350 109900000
   55 IF(T(J)-X(N)) 162,59,158                                          CSIU1360 110000000
  162 IF(T(J)-X(I)) 160,217,57                                          CSIU1370 110100000
  160 I=I-1                                                             CSIU1380 110200000
      GO TO 162                                                         CSIU1390 110300000
   56 IF(T(J)-X(I)) 60,217,57                                           CSIU1400 110400000
   57 I=I+1                                                             CSIU1410 110500000
      GO TO 56                                                          CSIU1420 110600000
   58 CONTINUE                                                          CSIU1430 110700000
C                                                                       CSIU1440 110800000
C     EXTRAPOLATE LOWER END                                             CSIU1450 110900000
C                                                                       CSIU1460 111000000
      I=1                                                               CSIU1470 111100000
      IW=0                                                              CSIU1480 111200000
      TSAV=T(J)                                                         CSIU1490 111300000
      T(J)=X(1)                                                         CSIU1500 111400000
      IEXP=1                                                            CSIU1510 111500000
      GO TO 17                                                          CSIU1520 111600000
C                                                                       CSIU1530 111700000
C     EXTRAPOLATE UPPER END                                             CSIU1540 111800000
C                                                                       CSIU1550 111900000
  158 I=IW=N                                                            CSIU1560 112000000
      I=I-1                                                             CSIU1570 112100000
      TSAV=T(J)                                                         CSIU1580 112200000
      T(J)=X(N)                                                         CSIU1590 112300000
      IEXP=2                                                            CSIU1600 112400000
      GO TO 17                                                          CSIU1610 112500000
   59 I=N                                                               CSIU1620 112600000
   60 CONTINUE                                                          CSIU1630 112700000
      I=I-1                                                             CSIU1640 112800000
  217 IW=I                                                              CSIU1650 112900000
   17 CONTINUE                                                          CSIU1660 113000000
      IM1=I-1                                                           CSIU1670 113100000
      ITD=IDLY+IM1                                                      CSIU1680 113200000
      IT2=IS2+IM1                                                       CSIU1690 113300000
      IT3=IS3+IM1                                                       CSIU1700 113400000
      DO 110 K=1,NCVS                                                   CSIU1710 113500000
      HT1=T(J)-X(I)                                                     CSIU1720 113600000
      II=I+1                                                            CSIU1730 113700000
      HT2=T(J)-X(II)                                                    CSIU1740 113800000
      PROD=HT1 * HT2                                                    CSIU1750 113900000
      DER2=WK(IT2)+HT1*WK(IT3)                                          CSIU1760 114000000
      DELSQS=(WK(IT2)+WK(IT2+1)+DER2)/6.                                CSIU1770 114100000
      F(J,K)=Y(I,K)+HT1*WK(ITD)+PROD*DELSQS                             CSIU178C 114200000
      IF(IEXP-1) 1000,1001,1002                                         CSIU1790 114300000
 1001 CONTINUE                                                          CSIU1800 114400000
      DER1      =WK(ITD)+(HT1+HT2)*DELSQS+PROD*WK(IT3)/6.               CSIU1810 114500000
      F(J,K)=F(J,K)+(TSAV-X(1))*DER1                                    CSIU1820 114600000
      GO TO 1000                                                        CSIU1830 114700000
 1002 CONTINUE                                                          CSIU1840 114800000
      DER1      =WK(ITD)+(HT1+HT2)*DELSQS+PROD*WK(IT3)/6.               CSIU1850 114900000
      F(J,K)=F(J,K)+(TSAV-X(N))*DER1                                    CSIU1860 115000000
 1000 CONTINUE                                                          CSIU1870 115100C00
      IT3=IT3+N                                                         CSIU1880 115200000
      IT2=IT2+N                                                         CSIU1890 115300000
      ITD=ITD+N                                                         CSIU1900 115400000
  110 CONTINUE                                                          CSIU1910 115500000
      IF(IEXP .NE. 0) T(J)=TSAV                                         CSIU1920 115600000
      IEXP=0                                                            CSIU1930 115700000
      IF(J.LT.M)GO TO 105                                               CSIU1940 115800000
      RETURN                                                            CSIU1950 115900000
      END                                                               CSIU1960 116000000
      SUBROUTINE SPLS(MNPTS,N,NCVS,X,Y,KI,KJ,PROXIN,IW,WK,IERR)         SPLS0010 116100000
      DIMENSION PROXIN(NCVS)                                            SPLS0020 116200000
      DIMENSION X(N),Y(MNPTS,NCVS)                                      SPLS0030 116300000
      DIMENSION WK(1)                                                   SPLS0040 116400000
      IERR=0                                                            SPLS0050 116500000
      JK=N*NCVS                                                         SPLS0060 116600000
      JK2=JK*2                                                          SPLS0070 116700000
      IT=0                                                              SPLS0080 116800000
      IP=0                                                              SPLS0090 116900000
      ID=0                                                              SPLS0100 117000000
      IM=1                                                              SPLS0110 117100000
C                                                                       SPLS0120 117200000
C     WK(1 TO N)            H                                           SPLS0130 117300000
C                                                                       SPLS0140 117400000
C     WK(N+1 TO JK+N)               IDLY                                SPLS0150 117500000
C                                                                       SPLS0160 117600000
C     WK(JK+N+1 TO JK+2N)           ICC                                 SPLS0170 117700000
C                                                                       SPLS0180 117800000
C     WK(JK+2N+1 TO JK+3N)          IE2                                 SPLS0190 117900000
C                                                                       SPLS0200 118000000
C     WK(JK+3N+1 TO JK+4N)          IDSQ                                SPLS0210 118100000
C                                                                       SPLS0220 118200000
C     WK(JK+4N+1 TO JK+5N)          ICD                                 SPLS0230 118300000
C                                                                       SPLS0240 118400000
C     WK(JK+5N+1 TO 2JK+5N)         IS2                                 SPLS0250 118500000
C                                                                       SPLS0260 118600000
```

```
C     WK(2JK+5N+1 TO 2JK+6N) IWW                                        SPLS0270 118700000
C                                                                       SPLS0280 118800000
C     WK(2JK+6N+1 TO 2JK+7N) IGG                                        SPLS0290 118900000
C                                                                       SPLS0300 119000000
C     WK(2JK+7N+1 TO 2JK+8N) ISV                                        SPLS0310 119100000
C                                                                       SPLS0320 119200000
C     DIMENSION WK(2JK+8N)                                              SPLS0330 119300000
C                                                                       SPLS0340 119400000
      IDLY=N+1                                                          SPLS0350 119500000
      ICC=JK+N+1                                                        SPLS0360 119600000
      IH2=JK+2*N+1                                                      SPLS0370 119700000
      IDSQ=JK+3*N+1                                                     SPLS0380 119800000
      IDD=JK+4*N+1                                                      SPLS0390 119900000
      IS2=JK+5*N+1                                                      SPLS0400 120000000
      IWW=JK2+5*N+1                                                     SPLS0410 120100000
      IGG=JK2+6*N+1                                                     SPLS0420 120200000
      ISV=JK2+7*N+1                                                     SPLS0430 120300000
      IF(IW-(-1)) 15,2,15                                               SPLS0440 120400000
    2 N1=N-1                                                            SPLS0450 120500000
      DO 118 I=1,N1                                                     SPLS0460 120600000
      WK(I)=X(I+1)-X(I)                                                 SPLS0470 120700000
      IF(WK(I))119,119,118                                              SPLS0480 120800000
  119 WRITE(6,122)I,X(I),X(I+1)                                         SPLS0490 120900000
  122 FORMAT(1H06X62HINDEPENDENT VARIABLE ARRAY NOT INCREASING IN SPLS ASPLS0500 121000000
     1T POSITION I4,2X2HX=2F10.4)                                       SPLS0510 121100000
      IERR=1                                                            SPLS0520 121200000
      RETURN                                                            SPLS0530 121300000
  118 CONTINUE                                                          SPLS0540 121400000
      IW=1                                                              SPLS0550 121500000
      DO 101 L=1,NCVS                                                   SPLS0560 121600000
    3 DO 51 I=1,N1                                                      SPLS0570 121700000
      II=I+1                                                            SPLS0580 121800000
      IM1=I-1                                                           SPLS0590 121900000
      WK(IDLY+IP)=(Y(II,L)-Y(I,L))/WK(I)                                SPLS0600 122000000
      IP=IP+1                                                           SPLS0610 122100000
      WK(ICC+IM1)=WK(I)                                                 SPLS0620 122200000
   51 CONTINUE                                                          SPLS0630 122300000
      DO 65 I=2,N1                                                      SPLS0640 122400000
      IM1=I-1                                                           SPLS0650 122500000
      WK(IH2+IM1)=(WK(IM1)+WK(I))*2.                                    SPLS0660 122600000
      WK(IDSQ+IM1)=(WK(ID LY+IM)-WK(ID LY+IM-1))*6                      SPLS0670 122700000
      IM=IM+1                                                           SPLS0680 122800000
   65 CONTINUE                                                          SPLS0690 122900000
  222 CONTINUE                                                          SPLS0700 123000000
      WK(IH2)=1.                                                        SPLS0710 123100000
      WK(IDSQ-1)=1.                                                     SPLS0720 123200000
      WK(ICC)=0.                                                        SPLS0730 123300000
      WK(N1)=0.                                                         SPLS0740 123400000
      WK(IDSQ)=0.                                                       SPLS0750 123500000
      WK(IDD-1)=0.                                                      SPLS0760 123600000
  223 CONTINUE                                                          SPLS0770 123700000
C                                                                       SPLS0780 123800000
C THIS ROUTINE SOLVES THE TRIDIAGONAL (EXCEPT TWO ELEMENTS)   MATRIX    SPLS0790 123900000
C                                                                       SPLS0800 124000000
      IIP=ISV-1                                                         SPLS0810 124100000
      WK(IWW)=WK(IH2)                                                   SPLS0820 124200000
      WK(ISV)=WK(ICC)/WK(IH2)                                           SPLS0830 124300000
      WK(IGG)=WK(IDSQ)/WK(IWW)                                          SPLS0840 124400000
      NM1=N-1                                                           SPLS0850 124500000
      DO 100 K=2,N                                                      SPLS0860 124600000
      KM2=K-2                                                           SPLS0870 124700000
      KM1 = K-1                                                         SPLS0880 124800000
      WK(IWW+KM1)=WK(IH2+KM1)-WK(KM1)*WK(ISV+KM2)                       SPLS0890 124900000
      IF (K.EQ.N)    GO TO 5                                            SPLS0900 125000000
    4 WK(ISV+KM1)=WK(ICC+KM1)/WK(IWW+KM1)                               SPLS0910 125100000
    5 WK(IGG+KM1)=(WK(IDSQ+KM1)-WK(KM1)*WK(IGG+KM2))/WK(IWW+KM1)        SPLS0920 125200000
  100 CONTINUE                                                          SPLS0930 125300000
      WK(IS2-1)=WK(ISV-1)                                               SPLS0940 125400000
      IBW=IS2-1                                                         SPLS0950 125500000
      DO 200 K=1,N1                                                     SPLS0960 125600000
      IBC=IBW-K                                                         SPLS0970 125700000
      KK= N-K                                                           SPLS0980 125800000
      KKM1=KK-1                                                         SPLS0990 125900000
      WK(IBC)=WK(IGG+KKM1)-WK(ISV+KKM1)*WK(IBC+1)                       SPLS1000 126000000
```

APPENDIX G

```
  200 CONTINUE                                                          SPLS1010 126100000
C                                                                       SPLS1020 126200000
C                                                                       SPLS1030 126300000
      DO 66 I=1,N                                                       SPLS1040 126400000
      IM1=I-1                                                           SPLS1050 126500000
      WK(IS2+ID)=WK(IDD+IM1)                                            SPLS1060 126600000
      ID=ID+1                                                           SPLS1070 126700000
   66 CONTINUE                                                          SPLS1080 126800000
      WK(N1)=WK(IM2-2)                                                  SPLS1090 126900000
      IM=IM+2                                                           SPLS1100 127000000
      IP=IP+1                                                           SPLS1110 127100000
  101 CONTINUE                                                          SPLS1120 127200000
   15 CONTINUE                                                          SPLS1130 127300000
      II2=IS2                                                           SPLS1140 127400000
      KJM=KJ-1                                                          SPLS1150 127500000
      DO 120 K=1,NCVS                                                   SPLS1160 127600000
      PROXIN(K)=0.                                                      SPLS1170 127700000
      DO 62 I=K1,KJM                                                    SPLS1180 127800000
      I1=I+1                                                            SPLS1190 127900000
      I22=II2+I-1                                                       SPLS1200 128000000
      PROXIN(K)=PROXIN(K)+.5*WK(I)*(Y(I,K)+Y(I1,K))-WK(I)**3            SPLS1210 128100000
     1*(WK(I22)+WK(I22+1))/24.                                          SPLS1220 128200000
   62 CONTINUE                                                          SPLS1230 128300000
      II2=IS2+(N*K)                                                     SPLS1240 128400000
  120 CONTINUE                                                          SPLS1250 128500000
      RETURN                                                            SPLS1260 128600000
      END                                                               SPLS1270 128700000
      SUBROUTINE BSSLS (X,F,N)                                          BSSL0010 128800000
C     THIS SUBROUTINE WAS ORIGINALLY THE SYSTEM SUBROUTINE BSSLS/TAS    BSSL0020 128900000
C 8/26/66 BUT HAS SINCE BEEN MODIFIED TO ALLOW BESSEL FUNCTIONS OF      BSSL0030 129000000
C ORDER GREATER THAN 30 TO BE CCMPUTED.                                 BSSL0040 129100000
      DIMENSION F(1)                                                    BSSL0050 129200000
      COMMON/FIX/NPR,NP,NPP                                             BSSL0060 129300000
      DO 1 I=1,1000                                                     BSSL0080 129400000
      F(I)=0.                                                           BSSL0090 129500000
    1 CONTINUE                                                          BSSL0100 129600000
      NP = N+28                                                         BSSL0110 129700000
      NPR =1.5*X+28                                                     BSSL0120 129800000
      IF(NP.LT.NPR)NP=NPR                                               BSSL0130 129900000
      IF(X.EQ.0.0) GO TO 19                                             BSSL0140 130000000
      F(NP)= 0.0                                                        BSSL0150 130100000
      F(NP-1)=.1E-99                                                    BSSL0160 130200000
      NPR=NP - 2                                                        BSSL0170 130300000
      NPP=NP - 1                                                        BSSL0180 130400000
      DO 11 I=1,NPR                                                     BSSL0190 130500000
      NP = NPP - I                                                      BSSL0200 130600000
      XN = NP                                                           BSSL0210 130700000
   11 F(NP) =  2.0*XN/X*F(NP+1)-F(NP+2)                                 BSSL0220 130800000
      XN = F(1)                                                         BSSL0230 130900000
      DO 7 I = 3,NPP,2                                                  BSSL0240 131000000
    7 XN = 2.0 *F(I)+XN                                                 BSSL0250 131100000
      XN = 1./XN                                                        BSSL0260 131200000
      DO 8 I = 1,NPP                                                    BSSL0270 131300000
      F(I)=XN*F(I)                                                      BSSL0280 131400000
    8 CONTINUE                                                          BSSL0290 131500000
      RETURN                                                            BSSL0310 131600000
   19 F(1)= 1.000                                                       BSSL0320 131700000
      DO  1831 I = 2,NP                                                 BSSL0330 131800000
 1831 F(I) = 0.0000                                                     BSSL0340 131900000
      RETURN                                                            BSSL0360 132000000
      END                                                               BSSL0370 132100000
      PROGRAM SPLPLT (INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,    SPLP0010 132200000
     +TAPE4=1001)                                                       SPLP0020 132300000
C***********************************************************************SPLP0030 132400000
C*              PURPOSE                                                *SPLP0040 132500000
C*                 TO PLOT (CALCOMP) THE SOUND PRESSURE LEVEL (SPL)    *SPLP0050 132600000
C*                 DATA GENERATED BY PROGRAM RNPPE4.  ONE PLOT OF      *SPLP0060 132700000
C*                 SPL .VS. FREQUENCY IS PRODUCED FOR EACH SET OF SPL  *SPLP0070 132800000
C*                 DATA GENERATED BY PROGRAM RNPPE4.  IN ADDITION IF   *SPLP0080 132900000
C*                 INTEGRATION OF THE STEADY LOADING DISTRIBUTION ALONG*SPLP0090 133000000
C*                 ANY LENGTH OF THE CHORD WAS PERFORMED, A PLOT OF    *SPLP0100 133100000
C*                 THE INTEGRATED BLADE LOADING COEFFICIENTS (AND AN   *SPLP0110 133200000
C*                 OPTIONAL PLOT CF THE INTEGRATED PHASE SPECTRUM) WILL*SPLP0120 133300000
C*                 BE PRODUCED.                                        *SPLP0130 133400000
```

```
C*                                                               *SPLP0140 133500000
C*         NAMELIST INPUT PARAMETERS                             *SPLP0150 133600000
C*                        NAMELIST FIXED                         *SPLP0160 133700000
C*            NSPL     - NUMBER CF SPL PLCTS TO BE PRODUCED FOR  *SPLP0170 133800000
C*                       GIVEN DATA                              *SPLP0180 133900000
C*                       (NSPL=6 IS DEFAULT VALUE)               *SPLP0190 134000000
C*            XMIN     - FREQUENCY RANGE OF AVAILABLE FLIGHT POINT *SPLP0200 134100000
C*                       (XMIN=0.0 IS DEFAULT VALUE)             *SPLP0210 134200000
C*            XMAX       LOADING DATA (HZ.) IS GIVEN BY XMAX-XMIN *SPLP0220 134300000
C*                       (XMAX=1000.C IS DEFAULT VALUE)          *SPLP0230 134400000
C*            YMIN     - MINUMUM SOUND PRESSURE LEVEL (DB)       *SPLP0240 134500000
C*                       (YMIN=10.0 IS DEFAULT VALUE)            *SPLP0250 134600000
C*            YSCALE   - SPL AXIS (Y-AXIS) SCALE FACTOR          *SPLP0260 134700000
C*                       (YSCALE=10.0 IS DEFAULT VALUE)          *SPLP0270 134800000
C*            RUN      - RUN NUMBER                              *SPLP0280 134900000
C*                                                               *SPLP0290 135000000
C*                        NAMELIST VARIABLE                      *SPLP0300 135100000
C*            NNPLOT   - 0 NO INTEGRATED BLADE LOADING COEFFICIENT OR*SPLP0310 135200000
C*                          INTEGRATED PHASE PLOTS WILL BE GENERATED *SPLP0320 135300000
C*                          FOR THIS SET CF SPL VALUES.          *SPLP0330 135400000
C*                       1 AN INTEGRATED BLADE LOADING COEFFICIENT *SPLP0340 135500000
C*                          PLCT IS GENERATED FOR THIS SET SPL VALUES *SPLP0350 135600000
C*                       2 BOTH AN INTEGRATED BLH COEFFICIENT AND *SPLP0360 135700000
C*                          PHASE PLOT IS GENERATED FOR THIS SET *SPLP0370 135800000
C*                          OF SPL VALUES.                       *SPLP0380 135900000
C*                      -1 INTEGRATED BLH COEFFICIENT AND PHASE PLOT *SPLP0390 136000000
C*                          IS NOT DESIREC, ALTHOUGH BLADE LOADING *SPLP0400 136100000
C*                          DATA WAS COMPUTED THROUGH INTEGRATION *SPLP0410 136200000
C*                       (REMARK: SINCE PLOTTING POINTS (BLH,PHASE, *SPLP0420 136300000
C*                                AND SPL) ARE INPUT SEQUENTIALLY AS *SPLP0430 136400000
C*                                THEY WERE GENERATED BY PROGRAM *SPLP0440 136500000
C*                                RNPPE4, NNPLOT MUST BE 0 IF CORRES-*SPLP0450 136600000
C*                                PONDING NTEGRAT WAS 0 IN RNPPE4) *SPLP0460 136700000
C*                                                               *SPLP0470 136800000
C*         DISC INPUT PARAMETERS                                 *SPLP0480 136900000
C*                        BLH PLOT                               *SPLP0490 137000000
C*            NREAD    - NUMBER OF BLOCKS (OF SIZE 250 OR LESS) OF *SPLP0500 137100000
C*                       BLADE LOADING HARMONICS                 *SPLP0510 137200000
C*            MPOIN    - SIZE OF THE LAST BLH DATA BLOCK         *SPLP0520 137300000
C*            DELTF    - BLADE LOADING FREQUENCY (HZ.)           *SPLP0530 137400000
C*            LO       - STEADY LOADING COEFFICIENT              *SPLP0540 137500000
C*            PRCNTGR  - PER CENTAGE OF CHORD INTEGRATED         *SPLP0550 137600000
C*            RS       - ROTOR ROTATIONAL SPEED (RPM)            *SPLP0560 137700000
C*            THRUST   - TOTAL LIFT (WEIGHT OF HELICOPTER) (LBS.) *SPLP0570 137800000
C*            PRESSUR  - AN ARRAY DIMENSIONED 250 CONTAINING THE *SPLP0580 137900000
C*                       BLADE LOADING HARMONICS                 *SPLP0590 138000000
C*            PHASE    - AN ARRAY DIMENSIONED 250 CONTAINING THE *SPLP0600 138100000
C*                       INTEGRATED PHASE DATA                   *SPLP0610 138200000
C*                        SPL PLOT                               *SPLP0620 138300000
C*            NB       - NUMBER OF BLADES                        *SPLP0630 138400000
C*            F        - BLADE PASSAGE FREQUENCY (HZ.)           *SPLP0640 138500000
C*            OASPL    - OVERALL SOUND PRESSURE LEVELS (DB)      *SPLP0650 138600000
C*            THRUST   - TOTAL LIFT (WEIGHT OF HELICOPTER) (LBS.) *SPLP0660 138700000
C*            RS       - ROTOR ROTATIONAL SPEED (RPM)            *SPLP0670 138800000
C*            OBSELV   - OBSERVER ELEVATICN ANGLE (RADIANS)      *SPLP0680 138900000
C*            OBSAZI   - OBSERVER AZIMUTH ANGLE (RADIANS)        *SPLP0690 139000000
C*            TRACK    - IF NC INTEGRATICN OF STEADY LOADING DIST. *SPLP0700 139100000
C*                       WAS PERFORMED, TRACK IS THE SPECTRUM DATA *SPLP0710 139200000
C*                       POSITION                                *SPLP0720 139300000
C*            ICHORD   - DETERMINES WHICH CHORD DISTRIBUTION FUNCTION*SPLP0730 139400000
C*                       IS USED TO CORRECT SPL VALUES           *SPLP0740 139500000
C*            MAXHAR   - MAXIMUN NUMBER OF HARMONICS  0≤MAXHAR≤500 *SPLP0750 139600000
C*            SPL      - AN ARRAY DIMENSIONED 500 CONTAINING THE *SPLP0760 139700000
C*                       SOUND PRESSURE LEVEL VALUES. (DB)       *SPLP0770 139800000
C*                                                               *SPLP0780 139900000
C*         SUBROUTINES USED                                      *SPLP0790 140000000
C*            NONE                                               *SPLP0800 140100000
C*                                                               *SPLP0810 140200000
C****************************************************************SPLP0820 140300000
C                                                                 SPLP0830 140400000
      COMMON PRESSUR(250),PHASE(250),SPL(500),ACOUST(500),X(500)  SPLP0840 140500000
C                                                                 SPLP0850 140600000
      REAL LO                                                     SPLP0860 140700000
C                                                                 SPLP0870 140800000
```

```
      DATA RADIAN /57.29577913/                                         SPLP0880 140900000
      NAMELIST /FIXED/ NSPL,XMIN,XMAX,YMIN,YSCALE,RUN /INPUT/ NNPLOT    SPLP0890 141000000
C                                                                       SPLP0900 141100000
C                                                                       SPLP0910 141200000
C              MAKE INITAL CALL TO CALCCMP PROCESSOR                    SPLP0920 141300000
      CALL PSEUDO(6LSPLBLH)                                             SPLP0930 141400000
      CALL LEROY                                                        SPLP0935 141500000
C                                                                       SPLP0940 141600000
C              FIXED NAMELIST DEFAULT PARAMETER VALUES                  SPLP0950 141700000
      NSPL = 6                                                          SPLP0960 141800000
      XMIN = 0.0                                                        SPLP0970 141900000
      XMAX = 1.0E03                                                     SPLP0980 142000000
      YMIN = 1.0E01                                                     SPLP0990 142100000
      YSCALE = 1.0E01                                                   SPLP1000 142200000
C                                                                       SPLP1010 142300000
C              READ, CHECK FOR END OF FILE, AND WRITE FIXED             SPLP1020 142400000
C              INPUT DATA                                               SPLP1030 142500000
   10 READ (5,FIXED)                                                    SPLP1040 142600000
      IF (EOF,5) 999,20                                                 SPLP1050 142700000
   20 WRITE (6,FIXED)                                                   SPLP1060 142800000
C                                                                       SPLP1070 142900000
C    ....................................................               SPLP1080 143000000
C              COMPUTE SCALE FACTORS AND X AND Y-AXIS LENGTHS           SPLP1090 143100000
C     SCALE FACTORS ARE DETERMINED FOR PLOTTING                         SPLP1100 143200000
      FRANGE=XMAX-XMIN                                                  SPLP1110 143300000
      IPOW10=ALOG10(FRANGE)                                             SPLP1120 143400000
      POW10 = 10.**IPOW10                                               SPLP1130 143500000
      IF (POW10 .GE. FRANGE) POW10 = POW10/10.                          SPLP1140 143600000
      IF(FRANGE/POW10-10.) 60,50,50                                     SPLP1150 143700000
   50 XSCALE=POW10                                                      SPLP1160 143800000
      GO TO 110                                                         SPLP1170 143900000
   60 IF(FRANGE/POW10-5.)  80,70,70                                     SPLP1180 144000000
   70 XSCALE=POW10/2.                                                   SPLP1190 144100000
      GO TO 110                                                         SPLP1200 144200000
   80 IF(FRANGE/POW10-2.) 100,90,90                                     SPLP1210 144300000
   90 XSCALE=POW10/5.                                                   SPLP1220 144400000
      GO TO 110                                                         SPLP1230 144500000
  100 XSCALE=POW10/10.                                                  SPLP1240 144600000
  110 IMIN=XMIN/XSCALE                                                  SPLP1250 144700000
      IF(IMIN*XSCALE.GT.XMIN) IMIN=IMIN-1                               SPLP1260 144800000
      XMIN=IMIN*XSCALE                                                  SPLP1270 144900000
      IMAX=XMAX/XSCALE                                                  SPLP1280 145000000
      IF(IMAX*XSCALE.LT.XMAX) IMAX=IMAX+1                               SPLP1290 145100000
      XLENGTH=IMAX-IMIN                                                 SPLP1300 145200000
      YMAX=10.*YSCALE+YMIN                                              SPLP1310 145300000
C    ....................................................               SPLP1320 145400000
C                                                                       SPLP1330 145500000
C.......................LOOP FOR THE NUMBER OF..........................SPLP1340 145600000
C                        SOUND PRESSURE LEVEL PLOTS                     SPLP1350 145700000
C                                                                       SPLP1360 145800000
      DO 300 NP=1,NSPL                                                  SPLP1370 145900000
C                                                                       SPLP1380 146000000
C              READ, CHECK FOR END OF FILE, AND WRITE VARIABLE          SPLP1390 146100000
C              INPUT DATA                                               SPLP1400 146200000
      READ (5,INPUT)                                                    SPLP1410 146300000
      IF (EOF,5) 999,130                                                SPLP1420 146400000
  130 WRITE (6,INPUT)                                                   SPLP1430 146500000
C                                                                       SPLP1440 146600000
C              TEST TO DETERMINE IF BLADE LOADING HARMONICS             SPLP1450 146700000
C              PLOT IS DESIRED (BLH)                                    SPLP1460 146800000
      IF (NNPLOT .EQ. 0) GO TO 190                                      SPLP1470 146900000
C                                                                       SPLP1480 147000000
C    ....................................................               SPLP1490 147100000
C                                                                       SPLP1500 147200000
C                    SECTION TO PLOT BLADE LOADING HARMONICS            SPLP1510 147300000
C                    COEFFICIENTS AND INTEGRATED PHASE SPECTRA          SPLP1520 147400000
C                                                                       SPLP1530 147500000
C              SET ALLOWABLE RANGE (MINIMUM AND MAXIMUM) FOR THE        SPLP1540 147600000
C              BLH COEFFICIENTS AND INTEGRATED PHASE SPECTRA            SPLP1550 147700000
      YBMIN = -100.0                                                    SPLP1560 147800000
      YBMAX = 0.0                                                       SPLP1570 147900000
      YPMIN = -180.0                                                    SPLP1580 148000000
      YPMAX = 180.0                                                     SPLP1590 148100000
C              SET Y-AXIS SCALE FACTOR FOR BLH AND PHASE PLOTS          SPLP1600 148200000
```

```
      YBSCAL = 10.0                                                     SPLP1610 148300000
      YPSCAL = 40.0                                                     SPLP1620 148400000
C             INITIALIZE OABLH PARAMETER                                SPLP1630 148500000
      SUM = 0.0                                                         SPLP1640 148600000
C                                                                       SPLP1650 148700000
C             READ NECESSARY BLH PLOTTING PARAMETERS                    SPLP1660 148800000
      READ (4) NREAD,MPOIN,DELTF,LO,PRCNTGR,RS,THRUST                   SPLP1670 148900000
C                                                                       SPLP1680 149000000
C             DETERMINE IF INTEGRATED BLH AND PHASE PLOTS               SPLP1690 149100000
C             EXIST AND ARE NOT DESIRED                                 SPLP1700 149200000
      IF (NNPLOT .NE. -1) GO TO 145                                     SPLP1710 149300000
      DO 140 NR=1,NREAD                                                 SPLP1720 149400000
      READ(4)                                                           SPLP1730 149500000
  140 CONTINUE                                                          SPLP1740 149600000
      GO TO 190                                                         SPLP1750 149700000
C                                                                       SPLP1760 149800000
C             SET LOOP INDEX DEPENDING ON DESIRE OF PHASE PLOT          SPLP1770 149900000
  145 NAP = 1                                                           SPLP1780 150000000
      IF (NNPLOT .EQ. 2) NAP=2                                          SPLP1790 150100000
C                                                                       SPLP1800 150200000
C         .....LOOP FOR BLH + INTEGRATED PHASE PLOTS    .....           SPLP1810 150300000
      DO 185 IAP=1,NAP                                                  SPLP1820 150400000
C                                                                       SPLP1830 150500000
C             INITIALIZE BLH AND PHASE POINT COUNTER                    SPLP1840 150600000
      NII = 0                                                           SPLP1850 150700000
C                                                                       SPLP1860 150800000
C             SET STANDARD RECORD SIZE                                  SPLP1870 150900000
      NPOIN = 250                                                       SPLP1880 151000000
C                                                                       SPLP1890 151100000
C           DRAW AND LABEL INTEGRATED BLH AND PHASE PLOT AXIS           SPLP1900 151200000
      IF (IAP .EQ. 1) CALL AXES (0.,0.,90.,10.,YBMIN,YBSCAL,-1.,10.,52HISPLP1910 151300000
     +NTEGRATED BLADE LOADING COEFFICIENTS (20LOG(LS/LO)),.15,52)       SPLP1920 151400000
      IF (IAP .EQ. 2) CALL AXES (0.,0.,90.,10.,-200.,YPSCAL,1.,10.,35HINSPLP1930 151500000
     +TEGRATED PHASE SPECTRUM (DEGREES),0.2,35)                         SPLP1940 151600000
      CALL AXES(0.,0.,0.,XLENGTH,XMIN,XSCALE,-1.,10.,14HFREQUENCY, HZ., SPLP1950 151700000
     10.15,-14)                                                         SPLP1960 151800000
C                                                                       SPLP1970 151900000
C          ..........LOOP FOR THE NUMBER OF BLH RECORDS.........        SPLP1980 152000000
      DO 180 NR=1,NREAD                                                 SPLP1990 152100000
C                                                                       SPLP2000 152200000
C             TEST FOR LAST INTEGRATED BLH OR PHASE BLOCK               SPLP2010 152300000
      IF (NR .EQ. NREAD) NPOIN=MPOIN                                    SPLP2020 152400000
C             READ INTEGRATED BLH COEFFICIENT AND PHASE RECORD          SPLP2030 152500000
      READ (4) (PRESSUR(I),PHASE(I),I=1,NPOIN)                          SPLP2040 152600000
C                                                                       SPLP2050 152700000
C             ..........LOOP TO CREATE AND SCALE.........               SPLP2060 152800000
C                       FREQUENCY AND BLH POINTS                        SPLP2070 152900000
      DO 160 I=1,NPOIN                                                  SPLP2080 153000000
      NII = NII+1                                                       SPLP2090 153100000
      X(I) = (NII-1)*DELTF                                              SPLP2100 153200000
      X(I) = (X(I)-XMIN) / XSCALE                                       SPLP2110 153300000
      IF (IAP .EQ. 2) GO TO 150                                         SPLP2120 153400000
      SUM = SUM + PRESSUR(I)*PRESSUR(I)                                 SPLP2130 153500000
      PRESSUR(I) = 20. * ALOG10 (ABS(PRESSUR(I)/LO))                    SPLP2140 153600000
      IF (PRESSUR(I) .LT. YBMIN) PRESSUR(I) = YBMIN                     SPLP2150 153700000
      IF (PRESSUR(I) .GT. YBMAX) PRESSUR(I) = YBMAX                     SPLP2160 153800000
      PRESSUR(I) = (PRESSUR(I)-YBMIN) / YBSCAL                          SPLP2170 153900000
      GO TO 160                                                         SPLP2180 154000000
  150 IF (PHASE(I) .LT. YPMIN) PHASE(I) = YPMIN                         SPLP2190 154100000
      IF (PHASE(I) .GT. YPMAX) PHASE(I) = YPMAX                         SPLP2200 154200000
      PHASE(I) = (PHASE(I)+200.0) / YPSCAL                              SPLP2210 154300000
  160 CONTINUE                                                          SPLP2220 154400000
C             .......................................                   SPLP2230 154500000
C                                                                       SPLP2240 154600000
C             .........LOOP TO PLOT BLADE LOADING.......                SPLP2250 154700000
C                 COEFFICIENTS (OR PHASES) .VS. FREQUENCY               SPLP2260 154800000
      IF (NR.EQ.1 .AND. IAP.EQ.1) CALL CALPLT(X(1),PRESSUR(1),1)        SPLP2270 154900000
      IF (NR.EQ.1 .AND. IAP.EQ.2) CALL CALPLT(X(1),PHASE(1),1)          SPLP2280 155000000
      II = 1                                                            SPLP2290 155100000
      IF (NR .EQ. 1) II=2                                               SPLP2300 155200000
      DO 170 I=II,NPOIN                                                 SPLP2310 155300000
      IF (X(I) .GT. XLENGTH .OR. X(I) .LT. 0.0) GO TO 170               SPLP2320 155400000
      IF (IAP .EQ. 1) CALL CALPLT (X(I),PRESSUR(I),2)                   SPLP2330 155500000
      IF (IAP .EQ. 2) CALL CALPLT (X(I),PHASE(I),2)                     SPLP2340 155600000
  170 CONTINUE
C                ........................................
C
  180 CONTINUE
C
C                COMPUTE OVERALL RMS
      RMS = SQRT (SUM*DELTF)
      OABLH = 20. * ALOG10(RMS/L0)
C
C                DRAW BLH PLOT IDENTIFICATION PARAMETERS
      RPM  = RS
      CALL NOTATE (1.,9.4,.15,5HRPM =,0.,5)
      CALL NUMBER (1.7,9.4,.15,RPM,0.,0)
      CALL NOTATE (1.,9.7,.15,3HRUN,0.,3)
      CALL NUMBER (1.7,9.7,.15,RUN,0.,0)
      IF (IAP .EQ. 1) CALL NOTATE(3.0,9.7,.15,15HOABLH =      DB,0.,15)
      IF (IAP .EQ. 1) CALL NUMBER(3.9,9.7,.15,OABLH,0.,1)
      CALL NOTATE (3.,9.4,.15,14HTHRUST=     LB,0.,14)
      CALL NUMBER (3.9,9.4,.15,THRUST,0.,-1)
      CALL NOTATE (5.5,9.4,.15,4HL0 =,0.,4)
      CALL NUMBER (6.1,9.4,.15,L0,0.,3)
      CALL NOTATE (1.,9.1,.15,32HPERCENTAGE OF CHORD INTEGRATED =,0.,32)
      CALL NUMBER (5.2,9.1,.15,PRCNTGR,0.,1)
C
C                IF INTEGRATED PHASE SPECTRA PLOT IS REQUESTED,
C                REPOSITION DISC TO READ INTEGRATED PHASE SPECTRA
C                AND POSITION NEW PHASE PLOT ORIGIN.
      IF (NAP.EQ.1 .OR. IAP.EQ.2) GO TO 183
      DO 182 NR=1,NREAD
      BACKSPACE 4
  182 CONTINUE
C
C                MOVE PLOT-PEN TO ORIGIN OF NEXT PLOT)
  183 CALL NFRAME
C
  185 CONTINUE
C
C
C   .............................................................
C
C                READ ARRAY OF SOUND PRESSURE LEVELS
C                READ NECESSARY SPL PLOTTING PARAMETERS
  190 READ (4) NB,F,RS,OASPL,THRUST,OBSELV,OBSAZI,TRACK,PRCNTGR,ICHORD
      READ (4) MAXHAR,(SPL(M),M=1,MAXHAR)
C
C           ...........LOOP TO CREATE AND SCALE SPL...............
C                             AND FREQUENCY ARRAYS
C
      DO 200 M=1,MAXHAR
      MB = M*NB
C
C                THE SOUND PRESSURE LEVELS ARE SCALED
      IF (SPL(M) .GT. YMAX) SPL(M)=YMAX
      IF (SPL(M) .LT. YMIN) SPL(M)=YMIN
      ACOUST(M) = (SPL(M)-YMIN) / YSCALE
      X(M) = FLOAT(MB)*F*(1./XSCALE)
      IF (FLOAT(MB)*F .LT. XMIN) X(M) = XMIN*(1./XSCALE)
      IF (FLOAT(MB)*F .GT. XMAX) X(M) = XMAX*(1./XSCALE)
  200 CONTINUE
C
C                CONVERT OBSERVER ELEVATION AND AZIMUTH ANGLES
      OBSELV = OBSELV*RADIAN
      OBSAZI = OBSAZI*RADIAN
C
C                DRAW AND LABEL SPL PLOT AXIS
      CALL AXES(0.,0.,90.,10.,YMIN,YSCALE,-1.,10.,13HSPL, DECIBELS,.2,13
     1)
      CALL AXES(0.,0.,0.,XLENGTH,XMIN,XSCALE,-1.,10.,14HFREQUENCY, HZ.,
     1.15,-14)
C
C                DRAW SPL PLOT IDENTIFICATION PARAMETERS
      RPM = RS
      CALL NOTATE(1.,9.4,.15,5HRPM =,0.,5)
      CALL NUMBER(1.7,9.4,.15,RPM,0.,0)
      CALL NOTATE(1.,9.7,.15,3HRUN,0.,3)                                SPLP3080 163100000
      CALL NUMBER(1.7,9.7,.15,RUN,0.,-1)                                SPLP3090 163200000
      CALL NOTATE(3.0,9.7,.15,15HOASPL =      DB,0.,15)                 SPLP3100 163300000
      CALL NUMBER(3.9,9.7,.15,OASPL,0.,1)                               SPLP3110 163400000
      CALL NOTATE (3.,9.4,.15,14HTHRUST=     LB,0.,14)                  SPLP3120 163500000
      CALL NUMBER (3.9,9.4,.15,THRUST,0.,-1)                            SPLP3130 163600000
      SIGMA = OBSELV                                                    SPLP3140 163700000
      CALL NOTATE(5.5,9.7,.15,14HSIGMA =    DEG,0.,14)                  SPLP3150 163800000
      CALL NUMBER(6.4,9.7,.15,SIGMA,0.,-1)                              SPLP3160 163900000
      THETA = OBSAZI                                                    SPLP3170 164000000
      CALL NOTATE(5.5,9.4,.15,14HTHETA =    DEG,0.,14)                  SPLP3180 164100000
      CALL NUMBER(6.4,9.4,.15,THETA,0.,-1)                              SPLP3190 164200000
C                                                                       SPLP3200 164300000
      IF (NNPLOT .NE. 0) GO TO 210                                      SPLP3210 164400000
      CALL NOTATE(1.,9.1,.15,5HGAGE ,0.,5)                              SPLP3220 164500000
      CALL NUMBER(1.7,9.1,.15,TRACK,0.,-1)                              SPLP3230 164600000
      GO TO 220                                                         SPLP3240 164700000
C                                                                       SPLP3250 164800000
  210 CALL NOTATE (1.,9.1,.15,32HPERCENTAGE OF CHORD INTEGRATED =,0.,32)SPLP3260 164900000
      CALL NUMBER (5.2,9.1,.15,PRCNTGR,0.,1)                            SPLP3270 165000000
C                                                                       SPLP3280 165100000
  220 IF (ICHORD .EQ. 0) CALL NCTATE (1.,8.8,.15,23HPOINT LOADING DATA USPLP3290 165200000
     +SED,0.,23)                                                        SPLP3300 165300000
      IF (ICHORD .EQ. 1) CALL NCTATE (1.,8.8,.15,26HRECTANGULAR CHORD FUSPLP3310 165400000
     +NCTION,0.,26)                                                     SPLP3320 165500000
      IF (ICHORD .EQ. 2) CALL NOTATE (1.,8.8,.15,26HHALF COSINE CHORD FUSPLP3330 165600000
     +NCTION,0.,26)                                                     SPLP3340 165700000
      IF (ICHORD .EQ. 3) CALL NCTATE (1.,8.8,.15,25HTRIANGULAR CHORD FUNSPLP3350 165800000
     +CTION,0.,25)                                                      SPLP3360 165900000
      IF (ICHORD .EQ. 4) CALL NCTATE (1.,8.8,.15,23HSAWTOOTH CHORD FUNCTSPLP3370 166000000
     +ION,0.,23)                                                        SPLP3380 166100000
C         ...................................................           SPLP3390 166200000
C                                                                       SPLP3400 166300000
C             ...........LOOP TC PLOT SOUND   ..........                SPLP3410 166400000
C                    PRESSURE LEVEL .VS. FREQUENCY                      SPLP3420 166500000
      CALL PNTPLT (X(1),ACOUST(1),11,1)                                 SPLP3430 166600000
      DO 250 M=2,MAXHAR                                                 SPLP3440 166700000
      CALL CALPLT(X(M),ACOUST(M),2)                                     SPLP3450 166800000
      CALL PNTPLT(X(M),ACOUST(M),11,1)                                  SPLP3460 166900000
  250 CONTINUE                                                          SPLP3470 167000000
C             ..........................................                SPLP3480 167100000
C                                                                       SPLP3490 167200000
C             MOVE PLOT PEN TO ORIGIN OF NEXT PLOT                      SPLP3500 167300000
      CALL NFRAME                                                       SPLP3510 167400000
C                                                                       SPLP3520 167500000
  300 CONTINUE                                                          SPLP3530 167600000
C.......................................................................SPLP3540 167700000
C                                                                       SPLP3550 167800000
C             DETERMINE IF ANOTHER CASE IS TO BE RUN                    SPLP3560 167900000
      GO TO 10                                                          SPLP3570 168000000
  999 CALL CALPLT (0.,0.,999)                                           SPLP3580 168100000
C                                                                       SPLP3590 168200000
      STOP                                                              SPLP3600 168300000
      END                                                               SPLP3610 168400000
```

REFERENCES

1. Ramakrishnan, Ramani: On Discrete Noise Radiation From Helicopters. M. S. Thesis, George Washington Univ., Sept. 1973.

2. Hosier, Robert N.; and Ramakrishnan, Ramani: Helicopter Rotor Rotational Noise Predictions Based on Measured High-Frequency Blade Loads. NASA TN D-7624, 1974.

3. Wright, S. E.: Sound Radiation From a Lifting Rotor Generated by Asymmetric Disk Loading. J. Sound & Vib., vol. 9, no. 2, Mar. 1969, pp. 223-240.

4. Wright, S. E.: Discrete Radiation From Rotating Periodic Sources. J. Sound & Vib., vol. 17, no. 4, Aug. 22, 1971, pp. 437-498.

5. Brown, Thomas J.; Brown, Christine G.; and Hardin, Jay C.: Program for the Analysis of Time Series. NASA TM X-2988, 1974.

6. Computer Programming Manual. Langley Research Center, NASA, 1975.
   Vol. I.- General Information.
   Vol. II.- Subprogram Library.
   Vol. III.- Manufacturers Manuals.
   Vol. IV.- Special Capabilities.

7. Gutin, L.: On the Sound Field of a Rotating Propeller. NACA TM 1195, 1948. (From Physik. Zeitschr. der Sowjetunion, Bd. 9, Heft 1, 1936, pp. 57-71.)

TABLE I.- CHORDWISE DISTRIBUTIONS AND ASSOCIATED χ-FUNCTIONS

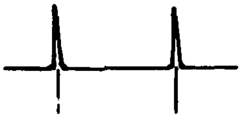
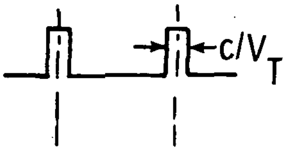
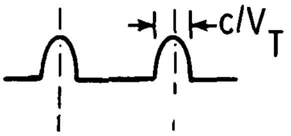
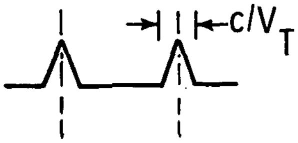
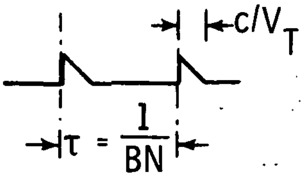
DISTRIBUTION		χ - FUNCTION
	POINT LOADING	$\chi_{mB} = 1$
	RECTANGULAR LOADING	$\chi_{mB} = \dfrac{\sin\left(\dfrac{mBc}{2r}\right)}{(mBc/2r)}$
	HALF-COSINE LOADING	$\chi_{mB} = \dfrac{\cos\left(\dfrac{mBc}{2r}\right)}{\left[1-\left(\dfrac{mBc}{\pi r}\right)^2\right]}$
	TRIANGULAR LOADING	$\chi_{mB} = \left[\dfrac{\sin\left(\dfrac{mBc}{4r}\right)}{(mBc/4r)}\right]^2$
	SAWTOOTH LOADING	$\chi_{mB} = \dfrac{\left\lvert 1 - \exp\left(-j\,\dfrac{mBc}{\pi r}\right) - j\,\dfrac{mBc}{\pi r}\right\rvert}{1/2\,(mBc/r)^2}$

TABLE II.- SRRNPP PROGRAM INTERRELATION

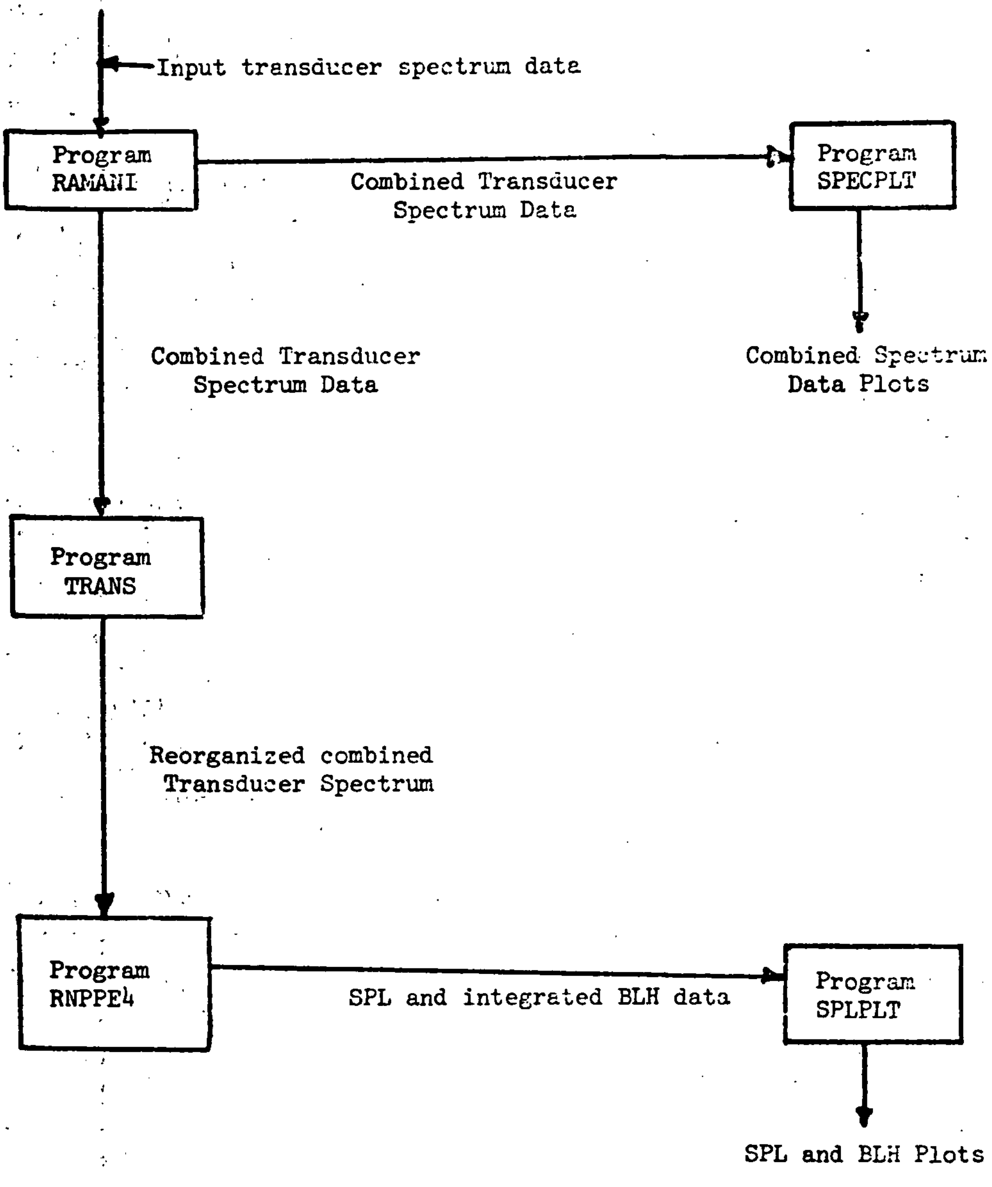

Input transducer spectrum data
Program RAMANI
Combined Transducer Spectrum Data
Program SPECPLT
Combined Spectrum Data Plots
Combined Transducer Spectrum Data
Program TRANS
Reorganized combined Transducer Spectrum
Program RNPPE4
SPL and integrated BLH data
Program SPLPLT
SPL and BLH Plots

TABLE III.- RNPPE4 FLOW CHART

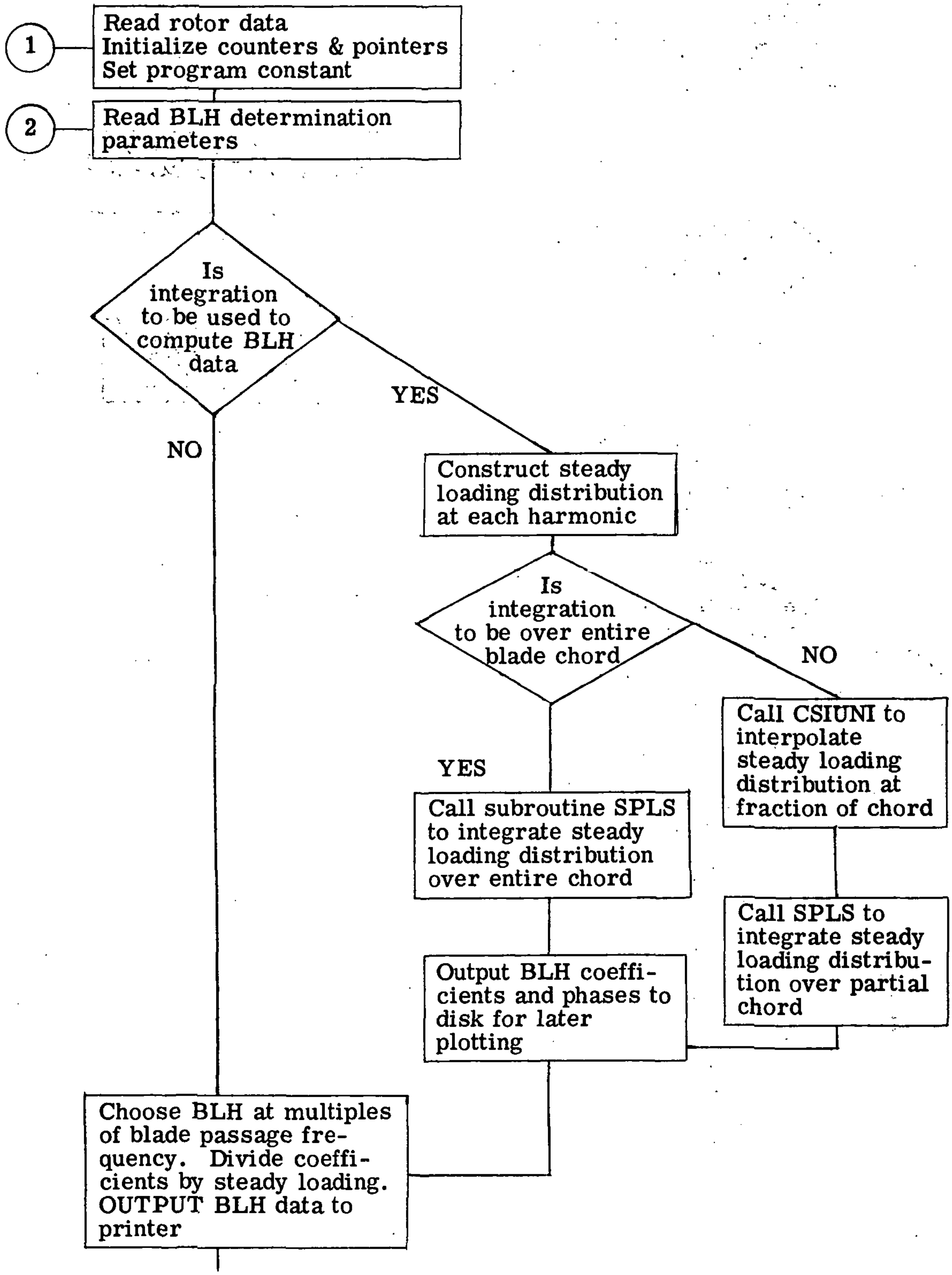

1
Read rotor data
Initialize counters & pointers
Set program constant
2
Read BLH determination parameters
Is integration to be used to compute BLH data
YES
NO
Construct steady loading distribution at each harmonic
Is integration to be over entire blade chord
NO
YES
Call CSIUNI to interpolate steady loading distribution at fraction of chord
Call subroutine SPLS to integrate steady loading distribution over entire chord
Call SPLS to integrate steady loading distribution over partial chord
Output BLH coefficients and phases to disk for later plotting
Choose BLH at multiples of blade passage frequency. Divide coefficients by steady loading. OUTPUT BLH data to printer

TABLE III.- Concluded

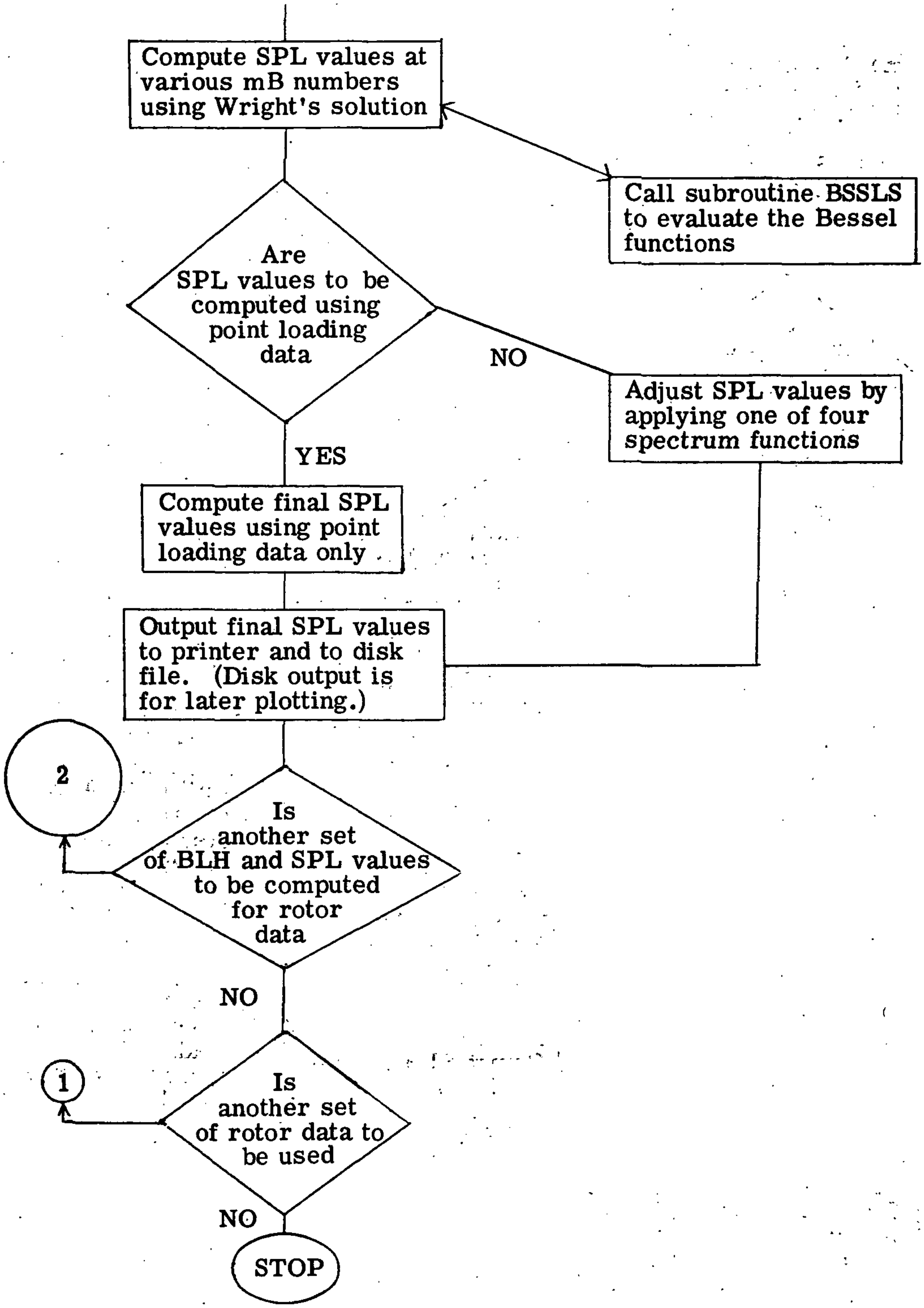

Compute SPL values at various mB numbers using Wright's solution
Call subroutine BSSLS to evaluate the Bessel functions
Are SPL values to be computed using point loading data
NO
Adjust SPL values by applying one of four spectrum functions
YES
Compute final SPL values using point loading data only
Output final SPL values to printer and to disk file. (Disk output is for later plotting.)
2
Is another set of BLH and SPL values to be computed for rotor data
NO
1
Is another set of rotor data to be used
NO
STOP

TABLE IV.- SPECTRUM-DATA INPUT PARAMETERS

Variable name	Variable description
ICH	Spectrum data transducer (or combined position) number ($1 \leqq ICH \leqq 14$) (integer).
DELTF	Blade-loading frequency (Hz) (floating point).
NSPCT	Number of spectrum points per transducer ($1 \leqq NSPCT \leqq 30{,}000$) (integer).
[a]NREAD	Number of spectrum data records per transducer ($1 \leqq NREAD \leqq 20$) (integer).
NUAMP	An array dimensioned 1500 containing the spectrum-data amplitudes (floating point).
NUPHASE	An array dimensioned 1500 containing the spectrum-data phases (degrees) (floating point).

[a] NREAD is not part of the transducer identification record of the program RAMANI input tape. It is part of the identification record of the RAMANI output tape used by programs SPECPLT and TRANS.

TABLE V.- PROGRAM RAMANI: NAMELIST INPUT

Variable name	Variable description
NTRACKS	Number of tracks of transducer spectrum data being input (integer) ($1 \leq$ NTRACKS ≤ 24).
DBREF	An array dimensioned 24 containing the decibel reference frequency constant for each track of spectrum data (floating point) (decibels). (DBREF(I) = 0.0 is default value for I = 1, 2, . . ., 24.)
NPRINT	Output control parameter specifying the number of combined spectrum data records (Frequency, Amplitude, and Phase) output to printer for each combined spectrum position (integer). (NPRINT = 50 is default value.)
NSKIP	Input control parameter specifying the number of records of transducer spectrum data to be skipped at the beginning of input tape. (Since data is input in blocks of 1500, NSKIP = $\left\langle [(N - 1)/1500] + 1 \right\rangle + 1$ is required to skip spectrum data from one transducer containing N spectrum data points.) (integer) (NSKIP = 0 is default value.)
MTRACKS	Number of spectrum data positions after combining (adding or scaling) in addition to the number of transducers whose data is to be skipped. (Exclude records skipped by NSKIP.) (integer) ($1 \leq$ MTRACKS ≤ 24.)
ISTN,JSTN	Two arrays dimensioned 24 describing the spectrum-data combining sequence. (integers) ($0 \leq$ ISTN(K),JSTN(K) ≤ 24 where K = 1, 2, . . ., 24.) (i) ISTN(K) = JSTN(K), JSTN(K) ≠ 0 implies that transducer spectrum data from transducer ISTN(K) will be weighted by the scaling factor WGHT. (ii) ISTN(K) ≠ JSTN(K), JSTN(K) ≠ 0 implies that transducer spectrum data from transducers ISTN(K) and JSTN(K) will be added. (iii) ISTN(K) ≠ JSTN(K), JSTN(K) = 0 implies spectrum data from transducers ISTN(K) will be skipped. Restrictions: The ISTN array must be strictly increasing and ISTN(I) ≧ JSTN(J) for I ≧ J. (ISTN(I)) = JSTN(I) = 0.0 is default for I = 1, . . ., 24.)
WGHT	A scaling factor by which the spectrum data amplitudes are to be weighted. (If no weighting is desired, use default.) (WGHT = 1.0 is default value.) (floating point)
IOPTN	An output control option (integer) = 0 Combined spectrum data will be output to tape only. = 1 Combined spectrum data will be output to random access for immediate use by programs SPECPLT or TRANS. An output tape is also created. (IOPTN = 1 is default value.)
NTAPE	An input control parameter (integer) = 1 Transducer spectrum data will be input from one tape. = 2 Transducer spectrum data will be input from two tapes. (NTAPE = 1 is default value.)
LO	An array dimensioned 24 (and specified REAL) containing the steady loading frequency for each transducer (psi). (LO(I) = 0.0 is default for I = 1, . . ., 24.)

TABLE VI.- PROGRAM SPECPLT NAMELIST INPUT

Variable name	Variable description
MTRACKS	Number of combined spectrum-data positions (integer) (1 ≤ MTRACKS ≤ 14).
THRUST	Total lift of helicopter (lbs) (floating point).
RPM	Rotational speed of helicopters (rpm) (floating point).
XMIN,XMAX	(XMAX - XMIN) is the allowable frequency range (X-axis) for plotting the combined spectrum data (Hz) (floating point). (XMIN = 0.0 is default value.) (XMAX = 1000.0 is default value.)
RUN	Flight run number (floating point).
YMIN,YMAX	(YMAX - YMIN) is the allowable amplitude range (Y-axis) for plotting the combined spectrum data (floating point). (YMIN = -100.0 is default value.) (YMAX = -0.0 is default value.)
YSCALE	Combined spectrum-data amplitude (Y-axis) scale factor (floating point). (YSCALE = 10.0 is default value.)
NNPLOT	An array dimensioned 14 containing the plotting control for each of the I = 1, . . ., MTRACKS combined spectrum-data positions (integer). = 0 for some I implies that no combined amplitude plot will be generated for the Ith position. = 1 for some I implies that only a combined amplitude plot will be generated for the Ith position. = 2 for some I implies that both a combined amplitude and a combined phase plot will be generated for the Ith position. (NNPLOT(I) = 0 for I = 1, . . ., 14 is default value.)
DBREF	An array dimensioned 14 containing the decibel reference frequency constants for each of the combined spectrum data positions (floating point) (decibels). (DBREF(I) = 0.0 for I = 1, . . ., 14 is default value.)
IOPTN	Combined spectrum-data input control parameter. = 0 The combined spectrum data is to be input by magnetic tape. (Programs SPECPLT and RAMANI are not job-stepped.) = 1 The combined spectrum data is input by random access. (Programs SPECPLT and RAMANI are to be job-stepped.) (Integer) (IOPTN = 1 is default value.)

TABLE VII.- PROGRAM TRANS NAMELIST INPUT

Variable name	Variable description
IOPTN	Input control parameter describing the means of combined spectrum-data input (integer). = 0 Combined transducer data will be input by magnetic tape. = 1 Combined transducer data will be input by random access file. (Programs TRANS and RAMANI are job-stepped.) = -1 Combined transducer data will be input by data card. (IOPTN = 0 is default condition.)
MTRACKS	Number of positions of combined spectrum data (1 ≦ MTRACKS ≦ 14) (integer).

TABLE VIII.- PROGRAM TRANS OPTIONAL CARD INPUT FORMAT

For each combined transducer station the optional card input has the following format:

CARD 1 Station Identification Record

ICH	Columns 1-5	(I5)
NSPCT	Columns 6-10	(I5)
DELTP	Columns 11-30	(G20.10)

CARD 2 CARD (NSPCT + 1)

NUAMP	Columns 1-20	(G20.10)
NUPHASE	Columns 21-40	(G20.10)

The individual parameters are described in table V.

TABLE IX.- PROGRAM RNPPE4 RANDOM ACCESS INPUT PARAMETERS

Variable name	Variable description
MTRACKS	Number of combined spectrum-data positions (integer). (1 ≦ MTRACKS ≦ 14.)
NSPCT	Number of spectrum data points per position (integer). (1 ≦ NSPCT ≦ 30,000.)
NRCSUM,KREC	Beginning and ending random access record locations for the reorganized combined spectrum data (integer).
MPOIN,MWORDS	Number of combined spectrum points per record in the last random access record for nonintegration and integration (integer). (1 ≦ MPOIN ≦ 250) (1 ≦ MWORDS ≦ 250*MTRACKS)
SPECTRA	An array dimensioned 250 by 16 containing the block of reorganized combined spectrum (amplitude and phase) data (floating point).

TABLE X.- PROGRAM RNPPE4 NAMELIST INPUT

[NAMELIST ROTOR – ROTOR is reinput whenever BLH and SPL data are to be computed using a new rotor, or whenever a switch to or from the inline computation of BLH data is made.]

Variable name	Variable description
NSPL	Number of sets of BLH and SPL data to be computed using this helicopter rotor (integer). (NSPL = 6 is default value.)
EFMACH	Effective radial Mach number (floating point).
OBSELV	Observer elevation angle (radians) (floating point).
EFPTCH	Effective blade pitch (radians) (floating point).
OBSDIST	Observer distance from rotor center (feet) (floating point).
NB	Number of blades (integer). (NB = 4 is default value.)
THRUST	Total lift (weight of the helicopter in pounds) (floating point).
TORQUE	Total rotor drag force (lbs) (floating point).
C	Speed of sound (feet/second) (floating point). (C = 1084.8 is default value.)
RS	Rotational speed (rpm) (floating point).
F	Blade passage frequency (rps) (floating point).
EFRAD	Effective blade radius (feet) (floating point).
EFCORD	Effective blade chord (feet) (floating point).
XMAX	Maximum frequency of available flight point loading data (Hz) (floating point). (XMAX = 1000.0 is default value.)
OBSAZI	Observer azimuth angle (radians) (floating point).
STDYLO	An array dimensioned 14 containing the steady loading coefficients for each of the combined spectrum data positions (floating point). (STDYLO(I) = 0 for I = 1, . . ., 14 is default value.)
RP	An array dimensioned 14 containing the relative positions of the combined spectrum data. The RP array must be strictly increasing and $0.0 \leq RP(I) \leq 1.0$ for I = 1, . . ., 14 (floating point). (RP(I) = 0.0 is default value for I = 1, . . ., 14.)
[NAMELIST INPUT	INPUT is reinput whenever a new set of BLH and SPL data is to be computed from the combined spectrum data.]

TABLE X.- Concluded

Variable name	Variable description
NTEGRAT	BLH data computation control parameter (integer). = 0 BLH coefficients and phases are to be derived from the spectrum data of one combined position. (No integration of the steady loading distribution is performed.) = K BLH coefficients are to be computed through integration of the steady loading distribution. The steady loading distribution is to be integrated from one edge of the blade (the first position) to the Kth spectrum data position. Therefore, the bounds on K are from 2 to the number of spectrum data positions plus two (for the blade edges). Full chord integration is accomplished by setting K =MTRACKS + 2 where MTRACKS is the number of combined spectrum-data positions. = -1 BLH's coefficients and phases are to be computed through integration of the steady loading distribution over a fraction of the chord. (If partial chord integration upper limit occurs at the K - 1 spectrum-data position, use NTEGRAT = K.) (NTEGRAT = 0 is default value.)
ITRACK	Number of the spectrum-data position whose data is to be used to compute BLH data. If integration is used to compute BLH data, use default ITRACK = 1 (integer).
PRTLINT	Fraction of the chord along which the steady loading distribution is to be integrated. If no integration is desired, use default PRTLINT = 1.0 (0.0 ≦ PRTLINT ≦ 1.0.) (floating point).
LO	Steady loading coefficient for the ITRACKth combined spectrum-data position. If integration is desired, use default condition LO = 0.0 (floating point).
ICHORD	Spectrum chord function control parameter (integer). The SPL values at each harmonic will be adjusted by the chord function indicated. = 0 Point loading data spectrum chord function. = 1 Rectangular spectrum chord function. = 2 Half-cosine spectrum chord function. = 3 Triangular spectrum chord function. = 4 Saw-tooth spectrum chord function. (ICHORD = 1 is default value.)
INCOF	BLH coefficient output control parameter. = K Every K - 1th BLH coefficient and phase angle will be output. (If K = 1 all coefficients and phases will be output.) (integer) (INCOF = 5 is default value.)
NAMELIST INBLH	Optional namelist used in computing an experimental set of BLH coefficients. INBLH should be reinput whenever a new BLH function is desired.
NBLHPT	Number of BLH coefficients and phase angles to be determined (1 ≦ NBLHPT ≦ 2000) (integer).
X,C	Constants used in defining the BLH coefficient function given by BLH(i) = c/i^x where i = 1, . . ., NBLHPT.
ICHORD	Same as described in NAMELIST INPUT.
INCOF	Same as described in NAMELIST INPUT.

TABLE XI.- PROGRAM SPLPLT DISK INPUT PARAMETERS

Variable name	Variable description
NREAD	Number of records (of length 500 or less) of integrated blade-loading coefficients and integrated phases (integer).
MPOIN	Size of the last blade-loading coefficient and phase data record (1 ≦ MPOIN ≦ 250) (integer).
DELTF	Blade-loading frequency (Hz) (floating point).
LO	Integrated steady loading coefficient (floating point).
RS	Rotor rotational speed (rpm) (floating point).
THRUST	Total lift (lbs) (floating point).
PRESSUR	An array dimensioned 250 containing a block of integrated blade-loading coefficients (floating point).
PHASE	An array dimensioned 250 containing a block of integrated phases (degrees) (floating point).
NB	Number of blades (integer).
F	Blade passage frequency (Hz) (floating point).
OASPL	Root mean square sound pressure level average (dB) (floating point).
OBSELV	Observer elevation angle (radians) (floating point).
OBSAZI	Observer azimuth angle (radians) (floating point).
TRACK	Spectrum position number for SPL values determined from nonintegrated BLH data (floating point).
PRCNTGR	Percentage of chord integrated, if integration is used to determine BLH data (floating point) (0.0 ≦ PRCNTGR ≦ 100.0).
ICHORD	Spectrum chord function used in adjusting SPL values (integer).
MAXHAR	Maximum number harmonics for which SPL values are computed (1 ≦ MAXHAR ≦ 500) (integer).
SPL	An array dimensioned 500 containing the sound pressure level values (dB) (floating point).

TABLE XII.- PROGRAM SPLPLT NAMELIST PARAMETERS

[NAMELIST FIXED – FIXED is input once at beginning of program to control SPL plotting.]

Variable name	Variable description
NSPL	Number of sound pressure level plots to be generated (integer). (NSPL = 6 is default value.)
XMAX,XMIN	(XMAX - XMIN) is the allowable frequency (X-axis) range for the SPL and BLH plotting (Hz) (floating point). (XMIN = 0.0 is default value.) (XMAX = 1000.0 is default value.)
YMIN	Minimum allowable sound pressure level plotting value (dBs) (floating point). (YMIN = 10.0 is default value.)
YSCALE	Sound pressure level scale factor (Y-axis) (floating point). (YSCALE = 10.0 is default value.)
RUN	Flight run number (floating point).
[NAMELIST INPUT	Consists of the single parameter NNPLOT and is reinput after the completion of each SPL plot to determine if integrated BLH plotting is to be done for the next set of SPL data to be plotted.]
NNPLOT	BLH plot-control parameter (integer). = 0 No integrated BLH coefficient or phase plot is generated. Corresponding NTEGRAT for program RNPPE4 is zero. = 1 An integrated blade-loading coefficient plot is generated. Corresponding NTEGRAT in program RNPPE4 is nonzero. = 2 Both integrated blade-loading coefficient and phase plots are generated. Corresponding NTEGRAT in program RNPPE4 is nonzero. = -1 No integrated BLH coefficient or phase plots are generated. Corresponding NTEGRAT in program RNPPE4 is nonzero.

TABLE XIII.- RAMANI INPUT TAPE FORMAT

Record number	Record content[a]	
1	Transducer identification	
2	1500 amplitudes	1500 phases
3	1500 amplitudes	1500 phases
4	400 amplitudes	400 phases
5	Transducer identification	
6	1500 amplitudes	1500 phases
7	1500 amplitudes	1500 phases
8	400 amplitudes	400 phases
•	•	
•	•	
•	•	
25	Transfer identification	
26	1500 amplitudes	1500 phases
27	1500 amplitudes	1500 phases
28	400 amplitudes	400 phases

[a] The transducer identification record consists of: transducer (or track) number, blade-loading frequency, and the number of spectrum data points per station.

The spectrum data (amplitude and phase) is in a frequency domain.

TABLE XIV.- RAMANI RANDOM ACCESS OUTPUT (OPTIONAL)

Record number	Record content
1	Identification
2	1500 combined amplitudes
3	1500 combined phases
4	1500 combined amplitudes
5	1500 combined phases
6	400 combined amplitudes
7	400 combined phases
.	.
.	.
.	.
22	Identification
23	1500 combined amplitudes
24	1500 combined phases
25	1500 combined amplitudes
26	1500 combined phases
27	400 combined amplitudes
28	400 combined phases
29	Steady loading
30	Amplitude average

TABLE XV.- TRANS REORGANIZED SPECTRUM DATA [a]

Record number	Record content			
NRCSUM	$A(S_1)$	$A(S_2)$	$A(S_3)$	$A(S_4)$
NRCSUM + 1	$A(S_1)$	$A(S_2)$	$A(S_3)$	$A(S_4)$
•		•		
•		•		
•		•		
NRCSUM + 13	$A(S_1)$	$A(S_2)$	$A(S_3)$	$A(S_4)$
NRCSUM + 14	$P(S_1)$	$P(S_2)$	$P(S_3)$	$P(S_4)$
NRCSUM + 15	$P(S_1)$	$P(S_2)$	$P(S_3)$	$P(S_4)$
•		•		
•		•		
•		•		
NRCSUM + 27	$P(S_1)$	$P(S_2)$	$P(S_3)$	$P(S_4)$

[a] From table XIV NRCSUM = 31 for this sample. NRCSUM is internally computed and dependent upon the number of combined positions and the number of points per position.

$A(S_i)$ indicates a block of amplitudes from combined station S_i

$P(S_i)$ indicates a block of phases from combined station.

In this example all records are of length 1,000 except for 44 and 58 which are of length 600.

TABLE XVI.- SPL AND BLH DISK FORMAT

Record number	Record content
1	SPL values for position 1
2	SPL values for position 2
3	SPL values for position 3
4	SPL values for position 4
5 to 17	BLH data for full-chord integration
18	SPL values for full-chord integration
19 to 21	BLH data for full-chord integration
22	SPL values for full-chord integration

Remarks:

(1) Assume six sets of SPL values are computed: the first four using BLH data from the separate positions and the last two involving full- and half-chord integration.

(2) A typical integrated BLH data record contains up to 250 of both integrated blade-loading coefficients and integrated phases.

TABLE XVII.- FRAME STATISTICS AND OPERATIONS CONTROL OUTPUT FOR PROGRAM SPECPLT

```
THE PLOT CONTROL CARD IMAGE IS,

PLOT.CALPOST,12(PVF=SPCTRA,XO=2.0,YO=0.5,FSH=11)
     //SINGLE PLOT MODE.
       LEROY .3MM PEN TYPE      BLACK INK COLOR.
       RAG TYPE PAPER.          PAPER NO. 00.//

FRAME         XO                 YO                XM                 YM              CAL. POS.

  1       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.
  2       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.
  3       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.
  4       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.
  5       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.
  6       2.000000E+00       5.000000E-01      1.000000E+00       1.000000E+00          0.

LB22120.                                                     LRC COMPUTER COMPLEX
JOB,1,100,50000,2000.          Z4181   33088T     100677     BIN11
USER. RANDALL,DONALD              477544121C 11100 NAS       CSC


THE PLOT CONTROL CARD IMAGE IS,

PLOT.CALPOST,12(PVF=SPCTRA,XC=2.0,YO=0.5,FSH=14,FSV=11)
     //SINGLE PLOT MODE.
       LEROY .3MM PEN TYPE      BLACK INK COLOR.
       RAG TYPE PAPER.          PAPER NO. 00.//

OPERATIONS CONTROL MESSAGES.

THIS FILE CONTAINS      6 FRAMES.
                        6 PLOTS.
                    18034 POINTS.
ESTIMATED PLOTTING TIME 0 HRS 11 MINS.
ESTIMATED PLOTTING 7.0 FEET

PLOT TAPE NO. PLT056     ON CALCOMP 12 INCH DRUM.
DATE - 05/06/74      TIME - 02.15.31.
```

TABLE XVIII.- DESCRIPTION OF PERIPHERAL EQUIPMENT

DISK STORAGE DRIVES (TRANSIENT STORAGE)
524 M CHARACTERS
60 MS AVERAGE ACCESS TIME
25 MS AVERAGE ROTATIONAL DELAY
128 FILES AVAILABLE AT 1 ACCESS POSITION
(FILE = 32,000 CH)

DATA CELL DRIVES (PERMANENT STORAGE)
2.2 BILLION CHARACTERS
40 REMOVABLE WEDGES
DATA MANAGEMENT SOFTWARE FOR
PROGRAM & STORAGE STASH, FETCH
& MODIFY, REPLACE

MAGNETIC TAPE DRIVES
MAX. DATA TRANSFER SPEED 120,000 CHAR./SEC.
BURST RATE AT 800 BPI (BITS/INCH)
TRANSPORT SPEED 150 INCHES/SECOND
LONGITUDINAL DENSITY 200,556 AND 800 BPI
DATA CODE 7 TRACK, EVEN PARITY BINARY CODED DECIMAL
OR 7 TRACK, ODD PARITY BINARY

CALCOMP MODEL 765 (12 INCH) PLOTTER
DRUM PLOTTER
16 SIGNALS RECOGNIZED BY PLOTTER
USABLE SURFACE
X 120 FEET
Y 11 INCHES

TABLE XIX.- RAMANI-SPECPLT PROGRAM SETUP (SOURCE DECK)

```
JOB,1,100,50000,2000.              Z4181      33088T      100677      BIN11
USER.RANDALL,DONALD                      477544121C  11100 NAS        CSC
LINECNT(10000)
NORFL.
RUN(S,,,,,RAMANI,,,1)
RUN(S,,,,.SPECPLT,,,1)
REQUEST,TAPE8,HY.          301022,ROS,DPR,WT10   BLH RXXX TK1-7
REWIND(TAPE8,TAPE7)
COPYBR(TAPE8,TAPE7)
DROPFIL(TAPE8)
REWIND(TAPE7,TAPE4)
SETINDF.
RAMANI.
REQUEST,TAPE51,HY.         SAVTP,RIS,DPR,WT10   BLH RXXX TK1-4
REWIND(TAPE4,TAPE51)
COPYBF(TAPE4,TAPE51)
REWIND(TAPE4)
DROPFIL(TAPE51)
REWIND(TAPE4)
SETINDF.
SPECPLT.
RFL(30000)
PLOT.CALPOST,12(PVF=SPCTRA,X0=2.0,Y0=0.5,FSH=14,FSV=11)
CONT.//SINGLE PLOT MODE.
CONT. 'LEROY .3MM PEN TYPE      BLACK INK COLOR.
CONT.  RAG TYPE PAPER.          PAPER NO. 00.//
EXIT.
DROPFIL(TAPE8)
DROPFIL(TAPE51)
7/8/9
 SOURCE DECK  PROGRAM RAMANI
7/8/9
 SOURCE DECK  PROGRAM SPECPLT
7/8/9
 DATA FOR PROGRAM RAMANI
7/8/9
 DATA FOR PROGRAM SPECPLT
6/7/8/9
```

TABLE XX.- RAMANI-SPECPLT PROGRAM SETUP (DATA CELL)

NOTE: The * control cards refer only to the plotting program SPECPLT. If these cards are removed, no combined transducer spectrum data plots are generated. This deletion produces the combination of program RAMANI alone. *-This sign is not a part of the control cards.

```
JOB,1,100,50000,2000.             Z4181     33088T     100677      BIN11
USER.RANDALL,DONALD                    477544121C  11100 NAS       CSC
NOMAP.
LINECNT(10000)
NORFL.
FETCH(Z4186,,BINARY)
COPYBR(BNFILE,RAMANI,1)
*COPYBF(BNFILE,SPECPLT)
REQUEST,TAPE8,HY.         301022,ROS,DPR,WT10  BLH RXXX TK1-7
REWIND(TAPE8,TAPE7)
COPYBF(TAPE8,TAPE7)
DROPFIL(TAPE8)
REWIND(TAPE7,TAPE4)
SETINDF.
RAMANI.
REQUEST,TAPE51,HY.        SAVTP,RIS,DPR,WT10  BLH RXXX TK1-4
REWIND(TAPE4,TAPE51)
COPYBF(TAPE4,TAPE51)
REWIND(TAPE4)
DROPFIL(TAPE51)
*REWIND(TAPE4)
*SETINDF.
*SPECPLT.
*RFL(30000)
*PLOT.CALPOST,12(PVF=SPECTRA,X0=2.0,Y0=0.5,FSH=14,FSV=11)
*CONT.// SINGLE PLOT MODE.
*CONT.   LEROY .3MM PEN TYPE     BLACK INK COLOR.
*CONT.   RAG TYPE PAPER.         PAPER NO. 00.//
EXIT.
DROPFIL(TAPE8)
DROPFIL(TAPE51)
7/8/9
 DATA FOR PROGRAM RAMANI
*7/8/9
* DATA FOR PROGRAM SPECPLT
6/7/8/9
```

TABLE XXI.- TRANS-RNPPE4-SPLPLT PROGRAM SETUP (SOURCE DECK)

```
JOB,1,100,50000,2000.           Z4181      33088T      100677      BIN11
USER.RANDALL,DONALD                  477544121C  11100 NAS         CSC
LINECNT(10000)
NORFL.
RUN(S,,,,,TRANS,,,1)
RUN(S,,,,,RNPPE4,,,1)
RUN(S,,,,,SPLPLT,,,1)
REQUEST,TAPE8,HY.         3208057,ROS,DPR,WT10  BLH RXX TK1-4
REWIND(TAPE8,TAPE7)
COPYBF(TAPE8,TAPE7)
REWIND(TAPE7)
DROPFIL(TAPE8)
SETINDF.
TRANS.
REWIND(TAPE4)
SETINDF.
RNPPE4.
REWIND(TAPE4)
SETINDF.
SPLPLT.
RFL(30000)
PLOT.CALPOST,12(PVF=SPLBLH,XO=2.0,YO=0.5,FSH=14,FSV=11)
CONT.//SINGLE PLOT MODE.
CONT.  LEROY .3MM PEN TYPE      BLACK INK COLOR.
CONT.  RAG TYPE PAPER.          PAPER NO. 00.//
EXIT.
DROPFIL(TAPE8)
7/8/9
 SOURCE DECK PROGRAM TRANS
7/8/9
 SOURCE DECK PROGRAM RNPPE4
7/8/9
 SOURCE DECK PROGRAM SPLPLT
7/8/9
 DATA FOR PROGRAM TRANS
7/8/9
 DATA FOR PROGRAM RNPPE4
7/8/9
 DATA FOR PROGRAM SPLPLT
6/7/8/9
```

TABLE XXII.- TRANS-RNPPE4-SPLPLT PROGRAM SETUP (DATA CELL)

```
JOB,1,100,50000,2000.              Z4181     33088T     100677     BIN11
USER.RANDALL,DONALD
NOMAP.
LINECNT(10000)
NORFL.
FETCH(Z4188,,BINARY)
COPYBR(BNFILE,TRANS,1)
COPYBR(BNFILE,RNPPE4,4)
COPYBF(BNFILE,SPLPLT)
REQUEST,TAPE8,HY.        3208057,ROS,DPR,WT10  BLH RXX TK1-4
REWIND(TAPE8,TAPE7)
COPYBF(TAPE8,TAPE7)
REWIND(TAPE7)
DROPFIL(TAPE8)
SETINDF.
TRANS.
REWIND(TAPE4)
SETINDF.
RNPPE4.
REWIND(TAPE4)
SETINDF.
SPLPLT.
RFL(30000)
PLOT.CALPOST,12(PVF=SPLBLH,XO=2.0,YO=0.5,FSH=14,FSV=11)
CONT.//SINGLE PLOT MODE.
CONT.  LEROY .3MM PEN TYPE      BLACK INK COLOR
CONT.  RAG TYPE PAPER.          PAPER NO. 00.//
EXIT.
DROPFIL(TAPE8)
7/8/9
 DATA FOR PROGRAM TRANS
7/8/9
 DATA FOR PROGRAM RNPPE4
7/8/9
 DATA FOR PROGRAM SPLPLT
6/7/8/9
```

TABLE XXIII.- RAMANI-SPECPLT-TRANS-RNPPE4-SPECPLT PROGRAM SETUP (DATA CELL)

```
JOB,1,150,50000,4000.             Z4181     33088T     10067      BIN11
USER.RANDALL,DONALD                    477544121C  11100 NAS      CSC
NOMAP.
LINECNT(10000)
NORFL.
FETCH(Z4186,,BINARY)
COPYBR(BNFILE,RAMANI,1)
COPYBF(BNFILE,SPECPLT)
FETCH(Z4188,,BINARY)
COPYBR(BNFILE,TRANS,1)
COPYBR(BNFILE,RNPPE4,4)
COPYBF(BNFILE,SPLPLT)
REQUEST,TAPE8,HY.          301022,ROS,SPR,WT10  BLH RXXX TK1-7
REWIND(TAPE8,TAPE7)
COPYBF(TAPE8,TAPE7)
DROPFIL(TAPE8)
REWIND(TAPE7,TAPE4)
SETINDF.
RAMANI.
REQUEST,TAPE51,HY.         SAVTP,RIS,DPR,WT10  BLH RXXX TK1-4
REWIND(TAPE4,TAPE51)
COPYBF(TAPE4,TAPE51)
DROPFIL(TAPE51)
REWIND(TAPE4)
SETINDF.
SPECPLT.
PLOT.CALPOST,12(PVF=SPECTRA,XO=2.0,YO=0.5,FSH=14,FSV=11)
CONT.//SINGLE PLOT MODE.
CONT.  LEROY .3MM PEN TYPE      BLACK INK COLOR.
CONT.  RAG TYPE PAPER.          PAPER NO. 00.//
REWIND(TAPE4)
SETINDF.
TRANS.
REWIND(TAPE4)
SETINDF.
RNPPE4.
REWIND(TAPE4)
SETINDF.
SPLPLT.
RFL(30000)
PLOT.CALPOST,12(PVF=SPLBLH,XO=2.0,YO=0.5,FSH=14,FSV=11)
CONT.//SINGLE PLOT MODE.
CONT.  LEROY .3MM PEN TYPE      BLACK INK COLOR.
CONT.  RAG TYPE PAPER           PAPER NO. 00.//
EXIT.
```

TABLE XXIII.- Concluded

```
DROPFIL(TAPE8)
DROPFIL(TAPE51)
7/8/9
 DATA FOR PROGRAM RAMANI
7/8/9
 DATA FOR PROGRAM SPECPLT
7/8/9
 DATA FOR PROGRAM TRANS
7/8/9
 DATA FOR PROGRAM RNPPE4
7/8/9
 DATA FOR PROGRAM SPLPLT
6/7/8/9
```

TABLE XXIV.- RAMANI-SPECPLT-TRANS-RNPPE4-SPLPLT PROGRAM SETUP (SOURCE DECK)

> NOTE: The * control cards refer only to the plotting programs SPECPLT and SPLPLT. If these cards are removed, no combined transducer spectrum plots, or sound pressure level and integrated blade loading data plots will be generated. This deletion results in the program combination RAMANI-TRANS-RNPPE4 being executed. *-This sign is not apart of the control card deck.

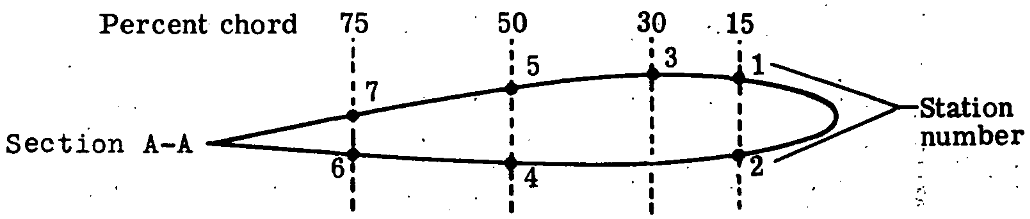
```
JOB,1,150,50000,4000.              Z4181    33088T    100677      BIN11
USER.RANDALL,DONALD                      477544121C  11100 NAS    CSC
LINECNT(10000)
NORFL.
RUN(S,,,,,RAMANI,,,1)
*RUN(S,,,,,SPECPLT,,,1)
RUN(S,,,,,TRANS,,,1)
RUN(S,,,,,RNPPE4,,,1)
*RUN(S,,,,,SPLPLT,,,1)
REQUEST,TAPE8,HY.           301022,ROS,DPR,WT10  BLH RXXX TK1-7
REWIND(TAPE8,TAPE7)
COPYBF(TAPE8,TAPE7)
DROPFIL(TAPE8)
REWIND(TAPE7,TAPE4)
SETINDF.
RAMANI.
REQUEST,TAPE51,HY.          SAVTP,RIS,DPR,WT10  BLH RXXX TK1-4
REWIND(TAPE4,TAPE51)
COPYBF(TAPE4,TAPE51)
DROPFIL(TAPE51)
*REWIND(TAPE4)
*SETINDF.
*SPECPLT.
*PLOT.CALPOST,12(PVF=SPECTRA,X0=2.0,Y0=0.5,FSH=14,FSV=11)
*CONT.//SINGLE PLOT MODE.
*CONT.  LEROY .3MM PEN TYPE     BLACK INK COLOR.
*CONT.  RAG TYPE PAPER.         PAPER NO. 00.//
REWIND(TAPE4)
SETINDF.
TRANS.
REWIND(TAPE4)
SETINDF.
RNPPE4.
*REWIND(TAPE4)
*SETINDF.
*SPLPLT.
*RFL(30000)
*PLOT.CALPOST,12(PVF=SPLBLH,X0=2.0,Y0=0.5,FSH=14,FSV=11)
```

TABLE XXIV.- Concluded

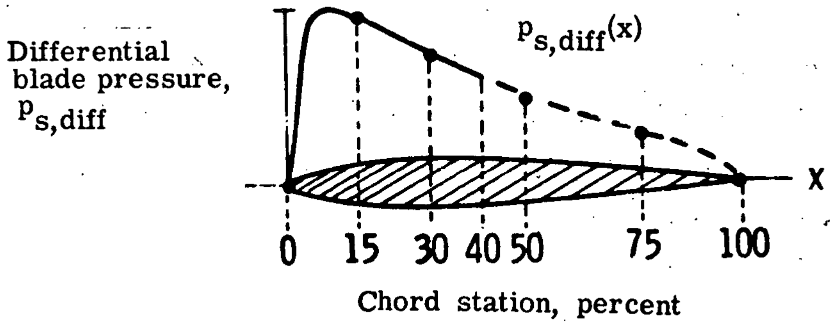
```
*CONT.//SINGLE PLOT MODE.
*CONT.  LEROY .3MM PEN TYPE      BLACK INK COLOR.
*CONT.  RAG TYPE PAPER.          PAPER NO. 00.//
EXIT.
DROPFIL(TAPE8)
DROPFIL(TAPE51)
7/8/9
 SOURCE DECK PROGRAM RAMANI
*7/8/9
* SOURCE DECK PROGRAM SPECPLT
7/8/9
 SOURCE DECK PROGRAM TRANS
7/8/9
 SOURCE DECK PROGRAM RNPPE4
*7/8/9
* SOURCE DECK PROGRAM SPLPLT
7/8/9
 DATA FOR PROGRAM RAMANI
*7/8/9
* DATA FOR PROGRAM SPECPLT
7/8/9
 DATA FOR PROGRAM TRANS
7/8/9
 DATA FOR PROGRAM RNPPE4
*7/8/9
* DATA FOR PROGRAM SPLPLT
6/7/8/9
```

TABLE XXV.- PROGRAM PERFORMANCE

Program combination	CPU time (seconds)	PPU time (seconds)	OS calls
RAMANI-SPECPLT	38.7	448.3	923
TRANS-RNPPE4-SPLPLT	66.2	299.1	1186
RAMANI-SPECPLT-TRANS-RNPPE4-SPLPLT	104.3	749.2	1901

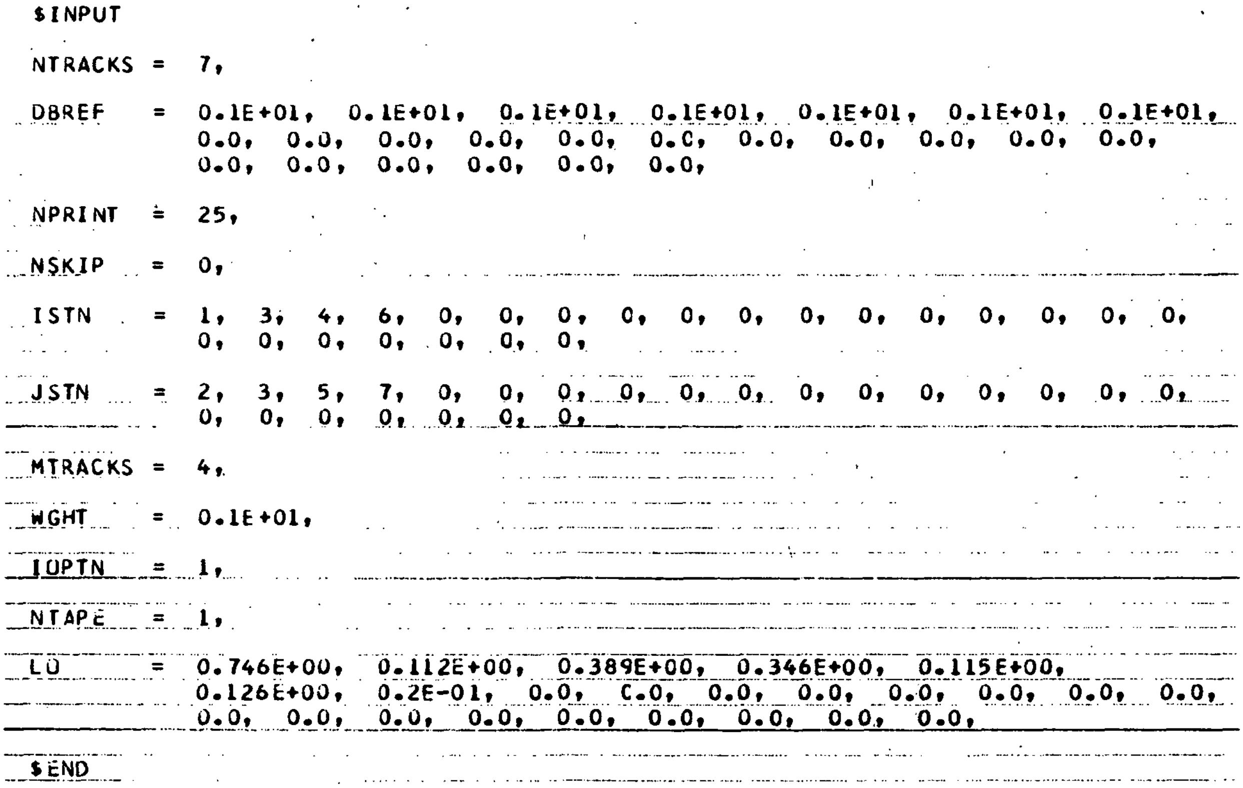


(a) Location of pressure transducers.



(b) Typical 40 percent and 100 percent chordwise integration of pressure distribution.

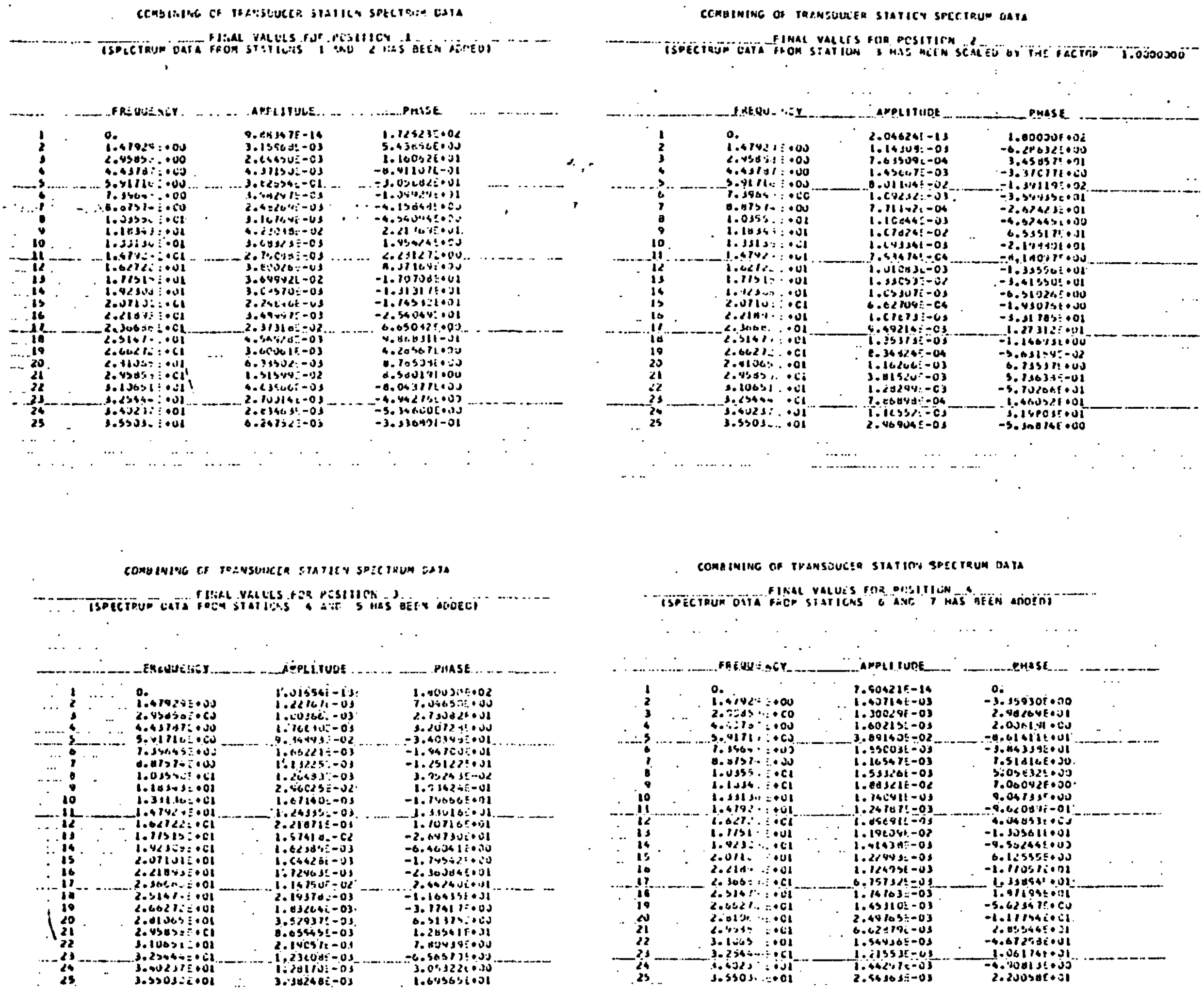
Figure 1.- Pressure transducers and pressure distribution.



```
$INPUT

NTRACKS  =  7,

DBREF    =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.C,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

NPRINT   =  25,

NSKIP    =  0,

ISTN     =  1,  3,  4,  6,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,

JSTN     =  2,  3,  5,  7,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,

MTRACKS  =  4,

WGHT     =  0.1E+01,

IOPTN    =  1,

NTAPE    =  1,

LO       =  0.746E+00,  0.112E+00,  0.389E+00,  0.346E+00,  0.115E+00,
            0.126E+00,  0.2E-01,  0.0,  C.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

$END
```

(a) Input.

Figure 2.- Sample input and output for program RAMANI.



```
                 COMBINING OF TRANSDUCER STATION SPECTRUM DATA

                          FINAL VALUES FOR POSITION 1
             (SPECTRUM DATA FROM STATIONS  1 AND  2 HAS BEEN ADDED)


              FREQUENCY            AMPLITUDE            PHASE

     1        0.                   9.88367E-14          1.72523E+02
     2        1.47929E+00          3.15968E-03          5.43656E+00
     3        2.95858E+00          2.64450E-03          1.16062E+01
     4        4.43787E+00          4.37150E-03         -8.91107E-01
     5        5.91716E+00          3.82554E-01         -3.05082E+01
     6        7.39645E+00          3.94297E-03         -1.09929E+01
     7        8.87574E+00          2.45269E-03         -4.15844E+00
     8        1.03550E+01          3.16764E-03         -4.54094E+00
     9        1.18343E+01          4.23018E-02          2.21769E+01
    10        1.33136E+01          3.68273E-03          1.95474E+00
    11        1.47929E+01          2.76095E-03          2.23127E+00
    12        1.62722E+01          3.80026E-03          8.37169E+00
    13        1.77515E+01          3.69992E-02         -1.70708E+01
    14        1.92308E+01          3.04570E-03         -1.31317E+01
    15        2.07101E+01          2.26046E-03         -1.74512E+01
    16        2.21893E+01          3.49497E-03         -2.54049E+01
    17        2.36686E+01          2.37316E-02          6.65042E+00
    18        2.51479E+01          4.56978E-03          9.86831E-01
    19        2.66272E+01          3.60061E-03          4.26567E+00
    20        2.81065E+01          6.35502E-03          8.76534E+00
    21        2.95858E+01          1.51599E-02          8.56019E+00
    22        3.10651E+01          4.43566E-03         -8.04377E+00
    23        3.25444E+01          2.70314E-03         -4.94276E+00
    24        3.40237E+01          2.83463E-03         -5.34600E+00
    25        3.55030E+01          6.24752E-03         -3.33649E-01
```

```
                 COMBINING OF TRANSDUCER STATION SPECTRUM DATA

                          FINAL VALUES FOR POSITION 2
    (SPECTRUM DATA FROM STATION  3 HAS BEEN SCALED BY THE FACTOR   1.0000000


              FREQUENCY            AMPLITUDE            PHASE

     1        0.                   2.04624E-13          1.80000E+02
     2        1.47929E+00          1.14309E-03         -6.28632E+00
     3        2.95858E+00          7.63509E-04          3.45857E+01
     4        4.43787E+00          1.45667E-03         -3.37077E+00
     5        5.91716E+00          8.01104E-02         -1.39119E+02
     6        7.39645E+00          1.09232E-03         -3.59935E+01
     7        8.87574E+00          7.71142E-04         -2.67423E+01
     8        1.03550E+01          1.10644E-03         -4.67445E+00
     9        1.18343E+01          1.07824E-02          6.55517E+01
    10        1.33136E+01          1.09336E-03         -2.19940E+01
    11        1.47929E+01          7.59474E-04         -4.14097E+00
    12        1.62722E+01          1.01083E-03         -1.33556E+01
    13        1.77515E+01          1.33059E-02         -3.41550E+01
    14        1.92308E+01          1.05307E-03         -6.51026E+00
    15        2.07101E+01          6.62709E-04         -1.93075E+00
    16        2.21893E+01          1.07873E-03         -3.31785E+01
    17        2.36686E+01          5.49214E-03          1.27312E+01
    18        2.51479E+01          1.35373E-03         -1.16673E+00
    19        2.66272E+01          8.34424E-04         -5.63159E-02
    20        2.81065E+01          1.16266E-03          6.73537E+00
    21        2.95858E+01          3.81520E-03          5.73634E-01
    22        3.10651E+01          1.28299E-03         -5.70266E+01
    23        3.25444E+01          7.26898E-04          1.46052E+01
    24        3.40237E+01          1.16557E-03          3.19803E+01
    25        3.55030E+01          2.96904E-03         -5.36874E+00
```

```
                 COMBINING OF TRANSDUCER STATION SPECTRUM DATA

                          FINAL VALUES FOR POSITION 3
             (SPECTRUM DATA FROM STATIONS  4 AND  5 HAS BEEN ADDED)


              FREQUENCY            AMPLITUDE            PHASE

     1        0.                   1.01654E-13          1.80000E+02
     2        1.47929E+00          1.22767E-03          7.04650E+00
     3        2.95858E+00          1.00366E-03          2.73082E+01
     4        4.43787E+00          1.76010E-03          3.20724E+00
     5        5.91716E+00          9.34993E-02         -3.40399E+01
     6        7.39645E+00          1.66221E-03         -1.94700E+01
     7        8.87574E+00          1.13225E-03         -1.25122E+01
     8        1.03550E+01          1.26433E-03          3.95243E-02
     9        1.18343E+01          2.96025E-02          1.31624E-01
    10        1.33136E+01          1.67140E-03         -1.79666E+01
    11        1.47929E+01          1.24335E-03          3.31016E+01
    12        1.62722E+01          2.21871E-03          1.70716E+01
    13        1.77515E+01          1.57418E-02         -2.69730E+01
    14        1.92308E+01          1.62384E-03         -6.46041E+00
    15        2.07101E+01          1.04426E-03         -1.79542E+00
    16        2.21893E+01          1.72463E-03         -2.36084E+01
    17        2.36686E+01          1.16750E-02          2.44240E+01
    18        2.51479E+01          2.19376E-03         -1.16435E+01
    19        2.66272E+01          1.83264E-03         -3.77417E+00
    20        2.81065E+01          3.57937E-03          6.51375E+00
    21        2.95858E+01          8.65545E-03          1.28541E+01
    22        3.10651E+01          2.14057E-03          7.80939E+00
    23        3.25444E+01          1.23608E-03         -6.56573E+00
    24        3.40237E+01          1.28170E-03          3.05322E+00
    25        3.55032E+01          3.38248E-03          1.69565E+01
```

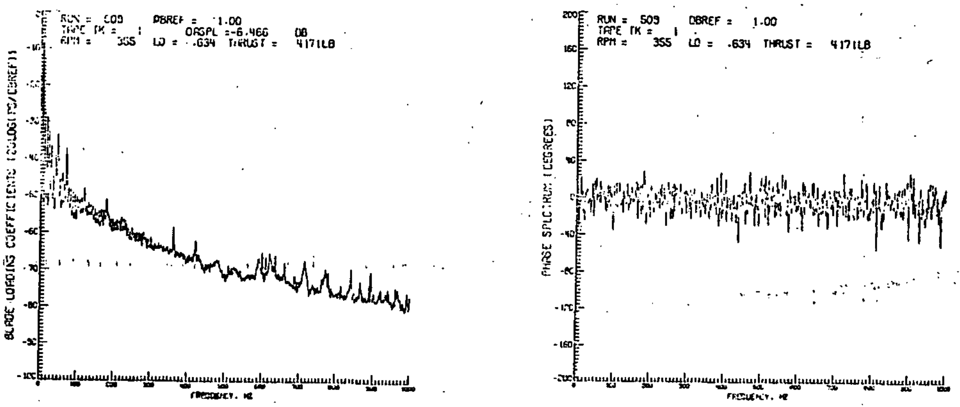
```
                 COMBINING OF TRANSDUCER STATION SPECTRUM DATA

                          FINAL VALUES FOR POSITION 4
             (SPECTRUM DATA FROM STATIONS  6 AND  7 HAS BEEN ADDED)


              FREQUENCY            AMPLITUDE            PHASE

     1        0.                   7.90421E-14          0.
     2        1.47929E+00          1.40714E-03         -3.35930E+00
     3        2.95858E+00          1.30029E-03          2.98269E+01
     4        4.43787E+00          1.60215E-03          2.03619E+00
     5        5.91716E+00          3.89140E-02         -8.61481E+01
     6        7.39645E+00          1.55003E-03         -3.84339E+01
     7        8.87574E+00          1.16547E-03          7.51416E+00
     8        1.03550E+01          1.53326E-03          5.05632E+00
     9        1.18343E+01          1.88321E-02          7.06092E+00
    10        1.33136E+01          1.74091E-03          9.04733E+00
    11        1.47929E+01          1.24787E-03         -9.62089E-01
    12        1.62722E+01          1.05691E-03          4.04853E+00
    13        1.77515E+01          1.19809E-02         -1.30561E+01
    14        1.92308E+01          1.41438E-03         -9.55244E+00
    15        2.07101E+01          1.27993E-03          6.12555E+00
    16        2.21893E+01          1.72495E-03         -1.77057E+01
    17        2.36686E+01          6.75732E-03          1.33894E+01
    18        2.51479E+01          1.74763E-03          1.97195E+01
    19        2.66272E+01          1.45310E-03         -5.62347E+00
    20        2.81065E+01          2.49765E-03         -1.17754E+01
    21        2.95858E+01          6.62479E-03          2.85544E+01
    22        3.10651E+01          1.54936E-03         -4.67258E+01
    23        3.25444E+01          1.21553E-03          1.06174E+01
    24        3.40237E+01          1.44247E-03         -4.90813E+00
    25        3.55030E+01          2.54363E-03          2.20058E+01
```

(b) Output.

Figure 2.- Concluded.

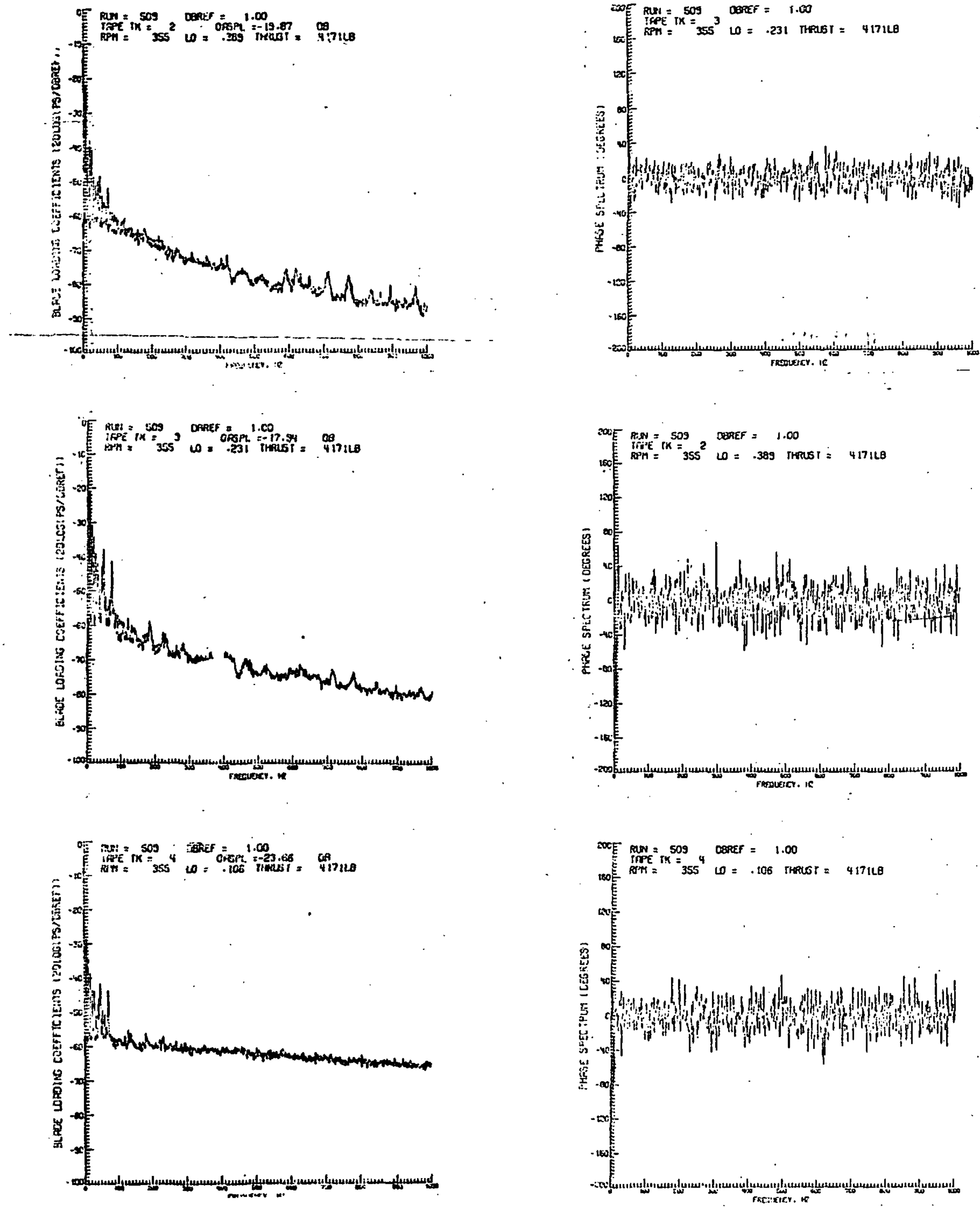
```
$INPUT
MTRACKS =  4,
THRUST  =  0.4171E+04,
RPM     =  0.355E+03,
XMIN    =  0.0,
XMAX    =  0.1E+04,
RUN     =  0.509E+03,
YMIN    = -0.1E+03,
YMAX    = -0.0,
YSCALE  =  0.1E+02,
NNPLOT  =  2,  2,  2,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
DBREF   =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,
IOPTN   =  1,
$END
```

(a) Input.



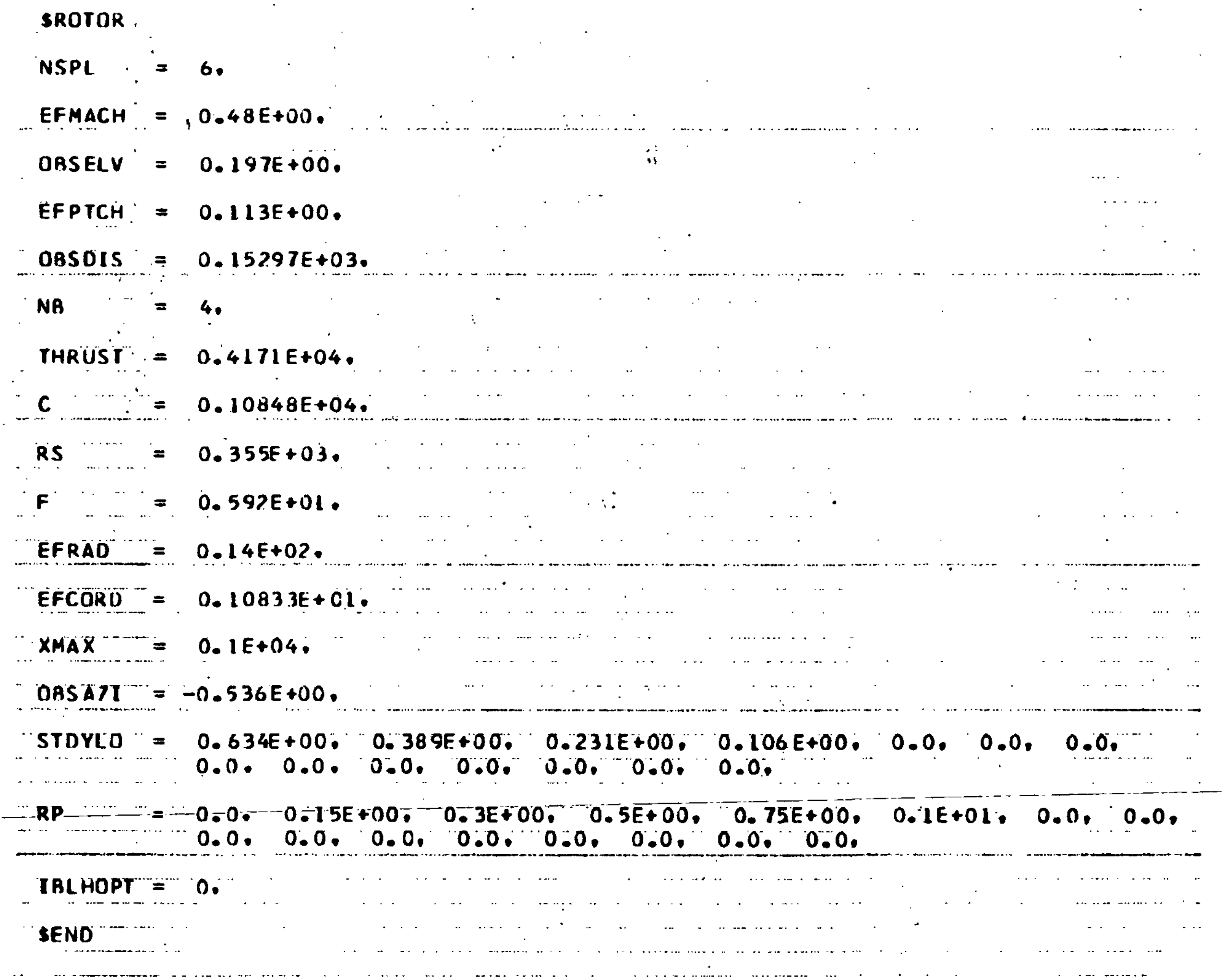
(b) Output.

Figure 3.- Sample input and output (graphic) for program SPECPLT.



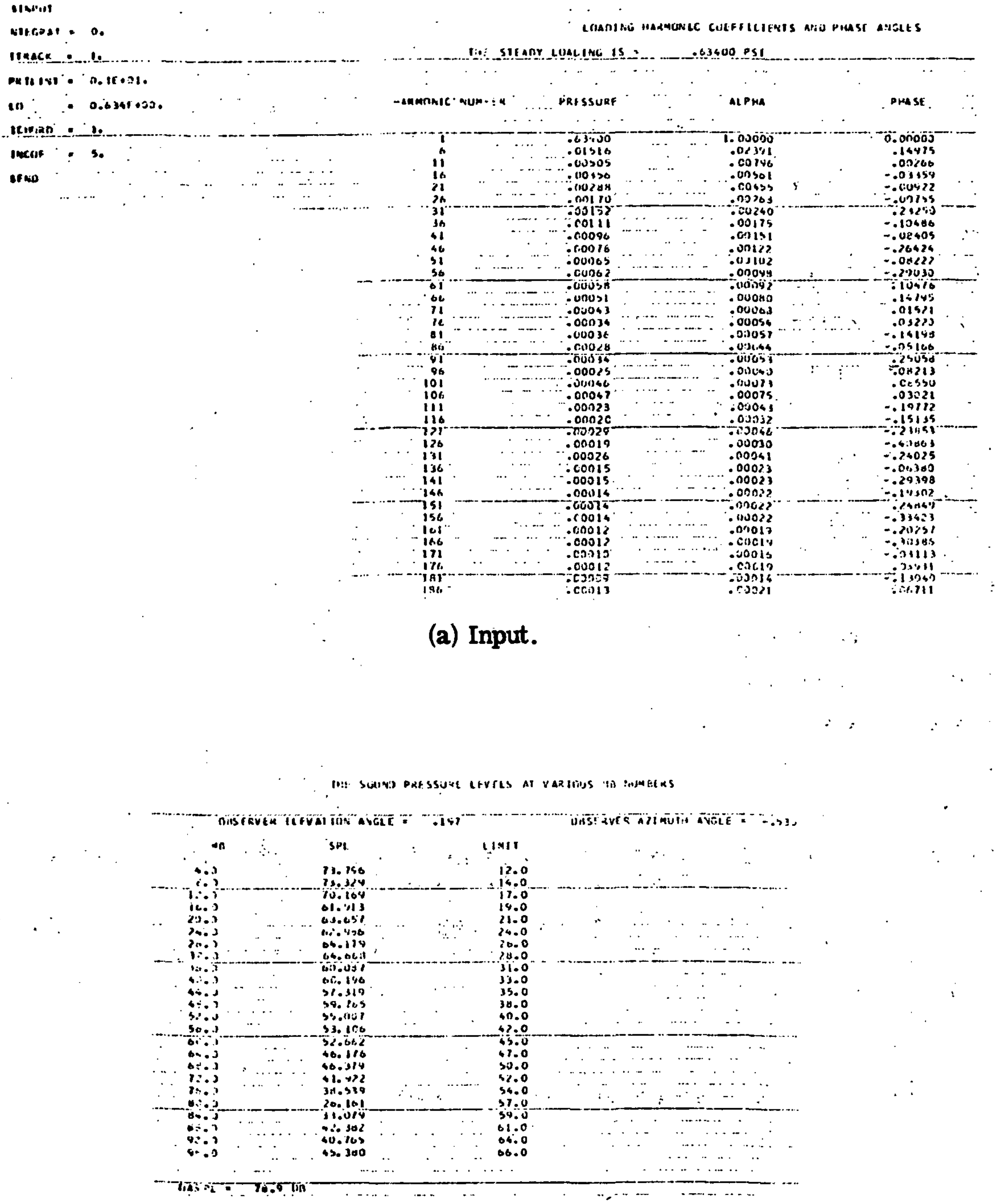
(b) Concluded.

Figure 3.- Concluded.



```
$ROTOR
NSPL      =  6,
EFMACH    =  0.48E+00,
OBSELV    =  0.197E+00,
EFPTCH    =  0.113E+00,
OBSDIS    =  0.15297E+03,
NB        =  4,
THRUST    =  0.4171E+04,
C         =  0.10848E+04,
RS        =  0.355E+03,
F         =  0.592E+01,
EFRAD     =  0.14E+02,
EFCORD    =  0.10833E+01,
XMAX      =  0.1E+04,
OBSA7I    = -0.536E+00,
STDYLD    =  0.634E+00,  0.389E+00,  0.231E+00,  0.106E+00,  0.0,  0.0,  0.0,
             0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
RP        =  0.0,  0.15E+00,  0.3E+00,  0.5E+00,  0.75E+00,  0.1E+01,  0.0,  0.0,
             0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
IBLHOPT   =  0,
$END
```

Figure 4.- Sample input (parametric) for program RNPPE4.



```
$INPUT
NTEGRAT =  0.
ITRACK  =  1.
PRTLIST =  0.1E+21.
LO      =  0.634E+00.
ICURVD  =  1.
INCOF   =  5.
$END
```

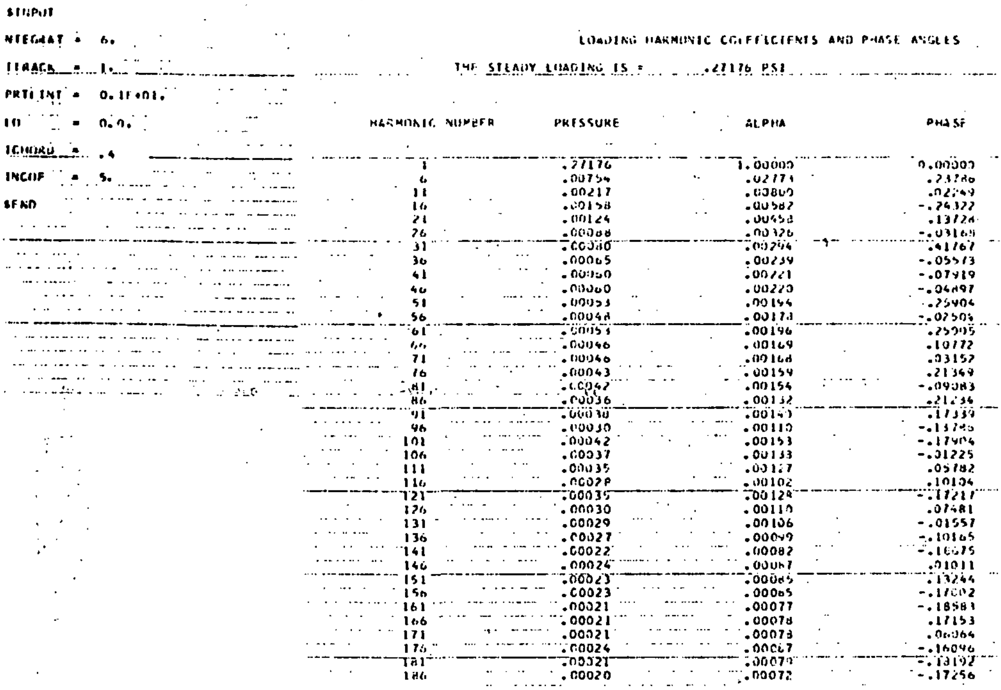
```
                     LOADING HARMONIC COEFFICIENTS AND PHASE ANGLES

            THE STEADY LOADING IS =      .63400 PSI

-HARMONIC NUMBER      PRESSURE            ALPHA              PHASE

        1              .63400            1.00000            0.00000
        6              .01516             .02391             .14975
       11              .00505             .00796             .00266
       16              .00456             .00561            -.03359
       21              .00288             .00455            -.00922
       26              .00170             .00263            -.09755
       31              .00152             .00240             .24293
       36              .00111             .00175            -.10486
       41              .00096             .00151            -.02405
       46              .00076             .00122            -.26424
       51              .00065             .00102            -.08222
       56              .00062             .00098            -.29030
       61              .00058             .00092             .10476
       66              .00051             .00080             .14795
       71              .00043             .00063             .01521
       76              .00034             .00054             .03223
       81              .00036             .00057            -.14198
       86              .00028             .00044            -.05166
       91              .00034             .00053             .25058
       96              .00025             .00040            -.08213
      101              .00046             .00073             .02550
      106              .00047             .00075             .03021
      111              .00023             .00043            -.19772
      116              .00020             .00032            -.15135
      121              .00029             .00046            -.21853
      126              .00019             .00030            -.40863
      131              .00026             .00041            -.24025
      136              .00015             .00023            -.06380
      141              .00015             .00023            -.29398
      146              .00014             .00022            -.19302
      151              .00014             .00022             .24849
      156              .00014             .00022            -.33423
      161              .00012             .00019            -.20257
      166              .00012             .00019            -.30385
      171              .00010             .00016            -.03113
      176              .00012             .00019             .35911
      181              .00009             .00014            -.13049
      186              .00013             .00021             .06711
```

(a) Input.

```
              THE SOUND PRESSURE LEVELS AT VARIOUS HD NUMBERS

     OBSERVER ELEVATION ANGLE =   .197              OBSERVER AZIMUTH ANGLE =  -.533

       HD              SPL              LIMIT

       4.0            73.756             12.0
       8.0            73.329             14.0
      12.0            70.169             17.0
      16.0            61.913             19.0
      20.0            63.657             21.0
      24.0            62.956             24.0
      28.0            64.119             26.0
      32.0            64.668             28.0
      36.0            60.037             31.0
      40.0            60.196             33.0
      44.0            57.319             35.0
      48.0            59.765             38.0
      52.0            55.007             40.0
      56.0            53.106             42.0
      60.0            52.662             45.0
      64.0            46.176             47.0
      68.0            46.379             50.0
      72.0            41.922             52.0
      76.0            38.539             54.0
      80.0            26.161             57.0
      84.0            31.079             59.0
      88.0            42.382             61.0
      92.0            40.765             64.0
      96.0            45.380             66.0

   OASPL =   76.9 DB
```

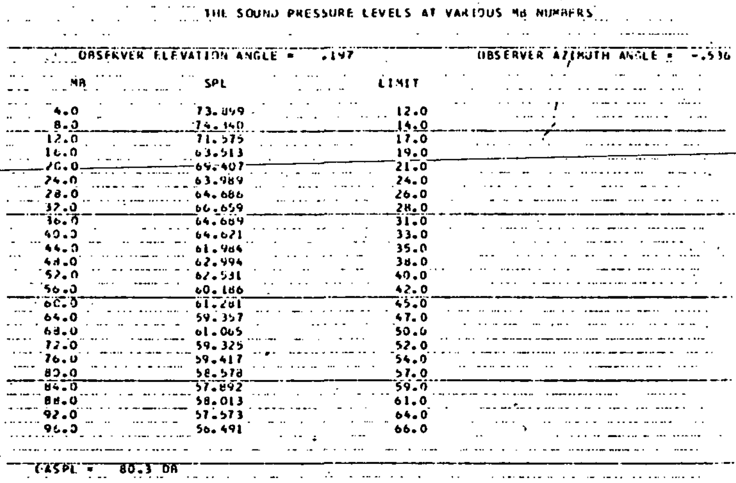
(b) Output.

Figure 5.- Printed input and output for Case A.



```
$INPUT
NTEGRAT =  6.                              LOADING HARMONIC COEFFICIENTS AND PHASE ANGLES
ITRACK  =  1.                              THE STEADY LOADING IS =      .27176 PSI
PRTLINT =  0.1E+01,
IO      =  0.0,                 HARMONIC NUMBER      PRESSURE          ALPHA             PHASE
ICHORD  =  .4
INCOF   =  5.                          1             .27176           1.00000           0.00000
$END                                   6             .00754            .02771            .23786
                                      11             .00217            .03809            .02249
                                      16             .00158            .00582          -.24322
                                      21             .00124            .00458            .13724
                                      26             .00088            .00326          -.03169
                                      31             .00080            .00294            .41767
                                      36             .00065            .00239          -.05573
                                      41             .00060            .00221          -.07919
                                      46             .00060            .00220          -.04897
                                      51             .00053            .00194            .25904
                                      56             .00048            .00178          -.07504
                                      61             .00053            .00196            .25905
                                      66             .00046            .00169            .10772
                                      71             .00046            .00168            .03152
                                      76             .00043            .00159            .21369
                                      81             .00042            .00154          -.09083
                                      86             .00036            .00132            .21234
                                      91             .00038            .00140            .17339
                                      96             .00030            .00110          -.13745
                                     101             .00042            .00153          -.17904
                                     106             .00037            .00133          -.01225
                                     111             .00035            .00127            .05782
                                     116             .00028            .00102            .10104
                                     121             .00035            .00128          -.17217
                                     126             .00030            .00110            .07481
                                     131             .00029            .00106          -.01557
                                     136             .00027            .00099          -.10165
                                     141             .00022            .00082          -.16675
                                     146             .00024            .00087            .01011
                                     151             .00023            .00085            .13244
                                     156             .00023            .00085          -.17002
                                     161             .00021            .00077          -.18583
                                     166             .00021            .00078            .17153
                                     171             .00021            .00078            .06064
                                     176             .00024            .00087          -.16096
                                     181             .00021            .00079          -.13192
                                     186             .00020            .00072          -.17256
```

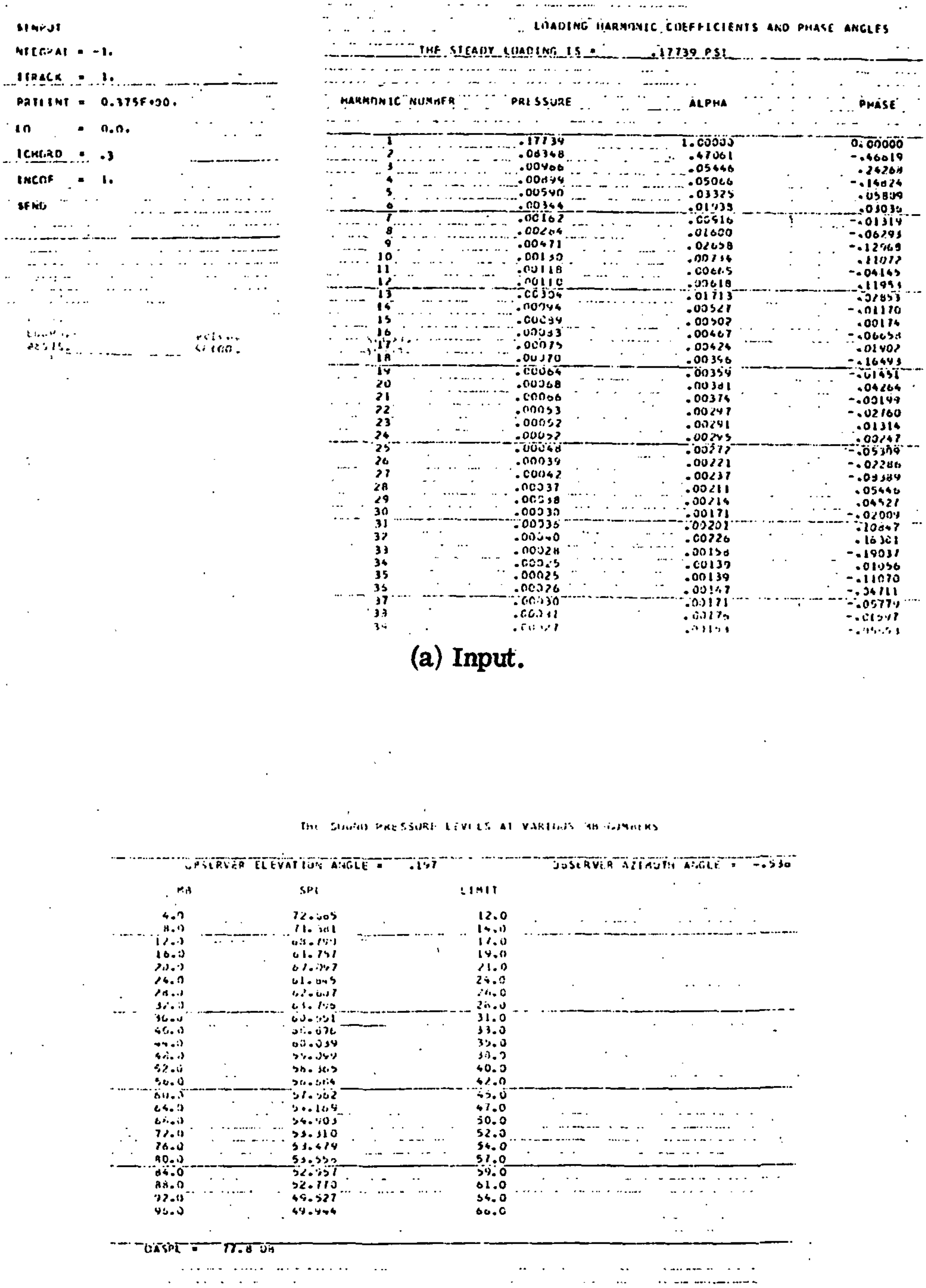
(a) Input.



```
              THE SOUND PRESSURE LEVELS AT VARIOUS MB NUMBERS

   OBSERVER ELEVATION ANGLE =    .197           OBSERVER AZIMUTH ANGLE =  -.536

      MB              SPL              LIMIT

      4.0            73.899            12.0
      8.0            74.140            14.0
     12.0            71.575            17.0
     16.0            63.513            19.0
     20.0            69.407            21.0
     24.0            63.989            24.0
     28.0            64.686            26.0
     32.0            66.659            28.0
     36.0            64.689            31.0
     40.0            64.621            33.0
     44.0            61.984            35.0
     48.0            62.994            38.0
     52.0            62.531            40.0
     56.0            60.186            42.0
     60.0            61.281            45.0
     64.0            59.357            47.0
     68.0            61.065            50.0
     72.0            59.325            52.0
     76.0            59.417            54.0
     80.0            58.578            57.0
     84.0            57.892            59.0
     88.0            58.013            61.0
     92.0            57.573            64.0
     96.0            56.491            66.0

   CASPL =   80.3 DB
```

(b) Output.

Figure 6.- Printed input and output for Case B.



```
$INPUT

NTEGRAL = -1.

ITRACK  =  1.

PATIENT =  0.375E+00.

LO      =  0.0.

ICHORD  =  .3

INCOF   =  1.

$END
```

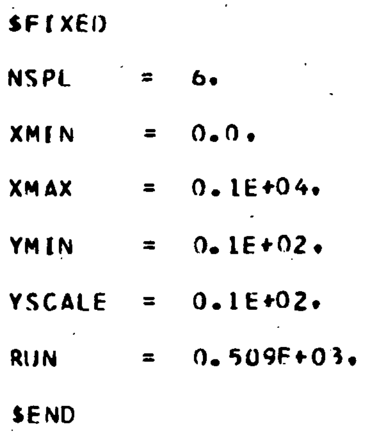
```
                    LOADING HARMONIC COEFFICIENTS AND PHASE ANGLES

           THE STEADY LOADING IS =        .17739 PSI

  HARMONIC NUMBER        PRESSURE              ALPHA                 PHASE

          1              .17739               1.00000               0.00000
          2              .08348                .47061               -.46619
          3              .00966                .05446                .24268
          4              .00899                .05066               -.14824
          5              .00590                .03325                .05809
          6              .00344                .01938                .03036
          7              .00162                .00916               -.01319
          8              .00284                .01600               -.06293
          9              .00471                .02658               -.12969
         10              .00130                .00734                .11072
         11              .00118                .00665               -.04145
         12              .00110                .00618                .11953
         13              .00304                .01713                .07853
         14              .00094                .00527               -.01170
         15              .00089                .00502                .00174
         16              .00083                .00467               -.06658
         17              .00075                .00424                .01902
         18              .00070                .00396               -.16493
         19              .00064                .00359               -.01451
         20              .00068                .00381                .04264
         21              .00066                .00374               -.00199
         22              .00053                .00297               -.02760
         23              .00052                .00291                .01314
         24              .00052                .00295                .00747
         25              .00048                .00272               -.05309
         26              .00039                .00221               -.02286
         27              .00042                .00237               -.09389
         28              .00037                .00211                .05446
         29              .00038                .00214                .04527
         30              .00030                .00171               -.02009
         31              .00036                .00201                .10847
         32              .00040                .00226                .16301
         33              .00028                .00158               -.19037
         34              .00025                .00139                .01056
         35              .00025                .00139               -.11070
         36              .00026                .00147               -.34711
         37              .00030                .00171               -.05779
         38              .00031                .00175               -.01597
         39              .00027                .00154               [illegible]
```

(a) Input.

```
              THE SOUND PRESSURE LEVELS AT VARIOUS HARMONICS

      OBSERVER ELEVATION ANGLE =    .197              OBSERVER AZIMUTH ANGLE =  -.536

      MB              SPL              LIMIT

     4.0            72.665             12.0
     8.0            71.581             14.0
    12.0            63.799             17.0
    16.0            61.757             19.0
    20.0            67.097             21.0
    24.0            61.645             24.0
    28.0            62.607             26.0
    32.0            63.795             28.0
    36.0            60.551             31.0
    40.0            58.676             33.0
    44.0            60.039             35.0
    48.0            59.099             38.0
    52.0            56.365             40.0
    56.0            56.564             42.0
    60.0            57.562             45.0
    64.0            54.169             47.0
    68.0            54.903             50.0
    72.0            53.310             52.0
    76.0            53.479             54.0
    80.0            53.555             57.0
    84.0            52.957             59.0
    88.0            52.770             61.0
    92.0            49.527             64.0
    96.0            49.944             66.0

   OASPL =    77.8 DB
```

(b) Output.

Figure 7.- Printed input and output for Case C.

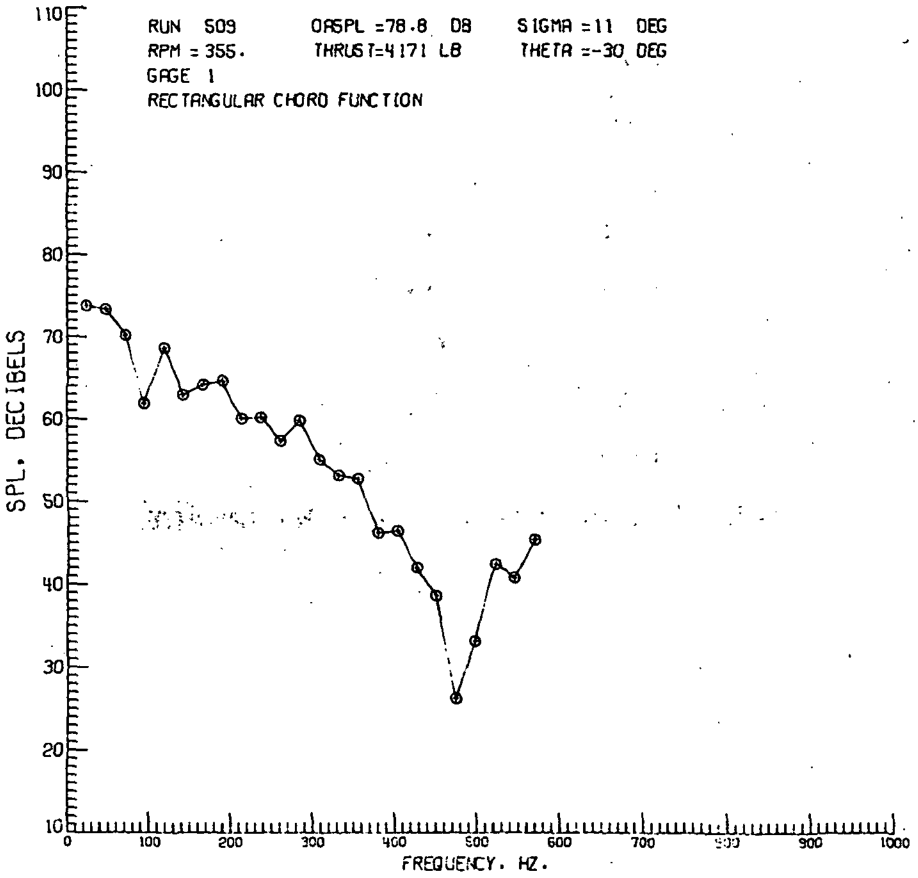


```
$FIXED
NSPL    =  6,
XMIN    =  0.0,
XMAX    =  0.1E+04,
YMIN    =  0.1E+02,
YSCALE  =  0.1E+02,
RUN     =  0.509E+03,
$END
```

Figure 8.- Parametric input for program SPLPLT.

```
$INPUT
NNPLOT   =   0.
$END
```

(a) Input.

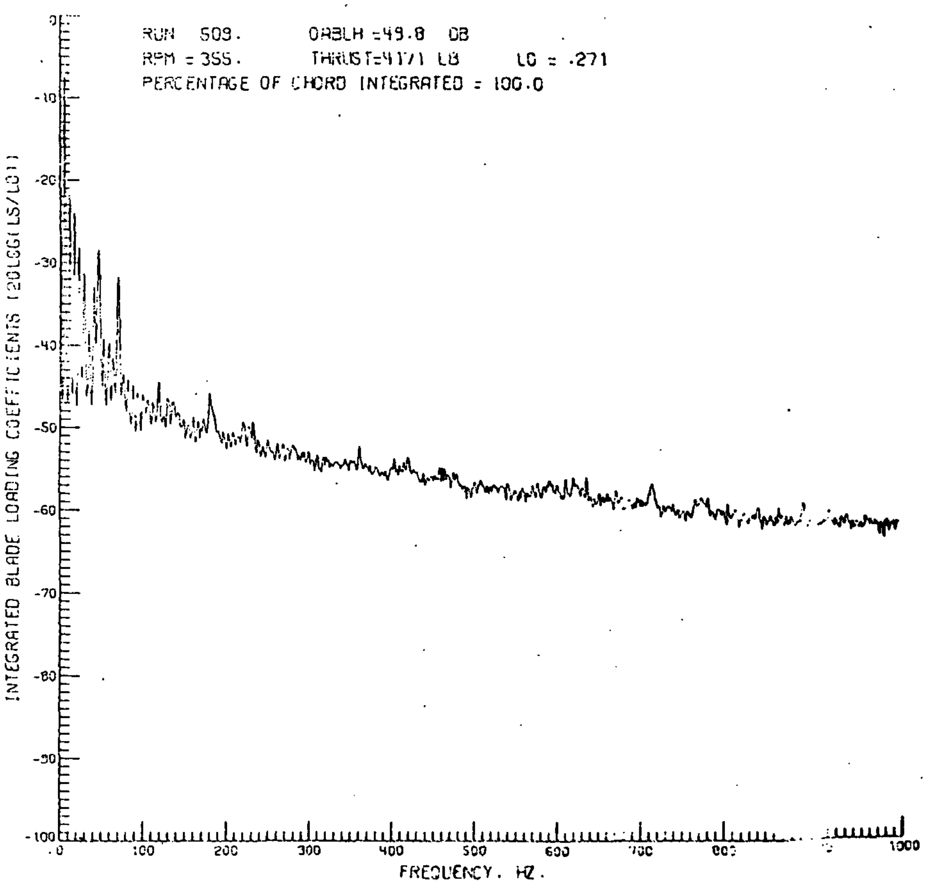


(b) Graphic output.

Figure 9.- SPLPLT input and graphic output for Case A.

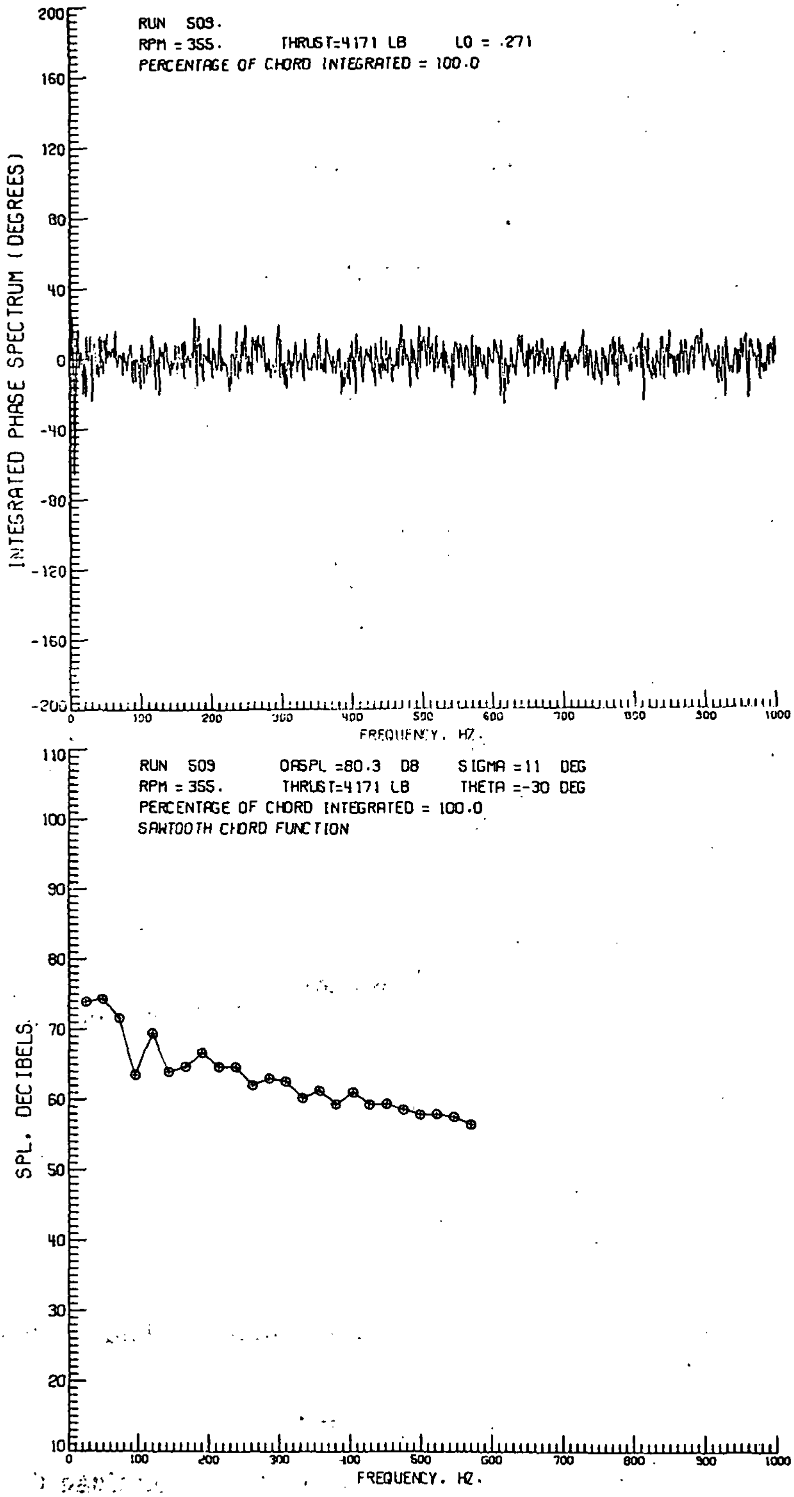
```
$INPUT
NNPLOT   =   2.
$END
```

(a) Input.



(b) Graphic output.

Figure 10.- SPLPLT input and graphic output for Case B.

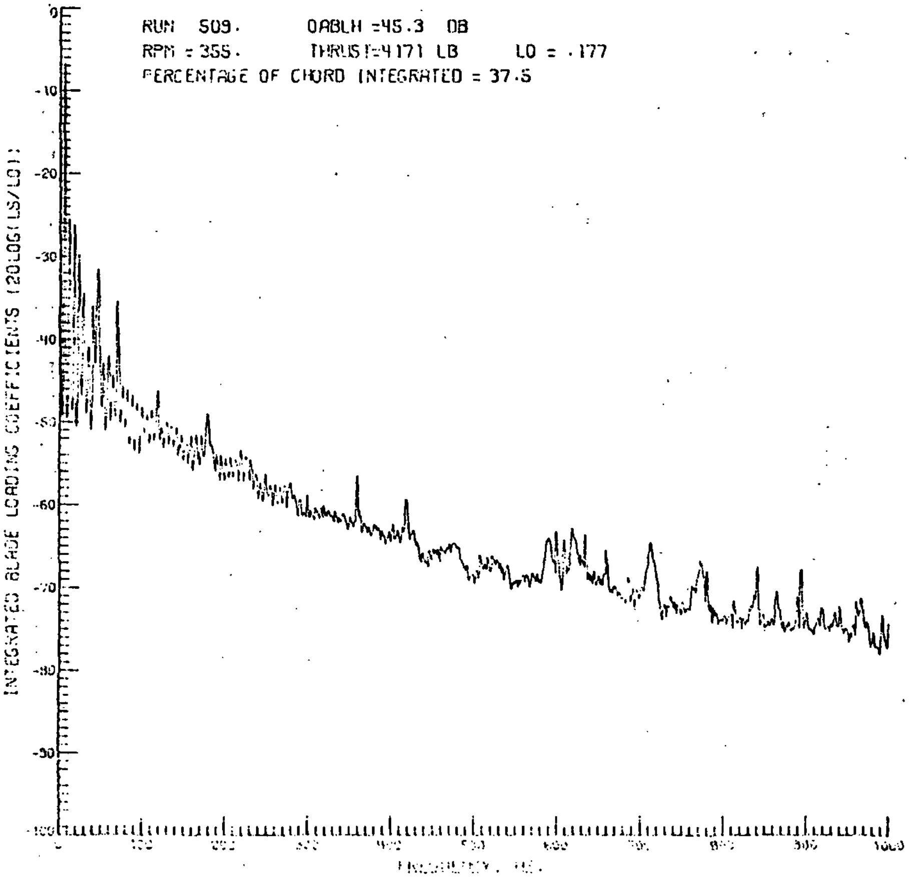


(b) Concluded.

Figure 10.- Concluded.

```
$INPUT

NNPLOT   =  2.

$END
```

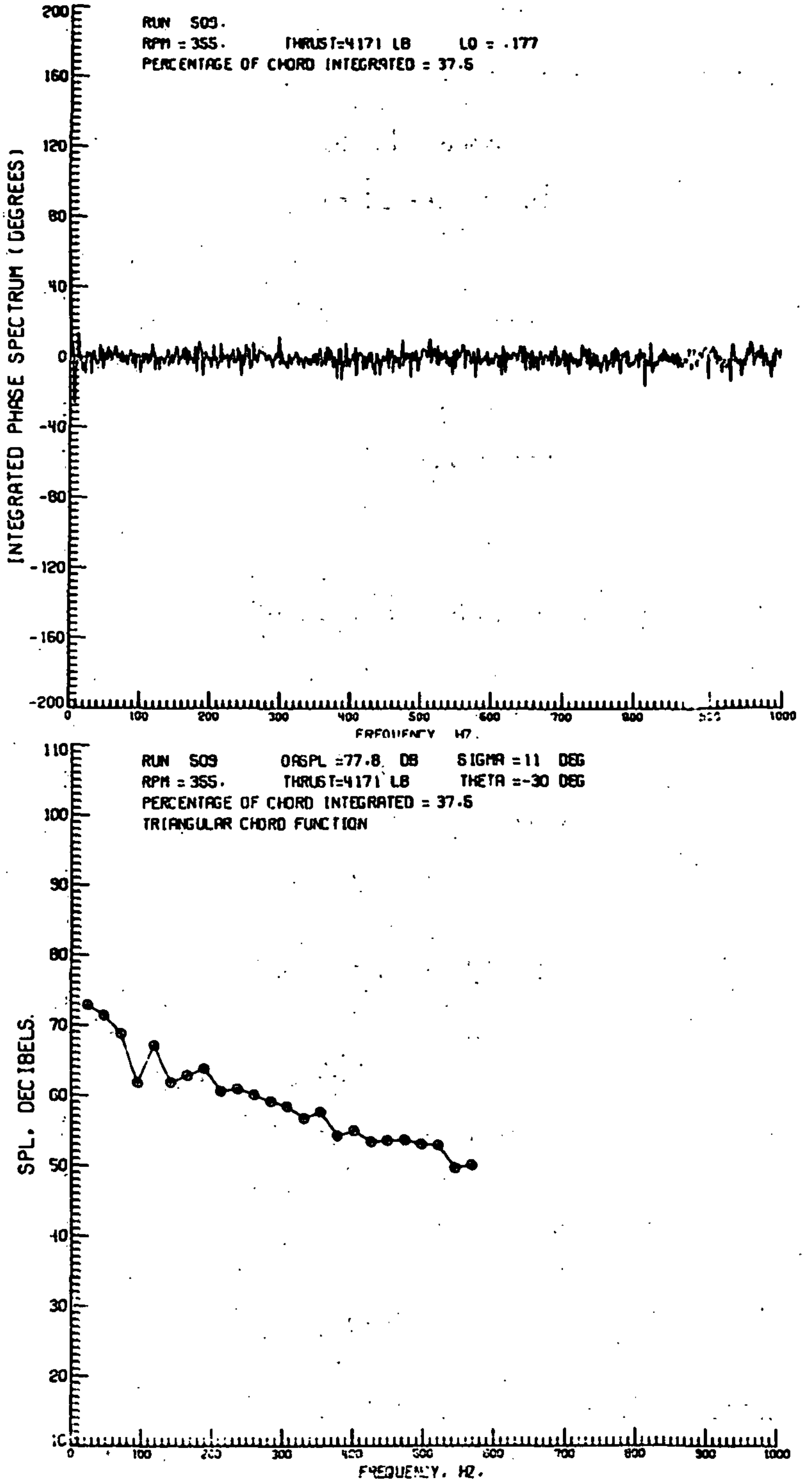
(a) Input.



RUN 509.   OABLH =45.3 DB
RPM = 355.   THRUST=4171 LB   LO = .177
PERCENTAGE OF CHORD INTEGRATED = 37.5

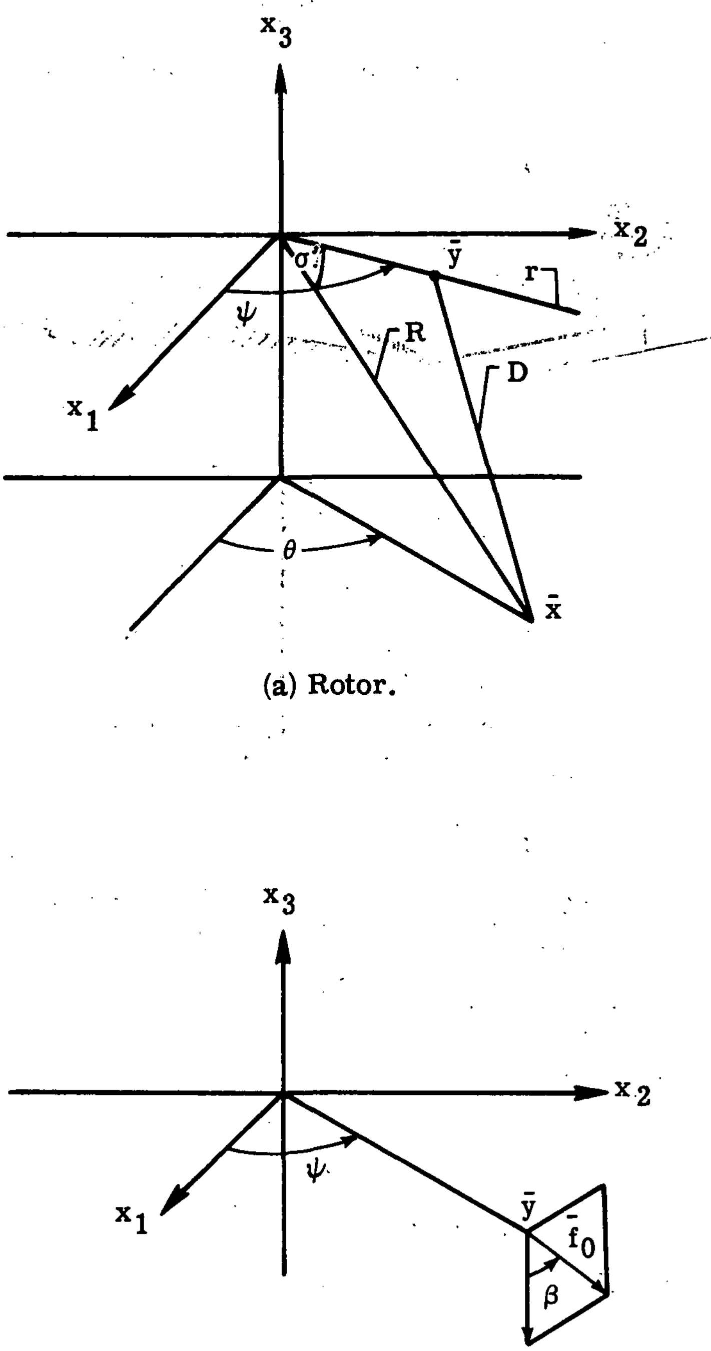
(b) Graphic output.

Figure 11.- SPLPLT input and graphic output for Case C.



(b) Concluded.

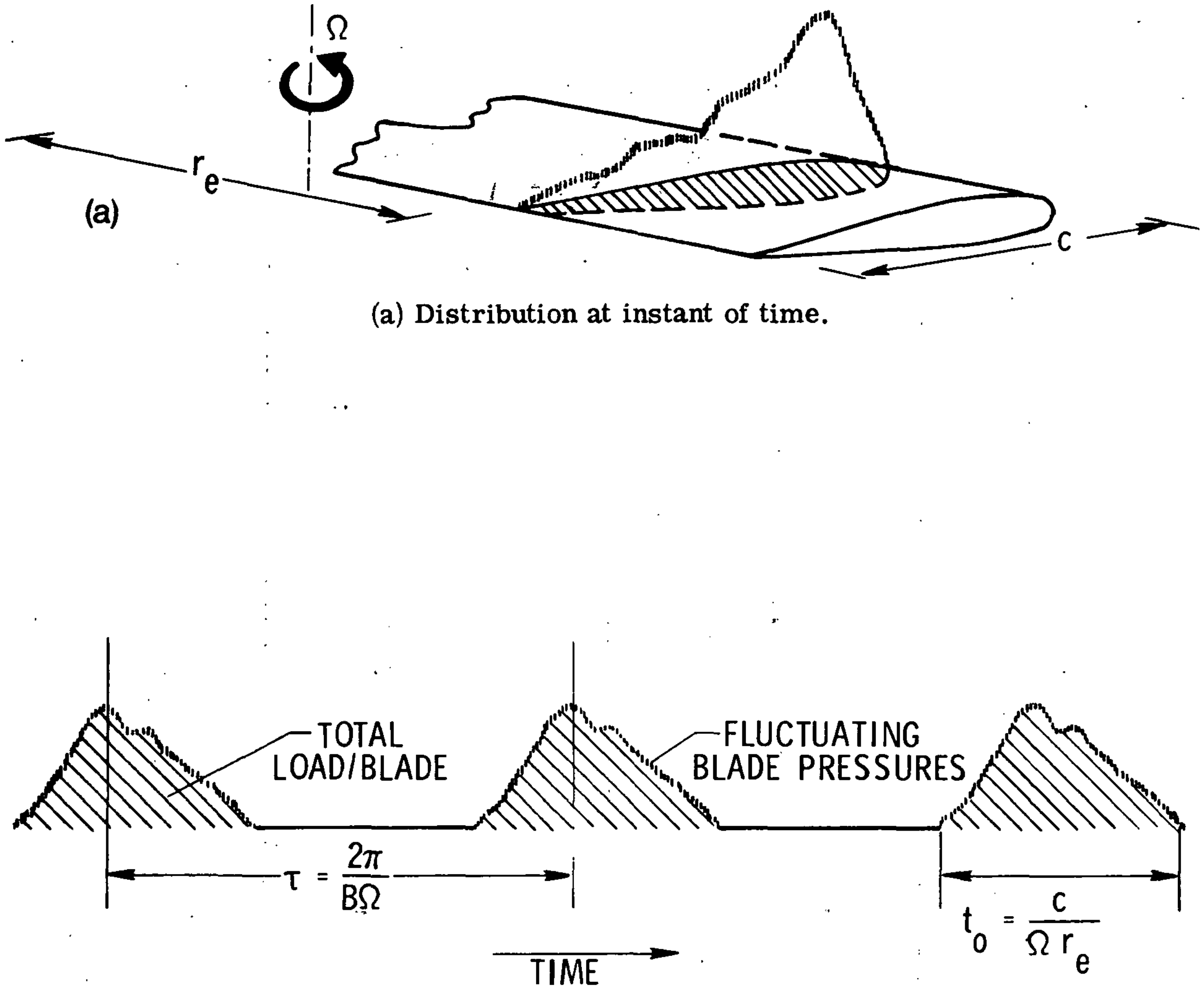
Figure 11.- Concluded.



(a) Rotor.

(b) Forces.

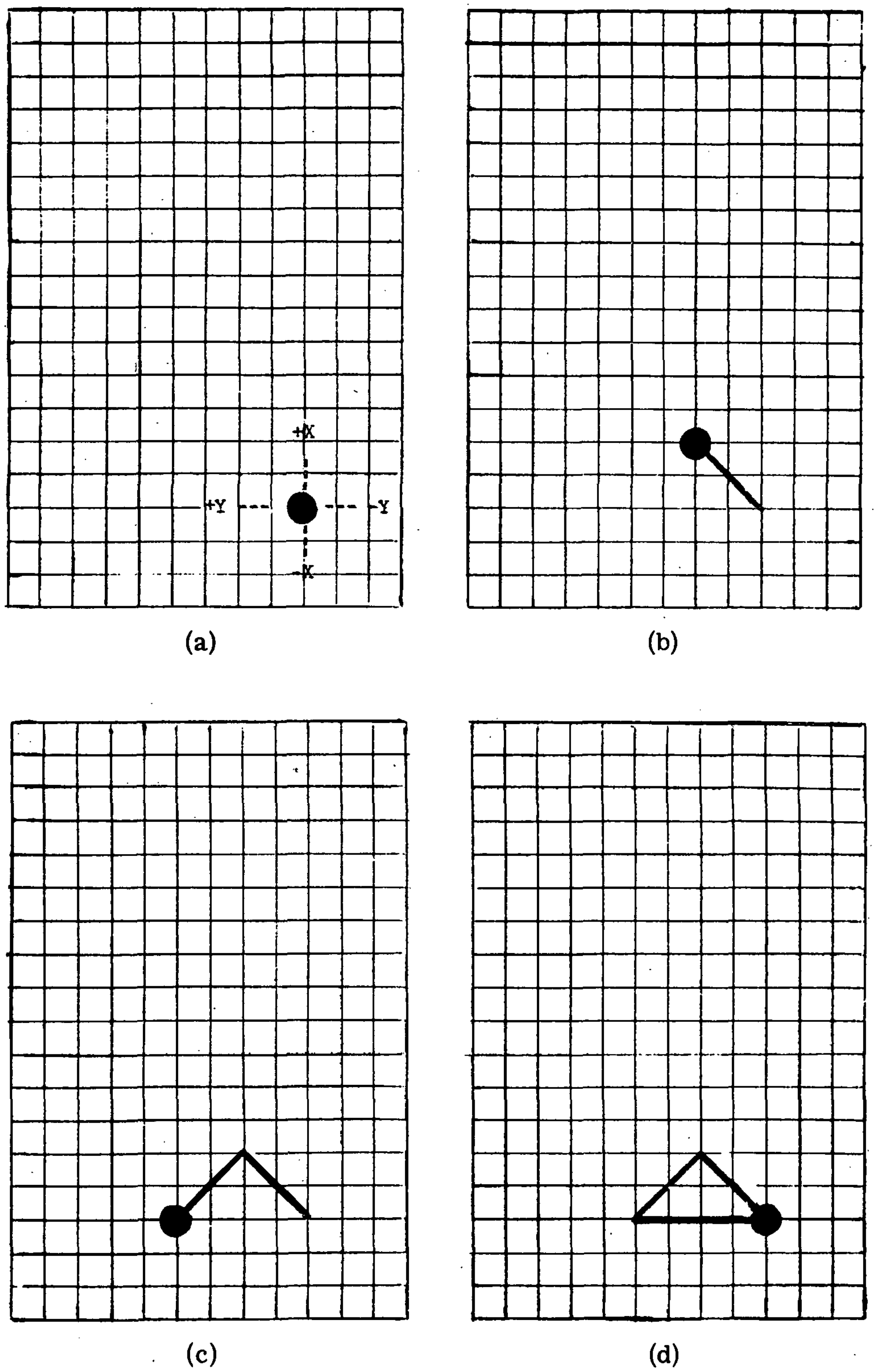
Figure 12.- Coordinate system.


Ω
r_e
(a)
c
TOTAL LOAD/BLADE
FLUCTUATING BLADE PRESSURES
$\tau = \frac{2\pi}{B\Omega}$
$t_0 = \frac{c}{\Omega r_e}$
TIME

(a) Distribution at instant of time.

(b) Time history of loading distribution.

Figure 13.- Arbitrary chordwise loading distribution.



(a) (b) (c) (d)

Figure 14.- CALPLT pen positions.

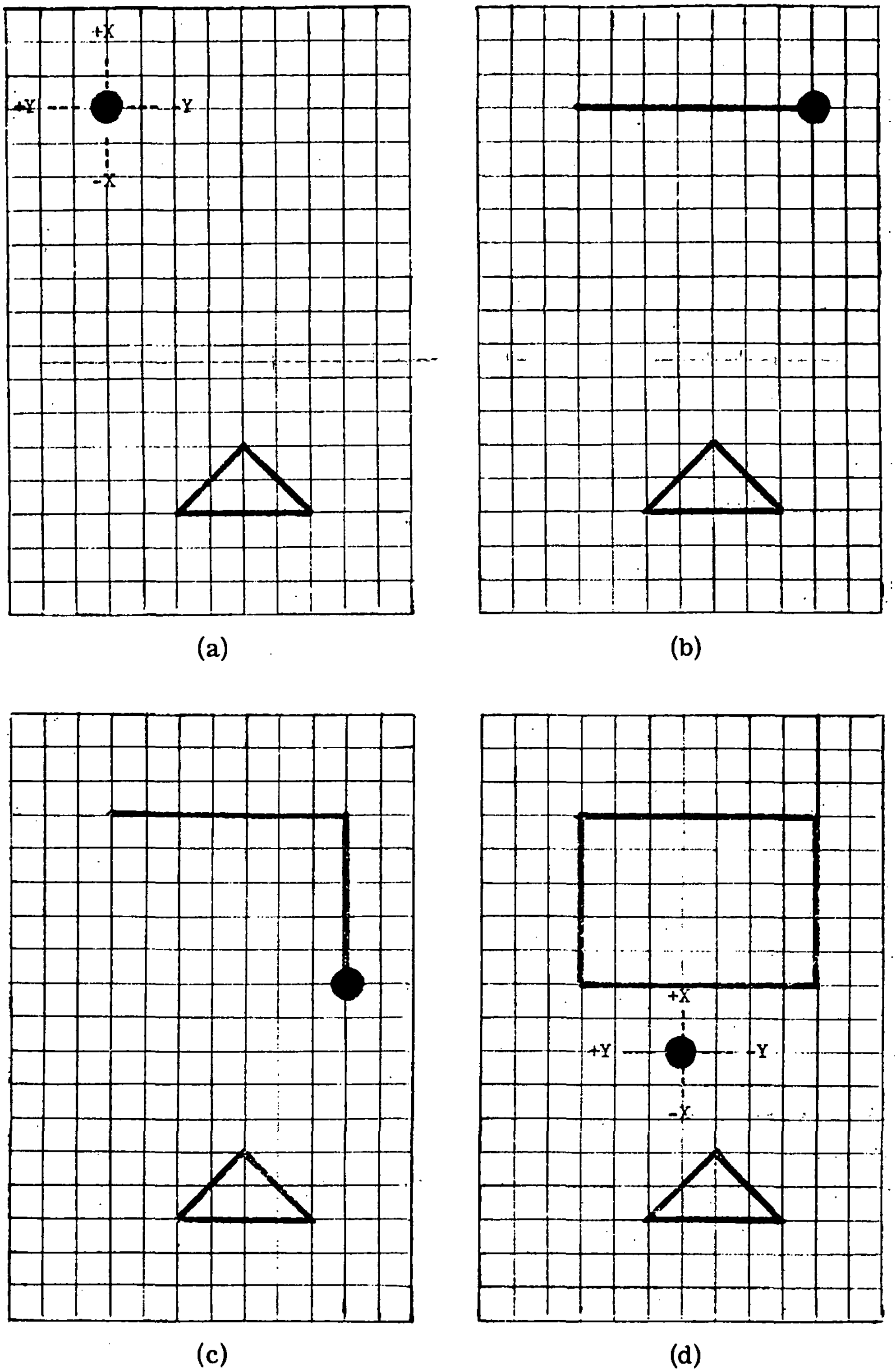
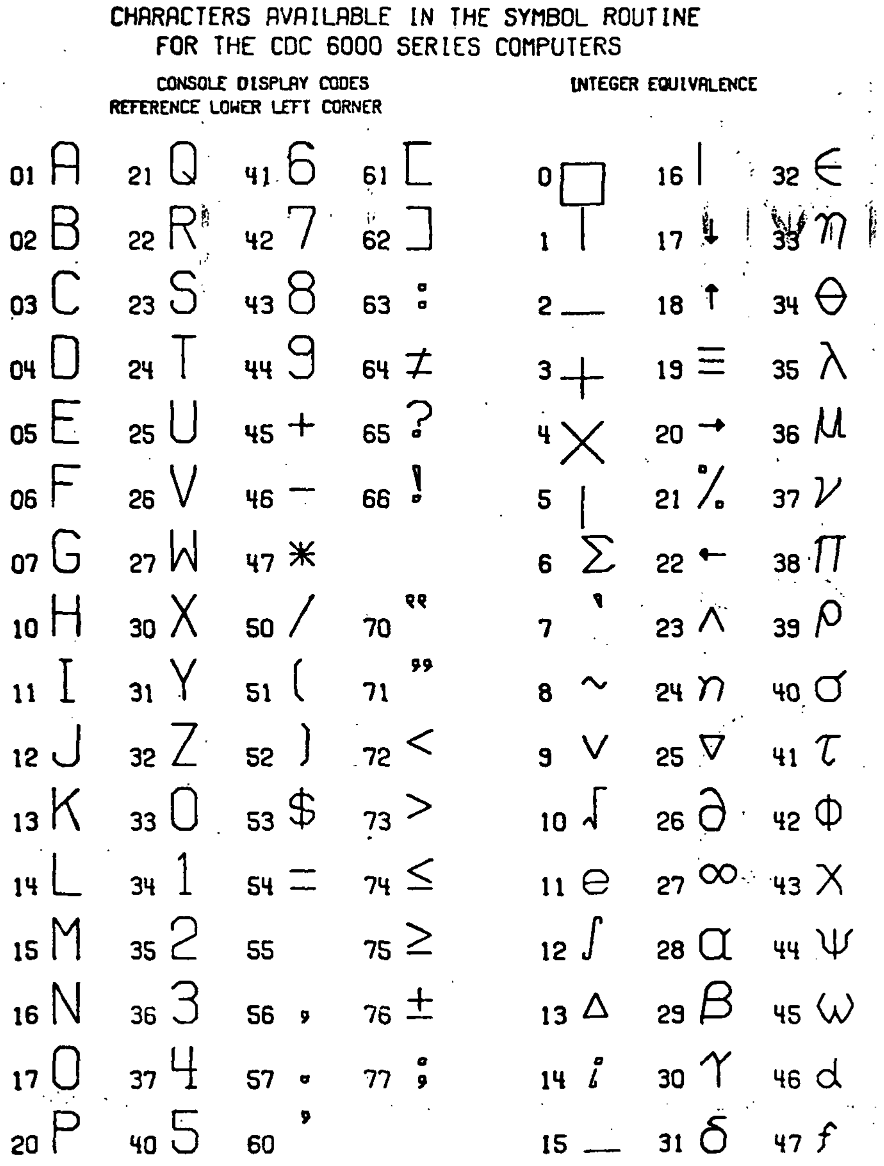


Figure 15.- CALPLT pen positions.



CHARACTERS AVAILABLE IN THE SYMBOL ROUTINE
FOR THE CDC 6000 SERIES COMPUTERS

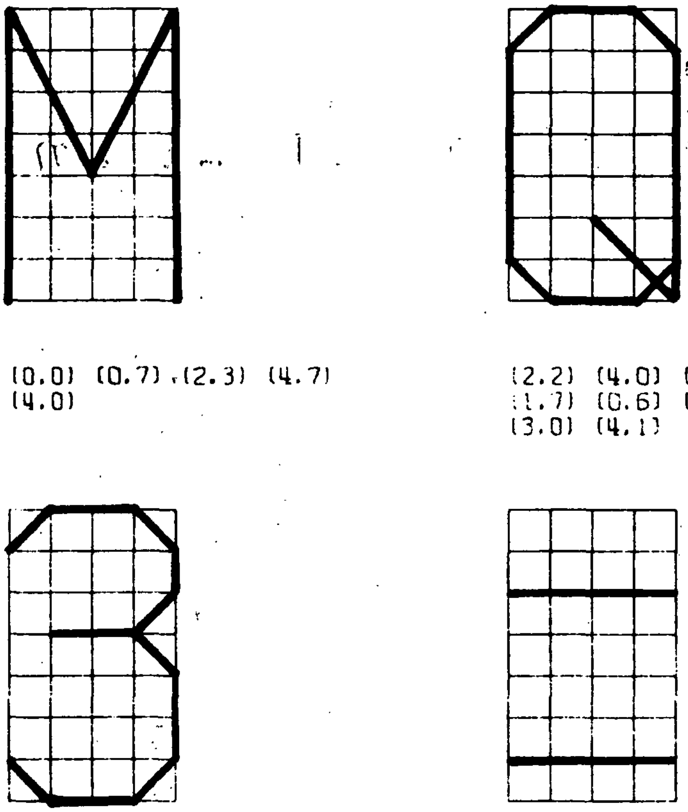
CONSOLE DISPLAY CODES
REFERENCE LOWER LEFT CORNER

01	A	21	Q	41	6	61	[
02	B	22	R	42	7	62	]
03	C	23	S	43	8	63	:
04	D	24	T	44	9	64	≠
05	E	25	U	45	+	65	?
06	F	26	V	46	−	66	!
07	G	27	W	47	*		
10	H	30	X	50	/	70	“
11	I	31	Y	51	(	71	”
12	J	32	Z	52	)	72	<
13	K	33	0	53	$	73	>
14	L	34	1	54	=	74	≤
15	M	35	2	55		75	≥
16	N	36	3	56	,	76	±
17	O	37	4	57	.	77	;
20	P	40	5	60	’		

INTEGER EQUIVALENCE

0	□	16	|	32	∈
1	⊤	17	↓	33	η
2	_	18	↑	34	θ
3	+	19	≡	35	λ
4	×	20	→	36	μ
5	|	21	%	37	ν
6	Σ	22	←	38	π
7	’	23	∧	39	ρ
8	~	24	η	40	σ
9	∨	25	∇	41	τ
10	√	26	∂	42	Φ
11	e	27	∞	43	χ
12	∫	28	α	44	ψ
13	△	29	β	45	ω
14	i	30	γ	46	d
15	—	31	δ	47	f

Figure 16.- Plotting characters.



(0.0) (0.7) (2.3) (4.7)
(4.0)

(2.2) (4.0) (4.6) (3.7)
(1.7) (0.6) (0.1) (1.0)
(3.0) (4.1)

(0.1) (1.0) (3.0 (4.1)
(4.3) (3.4) (1.4 (3.4)
(4.5) (4.6) (3.7 (1.7)
(0.6)

(0.1) (4.1) (7.0) (4.5)
(0.5)

Figure 17.- Sample of characters with their X and Y offsets.

CALCOMP SYMBOL	HOLLERITH CARD CODE (MULTIPUNCHED COLUMN)	CONSOLE DISPLAY CODE	CDC PRINTER SYMBOL
'	$^{0}6_{8}$	60	≡
[	7_{8}	61	[
]	$^{0}2_{8}$	62	]
:	2_{8}	63	:
≠	4_{8}	64	≠
?	$^{0}5_{9}$	65	→
!	11_{0}	66	∨
“	$^{11}5_{8}$	70	↑
”	$^{11}6_{8}$	71	↓
<	12_{0}	72	<
>	$^{11}7_{8}$	73	>
≤	5_{8}	74	≤
≥	$^{12}5_{8}$	75	≥
±	$^{12}6_{8}$	76	¬
;	$^{12}7_{8}$	77	;

Figure 18.- Extended card codes.

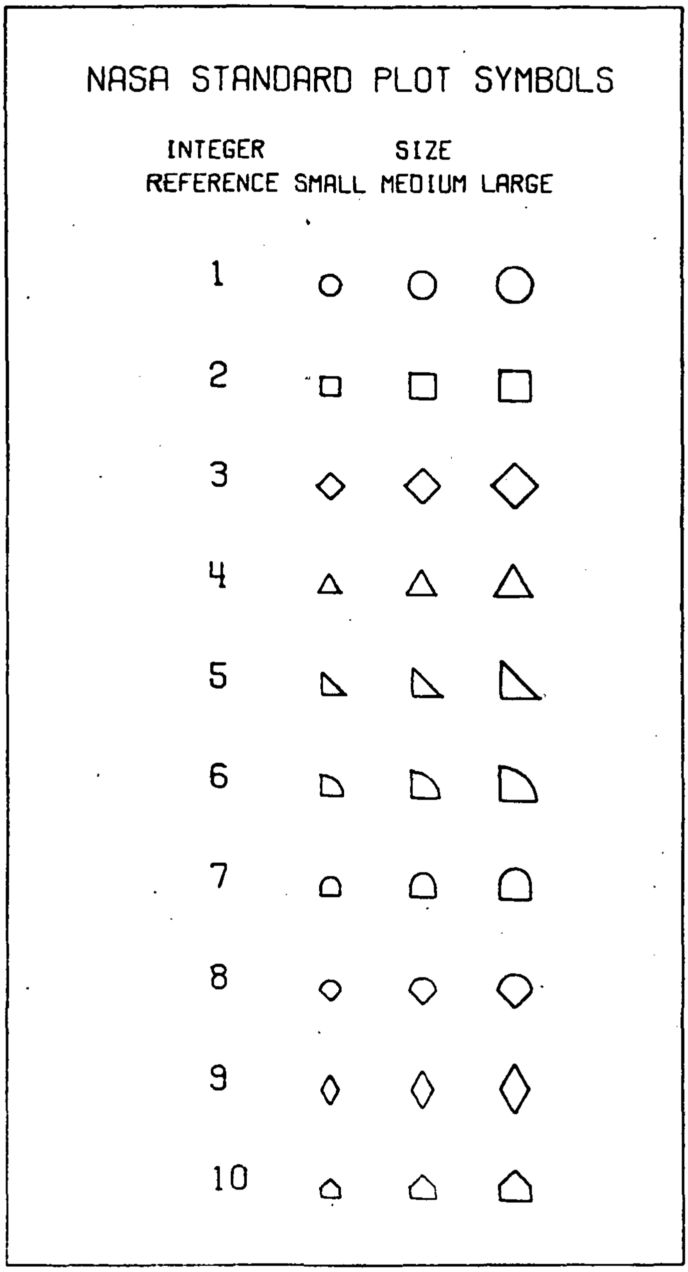


Figure 19.- Plot symbols.

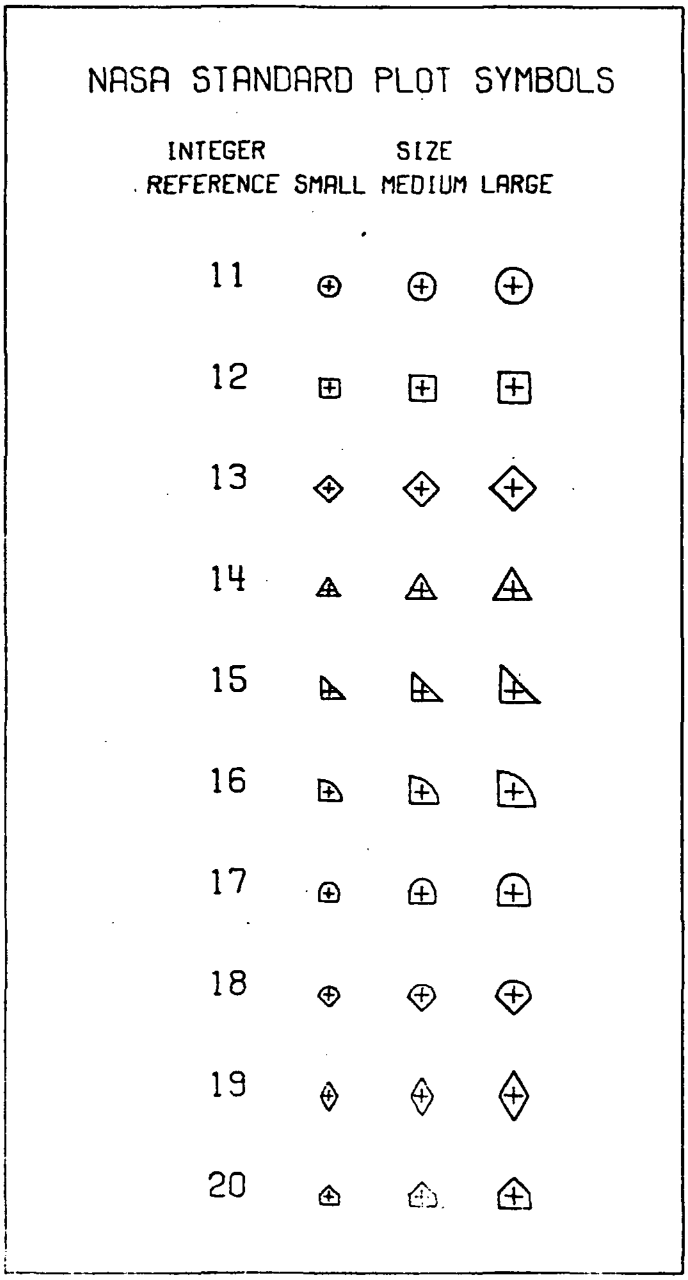


Figure 20.- Plot symbols.

*U.S. GOVERNMENT PRINTING OFFICE: 1976 - 635-275/70

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546