FROM BCSL
Copy Authorized

UNIVERSITY OF WASHINGTON

DEPARTMENT OF MECHANICAL ENGINEERING



# Sound and Vibration Research Laboratory

SEATTLE, WASHINGTON 98195

(NASA-CR-146420) A STUDY OF SOUND GENERATION IN SUBSONIC ROTORS, VOLUME 2 Final Report (Washington Univ.) 63 p HC $4.50 CSCL 20A

N76-18121

Unclas
G3/07 09635

Final Report on

NASA Grant No. ~~NGR-002-144~~
NGR-48-002-144

# A STUDY OF SOUND GENERATION IN SUBSONIC ROTORS

Report No. ME 73-11

Volume 2 of 2

J. D. Chalupnik, Associate Professor

L. T. Clark, Research Assistant Professor

University of Washington

Department of Mechanical Engineering

Seattle, Washington 98195

## 1.0 INTRODUCTION

Volume 2 of the final report is used to outline four computer programs which were developed during the grant period. All were coded in FORTRAN IV Language for usage on the University of Washington CDC 6400 digital computer.

The discussions here are primarily user input instructions. For two of the programs, ROTOR and AIRFOIL, the background models are developed in some detail in Volume 1 of the report. Program TONE is developed herein and SDATA is outlined here with reference to the literature where the detailed development requires a book to describe. The function of each program is discussed below.

(1) Program AIRFOIL: This program computes the spectrum of radiated sound from a single airfoil immersed in a laminar flow field. The mechanism is force fluctuations which are related to the unsteady wake momentum. Input required is the aerodynamic properties of the turbulent wake.

(2) Program ROTOR: The purpose of program ROTOR is to provide an extension of the single airfoil in AIRFOIL to a rotating frame. This is a model for sound generation in subsonic rotors. The only broad band mechanism currently in the program is the airfoil wake mechanism. Program ROTOR also computes tone sound generation due to the steady state forces on the blades (the Gutin result). Input requirements are similar to those for AIRFOIL.

(3) Program TONE: A moving source analysis is used to generate a time series for an array of forces moving in a circular path as would be the case for a rotor. This analysis has great flexibility in that arbitrary blade spacings, blade forces and source motions can be easily accommodated. Broad band effects have been included using autoregression methods. The resultant time series are Fourier transformed using a Fast Fourier Transform to present the sound radiation in the more normal spectral form. With some study, a user can make this program a very valuable tool in sound generation investigations.

(4) Program SDATA: This program is a standard time series analysis package. It will read in two discrete time series and form auto and cross covariances and normalize these to form correlations. The program will then transform the covariances to yield auto and cross power spectra by means of a Fourier transformation. The spectral data are presented in terms of amplitude and phase as well as the coherency spectrum.

A final note on these programs is in order. These were developed as tools to be used in the work performed under the grant. There was no intention to provide the programs as complete packages for general usage. The content is general enough, however, that with some effort a user should be able to employ the programs to advantage. He is cautioned to understand the analytical models before he becomes too involved in turning around decks. No program is a substitute for this understanding. Tools must be skillfully used.

To the end of providing a usable tool the programs are coded in variable names which read very like those of the physical variables they represent. For example the density will be RHO and the longitudinal turbulence covariance will be U1U1, etc. This should facilitate the reading of the program listings.

Each of the programs is discussed in the following sections.

## 2.0 PROGRAM AIRFOIL

The fortran program AIRFOIL is used to compute the sound radiated from a single airfoil due to those mechanisms related to the airfoil wake turbulence. A development of the model was presented in detail in Volume 1 of this report. Through the model, the sound radiation is related to the turbulent structure in the wake of the airfoil.

The equation which forms the basic result of the model and that programmed here is

$$G_p(\vec{x},\omega) = \sum_{i=1}^{2} \sum_{j=1}^{2} 16\ S\ L_1^2(u_i u_j,\omega)\ L_2(u_i u_j,\omega)\ L_3(u_i u_j,\omega)$$

$$\omega^4 \int_{y_2} A_i A_j\ \overline{u_i u_j}(y_2,\omega)\ dy_2$$

with: $A_1 = \rho_o U_1 \sin\psi/(4\pi a_o^3\ |\vec{x}|)$

$A_2 = \rho_o \overline{U}_1 \cos\psi/(4\pi a_o^3\ |\vec{x}|)$ .

Nomenclature used here is the same as that of Volume 1.

The integration noted above is performed using a Lagrangian method. This subroutine is part of the University of Washington Computer Library. Any simple numerical integration scheme may be substituted for this subroutine. In the program the subroutine is labeled LAGRAN.

### 2.1 Program Input

AIRFOIL is a simple computational program. It consists of three subprograms. The main program is AIRFOIL in which the actual computations are made. The next is subroutine DREAD in which data are read into and conditioned to form the variables used in the actual computations. Finally, DWRITE in which the results of the computations are written out.

Input to the program is accomplished using IBM cards. In the following we will list each card type, the variable names and their descriptions.

CARD 1, FORMAT(7F10.2)

| | | |
|---|---|---|
| PATM | atmospheric static pressure | (atm) |
| TATM | " " temperature | ($^{o}$C) |
| SPAN | airfoil length | (cm) |
| XLAM | length in the stream direction (this is a dead variable in the current program so any input is okay | (cm) |
| DEL | wake thickness | (cm) |
| RAD | distance from airfoil to sound observer | (m) |

CARD 2, FORMAT(2I5)

| | |
|---|---|
| NPF | number of frequency points where data will be input |
| NPY | number of positions in the wake where data will be given |

CARD 3, FORMAT(7F10.2)

| | | |
|---|---|---|
| Y(I) I = 1,NPY | positions in the wake measured relative to the wake centerline | (cm) |

Note that if more than 7 entries are needed, additional cards will be required for this variable.

CARD 4, FORMAT(7F10.2)

| | | |
|---|---|---|
| FREQ(I) | frequencies at which the following correlation lengths are to correspond | (Hz) |
| XL(I) | correlation length in stream direction for the longitudinal turbulent component | (cm) |

| | | |
|---|---|---|
| YL(I) | correlation length across the wake for the longitudinal turbulence component | (cm) |
| ZL(I) | correlation length in the span direction for the longitudinal turbulence component | (cm) |

CARD 5, FORMAT(7F10.2)

| | | |
|---|---|---|
| FREQ(I) | same as for CARD 4 | |
| XT(I) | same as for CARD 4, but for the transverse turbulence component | (cm) |
| YT(I) | same as for CARD 4, but for the transverse turbulence component | (cm) |
| ZT(I) | same as for CARD 4, but for the transverse turbulence component | (cm) |

Notice that CARD's 4 and 5 are read inside a DO loop. The statements are

```
    DO 16 I = 1,NPF
    READ(5,3)  FREQ(I),XL(I),YL(I),ZL(I)
 16 READ(5,3)  FREQ(I),XT(I),YT(I),ZT(I)
```

Certainly more than just the two cards will be required here since more than one spectral point will generally be required. For this outline, we will continue with CARD 6 keeping in mind that CARD 6 refers to a new input type of data rather than the actual number of the card in the input file.

CARD 6, FORMAT(7F10.2)

| | | |
|---|---|---|
| XI(I)<br>1 = 1,NPY | ratio of wake velocity at Y(I) to the velocity in the free stream UMEAN | (--) |

CARD 7, FORMAT(7F10.2)

| | | |
|---|---|---|
| DDB1(I)<br>1 = 1,NPF | ratio of spectral level at FREQ(I) to the RMS level; this is for the longitudinal component of turbulence | (dB) |

DDB1 can be interpreted from the equation for its usage which is

$$u_1'(f) = u_1^2 * 10. **(-DDB1(I)/10.)$$

$u_1'(f)$ is the actual spectral level sought, and

$u_1^2$ is the RMS level for the longitudinal component of turbulence.

CARD 8, FORMAT(7F10.2)

| | | |
|---|---|---|
| DDB2(I)<br>I = 1,NPF | the ratio of the spectral level at FREQ(I) to the RMS level for the shear component of turbulence | (dB) |

see the note for DDB1

CARD 9, FORMAT(7F10.2)

| | | |
|---|---|---|
| DDB3(I) | the ratio of the spectral level at FREQ(I) to the RMS level for the transverse component of turbulence | (dB) |

CARD 10, FORMAT(7F10.2)

| | |
|---|---|
| U11(I)<br>I = 1,NPY | spatial distribution of the RMS level of longitudinal turbulence across the wake, normalized with UMEAN |

CARD 11, FORMAT(7F10.2)

| | |
|---|---|
| U12(I)<br>I = 1,NPY | spatial distribution of the RMS level of the shear stress across the wake, normalized with UMEAN |

CARD 12, FORMAT(7F10.2)

| | |
|---|---|
| U22(I) | spatial distribution of the RMS level of the transverse turbulence across the wake, normalized with UMEAN |

This completes the input for the program.

Program output consists of a listing of the input variables and of the spectral distribution of the radiated sound pressure at each radiation angle $0^\circ$ through $180^\circ$ in $10^\circ$ increments. The radiated sound is presented in dB by normalizing the pressure with a reference $20 \mu N/m^2$.

## 2.2 Program Listing

```
      PROGRAM AIRFOIL (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

C     THIS PROGRAM COMPUTES THE SPECTRUM OF RADIATED SOUND FROM
C     CORRELATION AND SPECTRAL DATA GIVEN FOR TWO POINT VELOCITY
C     CORRELATIONS IN THE WAKE OF A SINGLE AIRFOIL

      DIMENSION XINTG(11),XINT1(11),XINT(11)
      DIMENSION FREQ(20),XL(20),YL(20),ZL(20),U1U1(11,20),XI(11),
     1U1U2(11,20),U2U2(11,20),Y(11),XM(11),XOMEG(20),
     2PSD(20),SPL(20),VCL(20),VCT(20),VAR(3),XT(20)
      DATA PI/3.14159/

C     READ IN DATA .

C     PATM = STATIC PRESSURE (MM HG)
C     TATM = STATIC TEMPERATURE (DEG C)
CC     SPAN = AIRFOIL LENGTH (CM)
C     XLAM = STREAM INTEGRATION LENGTH (CM)
C     DEL = WAKE THICKNESS (CM)
C     UMEAN = FREE STREAM VELOCITY (M/SEC)
C     RAD = DISTANCE TO OBSERVER (M)
C     NPF = NUMBER OF FREQUENCY POINTS IN SPECTRUM
C     NPY = NUMBER OF SPATIAL POINTS IN WAKE
C     Y = LOCATIONS OF THE POINTS IN THE WAKE (CM)
C     FREQ = FREQUENCY ARRAY (HZ)
C     XL = STREAM CORRELATION FOR U1U1 (CM)
C     YL = CORRELATION ACROSS THE WAKE FOR U1U1 (CM)
C     ZL = SPAN CORRELATION FOR U1U1 (CM)
C     XT = STREAM CORRELATION FOR U2U2 (CM)
C     YT = CORRELATION ACROSS THE WAKE FOR U2U2 (CM)
C     ZT = SPAN CORRELATION FOR U2U2 (CM)
C     XI = RATIO OF WAKE VELOCITY TO UMEAN
C     DDB1 = RATIO OF SPECTRAL LEVEL TO RMS - U1U1 (DB)
C     DDB2 = RATIO OF SPECTRAL LEVEL TO RMS - U1U2 (DB)
C     DDB3 = RATIO OF SPECTRAL LEVEL TO RMS - U2U2 (DB)
C     U11 = RMS VALUE OF U1U1 (PERCENT)
C     U12 = RMS VALUE OF U1U2 (PERCENT)
C     U22 = RMS VALUE OF U2U2 (PERCENT)

      CALL DREAD(AO,RHO,PI,BM,XOMEG,VCL,VCT,SPAN,XLAM,DEL,XM,
     1RAD,U1U1,U1U2,U2U2,NPF,NPX,NPY,X,Y,FREQ,PATM,TATM,XL,XT)

C     COMPUTE RADIATION PATTERN

      COEF1=(RHO/(4.*PI*RAD))**2
      COEF2=COEF1*SPAN*BM**2
      DO 2000 KK=1,19
      UKK=KK
      CODE1=UKK
      THETA=-90.+10.*(UKK-1.)
      THET=THETA*PI/180.
      ST=SIN(THET)
      CT=COS(THET)
      STS=ST**2
      CTS=CT**2
      STCT=ST*CT
```

```
      COEF3=COEF2
C     COMPUTE SPECTRUM
      DO 1000 K=1,NPF
C     COMPUTE INTEGRALS OVER THE WAKE VOLUME
      XINT(K)=0.0
      DO 300 II=1,3
      DO 100 I=1,NPY
      VAR(1)=STS*U1U1(K,I)**2*VCL(K)*XL(K)
      VAR(2)=STCT*U1U2(K,I)**2*(VCL(K)+VCT(K))/2.*(XL(K)+XT(K))/2.
      VAR(3)=CTS*U2U2(K,I)**2*VCT(K)*XT(K)
  100 XINTG(I)=VAR(II)
      WRITE(6,10)  (XINTG(I),I=1,NPY)
   10 FORMAT(7E10.2)
      CALL LAGRAN(Y,XINTG,NPY,1,1,0,ANS,IRROR)
  300 XINT(K)=ANS+XINT(K)
      PSD(K)=XINT(K)*COEF3*XOMEG**4
      PSD(K)=ABS(PSD(K))
      WRITE(6,11) ANS
   11 FORMAT(4H ANS   1E20.5)
 1000 SPL(K)=10.*ALOG10(PSD(K)/400.E-12)
      CALL DWRITE(AO,RHO,PI,BM,XOMEG,VCL,VCT,SPAN,XLAM,DEL,
     1XM,RAD,U1U1,U1U2,U2U2,X,Y,THETA,PSD,SPL,NPY,NPX,NPF,
     2FREQ,CODE1,PATM,TATM,XL,XT)
 2000 CONTINUE
      STOP
      END
      SUBROUTINE DREAD(AO,RHO,PI,BM,XOMEG,VCL,VCT,SPAN,XLAM,DEL,XM,
     1RAD,U1U1,U1U2,U2U2,NPF,NPX,NPY,X,Y,FREQ,PATM,TATM,XL,XT)
C     THIS SUBROUTINE READS AND CONDITIONS INPUT DATA
      DIMENSION FREQ(20),XL(20),YL(20),ZL(20),U1U1(11,20),XI(11),
     1U1U2(11,20),U2U2(11,20),Y(11),XM(11),XOMEG(20),
     2 PSD(20),SPL(20),VCL(20),VCT(20)
      DIMENSION DDB1(20),DDB2(20),DDB3(20),U11(11),U12(11),
     1U22(11),DF1(20),DF2(20),DF3(20),XT(20),YT(20),ZT(20)
      DATA CR,CO,GAM/288.,1.0,1.4/
      READ(5,1) PATM,TATM,SPAN,XLAM,DEL,UMEAN,RAD
    1 FORMAT(7F10.2)
      CMFT = .0328
      PATM = 2116.*PATM
      TATM=(TATM+32.)*9./5.+460.
      SPAN=SPAN*CMFT
      XLAM=XLAM*CMFT
      DEL=DEL*CMFT
      UMEAN = UMEAN*CMFT*100.
      RAD = RAD*CMFT*100.
```

```
      DSCALE=1.0
      AO=SQRT(GAM*CR*CG*TATM)
      RHO=PATM/(CG*CR*TATM)
      BM=UMEAN/AO
      READ(5,2)   NPF,NPY
    2 FORMAT(3I5)
      READ(5,1)   (Y(I),I=1,NPY)
      DO 18 I=1,NPY
   18 Y(I)=CMFT*Y(I)

      DO 16 I=1,NPF
      READ(5,3)   FREQ(I),XL(I),YL(I),ZL(I)
   16 READ(5,3)   FREQ(I),XT(I),YT(I),ZT(I)

      DO 10 K=1,NPF
      XOMEG(K)=2.*PI*FREQ(K)
      FREQ(K)=FREQ(K)/DSCALE
      XOMEG(K)=XOMEG(K)/DSCALE
      COEF=2./(2.54*12.)
      XL(K)=XL(K)*COEF
      YL(K)=YL(K)*COEF
      ZL(K)=ZL(K)*COEF
      XT(K)=XT(K)*COEF
      YT(K)=YT(K)*COEF
      ZT(K)=ZT(K)*COEF
      VCL(K)=XL(K)*YL(K)*ZL(K)
   10 VCT(K)=XT(K)*YT(K)*ZT(K)

      READ(5,1)   (XI(I),I=1,NPY)

      DO 13 I=1,NPY
   13 XM(I)=UMEAN*XI(I)/AO

      READ(5,1) (DDB1(K),K=1,NPF)
      READ(5,1) (DDB2(K),K=1,NPF)
      READ(5,1) (DDB3(K),K=1,NPF)
      DO 12 K=1,NPF
      DF1(K)=10.**(-DDB1(K)/20.)
      DF2(K)=10.**(-DDB2(K)/20.)
   12 DF3(K)=10.**(-DDB3(K)/20.)
      READ(5,1)   (U11(I),I=1,NPY)
      READ(5,1)   (U12(I),I=1,NPY)
      READ(5,1)   (U22(I),I=1,NPY)
      DO 15 K=1,NPF
      DO 15 I=1,NPY
      U1U1(K,I)=DF1(K)*U11(I)
      U1U2(K,I)=DF2(K)*U12(I)
   15 U2U2(K,I)=DF3(K)*U22(I)

    3 FORMAT(4F10.2)
      RETURN
      END
      SUBROUTINE DWRITE(AO,RHO,PI,BM,XOMEG,VCL,VCT,SPAN,XLAM,DEL,XM,RAD,
     1U1U1,U1U2,U2U2,X,Y,THETA,PSD,SPL,NPY,NPX,NPF,FREQ,CODE1,
     2PATM,TATM,XL,XT)
```

```
      DIMENSION FREQ(20),XL(20),YL(20),ZL(20),U1U1(11,20),X1(11),
     1U1U2(11,20),U2U2(11,20),Y(11),XM(11),XOMEG(20),
     2 PSD(20),SPL(20),VCL(20),VCT(20)

      IF(CODE1.GT.1.) GO TO 21

      WRITE(6,1)
    1 FORMAT(1H1,80H SOUND RADIATION FROM A SINGLE AIRFOIL AS RELATED TO
     1WAKE AERODYNAMIC PARAMETERS    //)

      WRITE(6,2) PATM,TATM,SPAN,XLAM,DEL,RAD
    2 FORMAT(17H AMBIENT PRESSURE 1F10.2,20H AMBIENT TEMPERATURE
     11F10.2//26H DIMENSIONS OF INTEGRATION  /13H AIRFOIL SPAN
     21E20.5,10X,17H STREAM DIRECTION 1E20.5/15H WAKE THICKNESS
     31E20.5//19H RADIUS TO OBSERVER   1E20.2///)

      WRITE(6,3) (Y(I),I=1,NPY)
    3 FORMAT(30H STEADY FLOW MACH NUMBER FIELD/17H STREAM DIRECTION
     130X,16H ACROSS THE WAKE  /20X,10F10.4/)

      WRITE(6,4)  (XM(I),I=1,NPY)
    4 FORMAT(5X,1F10.4,5X,10F10.5)

      DO 20 K=1,NPF
      WRITE(6,6)  FREQ(K),VCL(K),VCT(K)
    6 FORMAT(1H1,10H FREQUENCY   1E20.5,19H CORR VOLUME (LONG)  1E20.5,
     119H CORR VOLUME (TRAN)  1E20.5//)
      WRITE(6,7)  (Y(I),I=1,NPY)
    7 FORMAT(34H LONGITUNINAL TURBULENCE COMPONENT //17H STREAM DIRECTIO
     1N    30X,16H ACROSS THE WAKE/20X,10F10.2/)

      WRITE(6,4)  (U1U1(K,I),I=1,NPY)

      WRITE(6,8)  (Y(I),I=1,NPY)
    8 FORMAT(//32H TRANSVERSE TURBULENCE COMPONENT//17H STREAM DIRECTION
     130X,16H ACROSS THE WAKE /20X,10F10.2)

      WRITE(6,4)  (U2U2(K,I),I=1,NPY)

      WRITE(6,9)  (Y(I),I=1,NPY)
    9 FORMAT(//27H SHEAR TURBULENCE COMPONENT //17H STREAM DIRECTION
     130X,16H ACROSS THE WAKE /20X,10F10.2/)

      WRITE(6,4)  (U1U2(K,I),I=1,NPY)

   20 CONTINUE
   21 CONTINUE

      WRITE(6,11) THETA
   11 FORMAT(1H1,25H PREDICTED SOUND SPECTRUM  //16H RADIATION ANGLE
     11F10.2//10H FREQUENCY  10X,17H RADIATED SPL(DB)    )

      DO 12 I=1,NPF
   12 WRITE(6,13)  FREQ(I),SPL(I)
   13 FORMAT(1F10.2,10X,1F10.2)
```

```
      RETURN
      END
2116.     530.      .292      .042      .013      269.      3.0
1.0
   6    5
 .08      -.04      0.0       0.4       0.8
5000.     .6        .082      .324
5000.     .49       .066      .084
7500.    .6        .082      .324
7500.     .56       .071      .094
10000.    .6        .082      .273
10000.    .62       .077      .126
15000.    .43       .073      .189
15000.    .58       .077      .159
20000.    .42       .073      .157
20000.    .58       .073      .137
30000.    .28       .059      .105
30000.    .45       .071      .105
.92       .82       .54       .78       .64
16.       15.5      15.5      12.       15.       19.
20.       20.       20.       20.       20.       20.
23.       21.       19.       10.       10.       17.
.0055     .015      .04       .038      .007
0.0       0.0       0.0       0.0       0.0       0.0
.0055     .015      .038      .021      .005
```

## 3.0 PROGRAM ROTOR

The purpose of developing program ROTOR was to provide a method for the prediction of sound generation from turbomachinery rotors. Aerodynamic operation was limited to the subsonic flow regime. Both tone noise and broad band effects were included. The tone noise part of the model was simply the steady state forces on the rotor blades. The broad band mechanism was taken as the force fluctuations which were relatable to the wake turbulence of the individual airfoils of the rotor. The mathematical model used for the basis of the sound radiation was that presented by Ffowcs Williams and Hawkings*.

To use the program one must supply the aerodynamic parameters for the particular design. Program ROTOR then computes the spectrum of sound radiated by the machine due to the mechanisms noted above.

The program is divided into five subroutines. A brief discussion of the contents of these subroutines will be given to provide some familiarization with the program structure. The names of the subroutines are: DATAIN, WAKE, ORTH1, TONE, and ORTH2..

### 3.1 Subroutine DATAIN

As the name implies, DATAIN is used to read and condition the input data. The data input can be listed with the required FORMAT and physical units. The following are the only inputs required to run the program.

CARD 1, FORMAT(7F10.2)

| | | |
|---|---|---|
| RPM | rotor speed | (RPM) |
| R1 | distance to observer | (M) |

* "Theory Relating to the Noise of Rotating Machinery," J. Sound if Vib., 10, 1, 1969.

| | | |
|---|---|---|
| B | number of rotor blades | (-) |
| U1 | blade relative velocity | (M/SEC) |
| XLAM | blade stagger angle | (DEGREES) |
| RS | source radius | (M) |
| FMAX | maximum frequency in spectrum | (HZ) |

CARD 2, FORMAT(7F10.2)

| | | |
|---|---|---|
| FB | magnitude of force on each blade | (kg) |
| U1U1B | RMS turbulence level for the longitudinal component, normalized with U1 | (--) |
| U1U2B | RMS shear turbulence, normalized with U1 | (--) |
| U2U1B | RMS shear turbulence, normalized with U1 | (--) |
| U2U2B | RMS turbulence level for the transverse component, normalized with U1 | (--) |

CARD 3, FORMAT(7F10.2)

| | | |
|---|---|---|
| ZL | blade span or other span length used to define the length of the source in the span direction | (M) |
| YL | thickness of the airfoil wake | (M) |

CARD 4, FORMAT(1I5)

| | | |
|---|---|---|
| NF | number of frequency points to be used to input random data for wake related sound generation | (--) |

In the remaining inputs a FORMAT(7F10.2) is used but if more than NF = 7 is used then more than one card will be needed for each of the input variables. Here we will list the variable arrays as if only one card will be required for the input.

ALL FOLLOWING CARDS, FORMAT(7F10.2)

| | | |
|---|---|---|
| F(I) | array of frequency values | (HZ) |
| CORR(I) | an arbitrary spectral shaping factor, normally set equal to 1 | (--) |
| XLL(I) | correlation length in the stream direction, based upon the longitudinal component of turbulence | (CM) |
| YLL(I) | correlation length in the direction across the wake, based upon the longitudinal component of turbulence | (CM) |
| ZLL(I) | correlation length in the span direction, based upon the longitudinal component of turbulence | (CM) |
| XLT(I) | correlation length in the stream direction, based upon the transverse component of turbulence | (CM) |
| YLT(I) | correlation length in the direction across the wake, based upon the transverse component of turbulence | (CM) |
| ZLT(I) | correlation length in the span direction, based upon the transverse component of turbulence | (CM) |

The following variables, U1U1, U1U2, U2U1, and U2U2, are spectral levels given in dB relative to the RMS levels for the component. The result of this procedure will be: if these turbulence quantities are given in dB down from RMS on a 1/3 octave spectrum then the radiated sound will also be 1/3 octave. If 6% spectra are used then the radiated sound will be 6%, also.

| | | |
|---|---|---|
| U1U1(I) | longitudinal component of turbulence | (dB) |
| U1U2(I) | shear stress component | (dB) |
| U2U1(I) | shear stress component | (dB) |
| U2U2(I) | transverse component of turbulence | (dB) |

All input data are printed out in the physical units used in the computations.

## 3.2 Subroutine WAKE

This subroutine computes the wake related sound generation by the rotor. The model used to relate the force fluctuations on the rotor blading to aerodynamic variables is the single airfoil model. The expression which results and is here computed is

$$\left|p(\vec{x},\omega)\right|^2 = \frac{1}{16\pi^2 \; r^2} \sum_{n=-\infty}^{\infty} \sum_{k=1}^{B} \left(\frac{\omega}{a_o}\right)^2$$

$$d_r(\theta_k,\; f-n\Omega)\; d_r^*\;(\theta_k,\; f-n\Omega)\; J_n^2\left(\frac{-\omega R_s \; \sin\phi}{a_o}\right) .$$

For the present case, the force fluctuations are

$$d_r(\theta_k,\; f-n\Omega)\; d_r^*(\theta_k,\; f-n\Omega) = S\;\delta_\omega\;(\omega-2\pi n\Omega)^2\;\rho_o{}^2$$
$$\sum_{\ell=1}^{3} \sum_{m=1}^{3} \hat{r}_\ell \hat{r}_m \; 16\; L_1^2(u_\ell u_m,\omega)\; L_2(u_\ell,u_m,\omega)\; L_3(u_\ell,u_m,\omega)\; \overline{u_\ell u_m}(\omega).$$

where: $S$ = span, $\delta_w$ = wake thickness, $L_i$ = correlation lengths ($i$ = 1,3), $\overline{u_1 u_m}$ = spectral level of turbulent stress, $\omega$ = radian frequency, $\Omega$ = rotor speed, and $f$ = frequency in Hz. $J_n$ is a Bessel function of the first kind of integer order.

The output from this subroutine is the radiated sound pressure level expressed in dB relative to a reference pressure of 20 $\mu N/m^2$.

## 3.3 Subroutine ORTH1

This subroutine computes the components of the position vector of the sound observer in the coordinate system relative to each of the rotor blades.

The product of these components with the force components on the blades yields the force in the direction of the observer. To do this the position vector must first be expressed in terms of the rotor coordinate system. It must then be rotated by the blade stagger angle to be in the blade coordinate system.

The observer position vector components for a typical blade now may be written:

$$r_1 = (r \sin\phi - R_s \cos\theta)/[r(1 - \frac{R_s}{r} \sin\phi \cos\theta)]$$

$$r_2 = - R_s \sin\theta/[r(1 - \frac{R_s}{r} \sin\phi \cos\theta)]$$

$$r_3 = \cos\phi/(1 - \frac{R_s}{r} \sin\phi \cos\theta)$$

The change from the observer coordinates to those of the rotor blade can be expressed in the rotation

$$\begin{pmatrix} -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \cos\theta & \sin\theta & 0 \end{pmatrix}$$

and the further rotation by the stagger angle, $\lambda$,

$$\begin{pmatrix} \sin\lambda & \cos\lambda & 0 \\ -\cos\lambda & \sin\lambda & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The stagger angle, $\lambda$, is the angle between the airfoil chord and a radial plane passing through the axis of the rotor.

## 3.4 Subroutine TONE

Subroutine TONE as currently set up computes the sound generation related to the steady state blade forces. This is the "Gutin" mechanism. The program has been developed for a rotating steady state stress term also but as yet it is not known how to model this in terms of aerodynamic design parameters.

The expression which is computed for the radiated sound pressure is

$$\left|p(\vec{x},\omega)\right|^2 = \frac{1}{16\pi^2 r^2} \left(\frac{B}{R_s}\right)^2 \sum_{k=1}^{B} \sum_{m=-\infty}^{\infty} m^2 F_r^2 (\theta_k,\omega) J_{mB}^2\left(\frac{\omega R_s \sin\phi}{a_o}\right)$$

where: $B$ = number of blades in the rotor

$F_r$ = blade force in the direction of the observer

$R_s$ = source of radius

$\phi$ = observer angle to rotor axis

$r$ = distance to the observer

$\omega$ = radian frequency

## 3.5 Subroutine ORTH2

ORTH2 does exactly the same computation to the blade force as ORTH1 did for the random blade forces. The procedures are parallel.

### 3.6 Program Listing

```
      PROGRAM ROTOR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

C     AERODYNAMIC SOUND GENERATION FROM MULTIBLADED ROTORS
C     ...THE RADIATED SOUND SPECTRUM IS PREDICTED BASED
C        ON STEADY LOADS AND AIRFOIL WAKE STATISTICAL PROPERTIES...

      DIMENSION U1U1(50),U1U2(50),U2U1(50),U2U2(50),F(50),FT(200),
     1RHOT(200),RHOS(30),CORR(30),XLL(30),XLT(30),VCL(30),VCT(30)

C     CALL THE SUBROUTINE TO INPUT AND CONDITION THE DATA

      CALL DATAIN(U1U1,U1U2,U2U1,U2U2,F,RPS,R1,B,U1,XLAM,RS,FMAX,FB,NF,
     1ZL,YL,XLL,XLT,VCL,VCT,CORR)
      CALL WAKE(F,U1U1,U1U2,U2U1,U2U2,NF,RPS,FMAX,RS,RHOS,B,
     1R1,XLAM,U1,ZL,YL,XLL,XLT,VCL,VCT,CORR)

C     WAKE COMPUTES BROAD BAND SOUND GENERATION

      CALL TONE(R1,B,FB,RHOT,XLAM,RPS,RS,U1,FT)
C     TONE COMPUTES LEVELS OF PURE TONE GENERATED BY ROTATING
C     STEADY BLADE FORCE AND MEAN VALUES OF STRESS TENSOR
      STOP
      END
      SUBROUTINE DATAIN(U1U1,U1U2,U2U1,U2U2,F,RPS,R1,B,U1,XLAM,RS,
     1FMAX,FB,NF,ZL,YL,XLL,XLT,VCL,VCT,CORR)

C     READS AND CONDITIONS DATA

      DIMENSION U1U1(50),U1U2(50),U2U1(50),U2U2(50),F(50)
      DIMENSION VC(30),CORR(30),XLL(30),YLL(30),ZLL(30),
     2XLT(30),YLT(30),ZLT(30),VCT(30),VCL(30)

C     RPM = ROTOR SPEED (RPM)
C     R1 = OBSERVER RADIUS (M)
C     B = NUMBER OF ROTOR BLADES
C     U1 = BLADE RELATIVE VELOCITY (M/SEC)
C     XLAM = STAGGER ANGLE - MEASURED FROM FAN AXIS (DEG)
C     RS = SOURCE RADIUS (M)
C     FMAX = MAXIMUM FREQUENCY OF INTEREST (HZ)
C     FB = BLADE FORCE (KG)
C     U1U1B = RMS LONGITUDINAL TURBULENCE - NORMALIZED WITH U1
C     U1U2B = RMS SHEAR - NORMALIZED WITH U1
C     U2U1B = RMS SHEAR - NORMALIZED WITH U1
C     U2U2B = RMS TRANSVERSE TURBULENCE - NORMALIZED WITH U1
C     ZL = SPAN DIMENSION (CM)
C     ZL = WAKE THICKNESS (CM)
C     CHD = BLADE CHORD (CM)
C     NF = NUMBER OF FREQUENCY POINTS
C     F(I) = FREQUENCY ARRAY (HZ)
C     CORR(I) = A SPECTRAL WEIGHTING FUNCTION (NORMALLY UNITY)
C     XLL(I) = STREAM CORRELATION LENGTH FOR U1U1B
C     YLL(I) = NORMAL CORRELATION LENGTH FOR U1U1B (CM)
C     ZLL(I) = SPAN CORRELATION LENGTH FOR U1U1B  (CM)
C     ZLT(I) = STREAM CORRELATION LENGTH FOR U2U2B  (CM)
C     YLT(I) = NORMAL CORRELATION LENGTH FOR U2U2   (CM)
C     ZLT(I) = SPAN CORRELATION LENGTH FOR U2U2  (CM)
```

```
C       U1U1(I) =  DB DOWN FROM U1U1B  (RMS)
C       U1U2(I) = DB DOWN FROM U1U2B  (RMS)
C       U2U1(I) = DB DOWN FROM U2U1B  (RMS)
C       U2U2(I) = DB DOWN FROM U2U2B  (RMS)

        DATA AO,CO,PI/342.,1.0,3.14159/

        READ(5,1) RPM,R1,B,U1,XLAM,RS,FMAX
        READ(5,1) FB,U1U1B,U1U2B,U2U1B,U2U2B
        READ(5,1)   ZL,YL,XL,CHD
        CMFT = .0328
        R1=R1*CMFT*100.
        U1=U1*CMFT*100.
        RS=RS*CMFT*100.
        FB=FB*2.21
        ZL=ZL*CMFT
        YL=YL*CMFT
        CHD=CHD*CMFT
      1 FORMAT(7F10.2)
        TPI=2.*PI
        XLAM=XLAM*PI/180.
        RPS=TPI*RPM/60.
        READ(5,2) NF
      2 FORMAT(1I5)
        READ(5,1)  (F(I),I=1,NF)
        READ(5,1)  (CORR(I),I=1,NF)
        READ(5,1)  (XLL(I),I=1,NF)
        READ(5,1)  (YLL(I),I=1,NF)
        READ(5,1)  (ZLL(I),I=1,NF)
        READ(5,1)  (XLT(I),I=1,NF)
        READ(5,1)  (YLT(I),I=1,NF)
        READ(5,1)  (ZLT(I),I=1,NF)
        READ(5,1)  (U1U1(I),I=1,NF)
        READ(5,1)  (U1U2(I),I=1,NF)
        READ(5,1)  (U2U1(I),I=1,NF)
        READ(5,1)  (U2U2(I),I=1,NF)

C       INTRODUCE SCALING FACTORS

        SCL=(CHD/3.0)**0.8*(U1/150.)**(-0.2)
        SCF=(U1/150.)**1.2*(CHD/3.0)**(-0.8)
        FMAX=FMAX*SCF
        XL=XL*SCL
        YL=YL*SCL
        ZL=ZL*SCL
        DO 30 I=1,NF
        F(I)=F(I)*SCF
        XLL(I)=XLL(I)*SCL
        YLL(I)=YLL(I)*SCL
        ZLL(I)=ZLL(I)*SCL
        XLT(I)=XLT(I)*SCL
        YLT(I)=YLT(I)*SCL
     30 ZLT(I)=ZLT(I)*SCL
        DO 20 I=1,NF
        VCL(I)=8.*XLL(I)*YLL(I)*ZLL(I)/28350.
     20 VCT(I)=8.*XLT(I)*YLT(I)*ZLT(I)/28350.
```

```
C     WRITE TIE INPUT DATA

      WRITE(6,3)R1,RPM,FB,XLAM,B,RO,U1,FMAX,U1U1B,U1U2B,U2U1B,U2U2B
    3 FORMAT(1H1,54H AERODYNAMIC SOUND GENERATION FROM MULTIBLADED ROTOR
     15//23H OBSERVER DISTANCE (M)    1F7.2,10X,12H ROTOR SPEED
     21F10.2/19H STEADY BLADE FORCE  1F11.2,10X,20H BLADE STAGGER ANGLE
     31F10.2/17H NUMBER OF BLADES  1F3.0,20X,14H SOURCE RADIUS
     41F6.2/24H BLADE RELATIVE VELOCITY  1F6.1,10X,15H MAX. FREQUENCY
     51F15.2/10H U1U1(RMS)  1E10.3,5X,10H U1U2(RMS) 1E10.3,5X,
     610H U2U1(RMS) 1E10.3,5X,10H U2U2(RMS)  1E10.3////)
      WRITE(6,9)  ZL,YL
    9 FORMAT(21H WAKE VOLUME  L(Z) =  1E15.5,5X,
     18H L(Y) =   1E15.5//)
      WRITE(6,11)
   11 FORMAT(100H    FREQUENCY                L(Z)                   L(Y)
     1               L(X)                SCALR FACTOR          )
      WRITE(6,12)  (F(I),ZLL(I),YLL(I),XLL(I),CORR(I),I=1,NF)
      WRITE(6,12)  (F(I),ZLT(I),YLT(I),XLT(I),CORR(I),I=1,NF)
   12 FORMAT(5E20.5)
      WRITE(6,4)
    4 FORMAT(14H SPECTRUM U1U1  //)
      WRITE(6,5) (F(I),U1U1(I),I=1,NF)
    5 FORMAT(4(1F10.3,1E20.4))
      WRITE(6,6)
    6 FORMAT(///14H SPECTRUM U1U2  //)
      WRITE(6,5)  (F(I),U1U2(I),I=1,NF)
      WRITE(6,7)
    7 FORMAT(///14H SPECTRUM U2U1  //)
      WRITE(6,5)  (F(I),U2U1(I),I=1,NF)
      WRITE(6,8)
    8 FORMAT(///14H SPECTRUM U2U2 //)
      WRITE(6,5) (F(I),U2U2(I),I=1,NF)

C     CONDITION TURBULENCE DATA TO LINEAR SCALE

      DO 10 I=1,NF
      XLL(I)=XLL(I)/(2.54*12.)
      XLT(I)=XLT(I)/(2.54*12.)
      U1U1(I)=U1U1B/(10.**(U1U1(I)/20.))
      U1U2(I)=U1U2B/(10.**(U1U2(I)/20.))
      U2U1(I)=U2U1B/(10.**(U2U1(I)/20.))
   10 U2U2(I)=U2U2B/(10.**(U2U2(I)/20.))
      RETURN
      END
      SUBROUTINE WAKE(F,U1U1,U1U2,U2U1,U2U2,NF,RPS,FMAX,RS,RHOS,B,
     1R1,XLAM,U1,ZL,YL,XLL,XLT,VCL,VCT,CORR)

C     COMPUTES SOUND RADIATION FROM WAKE TURBULENCE PROPERTIES

      DIMENSION F(30),U1U1(30),U1U2(30),U2U1(30),U2U2(30),BES(1000),
     1BEN(1000),PHI(20),SX(30,3),S1(30),
     2RHOS(30),V1(3),VC(30),CORR(30),
     3XLL(30),XLT(30),VCL(30),VCT(30)

      DATA CG,AO,PI,RHO/1.0,342.,3.14159,1.175/
```

```
      PREF=20.E-06/AO**2
      PREF=PREF*PREF
      C1=1./(4.*PI*AO**2*R1)**2
      TPI=2.*PI
      C2=C1*(RHO/CG)**2
      REV=RPS/TPI
      ATOT=ZL*YL

C     COMPUTE THE RADIATION RATTERN

      DO 1000 KK=1,3
      PHI(KK)=45.*FLOAT(KK-1)
      PHIP=PHI(KK)*PI/180.

C     COMPUTE THE SPECTRUM

      DO 11 I=1,NF
      DVL=VCL(I)*ATOT*CORR(I)
      DVT=VCT(I)*ATOT*CORR(I)
      DVL=SQRT(DVL)
      DVT=SQRT(DVT)
      DV=0.0
      SX(I,1)=-U1U1(I)*DVL*SQRT(XLL(I))
      SX(I,2)=U2U2(I)*DVT*SQRT(XLT(I))
      DO 40 J=1,3
   40 WRITE(6,41)  SX(I,J)
   41 FORMAT(10H SX(I,J) =     1E20.5)
      DO 32 J=1,3
   32 V1(J)=SX(I,J)
      V3=U1U2(I)
      V4=U2U1(I)
      FRA=F(I)*TPI

      CALL ORTH1(V1,FRA,V3,V4,PHIP,XLAM,B,R1,RS,DD)

   11 S1(I)=DD
      DO 2000 K=1,NF
      C3=(TPI*F(K)/AO)**2*U1**2

C     THE ARGUEMENT FOR THE BESSEL FUNCTION

      ARG=-TPI*F(K)*RS*SIN(PHIP)/AO

C     NMAX IS THE MAXIMUM NECESSARY VALUES OF N FOR THE BESSEL FUNCTION

      NMAX=IFIX(FMAX*TPI/RPS)*2+1
      NMAX=250
      IF(NMAX.GT.400)  NMAX=400
      IF(ARG.NE.0.0)  GO TO 60
      BES(1)=1.0
      DO 61 LL=2,NMAX
   61 BES(LL)=0.0
      GO TO 62
   60 CONTINUE
      DO 63 LL=1,NMAX
```

```
   63 BES(LL)=0.0
      CALL NBESJ(ARG,NMAX,BES)
   62 CONTINUE
C     SQUARE THE BESSEL FUNCTIONS
      DO 44 J=1,NMAX
      BEN(J)=(-1.)**(J-1)*BES(J)
      BES(J)=BES(J)*BES(J)
   44 BEN(J)=BEN(J)*BEN(J)
C     PROVIDE THE SUMMANDS
      SUM1=0.0
      DO 14 N=1,NMAX
      FP=F(K)+FLOAT(N-1)*REV
      IF(FP.GT.FMAX)  GO TO 13
      DUM1=TBLU1(FP,F,S1,1,NF)
      DUM1=DUM1*(TPI*FP)**2
      GO TO 14
   13 DUM1=0.0
   14 SUM1=SUM1+DUM1*BES(N)
      DO 12 N=2,NMAX
      FN=F(K)-FLOAT(N-1)*REV
      IF(FN.LT.-FMAX)  GO TO 15
      FN=ABS(FN)
      DUM1=TBLU1(FN,F,S1,1,NF)
      DUM1=DUM1*(TPI*FN)**2
      GO TO 12
   15 DUM1=0.0
   12 SUM1=SUM1+DUM1*BEN(N)
      WRITE(6,31)  SUM1,DUM1
   31 FORMAT(5H SUM1  1E20.5,5H DUM1   1E20.5)
C     THIS COMPLETES THE DIPOLE TERM
C     WRITE THE MODULUS OF THE RADIATED SOUND DENSITY
      VAR1=C2*C3*SUM1/PREF
      WRITE(6,50)  F(K),VAR1
   50 FORMAT(1F10.3,1E20.5)
 2000 RHOS(K)=C2*C3*SUM1
      WRITE(6,1)  PHI(KK)
    1 FORMAT(1H1,37H SPECTRAL DISTRIBUTION OF ROTOR SOUND  ///
     130H WAKE TURBULENCE RELATED SOUND  //17H RADIATION ANGLE  1F7.2//
     210X,10H FREQUENCY  10X,8H SPL(DB)     /)
      DO 10 I=1,NF
      VAR1=RHOS(I)/PREF
      IF(VAR1.LT.1.0)  GO TO 10
      RHOS(I)=10.*ALOG10(VAR1)
   10 CONTINUE
      WRITE(6,2)  (F(I),RHOS(I),I=1,NF)
    2 FORMAT(12X,1F10.2,10X,1E20.4)
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ORTH1(SX,RPS,U1U2,U2U1,PHI,XLAM,B,R1,RS,DRDR)

C     COMPUTES THE SUM OF THE COMPONENTS IN THE DIRECTION OF THE
C     OBERVER  FROM EACH BLADE OF THE ROTOR

      DIMENSION ZY(3,3),THE(30),YX(3,3),SX(3),SYY(3,3),R(3),V(3)

C     INTRODUCE THE COMPONENTS OF THE FIRST ROTATION

      PI=3.14159
      ZY(1,1)=SIN(XLAM)
      ZY(1,2)=COS(XLAM)
      ZY(1,3)=0.0
      ZY(2,1)=-COS(XLAM)
      ZY(2,2)=SIN(XLAM)
      ZY(2,3)=0.0
      ZY(3,1)=0.0
      ZY(3,2)=0.0
      ZY(3,3)=1.0

C     THE NEXT ROTATION DEPENDS ON THE INITIAL BLADE POSITION AS DOES
C     THE OBSERVER RADIUS VECTOR

      NB=IFIX(B)
      DRDR=0.0
      DO 10 K=1,NB
      THE(K)=360./B*FLOAT(K)
      THET=THE(K)*PI/180.
      DEN=R1*(1.-RS/R1*SIN(PHI)*COS(THET))
      R(1)=(R1*SIN(PHI)-RS*COS(THET))/DEN
      R(2)=(-RS*SIN(THET))/DEN
      R(3)=(R1*COS(PHI))/DEN
      YX(1,1)=-SIN(THET)
      YX(1,2)=COS(THET)
      YX(1,3)=0.0
      YX(2,1)=0.0
      YX(2,2)=0.0
      YX(2,3)=1.0
      YX(3,1)=COS(THET)
      YX(3,2)=SIN(THET)
      YX(3,3)=0.0

C     NOW SUM ON THE COMPONENTS TO GET FINAL VALUES IN OBSERVER FRAME

      DD=0.0
      DO 11 I=1,3
      V(I)=0.0
      DO 11 J=1,3
      DO 11 L=1,3
   11 V(I)=R(L)*ZY(I,J)*YX(J,L)+V(I)
      WRITE(6,30)  (V(I),I=1,3)
   30 FORMAT(5H V(B)  3E15.3)

C     DETERMINE THE COMPONENTS IN THE DIRECTION OF THE OBSERVER

C     DIPOLE TERMS
```

```
      DO 12 I=1,2
      DO 12 J=1,2
      IF(I.EQ.1.AND.J.EQ.2)   GO TO 13
      IF(J.EQ.1.AND.I.EQ.2)   GO TO 131
      VAR1=SX(I)*SX(J)
      GO TO 12
   13 VAR1=-U1U2**2
      GO TO 12
  131 VAR1=-U2U1**2
   12 DD=DD+V(I)*VAR1*V(J)
   10 DRDR=DRDR+DD
      WRITE(6,20)  DRDR
   20 FORMAT(1E20.5)
C
C     THIS COMPLETES COMPUTATION OF THE COMPONENTS IN THE
C     DIRECTION OF THE OBSERVER
C
      RETURN
      END
      SUBROUTINE TONE(R1,B,FB,RHO,XLAM,RPS,RS,U1,F)
C
C     COMPUTES PURE TONES GENERATED BY STEADY BLADE FORCES AND
C     THE MEAN VALUES OF THE STRESS TENSOR
C
      DIMENSION BES(200),SX(3),SYY(3,3),TX(200),TYY(200),F(200),
     1RHO(200),PHI(20),TT(30),DD(30)
      DATA AO,PI,CG/342.,3.14159,1.0/
      PREF=20.E-06/AO**2
      PREF=PREF*PREF
      RATM=1.175
      RATM=0.0
      NB=IFIX(B)
C
C     PROVIDE THE TENSOR AND FORCE COMPONENTS
C
      C1=1./(2.*PI*AO**2*R1)**2
      REV=RPS/(2.*PI)
      FAX=0.05*FB
      FT=0.95*FB
      DO 1 I=1,3
    1 SX(I)=0.0
      SX(1)=-FAX
      SX(2)=FT
      DO 2 I=1,3
      DO 2 J=1,3
    2 SYY(I,J)=0.0
      SYY(1,1)=1.0
      DO 1000 KK=1,1
      PHI(KK)=30.+10.*FLOAT(KK)
      PHI(KK)=60.+10.*FLOAT(KK)
      PHI(KK)=80.
      PHIP=PHI(KK)*PI/180.
      C2=(B/RS)**2
      C3=(B*RATM*U1*U1/(AO*RS*SIN(PHIP)))**2
```

```
C       COMPUTE THE SPECTRUM

        CALL ORTH2(SX,SYY,PHIP,XLAM,B,R1,RS,DD,TI)

C       COMPUTE 5 HARMONICS OF BLADE PASSAGE FREQUENCY

        NH=10
        DO 10 N=1,NH
        F(N)=FLOAT(N)*REV*B
        ARG=2.*PI*F(N)*RS*SIN(PHIP)/A0
        NN=IFIX(B)*N+1
        CALL NBESJ(ARG,NN,BES)
        WRITE(6,30)  ARG,BES(NN)
     30 FORMAT(16H BESSEL FUNCTION   2E20.5)
        TYY(N)=0.0
        TX(N)=0.0
        DO 40 K=1,NB
        DO 40 L=1,NB
        TX(N)=TX(N)+BES(NN)*BES(NN)*DD(K)*DD(L)
     40 TYY(N)=TYY(N)+BES(NN)*BES(NN)*TI(K)*TI(L)*2.*PI*F(N)

C       THIS COMPLETES THE SPECTRAL LEVEL COMPUTATION - NOW CALL ROUTINE
C       TO PERFORM COORDINATE TRANSFORMATIONS

        VAR1=C1*C2*TX(N)/PREF
        VAR2=C1*C3*TYY(N)/PREF
        WRITE(6,20)  F(N),VAR1,VAR2
     20 FORMAT(3(1F10.2,2F15.3))
        FNN=FLOAT(N)*FLOAT(N)
     10 RHO(N)=C1*C2*FNN*TX(N)
        WRITE(6,5)
      5 FORMAT(1H1,37H SPECTRAL DISTRIBUTION OF ROTOR SOUND     ///)
        DO 11 I=1,NH
        VAR2=RHO(I)/PREF
        IF(VAR2.LT.1.0)  GO TO 11
        RHO(I)=10.*ALOG10(VAR2)
     11 CONTINUE
        WRITE(6,3)  PHI(KK)
      3 FORMAT(/////36H SPECTRAL DISTRIBUTION OF TONE SOUND    ///
       117H RADIATION ANGLE  1F5.2///
       29H FREQ(HZ)    11X,8H SPL(DB)        //)
        WRITE(6,4)  (F(I),RHO(I),I=1,NH)
      4 FORMAT(4(1F10.2,1F20.3))
   1000 CONTINUE
        RETURN
        END
        SUBROUTINE ORTH2(SX,SYY,PHI,XLAM,B,R1,RS,DRDR,TRTR)

C       COMPUTES THE SUM OF THE COMPONENTS IN THE DIRECTION OF THE
C       OBSERVER  FROM EACH BLADE OF THE ROTOR

        DIMENSION ZY(3,3),THE(30),YX(3,3),SX(3),SYY(3,3),R(3),V(3)
        DIMENSION DRDR(30),TRTR(30)

C       INTRODUCE THE COMPONENTS OF THE FIRST ROTATION
```

```
      PI=3.14159
      ZY(1,1)=SIN(XLAM)
      ZY(1,2)=COS(XLAM)
      ZY(1,3)=0.0
      ZY(2,1)=-COS(XLAM)
      ZY(2,2)=SIN(XLAM)
      ZY(2,3)=0.0
      ZY(3,1)=0.0
      ZY(3,2)=0.0
      ZY(3,3)=1.0

C     THE NEXT ROTATION DEPENDS ON THE INITIAL BLADE POSISTION AS DOES
C     THE OBSERVER RADIUS VECTOR

      NB=IFIX(B)
      DO 10 K=1,NB
      THE(K)=360./B*FLOAT(K)
      THET=THE(K)*PI/180.
      DEN=R1*(1.-RS/R1*SIN(PHI)*COS(THET))
      R(1)=(R1*SIN(PHI)-RS*COS(THET))/DEN
      R(2)=(-RS*SIN(THET))/DEN
      R(3)=(R1*COS(PHI))/DEN
      YX(1,1)=-SIN(THET)
      YX(1,2)=COS(THET)
      YX(1,3)=0.0
      YX(2,1)=0.0
      YX(2,2)=0.0
      YX(2,3)=1.0
      YX(3,1)=COS(THET)
      YX(3,2)=SIN(THET)
      YX(3,3)=0.0

C     NOS SUM ON THE COMPONENTS TO GET FINAL VALUES IN OBSERVER FRAME

      DD=0.0
      TT=0.0
      DO 11 I=1,3
      V(I)=0.0
      DO 11 J=1,3
      DO 11 L=1,3
   11 V(I)=R(L)*YX(J,L)*ZY(I,J)+V(I)
      WRITE(6,20)  (V(I),I=1,3)
   20 FORMAT(//31H OBSERVER VECTOR IN BLADE BASIS  /3E20.5//)
C     DETERMINE THE COMPONENTS IN THE DIRECTION OF THE OBSERVER
C     DIPOLE TERMS
      DO 12 I=1,2
   12 DD=DD+V(I)*SX(I)
      WRITE(6,30)  DD
   30 FORMAT(16H FORCE COMPONENT    1E20.5)
C     QUADRAPOLES
      DO 18 I=1,2
      DO 18 J=1,2
   18 TT=TT+V(I)*SYY(I,J)*V(J)
      DRDR(K)=DD
   10 TRTR(K)=TT
C     THIS COMPLETES COMPUTATION OF THE COMPONENTS IN THE
C     DIRECTION OF THE OBSERVER
      RETURN
      END
```

## 4.0 PROGRAM TONE

This model was set up in a computer program so that direct numerical computations of the time series described by the equation for the radiated sound could be made. This direct approach of calculating the actual time series is the difference between the present approach and those of other rotor models. Typically the analytical results for the rotor are Fourier transformed and reduced to a closed form (such as program ROTOR described herein). This approach normally requires that simplifications be made so that the closed form can be obtained. The advantage of the present approach is no simplifications in the analysis need be made.

The time series is computed for a sufficient number of values so that a good statistical sample will result. The time series is then Fourier transformed using a Fast Fourier Transform algorithm to yield the spectrum of the radiated sound. In other words, the model is used to generate a time series which is handled exactly as one would handle experimental data. Very complex source histories can be studied with no more effort than the very simple steady force case. For example, rotor wobble can be simulated in order to study how it affects the generation of sound; the individual blades can be allowed to vibrate as might occur in flutter cases; or each blade can be assigned a different value of force to determine the effect of such blade to blade differences as might be caused by manufacturing imperfections.

Of most importance, is that complex random effects can be included in this model, also. Such mechanisms as force fluctuations related to wake turbulence, inflow turbulence, or even rotating stalls can be studied. The way this must be done is to express the force fluctuation in a time series that has the required statistical nature. If the spectrum of the force fluctuation relative to the blade is known, then it is possible to generate a random time series which will give the same statistics as the known result. The methods for generating an auto-regres .on analysis can be used to simulate the spectrum of the force on the blades. Once the time series are known, the results from the different blades may be combined in different ways to simulate different mechanisms of sound generation. For example, the effect of large scale turbulence can be simulated by allowing several of the blades to experience the same force fluctuation, or allow all of them to experience it. The case where there is no blade to blade correlation would model the wake related sound generation.

The idea of the model is to provide a tool where the various mechanisms which might produce appreciable sound generation can be evaluated. In addition, it is sought to set up the model in such a way that its usage will be reasonably simple and straightforward. As an example, if one wants to study the effect of different forces on each of the blades, he merely reads in the array of steady loads he wishes to rotor to have, and that is all. Rotor wobble is accounted for by replacing the velocity of the source, the rotor velocity, with a time dependent function describing the wobble characteristics.

## 4.1 Tone Noise Generation Model

The generation of pure tone sound due to the steady state forces on the rotor blading is studied in the model. In addition, an auto-regression method is used to write a time series which approximates a random force distribution resulting in broad band noise generation.

The model is based upon M.V. Lowson's development for rotating point forces*. The development is done for a force per unit volume. In this representation only the point force case will be considered. Extension to the distributed case can be treated as an array of point forces where the array is distributed along the span as well as circumferentially. This has been done by the author and the mechanics are straightforward but the expense in computer time is correspondingly increased.

For a point force, the radiated sound in the far field can be written:

$$p(\vec{x},t) = \left[\frac{x_i - y_i}{4\pi a_o r^2 (1-M_r)^2}\left\{\frac{\partial F_i}{\partial t} + \frac{F_i}{1-M_r}\frac{\partial M_r}{\partial t}\right\}\right]_{t_R}$$

where the evaluation of the source terms of the retarded time

$$t_R = t - \frac{|\vec{x}-\vec{y}|}{a_o}$$

$t_R$ = time of sound generation

$t$ = time of observation, and $|\vec{x} - \vec{y}|/a_o$ = propagation time.

The position vector $\vec{r} = \vec{x}-\vec{y}$ can be written in the form

$$\vec{r} = \sum_{i=1}^{3} (x_i - y_i)\,\vec{e}_i$$ in rectangular cartesian coordinates.

The Mach number $M_r$ is the component of the source Mach number in the direction of the observer. It may be written

$$M_r = \vec{M}_s \cdot \vec{r}/|\vec{r}|$$

or in components as

---

* "The Sound Field for Singularities in Motion," Proc. Roy. Soc., A, 286, 559, 1965.

$$M_r = \sum_{i=1}^{3} M_i(x_i - y_i)/|\vec{r}|.$$

The components of the blade force F can be chosen to represent the thrust and torque loads on the individual blades. Using the blade stagger angle, $\xi$, write

$$F_1 = -F \cos \xi$$

$$F_2 = -F \sin \xi \sin \theta$$

$$F_3 = F \sin \xi \cos \theta.$$

Take the time derivatives of these force components:

$$\partial F_1/\partial t = 0$$

$$\partial F_2/\partial t = -F\Omega \sin \xi \cos \theta$$

$$\partial F_3/\partial t = -F\Omega \sin \xi \sin \theta.$$

The components of the position of the observer are written in polar form as

$$x_1 = r \cos \phi$$

$$x_2 = r \sin \phi$$

$$x_3 = 0 .$$

The location of the source is also written in polar form:

$$y_1 = 0$$

$$y_2 = R_s \cos \theta$$

$$y_3 = R_s \sin \theta.$$

Finally, the components of the source Mach number are

$$M_1 = 0$$

$$M_2 = -M \sin \theta$$

$$M_3 = M \cos \theta .$$

The value of the source Mach number is

$$M_s = R_s \Omega / a_o .$$

The Mach number component $M_r$ may now be computed:

$$M_r = \sum_{i=1}^{3} M_i (x_i - y_i)/r .$$

Substitution yields

$$M_r = (r_1 \cos \phi)(0)/r + (r_1 \sin \phi - R_s \cos \theta)(-M_s \sin \theta)/r$$

$$= -(r_1/r) M_s \sin \phi \sin \theta + (R_s/r) M_s \cos \theta \sin \theta ,$$

so that to the first order

$$M_r = -M_s (r_1/r) \sin \phi \sin \theta .$$

The time derivative of the Mach number becomes

$$\partial M_r / \partial t = -M_s (r_1/r) \Omega \sin \phi \cos \theta .$$

All the terms required for the computation of the pressure radiated are now complete. The radiated sound is

$$p(\vec{x},t) = [\frac{F}{4\pi a_o r^2(1-M_r)^2}\{\frac{r_1 \sin\phi - R_s \cos\theta}{r}(-\Omega \sin\xi \cos\theta)$$

$$\frac{- R_s \sin\theta}{r}(-\Omega \sin\xi \sin\theta) + \frac{1}{1-M_r}\frac{\partial M_r}{\partial t}[\frac{r_1 \cos\phi}{r}(-\cos\xi)$$

$$+ \frac{r_1 \sin\phi - R_s \cos\theta}{r}(-\sin\xi \sin\theta) + \frac{-R_s \sin\theta}{r}\sin\xi \cos\theta]\}].$$

And, expanding:

$$p(\vec{x},t) = [\frac{F}{4\pi a_o r(1-M_r)^2}\quad\frac{-r_1 \Omega \sin\phi \sin\xi \cos\theta}{r}$$

$$+ \frac{R_s \Omega \sin\xi}{r}(\cos^2\theta + \sin^2\theta) + \frac{1}{1-M_r}\frac{\partial M_r}{\partial t}[\frac{-r_1 \cos\phi \cos\xi}{r}$$

$$\frac{-r_1 \sin\xi \sin\phi \sin\theta}{r}]\}].$$

The solution reduces to the form

$$p(\vec{x},t) = \frac{F}{4\pi a_o r^2(1-M_r)^2}[\Omega \sin\xi (R_s - r_1 \sin\phi \cos\theta)$$

$$- \frac{1}{1-M_r}\frac{\partial M_r}{\partial t}(r_1 \cos\phi \cos\xi + r_1 \sin\xi \sin\phi \sin\theta)].$$

This will give the radiated sound from each of the blades. The above expression is to be evaluated at the retarded time. This will require some further manipulations.

$$t_R = t \quad r/a_o \qquad\qquad \text{where } r = |\vec{x}-\vec{y}|$$

square the distance r

$$r^2 = (x_i - y_i)^2 = (r_1 \cos\phi)^2 + (r_1 \sin\phi - R_s \cos\theta)^2 + (R_s \sin\theta)^2$$

$$= r_1^2 - 2\, r_1\, R_s \sin\phi \cos\theta + R_s^2 \,.$$

Now, $r_1 >> R_s$, so it will be sufficient for the far field solution to write

$$r^2 = r_1^2 - 2\, r_1\, R_s \sin\phi \cos\theta$$

or, equivalently,

$$r^2 = (r_1 - R_s \sin\phi \cos\theta)^2 - R_s^2 \sin^2\phi \cos^2\theta \,.$$

For the far field solution we can therefore use

$$r \approx r_1 - R_s \sin\phi \cos\theta \,.$$

The retarded time can be written as

$$t_R = t - r_1/a_o + R_s/a_o \sin\phi \cos[\theta] \,,$$

where the $[\theta]$ is the retarded value of the blade location.

This source position may be written as

$$[\theta] = \Omega t_R + \theta_{t_R=0} = \Omega(t - r_i/a_o) + \frac{\Omega R_s}{a_o} \sin\phi \cos[\theta] + \theta_{t_R=0} \,.$$

Now in the solution for the radiated sound substitute $t_R$ for t and $[\theta]$ for $\theta$, the radiated sound pressure becomes

$$p_K(\vec{x},t) = \frac{F}{4\pi a_o r^2 (1-[M_r])^2} \{\Omega \sin\xi (R_s - r_1 \sin\phi \cos[\theta]_k)$$

$$- \frac{r_i}{(1-[M_r]_k)} [\frac{\partial M_r}{\partial t}]_k (\cos\xi \cos\phi + \sin\xi \sin\phi \sin[\theta]_k)\}$$

where: $r = r_1 - R_s \sin\phi \cos[\theta]_k$

$$[M_r]_k = -M_s \sin\phi \sin[\theta]_k$$

$$[\partial M_r/\partial t]_k = -M_s\, \Omega \sin\phi \cos[\theta]_k \;,$$

with: $[\theta]_k = \Omega(t - r_1/a_o) = (\Omega\, R_s/a_o) \sin\phi \cos[\theta]_k + \theta_{k,t_R=0}$

$$\theta_{k,t=0} = \text{location of kth source at } t_R = 0 \;.$$

The total sound from all the rotor blades becomes

$$p(\vec{x},t) = \sum_{K=1}^{B} p_k(\vec{x},t) \;.$$

## 4.2 How to Use Program TONE

As was seen in the previous section, the model used to predict the sound radiation generates a discrete time history using the solution for a rotating array of forces, then treats that history in the same way one would treat experimental data.

We previously mentioned the flexibility of this approach. This flexibility is achieved in the program by changing the program listing. A multiple option approach could have been taken but this is wasteful of computer time and assumes one knows all the options which will be needed. The reason for this approach was to provide a flexible tool; however, it does depend upon the operator's ability to change the Fortran statements used to generate the time series. This is easily done since the equations are short and the program is straightforward. The part of the program actually used to generate the time series is about 20 cards long.

The form of the program set up here is for the case where the force on each of the blades is the following function of time

$$F(t) = F_B \ (1 + C_d x(t))$$

where $F_B$ is the steady state blade force, $C_d$ is a dynamic loading coefficient, and x(t) is a time series simulating the unsteady force fluctuations on the blade. Depending upon the particular mechanism, x(t) can be chosen to simulate either wake related force fluctuations or the fluctuations caused by inflow turbulence interacting with the airfoil. Processes of this type can be represented using autoregressive processes. This will be discussed in section 4.5 below.

## 4.3 Program Inputs

Input to the program is done using cards. Four cards are required to run the program in its present form.

CARD 1, FORMAT(1I5)

NOPT = 0, no plots
= 1, plot the time history
= 2, plot the sound spectrum
= 3, plot both the history and the spectrum

NOPT controls the type of output desired. Two CALCOMP subroutines are contained in the program so that the time history for the process can be examined and/or the spectrum of the radiated sound can be plotted. If neither plot is desired then listings of the time history and the spectrum will be the only output.

CARD 2, FORMAT(7F10.2)

| | | |
|---|---|---|
| BE | bandwidth desired in the spectrum | (Hz) |
| FMAX | maximum frequency in the spectrum | (Hz) |

See Section 4.4 for the implications of these selections.

CARD 3, FORMAT(7F10.2)

| | | |
|---|---|---|
| ZR | number of rotor blades | (--) |
| XI | blade stagger angle, ueasred from blade chord to radial plane passing through rotor axis | (DEG) |
| PHI | radiation angle, measured from inlet rotor axis | (DEG) |
| RPM | rotor speed | (RPM) |
| RS | radial distance to source | (M) |
| R1 | radial distance to observer | (M) |
| F | magnitude of force on each blade | (KG) |

CARD 4, FORMAT(7F10.2)

| | | |
|---|---|---|
| CDYN | dynamic force coefficient | (--) |
| A1 | autoregression coefficient | (--) |
| A2 | autoregression coefficient | (--) |

See Section 4.5 for a discussion of the selection of the autoregression coefficients.

## 4.4 Time Sampling

The setting of FMAX and BE determines the sampling rate and the total number of required samples. Letting

$$f_{max} = FMAX$$

$$B_e = BE$$

we can illustrate how this is done. First, estimate the number of samples required as

$$N = 2\ f_{max}/B_e.$$

Since we will use a Fast Fourier Transform to find the spectrum, the number of samples must be of the form $N' = 2^m$ where m is an integer. We choose m such that

$$2^{m-1} < N \leq 2^m.$$

Then compute $N' = 2^m$ samples of the time series. The new bandwidth becomes

$$B_e' = 2\ f_{max}/N'\ .$$

The sampling rate for the time series is

$$H = 1/(2f_{max})\ .$$

4.5 Use of Autoregression Processes

A second order autoregression process is used in the current form of the program to represent the force fluctuations. This process may be written

$$x(t) = a_1\ x(t-1) + a_2\ x(t-2) + Z(t)$$

where $a_1$ and $a_2$ are the autoregression coefficients (A1 and A2 in the program. Z(t) is a random number selected using a randum number generator.

A time series is generated separately for each blade. If there is to be no correlation between different blades (as for wake related forces) then

Z will be taken from a different random series for each blade. If correlation is desired between blades then the time series can be selected to reflect the correlation. Correlation would be expected to exist between blades for the mechanism of inflow turbulence.

The main idea is to fit the time series to represent what is desired, then let the program compute the resulting sound radiation.

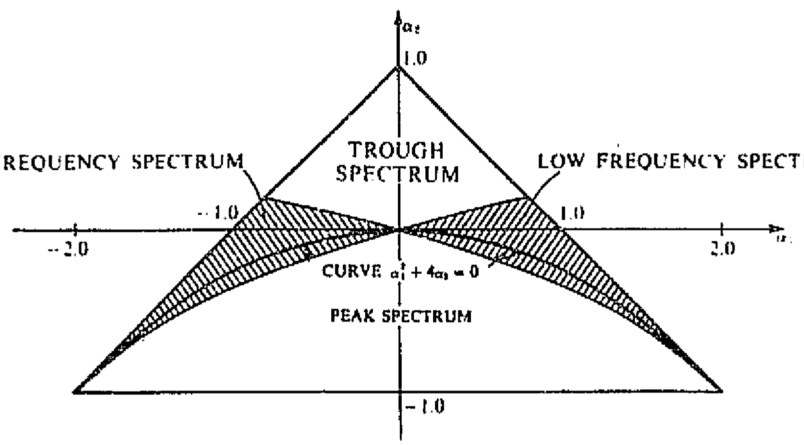
The spectrum for the second order process above can be solved for directly. The spectrum is

$$G_{xx}(f) = H\sigma_z^2/(1 + a_1^2 + a_2^2 - 2a_1(1-a_2) \cos 3\pi fH - 2a_2 \cos 4\pi fH)$$

$$- 1/2H \leq f \leq 1/2H.$$

Selection of $a_1$ and $a_2$ gives the spectral shape to be expected from x(t).

A plot of $a_1$ and $a_2$ for different spectral shapes is shown below to give an idea of the values of the coefficients needed for particular shapes.



A peaked spectrum is what we would expect for a wake related sound generation mechanism. A low frequency spectrum would be expected for the cases of inflow turbulence. The frequency where the peak will occur is givey by

$$\cos 2\pi f_o H = -a_1 (1 - a_2)/4a_2 .$$

As may be expected, some experimentation will be required to fit the process to an actual experimental spectrum.

To obtain more complicated power spectral shapes higher order processes may be needed. These can be handled as easily as the above on the computer, but the preliminary efforts to get a particular spectrum increase.

For a good general discussion of the above method the reader is referred to SPECTRAL ANALYSIS AND ITS APPLICATIONS by G.M. Jenkins and D.G. Watts (1969).

If more simply definable fluctuations are needed these can be used for x(t). For example, inlet distortion or the effect of another blade row can be simulated by a Fourier series representation of the force fluctuation resulting from the distortion. A simple cosine with period equal to the stator spacing will give a good qualitative model of blade row interaction noise.

The model can also be extended to account for span distributions of the blade force. This is straightforward but the computer time required does increase substantially.

## 4.6 Program Listing

```
      PROGRAM TONE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

C     COMPUTE SOUND RADIATION FROM A ROTATING ARRAY
C     OF POINT FORCES

      COMPLEX C
      DIMENSION PRAD(1100),THEP(20),THE(20),THER(20),THERP(20)
      DIMENSION C(1024),B(2,1024),INV(256),M(3),S(256),AMP(1024),
     1SPL(1024),FREQ(512),PRA(1024),Z(1024),X(4,1024),DATO(2048)
      EQUIVALENCE(C,B)

C     NHAR = HARMONICS TO BE ANALYZED
C     NOPT = OPTION
C     BE = FREQUENCY BANDWIDTH DESIRED
C     ZR = NUMBER OF ROTOR BLADES
C     XI = STAGGER ANGLE, MEASURED FROM FAN AXIS (DEG)
C     PHI = RADIATION ANGLE - MEASURED FROM FAN AXIS
C     RPM = ROTOR SPEED (RPM)
C     RS = SOURCE RADIUS (M)
C     R1 = OBSERVER RADIUS (M)
C     F = FORCE ON EACH ROTOR BLADE (KG)
C     CDYN = DYNAMIC LOADING - PER CENT OF TOTAL FORCE
C     A1 = AUTO-REGRESSION CONSTANT
C     A2 = AUTO-REGRESSION CONSTANT

C          X(J,I) = A1*X(J,I-1) + A2*X(J,I-2) + Z(I)'
C     Z(I) = RANDOM VARIABLE

      DATA PI,AO/3.14159,342./


C     INPUT M1 SUCH THAT C HAS LENGTH N=2**M1

      READ(5,1)  NHAR,NOPT
    1 FORMAT(2I5)
      READ(5,2)  BE

C     OPTION = 0  NO PLOTS
C     OPTION = 1  PLOT TIME HISTORY
C     OPTION = 2  PLOT SPECTRUM
C     OPTION = 3  PLOT BOTH TIME HISTORY AND SPECTRUM

      READ(5,2) ZR,XI,PHI,RPM,RS,R1,F
      READ(5,2)  CDYN,A1,A2
    2 FORMAT(7F10.2)

C     CONDITION INPUT DATA

      CMFT = 3.28
      RS = RS*CMFT
      R1=R1*CMFT
      F=F*2.21
      FMAX=FLOAT(NHAR)*RPM/60.*ZR
      FMAX=10000.
      N1=IFIX(2.*FMAX/BF)
      DO 75  I=1,10
```

```
      N2=2**I
      IF(N2.GT.N1)  GO TO 71
   75 CONTINUE
   71 NSAMP=N2
      M1=I
C
C     THE MAXIMUM NUMBER OF POINTS ARE 2**10 (1024)
C
      BE=2.*FMAX/FLOAT(NSAMP)
      XOMEG=RPM*2.*PI/60.
      XIP=XI*PI/180.
      PHIP=PHI*PI/180.
      SINPHI=SIN(PHIP)
      COSPHI=COS(PHIP)
      SINXI=SIN(XIP)
      COSXI=COS(XIP)
      XMS=RS*XOMEG/AO
      NZR=ZR
C
C     RANDOM FORCE ON THE ROTOR BLADES MODELLED BY UNCORRELATED
C     SECOND ORDER PROCESS
      IX=2**24+3
      NRAND=NSAMP
      DO 31 J=1,NZR
      IX=IX+10
      CALL RAND(I0,IX,1,NRAND,Z)
      X(J,1)=1.0
      X(J,2)=1.0
      DO 31 I=3,NSAMP
   31 X(J,I)=A1*X(J,I-1)+A2*X(J,I-2)+Z(I)
C
C     WRITE INPUT DATA
C
      WRITE(6,20)  ZR,XI,RPM,F,RS,PHI,R1
   20 FORMAT(1H1,51H RADIATION OF SOUND FROM A ROTATING ARRAY OF FORCES
     1///25H NUMBER OF ROTOR BLADES    1F5.0/20H BLADE STAGGER ANGLE
     21F5.2/18H ROTOR SPEED (RPM)    1F10.2/12H BLADE FORCE   1F10.2/
     331H RADIAL LOCATION OF POINT FORCE    1F5.2/
     416H RADIATION ANGLE  1F5.2/16H OBSERVER RADIUS   1F5.1///)
      WRITE(6,22)  FMAX,BE
   22 FORMAT(//18H MAXIMUM FREQUENCY
     11F20.3,15H BANDWIDTH (HZ)    1F10.2//)
      WRITE(6,23)  CDYN,A1,A2
   23 FORMAT(27H RANDOM FORCE FLUCTUATIONS     //
     137H X(I) = A1*X(I-1) + A2*X(I-2) + Z(I)       /
     223H DYNAMIC COEFFICIENT =    1F10.3  /
     35H A1 =  1F10.3,10X,5H A2 =  1F10.3    //)
C
C     INITIALIZE BLADE POSITIONS
C
      DO 10 I=1,NZR
      UI=I
   10 THE(I)=2.*PI/ZR*(UI-1.)
C
C     COMPUTE TIME SERIES BASED ON RETAINING SUFFICIENT
C     INFORMATION FOR 10 HARMONICS .....
```

```
C      SO, SAMPLE AT 20 SAMPLES PER BLADE SPACING
C      SET TIME INCREMENT
       FB=F
       DT=0.5/FMAX
       DO 200 K=1,NSAMP
       UK=K
       T=DT*(UK-1.)
C      COMPUTE SOUND RADIATION FROM THE BLADE ARRAY
       PRAD(K)=0.0
       DO 300 J=1,NZR
       UJ=J
       IF(K.NE.1) GO TO 301
       THEP(J)=THE(J)
       GO TO 302
   301 THEP(J)=THERP(J)
   302 CONTINUE
       THER(J)=XOMEG*(T-R1/AO+RS/AO*SINPHI*COS(THEP(J)))+THE(J)
       THERP(J)=THER(J)
       XMR=-XMS*SINPHI*SIN(THER(J))
       DXMR=-XMS*XOMEG*SINPHI*COS(THER(J))
       R=R1-RS*SINPHI*COS(THER(J))
       F=FB*(1.+CDYN*X(J,K))
       VAR1=F/((4.*PI*AO*R**2)*(1.-XMR)**2)
       VAR2=XOMEG*SINPHI*(RS-R1*SINPHI*COS(THER(J)))
       VAR3=-R1*DMR/(1.-XMR)*(COSXI*COSPHI+
      1SINXI*SINPHI*SIN(THER(J)))
       PBLADE=VAR1*(VAR2+VAR3)
   300 PRAD(K)=PRAD(K)+PBLADE
   200 CONTINUE
C      THIS COMPLETES COMPUTATION OF THE TIME SERIES FOR
C      ONE RADIATION ANGLE
       WRITE(6,21)  (I,PRAD(I),I=1,NSAMP)
    21 FORMAT(4(5X,I15,5X,1E15.3))
       USAMP=NSAMP
C      FIND THE MEAN VALUE
       PSUM=0.0
       DO 400 I=1,NSAMP
   400 PSUM=PSUM+PRAD(I)
       PMEAN=PSUM/USAMP
       WRITE(6,30) PMEAN
    30 FORMAT(///29H MEAN RADIATED SOUND PRESSURE     1E20.5///)
       DO 500 I=1,NSAMP
   500 PRA(I)=PRAD(I)/PMEAN
       IF(NOPT.EQ.1)  GO TO 701
       IF(NOPT.EQ.4)  GO TO 701
```

```
      GO TO 702
  701 CONTINUE

C     CONDITION THE DATA FOR THE TIME PLOT

C     FIND MAXIMUM AND MINIMUM VALUES IN THE PRESSURE ARRAY

      PMAX=-999 $ PMIN=999
      DO 600 I=1,NSAMP
      IF(PRA(I).LT.PMIN)  PMIN=PRA(I)
      IF(PRA(I).GT.PMAX)   PMAX=PRA(I)
  600 CONTINUE

C     PLOT THE NORMALIZED ACOUSTIC PRESSURES

      CALL TPLOT(PRA,NSAMP,DT,PMIN,PMAX,NZK)

  702 CONTINUE

C     PROVIDE ARRAY C
      DO 40 I=1,NSAMP
      II=2*I
   40 DATO(II-1)=PRAD(I)
      NNN=2*NSAMP
      WRITE(6,41)  (DATO(I),I=1,NNN)
      CALL FOUR1(DATO,NSAMP,-1)
      DO 70 I=1,NSAMP
      II=2*I
   70 C(I)=CMPLX(DATO(II-1),DATO(II))
C     ARRAY C NOW CONTAINS THE FREQUENCY COMPONENTS
C     COMPUTE AMPLITUDE AND PHASE ANGLE FORM
      WRITE(6,41)  (C(I),I=1,50)
   41 FORMAT(2(10X,1E15.3,5X,1E15.5))
      NF=NSAMP/2
      DO 50 I=1,NF
      AMP(I)=CABS(C(I))
      AMP(I)=AMP(I)/NF
   50 SPL(I)=20.*ALOG10(AMP(I)/20.*E-06)

C     USE FAST FOURIER TRANSFORM TO EXPRESS IN TERMS OF FREQUENCY


C     COMPUTE THE FREQUENCY ARRAY

      FMAX=1./(2.*DT)
      DF=FMAX*2./FLOAT(NSAMP)
      DO 60 I=1,NF
      UI=I
   60 FREQ(I)=DF*UI

      WRITE(6,61)
   61 FORMAT(//24H RADIATED SOUND SPECTRUM    //)
      WRITE(6,62)  (FREQ(I),SPL(I),I=1,NF)
   62 FORMAT(4(1F10.2,5X,1F10.2,5X))
```

```
      IF(NOPT.EQ.2)  GO TO 703
      IF(NOPT.EQ.4)  GO TO 703
      GO TO 1000
  703 CONTINUE
C     NOW PLOT THE SPECTRUM OF THE RADIATED SOUND
C     CONDITION THE DATA FOR THE SPECTRAL PLOT

      NF=NSAMP/2
      CALL FPLOT(SPL,FREQ,NF)

 1000 CONTINUE
      STOP
      END
      SUBROUTINE FOUR1(DATA,N,ISIGN)

C     A SIMPLE FAST FOURIER TRANSFORM

      DIMENSION DATA(2048)
      PI=ACOS(-1.)
      TPI=2.*PI
      IP0=2
      IP3=IP0*N
      I3REV=1
      DO 50 I3=1,IP3,IP0
      IF(I3-I3REV)10,20,20
   10 TEMPR=DATA(I3)
      TEMPI=DATA(I3+1)
      DATA(I3)=DATA(I3REV)
      DATA(I3+1)=DATA(I3REV+1)
      DATA(I3REV)=TEMPR
      DATA(I3REV+1)=TEMPI
   20 IP1=IP3/2
   30 IF(I3REV-IP1)  50,50,40
   40 I3REV=I3REV-IP1
      IP1=IP1/2
      IF(IP1-IP0)50,30,30
   50 I3REV=I3REV+IP1
      IP1=IP0
   60 IF(IP1-IP3)  70,100,100
   70 IP2=IP1*2
      THETA=TPI/FLOAT(ISIGN*IP2/IP0)
      SINTH=SIN(THETA/2.)
      WSTPR=-2.*SINTH*SINTH
      WSTPI=SIN(THETA)
      WR=1.
      WI=0.
      DO 90 I1=1,IP1,IP0
      DO 80 I3=I1,IP3,IP2
      I2A=I3
      I2B=I2A+IP1
      TEMPR=WR*DATA(I2B)-WI*DATA(I2B+1)
      TEMPI=WR*DATA(I2B+1)+WI*DATA(I2B)
      DATA(I2B)=DATA(I2A)-TEMPR
      DATA(I2B+1)=DATA(I2A+1)-TEMPI
      DATA(I2A)=DATA(I2A)+TEMPR
   80 DATA(I2A+1)=DATA(I2A+1)+TEMPI
```

```
      TEMPR=WR
      WR=WR*WSTPR-WI*WSTPI+WR
   90 WI=WI*WSTPR+TEMPR*WSTPI+WI
      IP1=IP2
      GO TO 60
  100 RETURN
      END
      SUBROUTINE TPLOT(PRAD,NSAMP,DT,PMIN,PMAX,NPAS)
      DIMENSION PRAD(1100),T(1100)

C     INITIALIZE THE PLOT

      CALL PRNTON

C     PLOT SIZE TO BE 5 BY 7 INCHES
C     SET OBJECT SPACE

C     NPAS IS THE NUMBER OF BLADE PASSAGES TO BE PLOTTED

      UPAS = NPAS
      TMAX=UPAS*20.*DT

      CALL STSZOB(2.,9.,2.,7.)
      CALL STSOBJ(0.,TMAX,PMIN,PMAX)

C     CONSTRUCT AXES

      CALL STNDIV(1,2)
      CALL AXLILI
      CALL SAXLIT
      CALL SAXLIR

C     LABEL SCALES

      CALL STNDIV(1,2)
      CALL STNDEC(0)
      CALL NOSLIB
      CALL NOSLIL

C     TITLE SCALES

      CALL STNCHR(11)
      CALL TITLEB(11H TIME (SEC) )
      CALL STNCHR(29)
      CALL TITLEL(29H FAR FIELD ACOUSTIC PRESSURE   )

C     GENERATE THE TIME ARRAY

      NT=NPAS*20
      TI=0.0
      DO 10 I=1,NT
      UI=I
   10 T(I)=TI+DT*(UI-1.)

C     PLOT THE DATA USING STRAIGHT LINES
```

```
      CALL STNPTS(NT)
      CALL SLLILI(T,PRAD)
C     TERMINATE PLOTTING
      CALL EXITPL
      RETURN
      END
      SUBROUTINE FPLOT(SPL,FREQ,NF)
      DIMENSION SPL(1024),FREQ(1024)
C     INITIALIZE THE PLOT
      CALL STCCON(48H TONE NOISE GENERATION SPECTRUM                  )
C     PLOT SIZE TO BE 5 BY 7 INCHES
C     SET OBJECT SPACE
      CALL STSZOB(2.,9.,2.,7.)
      CALL STSOBJ(10.,10000.,50.,150.)
C     CONSTRUCT AXES
      CALL STNDIV(30,20)
      CALL AXLGLI
      CALL SAXLGT
      CALL SAXLIR
C     LABEL SCALES
      CALL STNDIV(3,10)
      CALL STNDEC(0)
      CALL NOSLIL
      CALL NOLGB
C     TITLE SCALES
      CALL STNCHR(15)
      CALL TITLEB(15H FREQUENCY (HZ)   )
      CALL STNCHR(37)
      CALL TITLEL(37H REDIATED SPL (DB RE 20 MICRO N/M**2)    )
C     PLOT THE DATA USING STRAIGHT LINES
      CALL STNPTS(NF)
      CALL SLLGLI(FREQ,SPL)
C     TERMINATE PLOTTING
      CALL EXITPL
      RETURN
      END
```

## 5.0 PROGRAM SDATA

Program SDATA is used to perform correlation and spectral analysis on discrete time series. The particular operations carried out on two given time series are:

1) Autocovariance function for each sample and cross covariance between the two samples

2) Autocorrelation function for each sample and cross correlation between the two samples

3) Autospectrum for each sample

4) Magnitude and phase representation of cross spectrum between the two samples

5) Squared coherency function

The analysis used as the basis for the program is that presented in SPECTRAL ANALYSIS AND ITS APPLICATIONS, by G.M. Jenkins and D.G. Watts. The method used is exactly that recommended in this reference.

Two input possibilities exist. The first is to input two time series of arbitrary length. For this case the program will compute all the above covariances, correlations, and spectral quantities. The second possibility is to input two covariance functions. In this case the program will only carry out the spectral analysis. Some electronic instruments have become available recently which quickly compute correlation functions. This program can be used in conjunction with these instruments to obtain the spectral representation of the data.

A classical Fourier transform was used in the program. For this reason long samples, in terms of correlation lags, will require substantial computer time. Therefore, a Fast Fourier Transform program should be used for large samples.

## 5.1 Outline of Computations

This outline is substantially the same as that presented by Jenkins and Watts (pages 382 and 383).

Two time series of data, $x_{t1}$ and $x_{t2}$, sampled at increments of H sec for a total of N points. For convenience H is assumed equal to one. The frequency range is then $0 \leq f \leq 1/2$ Hz. For non-unity values of H the frequency range is $0 \leq f \leq 1/2H$. The actual spectral levels are also to be multiplied by H to obtain quantitatively correct values.

The computations are:

(1) For the $x_{t1}$ data:

(a) the autocovariance estimate is

$$c_{11}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (x_{1t}-\bar{x}_1)(x_{1t+k}-\bar{x}_1) \qquad 0 \leq k \leq M-1$$

where M is the number of time lags in the estimate and the mean value is

$$\bar{x}_1 = \frac{1}{N} \sum_{t=1}^{N} x_{1t}$$

(b) the smoothed spectral estimate using a Tukey window

$$\overline{C_{11}}(j) = 2 \{c_{11}(0) + 2 \sum_{k=1}^{M-1} c_{11}(k)\ w\ (k) \cos \frac{\pi k j}{F}\} \quad 0 \leq j \leq F$$

F is a measure of the spacing of the frequency points. The smoothed spectral estimates are to be computed at 0,1/2F, ...,1/2 where F is of the order 2 to 3 times M. The Tukey window w(k) is defined by

$$w(k) = \frac{1}{2}\,(1 + \cos\frac{\pi k}{M}) \qquad |k| \le M$$

$$= 0 \qquad |k| > M\,.$$

(2) For the $x_{t2}$ data:

(a) the autocovariance function estimate

$$c_{22}(k) = \frac{1}{N}\sum_{t=1}^{N-k} (x_{2t}-\bar{x}_2)(x_{2t+k}-\bar{x}_2) \qquad 0 \le k \le M-1$$

with the mean value

$$\bar{x}_2 = \frac{1}{N}\sum_{t=1}^{N} x_{2t}\,.$$

(b) the smoothed spectral estimate

$$\bar{c}_{22}(j) = 2\{c_{22}(0) + 2\sum_{k=1}^{M-1} c_{22}(k)\,w\,(k)\,\cos\frac{\pi kj}{F}\} \qquad 0 \le j \le F$$

where the same window is used (Tukey) for w(k).

(3) For the $x_{t1}$ and the $x_{t2}$ data:

(a) the cross covariance estimate

$$c_{12}(k) = \frac{1}{N}\sum_{t=1}^{N-k} (x_{1t}-\bar{x}_1)(x_{2t+k}-\bar{x}_2) \qquad 0 \le k \le M-1$$

$$c_{21}(k) = c_{12}(-k) = \frac{1}{N}\sum_{t=1}^{N-k} (x_{1t+k}-\bar{x}_1)(x_{2t}-\bar{x}_2) \qquad 0 \le k \le M-1$$

(b) the even and odd cross covariance estimates

$$\ell_{12}(k) = \frac{1}{2}\,\{c_{12}(k) + c_{12}(-k)\} \qquad 0 \le k \le M-1$$

$$q_{12}(k) = \frac{1}{2}\,\{c_{12}(k) - c_{12}(-k)\} \qquad 0 \le k \le M-1$$

(c) the smoothed co- and quadrature spectral estimates

$$\bar{L}_{12}(j) = 2\ \{\ell_{12}(0) + 2 \sum_{k=1}^{M-1} \ell_{12}(k)\ w\ (k)\ \cos \frac{\pi jk}{F}\} \qquad 0 \le j \le F$$

$$\bar{Q}_{12}(j) = 4 \sum_{k=1}^{M-1} q_{12}(k)\ w\ (k)\ \sin \frac{\pi jk}{F} \qquad 1 \le j \le F-1$$

note that $q_{12}(0) = 0$ and that $Q_{12}(0) = Q_{12}(F) = 0$.

(d) the smoothed cross amplitude spectral estimate

$$\bar{A}_{12}(j) = \sqrt{\bar{L}^2_{12}(j) + \bar{Q}^2_{12}(j)} \qquad 0 \le j \le F$$

(e) the smoothed phase spectral estimate

$$\bar{F}_{12}(j) = \arctan\ \{- \frac{\bar{Q}_{12}(j)}{\bar{L}_{12}(j)}\} \qquad 0 \le j \le F$$

(f) the smoothed squared coherency spectral estimate

$$\bar{K}^2_{12}(j) = \frac{\bar{A}^2_{12}(j)}{\bar{c}_{11}(j)\ \bar{c}_{22}(j)} \qquad 0 \le j \le F$$

## 5.2 Correction for Bias Error Reduction

When there is a large delay in the correlation a large error can result in the coherency and phase spectrum. This bias error can be reduced by shifting the cross covariance function so that the maximum correlation occurs at zero time lag.

The way this correction is made in the present program is to search for the time lag corresponding to the largest absolute value of covariance. A new covariance function is then generated such that the peak occurs at the zero time lag. This function is Fourier transformed and the phase shift introduced by the time shift is applied to the spectral estimate to correct it back. A consequence of the generation of a new covariance function is that it necessarily will contain fewer terms than the original. This results in a wider frequency bandwidth in the spectral estimate. This is why the program will seem to generate different bandwidths for different batches of data even when the same number of lags are called out in the input.

## 5.3 Input to Program

Two possibilities exist for input to this program. The first is to read in two time series of variables for which correlation and spectral computations are required. This is obtained from data card one by setting the variable OPTION = 1, FORMAT(1I5). If OPTION is not set equal to one the program will read in the data as if it were covariance functions. Input for OPTION = 1 is described in Section 5.3.1 and for OPTION ≠ 1 in Section 5.3.2.

CARD 1, FORMAT(1I5)

OPTION = 1 read in time series

≠ 1 read in covariance functions

### 5.3.1 OPTION = 1

CARD 2, FORMAT(2I5)

NSAMP number of points in the time sample

MLAG number of time lags desired in the covariance functions

typically MLAG is chosen to be approximately one tenth as large as NSAMP.

CARD 3, FORMAT(1F10.2)

DELT time increment in samples (Sec)

The remaining inputs under this option are the actual values of the variable in the time series. They are read 7 values to each card in the FORMAT(7F10.2). The number of cards required depends upon the size of the sample. The computer call is

```
     DO 10  I = 1,2
  10 READ(5,2)  (X(J,I),I = 1,NSAMP)
```

This is the total input required for this option.

5.3.2 OPTION ≠ 1

CARD 2, FORMAT(2F10.2)

| | | |
|---|---|---|
| XLAG | number of values in the covariance function samples | |
| DELT | time increment in the functions | (Sec) |

The remaining cards are used to read in the covariance functions. In the present format the functions are read in as follows.

```
     DO 10  I = 1,2
     DO 10  J = 1,2
  10 READ(5,2)  (COV(K,I,J), K = 1,M1)
```

where M1 = XLAG + 1 and the indices I,J mean the following:

COV(K,1,1) = $C_{11}(k)$ = autocovariance function for series 1, k = 0,M

COV(K,2,2) = $C_{22}(k)$ = autocovariance function for series 2, k = 0,M

COV(K,1,2) = $C_{12}(k)$ = cross covariance function for positive time lags, k = 0,M

COV(K,2,1) = $C_{21}(k)$ = cross covariance function for negative time lags, k = 0,M

The format used to read in these functions is FORMAT(7F10.2). This completes the input requirements for this option.

## 5.4 Program Input

The output of the present form of this program is simply a listing of each of the functions computed. The list of functions computed was presented in Section 1.0.

## 5.5 Program Listing

```
      PROGRAM SDATA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     A STATISTICAL DATA ANALYSIS PROGRAM WHICH COMPUTES THE FUNCTIONS
C     AUTO COVARIANCE AND CROSS COVARIANCE AS WELL AS
C     AUTO AND CROSS SPECTRUM
      DIMENSION X(1024,2),COV(101,2,2),Z(200)

C     READ IN THE TIME SERIES

C     NSAMP = NUMBER OF TIME SAMPLES
C     MLAG = NUMBER OF CORRELATION LAGS
C     DELT = TIME INCREMENT (SEC)
C     X(I,J) = THE TWO SAMPLES TO BE ANALYZED

      DATA PI/3.14159/
      READ(5,1)  OPTION

C     OPTION = 1   READ TIME SERIES OTHERWISE READ COVARIANCES

      IF(OPTION.EQ.1)  GO TO 50
      READ(5,1) NSAMP,MLAG
      READ(5,2)  DELT
      DO 10 I=1,2
   10 READ(5,2)  (X(J,I),J=1,NSAMP)
      WRITE(6,40)  (I,X(I,1),X(I,2),I=1,NSAMP)
   40 FORMAT(2(5X,1I5,5X,2E20.5))

C     FORM COVARIANCES AND CORRELATIONS

      CALL COVAR(NSAMP,X,COV,MLAG)

C     COMPUTE THE CROSS POWER SPECTRAL DENSITY

      GO TO 22
   50 READ(5,2)  XLAG,DELT
       MLAG=IFIX(XLAG)
      M1=MLAG+1
      DO 21 J=1,2
      DO 21 I=1,2
   21 READ(5,2) (COV(K,I,J),K=1,M1)
   22 CONTINUE

C     COVARIANCES ARE READ IN THE ORDER (1,1),(1,2),(2,1),(2,2)

      CALL CRSPEC(COV,MLAG,DELT,NSAMP)
    1 FORMAT(2I5)
    2 FORMAT(7F10.2)
      STOP
      END
      SUBROUTINE COVAR(NSAMP,X,COV,MLAG)
      DIMENSION SM(2),X(1024,2),COV(101,2,2),DCOV(101,2,2),
     1COR(101,2,2),DCOR(101,2,2),SUM(2),XM(2)

C     THE AUTO AND CROSS COVARIANCES AS WELL AS THE AUTO AND
C     CROSS CORRELATIONS ARE COMPUTED
C     COMPUTE MEAN VALUES
```

```
      DO 1 J=1,2
      SUM(J)=0.0
      DO 2 I=1,NSAMP
    2 SUM(J)=SUM(J)+X(I,J)
    1 XM(J)=SUM(J)/NSAMP
C     PRINT MEAN VALUES
      WRITE(6,10)
   10 FORMAT(1H1,36H RESULTS OF CORRELATION COMPUTATIONS    )
      WRITE(6,11)   (XM(J),J=1,2)
   11 FORMAT(//24H MEAN VALUE SERIES ONE =    1E20.5   /
     124H MEAN VALUE SERIES TWO =    1E20.5 //)
C     COMPUTE DEVIATIONS FROM THE MEAN
      DO 3 J=1,2
      DO 3 I=1,NSAMP
    3 X(I,J)=X(I,J)-XM(J)
C     COMPUTE COVARIANCES
      M1=MLAG
      DO 4 K=1,M1
      DO 4 J=1,2
      DO 4 L=1,2
      SUM1=0.0
      M2=NSAMP-K
      DO 5 I=1,M2
    5 SUM1=X(I,J)*X(I+K-1,L)+SUM1
    4 COV(K,J,L)=SUM1/FLOAT(NSAMP)
C      COMPUTE DIFFERENCED COVARIANCES
      DO 6 L=1,2
      DO 6 J=1,2
      DCOV(1,J,L)=-COV(2,L,J)+2.*COV(1,J,L)-COV(2,J,L)
      M3=M1-1
      DO 6 K=2,M3
    6 DCOV(K,J,L)=-COV(K-1,J,L)+2.*COV(K,J,L)-COV(K+1,J,L)
      WRITE(6,12)
   12 FORMAT(31H TABLE OF AUTO COVARIANCE (1-1)     )
      WRITE(6,14)   (K,COV(K,1,1),K=1,M1)
   14 FORMAT(4(1I5,5X,1F15.5,5X))
      WRITE(6,15)
   15 FORMAT(31H TABLE OF AUTO COVARIANCE (2-2)     )
      WRITE(6,14)   (K,COV(K,2,2),K=1,M1)
      WRITE(6,16)
   16 FORMAT(32H TABLE OF CROSS COVARIANCE (1-2)     )
      WRITE(6,14)   (K,COV(K,1,2),K=1,M1)
      WRITE(6,17)
   17 FORMAT(32H TABLE OF CROSS COVARIANCE (2-1)     )
      WRITE(6,14)   (K,COV(K,2,1),K=1,M1)
C     OUTPUT DIFFERENCED COVARIANCES
```

```
      WRITE(6,18)
   18 FORMAT(1H1,46H DIFFERENCED COVARIANCES (SAME ORDER AS ABOVE)   //)

C     COMPUTE CORRELATIONS NORMALIZED COVARIANCES
C     BOTH NORMAL AND DIFFERENCED FORMS

      DO 20 L=1,2
      DO 20 J=1,2
      COVM=SQRT(COV(1,J,J)*COV(1,L,L))
      DCOVM=SQRT(DCOV(1,J,J)*DCOV(1,L,L))
      DO 20 K=1,M1
      COR(K,J,L)=COV(K,J,L)/COVM
   20 DCOR(K,J,L)=DCOV(K,J,L)/DCOVM

C     PRINT THE CORRELATIONS AND THEIR DIFFERENCED FORMS

      WRITE(6,21)
   21 FORMAT(1H1,50H AUTO AND CROSS CORRELATIONS AND DIFFERENCED FORMS
     1///48H AUTO CORRELATION (WITH DIFFERENCED FORM) (1-1)   //)
      WRITE(6,14)  (K,COR(K,1,1),K=1,M1)
      WRITE(6,22)
   22 FORMAT(//23H AUTO CORRELATION (2-2)    )
      WRITE(6,14)  (K,COR(K,2,2),K=1,M1)
      WRITE(6,23)
   23 FORMAT(///25H CROSS CORRELATION (1-2)     )
      WRITE(6,14)  (K,COR(K,1,2),K=1,M1)
      WRITE(6,24)
   24 FORMAT(///25H CROSS CORRELATION (2-1)     )
      WRITE(6,14)  (K,COR(K,2,1),K=1,M1)

C     THIS COMPLETES THE CORRELATION PHASE

      RETURN
      END
      SUBROUTINE CRSPEC(COV,MLAG,DELT,NSAMP)

C     THE CROSS POWER SPECTRAL DENSITY IS COMPUTED

      DIMENSION COV(101,2,2),COVP(101),SPECP(303),SPEC(303,2),FREQ(303),
     1EV(101),XOD(101),SQ(303),COSPEC(303),QSPEC(303),PHASE(303),
     2COHSQ(303),ASPEC(303,2),DUM(202),DUP(202)
      DIMENSION CSPEC(303),FREC(303)
      DATA PI/3.14159/
      M1=MLAG

C     FIND S  WHERE S IS THE NUMBER OF TIME LAGS TO ALIGN THE TWO
C     PROCESSES WITH MAXIMUM CORRELATION AT ZERO
C     DETERMINE LAG TIME FOR SERIES ALIGNMENT

      WRITE(6,70)  (COV(I,1,2),COV(I,2,1),I=1,M1)
      DO 21 K=1,MLAG
   21 DUM(K)=COV(M1+1-K,2,1)
      DO 22 K=2,M1
   22 DUM(MLAG+K-1)=COV(K,1,2)
      MM=2*MLAG-1
      WRITE(6,26)  (I,DUM(I),I=1,MM)
```

```
   26 FORMAT(4(5X,I15,5X,1E15.5))
      XMAX=-999.
      DO 23 K=1,MM
      DUP(K)=ABS(DUM(K))
      IF(DUP(K).GT.XMAX)  GO TO 24
      GO TO 23
   24 XMAX=DUP(K)
      IS=K-M1
   23 CONTINUE
      WRITE(6,25)  IS
   25 FORMAT(///22H ALIGNMENT LAG NUMBER   1I10///)

C     CONDITION THE DATA FOR THE CALCULATION OF THE
C      CROSS SPECTRUM

      M2=M1-IABS(IS)

C     CONDITION THE DATA FOR AUTOSPECTRUM CALCULATION

      DO 20 K=1,2
      DO 10 I=1,M1
   10 COVP(I)=COV(I,K,K)
      CALL AUSPEC(MLAG,DELT,COVP,SPECP,FREQ,IS)

C     STORE AUTOSPECTRUM

      NF=2*M2+1
      DO 11 I=1,NF
   11 SPEC(I,K)=SPECP(I)
   20 CONTINUE
      EV(1)=DUM(M1+IS)
      XOD(1)=0.0
      DO 40 K=2,M2
      MMM=M1+IS+K-1
      MMMM=M1+IS-K+1
      EV(K)=0.5*(DUM(MMM)+DUM(MMMM))
   40 XOD(K)=0.5*(DUM(MMM)-DUM(MMMM))
      WRITE(6,70)  (EV(I),XOD(I),I=1,M2)

C     DO THE FOURIER TRANSFORM

      CALL CROSPEC(EV,XOD,COSPEC,QSPEC,MLAG,DELT,IS)

C     COMPUTE PHASE ANGLE AND COHERENCE FUNCTION


C     RETURN CROSS SPECTRUM TO UNSHIFTED FORM

      DO 60 I=1,NF
      ARG=PI*FLOAT(I-1)*FLOAT(IS)/FLOAT(NF)
      CN=COS(ARG)
      SN=SIN(ARG)
      DUM1=COSPEC(I)*SN+QSPEC(I)*CN
      DUM2=COSPEC(I)*CN-QSPEC(I)*SN
      COSPEC(I)=DUM2
      QSPEC(I)=DUM1
```

```
      SQ(I)=COSPEC(I)*COSPEC(I)+QSPEC(I)*QSPEC(I)
       IF(COSPEC(I).EQ.0.0)  GO TO 80
      PHASE(I)=ATAN2(-QSPEC(I),COSPEC(I))
      GO TO 81
   80 PHASE(I)=PI/2.
   81 CONTINUE
      PHASE(I)=PHASE(I)*180./PI
      CSPEC(I)=SQRT(SQ(I))
      COHSQ(I)=SQ(I)/(SPEC(I,1)*SPEC(I,2))
      IF(SPEC(I,1).LE.0.0)  GO TO 51
      GO TO 55
   52 CONTINUE
      IF(SPEC(I,2).LE.0.0)  GO TO 53
      GO TO 54
   51 ASPEC(I,1)=-100.
      GO TO 52
   53 ASPEC(I,2)=-100.
      GO TO 60
   55 CONTINUE
      ASPEC(I,1)=ALOG10(SPEC(I,1))
      GO TO 52
   54 CONTINUE
      ASPEC(I,2)=ALOG10(SPEC(I,2))
   60 CONTINUE
C     GENERATE THE FREQUENCY ARRAY
      BL=1.33/(FLOAT(MLAG)*DELT)
      DO 41 I=1,NF
   41 FREQ(I)=FLOAT(I-1)/(2.*DELT*FLOAT(NF-1))
C     WRITE THE SPECTRAL LEVELS (CROSS PSD) ,PHASE AND COHERENCY
      WRITE(6,61)
   61 FORMAT(1H1,35H RESULTS OF POWER SPECTRAL ANALYSIS   ///
     119H AUTOSPECTRUM (1-1)    //)
   70 FORMAT(3(5X,1F10.4,5X,1E15.5))
      WRITE(6,70)  (FREQ(I),SPEC(I,1),I=1,NF)
      WRITE(6,62)
   62 FORMAT(///19H AUTOSPECTRUM (2-2)     ///)
      WRITE(6,70)  (FREQ(I),SPEC(I,2),I=1,NF)
      WRITE(6,65)
   65 FORMAT(///15H CROSS SPECTRUM   ///)
      WRITE(6,70)  (FREQ(I),CSPEC(I),I=1,NF)
      WRITE(6,63)
   63 FORMAT(///12H PHASE ANGLE   ///)
      WRITE(6,70)  (FREQ(I),PHASE(I),I=1,NF)
      WRITE(6,64)
   64 FORMAT(///11H COHFRENCY       ///)
      WRITE(6,70)  (FREQ(I),COHSQ(I),I=1,NF)
      RETURN
      END
      SUBROUTINE CROSPEC(EV,XOD,COSPEC,QSPEC,MLAG,DELT,IS)
      DIMENSION EV(101),XOD(101),COSPEC(303),QSPEC(303),W(101)
C     COMPUTES SMOOTHED DO- AND QUAD- SPECTRA AND SQUAREDAMPLITUDE
```

```
      DATA PI/3.14159/
      M1=MLAG
      M=M1-IABS(IS)
      M2=M-1
      NF=2*M+1
C     COMPUTE WEIGHTS USING TUKEY WINDOW
      DO 20 J=1,M2
   20 W(J)=0.5*(1.+COS(PI*FLOAT(J)/FLOAT(M)))
      DO 10 I=1,NF
      SUM1=0.0
      SUM2=0.0
      DO 30 K=1,M2
      ARG=PI*FLOAT(K)*FLOAT(I-1)/FLOAT(NF-1)
      SN=SIN(ARG)
      CS=COS(ARG)
      SUM1=SUM1+EV(K+1)*W(K)*CS
   30 SUM2=SUM2+XOD(K+1)*W(K)*SN
      COSPEC(I)=2.*DELT*(EV(1)+2.*SUM1)
   10 QSPEC(I)=4.*DELT*SUM2
      RETURN
      END
      SUBROUTINE AUSPEC(MLAG,DELT,COV,SPEC,FREQ,IS)
C     COMPUTE AUTOSPECTRUM OF COVARIANCES
      DIMENSION COV(101),W(101),SPEC(303),FREQ(303)
      DATA PI/3.14159/
C     THE TUKEY WINDOW IS USED FOR DATA SMOOTHING
C     COMPUT WEIGHTS
      M1=MLAG
      M=M1-IABS(IS)
      M2=M-1
      NF=2*M+1
      DO 10 I=1,M
   10 W(I)=0.5*(1.+COS(PI*FLOAT(I)/FLOAT(M)))
C     CALCULATE A SMOOTHED AUTOSPECTRAL ESTIMATE
      MM=M-1
      DO 20 I=1,NF
      SUM=0.0
      DO 21 K=1,MM
      ARG=PI*FLOAT(K)*FLOAT(I-1)/FLOAT(NF-1)
      VAR=COV(K+1)*W(K)*COS(ARG)
   21 SUM=SUM+VAR
   20 SPEC(I)=2.*DELT*(COV(1)+2.*SUM)
C     COMPUTE BANDWIDTH AND DEGREES OF FREEDOM
      RETURN
      END
```